

УДК 005.32

DOI: [https://doi.org/10.31521/modecon.V47\(2024\)-10](https://doi.org/10.31521/modecon.V47(2024)-10)

**Михайлов Т. О.**, аспірант кафедри менеджменту та адміністрування, Івано-Франківський національний технічний університет нафти і газу, м. Івано-Франківськ, Україна

**ORCID ID:** 0009-0006-3071-2529

**e-mail:** taras.mykhailov@gmail.com

**Слабінога М. О.**, к.т.н., доцент, доцент кафедри комп'ютерних систем і мереж, Івано-Франківський національний технічний університет нафти і газу, м. Івано-Франківськ, Україна

**ORCID ID:** 0000-0002-7296-0356

**e-mail:** marian.slabinoha@nung.edu.ua

### Особливості самоорганізації робочих груп та її вплив на гнучкість проєкту

**Анотація.** У сучасному управлінському середовищі, яке характеризується постійними змінами та новими викликами, підприємства повинні бути здатними швидко адаптуватися та демонструвати високу гнучкість в управлінні проєктами. Одним із ефективних інструментів для досягнення такої гнучкості є самоорганізація робочих груп. Цей підхід передбачає, що команди можуть самостійно визначати методи та способи виконання своїх завдань, що дозволяє їм швидко реагувати на будь-які зміни в проєктному середовищі.

За останнє десятиліття самоорганізовані команди стали центральною ланкою в інженерії програмного забезпечення, коли їх почали використовувати як відмінну рису Agile-методів. Особливо актуальною є самоорганізація в рамках проєктного менеджменту, де широко застосовується підхід Scrum. Scrum не лише слугує практичним інструментом управління процесами, але й виступає стратегічним напрямом, який сприяє ефективному вирішенню сучасних управлінських завдань.

Самоорганізація в робочих групах полягає у здатності команди взяти на себе відповідальність за свою роботу та організацію процесу. Це дозволяє командам бути більш гнучкими, адаптивними та швидко реагувати на зміни в середовищі проєкту. В контексті Scrum, самоорганізація підтримується трьома основними принципами: прозорість, перевірка та адаптація, які забезпечують ефективну комунікацію та постійне вдосконалення процесів. Прозорість гарантує чіткість та доступність інформації для всіх учасників команди, перевірка дозволяє систематично оцінювати результати роботи, а адаптація дає можливість швидко реагувати на зміни та коригувати плани.

**Ключові слова:** самоорганізація; гнучкість проєкту; scrum, agile; прозорість; перевірка; адаптація.

**Mykhailov Taras**, Postgraduate Student at the Department of Management and Administration, Ivano-Frankivsk National Technical University of Oil and Gas, Ivano-Frankivsk, Ukraine

**Slabinoha Marian**, PhD in Engineering, Associate Professor of Computer Systems and Networks Department, Ivano-Frankivsk National Technical University of Oil and Gas, Ivano-Frankivsk, Ukraine

### Specifics of Work Group Self-Organization and its Impact on Project Flexibility

**Abstract. Introduction.** In today's management environment, characterized by constant change and new challenges, organizations need to adapt quickly and demonstrate high flexibility in project management. An effective tool for achieving such flexibility is the self-organization of work groups. This approach allows teams to independently determine methods and ways to accomplish their tasks, enabling them to quickly respond to any changes in the project environment. Over the past decade, self-organized teams have become central to software engineering, especially as a distinguishing feature of agile methodologies. Self-organization is particularly relevant to project management, where the Scrum approach is widely used. Scrum serves not only as a practical process management tool, but also as a strategic orientation that facilitates the effective resolution of today's management challenges.

**Purpose.** The purpose of this study is to analyze the fundamental principles of the Scrum approach (Transparency, Inspection, and Adaptation) and how they ensure the team's ability to self-organize and contribute to project flexibility. It is essential to examine how these principles affect the team organization, creating conditions for effective collaboration, rapid adaptation to changes, and achieving high results.

**Results.** The analysis of the Scrum core principles (Transparency, Inspection and Adaptation) shows how crucial they are for self-organizing teams. Transparency ensures that information is clear and accessible, fostering effective communication and rapid response to change. Inspection involves regular reviews of the team's progress, ensuring alignment with goals and maintaining quality standards. Adaptation allows the team to modify plans and processes based on new information or evolving requirements, enabling them to remain flexible and responsive in a dynamic project environment.

**Conclusions.** It has been found that the core principles of Scrum help teams to continuously improve and effectively manage project challenges. The principles of transparency, inspection and adaptation embedded in Scrum enable teams to effectively respond to new circumstances, continuously improve work processes and achieve high results in a dynamic business

<sup>1</sup>Стаття надійшла до редакції: 11.09.2024

Received: 11 September 2024

*environment. Self-organization is proving to be not only an effective strategy, but a necessity for achieving high results and competitive advantage.*

**Keywords:** *self-organization; project flexibility; scrum, agile; transparency; inspection; adaptation.*

**JEL Classification:** M15.

**Постановка проблеми.** В умовах сучасної роботи команд з розробки програмного забезпечення, фактор самоорганізації команд та гнучкості реалізації проекту грають ключову роль, оскільки масштаб проєктів, розміри команд, їх розосередженість в різних країнах, та інші фактори роблять неможливим централізоване управління командою. Тому метою дослідження було проаналізувати, як основні принципи підходу Scrum (прозорість, перевірка та адаптація) забезпечують здатність команди до самоорганізації та сприяють гнучкості реалізації проєкту. Необхідно розглянути, яким чином ці принципи впливають на організацію роботи в команді, створюючи умови для ефективної співпраці, швидкої адаптації до змін та досягнення високих результатів. Розширити ідеї, наведені в авторській публікації [1].

**Аналіз останніх досліджень і публікацій.** Одним з принципів Agile-маніфесту [2] є самоорганізація, яка вважається основною ідеєю гнучкої розробки програмного забезпечення. З цим принципом погоджуються багато науковців, які сходяться в думці, що самоорганізовані команди є фундаментальними ознаками гнучкості в розробці програмного забезпечення [3-7].

Згідно з дослідженням Agile Alliance [8], самоорганізовані команди здатні швидше адаптуватися до змін і підвищувати ефективність завдяки постійному вдосконаленню та внутрішнім зворотнім зв'язкам. Інше дослідження, представлене на Agile42 [9], показує, що створення умов для самоорганізації сприяє підвищенню продуктивності та задоволеності роботою.

Дослідження показують, що підвищена автономія сприяє кращій мотивації та залученості працівників, що на пряму впливає на продуктивність. Зокрема, експеримент з вимірюванням нейрофізіології продемонстрував, що автономія в роботі підвищує продуктивність на 5.2%, одночасно покращуючи настрої і задоволеність команд на 31% [10]. Також дослідження показують, що все більше компаній використовують підхід самоорганізованих команд. В 1987 році лише 27% компаній зі списку Fortune 1000 використовували такий підхід. В 1993 році їх кількість зросла до 68%. В 2004 ця цифра складає вже 75% [11].

Дослідження також вказують на позитивний вплив самоорганізації на мотивацію та залученість

співробітників, що в свою чергу сприяє зниженню плинності кадрів та підвищенню загальної ефективності роботи команди. Таким чином, сучасні дослідження однозначно підтримують важливість самоорганізації для успішного та гнучкого управління проєктами.

#### **Формулювання цілей дослідження.**

Метою статті є:

- аналіз впливу принципів Scrum на організацію команд, а саме дослідження того, як основні принципи підходу Scrum (прозорість, перевірка та адаптація) сприяють самоорганізації команд та як це впливає на їх здатність ефективно виконувати завдання;
- визначення ролі прозорості, перевірки та адаптації у підвищенні гнучкості проєктів, а саме оцінка того, яким чином ці принципи впливають на організацію роботи команд, забезпечуючи можливість швидко адаптуватися до змін та досягати високих результатів;
- обґрунтування важливості самоорганізації для успіху проєктів в умовах сучасного динамічного середовища через дослідження того, як самоорганізація, підкріплена принципами Scrum, допомагає командам справлятися з викликами та невизначеністю в процесі розробки програмного забезпечення.

**Виклад основного матеріалу дослідження.** Scrum ґрунтується на теорії емпіричного процесуального контролю або емпіризмі. Емпіризм стверджує, що знання виникає з досвіду і прийняття рішень на основі того, що відомо. Scrum використовує ітеративний, інкрементальний підхід для оптимізації передбачуваності і контролю ризиків [12]. Три стовпи підтримують кожну реалізацію емпіричного процесуального контролю: прозорість, перевірка та адаптація (рис. 1).

Ці "стовпи" повинні працювати циклічно і цикл повинен повторюватись настільки часто, наскільки це необхідно для виявлення відхилень, відкриттів і потенційних можливостей, що виникають в процесі виконання проєкту. Процес повинен відбуватись не один раз на рік чи по завершенню проєкту, а постійно - щодня, щотижня та щомісяця [14, с. 38].

Проаналізуємо як базові принципи підходу Scrum забезпечують здатність команди до самоорганізації та сприяють гнучкості реалізації самого проєкту.

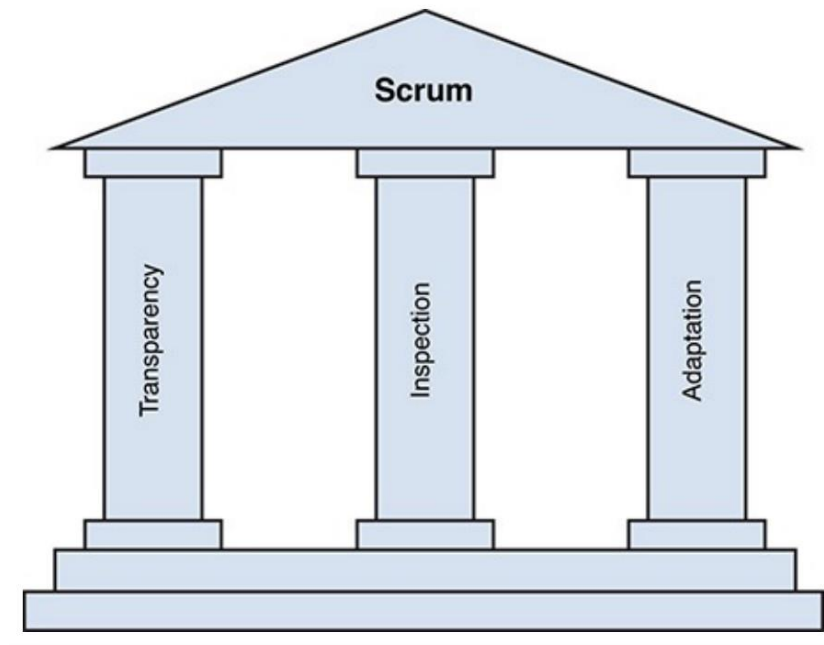


Рисунок 1 – Елементи концепції Г. Хофстеде

Джерело: [13, с. 139]

1. Прозорість (Transparency). Прозорість, як фундаментальний принцип Scrum, гарантує чіткість та доступність інформації для всіх учасників команди. Це означає, що всі завдання та їхні статуси легко доступні для всіх членів команди, дозволяючи кожному учаснику мати чітке уявлення про напрямок роботи та виконання завдань.

Прозорість також передбачає відкритість стосовно потенційних ризиків та викликів, з якими команда може зіткнутися. Це дозволяє всім учасникам розуміти потенційні труднощі та вчасно приймати заходи для їх вирішення.

Прозорість також включає чіткість цілей та вимог. Всі члени команди повинні мати спільне розуміння цілей проекту та вимог до кожного завдання. Це забезпечує узгодженість дій та уникнення непорозумінь.

Згідно з Project Management Body of Knowledge (PMBOK), прозорість визначається як забезпечення чіткості та доступності інформації для всіх учасників проекту [15]. Це є ключовим елементом ефективної комунікації, що дозволяє команді оперативно реагувати на будь-які зміни в середовищі проекту.

Прозорість повинна працювати разом з перевіркою та адаптацією. Припустимо, що організація має всю необхідну інформацію для перевірки (Перевірка) та можливість адаптуватися до змін (Адаптація), але відсутня прозорість. Інформація може бути недоступною або не відображати реальну ситуацію, що може призвести до неправильних висновків під час перевірки і неефективної адаптації до змін [13, с. 140].

Прозорість слугує основою для ефективної комунікації в команді, сприяючи взаєморозумінню

між учасниками і допомагаючи уникнути непорозумінь. Цей принцип у Scrum не лише створює умови для ефективного управління проектами, але й робить команди більш гнучкими та готовими реагувати на будь-які виклики та можливості, що можуть виникнути під час виконання завдань. Прозорість дозволяє всім учасникам проекту бути в курсі поточного стану справ, що підвищує їхню відповідальність та мотивацію. Завдяки прозорості команди можуть швидше реагувати на зміни, ухвалювати обґрунтовані рішення та досягати поставлених цілей з меншою кількістю помилок та затримок.

2. Перевірка (Inspection). У підході Scrum перевірка є невід'ємною частиною процесу розвитку продукту. Цей етап передбачає систематичний огляд результатів команди на кожному етапі розробки, щоб забезпечити відповідність поставленим цілям та вимогам. Регулярна перевірка допомагає командам бути впевненими, що вони рухаються в правильному напрямку та досягають високих стандартів якості.

Перевірка в Scrum проводиться через регулярні ітерації, відомі як "спринти". Кожен спринт триває зазвичай від одного до чотирьох тижнів, протягом яких команда працює над конкретними завданнями та функціоналом. По завершенні кожного спринту результати роботи перевіряються на відповідність з поставленими вимогами. Це включає в себе огляд готових елементів продукту, тестування функціональності та оцінку прогресу.

Під час спринт-огляду (Sprint Review) команда представляє завершену роботу зацікавленим сторонам, отримує зворотній зв'язок та обговорює

можливі покращення. Це дозволяє ідентифікувати можливі проблеми та швидко реагувати на них, що сприяє безперервному покращенню процесу розробки.

Перевірка дозволяє команді оцінити свій прогрес у виконанні завдань та досягненні поставлених цілей. Вона допомагає виявляти будь-які недоліки або невідповідності між очікуваннями та фактичними результатами роботи. Це забезпечує високу якість продукту та своєчасне внесення необхідних змін.

Завдяки регулярним перевіркам команда може:

1) ідентифікувати проблеми на ранніх етапах: Регулярна перевірка дозволяє швидко виявляти та виправляти помилки або дефекти, що зменшує ризик накопичення проблем до кінця проєкту;

2) отримувати зворотній зв'язок: Під час спринт-оглядів зацікавлені сторони можуть надавати свої коментарі та пропозиції, що допомагає команді краще розуміти потреби користувачів та вносити відповідні корективи;

3) підтримувати високий рівень якості: Регулярна перевірка дозволяє команді дотримуватися високих стандартів якості на всіх етапах розробки.

За стандартами Project Management Body of Knowledge (PMBOK), перевірка визначається як процес оцінки продукту або результатів проєкту. Це включає в себе перевірку, аналіз та оцінку, щоб забезпечити відповідність вимогам та досягнення поставлених цілей [15]. PMBOK підкреслює важливість систематичного підходу до перевірки, що включає планування, проведення та документування результатів перевірки.

Перевірка повинна працювати разом з прозорістю та адаптацією. Припустимо, що організації нічого приховувати і вона має стійку комунікацію (Прозорість) та має можливість адаптуватися до змін (Адаптація). Але при цьому, організація не виділяє час на збір та аналіз даних будь-яким послідовним способом. У кінцевому підсумку команда буде відчувати, що їй потрібно змінюватись, але за відсутності вагомих доказів, які це могли б підтвердити, реальні зміни стають важчими [13, с. 140-141].

Перевірка грає важливу роль в забезпеченні якості продукту. Вона дозволяє вчасно виявляти та виправляти будь-які проблеми або дефекти, що можуть виникнути під час розробки. Завдяки систематичному підходу до перевірки, команди можуть забезпечити відповідність поставленим вимогам та досягнення поставлених цілей, що сприяє успішній реалізації проєктів.

3. Адаптація (Adaptation). Адаптація є одним із ключових аспектів підходу Scrum, що передбачає гнучке реагування на зміни та адаптацію до нових вимог проєкту. В контексті Scrum, адаптація

відбувається на різних рівнях – від коригування планів до змін у процесі розробки, що дозволяє команді ефективно реагувати на будь-які зміни та максимально використовувати наявні ресурси.

Адаптація в Scrum означає здатність команди швидко і ефективно змінювати свої плани та підходи на основі нової інформації або зміни вимог клієнта. Це передбачає гнучкість у прийнятті рішень і готовність до змін, що є критично важливими для успішного завершення проєкту в умовах невизначеності.

Припустимо, що організація має точні дані завдяки прозорості та регулярним перевіркам. Однак, якщо ніхто не має повноважень або бажання діяти на основі цих даних, ефективність роботи буде низькою [13, с. 141]. Здатність адаптуватися та реалізовувати необхідні зміни є життєво важливою для успіху будь-якого емпіричного процесу.

Адаптація в Scrum відбувається на різних етапах розробки:

1) щоденні стендапи (Daily Stand-ups): Кожного дня команда обговорює поточний стан справ, обмінюється інформацією про виконану роботу та можливі проблеми. Це дозволяє швидко реагувати на будь-які зміни та вносити корективи в план дій;

2) спринт-ретроспективи (Sprint Retrospectives): Після кожного спринту команда проводить ретроспективу, де оцінює свою роботу та шукає можливості для покращення. Це дозволяє постійно вдосконалювати процес розробки та підвищувати продуктивність;

3) спринт-огляди (Sprint Reviews): Під час спринт-оглядів команда отримує зворотний зв'язок від зацікавлених сторін, що дозволяє вчасно вносити зміни до продукту відповідно до нових вимог або умов.

Адаптація дозволяє команді залишатися гнучкою та ефективною в умовах постійних змін. Основні переваги адаптації включають:

1) швидка реакція на зміни: Команда може оперативно реагувати на зміни вимог клієнта або ринкові умови, що підвищує конкурентоспроможність продукту.

2) покращення якості продукту: Постійна адаптація дозволяє вчасно виявляти та виправляти помилки, що підвищує загальну якість продукту.

3) оптимізація використання ресурсів: Завдяки адаптації команда може ефективніше використовувати наявні ресурси та зосереджуватися на найважливіших завданнях.

За стандартами Project Management Body of Knowledge (PMBOK), адаптація визначається як процес зміни планів або методів управління проєктом на основі зміни умов або вимог [15]. Це передбачає гнучке реагування на зміни та використання нової інформації для досягнення кращих результатів. PMBOK підкреслює важливість

постійного моніторингу та коригування планів для забезпечення успішного завершення проєкту [16].

Адаптація є важливим елементом підходу Scrum, що забезпечує гнучкість та ефективність розробки продукту. Завдяки здатності команди швидко реагувати на зміни та постійно вдосконалювати свої процеси, можна досягти високої якості продукту та успішно завершити проєкт в умовах динамічного ринкового середовища. Адаптація дозволяє команді бути більш готовою до викликів та можливостей, що можуть виникнути під час виконання завдань, забезпечуючи таким чином гнучкість проєкту та його успішну реалізацію.

**Висновки.** Самоорганізація робочих груп на принципах Scrum значно підвищує гнучкість проєкту, забезпечуючи адаптивність та швидку реакцію на зміни. Це стає ключовим фактором

успіху, який сприяє не лише досягненню поставлених цілей, але й посиленню спроможності команди справлятися з сучасними викликами. Принципи прозорості, перевірки та адаптації, закладені в Scrum, дозволяють командам ефективно реагувати на нові обставини, постійно вдосконалювати робочі процеси та досягати високих результатів в умовах динамічного бізнес-середовища.

У динамічному бізнес-середовищі, де зміни відбуваються постійно, здатність команди адаптуватися та працювати ефективно стає вирішальним фактором успіху проєкту. Самоорганізація виявляється не просто ефективною стратегією, але й необхідною для досягнення високих результатів і конкурентних переваг.

### Література:

1. Станьковська І. М., Михайлов Т.О. Планування професійного зростання в умовах сталого розвитку. Сталий розвиток складних соціально-економічних систем в умовах сучасних змін та викликів: матеріали І Міжнародної науково-практичної конференції (м. Івано-Франківськ, 16-17 травня 2024 р.). Івано-Франківськ: ІФНТУНГ, 2024. С. 295-297.
2. Agile-маніфест розробки програмного забезпечення. Agile Manifesto. 2001. URL: <https://agilemanifesto.org/>.
3. Hoda R. Self-Organizing Agile Teams: A Grounded Theory. Open Access Te Herenga Waka-Victoria University of Wellington. 2021. DOI: <https://doi.org/10.26686/wgtn.16985179.v1>.
4. Hoda R., Noble J., Marshall S. Self-Organizing Roles on Agile Software Development Teams. IEEE Transactions on Software Engineering. 2013. Vol. 39(3). Pp. 422-444. DOI: <https://doi.org/10.1109/TSE.2012.30>.
5. Hoda R., Murugesan L.K. Multi-level agile project management challenges: A self-organizing team perspective. Journal of Systems and Software. 2016. Vol. 117(1). Pp. 245-257. DOI: <https://doi.org/10.1016/j.jss.2016.02.049>.
6. McAvoy J., Butler T. The role of project management in ineffective decision making within Agile software development projects. European Journal of Information Systems. 2009. Vol. 18(4). Pp. 372-383. DOI: <https://doi.org/10.1057/ejis.2009.22>.
7. Cohn M. Succeeding with Agile: Software Development Using Scrum. Boston: Addison-Wesley Professional, 2010.
8. Self-organizing teams - Challenges and Strategies. Agile Alliance. 2016. URL: [https://www.agilealliance.org/wp-content/uploads/2016/01/Self-organizing-teams-Challenges-and-Strategies\\_0.pdf](https://www.agilealliance.org/wp-content/uploads/2016/01/Self-organizing-teams-Challenges-and-Strategies_0.pdf).
9. Giuseppe D.S. Self-Organizing Teams: A Complete Guide. Agile 42. 2023. URL: <https://www.agile42.com/en/blog/self-organizing-teams>.
10. Johannsen R., Zak P.J. Autonomy Raises Productivity: An Experiment Measuring Neurophysiology. Frontiers in Psychology. 2020. Vol. 11(963). DOI: <https://doi.org/10.3389/fpsyg.2020.00963>.
11. Smith N., Pazos P. Self-Managing Team Performance: A Systematic Review of Multilevel Input Factors. Small Group Research. 2018. Vol. 49(1). DOI: <https://doi.org/10.1177/1046496417710500>.
12. Schwaber K., Sutherland J. The 2020 Scrum Guide. Scrum Guides. 2020. URL: <https://scrumguides.org/scrum-guide.html>.
13. McGreal D., Jocham R. The Professional Product Owner: Leveraging Scrum as a Competitive Advantage. Boston: Addison-Wesley Professional, 2018.
14. Verwijs C., Schartau J., Overeem B. Zombie Scrum Survival Guide: A Journey to Recovery. Boston: Addison-Wesley Professional, 2021.
15. Стандарт з управління проєктами та Настанова до зводу знань з управління проєктами (Настанова РМВБК). Пенсильванія: Project Management Institute, 2021.
16. Fitsilis P. Comparing PMBOK and Agile Project Management Software Development Processes. Advances in Computer and Information Sciences and Engineering. Springer, Dordrecht. 2007. Pp. 378-383. DOI: [https://doi.org/10.1007/978-1-4020-8741-7\\_68](https://doi.org/10.1007/978-1-4020-8741-7_68).

### References:

1. Stanjkovsjka, I.M., & Mykhailov, T.O. (2024, May 16-17). *Planning professional growth in conditions of sustainable development*. Sustainable development of complex socio-economic systems in conditions of modern changes and challenges, Ivano-Frankivsk National Technical University of Oil and Gas, Ivano-Frankivsk, Ukraine, 295-297.
2. Agile Manifesto (2001). *Manifesto for Agile Software Development*. <https://agilemanifesto.org/>
3. Hoda, R. (2021). Self-Organizing Agile Teams: A Grounded Theory. *Open Access Te Herenga Waka-Victoria University of Wellington*. <https://doi.org/10.26686/wgtn.16985179.v1>
4. Hoda, R., Noble, J., & Marshall, S. (2013). Self-Organizing Roles on Agile Software Development Teams. *IEEE Transactions on Software Engineering*, 39(3), 422-444. <https://doi.org/10.1109/TSE.2012.30>.
5. Hoda, R., & Murugesan, L.K. (2016). Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software*, 117(1), 245-257. <https://doi.org/10.1016/j.jss.2016.02.049>.

6. McAvoy, J., & Butler, T. (2009). The role of project management in ineffective decision making within Agile software development projects. *European Journal of Information Systems*, 18(4), 372–383. <https://doi.org/10.1057/ejis.2009.22>.
7. Cohn, M. (2010). *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional.
8. Agile Alliance (2016). *Self-organizing teams - Challenges and Strategies*. [https://www.agilealliance.org/wp-content/uploads/2016/01/Self-organizing-teams-Challenges-and-Strategies\\_0.pdf](https://www.agilealliance.org/wp-content/uploads/2016/01/Self-organizing-teams-Challenges-and-Strategies_0.pdf).
9. Giuseppe, D. S. (2023, July 3). Self-Organizing Teams: A Complete Guide. *Agile 42*. <https://www.agile42.com/en/blog/self-organizing-teams>.
10. Johannsen, R., & Zak, P. (2020). Autonomy Raises Productivity: An Experiment Measuring Neurophysiology. *Frontiers in Psychology*, 11(963). <https://doi.org/10.3389/fpsyg.2020.00963>.
11. Smith, N., & Pazos, P. (2018). Self-Managing Team Performance: A Systematic Review of Multilevel Input Factors. *Small Group Research*, 49(1). <https://doi.org/10.1177/1046496417710500>.
12. Schwaber, K., & Sutherland, J. (2020). The 2020 Scrum Guide. *Scrum Guides*. <https://scrumguides.org/scrum-guide.html>.
13. McGreal, D., & Jocham, R. (2018). *The Professional Product Owner: Leveraging Scrum as a Competitive Advantage*. Addison-Wesley Professional.
14. Verwijs, C., Schartau, J., & Overeem, B. (2021). *Zombie Scrum Survival Guide: A Journey to Recovery*. Addison-Wesley Professional.
15. Project Management Institute. (2017). *A Guide to the project management body of knowledge (PMBOK® guide)*.
16. Fitsilis, P. (2007). Comparing PMBOK and Agile Project Management Software Development Processes. *Advances in Computer and Information Sciences and Engineering*. Springer, Dordrecht, 378-383. [https://doi.org/10.1007/978-1-4020-8741-7\\_68](https://doi.org/10.1007/978-1-4020-8741-7_68).



Ця робота ліцензована Creative Commons Attribution 4.0 International License