

Що таке Agile-маніфест, ким і коли він був складений, які проблеми мав вирішити.

Agile-маніфест - це документ, який визначає основні принципи і цінності розробки програмного забезпечення за підходом Agile. Agile визначається як гнучкий, адаптивний підхід до розробки, який спрямований на швидку і ефективну реакцію на зміни вимог клієнта та умови ринку.

Маніфест був складений у 2001 році групою видатних фахівців у сфері розробки програмного забезпечення, які шукали альтернативні методики розробки, більш гнучкі та адаптивні, ніж традиційні підходи. Ця група включає в себе представників різних методологій розробки, таких як Scrum, Extreme Programming (XP), Crystal, адже їхні спільні цінності виявилися ключовими для нового підходу.

Маніфест Agile складається з чотирьох цінностей та 12 принципів. Основні цінності маніфесту Agile виглядають наступним чином:

1. Індивідуали та взаємодія важливіші за процеси та інструменти.
2. Робоче програмне забезпечення важливіше за вичерпну документацію.
3. Співпраця з клієнтом важливіша за угоду про обсяги робіт.
4. Реагування на зміни важливіше за виконання плану.

Принципи Agile орієнтовані на створення більш гнучких, колаборативних і швидких методів розробки, здатних ефективно адаптуватися до змін у вимогах та умовах ринку. Agile прагне розв'язати проблеми, пов'язані з важкими, малоефективними та негнучкими традиційними методологіями розробки, а також зробити процес розробки більш відкритим для змін та взаємодії з клієнтом.

Основні відмінності гнучких технологій від класичних.

Гнучкі технології (які часто визначаються терміном "гнучка розробка" або "Agile розробка") відрізняються від класичних (традиційних) технологій в кількох аспектах, включаючи підхід до процесу розробки, комунікацію та спрямованість на зміни. Ось деякі основні відмінності:

Гнучкість та адаптивність:

- Гнучкі технології: Спрямовані на адаптацію до змін у вимогах та умовах ринку. Процес розробки дозволяє ефективно взаємодіяти з невизначеністю та реагувати на нові вимоги.
- Класичні технології: Зазвичай спираються на жорсткі плани та статичні специфікації, що може ускладнити адаптацію до змін.

Цикли розробки:

- Гнучкі технології: Використовують короткі ітерації або спроби для розробки продукту (наприклад, Scrum має двотижневі спроби).
- Класичні технології: Зазвичай використовують більш довгі фази розробки, такі як визначення вимог, проектування, реалізація, тестування і т. д.

Залучення клієнта:

- Гнучкі технології: Активно залучають клієнта протягом усього процесу розробки, сприяючи змінам та взаємодії.
- Класичні технології: Клієнт зазвичай визначає вимоги на початку проекту, і внесення змін може бути важким.

Комунікація:

- Гнучкі технології: Підкреслюють комунікацію та співпрацю серед учасників проекту, включаючи замовника, розробників і тестувальників.
- Класичні технології: Зазвичай мають більше формальні комунікаційні канали та процеси.

Тестування:

- Гнучкі технології: Зазвичай включають неперервне тестування та автоматизацію тестів.
- Класичні технології: Тестування зазвичай проводиться після завершення фази розробки.

Ці різниці в підходах дозволяють гнучким технологіям бути більш адаптивними та спроможними швидше реагувати на зміни вимог та умовах ринку.

Відомі Вам гнучкі технології.

Існує кілька популярних гнучких технологій, які використовуються в розробці програмного забезпечення. Деякі з найбільш відомих та широко використовуваних гнучких технологій включають:

Scrum:

- Scrum є однією з найпопулярніших гнучких методологій розробки. Вона використовує короткі ітерації, називані "спробами" або "спринтами", які зазвичай тривають від 2 до 4 тижнів. Scrum визначає ролі, такі як Scrum Master, Власник продукту (Product Owner) та розробник, і забезпечує регулярні зустрічі для планування, огляду та ретроспективи.

Kanban:

- Kanban є системою управління роботою, яка базується на візуальному представленні робочого процесу за допомогою дошки з завданнями та картками. Картки переміщуються вздовж дошки, відображаючи поточний стан завдань. Основна ідея - обмеження кількості одночасно виконуваних завдань для покращення продуктивності та управління потоком роботи.

Extreme Programming (XP):

- Extreme Programming - це методологія розробки, яка покладає акцент на якість коду та взаємодію команди. Вона включає практики, такі як парне програмування, тестування на ранніх етапах розробки, часті релізи та взаємодію з клієнтом. XP розвивається на основі коротких ітерацій та постійного вдосконалення.

Feature-Driven Development (FDD):

- Feature-Driven Development - це методологія розробки, яка розглядає розробку програмного забезпечення як набір невеликих функціональних "рис". Кожна команда працює над конкретною функцією, що дозволяє швидше реалізовувати та впроваджувати функції.

Lean Software Development:

- Lean Software Development виникла як адаптація Lean-принципів, які спочатку використовувались у виробництві. Основні принципи Lean включають уникання зайвої праці, оптимізацію вартості та управління потоком роботи. У розробці програмного забезпечення Lean

спрямована на максимізацію цінності для клієнта та мінімізацію витрат.

Ці гнучкі технології використовують різні підходи та практики, але всі вони спрямовані на створення ефективних та гнучких процесів розробки програмного забезпечення. Команди можуть вибрати той метод, який найкраще відповідає їхнім потребам та конкретностям проекту.