

DOI: <https://doi.org/10.32782/2524-0072/2024-67-169>
УДК 005

МЕТОДИКА SCRUM: ВПЛИВ НА КУЛЬТУРУ ТА ПРОДУКТИВНІСТЬ ІТ-КОМАНД

SCRUM METHODOLOGY: IMPACT ON CULTURE AND PRODUCTIVITY OF IT TEAMS

Чабан Сергій Валерійович

аспірант,

Івано-Франківський національний університет нафти і газу

ORCID: <https://orcid.org/0009-0009-8162-6964>

Король Світлана Василівна

кандидат економічних наук, доцент,

Івано-Франківський національний університет нафти і газу

ORCID: <https://orcid.org/0000-0002-4804-7612>

Chaban Serhii, Korol Svitlana

Ivano-Frankivsk National Technical University of Oil and Gas

У статті досліджується вплив методика Scrum на культуру та продуктивність ІТ-команд у контексті сучасних викликів у сфері інформаційних технологій. Акцентується на тому, що, незважаючи на зростаючу популярність Scrum, його впровадження може призводити до негативних наслідків, таких як вигорання працівників і зайве ускладнення робочих процесів. Проаналізовано актуальні дослідження, які підтверджують як позитивний, так і негативний вплив Scrum на продуктивність команд. Однак наголошується на важливості вивчення довгострокових наслідків використання цієї методика. Стаття також виявляє невирішені аспекти, такі як вплив постійного зворотного зв'язку на робочу культуру, та формулює мету дослідження: вивчення викликів впровадження Scrum і розробка рекомендацій для оптимізації процесу. Результати дослідження можуть сприяти глибшому розумінню теорії гнучкого управління та поліпшенню практик в ІТ-індустрії.

Ключові слова: Agile, Scrum, культура ІТ-команд, продуктивність scrum команд, адаптація до змін, впровадження Scrum, вигорання ІТ-працівників.

This article explores the impact of the Scrum methodology on the culture and productivity of IT teams amidst contemporary challenges in the information technology sector. The relevance of this topic stems from the growing adoption of Agile practices, particularly Scrum, which have revolutionized project management in software development. However, the widespread implementation of Scrum raises significant questions regarding its effectiveness and potential drawbacks. The primary aim of this study is to investigate the multifaceted challenges associated with implementing Scrum and to develop practical recommendations for optimizing its application in diverse organizational contexts. The research employs a qualitative approach, utilizing interviews and case studies to gain insights from professionals and teams actively engaged in Scrum practices. Through these methodologies, the study captures a range of experiences, highlighting common themes and divergences in how Scrum is perceived and applied. The findings reveal a complex landscape where Scrum's advantages, such as enhanced collaboration, improved flexibility, and faster delivery times, coexist with challenges including employee burnout, confusion over roles, and increased complexity in workflows. Notably, the study highlights the importance of understanding long-term consequences and the necessity for continuous feedback within teams to foster a supportive work culture. It suggests that organizations must be mindful of these challenges and proactively address them to harness the full potential of Scrum. The practical value of this article lies in its contribution to the ongoing discourse on Agile methodologies, providing insights that can help organizations refine their Scrum practices. By articulating the challenges and offering actionable recommendations, this research serves as a valuable resource for IT managers and team leaders seeking to improve productivity while maintaining a healthy work environment. Ultimately, this study underscores the need for a nuanced approach to implementing Scrum, balancing its benefits against potential pitfalls to cultivate a more effective and sustainable work culture in the IT industry.

Keywords: Agile, Scrum, culture of IT teams, productivity of scrum teams, adaptation to changes, implementation of Scrum, burnout of IT employees.

Постановка проблеми. Сучасний світ інформаційних технологій вимагає від компаній постійної адаптації до змін, швидкого реагування на нові виклики та забезпечення високої продуктивності команд. Одним із найбільш популярних підходів для досягнення цих цілей є методика Scrum, що базується на принципах Agile. Scrum покликаний підвищити гнучкість у вирішенні проблем, швидкість реагування на нові виклики та ефективність розробки продукту. Проте, не дивлячись на популярність та поширення цього підходу, в ІТ-компаніях все частіше постає питання про його негативні аспекти. Впровадження Scrum може змінювати внутрішню культуру організації, впливаючи на відносини між працівниками, менеджерами та клієнтами. Виникає потреба розуміти, як методика Scrum впливає на взаємодію та культуру в команді, а також на її продуктивність. Постійний тиск на оперативне виконання завдань у коротких циклах (спринтах) часто призводить до вигорання команд. Відсутність чіткої довгострокової стратегії також може сприяти дезорганізації робочих процесів та зниженню загальної продуктивності. Отож, вивчення впливу методики Scrum на культуру та продуктивність команд в ІТ-сфері є актуальним науково-практичним завданням, що сприятиме глибшому розумінню теорії гнучкого управління та поліпшенню реальних бізнес-процесів.

Аналіз останніх досліджень і публікацій.

У дослідженнях, присвячених методологіям гнучкого управління, зокрема Scrum, значна увага приділяється їх позитивному впливу на продуктивність команд і гнучкість організацій. Такі автори, як Schwaber і Sutherland (2020) у своїй роботі «The Scrum Guide», описують основні принципи Scrum, його структуру та переваги для розробки програмного забезпечення [4]. Власне, цей документ є основою для розуміння не тільки теоретичних основ Scrum, але й інших досліджень. Аналізуючи останні дослідження та публікації, необхідно звернути увагу на «17-й звіт про стан Agile» за 2023 рік, який містить цінну інформацію про проблеми та результати методики Scrum. Згідно з представленими даними, можемо відзначити: 71% опитаних організацій використовують Agile-практики, однак лише 11% «дуже задоволені» їхніми результатами, при цьому великі підприємства стикаються зі значними перешкодами при масштабуванні Agile-практик, таких як Scrum [2]. Ці перешкоди включають організаційний опір змінам, непорозуміння керівництва та відокремлені

операції, що ускладнює підтримку співпраці між відділами та командами.

Виділення невирішених раніше частин загальної проблеми. Попри значну кількість досліджень щодо впровадження Scrum у сфері ІТ, залишаються невирішені аспекти, які потребують подальшого вивчення. Хоча методика Scrum сприяє швидкому виконанню завдань у межах спринтів, довгострокові наслідки такого ритму роботи залишаються недостатньо дослідженими. Наразі не до кінця зрозуміло, яким чином постійний тиск на короткострокову продуктивність впливає на загальний рівень вигорання, мотивацію та ефективність команд у довгостроковій перспективі. Хоча багато досліджень звертають увагу на позитивний вплив гнучких методологій на комунікацію та співпрацю в командах, залишаються відкриті питання щодо того, як Scrum впливає на формування робочої культури в різних організаційних контекстах. Зокрема, постає питання: чи дійсно практика постійного зворотного зв'язку сприяє зростанню продуктивності, чи вона стає джерелом стресу і конфліктів у командах? Власне, на цьому і зосередимо свою увагу у даному дослідженні.

Формулювання цілей статті (постановка завдання). Мета даної статті полягає в дослідженні впливу методики Scrum на продуктивність і культуру ІТ-команд, що дозволить виявити виклики при впровадженні даної методики та запропонувати рекомендації для оптимізації цього процесу.

Виклад основного матеріалу дослідження. Впровадження методики Scrum істотно впливає на культуру команди, змінюючи не лише робочі процеси, але й динаміку взаємовідносин між учасниками. Однією з ключових змін є акцент на співпраці та безперебійній комунікації, що закладено в основу Scrum. Регулярні зустрічі команди, серед яких щоденні стендапи, спринт-рев'ю та ретроспективи, сприяють відкритості у спілкуванні та створюють атмосферу прозорості, де кожен учасник може вільно висловлювати свою думку, ділитися проблемами та обговорювати шляхи їх вирішення. Такий підхід не тільки зміцнює довіру серед членів команди, але й формує культуру колективної відповідальності, де кожен розуміє свій внесок у загальний результат.

Проте часті й інтенсивні обговорення можуть створювати додаткове навантаження на команду. Для деяких учасників постійний контроль і фідбек можуть бути джерелом

стресу, особливо якщо вони не звикли до таких коротких циклів виконання завдань і регулярного перегляду своїх результатів. Крім того, часті взаємодії також можуть загострювати конфлікти, так як різноманітні підходи до вирішення задач і оцінки продуктивності здатні провокувати напруженість. У середовищі, де важливим є швидке реагування та оперативне вирішення проблем, інколи з'являється небезпека втрати уваги до довгострокових цілей та стратегічного розвитку. Очевидно, що такий процес негативно впливає на внутрішню культуру команди, спричиняючи дезорганізацію.

Однією з проблем застосування Scrum є традиційна структура двотижневих спринтів, яку більшість компаній використовують за замовчуванням. Разом із постійними щоденними зустрічами, така структура часто стає негнучкою і не дозволяє менеджменту адаптувати тривалість спринтів до реальних потреб команди та проекту. Двотижневий спринт не завжди відповідає оптимальним вимогам конкретних етапів розробки, оскільки в різних фазах проекту необхідна різна частота узгоджень і комунікацій (рис. 1).

На початкових етапах розробки проекту, коли багато аспектів ще не визначені, а напрям розвитку продукту може коригуватися, ідеальною стратегією є використання коротших спринтів, – тривалістю один тиждень. Така тривалість дозволяє команді частіше синхронізуватися з клієнтами та іншими зацікавленими сторонами, щоб регулярно підтверджувати правильність обраного напрямку розробки. Це також сприяє швидкому виявленню проблем і прийняттю своєчасних рішень, що особливо важливо на етапі встановлення чітких вимог до проекту. Очевидно, що зі збільшенням кількості узгоджень і чіткішим розумінням ключових елементів проекту потреба в таких частих синхронізаціях зменшується. У цьому випадку довжина спринта може поступово збільшуватися до двох або навіть чотирьох тижнів. Це дозволяє

команді менше витратити часу на щоденні комунікації та сфокусуватися на більш тривалих завданнях, підвищуючи продуктивність і знижуючи навантаження, пов'язане з частими зустрічами.

Оптимальний підхід до довжини спринтів полягає в її гнучкості, адаптованій до етапу проекту. На початку розробки варто використовувати один тиждень для більшої узгодженості, а надалі поступово збільшувати тривалість спринтів, коли напрям розробки чітко визначено. Алгоритм такого підходу виглядає наступним чином: на початку проекту встановлюється спринт на один тиждень. В кінці спринта команда оцінює, чи рухається вона в правильному напрямі. Якщо так, то можна збільшити тривалість спринта, щоб зменшити потребу в частих обговореннях. Якщо ж з'являються нові питання або виникає потреба в додаткових узгодженнях, довжина спринта залишається максимально короткою (один тиждень) для ефективного реагування на зміни (рис. 2).

Одним із ключових елементів методики Scrum є щоденні зустрічі (daily), які спрямовані на забезпечення регулярної комунікації між членами команди та менеджментом. Вони дозволяють оперативно обговорювати прогрес, виявляти перешкоди та визначати подальші кроки. Проте стандартний підхід до цих зустрічей може мати низку недоліків. У більшості компаній вони проводяться незалежно від реальних потреб, що стає частиною рутинної діяльності, а не інструментом для вирішення проблем. Це може призводити до перенасичення комунікаціями, коли фіксовані двотижневі спринти включають щоденні зустрічі протягом усього процесу розробки, навіть на етапах, де частота таких обговорень не є доцільною.

Такі зустрічі особливо корисні на початкових етапах проекту, коли команда ще формує спільне бачення та узгоджує ключові аспекти з клієнтами. Вони забезпечують постійну

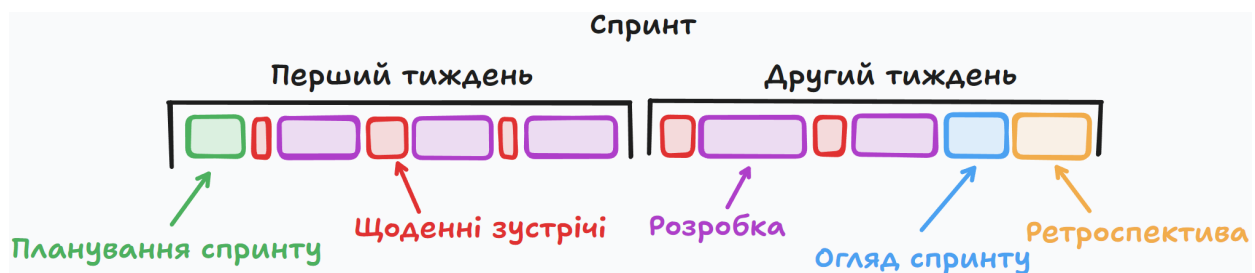


Рис. 1. Найпопулярніша структура Scrum спринту

Джерело: побудовано авторами за допомогою tldraw

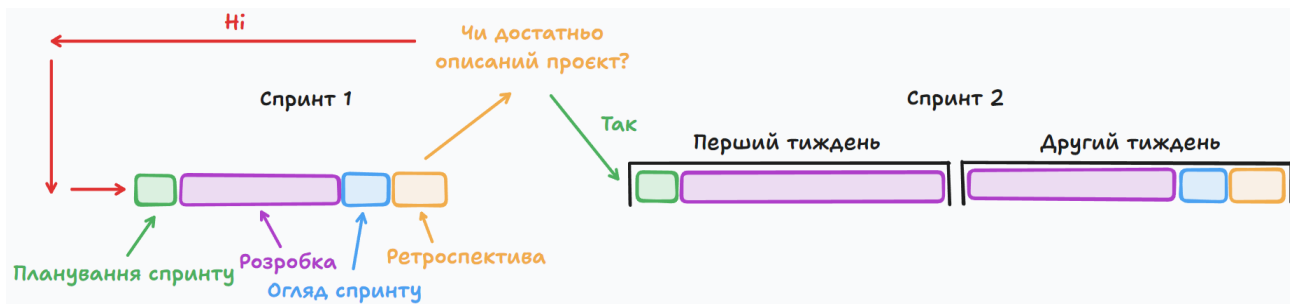


Рис. 2. Динамічна структура Scrum спринту

Джерело: побудовано авторами за допомогою tldraw

синхронізацію дій та дозволяють швидко реагувати на зміни чи невизначеності. Проте, як тільки більшість аспектів проєкту узгоджені, і команда отримує чітко розуміння вимог, необхідність щоденних зустрічей зменшується.

Часті зустрічі можуть стати джерелом додаткового навантаження на команду, оскільки постійне обговорення поточних завдань може відволікати від глибинної роботи над проєктом. Крім того, надмірна комунікація може знижувати загальну мотивацію та продуктивність, перетворюючи робочий процес на серію формальних обговорень, замість фокусування на вирішенні реальних проблем. Важливо розуміти, що суть ефективної комунікації полягає не в її частоті, а в її доцільності та змістовності.

Замість того, щоб дотримуватися жорсткого правила щоденних зустрічей, доцільно застосовувати гнучкий підхід до комунікації. Обговорення повинні виникати з реальної потреби, коли це дійсно необхідно для вирішення проблем чи ухвалення рішень. Часто такі обговорення краще проводити у форматі індивідуальних зустрічей (один на один) між тими, кого безпосередньо стосується проблема. Це дозволяє не тільки скоротити час на непотрібні зустрічі, але й зосередити увагу на ключових питаннях, що потребують розв'язання в конкретний момент часу.

Така зміна підходу до комунікації може позитивно вплинути на культуру роботи команди, дозволивши їй зосереджуватись на основних завданнях та знижуючи рівень стресу через необхідність постійно звітувати про кожен етап роботи. Відмова від щоденних зустрічей сприяє більш самостійній роботі, підвищуючи рівень відповідальності кожного учасника команди за свої завдання. У результаті, команда отримує більше часу на вирішення реальних проблем, що сприяє підвищенню продуктивності та якості розробки.

На початку проєкту, коли scrum спринти проводяться в коротких циклах, старші спеціалісти, серед яких є як менеджери так і програмісти, працюють над ключовими елементами рішення, перш ніж розпочати повноцінну роботу над проєктом. Задачі повинні визначатись на правильному рівні абстракції: вони мають бути достатньо конкретними, щоб команди знали, що робити, але водночас досить абстрактними, щоб мати простір для самостійного опрацювання конкретних деталей (рис. 3). Коли формуються задачі, потрібно робити це на правильному рівні абстракції: не надто розпливчато й не надто конкретно. Продукт-менеджери часто помиляються в одній із цих двох дій [3].

Коли замовники переходять безпосередньо до варфреймів (wireframes) або високоякісних макетів, вони визначають дуже багато деталей, однак занадто рано. Це не залишає дизайнерам місця для творчості. Надмірна специфікація дизайну також призводить до помилок при оцінці часу на виконання завдання. Чим конкретніша робота, тим важче її оцінити, тому що створення інтерфейсу саме таким, як хоче замовник, може вимагати вирішення прихованих складнощів і деталей реалізації, які не були видимі в макеті. Коли масштаб завдання не можливо змінити, команда не може переглянути дизайнерське рішення, яке виявилось, скажімо, дорожчим, ніж воно здавалось на початку.

З іншого боку є проєкти, які мають занадто розпливчастий опис, проте вони не працюють. Коли проєкт описується кількома словами, ніхто не знає, що це означає. «Створити перегляд календаря» або «дати групі співіщення» звучить зрозуміло, але що саме вони мають під собою? Члени команди не мають достатньо інформації, щоб робити компроміси. Вони не знають, що включити або виключити. Це означає, що має вирішуватися

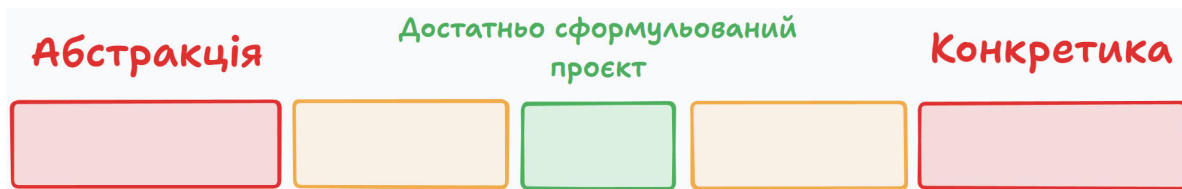


Рис. 3. Схема збереження балансу між абстракцією та конкретикою про формулюванні проекту

Джерело: побудовано авторами за допомогою tldraw

проблема без контексту. Виконавець, фактично, повинен бути читачем думок. Це як: «ми дізнаємося, коли побачимо».

Формування проекту є творчим і інтегративним процесом. Це вимагає поєднання ідей інтерфейсу з технічними можливостями та з пріоритетами бізнесу [3]. Для цього потрібно бути фахівцем широкого профілю, або співпрацювати з одним чи декількома іншими людьми з відповідних галузей.

Формування проекту – це насамперед дизайнерська робота. Сформована концепція – це дизайн взаємодії, який розглядається з точки зору кінцевого споживача. Вона визначає, що функція повинна робити, як вона працює та де вона вписується в схемі проекту.

На практиці, не потрібно бути програмістом, щоб формувати проект, однак необхідно бути технічно грамотним. Це означає, що потрібно вміти оцінювати: що можливо, що легко, що складно. Знання про те, як працює система, допомагає побачити можливості чи перешкоди для реалізації ідеї, що є стратегічною роботою.

Крім того, розробка рішення вимагає критичного підходу до проблеми: Що потрібно вирішити? Чому це важливо? Що вважати успіхом? Яких клієнтів це стосується? Яка вартість виконання цього підходу, замість якогось іншого?

Насправді, формування проекту є закритим творчим процесом. Тобто, можна на самоті креслити на папері або перед дошкою у колаборації з деякими співробітниками. Відтак, результатом будуть приблизні діаграми, які ніхто поза кімнатою не зможе розтлумачити. При цьому працюючи зі певними співробітниками, дія змінюється: рухи швидші, відверта мова, перехід з однієї перспективної позиції на іншу. Свого роду це приватна попередня робота.

Водночас у класичному підході до двотижневих спринтів існує значний ризик того, що проблеми можуть виявлятися на пізніх

етапах розробки. Тому через жорстку структуру спринтів команди часто можуть переходили до безпосередньої розробки функціональності, однак без достатнього часу на дослідження та експерименти. Власне, такі дії можуть призвести до ситуацій, коли приховані технічні або дизайнерські проблеми можуть стати очевидними лише на фінальних етапах. Це уможливить значне ускладнення процесу їх вирішення та спровокує підвищення витрат.

З іншого боку, більш гнучкий підхід до спринтів, зокрема короткі цикли на початку проекту, дозволяють зменшити цей ризик. Коли проект знаходиться на початковій стадії, важливо в більшій мірі зосередитися на дослідженні та розробці (R&D), ніж на безпосередньому створенні готового продукту. Це дозволяє розробникам краще зрозуміти суть проекту, виявити потенційні проблеми та незгодженості ще на ранніх етапах, перед тим як почати повноцінну реалізацію. Відтак, короткі спринти, що тривають один тиждень, дають можливість часто переглядати напрям розвитку проекту, отримуючи зворотний зв'язок від клієнтів або зацікавлених сторін. У цьому контексті задачі формуються на досить абстрактному рівні, що надає команді гнучкість для експериментів і пошуку найкращих рішень.

На етапі R&D команда повинна зосередитися на вивченні технічних можливостей, тестуванні гіпотез та вирішенні основних питань архітектури проекту. Це дозволяє виявити технічні обмеження та потенційні ризики ще до того, як буде витрачено значні ресурси на розробку функціоналу, який може виявитися недоцільними. Таким чином, цей період коротких спринтів є фундаментальним для того, щоб знизити рівень невизначеності та підготувати команду до більш структурованої і продуктивної роботи на пізніших етапах, коли довжина спринтів збільшиться.

Як тільки основні напрями проекту будуть узгоджені, можна поступово переходити до більш тривалих спринтів, де задачі стають



Рис. 4. Динамічна структура Scrum спринту з урахуванням R&D процесу

Джерело: побудовано авторами за допомогою tldraw

конкретнішими, а команда зосереджується на їх чіткій реалізації (рис. 4).

Такий підхід забезпечує природний баланс між абстракцією і конкретизацією задач, дозволяючи командам спершу зрозуміти й дослідити проєкт, а потім ефективно втілити його в життя.

Висновки. У результаті дослідження впливу Scrum на культуру та продуктивність IT-команд, необхідно акцентувати увагу на результируючих даних. Зокрема, Scrum сприяє розвитку культури відкритої комунікації, співпраці та колективної відповідальності, що позитивно впливає на загальну атмосферу в команді. Однак, одночасно часті зустрічі та інтенсивні комунікації можуть створювати додаткове навантаження на команду, особливо коли це не є необхідним на певних етапах проєкту.

Важливим аспектом є гнучкість у застосуванні тривалості спринтів. Рекомендовано на початкових стадіях проєкту використовувати короткі спринти, щоб дозволити часті узгодження й оперативне виявлення проблем. Зі

збільшенням визначеності проєкту тривалість спринтів може зростати, надаючи команді більше часу для роботи над складнішими задачами. Це допомагає підтримувати баланс між комунікацією та продуктивною роботою.

Подальші дослідження в цьому напрямі можуть бути зосереджені на вивченні альтернативних методів управління комунікацією в рамках Scrum. Зокрема, цікавим є питання про ефективність гнучких підходів до зустрічей, де вони відбуваються за потребою, а не за розкладом. Також варто глибше вивчити взаємозв'язок між рівнем абстракції задач і ефективністю їх виконання в різних фазах проєкту, щоб оптимізувати процес формування завдань для команд.

Крім того, перспективи подальших досліджень повинні включати аналіз впливу гнучкої структури спринтів на різні типи проєктів і команди. Це дозволить розробити більш персоналізовані рекомендації для впровадження Scrum у різних галузях, що допоможе уникати типових помилок і підвищить загальну ефективність застосування методики.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Стан Agile. URL: <https://stateofagile.com/>
2. 17-й звіт про стан Agile. URL: <https://info.digital.ai/rs/981-LQX-968/images/RE-SA-17th-Annual-State-Of-Agile-Report.pdf?version=0>
3. Сінгер Р. Shape Up, Перестаньте бігати по колу та почніть працювати над тим, що має значення. *Basecamp*. URL: <https://basecamp.com/shapeup>
4. Сазерленд Дж., Швабер К. Scrum Guide™. URL: <https://scrumguides.org/scrum-guide.html>

REFERENCES:

1. State of Agile. URL: <https://stateofagile.com/>
2. The 17th State of Agile is Report. URL: <https://info.digital.ai/rs/981-LQX-968/images/RE-SA-17th-Annual-State-Of-Agile-Report.pdf?version=0>
3. Singer R. Stop Running in Circles and Ship Work that Matters. URL: <https://basecamp.com/shapeup>
4. Sutherland J., Schwaber K. Scrum Guide™. URL: <https://scrumguides.org/scrum-guide.html>