12\_03\_23

Домашня робота

Урок 3

Склади порівняльну таблицю найбільш поширених методологій:

|  | Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | Waterfall | 1)Вартість та термін виконання зрозумілі ще до початку робіт. Тому замовник точно знатиме, коли проект завершиться і який бюджет потрібно витратити.  2)Детально структурований план робіт та продумана документація.  Завдяки зручній звітності легко відстежити витрачений час, можливі ризики та ресурси, що використовуються в процесі роботи над проектом.  3)Завдання, які ставляться перед командою зрозумілі та не змінюються протягом усього проекту.  Якість проекту займає першочергове місце, а витрачений час та бюджет відходять на другий план. | 1)Вимоги до проекту закріплюються на початку і не можуть змінюватись до закінчення робіт. Цей факт позбавляє проект гнучкості.  2)Витрачається великий обсяг коштів, часу та ресурсів.  3)Неможливість внесення змін у процесі розробки.  4)Замовник побачить готовий проект тільки після його релізу, при необхідності змін можуть знадобитися додаткові кошти та час.  5)Взаємодія між етапами розроблення повністю відсутня.  У разі використання каскадної моделі продукт тестується після його випуску. Тому в більшості випадків проблеми виявляються лише на етапі тестування. | Космічна галузь, металургійна, машинобудівна.  Це галузі в яких наперед відомі технології, вартість проекту, в пріоритеті є якість кінцевого продукту. Сам продукт складний та потребує великих затрат. |
| 2 | Agile:  Kanban, scrum etc | 1)Внесення необхідних змін та запровадження нового функціоналу може відбуватися незалежно від циклу розробки продукту, що значно підвищує конкурентні переваги готового проекту.  2)Проект складається з коротких та зрозумілих циклів, по закінченні яких клієнт отримує робочий продукт.  3)Гнучкий процес коригування в будь-якій ітерації дозволяє знизити виробничі ризики. Досить швидкий реліз пробної версії для подальших коригувань та тестування.  4)Високий ступінь залучення всіх членів команди та постійна взаємодія із замовником. Він завжди знає, на якій стадії знаходиться проект. | 1)Розрахувати кінцеві витрати практично неможливо – вимоги можуть постійно змінюватись в залежності від особливостей проекту. 2)вимагає великого залучення до процесу і повного занурення в нього, що буває складно.  3)Можливість частого внесення правок може обернутися ризиком у безкінечному вдосконаленні проекту. | Інформаційні технології, ремсервіс, сфера обслуговування (ресторан, піцерія, перукарня, турсервіс)  Тобто там, де перелік вимог остаточно не визначено, а зміни мають бути внесені максимально швидко.  Новий проект є стартапом.  Ніша, для якої розробляє продукт, схильна до постійних змін. |

* На твою думку, чому з’явився Agile-маніфест?

Маніфест з'явився тому, що цього вимагає ринок ІТ технологій.

З одного боку - з боку розробника - кожен ІТ проект це унікальний стартап - неможливо прорахувати наперед з якими труднощами доведеться зіткнутись, скільки це буде коштувати та скільки часу потрібно. З іншої сторони - конкуренція та вимоги споживача - ринок розвивається швидко, вила кількість конкурентов, вимоги постійно змінюються. Високій рівень невизначеності привів до необхідності: ставити вимоги клієнта на перше місце, йти до мети маленькими інкрементним блоками та робити постійні ітерації для перевірки вимог.

* Які проблеми він мав вирішити і чи це вдалося?

1. неповороткість, негнучкість, неможливість внесення змін від етапу до етапу всередині процесса розробки ПЗ та високу ресурсозатратність Waterfall.
2. довгий час на очікування при отриманні робочого продукту
3. неможливість коригування вимог у заказика

Ти – засновник/ця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків. Яку методологію ти обереш для процесу розробки і чому? Відповідь текстово обґрунтуй.

Згідно існуючим методологіям - оберу комбінацію kanban+scrum.

Я планую:

1) згідно вимог сформувати User Stories з Acceptance Criterias (Scrum). Це мій Product Backlog .

2) На Sprint Planning Meeting Product Backlog буде розбитий на задачі (Tickets для программеров - еквівалент для тестеров - Test Cases).

Для кожної задачі будуть виставлені приоритет та оцінка в чол./годинах. Згідно пріоритету та показнику чол./годин об'єдную задачі в спринти, маленьки ітерації (2-4 тижня, Scrum). Таким чином формую Sprint Backlog из Product Backlog (витягуючий принцип Kanban).

3) Кожного дня проводимо Daily Scrum meeting(Scrum)) - та відображаємо на дошці (Kanban) що зробили-робимо-будемо робити(, які є проблеми, як їх можно вирішити. По ітогам спринта робимо Sprint review та Sprint Retrospective(Scrum))

Оскільки це арр, то високий рівень конкуренції буде вимагати від мене:

1. Розпочати запуск продукту якомога швидше - з мінімально робочої версії. Чим швидше я зайду - тим більше в мене буде вірогідність зібрати свою умовно-монопольний прибуток.
2. Інкрементність дозволяє будувати блоками, перевіряти і тестувати кожен блок окремо - це збільшує стійкість ПЗ і знижує в цілому ресурсозатратність
3. Ітеративність моделі дає право постійно перевіряти як йде процес і при необхідності швидко позицинуватись від конкурентів - достроїти опції краще ніж у них, або унікальні - такі які буде чекати ринок (feedback)