

Лекція 6. Agile та Scrum

Вступ

У даній лекції розглядаються гнучкі підходи до управління проектами, зокрема Agile та Scrum. Ці підходи використовуються у проектах із високим рівнем невизначеності, де вимоги можуть змінюватися протягом усього життєвого циклу. На відміну від класичних моделей, гнучкі методології дозволяють швидко адаптуватися до змін і поступово створювати цінність для замовника.

Поняття Agile

Agile - це підхід до управління проектами, який базується на принципах гнучкої розробки. Його основна ідея полягає у тому, щоб не намагатися детально спланувати весь проект наперед, а рухатися невеликими кроками, постійно отримуючи зворотний зв'язок і коригуючи напрям роботи.

Такий підхід особливо ефективний у ситуаціях, коли неможливо точно визначити всі вимоги на початку. Наприклад, при розробці програмного забезпечення замовник може змінювати свої очікування у процесі роботи.

Цінності Agile

Agile базується на чотирьох ключових цінностях, які визначають пріоритети в роботі команди:

Люди та взаємодія важливіші за процеси та інструменти.

Працюючий продукт важливіший за вичерпну документацію.

Співпраця із замовником важливіша за формальні домовленості.

Готовність до змін важливіша за дотримання початкового плану.

Ці цінності визначають, як саме команда повинна приймати рішення в умовах невизначеності.

Принципи Agile

Принципи Agile деталізують ці цінності. Наприклад, одним із ключових принципів є регулярна поставка працюючого продукту, що дозволяє замовнику бачити результат і вчасно вносити корективи.

Іншим важливим принципом є постійна взаємодія між бізнесом і розробниками. Це означає, що команда не працює ізольовано, а постійно узгоджує свої дії із зацікавленими сторонами.

Також важливими є принципи самоорганізації команди, простоти та постійного вдосконалення процесу роботи.

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Найвищим пріоритетом для нас є задоволення потреб замовника завдяки регулярній та ранній поставці цінного програмного забезпечення.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Зміна вимог вітається навіть у пізніх стадіях розробки. Agile-процеси дозволяють використовувати зміни для забезпечення конкурентної переваги замовника

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Працюючий продукт слід випускати якнайчастіше, з періодичністю від двох тижнів до двох місяців

4. Business people and developers must work together daily throughout the project.

Протягом усього проекту розробники та представники бізнесу мають щодня працювати разом

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

Над проектом мають працювати вмотивовані професіонали. Щоб роботу було зроблено, створіть їм умови, забезпечте підтримку і повністю їм довіртеся

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Безпосереднє спілкування - найбільш практичний та ефективний спосіб обміну інформацією як із самою командою, так і всередині команди

7. Working software is the primary measure of progress.

Працюючий продукт – основний показник прогресу

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Agile допомагає налагодити сталий процес розробки. Інвестори, розробники та користувачі повинні мати можливість нескінченно підтримувати постійний ритм

9. Continuous attention to technical excellence and good design enhances agility.

Постійна увага до технічної досконалості та якості проектування підвищує гнучкість проекту

10. Simplicity - the art of maximizing the amount of work not done – is essential.

Простота як мистецтво скоротити до мінімуму зайву роботу вкрай необхідна

11. The best architectures, requirements, and designs emerge from self-organizing teams.

Найкращі вимоги, архітектурні та технічні рішення народжуються у команд, що самоорганізуються.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Команда повинна систематично аналізувати можливі способи покращення ефективності та відповідно коригувати стиль своєї роботи

Scrum як реалізація Agile

Scrum є однією з найпоширеніших реалізацій Agile. Він базується на ітеративному підході, коли робота виконується невеликими циклами - спринтами.

Кожен спринт має фіксовану тривалість і завершується створенням частини продукту, яка має цінність для замовника.

Такий підхід дозволяє зменшити ризики, оскільки проблеми виявляються на ранніх етапах, а не в кінці проекту.

Принципи Scrum

Scrum базується на трьох ключових принципах:

Прозорість - всі процеси та результати повинні бути зрозумілими і доступними.

Перевірка - необхідно регулярно перевіряти результати роботи.

Адаптація - у разі виявлення проблем необхідно швидко вносити зміни.

Ці принципи забезпечують контроль і гнучкість у процесі виконання проекту.

Ролі в Scrum

Scrum-команда складається з трьох ролей: розробники, власник продукту та Scrum-майстер. Кожна роль має чітко визначені обов'язки і відповідальність.

Розробники

Розробники безпосередньо виконують роботу і створюють інкремент продукту.

Команда є крос-функціональною, тобто включає всі необхідні компетенції для виконання роботи.

Важливою особливістю є самокерованість: команда сама визначає, хто і як буде виконувати задачі.

Наприклад, якщо виникає нова задача, команда самостійно розподіляє її між учасниками без втручання зовнішнього менеджера.

Власник продукту

Власник продукту відповідає за максимізацію цінності продукту.

Його основне завдання - управління беклогом продукту: формування задач, визначення пріоритетів та постановка цілей.

Він виступає як зв'язуюча ланка між бізнесом і командою розробки.

Наприклад, якщо замовник змінює вимоги, саме власник продукту визначає, як ці зміни вплинуть на пріоритети задач.

Scrum-майстер

Scrum-майстер відповідає за правильне застосування Scrum-підходу.

Він допомагає команді працювати ефективно, усуває перешкоди та забезпечує дотримання процесу.

Його роль часто порівнюють із фасилітатором або сервісним лідером.

Наприклад, якщо команда стикається з організаційними проблемами, Scrum-майстер допомагає їх вирішити.

Артефакти Scrum

Артефакти Scrum представляють собою результати роботи та інформацію про прогрес.

До них належать:

- Беклог продукту - список усіх вимог до продукту.
- Беклог спринту - задачі, які команда планує виконати у поточному спринті.

- Інкремент - готова частина продукту.

Кожен артефакт має відповідну ціль, яка дозволяє оцінювати прогрес.

Події Scrum

Scrum включає набір регулярних подій, які забезпечують контроль і координацію роботи команди.

Спринт

Спринт є центральним елементом Scrum.

Це період, протягом якого команда створює інкремент продукту.

Усі інші події відбуваються всередині спринту, що робить його основною одиницею роботи.

Планування спринту

Під час планування команда визначає, що саме буде зроблено у спринті.

Це включає вибір задач із беклогу та оцінку їх виконання.

Наприклад, команда може вирішити реалізувати конкретний функціонал, який є пріоритетним для замовника.

Daily Scrum

Daily Scrum - це коротка щоденна зустріч (до 15 хвилин), яка допомагає синхронізувати роботу команди.

На цій зустрічі обговорюються поточні задачі, проблеми та плани на день.

Це дозволяє швидко виявляти перешкоди і оперативно реагувати.

Огляд спринту

Огляд спринту - це демонстрація результату роботи.

Команда показує створений продукт і отримує зворотний зв'язок від стейкхолдерів.

Це дозволяє переконатися, що проект рухається у правильному напрямку.

Ретроспектива

Ретроспектива спрямована на покращення процесу роботи.

Команда аналізує, що було зроблено добре, а що можна покращити.

Наприклад, команда може вирішити змінити підхід до планування або комунікації.

Правила Scrum

Scrum має ряд правил, які забезпечують стабільність процесу:

- Спринти мають однакову тривалість.
- Після завершення одного спринту одразу починається наступний.
- Усі події є обов'язковими.
- Зміни в беклозі під час спринту не рекомендуються.

Обмеження Scrum

Scrum має певні обмеження.

- Його складно застосовувати у великих командах без додаткових підходів.
- Ефективність залежить від мотивації команди.
- Відсутність загального дедлайну може створювати ризики.
- Також значною проблемою може бути втрата ключових учасників команди.