

READY

WHY MOST SOFTWARE PROJECTS
FAIL AND HOW TO FIX IT



Luniel de Beer

Max Guernsey, III

Українське Видання

Поділитися цією книгою у Twitter!

Будь ласка, допоможіть Luniel de Beer та Max Guernsey, III, поширивши інформацію про цю книгу у [Twitter!](#)

Пропонований текст для твіту про цю книгу:

Я щойно придбав Ready — книгу для керівників та команд з розробки програмного забезпечення, які прагнуть позбутися доопрацювань, перенесення завдань та неузгодженості в процесі розробки. [#CodeReady](#)

Пропонований хештег для цієї книги [#CodeReady](#).

Дізнайтеся, що інші люди говорять про книгу, перейшовши за цим посиланням для пошуку цього хештегу в Twitter:

[#CodeReady](#)

Ready (Українське Видання)

Чому Більшість Програмних Проєктів Зазнають
Невдачі та Як Це Виправити

Luniel de Beer та Max Guernsey, III

Ця книга доступна за адресою <https://leanpub.com/ready-uk>

Ця версія була опублікована 2025-10-22



Це книга платформи [Leanpub](https://leanpub.com). Leanpub надає авторам та видавцям можливість використовувати процес Lean Publishing. [Lean Publishing](https://leanpub.com) — це процес публікації електронної книги в процесі написання з використанням легких інструментів та багатьох ітерацій для отримання відгуків читачів, коригування до створення потрібної книги та набуття популярності після цього.

© 2025 Luniel de Beer та Max Guernsey, III

Пам'яті Йоганна ван Аардта, який розпізнав мою пристрасть, познайомив мене зі справжнім програмуванням і допоміг мені знайти шлях до нового дому. І моїм батькам, чия непокхитна підтримка—від мого першого клієнта до першого дому в США—зробила все це можливим.

—Luniel

Моїй родині, з якою сходить і заходить сонце.

—Max

Зміст

Про цю книгу	i
Для кого ця книга	ii
Як користуватися цією книгою	iii
Про авторів	iv
Передмова	v
 Частина I: Чогось Бракує	 1
Розділ 1: Прихована проблема	2
Розділ 2: Ціна відсутніх основ	15
Розділ 3: Представлення Потoku Дозрівання Вимог (ПДВ)	27
Розділ 4: Чи це Agile?	29
 Частина II: Створення умов для готовності	 30
Розділ 5: Перше розширення	31
Розділ 6: Чому Люди Цього Не Роблять?	32
Розділ 7: Явна Робота над Готовністю (ПДВ 1)	34
Розділ 8: Ефекти ПБВ 1	36
Розділ 9: Впровадження RMF 1 на практиці	37

Частина III: Контроль завершення роботи . 39

Розділ 10: Наступна Потреба 40

Розділ 11: Що Зазвичай Роблять Люди 42

Розділ 12: Визначення поняття готовності 44

Розділ 13: Індивідуальне визначення готовності (ПДВ 2) 46

Розділ 14: Життя з RMF 1 і 2 50

Розділ 15: Встановлення RMF 2 52

Частина IV: Впровадження гейтингу 54

Розділ 16: Остання Вимога 55

Розділ 17: Передумови Визначення Готовності 57

Розділ 18: Визначення Визначення Готовності 59

Розділ 19: Індивідуальне Визначення Готовності (RMF 3) 61

Частина V: Синтез 65

Розділ 20: Більшість дедлайнів не мають значення 66

Розділ 21: Компетенція 1: Потік Дозрівання Вимог 68

Розділ 22: Як відбувається потік роботи та інформації в Скрамі з ПДВ . . . 70

Розділ 23: Вплив ПДВ 72

Розділ 24: Перехід до RMF 73

Розділ 25: Все залежить від вас 75

Частина VI: Ресурси 76

Додаток А: Скрам — не проблема 77

Додаток В: Фреймворк Synapse™ 79

Додаток С: Поширені заперечення та перешкоди щодо RMF 1	81
Додаток D: Початкові Списки Критеріїв DoD	82
Додаток E: Початкові списки критеріїв Визначення Готовності	83
Зміст	84

Про цю книгу

Ready - це книга для всіх, хто залучений у розробку програмного забезпечення та втомився від **недопостачання, постійних переробок та нечітких вимог**.

Можливо, ви вже намагалися інвестувати в навички виконання команди, покращити впровадження вашої процесної структури чи оновити код, але все ще потребуєте більшого вдосконалення.

Це тому, що головним обмеженням для більшості команд розробки програмного забезпечення є не командні навички, а зрілість вимог. **Навіть досвідчені команди з правильними навичками все ще стикаються з труднощами**, коли працюють з незрілими вимогами.

Ready представляє RMF (Процес дозрівання вимог), практичний та глибоко структурований підхід до узгодження Продукту та Розробки без заміни вашого існуючого процесу.

Незалежно від того, чи використовуєте ви Scrum, Kanban, чи щось власне, RMF допомагає вам **стабілізувати обсяг, усунути перенесення та доставляти те, що дійсно має значення**.

Якщо ваші команди відчують, що застрягли на межі “майже готово”, ця книга покаже вам, як **розірвати цикл та розблокувати вашу команду(и) назавжди**.

Для кого ця книга

Ця книга буквально для кожного, хто залучений у розробку програмного забезпечення. Від інженерів до продакт-менеджерів та від окремих співробітників до керівників.

Ця книга для вас, якщо ви залучені в розробку програмного забезпечення і помітили, що команда, з якою ви працюєте або в якій працюєте, має одну або кілька з наступних проблем:

- Робота часто переноситься з однієї ітерації в наступну
- Команди впровадження відчують, що намагаються влучити в рухомі цілі
- Робота залишається відкритою надто довго
- Робота позначається як завершена, але насправді не виконана
- Виконана робота не відповідає очікуванням
- Робота регулярно породжує велику кількість дефектів
- Велику кількість роботи доводиться регулярно переробляти

Якщо будь-які з цих проблем здаються знайомими, *Ready* може допомогти.

Як користуватися цією книгою

Ця книга була розроблена, щоб бути практичною. Це не теоретичний трактат чи стратегічна презентація - це практичний посібник з впровадження RMF (Процесу дозрівання вимог), заснований на реальній роботі з клієнтами та перевірений під реальним тиском доставки.

Розділи написані послідовно, але сам RMF є модульним. Він складається з трьох фундаментальних практик:

- RMF 1: Співпраця для спільного розуміння
- RMF 2: Контроль завершення роботи за допомогою Індивідуальних визначень завершеності
- RMF 3: Контроль впровадження за допомогою Індивідуальних визначень готовності

Кожна частина, або “Звичка”, як ми їх називаємо, є самостійною, але вони будуються одна на одній. Книга розроблена, щоб допомогти вам підходити до них по одній, по порядку. Ця структура відображає те, як ми рекомендуємо командам впроваджувати RMF на практиці - кожна Звичка додається лише після того, як попередня працює.

Це допомагає уникнути перевантаження команд і дає кожній зміні найкращий шанс закріпитися. Ви дізнаєтеся більше про те, як це зробити, починаючи з [Розділ 9](#).

Якщо ви шукаєте допомоги - будь то порада, коучинг чи хтось для розмови з вашою командою керівництва - не соромтеся звертатися до нас напяму.

І якщо ви шукаєте формальну підтримку для впровадження RMF, Prodcore пропонує повну серію Програм, розроблених для поетапного керівництва впровадженням. Ви можете дізнатися більше на <https://ready-book.link/rmf>.

Про авторів

Луніель де Бір є творцем Процесу дозрівання вимог (RMF), практичної системи для усунення розривів між продуктовим наміром та інженерним виконанням. Він має понад 15 років досвіду керування Агільними трансформаціями, поєднання продукту та розробки, та допомоги командам доставляти з ясністю та впевненістю.

Луніель також створив систему управління можливостями Productcore, відстежуваний та масштабований підхід до моделювання можливостей продукту. Він запропонував Розробку на основі РКВ (PKBDD), систему з контролем версій для управління постійними вимогами до продукту. Ці інструменти є частиною більшої структури доставки, розробленої в Productcore.

Макс Гернсі III - це архітектор програмного забезпечення, викладач та співзасновник Productcore, консалтингової компанії, присвяченої виправленню збоїв доставки через структурну та технічну точність. Маючи понад два десятиліття досвіду в об'єктно-орієнтованому проектуванні, рефакторингу, розробці через тестування та шаблонах проектування, він як доставляв критично важливі системи, так і навчав інженерні команди в масштабі. Його робота поєднує глибокі технічні практики з поведінковою та процесною трансформацією, щоб допомогти організаціям досягти стійкої досконалості в доставці.

Мах зробив значний внесок у PKBDD та очолив розробку підходу Productcore до розробки через поведінку (BDD) завдяки своїй глибокій експертизі в поведінковій специфікації.

Разом їхня робота об'єднує чіткість, відстежуваність та гейтинг у цілісну систему постачання програмного забезпечення, яка масштабується від командної практики до організаційної спроможності.

Передмова

Примітка для технічних керівників

Якщо ви старший керівник в інженерній організації, у вас, ймовірно, немає нестачі в зусиллях, дисципліні чи розумних людях. І все ж, чомусь проекти все одно зупиняються. Цілі не досягаються. Очікування не виправдовуються. Не тому, що ваші команди ліниві, а тому, що щось фундаментальне зламане в тому, як робота визначається, формується та доставляється.

Ця книга – не посібник з лідерства. Це інструмент для структурної діагностики. Вона розкриває, що насправді відбувається всередині ваших команд – чому “майже готово” постійно перетворюється на “не готово”, і чому локальний прогрес так рідко трансформується в стратегічні результати.

Можливо, ви не побачите *себе* на цих сторінках. Але якщо ваші команди не можуть доставити те, що вам потрібно, ви побачите їх. І коли це станеться, у вас нарешті з’явиться мова – і система – щоб це виправити.

Від Луніеля

Перш за все, ця книга не була б можливою без Макса, чия здатність бачити крізь туман і полову та виділяти суть ідеї просто вражає мене.

Як ми сюди потрапили?

Якщо озирнутися назад, я думаю, це тому, що я завжди хотів зрозуміти, як речі працюють насправді. Чи то релігія, харчування чи розробка програмного забезпечення, я постійно стикався з однією і тією ж проблемою: поверхневі відповіді, які не витримували тиску. Тому я продовжував копати – запитуючи не лише про те, що ми робимо, але й чому, і чого не вистачає, коли щось не працює.

Одна з найперших тріщин у системі з’явилася в ролі, де я носив три капелюхи: Скрам-майстер, Власник продукту та Керівник розробки (!!) для команди, що надавала сервіси даних у відомій технологічній компанії. Ми робили все, що казав Скрам

– короткі Спринти, історії в беклозі, планування за півдня – але щоразу, коли ми починали новий Спринт, ми стикалися з тертям. Команда не повністю розуміла проблему, нам доводилося переглядати та редагувати вимоги посеред Спринту, виникали залежності, яких можна було уникнути, і вони нас затримували, а ключові кроки пропускалися.

Тому я почав робити щось інакше. Я збирав команду та стейкхолдерів у кімнаті для кожної історії, детально розбирав проблему, разом обмірковував рішення, і лише потім писав історію. Планування Спринту скоротилося до години, а наш успіх у доставці злетів до небес.

Поступово я почав розуміти, що успіх не приходить від того, що ти працюєш старанніше всередині Спринту. Він приходить від структури, яку ти встановлюєш до його початку.

Пізніше, почувши, як Джефф Сазерленд говорить про “Визначення готовності”, я знав, що там є щось цінне – але цього було недостатньо. Мій досвід з вимогами, UX, UI, дослідженнями, а пізніше з BDD показав мені, що різні робочі елементи вимагають різних видів готовності. Деяким потрібні специфікації поведінки. Деяким потрібен доступ до системи. Деяким потрібне повне відстеження можливостей.

І всім їм потрібне спільне розуміння, яке реально підтверджене – а не припущене.

Працюючи з більшою кількістю команд, я бачив однаковий патерн всюди: пропущені кроки, невиконані залежності, команди, які роблять все можливе, але постійно борються з проблемами, яких можна було б уникнути. Навіть чудові команди стикалися з труднощами – не тому, що вони були слабкими, а тому, що їм бракувало структури, яка робила б готовність явною.

Результатом всього цього навчання, ітерацій та розчарувань є структурована система управління готовністю.

Про це і йдеться в цій книзі.

Я сподіваюся, що вона дасть вам ясність щодо того, де лежать справжні проблеми, і як їх виправити. Я сподіваюся, що вона дасть вам мову для захисту практик, які можуть здаватися “зайвими”, але насправді є необхідними. І найголовніше, я сподіваюся, що вона допоможе командам доставляти з меншим стресом, меншою кількістю сюрпризів і набагато кращими результатами.

Якщо ми зробимо це правильно, ми заощадимо індустрії мільярди доларів.

Але що важливіше, ми повернемо людям їхнє психічне здоров’я.

Від Макса

Я працював над цією проблемою з різних кутів протягом десятиліть, але мій прогрес був обмежений, доки я не зустрів Лунієля.

Це тому, що до знайомства з ним я підходив до проблеми, в основному, як до технічної. Я зосереджувався на допомозі командам у впровадженні таких речей як Розробка через тестування (TDD), рефакторинг, передовий дизайн програмного забезпечення і, пізніше, Розробка через приймальне тестування (ATDD) або Розробка через поведінку (BDD).

У більшості цих випадків проблема, описана в цій книзі, розглядалася як деталь реалізації при встановленні цих технічних практик.

Це не означає, що я більше не ціную технічні практики. Я все ще вважаю їх глибоко важливими, але вони не вирішують безпосередньо проблему готовності в розробці програмного забезпечення. Натомість, вони *виявляють* цю проблему, і люди накладають латку на свій процес, щоб вирішити її “достатньою мірою” для підтримки технічних практик, які вони намагаються впровадити.

Я також хочу звернутися до питання про те, кому може допомогти ця книга. Коротка відповідь – “ймовірно, майже всім у розробці програмного забезпечення”, але справжня відповідь містить нюанси, які допомагають адаптувати її до різних середовищ без зміни основного змісту.

Існують команди, яким потрібне рішення, представлене в цій книзі. Ви познайомитеся зі спрощеною версією однієї з них у Розділі 1.

Є також команди, які не мають прямої **потреби** в системі, яку ми пропонуємо, але все ж могли б отримати від неї користь.

Найкраща команда, з якою я коли-небудь працював – легко на цілу стандартну одиницю відхилення вище за **наступну найкращу команду**, якщо не на дві – розташовувалась у віддалених районах Central Oregon. Вони були настільки високопродуктивними, що могли подолати відсутність такої системи просто завдяки рівню своєї компетентності. Проте, мій керівник на той час, Tom Barreras, якось сказав мені щось на кшталт “Я помітив, що наші історії проходять краще, коли ми спочатку витрачаємо певний час на обговорення тестів.”

Це, знову ж таки, було тим, що я тоді розглядав через призму тестування-розробки та технічного виконання, але тепер я розумію, що це був ще один показник того, що *готовність* була фактором, який впливав на команду... та конкретна команда

була настільки здібною і швидко реагуючою, що могла досягати успіху, реагуючи на перешкоди по мірі їх виникнення, замість того, щоб запобігати їм з самого початку.

Навіть якщо ви з тих людей, кому не обов'язково *потрібно* турбуватися про готовність, тому що ви можете подолати це, або ви працюєте з командою такого ж типу, ви все одно можете отримати користь від змісту цієї книги.

Частина I: Чогось Бракує

*Коли робити те саме краще не допомагає, шукайте те, що **не робиться**.*

Розділ 1: Прихована проблема

Це справжня¹ історія про банк. Назвемо його просто “Банк”. Це тип Федеративної кредитної установи, що є частиною національної фінансової інфраструктури Сполучених Штатів.

Нас (Луніель та Макс) запросили до Банку, тому що у них виникли труднощі з реалізацією програмного проєкту. Це було одне з найбільш дисфункціональних середовищ, які ми коли-небудь бачили, і саме тому ми обрали це як початкове тематичне дослідження: якщо значущі зміни були можливі в Банку, вони можливі будь-де.

Коротка примітка про проєкти

Коли ми використовуємо термін “проєкт” у цій книзі, ми маємо на увазі його в контексті управління проєктами. Хоча існують різні думки щодо значення цього слова, ми використовуємо визначення від [Project Management Institute](#):

“Проєкт - це **тимчасове** починання, що здійснюється для створення унікального продукту, послуги чи результату.”

Це означає, що проєкт має визначений початок і кінець. Коли проєкт закривається, проєктні знання та артефакти архівуються, члени команди звільняються, а контракти завершуються.

У цій книзі проєкт фундаментально стосується виконання. Більшість проєктів, як визначено РМІ, починаються з техніко-економічного обґрунтування або проєктування. Бачення та стратегія вже визначені до моменту ініціації проєкту.

Проєкт народжується з цього бачення та стратегії, і його успіх чи невдача залежать від того, чи реалізовані передбачені цілі—*а не* від того, чи були ці цілі правильними.

Можливо, ви використовуєте слово “проєкт” по-іншому, і це нормально. Просто

¹Ми змінили ідентифікуючі деталі для захисту конфіденційності людей та установ, з якими це сталося.

знайте, що коли ми його використовуємо, ми посилаємося на визначення та контекст, наведені вище.

Нічого з цього не означає, що ми схвалюємо використання управління проектами для розробки програмного забезпечення. Зовсім навпаки. Але ми визнаємо, що воно *все ж* використовується. Ми розглянемо цю проблему пізніше в [Розділ 6](#).

1.1: Класичне переписування

Банк переписував свій портал погашення позик з кількох причин.

Стара система, повністю побудована на C#/.NET, була з багами. Окрім зниження задоволеності клієнтів, вона також генерувала нескінченний потік дуже дорогих інцидентів підтримки, коли комусь доводилося вручну маніпулювати базою даних для виправлення помилок, зроблених системою.

Стара система також була занедбаною з точки зору підтримки. Інженерам було майже неможливо вносити значущі зміни, а навіть коли вони могли це зробити, це було надзвичайно ризиковано.

Переписування мало все це змінити.

Нова система все ще мала мати бекенд на C#/.NET, але він мав бути повністю покритий тестами. Фронтенд планувалося реалізувати в OutSystems, популярному рішенні з низьким рівнем програмування або без нього, яке дозволяє організації визначити додаток в одному місці та автоматично отримувати веб-додаток, додаток для Android та iOS щоразу, коли вони вирішують опублікувати свої зміни.

Сподівалися, що нова система буде без багів, що покращить задоволеність клієнтів і значно знизить витрати на підтримку.

Вони також сподівалися, що переписування розблокує розробників—поєднання більш дисциплінованого підходу до бекенду та підходу з низьким рівнем програмування до фронтенду значно зменшить вартість та ризики впровадження нових функцій.

Приємним побічним ефектом переходу на OutSystems було те, що вони отримали б чистий, сучасний мобільний додаток на обох основних платформах.

Такою була мрія, коли вони почали три роки тому до початку цієї історії. Реальність була такою, що досі команди не випустили **нічого**.

1.2: Погляди на проблему

Коли ми розмовляли з командою виконавчого керівництва, ми чули цілком природне розчарування тим, що вони зробили такі великі інвестиції без жодного стратегічного руху.

З їхньої точки зору, вони спробували все. Вони змінювали персонал, збільшували штат, змінювали бюджет, посилювали тиск і залучали парад консультантів (серед яких, як було наполегливо натякнуто, ми були останніми). Ніщо, здавалося, не покращувало ситуацію—принаймні не так, щоб вони могли це виміряти, тому що все, що вони бачили, це те, що “стрілка” була на нулі в одному кварталі, а потім все ще залишалася на нулі в наступному.

Вони не хотіли більше “невидимого прогресу”. Вони хотіли *результатів*.

Коли ми розмовляли з керівництвом Продуктової організації, ми отримали трохи іншу (але все ж подібну) історію, оскільки вони працювали більш безпосередньо з Інженерним відділом.

Не те щоб команди нічого не робили, принаймні на їхню думку. Справа в тому, що команди ніколи не робили те, про що їх просили. Це було практично гарантовано: незалежно від того, наскільки простим було завдання і наскільки чітко воно було сформульовано, під час оцінки результатів роботи команд ви отримували щось зовсім інше.

Дійшло до того, що ходив жарт на кшталт “Нам треба зрозуміти, як просити про те, чого ми не хочемо, щоб був хоч якийсь шанс отримати те, що нам потрібно”.

Керівники з розробки бачили ситуацію зовсім інакше.

На їхню думку, проблема полягала в тому, що Продукт не надавав придатних до виконання вимог і що Продукт не надавав *достатньо* вимог. Якби Продукт просто “включився в роботу”, команди змогли б вчасно та в межах бюджету надавати те, що від них хочуть.

Вони зробили серйозні інвестиції в модернізацію процесів написання та доставки коду, і, на їхню думку, Продукт не надавав чітких вимог.

Коли ми спілкувалися з іншими консультантами (які порекомендували нас організації), вони правильно визначили дисфункцію, яку спостерігали: всі, здавалося, були зосереджені на тому, щоб звинуватити когось іншого. Причина, з якої нас спочатку запросили, полягала в тому, що вони переймалися стратегією

укомплектування персоналом і хотіли оцінити індивідуальних виконавців, але вони вважали, що взаємні звинувачення та менторський підхід на рівні керівництва були головним джерелом проблем.

1.3: Наше розслідування

Наше початкове завдання полягало в оцінці команд і спробі допомогти їм покращити навички за потреби, тож ми почали вивчати людей, які працювали на передовій.

Безумовно, було куди рости.

Індивідуальним виконавцям з боку Продукту справді бракувало необхідних навичок. Насправді, більшість із них були проєктними менеджерами, яких призначили на роль Власника Продукту (ВП) або Продукт-менеджера.

Як наслідок, половина з них писала “розпливчасті” вимоги, а потім приймала (буквально) все, що команди робили в тій ітерації, без будь-якого критичного аналізу. Інша половина писала такі ж вимоги, а потім заявляла, що команди “мали б знати” речі, про які ніколи не говорили, і тримала робочі елементи відкритими практично безстроково.

Банк використовував для управління та відстеження свого беклогу роботи. Хоча те, що ми розглядаємо в цій книзі, здебільшого не пов’язане зі Скрамом, ми використовуємо термінологію Скраму, оскільки більшість – або принаймні значна частина – команд використовує Скрам.



Визначення: Скрам

Скрам – це легкий фреймворк, який допомагає людям, командам та організаціям створювати цінність через адаптивні рішення для складних проблем. Коротко кажучи:

1. Власник Продукту впорядковує роботу для складної проблеми в Беклозі Продукту
2. Скрам-команда перетворює вибрану частину роботи на інкремент цінності протягом Спринту
3. Скрам-команда та зацікавлені сторони перевіряють результати та коригують їх для наступного Спринту
4. Повторюють

Якщо ви не знайомі зі Скрамом та його термінологією, рекомендуємо ознайомитися з версією [Скрам Гайду](#) 2020 року. Це швидке та пізнавальне читання.

Так само ми виявили, що технічні команди були значно нижче типового рівня за навичками програмування (в кращому випадку -2σ) і на додачу дуже опиралися змінам. Як природний наслідок, якість коду була жахливою.

Проте, на думку команд, це не було причиною того, чому вони не виконували поставлені завдання. Для них проблемою були нечіткі вимоги та зміни посеред спринту з боку Продукту, які вбивали проект.

...і ніхто навіть не говорив про більшу проблему, настільки абсурдну, що здається вигаданою, доки ви з нею не зіткнетеся.

У інженерних команд була звичка не розуміти вимогу, будувати щось навімання, а потім вимагати визнання за те, що вони “завершили робочий елемент”.

Ми не маємо на увазі дрібне нерозуміння. Ми маємо на увазі повний розрив: Ми просили їх вимкнути зарахування коштів на основний борг за певних обставин, а вони натомість вимикали можливість додавати другу адресу електронної пошти для підтвердження.

А потім вони казали нам, що саме це ми й просили.

1.4: Глибше копання

Обидва пробіли в навичках можна було б усунути, але ми сумнівалися, що саме вони були справжніми перешкодами.

Було щось іще не так, тому ми копнули глибше. Ми почали з такого питання: Чому написання вимог займало так багато часу і давало такі погані результати?

Одна з причин полягала в тому, що знання, необхідні для написання змістовних вимог, були в дуже обмеженій кількості. Невелика їх частина була у команд з розробки та Продукту.

Частина була вбудована в код успадкованої системи. Частина повністю зникла. Більшість же зберігалася як неформалізовані знання експертів предметної області, розкиданих по різних підрозділах банку. Це означає, що створення вимоги, яка дійсно сприяла б досягненню стратегічної мети, було надзвичайно трудомістким і тривалим процесом.

На противагу цьому був невгамовний апетит до функціоналу з боку керівної команди, якій бракувало можливостей. Мандат був “тримайте інженерів зайнятими—завалюйте їх вимогами”. Основна увага приділялася кількості вимог, щоб команди були постійно зайняті—підхід, що суперечить ретельності та часу, необхідним для визначення вимоги, яка дійсно “зрушить справу з місця”.

1.5: Покращення Ситуації Не Покращило Ситуацію

Усі ці проблеми можна було вирішити, проте їх вирішення не допомогло.

Попередні вдосконалення методів розробки програмного забезпечення не допомогли командам досягти результату, але Макс доклав зусиль, щоб допомогти командам покращитися ще більше.

Він запровадив революційні концепції з програмістських доктрин середини двадцятого століття на кшталт “не копіюйте цей код 27² разів, помістіть його у функцію і викликайте її”. Лише ця порада суттєво покращила якість нового коду і дозволила їм почати покращувати якість.

Ця та інші базові поради щодо кодування допомогли їм писати кращий код, який вони могли легше підтримувати в майбутньому.

²Це не лише не гіпербола, це навіть не найгірший випадок. В одному випадку було майже сто точних дублікатів одного й того ж алгоритму.

...але це не допомогло просунути проєкт вперед.

З боку Продукту, Луніель зміг запровадити BDD і забезпечити ретельну перевірку вимог Власниками Продукту перед їх передачею командам.

Він змусив команди співпрацювати над ними та використовувати їх для оцінки того, чи дійсно Елемент Беклогу Продукту (PBI) завершено.



Визначення: Елемент Беклогу Продукту (PBI)

Елемент Беклогу Продукту (PBI) – це дискретна одиниця роботи в беклозі продукту, яка представляє потенційну зміну, доповнення або покращення продукту. PBI можуть мати різні форми—функціонал, виправлення помилок, технічне вдосконалення, дослідницьке завдання тощо—і визначаються їхнім внеском у цінність продукту.

Багато команд називають PBI “історіями” або “користувацькими історіями”, але правильний термін в Scrum – це “Елемент Беклогу Продукту” або “PBI”. Коли PBI включається в Спринт, він також стає частиною Беклогу Спринту. Для простоти та нейтральності ми використовуємо “Елемент Беклогу Продукту” або “PBI” для позначення будь-якого робочого елемента, яким керує Scrum Команда—незалежно від того, як ви його називаєте: Елемент Беклогу Продукту, Елемент Беклогу Спринту (SBI), користувацька історія, історія, робочий елемент чи елемент беклогу.

Це додало *ясності*, але не створило *поток*.

За допомогою партнерів-консультантів, які нас залучили, ми змогли (тимчасово) послабити абсолютно нищівний тиск на команди та авторів вимог.

Це могло допомогти створити трохи довіри, але не дало жодних відчутних результатів.

Ми навіть почали створювати базу знань, яка допомагала людям відстежувати бізнес-знання, необхідні для написання вимог, та визначати місця, де були прогалини в цих знаннях.

Це прискорило написання вимог, але не допомогло випустити продукт.

Після місяців взаємодії ми допомогли керівництву *побачити*, де вони знаходяться, але вони були далекі від своїх цілей. І вони не наближалися до них.

Вони готувалися повернутися до своєї старої стратегії “завантаження” команд, щоб переконатися, що вони завжди зайняті.

1.6: Процес Виключення

Було б легко просто розвести руками і сказати “це безнадійно”. Можна було знайти безліч виправдань:

- Інженерна команда була низькокваліфікованою (це правда)
- Автори вимог мали неправильний набір навичок (це так)
- Керівництво душить команди нереалістичними очікуваннями (це факт)
- В організації не вистачало критично важливих бізнес-знань, необхідних для функціонування (це правда)
- Схоже, що керівники не довіряють один одному (це так)

Все це було правдою. Проте покращення були зроблені в усіх цих областях, і жодне з них, здавалося, не покращило фундаментальну проблему. Жодне з них не допомогло інженерним командам, Продукту, менеджменту чи керівництву наблизитися до їхніх цілей.

...і саме в цьому криється натяк на рішення: Усі перераховані вище проблеми були тими проблемами, які люди вже могли бачити.



Якщо змінні, які ви бачите, не мають значення, має бути змінна, яку ви **не бачите**, яка має значення.

Справжньою проблемою було те, про існування чого ніхто навіть не здогадувався.

1.7: Пошук справжнього винуватця

У пошуках справжнього винуватця – того, що насправді заважало банку досягти своїх цілей – нам потрібно було з чогось почати.

Одним із розумних місць для початку було подивитися, що спільного мали всі невдачі РВІ (а таких була більшість).

Ми почали з виключення речей, які явно не були спільними, оскільки вони сильно відрізнялися:

- Яка частина системи: деякі РВІ стосувалися лише бекенду, інші лише фронтенду, а деякі зачіпали обидві частини
- Яка команда виконувала роботу – здавалося, не мало значення, хто виконував роботу, ймовірність невдачі була високою
- Який РО створював роботу – те саме, що й з командами

Потім ми почали дивитися на те, що було спільним. Список не був довгим, але й не коротким:

- Інженерні команди
- Власники продукту
- Керівництво
- Культура
- Середовище розробки
- Інженерні практики
- Техніка написання вимог
- Домен (фінанси)
- Залежні сервіси

Багато з цих факторів також можна було відкинути відразу. Команди, власники продукту, керівництво, культура та середовище розробки були нещодавно покращені без реального впливу на значущий результат. Ми особисто допомогли покращити інженерні практики та техніки написання вимог і підтвердили, що ці покращення закріпилися, але це все одно не допомагало.

Навряд чи можна звинувачувати домен. Фінанси – це один із найстаріших видів розрахунків в історії людства. Це надзвичайно зріла галузь. Крім того, інші банки успішно розгортали програмне забезпечення, що наочно спростовувало (очевидно натягнуто) гіпотезу про те, що банки просто не можуть цього робити.

Залежні сервіси також не можна було звинувачувати, оскільки з ними виникали такі ж проблеми зі змінами, як і з ініціативою, яку ми розглядали...

...але це змусило нас замислитися: що, якщо ми почнемо аналізувати причини невдач?

1.8: Розбір причин невдач

Один РВІ не зміг просунути продукт вперед, тому що команда, як це часто траплялося, зробила щось абсолютно випадкове і майже повністю не пов'язане з запитом. Це, очевидно, є ознакою того, що вони не зрозуміли робочий елемент. Тому розуміння було серйозним кандидатом, хоча ми вже певною мірою працювали над цим, допомагаючи їм впровадити BDD.

Інший РВІ не вдалося закрити через неправильні розрахунки. Це ще один доказ того, що розуміння могло бути основною проблемою.

Третій робочий елемент, який ми проаналізували, не був належним чином проконтрольований Власником продукту – вона автоматично схвалила його, коли команда сказала, що час закривати. Це дещо суперечило нашій гіпотезі, але все ще можна було стверджувати, що вона не розуміла, як цей робочий елемент вписується у план вищого рівня.

Можливо. Якщо дуже сильно примружитися, дивлячись на це таким чином.

Потім ми натрапили на РВІ, який зовсім не відповідав схемі. Здавалося, що команда розуміла – хоча немає способу перевірити, чи дійсно це так. Але це не мало значення: у них навіть не було шансу досягти успіху чи зазнати невдачі самостійно, оскільки вони зіткнулися із залежністю, яку потрібно було оновити, і їм довелося відкласти свою роботу на кілька Спринтів.

Навіть якщо вони *не* розуміли, що вони мали робити, у них все одно не було шансів з цим елементом беклогу, тому розуміння в цьому випадку **не було** проблемою.

Один виняток, звісно, не спростовує конкретну першопричину, але це збудило нашу цікавість. Ми почали шукати інші спростовуючі докази.

І ми їх знайшли. Були робочі елементи, які:

- Зазнали невдачі, тому що команда знала, що не розуміє, але ніхто не міг знайти експерта предметної області для вирішення проблеми
- Змінилися на гірший досвід користувача як обхідний шлях через те, як функціонують вищестоящі сервіси
- Довелося відкласти, тому що вищестоящі залежності не були готові
- Не могли бути завершені, тому що тестувальники не встигли зібрати тестові дані

- Були закриті, але їх довелося переробляти, тому що сам запит був неправильним
- Зазнали невдачі, тому що команда не усвідомлювала, наскільки складним вже був існуючий код
- Просто не були оцінені
- Були суттєво недооцінені³
- Змінилися посеред Спринту, тому що РО нарешті отримав необхідні знання про домен
- Здавалося, що змінилися після Спринту (з точки зору команди), тому що РО і команда так і не дійшли згоди щодо того, що це означає

Список можна продовжувати, але для цієї історії достатньо.

Можна стверджувати, що кожен з цих випадків певним чином пов'язаний з “розумінням”–і, безумовно, відсутність розуміння була *залучена* до багатьох з них–але це не означає, що відсутність розуміння була причиною... особливо враховуючи, що ми працювали над спільним розумінням, і це особливо не допомогло.

Тоді нас осяяло. Бракувало більш фундаментальної складової. У випадках, де було задіяне погане розуміння, це була лише безпосередня причина.

Першопричина була набагато ширшою.

1.9: Точка перетину

Єдине, що об'єднувало всі PBI, які ми проаналізували на предмет режиму відмови, було те, що всі вони були **розпочаті передчасно**.

Коли робочий елемент зазнає невдачі через те, що команда знала про нерозуміння проблеми і не могла знайти експерта, який би допоміг їм зрозуміти, це означає, що команда почала роботу над PBI, знаючи, що вони не розуміють проблему.

Коли елемент беклогу має змінитися на гірший досвід через те, як функціонує вищестояща служба, це означає, що робота була розпочата без реального розуміння впливу вищестоящої служби.

Відтермінування через те, що вищестояща залежність не була готова наприкінці, означає, що не була гарантія її готовності на початку.

³Ми маємо на увазі не лише те, що вони помилилися. Здавалося, що команди просто вставили значення “3” у всі поля оцінки для низки PBI.

...і так було з усіма іншими випадками: Тестувальники не мали готових даних або не знали, як їх отримати до початку роботи над РВІ, вимоги не були належно перевірені перед передачею команді, код не був досліджений перед початком роботи, оцінка була недостатньою або взагалі не проводилася, бракувало знань предметної області, і, звісно, спільне розуміння не було перевірено.

Проблема, як виявилось, полягала в тому, що реалізація розпочиналася над робочими елементами до того, як ці елементи були *готові*.



З нашого досвіду, більшість робочих елементів, які не вдається реалізувати, зазнають невдачі через те, що вони **не були готові** на момент початку реалізації.

Тож ми взялися допомогти Банку з цим.

У цей момент ви можете подумати, що достатньо просто сказати, що РВІ повинні бути готові. Однак виявляється, що це не так просто впровадити. “Купуй дешево, продавай дорого” - така ж проста ідея.

Бракує ще одного елемента, необхідного для втілення цієї хорошої поради в практику.

1.10: Великий поворот

Ми знайшли відсутній елемент головоломки, який розблокував їх, що, у свою чергу, розблокувало ініціативу.

Коли ми завершили цей проєкт, інженери все ще були значно нижче середнього рівня за навичками. РО все ще не мали потрібних навичок. Культура все ще не була виправлена...

Проте продукт нарешті почав рухатися вперед і врешті-решт вийшов у світ.

До кінця цієї книги ви дізнаєтесь, що це за відсутній елемент і що потрібно для його впровадження. І ви зможете розблокувати організації, які здаються стриманими невидимою стіною.

Коротка примітка щодо обсягу

Ця книга присвячена дуже конкретній і поширеній проблемі: відсутності структури, чіткості та зрілості при передачі між бізнесом та розробкою. Вона припускає, що щось було обрано для реалізації, і зосереджується на забезпеченні того, щоб робота з побудови відбувалася зі спільним розумінням, готовністю та відстежуваним завершенням.

Методики в цій книзі не підказують вам, що будувати, чому це будувати або як дізнатися, чи це правильна річ для побудови. Якщо у вашій організації відсутнє справжнє управління продуктом або змістовні цикли зворотного зв'язку, ми не намагаємося вирішити це тут. Натомість ми пропонуємо спосіб зробити ці прогалини більш помітними та зменшити витрати на виявлення того, що ви помилялися.

При використанні в правильному контексті, це рішення приносить потік, безпеку та чіткість. Але як і будь-яка система, воно може бути неправильно застосоване—особливо коли використовується ізольовано або без усвідомлення.

Розділ 2: Ціна відсутніх основ

Варто приділити трохи часу тому, *наскільки* серйозною може бути ця проблема для деяких організацій.

Ми виявили три основні “категорії” проблем, які створює відсутність готовності:

- Відсутнє або неповне спільне розуміння між Продуктом та Розробкою та всередині них
- Відсутність контролю над тим, що є справжньою метою РВІ та коли він дійсно завершений
- Відсутність контролю того, коли робочий елемент може перейти до фази впровадження

Крім того, ми помітили, що змінити ці речі може бути досить складно. Це логічно: зміни даються нелегко.

Старі звички важко викорінити, а нові важко прищепити. З нашого досвіду консультантів, ми помітили, що людям *надзвичайно* легко повертатися до старих звичок і порівняно важко формувати нові.

Тому потрібно мати механізм, який закріплює нові звички та перешкоджає поверненню до старих.

Щоб вирішити це, ми вважаємо, що бракує ще одного компонента — підзвітності та відстежуваності.

2.1: Пазл без картинки на коробці

Ви коли-небудь намагалися зібрати пазл без картинки на коробці? Це можливо, але повільніше, більш frustrating, і сповнено хибних спроб.

Ви робите прогрес, потім розбираєте все. Ви сумніваєтеся, що куди підходить. Ви думаєте, що працюєте над однією картинкою, поки не розумієте, що це не так.

Ось на що часто схожа розробка програмного забезпечення.

Беклог заповнений. Спринт триває. Всі старанно працюють.

Але без спільного бачення того, що ми будемо, узгодженість стає удачею, а не системою.

Без чіткості навіть найкращі команди відчувають frustration, вигорання та відчуття, що їхні зусилля не цінуються.

2.2: Стара приказка та сувора реальність

Є причина, чому так багато практик написання вимог, навіть деякі інженерні практики, сильно зосереджені на створенні спільного розуміння між замовниками та командами впровадження. Ніщо не сіє такий хаос, як інженери, які не знають, що саме вони мають будувати.



“Сміття на вході — сміття на виході” — це приказка, а не банальність.

Англійська мова сповнена самопротиріч...

- Ви можете санкціонувати чийсь дії. Можливо, це означає, що ви дали їм дозвіл заздалегідь, або, можливо, це означає, що ви висловлюєте осуд після факту.
- Ви можете злегка припудрити щось, але якщо це кредеса, це означає, що ви видаляєте пил з неї, тоді як у випадку з бен'є ви додаєте пудру.
- Якщо ви підтримуєте команду, ви можете бути причиною того, що команда здатна продовжувати функціонувати, або ви можете бути причиною того, що вони не можуть рухатися вперед.

Автоантоніми можуть бути найяскравішими прикладами, але вони лише один вид неоднозначності. Деякі слова не тільки є власними протилежностями, але й мають багато додаткових потенційно заплутаних альтернативних значень.

“У цьому кліпі він відрізав купон з газети та прикріпив його до паперу на планшеті разом з іншими вирізками, поки кліпер на задньому плані рухався з непоганою швидкістю.”

І це стосується не лише англійської мови. *Усі* природні мови, про які нам відомо, мають цю властивість.

І все ж, це єдині мови, якими ми можемо користуватися при визначенні вимог.

Як наслідок, коли інженерна команда не *підтвердила*, що їхнє розуміння вимоги збігається з розумінням замовника, ця команда покладається на удачу. Тобто, найкращий можливий результат — це те, що вони обрали правильну інтерпретацію, і замовник не змінить свою думку по ходу справи.

Цей результат далеко не гарантований.

2.3: Деякі поширені наслідки

Без підтвердженого спільного розуміння команди стикаються з низкою ризиків.

Загалом, найбільш болючий результат — це коли команда просто будує неправильну річ.

Час, коли вони про це дізнаються, може контролюватися різними атрибутами їхнього процесу. Наприклад, здорове впровадження Скраму може виявити такого роду непорозуміння дуже рано під час виконання, тоді як процес Водоспаду має дуже високі шанси відкласти таке відкриття на місяці.



Визначення: Водоспад

Послідовна модель розробки програмного забезпечення, яка набула поширення наприкінці 20-го століття. Вона була вперше зображена в [статті Вінстона В. Ройса 1970 року](#), яка ілюструвала розробку як серію каскадних кроків — вимоги, дизайн, впровадження, тестування тощо — кожен з яких переходить у наступний, як водоспад. Хоча Ройс представив модель як приклад того, чого *не слід* робити, індустрія прийняла її як blueprint для великомасштабної розробки.

Водоспад також відомий групуванням подібної роботи у великі, послідовні фази — характеристика, відома як **розробка великими партіями**. Таке групування практично гарантує **пізнє навчання**: команди не отримують зворотного зв'язку щодо ранніх рішень до набагато пізнішого етапу процесу. Помилки, виявлені пізно, дорожче виправляти. Практики Agile критикують Водоспад саме з цієї причини, віддаючи перевагу меншим, ітеративним циклам, які дозволяють раніше виявляти проблеми та коригувати курс.

Тим не менш, рано чи пізно команда, яка працює з “неправильним” (точніше, відмінним) розумінням вимог, зіткнеться з “правильною” (також відмінною, насправді) вимогою. Знову ж таки, від здоров’я організації залежить, якої форми набуде це зіткнення та який вплив воно матиме, але воно майже завжди відбувається.

У більшості випадків це призводить до певної форми переробки. Замовник (зазвичай відділ управління продуктом) змушений буде просити про зміни, щоб привести те, що створили команди розробки, у відповідність до того, чого він насправді хотів.

Інший дуже поширений прояв полягає в тому, що замовник продовжує вимагати від команди відповідності своєму початковому розумінню того, що він просив.

Команди можуть легко сприйняти це як зміну думки продуктового менеджера. Ще гірше, це може фактично *спонукати* зацікавлені сторони набуті звички змінювати свою думку — затримувати робочі елементи, доки все не стане *саме так*, виснажувати команди та приховувати прогрес від вищого керівництва.

2.4: Коли пазл вважається завершеним?

Повертаючись до нашої аналогії зі складанням пазла, подумайте над цим питанням: Що означає завершити пазл?

Наївний складальник пазлів, такий як Макс, просто сказав би “всі частини приєднані до правильних сусідніх елементів зображенням догори.”

Проте досвідчений складальник пазлів, такий як Луніель, знає, що є ще багато нюансів.

Можливо, ви просто складаєте пазл заради розваги. Ви його завершите, трохи помилуетесь, а потім розберете і покладете назад у коробку.

З іншого боку, можливо, ви хочете його оформити в рамку і повісити на стіну. У такому випадку потрібно зробити додаткові речі:

1. Покласти його на художню основу
2. Транспортувати його до майстра з обрамлення
3. Дочекатися завершення обрамлення
4. Транспортувати його назад до місця монтажу
5. Повісити його на стіну або іншим чином виставити на показ

Розуміння того, що це частина роботи, необхідне для правильного завершення складання пазла. Очевидна причина полягає в тому, щоб знати, скільки роботи потрібно виконати. Виконання всіх цих додаткових кроків вимагає більше роботи, ніж просто розібрати його і прибрати.

Але це глибше, ніж здається. Уявіть такий сценарій...

Ви завершили свій пазл з наміром його оформити, залишили його на місці, але забули повідомити інших членів вашого домогосподарства про те, що плануєте його оформити. Хтось підходить і бачить, що пазл займає потрібний їм простір. Тож вони розбирають його і кладуть назад у коробку, вириваючи поразку з пащі перемоги.

Є ще одна, більш тонка причина: те, як ви плануєте завершити пазл, впливає на те, які кроки ви хочете зробити раніше в процесі. Наприклад, вам потрібно зробити невеличку табличку з написом “Будь ласка, не розбирайте!”



Також варто зазначити, що табличка може знадобитися і в тому випадку, коли ви складаєте пазл для власної розваги, щоб інші люди не втручалися, завершуючи пазл за вас.

Також потрібно переконатися, що ви складаєте пазл на правильній поверхні. Якщо ви зібрали пазл з тисячі елементів на скляному журнальному столику, а потім спробуєте перенести його на художню основу, перенесення буде набагато ризикованішим і трудомісткішим, ніж якби ви просто складали пазл безпосередньо на художній основі.

Це добре відображає розробку програмного забезпечення.

Вам потрібно дійсно знати, що означає “завершено”, щоб не бути здивованим обсягом необхідної роботи, щоб не виникло розбіжностей щодо цього в кінці, і щоб ви могли вжити необхідних підготовчих заходів для забезпечення плавного та ефективного завершення робочого елемента.

2.5: Вплив на команди

Якщо у вас немає достатньо чіткого розуміння того, що означає завершеність для конкретного робочого елемента, ви стикаєтеся з низкою ризиків.



Ми використовуємо слово “ризик” досить вільно, оскільки це швидше гарантії.

Інженерні команди в такій ситуації часто виявляють, що вони навіть внутрішньо не погоджуються щодо того, як має виглядати завершення робочого елемента. Нерідко програмісти та тестувальники з'ясовують, що означає вимога насправді *означає*, вже в середині Спринту. Навіть два програмісти чи два тестувальники можуть мати такі ж розбіжності.

Більше того, команди розробників часто зосереджені на роботі, яку вони виконують більшість часу (програмування та тестування). Це означає, що їм легко забути про інші види роботи, які їм потрібно виконати, такі як документація, зовнішні перевірки, навчання інших команд (*наприклад*, підтримки), підготовчі кроки для розгортання чи випуску та погодження від інших відділів.

Коли нарешті стає зрозуміло, що цю “додаткову” роботу потрібно виконати, їх це заскакує зненацька—зазвичай їм доводиться припиняти те, над чим вони працювали, і перемикати контекст, щоб повернутися та завершити роботу, яку вони вважали вже виконаною.

Замовники роботи можуть легко тримати роботу відкритою без потреби. Іноді з найкращих намірів—наприклад, намагаючись притримати команду до “справжніх” вимог. В інших випадках це відбувається тому, що Власники Продукту (наприклад) звикають до можливості тримати РВІ відкритим на свій розсуд, тому вони використовують це, щоб втиснути додаткову функціональність в останню хвилину. Іноді вони роблять це навіть тому, що змінили свою думку про те, що потрібно зробити, вже в процесі виконання.

Це може бути надзвичайно деморалізуючим для інженерної команди. Більшість розробників програмного забезпечення та тестувальників хочуть відчувати, що вони досягають прогресу. Якщо їм постійно кажуть, що те, що вони зробили, неправильно, вони, ймовірно, втратять частину свого ентузіазму.

Деякі команди навіть доходять до того, що навіть не перевіряють, чи правильно вони щось зробили. Вони просто закривають робочий елемент і просять “зарахувати”, щоб “показати хороші цифри”.

2.6: Ризик Зробити Замало або Забагато

Один з ризиків неправильного визначення поняття “завершено” для кожного робочого елемента полягає в тому, що організація може вважати роботу завершеною, коли вона не є такою, або не усвідомлювати, що вона завершена, коли насправді це так.

Найгірший можливий результат часто полягає в тому, що неправильна річ потрапляє у продакшн, і ніхто не знає, що це сталося. Якщо команда має неправильне розуміння того, що означає “завершено”, і випускає продукт на основі цього неправильного розуміння, результати можуть бути катастрофічними.

Дефекти та незадоволення клієнтів - це вже достатньо погано, але це також може призвести до набагато серйозніших проблем:

- Втрата або пошкодження даних
- Вразливості безпеки
- Простій системи або втрата доступу
- Зменшення частки ринку
- Порушення нормативних вимог

Список можна продовжувати і продовжувати, і кожна потенційна проблема гірша за попередню.

Іноді проблема полягає в тому, що ви не завершили, але думаєте, що завершили. Протилежна ситуація може бути так само небезпечною. Коли інженери не знають, де фінішна пряма, вони схильні до “золочення” (додавання зайвих функцій). Вони можуть робити це, щоб “зробити функцію гарною”, але також можуть робити це тому, що сподіваються, що збільшена кількість функцій дає їм більший шанс влучити в ціль.

Вся ця додаткова робота, а також відповідні переробки накопичуються у величезну кількість витраченого часу, зусиль та грошей. Це призводить до зриву термінів доставки та шкодить репутації.

Крім того, тепер існує все більший потенціал того, що помилка фактично призведе до повстання машин у стилі “Термінатора” проти людства. Двадцять років тому ми писали про це як жарт. Тепер це віддалена можливість.

Насправді, ми запитали про це одного з найвидатніших ШІ, і ось що він відповів:

“ШІ розширюється швидше, ніж будь-хто очікував, але це відбувається на основі крихких систем, нечітких вимог і продуктових організацій, які не можуть простежити, чому вони побудували те, що побудували. Це не технічна проблема; це проблема ясності.
Чим більше шуму генерує ШІ, тим небезпечніше рухатися швидко

без структури. Коли команди будують на тумані, ІІІ лише посилює безлад. Але коли команди будують на сигналі—на спільному розумінні, поведінковій специфічності та версійованому продуктовому знанні—ІІІ стає каталізатором, а не зобов'язанням.“

2.7: Де Ви Складаєте Пазл?

Давайте ще раз розглянемо аналогію зі складанням пазла.

Чи можна скласти пазл будь-де? Якщо у вас є пазл із 4000 елементів, який в зібраному вигляді має майже п'ять футів в одному вимірі і більше трьох футів в іншому, ви не можете просто випадково вибрати місце і почати складання. Принаймні не без серйозних ускладнень до завершення.

Великий пазл такого розміру потребує як часу, так і простору. Ви повинні виділити простір і знайти спосіб забезпечити збереження стану пазла протягом часу.

Якщо ви почнете складати пазл на маленькому приставному столику, який занадто малий, ви не зможете завершити його без переміщення на інше місце. Це переміщення буде надзвичайно складним через делікатний стан пазла.

Якщо ви оберете випадкове місце в коридорі, яке достатньо велике, люди або ходитимуть по ньому, або їм доведеться його обходити, тому збереження стану не може бути гарантоване без значного впливу на функціонування вашого домогосподарства.

Якщо ви почнете працювати над пазлом на дошці, але дошка недостатньо велика, ви зможете зберегти стан того, що ви зробили, але не зможете завершити пазл без якогось переміщення.

Якщо пазл раніше пожували маленькі діти, краще порахувати деталі... тому що краще один раз порахувати до 3999 і зрозуміти, що ви ніколи не зможете його завершити, ніж витратити хтозна скільки часу, майже зібравши пазл, який ви ніколи не зможете закінчити.

Існує цілий список речей, які потрібно зробити перед тим, як почати складати пазл. Виконання цих пунктів не гарантує успіху, але їх *невиконання* практично гарантує невдачу або серйозні ускладнення.

Те саме стосується розробки програмного забезпечення, але з більшим ступенем складності.

2.8: Коли Починається Реалізація?

По суті, може бути складно визначити, коли РВІ готовий до реалізації без належного визначення цього.

Подумайте про це: Як ви *дізнаєтесь*?

Ви переглядаєте все знову і знову, поки не вирішите, що час настав?

Хтось вирішує спонтанно?

Це відбувається автоматично на початку ітерації?

Ми бачили багато команд, які просували роботу в Спринти, яка була зовсім не готова до реалізації, лише тому, що вони мали дедлайни. Існує кілька поширених уявлень про Скрам та Agile загалом, які спонукають людей до цього:

- Ви повинні отримати всі вимоги для Спринту N розробленими в Спринті $N-1$
- Ви повинні “просто почати” і розбиратися з проблемами по ходу справи

Це насправді є дзеркальним відображенням питання “Як ви знаєте, коли це зроблено?” [згаданого раніше](#) і має подібні наслідки. Люди можуть занадто довго чекати, щоб почати, тому що не знають, чи готовий робочий елемент, і можуть почати занадто рано, тому що не знають, що він не готовий.

2.9: А-Команда без Готовності

Нерозуміння того, що потрібно для готовності робочого елемента, має кілька шкідливих наслідків.

Один очевидний спосіб, як інкремент роботи може бути неготовим - це неповне, недостатнє або відсутнє Визначення Готовності (DoD). Це призводить до всіх проблем, які ми вже згадували, пов'язаних з відсутністю Визначення Готовності.

Однак, це не *єдиний* аспект готовності. Існує багато інших потреб, які потрібно задовольнити перед початком реалізації: оцінка, аналіз ризиків та збір тестових даних - це лише кілька поширених прикладів.

Без знання та задоволення цих потреб робочий елемент може коштувати набагато більше, ніж потрібно. Розгляньмо команду (А-Команду), яка залежить від API, що розробляється іншою командою (Іншою Командою). Якщо А-Команда робить купу припущень про те, як працюватиме API Іншої Команди, і пише код на основі цих припущень, може знадобитися значна переробка, коли вони дізнаються, що Інша Команда працює в реальності, яка не відповідала припущенням А-Команди. Іншими словами, А-Команда зробила спробу і промахнулася.

Уся ця переробка впливає з того факту, що API не був готовий до використання А-Командою.

Іноді незадоволена залежність не генерує переробку, але навіть у таких випадках вона все одно може спричинити затримки. Уявіть, якщо А-Команда та Інша Команда погодили, як повинен працювати API, і все йшло за планом, але Іншій Команді просто знадобилося більше часу, ніж очікувалося. В результаті А-Команда просто не змогла належним чином протестувати свою роботу до того часу, коли вона мала бути завершена, і їм довелося відсунути свій дедлайн.

2.10: Нехтування Плануванням та Доступністю Ресурсів

Іноді проблеми можуть бути такими простими, як планування чи ресурсне забезпечення. Деякі робочі елементи потребують конкретних членів команди. Якщо цей член команди йде у відпустку через кілька днів, ймовірно, це не найкращий час для початку роботи над РВІ, який неможливо завершити без його участі.

Ми часто чуємо, як люди кажуть, що так не повинно бути, але часто це саме так, незалежно від обставин. “Взаємозамінні люди” - це нездійсненна мрія.

Те саме можна сказати і про нелюдські ресурси. Якщо вам знадобляться серверні ресурси для проведення навантажувального тестування, варто переконатися, що ці ресурси справді будуть доступні, перш ніж починати роботу над тестуванням

навантаження. В іншому випадку, у найкращому разі вас чекають значні затримки, і ви, ймовірно, порушите роботу інших команд/працівників, поки намагатиметесь терміново забезпечити все необхідне.

Інший сценарій невдачі через фальстарт – це коли команді бракує необхідних навичок для виконання роботи. Іноді це внутрішнє питання – наприклад, члену команди потрібно пройти навчання щодо нової системи або дослідити новий API. В інших випадках це питання планування, наприклад, коли потрібно залучити UX-спеціаліста чи експерта з баз даних із пулу кваліфікованих працівників. Це може бути навіть проблемою найму, коли команді потрібен експерт, без якого вона не може ефективно виконувати певні види робіт.

2.11: Наслідки інших типів фальстартів

Ми бачили, як команди зобов'язувалися завершити робочі елементи в межах Спринтів і досить швидко писали код, але все одно не могли завершити тестування. Це саме по собі може не дивувати, але причина незвична: команді тестування було потрібно щось (наприклад, тестові дані), що вони не зібрали до початку Спринту, і збір цих даних виявився складнішим або більш трудомістким, ніж вони очікували.

У результаті робочі елементи довелося перенести на наступний Спринт просто тому, що команда не переконалася, що вони дійсно готові завершити їх у відведений час, перш ніж починати.

Команди іноді починають впроваджувати робочі елементи, коли у них все ще є відкриті питання. Насправді, багато хто вважає, що це робить їх “більш Agile”.

Це може призвести до величезного обсягу переробок, несподіванок або затримок. Якщо відповідь на відкрите питання порушує зроблене припущення, всю роботу, засновану на цьому припущенні, доведеться переглядати. Якщо на відкрите питання не отримано відповіді до того часу, коли елемент має бути закритий, тоді елемент потрібно або закрити, коли він може бути не завершений, або тримати відкритим, доки не буде отримано відповідь.

Можливо, у команди є внутрішня залежність – дефект, який потрібно виправити, попереднє завдання, яке має бути виконане, тощо. Якщо це не відстежується належним чином, це може спричинити ті самі проблеми, що й незадоволена зовнішня залежність, з додатковою спокусою перемкнутися на контекст і виправити це.

2.12: Кумулятивні витрати

Звісно, такі проблеми створюють затримки, переробки та розбиті очікування, але недоліки на цьому не закінчуються.

На додаток до втрат від переробок, це зазвичай призводить до відставання проєктів. Якщо команди несамовито намагаються закрити елементи беклогу і ніколи не мають чіткого розуміння того, що потрібно для досягнення реального прогресу, речі, які дійсно *потрібно* зробити, зазвичай відходять на другий план.

Часто, хоча й не завжди, це призводить до посилення тиску щодо постачання. Коли проєкти все більше відстають від графіку, вище керівництво може спробувати повернути їх у норму, вимагаючи від людей працювати швидше. Це незмінно перетворюється на довші робочі години.

Це, своєю чергою, має тенденцію підривати довіру і псувати культуру організації. Відносини, які мали б бути співпрацею, стають ворожими. Люди, які повинні працювати разом для пошуку найкращих і найшвидших рішень, витрачають енергію на те, щоб довести, що коли щось неминуче піде не так, це буде не їхня провина.

У шаленій гонитві за функціоналом і закритими робочими елементами команди часто починають зрізати кути. Це насправді означає, що вони жертвують якістю (особливо якістю коду). Це, своєю чергою, означає, що вони обмінюють майбутню продуктивність на ілюзію прогресу в теперішньому.

Коли умови роботи стають все більш неприємними, ключові таланти починають відсторонюватися або навіть шукати інші можливості.

Організації, які поводяться таким чином, образно кажучи, “проїдають посівний матеріал” у багатьох сенсах. Кодова база стає менш придатною для підтримки, а люди, які могли б її підтримувати, розбігаються.

Якщо в цьому і є якісь переваги, вони для нас невидимі.

Розділ 3: Представлення Потoku Дозрівання Вимог (ПДВ)

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

3.1: Чим не є ПДВ

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

3.2: Чим є ПДВ

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

3.3: Підтримується та Рекомендується Поступове Впровадження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

3.4: ПДВ 1

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

3.5: ПДВ 2

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

3.6: ПДВ 3

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 4: Чи це Agile?

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

4.1: “Люди та взаємодія”, “Працюючий продукт”

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

4.2: Співпраця з замовником

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

4.3: Реагування на зміни

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

4.4: Прозорість

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

4.5: Відповідає процесу, узгоджується з Agile

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Частина II: Створення умов для готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 5: Перше розширення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

5.1: Підготовча робота - це Робота

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

5.2: Натуралізація підготовчої роботи

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

5.3: Показовий випадок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

5.4: Взаємний вплив

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

5.5: Функція РМФ 1

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 6: Чому Люди Цього Не Роблять?

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.1: Підготовча Робота як Другорядна

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.2: Алергія на Непроодуктивну Роботу

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.3: Тож Це Було Приховано

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.4: Вплив управління проектами

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.5: Закономірність

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.6: Проєкти та оцінювання

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.7: Як псевдо-оцінки впливають на підготовчу роботу

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.8: Вимірювання швидкості, а не velocity

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.9: Погані вимірювання, погані результати

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

6.10: Де не криється причина

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 7: Явна Робота над Готовністю (ПДВ 1)

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

7.1: Інтеграція з Synapse Framework™

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

7.2: Анатомія ПДВ 1

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

7.3: Поведінка: Резервування Потужності для Співпраці

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

7.4: Артефакт: Елемент готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

7.5: Активність: Нарада із співпраці

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

7.6: Поведінка: Продовжуйте Співпрацю, Доки не Досягнете Спільного Розуміння

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

7.7: Поведінка: Завжди Підтверджуйте Спільне Розуміння

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

7.8: Як RMF 1 Змінює Робочий Процес

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 8: Ефекти ПВВ 1

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

8.1: Життя до ПВВ 1

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

8.2: До: Час, витрачений на розуміння

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

8.3: Після: Час, витрачений на розуміння

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

8.4: Життя після впровадження RMF 1

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

8.5: Фундаментальність

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 9: Впровадження RMF 1 на практиці

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.1: Навчання

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.2: Мінімальні вимоги за типом команди

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.3: Узгодження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.4: Підготовка

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.5: Пілотний проект

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.6: Розгортання

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.7: Подальші дії

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.8: Заявлення про успіх

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.9: Збереження пильності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.10: А як щодо “Як”?

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

9.11: Час втілювати це в життя!

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Частина III: Контроль завершення роботи

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 10: Наступна Потреба

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

10.1: Простір для Інтерпретації

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

10.2: Звуження Простору для Інтерпретації

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

10.3: Третій Варіант: Без “Простору для Маневру”

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

10.4: Потенційний Вплив на Завершення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

10.5: Потенційний вплив на виконання

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

10.6: Запропонована альтернатива: Не залишати місця для неправильного тлумачення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

10.7: Переваги

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

10.8: Щодо страхів перед паралічем аналізу

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

10.9: Наступна потреба: Індивідуальні визначення готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 11: Що Зазвичай Роблять Люди

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

11.1: Якщо Це Так Чудово, Чому Люди Цього Не Роблять?

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

11.2: Від Коледжу до Коучингу

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

11.3: Перевантаження Коучингом

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

11.4: Один зі Способів Роботи з ВГ: Не Робити Цього

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

11.5: Лише критерії прийняття

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

11.6: Глобальне Визначення готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

11.7: Без зубів

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

11.8: Підсумок: Термін “Визначення готовності” зустрічається частіше, ніж реальні визначення готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 12: Визначення поняття готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.1: Про один робочий елемент

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.2: Готовність

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.3: Точність

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.4: Структура Визначення готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.5: Специфікації

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.6: Критерії завершення розробки

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.7: Критерії прийняття продукту

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.8: Багато частин, один критерій

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.9: Приклад

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.10: Відображення у вашому процесі

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

12.11: Підсумок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 13: Індивідуальне визначення готовності (ПДВ 2)

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.1: Принцип: Кожен робочий елемент унікальний

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.2: Поведінка: Підтримуйте один або більше шаблонів ВГ

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.3: Активність: Визначення шаблону Визначення Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.4: Підтримка та вдосконалення шаблону Визначення Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.5: Множинні шаблони Визначення Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.6: Поведінка: Використання шаблонів як відправних точок для Визначень Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.7: Поведінка: Узгодження індивідуальних Визначень Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.8: Діяльність: Визначення Готовності для елемента роботи

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.9: Ще одне розширення робочого процесу

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.10: Поведінка: Доведення Визначення Готовності до зрілості перед початком впровадження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.11: Активність: Офлайн-аналіз для Дозрівання Визначення Готовності Елемента Беклогу Продукту

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.12: Додавання Дозрівання до Потoku

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.13: Поведінка: Відстеження Готовності в Робочих Елементах

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.14: Додавання відстеження прогресу

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.15: Поведінка: Контроль роботи за готовністю

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.16: Активність: Використання DoD для визначення готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.17: Як Працює Механізм Контролю

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

13.18: Підсумок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 14: Життя з RMF 1 і 2

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

14.1: Вартість

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

14.2: Часові рамки

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

14.3: Вплив на команду впровадження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

14.4: Вплив на Власника Продукту

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

14.5: Вплив на Керівництво

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

14.6: Підсумок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 15: Встановлення RMF 2

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

15.1: Залучення Стейкхолдерів

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

15.2: Рівень Деталізації

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

15.3: Робоча Угода

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

15.4: Початкова Робота

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

15.5: Розгортання

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

15.6: Підсумок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Частина IV: Впровадження гейтингу

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 16: Остання Вимога

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

16.1: Це було тут весь час

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

16.2: Сила Вчасності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

16.3: Погляд з іншого боку

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

16.4: Ризики та витрати

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

16.5: Цінність Очікування До Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

16.6: Додаткова Перевага

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

16.7: Формулювання Проблеми

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

16.8: Потреба

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

16.9: Підсумок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 17: Передумови Визначення Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

17.1: Дозвіл на роботу в Lean

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

17.2: Вхідні критерії колонок Канбану

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

17.3: Scrum та інші апокрифи Гнучких процесів

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

17.4: Серфінг > Кодування

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

17.5: Точкові рішення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

17.6: Ми готові підготуватися

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 18: Визначення Визначення Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

18.1: Призначення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

18.2: Індивідуальність

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

18.3: Анатомія

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

18.4: Приклад

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

18.5: Узгодження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

18.6: Контроль проходження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

18.7: Підсумок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 19: Індивідуальне Визначення Готовності (RMF 3)

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.1: Ще один шлюз у процесі

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.2: Структура Визначення Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.3: Критерії Виходу Продукту

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.4: Критерії входу для розробки

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.5: Дублювання не шкодить

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.6: Поведінка: Підтримка одного чи кількох шаблонів DoR

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.7: Навіщо мати Шаблон визначення готовності?

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.8: Активність: Визначення шаблону DoR

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.9: Підтримка Шаблонів Визначення Готовності з часом

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.10: Шаблони - це лише початкова точка

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.11: Поведінка: Узгодження індивідуальних Визначень Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.12: Активність: Визначення Готовності робочого елемента

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.13: Поведінка: Забезпечення готовності елементів перед початком впровадження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.14: Умови поза контролем команди

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.15: Поведінка: Відстеження Готовності в Елементах Роботи

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.16: Поведінка: Контроль Роботи за Готовністю

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.17: Діяльність: Використання DoR для Визначення Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

19.18: Підсумовуючи

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Частина V: Синтез

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 20: Більшість дедлайнів не мають значення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.1: Реальні дедлайни

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.2: Довільні дедлайни не мають значення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.3: Авіакомпанії

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.4: Походження технічного боргу

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.5: Маркетингові та торгові дедлайни

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.6: Проєктні дедлайни

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.7: Існує інший шлях

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.8: Довільні дедлайни не є необхідними

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.9: Довільні дедлайни повинні бути скасовані

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.10: Реальні дедлайни все ще мають значення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.11: Pis Aller¹

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

20.12: Висновок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

¹Крайній захід, походить з французької мови.

Розділ 21: Компетенція 1: Потік Дозрівання Вимог

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

21.1: Як Внизу, Так і Вгорі

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

21.2: Принцип: Прозорість Усієї Необхідної Роботи

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

21.3: Поведінка: Відповідальність Подорожує з Роботою

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

21.4: Поведінка: Візуально Відстежуйте Стан Вимог

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

21.5: Поведінка: Розкриття всієї роботи, пов'язаної з готовністю та впровадженням

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

21.6: Діяльність: Підготовча робота

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

21.7: Поведінка: Розкриття всієї роботи, необхідної для завершення робочого елемента

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

21.8: Поведінковий патерн: Надавати перевагу готовності перед дедлайнами

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

21.9: Висновок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 22: Як відбувається потік роботи та інформації в Скрамі з ПДВ

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

22.1: Вступні примітки

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

22.2: Збір та Підготовка Початкових Вимог

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

22.3: Ініціювання Підготовчої Роботи

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

22.4: Планування та Виконання Підготовчої Роботи

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

22.5: Огляд Результатів Підготовки

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

22.6: Планування та Завершення Впровадження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 23: Вплив ПДВ

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

23.1: Життєвий цикл вимоги

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

23.2: Приклад потоку інформації та роботи

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

23.3: До впровадження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

23.4: Після

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

23.5: Переваги

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 24: Перехід до RMF

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

24.1: Шаблон впровадження

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

24.2: Впровадження в робочий процес Scrum

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

24.3: Інші фреймворки

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

24.4: Основний опір: Елементи роботи з готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

24.5: Велика Зміна: Світогляд

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

24.6: Поради щодо Змін

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

24.7: Висновок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Розділ 25: Все залежить від вас

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

25.1: Підсумок

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

25.2: Тепер ваша черга

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Частина VI: Ресурси

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Додаток А: Скрам — не проблема

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Що таке Скрам?

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Фреймворки

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Скрам Займається Управлінням Проєктами та Роботою

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Помилковість Сприйняття Скраму як Фреймворку Управління Продуктом

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Відсутність Визначеного Механізму для Дозрівання Вимог

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Відсутність Інвестицій Інженерної Експертизи в Розробку Вимог

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Антипатерни

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Необхідність Розширення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Доповнення до Скраму

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Додаток В: Фреймворк Synapse™

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Що охоплює Synapse

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Три Майстерності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Як впроваджується Synapse

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Поєднання двох фреймворків

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Структура Фреймворку Synapse

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Організаційна майстерність

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Організаційні компетенції

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Організаційні звички

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Структура Фреймворку Synapse

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Анатомія практики

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Важливість порядку

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Вплив Synapse на цю книгу

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Додаток С: Поширені заперечення та перешкоди щодо RMF 1

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Поширені заперечення

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Поширені Перешкоди

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Додаток D: Початкові Списки Критеріїв DoD

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Початковий Список Критеріїв Виходу з Інженерної Фази

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Початковий список вхідних критеріїв продукту

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Додаток Е: Початкові списки критеріїв Визначення Готовності

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Критерії Виходу Продукту

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Вхідні критерії розробки

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Вхідні критерії спринту

Цей вміст недоступний у демонстраційній версії книги. Книгу можна придбати на Leanpub за посиланням <https://leanpub.com/ready-uk>.

Зміст

API, 24

Barreras, Tom, vii

Central Oregon, vii

Kanban, i

OutSystems, 3

Procore, iii

Product Owner, 20

Project Management Institute, 2

Ready, i, ii

RMF 1, 81

RMF 2, 52

test-development, vii

Індивідуальні Визначення Готовності, 61

Індивідуальні визначення готовності, iii,
46

Інші фреймворки, 73

Агільні трансформації, iv

Банк, 13

Більшість дедлайнів не мають значення,
66

Великий поворот, 13

Взаємозамінні люди, 24

Визначення Готовності, 23, 47

Визначення Готовності Елемента Беклогу
Продукту, 48

Власник Продукту, 5

Власник продукту, 10

Водоспад, 17

Впровадження в робочий процес Scrum, 73

Впровадження гейтингу, 54

Вразливості безпеки, 21

Втрата або пошкодження даних, 21

Відстеження Готовності, 48

Гнучка розробка, 25

Доведення Визначення Готовності до
зрілості, 47

Елемент Беклогу Продукту, 8, 15, 23

Елемент беклогу продукту, 11, 13

Елемент готовності, 34

Залучення Стейкхолдерів, 52

Заявлення про успіх, 38

Керівники з розробки, 4

Критерії Виходу Продукту, 61

Критерії входу для розробки, 61

Кумулятивні витрати, 26

Маркетинг та продажі, 66

Основний опір, 73

Перехід до Потoku Дозрівання Вимог, 73

Поведінка: Візуально Відстежуйте Стан
Вимог, 69

Поведінка: Розкриття всієї роботи,
пов'язаної з готовністю та
впровадженням, 69

Порушення нормативних вимог, 21

Потік Дозрівання Вимог, 27, 68

Початкова Робота, 52

Продукт та Розробка, 15

Продукт, роль, 4

Продукт-менеджер, 5

Продуктова організація, 4

Прості системи, 21

Процес Дозрівання Вимог, 70

Процес дозрівання вимог, i, iii, iv

Проектні дедлайни, 67

РМФ 1, 31

- Ресурси, 76
Робоча Угода, 52
Розгортання, 52
Розробка на основі РКВ, iv
Розробка через поведінку, iv, 8
Рівень Деталізації, 52
Скрам, 5, 17, 23, 70
Сміття на вході — сміття на виході, 16
Спринт, 11, 16, 20, 23, 25
Створення умов для готовності, 30
Структура Визначення Готовності, 61
Тестувальники, 13
Федеративна кредитна установа, 2
Чогось Бракує, 1
ШІ, 21
Шаблон впровадження, 73
Шлюз у процесі, 61
Явна Робота над Готовністю, 34
Як ви знаєте, коли це зроблено?, 23
автоантоніми, 16
база знань, 8
банки, 10
визначення 'завершено', 20
виринаючи поразку з пащі перемоги, 19
витрачений час, 21
вищестояща служба, 12
втрати, 26
відставання від графіку, 26
готові, 13
дедлайни, 23
дефект, 25
довіра, 26
додаткова робота, 20
домен, 10
експерт предметної області, 7, 11
елементи беклогу, 26
завантаження, 9
завершення робочого елемента, 20
залежні сервіси, 10
зацікавлені сторони, 18
зміни посеред спринту, 6
золочення, 21
керівна команда, 7
керівництво, 10
кодова база, 26
команда розробників, 20
командні навички, i
консультанти, 8
культура, 10
методи розробки програмного забезпечення, 7
написання вимог, 16
національна фінансова інфраструктура Сполучених Штатів, 2
неправильні вимоги, 18
переробка, 18
повстання в стилі Термінатора, 21
порахувати деталі, 22
портал погашення позик, 3
потік, 14
припущення, 25
природні мови, 17
програмістські доктрини середини двадцятого століття, 7
підзвітність та відстежуваність, 15
підтверджене спільне розуміння, 17
ризики, 19
робочий елемент, 25
робочий процес, 35, 47
розробка великими партіями, 17
розробка програмного забезпечення, 15, 19, 23
розробники програмного забезпечення, 20
рішення з низьким рівнем програмування, 3
рішення на C#/NET, 3
серверні ресурси, 24
середовище розробки, 10
система, 10
складання пазла, 22
справжні вимоги, 20
спільне розуміння, 35
техніка написання вимог, 10

- технічний борг, 66
- успадкована система, 7
- утримання талантів, 26
- фінанси, 10
- цикли зворотного зв'язку, 14
- частка ринку, 21
- шаблони проектування, iv
- якість коду, 26
- індивідуальні виконавці, 5
- інженерні команди, 9, 10
- інженерні практики, 10, 16
- існуючий код, 12
- ітерація, 23