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

1. Склади порівняльну таблицю найбільш поширених методологій

|  |  |  |  |
| --- | --- | --- | --- |
| **Методология** | **Сильные стороны** | **Слабые стороны** | **Сфера** |
| **Waterfall** | 1) процесс идет быстро, потому что мы не возвращаемся несколько раз к предыдущим этапам | 1) негибкость. Если мы нашли critical bug, его крайне сложно исправить, тк нельзя вернуться к предыдущим этапам разработки | 1) для мелких проектов, тк в них проще на самом старте продумать все до мелочей  2) для safety critical functional проектов (мед., космических и тд), потому что уровень требований к их надежности так высок, что каждый этап продуман и реализован досконально. Здесь нет права на ошибку, тк риски тождественны жизням людей, мировым катастрофам и тд |
| **V-model** | 1. процесс идет быстро, потому что мы не возвращаемся несколько раз к предыдущим этапам 2. на каждом этапе предусмотрено тестирование, а значит риски дефектов уменьшаются | 1) негибкость. Если мы нашли critical bug, его крайне сложно исправить, тк нельзя вернуться к предыдущим этапам разработки | 1) для мелких проектов, тк в них проще на самом старте продумать все до мелочей  2) для safety critical functional проектов (мед., космических и тд), потому что уровень требований к их надежности так высок, что каждый этап продуман и реализован досконально. Здесь нет права на ошибку, тк риски тождественны жизням людей, мировым катастрофам и тд |
| **Итерационная** | 1. процесс идет быстро, потому что мы не возвращаемся несколько раз к предыдущим этапам 2. можно учесть ошибки прошлой итерации в последующей, что уменьшает риск дефектов | 1. негибкость. Если мы нашли critical bug, его крайне сложно исправить, тк нельзя вернуться к предыдущим этапам разработки 2. после каждой итерации не видим результат и систему в целом, что не дает сделать корректную оценку предыдущих итераций | 1. для safety critical functional проектов (мед., космических и тд), потому что уровень требований к их надежности так высок, что каждый этап продуман и реализован досконально. Здесь нет права на ошибку, тк риски тождественны жизням людей, мировым катастрофам и тд |
| **Инкрементальная** | 1. процесс идет быстро, потому что мы не возвращаемся несколько раз к предыдущим этапам 2. позволяет опередить конкурентов за счет скорого первого релиза. Это может быть важно для новых идей | 1. негибкость. Если мы нашли critical bug, его крайне сложно исправить, тк нельзя вернуться к предыдущим этапам разработки 2. изначально выпускается сырой продукт, тк первая версия – это так называемый набросок, костяк | 1. для проектов, требующих скорого выпуска первой версии, для стартапов, тк эта система позволяет первыми выйти в мир с новой идеей |
| **Спиральная** | 1. процесс идет быстро, потому что мы не возвращаемся несколько раз к предыдущим этапам 2. можно учесть ошибки прошлой итерации в последующей, что уменьшает риск дефектов | 1. негибкость. Если мы нашли critical bug, его крайне сложно исправить, тк нельзя вернуться к предыдущим этапам разработки | 1. в банковской сфере, тк в этой методологии после каждого этапа идет оценка рисков, что крайне важно для банковской сферы |
| **Scrum** | 1. много коммуникации в команде, что в разы повышает эффективность работы. Есть возможность обсудить, попросить помощи, подсказать друг другу, объяснить 2. после каждого спринта имеем готовую часть продукта или код, потому что спринты именно это и подразумевают 3. о проблемах узнает вся команда и сразу, что позволяет их быстро решать. Это реализуемо благодаря церемонии scrum daily meeting, к примеру 4. подразумевает skilled команду | 1. не подходит для больших команд, тк в них сложно организовать коммуникацию необходимого уровня 2. зависит от профессионализма рroduct Owner 3. команда должна быть skilled & motivated. Неопытные и немотивированные сотрудники затормозят весь процесс | 1. в современных проектах любого масштаба 2. для стартапов, тк позволяет дорабатывать требования в процессе всей разработки, основываясь на полученном личном и пользовательском опыте 3. для проектов, где критически важна скорость выпуска 1й версии 4. для продуктов той сферы, которая быстро меняется. Методология позволяет изменять продукт и поддерживать на уровне, соответствующем современности |
| **Kanban** | 1. много коммуникации в команде, что в разы повышает эффективность работы. Есть возможность обсудить, попросить помощи, подсказать друг другу, объяснить 2. о проблемах узнает вся команда и сразу, что позволяет их быстро решать. 3. подразумевает skilled команду | 1. не подходит для больших команд, тк в них сложно организовать коммуникацию необходимого уровня 2. команда должна быть skilled & motivated. Неопытные и немотивированные сотрудники затормозят весь процесс | 1. в современных проектах любого масштаба 2. для стартапов, тк позволяет дорабатывать требования в процессе всей разработки, основываясь на полученном личном и пользовательском опыте 3. для продуктов той сферы, которая быстро меняется. Методология позволяет изменять продукт и поддерживать на уровне, соответствующем современности |

**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?

|  |  |
| --- | --- |
| Мне кажется, что созданию Agile манифеста привело развитие и изменение окружающего мира и общества. Мир стал более динамичным. Изменения в тех или иных сферах стали происходить намного быстрее, чем раньше. В таких условиях ранее использованные методологии управления процессом создания ПО стали практически нежизнеспособными. Они не позволяли быстро реагировать на нововведения и оперативно менять продукт так, чтобы он соответствовал свежим тенденциям. Из-за того, что в старых методология возврат к предыдущим этапам разработки невозможен, требуется много времени и сил, чтобы обновить продукт в соответствии с новыми реалиями. Грубо говоря, программный продукт необходимо попросту переписать с нуля. А это не рационально с экономической, временной и тд точек зрения. Также я поинтересовалась, а когда же был создан этот манифест, и выяснила, что ему около 20 лет. Могу предположить, что это именно тот период, когда ИТ стали входить в обыденную жизнь человека в противовес тому, как это было ранее. Раньше ИТ продукты были, скажем так, для избранных: для сферы высоких технологий, космоса, науки, медицины. Позже программные продукты стали входить в повседневную жизнь, а это также предполагает свою специфику и подход к управлению процессами разработки. Основываясь на этих мыслях, могу сделать вывод, что именно реалии современного мира подтолкнули к созданию Agile манифеста с его принципами и подходами. Он дал ту гибкость, которая позволяет оперативно обновлять продукт, поддерживать его в состоянии, соответствующем современности. |  |

**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.

Исходя из условий задания, делаю вывод, что я заинтересована выпустить продукт как можно быстрее. Это новая идея, и я не хочу, чтобы ее перехватили конкуренты. Поэтому мне подойдет методология, позволяющая в кратчайшие сроки выпустить первую версию приложения, пускай даже с минимальным набором функций и в относительно сыром состоянии. Сейчас важнее всего – заявить о себе и присвоить эту идею именно себе. Поэтому в рассмотрение могу взять только инкрементальную методологию и scrum. Именно эти методологии позволяют быстро выпустить первый релиз. При выборе между ними двумя я предпочту scrum по той причине, что с большой долей вероятности требования и ожидания к приложению будут скорректированы. Я буду смотреть на отзывы конечных пользователей, также получу собственный опыт от использования своего приложения, что даст мне возможность продумать, где и как можно улучшить приложение, сделать его более удобным, популярным. Возможно, что-то захочу скорректировать, что-то вовсе удалить, а что-то добавить. И именно scrum позволит мне это сделать. Инкрементальный подход будет не так удобен, потому что по своей сути он не подразумевает возвратов к предыдущим шагам, а значит и внесения корректив. В совокупности этих аргументов мой выбор – scrum.