

Урок 6. Как выбрать гибкую методологию и не ошибиться

Задание:

Вы работаете в команде компании, которую привлекли для разработки мобильного приложения крупной логистической компании. С помощью этого приложения клиенты будут создавать заказы на перевозку грузов.

У заказчика есть требования:

- рассчитать бюджет проекта в самом начале. Это необходимо для корректного бюджетирования деятельности компании. Рассчитать бюджет можно примерно, оценка бюджета может уточняться по мере разработки;
- заказчик готов участвовать в процессе разработки;
- у заказчика есть идея, но нет чёткого представления о том, как сформировать техническое задание. Он хочет обсудить это с вами.

Проанализируйте кейс и предложите подходящую модель разработки. Обоснуйте свой выбор.

Решение:

Выбор модели разработки

Характеристика модели разработки Scrum:

— итеративный подход: Scrum основывается на итерациях, называемых спринтами, которые длительностью обычно составляют 2...4 недели. Каждый спринт представляет собой мини-проект, который включает в себя планирование, разработку, тестирование и обратную связь от заказчика. Это позволяет клиенту активно участвовать в процессе разработки и быстро видеть результаты.

— гибкость в изменениях: заказчик не имеет четкого представления о техническом задании, и Scrum позволяет гибко реагировать на изменения требований и уточнения. Новые или измененные требования могут быть добавлены в следующие спринты.

— прозрачность: Scrum обеспечивает высокую степень прозрачности и коммуникации. Регулярные стендап-встречи (stand-up meetings) позволяют команде и заказчику быть в курсе текущего состояния проекта, прогресса работы и возникающих препятствий;

— сотрудничество: Scrum акцентирует сотрудничество между заказчиком и разработчиками. Заказчик является частью команды разработки и принимает активное участие в планировании и приоритизации задач.

— доставка ценности: Scrum нацелен на представление заказчику рабочей версии программного обеспечения в конце каждого спринта, что позволяет заказчику получать реальные результаты и своевременно вносить корректировки.

Таким образом, в качестве модели разработки применим Scrum, потому что:

- бюджет нужен только примерный - Scrum позволяет нам это сделать;

- заказчик готов участвовать в разработке, следовательно можно выделить ему роль Product owner-a;

- у заказчика есть идея, но нет чёткого представления о том, как сформировать техническое задание. Он хочет обсудить это с нами, с командой профессионалов, с большим опытом работы в этой области - это Scrum.

Модель разработки Scrum, т.е. работа по спринтам, ежедневные собрания и желание заказчика участвовать в процессе разработки, идеально подходит для данного проекта.

Вывод. Исходя из требований заказчика, подходящей моделью разработки для данного случая может быть гибкая методология разработки, такая как Scrum.

Расчет бюджета

Определим стоимость каждого спринта, учитывая затраты на команду разработчиков и другие операционные расходы.

Умножим стоимость одного спринта на количество спринтов, необходимых для завершения проекта, чтобы получить общую оценку бюджета.

Подход к расчету бюджета по Scrum:

- определение проектных требований: в начале проекта определим и задокументируем предварительные требования к продукту или проекту. Это поможет определить, какие работы должны быть выполнены;

- формирование команды: сформируем команду разработчиков, которая будет отвечать за выполнение работы;

- оценка объема работы: Команда разработчиков должна оценить объем работы, необходимой для выполнения каждого требования.

- определение скорости команды: определим скорость команды, то есть количество работы, которое команда может выполнить за одну итерацию (спринт). Для этого можно использовать данные о предыдущих итерациях или произвести первоначальную оценку на основе ранее выполненных проектов.

- расчет продолжительности проекта: поделив оценку объема работы на скорость команды, мы сможем примерно определить, сколько спринтов понадобится для выполнения данного проекта.

Важно! Scrum предполагает гибкость и изменчивость проекта, поэтому бюджет может меняться в процессе выполнения работ.

Используем Kanban-доску

Используем Kanban-доску как визуальный инструмент для отслеживания и управления задачами внутри итераций Scrum.

В этом случае, Kanban-доска может быть разделена на колонки, представляющие различные стадии жизненного цикла задачи (например, "В ожидании", "В работе", "Готово").

Задачи могут перемещаться по колонкам в соответствии с их текущим состоянием.