**SDLC та STLC. Методології розробки ПЗ.**

1. **Склади порівняльну таблицю найбільш поширених методологій:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Назва методології** | **Сильні сторони** | **Слабкі сторони** | **Для якої галузі є доцільною** |
| Waterfall | * проста і зрозуміла (є чіткий алгоритм усіх фаз) * зручна для PM (виконання усіх фаз легко проконтролювати) * забезпечує легкість контролю над якістю проекту (прослідковування відповідно до термінів в документації) * відсутні переробки (тому що неможливо повернутися на попередній етап) * має чіткі критерії, які є сталими до кінця проекту (завжди актуальні та правильні специфікації) | * відсутній механізм виправлення помилок (для виправлення помилки потрібно знову пройти всі етапи) * важко або неможливо вносити зміни, внесення змін затратним (передбачає повне визначення критеріїв замовника на початку проекту) * відсутня паралельна робота відразу на кількох етапах (відповідно до методології – працювати можна тільки на одному етапі) * необхідність жорсткого контролю (при недостатньому контролі терміни можуть бути зірваними) * недостатня рол замовника (замовник бере участь в проекті тільки на початку проекту та вкінці – на етапі тестування) | Проекти, пов’язані із життєзабезпеченням, наприклад, автомобільна промисловість, космічна промисловість, авіапромисловість, військова промисловість, медицина |
| V-модель | * Зменшення ризиків (постійне тестування мінімізує ймовірність виникнення помиклки) * Скорочення витрат (ціна всіх стадій легко прогнозується і не змінюється) * Адаптивність для користувачів. (V-модель чітко фіксує і реалізує основні вимоги користувачів до продукту, що розробляється) | * Складність виправлення помилок (Відсутній конкретний механізм вирішення проблем, виявлених на етапі тестування) * Брак гнучкості в процесі розробки (Розробка починається тільки після переходу до наступного етапу) | Галузі, де робота продукту не може бути зупинена (наприклад, галузь авіації) |
| Спіральна | * Усунення ризиків на ранніх етапах реалізації проекту. * Гнучкість на всіх етапах розробки (можливість внесення змін існує протягом усього проекту) | * Тривала і дорога розробка (спіральна модель вимагає великих часових і грошових витрат на здійснення основних принципів і залучення кваліфікованих фахівців. * Висока залежність результату від стадії аналізу (якщо на цьому етапі буде допущена помилка, то зміни проекту потребуватимуть великих витрат) | Галузі, де потрібно враховувати велику кількість ризиків (розробка супутників та військових об’єктів, банківські операції) |
| Scrum | * Ефективна взаємодія між учасниками проекту. (процес прийняття рішень повністю залежить тільки від членів команди. Усі внутрішні процеси регулюють самі розробники. Це дає змогу всім учасникам проекту чітко розуміти свої функції та завдання) * Мінімум контролю і фокус на постійні оновлення (будь-які зміни відбуваються дуже швидко і не вимагають зайвих витрат і витрат) | * Висока вартість розробників (результат сильно залежить від професіоналізму команди. Співробітники повинні мати здібності до самодисципліни та самоконтролю) * Брак гнучкості у великих проектах (втрата навіть одного члена команди стане серйозною проблемою і знизить ефективність реалізації проекту) | Застосовують у стартапах, де необхідно випередити конкурентів і випустити продукт якнайшвидше, і у сфері нових технологій, де результати розроблення продукту не можна передбачити заздалегідь |
| Kanban | * Висока концентрація на поточній роботі (команда фокусується на конкретному завданні і спрямовує всі зусилля на його вирішення. Пріоритетність завдань може змінюватися) * Швидке усунення проблем (усі члени команди можуть відстежувати прогрес і давати зворотний зв'язок, який допомагає оперативно виправляти помилки) * Оптимізація витрат (канбан дає змогу аналізувати і прогнозувати точний час, необхідний для реалізації) | * Не задовольняє вимогам великих команд (метод не призначений для груп чисельністю понад 5 осіб, і команд, де співробітники не знають функції один одного. У таких умовах неможливо ефективно контролювати реалізацію) |
| Agile | * Постійна взаємодія з власником продукту (можна відстежити, чи підходить продукт ринку, що потрібно змінити й одразу внести необхідні зміни) * Ефективність роботи (співробітники самі приймають рішення щодо основних елементів роботи. Документи та інструменти не визначають роботу команди. Усі процеси та структури максимально спрощені. Команда концентрується тільки на найважливіших пріоритетах у розвитку проекту) * Швидке виявлення та усунення помилок (усі можливі проблеми виявляються на ранніх етапах і тут же усуваються. Це також дає змогу уникнути проблем із розбіжністю очікуваного і реального результату) | * Небезпека затягування термінів. (постійний зворотний зв'язок може відтягувати завершення проекту. Необхідно завжди враховувати зміни, що відбуваються, і адаптувати дедлайни під нові завдання) * Складно оцінити кінцеву вартість продукту (усі нові й нові ітерації розширюють бюджет і не дають змоги точно спрогнозувати фінальну суму) |

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

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

Agile-маніфест ставить на перше місце співпрацю, людей, зміни і продукт, який працює. Думаю, що передумовами його виникнення були прецеденти, які не дозволяли зробити продукт кращим, тому що були обмежені стандартними речами, такими як контракти, процеси, точна документація та початковий план. Якщо наприклад, документація не є вичерпною, але продукт досить добре працює, то чи має значення, що документація не є вичерпною? На мою думку, продукт, який працює є важливішим. Чи, наприклад, на початку проекту було складено план реалізації проекту. Під час самої реалізації було помічено, що деякі елементи можна покращити, але план доведеться підкорегувати. То чи справді важливіше дотримуватися плану і не покращити продукт? Думаю, що ні. Важливо змінити план та покращити продукт. Замовник очікує від нас якісний продукт, який відповідатиме його вимогам. І сам же замовник буде задоволений, якщо ми, будучи кваліфікованими працівниками, зможемо дати пропозиції покращення продукту. Тому, тут важливо співпрацювати із замовником, пропонувати нові ідеї та можливості, бо саме від співробітництва із ним залежатиме і наша репутація, яка можливо і впливатиме на майбутні замовлення.

Думаю, що такий маніфест мав на меті вирішити наступні проблеми:

* чітке дотримання початкового плану і негативна реакція на зміни, відсутність гнучкості;
* зосередженість на документації, а не на продукті;
* спілкування із замовником заради умов контракту, а не заради продукту;
* зосередженість на процесах, а не на співпраці команди.

Чи вдалося вирішити ці проблеми? Думаю, що так. З’явилося багато гнучких методологій, в тому числі похідних від Agile, які не мають чітких рамок в етапах реалізації, зосереджені на внутрішній співпраці і навіть в своїй команді можуть мати замовника, який допомагає у створенні продукту. Поява гнучких методологій, таких як Scrum, Kanban, показує нам, що не обов’язково мати чіткий план, достатньо щодня перевіряти завдання та корегувати їх виконання, тому що, не завжди можна передбачити ризики, терміни виконання деяких завдань та проблеми, з якими можна зіткнутися під час реалізації.

1. **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.**

Для процесу розробки виберу методологію Scrum. Тому що:

* Scrum дає змогу швидко випустити тестовий варіант продукту, оцінити його працездатність і потім почати доопрацювання. Це знижує ризики провалу порівняно з підходом, коли продукт нескінченно покращується до запуску;
* методологія Scrum і Agile-принципи заточені на те, щоб вивчати потреби користувачів. Відповідно до цих потреб, команда зможе вдосконалювати продукт;
* є можливість швидко змінювати продукт;
* не потрібна велика команда.