

Стефані Окерман, Саймон Рейндл

Опанування професійного Scrum

**Практичний посібник
із подолання труднощів
і максимізації
переваг Agile-підходу**

ВИДАВНИЧИЙ ДІМ
ФАБУЛА
#PRO

Авторизований переклад з англomовного видання під назвою
Mastering Professional Scrum: A Practitioner's Guide to Overcoming Challenges
and Maximizing the Benefits of Agility, 1-ше видання, авторів Стефані Окерман
і Саймона Рейндла, опубліковано Pearson Education, Inc, що виступає
як Addison Wesley Professional, Copyright © 2020 Pearson Education, Inc.

Authorized translation from the English language edition, entitled
Mastering Professional Scrum: A Practitioner's Guide to Overcoming Challenges
and Maximizing the Benefits of Agility, 1st Edition by Stephanie Ockerman; Simon Reindl,
published by Pearson Education, Inc, publishing as Addison Wesley Professional,
Copyright © 2020 Pearson Education, Inc.

*Усі права збережено. Жодної частини цієї книжки не може бути відтворено
або передано в будь-якій формі або будь-якими засобами, електронними
чи механічними, включно з фотокопією, записом чи будь-якою системою
зберігання та пошуку інформації, без письмового дозволу Pearson Education, Inc.*

*All rights reserved. No part of this book may be reproduced or transmitted in any form
or by any means, electronic or mechanical, including photocopying, recording
or by any information storage retrieval system, without permission from Pearson Education, Inc.*

Видання українською мовою опубліковано ТОВ Видавничий дім «Фабула» © 2023
Ukrainian language edition published by Publishing House Fabula LLC, Copyright © 2023

Окерман Стефані

О50 Опанування професійного Scrum: практичний посібник із по-
долання труднощів і максимізації переваг Agile-підходу / Стефані
Окерман, Саймон Рейндл / пер. з англ. В. Луненко. — Харків :
ВД «Фабула», 2023. — 224 с.

ISBN 978-617-522-087-0

«Опанування професійного Scrum» — це книжка для тих, хто хоче
поставляти підвищену цінність завдяки більш ефективному використан-
ню Scrum. Провідні практики Scrum Стефані Окерман та Саймона Рейндл
спираються на роки навчання та тренінгу Scrum, щоби допомогти чи-
тачам повернутися до первинних принципів та застосовувати Scrum
із професіоналізмом, необхідним для досягнення його трансформацій-
ного потенціалу.

Автори мають на меті допомогти зосередитися на перевірених під-
ходах Scrum для поліпшення якості, отримання та використання швид-
кого зворотного зв'язку, а також стати більш пристосованим, замість
«виконувати дії» та задовольнятися лише скромними вдосконаленнями.

УДК 338.24:65.01

Copyright © 2020 Stephanie Ockerman
and Simon Reindl

© В. Луненко, переклад з англ., 2023

© ВД «Фабула», макет, 2023

ISBN 978-617-522-087-0

*Людам, які приходять щодня і роблять важку ро-
боту, щоби створити більш інклюзивні, добрі та
стійкі організації, громади та суспільства. І всім жін-
кам та союзникам, які надали мені сили, надихнули
й дали змогу здійснити мою власну подорож.*

Стефані

*Мойй чудовій родині, без терпіння якої це було б не-
можливим. Мойй дивовижній і люблячій дружині Сарі,
що підтримувала мене, і двом чудовим дітям, Еллі
та Джорджу. Дякую вам за радість, і сміх, і свіжий
погляд, який ви приносите в мій світ.*

Симон

Зміст

Передмова Кена Швабера	9
Передмова Дейва Веста	13
Вступ	17
Scrum — це шлях уперед, якщо прагнути до нього професійно . . .	18
Кому слід читати цю книжку	18
Як упорядковано цю книжку	19
Заклик до дії	20
Слова подяки	23
Розділ 1. Постійне вдосконалення вашої практичної діяльності в Scrum	25
Зосередьтеся на семи ключових напрямках для вдосконалення практичної діяльності в Scrum	25
Гнучке (agile) мислення	26
Емпіризм лежить в основі Scrum.	26
Опанування Scrum означає поліпшення командної роботи	28
Кожна Scrum-команда повинна зосередитися на поліпшенні цінності свого продукту	29
Кожна сильна команда має чітку командну ідентичність.	29
Команди повинні відшліфувати свої командні процеси з метою вдосконалення	30
Організація може значно вплинути на результати роботи команди	31
Зростання Scrum вимагає від команди вдосконалення інших умінь	31
Викладацькі навички	32
Навички фасилітації	33
Навички навчання	34
Технічна досконалість	34
Лідерство як служіння	35
Процес безперервного вдосконалення	36
Що дошкуляє найбільше?	37
Аналіз першопричини	39
Експериментуємо з різними підходами	43
Успіх чи невдача?	44
Підсумки	46
Заклик до дії	46

Розділ 2. Створення сильної команди	47
Формування командної ідентичності	47
Що створює хорошого члена команди?	48
Хто має бути в Scrum-команді?	51
Команди розробників повинні знати більше, ніж просто розробка	52
Як Scrum-команди створюють робочі угоди?	53
Який вигляд має самоорганізація?	55
Спільні цілі	57
Чітка відповідальність	57
Межі	58
Як співпрацюють Scrum-команди?	60
Як просуваються команди?	66
Характеристики продуктивних і адаптивних команд.	71
Підсумки	73
Заклик до дії	73
Розділ 3. Постачання «готових» приростів цінності продукту	75
Що означає «повна готовність завдання»?	76
Переваги критерію «повної готовності завдання»	77
Як створити критерій «повної готовності завдання»	78
Використання цілей спринту для досягнення «повної готовності завдання»	81
Створення гарних цілей спринту	82
Використання цілі спринту для ефективної щоденної Scrum-наради	84
Приведення елементів беклогу продукту до «готового» стану раніше в межах спринту.	85
Обмеження кількості незавершених завдань	89
Вимірювання та аналіз потоку виконання	90
Нарощування якості від самого початку	92
Автоматизація та «повна готовність завдання»	94
DevOps.	96
Перегляд коду.	96
Метрики якості.	97
Пропрацювання технічного боргу	98
Надання прозорості технічному боргу	100
Забезпечення помітності «погашення» технічного боргу	102
Підсумки	103
Заклик до дії	103

Розділ 4. Поліпшення наданої цінності	105
Що таке цінність?	105
Швидше постачання — хороший початок, але цього недостатньо	106
Цінність продукту і Scrum-команда	108
Використання концепції продукту для підсилення командної мети, зосередженості та ідентичності	109
Вимірювання цінності	111
Орієнтація запланованих для виконання елементів продукту (беклогу) на результати користувачів	113
Поліпшення наданої цінності під час спринту	119
Перевірка та пристосування на основі відгуків	120
Навчання як цінність	120
Ефективні огляди спринту містять реалізовану цінність	121
Збирання відгуків від зацікавлених сторін	121
Підсумки	123
Заклик до дії	123
Розділ 5. Поліпшення планування	125
Планування за допомогою продуктового мислення	126
Вимірювання успіху	127
Емпіричне планування	128
Створення узгодженої структури	130
Уточнення беклогу продукту	131
Удосконалення запланованих для виконання задач (беклогу) мінімально життєздатного продукту	132
Оцінка	134
Розбиття запланованих для виконання елементів продукту (беклогу) для зосередження на результатах	137
Планування спринту	137
Скільки «готового» приросту цінності ви можете отримати за спринт?	138
Скільки часу слід витратити на вдосконалення цього спринту?	142
Наскільки далеко ви від удосконалення	142
Планування релізів	143
Наскільки великим повинен бути реліз?	144
Наскільки малим за обсягом може бути реліз?	144
Підсумки	145
Заклик до дії	145

Розділ 6. Допомога Scrum-командам у розвитку та вдосконаленні	147
Використання ретроспективи спринту для виявлення сфер для вдосконалення	147
Виявлення та усунення перешкод	151
Відстеження перешкод та кількісна оцінка наслідків	153
Усунення перешкод	156
Зростання індивідуальних та командних можливостей	158
Знайдіть час для постійного навчання та зростання	158
Використовуйте знання та досвід в організації	160
Бути відповідальним Scrum-майстром	162
Вимірювання успіху Scrum-майстра	163
Ефективні Scrum-майстри змінюють свій підхід на основі контексту	166
Підсумки	171
Заклик до дії	171
Розділ 7. Ефективне залучення організації до вдосконалення	173
Організаціям необхідно розвиватися, аби досягти успіху	173
Розвиток людей і команд	174
Вплив оглядів продуктивності та компенсації	175
Індивідуальні кар'єрні шляхи	176
Стратегії пошуку ресурсів і вплив на команди	177
Розподілені команди	179
Почуття зручності в стані прозорості	180
Культура відповідальності за результат, а не звинувачення	181
Відмова від (ілюзії) контролю	182
Справжня сила залізного трикутника	183
Фінансові ініціативи	185
Оцінка на основі обсягу	185
Ітеративне та інкрементальне бюджетування	186
«Бути Agile» — це не мета	189
Зробіть щось добре, перш ніж збільшувати використання	190
Підсумки	192
Заклик до дії	192
Розділ 8. Висновки й подальші перспективи	193
Гнучкість бізнесу вимагає нових рішень	193
Заклик до дії	196

Додаток А. Самооцінка для розуміння того, де ви перебуваєте	197
Бізнес-гнучкість	197
Ефективний емпіризм зі Scrum	198
Ефективна робота в команді зі Scrum	203
Аналіз оціночних відповідей	204
Додаток Б. Поширені хибні думки щодо Scrum	205
Scrum не є методологією або процесом управління	205
Scrum — це не «панацея» чи спосіб пришвидшити розробників	206
Власник продукту не документує вимоги	207
Беклог продукту не є Agile-версією традиційного документа з вимогами	207
Беклог продукту не є списком усіх запитів	208
Щоденна Scrum-нарада не є статусною	208
Спринт може бути успішним, навіть якщо всі заплановані елементи беклогу спринту не завершено	208
Scrum-майстер не відповідає за відстеження роботи команди розробників	209
Огляд спринту — це не зустріч заради схвалення	209
Для початку спринту не потрібно багато готуватися	209
Предметний покажчик	211

Передмова Кена Швабера

Я створив Scrum для поліпшення способу розробки програмного забезпечення. Упродовж останніх 27 років його вдосконалювали здебільшого завдяки створенню, публікації та акуратному поліпшенню «Посібника зі Scrum» («Scrum Guide»). Ми з Джеффом Сазерлендом (співавтором Scrum) опублікували його в інтернеті, як зазначено в посібнику, щоби люди могли пропонувати вдосконалення. Протягом багатьох років ми вдосконалювали Scrum на основі цих коментарів, роблячи його простішим у використанні та розумінні.

Коли я вперше застосував термін «Scrum-майстер», багато людей розгубилися. Ніхто не опанував Scrum; ми всі вчилися, як ним користуватися і як нарощувати його за допомогою практик та інструментів, щоби поліпшити результати й допомогти членам команди використовувати його з правильними цінностями, методиками, артефактами («готові» природи цінності продукту) та ролями, коли всі працюють разом заради досягнення цілей Scrum.

Робота Scrum-майстра полягає в допомозі організації та Scrum-команді правильно використовувати Scrum заради поліпшення їхньої здатності приносити цінність. Scrum-майстер має допомогти членам команди та людям, які залежать від Scrum (спеціалісти з людських ресурсів, фінансів тощо), зрозуміти, як вони можуть налагодити оптимальну роботу. Будь-хто зі Scrum-команди може вдосконалити свою *майстерність* роботи зі Scrum; вони можуть стати кращими у використанні Scrum та емпіризму для досягнення ліпших результатів і більшої цінності в складних сферах. Будь-хто може стати *професійнішим*.

Професіонал — це той, хто працює за гроші й дотримується правил, установлених для цієї професії. Професіонали діють і пра-

цюють відповідно до стандартів, якщо такі існують (наприклад, дотримуючись правил, викладених у «Посібнику зі Scrum»). Вони також приймають і втілюють набір етичних принципів, установлених їхньою професією (наприклад, цінності Scrum: зосередженість, обов'язок, відкритість, повага та сміливість).

Іноді Scrum-професіонал починає розриватися між двома альтернативами. За таких обставин Agile-маніфест забезпечує вказівками вищого рівня.

- Люди та співпраця важливіші за процеси та інструменти.
- Продукт, що працює, важливіший за вичерпну документацію.
- Співпраця із замовником важливіша за обговорення умов контракту.
- Готовність до змін важливіша за дотримання плану.

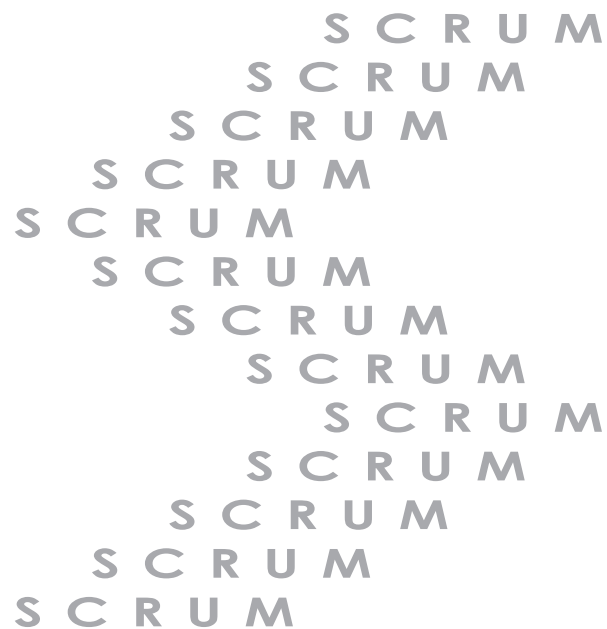
Scrum-професіонали не переосмислюють і не «припасовують» його до своїх організацій. Scrum є Scrum. Щоби створити приріст цінності продукту та досягти бажаних результатів, вони *справді* використовують допоміжні та додаткові методології, зокрема DevOps, Kanban, а також тестування, урегулювання питань та комунікаційні практики. Методики, які відрізняють Scrum від інших підходів до складної роботи, містять низку тверджень.

1. Команда організовує роботу в *коротких циклах*.
2. Керівництво *не втручається* в роботу команди під час робочого циклу.
3. Команда звітує *перед клієнтом*, а не перед менеджером.
4. Команда оцінює, *скільки часу та зусиль* забере ця робота.
5. Команда вирішує, *скільки роботи* вона може виконати за ітерацію.
6. Команда вирішує, *як виконувати* роботу в ітерації.
7. Команда *вимірює власну продуктивність*.
8. Команда визначає цілі роботи *перед* початком кожного циклу.
9. Команда визначає роботу та бажані результати за допомогою поступово уточнювального опису результатів (так званого списку задач (беклогу) продукту).
10. Команда працює над систематичним та постійним удосконаленням та усуненням перешкод.

Наша робота як Scrum-професіоналів полягає в постійному вдосконаленні наших умінь використовувати Scrum для постачання продуктів і послуг, які допомагають клієнтам досягти значних результатів. Ця книжка надасть вам змогу поліпшити свої вміння в застосуванні Scrum. Її автори діляться власним досвідом та порадами, зібраними завдяки допомозі багатьом клієнтам та студентам у вивченні та застосуванні Scrum у своїх організаціях. Сподіваюся, це допоможе вам у вашій професійній подорожі.

Продовжуйте скрамити!

Кен Швабер, співтворець Scrum і власник Scrum.org



Передмова Дейва Веста

Що таке професійний Scrum?

Немає сумніву, що світ праці стає дедалі складнішим. Річ не в тім, що проста робота відходить у небуття, а в тому, що більшу частину цієї роботи замінять автоматизація, алгоритми та робототехніка. Складну роботу найкраще визначити як невідому не лише з огляду на те, як ми це робимо, але й з погляду її результатів та наслідків. Навіть коли маємо на увазі чіткий результат, ми розуміємо, що вплив змін може відрізнитися від раніше передбаченого лише після того, як щось зробили.

Scrum було розроблено, щоби допомогти нам прокласти шлях у складному світі. Проста, але потужна бібліотека (наборів інструкцій, або фреймворк), яка дозволяє впорядкувати та структурувати складне шляхом відкриттів та навчання. Але для ефективності роботи зі Scrum потрібно щось більше, ніж просто слідування принципу дії фреймворку: вона вимагає професійного ставлення.

Кен Швабер, співторець Scrum, описує *професіонала* як того, хто працює за гроші й дотримується встановлених правил у професії. Він також додає, що бути професіоналом означає дотримуватися набору етичних стандартів. Ці стандарти не лише уніфікують представників певної професії, а й визначають її для зовнішнього світу, як це робить клятва Гіппократа для медичної професії.

Спираючись на цю модель професіоналізму, можна визначити чотири додаткові елементи, які є ключовими для його досягнення з використанням Scrum.

- **Дисципліна.** Щоби бути ефективним у використанні Scrum, потрібна дисципліна. Ви повинні здійснити постачання продукту, щоби навчитися; ви мусите виконувати механіку Scrum;

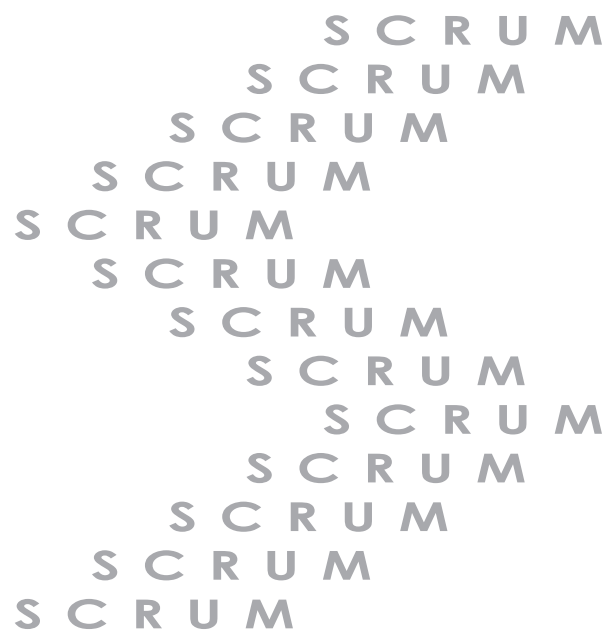
ви маєте кинути виклик своїм заздалегідь продуманим уявленням про власні вміння, роль та розуміння проблеми; і вам треба працювати прозоро та структуровано. Дотримуватися дисципліни важко, і часом вона може здатися несправедливою, оскільки робота ставить проблему за проблемою, а ваші зусилля здаються марними.

- **Поведінка.** Цінності Scrum було представлено в «Посібнику зі Scrum» 2016 року у відповідь на необхідність підтримки культури, яка б дозволила Scrum бути успішним. Цінності Scrum описують п'ять простих ідей, котрі на практиці заохочують до становлення agile-культури (сміливість, зосередженість, обов'язок, повага та відкритість), описують поведінку, що її повинні демонструвати як Scrum-команди, так і організації, у яких вони працюють.
- **Цінність.** Scrum-команди працюють над проблемами, розв'язання яких приносить користь споживачам та зацікавленим сторонам. Команди працюють на клієнта, який винагороджує їх за цю роботу. Але відносини складні, оскільки складними є проблеми: клієнти можуть не знати, чого вони хочуть, економіка рішення може бути незрозумілою або якість та безпека рішення невідомими. Робота професійної Scrum-команди полягає в тому, щоб, наскільки це можливо, зробити правильно для всіх цих сторін, запропонувавши рішення, котре найкраще відповідає потребам клієнтів у межах накладених обмежень. Для цього потрібні прозорість, повага одне до одного та до клієнтів і здорова доза цікавості, щоби розкрити правду.
- **Допомога іншим.** Scrum — це командний вид спорту, але такий, де кожна команда невелика. Як наслідок, команда часто є аутсайдером у намаганні розв'язати проблеми, для подолання яких вона заледве має навички та досвід. Щоби бути ефективними, професійні Scrum-команди повинні працювати з іншими членами своєї спільноти, аби опановувати нові навички та ділитися досвідом. Допомога у зміцненні гнучкості спільноти не є суцільним альтруїзмом, оскільки помічники часто отримують цінні для себе знання, які вони можуть застосовувати для допомоги власним командам. Професійний Scrum закликає людей створювати професійні мережі, у яких можна обмінюватися помічними для команд ідеями та досвідом.

Простий опис професійного Scrum мало чим допомагає вам у реалізації цієї ідеї у вашій організації чи продукті. Ось де на допомогу й приходить книжка Стефані та Саймона. Вона є підтримкою для професійного Scrum. Книжка бере фреймворк Scrum і ставить його в контекст професіоналізму, описуючи, чому багато з цих ідей мають сенс і як вони розвинулися з різних дисциплін та концепцій. Незалежно від того, читаете ви її повністю або поринаєте в глибини певного розділу в пошуках конкретних указівок, вона надає практичні поради щодо того, як можна освоїти Scrum та стати професійнішим. І ця подорож довга й ніколи не закінчується.

Щастя вам і приємної подорожі.

Дейв Вест, генеральний директор і власник Scrum.org



Вступ

Ми живемо у світі, де впевненим можна бути лише в невизначеності. Оскільки світ стає більш взаємопов'язаним і взаємозалежним, він також стає більш складним. І він швидко змінюється: нові клієнти та конкуренти можуть з'являтися, розвиватися й зникати, перш ніж ви отримаєте можливість відповісти. Технології постійно змінюються, нові політичні реалії можуть створити нові нормативно-правові вимоги, на які доведеться зважати, та й зловмисні хакери, здається, навчаються швидше, ніж розвивається наша здатність ставати їм на заваді.

У відповідь на цю невизначеність ми приймаємо факт немоги передбачити майбутнє. Найкраще, що ми можемо вчинити, це діяти з наміром, роблячи невеликі кроки вперед, приймаючи невизначеність та емпіризм і використовуючи цикли зворотного зв'язку для навчання. У цьому суть гнучкості (*agility*) й основа Scrum: планування з невеликим кроком (приростом), надання робочого просту продукту, перевірка результату та адаптація, багаторазово та повністю прозоро. Але щоби гнучкість працювала, її треба прагнути професійно.

Докази відсутності професіоналізму скрізь: від отриманого замовлення з неправильними товарами до телефонного застосунку, який не працює, та повідомлень про порушення безпеки, що розкривають нашу приватну інформацію стороннім особам. Це також проявляється в проектах, які виходять з-під контролю, коштуючи мільйони доларів і не приносячи жодної цінності. Вони проявляються й на особистому рівні: цінні робочі роки витрачаються без розвитку нових навичок або відкриття нових можливостей. Це підриває довіру та шкодить робочим відносинам. Кожен, хто працював над розробкою продукту, відчував принаймні деякі з його симптомів.

- Відсутність прозорості щодо прогресу, якості та результатів.
- Обіцянки помилкової впевненості та уникання відкритих та чесних розмов про складність і ризик.
- Жертвування якістю заради економії грошей або часу.
- Уникання підзвітності.
- Постачання продуктів, які не досягають прийнятної якості, щоби вкластися в терміни постачання.
- Ігнорування нової інформації та виконання плану.

Scrum — це шлях уперед, якщо прагнути до нього професійно

Scrum — це емпіричний підхід до постачання продуктів у складному та невизначеному світі. Хоча Scrum доволі розповсюджений, здебільшого він не надто професійний. Як зазначає співзасновник Scrum Кен Швабер, «Scrum простий для розуміння, але складний для освоєння»¹. З-поміж різних команд та організацій, які справді практикують Scrum, багато хто просто проходить етапи й здійснює рухи. Ми називаємо це «механічний Scrum». Такі команди використовують Scrum-термінологію, не розуміючи намірів, що за нею криються, або не демонструючи дисципліни, якої вимагає Scrum.

Ця книжка має на меті розвіяти міфи, виправити непорозуміння та допомогти організаціям використовувати Scrum для постачання високоякісних продуктів і досвіду для своїх клієнтів. Якщо коротко, то вона спрямована на допомогу організаціям у застосуванні Scrum у процесі набуття професіоналізму.

Кому слід читати цю книжку

Цю книжку призначено для людей, які мають практичні навички роботи зі Scrum і, можливо, багато чого роблять правильно, але хочуть удосконалитися. Ви можете бути Scrum-майстром або ж членом команди розробників чи власником продукту. Найголовніше,

¹ www.Scrumguides.comBOOK.

що ви хочете і маєте потребу вдосконалюватися. Якщо ви бажаєте дізнатися про Scrum, ми пропонуємо вам розпочати з «Посібника зі Scrum», занять, присвячених Scrum, або однієї з багатьох чудових ознайомчих книжок із цього питання¹.

Як упорядковано цю книжку

Пишучи цю книжку, ми мали намір надати вам віртуального Scrum-тренера для допомоги у вашій подорожі, розв'язанні проблем із прозорістю та сміливістю, а також заради ознайомлення вас із новими підходами, які допоможуть оволодіти Scrum, проявити професіоналізм та забезпечити гнучкість бізнесу. У нас немає всіх відповідей, але ми надамо інструменти, за допомогою яких ви зможете знайти власні відповіді на свої унікальні виклики.

Щоби спростити подорож до опанування Scrum, ми розробили підхід для професійного Scrum, який синтезує те, про що ми дізналися, і надає компас, котрий допоможе вам зорієнтуватися під час власної подорожі. Цей підхід базується на нашому досвіді як практиків, професійних Scrum-тренерів і знаннях, які ми отримали з ширшої спільноти Scrum.org. Вам вирішувати, де розпочати свою подорож.

Кожен розділ присвячений певному набору проблем.

- **Розділ 1 — Постійне вдосконалення вашої практичної діяльності в Scrum** — надає спосіб оцінки й узагальнення вашої поточної діяльності, підкріплений самооцінкою в Додатку А з прицілом на визначення напрямів для вдосконалення.
- **Розділ 2 — Створення сильної команди** — допомагає зрозуміти, як ваша команда працює разом, визначити напрями вдосконалення та використати відповідні методи для поліпшення складу та способу роботи команди.
- **Розділ 3 — Постачання «готових» приростів цінності продукту** — пояснює, чому «готовий» є найважливішим понят-

¹ Для пошуку курсу див. <https://www.Scrum.org/courses>. Якщо ви шукаєте стислу книжку про Scrum, рекомендуємо Scrum: Pocket Guide від Gunther Verheyen (Van Haren Publishing, 2013).

тям у Scrum і чому «неготова» робота є ознакою того, що щось іде відверто не так і потребує виправлення.

- **Розділ 4 — Поліпшення наданої цінності** — зосереджується на вимірюванні цінності, поставленої завдяки вашому «готовому» приросту цінності продукту, і забезпечує методики для поліпшення цінності, яку ви поставляєте з плином часу.
- **Розділ 5 — Поліпшення планування** — допомагає вам поліпшити спосіб прийняття рішення щодо того, яку роботу ви збираєтесь виконувати, і зосередитися на забезпеченні високоцінних приростів продуктів, водночас усуваючи «неготову» роботу.
- **Розділ 6 — Допомога Scrum-командам у розвитку та вдосконаленні** — розкриває перешкоди, з якими може стикнутися команда під час постачання «готових» приростів цінності продукту, та пропонує стратегії їх подолання.
- **Розділ 7 — Ефективне залучення організації до вдосконалення** — розглядає, як організаційні перешкоди можуть обмежити здатність команди виконувати завдання, та звертає увагу на те, що ви можете зробити задля подолання цих перешкод.
- **Розділ 8 — Висновки й подальші перспективи** — дає огляд подорожі, яку ви здійснили в цій книжці, та пропонує кілька способів її продовження.
- **Додаток А — Самооцінка для розуміння того, де ви перебуваєте**, — забезпечує засіб оцінки вашої поточної практичної діяльності в Scrum.
- **Додаток Б — Поширені хибні думки щодо Scrum** — описує та виправляє типові непорозуміння стосовно того, що вважається Scrum, а що ні.

Заклик до дії

У процесі просування вперед ми рекомендуємо розпочати потужні та продуктивні бесіди, які допоможуть вам отримати максимум користі від цієї книжки. По-перше, вам слід подумати про те, у якій точці ви зараз перебуваєте й куди хочете йти.

Визначаючи спільні цілі, не забувайте цікавитися різними поглядами.

Наведемо кілька запитань для початку.

1. Що означає бізнес-гнучкість для нашої організації? Як це пов'язано з нашою місією? Які переваги ми очікуємо бачити як організація? Який вигляд це буде мати, коли ми реалізуємо наше бачення гнучкості? Як це буде по-іншому?
2. Що означає бізнес-гнучкість для нашої команди? Які переваги ми очікуємо побачити? Які дані ми можемо використати для розуміння гнучкості на командному та продуктовому рівнях?
3. Як бізнес-гнучкість співвідноситься з баченням продукту?
4. Як швидко ми можемо отримати рентабельність інвестицій (ROI)? Чого ще ми хочемо?
5. Скільки гнучкості та контролю має наш бізнес під час прийняття інвестиційних рішень? Що ще потрібно?
6. Як швидко ми можемо скористатися можливостями та зреагувати на ризики? Чого ще ми хочемо?
7. Як ми демонструємо професіоналізм у команді? Як цінності та поведінка організації співвідносяться з професіоналізмом?
8. Які приклади непрофесійної поведінки ми бачили (чи брали в них участь) у нашій організації?

SCRUM
SCRUM
SCRUM
SCRUM
SCRUM
SCRUM
SCRUM
SCRUM
SCRUM
SCRUM
SCRUM
SCRUM

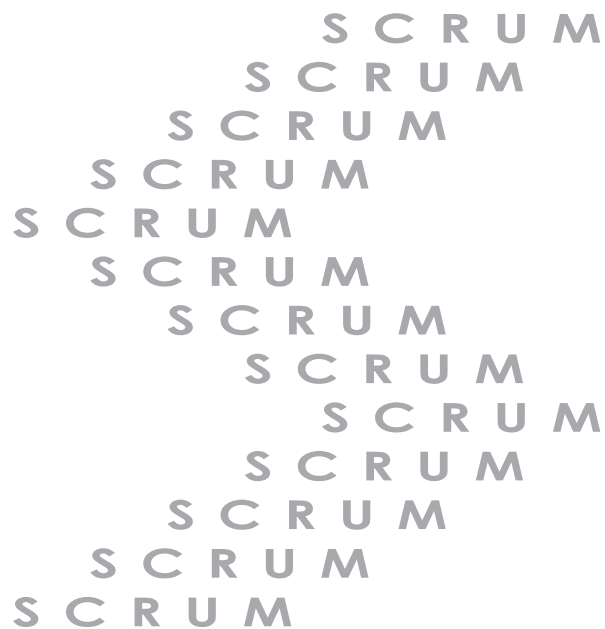
Слова подяки

Під час написання цієї книжки ми отримали велику допомогу й підтримку. По-перше, ми повинні подякувати Дейву Весту, Курту Бітнеру та Кену Шваберу за їхню довіру, підтримку, заохочення та об'єктивний погляд на завдання, яке здавалося заледве здійсненним,— написання книжки, яка виявляє силу Scrum, надає практичні поради та надихає людей на відкриття власного шляху до Scrum-майстерності. Курт Бітнер здійснив магію, допомігши нам висловити наші ідеї простіше та ефективніше. Звичайно, ми повинні подякувати Кену Шваберу та Джеффу Сазерленду за те, що вони створили сам Scrum, який дав нам шляхи для виконання роботи, котра відповідає нашим цінностям і цілям.

Ми дуже вдячні спільноті професійних тренерів Scrum, члени якої підтримували нас у нашому зростанні як спеціалісти з постачання продуктів, професійні тренери Scrum та підприємці. Вдячні, що ми є частиною цієї спільноти, за їхню щедрість в обміні знаннями та досвідом, відданість навчанню й зростанню та бажання вповні себе проявити.

Багато хто надихав і критикував нас на нашому шляху до опанування Scrum. Наша найбільша вдячність Тодду Гріну, Річарду Гундгаузену, Стейсі Мартін, Дону МакГрілу, Стіву Портеру, Раяну Ріплі, Стіву Траппсу та багатьом іншим.

Саймон і Стефані



Розділ 1

Постійне вдосконалення вашої практичної діяльності в Scrum

Scrum — це простий набір правил і технік, який допомагає командам часто створювати цінні продукти, що виходять у реліз. Правила, які існують для практики Scrum, важливі для гарантування прозорості, уможливлення ефективного контролю та адаптації, економії та забезпечення гнучкості бізнесу¹.

Незалежно від рівня досвідченості, кожна команда може вдосконалити свою здатність перевіряти та пристосовуватися, щоби забезпечити приріст цінності продукту. Клієнти постійно розвиваються, і їхні потреби постійно змінюються. Конкуренти постійно розвиваються та адаптуються. Так само й технології постійно змінюються, відкриваючи нові можливості й водночас створюючи нові виклики. Нові члени команди приносять нові навички та погляди, але можуть змінити динаміку команди. Подолання цих викликів означає не лише опанування процесу постачання чудових продуктів за допомогою емпіризму, а й перевірку, адаптацію та вдосконалення можливостей Scrum-команди.

Зосередьтеся на семи ключових напрямках для вдосконалення практичної діяльності в Scrum

Щоби допомогти вдосконалитися вам та вашим командам, ми розбили проблему на сім ключових кроків.

¹ Щоби дізнатись більше про Scrum, див. <https://www.Scrum.org/resources/what-is-Scrum>.

- Гнучке (*agile*) мислення
- Емпіризм
- Командна робота
- Командний процес
- Командна ідентичність
- Цінність продукту
- Організація

Гнучке (*agile*) мислення

Гнучкий спосіб мислення має важливе значення для поліпшення ставлення та світогляду членів Scrum-команди, формуючи спосіб, у який вони інтерпретують світ і працюють між собою та світом в цілому. Коли ми говоримо про гнучкий спосіб мислення, то додаємо сюди й цінності Scrum¹, цінності та принципи з Agile-маніфесту про гнучку розробку програмного забезпечення², принципи Lean³. Ці цінності та принципи керують рішеннями, які ухвалює команда Scrum, і вони безпосередньо впливають на ефективність цієї команди у співпраці, використовуючи емпіричний процес для отримання приросту цінності продукту.

Постачання цінності в складному світі означає, що існує декілька правил і немає «найкращих практик», які команда може застосувати. Натомість, члени команди керуються гнучким мисленням, щоби знаходити розв'язання на підставі найкращих доступних для них даних.

Емпіризм лежить в основі Scrum

Scrum створено для забезпечення емпіризму. Прийняття емпіризму поліпшує прозорість, перевірку й адаптацію. Розуміння цих трьох підвалин будь-якого емпіричного процесу є надзвичайно важливим для поліпшення здатності Scrum-команди поставляти прирости цінності продукту.

¹ www.Scrumguides.org

² agilemanifesto.org

³ Див. книжку Мєрі Поппендік та Тома Поппендіка «Lean Software Development: An Agile Toolkit» (Boston: Addison-Wesley, 2003).

- **Прозорість** означає, що Scrum-команда цілком розуміє ситуацію; вони бачать усі аспекти процесу, які впливають на результати. Прозорість допомагає зрозуміти, які характерні особливості та функції заплановані для продукту, як Scrum-команда просувається до своїх цілей та яку цінність отримують клієнти, коли користуються продуктом.
- **Перевірка** означає, що Scrum-команда здатна за невеликі часові проміжки спостерігати за результатами та вчитися на новій інформації. Члени команди активно шукають інформацію як про досягнення, так і про недоліки бажаних результатів та цілей.
- **Адаптація** означає, що Scrum-команда за невеликі часові проміжки використовує інформацію, отриману в результаті перевірки, для зміни своїх стратегії, планів, методів та поведінки, щоб узгодити їх із бажаними результатами й цілями.

Фреймворк Scrum пропонує набір легких правил, які допомагають Scrum-команді досягти мінімального рівня емпіризму.

- Часові рамки (таймбокси) допомагають Scrum-команді створювати емпіричні цикли зворотного зв'язку.
- Виробляючи **приріст цінності продукту** принаймні раз під час спринту, Scrum-команда забезпечує прозорість, яка дозволяє їй підтвердити свої припущення щодо цінності.

Щоби по-справжньому максимізувати переваги Scrum, Scrum-команди повинні розширити діапазон (кількість) та збільшити глибину (якість) свого емпіризму. Наприклад:

- підвищуючи прозорість виконання своєї роботи, вони можуть виявити поліпшення у своїх процесах, інструментах і взаємодіях;
- підвищуючи прозорість цінності, яку споживачі усвідомлюють унаслідок використання продукту, вони отримують глибше розуміння того, як можна його вдосконалити;
- збільшуючи частоту співпраці впродовж дня, окрім щоденної Scrum-наради, вони можуть швидше виявляти та розв'язувати проблеми, відтак поліпшуючи процес здійснення діяльності;

- співпрацюючи з власником продукту в процесі роботи, вони можуть збільшити швидкість, з якою будуть здатні вдосконалити продукт.

Опанування Scrum означає поліпшення командної роботи

Щоби змусити емпіризм працювати, Scrum-команди мають співпрацювати заради пошуку цінних розв'язань складних проблем, тоді виміряти результати й згодом пристосуватися на основі зворотного зв'язку. Ефективній Scrum-команді можна дати низку характеристик.

- *Кросфункціональна.* Кросфункціональна команда має всі навички, необхідні для досягнення мети. Це зменшує ризик, спричинений залежностями, які існують поза командою, зокрема потенційні втрати від частково виконаної роботи. Кросфункціональність не означає, що кожна людина повинна вміти виконувати будь-яку роботу. Натомість команда мусить виявити поєднання навичок та способи їх розподілу серед своїх членів, щоби зменшити втрати, поліпшити інновації та якість і пристосуватися до мінливих потреб.
- *Самоорганізована.* Команда, яка самоорганізовується, визначає, чого вона може досягти і як її члени працюватимуть разом, щоби це відбулося. Перший крок для забезпечення підзвітності полягає в тому, щоби команда відчувала відповідальність за свою роботу. Її членам необхідно довіряти як експертам і дозволяти експериментувати, випробовувати нові речі та змінювати напрям — і все це заради забезпечення цінності.
- *Здатна на співробітництво.* Щоби використати силу колективного інтелекту, самоорганізована, кросфункціональна команда повинна подолати відособленість заради отримання переваг від співпраці. Відособлена робота створює перепони для інновацій або навіть для швидкого постачання чогось важливого для замовника. Передача даних створює прогалини в розумінні, затримки та інші втрати.
- *Стабільна.* Самоорганізована, кросфункціональна, здатна на співробітництво команда — це більше, ніж сукупність працівників; це абсолютно нове утворення, що складається з людей, які самі є дивовижно складними істотами. Необхідні свідомі зусилля і час, щоб об'єднати групу людей у згуртовану

команду, яка може постійно розвиватися з погляду того, чим вона є і як працює. Без стабільності команда ніколи не сформується цілком, а організація, що її спонсорує, ніколи не зможе по-справжньому користатися перевагами високоефективної команди. Це не означає, що склад команди ніколи не повинен змінюватися, а тоді, коли це станеться, знадобиться час і свідомі зусилля, щоби допомогти людям знову працювати в команді.

Кожна Scrum-команда повинна зосередитися на поліпшенні цінності свого продукту

Мета Scrum-команди — поставити цілу низку цінних продуктів. Щоби забезпечити цінність, Scrum-команда повинна:

- зрозуміти мотивацію, поведінку та потреби (як явні, так і приховані) користувачів і клієнтів;
- узгодити бачення товару, його стратегію, місію та цілі організації;
- виміряти фактичну доставлену цінність.

По суті, Scrum дозволяє команді часто постачати багато речей. Однак, якщо команда не оптимізує цінність Продукту, вона досягне дуже мало.

Кожна сильна команда має чітку командну ідентичність

Команда починається як добірка людей. Разом вони утворюють цілком новий живий і дієвий організм. Цей новий організм із часом формує ідентичність. Подібно до того, як дитина зростає і стає підлітком, а потім юнаком, команда повинна постійно прагнути виявляти та розвивати свою особистість.

На фундаментальному рівні встановлення ідентичності полягає у відповіді на три наведені нижче суттєві запитання, які спрямовують команду на шлях до високої продуктивності.

1. Чому ми існуємо? (Мета)
2. Що для нас важливо? (Значення)
3. Чого ми хочемо? (Бачення)

Команди повинні відшліфувати свої командні процеси з метою вдосконалення

Scrum-команда визначає свій спосіб роботи в межах захисних зон, установлених фреймворком Scrum, тобто меж і вказівок, зумовлених рольовою підзвітністю, цілями подій та артефактів. Як саме Scrum-команда виконує ролі, використовує артефакти та проводить події, залишається на її розсуд. Те, як вона створює приріст цінності продукту та забезпечує якість, також залежить від неї.

Вимір командного процесу містить практики, інструменти та способи спільної роботи. Він охоплює широкий спектр сфер.

- Інженерні методики та інструменти.
- Методики та інструменти забезпечення якості.
- Методики та інструменти управління продуктами.
- Методики та інструменти управління беклогом продукту.
- Ефективне використання подій та артефактів Scrum.
- Ефективне спілкування та співпраця.
- Ідентифікація та усунення джерел збитків.
- Виявлення та усунення перешкод.
- Ефективне використання й зростання знань, навичок та можливостей команди.

На практики та інструменти, які використовує команда, впливатимуть тип товару, його технологічна платформа, середовище, у якому він використовується, користувачі продукту та спосіб використання, нормативно-правові умови, ринкові тенденції, мінливі потреби бізнесу тощо. Багато чого! Ба більше, значна частина цих речей змінюється з часом.

Як наслідок, Scrum-команди повинні залишатися пильними, пристосовуючи те, що вони роблять, перевіряючи, чому вони це роблять, як вони це роблять та які переваги вони від цього отримують. Нові практики та інструменти постійно створюються та розповсюджуються спільнотами розробників продуктів у всьому світі, тому важливо залишатися на зв'язку та продовжувати навчання. Може статися, що насправді командам доведеться винаходити нові практики та інструменти, щоби відповісти своїм унікальним викликам та потребам.

Організація може значно вплинути на результати роботи команди

Організації забезпечують і структуру, і культуру. Обидва ці аспекти впливають на команди та продукти, що існують в організації, як позитивно, так і негативно.

Структура містить бізнес-модель, яка, по суті, є схемою для успішного ведення бізнесу. Ще вона містить місію, стратегію, продукти та послуги, а також те, як усе це співвідноситься із джерелами доходу, базою споживачів та фінансуванням. Структура також містить спосіб, у який організовано працівників, партнерів та постачальників послуг. Це часто впливає на організаційні процеси та лінію поведінки.

Культура — це сукупність звичок, які пов'язують людей в єдине ціле. Культура — це спосіб бачити ситуацію, знати, що робити за конкретних обставин. Вона розвивається із суми всіх видів поведінки людини в організації. На це часто впливають організаційна структура та процеси, зокрема ролі, цілі та стимули.

На успіх Scrum-команди значний вплив мають організаційна структура та культура.

Максимізація переваг Scrum часто означає розвиток організаційної культури, процесів і, можливо, структури. Хоча вам не доведеться розв'язувати ці справи негайно, зазвичай Scrum-команди з часом починають стикатися з перешкодами, які не здатні контролювати. Вони можуть певний час діяти в обхід цих перешкод, та це означає, що вони вийдуть на плато.

Зростання Scrum вимагає від команди вдосконалення інших умінь

Scrum-командам потрібна низка можливостей, які допоможуть їм постійно вдосконалюватися та адаптуватися до змін. Під «умінням» ми маємо на увазі здатність застосовувати знання, навички та досвід для розв'язання проблем. Зокрема Scrum-команди повинні мати знання (наприклад теорію, методи, домени), здатність уміло застосовувати ці знання для отримання бажаних результатів та досвід для побудови цих навичок, а також керуватися інтуїцією та передбаченням.

Залежно від виду продукту, який вони розробляють, Scrum-команди потребують різних умінь та обмежень організації, що у ній вони працюють. Види необхідних їм умінь можна розділити на п'ять категорій:

- викладацькі навички;
- навички фасилітації;
- навички навчання;
- технічна досконалість;
- лідерство як служіння.

Люди в Scrum-команді повинні мати згадані вміння та продовжувати розвивати їх, щоби бути успішними в тих напрямках, які уможливають появу професійного Scrum.

Викладацькі навички

Викладання — це просвіта інших з метою дати їм нові знання та навички. Часто Scrum-майстри використовують свої навички викладання для допомоги членам команди в розумінні структури Scrum та його основних цінностей і принципів. Scrum-командам, імовірно, доведеться ознайомитися з техніками, які можуть допомогти їм рухатися вперед та ставати ефективнішими зі Scrum.

Навички та знання, необхідні Scrum-команді для постійного вдосконалення та розв'язання нових завдань, з часом змінюватимуться. Scrum-майстри з'ясовують, що потрібно Scrum-команді з огляду на її зростання та поточний контекст, щоби допомогти їй перейти на наступний необхідний рівень. Це може бути професійне навчання, короткі вправи та обмін знаннями, курс підвищення кваліфікації, ситуаційний виховний момент або поєднання всього названого вище.

Звичайно, Scrum-майстру не завжди треба навчати команду. Власники продуктів можуть забезпечувати навчання команд розробників стосовно товарного ринку, потреб клієнта та комерційної цінності. Члени команди розробників можуть навчати одне одного підходам до забезпечення якості, тестування та користування інструментами.

Навчати не означає просто розповідати людям щось; тобто викладання — це не лекція. Люди навчаються набагато ефективніше,

роблячи й відкриваючи. Вони навчаються, пов'язуючи нові знання з тим, що вже знають. Також люди навчаються, коли нові знання й навички здійснюють емоційний вплив.

Викладання — це не те, що до снаги кожному. У декого вроджена здатність навчати, але, зрештою, викладання — це здатність, яку люди можуть розвивати та культивувати. На щастя, вам не потрібно сягати рівня професійного вчителя, щоби застосовувати та розвивати це вміння.

Навички фасилітації

Застосовуючи нейтральні погляди на ситуацію, ведучі спрямовують групи, щоби допомогти їм знайти та ухвалити власні рішення. Фасилітатор надає групі достатню структуру, щоб її члени могли долучитися до позитивної співпраці для досягнення прогресу під час зустрічей і бесід. Слово «*facile*» означає французькою «легкий» або «простий», тож фасилітатор намагається полегшити групі людей спільну роботу.

Навички фасилітації (сприяння) можуть допомогти поліпшити кожен подій Scrum. Крім того, це сприятиме поліпшенню в проведенні інших робочих зустрічей, а також спеціальних бесід, які відбуваються, коли групи разом виконують складну роботу.

Ступінь сприяння може коливатися від легкого до значного, залежно від потреб групи. Скрізь, коли зустріч чи розмова підпадає під цей діапазон, переконайтеся, що існує достатня структура для досягнення наведених нижче цілей:

- триматися своєї мети й цілей;
- створити середовище, що сприяє змістовному обговоренню та співпраці;
- уточнити рішення групи, ключові результати та наступні кроки.

Будь-який член команди може допомогти їй шляхом фасилітації (сприяння). «Посібник зі Scrum» не вимагає від Scrum-майстра сприяти всім подіям; натомість фасилітація — це навичка, яку можна й потрібно культивувати в Scrum-команді. Навички фасилітації також допомагають членам команди вести власні неформальні бесіди та робочі заняття одне з одним, щоби бути більш цілеспрямованими, креативними та продуктивними.

Навички навчання

Навчання та підтримка інших (коучинг) поліпшує здатність самої людини вчитися, вносити зміни та досягати бажаних цілей. Це творчий процес, який нашоує на роздуми, дозволяє людям ухвалювати свідомі рішення та дає їм можливість стати лідерами у власному житті¹.

На наш погляд, коучинг — не те саме, що давати поради чи консультувати. Ключова відмінність полягає в тому, що особа, якій надають коучингові послуги, бере на себе ініціативу. Отримуючи поради, людина не вчиться та не робить відкриттів на основі власного досвіду та бажань. Вона отримує інформацію, засновану на чужому досвіді та бажаннях. «Консалтинг» — це широкий і доволі вживаний термін, але він зазвичай передбачає виконання роботи (на противагу допомозі іншим знаходити рішення) та консультування людей, як це робити.

Навички коучингу сприяють зростанню Scrum-команди, оскільки вони допомагають членам команди поліпшити свою контрольованість та здатність до самоорганізації. Вони також допомагають команді стати витривалішою під час її зіткнення зі складнощами, новими проблемами та постійними змінами.

Технічна досконалість

Технічна досконалість означає високий рівень розвитку у виборі та застосуванні методів, і справа не лише в технології. Scrum не каже, ані як вам стати чудовою командою розробників, ані як стати прекрасним власником продукту. Підходи, навички та інструменти, що знадобляться вам у кожній ролі, повністю залежать від контексту, у якому ви працюєте. Хоча Scrum не визначає, що саме вам знадобиться для демонстрації технічної досконалості, його застосування абсолютно точно цього вимагає. Поняття технічної досконалості охоплює багато речей: від інженерних методик до мов програмування, від практики управління продуктами до забезпечення якості, від машинобудування до проектування користувацького досвіду та багато іншого.

¹ Для отримання додаткової інформації про коучинг зверніться до Міжнародної федерації тренерів (<https://coachfederation.org/>) та Інституту підготовки тренерів (<https://coactive.com/>).

Оскільки технології та бізнес змінюються дуже швидко, водночас з іншими змінами в середовищі, які впливають на можливості продукту, будь-яка спроба точно визначити, що потрібно для досягнення технічної досконалості, негайно застаріє. Крім того, продукт стає чимось набагато більшим, аніж просто програмне забезпечення. Як наслідок, Scrum-командам потрібно постійно корегувати своє розуміння технічної досконалості, оскільки бізнес та технології потребують змін.

Лідерство як служіння

«Посібник зі Scrum» описує Scrum-майстра як лідера-служителя і наводить приклади способів, у які Scrum-майстер служить власникові продукту, команді розробників та організації. Scrum-майстри — відповідальні лідери-служителі, а це означає, що успіх Scrum-майстра визначається успіхом його або його Scrum-команди. Scrum-майстер допомагає кожному розвинути свої вміння, ефективно долати обмеження та проблеми, а також використовувати емпіризм, щоби з високою частотністю поставляти цінні продукти в складному й непередбачуваному світі.

Однак існує витончена складність у виконанні відповідальної ролі Scrum-майстра. Коли успіх залежить від дій інших, легко захотіти спрямувати їх і втрутитися, якщо здається, що справи йдуть кепсько. Однак таке втручання може підірвати самоорганізацію й почуття відповідальності. Саме тут Scrum-майстра скеровують здібності лідера-служителя.

Наведемо приклади поведінки відповідальних Scrum-майстрів.

- Вони створюють середовище безпеки, заохочуючи до продуктивних дебатів, щоби люди відчували, що їх чують і поважають, відтак допомагаючи командам знаходити та ухвалювати кращі рішення.
- Вони сприяють досягненню консенсусу, допомагаючи командам уточнювати рішення та обов'язки задля підвищення зосередженості та створення однакового розуміння.
- Вони втримуються від розв'язання проблем і мають на меті збільшити прозорість, яка розширює можливості команд та допомагає їм краще самоорганізовуватися, беручи на себе відповідальність за процеси, розв'язання та результати.

- Вони не відчувають дискомфорту від невдач та неясності. Коли рішення команди не призводять до очікуваного результату, вони допомагають команді вчитися та зростати й набувати впевненості у використанні емпіричного підходу, який максимізує навчання та контролює ризик.
- Вони турбуються про людей, зустрічаються з ними на місцях і допомагають їм відшукати наступний крок до зростання, проте не бояться кинути виклик працівникам, коли ті здатні на більше.
- Вони виявляють низьку терпимість до організаційних перешкод і рішуче виступають за те, щоби команда усунула перешкоди, які заважають їй досягти кращих результатів.

Така поведінка сприяє більшій зацікавленості, швидшому зворотному зв'язку та кращим результатам під час створення продукту. Коли менеджери Scrum-команди та інші керівники в організації діють як відповідальні лідери-служителі, вони підтримують зростання і Scrum-команди, і гнучкості у всій організації¹.

Процес безперервного вдосконалення

Додаток А «Самооцінка для розуміння того, де ви перебуваєте» містить набір запитань для самооцінки. Якщо ви й ваша команда витратите час на її проведення, вона принесе додаткові бачення й висновки. Оцінка має допомогти вам визначити сфери, де ви можете вдосконалюватися як команда; вона не має на меті засуджувати вас за те, що ви робите неправильно. В ідеалі цей інструмент спонукає всю вашу Scrum-команду об'єктивно поглянути, де ви перебуваєте і де хочете бути, як на вихідну точку для вдосконалення вашої команди.

Після завершення самооцінки відшукайте запитання, у яких ви оцінили себе на 7 або нижче. Особливо зверніть увагу на запитання, де ви оцінили себе на 5 або нижче.

Список сфер, де, на вашу думку, вам потрібно вдосконалитися, може здатися величезним, але, як ми вже говорили раніше, прозорість важлива для поліпшення результатів, які ви отримуєте за

¹ Ви можете більше дізнатися про лідерство як служіння в контексті Scrum із книжки: Geoff Watts. *Scrum Mastery* (Inspect & Adapt Ltd., 2013).

допомогою Scrum. Щоби почати поліпшувати щось складне, варто поставити собі три наведені нижче запитання.

1. Що дошкуляє найбільше?
2. Чому?
3. Які невеликі експерименти, що принесуть найбільшу цінність, ми можемо провести?

Що дошкуляє найбільше?

Ви не можете виправити все відразу. Ваші енергія та фокус розчинятимуться, коли ви намагатиметеся одночасно змінити занадто багато речей, а це буде ускладнювати досягнення значущого результату для чогось одного. Коли люди в організації не бачать швидкої вигоди, вони зазвичай втрачають інтерес і відмовляються від підтримки, тож нові звички можуть не отримати шансу на закріплення. Розпорошення на надто багато напрямів також ускладнює вимірювання впливу кожної зміни або визначення того, які зміни здійснюють бажаний вплив.

Натомість поступово й безперервно впроваджуйте зміни, уносячи корективи відповідно до того, як будете дізнаватися більше. Інакше кажучи, поліпшуйтеся емпірично! Іноді ці зміни невеликі, а іноді вони мають бути великими. Усе залежить від того, що перебуває в неналежному стані. Зазвичай найкраще розпочати з того, що дошкуляє найбільше.

На практиці: сім поширених функціональних порушень Scrum

З нашого досвіду сім типових помилок заважають командам та організаціям цілком забезпечити гнучкість бізнесу за допомогою професійного Scrum. Ці помилки можуть трапитися навіть за наявності найкращих намірів.

1. Незавершений Scrum. Зі свого досвіду залучення широкого розмаїття команд ми знаємо, що найбільшою проблемою для Scrum-команд є неможливість отримати «готовий» приріст цінності

продукту до кінця спринту. Scrum-команди, які не надають простоту цінності, не можуть перевірити себе та адаптуватися й насправді не отримують жодної вигоди від використання Scrum. Це може призвести до зомбі-Scrum, каскадної Scrum-моделі (*water-Scrum-fall*) чи деяких інших функціональних порушень, перелічених далі.

2. Механічний Scrum (або зомбі-Scrum). Ця проблема полягає в простому русі без духу постійного вдосконалення, без розуміння або турботи про основні цінності та принципи. Це діяльність «для галочки», щоби сказати, що ви щось робите, але насправді процес неживий.

3. Догматичний Scrum. Ця проблема може трапитися, коли «експерт» розповідає Scrum-команді про «найкращі рішення» з власного досвіду. *У Scrum немає найкращих рішень.* Твердження, що команди повинні зважати на певні «кращі рішення», знеохочує самоорганізацію та зрештою обмежує гнучкість. Scrum має бути основою для спонтанних відкриттів. Причина такої легкості Scrum полягає в тому, що конкретні методики та методи не є універсальними. Постачання продукту складне й непередбачуване і вимагає творчого дослідження самоорганізованих команд. Найкраще рішення — те, яке працює для вашого продукту та вашої команди в даний момент. І це, мабуть, не буде кращим рішенням для вашого продукту та вашої команди за шість місяців.

4. Універсальний Scrum. В універсальному Scrum організація хоче стандартизувати та створити Scrum-методологію для всіх Scrum-команд в організації. Ця проблема, яка часто поєднується з догматичним Scrum, іноді виникає переважно через (невідповідне) відчуття контролю та передбачуваності, а не через відчуття створення справжньої цінності для організації. Це може бути спробою забезпечити «покриття» всіх попередніх дій і документів.

У Scrum мають значення не дії, а результати. Ми повинні бути відкритими для нових способів роботи, щоби задовольнити реальні потреби. Scrum — це фреймворк, і командам потрібно з'ясувати свій власний процес у межах Scrum.

5. Каскадна Scrum-модель. Ця проблема має дві версії. У першій Scrum-команда працює в серії спринтів, але по суті досі виконує каскадний процес у межах спринту, послуговуючись хаотичними знаннями та навичками й застосовуючи численні передачі. Це часто призводить до того, що до кінця спринту не з'явиться «готовий» приріст.

У другій версії Scrum-команда здійснює свою «розробницьку» діяльність у спринті, але існують попередні вимоги та цикли проектування й пізніші цикли тестування. Це насправді зовсім не Scrum, оскільки не існує наміру створювати придатний приріст цінності в кінці кожного спринту.

6. Достатньо хороший Scrum. З цією проблемою Scrum-команда отримує певні переваги в ефективності за рахунок регулярного планування та аналізу стану продукту, але вона толерує організаційні перешкоди та поточні обмеження, припускаючи, що «так завжди робилося». Члени команди не вимагають від себе вдосконалення технічних та інженерних практик, щоби наприкінці кожного спринту мати «готовий» приріст цінності.

7. Особливий Scrum/Scrum «у білому пальті». Така ситуація трапляється, коли команда чи організація вважає себе «унікальною», тому повинна адаптувати Scrum до своїх потреб. Ви або робите Scrum, або ні. Модифікація Scrum не розв'язує проблем. Зміна Scrum, найвірогідніше, приховає ваші проблеми... на певний час. Коли проблеми приховано, можливо, відчуватиметься поліпшення, але ж вони досі існують. Зрештою проблеми проявляться як відсутність бізнес-гнучкості та функціональні порушення.

Аналіз першопричини

Запитання «чому?» зводиться до пошуку першопричини. «П'ять чому» — це техніка, яка використовується для визначення основної причини проблеми шляхом повторення запитання «чому?»¹.

¹ Детальніше про техніку «П'ять чому» можна дізнатися за посиланням https://link.springer.com/chapter/10.1007%2F978-981-10-0983-9_32.

«П'ять» у назві цієї техніки походить від спостереження, що для досягнення кореня проблеми зазвичай потрібні п'ять ітерацій, хоча фактична кількість може бути меншою або більшою.

На практиці: використання «П'ять чому» для діагностики першопричин

Щоби проілюструвати використання «П'ять чому», розглянемо таку проблему: релізи постійно затримуються, і це спричиняє розчарування в клієнтів та інших зацікавлених сторін.

Перше запитання, яке ви можете поставити: «Чому релізи постійно затримуються?» Вашою відповіддю, можливо, буде: «Оскільки ми не отримали "готовий" приріст цінності продукту, наша робота мусить продовжуватися в наступному спринті».

У відповідь на це вашим другим запитанням, найвірогідніше, буде таке: «А чому ви не поставили "готовий" приріст цінності продукту?» Вашою відповіддю може стати: «Елементи беклогу продукту завжди більші та складніші, ніж ми вважаємо, і зазвичай ми запізно виявляємо це в спринті».

З огляду на свій досвід, ви, імовірно, уже думали про деякі можливі ключові причини:

- робота занадто велика (командний процес);
- робота недостатньо зрозуміла (командний процес, цінність продукту);
- спосіб, у який команда досягає «готового» приросту, непрозорий або неефективний (емпіризм, командна робота, командний процес);
- непрозорий прогрес (емпіризм, командний процес);
- члени команди можуть боятися порушувати питання проблем та ризиків (командна робота, командна ідентичність).

Тепер ви можете сформулювати точніші запитання, щоби почати заглиблюватися в першопричини. Вашим третім запитанням може стати: «Наскільки прозорим є щоденний процес роботи?» Відпо-

відь, вірогідно, буде такою: «У нас щоденна Scrum-нарада, і ми дивимосся на Scrum-дошку. Члени команди повідомляють про стан карток, над якими вони працюють. Для опрацювання більшості карток потрібно кілька днів, іноді понад тиждень. Отже, наприкінці спринту працівники починають повідомляти, що вони ризикують не виконати завдання. На той час, звичайно, у тестувальників уже не буде достатньо часу».

Виходячи з нашого досвіду, можна стверджувати, що можливі ключові причини полягають:

- у нерозумінні мети щоденної Scrum-наради та поганій її фасилітації (емпіризм, командна робота, командний процес);
- відсутності спільних цілей у команди розробників та відповідальності одне перед одним (командна робота, командна ідентичність);
- у розрізнених знаннях та навичках, які перешкоджають співпраці та швидшому виконанню завдань у спринті (командна робота, командна ідентичність, командний процес).

Беручи до уваги отримані відповіді, поставте запитання для уточнення.

- Наскільки прозорою є щоденна робота?
- Чому члени команди працюють над різними задачами?
- Як Scrum-команда адаптується, коли виявляється, що для закінчення роботи не вистачає часу?

Існує багато шляхів розгортання подібної прикладної розмови, і на практиці вона забере більше часу й вимагатиме більше запитань у процесі пошуку першопричин проблеми. Основні недоліки часто є комплексними і мають кілька ключових причин. Отже, вам доведеться розставити пріоритети, якими шляхами ви спершу підете. Ви будете бачити, як розвиваються теми чи шаблони. Шукайте ключові причини, які є фундаментальними, тобто вони будуть перешкоджати прогресу у розв'язанні інших питань, важливих для ефективності Scrum.

Scrum-команди можуть використовувати цей прийом у ретроспективах спринту, щоби зрозуміти, чому вони стикаються з певною проблемою (рис. 1.1).

На рис. 1.1 обведено три основні недоліки Scrum-команди, і кожна можлива причина показана як така, що сприяє одному або кільком із них. Тепер, візуалізуючи проблеми та їхні ключові причини, Scrum-команда може ухвалювати обґрунтовані рішення щодо вибору відправної точки для розв'язання найбільш критичних проблем. Хоча не існує чарівної формули для усунення всіх можливих першопричин, ітеративний та поступовий підхід дозволить команді знайти варіанти, найкращі для неї в певний момент часу. Поступове вдосконалення здійснюється шляхом використання емпіризму. Обговорюючи складнощі та їхні можливі ключові причини, ви досягли прозорості й уможливили перевірку цієї прозорості інформації.

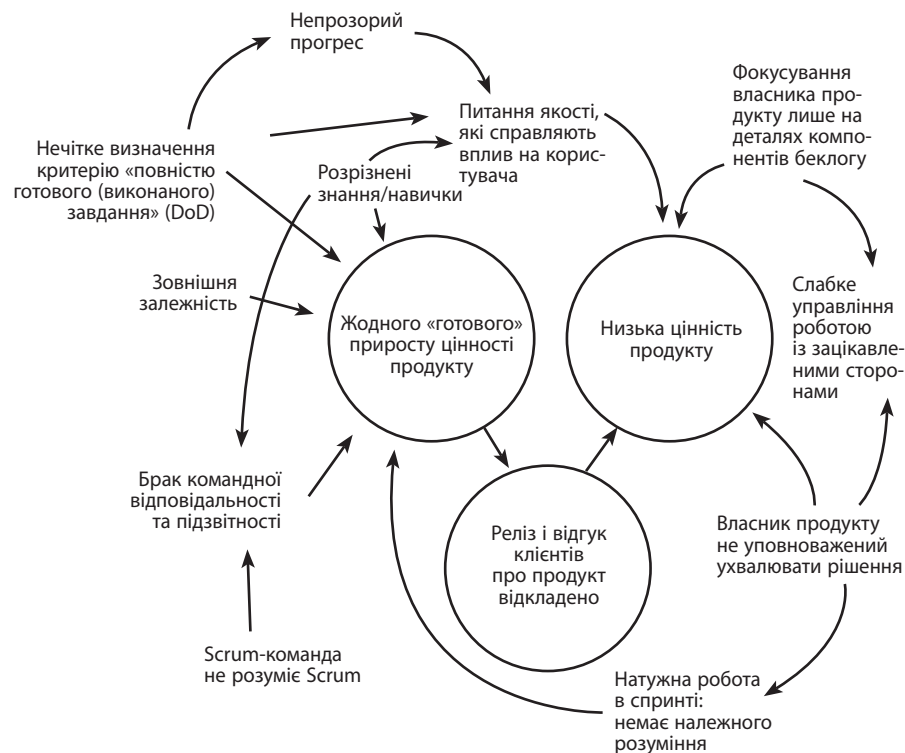


Рис. 1.1. Зразок результату аналізу ключових причин

Експериментуємо з різними підходами

Складні проблеми не мають простих або очевидних рішень. Перш ніж добре фінансово вкластися в конкретне рішення, переконайтеся, що ви розумієте проблему та маєте життєздатне рішення для її усунення. Незалежно від ваших даних, інтуїції та досвіду завжди знайдуться речі, про які ви не знаєте.

Щоби просуватися вперед і не бути паралізованим цими багатьма невідомими, ви можете спробувати кілька експериментів і побачити, що саме спрацює, або зібрати більше інформації¹. Звучить, як щось складне для реалізації та непередбачуване, чи не так?

Щоб ефективно використовувати експерименти з метою вдосконалення, виконайте низку наведених нижче дій.

1. Визначте проблему, яку намагаєтеся розв'язати. Ви, мабуть, отримали певне уявлення про це внаслідок власного аналізу першопричин.
2. Висуньте гіпотезу про певні дії, які, на вашу думку, можна поліпшити.
3. Вирішіть, що ви будете робити для перевірки цієї гіпотези.
4. Проведіть експеримент.
5. Проаналізуйте результати. Сюди входять порівняння фактичних результатів і очікувань, розмірковування про навчання й отримання відгуків (зворотного зв'язку).
6. Уточніть і повторіть. Може відбутися зміна гіпотези або експерименту.

Під час планування експерименту чітко усвідомлюйте наведені нижче моменти.

- Чого ви намагаєтеся навчитися?
- Як ви будете вимірювати успіх?
- Як швидко ви можете отримати відгук (зворотний зв'язок)?

Створюючи експеримент, ви також хочете враховувати потенційну рентабельність інвестицій (ROI) експерименту. В ідеалі експеримент повинен бути досить невеликим, щоби ви могли мінімізувати

¹ Структурований та дисциплінований підхід до випробування речей є основою наукового методу; див. <https://www.britannica.com/science/scientific-method>.

інвестиції та швидше отримати зворотний зв'язок. Також він повинен забезпечити достатню цінність. Плоди, що низько звисають, можна швидко й легко зібрати, але й прибутку від них ви можете отримати менше. На створення більш цінних речей може знадобитися більше інвестицій, часу та енергії.

Правильної відповіді немає. Ви мусите зважати на унікальні недоліки та унікальні потреби вашої команди. Ви маєте проявити творчий підхід до розбиття великих сегментів матеріалу на менші експерименти з вищою цінністю. Чинячи в такий спосіб, ви можете здійснювати вдосконалення ітеративно та з поступовим приростом.

Тепер ви знаєте, у якій точці ви перебуваєте і де хочете бути. Коли ви почнете визначати експерименти, які слід провести в намаганні наблизитися до бажаної точки, створіть беклог удосконалення. Упорядкуйте всі елементи й розпочинайте.

Так само як Scrum використовує емпіричний підхід для розв'язання складних проблем та постачання цінних продуктів, ви можете використовувати емпіричний підхід для розв'язання складних проблем та максимізації переваг Scrum. Ви в змозі зробити це на рівні Scrum-команди та за її межами, на інших рівнях організації. Для окремої команди цей цикл постійного вдосконалення вже вбудований у послідовність спринту та використання ретроспективи спринту для перевірки та пристосування як Scrum-команди. Крім того, кожна така команда визначає кількість часу, який потрібно виділити на вдосконалення кожного спринту, а також спосіб організації та перевірки вдосконалень, внесених під час кожного спринту.

Успіх чи невдача?

Чи можливо, щоб успіх був провалом? Чи можливо, щоби невдача була успіхом?

Ви, можливо, помітили, що багато запитань щодо оцінки гнучкості бізнесу в Додатку А стосуються результатів (наприклад, цінність, швидке постачання). Хоча результати є найважливішими, варіанти поведінки також можуть бути важливими, коли допомагають розвивати командні вміння.

Ви не можете контролювати всі змінні в складній роботі та передбачуваних умови в середовищі навколо вас. Якби ви могли, то

спланували б усе заздалегідь, дотримувалися б цього плану й отримували гарантовані результати. Однак у заплутаному реальному світі ви можете робити всі «правильні речі» і все одно не отримати бажаних результатів. Ось чому також важливо звернути увагу на поведінку.

Аналізуючи результати експериментів або спрямованих на вдосконалення кроків, беріть до уваги результати та поведінку, особливо їхні тенденції з плином часу. Наприклад, розглянемо ситуацію, коли команда розробників виявляє основні технічні проблеми під час нової інтеграції. Команда розпочала цю роботу в перший день спринту, оскільки її члени знали, що завдання буде складнішим, і до цього вже на власному гіркому досвіді зрозуміли, що слід раніше розв'язувати більш ризиковані питання. Вони аж роїлися. Власника продукту повідомили про ситуацію та спільно працювали над тим, щоби розбити цю роботу на менші частинки. Зрештою, до «готового» приросту цінності вони так і не просунулися.

У цьому прикладі видно явну невдачу: команда не має «готового» приросту. Проте є й успіх: команда використовувала знання зі свого попереднього досвіду й зробила все, що могла, застосувавши те, що було їй відомо на той час. Вони співпрацювали, вели переговори та адаптувалися протягом спринту. Головне — набути нових знань, щоби наступного разу зробити ще краще. Можливо, команда вирішить відкорегувати спосіб удосконалення беклогу продукту, щоби по-іншому розбити елементи. Імовірно, вони виявлять недолік кваліфікації, який необхідно буде усунути. Можливо, вони вирішать змінити свою методику розробки або інструменти.

Зрештою, є два запитання, які слід поставити.

- Чи зробили ми все, що від нас залежало, зі знаннями, які мали на той час?
- Як ми можемо зробити краще?

Підсумки

Сім основних сфер удосконалення, на яких ми зосереджуємося,— гнучке мислення, емпіризм, командна робота, командний процес, командна ідентичність, цінність продукту та організація — дають лінзу, за допомогою якої ви можете роздивитися й перевірити здатність вашої команди досягати своїх цілей та знаходити шляхи вдосконалення. Шукаючи ключові причини, проводячи експерименти в спробах поліпшення, а потім перевіряючи та пристосовуючись, ви можете поступово, послідовно та постійно вдосконалювати свою здатність досягати кращих результатів.

Сім ключових сфер також дають лінзу, крізь яку ви можете спостерігати за результатами й поведінкою та шукати ключові причини, знімаючи шар за шаром. Ця лінза створює фокус і чіткість, щоби ви могли вдатися до роздумів і, сформувавши намір, виконати дії.

Ви поліпшуєте емпіризм, застосовуючи його. Вам потрібно створити прозорість щодо бажаних результатів удосконалень та регулярно перевіряти й пристосовувати свій шлях до максимального використання переваг Scrum.

Заклик до дії

Перегляньте свої нотатки щодо запитань для самооцінки й характеристики та візьміть до уваги наведені нижче моменти.

- Що ви помітили в даних?
- Які тенденції бачите?
- Яку нову аналітичну інформацію ви отримали внаслідок цього оцінювання?

Використовуючи інформацію, що ми обговорили в цьому розділі, як посібник, проведіть зі своєю Scrum-командою колективну дискусію, щоби зробити низку кроків.

1. Визначте дві-три головні невдачі.
2. Визначте можливі ключові причини для кожної з них.
3. Оберіть дві-три ключові причини, які необхідно усунути.
4. Створіть упорядкований список перших поліпшень, які ви хочете здійснити. Для кожного з цих «експериментів» обов'язково уточнюйте очікувані результати та способи їх вимірювання.
5. Починайте.

Розділ 2

Створення сильної команди

Успіх гнучкого (agile) підходу до управління починається із сильної команди. В ідеалі ця команда є згуртованою, кросфункціональною та самоорганізованою, хоча більшість команд створюються як сукупність людей, які мусять навчитися працювати разом заради досягнення спільних цілей. Члени нових Scrum-команд, особливо ті, хто не звик працювати в складі кросфункціональної спільноти, спершу часто намагаються забезпечити «готові» прирости цінності продукту в кожному спринті. У цьому розділі ми розглянемо, як команди можуть подолати ці виклики.

Формування командної ідентичності

Ви не можете просто зібрати групу людей, сказати їм: «Ви — команда» і чекати, що вони досягнуть значних успіхів. Формування команди означає інвестування в об'єднання групи людей для досягнення того, чого їм не до снаги досягти поодиночі. Разом члени команди формують абсолютно новий живий і дієвий організм, ідентичність якого з часом розвивається.

На базовому рівні формування командної ідентичності відбувається шляхом відповіді на три запитання, які спрямовують команду на її шляху до успіху.

- Чому ми існуємо — яка наша мета?
- Що для нас важливо — які цінності є для нас пріоритетними?
- Чого ми хочемо досягти разом?

Так само як і окремих індивідів, кожна команда буде постійно вдосконалювати та уточнювати власну ідентичність у процесі навчання

й зростання її членів. Переконавання команди щодо своєї ідентичності допомагають (або перешкоджають) їй на шляху до досягнення спільних цілей та постійного підвищення ефективності. У контексті гнучкої розробки продукту agile-світогляд дає корисну вихідну точку для формування ідентичності Scrum-команди¹.

Що створює хорошого члена команди?

Ми — люди, і тому робимо помилки, навіть якщо маємо добрі наміри. Ми хочемо зробити все можливе. Хочемо рости й навчатися. Ми всі здатні на набагато більше, аніж знаємо. Ми процвітаємо, коли відчуваємо зв'язок і комунікуємо. Також ми надзвичайно унікальні.

Навіть у командному середовищі важливі розуміння та поцінування особистості. Хоча людям необхідно відмовитися від індивідуального статусу та его, щоби зосередитися на спільних цілях та результатах, кожен у команді все одно має задовольнити власні потреби. Зрештою, зацікавлені та задоволені люди будуть креативнішими та продуктивнішими, і, певно, з ними буде ще приємніше працювати.

Щоб оцінити унікальні навички та таланти, які окремі члени привнесуть у команду, важливо брати до уваги три аспекти й зрозуміти, як участь у такій команді може допомогти цим людям зростати й отримувати задоволення.

- **Особистість.** Люди надзвичайно унікальні. У них різний характер, який виявляється в уподобаннях та поведінці, котрі зазвичай зберігаються впродовж усього життя. Деякі характеристики мають генетичне походження, а інші визначаються досвідом. Однак для досягнення бажаних командних цілей люди можуть свідомо вирішити діяти інакше, не за притаманними їм уподобаннями².

Особистісні відмінності породжують конфлікт, але ці самі відмінності породжують і розмаїття, яке розширює перспек-

тиви в такий спосіб, що робить команди інноваційнішими та ефективнішими¹. Коли команди ефективні, їхні члени з'ясовують, як правильно долати конфліктні ситуації.

- **Емоційний інтелект.** Емоційний інтелект — це розуміння власних емоцій і поведінки та управління ними, а також здатність розпізнавати емоції інших та впливати на них². Емоційний інтелект допомагає людям зрозуміти, коли і як виражати свої особистісні якості.

Згідно з дослідженнями *TalentSmart*, у складі емоційного інтелекту міститься гнучкий набір навичок, які можна вдосконалити завдяки практиці. Хоча деякі люди, природно, мають вищий за інших емоційний інтелект, ви можете набути та розвинути цей набір. Крім того, *TalentSmart* виявив, що 90 % працівників із найкращою продуктивністю також мають високий рівень емоційного інтелекту. З іншого боку, лише 20 % працівників з низькою продуктивністю мають високий емоційний інтелект³. У цьому є сенс, коли ми замислюємося про природу розробки продукту.

- **Внутрішня мотивація.** Хоча мотивація має значення для будь-якої роботи, самоорганізовані команди просто неефективні, якщо кожен їхній член не має внутрішньої мотивації. Як зазначає Ден Пінк у книжці «Драйв», працівників, зайнятих розумовою працею, не мотивують зовнішні винагороди на кшталт грошей. Натомість їх мотивують три фактори⁴.
 - **Автономія.** Люди самі контролюють свою роботу.
 - **Майстерність.** У людей з'являється можливість у чомусь досягти успіху, зростати та відточувати свої знання й уміння.
 - **Мета.** Люди відчувають, що працюють над чимось більшим за самих себе. Вони бачать сенс у своїй роботі.

¹ <https://hbr.org/2013/12/how-diversity-can-drive-innovation>

² Докладніше про емоційний інтелект див. у книжці: Daniel Goleman *Emotional Intelligence: Why It Can Matter More Than IQ* (Bantam Books, 2005).

³ Travis Bradbury і Jean Greaves, *Emotional Intelligence 2.0* (TalentSmart, 2009).

⁴ Den Pink, *Drive*. (Riverhead Books, 2011). У книжці подається переконлива дискусія про те, що спонукає людей працювати над складними інтелектуальними завданнями. Це питання узагальнено в цьому короткому відео: <https://www.youtube.com/watch?v=u6XAPnuFjJc>. Більш детально див. <https://www.danpink.com/drive>.

¹ Стосовно ідей використання цінностей Scrum та Agile-маніфесту (www.agilemanifesto.org) для уточнення ідентичності команди див. <https://www.Scrum.org/resources/blog/maximize-Scrum-Scrum-values-focus-part-1-5>.

² <https://carleton.ca/economics/wp-content/uploads/little08.pdf>

На практиці:**особистість та емоційний інтелект**

Scrum-команда розчарувалась у працівнику на ім'я Алекс, якого інші члени вважають «негативним», «антагоністичним» і «хитрим». Нова Scrum-майстриня отримувала ці скарги від окремих осіб упродовж декількох тижнів. Вона також уважно спостерігала за взаємодією між Алексом та рештою команди й відчуває, що особистісні характеристики Алекса мають низьку оцінку на шкалі доброзичливості¹.

Завдяки низці індивідуальних бесід з Алексом та членами команди, а також за рахунок використання можливостей, які надаються в процесі групових обговорень командної роботи, Scrum-майстриня допомагає краще зрозуміти особистісні відмінності й переваги, що з ними пов'язані. Члени команди починають розуміти, що Алекс не намагається негативно або скептично ставитися до їхніх ідей. Для нього природно спочатку ставити під сумнів речі, намагаючись краще їх зрозуміти, щоби впевнитися, що було проведено достатнє дослідження. Співробітники навіть починають цінувати і частіше цікавитися його думкою. Алекс також краще розуміє, наскільки негативно можуть сприйматися членами команди його коментарі і що вони навіть можуть почуватись ображеними, тож починає уважніше підходити до того, як він ставить під сумнів і досліджує ідеї своїх колег.

Дізнавшись більше про власну особистість та якості особистості одне одного, члени команди отримують здатність краще обирати, як вони проявлятимуть себе в щоденній взаємодії. До того ж вони краще досягають бажаних результатів, відчуваючи легкість у співпраці.

Успішні команди складаються з працівників, які знають свої слабкі й сильні сторони настільки, щоби зрозуміти власні сили, мають достатній емоційний інтелект для адаптації своїх реакцій до людей, що їх оточують, і є внутрішньо мотивованими працюва-

¹ Хоча було розроблено багато моделей для розуміння особистісних параметрів, фактор доброзичливості походить від моделі п'яти факторів. Див. <https://positivepsychology.com/big-fivepersonality-theory>.

ти з іншими заради досягнення того, чого вони не змогли досягти самотужки¹.

Хто має бути в Scrum-команді?

В ідеальному світі, де кожен має всі необхідні для команди навички, ми сказали б, що відповідь на це запитання така: «Кожен, хто хоче бути в Scrum-команді». Насправді, коли вам потрібна відносно невелика кількість людей у команді, щоби мати змогу впоратися зі складністю комунікації, і вам необхідно багато різних навичок, щоби поставляти «готовий» приріст цінності продукту в кожному спринті, слід зробити розумний вибір і досягти компромісу щодо того, хто входить до Scrum-команди, а хто ні. Тоді як команда ви повинні працювати в рамках обмежень, установлених цим вибором.

З плином часу вам потрібно буде постійно вдосконалювати свої навички. У команді ви робите все від вас залежне, використовуючи наявні навички, а потім працюєте над розвитком нових, щоби заповнити будь-які прогалини. Перебування в кросфункціональній команді означає, що її члени готові робити те, що сприятиме загальному успіху, навіть якщо вони не є найкращими або найшвидшими в цій справі. Такі команди здогадуються, як в цілому скористатися з навичок та знань, якими вони володіють, тому вживають заходів для зростання, коли виявляють прогалини та фактори, що уповільнюють їхній розвиток. У перспективі це означає, що вони поступово розвиватимуть усе більше різних видів ґрунтовних навичок (див. рис. 2.1 на с. 52)².

¹ Детальніше про навички та риси, які люди можуть розвинути, щоби стати кращими в командній роботі, див. у книжках: Christopher Avery, *Teamwork Is an Individual Skill* (ReadHowYouWant, 2012) і Patrick Lencioni, *The Ideal Team Player* (Jossey-Bass, 2016).

² «Т-подібні навички» (*T-shaped skills*) — це метафора, яка стосується людини, що має розлогі навички, пов'язані з розв'язанням проблем або сферою ділової активності, а також глибші навички в межах власної спеціалізації (див. https://en.wikipedia.org/wiki/T-shaped_skills). π-подібні та комбіновані навички стосуються людини, яка має ґрунтовні вміння більш ніж в одній спеціалізації.



Рис. 2.1. З часом у членів команди прослідковується тенденція до розвитку ґрунтовних навичок у більшості сфер

Команди розробників наділені кросфункціональністю. І справді, це ключовий аспект самоорганізації. Команда розробників може взяти до себе когось, щоби підвищити власний рівень навичок та знань. Члени команди також можуть або обрати плановий тренінг, або витратити час на самоосвіту з метою розвитку ширших та ґрунтовніших навичок та знань. Крім того, команда розробників може обрати спосіб роботи, який підтримує наставництво, спрямоване на подальший розвиток наявних членів команди. Відповідно до розвитку продукту та команди також необхідно буде сприяти розподілу навичок та знань між її членами.

Команди розробників повинні знати більше, ніж просто розробка

Коли ви замислюєтеся над навичками, необхідними для постачання споживачам або реальним користувачам «готового» приросту продукту, розгляньте кілька запитань.

- Чи корисно членам команди розробників розуміти, як споживачі користуються продуктом?
- Чи корисно членам команди розробників дізнатися, як зміни продукту можуть вплинути на бізнес-процеси чи інші продукти, котрі використовуються клієнтами або в межах організації?
- Чи корисно членам команди розробників розуміти, як можуть вплинути на їхній продукт зміни інших продуктів, бізнес-процесів та політики в організації?

Звичайно, відповіді на всі ці запитання категорично ствердні! Навички бізнес-аналізу є важливим складником досягнення ефективного приросту цінності продукту, але це не завжди означає, що вам потрібен бізнес-аналітик в команді розробників; вам доведеться вийти за рамки старих способів створення продуктів.

А що, коли хтось із таким бізнес-контекстом приєднається до команди розробників? Як знання, досвід та навички цієї людини можуть сприяти роботі в спринті? Імовірно, новий учасник може відповісти на запитання й скерувати напрям роботи над елементом беклогу продукту (PBI) у процесі його розробки. Або може сприяти підвищенню якості, надаючи поради щодо підходів до тестування та деталей тест-кейсів, а також допомагаючи у виконанні самого тестування. Імовірно, учасник може зіграти свою роль у розробці онлайн-довідки, навчальних матеріалів або діяльності з управління бізнес-зміними. Можливо, він чи вона можуть зробити свій внесок у вдосконалення беклогу продукту. Усі ці випадки сприяння допоможуть команді розробників відповідати суті критерію «повна готовність завдання».

Не всі в команді розробників мусять знати, як писати код, та й створення «готового» приросту цінності продукту вимагає багато більше, ніж просто написання коду.

Як Scrum-команди створюють робочі угоди?

Робочі угоди допомагають групі людей із різними характерами, уподобаннями та досвідом ефективно працювати разом, чітко визначаючи, який вигляд має відданість команді. Майте на увазі, що робочі угоди не є правилом для Scrum-майстра, яке слід застосовувати. Натомість, уся Scrum-команда бере на себе цю відповідальність: її члени відповідають одне перед одним і розв'язують проблеми відповідно до їх виникнення.

Однак робочі угоди не статичні, і командам слід регулярно переглядати та оновлювати їх у процесі свого розвитку. Робочі угоди часто стосуються трьох сфер:

- завдання, або очікувані для команди заходи та результати;
- процес, або те, як буде виконуватися діяльність;
- норми, або способи взаємодії членів команди між собою.

Створення робочої угоди сприяє формуванню міцної командної ідентичності. Робочі угоди народжуються на основі цінностей та принципів, дорогих членам команди, і розробка таких угод часто конкретизує ці цінності та принципи. Подібна угода створює основу, на якій команда може самоорганізуватися. Команда більше прагне досягти власне командного успіху, аніж успіху окремого її члена, коли всі учасники поділяють спільний набір цінностей та принципів. Також легше дотримуватися командних рішень, навіть якщо працівник має певні перестороги, але знає, що певні рішення ґрунтуються на цінностях та принципах.

Постановка низки запитань може допомогти вам скласти робочу угоду для своєї команди.

- Якими є наші стандарти якості та як ми забезпечимо їх досягнення?
- Як ми будемо ефективно співпрацювати?
- Як ми будемо ділитися інформацією всередині команди та із зацікавленими сторонами?
- Які наші стандарти щодо відвідуваності, оперативності та участі?
- Як ми будемо ухвалювати рішення?
- Як ми виявлятимемо конфлікти чи розбіжності?
- Як ми маємо поводитися під час конфлікту?
- Як цінності Scrum керуватимуть нашою взаємодією та роботою?
- Як ми будемо розвивати знання та вміння серед членів команди?
- Який вигляд для нас має доброзичлива поведінка?
- Як ми будемо відслідковувати нашу роботу й прогрес?
- Як ми будемо вимагати одне від одного відповідальності за виконання взятих на себе зобов'язань?

Визначення критерію «повністю готове завдання» також є робочою угодою в команді розробників, вона про те, як члени команди забезпечуватимуть якість та повноту приросту цінності продукту.

На практиці: заява про цілі команди

Допомога в укладанні заяви про цілі (рис. 2.2) може бути корисною для об'єднання команди та внесення ясності й прозорості в її робочі угоди. Заява про цілі стосується низки елементів.

- Що: які потреби замовника команда намагається задовольнити.
- Як: методи або технології, що їх команда використовуватиме для задоволення цих потреб клієнта.
- Хто: опис пріоритетного замовника.
- Чому: причини важливості задоволення цих потреб клієнта.
- Родзинка: що вирізняє команду з-поміж інших.

Ми пропонуємо інтуїтивно зрозумілі рішення для управління даними, щоб наші користувачі могли ефективніше виконувати свою роботу та обслуговувати клієнтів й були впевнені, що дотримуються всіх нормативних, правових та безпекових стандартів.

Рис. 2.2. Зразок заяви про цілі команди

Який вигляд має самоорганізація?

Самоорганізована Scrum-команда здатна визначати, як вона виконує свою роботу. На практиці це може проявлятися по-різному.

- Scrum-команди беруть на себе відповідальність за власні процеси. Вони не звинувачують сам процес, а змінюють те, що не працює належно, і борються з організаційними процесами, які їм заважають.

- Scrum-команди визначають ціль спринту разом. Команди розробників ухвалюють рішення, скільки роботи разом вони можуть спрогнозувати в спринті. Вони вирішують, як саме вони будуть виконувати роботу разом.
- Члени команди розробників не чекають, поки щоденна Scrum-нарада порушить проблему чи вкаже на перепони.
- Члени команди розробників вирішують, як і коли почати боротися з проблемами, що загрожують цілі спринту.
- Члени команди розробників оновлюють свій беклог спринту для відображення поточного прогресу та нових знань.
- Scrum-команди визначають, як вони будуть командно вдосконалюватися, і беруть на себе відповідальність за виконання взятих зобов'язань у кожному спринті.
- Члени Scrum-команди звертають увагу на незгоди та конфлікти й розв'язують їх.
- Scrum-команди своєчасно ухвалюють узгоджені рішення. Вони визначають, чи потрібно їм консультуватися зі сторонніми експертами.

Ефективна самоорганізація вимагає трьох речей: спільних цілей, чіткої відповідальності та меж (рис. 2.3). Якщо якийсь зі складників послабиться, команда може втратити здатність до самоорганізації та стати менш ефективною.

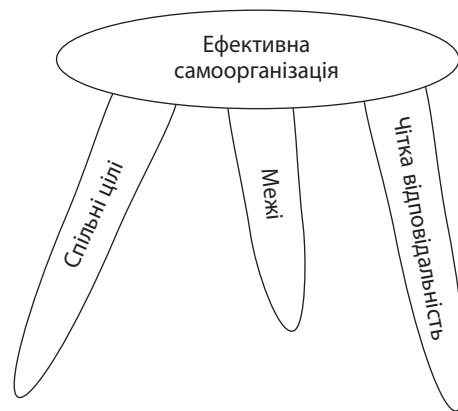


Рис. 2.3. Ефективна самоорганізація вимагає, щоб усі три опори залишалися в рівновазі

Спільні цілі

Усім чудовим командам потрібна ціль. І що зухваліша, то краще. Їм потрібно те, до чого вони можуть прагнути й дотягуватися, і досягнення, за яким вони можуть оцінити себе. Без спільних цілей шляхи членів команди можуть легко розійтися, а сама команда втратить цілеспрямованість та згуртованість.

Спільні цілі зазвичай беруть початок із цілей продукту і виражаються як добре сформульована бізнес-стратегія, чітке бачення продукту, розуміння його цінності для споживача та зрозумілий спосіб її виміру. Усі ці аспекти скеровують команди й допомагають їм зрозуміти, куди вони прямують і що важливо.

Мета спринту також важлива і забезпечує загальну ціль або завдання Scrum-команди під час проведення спринту. Так забезпечується фокусування команди, коли вона розкриває нову інформацію та стикається з проблемами в процесі створення приросту цінності продукту під час спринту. Ви можете розглядати цілі спринту як проміжні точки маршруту, що складають шлях до досягнення більших, довгострокових релізів чи бізнес-цілей.

Щоденна Scrum-нарада ще більше деталізується й зосереджується на роботі, яку команда розробників проведе впродовж наступних 24 годин для просування до цілі спринту.

Чітка відповідальність

Scrum забезпечує чітку відповідальність за кожну роль. Організація повинна поважати цю відповідальність. Це означає забезпечення членів Scrum-команди повноваженнями виконувати свої ролі. Члени команди також потребують знань та навичок для виконання взятих на себе зобов'язань. Для цього можуть знадобитися інвестиції в передачу знань та навчання. Це також може означати надання людям доступу до інформації, яка допоможе в прийнятті рішень.

Звичайно, членам Scrum-команди потрібно достатньо часу для виконання своїх ролей. Коли члени команди виконують декілька обов'язків, крім своєї ролі в Scrum, важливо оцінити вплив, який це може мати. Яка роль є пріоритетною? Чи достатньо в людей часу для виконання своєї ролі в Scrum? А як щодо інших ролей? Що трапляється, коли людині доводиться робити важкий вибір — або відкласти щось, щоби виконати роль у Scrum, або ж навпаки?

Ба більше, якщо виникає невідповідність між тим, як оцінюється індивідуальна діяльність, і тим, за що відповідають окремі члени команди, ці працівники опиняться в складній ситуації. Чи повинні вони робити те, що відповідає їхнім власним інтересам, щоб отримати «хорошу оцінку ефективності роботи»? Або ж вони мають зробити все від них залежне для виконання своєї ролі в Scrum?

Те, що потрібно для виконання ролей власника продукту, команди розробників та Scrum-майстра з часом зміниться в процесі розвитку продукту, бізнесу та Scrum-команди. Щоби забезпечити задоволення цих потреб, Scrum-командам важливо продовжувати оцінювати те, що необхідно зараз і що знадобиться найближчим часом.

Межі

Фреймворк Scrum, разом із 11 елементами та правилами, які пов'язують їх між собою, установлює межі, що роблять «безпечною» самоорганізацію Scrum-команди. Під «безпечною» ми маємо на увазі, що ризик невдач зменшується, а вартість відмов обмежується.

Часові рамки (таймбокси) в Scrum — це приклад меж, які забезпечують фокусування, створюють відчуття терміновості, зменшують витрати та обмежують ризик. Поміркуйте, як використання таймбоксів забезпечує ці переваги команді або де ці переваги можуть бути відсутні.

Поява «готового» приросту цінності продукту потрібна щонайменше до кінця спринту, а критерій «повної готовності завдання» дає чітку межу того, що означає якість і повнота для команди розробників. Зверніть увагу, що організація може мати мінімальний базовий критерій «повної готовності завдання» — це приклад організації, котра встановлює мінімальні межі, на які команда розробників може потім спиратися.

Може виникнути потреба у встановленні та уточненні меж поза Scrum. Ці межі можуть стосуватися технологічних рішень, розвитку команди або багатьох інших категорій. Поставте запитання для уточнення меж.

- Які рішення уповноважена ухвалювати Scrum-команда?
- З ким команда повинна консультуватися для прийняття певних рішень?

- Кого повинна інформувати команда, коли вона ухвалює певні рішення?

Наприклад, Scrum-команді, можливо, доведеться проконсультуватися з групою з архітектури підприємства, коли члени команди хочуть запровадити нову технологію у своїй платформі продуктів. Вірогідно, їм просто потрібно буде поінформувати групу з архітектури підприємства, коли вони вносять зміни, які збільшують розмір їхньої платформи продуктів.

Інший приклад — рішення, які може ухвалювати Scrum-команда щодо свого розвитку. Команда самостійно визначає, як її члени спільно працюють над створенням приросту цінності продукту, навчаються та навчають одне одного для розвитку навичок. Але їй (команді), можливо, доведеться проконсультуватися з менеджером, коли вона захоче інвестувати в тренінги та інші навчальні ресурси більше за конкретну суму.

Лідерство та самоорганізація

Лідерство має важливе значення задля створення умов для ефективної самоорганізації. У книжці Девіда Марке «Розверніть ваш корабель!» (*Turn the Ship Around!*) описується взаємозв'язок між трьома факторами контролю (тобто забезпечення більшого контролю), компетентністю та ясністю як засобом створення простору лідерства, що базується на намірах.

Робота керівника полягає в підтримці команд розробників в їхніх зусиллях для підвищення загальної компетентності. Лідер може допомогти командам набутися компетентності та взаєморозуміння, а потім дати їм більше можливостей самим контролювати процес. Як варіант, керівник може спершу послабити контроль зі свого боку, а потім заповнити лакуни компетентності та чіткості розуміння. Шлях до швидких змін і вдосконалення полягає в наданні більших можливостей щодо контролю власної роботи, а це вимагає від керівників довіри до своїх команд. Їм простіше це зробити, коли вони мають межі спринту для створення фокуса й мінімізації наслідків невдачі.

Люди найшвидше розвивають компетентність та чіткість розуміння, коли навчаються, виконуючи роботу,— у цьому суть емпі-

ризму! Навіть коли бракує власних цілей, працівники дізнаються щось нове, що допоможе їм наблизитися до цілі наступного разу, коли вони зроблять ще одну спробу. Деякі прихильники Agile називають цей підхід таким, що «швидко наражається на невдачу», але єдиною справжньою невдачею є невміння вчитися на експериментах.

Як співпрацюють Scrum-команди?

Самоорганізована, кросфункціональна команда повинна навчитися співпрацювати. Для цього її учасникам потрібно вкластися у створення «спільних активів». Скористатися перевагами ефективної співпраці командам допомагають п'ять активів:¹

- довіра;
- продуктивний конфлікт;
- відданість;
- відповідальність;
- спільні цілі та результати.

Ці активи, як показано на рис. 2.4, базуються одне на одному. Якщо у вас немає довіри, інші чотири активи буде неможливо підтримувати. Якщо у вас немає продуктивних конфліктів, неможливо проявляти відданість, відповідальність та мати спільні цілі. І так далі для кожного активу.

У цьому контексті довіра означає готовність показати себе вразливим перед колегами, наприклад визнати помилку або звернутися по допомогу. Коли члени команди довіряють одне одному, вони відкриті для продуктивних конфліктів: готові кинути виклик одне одному, заперечити припущення, бути відкритими й ділитися дивними й божевільними, на їхню думку, ідеями².

¹ Якщо ви прочитали книжку Патріка Ленсіоні «П'ять дисфункцій команди» (*The Five Dysfunctions of a Team*, Jossey-Bass, 2002), ці активи можуть здатися вам знайомими. У своїй надзвичайно відомій книжці Ленсіоні представляє п'ять дисфункцій. Тут ми зосереджуємося на протилежному цим дисфункціям, тобто на активах готових до співробітництва команд.

² Докладніше цю тему див. <https://www.agilesocks.com/build-trust-enable-agility/>.

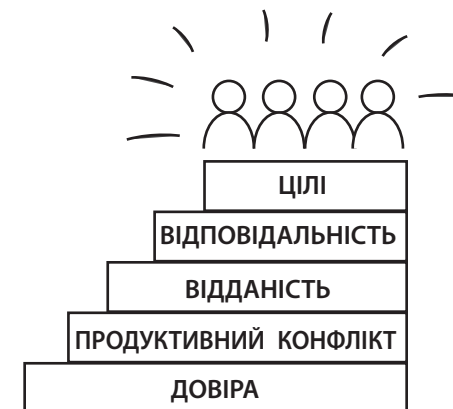


Рис. 2.4. Ці активи закладають основи для ефективної співпраці команд

Довіра — це готовність бути вразливим, готовність робити щось важливе для вас вразливим до дій інших людей. Люди, які працюють в атмосфері довіри, можуть плідно співпрацювати, дозволяючи розв'язувати справи швидше та з меншими витратами. І навпаки, коли довіра відсутня, бізнес сповільнюється, а витрати зростають, оскільки люди проводять більше часу, намагаючись захиститися від тих, кому вони не можуть довіряти¹.

На практиці: будування довіри

Немає жодного надійного способу побудувати довіру, але існує незліченна кількість способів її знищити. Ми розглядаємо довіру як постійну подорож, що насамперед полягає в послідовності того, як ви проявляєтеся у своїх стосунках та взаємодіях. Ця подорож ніколи не завершується, і ви можете легко почати рухатися у зворотному напрямі. Наведемо декілька прийомів, які ми використовуємо для формування довіри².

¹ Stephen M. R. Covey, *Speed of Trust* (Free Press, 2008).

² Для глибшого розуміння побудови довіри див. <https://brenebrown.com/videos/anatomy-trust-video/>.

- *Зробіть крок першими.* Можливо, вам доведеться спершу самому довіритися, перш ніж вам почнуть довіряти у відповідь. Будьте вразливими, щоби показати іншим, що це — нормально. Просіть про допомогу. Визнавайте свої помилки.
- *Будьте готові сказати «ні».* Коли ви берете на себе надмірні зобов'язання, то піддаєтеся ризику через те, що чогось не виконуєте й негативно впливаєте на інших. Вас можуть сприймати як ненадійну людину.
- *Сприймайте наміри як позитивні.* Зробіть усе можливе, щоби вбачати у вчинках чи словах іншої людини позитивні наміри. Хоча може бути доречним розглянути й ситуацію, коли чиїсь дії чи слова мали негативні наслідки, але проведіть розмову, припускаючи, що людина мала добрі наміри. Це допомагає усунути конфлікт, розв'язати проблеми та краще зрозуміти одне одного, демонструючи довіру іншій людині.
- *Уникайте пліток.* Розмова про людей часто сприймається як простий спосіб налагодити спілкування та зв'язок з іншими. Однак така поведінка може спричинити неочікувані наслідки, оскільки сприятиме ставленню до вас як до ненадійної людини. Якщо ви говорите про когось і ділитесь зі мною інформацією, яку ця людина вам довірила, як я можу знати, що ви не вчините так само з тим, що я вам сказав по секрету?
- *Узгодьте свої слова й дії.* Переконайтеся, що ви дотримуєтеся того, що, на вашу думку, є важливим. Якщо ви скажете своїй команді, що рівномірний темп роботи важливий, але працюєте понаднормово й відповідаєте на електронні листи у вихідні, між вашими переконаннями, які ви транслюєте у світ, та вашими діями виникає розбіжність.
- *Будьте відкритими та чесними з людьми.* Створення середовища, у якому люди можуть бути відкритими та чесними у вираженні своїх почуттів, проблем і бажань, має важливе значення для довіри. Можливо, вам доведеться бути першим, демонструючи це на власному прикладі. Створюючи робочі угоди, запитайте членів команди, які домовленості їм потрібні, щоби бути відкритими та чесними одне з одним.
- *Коли ви припускаєтеся помилок, поділіться досвідом.* Замість зосередитися на звинуваченні (чи ще гірше — на осуді), допоможіть усім зрозуміти, що кожна помилка — це можливість

навчання. Заохочуйте членів команди ділитися з іншими власними можливостями навчання. Імовірно, вам доведеться показати це на своєму прикладі. Під час виникнення навчальних моментів звертайтеся до власних помилок.

- *Пізнайте одне одного як люди.* Заохочуйте членів команди бачити одне в одному людей з багатим досвідом та повноцінним життям за межами офісу. Створюйте ситуації, які допомагають людям ділитися своїми особистими історіями. Ви можете делікатно запитати про родину, друзів, захоплення чи інтереси. Спершу поділіться особистою інформацією про себе.

У цьому контексті **конфлікт** означає продуктивне використання спірних моментів для генерування нових ідей та пошуку різних рішень. Ефективні команди використовують різні погляди своїх членів, щоби конструктивно боротися з проблемами та вдосконалювати рішення. Пристрасні суперечки, якщо вони ведуться із взаємною повагою та спрямовані на пошук найкращого можливого результату, іноді є каталізатором, який уможливорює появу передових рішень. Продуктивний конфлікт передбачає сумніви щодо поточного стану, перегляд поглядів та подолання обмежувальних переконань.

На практиці: управління конфліктним спектром

Щойно члени команди починають довіряти одне одному, вони можуть скористатися силою продуктивних конфліктів. Іноді команда може навіть мати бажання піти на конфлікт для розв'язання складних питань, тоді як в інших ситуаціях вони захочуть його уникнути.

То як команда знає, що конфлікт є продуктивним? Коли конфлікт існує, бо в членів команди різні погляди, але вони все ж мають спільне зобов'язання досягти найкращих результатів для своїх клієнтів, зацікавлених сторін та організації. Якщо конфлікт додає певного дискомфорту, майте на увазі, що позитивний конфлікт завжди виникає з бажання змінитися на краще. Це допомагає зрозуміти

власну природну реакцію на конфлікт, яка є елементом вашої особистості. Це не є правильним або неправильним; це лише перевага. Ви можете замінити цю первинну перевагу, якщо знаєте про це¹. Конфлікт має тенденцію до поступової ескалації. Він може початися з простих відмінностей у поглядах та ідеях. Імовірно, особисті фактори й фактори навколишнього середовища можуть сприяти прогресуванню конфліктів у такі способи, які зумовлені потребою в самозахисті, самоперевірці чи підтримці глибоко вкорінених систем вірувань. Крім того, люди можуть звернутися до методів на кшталт створення коаліцій, протидії або навіть погрози. Важливим є визнання рівня конфлікту та реагування в способи, які рухають людей до спільного зобов'язання шукати найкращі можливі результати.

Для самоорганізованих команд важливо вміти ефективно брати участь та розв'язувати конфлікти. Деяким командам може знадобитися допомога у формі навчання тому, як брати участь у продуктивних конфліктах. Інші команди потребуватимуть допомоги у сприянні деескалації нездорового конфлікту. А в деяких випадках (наприклад, переслідування, ризик нанесення фізичної чи емоційної шкоди) можуть знадобитися негайні дії щодо втручання, розділення та застосування відповідних заходів².

У цьому контексті **відданість** означає: тільки-но команда розв'язує конфлікти та досягає консенсусу, її члени з відчуттям обов'язку ставляться до рішення, оскільки усвідомлюють, що колеги поважають їхні ідеї та погляди. Ми часто вживаємо фразу «не погоджуюсь і зобов'язуюсь», щоби показати, що члени команди досі мають змогу дотримуватися власної думки, але зобов'язуються перед своїми побратимами поважати рішення команди.

¹ Інструмент Томаса-Кілмана (TKI) є одним з інструментів для розуміння способів реагування на конфлікти.

² Ми закликаємо вас шукати додаткові ресурси, щоби зрозуміти моделі конфлікту. Ми запропонуємо для вивчення дві з них, але майте на увазі, що найкраща модель — це та, що допомагає вам та вашій команді увійти у фазу продуктивного конфлікту. Хоча Спід Ліз застосовував свою модель рівнів конфліктів у релігійних організаціях, її прийняли в спільноті прихильників Agile і про неї написали тут: <https://dzone.com/articles/agilemanaging-conflict>. Ще однією моделлю, яку варто розглянути, є дев'ятиступенева модель ескалації конфлікту Фрідріха Гласля: <https://www.mediate.com/articles/jordan.cfm>.

На практиці: сприяння консенсусу

Описані тут методи консенсусу — це способи збирання швидких та прозорих даних про те, на якому етапі перебуває команда, ухвалюючи рішення. Наведемо кілька прикладів того, коли ці методи можна використовувати:

- щоби визначити розклад подій спринту;
- щоби підтвердити домовленість команди розробників стосовно цілі спринту та беклогу спринту;
- під час сеансів співпраці із застосуванням проєктувальних та архітектурних підходів;
- у сеансах уточнення беклогу продукту заради визначення того, як розбити конструктивні особливості/функції на менші елементи беклогу продукту та/або як і коли проводити експерименти для отримання зворотного зв'язку та навчання.

Використання цих методів передбачає, що вже відбулося достатнє обговорення, яке гарантувало вивчення всіх ідей, і всіх було почуто.

Fist of Five (кулак) — це метод консенсусу, який дозволяє групам людей швидко зрозуміти, у чому вони погоджуються або не погоджуються. Люди вказують свій рівень підтримки, тримаючи одну руку і вказуючи цифру. Щоби прийняти рішення на основі консенсусу, може знадобитися кілька раундів обговорення та реєстрації. Ви можете використовувати часові рамки, щоб обговорення було зосередженим і не впадало в аналітичний параліч¹.

Римське голосування бере свій початок зі способу, у який римляни заявляли про свою волю на гладіаторській арені. Застосовуючи цю техніку, люди вказують на рівень підтримки за допомогою великого пальця.

- Великий палець угору означає «Я це підтримую».
- Великий палець убік означає «Я буду діяти за волею групи».
- Великий палець униз означає «Я цього не підтримую і хочу звернутися до групи».

¹ Для отримання додаткової інформації див. книжку: Jean Tabaka, *Collaboration Explained: Facilitation Skills for Software Project Leaders* (Addison-Wesley Professional, 2006)

Якщо всі великі пальці опущені або всі вгору, у вас є цілком чіткий консенсус. У разі змішаного голосування не забудьте дозволити висловитися людям, які опустили великий палець униз. Будьте обережні з рішеннями, коли всі великі пальці повернуті вбік, оскільки в команді може існувати штучна гармонія або нездоровий конфлікт, який потрібно дослідити на глибшому рівні¹.

У цьому контексті **відповідальність** означає, що члени команди підзвітні одне перед одним за взяті на себе зобов'язання. Потрібна мужність, щоби кинути виклик співробітнику через те, що він не виконує своїх обов'язків. Оскільки взяття на себе зобов'язань будується на довірі й одночасному усвідомленні того, що всі мають однакові цілі, конфлікт, притаманний цим розмовам, нівелюється та спрямовується на продуктивні дискусії про те, як рухатися вперед.

Члени команд, які притягають одне одного до відповідальності, діють ефективніше за керівництво, яке робить те саме. Ось чому обов'язок є тим основним елементом, який призводить до притягнення одне одного до відповідальності. Члени команди почуватимуться більш відповідальними за *власні* зобов'язання, ніж за зобов'язання, які беруть від їхнього імені інші.

Коли члени команди бажають притягнути одне одного до відповідальності, вони уможливають для команди встановлення високих стандартів і прагнення до їх досягнення. Це може проявитися у високій якості, кращих рішеннях, більшому обсязі навчання та більшій кількості інновацій.

Коли в команді існують відповідальність і підзвітність, можна зосередитися на загальних цілях та результатах.

Як просуваються команди?

Командна робота необхідна, але вона відбувається не автоматично і зазвичай не швидко. Більшість команд проходять низку етапів, об'єднуючись та створюючи активи, необхідні для ефектив-

¹ <https://www.mountaingoatsoftware.com/blog/four-quick-ways-to-gain-or-assess-team-consensus>

ної співпраці. Модель групового розвитку Брюса Такмена — одна зі способів розгляду змін, крізь які проходить команда, коли члени навчаються працювати разом (рис. 2.5). Деякі команди ніколи не просуваються далі нижчих рівнів, вони можуть повернутися на попередні етапи, якщо трапляються невдачі, приєднуються нові члени або ключові виконавці залишають команду.

П'ять етапів групового розвитку

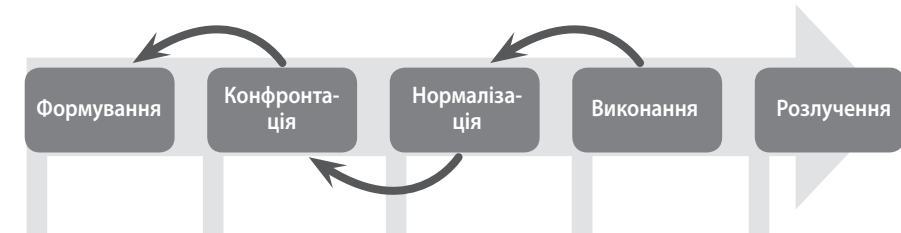


Рис. 2.5. Модель групового розвитку Такмена¹

Коли відбувається *формування* команди, її члени намагаються зрозуміти одне одного. Вони можуть бути стриманими у своїх взаємодіях, уникають конфліктів, поки працюють над встановленням меж.

Відповідно до того, як починають проявлятися відмінності (згадайте дискусію про особистості), а члени команди стають більш залежними одне від одного, виникають конфлікти. Такмен називає цей етап *конфронтація*.

Оскільки члени команди починають продуктивніше врегульовувати свої конфлікти та отримують краще розуміння того, як можна ефективніше працювати разом, вони, як кажуть, переходять на етап *нормалізації*. Учасники починають зосереджуватися на цілях команди та встановлюють стандарти якості й ефективності. Вони віддані команді й пишаються тим, що є частиною колективу, та роботою, яку вони виконують в цілому.

У процесі спільного розвитку команди її члени починають працювати більш рівномірно й автономно. Зараз вони на етапі *вико-*

¹ Для отримання додаткових відомостей про модель Такмена та її застосування до розвитку команди див. <https://project-management.com/the-five-stages-of-project-team-development>. Хоча модель Такмена показує, що команди прогресують лінійно, реальність набагато складніша. Та все ж ця модель дає уявлення про те, як розвиваються команди.

нання. Ви спостерігатимете пристрасні дебати серед членів команди, оскільки вони відповідальні й віддані створенню найкращих результатів, дотриманню високих стандартів та постійному самовдосконаленню. Хоча вони не можуть передбачити майбутнє, але випромінюють упевненість, що як команда можуть відповісти на будь-який виклик.

Розлучення відбувається, коли команду розпускають. Таке припинення діяльності може мати досить напружений характер, особливо коли це відбувається раптово і не є запланованим. Навіть найкращі команди можуть виявити, що цей етап дезорієнтує та демотивує.

Насправді, команди рухаються між цими етапами у відповідь на зовнішні події. Команда на етапі *виконання* може стикнутися з новими викликами, які вимагають від членів розширення їхніх навичок та знань, а невизначеність та конфлікти можуть повернути команду на етап *нормалізації* або навіть *формування*, якщо доведеться додавати нових членів.

На практиці: подолання проблем

Наведемо ситуацію, з якою ми стикаємося досить часто: команда, що була на етапі *виконання*, вирішила в останній ретроспективі спринту вдосконалити автоматизацію тестування, змусивши членів команди вивчати нові техніки та технології. Витіснені із зони комфорту, працівники намагаються засвоїти нові навички, а також виконати інші роботи під час спринту. Виникають конфлікти та розчарування, а прогнозована продуктивність членів команди падає.

Наступна демонстрація результатів спринту не вдається. Члени команди не лише не досягли прогнозованих результатів, але й значного прогресу в автоматизації тестів. У своїй ретроспективі спринту вони визнають, що не передбачали, наскільки це вплине на робочі стосунки їхньої команди або на зусилля, яких потребує автоматизація. Члени команди погоджуються бути більш уважними до цих змін, коли планують наступний спринт.

Сильніші команди з підтримкою керівництва та кращою культурою навчання швидше відновляться після такого відступу, тоді як команди в організаціях, що прагнуть звинуватити працівників у невдачах, ніколи не відновляться¹.

На практиці: наскільки стабільним має бути склад команди?

Стабільною є команда, учасники якої не надто поділяють свій час з іншими і склад якої з часом не дуже змінюється. Члени стабільної команди можуть налагодити кращі робочі стосунки, оскільки вони проводять більше часу разом, а це дає їм час розвивати довіру одне до одного. Такі команди також зазвичай більше знайомі з продуктом та сферою бізнесу, оскільки вони проводять більше часу, працюючи над одним продуктом.

Деяким організаціям важко підтримувати стабільні команди, оскільки їхні продукти не потребують швидких змін, тому й спеціалізована команда їм не потрібна. Крім того, члени команди можуть захотіти працювати над різними продуктами, вивчати нові технології та сфери бізнесу й працювати з різними людьми.

Навіть коли учасники команди можуть працювати в одній команді, з часом виникає потреба змін у наборі навичок, тому може стати на часі переведення людей між командами. Або працівниками можуть ділитися з іншими командами, коли всі вони потребують однакових рідкісних навичок.

Вірджинія Сатір створила п'ятиступеневу модель змін, яка описує вплив кожного етапу на почуття, мислення, результативність та фізіологію². Як показано на рис. 2.6 (с. 70), у момент змін команда вступає в *протидію* там, де вона відчуває зниження продуктивності. Ця зміна заважає стабільності та дружнім відносинам, і деякі люди можуть перебувати в режимі заперечення, уникання або навіть блокування. Хаос — це період нестабільної продуктивності, який триває, поки члени команди шукають спосіб упоратися

¹ Детальніше про культури навчання див. у книжці: Peter M. Senge, «*The Fifth Discipline: The Art & Practice of Learning Organization*» (New York: Doubleday/Currency, 1990).

² <https://stevenmsmith.com/ar-satir-change-model/>

зі змінами у своєму світі та врахувати їх. Коли вони знаходять цю ідею трансформації, то входять у процес *інтеграції*. Упродовж цього часу показники ефективності зростають, оскільки команда інтегрує цю нову зміну й навіть бачить, як це може бути корисним. Зрештою члени команди знаходять *новий статус-кво*, у якому вони повністю засвоїли зміну.



Рис. 2.6. Внесення змін спричиняє нестабільність та зниження продуктивності команди

Коли ми впроваджуємо заплановані зміни, як-от додавання члена команди, то сподіваємося, що новий статус-кво буде значущішим, ніж старий. Але, звичайно, немає гарантії, що ми досягнемо цього поліпшення.

То яка відповідь? Мета «досить стабільна». І вона полягає в тому, щоби Scrum-команда знайшла динамічну стабільність — могла швидко відновитися після змін, незалежно від того, заплановані вони чи ні. Коли ви впроваджуєте заплановані зміни, як-от додавання або видалення членів у наявній команді, а також коли створюєте нові команди, залучайте їхніх членів до процесу ухвалення рішень. Коли це можливо, нехай вони надають рішення. Якщо члени команди висловлюють свій погляд, то, найвірогідніше, неми-

нуче падіння продуктивності буде менш серйозним і триватиме коротший час, а відтак команда може відчувати вищу ефективність від змін.

Командна стабільність має бути чимось таким, що ви часто перевіряєте і до чого пристосовуєтеся. Спостерігайте за поведінкою та результатами. Отримуйте вказівки від членів команди. Поміркуйте, як кількість змін у складі вашої команди може сприяти менш бажаній поведінці та результатам.

Загальна порада: передавайте роботу командам, а не команди роботі. Чудові команди можуть навчитися нового, але їхнє створення вимагає часу й зазвичай є доволі складним. Коли команда знайде цю магію, скористайтеся нею, доки в цьому є сенс.

Характеристики продуктивних і адаптивних команд

Коли Scrum-команди досягають етапу виконання, ми очікуємо, що вони будуть продуктивними та пристосованими, «прагнучи бути якомога кращими. Це безперервна подорож до чогось кращого»¹. Коли вони зосереджуються на командних цілях, а не на окремих досягненнях, то можуть створити більш цінні результати. Продуктивна та пристосована команда досягає успіху або ж зазнає невдачі як команда.

На етапі *виконання* Scrum-команди зазвичай зосереджуються на таких аспектах:

- поліпшення якості та повноти «готового» приросту цінності продукту;
- поліпшення цінності, яку вони поставляють споживачам;
- усунення перешкод;
- зростання командних навичок та знань;
- поліпшення роботи за допомогою практичних удосконалень ретроспективи спринту.

¹ Див. книжку: Lyssa Adkins, «Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition» (Addison-Wesley Professional, 2010), с. 21.

Розглядаючи «готовий» приріст цінності продукту як спільну мету, Scrum-команда не зможе поставити придатний для релізу приріст цінності продукту, якщо вона не дала нічого цінного й не продемонструвала бізнес-гнучкості. Немає значення, що працівники нарізно здобули під час спринту,— як команда вони не змогли досягти цінного результату.

Для запобігання цьому члени команди пристосовуються до того, що вони роблять і як вони це роблять, коли виникають проблеми, щоби мати змогу досягти найціннішого результату відповідно до своїх спільних цілей. Продуктивні та пристосовані команди мають такі характеристики:

- упевненість, що вони як команда можуть розв'язати будь-яку проблему;
- прагнення досягти успіху в команді, а не індивідуально, беручи до уваги як короткострокові, так і довгострокові наслідки;
- націленість на результати, прагнення їх поліпшити та взяти на себе відповідальність за них;
- гіперпрозорість;
- прийняття ціннісних рішень на основі консенсусу;
- активний пошук продуктивного конфлікту;
- готовність вийти за межі своєї зони комфорту;
- постійне прагнення поліпшити свою ефективність та продуктивність;
- здатність плавно адаптуватися до несподіваних змін;
- узяття відповідальності за свої процеси, інструменти та взаємодію — якщо щось не працює, члени команди беруть на себе відповідальність змінювати свої практичні знання.

Які з цих характеристик ви прослідковуєте в командах, з котрими працюєте зараз або працювали раніше? Як це — відчувати себе частиною таких команд?

Підсумки

Сильні, стійкі, кросфункціональні та самоорганізовані команди забезпечують необхідну основу для швидкого успіху. Спільні цінності та цілі об'єднують команду й забезпечують її членів принципами, які вони можуть узгодити й використовувати для ухвалення рішень. Попри те, що всі команди будуть розвиватися у своєму власному темпі, створення середовища для забезпечення та розвитку самоорганізації, кросфункціональності та ефективної співпраці має важливе значення, якщо команди хочуть досягти високої ефективності та зрозуміти переваги Scrum.

Однак команди не є статичними. У процесі зміни свого складу та цілей їм потрібно переглянути та пристосувати власні цінності та цілі. Водночас вони будуть посилювати деякі аспекти своєї ідентичності, удосконалювати інші аспекти та розвиватися по-новому.

Заклик до дії

Розгляньте зі своєю командою наведені нижче запитання.

- Наскільки ясною є мета функціонування вашої команди і чи зрозуміла та сприйнята вона кожним членом команди?
- Як цінності Scrum керують рішеннями та методами роботи вашої команди?
- Як цінності та принципи Agile-маніфесту виявляються в щоденній взаємодії?
- Яких заходів ви можете вжити, щоби допомогти членам команди краще зрозуміти власну особистість та особистість своїх товаришів по команді?
- Які проблеми чи обмеження самоорганізації стримують вашу команду?
- Які активи ваша команда хоче збільшити для поліпшення співпраці?
- Скільки бізнес-процесів, цінностей продукту та знань користувачів має сенс тримати у вашій команді розробників для вашого продукту сьогодні?

- Що потрібно змінити, щоб скористатися перевагами динамічної стабільності?
- Які проблеми зараз найбільше дошкуляють? Визначте один-два експерименти, які допоможуть створити міцнішу основу команди. Для кожного експерименту обов'язково вкажіть бажані впливи та спосіб їх вимірювання.

Розділ 3

Постачання «готових» приростів цінності продукту

В основі Scrum лежить концепція «готовності». Scrum — це фреймворк, який забезпечує бізнес-гнучкість, і без «готового» приросту цінності продукту вона не може існувати: не існує такого поняття, як «частково готовий» або «майже готовий». Для досягнення рівня «готового» завдання потрібно, щоб Scrum-команди застосували командну роботу, емпіризм та гнучкий спосіб мислення, а також удосконалили командний процес для постачання продукту, що дозволить дати емпіричну оцінку мети спринту.

Після стадії *формування* нові Scrum-команди часто продовжують боротися, щоб кожного спринту створювати «готовий» приріст цінності продукту. Коли ми викладаємо курс «Професійний Scrum-майстер» (PSM)¹, одне з найпоширеніших запитань, яке ми отримуємо, таке: «Я розумію, чому “повна готовність” має значення, але як її досягнути, якщо не можна цього зробити сьогодні?»

Відповідно до того, як ідентичність команди зростатиме й створюватиме міцніший фундамент, члени команди одночасно працюватимуть над уточненням й удосконаленням процесу роботи. Фреймворк Scrum навмисно залишає за командою право визначати власну роботу. Пам'ятайте, що Scrum не має «найкращих практик», тобто найкраща практика чи інструмент — це те, що працює для вашого продукту та команди в цей момент. Ба більше, командна робота з часом повинна еволюціонувати відповідно до мінливих потреб та нових викликів.

Ця динамічна якість пояснює, чому Scrum-командам важливо спиратися на елементи командної роботи, емпіризм та гнучке мис-

¹ <https://www.Scrum.org/courses/professional-Scrum-master-training>

лення, щоби вони могли забезпечити «готовий» приріст цінності продукту. Краща співпраця та ефективніша самоорганізація вкажуть шлях до якіснішого та швидшого постачання продукту. Більша прозорість стосовно робочого циклу, прогресу та якості уможливить появу нових ідей та кращих адаптацій, які вдосконалять способи вироблення приросту цінності продукту, придатного для релізу. Очевидно, що цінності Scrum — цілеспрямованість та відданість — відіграють важливу роль у досягненні командою «готового» приросту¹.

Протягом цього розділу ми виділимо «позитивні практики», які слід урахувати. Однак майте на увазі, що ключовим моментом є розуміння мети практики. Чи це створення більшої прозорості щодо прогресу або якості? Чи сприяє це продуктивним конфліктам і більшій відданості командним рішенням? Чи це встановлення меж для створення більшої уваги та зменшення ризику? Шукайте першопричини своїх викликів і обирайте практики — або представлені в цьому розділі, або ті, що перебувають за межами цієї книжки,— які допоможуть вам надійно та послідовно забезпечувати «готовий» приріст цінності продукту.

Хоча команда розробників відповідає за досягнення рівня «готових» приростів цінності продукту, власник продукту та Scrum-майстер також сприяють успіху команди; існує внутрішнє протиріччя між рольовими відповідальностями, які доповнюють співпрацю². З цієї причини Розділ 3 призначений для всіх членів Scrum-команди та керівників, які надають підтримку та сприяють їм.

Що означає «повна готовність завдання»?

Критерій «повної готовності завдання» (DoD) описує спільне розуміння командою розробників діяльності, необхідної для створення придатного для релізу приросту цінності продукту. Воно часто опи-

¹ Детальніше про те, як цінності Scrum допомагають Scrum-командам ефективно використовувати цей підхід, читайте в цій публікації в блозі: <https://guntherverheyen.com/2013/05/03/theres-value-in-the-Scrum-values/>.

² Для кращого розуміння відповідальності ролей Scrum і їхньої спільної роботи прочитайте цю публікацію в блозі: <https://www.Scrum.org/resources/blog/accountability-quality-agile>.

сується конвенціями, стандартами й рекомендаціями та повинно відповідати будь-якому наявному організаційному визначенню або перевершувати його.

Критерій «повної готовності завдання» повинен бути доступним і відповідати продукту та команді. Коли приріст цінності продукту відповідає критерію «повної готовності завдання», власник продукту вирішує, відбудеться його випуск чи ні. Приріст цінності продукту може випускатися багато разів у спринті (наприклад, у режимі безперервної доставки або в моделі DevOps) або лише після декількох спринтів.

Критерій «повної готовності завдання» повинен забезпечити, щоби вся робота, яка виконується для отримання приросту цінності продукту, була інтегрована з попередніми приростами. Коли над продуктом працює кілька команд, їхня спільна робота повинна створити «готовий» приріст. Визначення окремих команд можуть відрізнитися, але вони мусять містити базовий мінімум розуміння якості та повноти, а це, звичайно, вимагає інтеграції для створення продукту, придатного для випуску.

Якщо це поняття настільки важливе, то чому Scrum не каже вам, що має бути частиною критерію «повної готовності завдання» (DoD)? Простіше кажучи, критерій «повної готовності завдання» настільки залежить від контексту, що його універсалізація неможлива. Тобто поняття «готовності роботи» дуже різне для команд, які працюють над мобільною грою, медичним пристроєм, системою управління польотом літака та міжнародною банківською системою. Використання продукту, окупність, безпека та вплив проблем на користувачів матимуть значення для формулювання критерію «повної готовності завдання».

Переваги критерію «повної готовності завдання»

Наявність надійного критерію «повної готовності завдання» та створення відповідного йому приросту цінності продукту мають наведені нижче переваги.

- *Прозорість процесу.* «Готовий» приріст цінності продукту забезпечує остаточну прозорість прогресу та постачання цінності. Коли команда не постачає «готового» приросту цінності

продукту в спринті, це ознака того, що їй не вистачає важливих знань чи можливостей. Ця невдача дає Scrum-команді можливість вивчити та вдосконалити власну роботу, свої інструменти та те, як вони працюють разом.

- *Прозорість приросту цінності продукту.* Коли є ясність щодо «повної готовності завдання», нікого зі Scrum-команди або зацікавлених сторін не повинні дивувати якість та повнота того, що показано в демонстрації результатів спринту.

Коли приріст цінності продукту відповідає критерію «повної готовності завдання», власник продукту може випустити його для клієнтів та отримати прибуток від інвестицій (ROI). Припущення щодо цінності й того, як зустрінуть продукт на ринку, можуть бути перевірені.

- *Прозорість щодо того, де ми перебуваємо.* Коли є чітке розуміння поняття «повної готовності завдання», власник продукту може краще спілкуватись із зацікавленими сторонами щодо завершеного завдання. Це дозволяє власнику продукту оновлювати прогнози, указуючи, коли, імовірно, будуть досягнуті додаткові цілі постачання продукту.
- *Прозорість щодо напрямку нашого руху.* Коли елементи беклогу продукту завершені раніше, ніж узгоджено критерій «повної готовності завдання», їхній розмір та складність можна використовувати для прогнозування майбутньої роботи подібного розміру та складності.
- *Прозорість щодо планування спринту.* Отримавши чітке розуміння необхідного для завершення певної роботи — отримання «повністю готового завдання», — команда розробників матиме краще уявлення про те, що вона може виробити під час спринта.

Як створити критерій «повної готовності завдання»

Хоча команда розробників є творцем критерію «повної готовності завдання», власник продукту може брати участь у його створенні, аби краще зрозуміти, що потрібно для появи приросту цінності продукту, придатного для випуску. Власник продукту може також допомогти визначити стандарти якості продукту, наприклад кіль-

кість одночасних користувачів, яких повинна підтримувати система, або максимально прийнятну тривалість транзакції.

Наведені нижче запитання корисні для стимулювання спільної дискусії щодо визначення критерію «повної готовності завдання».

- Що нам потрібно зробити, щоби допомогти людям, які підтримуватимуть продукт (наприклад, читабельний код, правила іменування змінних)?
- Як ми мінімізуємо технічні недоробки (наприклад, рефакторинг)?
- Як ми будемо тестувати продукт (наприклад, модульне тестування, функціональне тестування, регресійне тестування)?
- Яке тестування буде автоматизованим?
- Які дефекти необхідно усунути (наприклад, серйозність помилки, тип)?
- Як ми будемо задовольняти вимоги щодо продуктивності та масштабованості (наприклад, часу обробки транзакцій, одночасних користувачів)?
- Які стандарти розвитку приведуть нас до технічної досконалості?
- Як ми перевіримо відповідність стандартам розвитку нашої команди (наприклад, огляд роботи колегами)?
- Як ми перевіримо та забезпечимо якість даних?
- Як ми гарантуємо безпеку нашого продукту?
- Як ми забезпечимо відповідність нашого продукту нормативним, правовим або іншим стандартам?
- Що нам потрібно зробити, щоби задовольнити вимоги до брендингу?
- Що нам потрібно зробити, щоби забезпечити придатність нашого продукту для людей з інвалідністю (наприклад, стандарти доступності американського Закону про американців з інвалідністю [ADA])?
- Яка документація необхідна для випуску у виробництво (наприклад, онлайн-довідка, оновлення системи управління активами)?

На практиці: використання прийому «Зараз, наступний крок, майбутнє» для створення та вдосконалення критерію «повної готовності завдання»

Цей прийом здатний допомогти команді спільно створити початковий критерій «повної готовності завдання» та вдосконалити його з часом. Це також може посприяти командам, які намагаються подолати не дуже добре продуманий критерій «повної готовності завдання», сфокусуватися на тому, що в цей момент є найважливішим, та взяти продумані дії для просування вперед.

1. Команді розробників пропонується провести мозковий штурм для обговорення всього, що вона хотіла би бачити в складі критерію «повної готовності завдання» для отримання найякіснішого та найповнішого приросту цінності продукту. Члени команди припускають, що можуть зробити геть усе — жодних обмежень.
2. Члени команди спільно визначають елементи, які вони здатні виконати зараз, і переміщують їх у центр набору вкладених прямокутників, як показано на рис. 3.1.



Рис. 3.1. Використання прийому «Зараз, наступний крок, майбутнє» може допомогти командам розробників створити свій критерій «повної готовності завдання»

3. Члени команди визначають, якими будуть наступні вдосконалення, вони переміщують ці елементи в прямокутник із написом «Наступний крок».
4. Члени команди визначають бажані майбутні вдосконалення, які, імовірно, заберуть значну кількість часу або грошей, щоби перемістити їх у прямокутник із написом «Майбутнє».

До кінця цієї процедури команда повинна отримати оновлений критерій «повної готовності завдання», а також почати впорядковано впроваджувати вдосконалення, які потрібно буде отримати в наступних кількох спринтах. Може знадобитися більш детальна розбивка на елементи або зміна формулювань для більшої ясності, тому команда може вирішити зробити ці дії після сеансу¹.

Використання цілей спринту для досягнення «повної готовності завдання»

Scrum-команда дотримується мети спринту як спільної цілі для проведення спринту. Коли ви думаєте про успіх спринту, спершу запитайте: «Чи є у нас “готовий” приріст?», а потім: «Чи досягли ми мети спринту?»

Гарна ціль спринту має три характеристики: зосередженість, гнучкість та цілеспрямованість.

- Чи забезпечується достатня *зосередженість*, щоби ми могли отримати робочий приріст цінності продукту до кінця спринту?
- Чи забезпечується достатня *гнучкість*, щоби ми могли пристосувати наш план (тобто беклог спринту), коли виявляємо нові знання та складність структури?
- Чи забезпечується *відчуття цілеспрямованості* так, щоби можна було побачити цінність і значення нашої роботи? Чи надихає це зацікавлені сторони або принаймні створює зацікавлення щодо відвідування розбору спринту?

¹ Для отримання додаткової інформації та прикладів використання прийому «Зараз, наступний крок, майбутнє» для вдосконалення критерію «повної готовності завдання» див. <https://www.Scrum.org/resources/blog/improving-your-definition-done>.

Створення гарних цілей спринту

Не існує ідеальної формули для створення гарної цілі спринту. Кожна ціль спринту дуже залежна від контексту, і Scrum-команди повинні проєкспериментувати, щоб знайти, що для них підходить, і пристосуватися відповідно до зміни контексту (рис. 3.2)¹.

Ціль спринту	Контекст
Скоротити час завантаження звіту до 2 секунд	Команда, яка працює над звітами, керованими базами даних, необхідними для розв'язання архітектурних проблем і досягнення продуктивності
Наприкінці спринту показати цільову сторінку сайту з використанням виробничої інфраструктури	Команда, яка працює над новим продуктом, необхідним для перевірки моделі інфраструктури та процесу розгортання
Розширити варіанти способу оплати	Продукт підтримує одного постачальника карток і повинен містити інші картки, PayPal, Amazon Pay та інші варіанти оплати. Є багато невизначеності, тому мети можна досягнути, якщо існуватиме змога запропонувати ще один варіант

Рис. 3.2. Приклад цілі спринту

Наведемо поради, які допоможуть вам поліпшити цілі спринту.

- *Уникайте комбінованих цілей спринту.* Комбіновані цілі спринту, як-от «Побудувати X, i Y, i Z», розпорошують фокус Scrum-команди і стоять на заваді гнучкості. Іноді комбіновані цілі спринту з'являються, коли команди працюють над безліччю не пов'язаних між собою ініціатив або коли намагаються взяти на себе занадто багато роботи в спринті. Комбіновані цілі спринту залишають мало місця для роботи в процесі навчання Scrum-команди.

¹ Для отримання додаткової інформації про створення гарних цілей спринту див. <https://www.agilesocks.com/creating-good-sprint-goals/>.

- *Не намагайтеся здійснювати мікроменеджмент за допомогою цілей спринту.* Вам не потрібно додавати до своєї цілі спринту всі елементи беклогу продукту (PBI). Насправді ціль спринту, у якій сказано «виконати всі елементи беклогу продукту», еквівалентна відсутності цілі спринту як такої. Якщо ви хочете, щоб самоорганізовані, наділені повноваженнями команди були ефективними, то повинні вірити, що люди прагнуть робити все від них залежне. Якщо Scrum-команда досягне цілі спринту до його закінчення, члени команди зрозуміють, що ще вони можуть удіяти, аби зробити значний внесок.
- *Зробіть ціль спринту вимірною.* Коли ви дістанетеся кінця спринту, уся команда повинна дійти згоди в питанні, чи досягнуто ціль спринту. Щоби переконатися, що ціль спринту чітка, запитайте під час планування спринту: «Як ми дізнаємося, чи досягли цілі спринту?», а потім подумайте, чи можна визначити об'єктивні критерії.
- *Забезпечте консенсус команди щодо цілі спринту.* Під час планування спринту використовуйте метод консенсусу, щоб підтвердити розуміння та відданість кожного учасника досягненню цілі спринту.
- *Створюйте цілі спринту, які здійснюють вплив на діяльність.* Люди хочуть виконувати значущу роботу, яка має вплив. Щоб у цьому переконатися, спробуйте вжити низку заходів.
- *Зробіть бізнес орієнтованим на користувача.* Що зможе зробити користувач, коли ви впровадите цю функцію? Чого зможе досягти бізнес-сфера за допомогою цього вдосконалення?
- *Зосередьтеся на тестуванні бізнес-припущень та отриманні відгуків.* Важко зрозуміти, що користувачі насправді потребують або готові робити (адже навіть вони самі цього не знають). Власник продукту потребує раннього зворотного зв'язку для перевірки припущень щодо цінності.
- *Зосередьтеся на зниженні ризику.* Доказ використання нової технології або, можливо, нової архітектури є важливою частиною зниження ризику. Якщо ви дізнаєтеся, що технологія не буде відповідати вашим потребам у продуктивності, безпеці чи масштабованості, то можете змінити напрям. Що раніше ви це зробите, то дешевшою буде вартість зміни.

Використання цілі спринту для ефективної щоденної Scrum-наради

Щоденна Scrum-нарада призначена для того, щоби дозволити команді розробників перевірити прогрес, адаптувати свій план на підґрунті нових знань та виявити будь-які перешкоди для досягнення цілі спринту. Щоденна Scrum-нарада пропонує команді розробників можливість підтвердити одне одному свою спільну відповідальність.

Ефективна щоденна Scrum-нарада має низку характеристик.

- *Сприяє самоорганізації та співпраці.* Команда розробників сприяє щоденній Scrum-наradі, і вони оновлюють власний беклог спринту. Якщо всі зосереджені на своїй відданості цілі спринту і бачать прогрес, досягнутий на шляху її досягнення, існує більше можливостей для участі та співпраці.
- *Створює фокус і зменшує втрати.* Щоденна Scrum-нарада має бути схожою на швидке спільне планування. Чітка ціль спринту допомагає всім зосередитися на призначенні щоденної Scrum-наради і вкластися у 15-хвилинний таймбокс.
- *Сприяє прозорості та спільному розумінню прогресу в отриманні цінних результатів.* Щоденна Scrum-нарада — це не просто оновлення статусу завдань, над якими працює кожна людина¹. Унаслідок зосередження на отриманні «готового» приросту цінності продукту, який досягає цілі спринту, команди розробників нагадують про її призначення та обов'язки. Члени команди можуть досягти прогресу в контексті всього спринту. Якщо буде виявлено нову роботу, яка загрожує цілі спринту, вони можуть обговорити та адаптувати план. Якщо проблеми сповільнюють прогрес, члени команди можуть помітити їх завчасно й пристосувати план.

До кінця щоденної Scrum-наради кожен член команди розробників повинен розуміти прогрес у досягненні цілі спринту, знати поточний план та ймовірність її досягнення. Крім того, члени команди повинні знати, що вони планують досягти впродовж наступних 24 годин і як вони будуть працювати разом для цього.

¹ Щоденна Scrum-нарада не є поточною нарадою про хід виконання робіт; докладніше про це див. на <https://www.Scrum.org/resources/daily-Scrum-not-status-meeting>.

Знову ж таки, використання методу консенсусу в кінці щоденної Scrum-наради може стати корисним, аби гарантувати, що голоси всіх членів почуто і їхні думки збігаються.

Приведення елементів беклогу продукту до «готового» стану раніше в межах спринту

Є декілька методів, які можуть допомогти вам швидше привести елементи беклогу продукту до «готового» стану.

- *Застосуйте критерій «повної готовності завдання» до кожного елемента беклогу продукту.* Замість чекати до кінця спринту, щоби проаналізувати критерій «повної готовності завдання» та виконати дії, застосуйте це визначення до кожного елемента беклогу продукту в процесі роботи над ним і досягніть того, щоби він був по-справжньому «готовий», перш ніж переходити до чогось нового. Цей підхід має помітну побічну перевагу: він може дозволити вам фактично здійснити випуск приросту цінності продукту до закінчення спринту.
- *Розбийте елементи беклогу продукту на менші фрагменти.* Менші елементи, найвірогідніше, матимуть менші складності та невизначеність, а також потребуватимуть менше зусиль.
- *Спробуйте одноденний спринт.* Коли Scrum-команда намагається розбити роботу на менші кроки та співпрацювати над тим самим фрагментом, одноденний спринт може змусити їх кинути виклик своїм же припущенням. Цей вид спринту, найвірогідніше, не варто застосовувати постійно, але він може бути корисним заходом щодо вдосконалення команди, він допомагає зруйнувати переконання, котрі обмежують роботу, та заохотити до творчих підходів задля отримання цінності. Для початку ви можете запропонувати такий виклик: «Чи є якась невеличка цінність, яку ми можемо поставити користувачам за день?» Зменшуючи таймбокс, ви створюєте фокус і терміновість, змушуючи членів команди випробувати щось нове. Вони менше турбуватимуться про свою спеціалізацію та досвід, а замість цього більше думатимуть про внесок у результати команди. Ризик одноденного спринту невеликий: якщо він не спрацює, втрачено буде лише день. Але

потенційна вигода може бути суттєвою. Ідеї, що з'являються, часто є доволі переконливими і призводять до багатьох практичних зобов'язань щодо наступного спринту звичайної тривалості.

- *Залучити команду розробників до уточнення беклогу продукту.* Команда розробників повинна розуміти, що вони створюють, аби мати можливість задовольнити бізнес-потреби. Ці розмови можуть розпочатися до початку спринту, коли відбудуться заходи, пов'язані з уточненням беклогу продукту. Крім того, члени команди розробників матимуть знання, які допоможуть у розбитті елементів на дрібніші фрагменти і в такий спосіб, щоби забезпечити більшу гнучкість та креативність у постачанні цінних результатів. Залежності можна виявити та потенційно усунути до того, як елементи беклогу продукту будуть заплановані в спринті.

На практиці: використання Scrum-дошки для візуалізації прогресу

Scrum-дошка — це візуальне відображення беклогу спринту та його прогресу (рис. 3.3). Команди розробників можуть фізично тримати Scrum-дошку в просторі свого перебування, щоби вона була легкодоступною та розташовувалася в їхньому полі зору. Для членів команди, що працюють віддалено, доступно багато цифрових інструментів (як безкоштовних, так і платних), котрі задовольняють ці потреби. Команда розробників оновлює Scrum-дошку щоразу, коли отримує нову інформацію про елементи беклогу продукту, над якими працює.

На рис. 3.3 елементи беклогу продукту фіксуються на більших картках, і відповідний розподіл дій після їхнього завершення відображається в тому ж рядку. Стовпці на дошці демонструють поточний стан. Будьте обережні зі Scrum-дошками, які спроектовано в такий спосіб, щоби просувати каскадний або ізольований метод роботи під час спринту так, аби люди зосереджувалися лише на своєму відрізку роботи. Прикладом може бути напрям «Тестування», що, як вважають, «належить» людям із команди розробників, котрі спеціалізуються на тестуванні.

Елемент беклогу продукту (PBI)	Список завдань (To-Do)	Незавершене завдання (In-Progress)	Повністю готове завдання (Done)
			
	 	 	
	  		
	   		

Рис. 3.3. Приклад Scrum-дошки

Коли Scrum-команда ефективно використовує Scrum-дошку, ви, найвірогідніше, почуєте наведені нижче типи розмов.

- Члени команди обговорюють, як спільно працювати над елементами беклогу продукту, щоби швидше дійти до рівня «повністю готового завдання».
- Члени команди запитують, чи можуть вони допомогти в роботі над незавершеним елементом беклогу продукту, а не у виборі та початку роботи над новим елементом беклогу продукту.
- Один із членів команди дає поради іншому на основі свого досвіду роботи в певному сегменті продукту.
- Команда комплексно обговорює, хто хотів би працювати над певними елементами беклогу продукту, беручи до уваги поточні навички та те, як люди хотіли б розвивати свої навички.
- Члени команди висловлюють занепокоєння тим, що не можуть досягти цілі спринту та перегляду його обсягу: що можна вилучити із беклогу спринту, що можна розбити на менші фрагменти, щоби більше зосередитися на цілі спринту.
- Члени команди підтверджують, що нову роботу, яка загрожує цілі спринту, не будуть додавати до беклогу спринту, а спрямують власникові продукту.
- Команда загалом ставить запитання, чи ніщо не заважає роботі, й обговорює, що робити для поліпшення потоку.
- Команда загалом обговорює, як чинні зобов'язання ретроспективи спринту впливають на роботу команди.
- Команда загалом визнає нову залежність та обговорює, як її усунути.

- Команда загалом підтверджує, що існує відповідність між критерієм «повної готовності завдання» і самим завданням, коли елемент беклогу продукту перенесено у стовпець «Повністю готове завдання».

На практиці: використання графіка згоряння завдань у спринті для відстеження прогресу

Графік згоряння завдань у спринті відслідковує елементи беклогу продукту відповідно до їхнього завершення, а це означає, що вони відповідають критерію «повної готовності завдання» (рис. 3.4). Зверніть увагу, що цей графік відрізняється від графіка згоряння завдань у спринті, який відстежує завдання або години, оскільки основний фокус приділяється тому, що елементи беклогу продукту приводяться у відповідність до значення поняття «повністю готове завдання». Такий графік допомагає Scrum-команді візуалізувати свою роботу під час спринту.

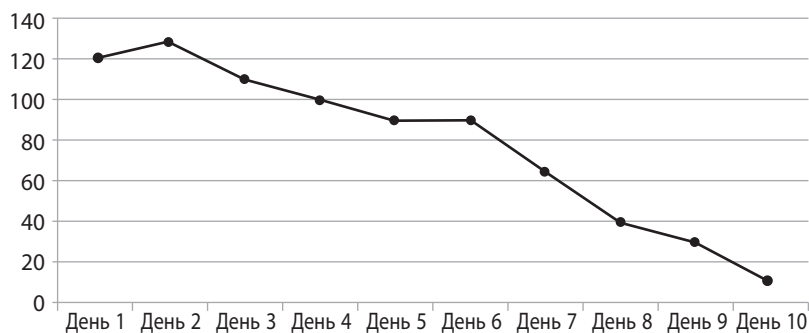


Рис. 3.4. Приклад графіка згоряння завдань у спринті

Графік згоряння завдань у спринті може бути використано в ретроспективі спринту, щоб розповісти про швидкість заповнення елементів беклогу продукту та визначити потенційні проблеми та напрями для вдосконалення. Це допомагає Scrum-команді ставити запитання, які ведуть до глибшого розуміння під час ретроспективи спринту.

- Що відбувалось у певні дні, коли лінія згладжувалася й це перешкоджало завершенню виконання елементів беклогу про-

дукту? Які зміни можуть зменшити втрати в нашому процесі та забезпечити кращий потік?

- У якому місці процесу роботи ми відкрили для себе нові завдання? Як ми відреагували на це відкриття? Що ми зробили б інакше в майбутньому?
- Що відбувалося в певні дні, коли лінія була крутішою і це допомагало нам завершити більше елементів беклогу продукту? Чого ми можемо навчитися з цього руху вперед? Члени команди можуть також поглянути на графіки вигоряння завдань із попередніх спринтів і побачити, як змінювався потік із плином часу. Чи мали бажаний ефект зроблені ними поліпшення? Чи з'явилися нові виклики? Візуалізація допомагає згадати деталі з минулого та належно пристосуватися.

Обмеження кількості незавершених завдань

Коли команда розробників застосовує до кожного елемента беклогу критерій «повної готовності завдання», вона також може захотіти обмежити кількість елементів, робота над якими триває. Цей прийом, який запроваджує принципи ощадливого виробництва (*Lean Principles*) для поліпшення робочого процесу, допомагає поліпшити співпрацю та навчання. Наприклад, команда розробників може встановити ліміт в кількості двох завдань для стовпця «Незавершене завдання». Коли завершується виконання двох елементів беклогу продукту, член команди розробників, який має можливість, зосереджується на тому, щоби допомогти команді досягти прогресу в роботі над їхнім завершенням, а не починати щось нове.

Цей підхід може бути складним для команд, члени яких намагаються зрозуміти, як найкраще застосувати свої навички та знання в процесі спільної роботи. Проте вони не вийдуть за межі цієї проблеми, якщо не спробують. Лімітування кількості незавершених завдань створює межу, яка змушує їх кидати виклик своїм старим способам роботи, проявляти творчі здібності до самоорганізації навколо роботи та бажання випробувувати щось нове.

Припустімо, наприклад, що під час щоденної Scrum-наради Боб заявляє, що хоче розпочати роботу над новим елементом беклогу продукту, оскільки проміжні елементи не мають жодних потреб

у фронтенд-проектуванні та розробці (його спеціалізація). Команда розробників може вказати на ліміт незавершених завдань і заохотити Боба взяти участь у діяльності, яка сприяє приведенню «незавершеного» елемента беклогу продукту до стану «готового», наприклад у проведенні певних тестів, оновленні документації або парному програмуванні з кимось.

Чому ми хочемо це зробити? Якщо коротко, то з огляду на закон Літтла¹, ми не будемо вдаватися до його детального пояснення, але по суті — що більше завдань, над якими ви працюєте в будь-який момент часу (у середньому), то більше часу вам знадобиться для закінчення кожного з них (у середньому). Коли робота триває довго, людям, здається, притаманно якомога швидше розпочинати робити щось нове, щоби вони могли це «закінчити вчасно», незалежно від того, робота над чим триває зараз. Результат полягає в тому, що, зрештою, завершення завдання триває довше з огляду на безліч форм витрат, породжених цією ситуацією.

Щоби загалом зробити більше, потрібно зосередитися на одночасному виконанні меншої кількості завдань. Позбавляйтеся від перешкод, які вповільнюють вас, а не вводьте більше завдань у свій робочий процес.

Вимірювання та аналіз потоку виконання

Ключем до вдосконалення процесу роботи команди є забезпечення прозорості всього, що насправді відбувається під час процесу. Об'єктивні дані можуть допомогти визначити потенційні закономірності, які можна прогледіти, просто покладаючись на спостереження або намагаючись згадати, як проходила робота.

Наведені нижче заходи та методи можуть допомогти забезпечити прозорість у процесі роботи команди.

- *Відстежуйте тривалість циклу.* Ви можете мати кілька «циклів», які визначаєте та вимірюєте в процесі діяльності. Для простоти пояснення визначимо час циклу як тривалість від моменту початку роботи над елементом беклогу продукту до моменту, коли він готовий до релізу. (*Примітка:* деякі

¹ Прочитайте цей довідковий матеріал заради додаткової інформації про те, як закон Літтла застосовується до Scrum-команд: <https://www.Scrum.org/resources/littles-law-professional-Scrum-kanban>.

команди визначають кінцеву точку як фактичний випуск продукту, особливо якщо це складник їхнього критерію «повної готовності завдання». Робіть те, що має сенс для вашої команди.)

- *Використовуйте графік вигоряння завдань або графік пропускної здатності.* Відстежуйте кількість елементів беклогу продукту, виконаних за одиницю часу. Деякі команди можуть мати вигоряння цих елементів на базі їхнього розміру (наприклад, якщо вони використовують відносні одиниці оцінки (сторі-пойнти) на позначення розмірності). Ключовим тут є те, що команда розуміє, коли окремі елементи беклогу продукту (не завдання) доходять до етапу «повної готовності».
- *Відстежуйте свої незавершені завдання.* Відстежуйте загальну кількість елементів беклогу продукту, які було розпочато й на певний момент часу ще не завершено. Ці елементи в розробці, але ще не дають цінності.
- *Відстежуйте заблоковані елементи беклогу продукту.* Коли робота над елементом беклогу продукту запускається, але потім, у процесі, зупиняється, такий елемент беклогу продукту вважається заблокованим. Крім того, команда може відстежити, як довго він залишається в заблокованому стані, та виявити, що до цього призвело.
- *Зверніть увагу на тенденції!* Сенс збирання цих даних полягає в пошуку тенденцій. Де тенденції вказують на можливості для поліпшення процесу?

На практиці: поліпшити прозорість та ефективність потоку за допомогою Канбан

Канбан — стратегія оптимізації потоку цінності для зацікавлених сторін за допомогою процесу, який використовує візуальну обмежену систему витягання незавершених завдань.

Потік — це рух цінності, орієнтованої на споживача, по всій системі розробки продукту. Канбан оптимізує потік шляхом поліпшення загальної ефективності, результативності та передбачуваності процесу¹.

¹ Для отримання додаткової інформації про використання Scrum із Канбан див. <https://www.Scrum.org/resources/kanban-guide-Scrum-teams>.

Орієнтована на потік концепція Канбан та зосередженість на прозорості й візуалізації в поєднанні з фреймворком Scrum можуть допомогти команді розробити методологію з метою оптимізації засобів для постачання цінності клієнтові. Scrum-команди досягають оптимізації потоку, додаючи до свого Scrum-процесу такі чотири практики:

- візуалізація робочого процесу;
- лімітування кількості незавершених завдань;
- активне управління незавершеними робочими завданнями;
- перевірка та пристосування визначення поняття «робочий процес».

Канбан акцентує увагу на більшій прозорості робочого процесу та визначає мінімальний набір показників потоку, які використовуються для активного управління виконуваними робочими елементами та забезпечення перевірки та адаптації, орієнтованої на потік. Він використовує теорію черг та закон Літтла, щоби показати, що для виконання більшої кількості роботи необхідно працювати менше. Так, це звучить божевільно, але вражає своєю правдивістю. Зрештою, Канбан забезпечує більший емпіризм, надаючи ще кілька практик та метрик, які допоможуть вам віднайти свій найкращий процес у межах Scrum-фреймворку¹.

Нарощування якості від самого початку

Якість охоплює досвід користувача стосовно продукту, зокрема те, наскільки добре він поєднує гнучкість, придатність до ремонту, ефективність та оперативність. Scrum-команди потребують раннього та постійного розуміння якості продукту, щоби вони могли пристосувати його до потреб замовника. На практиці ця рушійна сила спрямовує команди до застосування підходу «спершу тестування», щоби від самого початку зосередити увагу на якості.

Застосування підходу «спершу тестування» означає: Scrum-команда визначає свої тести перед створенням рішення, щоб отри-

¹ Для отримання додаткової інформації про Канбан і Scrum див. <https://www.Scrum.org/resources/blog/understanding-kanban-guide-Scrum-teams>.

мати швидкий відгук про те, що рішення, яке розробляється, працює належно. Це призводить до більшого покриття тестами, а також сприяє вищій якості, оскільки все, що побудовано, також буде перевірено. Наведемо практики, що використовують підхід «спершу тестування», деякі з них поєднуються з автоматизацією.

- *Керована тестами розробка (TDD)*, у якій перед написанням коду створюються автоматизовані модульні тести для керування розробкою програмного забезпечення та примусового зняття залежностей. Робота складається:
 - з написання «одиночного» модульного тесту, який описує аспект функції;
 - запуску тесту, який повинен провалитися, оскільки в програмі відсутня ця функція;
 - простого написання коду до «достатнього рівня», який уможливив би проходження тесту;
 - з рефакторингу коду, поки він не почне відповідати очікуваним стандартам кодування (як зазначено в критерії «повної готовності завдання»).

Цей процес повторюється, створюючи набір автоматизованих модульних тестів, який із часом зростає. Потім цей набір тестів можна запустити в будь-який час, щоби підтвердити, що продукт усе ще працює належно¹.

- *Керована поведінкою розробка (BDD)*, яка базується на загальних принципах керованої тестами розробки (TDD) та додає ідеї предметно-орієнтованого проектування (DDD) для створення автоматизованих тестів, що перевіряють конкретну поведінку, яку ви хочете бачити у вашому застосунку. Функціональні компоненти будуються поступово, керуючись бажаною поведінкою. Розробники, які застосовують BDD, зазвичай використовують базову мову в поєднанні з мовою предметно-орієнтованого проектування, щоб описати мету та переваги свого коду².
- *Розробка на основі приймального тестування (ATDD)* — практика спільної роботи, у якій користувачі та команда

¹ <https://www.agilealliance.org/glossary/tdd/>

² <https://www.agilealliance.org/glossary/bdd/>

розробників визначають критерії приймання робіт до створення будь-якої функціональності. Ці тести представляють погляд користувача й описують, як функціонуватиме система. У деяких випадках команда автоматизує приймальні тести. Цей підхід вважає пріоритетною перевірку ділової функціональності¹.

Автоматизація та «повна готовність завдання»

Нині автоматизація стає все більш значущою в розробці продуктів завдяки швидкості, з якою багатьом підприємствам потрібно постачати цінність споживачам, складності та масштабу продуктів і систем, а також важливості якості у світі, що стає дедалі залежнішим від технологій. Автоматизація роботи може здатися командам без досвіду роботи складним завданням. Автоматизація вимагає значних зусиль, але переваги її використання того варті:

- зменшення ручних зусиль членів команди;
- зменшення стримувальних факторів, пов'язаних із конкретними наборами навичок;
- раннє виявлення дефектів у циклі постачання;
- знижена ймовірність помилки людини;
- фокусування навичок та енергії членів команди на розв'язанні нових складних проблем та здобутті нових знань.

Загалом ці переваги сприяють здатності швидше поставити цінність із вищою якістю, а також дозволяють людям швидше вчитися. Якщо ви створюєте новий продукт, не відкладайте автоматизацію на потім. Простіше, коли ви починаєте з невеликого продукту й розбудовуєте автоматизацію відповідно до його зростання,— тоді ви починаєте отримувати переваги раніше. І якщо ви потрапили в ситуацію, коли продукт уже досить великий, вам доведеться розбивати його на менші фрагменти й працювати з ними.

Деякі команди знають, що їм слід автоматизуватися, але вони завжди «занадто зайняті», щоби досягти будь-якого прогресу в до-

сягненні цієї мети. Рішення такої дилеми полягає в тому, щоби взяти на себе менше роботи в спринті, а решту зусиль докласти до автоматизації, щоби створити додаткові можливості для виконання більшої кількості роботи в майбутньому. Команда розробників повинна співпрацювати з власником продукту, щоби досягти компромісу стосовно зменшення прогресу зараз для досягнення більшого прогресу пізніше.

До автоматизації часто звертаються в наведеному нижче порядку.

1. Контроль версій.
2. Автоматизована збірка.
3. Автоматизований тест.
4. Автоматизоване компонування пакетів програм.
5. Автоматизоване розгортання.

Діяльність з автоматизації та контроль якості можуть бути частиною критерію «повної готовності завдання» команди розробників. Можна розглянути кілька різних рівнів тестування, як показано на рис. 3.5. Зрештою, що більше автоматизованих рівнів, то більшу впевненість ви маєте щодо якості, яку вкладаєте в продукт.

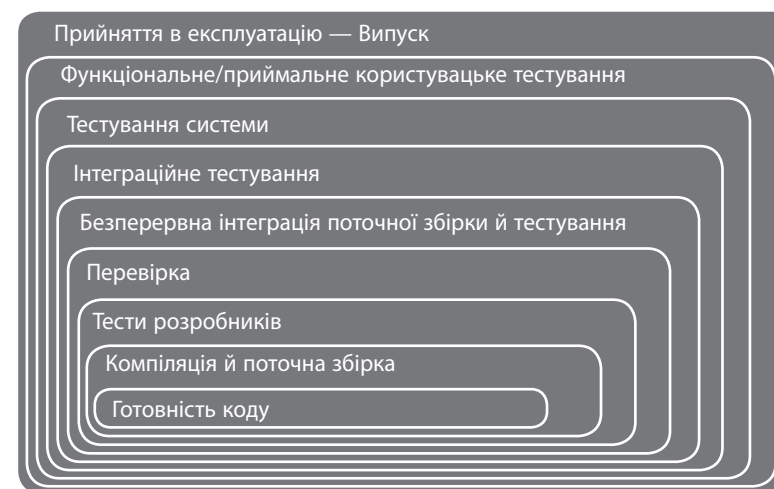


Рис. 3.5. Рівні тестування

¹ <https://www.agilealliance.org/glossary/atdd/>

В Agile-маніфесті говориться, що гнучкі команди цінують робоче програмне забезпечення, а не вичерпну документацію. Це означає, що команда повинна доводити розробку програмного забезпечення до збірки та тестування. У великих масштабах. По суті, увесь час. В ідеалі команді потрібно створювати збірку щоразу, коли вона робить зміни в репозиторії вихідного коду. Але збірка — це лише відправна точка, команда також повинна проводити тестування кожного разу, коли вносяться зміни. Не просто модульні тести, а й функціональні, регресивні, тести на продуктивність та масштабованість — усі на основі API. І не тільки в простих середовищах, але й тих, які максимально схожі на виробничі. Коли команди роблять це, вони забезпечують кращий код¹.

Технічна досконалість є важливою здатністю команд розробників, яка сприяє зростанню, а практики автоматизації — основний складник цих зусиль. Безперервне постачання (CD) та рух DevOps — сфери, які слід досліджувати надалі².

DevOps

Існує велика плутанина стосовно того, що таке DevOps. Багато джерел зосереджуються на технічних практиках, деякі з яких уже було пояснено (наприклад, безперервна інтеграція, автоматизоване тестування, безперервне постачання, інфраструктура як код [IaC]). Але важливо не випускати з уваги всеосяжну мету DevOps: він руйнує бар'єри між операціями та розвитком для досягнення підвищеної гнучкості³.

Перегляд коду

Під час перегляду коду член команди розробників (або декілька членів) аналізує роботу, виконану іншим учасником команди, щоби забезпечити перевірку якості. Команди, що використовують

¹ <https://www.Scrum.org/resources/blog/what-devops-taught-me-about-agile>

² Див. Jez Humble, David Farley, *Continuous Delivery*. Ця книжка — чудовий ресурс для вивчення ширших технічних тем, зокрема управління конфігурацією, постійної інтеграції, конвеєрів розгортання, управління інфраструктурою та середовищами, тестування та управління даними.

³ Детальніше про те, як гнучкість та DevOps доповнюють одне одного, див. <https://www.Scrum.org/resources/convergence-Scrum-and-devops>.

цю практику, часто мають контрольний список, який вони застосовують під час перегляду коду і який, імовірно, міститиме критерій «повної готовності завдання». Крім підвищення якості роботи, перегляд коду — це чудова навчальна можливість для людей продовжувати розвивати свої знання про систему, а також знання та навички як розробників¹.

Метрики якості

Вимірювання складників продукту, які є показниками якості, допомагає створити прозорість, щоби Scrum-команда могла перевірити та адаптувати вплив на якість способу побудови продукту та вибору, який вони роблять. Деякі з цих метрик генеруються автоматизованими інструментами, інші можна відстежувати вручну. Пам'ятайте, що з метриками тенденції більш інформативні, аніж конкретні точки даних. Ба більше, Scrum-команди захочуть перевірити кілька показників, беручи до уваги безліч змінних (як відомих, так і невідомих), які можуть впливати на метрику. Зауважте, що ці показники призначені для використання Scrum-командою. Щоби не порушувати та підтримувати прозорість, вони не повинні використовуватися для вимірювання її продуктивності людьми за межами команди, а також їх не можна стимулювати.

Наведемо деякі загальні метрики якості.

- *Показники покриття коду* вказують на те, яку частину бази коду продукту охоплюють автоматизовані тести. Це загальний пункт даних для команд, які використовують керовану тестами розробку. Хоча високий відсоток не гарантує, що код «написано як слід» і що тести підтримуються, цей захід є корисним для команд.
- *Показники складності* допомагають команді розробників зрозуміти ступінь ремонтпридатності коду. Цей фактор якості відображає довгострокову ефективність та продуктивність роботи Scrum-команди. Приклади того, як виміряти складність, містять цикломатичну складність, глибину

¹ Для ознайомлення з поясненням, чому перегляди коду насправді економлять час, див. <https://www.atlassian.com/agile/softwaredevelopment/code-reviews>.

наслідування об'єктів, зв'язок між класами, вкладеність та метрики Холстеда.

- *Показники стабільності збірки* — провідний індикатор загальної стабільності продукту. Вони вказують на те, чи стабільний процес збірки, та викривають проблеми, пов'язані з якістю коду. Стабільність збірки можна виміряти кількістю днів від моменту останнього червоного статусу (збій збірки) та кількістю послідовних днів, у які виникало повідомлення з червоним статусом.
- *Показники дефектів* дозволяють оцінити якість товару. Виправлення помилок безпосередньо у виробничому середовищі дороговартісні, і вони часто перешкоджають здатності команди поставляти нові функціональні особливості й можливості. Помилки можуть мати значний вплив на користувачів, зрештою впливаючи на бренд, репутацію та фінансовий результат.

Крім самих заходів, важливі також і тенденції.

- Скільки дефектів виявлено у виробництві та скільки дефектів усунуто до виробництва і як це співвідношення змінюється з часом?
- Скільки часу в середньому займає виправлення виробничого дефекту і як це змінюється з часом?
- Якою є середня критичність дефектів (тобто їхня вартість або вплив на бізнес та клієнтів) і як це змінюється з часом?
- Скільки дефектів повторюється і як це змінюється з часом?
- Як довго дефекти лишаються відкритими і як це змінюється з часом?

Пропрацювання технічного боргу

Технічний борг — це відкладена робота над продуктом, яка часто виникає внаслідок рішень, прийнятих командою розробників з метою обміну якістю на швидкість. Технічний борг можна розглядати як ламкий або важкозамінний код. Зауважте, що вдаватися до технічного боргу — це не обов'язково погана річ, доки існує реальний

план його «погашення». Як і в ситуації з фінансовим боргом, іноді може статися, що прибуток перевищуватиме виплачені відсотки. Наприклад, створення прототипу для тестування чогось на ринку перед тим, як інвестувати у функціональну серверну частину системи, може бути прийнятним рішенням. Технічний борг повинен бути прозорим, а вплив його створення на майбутню працездатність продукту має приносити цінність.

До прикладів технічного боргу належать:

- відсутність автоматизованих тестів, збірки або розгортання проекту;
- відсутність модульних тестів;
- висока складність коду;
- значна зв'язаність коду;
- бізнес-логіка в неправильних місцях;
- замало приймальних випробувань;
- дубльований код або модулі;
- нечитабельні імена чи алгоритми.

Якщо технічний борг у продукті накопичується настільки, що стає важко отримати придатний для релізу значущий приріст цінності до кінця кожного спринту, то потрібно почати його гасити — і дуже швидко. Звичайно, ви не хочете дійти до такої ситуації, і для цього необхідно розпочати усунення технічного боргу раніше, а не пізніше¹.

На практиці:

оновіть критерій «повної готовності продукту», щоби відображати низьку терпимість до технічного боргу

Може бути корисно додати конкретні очікування щодо технічної заборгованості до критерію «повної готовності продукту». Що робитиме команда розробників, щоб уникнути створення нових технічних боргів? І що буде робити команда розробників для усунення наявного технічного боргу?

¹ Для подальшого вивчення теми технічного боргу перегляньте вебсеминар Scrum.org від Марка Нонемана «Мати справи з технічним боргом: уникнення технічного банкрутства» (*Dealing with Technical Debt: Avoiding Technical Bankruptcy*): <https://www.Scrum.org/resources/dealing-technical-debt>.

Наведемо кілька прикладів того, на що команда розробників може звернути увагу в критерії «повної готовності завдання» у зв'язку з технічним боргом.

- Що слід піддати рефакторингу?
- Як залишити модуль в кращому стані, ніж ми його знайшли (Правило бойскаута)?
- Що ми будемо робити, коли стикнемося з технічним боргом під час досягнення приросту цінності продукту? Який технічний борг ми будемо шукати та усувати в процесі?
- Що ми будемо робити з технічним боргом, який залишається нерозв'язаним? Критерій «повної готовності завдання», створений командою розробників, є формою робочої угоди, тому важливо чітко вказати, як запобігати технічній заборгованості та управляти нею. Ця угода допомагає членам команди притягати одне одного до відповідальності стосовно дотримання високих стандартів якості, а також робить якість більш прозорою для зацікавлених сторін.

Надання прозорості технічному боргу

Додавання технічного боргу до беклогу продукту зробить його прозорим для Scrum-команди та зацікавлених сторін. Цей крок також дозволяє власнику продукту приймати кращі рішення щодо замовлення роботи. Якщо ви це зробите, не забудьте чітко відобразити цінність розв'язання технічного боргу в бізнес-плані. Що виграє бізнес, усунувши цей технічний борг? Скільки коштуватиме бізнесу відсутність розв'язання цього технічного боргу? Як впливатиме на користувачів відсутність розв'язання цього технічного боргу?

Наведемо кілька прикладів отримання цінності з погляду бізнесу. Розгляньте можливість додавання даних, які відображають наявні проблеми якості або очікуваний приріст ефективності на основі історії для конкретного продукту.

- Здійсніть рефакторинг, щоб зменшити кількість шляхів проходження коду, скоротивши час тестування на 30 %.

- Застосуйте узгоджені та структурні домовленості, які дозволяють членам команди бути ефективнішими та дієвішими під час створення нових функцій та розв'язання проблем.
- Централізуйте бізнес-логіку для функції X, спростивши оновлення бізнес-логіки в майбутньому та знизивши ймовірність помилок. За останні 4 місяці в нас було 24 баги (помилки), які безпосередньо вплинули на клієнтів і призвели до втрати 100 000 доларів прибутку.
- Здійсніть рефакторинг функції Y, що збільшує продуктивність системи на 35 % і призводить до скорочення часу транзакції на 2,5 секунди.
- Здійсніть рефакторинг функції Z, щоб тепер, коли цінність функціоналу для клієнтів доведено, ми могли масштабувати рішення для ширшої бази користувачів, що призведе до збільшення доходу.

На практиці: розглядайте технічний борг як заборгованість за кредитною картою

Припиніть його створювати. І почніть виплачувати, потрохи в кожному спринті. Якщо технічний борг настільки погіршився, що завдання забезпечити «готовий» приріст цінності продукту здається команді складним, вона повинна провести обговорення цього питання й визначити, чи настав час скоротити кредитну картку та почати виплачувати трохи більше, ніж відсоткові платежі кожного спринту.

Команда розробників може допомогти власнику продукту зрозуміти важливість розв'язання проблеми технічного боргу й домовитися про кількість часу, який слід виділити в майбутніх спринтах для елементів беклогу продукту, пов'язаних із технічним боргом. Власник продукту може допомогти зацікавленим сторонам зрозуміти цінність, яку вони отримують від розв'язання цих питань.

Слід продовжувати перевіряти та адаптувати цей підхід, щоб зрозуміти отримані вигоди та, за необхідності, відрегулювати обсяг часу, витраченого на розв'язання технічного боргу.

Забезпечення помітності «погашення» технічного боргу

Якщо продукт має для погашення значну суму технічного боргу, подумайте про створення помітного показника прогресу для команди. Цей крок починається з того, що Scrum-команда встановлює початкову ціль або, можливо, кілька цілей. Потім члени команди можуть створити інформаційне джерело, яке відстежує їхній прогрес у поверненні технічного боргу під час кожного спринту.

На рис. 3.6 показано приклад використання термометра як метафори та індикатора прогресу. Можливо, Scrum-команда ставить за мету усунути певний обсяг технічного боргу протягом декількох спринтів, і термометр допомагає їм святкувати невеликі прибутки, отримані кожним спринтом. Іншим підходом є графік вигорання завдань, що показує загальну суму відомого технічного боргу (тобто прозорого в беклогу продукту) та відстеження трендів із часом, коли технічний борг усувається, виявляється або, можливо, створюється навмисно.

4-квартальна ціль технічного боргу



Рис. 3.6. Візуалізація технічного боргу може мотивувати щось із цим робити

Підсумки

Досягнення продуктом рівня «повної готовності» має важливе значення для отримання цінності під час одночасного контролю ризиків та керування складністю. Оскільки Scrum-команда формує міцну основу, вона, природно, зможе розвивати командний процес для ефективнішого постачання високоякісного продукту високої вартості. У Scrum не існує «найкращих практик», тому команди повинні постійно перевіряти, що вони роблять, чому вони це роблять і які переваги вони отримують (або вже не отримують). Технології та бізнес постійно змінюються, тому команди повинні думати про те, що буде далі і що їм потрібно робити інакше, аби відповідати новим потребам, випереджати конкурентів і радувати клієнтів.

Гнучкість, яка забезпечується мінімальними межами фреймворку Scrum, відкриває багато можливостей для розкриття творчості спільних команд. Вони можуть орієнтуватись у можливостях, завжди прагнучи до більшої прозорості свого процесу та його впливу на результати, а також, часто перевіряючи та адаптуючи командний процес на основі цих ідей. Команди повинні мінімально враховувати, наскільки добре вони застосовують критерій «повної готовності завдання» та цілі спринту, як протікає робота, а також які методи та заходи, спрямовані на досягнення якості, є необхідними.

Заклик до дії

Розгляньте зі своєю командою наведені нижче запитання.

- Чи гарантовано ви досягаєте рівня «повної готовності» продукту принаймні до кінця кожного спринту? Якщо ні, то що команда з цим робила? В іншому разі — де є можливості робити щось ефективніше, якісніше та з більшою радістю?
- Як цінності Scrum проявляються у вашій командній роботі?
- Наскільки надійним є критерій «повної готовності завдання»? Як він еволюціонував із часом і які є можливості для його вдосконалення?
- Як цілі спринту використовуються впродовж спринту?

- Наскільки прозора ваша робота в спринті? Де вам потрібно більше прозорості?
- Якими є тенденції якості продукту?
- Скільки технічного боргу в продукті? Він збільшується чи зменшується?
- Де команді потрібно нарощувати знання та навички, щоби швидше створити придатний для релізу приріст належної якості?
- Які проблеми зараз найбільше дошкуляють? Визначте один або два експерименти, які допоможуть розвинути командну роботу та ефективніше дійти до «готового» продукту. Для кожного експерименту обов'язково визначайте бажані впливи та спосіб їх вимірювання.

Розділ 4

Поліпшення наданої цінності

Створення «готового» функціоналу для продукту — це не кінець подорожі, а лише її початок заради навчання з метою постачання більшої цінності. Scrum-команда тепер має змогу виміряти поставлену цінність та застосувати емпіризм для поліпшення цінності, отримуваної клієнтами.

Що таке цінність?

Термін «цінність» багаторазово використовується в «Посібнику зі Scrum». Уперше його застосували у визначенні Scrum: «постачання продуктів найвищої можливої цінності». Цікавий експеримент — запитати людей, як вони визначають поняття «цінність». І дійсно, важко визначитися зі значенням «цінності» без використання слів «цінність» або «цінний». Цінність, зрештою, визначається враженнями споживачів, отриманими від продукту.

Наведемо запитання, які можуть допомогти вам визначити, чи поставляєте ви цінність.

- Чи задоволені ваші клієнти? Чи допомагаєте ви їм досягати результатів, які вони вважають важливими?
- Чи знаходить ця задоволеність відображення в способах, які можна вигідно монетизувати (перетворити на гроші)?
- Кількість ваших клієнтів зростає чи зменшується?
- Як швидко ви можете донести нову ідею до клієнта та виміряти результат?
- Чи задоволені ваші співробітники?

Некомерційні та соціальні підприємства не дуже турбуються стосовно вигідної монетизації від споживачів. Та навіть у цьому випадку вони все ж опікуються результатами, які отримують користувачі, хоча й можуть послуговуватися найменуваннями, як-от «громадяни» або «клієнти», замість «споживачі». Деякі комерційні підприємства також керуються місіями, але їх можна описати з погляду досягнення певних результатів для групи людей, як у наведених прикладах:

- збільшення зайнятості населення;
- поліпшення добробуту громади;
- зменшення негативних впливів на навколишнє середовище.

Відомий консультант з управління, викладач і автор Пітер Друкер зауважив: «Якщо ви не можете чогось виміряти, ви не можете це вдосконалити». Те саме стосується й цінності: створення та постачання «готового» приросту цінності продукту недостатньо — ви повинні виміряти надану цінність, щоби поліпшити її.

Швидше постачання — хороший початок, але цього недостатньо

Хоча багато організацій вдаються до Scrum для здійснення *швидшого постачання*, щойно вони починають здійснювати його споживачам та оцінювати результати, як виявляють, що справжньою перевагою Scrum є отримання зворотного зв'язку в короткі терміни для швидшого поліпшення. Насправді, якби швидше постачання могло само собою розв'язати проблеми, з якими стикаються організації в процесі задоволення потреб споживачів, було б достатньо традиційного підходу з багатьма дуже короткими релізами.

Проблема в тому, що, перефразовуючи Джона Вонамейкера, більше половини наших ідей не несе ніякої цінності, але ми не знаємо, яка саме з тих половин¹. Щоби поліпшити свою здатність постачати цінність, ви мусите не лише поліпшити швидкість, з якою це відбувається, але й оцінити, що ви постачаєте, щоби визначити цінність.

¹ Джон Вонамейкер (1838–1922), заможний власник універмагу, чудово підмітив: «Половина грошей, які йдуть на рекламу, витрачаються марно; біда в тому, що я не знаю, яка саме половина».

І ви повинні використовувати відгуки заради поліпшення цінності для постачання в наступному релізі.

Дослідження показали, що 65 % функцій використовуються зрідка або ж ніколи (рис. 4.1)¹. У статті Гарвардського бізнес-огляду (*Harvard Business Review*) за 2017 рік також було зазначено, що «переважна більшість [нових ідей] зазнають невдачі в експериментах, і навіть експерти часто неправильно оцінюють їхню окупність. У *Google* і *Bing* лише приблизно 10–20 % експериментів дають позитивні результати. У цілому в *Microsoft* третина виявляється ефективною, третина має нейтральні результати й третина — негативні»².

Standish Group, використання характерних особливостей та функцій

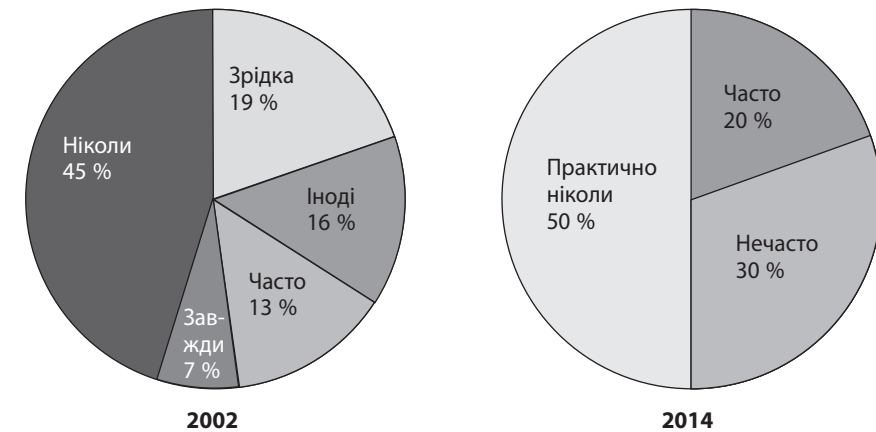


Рис. 4.1. Більшість функцій використовуються зрідка або ніколи

Цінність легко досягнути, оцінюючи її критеріями надходжень, прибутку та прямих витрат, але не будь-яка цінність може бути оцінена в такий спосіб. Темпи зростання частки ринку, розмаїтість клієнтської бази, задоволеність споживачів та працівників і коефіцієнт плинності кадрів теж є важливими показниками цінності. Так само й простота використання та легкість прийняття продукту можуть бути важливими ознаками, які вказують на поліпшення продукту.

¹ Don McGreal, Ralph Jocham. *The Professional Product Owner: Leveraging Scrum as a Competitive Advantage* (Addison-Wesley, 2018), частина серії Professional Scrum від Scrum.org.

² <https://hbr.org/2017/09/the-surprising-power-of-online-experiments>

Цінність продукту і Scrum-команда

У Scrum власник продукту несе відповідальність за максимізацію результатів, які продукт надасть споживачам, збільшуючи цінність, реалізовану організацією.

Цей акцент на вартості та результатах означає собою зміни для організацій, у яких мірилом успіху історично був обсяг виробництва. За результатом вимірюється те, що виробляється або споживається, наприклад поставлений функціонал або одиниці складності. Те, що на виході, легко виміряти, але воно має лише другорядне значення: кількість наданих функцій не має значення, якщо жодна з цих функцій не поліпшує життя або можливості замовника. Поставлений функціонал має значення лише з урахуванням прибутковості чи часу виходу на ринок, але якщо він не дає нічого цінного, тоді це лише марнотратство.

На практиці: вимірювання прогресу й успіху

Коли люди ведуть розмову про «статус» чи «прогрес», придивіться до способу мислення, що рухає дискусію. Якщо йдеться про відсоток виконання, кількість створених функцій або червоний/жовтий/зелений статус, поставте кілька важливих запитань, які зосередять увагу на цінності продукту.

- Чи можемо ми здійснити всі ці заходи й все одно не досягти успіху?
- Як ми перевіряємо припущення щодо потреб користувачів чи попиту на ринку?
- Що ми дізнаємося про цінність? Як це впливає на наші рішення щодо продуктів?
- Що змінилось у наших користувачів або в нашому конкурентному середовищі з початку цієї ініціативи?

Надто часто «відсоток виконання» та подібні дискусії приховують припущення, що все в «списку вимог» є важливим і цінним. Значені вище дослідження доводять, що це не так. Замість турбуватися, чи все буде зроблено, зосередьтеся на обговореннях, як можна швидше перевірити припущення щодо цінності задля зменшення втрат та збільшення цінності, яку ви поставляєте.

Scrum-команда визначає свою роботу в межах Scrum-фреймворку. Цей процес містить визначення цінності, її постачання та вимірювання.

Попри те що відповідальним залишається власник продукту, цілком імовірно, що йому потрібна допомога. Він потребує участі зацікавлених сторін, зокрема замовників, користувачів та членів команди розробників. Власник продукту також залежить від команди розробників, щоби справді забезпечити цінність, тому важливо, аби члени цієї команди розуміли, які результати прагнуть отримати споживачі, щоби краще донести їм свої рішення¹.

Беклог продукту забезпечує прозорість щодо відносної важливості роботи, яка, на думку власника продукту, максимізує поставлену цінність. Коли його застосовують найбільш ефективно, він створює основу для діалогу з рештою Scrum-команди, а також із зацікавленими сторонами про те, що є цінним.

Використання концепції продукту для підсилення командної мети, зосередженості та ідентичності

Концепція продукту виражає причину його існування — для кого він призначений і що він мав би для них зробити. Це важливо під час визначення й фінансування продукту, проте це також має значення для донесення мети й фокусу до Scrum-команди та допомоги у формуванні її ідентичності. Періодичне повернення до цієї концепції є корисним способом нагадати всім членам команди, чому власне вона існує². Різні пов'язані методики допомагають команді формувати та зміцнювати свою ідентичність.

¹ Для більшого заглиблення в процес визначення продуктів і розуміння того, що споживачі вважатимуть цінним, ми рекомендуємо курс від Scrum.org для власників продукту: <https://www.Scrum.org/courses/professional-Scrum-product-owner-training>. Якщо ви не можете потрапити на курс або навіть маєте таку змогу, ми також порекомендуємо вам книжку: Don McGreal, Ralph Jocham. *The Professional Product Owner: Leveraging Scrum as a Competitive Advantage* (Addison-Wesley, 2018), частина серії Professional Scrum від Scrum.org.

² Для отримання додаткової інформації про створення чіткої концепції продукту див. книжку: Don McGreal, Ralph Jocham. *The Professional Product Owner: Leveraging Scrum as a Competitive Advantage* (Addison-Wesley, 2018), частина серії Professional Scrum від Scrum.org.

- *Цінність товару.* Чітке розуміння цінності допомагає команді з'ясувати причину, через яку вона виконує роботу, та перевірити, чи сприяє її праця появі цінності. Визначення цінності конкретним способом (наприклад, беручи до уваги дохід, частку ринку, задоволеність споживачів) має велике значення для виховання цілеспрямованості.
- *Образ споживача.* Образи споживачів допомагають команді краще зрозуміти користувачів та клієнтів, щоби розвинути вміння ставити себе на їхнє місце. Це зрештою допомагає членам команди бачити мету своєї роботи та ухвалювати кращі рішення. Деякі команди розміщують описи уявних покупців у робочому середовищі як постійне нагадування про людей, для допомоги яким вони працюють¹.
- *План розвитку продукту.* План розвитку, або ж роадмап, продукту — це візуалізація плану високого рівня, призначеного для того, щоби допомогти команді побачити напрям розвитку продукту в часі. Що більше роадмап зосереджено на бізнес-цілях та цінності для бізнесу, то більше це забезпечує ґрунтовну мету.

На практиці: підживлення концепції продукту

Концепцію продукту легко визначити на початку його життєвого циклу, але згодом під час окремих релізів про неї забувають. Продукти змінюються з часом, і це може відбуватися усвідомлено або випадково. Роздування певної функціональності та інші форми шкідливого внесення доповнень часто спричинені відсутністю фокусування на певній меті, ця відсутність виникає внаслідок заплутаної або нечіткої концепції продукту.

Розгляньмо способи підтримки чіткості концепції.

1. *Охоплюйте ширшу спільноту.* Ваші зацікавлені сторони та ваша команда розробників, вірогідно, мають цінні погляди та ідеї. А охоплюючи інших, ви допомагаєте їм відчувати, що їх чують і розуміють, і це, імовірно, збільшить їхню залученість до процесу та розуміння концепції продукту. Використання концепції про-

¹ Для отримання додаткової інформації про образи споживачів див.: <http://gamestorming.com/empathy-mapping/> і <https://www.romanpichler.com/blog/10-tips-agile-personas/>.

дукту для періодичного обговорення того, кому продукт мусить служити і що він повинен зробити для них, є хорошим способом розвинути її вдумливо, а не випадково.

2. *Розвивайте її на основі нової інформації.* Початкова концепція продукту — це вихідна точка, заснована на припущеннях і здогадах, багато з яких будуть правильними, а деякі виявляться помилковими. Вам потрібно буде перевірити, пристосувати та розвинути концепцію продукту на основі нової інформації.
3. *Не відволікайтеся від основної думки.* Успішні продукти мають чіткий фокус; команди, які їх створюють, знають, кому і як вони служать. Погані продукти намагаються зробити дещо для кожного, але не надто багато.
4. *Постійно підкріплюйте її.* Кажучи словами професійного Scrum-тренера Дона Макґріла, «Будьте надокучливим щодо цього». Власники продуктів повинні шукати можливості для підкріплення концепції продукту, а також перевірити її відповідність. Власники продуктів можуть додати фізичне втілення концепції продукту до оглядів спринту або інших обговорень із зацікавленими сторонами та командою розробників (наприклад, у формі *Product Box*¹ або постеру з мініпрезентацією²).

Вимірювання цінності

Scrum-команди можуть по-різному вимірювати цінність, яку вони надають, до того ж різні типи цінності потрібно вимірювати в різні способи, від дуже загальних, які досліджують продукт у цілому, до вкрай конкретних, що розкривають певні елементи беклогу продукту. Насправді вам у різний час, мабуть, знадобляться всі види перелічених нижче показників.

Загальні показники задоволеності клієнтів:

- індекс лояльності клієнтів;
- дохід або прибутковість на одного клієнта;

¹ Більше інформації про метод *Product Box* див. <https://www.innovationgames.com/product-box/>.

² Більше прикладів мініпрезентацій див. на <https://strategypeak.com/elevator-pitch-examples/>.

- наявність постійних покупців;
- зниження за рахунок вартості володіння;
- поліпшений коефіцієнт конверсії;
- зростання кількості клієнтів або користувачів;
- реферали клієнтів (нові клієнти внаслідок рекомендацій наявних клієнтів).

Досягнення бізнес-цілей:

- частка ринку;
- сукупний дохід або прибуток;
- вартість залучення нового клієнта;
- скорочення тривалості циклу, зменшення наявних запасів, економія коштів або збільшення частки ринку.

Конкретні показники для оцінки взаємодії продукту та споживачів:

- економія споживачем часу, що витрачається на досягнення мети;
- частота використання послуги (функції);
- тривалість використання послуги (функції);
- кількість клієнтів або користувачів, які використовують певну послугу (функцію);
- показники завершення/відмови від транзакції.

На практиці:

використання кількох показників для діагностики проблем із продуктами

Наведені вище списки не є вичерпними, а швидше ілюструють, що різні типи показників можуть бути використані для кількісної оцінки цінності різними способами. У більшості випадків одні показники спричиняють нові запитання, для пояснення яких потрібні інші показники. Наприклад, загальне зниження індексу лояльності клієнтів або рефералів клієнтів свідчить про те, що споживачі стали менш задоволені продуктом. Коли ви починаєте оцінювати те, що користувачі насправді роблять із продуктом, то можете виявити, що якась нова функція, яку ви впровадили,

ускладнила використання продукту, а це призвело до загального незадоволення, яке ви й виявили.

Scrum-команди можуть поліпшити свої показники цінності різними способами.

- *Залучайте інших.* Як і в ситуації з концепцією продукту, ваші зацікавлені сторони та команда розробників, вірогідно, мають цінні погляди та ідеї. Крім того, залучаючи інших, ви даєте їм можливість відчувати, що їх почули, і це, імовірно, збільшить їхню прихильність та розуміння цінності товару.
- *Зробіть показники видимими.* Показники повинні бути прозорими для всіх зацікавлених сторін та команди розробників. Подумайте про створення інформаційної панелі показників цінності продукту.
- *Поговоріть про показники й результати в оглядах спринту.* Оскільки особливості та функції демонструються учасникам, зверніться до питання очікуваної цінності та того, як стане відомо, якщо її буде досягнуто.
- *Пов'яжіть показники та результати з бізнес-цілями.* Це повертає нас до створення узгодженості та обґрунтування того, як ви визначаєте цінність. І якщо бізнес-цілі змінюються, то це може бути хорошим показником, що вам потрібно перевірити та, можливо, адаптувати свої визначення та показники цінності.

Орієнтація запланованих для виконання елементів продукту (беклогу) на результати користувачів

Розуміння потреб і бажань користувачів є ключовим для виявлення того, що і чому є цінним. Scrum-команди можуть застосувати низку методів, щоби краще зрозуміти користувачів. Зокрема, два методи, які часто поєднуються і допомагають Scrum-командам краще зрозуміти користувачів та зосередити на них увагу,— це образ споживача/результати та користувацькі історії.

Образи споживача та результати

Образ споживача — це вигаданий персонаж, створений для представлення користувача або типу клієнта, який може використовувати товар у той самий спосіб, що й інші. Образи споживачів часто ство-

рюються за допомогою даних маркетингових досліджень та опитувань клієнтів. Вони оживляють споживачів (якщо так можна сказати), зображуючи картину, пов'язану з демографічними показниками, способом життя, цілями та причинами використання продукту. Використання образів споживачів допомагає Scrum-команді сфокусуватися, сприяючи їй в отриманні дуже конкретної інформації про те, на кого вона націлюється за допомогою певного елемента беклогу.

Результат — це якась умова або ціль, що її хотіла б досягти людина, якій відповідає певний образ. Розуміння цих цілей допомагає Scrum-команді досягти фокуса, чітко сформулювавши, що хотів би отримати користувач або замовник. Використання образів споживачів та результатів дає такі основні переваги:

- допомога людям, які створюють продукти, більше наблизитися до користувачів та їхніх потреб;
- допомога у визначенні проблемних точок користувача та творчих рішень;
- допомога у концентрації зі збереженням можливості бачити всю картину.

Образи споживачів та результати виступають протиотрутою для функцій, які не мають добре окреслених цілей або чіткої цільової аудиторії. Образи споживачів також допомагають уникнути розмитих дискусій про «користувача», оскільки жоден продукт не має єдиного однорідного типу користувача. Натомість, люди використовують той самий продукт у дуже різні способи заради досягнення досить різних результатів.

На практиці:

використання карти впливу для отримання кращої концепції продукту

Картографування впливу може бути корисним прийомом для підключення елементів беклогу продукту до цілей, яких вони мають досягти¹. Ви можете використовувати модифіковану карту впливу,

¹ Для отримання опису розширеної карти впливів див. <https://www.Scrum.org/resources/blog/extending-impact-mapping-gain-better-product-insights>.

щоби пов'язати бізнес-ціль, якої намагаєтеся досягти, з образами споживачів, котрих ви будете обслуговувати, з результатами, яких ви сподіваєтеся допомогти їм досягти, впливом на вашу організацію у разі досягнення таких результатів та елементами беклогу продукту, що їх ви забезпечите в продукті для досягнення цих результатів (рис. 4.2).

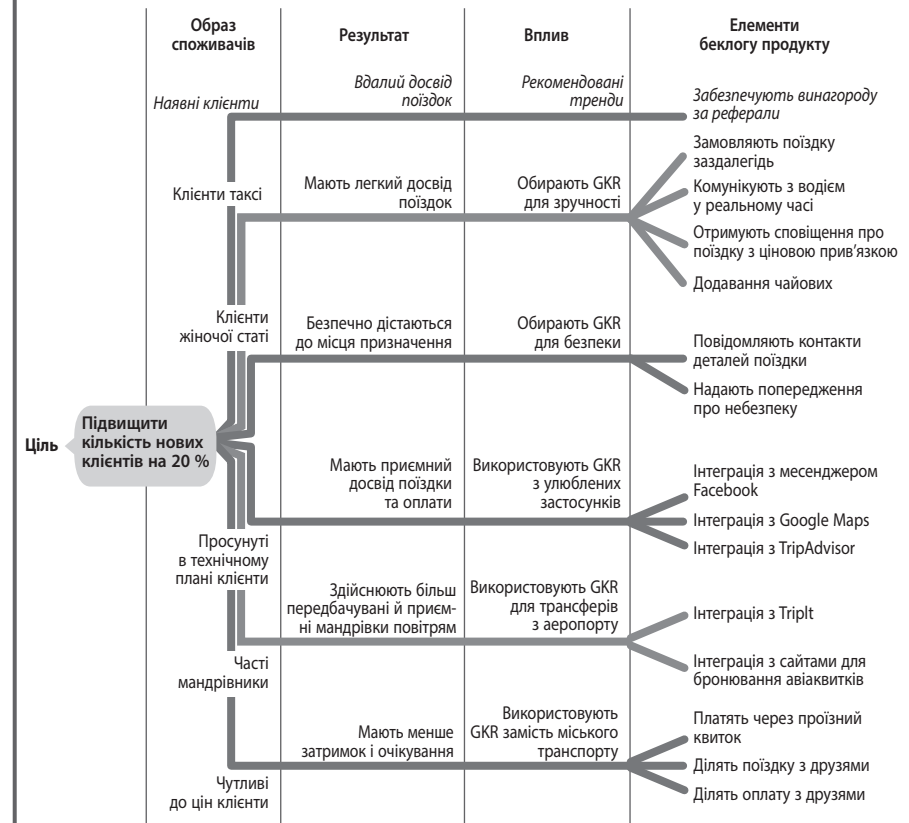


Рис. 4.2. Розширена карта впливу допомагає пов'язати елементи беклогу продукту з цілями

У прикладі на рис. 4.2 організація — компанія, яка виходить на ринок послуг спільного проїзду, — хотіла би збільшити кількість нових клієнтів на 20 %. Для цього потрібно звернутися до бага-

трьох різних видів потенційних користувачів таксі, кожен з яких представлений різним *образом*. У кожної людини різні результати, яких би вона хотіла досягти. Компанія вважає, що, досягнувши конкретних результатів, людина спричинить певний *вплив* на компанію або принесе їй певні результати, і, на думку представників компанії, постачання певних елементів беклогу продукту допоможе їм це зробити. Карти впливу можна використовувати в різні способи. У контексті вдосконалення беклогу продукту вони допомагають Scrum-команді подумати про те, як кожен з елементів беклогу продукту забезпечить певний результат.

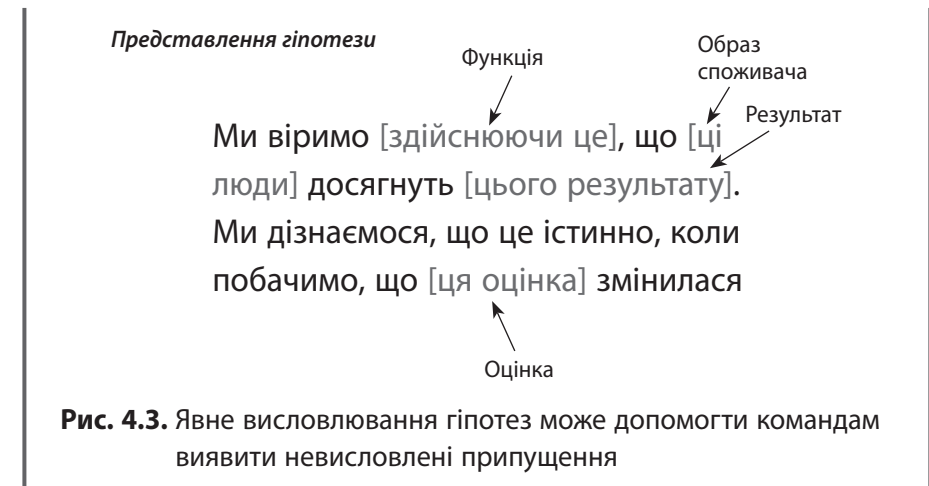
Карти впливу також допомагають власнику уявити, кому пропонується продукт, чого різні типи користувачів або зацікавлені сторони хотіли б досягти, використовуючи продукт, і яку користь отримає організація, допомагаючи клієнтам досягти конкретних результатів.

На практиці: представлення елементів беклогу продукту як гіпотез або експериментів

Заснована на гіпотезах розробка (HDD) — це спосіб представлення елемента беклогу продукту в такий спосіб, який дає змогу чітко визначити образ споживача, результат, спосіб вимірювання та очікуваний результат¹. Порядок виконання заснованої на гіпотезах розробки допомагає Scrum-команді висувати гіпотези та будувати експерименти й усвідомлювати, як вона дізнається про істинність гіпотези (або припущення). Це спонукає всіх думати не лише про те, чого вони намагаються досягти, а й про те, як вони це оцінюватимуть². На рис. 4.3 наведено приклад одного зі способів представлення гіпотези.

¹ Курс професійного Scrum із досвідом взаємодії з користувачем на Scrum.org (<https://www.Scrum.org/courses/professionalScrum-user-experience-training>) навчить, як поєднувати сучасні UX-практики зі Scrum, і базується на книжці: Jeff Gothelf, Josh Seiden, *Lean UX* (O'Reilly Media, 2016).

² Більше про Scrum та засновану на гіпотезах розробку (HDD) див. <https://www.Scrum.org/resources/blog/Scrum-and-hypothesis-driven-development>.



Користувацькі історії

Користувацькі історії водночас і широко використовуються, і так само широко неправильно застосовуються. Початковий намір їх запровадження полягав у тому, щоби вони слугували певним *шаблоном* чи маркером розмови про те, як хтось може використовувати продукт для досягнення певного результату. Коли вони відхиляються від цього нагадування про розмову та стають форматом для документування елементів беклогу продукту, то можуть перетворитися на абсолютну нісенітницю, особливо якщо їх використовуватимуть для представлення технічних вимог або обмежень. Щоби цього не сталося, зосередьтеся на трьох користувацьких історіях¹.

- *Картка*, яка є нагадуванням про розмову. Вона повинна мати простий мінімальний формат, що вміщується на індексній картці (або на стікері) і може складатися лише з «Поговоріть із Мері про те, як підбиваються рахунки наприкінці розрахункового періоду».
- *Розмова*, яка є фактичним обговоренням теми, згаданої на картці.
- *Підтвердження*, яке фактично є тестами, котрі доводять, що це працює².

¹ <https://www.agilealliance.org/glossary/user-stories/>

² Якщо ви хочете застосувати користувацькі історії як ефективний метод уточнення беклогу продукту, ми рекомендуємо книжку: Mike Cohn, *User Stories Applied* (Addison-Wesley Professional, 2004).

На практиці:**загальні пастки, пов'язані з беклогом продукту**

Незалежно від формату, який набувають ваші елементи беклогу продукту (образ споживача та результати, користувацькі історії чи щось інше), остерігайтеся низки загальних пасток.

1. *Припущення, що елементи беклогу продукту мають відповідати встановленому формату.* Формат, який часто використовується для користувацьких історій, не є ані оригінальним, ані обов'язковим. Ваша Scrum-команда може отримати користь від використання певного формату, але їй не потрібно припасовувати елемент беклогу продукту під цей формат. Пишіть те, що має сенс. Зрештою, картка є лише нагадуванням про розмову.
2. *Відсутність зрозумілих користувачеві чи замовнику переваг елементів беклогу продукту.* Якщо ви знову і знову пишете «Як користувач» на початку користувацької історії, ви не допомагаєте створити фокус і розуміння користувача, для якого розробляєте функцію чи послугу. Як зазначено в пункті 1, вам не завжди потрібно використовувати цей формат, але ви мусите мати спільне з користувачами та замовниками розуміння вашого продукту.
3. *Нез'ясоване поняття цінності.* Часто елементи беклогу продукту вказують на бажану послугу, функцію чи можливість, але не ясно, чому саме цей елемент є бажаним. Коли немає спільного розуміння, чому ви щось будujete, може трапитися, що альтернативні шляхи розв'язання проблеми не обговорюватимуться. Відтак утрачається можливість максимізувати цінність.
4. *Ставлення до елементів беклогу продукту як до контракту.* Беклог продукту — це не просто *agile-версія* традиційного документа про вимоги до програмного забезпечення. Елементи беклогу продукту відкриті для змін. Розгляньмо техніку «Користувацька історія»: вона призначена для відображення потреб клієнта, а не лише для їх документування. Користувацькі історії необхідно перетворювати навіть у процесі їх створення.
5. *Додавання деталей реалізації.* Елементи беклогу продукту повинні бути зосереджені на «що», а не на «як». Якщо ви приймає-

те реалізаційні рішення занадто рано, то можете обмежити свої можливості. До того ж у вас виникатимуть непотрібні витрати внаслідок детального опису того, що може змінитися, коли реалізація насправді відбудеться.

Поліпшення наданої цінності під час спринту

Поки команда розробників працює над елементами беклогу продукту під час спринту, її розуміння цінності, яку надаватимуть ці елементи, продовжує вдосконалюватися, оскільки члени команди дізнаються більше під час бесід, завдяки зворотному зв'язку із зацікавленими сторонами та навіть від реальних клієнтів або користувачів, якщо команда випускає продукт під час спринту. (Так, це можливо!)¹ Елемент беклогу продукту ніколи не «блокується» або не «затверджується», доки фактично не буде «готовим». Це містить деталі й того, що будуються, і того, як воно будуються. Якщо члени команди розробників співпрацюють між собою, а також із власником продукту, вони можуть постійно ставити запитання, пов'язані з цінністю, і дозволяти цьому керувати їхнім процесом прийняття рішень. Наведемо приклади.

- Scrum-команда вирішує розділити елемент беклогу продукту, щоби зосередитися на найцінніших критеріях прийнятності для користувацьких функцій, яких потребують зараз, що дозволяє пізніше змінити пріоритет частини з нижчою цінністю базуючись на розгляді інших бажаних функцій.
- Побачивши нову можливість, реалізовану в продукті, власник продукту дає вказівки щодо того, як зробити його більш помітним для цільових користувачів.
- Команда розробників бачить альтернативні способи збільшення конверсії користувачів, що є заявленою цінністю елемента беклогу продукту, який вони будують. Розробники передають ці варіанти власнику продукту, і вони домовляються про зміну обсягу для кращого задоволення бізнес-потреб, водночас поставляючи «готовий» приріст цінності в спринті.

¹ Щоб отримати глибше уявлення про реліз під час спринту, див. <https://www.Scrum.org/resources/blog/myth-3-Scrum-releases-are-done-only-end-sprint>.

Перевірка та пристосування на основі відгуків

Випустивши продукт або продемонструвавши його зацікавленим сторонам, ви отримаєте емпіричні дані, якими зможете підтвердити (або відхилити) свої гіпотези. Одна одиниця даних зазвичай не скаже вам багато, але з часом тенденції покажуть, чи відбувається поліпшення або погіршення в певному вимірі. І пам'ятайте, вам можуть знадобитися різні заходи для вимірювання, щоби по-справжньому зрозуміти, що відбувається.

Наприклад, у вас можуть бути дуже задоволені клієнти, які люблять ваш продукт, але жодні виміри їхньої задоволеності не підкажуть, чому люди його не купують. Якщо ви хочете розширити свою частку ринку, вам потрібно буде виміряти більше за *поточну цінність*, яку поставляє товар, тобто вам також необхідно буде зрозуміти, які фактори заважають вам реалізувати весь ринковий потенціал вашого продукту¹.

Аналізуючи ціннісні тенденції, подумайте, які зміни у вас вийшли в реліз та коли і як вони могли вплинути на цінність. Подумайте, які фактори не піддаються вашому контролю (наприклад, великий спад на фондовому ринку може вплинути на рішення користувачів, навіть якщо ви впровадили нові функції, котрі, за вашими очікуваннями, мали би збільшити обсяги продажу).

Навчання як цінність

Іноді цінність полягає в навчанні. Цей процес може керуватися даними або ні, але здатний бути корисним для чіткого розуміння навчання як цінності. Наприклад, Scrum-команда може захотіти дізнатися, яку з двох технологічних послуг буде легко як впровадити, так і вдосконалити, а також задовольнити бізнес-потреби. В іншому прикладі Scrum-команда може захотіти дізнатися, який досвід взаємодії з користувачем найімовірніше призведе до покупки.

¹ Scrum.org розробив фреймворк для розуміння того, як виміряти цінність та поліпшити вашу здатність її поставляти, і називається вона «доказовий менеджмент» (Evidence-Based Management). Два виміри цінності — це «поточна цінність», яка стосується цінності, реалізованої поточними споживачами вашого продукту, та «нереалізована цінність», що означає потенційну цінність, яку ви могли б поставити всім потенційним споживачам, але не сьогодні. Для отримання додаткової інформації див. <https://www.Scrum.org/resources/evidence-based-management>.

Ефективні огляди спринту містять реалізовану цінність

Нагадаємо, що результатом огляду спринту є адаптація беклогу продукту. На додаток до відгуків зацікавлених сторін про приріст цінності продукту та загальні ринкові тенденції, дані про фактичну цінність та тренди дають вам ще більше емпіричних даних для управління рішеннями щодо беклогу продукту. Зробіть прозорими свої фактичні показники цінності. Отримайте інформацію про те, що бачать зацікавлені сторони в тенденціях і як, на їхню думку, це має сприяти адаптації беклогу.

Збирання відгуків від зацікавлених сторін

Ваш підхід до збирання інформації від зацікавлених сторін буде залежати від багатьох факторів, зокрема (але не обмежуючись цим) складності вашого продукту, кількості зацікавлених сторін, розмаїття типів зацікавлених сторін та їхніх потреб, а також розташування ваших зацікавлених сторін. Часто для збирання найціннішої інформації потрібно звертати особливу увагу на зацікавлені сторони. Хоча зазвичай вони досить досвідчені в певній галузі, можливо, вам доведеться спрямувати їхню увагу на те, стосовно чого ви бажаєте отримати відгук. Майте на увазі, що огляди спринту — не єдиний випадок, коли власники продуктів можуть отримувати інформацію та співпрацювати із зацікавленими сторонами. Ви отримаєте більше від сесій співпраці із зацікавленими сторонами в цілому та від оглядів спринту зокрема, якщо зможете сфокусувати участь зацікавлених у наведені нижче способи.

- *Будьте конкретними стосовно того, що ви переглядаєте та які відгуки шукаєте.* Наявність простого, але чіткого порядку денного для сесій зворотного зв'язку допомагає всім зосередитися.
- *Зробіть сесії зворотного зв'язку сповненими активності та заохочуйте до участі.* Люди досягають успіхів завдяки активності. І навпаки, пасивне сидіння й слухання того, як хтось проходить по особливостях, функціях та можливостях, часто знуджує учасників. Організація сесій у спосіб, який змушує людей рухатися, триматиме їх у більшій зацікавленості, а це часто економить час. Крім того, люди частіше відчува-

ють, що їх чують, якщо вони фізично брали участь у певній діяльності.

- *Уможливіть співпрацю між зацікавленими сторонами.* Представники зацікавлених сторін можуть вчитися одне в одного. Не всі поділяють той самий погляд, й іноді це призводить до конфліктів, які можна розв'язати, якщо зацікавлені сторони розумітимуть погляди одне одного.
- *Зробіть співпрацю видимою.* Простіше обговорити широкий спектр ідей, коли ми можемо бачити їх у фізичному просторі та легко додавати, оновлювати та переміщувати інформацію. Крім того, такий підхід створює прозорість того, чого ми намагаємося досягти і чого насамкінець навчилися. Також корисно мати концепцію та визначення цінності, щоби тримати все у фокусі.
- *Розподіліться на менші групи під час сесії.* Обговорення в менших групах людей, що цікавляться певними темами, зазвичай ефективніші, ніж у великих. Залиште час для обговорення в менших групах та повернення результатів до великої групи.
- *Упровадьте методики, які заохочують до порівняння відносної цінності.* Легко загрузнути в деталях, особливо коли йде обговорення, які елементи беклогу продукту цінніші за інші. Порівнюючи значення відносно (тобто елемент X є ціннішим за елемент Y, але менш цінним за елемент Z), ми можемо швидко отримати достатньо інформації¹.

¹ Детально про методи полегшення збирання даних щодо відносної цінності для допомоги у впорядкуванні беклогу продукту див. на с. 213 у книжці: Don McGreal, Ralph Jocham. *The Professional Product Owner: Leveraging Scrum as a Competitive Advantage* (Addison-Wesley, 2018).

Підсумки

Scrum не призначений, щоби допомогти вам створити та випустити більше «матеріалів». Натомість Scrum допомагає максимізувати цінність, яку ви створюєте для своїх клієнтів, а отже, і для вашої організації, завдяки частому постачанню продукту, вимірюванню результатів, а отже — навчанню та пристосуванню для отримання ще більшої цінності з продукту.

У цьому розділі ми дослідили, як емпіризм, гнучкий спосіб мислення та робота в команді скеровують вас на шлях рішучого розв'язання складних питань щодо цінності продукту. Ви мусите зробити прозорою цінність і брати участь у досить частих перевірках фактичної цінності, усвідомлюючи, що можете продовжувати рухатися в найкращому напрямі. Так само як складність і непередбачуваність властиві створенню придатного для релізу продукту, з'ясування того, що саме має бути створено, тягне за собою певну складність і непередбачуваність. Scrum забезпечує мінімальний рівень емпіризму, і Scrum-команда повинна визначати свою діяльність у рамках фреймворку Scrum. До цього процесу входить те, як ви уможливорюєте появу цінності, вимірюєте фактичну цінність і пристосовуєтеся до нової інформації та мінливого середовища.

Власник продукту — це особа, відповідальна за оптимізацію цінності. Досвідчений власник продукту залучатиме інших і надаватиме їм повноваження для підтримки досягнення цієї мети. Сильний власник продукту сприятиме формуванню в організації мислення навколо продукту та змалює широку картину, створюючи узгодженість усередині команди розробників та серед зацікавлених сторін щодо напрямку розвитку продукту та способу визначення цінності. Власник продукту співпрацює з командою розробників та зацікавленими сторонами, щоби забезпечити появу цінності ітеративно та поступово, керуючись досвідом вимірювання фактичної цінності.

Заклик до дії

Обдумайте зі своєю командою наведені нижче запитання.

- Наскільки добре розуміють концепцію продукту команда розробників і зацікавлені сторони?

- Де вам потрібна більша прозорість у представленні бажаних результатів і припущень щодо цінності?
- Які заходи для вимірювання цінності допоможуть вам ухвалити більш обґрунтовані рішення стосовно того, що є в беклозі продукту та його порядку? Як часто вам потрібно було би перевіряти ці дані?
- Як команда розробників співпрацює з власником продукту або відповідними зацікавленими сторонами під час проведення спринту?
- Скільки відгуків та нових ідей можна отримати внаслідок огляду спринту чи інших спільних сесій комунікації із зацікавленими сторонами?
- Чи зосереджуються зацікавлені сторони на створенні та постачанні цінності як на ключовому вимірі успіху? Які розмови можуть допомогти вам змістити фокус у правильному напрямку?
- Які проблеми зараз найбільше дошкуляють? Визначтеся з одним або двома експериментами, які допоможуть краще зрозуміти й виміряти цінність. Для кожного експерименту обов'язково вкажіть бажані впливи та способи їх вимірювання.

Розділ 5

Поліпшення планування

Використання емпіричного підходу до постачання складних продуктів у непевному та дуже мінливому світі вимагає емпіричного планування. Тобто під час планування діяльності слід прагнути:

- до уможливлення прозорості заради прогресу;
- установлення реалістичних очікувань;
- мінімізації зайвої роботи за одночасної максимізації цінності;
- використання змін та нових знань для отримання конкурентних переваг;
- до проведення відкритого та чесного обговорення щодо непередбачуваності та складності розробки продукту.

Керівники бізнесу та клієнти все одно будуть — і повинні — ставити запитання «Коли це буде готово?» та «Скільки це коштуватиме?». Коли ви працюєте емпірично, ваші відповіді на ці запитання відображатимуть правдоподібні ймовірності та можливості, а не зобов'язання та впевненість. Ви не можете ідеально спланувати складну роботу, а навпаки, повинні залишатися відкритими для змін і нових знань. Навіть у межах часового інтервалу спринту є складнощі, невідомість та можливості для змін.

Є чотири основні моменти, які Scrum-команда повинна винести зі своєї діяльності з планування.

1. Установіть загальний набір понять, на базі яких можуть виникнути непередбачувана ситуація, адаптація та співпраця (гнучкий спосіб мислення, емпіризм, командна робота).
2. Сформулюйте очікування, що прогрес буде вимірюватися (емпіризм).

3. Переконайте джерело фінансування в тому, що рентабельність інвестицій в ініціативу того варта (гнучкий спосіб мислення).
4. Допоможіть людям, які беруть участь у процесі постачання цінності, ухвалювати кращі рішення (емпіризм, гнучке мислення).

Планування та прогнозування охоплюють широкий спектр тем. У цьому розділі ми зосередимося на висвітленні певних концепцій планування та надамо додаткові ресурси для вивчення конкретних тактик:

- як Scrum-команди планують емпірично та отримують максимальну користь від удосконалення беклогу продукту;
- як Scrum-команди планують ефективно отримати «виконаний» приріст прийнятної цінності продукту під час кожного спринту, зокрема навчання та постійне вдосконалення;
- як Scrum-команди можуть підійти до планування релізу без необхідності планувати безліч спринтів заздалегідь.

Планування за допомогою продуктового мислення

Продуктове мислення допомагає Scrum-командам та організаціям зосередитися на постачанні цінних результатів, а не просто на концентрації кількості результатів¹. Цей підхід легше зрозуміти, коли ми порівнюємо його з проектним мисленням.

Згідно з визначенням Інституту управління проектами (PMI), проект — це тимчасова робота, спрямована на створення унікального продукту, послуги чи результату². Два ключові моменти в цьому визначенні полягають у тому, що проект має початкову й кінцеву дату (що створює межі навколо сфери застосування та ресурсів) і його не можна повторити.

Коли ви думаєте про Scrum, то можете розглядати кожен спринт як проект. Спринт має дати початку та закінчення, і Scrum-команда

¹ Повніший опис відмінностей між проектним і продуктовим мисленням див. <https://www.Scrum.org/resources/blog/project-mindset-or-product-mindset>.

² <https://www.pmi.org/about/learn-about-pmi/what-is-project-management>

створює унікальний приріст цінності, який можна випустити протягом цього періоду. Або ви можете розглянути підмножину беклогу продукту як проект, який охоплює кілька спринтів і досягає передбачуваних переваг або бізнес-цілі завдяки постачанню декількох приростів цінності. Ми намагаємося підкреслити, що ці поняття пов'язані між собою. Однак проблема виникає, коли ви враховуєте частоту вимірювання успіху.

Вимірювання успіху

З нашого досвіду, успіх проекту традиційно вимірюється відповідями на три наведені нижче запитання.

- Чи поставили ви весь обсяг продукту?
- Чи залишилися ви в рамках бюджету?
- Чи зробили ви це вчасно?

Ці запитання походять із «трикутника управління проектами», також відомого як «залізний трикутник», або «потрійне обмеження», зображеного на рис. 5.1.

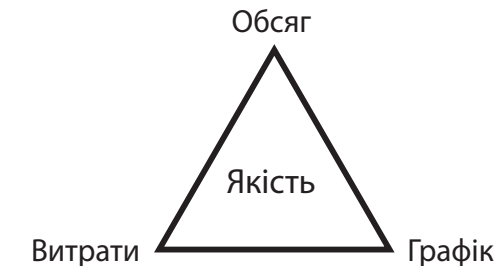


Рис. 5.1. Залізний трикутник управління проектами

Проблема використання як мірила успіху лише цих змінних полягає в тому, що в такий спосіб втрачаються цінні результати. Як дізнатися, чи інвестиції того варті? Звідки ви знаєте, чи потрібно змінити напрям або можна припинити інвестувати раніше? І навіть якщо організація справді вимірює рентабельність інвестицій порівняно з початковим бізнес-обґрунтуванням проекту, це зазвичай трапляється після завершення проекту, отже, із цим уже запізно

щось робити. Крім того, бізнес-кейси самі собою є лише обґрунтованими здогадками, припущення яких потрібно перевірити за допомогою емпіризму.

Емпіричне планування

«План — це просто здогадка, яку ви записали»¹. Замість заздалегідь робити багато аналізів та оцінок, аби створити довгостроковий детальний план, скористайтесь емпіричним підходом до планування. Емпіризм говорить, що навчання відбувається внаслідок дій. Ви трохи плануєте, а насправді виконуєте цю роботу, а потім перевіряєте результати, запитуючи: «Які припущення ми перевірили? Що ми дізналися?» На основі нової інформації ви відповідно пристосовуєте свій план. Змінюючи свої плани з огляду на нову інформацію, ви мінімізуєте ризик складної роботи в непередбачуваному та мінливому світі.

Емпіричне планування вимагає прозорості. Гнучкий спосіб мислення нагадує нам, що єдиним мірилом прогресу є дієвий продукт. Виконання деяких «дій» не приносить жодної користі, якщо немає робочого продукту. Не існує завершеності на 85 %, коли йдеться про «виконаний» приріст цінності продукту або будь-яку іншу складну роботу. Такий тип «оновлення статусу» в кращому разі часто є дикою здогадкою, а іноді намаганням уникнути складних питань та розмов. Крім того, коли організації зосереджуються на дотриманні графіка та/або бюджету, вони перешкоджають прозорості та готовності до змін і навчання. Це насправді збільшує, а не зменшує ризик.

Гнучкий спосіб мислення також нагадує нам про прагнення максимізувати цінність і зменшити втрати. Подумайте, скільки сміття утворюється внаслідок частого оновлення детальних довгострокових планів. Скільки втрат створюють процеси контролю змін, які потребують великого обсягу роботи для додавання нової інформації до вашого плану? Яка цінність потенційно втрачається або відкладається, коли люди відчують тягар важкого процесу контролю змін і надають перевагу простому «дотриманню плану і сподіванням на краще»? Коли команди змушені вкладатися у встановлені терміни, якість часто приносять у жертву.

¹ <https://m.signalnoise.com/planning-is-guessing/>

Зрештою, це дорого коштуватиме вам у довгостроковій перспективі, бо зашкодить вашій здатності поставляти майбутню цінність.

Ґрунтуючись на емпіризмі та agile-підході, ви зможете зосередитися на продуктовому мисленні, оскільки Scrum-команда планує роботу над продуктом, від короткотермінового до довгострокового горизонту планування. Закладання цих основ також буде всім нагадувати: що менше історії в Scrum-команди (тобто емпіричних даних) і що далі в майбутнє простягається прогноз, то ширшим буде діапазон невідомого, складнощів та ймовірності для змін.

На практиці: емпіризм зменшує невизначеність завдяки вживанню заходів, а не збільшенню планування

Традиційні підходи до управління проєктами припускають, що складання детальніших планів зменшує ризик, але більше планування насправді лише затримує протидію ризику, створюючи забагато припущень. Ось чому так багато традиційних проєктів зазнають невдачі. Наприклад, опис детальної архітектури продукту нічого не говорить про те, чи справді ця архітектура буде працювати. Найкращий спосіб зрозуміти, чи щось спрацює,— створити критичну підмножину рішення та випробувати його.

Це ще більше стосується уявлень про те, що сподобається споживачам. Єдиний спосіб точно дізнатися, чи сподобається ваш продукт споживачам,— насправді надати щось мінімально корисне для використання, яке часом називають мінімально життєздатний продукт (MVP)¹.

Конус невизначеності² — це модель управління проєктами, яка виражає зменшення невизначеності результатів, оскільки організація збирає більше інформації. На початку релізу невизначеність стосовно того, що і як будувати, зазвичай є досить високою. Плани будуть доволі «приблизними», імовірно, чимось на кшталт бажаних результатів, деяких можливих особливостей та функцій для досягнення цих результатів, приблизних уявлень про бажані дати

¹ Детальніше про концепцію мінімально життєздатного продукту див. <https://www.agilealliance.org/glossary/mvp>.

² <https://www.construx.com/books/the-cone-of-uncertainty>

релізу на основі ринкових умов. Коли Scrum-команда виконує спринти, вона збирає інформацію та уточнює беклог продукту, і розробники краще розуміють, скільки ще їм потрібно зробити для досягнення бажаних результатів. Власник продукту може довіряти емпіричному процесу, знаючи, що певна цінність набуде ознак «виконаної» принаймні до кінця кожного спринту. Відтак, нова інформація та нові знання допоможуть власникові продукту ухвалити рішення: продовжувати рух уперед, змінити напрям або припинити інвестування.

Створення узгодженої структури

Планування ефективно лише тоді, коли в організації узгоджено позицію щодо забезпечення цінності. Узгодженість стосується всіх, хто рухається в одному напрямі. Ви можете візуалізувати планування постачання продукту як очищення шарів цибулі, що показано на рис. 5.2.



Рис. 5.2. Планування постачання продукту вимагає узгодження між різними рівнями в організації

Scrum-фреймворк безпосередньо стосується двох внутрішніх (тобто короткотермінових) рівнів планування. Щоденна Scrum-нарада — це план на наступні 24 години для просування до цілі спринту. Ціль спринту створюється під час його планування та визначається замовленим беклогом продукту. Стратегію власника продукту щодо переходу до концепції продукту по суті відображено в беклозі продукту. Нарешті, щоби максимізувати цінність, концепція та стратегія розвитку продукту повинні узгоджуватися з бізнес-стратегією організації. Кожен рівень є іншим горизонтом планування в контексті продукту та того, як він вписується в напрям організації в цілому. Якщо поглянути на картину загалом, то, найвірогідніше, виникне низка запитань.

- Наскільки завчасно ви повинні планувати і з яким рівнем деталізації для кожного шару?
- Як часто слід перевіряти та адаптувати свої плани на кожному рівні?
- Хто повинен брати участь у плануванні зовнішніх шарів, які виходять за межі Scrum-команди?

Відповідь на всі ці запитання така: «Залежить від обставин!» Це залежить від складності вашого продукту та його відношення до організації в цілому, а також від середовища, у якому він використовується. Подумайте, як цикли планування у вашій організації забезпечують емпіричне планування та узгодження між загальними цілями бізнесу та роботою, яку виконує Scrum-команда.

Уточнення беклогу продукту

Уточнення беклогу продукту — це діяльність, яка допомагає забезпечити узгодженість. Беклог продукту — це план майбутніх спринтів; це життєвий план, який відкритий для змін і постійно розвивається, доки існує продукт. Отже, уточнення беклогу продукту — це те, як Scrum-команди планують наступний спринт, а також майбутні спринти, які потім впливають на плани вищого рівня для бізнес-сфер. На наших навчальних курсах люди часто запитують, чому вдосконалення беклогу продукту не є подією в Scrum. Оскільки ця

діяльність дуже залежить від контексту, вона є аспектом процесу роботи Scrum-команди, і її залишають на розгляд команди заради визначення найбільш ефективних методів та термінів.

Багато Scrum-команд намагаються знайти свій ритм за допомогою вдосконалення беклогу продукту. Виникають запитання, як часто це робити, скільки часу витратити на цю діяльність, наскільки детально це вивчати, хто бере участь та які практики застосовувати. Знову ж таки, це найліпше визначається Scrum-командою й найкраще засвоюється завдяки експериментам, вивченню та адаптації.

Удосконалення запланованих для виконання задач (беклогу) мінімально життєздатного продукту

Мета вдосконалення беклогу продукту є мінімальною, але достатньою. Елементи беклогу продукту мають атрибути опису, замовлення, оцінки й цінності. Цей набір атрибутів відображає мінімальну кількість інформації, якою повинна володіти ефективна команда перед початком роботи. Звичайно, усі ці атрибути можуть бути змінені в процесі отримання додаткової інформації¹.

Відповідно до наближення елементів беклогу продукту до спринту, Scrum-команда докладатиме певних зусиль, щоби розбити їх на менші фрагменти та додати деталі. Елементи беклогу продукту вважатимуть «готовими до роботи», коли команда розробників матиме достатнє спільне розуміння бажаної цінності та вважатиме, що елемент досить малий, щоби в рамках спринту він отримав статус «цілком готового».

На практиці: чи потрібне вам формальне визначення поняття «готовий до роботи»?

Ми бачили, як добре і як погано використовується додаткова практика офіційного визначення поняття «готовий до роботи». Таке визначення може бути корисним, коли Scrum-команда часто стикається із залежностями або прогалинами в розумінні між членами групи розробників, які найкраще вирішувати в процесі

вдосконалення задля уникнення затримок, тільки-но елемент беклогу буде внесено в спринт.

Наведемо кілька прикладів.

- Текст потребує перевірки юридичним відділом.
- Є потреба в оновленій графіці, яка відповідає вимогам щодо брендингу/стилю.
- Апаратне або програмне забезпечення потребує підготовки та налаштування.

Спосіб не використовувати офіційне визначення поняття «готовий до роботи» — розглядати елементи беклогу продукту як документ із «заблокованими» вимогами, де всі деталі повинні бути записані й затверджені. Це призводить до виникнення атмосфери, у якій «переговори про контракт важливіші за співпрацю»¹.

Не створюйте штучного бар'єра, який заважає роботі команди розробників або означає, що команда не може співпрацювати з власником продукту або відповідними зацікавленими сторонами. Елементи беклогу продукту варто продовжувати уточнювати навіть під час спринту, оскільки ми багато про що дізнаємося.

Scrum-команди повинні отримати шість переваг від процесу вдосконалення беклогу продукту.

1. *Підвищена прозорість.* Беклог продукту — «єдине джерело істини» для того, що планується в продукті. Додавання деталей збільшує прозорість того, що ви плануєте забезпечити, а також вашого прогресу.
2. *Уточнення цінності.* Коли ви з'ясовуєте деталі, що стосуються цінності, результати, яких ви намагаєтеся досягти за допомогою елемента беклогу продукту, стають зрозумілішими. Це допомагає команді розробників створити те, що потрібно. Таке уточнення може вплинути на формування запиту, на оцінку та замовлення, оскільки власник продукту та команда розробників з'ясовують, що ж насправді потрібно.
3. *Розбиття елементів на придатні для використання фрагменти.* Ви хочете, щоб елементи беклогу продукту були достатньо

¹ ScrumGuides.org

¹ Це протиставляється оцінці співпраці з клієнтами на підставі переговорів щодо контракту, що є однією з основних цінностей Agile-маніфесту.

малі, аби команда розробників могла завершити більше одного в спринті. Наявність декількох елементів беклогу продукту в спринті дає команді більшу гнучкість для досягнення цілі спринту та постачання «готового» приросту цінності продукту¹.

4. *Зменшення залежностей.* Залежності часто перетворюються на перешкоди. Хоча вам, можливо, і не вдасться уникнути усіх, та спробуйте зменшити їх там, де це вдається. Принаймні ви хочете, щоби залежності були прозорими.
5. *Прогнозування.* Доопрацьований беклог продукту в поєднанні з історичною інформацією про здатність Scrum-команди поставити продукт, який працює, допомагає вам прогнозувати. Прогнозування для деяких продуктів повинно розширюватися на кілька спринтів у майбутнє, щоби допомогти обговорити очікування від релізу із зацікавленими сторонами. Для інших продуктів прогнозування за межами поточного спринту не потрібне. Більшість продуктів потрапляють у проміжок між цими двома крайнощами.
6. *Застосування навчання.* Емпіризм полягає в застосуванні того, що ви здобуваєте під час створення продукту, оскільки ви краще розумієте, як реалізувати цінність продукту, і бачите зміни, що відбуваються у вашому середовищі².

У процесі накопичення досвіду ви дізнаєтеся, наскільки вам необхідні доопрацювання беклогу для полегшення планування та мінімізації потенційних утрат, що трапляються через надто велику кількість часу на переробку.

Оцінка

Мета оцінювання полягає в тому, щоби допомогти команді розробників спрогнозувати, що можна розробити в спринті, а власнику управляти беклогом продукту, зокрема визначаючи, чи має елемент достатньо передбачуваної цінності, щоби виправдати інвестиції (тобто гарантувати рентабельність інвестицій). Коли ми починаємо думати про ширшу картину прогнозування та планування за межами спринту, оцінки також можуть у цьому допомогти.

¹ Докладніше про те, як розділити користувацькі історії на менші елементи беклогу продукту, див. <https://agileforall.com/patterns-for-splitting-user-stories/>.

² Дізнайтеся більше про такі переваги з цієї публікації в блозі: <https://www.Scrum.org/resources/blog/art-product-backlog-refinement>.

Оцінка — це насправді лише «здогадка», яку ви робите з огляду на найкращу наявну інформацію про розмір, зусилля та складність елемента беклогу продукту. Оскільки це лише здогадка, ви повинні припускати, що кожна оцінка помилкова; не слід очікувати, що оцінки точно передбачать майбутнє, коли ви виконуєте складну роботу. Коли очікується, що люди будуть точними у своїх оцінках (за допомогою прямого стимулу або неявного показника ефективності), це часто призводить до підтасовування даних. Потім така «гра в цифри» потенційно може призвести до завищених оцінок (це означатиме, що вони більше не є прозорими і не служать своєму призначенню) або зрізання кутів для досягнення оцінки (це означає, що приріст цінності продукту більше не є прозорим і немає можливості постачати цінність). Scrum не робить приписів, як оцінювати, але вказує, що команда розробників володіє правом на оцінку, оскільки саме вона виконує роботу.

Члени команди несуть відповідальність за створення «готового» приросту цінності продукту й володіють беклогом спринту, а це означає, що вони вирішують, скільки роботи потрібно вкласти в спринт. Це важливий аспект самоорганізації.

Ви можете підійти до оцінки двома різними способами: оцінити необхідні зусилля, представлені годинами чи робочими днями, або зробити відносну оцінку, що означає порівняти частину роботи з чимось іншим на основі зрозумілої контрольної точки. Емпіризм та гнучке мислення схилиють команди до відносної оцінки, оскільки такий підхід допомагає їм брати до уваги складність та невідомість, ґрунтується на досвіді вже відомої роботи та мінімізує кількість часу, витраченого на оцінку (тобто потенційні втрати). Важливіше не те, які методи ви використовуєте, а те, як ви їх застосуєте і які переваги отримує Scrum-команда.

На практиці:

оптимізуйте командну роботу, емпіризм та гнучкість мислення завдяки методам відносного оцінювання

Ми досягли успіху, допомагаючи командам обрати метод відносної оцінки та максимально використати можливості групової оцінки. Наведемо короткий опис п'яти загальноновживаних методів відносної оцінки.

- *Відносні одиниці складності*: елементи беклогу продукту оцінюються за допомогою низки чисел, часто це послідовність Фібоначчі (1, 2, 3, 5, 8, 13 тощо), для присвоєння кожній задачі певної кількості балів залежно від розміру та складності.
- *Розміри футболок*: оцінки елементів беклогу продукту визначаються з використанням розмірів футболок, як-от XS, S, M, L та XL.
- *Тварини, фрукти тощо*: елементи беклогу продукту оцінюються за допомогою фізичного об'єкта для відображення відносного розміру. Наприклад, кавун більший за диню, яка більша за грейпфрут, а той більший за лайм тощо.
- *Елементи «однакового розміру»*: елементи беклогу продукту нарізають та розбивають так, щоб усі вони мали приблизно однаковий розмір.
- *Предмети «правильного розміру»*: по суті, це розбивання елементів на досить дрібні фрагменти, щоб хоча б один елемент можна було виконати за спринт.

Дослідження показують, що **групове оцінювання** точніше за індивідуальне. Воно вимагає від груп розмаїтості й незалежності, а також децентралізації влади всередині групи. Найкращі рішення приймаються внаслідок продуктивного конфлікту, коли вислуховуються всі погляди й група може дійти консенсусу¹.

Розгляньте можливість залучення до оцінки всієї Scrum-команди. Усі розробники можуть запропонувати унікальні погляди на основі власного досвіду, знань та навичок. Хоча власник продукту не оцінює роботу, участь цієї особи допомагає уточнити мету та бажану цінність елементів беклогу продукту. Scrum-майстер гарантує хорошу взаємодію в команді й допомагає їй не потрапляти в антишаблони проектування, пов'язані з оцінкою (наприклад, прикріплення, аналітичний параліч)².

Розбиття запланованих для виконання елементів продукту (беклогу) для зосередження на результатах

Команди часто вступають в боротьбу з розбиттям великих функціональностей та функцій на цінні елементи беклогу продукту, які є досить малими, щоби мати можливість отримати характеристику «готових» у спринті. Ми часто бачимо, що увага зупиняється на «достатньо малій» задачі, проте «цінність» задачі втрачається з поля зору.

Уточнення беклогу надає прозорість щодо планів команди зі створення того, що надає цінність як Scrum-команді, так і зацікавленим сторонам. Для побудови правильного продукту Scrum-команда повинна мати спільне розуміння бажаних результатів. Власник продукту отримує більше гнучкості та розуміння того, як замовити беклог продукту для оптимізації вартості. Зацікавлені сторони отримують більше прозорості щодо зв'язку роботи, яку виконує Scrum-команда на рівні спринту, з масштабнішими завданнями, що їх бізнес хоче досягти для продукту.

Розгляньмо методи колективної візуалізації, як-от візуалізація користувацької історії, що дозволяють Scrum-команді та зацікавленим сторонам бачити як загальну картину, так і можливі способи її розбиття без випускання з поля зору результатів і цінності для користувачів¹.

Незалежно від того, які методи ви використовуєте, переконайтеся, що ваша Scrum-команда та зацікавлені сторони отримують шість описаних раніше переваг від розбиття елементів беклогу продукту. Знайте, чому ви використовуєте ту чи іншу методику, і регулярно перевіряйте, наскільки вона відповідає вашим потребам. Пристосовуйте методики, щоби вони краще працювали для вас.

Планування спринту

У процесі планування спринту Scrum-команда повинна відповісти на три запитання.

- Яка робота є правильною для зосередження на цьому спринті і наскільки ми її розуміємо?

¹ James Surowiecki, *Wisdom of Crowds* (Anchor, 2005).

² Щоби дізнатися більше про методи оцінювання, див. книжку: Mike Cohn, *Agile Estimating and Planning* by (Prentice Hall, 2005).

¹ Щоби дізнатися більше про візуалізацію користувацьких історій, див. книжку: Jeff Patton. *User Story Mapping* (O'Reilly Media, 2014).

- Скільки ми можемо отримати «готового» приросту цінності за спринт?
- Скільки часу нам потрібно витратити на вдосконалення роботи на цьому спринті?¹

Ми вже обговорювали перше запитання. Тут ми зупинимося на двох інших.

Скільки «готового» приросту цінності ви можете отримати за спринт?

Плануючи спринт, члени команди розробників обирають елементи з беклогу продукту, які, на їхню думку, вони можуть завершити в поточному спринті для досягнення цілі спринту. Скільки вони можуть отримати «готового» приросту, залежить від їхнього досвіду роботи як команди, ефективності процесів цієї команди та підтвердженої здатності робити подібну роботу.

У процесі розвитку своєї ідентичності та уточнення й удосконалення своєї здатності створювати якісні прирости цінності, які підлягають релізу, Scrum-команди стануть краще інтуїтивно розуміти, скільки продукту вони можуть створити за певний проміжок часу. Спринт допомагає забезпечити гармонію або ритм, щоб це інтуїтивне відчуття зростало з часом.

Наведемо кілька загальних проблем, які ускладняють розвиток цього інтуїтивного відчуття:

- часта зміна складу команди;
- відсутність співпраці;
- часта зміна довжини спринтів;
- члени команди розподіляються по інших командах або вилучаються;
- часті перебої, що призводять до переключення контексту²;
- частково виконана робота в кінці спринту.

¹ Оновіть свої знання про основи заходів із планування спринту (ціль, ресурси/вхідні параметри, результати) тут: <https://www.Scrum.org/resources/what-is-sprint-planning>.

² Після кожної перерви людині може знадобитися більше 20 хвилин, щоб повернутися туди, де вона зупинилася; див. https://www.washingtonpost.com/news/inspired-life/wp/2015/06/01/interruptions-at-work-can-cost-you-up-to-6-hours-a-day-heres-how-to-avoid-them/?noredirect=on&utm_term=.8f892c76b64d.

Щоби подолати ці виклики, прагніть до створення більшої стабільності та зосередженості. Чи може змінитися склад команди? Так, але це слід робити навмисно і в такий спосіб, щоб забезпечити команді достатній контроль. Чи може змінюватися довжина спринту? Так, але це також слід робити навмисно. Чи можуть бути перебої? Так, але це повинно відбуватися з поважних причин і робитися так, щоб мінімізувати їхній негативний вплив.

Насправді, що довший спринт, то складніше його планувати. Більші спринти мають вищу складність і більше невідомих. Якби ми попросили вас спланувати все, що ви можете виконати впродовж наступного місяця, це завдання, найімовірніше, здалося б вам надскладним. Але якби ми попросили вас спланувати свої досягнення на наступний тиждень, це, мабуть, виявилось б набагато простішим¹.

На практиці: скільки має тривати спринт?

«Спринт повинен бути якомога коротшим, але не коротшим за необхідний».— Кен Швабер

Scrum каже, що спринт повинен тривати місяць або менше. Але звідки ви знаєте, що для вас підходить? Тривалість спринту визначають два наведені нижче фактори:

1. Як швидко бізнесу необхідно змінити напрям?
2. Як швидко команда розробників може створити «готовий» приріст цінності продукту?

Перший фактор стосується надання бізнесу змоги скористатися можливостями, реагувати на зміни на ринку та керувати інвестиційним ризиком. Власник продукту також мусить брати до уваги й те, як часто будуть збиратися відгуки та нова інформація, беручи до уваги цикл перевірки та адаптації спринту.

¹ Щоби дізнатися про підхід до організації події, присвяченій плануванню спринту, перегляньте це коротке відео: <https://www.Scrum.org/resources/effective-sprint-planning>.

Часто ми бачимо, що Scrum-команди зосереджуються на другому факторі. Але якщо відповідь на друге запитання обмежує гнучкість бізнесу, тоді команда розробників повинна подумати, як вдосконалити необхідні процеси, інструменти та можливості для задоволення бізнес-потреб. Наприклад, якщо ринок змінюється так часто, що чотирьох тижнів стабільної роботи ніколи не буде в наявності, тоді чотиритижневий спринт не відповідає потребам бізнесу.

Після початку спринту його довжина не змінюється. Однак Scrum-команда може змінити довжину спринту для безперервного вдосконалення (зазвичай визначається в ретроспективі спринту, коли завершується поточний спринт), і це рішення застосовується до всіх майбутніх спринтів. Сенс у тому, щоби дозволити команді ввійти в ритм.

Узгодження довжини спринту з «пульсом організації» може принести користь. Наприклад, коли кілька Scrum-команд працюють над одним продуктом або набором продуктів, координація роботи та усунення залежностей шляхом вирівнювання періодичності стають більш важливими.

На практиці: прогнозування спринту на основі емпіричних даних

Швидкість — це додаткова практика, яка забезпечує вказівку середнього обсягу беклогу продукту, перетвореного командою розробників на «готовий» приріст цінності продукту під час спринту. Команда розробників відстежує його для використання в Scrum-команді. Коли з плином часу збираються історичні дані, Scrum-команда отримує уявлення про швидкість, з якою буде завершено роботу, а потім може скористатися цими знаннями для прогнозування майбутньої роботи (тобто для вибору елементів беклогу продукту для беклогу спринту).

Однак швидкістю можна легко зловживати. Дуже важливо розуміти, що нею не є:

- показник ефективності роботи команди розробників;
- обіцянка того, що буде поставлено в майбутньому;

- спосіб порівняння Scrum-команд або команд розробників;
- спосіб порівняння людей у команді розробників;
- індикатор того, наскільки важко працює команда розробників.

На практиці: імовірнісне прогнозування

Імовірнісне прогнозування — це підхід, за допомогою якого дані історичного процесу використовуються разом із методом статистичної вибірки (наприклад, моделювання Монте-Карло), щоби зробити прогноз щодо часу завершення певного елемента чи елементів. Цей прийом популярний серед потокових процесів, оскільки показники потоку особливо сприяють статистичному аналізу.

Результатом такого прогнозу є твердження про майбутнє, сформульоване за допомогою ймовірнісних показників з певним узгодженим рівнем достовірності. Інакше кажучи, імовірнісний прогноз містить цілу низку можливих майбутніх результатів, а також рівень довіри до цього діапазону. Наприклад, команда може використовувати історичні дані про час циклу, щоби запропонувати очікування, що окремі елементи рухатимуться в процесі «за 8 днів або менше 85 % часу». Або вони можуть використовувати історичні дані стосовно пропускну здатності як вхідні дані в симуляцію Монте-Карло для прогнозування шансів заповнити всі елементи беклогу продукту як «не пізніше 1 жовтня з упевненістю 95 %».

Ефективність імовірнісного прогнозування полягає в тому, що воно охоплює внутрішню невизначеність, пов'язану з передбаченням майбутнього. Щоразу, коли присутня невизначеність, необхідний імовірнісний підхід. Ба більше, розуміння ймовірностей, пов'язаних з певними результатами, є вигідним під час оцінки ризику, який супроводжує цей план¹.

¹ Детальніше про ймовірнісне прогнозування див. у книжці: Daniel Vacanti, *When Will It Be Done*; <https://leanpub.com/whenwillitbedone>.

Скільки часу слід витратити на вдосконалення цього спринту?

Завжди існує необхідність постачати більше цінності. На жаль, це часто трактується як «постачати більше». Ми нерідко бачимо, що для отримання більшої цінності Scrum-командам потрібно зосередитися на поліпшенні спільної роботи, удосконаленні своїх інструментів, усуненні технічного боргу та впровадженні нового навчання.

2017 року «Посібник зі Scrum» було оновлено, щоби чітко вказати: беклог спринту має містити принаймні одне високопріоритетне поліпшення процесу, визначене в попередній ретроспективі спринту¹. Скільки часу витрачати на постійне вдосконалення, залишається на розгляд Scrum-команди. Якщо команда продовжує намагатися створити «готовий» приріст цінності продукту під час кожного спринту, учасникам команди, імовірно, доведеться витратити більше часу на вдосконалення своїх практик, інструментів та взаємодії. Звичайно, це означає, що просто зараз потрібно почати брати на себе менше роботи, водночас визнаючи, що такі інвестиції в поліпшення створять можливості для збільшення потоку цінності в майбутньому.

Наскільки далеко ви від удосконалення

Власник продукту забезпечує наявність здорового беклогу продукту, у якому міститься «достатньо» досконалої роботи, зрозумілої команді розробників. Прагнення до визначення пріоритетів чітко видно завдяки порядку беклогу продукту й загальному розумінню цінності та її співвідношення з бізнес-цілями. Команда розробників визначає, скільки вона може витратити в спринті, і вчиться правильно збалансовувати час. Вони можуть використовувати емпіричні дані, зокрема, яку частину беклогу продукту вони закінчили в попередніх спринтах, для прогнозування того, скільки вони можуть закінчити в наступних спринтах, але цей прогноз також бере до уваги кількість очікуваних змін у меню та потреби розробників упродовж прогнозованого періоду.

¹ Цього завжди очікували від самого зародження Scrum, оскільки перевірка без пристосування безглузда.

Вам, можливо, також знадобиться прогнозування на більш високих рівнях. Вірогідно, вам потрібно спрогнозувати, коли для досягнення бізнес-мети, імовірно, буде поставлено підмножину елементів беклогу продукту, а потім заради наступної бізнес-мети... й наступної. Можливо, вам потрібно спрогнозувати, коли буде поставлено конкретний елемент беклогу продукту.

Планування релізів

Мета планування полягає в спілкуванні та керуванні очікуваннями. Ми також плануємо, щоби було чим виміряти прогрес. В умовах стрімкого розвитку сучасного суспільства люди шукають швидших шляхів розв'язання своїх проблем. Щоб їх задовольнити, треба або попередити проблеми, або виховати в людей бажання певної функції, або бути надзвичайно чутливим. Цей підхід лежить в основі Scrum, тобто в можливості випускати продукт якнайшвидше, беручи до уваги контекст його існування та ринок, на якому він використовується. Реліз — це те, як ви насправді поставляєте цінність для користувачів/клієнтів.

Слово «реліз» часто зустрічається в «Посібнику зі Scrum», проте Scrum не передбачає стратегій релізу або його планування. Що Scrum насправді нам каже, так це те, що як частину відповідальності за оптимізацію цінності власник продукту несе відповідальність за рішення, коли випустити «готовий» приріст цінності продукту.

Дехто запитує: «Чому реліз не планує подію Scrum?» Ну що ж, у вас зовсім немає необхідності здійснювати реліз у спринті. Фактично ви можете пройти кілька спринтів до релізу. Або ж можете провести реліз кілька разів за спринт — можливо, щоразу, коли елемент беклогу продукту відповідає визначенню поняття «виконано». Інженери Netflix роблять релізи тисячі разів на день. Оскільки вони проводять безліч невеликих експериментів, ризик будь-якої з цих змін, що спричиняє проблему, дуже низький, і вартість відшкодування цих змін також дуже низька¹. Навпаки, команда, яка розробляє програмне забезпечення для імплантованих медичних пристроїв, проводи-

¹ Короткий вступ до сучасних методів проведення релізів див. <https://medium.com/data-ops/how-software-teams-accelerated-average-release-frequency-from-three-weeks-to-threeminutes-d2aaa9cca918>.

тиме релізи дуже рідко, оскільки ризик заподіяння шкоди в разі виникнення помилки дуже великий, а вартість повторного випуску або відкликання виробу надзвичайно висока. У кожного продукту своя ідеальна частота релізу, яка відображає ризик та вартість змін.

Наскільки великим повинен бути реліз?

Стратегія релізу залежатиме від типу продукту, середовища, у якому він використовується, та можливостей Scrum-команди. Отже, рівень необхідного планування та його спосіб будуть значною мірою залежати від стратегії релізу. Планування релізу найкраще вважати частиною емпіричної діяльності Scrum-команди, щоби вона визначала, коли і як найкраще це зробити, аби максимізувати вигоди та зменшити втрати.

Наскільки малим за обсягом може бути реліз?

Найменший реліз, який ви маєте розглядати,— це *новий або поліпшений результат* для одного образу споживача. Якщо ви не постачатимете принаймні такий же обсяг своїм клієнтам, вам доведеться докласти чимало зусиль і ваші клієнти не отримають жодної вигоди. Якщо ви поставили більше одного нового або поліпшеного результату, ви завадили своєму клієнту отримати вигоду від одного з результатів раніше.

Можливо, вам не вдасться випустити окремий результат за раз. У вас може бути занадто багато фізичних зусиль або складності в процесі релізу, щоби його уможливити. Вам може не вистачати достатнього тестування для забезпечення якості або автоматизації ключових операцій для релізу. Незалежно від частоти проведення релізів або рівня приросту, усунення цих бар'єрів допоможе вам поліпшити якість, вартість та частоту релізів вашого продукту¹.

¹ Для отримання додаткової інформації про стратегії планування релізу див. видання; *The Professional Product Owner* Don McGreal, Ralph Jocham, яке є частиною серії *Professional Scrum* від Scrum.org, опублікованої Pearson Addison-Wesley (2018).

Підсумки

Плани — це живі артефакти: важлива діяльність із планування. Працуйте над тим, що ви можете контролювати, створюючи спільне розуміння, досягаючи справді добрих результатів у постачанні невеликих за обсягом цінностей і переконуючись, що завжди працюєте над наступною потрібною річчю. Буде легше пристосувати свій план тоді, коли зміни неминуче відбудуться.

Беклог продукту в поєднанні з емпіричними даними — усе, що вам необхідно для планування та прогнозування. Це дійсно так просто. З часом ви будете корегувати свій план у процесі навчання, виконання роботи та спостереження за тим, що змінюється навколо вас.

- Беклог продукту — це план майбутніх спринтів.
- Беклог спринту — це план поточного спринту.
- Удосконалення беклогу продукту є плановою діяльністю.

Прагніть отримати максимальну віддачу від планування діяльності, одночасно зменшуючи втрати. Прийміть той факт, що плани будуть розвиватися, і будьте пильні щодо нових планів у процесі навчання. Забезпечте прозорість прогресу, щоби ви могли встановлювати реалістичні очікування та відкрито й чесно обговорювати складність та непередбачуваність, властиві розробці продукту.

Заклик до дії

Розгляньте зі своєю командою наведені нижче запитання.

- Чи здається діяльність із планування вам легкою, важкою?
- Наскільки добре ви співпрацюєте в команді під час планування та прогнозування?
- Наскільки змінився беклог продукту від спринту до спринту? Який обсяг беклогу продукту є сенс отримати з позначкою «готовий до роботи»?
- Наскільки добре Scrum-команда та зацікавлені сторони розуміють цінність, бажану для окремих елементів беклогу продукту?

- Які переваги можна отримати від частіших релізів? Що заважає вам частіше здійснювати релізи?
- Як цикли планування у вашій організації дозволяють емпіричне планування та узгодження між ширшою картиною бізнес-цілей загальної картини й роботою, яку виконує Scrum-команда?
- Які проблеми зараз найбільше дошкуляють? Визначте один або два експерименти, які допоможуть підвищити ефективність планування. Для кожного експерименту обов'язково вкажіть бажані впливи та спосіб їх вимірювання.

Розділ 6

Допомога Scrum-командам у розвитку та вдосконаленні

Удосконалення планування важливе, але це лише один невеликий аспект того, як Scrum-команда може поліпшити свою спільну роботу для досягнення цінності. Потреба в цих удосконаленнях часто виявляється в ретроспективі спринту, але вона також може бути виявлена й завдяки відгукам, отриманим в оглядах спринту, та перешкодам, що виникають під час його перебігу.

Хто допомагає Scrum-команді вдосконалюватися? Безумовно, Scrum-майстер, але не лише він один. У багатьох ситуаціях члени команди допомагають одне одному вдосконалюватися, а в ідеалі й решта організації допомагає із цим команді. Керівники організації можуть зіграти велику роль у допомозі колективу. Та зрештою, сама Scrum-команда відповідає за самовдосконалення та просить про допомогу, коли її потребує.

Використання ретроспективи спринту для виявлення сфер для вдосконалення

Коли Scrum-команді не вдається поставити «готовий» приріст продукту, у центрі уваги ретроспективи спринту має бути те, як команда може досягти позначки «виконано» в наступному спринті. Команда, яка здатна ефективно самоорганізуватися, повинна мати можливість зробити крок назад і зрозуміти, що перешкодило їй дійти до «готового» приросту, але сам факт, що команда не досягла результату, указує на те, що вона може бути неефективною, принаймні поки що.

Так само, якщо клієнти або інші зацікавлені сторони незадоволені цінністю, наданою внаслідок спринту (звичайно, передбачається, що Scrum-команда дістається «готового» приросту), це опиниться в центрі уваги ретроспективи спринту. Наскільки ефективно власник продукту працює з командою розробників для створення спільного розуміння цінності? Наскільки ефективно Scrum-команда працює із зацікавленими сторонами для кращого розуміння та надання цінності?

Scrum-майстер відіграє вирішальну роль у тому, наскільки ефективно Scrum-команда діагностує проблеми й інші перешкоди. Команді може знадобитися допомога для подолання власноруч створених меж. Ми бачили, як команди перешкоджають виконанню власних завдань, накладаючи обмеження та ліміти на те, що їм дозволено робити. Scrum-майстер повинен пояснити команді, що в її рамках необхідно ставити під сумнів, що фіксувати й проти чого слід постійно й рішуче боротися.

На практиці:

використовуйте факторний аналіз, щоб зрозуміти мотивацію до змін

Один зі способів виявити мотивацію команди до змін — провести простий факторний аналіз з вашою групою. Цей процес окреслить фактори, які впливають на зміни¹. Після переліку факторів, що впливають на пропоновану зміну, зважте їх з огляду на їхній вплив. Рис. 6.1 зображує фактори, що впливають на визначення поняття «виконано».

Хороший початок — зробити процес роботи видимим і забезпечити прозорість будь-якої перешкоди, яка сповільнює роботу Scrum-команди. Це може розпочатися з моменту, коли відбувся запит на функцію, й охоплювати весь процес упровадження у виробниче середовище. Команди та організації зазвичай схильні сприймати зміни лише тоді, коли вартість без змін перевищує вартість зі змінами. Факторний аналіз може допомогти вам обговорити в команді питання про те, чи будуть зміни окупними.

¹ Детальніше про факторний аналіз див. www.mindtools.com.



Рис. 6.1. Факторний аналіз «готового» приросту

З нашого досвіду, Scrum-команди найбільше стикаються з труднощами в ретроспективі спринту. Мало хто звик до саморефлексії та активно шукає сфери, у яких потрібно вдосконалюватися. Набагато простіше зосередитися на тому, як зробити продукт надійнішим («Ми повинні вдосконалити цю функцію для підтримки інших сфер бізнесу») або скаржитися на інші команди («З ними завжди так складно працювати»). Справжня саморефлексія означає бути сприйнятливим, протистояти обмеженням і прагнути побачити інші погляди, але для більшості людей це здається дуже неприємним.

На практиці:

поради щодо фасилітації ретроспектив спринту (та будь-яким іншим робочим зустрічам)

Фасилітація (сприяння) — це важлива навичка, коли йдеться про створення простору для висловлювання ідей та поглядів кожного, а також розкриття творчого потенціалу групи, наділеної різноманітним досвідом, знаннями та навичками в різних галузях. Мета полягає у створенні продуктивного конфлікту в такий спосіб, щоб залучати всіх до позитивної та творчої співпраці, одночасно

отримуючи рішення на основі консенсусу та чіткі результати¹. Наведемо кілька порад щодо вдосконалення ваших ретроспектив спринту, які можна застосувати до будь-якої сесії спільної роботи.

- *Використовуйте тишу.* Зазвичай прийоми мовчазного сприяння створюють умови для людей з особистими вподобаннями, що може спонукати їх спершу обдумати, а потім говорити, або змусити вагатися перед тим, як перебивати інших під час жвавого обговорення на підвищених тонах. Ці прийоми також можуть стати в пригоді, коли в грі задіяно сильні емоції, і ви хочете знизити рівень конфлікту, або, навпаки, коли побутує штучно створена гармонія і вам потрібно створити певний конфлікт.
- *Використовуйте різні групові конфігурації.* Під час групових обговорень може виникнути багато проблем, особливо в процесі збільшення розміру групи. У розмові можуть домінувати одна або дві людини. Слухаючи, як говорять інші, люди можуть формувати запитання та нові ідеї, але зрештою чекають, доки не побачать можливості заговорити. На цьому етапі їхні коментарі можуть здатися нерелевантними або навіть забутись. Крім того, декому зручніше говорити один на один або в невеликих групах.
- *Змушуйте людей рухатися і діяти.* Це одразу охоплює й задіює всіх. Люди рідше відволікаються або нудьгують. Наприклад, якщо люди змінюють партнерів або групи, вони рухаються фізично. Інший приклад — це попросити людей зробити свої розмови та результати видимими за допомогою фліпчартів.
- *Зробіть струс.* Ми маємо на увазі дещо шокувати людей, закликаючи їх думати по-іншому і створювати нові погляди. Зміна форматів та прийомів фасилітації допомагає запобігти простому «виконанню дій» або втраті інтересу.
- *Приходьте з розпорядком дня, але будьте гнучкими.* Зазвичай ви прагнете (1) переглянути прозору інформацію, щоб (2) отримати нову інформацію, а потім (3) ухвалити рішення про те, які дії буде вжито для вдосконалення. Складіть план виконання

¹ Фасилітація — це ціла професія, і ми закликаємо вас до дослідження безлічі доступних ресурсів. Щоб отримати глибше розуміння того, як розробляти ефективні ретроспективи та сприяти цьому процесу, див. книжку: Esther Derby, Diana Larsen. *Agile Retrospectives: Making Good Teams Great* (Pragmatic Bookshelf, 2006).

кожної з цих речей, зокрема визначте час, який буде витрачено на кожну з них. Але будьте гнучкими й реагуйте на потреби команди, дотримуючись мети заходу та часових рамок (таймбоксу).

Виявлення та усунення перешкод

Щоб отримати потужну ретроспективу спринту, члени Scrum-команди повинні вміти виявляти перешкоди, в ідеалі — упродовж усього спринту, що заважають їм створити приріст, який придатний для релізу і відповідає цілі спринту. Перешкодою є те, що блокує або сповільнює здатність Scrum-команди поставляти цінний продукт, придатний для релізу. Команда розробників та власник продукту можуть і повинні самостійно усувати деякі перешкоди (наприклад, як вони виконують свою роботу на рівні власної відповідальності), можливо, за допомогою настанови та підтримки від Scrum-майстра. Перешкоди, які члени Scrum-команди не можуть усунути самостійно, бере на себе Scrum-майстер.

Завдяки розумінню системи, у якій працює команда в цілому, ви зможете визначити, що стоїть на заваді або є фактором стримування. Іноді ви можете прискорити рух, просто відпустивши гальмо.

Scrum-команди максимізують потік, коли вони переміщують елементи в процесі якомога швидше, не спричиняючи будь-якого ризику для якості та задоволення споживачів. Видалення відходів дозволяє максимізувати потік, а під «відходами» ми маємо на увазі все, що не додає цінності для клієнта. Щоби максимізувати потік, слід звернути увагу не лише на *блокування* перешкод, а й зосередитися на речах, які уповільнюють роботу команди та заважають їй постачати максимальну цінність; не чекайте, доки перешкоди стануть повноцінними бар'єрами.

Команди вдосконалюються завдяки створенню більшої прозорості щодо речей, які можуть уповільнювати їх у процесі роботи, задля уможливлення визначення перешкоди, а відтак отримання розуміння, як найкраще усунути цю проблему. Scrum-дошка — це узвичаєна практика візуалізації прогресу роботи. Візуалізація роботи допомагає команді розробників зрозуміти, коли робота заблокована або повільно просувається, і, до речі, є багато деталей, які необхідно візуалізувати, крім простого використання Scrum-дошки.

На практиці: створіть прозорість за допомогою «змії відходів»

На рис. 6.2 зображено відходи в роботі команди. Ви можете створити схожий на змію ланцюжок наліпок, кожна з яких підсумовує те, що повинна зробити команда і що її члени вважають марнотратством. Підготуйте команду, обговоривши джерела відходів та інформацію, яку потрібно зафіксувати на кожній наліпці, наприклад ініціали особи, коротку заяву про надлишкову діяльність, категорію відходів та скільки часу було витрачено на таку діяльність. Ця вправа забезпечує простий спосіб визначення членами команди зайвого під час роботи в спринті. Ви можете використовувати її в ретроспективі спринту, щоб отримати повнішу картину тенденцій і витрат, а потім обговорити дії щодо зменшення надлишкової діяльності.

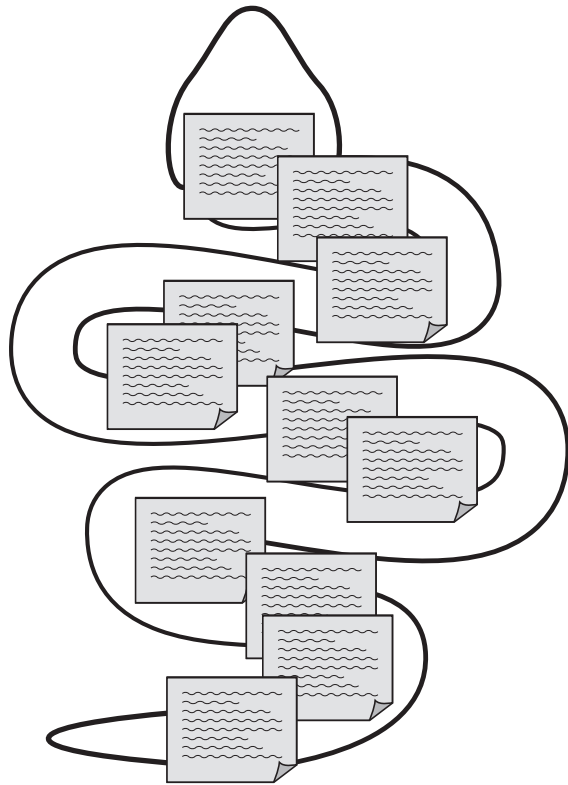


Рис. 6.2. «Змія відходів» візуалізує надлишкову діяльність команди

До розповсюджених різновидів надлишкової діяльності належать:

- ручне регресійне тестування;
- середовища, недоступні за потреби, зокрема збої та перезавантаження;
- початок роботи над елементом беклогу продукту, а потім виявлення, що він залежить від іншої команди і не може бути розв'язаний під час спринту;
- заповнення запитів на виконання іншими командами та затримки у відповідях (наприклад, доступ до логів, зміна налаштувань мережі, перезапуск системи);
- заповнення форми перевірки безпеки, яка містить багато нерелевантних або незрозумілих питань, що забирає багато часу і часто вимагає додаткових уточнювальних заходів;
- наявність члена команди, якого постійно перебивають і йому доводиться витрачати час, щоб повернутися до роботи;
- погано задокументований аспект системи, який змушує учасників команди перевчатися, запитуючи інших членів команди та проводячи дослідження щоразу, коли їм потрібно працювати в цій галузі;
- надто детальний опис «як» для всіх елементів беклогу продукту під час планування спринту та необхідність переробляти багато завдань у беклогу спринту під час розгортання роботи.

Відстеження перешкод та кількісна оцінка наслідків

З огляду на численні перешкоди, з якими стикаються команди в процесі виконання складної роботи, цілком імовірно, що не всі перешкоди будуть негайно усунені одразу після їх виявлення в спринті. Зробіть ці дані видимими та шукайте тенденції, указуючи, які перешкоди перебувають під контролем Scrum-команди, які під її впливом, а які стосовно неї є зовнішніми (див. рис. 6.3 на с. 154).

Наведемо кілька корисних запитань для кращого розуміння наслідків.

- Скільки часу ми витрачаємо на боротьбу з перешкодами?
- Як часто виникає ця перешкода?

N	Перешкода	Наслідки	Дата визначення	Частота	Дія	Відповідальність
1	Не використовується більш раннє середовище, досить близьке до виробничого	Проблеми з розгортанням, які впливають на клієнтів (найважливіший реліз у березні)	Спринт 03	Кожен реліз	Організація за-мовлення для забезпечення фінансування за-ради поліпшення середовища та додавання цього до беклогу продукту. Команда розробників працюватиме безпосередньо з інфраструктурою для розробки рішення	Власник продукту
2	У команди розробників немає доступу до бази даних	Затримки, пов'язані з усуненням несправностей під час спринту зазвичай займають від 2 годин до доби	Спринт 03	У середньому тричі за спринт	Виникне необхідність застосування винятків із наявної в організації політики. Scrum-майстер працюватиме з IT-менеджментом для отримання деталей про процес створення винятків	Scrum-майстер
3	Команда розробників часто залучається до зустрічей із метою консультацій щодо інтеграції в майбутні проекти	Переривання роботи/перемикання контексту вимагає часу для роботи над поточною метою спринту	Спринт 04	Спринт 03 — 10 годин, спринт 04 — 14 годин	Для обговорення — «Як ми можемо підтримати інші команди в організації, приділяючи належну увагу постачанню?»	Для обговорення

Рис. 6.3. Таблиця прикладів перешкод

- Як наслідок впливає на якість? Збільшується чи зменшується кількість виявлених у виробництві дефектів? Якою є вартість виправлення дефекту у виробництві порівняно з виправленням дефекту, виявленого під час розробки?
- Як перешкода впливає на моральний дух? Чи звільняються працівники через те, що незадоволені перешкодою? Якою є вартість найму когось нового?

На практиці: відстеження переривань у динаміці

Відстеження переривань як явного типу перешкоди допомагає кількісно оцінити вплив втрати уваги. Не має значення, чи команда насправді переймається через роботу, проте такі переривання змушують її членів утрачати продуктивний час, доки вони намагаються відновити свою концентрацію.

Ви можете дуже просто відстежувати ці дані. Намалюйте в кімнаті команди часову шкалу з відведеним місцем для кожного дня спринту. Потім щоразу, коли члена команди розробників переривають незапланованою роботою, яка не відповідає поточній цілі спринту, він або вона може написати про це на наліпці й розмістити її на часовій шкалі. Може бути корисною вказівка на джерело запиту (наприклад, іншу команду, менеджера, користувача, виробничу проблему) та кількість часу, витраченого на розв'язання проблеми. Цю інформацію можна переглянути в ретроспективі спринту, щоби допомогти Scrum-команді проаналізувати наслідки та допомогти їй виробити дієві зобов'язання, спрямовані на вдосконалення. Прикладами переривань можуть бути такі події:

- помилки, інциденти та питання підтримки виробництва;
- додаткова робота, доручена керівником команди;
- запити на час з командою профільних експертів;
- «приїзд» зацікавленої сторони, яка бажає обговорити з розробником нову ідею;
- запити на оцінювання потенційної майбутньої роботи;
- вигнання з конференц-залу та необхідність переміститися в інше місце для завершення сеансу спільної роботи.

Не всі переривання в роботі є поганими, але підвищення прозорості таких перебоїв та їхнього впливу допомагає Scrum-команді визначити, як зробити більш цінними й ефективними необхідні перерви (наприклад, запланувати конкретний час для певного виду діяльності) та як усунути або зменшити непотрібні.

Усунення перешкод

Щойно перешкоди стануть видимими, ви зможете пояснити необхідність інвестування часу й грошей для їх усунення. Звичайно, вам потрібно буде розставити пріоритети.

Наведені нижче запитання можуть допомогти вам вирішити, з чого почати.

1. Якої мети нам потрібно досягти? Яким є наш бажаний результат?
2. Що станеться, якщо ми не усунемо цю перешкоду зараз?
3. Що б ми зробили, якби могли зробити що завгодно для розв'язання цієї проблеми? Які рамки обмежують наші можливості? Чи насправді вони є обмеженнями або ми робимо припущення?
4. Якщо нам перешкоджають організаційна політика чи стандарт, чи можемо ми їх змінити, призупинити чи обійти?
5. Як ми дізнаємося, що ситуація (пов'язана з бажаним результатом) поліпшується?
6. Чиї знання, підтримка або вплив нам потрібні?

На практиці: залучення зацікавлених сторін до усунення перешкод

Не бійтеся говорити про певні перешкоди в огляді спринту. Деякі проблеми найкраще долати в Scrum-команді під час ретроспективи спринту, але може бути слушним порушення питання про

організаційні перешкоди, які вимагають інвестування в Scrum-команду, у контексті огляду спринту. У цих сесіях задіяне широке коло зацікавлених сторін, і вони можуть мати вплив (або гроші), щоби допомогти в усуненні цих перешкод.

Scrum-майстри відповідають за усунення перешкод, яких Scrum-команда не може позбутися самотійно. Однак вони не повинні відчувати, що мусять розв'язати ці проблеми самотужки. Часто менеджери або інші працівники організації (наприклад, менеджери проєктів) добре обізнані стосовно її правил роботи та історії. Scrum-майстри повинні прагнути до співпраці з людьми, які можуть надати додаткові знання, розуміння та вплив на подолання організаційних перешкод. А часом Scrum-майстру просто необхідно попросити когось надати невеличке «прикриття».

На практиці: не зосереджуйтеся на швидкості

Багато організацій прагнуть гнучкості заради поліпшення власних «темтів просування», що змушує їх зосередитися на поліпшенні швидкості. Швидкість — це емпіричне свідчення, історичний факт, який корисний лише для Scrum-команди. Зазвичай, коли менеджер хоче знати, як «поліпшити швидкість», він насправді просто запитує: «Як ми можемо частіше постачати більшу цінність?» З нашого досвіду, найкраще зосередитися на усуненні перешкод та видаленні відходів. Зосередження на швидкості може призвести до зрізання кутів і зниження якості через поспіх «швидшого просування». Зосередження уваги на видаленні відходів та усуненні перешкод, навпаки, підвищує якість і залишає більше часу для першочергової роботи, водночас підвищуючи швидкість постачань.

Підвищення або зниження швидкості безглузде без контексту. Команда розробників може частіше та якісніше постачати більшу цінність без будь-яких змін у швидкості, просто беручи до уваги вдосконалення у своїй роботі під час оцінювання.

Натомість, зосередьтеся на усуненні того, що стримує Scrum-команду. Нехай вона має власні «показники постачання» та зосереджує увагу зацікавлених сторін на «показниках цінності»¹.

Зростання індивідуальних та командних можливостей

Scrum сам собою не усуває проблем. Натомість, працівникам доводиться розв'язувати проблеми, які виявляє Scrum-фреймворк, створюючи цінні рішення завдяки гнучкому мисленню та командній роботі. Для цього їм потрібно надати простір та підтримку, щоби вони розвивали свої можливості. Кожен член Scrum-команди повинен відточити свою «технічну майстерність».

Під «технічною майстерністю» ми маємо на увазі не лише інженерні навички або майстерність розробки програмного забезпечення. Власники продуктів повинні розвивати широкий спектр умінь управління продуктами, тоді як командам розробників потрібні навички, пов'язані зі створенням придатного для релізу приросту цінності. Кожен повинен перевірити та пристосувати свої вміння, знання та можливості й подивитися, де Scrum-команда повинна зрости, щоби мати змогу відповідати новим викликам та потребам бізнесу.

Знайдіть час для постійного навчання та зростання

Розробка продуктів є складною й постійно змінюється. Нові технології та погляди на проблеми споживачів створюють нові можливості. Scrum-команди не просто перевіряють та пристосовують свій продукт на основі нової інформації; вони також перевіряють і пристосовують власні навички з огляду на свій досвід та проблеми, наявність яких вони передбачають. Навчання та розвиток є аспектом самоорганізації. Найкраще дозволити людям та команді володіти ним, за підтримки та керівництва.

Безперервне навчання може мати різні форми: самостійне навчання, користування онлайн-форумами, відвідування вебінарів

¹ <https://www.Scrum.org/resources/blog/why-focus-velocity-inhibits-agility>

та зустрічей, відвідування конференцій або виступ на них, читання книжок, статей і тематичних досліджень. Воно може містити онлайн-тренінг або поглиблене навчання в класі, можливо, передбачаючи отримання сертифікатів (тобто підтвердження навчання). Це також може мати на увазі тренерів або наставників, взаємодія з якими буде провадитися як офіційно, так і неформально. Або люди, які співпрацюють у процесі роботи, навчаються одне в одного та отримують інформацію й відгуки в режимі реального часу.

На практиці: парна робота за межами програмування

Можливо, ви чули про парне програмування, у якому двоє членів команди розробників співпрацюють в межах однієї діяльності з програмування. Вони працюють разом на одному робочому місці. Одна людина, яку вважають «ініціатором», пише код; інша, відома як «оглядач», переглядає кожен рядок коду щодо того, як його написано. Партнери часто міняються ролями. Упродовж цього процесу вони допомагають роботі одне одного, обговорюють проєктні рішення та надають негайний відгук про огляд коду.

Ви можете розширити цю ідею парної роботи й створити ще більше можливостей та переваг для команди. Наприклад, два члени команди розробників можуть працювати в парі, де один має сильніші навички програмування, а інший — сильніші навички тестування (або певні інші навички). Два члени групи розробників можуть також створити пару для тестування (наприклад, розробки тестів, написання тестів, виконання тестів).

Парна робота використовує дві пари очей, що надає різні погляди від працівників із різними навичками й досвідом та розширює спільні знання та вміння команди. Якісні дані свідчать про те, що парна робота зрештою поліпшує проєктне рішення, зменшує кількість помилок, збільшує рівень навчання, пришвидшує розв'язання проблем та підвищує відповідальність¹.

¹ Ця доповідь з конференції, виголошена професійною Scrum-тренеркою Падіпою Нараянасвами, висвітлює переваги парного тестування як засобу для розвитку навичок: <https://www.youtube.com/watch?v=DJBKWUjw01w>.

**На практиці:
поважайте людей, навчаючи кожного,
хто цього потребує**

Ми часто стикались із ситуаціями, коли організації направляли Scrum-майстрів на професійне навчання зі Scrum, а решту Scrum-команди — ні. Організація очікувала, що після відвідування занять Scrum-майстер зможе провести для всіх мінітренінги.

Така ситуація зазвичай має кілька негативних наслідків. По-перше, члени Scrum-команди можуть почати відчувати, що організація не поважає їх або виконувати ними роботу настільки, щоб інвестувати в їхнє навчання. По-друге, члени Scrum-команди часто погано розуміють Scrum-фреймворк та його інтенції, коли починають свої спринти, оскільки вони не отримали повного досвіду від професійного тренера. Це не налаштовує людей на успіх.

Використовуйте знання та досвід в організації

Хай що то буде — Scrum, фасилітація, Java, аналіз даних, алгоритм взаємодії з користувачем чи будь-яка інша сфера,— імовірно, існує широкий спектр знань та досвіду, які тільки й чекають, щоби ними скористалися в організації. Зазвичай, люди прагнуть поділитися знаннями й підтримати інших. До того ж викладання та наставництво допомагають їм відточувати свою майстерність, а також забезпечують краще відчуття життєвої мети.

**На практиці:
професійні спільноти уможливляють обмін
інформацією та вдосконалення**

Професійна спільнота — це група людей, які мають спільні інтереси. Вони діляться між собою досвідом та мають спільне бажання вдосконалюватися. Додатково до підтримки зростання учасників професійна спільнота може допомогти, сприяючи виявленню організаційних перешкод та керівництву організацією.

Професійна спільнота досить відкрита з погляду своєї форми та організації заходів і активностей. До різновидів її діяльності можуть належати презентації доповідей з-поза її меж, зустріч у форматі Lean Coffee¹, сесії з розв'язання проблем, семінари та соціальні заходи.

Що ж до структури, то професійні спільноти можуть формуватися на основі різноманітних факторів: ролі (наприклад, Scrum-майстер), дисципліни (наприклад, управління продуктами, тестування), навичок (наприклад, Java, архітектура) або будь-чого іншого на ваш смак. Нічого страшного, якщо є часткові збіги, оскільки люди можуть захотіти приєднатися до декількох професійних спільнот.

Наведемо декілька порад щодо створення ефективних професійних спільнот.

- Менеджери повинні підтримувати спільноти, надаючи простір для проведення зустрічей (фізичний, віртуальний або й те й те), час для працівників, щоби вони могли брати участь й робити власний внесок у свої спільноти, а також надаючи фінансування з метою підтримки діяльності спільноти.
- Дозвольте працівникам добровільно приєднуватися до спільнот; примусове членство зазвичай зводить нанівець переваги професійної спільноти.
- Дозвольте працівникам самоорганізовуватися. Члени професійної спільноти повинні визначитися, як вона працюватиме, як буде розвиватися та хто буде керувати її діяльністю.
- Оберіть вдале місце для проведення заходів, яке дозволить вам організовувати співпрацю. Це допомагає періодично зустрічатися особисто, але також забезпечує віртуальні способи зустрічей, коли фізичні неможливі.²

¹ Гнучкий варіант обміну знаннями та досвідом у формі зустрічі.— *Прим. перекл.*

² Перегляньте це коротке відео професійних Scrum-тренерів Стейсі Мартін і Тая Крокета про їхній досвід щодо запуску, фінансування професійної спільноти, підвищення рівня участі та багато іншого: <https://www.Scrum.org/resources/overview-communities-practice>.

Бути відповідальним Scrum-майстром

Scrum-майстер відповідає за те, щоб Scrum був зрозумілий та реалізований, у такий спосіб допомагаючи Scrum-командам та організації максимально скористатися з його переваг. Часто це інтерпретується як просто переконання в тому, що Scrum-команда дотримується «правил» Scrum-фреймворку та усуває організаційні перешкоди, які, можливо, заважають команді слідувати йому. Хоча це й правда, проте роль Scrum-майстра набагато більша за це.

«Посібник зі Scrum» описує Scrum-майстра як слугу-лідера. Завдяки такому стилю керівництва, успіх Scrum-майстра вимірюється зростанням та успіхом інших. Це відбувається завдяки здатності Scrum-майстра впливати на людей та команди, щоби вони брали на себе більшу відповідальність за свої дії та результати, надихаючи людей просуватися до вершин.

Хоча Scrum-майстер справді має владу над Scrum-процесом і йому може знадобитися посилення основних правил фреймворку в нових командах, його намір допомогти іншим поліпшитися викликає довіру та надихає членів команди виконувати все за власним вибором, а не через примушування. Ефективний Scrum-майстер виявляє низку якостей.

- *Подає приклад.* Scrum-майстер утілює цінності Scrum та дух співпраці в команді. Він відкритий до змін і покладається на емпіризм для боротьби з двозначністю та непередбачуваністю. Моделюючи це позитивне мислення та адаптивний підхід, він показує шлях іншим.
- *Дозволяє та надає можливість іншим.* Scrum-майстер не розв'язує проблеми людей, а натомість прагне відкрити можливості для вдосконалення за допомогою прозорої інформації та відкритих дискусій. Він знає, що не має ідеальних відповідей, і цінує колективний інтелект Scrum-команди.
- *Створює безпечне середовище та не боїться невдач.* Коли люди навчаються й виконують складну роботу, вони повинні почуватися в безпеці, вступаючи в конфлікт, кидаючи виклик одне одному та випробовуючи щось нове.
- *Спершу слухає і вчиться адекватно оцінювати ситуацію.* Scrum-майстер прагне сприяти досягненню консенсусу, щоби

люди почувалися почутими й були відкритими для інших поглядів.

- *Дуже піклується про людей, а також готовий кинути виклик, коли вони здатні на більше.* Scrum-майстер виходить із позитивних намірів і не засуджує людей. Він зустрічає людей там, де вони є, і допомагає їм здійснити наступний крок, надихаючи їх дотримуватися ще вищих стандартів.
- *Працює згідно з професійною етикою й зберігає спокій у стресовій ситуації.* Керівництво Scrum-майстра забезпечує послідовність і стабільність, за яку інші можуть триматися, коли відчувають стрес і переповнюються невпевненістю, яка панує навколо.
- *Виявляє низьку толерантність до організаційних перешкод.* Scrum-майстер готовий говорити правду владі, кидаючи виклик статус-кво й захищаючи команду.

Кому підзвітний Scrum-майстер? І Scrum-команді, і, зрештою, організації. Scrum-майстер служить Scrum-команді для максимізації переваг Scrum. Scrum-команда обслуговує організацію в плані постачання цінних продуктів. Однак існує багато парадоксів, які мусить подолати Scrum-майстер, щоби досягти успіху.

Вимірювання успіху Scrum-майстра

Успіх Scrum-майстра ґрунтується на успіху Scrum-команди. Однак Scrum-майстер не повинен бути недалекоглядним і вживати заходів, які підривають довгостроковий успіх Scrum-команди. Так само, як ви застосовуєте аналіз тенденції різних типів даних для розуміння цінності продукту, вам потрібні й кілька типів даних та вивчення тенденцій для визначення, наскільки добре працює Scrum-майстер. Наведемо декілька запитань для розгляду.

- Чи надійно Scrum-команда створює придатний для релізу приріст цінності продукту під час кожного спринту?
- Чи прийнятне значення приросту цінності? Чи поліпшується воно?
- Чи прийнятна якість приросту цінності? Чи поліпшується вона?

- Чи отримують члени команди задоволення від своєї роботи і чи постійно вони навчаються та зростають? Чи здається команда такою, що перебуває в стані застою й відчуває занепад морального духу?
- Чи прагне Scrum-команда постійно вдосконалювати кожен спринт?
- Чи чітко члени Scrum-команди розуміють Scrum-фреймворк і чи застосовують вони його послідовно та цілеспрямовано? Або ж вони просто «дотримуються правил», тому що так наказав їм Scrum-майстер?

Успіх Scrum-майстра можна виміряти на континуумі, як показано на рис. 6.4. На цьому континуумі тенденції (поліпшення/погіршення) важливіші за заходи, спрямовані на певний момент часу.

Вимірювання успіху Scrum-майстра

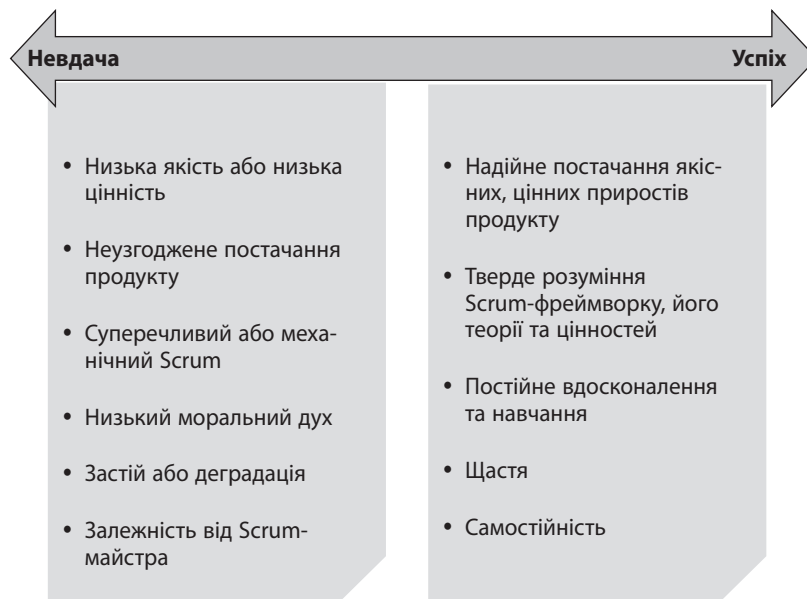


Рис. 6.4. Успіх демонструється завдяки зростанню та успіху інших у багатьох сферах

На практиці: ознаки того, що Scrum-майстер може пожертвувати довготривалим успіхом

На Scrum-майстра часто тиснуть, щоби «швидко показати поліпшення». Це складне середовище може призвести до певних видів поведінки, які фактично підривають успіх Scrum-команди.

- Scrum-майстер сприяє щоденній Scrum-наradі або оновлює Scrum-дошку, щоби допомогти «тримати все на потрібному шляху». Самоорганізована команда здатна керувати власним прогресом та оновлювати власний план. На ранньому етапі Scrum-майстру може знадобитися навчити їх техніці фасилітації та поставити запитання, щоби допомогти сфокусувати увагу та забезпечити консенсус. Однак із часом безпосередня участь Scrum-майстра має дедалі зменшуватися. Насправді, якщо щось піде не за планом, ця невдача може бути найпотужнішим навчальним досвідом для Scrum-команди.
- Scrum-майстер стає посередником, коли виникає конфлікт між членами команди. Людям легше служити, коли вони навчаться розв'язувати власні конфлікти безпосередньо одне з одним.
- Коли Scrum-майстер перебуває поза межами офісу, Scrum-команда пропускає події Scrum. Це може свідчити або про недостатнє розуміння мети (тобто емпіризм ще не сприйнято), або про відсутність навичок (наприклад, ніхто не відчуває себе комфортно під час проведення ретроспективи спринту). Кожна з цих форм поведінки демонструє залежність від Scrum-майстра. Насправді Scrum-майстри повинні прагнути позбавитися роботи. Хоча в дійсності цього ніколи не станеться, оскільки команди завжди будуть стикатися з новими викликами та матимуть можливості вдосконалюватися, Scrum-майстер повсякчас має шукати кращі способи, щоби Scrum-команда була більш самодостатньою.

Scrum-майстер підзвітний організації, але часто він може кинути їй виклик, коли вона заважає успіху Scrum-команди. Він мусить проявляти і сміливість, і співпереживання, щоби прямо кидати виклик людям, які обіймають керівні посади; у край важливо навчитися з повагою передавати важкі повідомлення. Кращому підходу до людей сприяють відкритість і цікавість. У всьому, що він робить, Scrum-майстер створює узагальнене розуміння та залучає людей до розробки спільних рішень.

Ефективні Scrum-майстри змінюють свій підхід на основі контексту

Scrum-майстри покладаються на спостереження, щоби зрозуміти, де люди перебувають сьогодні і в якій сфері їм найбільше потрібно зростати. Маючи досвід, вони інтуїтивно розуміють, що робити в тій чи іншій ситуації. Вони також постійно шукають нову інформацію та усвідомлюють, коли настав час змінити свій підхід (рис. 6.5).



Рис. 6.5. Scrum-майстер повинен вибрати найкращий підхід/підходи на основі контексту

Реакція Scrum-майстрів залежатиме від ситуації. Наведемо можливі варіанти.

- *Відстоювати Scrum.* Нова Scrum-команда може не повністю зрозуміти та прийняти емпіричну природу Scrum-фреймворку. Учасники команди відчуватимуть, що можуть пропустити подію, вийти за часові межі або втратити з поля зору відповідальність своєї ролі — і все це без жодних наслідків. Таке може трапитися тому, що вони досі намагаються створити потужний командний фундамент через тиск з боку організації, щоби зрізати кути, або тому, що стали самовдоволеними й почали ставитися до роботи безвідповідально. Підтримка Scrum означає повернення Scrum-команди до «чому» поза межами фреймворку, посилюючи переваги, які забезпечують вірність принципам Scrum.
- *Навчати.* Коли Scrum-команді потрібно поліпшити своє розуміння основ, головних принципів і додаткових практик Scrum-фреймворку, Scrum-майстер мусить допомогти членам команди поліпшити їхнє розуміння та застосувати його для ефективного постачання цінності. Найкраще вони роблять це, створюючи простір, у якому люди навчаються на власному досвіді завдяки керованим відкриттям¹.
- *Указувати напрям.* У контексті Scrum указівка напряму означає створення та розвиток емпіричної культури перевірки та адаптації з метою прийняття кращих рішень шляхом постачання цінності, збору відгуків та зміни курсу на їхній основі.
- *Надавати настанови.* Коли Scrum-майстри надають настанови, вони втілюють віру в те, що людина чи команда здатні знаходити власні відповіді, і допомагають їм знайти ці відповіді. Вони зосереджуються на тому, щоби допомогти членам команди зрозуміти власну ситуацію й те, що працівники хочуть робити далі. Зазвичай для цього ставлять запитання, щоби допомогти членам команди глибше зрозуміти, визначити дії, які вони будуть виконувати, та допомогти їм узяти

¹ Tastycupcakes.org — чудовий краудсорсинговий репозиторій про навчальну діяльність для загальних agile-концепцій.

на себе відповідальність за них¹. Коучинг допомагає людям розвивати їхню здатність демонструвати відповідальність, підзвітність, самоорганізацію та відчуття комфорту в умовах змін і непередбачуваності².

**На практиці:
ефективні Scrum-майстри (та лідери) здійснюють
коучинг для окремих людей заради досягнення
кращих командних результатів**

З власного досвіду можемо сказати, що корисно виділити регулярний час на проведення настановчих бесід з особами, які входять до Scrum-команди (і навіть із тими, хто не належать до неї). Такі регулярні бесіди допомагають людям навчитися краще розуміти себе. Цей процес сприяє тому, що люди можуть зрозуміти свою мету та прояснити власні цінності та погляди. Це також допомагає людям зрозуміти власні вподобання, мотивацію та поведінку, щоби мати змогу краще керувати своєю взаємодією з іншими. Наведемо кілька порад, щоб отримати максимальну віддачу від регулярних розмов із наставником.

- *Спробуйте вийти зі звичного робочого середовища.* Можливо, прогуляйтеся навколо будівлі, відвідайте місцеву кав'ярню або посидьте в сусідньому парку. Проста зміна декорацій, фізичні рухи чи свіже повітря можуть забезпечити збільшення кута зору та креативності.
- *Не завжди потрібен порядок денний.* Хоча іноді ви можете звернутися до конкретних спостережень чи ситуацій, нехай той, з ким ви проводите коучинг, веде бесіду. Коучинг стосується

¹ Ми закликаємо Scrum-майстрів та інших, хто хотів би вдосконалити свої коучингові навички, розглянути можливість навчання в організації, акредитованій Міжнародною коучинговою федерацією (ICF). До організацій, з якими ми маємо особистий досвід навчання, належать Інститут підготовки коучів (CTI) та Інститут Agile-коучингу (ACI).

² Приклад того, як поєднати коучинг із використанням цінностей Scrum, можна знайти в цій публікації в блозі: <https://www.agilesocks.com/4-ways-to-coach-with-the-Scrum-values/>.

іншої людини, а не вас. Просто розпочніть розмову із запитання «Що у вас на думці?»¹.

- *Святкуйте успіхи та зростання.* Хоча ми віримо в постійне вдосконалення, його можна досягти, лише святкуючи успіхи та зростання на цьому шляху.
- *Знайдіть час для обробки спостережень та розкриття знань.* Якщо люди завжди «в дії», вони можуть не отримати стільки інформації про себе та своє оточення.

- *Здійснюйте фасилітацію.* Scrum-майстри можуть сприяти проведенню Scrum-подій, додатковим робочим сесіям або навіть імпровізований співпраці впродовж усього спринту. Ефективне сприяння вимагає усвідомлення (розуміння ситуації), конструктивних конфліктів і збереження уваги до спільних цілей та результатів. Фасилітація забезпечує більшу структуру для самоорганізації команд і створює середовище, у якому члени команди можуть вступати в продуктивний конфлікт, досліджувати різні погляди, дотримуватися рішень команди та використовувати свої творчі здібності.
- *Уживайте заходи.* У деяких випадках Scrum-майстру потрібно буде вжити заходи, іноді рішуче й негайно. Діючи, Scrum-майстер чинить як захисник команди, її емпіричного процесу та інтересів організації в цілому. Уживання заходів означає, що Scrum-майстер повинен досягти тонкого балансу, керуючись питаннями безпеки. Якщо Scrum-майстер втручається занадто або занадто часто, це шкодить здатності Scrum-команди до самоорганізації, але там, де безпека або цілісність команди перебувають під загрозою, Scrum-майстер може не мати альтернативи.
- *Активно нічого не робіть.* Якщо робити щось для людей, коли вони можуть зробити це самотужки, це позбавить їх сил. Суть емпіризму та культури навчання полягає в експериментуванні та навчанні на практиці. Щоби члени команди навчалися, їм потрібно діяти самостійно. Це свідоме рішення, і воно завжди служить навчанню та зростанню інших.

¹ «Що у вас на думці?» — одне із семи важливих запитань, які ставляться в книжці: Michael Bungay Stanier, *The Coaching Habit* (Box of Crayons Press, 2016). Це чудове видання, що містить кілька простих основ для проведення коучингу.

«Активно» означає, що ви продовжуєте активно спостерігати, доки команда навчається та досліджує. Потім, з огляду на те, що відбувається, ви можете продовжити цей підхід або вибрати щось інше.

На практиці: коли втручання Scrum-майстра є доречним

«Посібник зі Scrum» подає простий приклад належного втручання Scrum-майстра: усунення перешкод, яке зазвичай відбувається за допомогою втручання Scrum-команди щодо бажаного результату, що його потрібно досягти. У цій ситуації Scrum-майстер буде взаємодіяти з іншими членами організації, щоби допомогти реалізувати рішення, яке часто вимагає навчання, фасилітації та наставництва.

Коли безпека будь-якої людини, команди або структури опиняється під загрозою, Scrum-майстер повинен діяти. Крайній випадок — це ситуація знущання чи переслідування. Така поведінка не тільки не узгоджується з цінностями Scrum та основними принципами цього фреймворку, але і є незаконною. Коли щось подібне трапляється, Scrum-майстер мусить негайно діяти, щоби захистити безпеку команди.

Менш екстремальним прикладом є втручання в активне управління конфліктом, який загострюється в нездоровий спосіб. Іноді розумно дозволити команді вчитися, досліджуючи та роблячи щось, однак в інших ситуаціях втручання допоможе команді внести суттєві зміни. Це особливо характерно для команд-початківців, оскільки вони не впевнені, як працювати по-новому разом. Активніше керуючи командою, Scrum-майстер може допомогти поліпшити темп навчання.

Підсумки

Scrum-команди потребують допомоги та підтримки для вдосконалення. У своїх ретроспективах спринту вони, найвірогідніше, визначать багато сфер, у яких їм потрібно вдосконалитися, але їм не слід відсувати на другий план потужність та оперативність щоденної Scrum-наради як збільшувального скла для вдосконалення. Швидке усунення перешкод за першого ж зіткнення з ними Scrum-команд є чи не найкращим, що ви можете зробити, аби їм допомогти.

Прозорість допомагає кожному зрозуміти, що стримує команду, а відстеження постійних проблем може дати змогу зробити їх більш помітними. Візуалізація різних форм зайвих витрат та розповсюдження інформації про них серед людей, які можуть допомогти це усунути, — гарна звичка, яку слід прищепити.

Scrum-командам також потрібно інвестувати у своє власне вдосконалення й отримувати можливість для здійснення цього від тих, хто не входить до команди. Під час планування спринтів їм потрібно брати до уваги особисті та командні інвестиції в розвиток, і вони мусять зробити свої потреби прозорими для людей з-за меж команди, яким, можливо, необхідно буде виділяти гроші, час або допомогу для розвитку можливостей інших.

Scrum-майстри відіграють важливу роль як у своїх командах, так і поза ними. Вони допомагають своїм командам навчатися та розвивати власні навички, сприймати емпіризм та гнучкий спосіб мислення, закликають їх до вдосконалення та допомоги у відточуванні власних умінь. Щоби бути ефективним Scrum-майстром, потрібно мати широкий спектр навичок, а також мудрість і досвід, щоби знати, коли і як застосовувати різні техніки. Крім безпосередньої допомоги командам, Scrum-майстри сприяють їм, впливаючи на організацію в цілому, щоби допомогти командам стати більш ефективними з часом.

Заклик до дії

Розгляньте зі своєю командою наведені нижче запитання.

- Скільки значущих роздумів трапляється в ретроспективах спринту?

- Як ретроспективи спринту можуть стати більш цікавими, креативними та колективними?
- Які наявні дані можуть забезпечити інформаційну відкритість щодо частоти та впливу перешкод?
- Які дії було вжито для усунення перешкод за останні кілька спринтів?
- Які основні перешкоди допускаються?
- У яких сферах члени команди найбільше потребують зростання й хочуть цього?
- У яких сферах Scrum-команда вдосконалюється і як Scrum-майстер вплинув на це зростання?
- Чи є щось, що необхідно припинити робити Scrum-майстру?
- Що потрібно від керівників організації для підтримки вдосконалення команди?
- Які проблеми зараз найбільше дошкуляють? Визначте один-два експерименти, які допоможуть у поліпшенні. Для кожного експерименту обов'язково визначайте бажані впливи та спосіб їх вимірювання.

Розділ 7

Ефективне залучення організації до вдосконалення

Як ми з'ясували в попередньому розділі, Scrum-майстер, менеджери та навіть окремі члени Scrum-команди допомагають одне одному та команді в цілому вдосконалюватися. Під час процесу їм нерідко доводиться використовувати структуру та культуру організації для створення необхідних умов, щоб допомогти командам розвивати свої можливості та краще зосереджуватися на частішому постачанні цінності. Структура та культура організації мають значний вплив на ідентичність команди, її діяльність, а також на те, як її члени розуміють і вимірюють цінність продукту. Завдання для всіх полягає в тому, щоб зробити вплив організації позитивним, а не таким, що заважає зростанню Scrum-команд.

Організаціям необхідно розвиватися, аби досягти успіху

Кожна організація потребує структури та обмежень стосовно своєї роботи, щоб мати змогу успішно вести власний бізнес. Організація визначає, які товари та послуги й для яких клієнтів вона пропонує, а також бізнес-моделі, що роблять можливим розвиток фінансування та підтримку діяльності. Ця структура також визначає, як співробітники, партнери та постачальники послуг працюють разом над постачанням продуктів і послуг. Структура часто виражається в процесах та політиці, які організація встановлює та застосовує.

Кожна організація також має культуру, набір звичок, які об'єднують людей і встановлюють неписані норми поведінки. Ця культура еволюціонує із суми всієї людської поведінки в організації й перебуває під впливом організаційної структури та процесів, зокрема ролей, цілей і стимулів.

Для допомоги організації в отриманні максимуму від Scrum, вам зазвичай потрібно розвивати її культуру, процеси та, можливо, навіть структуру. Щоби мінімізувати шкоду для організації та Scrum-команди, потрібно мати намір це зробити; організації та культурі притаманні багато механізмів — як явних, так і неявних — для запобігання випадковим змінам. Упроваджуючи Scrum, ви неминуче вносите зміни, і зворотна реакція від некерованих змін є частою причиною того, чому Scrum-команди борються, часто навіть не розуміючи, що відбувається.

Scrum-команди іноді можуть працювати певний час, не розв'язуючи структурні та культурні проблеми організації, але зрештою вони почнуть стикатися з перешкодами, які не підвладні їхньому контролю. Коли ці перешкоди неможливо буде усунути, Scrum-команда вийде на плато і їй буде складно підтримувати прогрес.

У цьому розділі ми коротко досліджуємо найпоширеніші проблеми, з якими стикаються Scrum-команди на організаційному рівні. Ми також описуємо, як змінити свій спосіб мислення, перейшовши від «захисту команди від організації» до «залучення організації для поліпшення здатності Scrum-команди надавати цінність».

Розвиток людей і команд

Організації вкладають багато грошей у найм та розвиток людей, проте своїми діями вони можуть підірвати власну здатність утримувати та залучати тих самих людей упродовж їхньої кар'єри. Такі не залучені співробітники коштують американським компаніям приблизно пів *трильйона* доларів на рік утраченої цінності, не кажучи вже про вартість просто «не зайнятих» працівників. Хоча для працівника наразі нерозумно сподіватися на роботу в тій самій компанії впродовж усієї своєї кар'єри, але цілеспрямована зайнятість працівників приносить значні вигоди¹.

¹ <https://www.inc.com/ariana-ayu/the-enormous-cost-of-unhappy-employees.html>

Вплив оглядів продуктивності та компенсації

Традиційні організації зосереджуються насамперед на індивідуальних результатах. Дивлячись на речі більш реалістично, скажемо, що в сучасному світі, у якому ми живемо й працюємо, майже все, що ми робимо, вимагає для досягнення бажаних результатів співпраці між багатьма людьми. Винагородження людей за роботу, досягнуту командою, — це недалекоглядна та застаріла практика, яка надсилає неправильне повідомлення про те, що є важливим. Часто колективна робота та результати команди навіть не розглядаються або є лише другорядною частиною оцінки. Крім того, багато систем компенсації можуть здаватися довільними та базуватися на титулах і статусі, а не на зусиллях, зростанні та досягнутих результатах. І чи мають сенс щорічні огляди продуктивності?¹

У сучасному швидкозмінному світі високопродуктивні команди є рушіями створення цінності організації. Організації, які хочуть створювати та розвивати високопродуктивні команди, повинні винагороджувати колективну роботу, а не індивідуальну продуктивність. Вони можуть зробити це різними способами, зокрема такими:

- винагородження команд бонусами за результатами їхньої роботи;
- дозвіл командам вирішувати, як розподілити підвищення та бонуси між членами команди;
- оцінка вагомості внеску в роботу команди в процесі огляду результатів діяльності окремої людини;
- надання людям частих, значущих та дієвих відгуків;
- збір відгуків про результати діяльності окремої особи від широкого кола людей — членів команди, клієнтів та менеджерів (іноді їх називають 360-градусними відгуками);²
- розгляд внутрішніх винагород, які підвищують почуття автономії, майстерності та цілі людини або команди³.

¹ <https://www.forbes.com/sites/lizryan/2018/01/14/performance-reviews-are-pointless-and-insulting-so-why-do-they-still-exist/#1cf4ca8972d1>; <https://getlighthouse.com/blog/get-rid-of-the-performancereview/>

² <https://www.thebalancecareers.com/360-degree-feedback-information-1917537>

³ *Drive*, Den Pink, (Riverhead Books, 2011).

Індивідуальні кар'єрні шляхи

Під час роботи з організаціями нас часто запитують про визначення кар'єрних шляхів. Роль Scrum-майстра нова для багатьох організацій і не схожа на жодну іншу, котру вони виконують, тому вони не впевнені, як визначити свій кар'єрний шлях. Те саме стосується й власника продукту. Ба більше, коли менеджери починають усвідомлювати самоорганізовану, кросфункціональну природу команд розробників, вони можуть зрозуміти, що те, як було визначено кар'єрний шлях, може спричинити конфлікти.

Традиційні організації часто поділяються на групи за навичками, побудованими навколо певної галузі спеціалізованих знань, кожна з яких має власний кар'єрний шлях. Ця структура мала сенс, коли робота справді була спеціалізованою і здійснювалась у світі, де людям не доводилося глибоко співпрацювати для її виконання. Ми вже показали силу самоорганізованих, кросфункціональних команд під час доставки складних продуктів у світі, де переважає невизначеність. Ізольовані організації з чітко визначеними ролями не можуть йти в ногу із сучасними потребами. Найкраще рішення — дозволити людям, що перебувають найближче до продукту, та клієнтам вирішити, які навички їм потрібні.

Кар'єра — і так було завжди — то історія, яку ви розповідаєте собі про те, що робите зараз і до чого це вас приведе. Не потрібно багато саморефлексії, щоби усвідомити: більшість із того, що, на вашу думку, мало відбутися, так і не трапилось, але замість цього сталися всілякі чудові речі, про які ви не могли навіть мріяти. Інша проблема поняття кар'єри полягає в тому, що інновації постійно створюють нові види роботи та вбивають багато інших видів праці. Робота, яку ви будете виконувати за десять років, або ще не існує, або буде настільки перетворена, що не матиме жодної схожості з тією, що існує сьогодні.

Гнучкий світ — це здебільшого світ посткар'єрний. У складних умовах, що характеризуються постійними змінами, навички та знання, необхідні для надання цінності споживачам, будуть постійно зазнавати метаморфоз. Замість планування кар'єри організаціям та особам потрібно зосередитися на формуванні гнучких навичок розв'язання проблем та критичного мислення. Команди, що самоорганізуються, є чудовим засобом для такого типу зростання, оскільки вони дозволяють будь-якому своєму

члену добровільно опановувати нові навички, коли команда їх потребує.

Водночас члени команди потребують певних указівок, які допоможуть їм знаходити власні рішення щодо професійного розвитку. Їхні організації можуть допомогти цьому в такий спосіб:

- зробити прозорим перелік навичок, які потрібні в різних частинах організації, та вказати місця, де існують можливості для їх розвитку;
- допомогти зрозуміти, де може знадобитися розширення навичок, а також поглиблення навичок;
- поєднати працівників із наставниками, які можуть забезпечити кар'єрну перспективу та допомогти створити професійну мережу;
- допомагати в зростанні впливу людей у організації поза межами команди;
- підтримувати професійні спільноти для обміну й розвитку навичок та досвіду;
- допомагати працівникам розвивати лідерські навички додатково до технічних умінь або навичок розв'язання спеціалізованих задач.

Стратегії пошуку ресурсів і вплив на команди

Організації використовують стратегії пошуку ресурсів для досягнення різноманітних цілей: зниження витрат, збільшення гнучкості під час найму, отримання швидкого доступу до дефіцитних навичок тощо. Але ті самі стратегії можуть вплинути на здатність команд об'єднуватися, розвивати самостійність та ідентичність, що позначатиметься на їхній здатності поставляти цінність. Члени команди з різних організацій можуть мати відмінну мотивацію та різний рівень прихильності до цілей команди. Тимчасові працівники можуть не відчувати того самого рівня поваги чи довіри, як інші члени команди. Члени команди, які працюють віддалено, можуть не відчувати однакового ступеня прозорості чи залученості. Стратегії пошуку, що сприймають людей як взаємозамінний ресурс, завдають серйозної шкоди процесу створення відповідальних, прозорих, самоорганізованих команд, які необхідні для успіху.

Стратегії пошуку, які орієнтовані на зниження оплати праці без урахування ефективності команди, є недалекоглядними. Довготривале очікування, спричинене надмірними передачами, утратою ефективності через надмірне переключення контексту, та перероблення, пов'язане з неефективним спілкуванням, можуть легко перекрити невелику економію, отриману внаслідок різниці в заробітній платні всередині компанії. Спершу зосередьтеся на створенні високоефективних команд, а потім на усуненні перешкод та інших джерел відходів, інакше ви можете виявити, що ваша стратегія пошуку призводить, як нарікав один менеджер, що з ним ми працювали, «до провалу зі знижкою 70 %».

На практиці: невід'ємний конфлікт у виробничій підтримці аутсорсингу

Багато організацій хочуть передати свою підтримку на аутсорсинг, щоби заощадити гроші, але він часто базується на хибному припущенні, що така робота вимагає лише навичок програмування та виправлення помилок. Підтримка аутсорсингу часто базується на світогляді проекту, який передбачає, що зміни в застосунках є рідкісними й просто пов'язані з виправленням дефектів.

Коли застосунок є частиною потоку цінності, що постачається для клієнтів, підтримка аутсорсингу може завдати шкоди кількома способами. Припущення, що програми не повинні часто змінюватися,— погане. Насправді застосунки та інші процеси в потоці цінності можуть дуже часто змінюватись у відповідь на мінливі ринки або потреби споживачів. Процеси підтримки, які навмисно ускладнюють ці зміни, перешкоджають підвищенню рівня задоволеності та утриманню споживачів.

Так само аутсорсинг позбавляє Scrum-команди цінної інформації, необхідної їм для розуміння того, як поліпшити програму — їхній продукт. Вони ніколи не отримують досвіду з підтримки системи, тому їм не вдається інвестувати в можливості продукту, які могли би поліпшити якість обслуговування клієнтів, зробивши застосунок більш надійним, масштабованим або безпечним.

Підтримка аутсорсингу змушує розробників та спеціалістів з експлуатації конфліктувати, указувати одне на одного пальцями та ворогувати. Зрештою це позбавляє допоміжний персонал цінного досвіду, який можна отримати, лише працюючи над продуктом. Реальність така, що підтримка/операції є складним завданням, і розв'язання питань підтримки окремо від розробки створює багато проблем, а усуває мало.

Якщо аутсорсинг є частиною вашої стратегії, розгляньте наведені нижче питання.

- Це транзакційні відносини чи довгострокове партнерство?
- Як тимчасово залучені працівники проходять навчання та підвищення кваліфікації? І чи буде це узгоджуватися з результатами постійних членів команди, з якими вони працюють?
- Чи має партнер agile-мислення та прихильність до емпіризму?
- Як оцінюється партнерство? Що таке процес перевірки та адаптації для вдосконалення?
- Якими є ризики аутсорсингу для вашого бізнесу? Які переваги надає аутсорсинг вашому бізнесу?

Аутсорсинг має свої переваги в разі правильного використання, але пам'ятайте і про витрати (прямі й непрямі). Поміркуйте, що може зробити організація для зменшення ризиків аутсорсингу. Спостерігайте за результатами та поведінкою.

Розподілені команди

Подібно до аутсорсингу, розподіл членів команди може зменшити згуртованість та ефективність її роботи. Люди, які працюють у різних часових поясах, матимуть менше часу на співпрацю — залежно від того, наскільки збігаються їхні робочі години. Члени команди, які працюють нарізно, виявлять, що їхня співпраця менш ефективна порівняно з тими, хто працює пліч-о-пліч. Їм буде важче працювати разом у команді. Змусити працювати розподілені команди не є неможливим, але це, безумовно, набагато складніше.

Члени команд — представники різних культур — також іноді відчують труднощі в комунікації. Деякі культури більш відкриті для висловлення розбіжності в думках, тоді як інші виявляють більшу повагу в умовах різного соціального статусу. Ці та інші фактори ускладнюють досягнення прозорості. Довіра та спільна робота зазвичай допомагають членам команди досягти більш високих рівнів взаєморозуміння, що робить прозорість та співпрацю ефективнішими, але досягти спільного мислення важче, коли члени команди віддалені. У будь-якому разі вам доведеться працювати більш старанно, щоби подолати бар'єри, створені розподіленими командами. Для цього слід зробити низку кроків¹.

- Допоможіть команді самоорганізуватися, а не намагайтеся розв'язати проблеми за неї.
- Інвестуйте в зростання команди за допомогою сесій співпраці на місцях (принаймні раз на рік, але бажано щоквартально). Задійте заходи, спрямовані на особисте знайомство працівників, установлення чітких робочих домовленостей, узгодження бачення продукту та розуміння споживачів, а також на досягнення спільних цілей.
- Інвестуйте в засоби спілкування та співпраці (наприклад, відеозв'язок, інтерактивні дошки).

Почуття зручності в стані прозорості

У цій книжці ми обговорювали важливість прозорості, коли ми маємо справу з комплексністю, неоднозначністю та непередбачуваністю, а також те, як це проявляється в нашій повсякденній взаємодії. Оскільки Scrum-команди починають використовувати прозорість та емпіричний процес, їм часто доводиться стикатись із суворою реальністю, що організації в цілому не обов'язково почувуються зручно в стані прозорості. Прозорість звучить як гарна ідея, доки відгуки клієнтів не покажуть, що елемент беклогу продукту — важливий, на думку зацікавленої сторони,— зовсім не те,

¹ Про інші техніки див. <https://techbeacon.com/app-dev-testing/distributed-agile-teams-8-hacks-make-them-work>.

що цікавить клієнтів, або ж коли виявиться, що найстарший розробник у команді повинен поліпшити якість свого коду.

Прозорість вимагає мужності, оскільки вона може кинути виклик загальноприйнятим ієрархіям або догмам. Вона вимагає справжнього керівництва, оскільки команда повинна мати безпеку та психологічний простір, щоб іноді висловлювати правдиві, але непопулярні думки. Зокрема керівники повинні гарантувати, що решта організації не буде відкрито чи опосередковано «карати» вісника. Як казав один із наших колишніх менеджерів, «факти доброзичливі». Тобто завжди краще мати більше інформації для прийняття кращих рішень. Маючи кращу інформацію, команди можуть ефективніше досліджувати та адаптуватися.

Багато організацій та їхні керівники позитивно налаштовані щодо людей із позицією «ми можемо це зробити». Команда, яка вірить, що може зробити все, має велику цінність. Водночас між упевненістю та оманю дуже тонка грань. Прозорість є протитрутою від нереальних очікувань.

Психологічна безпека має важливе значення для створення ефективних команд¹. Так само, як люди в команді повинні відчувати довіру одне до одного, організація повинна демонструвати довіру до команди. Довіра означає готовність:

- проводити відкриті та чесні обговорення;
- ділитися та досліджувати різні думки;
- вірити, що кожен робить усе можливе з інформацією, відомою йому на той момент.

Культура відповідальності за результат, а не звинувачення

Коли лідери кажуть: «Мені потрібні більш відповідальні люди», вони іноді справді мають на увазі, що хочуть, аби був хтось, кого можна звинуватити, якщо щось піде не так. Це створює культуру страху, яка заважає людям пробувати нове, збирати відгуки й адаптуватися. Agile-підхід не може вижити в організації з культурою,

¹ <https://hbr.org/2017/08/high-performing-teams-need-psychological-safety-heres-how-to-create-it>.

заснованою на страху та звинуваченнях. Відповідальність за результат відрізняється від звинувачення. Так, це означає приймати самостійні рішення та їхні результати, але кінцева мета — не робити когось винним. Натомість відповідальність означає зосередженість на цілі та належні наміри. Ідеться про прозорість ухвалення рішень, що дозволяє навчатися та адаптуватися; про прийняття складності й непередбачуваності та про створення коротших циклів зворотного зв'язку для зменшення ризику й пришвидшення навчання.

Організація з культурою відповідальності за результат дозволяє людям ухвалювати самостійні рішення (тобто ніхто не може перевершити рішення власника продукту про те, що створювати), вимагаючи, щоби вони були емпіричними у своєму підході, підтримуючи у доступі до потрібних людей та інформації та приймаючи той факт, що на цьому шляху траплятимуться помилки та дивовижні успіхи.

Коли організації стикаються з прозорістю та переходять від культури звинувачення до культури відповідальності за результат, це вимагає від лідерів відмови від контролю — точніше, ілюзії контролю.

Відмова від (ілюзії) контролю

Менеджери традиційних організацій живуть у культивованій ілюзії, що вони «головні». Та той, хто обіймав керівну посаду, розуміє, наскільки насправді мало контролю в менеджерів, особливо в ситуаціях, коли їхні організації постачають складну продукцію. Менеджер просто не може мати всіх знань та досвіду, необхідних для цього. Насправді потрібна кросфункціональна команда, яка здатна на ухвалення самостійних рішень та наділена повноваженнями робити це з відповідальністю за рішення.

Поки менеджери не допоможуть командам самоорганізуватися та взяти на себе відповідальність, вони ніколи не розвинути навичок довіри, прозорості, відданості, зосередженості, відкритості, сміливості й поваги, необхідних для досягнення чудових результатів¹. Якщо менеджер не позбавиться ілюзії контролю, щоби стати лідером, який служить, його команди поринуть в неефективність та розгубленість.

¹ <https://guntherverheyen.com/2013/05/03/theres-value-in-the-Scrum-values/>

Справжня сила залізного трикутника

Згадайте залізний трикутник із Розділу 5 «Поліпшення планування». Як ми зазначали в тому розділі, залізний трикутник часто змушує людей ігнорувати результати як критерій успіху. Дещо важливе щодо цього трикутника полягає в тому, що всі три обмеження — витрати, графік та обсяг — неможливо усунути, оскільки ми маємо справу зі складною роботою.

Десь дорогою цей момент утрачається в багатьох організаціях, які вимагають, щоб обсяг, графік та витрати були оцінені та гарантовані. Справжня сила залізного трикутника з'являється тоді, коли організації усвідомлюють, що не можуть виправити всі ці змінні, і натомість починають вести правильні розмови — відкрито й чесно — про обмеження, які впливають на рішення використати (або продовжувати використовувати) надану можливість.

- Чи можете ви поставити щось достатньо цінне досить швидко та за ціну, яка виправдовує інвестиції?
- Чи знайомі ви з роботою, яку потрібно буде виконати? Як ви берете до уваги цю невизначеність у рішенні скористатися можливістю?
- Як часто ви можете підтверджувати, що ваші відповіді на попередні запитання досі дійсні?

Давній вислів говорить, що ви можете зафіксувати лише дві сторони трикутника. Scrum дозволяє досить легко дотримуватися «правил» залізного трикутника. Якщо ви думаєте про спринт, це фіксована кількість часу та фіксовані витрати (припускаючи, що люди в Scrum-команді та їхні можливості не змінюються). Отже, стає досить легко визначити час і вартість за допомогою Scrum, і тоді ви дозволяєте обсягу бути гнучким (див. рис. 7.1 на с. 184).

Хоча вас може турбувати гнучкість сфери застосування, реальність така, що ми не можемо передбачити, який обсяг призведе до бажаного результату. Пам'ятайте, що Scrum дає нам власника продукту, який відповідає за оптимізацію цінності. Власник продукту забезпечує узгодження з баченням продукту, замовляє беклог продукту та визначає й перевіряє цінність. Власник продукту гарантує роботу Scrum-команди над наступною правильною річчю.



Рис. 7.1. Залізний трикутник із продуктовим мисленням

Звичайно, власник продукту та зацікавлені сторони можуть насправді не знати, що саме потрібно для досягнення бажаних результатів. У них можуть бути ідеї та гіпотези, але складний характер розробки продукту робить важчим отримання точної інформації, у якій можна бути абсолютно впевненим. Це бере початок з моменту перевірки результатів. Гнучкість в обсягах є силою залізного трикутника. І якщо ви часто здійснюєте релізи заради підтвердження своїх припущень та отримання значущих відгуків від ринку або користувачів, то можете пристосувати процес роботи для більш точного узгодження з бажаними результатами.

Ви можете поставити запитання: «А якщо я хочу виправити дві різні сторони залізного трикутника?» Ми вважаємо, що традиційний погляд на управління проектами створює помилку, зокрема через складність. Часто бажаним є встановлення часу та обсягу, але якщо ви викинете більше грошей на складну проблему без збільшення графіка, то зазвичай отримаєте протилежний ефект. Є варіанти: понаднормова робота або збільшення кількості людей у команді. Перший випалює людей, створюючи проблеми з якістю, і врешті-решт знижує моральний дух. Другий уповільнює роботу, оскільки тепер вам доводиться мати справу з переформуванням

ідентичності команди та розв'язанням завдань командного процесу з новими учасниками.

Логічний спосіб використання залізного трикутника — зекономити час і гроші та дозволити обсягу бути гнучким. Як альтернативу ви можете зафіксувати обсяг і дати можливість часу та грошам бути гнучкими, але лише за умови вашої абсолютної впевненості в тому, що все необхідне, на вашу думку, справді потрібне вашим клієнтам.

Фінансові ініціативи

Багато ініціатив починаються з поставленої мети або бажаного результату. Іноді організація висуває гіпотезу про те, який обсяг може допомогти в досягненні бажаного результату. Можливо, ви також захочете зрозуміти приблизну вартість досягнення такого обсягу для отримання бажаних результатів. Це коли ви робите оцінку на основі обсягу. Ви можете застосувати відносні методи оцінки, описані в Розділі 5, але, найвірогідніше, зробіть це на більш високому рівні.

Оцінка на основі обсягу

Члени Scrum-команди, які спільно працюють над продуктом, можуть зробити відносну оцінку та порівняти цю роботу з попередніми ініціативами, що були ними реалізовані. І якщо вони не мають історії спільної праці, їм потрібно буде провести достатньо аналізу, щоб отримати уявлення про роботу, знаючи, що вони можуть здійснювати перегляд оцінки, коли будуть дізнаватися більше під час кожного спринту.

Під час здійснення загальної оцінки ми рекомендуємо робити її простою. Власник продукту повідомляє, що йому відомо про ініціативу, зокрема про бажані результати та конкретні особливості чи функції, котрі, як очікується, дозволять досягти необхідних результатів. Команда розробників відповідно робить оцінки на основі відомої на сьогодні інформації. Найменша одиниця, яку можна передбачити для Scrum-команди, — це спринт. Тому має сенс створювати бюджети в цих підрозділах. Можливо, команда розробників

оцінить кількість спринтів. Або, вірогідно, вони оцінять роботу як малу, середню чи велику й закладуть розуміння того, який рівень зусиль і складності підпадає під ці категорії. Тоді команда матиме приблизний прогноз, скільки спринтів знадобиться для завершення роботи, а кількість спринтів дозволить визначити вартість. Зрештою, усе зводиться до кількості спринтів. На рис. 7.2 наведено приклад оцінки Scrum-командою ініціатив високого рівня.

Можливість	Оцінка високого рівня (спринти)	Бюджет команди (50 тис. дол./спринт)
A	2	\$100 000
B	4	\$200 000
C	0.5	\$25 000
D	3	\$150 000
E	7	\$350 000
F	1	\$50 000
G	3	\$150 000

Рис. 7.2. Приклад простого високорівневого бюджетування

Звичайно, у вас можуть бути додаткові витрати не лише на людей зі Scrum-команди, тому вам потрібно буде оцінити та додати їх до свого бюджету. Не забувайте регулярно перевіряти та пристосовуватись, якщо очікується, що якісь із цих не пов'язаних із роботою витрат зміняться впродовж виконання роботи.

Навіть із бюджетуванням, заснованим на обсязі, ви досі хочете, щоби власне обсяг залишався доволі гнучким, аби команда могла вивчати та пристосовувати беклог продукту, наближаючи його до бажаних результатів. Це дозволяє корегувати конкретні пункти в релізі, водночас фіксуючи обсяг роботи, який потрібно виконати.

Ітеративне та інкрементальне бюджетування

Добре виконаний Scrum дозволяє бізнесу ітеративно (покроково) та інкрементально (поступово) формувати бюджет та фінансувати Scrum-команду для низки спринтів, довіряючи власнику продукту

ту ухвалення відповідних рішень щодо порядку беклогу продукту з метою оптимізації цінності. «Виконані» прирости цінності, а також нові знання про роботу, показники цінності та відгуки клієнтів стають прозорими для зацікавлених сторін під час кожного спринту. (Огляд спринту — чудова можливість обговорити отримані уроки.) Власник продукту змінює беклог продукту на основі емпіричних даних та співпраці із зацікавленими сторонами й рештою Scrum-команди. Майте на увазі: що довше ваш спринт, то більший інвестиційний ризик ви несете.

У будь-який момент власник продукту або організація можуть вирішити припинити фінансування додаткових спринтів. Таке може статися, коли рентабельність інвестицій перестає бути достатньо високою для виправдання вдосконалення продукту. Це може вказувати на те, що організація має змогу більше інвестувати в інший або новий продукт, і Scrum-команда здатна поставляти більше цінності, працюючи над чимось іще. Ось чому збільшення «готового» приросту цінності має важливе значення, оскільки дозволяє бізнесу реагувати на можливості.

Якщо продукти існують рік або більше й повсякчас демонструють цінність унаслідок постійного вдосконалення, може існувати сенс у переході до ітеративної та поступової моделі бюджетування. Можливо, є сенс у фінансуванні продукту та його Scrum-команди впродовж усього фінансового року. Подумайте, яку кількість накладних витрат ви могли б заощадити, не проходячи окремих процесів бюджетування та підтримуючи окремі бюджети для кожної ініціативи, що впливає на продукт цього року.

Цей тип укладання бюджету узгоджується з ідеями, які популяризуються рухом *Beyond Budgeting* (Поза бюджетом)¹. Він стверджує, що з огляду на витрачені зусилля традиційні процеси бюджетування зрідка приносять користь і часто працюють проти того, щоб організація стала стійкою та адаптивною. Це свідчить про те, що організації повинні замінити свої централізовано контролювані й заздалегідь визначені цілі саморегульованими, відносними показниками та передати повноваження щодо ухвалення рішень працівникам на передовій.

¹ Більше інформації про рух *Beyond Budgeting* див. на <https://bbbt.org/the-beyond-budgeting-principles/>.

На планування й прогнозування на більш тривалий час впливають три фактори.

1. Чи ви вже маєте фінансування?
2. Чи є у вас історія (тобто емпіричні дані), яка допоможе вам планувати?
3. Яким є рівень довіри в організації?

На практиці:

зміна планування та фінансування продукту відповідно до зростання довіри та навчання під час роботи в команді

Наявна Scrum-команда працює над продуктом уже три місяці, тому її члени мають спільну історію зі знайомим видом роботи. З цієї історії вони інтуїтивно розуміють, наскільки великими та складними є елементи, з якими вони працюють. Оскільки вони послідовно отримують «виконані» прирости цінності та прозоро поводяться із зацікавленими сторонами щодо прогресу та того, як ухвалюються та підтверджуються рішення стосовно цінності, вони створили довіру. Якщо цей продукт уже має фінансування ще на кілька місяців, цілком імовірно, що беклог продукту команди детально описано для наступної пари спринтів. Власник продукту гарантує, що робота у верхній частині беклогу продукту — це робота, яка найкраще оптимізує цінність для організації.

Далі розглянемо бізнес-лідера, який хоче створити новий продукт та сформувати для цього нову Scrum-команду. Керівнику бізнесу потрібно обґрунтувати, що продукт передбачає розумну рентабельність інвестицій, тому необхідно провести достатнє планування, щоб отримати загальне уявлення про рівень витрат для досягнення початкових цілей продукту. Якщо в цій організації є культура довіри, гнучкий спосіб мислення та досвід ефективного Scrum, Scrum-команда проведе достатньо аналізу для складання початкового плану. Вони регулярно перевірятимуть та пристосовуватимуть загальну ініціативу щодо продукту залежно від зміни

плану та процесу побудови історії з новою Scrum-командою. Власник продукту перевірятиме припущення щодо рентабельності інвестицій під час побудови продукту. Можна зробити запит на додаткове фінансування, якщо є сенс продовжувати інвестувати у вдосконалення продукту.

Тепер розглянемо приклад, коли всередині організації існує недовіра. Попереднє планування часто проводиться через помилкове переконання, що такий вид планування здатний створити впевненість. Scrum-команда може витратити тижні або навіть місяці на деталізацію беклогу продукту, а це коштує грошей та затримує поставку цінності. Із нашого досвіду можемо сказати, тільки-но починається робота, план зазнає частих змін унаслідок зворотного зв'язку з робочим продуктом. Команда, найвірогідніше, стикнеться з великою кількістю відходів у формі відмови від попередніх детальних аналізів та зміни деталей у беклогу продукту.

Зі Scrum ви отримуєте саме те, що треба. Тобто ви хочете, щоби планування було мінімальним, але достатнім.

«Бути Agile» — це не мета

Багато організацій проводять «гнучку трансформацію»; ваша може бути однією з них. Бажати бути більш гнучким — це добре в цілому, за винятком ситуацій, коли «гнучкість» стає метою. Agile-мислення, гнучкі фреймворки (наприклад, Scrum) та agile-практики можуть допомогти вашій організації досягти своїх цілей, але справжньою метою є (або мусить стати) поліпшення цінності, яку надає організація, хай що це означає в цьому конкретному разі¹.

То що ж поганого в зосередженні на «гнучкій трансформації»? Шкода передусім полягає у відсутності зосередженості, а також у ризику розвитку зомбі-Scrum, коли люди просто виконують рухи, навіть гадки не маючи, що вони роблять і навіщо. Якщо ви чітко

¹ Більше на цю тему див. на <https://www.Scrum.org/resources/blog/Scrum-transport-not-destination>.

розумієте, чого хочете досягти, висловлюючись поняттями клієнта, то набагато ефективніше будете йти на компроміси, котрі вам неодмінно знадобляться¹.

**На практиці:
чому бути «гнучким на 100 %» дуже погано**

Якщо бути Agile — погана мета, то «бути гнучким на 100 %» навіть ще гірша, оскільки вона нав'язує agile-підхід людям і командам, які, можливо, не потребують емпіризму чи інновацій. Якщо ви хочете, щоби команди працювали швидше, розбийте їхню роботу на менші кроки та зосередьтеся на усуненні перешкод. Вам потрібен емпіризм, коли ви не можете точно визначити, що будувати або як це робити. Але якщо ви знаєте, що вам потрібно і як це побудувати, тоді просто зробіть це. Не ховайтеся за фасадом гнучкості, думаючи, що це ефективний підхід.

Організації, які потрапляють у пастку «якщо Agile — це добре, нумо його застосовувати скрізь», втратили з поля зору те, чого вони намагаються досягти. Вони забігають наперед. Коли Agile працює, він робить це, оскільки високопрофесійні команди з повноваженнями та самоорганізацією використовують гнучкий підхід для постійної перевірки, адаптації та вдосконалення як свого продукту, так і способу роботи. Для створення цих високоефективних agile-команд необхідні інвестиції часу та грошей, а також потрібні люди.

**Зробіть щось добре,
перш ніж збільшувати використання**

Часом організації стають нетерплячими і хочуть додати більше Scrum-команд, щоби поставити більше цінності. Але якщо наявна команда дуже старається надійно забезпечити доступні додаткові

¹ Scrum.org створив емпіричну систему вдосконалення на основі вимірювань, яка називається «Управління на основі доказів» (Evidence-Based Management), що надає способи вираження цілей та використання заходів для поліпшення вашої здатності досягти цих цілей. Більше інформації див. на <https://www.Scrum.org/resources/evidence-based-management>.

природи цінності продукту, що є придатними для релізу й цінними для споживачів та користувачів, тоді масштабування — це погана ідея. Коли ви намагаєтеся збільшити обсяг, додаючи більше команд, бо хочете, щоби вони дали більше, насправді ви просто масштабуєте наявні виклики й посилюєте хаос і плутанину.

Натомість зосередьтеся на тому, щоби допомогти Scrum-команді бути чудовою. Часто усунення організаційних перешкод, уможливлення вдосконалення командного процесу та створення кращого розуміння бізнесу та клієнтів забезпечують бажаний потік цінності без витрат та додаткових складнощів, пов'язаних з кількома командами. Якщо ви досі вважаєте, що вам потрібно збільшувати обсяги, аби дістатися бажаної точки, у вас тепер є міцний фундамент, на якому можна будувати. Додайте ще одну команду й працюйте над новими завданнями, щоби повернутися до більшого. Потім додайте ще одну команду й ще одну, доки не задовольните свої потреби в командах з постачання продуктів.

Під час масштабування кількох Scrum-команд, які працюють над одним продуктом, робіть це навмисно й використовуйте те, що принесло вам успіх. Самоорганізовані команди, що мають повноваження, визначатимуть, коли і як масштабуватися, постійно перевіряючи та пристосовуючись для вдосконалення як свого продукту, так і способу роботи¹.

¹ Nexus — це підхід для масштабування Scrum для кількох команд, які працюють над одним продуктом. Для отримання додаткової інформації див. <https://www.Scrum.org/resources/scaling-Scrum>.

Підсумки

Структура організації, її процеси та політика, а також її культура мають сильний, а іноді й переважний вплив на колективи, які працюють у ній. Щоби досягти довгострокових переваг від agile-підходів, як-от Scrum, необхідно навчитися використовувати організацію для просування міцної командної ідентичності, аби допомогти командам поліпшити потік у своєму процесі та забезпечити постачання більшої цінності продукту.

Усім організаціям необхідно постійно розвиватися незалежно від того, використовують вони Scrum чи ні. Швидкість їхнього розвитку залежить від того, де вони перебувають сьогодні, куди їм потрібно йти і які зміни впливають на їхній бізнес. Організації, які використовують agile-мислення та емпіризм для посилення сили колективної роботи, можуть бути більш гнучкими та витривалими в дуже мінливому й висококонкурентному світі.

Заклик до дії

Розгляньте зі своєю командою наведені нижче запитання.

- Чи заважають організаційна політика або процеси розвитку людей та команд?
- Чи може організація підтримати (за допомогою часу, грошей чи інших ресурсів) певні сфери, щоби посприяти зміцненню командної ідентичності?
- Як організація, особливо ваше керівництво та зацікавлені сторони, реагує на прозору інформацію, коли оновлення сприймається як позитивне? А коли воно сприймається як негативне?
- Якими способами ви можете допомогти організації прийняти зміни та адаптивність як конкурентну перевагу?
- Як організація використовує емпіризм та зосереджується на цінних результатах у своїй діяльності та політиці (як-от бюджетування)?
- Які проблеми наразі найбільше дошкуляють? Визначте один або два експерименти, які допоможуть залучити організацію до вдосконалення Scrum-команди. Для кожного експерименту обов'язково визначте бажані впливи та спосіб їх вимірювання.

Розділ 8

Висновки й подальші перспективи

Наразі вам слід розуміти, що Scrum — це легкий, простий, але потужний фреймворк, що вводиться в дію кваліфікованою, згуртованою, самоорганізованою командою, яка втілює цінності Scrum і приймає емпіризм.

У книжці ми досліджували структуру Scrum та багато загальних проблем у його застосуванні, а також шляхи їх подолання. Ми представили сім ключових сфер удосконалення професійного Scrum як спосіб допомогти вам досягти більших переваг за допомогою Scrum. Можливості для роздумів і дій розкидано по всій книжці. Ми сподіваємося, що поставили перед вами завдання поліпшити свій спосіб роботи й дали вам нове уявлення про шляхи вдосконалення вашої організації.

Гнучкість бізнесу вимагає нових рішень

Упродовж цієї книжки ми досліджували ітеративний та інкрементальний (поступовий) підхід до отримання цінності в складному та невизначеному світі. В ідеалі ви зосередитеся на простому процесі для розв'язання проблем: спробуйте щось, навчіться, повторіть. Вибудуйте рішення відповідно до вашого зростлого колективного розуміння, реагуючи на зміни відповідно до їх появи.

Організації хочуть забезпечити гнучкість бізнесу, а не просто «робити Scrum». Гнучкість бізнесу означає досить швидку рентабельність інвестицій (ROI), гнучкість та контроль над інвестиційними рішеннями, а також здатність легко змінювати напрям, коли

виникають нові можливості або ризики. Ітеративний та інкрементальний підходи самі собою не завжди досягають цих цінних результатів. Ядро Scrum — гнучкий спосіб мислення, емпіризм та робота в команді — забезпечує необхідну основу для усвідомлення переваг згаданих підходів.

Звертаючи увагу на відгуки Scrum-команди, ринку та зацікавлених сторін, організації можуть отримати цінну інформацію, яка допоможе їм постійно вдосконалювати свої продукти та процеси, що використовуються для їх створення. Ці цикли зворотного зв'язку керуються трьома елементами емпіризму: прозорістю, перевіркою та адаптацією. Зворотний зв'язок, підкріплений цінностями Scrum, створює культуру навчання, яка заохочує до експериментів та постійного вдосконалення.

Ця культура навчання пронизує всі аспекти того, як організація поставляє продукт, від найперших бесід про те, що може бути гарною ідеєю (гіпотеза), до постачання кожного «готового» приросту цінності. Організація постійно вдосконалює як продукт, так і підхід до його постачання, слідуючи своєму баченню продукту.

У Розділі 1 «Постійне вдосконалення вашої практичної діяльності в Scrum» ми представили сім ключових сфер, які підтримують удосконалення: гнучке мислення, емпіризм, командна робота, командний процес, командна ідентичність, цінність продукту та організація. Протягом цієї книжки ми розширили кожен з цих ключових сфер, показавши, як їх використовувати, щоби допомогти Scrum-команді поліпшитися.

Ми сподіваємося, що за цей час ви засвоїли кілька важливих уроків.

- *Без обізнаних, кваліфікованих та відданих людей Scrum не працює.* Створення цінних рішень у складному світі вимагає команд, які мають мотивованих людей, наділених різними видами знань, умінь та досвіду. Цим командам потрібен сприятливий простір, у якому їхні члени можуть співпрацювати для створення креативних рішень та отримувати з цього практичні уроки.
- *Команди потребують допомоги організації.* Команди живуть всередині організацій, чия культура буде формувати їх, або допомагаючи розширюватися, або обмежуючи їхнє зростан-

ня. Коли виникає суперечка між командою та організацією, придивіться, що зумовлює конфлікт між тим, ким є команда, і тим, як працює організація. Спробуйте зрозуміти, де і чому виникає конфлікт між цінностями Agile та принципами й поведінкою, притаманними організації.

- *Спосіб мислення формує культуру, а культура впливає на процес.* Наявність позитивного agile-мислення допоможе сформувати культуру, яка сприяє максимізації переваг Scrum та стійкості до складного й швидкозмінного середовища.
- *Опанування Scrum — це подорож, а не пункт призначення.* Практика поліпшує, але ніколи не досягає ідеалу. Що більше ви практикуєте усвідомлення (себе та навколишніх), розмірковуєте, а потім дієте з намірами, то більш творчо й продуктивно розв'язуватимете проблеми та надаватимете рішення максимально можливої цінності. Напрямою вашої подорожі, напевно, також зміниться, тому залишайтеся відкритими для можливостей.

Регулярний перегляд запитань оцінювання в Додатку А «Самооцінка для розуміння того, де ви перебуваєте» допоможе вам в пошуках шляхів самовдосконалення та вдосконалення своїх команд. Поставте собі наведені нижче запитання.

- Що ви дізналися?
- Що змінилося?
- Які ризики чи можливості відкриваються?
- Де ви бачите загальні теми?
- Які тенденції ви помічаєте?

Подібно до того, як поліпшуються індивідуальні навички, Scrum-команда та організація можуть вдосконалюватися шляхом активних експериментів для зростання й навчання, а також щодалі швидше реагуючи на ідеї та зміни. Команди та організації вдосконалюються, розвиваючи наведені нижче активи.

- Суворорієнтація на цінність.
- Невпинне акцентування на технічній досконалості та досягненні результату.
- Дух постійного вдосконалення, постійний пошук «кращого».

Усвідомте вашу поточну ситуацію й шукайте той найменший наступний крок, який принесе найбільшу цінність у просуванні до вашої мети. Змінювати важко, а використання фреймворку сприятиме тому, що зміни будуть більш цілеспрямованими, дисциплінованими та ефективними. Чітке розуміння застосовуваної структури та використання її за призначенням забезпечить найкращі переваги.

Заклик до дії

Наприкінці кожного розділу ми публікували «Заклик до дії», покликаний змусити вас рухатися вперед у вашій поточній подорожі, щоби максимізувати переваги Scrum. Ваша подорож не завершується тут, наприкінці цієї книжки; насправді вона тільки починається. Поміркуйте над своїми ключовими поглядами, діями, які ви вже вчинили, та результатами, які ви спостерігали від цих дій. Спочатку зосередьтеся на створенні сильної команди та на основах емпіризму, а потім будуйте на основі зворотного зв'язку. Завдяки досвіду та практиці діяльність, яка спочатку була дуже важкою, стане такою, яка майже не потребує зусиль.

Подумайте про те, щоби запланувати у вашому календарі час для регулярного міркування про своє навчання, спостереження та розуміння своєї подорожі до майстерності Scrum. Поміркуйте, як ви берете до уваги ці знання, щоби краще служити своїй Scrum-команді та організації (а можливо, і за її межами). Постійно перевіряйте свої результати та адаптуйте свій підхід.

Сила Scrum полягає в простоті. Оскільки світ стає більш мінливим, невизначеним і неоднозначним, вам знадобиться простота Scrum, щоб орієнтуватися в складнощах середовища, які постійно зростають. Лідери, що сприймають простоту та емпіризм, дають можливість своїм людям, командам та організаціям процвітати й, відповідно, надавати кращі рішення світових викликів. Ми закликаємо вас бути частиною місії Scrum.org щодо вдосконалення професії постачання продукту. Нехай професіоналізм керує вами.

Ми бажаємо вам усього найкращого у вашій власній подорожі навчання та вдосконалення.

Хай буде Scrum!

Додаток А

Самооцінка для розуміння того, де ви перебуваєте

Перш ніж зможете стати кращими, вам необхідно знати, де ви, можливо, не досягли своїх цілей. Наведені запитання допоможуть визначити слабкі місця, які найбільше вам дошкуляють. Ці проблеми часто є найбільш помітними для більшості людей, і вони зазвичай взаємопов'язані. Вони часто викликають у вашої команди відчуття тиску, тривоги або апатії. Спробуйте відповісти на кожне запитання відкрито та об'єктивно, а якщо ви не впевнені, подумайте, як отримати більше інформації.

Бізнес-гнучкість

Бізнес-гнучкість та ефективне використання Scrum безпосередньо пов'язані. Якщо ви «робите Scrum», але не усвідомлюєте бажаних результатів бізнес-гнучкості, вам слід подумати про те, як ви застосовуєте Scrum-фреймворк із гнучким мисленням. Щоби визначити, наскільки гнучкий у вас бізнес, оцініть ступінь згоди з наведеними нижче твердженнями за шкалою 1–10 (1 — максимально не згоден, 10 — максимально погоджуюсь).

- Ваша організація задоволена рентабельністю вашого продукту.
- Ви отримуєте «готовий» приріст цінності продукту (тобто потенційно доступний) принаймні раз на спринт.
- Ваші клієнти задоволені частотою релізів.
- Відгуки зацікавлених сторін та покупців беруться до уваги в продукті для поліпшення його цінності.

- Ви підтверджуєте припущення про цінність роботи, яку виконуєте, на основі відгуків ринку, клієнтів або користувачів.
- Ви можете поставити нові можливості продукту за прийнятний проміжок часу.
- Ви можете відповісти на нові можливості або ризики за прийнятний проміжок часу.
- Ви розумієте та маєте докази, які підтверджують потреби ваших клієнтів.
- Ви розумієте, як користувачі чи клієнти застосовують продукт, зокрема те, які функції вони використовують.
- Ви розумієте поточні та актуальні ринкові умови для вашого продукту.
- Ваші клієнти відчують, що рівень якості вашого продукту високий.
- Ви витрачаєте прийнятне співвідношення своїх інвестицій у продукт, на його підтримку або виправлення дефектів (порівняно з новими можливостями продукту).
- Ваші команди дуже задоволені своєю роботою.
- Ваші команди дуже задоволені своїми можливостями навчання та зростання.

Ефективний емпіризм зі Scrum

Щоби визначити рівень своєї емпіричності, оцініть ступінь згоди з наведеними нижче твердженнями за шкалою 1–10 (1 — максимально не згоден, 10 — максимально погоджуюсь).

У нас є власник продукту й...

- У нас є єдиний власник продукту, який уповноважений ухвалювати рішення щодо максимізації цінності продукту.
- Власник продукту регулярно повідомляє про власне чітке бачення продукту.
- Власник продукту має легкий доступ до даних, щоби допомогти команді виміряти вплив змін на продукт.
- Власник продукту обізнаний про стратегію продукту та про те, як вона узгоджується з бізнес-цілями.
- Власник продукту активно збирає думки зацікавлених сторін та вловлює їхні очікування.

- Власник продукту може допомогти відповісти на запитання та визначити цінність, створену командою розробників під час спринту.
- Власник продукту може приділити достатньо часу вивченню потреб споживачів/користувачів, формуванню й обговоренню концепції продукту та вивченню нових способів постачання цінності.

У нас є Scrum-команда й...

- Scrum-команда відчуває себе вповноваженою вносити зміни до свого процесу діяльності та інструментів.
- Scrum-команда активно прагне зменшити втрати у своїх процесах.

Ми плануємо спринт і...

- Уся Scrum-команда повноцінно бере участь і досягає мети протягом відведеного часу (таймбоксу).
- Scrum-команда створює ціль спринту, яка чітко пояснює необхідність його проведення.
- Scrum-команда здатна ефективно й дієво планувати спринт завдяки своїм знанням та інформації, яка міститься в беклогу продукту.
- Наприкінці спринту стає зрозуміло, чи досягли ми його цілі.

Ми проводимо спринти й...

- Спринт триває місяць або менше, а його тривалість залишається незмінною.
- Спринт досить короткий, щоби дати бізнесу необхідну гнучкість для обмеження інвестиційного ризику, отримання зворотного зв'язку заради підтвердження припущень та досить частої зміни напрямку.
- Завжди є потенційно доступний приріст, який забезпечує цінність для бізнесу до кінця кожного спринту.
- Scrum-команда послідовно відповідає своїм цілям спринту.
- Команда розробників кожного спринту дізнається більше про бізнес-потреби завдяки співпраці з власником продукту та іншими зацікавленими сторонами.

У нас є приріст цінності й...

- Визначення «готового» приросту відображає продукт, який випускається, і з часом розширюється для поліпшення якості та повноти.
- Scrum-команда не знижує якість під тиском необхідності постачати більше.
- Власник продукту ніколи не дивується приросту, перевіреному в огляді спринту.
- Зацікавлені сторони майже завжди задоволені приростом цінності, показаним в огляді спринтів. Коли ж ні, тоді власник продукту відповідає, використовуючи зворотний зв'язок для адаптації беклогу продукту.

У нас є команда розробників і...

- Члени команди розробників мають право самостійно вирішувати, як розробляти й постачати приріст цінності, і вони відчують себе уповноваженими ухвалювати ці рішення.
- Усі члени команди розробників чітко розуміють визначення «готового» приросту.
- Команда розробників працює над уточненням і розширенням визначення «готового» приросту, щоб з часом поліпшити якість та повноту.

У нас є беклог продукту й...

- Беклог продукту доступний і зрозумілий для Scrum-команди та зацікавлених сторін.
- Беклог продукту — це впорядкований список, який відображає поточні плани щодо продукту.
- Беклог продукту чітко визначає цінність кожного елемента.
- Беклог продукту часто вдосконалюється та оновлюється, оскільки більше інформації отримується шляхом постачання продукту та перевірки змін у навколишньому середовищі.
- Беклог продукту — це не просто перетворення інших документів із вимогами, переданих Scrum-команді.
- Беклог продукту — це не просто список кожного запиту клієнта/зацікавленої сторони, який постійно зростає, він радше відображає продуману відповідь на потреби клієнтів та бажані результати.

У нас є беклог спринту й...

- Беклог спринту чітко повідомляє про прогрес на шляху до цілі спринту.
- Принаймні одне дієве поліпшення, яке можна здійснити порівняно з попередньою ретроспективою спринту, міститься в беклогу спринту.
- Команда розробників часто оновлює беклог спринту, щоб відображати нові знання під час просування роботи.
- У кінці спринту робота рідко буває частково завершеною.

У нас щоденна Scrum-нарада й...

- Уся команда розробників повністю бере в ній участь і досягає мети впродовж установленого строку.
- Щоденна Scrum-нарада — це сесія спільного планування, яку проводить команда розробників.
- До кінця щоденної Scrum-наради команда розробників розуміє прогрес у досягненні цілі спринту, будь-які перешкоди, що блокують або вповільнюють прогрес, і план на наступні 24 години.
- Щоденна Scrum-нарада — це не просто оновлення статусу, а активний спосіб співпраці, який допомагає поліпшити здатність команди постачати цінність.

У нас є огляд спринту й...

- Уся Scrum-команда та запрошені зацікавлені сторони повноцінно беруть участь і досягають мети впродовж встановленого строку.
- Огляди спринтів є спільними й генерують корисні відгуки про приріст та напрям розвитку продукту.
- Усі необхідні зацікавлені сторони беруть участь, щоб надати відповідні та корисні відгуки.
- Зацікавлені сторони розуміють значення емпіризму під час розв'язання складних проблем; вони не розглядають Scrum як простий спосіб виконати більше роботи за короткий час.
- Зацікавлені сторони стурбовані тим, щоб приріст цінності, який буде заявлений у релізі, відповідав комерційній цілі, а не тим, чи було виконано всі передбачувані еле-

менти беклогу продукту або дотримано всіх відповідних процедур.

- Більший акцент робиться на придатному для релізу високоякісному прирості, який забезпечує цінність, аніж на виконанні всього, що було заплановано під час планування спринту.

У нас є ретроспектива спринту й...

- Уся Scrum-команда повністю бере участь і досягає мети впродовж визначеного строку.
- Ретроспективи спринту містять відкриті й змістовні дискусії про те, як працює Scrum-команда.
- Під час ретроспективи спринту Scrum-команда визначає практичні поліпшення, які необхідно впровадити в наступному спринті.
- Scrum-команда своєчасно виконує взяті на себе зобов'язання й оцінює результативність.

У нас є Scrum-майстер і...

- Scrum-майстер гарантує, що Scrum-команда розуміє та дотримується Scrum-фреймворку, водночас дозволяючи самоорганізацію та заохочуючи до неї.
- Scrum-майстер втілює в собі цінності Scrum та емпіризм.
- Scrum-майстер активно допомагає команді розробників зробити свою роботу прозорою.
- Scrum-майстер допомагає Scrum-команді прийняти цінності Scrum та емпіризм.
- Scrum-майстер утілює гнучкість в організації та активно працює над усуненням організаційних перешкод.
- Scrum-майстер розуміє стан здоров'я Scrum-команди й допомагає поліпшити спільну роботу в команді.
- Scrum-майстер вносить зміни для підвищення продуктивності та якості, не підриваючи самоорганізації.
- Scrum-майстер діє як лідер, який служить, вимірюючи власний успіх зростанням команди та її успіхом в досягненні переваг Scrum.

Ефективна робота в команді зі Scrum

Щоби визначити ефективність командної роботи у впровадженні Scrum, беріть до уваги рівень застосовності наведених нижче тверджень, оцінюючи їх за шкалою 1–10 (1 — максимально не згоден, 10 — максимально погоджуюсь).

Прихильність та зосередженість

- Ми вимірюємо успіх на основі досягнення спільних цілей та результатів.
- Ми несемо відповідальність одне перед одним за виконання взятих на себе зобов'язань та дотримання стандартів якості.
- Ми ухвалюємо рішення, засновані на цінностях та консенсусі.
- Усі члени команди підтримують рішення команди та план дій.
- Усі члени команди готові вийти за межі своєї зони комфорту.
- Ми зосереджуємося на одній-двох дрібних справах, виконуємо їх, а потім переходимо до наступної справи.

Відкритість і сміливість

- Усі члени команди активно допомагають іншим у команді.
- Усі члени команди активно звертаються по допомогу.
- Усі члени команди швидко визнають помилки та шукають рішення.
- Усі члени команди готові поділитися своїми ідеями та поглядами.
- Ми можемо впоратися з провалами або невдачами, розмірковуючи про те, що дізналися, та намагаючись застосувати інший підхід.
- Ми активно й прямо порушуємо питання.
- Ми можемо адаптуватися до несподіваних змін.
- Члени команди кидають виклик припущенням як власним, так і притаманним організації в цілому.

Повага

- Усі члени команди здібні й бажають розв'язувати конфлікти з повагою та продуктивно.

- Ми часто надаємо одне одному конструктивний зворотний зв'язок з метою індивідуального й командного зростання.
- Усі члени команди відкриті, щоби почути й розглянути різні ідей та погляди.
- Ми працюємо над поліпшенням кросфункціональності й розширенням знань про бізнес, технології та процеси в нашій команді.

Аналіз оціночних відповідей

Поверніться й перегляньте свої оціночні відповіді. А тепер розгляньмо поточні тенденції щодо кожної з них.

- У яких сферах ви рухаєтеся до бажаних результатів?
- У яких сферах у вас спостерігається застій?
- У яких сферах ви рухаєтеся назад?

А тепер розгляньте наведені нижче запитання.

- Чи здається вам, що ви просто виконуєте якісь дії? Чи ви приходите працювати на совість, віддано та з наміром?
- Наскільки відкрито й чесно обговорюються в Scrum-команді проблеми й ідеї щодо поліпшення?
- Наскільки члени Scrum-команди контролюють свою роботу з розробки й постачання продукту?
- Яку підтримку надає керівництво для усунення організаційних перешкод?

Додаток Б

Поширені хибні думки щодо Scrum

Scrum не є методологією або процесом управління

Методологія — це «сукупність методів, правил і постулатів, які застосовуються в усталеній практиці; конкретна процедура або набір процедур»¹. Методологія не працює в складній і непередбачуваний сфері діяльності, оскільки вона покладається на знання рішень та кроків, які вас до цього приведуть. У складних сферах діяльності команди повинні використовувати емпіричний підхід для отримання цінності або розв'язання проблем. Вони мусять уміти визначати власні дії та техніки й пристосовувати їх у процесі того, як дізнаватимуться більше під час виконання роботи.

Scrum — це фреймворк процесу, який забезпечує мінімальні межі, що дозволяють самоорганізуватися, одночасно обмежуючи ризик. Упровадження Scrum у всіх буде відбуватися по-різному, оскільки кожен продукт і кожна команда матимуть унікальні потреби.

Багато Scrum-команд працюють в організаціях, які глибоко піклуються про управління ризиками. Вони хочуть переконатися, що кошти витрачаються ефективно та узгоджуються з бізнес-цілями, що стандарти безпеки та захисту виконуються, і не допускають юридичних або нормативних зобов'язань. Ці організації впроваджують процеси управління, щоби спробувати контролювати ці ризики.

¹ Узято з дозволу з Merriam-Webster.com © 2019 by Merriam-Webster, Inc. <https://www.merriamwebster.com/dictionary/methodology>.

Scrum — це не процес управління, але він може допомогти вам контролювати ризики. Scrum може зробити відходи в організаційних процесах більш прозорими, що дозволить проводити подальшу перевірку та адаптацію, аби зробити ці процеси більш ефективними. Scrum може допомогти забезпечити узгодженість з бізнес-цілями та надає можливості для ранньої адаптації, приводячи постачання у відповідність до бізнес-цілей. Коли організація зосереджується на визначенні бажаних результатів щодо управління, ризику та дотримання вимог, а не методів, це дозволяє командам досягти таких результатів у найбільш ефективний із можливих способів. Невпинно зосереджуючись на створенні «готового» приросту, команда водночас управляє ризиком постачання продукту.

Scrum — це не «панацея» чи спосіб пришвидшити розробників

Просто «робити Scrum» не розв'яже ваших проблем. Люди роблять Scrum ефективним; вони розробляють креативні виходи із ситуації, ухвалюючи обґрунтовані рішення на основі емпіричних даних. Коли люди втілюють цінності Scrum, речі, що їх стримують, стають більш помітними. А коли все на видноті, є шанс на поліпшення. Добре виконаний Scrum — це як прожектор, що світить на вашу організацію й те, як ви працюєте,— і на гарне, і на погане.

Використання Scrum, імовірно, підвищить ефективність у процесі розробки, а це означає, що деякі дії заберуть менше часу через акцент на постійному вдосконаленні. Але це не відповідає суті Scrum. Scrum зосереджується на частому постачанні цінності. Scrum — це робота в інший спосіб, щоби цінні частини продукту можна було випускати частіше.

Швидко постачання більшої кількості неправильних матеріалів не дуже продуктивна річ. Ви повинні ефективно визначати, що є цінним. Мусите ефективно адаптуватися до зворотного зв'язку, емпіричних вимірювань реальної цінності та змін на ринку.

Ви повинні бути ефективними, щоби дійти до «готового» приросту. Scrum допомагає досягти такого рівня продуктивності.

Що ж до постачання продукту, ми обираємо результативність, а не оптимальність.

Оптимальність — це вторинна перевага Scrum, і команди розробників визначають, де в них є можливості для її підвищення у своїй роботі. Ключ у тому, щоби застосувати прозорість, отриману завдяки «готовому» приросту, для перевірки цінності. Можливість досягти «готового» приросту і розблокувати потік цінності є природною перевагою системи.

Власник продукту не документує вимоги

Відповідальність власника продукту зосереджена на максимізації цінності товару. Це набагато більше, ніж просто написання вимог.

Власник продукту співпрацює із зацікавленими сторонами, щоби зрозуміти їхні потреби та визначити очікування. Він передає своє бачення продукту відповідно до бізнес-стратегії та місії організації. Власник продукту ухвалює важкі рішення щодо конкурентних пріоритетів і бажань. Він повинен розуміти користувачів і те, як вони використовують продукт.

Попри те, що власник продукту відповідає за вміст беклогу продукту та порядок (пріоритетність) цих елементів, він, імовірно, делегує деякі або більшість деталей.

Беклог продукту не є Agile-версією традиційного документа з вимогами

За допомогою Scrum організація зосереджується на постачанні ситуативно обумовлених елементів цінного функціоналу. Беклог продукту — це представлення вимог до продукту, воно може бути змінене в будь-який час, і має змінюватися, коли команда більше дізнається про продукт, надаючи збільшення приросту його цінності. Він повинен розвиватися відповідно до того, як команда більше дізнається про користувачів, зміни на ринку та зміну бізнес-потреб. Навіть під час спринту елементи беклогу продукту можуть змінюватися, оскільки в процесі його створення команда уточнює та підтверджує загальне розуміння.

Продукт беклогу не передається Scrum-команді, коли вона починає проект. Так само його не слід розглядати і як контракт, що перешкоджає спільним переговорам та відкритості до змін.

Беклог продукту не є списком усіх запитів

Беклог продукту забезпечує прозорість того, що наразі заплановано в продукті. Власник продукту несе відповідальність за оптимізацію цінності, що здійснюється шляхом замовлення беклогу продукту. Якщо беклог продукту містить елементи, які ви не збираєтеся реалізовувати, оскільки вони мають низьку цінність або не узгоджуються з напрямом продукту, тоді ви ускладнюєте досягнення прозорості.

Кожен запит не повинен додаватися до беклогу продукту лише для того, щоби зацікавлені сторони почувалися добре, бо їхній запит було туди записано. Власник продукту виконує свою відповідальність, ухвалюючи важкі рішення, що буде зроблено стосовно продукту. Власник продукту може сказати: «Не зараз». Якщо цей елемент стане актуальним у майбутньому, він з'явиться знову і може бути доданий до беклогу продукту в той момент.

Щоденна Scrum-нарада не є статусною

Щоденна Scrum-нарада — це сесія спільного планування для перевірки прогресу щодо досягнення цілі спринту та адаптації плану. Вона не призначена для обліковування того, як кожна людина провела попередній робочий день. Scrum-нарада не має на меті зосередитися на окремих оновленнях статусу, оскільки це одночасно спричиняє втрату з поля зору цілі спринту та перешкоджає спільній відповідальності команди розробників щодо створення «готового» приросту.

Щоденна Scrum-нарада не призначена для надання оновлення статусу людям, які не належать до команди розробників. Така динаміка може призвести до відсутності прозорості в процесі та відволікти увагу від співпраці з командою розробників.

Спринт може бути успішним, навіть якщо всі заплановані елементи беклогу спринту не завершено

Навіть упродовж коротших часових рамок спринту постачання продукту тягне за собою великі складнощі й непередбачуваність. Беклог спринту — це не обіцянка поставити певний набір елементів протягом певного часу. Ми усвідомлюємо, що не можемо

ідеально планувати, тому погоджуємося на цю неоднозначність і ухвалюємо обґрунтовані рішення стосовно адаптації нашого плану на основі того, про що ми дізнаємося в процесі появи рішення. Ми використовуємо ціль спринту, щоби зосередитися й адаптувати беклог спринту відповідно до появи більшої кількості інформації.

Успішний спринт надає доступний для релізу продукт. Успішний спринт забезпечує цінність.

Scrum-майстер не відповідає за відстеження роботи команди розробників

Команда розробників є самоорганізованою, а це означає, що члени команди відповідають за моніторинг власного прогресу та адаптацію власного плану. Scrum-майстер може підтримати команду розробників, навчаючи їх методів ефективнішої спільної роботи та підвищення прозорості власного прогресу й результатів.

Огляд спринту — це не зустріч заради схвалення

Придатний для релізу приріст цінності вже створюється, коли команда потрапляє до огляду спринту. Огляд спринту — це можливість зібрати відгуки та разом обговорити, що робити далі, але він не призводить до ухвалення чи відхилення рішення.

Якщо зацікавлені сторони незадоволені продуктом, який вони бачать на огляді спринту, власник продукту так само ухвалює рішення про те, як адаптувати беклог продукту. Якщо власник продукту не задоволений тим, який продукт бачить на огляді спринту, йому слід подумати, як співпрацювати з командою розробників під час спринту, щоб отримати бажаний продукт.

Для початку спринту не потрібно багато готуватися

Scrum-команда може запустити спринт, якщо в неї є власник продукту з концепцією продукту, команда розробників, яка володіє всіма навичками для створення «готового» приросту цінності, та Scrum-майстер. Доопрацювання беклогу продукту відбувається в процесі.

Звичайно, корисно мати певні початкові заходи, як-от навчання та спрощене визначення процесу (наприклад, створення розуміння «готового» приросту).

Це також, імовірно, гарний час для пояснення, що не існує поняття, як-от спринт 0, інфраструктурний або дизайнерський спринт. Призначенням спринту є отримання «готового» приросту. Якщо ви не плануєте робити «готовий» приріст, то не проводите спринт. Робіть усе, що вважаєте за потрібне, щоби підготуватися до початку (у межах розумного — не втрапте в аналітичний параліч!). Лишень не називайте цей час, відведений на підготовку, спринтом 0.

Предметний покажчик

А

автоматизація, «виконані» прирости, 94–96
автономія членів команди, 49
адаптивні команди, характеристики, 71–72
аналіз, 90–91
аналіз першопричини, 39–41
аутсорсингова підтримка продукту, 178–179

Б

безперервне вдосконалення (самооцінювання), потреба в, 36
беклог продукту, 109, 118, 130, 145
 прогнозування, 134
 прозорість, 133
 самооцінювання, 200
 уточнення, 131
бізнес-гнучкість,
 негайні рішення, 193–196
 самооцінювання, 197–198
бізнес-цілі, вимірювання,
 досягнення, 112
 цілі спринту, 82–83
бюджетування, ітеративне та інкрементальне, 186–187

В

великі релізи, 144
викладацькі навички, 32
вимірювання,
 аналіз потоку виконання, 90–91
 Scrum-майстри, 163
 показників дефектів, 98

- показників покриття коду, 97
- показників складності, 97
- показників стабільності збірки, 98
- поточку виконання, 90
- прогресу й успіху, 108
- успіху, 127
- цінності, 111
- відгук (зворотний зв'язок), 43
 - зацікавлена сторона, 121
 - та швидке постачання, 107
- відкритість (командна робота), самооцінювання, 203
- відповідальність, 57–58, 66
 - Scrum-команди, 57–58
 - Scrum-майстри, 161–163
 - організації, поліпшення, 181
 - співпраця команд, 66
- відповіді, аналіз, 204
- відстеження,
 - переривання, 155
 - перешкод, 153–155
 - прогресу, 108
- візуалізація,
 - Scrum-дошки, 86–88
 - згоряння завдань у спринті, 88–89
 - «змія відходів», 152–153
 - технічний борг, 99–100
- власники продукту, 18, 28, 32, 123, 130, 143, 183
 - бюджетування, 186–187
 - індивідуальні та командні можливості, 158
 - карти впливу, 116
 - перешкоди, 151
 - самооцінювання, 198
 - цінність продукту, 78, 108
- внутрішня мотивація, члени команди, 49

Г

- гіпотеза як елемент беклогу продукту, 116
- «готовий до роботи», 132, 145
- «готовий» приріст цінності, 58, 75–76
- графік пропускну здатності, 91
- групове оцінювання, 136
- груповий розвиток, модель Такмена, 67

Д

- діагностування проблем продукту, 112–113
- довіра
 - будування, 61–62
 - згуртовані команди, 61
- догматичний Scrum, 38
- документування вимог,
 - беклог продукту, 207–208
 - власники продукту, 207

Е

- еволюція організацій, потреба в, 174
- емоційний інтелект, члени команди, 49
- емпіризм, 9, 17, 35, 46, 75, 105, 123–126, 128, 135, 162, 165, 190, 201
 - адаптація та вдосконалення, 25–26, 42
 - ефективність, 198
 - застосування навчання, 134
 - командна робота, 28, 40
 - оптимізація, 135
 - планування, 128–129
 - суть, 169
 - як основа Scrum, 26–27, 194

З

- заблокований елемент беклогу, 91
- задоволеність клієнтів, вимірювання, 111
- залізний трикутник, 127, 183–185
- залучення зацікавлених сторін, 113, 156
 - відчуття цілеспрямованості, 81
 - зворотний зв'язок, 119, 121
 - Канбан, 91
- захисні зони, Scrum-фреймворк, 30
- збільшення релізів, 144
- згоряння завдань у спринті, 88
 - вимірювання та аналіз потоку, 91
 - графік, 88
- згуртовані команди, 28, 60
 - Fist of Five (кулак), 65
 - відповідальність, 66
 - довіра, 61–63
 - консенсус, 64

конфлікт, 63
 римське голосування, 65
 зміна планування/фінансування продукту, 188–189
 зомбі-Scrum (механічний Scrum), 38, 189

I

ідентичність Scrum-команд, 47–48
 ілюзії контролю, відмови від, 182
 ілюзія контролю, 182
 імовірнісне прогнозування, 141
 індивідуальний/командний розвиток, 174
 аутсорсинг, 178–179
 вплив оглядів ефективності та компенсації, 175
 індивідуальні кар'єрні шляхи, 176
 розподілені команди, 179–180
 стратегії пошуку ресурсів, 177
 індивідуальні можливості, зростання, 158
 ініціативи, фінансування, 185
 інтелект (емоційний), члени команди, 185

K

Канбан, поліпшення прозорості та ефективності потоку, 91–92
 кар'єрні шляхи, 176–177
 команда розробників, 32–35, 52–53, 56, 80, 86, 96, 200
 команди,
 автономія, 49
 адаптивні, характеристика, 71
 безперервне вдосконалення, потреба, 36–37
 викладацькі навички, 32–33
 вимірювання успіху, 108
 відповідальність, 57
 внутрішня мотивація, 49
 впливи/результати, 177–178
 добір учасників, 51–53
 емоційний інтелект, 49–50
 емпіризм, 25–28
 згуртовані, 29, 47, 57, 179, 193
 ідентичність, 29, 40–41, 46–48, 54, 109
 ітеративне та інкрементальне бюджетування, 186–187
 кар'єрні шляхи, 176–177
 конфронтація, 67
 кросфункціональні, 176

лідерство як служіння, 35–36
 межі, 58–59
 мета діяльності, 29, 49, 55
 модель групового розвитку Такмена, 66–67
 можливості, 31–36, 158
 мотивація до змін, 148
 навички навчання, 34
 навички/таланти членів команди, 48
 навички фасилітації, 33
 прогрес, 66–68, 108
 продуктивні, 70–72
 професійні спільноти, 160–161
 робочі угоди, 53–54
 розвиток, 174–175
 розподілені, 179–180
 розробників, 52–53
 самоорганізовані, 28, 38, 47, 49, 55, 60, 64, 73, 176
 самооцінювання, 199–200
 спільні цілі, 57
 стабільні, 28–29, 69
 стратегії пошуку ресурсів, 177–178
 формування, 67–68
 цикли зворотного зв'язку, 17
 командна робота,
 конфронтація, 67
 командні можливості, зростання, 158
 командні процеси, 26
 консенсус,
 згуртовані команди, 65
 цілі спринту, 83
 конфлікт,
 згуртовані команди, 63
 спектр, 63–64
 концепція продукту, 109
 карта впливу, 114
 підживлення, 110
 користувачькі історії, 113, 117–118
 кросфункціональні команди, 176

Λ

лідерство,
 та самоорганізація, 59
 як служіння, 32, 35–36

М

маленькі релізи, 144
 масштабування організацій, 190–191
 межі, Scrum-команди, 58–59
 мінімально життєздатний продукт (MVP), 129
 модель змін Сатір, 69–71
 можливості (індивідуальні/командні), 158

Н

навички наставництва,
 Scrum-команди, 34
 Scrum-майстри, 166–167
 навички/таланти членів команди, 48
 навички фасилітації, 33
 навчання, 159
 застосування, 134
 культура, 69
 навички, 32, 34
 постійне, 158
 чітка відповідальність, 57
 як цінність, 120
 наслідки (результати),
 кількісна оцінка, 153
 команди, 177
 негайні рішення та бізнес-гнучкість, 193–196
 незавершена робота,
 вимірювання потоку виконання, 90
 відстеження, 91
 обмеження, 89–90
 прозорість, 91–92
 незавершений Scrum, 37
 неправильне розуміння Scrum, 205

О

образ споживача, 110, 113–116, 144
 огляди спринту, 113, 124, 147, 186
 перешкоди, 156
 самооцінювання, 200–201
 суть, 209
 цінність, 121

одноденні спринти, 85
 особистість, члени команди, 48
 оцінка,
 на основі обсягу, 135–136
 планування, 134–136

П

парне програмування, 90, 159
 перегляд коду, 96
 переривання, відстеження, 155
 перешкоди, 20, 172
 Scrum-майстри, 36, 148, 169, 202
 виявлення, 160
 графік згорання завдань, 88
 зменшення залежностей, 134
 «змія відходів», 152
 кількісна оцінка, 153–155
 командна робота, 30, 71
 та зосередження на швидкості, 157
 усунення, 151, 156–157, 190
 підготовка до спринтів, 209
 підтвердження, користувацькі історії, 117
 підтримка продукту, аутсорсингова, 110
 підходи,
 на основі контексту, Scrum-майстри, 165–166
 різні підходи, експериментування, 43
 план розвитку продукту, 110
 планування,
 експерименту, 43
 емпіричне, 128
 зміна, 188
 поліпшення, 20, 125
 продуктове мислення, 126
 релізів, 143–144
 спринту, 78, 83, 137–138
 створення узгодженої структури, 130
 показники
 дефектів, 98
 покриття коду, 97
 стабільності збірки, «виконані» прирости, 98
 потік,
 вимірювання та аналіз, 90–91
 застосування Канбан, 91–92

прийом «Зараз, наступний крок, майбутнє», 80
 природи цінності, самооцінювання, 200
 пристосування, емпіризм, 9, 17
 проблеми з продуктом, діагностування, 112
 провал як успіх, 44
 прогнозування,
 імовірнісне, 141
 спринти, 139–140
 прогрес,
 Scrum-дошки, 86–88
 вимірювання, 108, 143
 графік згоряння завдань, 88–89
 прозорість, 40, 209
 самооцінка, 201
 технічний борг, 102
 щоденні наради, 84
 продуктивні команди, характеристики, 71
 продуктивність,
 Scrum-команди, 70
 огляди, 175
 прозорість, 180–181
 професійні спільноти, 160–161
 процеси управління, 205–206

Р

релізи, 57, 76
 короткі, 106
 обсяг, 144
 планування, 126, 143
 ретроспективи спринту, 42–44, 68, 71, 147–152, 165
 різні підходи, експериментування, 43–45
 робочі угоди, 53–54
 розв'язання проблем,
 бізнес-гнучкість, 193–196
 експериментування з підходами, 43–44
 розподілені команди, 179–180

С

самоорганізовані команди, 28, 38, 47–49, 55, 60, 64, 73, 83, 191
 самооцінювання, 197–204

сміливість (командна робота), самооцінювання, 203
 спринти, 103–104, 171, 185
 беклоги, 84–87, 132, 137, 142, 145, 209
 «готовий» приріст цінності, 39, 47, 58, 138, 197
 графік згоряння завдань, 88
 огляди, 113, 121, 156
 підготовка до, 209
 планування, 78, 137–138, 153
 поліпшення наданої цінності, 119
 приріст цінності продукту, 27, 38, 40, 77
 прогнозування, 140
 релізи, 143
 ретроспективи, 42–44, 68, 71, 147–152, 165
 самоорганізація команди, 135
 самооцінювання, 199–202
 сприяння консенсусу, 65
 технічний борг, 101–102
 тривалість, 139–140
 цілі, 56–57, 75, 81–83
 щоденні Scrum-наради, 131, 208
 стабільні команди, 28–29, 69–71
 стабільність, створення метрик стабільності, 98
 стратегії пошуку ресурсів, удосконалення організацій, 177–178

Т

таймбокси (часові обмеження), 27
 тестування, рівні, 95
 технічна досконалість, Scrum-команди, 34
 технічний борг, 98–99
 «погашення», 102
 прозорість, 100–101
 тренди, 102, 121

У

угоди, робочі/про співпрацю, 53–54
 узгодження, створення, 130–131
 універсальний Scrum, 38
 управління цілями спринту, 82–83
 утручання Scrum-майстра, доречність, 169

Ф

фінансування,
 зміна планування та фінансування продукту, 188
 ініціативи, 185–187
 формування команд, 67
 фреймворки, 13, 15, 38
 емпіризм, 27
 командна робота, 58
 командні процеси, 30
 концепція «готовності», 75

Ц

цілі,
 Scrum-команди, 55–56
 бізнес-цілі, вимірювання досягнень, 112
 спільні, 57
 спринту, 82–83

цінність,
 визначення, 105–106
 вимірювання, 111–112
 відгуки зацікавленої сторони, 121–122
 концепція продукту, 110–111
 користувацькі історії, 117
 навчання як, 120
 образ споживача, 110
 огляди спринту, 121
 план розвитку продукту, 110
 поліпшення під час спринту, 119
 продукту, визначення, 105
 швидке постачання, 106–107

Ш

швидке постачання, 106
 швидкість удосконалення, 157

Щ

щоденна Scrum-нарада, 201
 ефективність, 84
 статусність, 208
 цілі спринту, 84

Я

якість,
 метрики, 97–98
 нарощування, 92
 самооцінювання, 200

А

Agile-маніфест, 10, 26, 48, 73, 96
 agile-мислення, 17, 48, 179, 189, 192, 195
 оцінка і, 135–136
 планування і, 128–129
 agile-трансформації, поліпшення організацій, 189–190
 ATDD (розробка на основі приймального тестування), 93

В

BDD (керована поведінкою розробка), 93

Д

DDD (предметно-орієнтоване проектування), 93
 DevOps, «виконані» природи, 96
 DoD (критерій повністю готового завдання),
 визначення, 76–77
 елемент беклогу продукту, 85–86
 переваги, 77–78
 прийом «Зараз, наступний крок, майбутнє», 80–81
 створення, 78–79
 технічний борг, 98–99

Н

HDD (заснована на гіпотезах розробка), 116

Р

PBI (елементи беклогу продукту), 53, 83, 136
 Scrum-дошки, 86–87
 гіпотези, експерименти, 116
 графік згорання завдань, 88, 91

загальні пастки, 118
карта впливу, 115
поліпшення цінності, 119
приведення до «готового» стану, 85
прозорість на пряму руху, 78
розбиття, 137
сприяння консенсусу, 65
успішність спринту, 208

S

Scrum, 195, 206

догматичний, 38
достатньо хороший, 39
емпіризм, 26, 44, 198
індивідуальні та командні можливості, 158
ітеративне та інкрементальне бюджетування, 186
каскадна модель, 39
командна робота, 28, 30–31
механічний (зомбі), 38
оптимальність, 207
особливий, 39
«повна готовність завдання», 77–78
постійне вдосконалення, 25, 36
поширені порушення, 37
поширені хибні думки, 205
та бізнес-гнучкість, 197
універсальний, 38
фреймворк, 27, 58, 75, 109, 131, 193
цінності, 76
чітка відповідальність, 57
швидке постачання, 106
як шлях уперед, 18

Scrum-дошка, 86–88

Scrum-команди, 27, 42, 51, 119

беклог продукту, 133–134
види вмінь, 32
вимірювання цінності, 111, 113
виявлення та усунення перешкод, 151
графік згоряння завдань, 88
емоційний інтелект, 50
кроки, 46
навички, 31–33
оптимізація потоку, 92
оцінка на основі обсягу, 185

планування, продуктове мислення, 126
планування скрипту, 82, 137
прозорість, 180
процеси, 30
робочі угоди, 53
самоорганізація, 55
самооцінювання, 199
технічний борг, 100
удосконалення, 147–148, 173
усунення перешкод, 156
характеристика, 71
цінність продукту, 29, 108

Scrum-майстри, 165

вимірювання успіху, 163–164
відповідальність, 161–163
втручання, 169
ефективність, 167
підхід, 166
роль, 176
самооцінювання, 202

Scrum-наради, 84, 131

самооцінювання, 201
щоденні, 208

T

TDD (керована тестами розробка), 93, 97

Переклад з англійської
ВІКТОРІЇ ЛУНЕНКО

ВИДАВНИЧИЙ ДІМ



Науково-популярне видання

Стефані Окерман, Саймон Рейндл
ОПАНУВАННЯ ПРОФЕСІЙНОГО SCRUM
Практичний посібник із подолання труднощів
і максимізації переваг Agile-підходу

Stephanie Ockerman, Simon Reindl
MASTERING PROFESSIONAL SCRUM
A Practitioner's Guide to Overcoming Challenges
and Maximizing the Benefits of Agility

Дизайн обкладинки І. І. Нестеренко

Головний редактор А. А. Клімов

Редактор Т. О. Попова

Технічний редактор Т. Г. Орел

Коректор Н. В. Красна

ФБ1332050У. Підписано до друку 17.01.2023.

Формат 70 × 100/16. Папір офсетний.

Гарнітура Minion. Друк офсетний.

Ум. друк. арк. 18,06.

ТОВ Видавничий дім «Фабула»

Свідоцтво ДК № 7005 від 12.12.2019.

вул. Котельніківська, 5, м. Харків, 61071

Для листів: вул. Космічна, 21а, м. Харків, 61145.

Надруковано у ПП «Юнісофт»

UNISOFT

61036, м.Харків, вул. Морозова, 136.

www.unisoft.ua

Свідоцтво ДК №5747 від 06.11.2017 р.

Наклад 1500 прим. Замовлення № 66/01.