**Домашнє завдання з пройденої теми:**

**Урок №3 SDLC та STLC. Методології розробки ПЗ.**

**1.Виконання завдання першого рівня**

Створимо таблицю найбільш поширених методологій розробки програмного забезпечення та порівнюємо їхні слабкі та сильні сторони.

| № | назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільною |
| --- | --- | --- | --- | --- |
| 1 | waterfall model  (водоспадна модель) | -в процесі виконання цієї моделі відсутні доповнення, або зміни до проекту;  -існує загальний план в якому чітко обговорені терміни розробки, кількість необхідних працівників;  -замовник чітко розуміє на якому етапі йде розробка. | -на теперішній час ця модель вважається застарілою і не використовується у сучасних проектах;  -відсутність гібкості, тобто її специфіка постанови дозволяє однократне виконання кожної з фаз проекту, котрі в свою чергу слідують одне за одним;  -процес тестування в цій моделі розпочинається майже з середини, а максимум припадає майже наприкінці на стадії реалізації проекту;  -не можна переходити на інший етап не зробивши попередній;  -не можна повертатися на попередній єтап розробки щоб зробити якісь зміни; | -доцільним є застосування в галузях де не потребує постійного оновлення;  -до розробки малих проектів;  -розробка медичного ПО;  -розробка робототехнічного обладнання. |
| 2 | V-model  (V-образна модель) | -планування тестування проводиться на ранніх стадіях проекту;  -мінімізація ризиків появи серйозних помилок наприкінці проекту;  -більш гнучка взаємодія з учасниками проекту (можливе вирішення спірних ситуацій з розробником чи клієнтом проекта) | -недостатній аналіз ризиків;  -весь пройдений результат можна побачити тільки з повною реалізацію циклу. | -для задач які припускають більш широке тестове покриття проекту. |
| 3 | Iterative model  (Ітераційна модель) | -можливість розбивки великого проекту на окремі невеликі частки, розробкою однієї з яких може проходити по схемі іншої моделі;  -простота та гнучкість керування інтераціями;  -можливість повторно тестування після виправлення; | -високі накладні розходи на її утримання (це може утворюватись тоді коли при розбивці великого проекту на більш менші доводиться залучати додаткових працівників) | -для реалізації великих проектів. |
| 4 | Rational Unified Process  (Раціональний уніфікований процес) | -розбиває весь процес створення проекту на чотири етапи (в фазі побудова вже проходять вирішальні етапи початкової реалізації проекту, що в свою чергу зменшують ризики);  -постійне забезпечення якості на всіх фазах проекту | -в цілому методологія зазвичай буде складною для її освоєння невеликими командами. | -для реалізації великих проектів. |
| 5 | Ітераційна модель  Scrum model  (Скрам модель) | -в процесі роботи команда обирає певний етап розробки із загального проекту (Product Backlog) та створює спрінт (Sprint Backlog) який триває 2-4 тижня;  -у встановленому проміжку часу команда поглиблено вивчає певну частину проекту. | -при відсутності в моделі Scram Mastera існує вірогідність неконтрольованого затягування розробки всього циклу проекту;  -якщо в команді знаходяться малодосвідчені спеціалісти то спрінт може провалитися;  -не можна вносити зміни в етап спрінту. | -для впровадження реалізації великих проектів які розбиваються на спрінти. |
| 6 | Ітераційна модель  Kanban model  (Канбан модель) | -даний метод немає тимчасових рамок і команда намагається вирішити завдання як набагато швидше;  -дозволяє перерозподіляти ресурси команди та координувати процесами які довгий час не вирішувались. | -не підійде для виконання високонавантажених проектів;  -відсутність тимчасових рамок може призвести до провалу задачі випуску або виправлення продукту. | -для вирішення спірних питань проекту;  - |
| 7 | Ітераційна модель  Spiral model  (Спіральна модель) | -можливість повторного тестування після доработки;  -найбільше підходить в процесі створення нових проектів так як має покращений аналіз ризиків та процесів розробки, завдяки розбивки на 4 сектора;  -можливість внесення змін на всіх етапах розробки;  -реалізація проекту проходить вже на ранніх етапах. | -весь проект має складну суцільну ієрархічну структуру і тому команді розробників та менеджерам проекту буде складно орієнтуватися в ній;  -так як процес розробки може бути довготривалим він з більшою вірогідністю стає занадто дорогим для замовника; | -коли кінцеву перевагу розробки неможливо чітко визначити;  -коли вимоги занадто складні;  -для реалізації великих проектів. |
| 8 | Agile model  (Аджайл модель) | -процес тестування проходить на ранніх етапах;  -використання цієї методології дає наприкінці проекту відмінну якість вихідного продукту так як ви майже на протязі всього часу розробки будите пов'язані з клієнтом;  -методологія дозволяє контролювати прозорість та ефективний зворотній зв'язок протягом усього проекту;  -зниження ризиків завдяки виконання завдань в невеликих спринтах;  -підвищена гнучкість моделі досягається завдяки того, що всі розробники команди взаємодіють один з одним, виконується невеликі всплески, а також постійна підтримка зворотного зв'язку;  -на процесі всього етапу розробки проект безперервно вдосконалюється. | -відсутність чіткого плану розробки;  -постійна загроза переробки більшої частини роботи. | -якщо проект технологічно складний;  -якщо клієнт бажає приймати участь на всіх етапах розробки;  - |

**2.Виконання завдання другого рівня**

Чому з'явився Agile маніфест ? Які проблеми він мав вирішити і чи це вдалося ?

Постійно усталені принципи розробки проектів не могли забезпечувати такі аспекти як швидкість створення програмного продукту та можливість його скорішої адаптації під поточні потреби ринку, тому було прийнято рішення про впровадження методології Agile, яка б одночасно відповідала вимогам як замовника (клієнта) так і команді розробників.

Головною метою створення Agile методології розробки програмного забезпечення було впровадження гнучкої методики, яка б забезпечувала безпосередню участь замовника (клієнта) та команди розробників на всіх етапах реалізації проекту. На початкових стадіях впровадження проекту вже починається активне тестування, що в кінцевому результаті на виході ми отримуємо якісний продукт. Великою перевагою Agile є адаптивний підхід до проекту, що дозволяє вносити зміни навіть на пізніх стадіях.