**QA – Homework 2**

**Ilya Nechytailo**

SDLC і STLC. Методології розробки ПЗ.

**Перший рівень**— відпрацюй навички на базовому рівні.

Склади порівняльну таблицю найбільш поширених методологій:

|  |  |  |  |
| --- | --- | --- | --- |
| Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі |
| Waterfall(каскадна) | - Чітка послідовність кожного етапу з дотриманням усіх попередніх вимог  - Можливість з боку замовника контролю за виконанням роботи, оскільки по завершенню кожного з етапів – є якісь кінцевий продукт ( мається на увазі, наприклад, на етапі аналізу вимог – маємо як фінальний продукт user requirements, тощо)  -дисциплінований підхід | - Важко або навіть неможливо реагувати на зміни вимог  - ПЗ як кінцевого продукту не має до кінця життєвого циклу | Оскільки, ми пам’ятаємо, що каскадна модель з’явилася перша ще до 2000х років, тоді були часи, коли розвивались сфери пов’язані, з воєнним комплексом, суднобудування, літаки, тощо. Тому, на мою думку, дану модель більш доцільно використовувати для сфер, які не потребують постійних змін під час розробки, які за своєю суттю є більш надійними. Космічні галузі, літаки, тощо. |
| V – модель | - Час витрачений на початку циклу, може зекономити час на пізніх етапах.  - Планування, проектування тестів вже відбуваються за довго до кодування, що є суттєвою перевагою  - Дефекти виявляються на ранніх стадіях, оскільки вже тестування, як раніше, зазначалось виконується раніше кодування. | - Не дивлячись навіть на те, що тестування може відбуватись на ранніх етапах, все ж таки дана модель залишається в категорії негнучких, які тяжко реагують на зміни в вимогах.  - Якщо відбуваються якісь зміни треба оновлювати велику кількість документації | Я вважаю, що дана модель є в цілому дуже схожою на каскадну, через те, що вона, як мінімум, також з’явилась в 1980х роках, тому доцільно сказати, що я б використовував її також в галузі – аерокосмічних компаній, міністерство оборони, урядові проекти, тощо. Саме для цих, бо дана модель також є жорсткою, дисциплінованою, зрозумілою для замовника і чіткою. |
| Спіральна модель | - Безумовно, це покращений аналіз ризиків  - Документація під час процесу розробки  - Можливість внесення змін, навіть на пізніх етапах, бо кожна ітерація проходить оцінку ризиків та вимог, після чого починається новий етап. | - Оскільки під час кожного етапу(ітерації) взаємодіє дуже багато осіб, кожен витрачає час, а час – це гроші. Тому, логічний мінус - ця модель може бути дорогою у використанні.  - Управління та оцінка ризиків – це доволі складний та важливий процес, який вимагає для свого виконання залучення компетентних осіб, яких знайти дуже важко, а без знань та досвіду цей процес втрачає свого значення.  - не підходить для невеликих проектів | На мою думку, дану модель слід впроваджувати для великих проектів, де можна створювати невеликі прототипи і далі їх покращувати та вдосконалювати для створення більшого ПЗ. Дана модель буде гарно працювати, якщо немає чітких вимог або є великі ризики динамічної зміни цих вимог.  Наприклад, ознайомившись зі сторонніми джерелами, я дізнався, що дану модель військові штати США використовували для розробки та модернізації програми модернізації майбутніх бойових систем (SCF). |
| Scrum | - Командна робота, кожен член команди є взаємопов’язаним  - Команда стає автономною  - Ітеративна модель, яка після кожного спринта має якийсь результат  - Гнучка до змін | - Команда повинна віддаватись проекту та заохочуватись для вчасної реалізації  - Обмежена кількість «гравців» в команді  - Залежність від досвіду роботи, бо кожен між собою пов'язаний | Я вважаю, що дану модель доцільно використовувати, наприклад, для розробки застосунків, якихось мінімальних та нескладних ігор. Оскільки – це гнучка модель, яка може підлаштуватись під побажання конкретного замовника або користувача та команда обмежена певною кількістю осіб, тому дуже великі проекти відпадають одразу. |
| Kanban | - Наглядно демонструє команді – де вона має «пробки» та навантаження на якій ділянці виконання завдання, через свою прозорість на дошці  - Легкий метод для розуміння та впровадження в існуючі команди, процес адаптації не займає багато часУ | - Не призначений для довгострокового планування, задачі є сьогодні і зараз, концентрується увага на швидкості виконання кожного таску, процес проходження по всім етапам на дошці | Я гадаю, що даний метод є, як з одного боку, дуже схожим на Скрам, так має і свої індивідуальні відмінності. Тому, його можна використати, як для створення застосунків, але гарно дуже підійде, на мою думку, цей метод для тех. підтримки вже існуючого продукту, аби ефективно та швидко вирішувати поставлені задачі за короткий термін часу, або, наприклад, для оптимізації вже існуючого процесу. |

**Другий рівень**— детальніше заглибся в практику.

1. Виконай завдання попереднього рівня.

2. Напиши розгорнуті відповіді (0,5 - 1 сторінки тексту) на такі два питання:

* На твою думку, чому з’явився Agile-маніфест?
* Які проблеми він мав вирішити і чи це вдалося?
* На твою думку, чому з’явився Agile-маніфест?

Щоб дати чітку відповідь на перше запитання, хочу спершу, більш детально пояснити значення для цього поняття, а саме: це так звані «принципи» та «цінності», якими користуються команди під час розробки ПЗ. Це загальна філософія, стиль управління проектами. Чому б вона могла з’явитись, почнемо аналіз. У попередньому рівні, ми розглядали суттєві слабкі і сильні сторони різних методологій, для перших трьох, а саме тих, які реалізовувались до початку 2000х років – були притаманні риси жорстокості, чіткості, неможливості зміни вимог на пізніх етапах, тощо. Тому, такі методології мали місце для більш урядових та державних організацій.

Початок 21 століття, нові технології, нові технологічні прориви, нові застосунки, нові гаджети, все нове – усе це користувачі, ми люди, отримали протягом останніх 20-25 років. Але давайте дивитись з боку підприємців, айті компаній, стартапів, тощо, усі вони повинні розвиватись та підлаштовуватись під реалії та потреби сьогодення, тому виникає потреба розроблення **такого ведення проекту, розробки ПЗ**, аби воно було ефективним та необхідним, як це зробити – відповідь проста, підлаштуватись під потребу. Але потреби динамічно змінюються і продукт повинен мати змогу оптимізуватись. Саме тому створення та застосування більш гнучких моделей було архіважливо в певний момент. Саме тому, на мою думку, зародився Agile-маніфест. **Люди та взаємодія, працюючий продукт, взаємодія з замовником, готовність до змін** були покладені в основу цього маніфесту, та в основу сьогодення.

* Які проблеми він мав вирішити і чи це вдалося?

Вважаю, що основною проблемою покладеною для вирішення маніфестом було саме - оптимізація роботи та підвищення ефективності ведення проектів в рамках технологічної революції та збільшення запросів та очікувань з боку кінцевих користувачів та замовників.

Чи це вдалося? Питання доволі цікаве, бо, щоб точно відповісти на це питання слід поставити зустрічне: Чи задовольняються наші потреби, як кінцевих користувачів, тих чи інших новітніх застосунків? На мою думку, так, постійний контакт та комунікація з замовником, комунікація з командою розробки та з кінцевими користувачами дозволяють робити акцент саме на якості продукту для певного сегменту людей в рамках перенасиченості ринку існуючих систем та проходження етапу валідації, як для замовника, так і для споживачів, оскільки філософія робить акцент на гнучкості та пристосуванні.

**Третій рівень**— різнобічно опануй тематику уроку.

1. Виконай завдання двох попередніх рівнів.

2. 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.

В ролі засновника даного стартапу, я обрав би методологію – Scrum, з подальшим впровадженням Kanban. Чому саме так? За своєю суттю даний проект не є дуже великим за масштабом – це звичайний застосунок для обміну світлинами. Виходячи з масштабів можна сказати, що команда може бути невеликою, а під невелику команду – ефективно можна впровадити саме модель Scrum. По-друге, слід не забувати, що це стартап, згадуючи, попередні теми, пам’ятаємо, що стартап, як правило, обмежений ресурсами та коштами, тому аби залучити інвестиції, знову ж таки повертаємося до переваг скраму, в сенсі, невелика команда + спринти, які зможуть демонструвати наглядно обсяг виконаної роботи для інвесторів та звітності періодів перед новими інвестиційними раундами. Окрім цього, заздалегідь, провівши аналіз та враховуючи всі ризики, що з’являються нові застосунки і замовники захочуть впровадити якісь технології на пізніх етапах, треба мати таку можливість – бути гнучким, тому знов ж таки плюсик та перевага в цьому випадку для Scrum моделі. Що означає з подальшим впровадженням Kanban? Це застосунок, який буде використовуватись, як певна соц.мережа, тому кінцеві користувачі після релізу волі-не-волі зустрінуться з багами, дефектами продукту, треба буде їх швидко фіксити, аби не втратити лояльність. Тому на моменті тех.підтримки та оптимізації застосунку, я б ввів саме цю методологію, аби підвищити ефективність та час вирішення необхідних завдань задля уникнення втрати обсягу кінцевих користувачів та лояльності, як зазначалося мною вище.