

# Гибкие методологии (Agile, SCRUM, Канбан и другие).

## Обучение в записи

### Урок 12. Семинар. Как выбрать гибкую методологию и не ошибиться

Условие задания промежуточной аттестации:

Вы находитесь в команде компании-разработчика, которую привлекли для реализации проекта по созданию мобильного приложения для крупной логистической компании. В рамках проекта Вам предстоит создать мобильное приложение, с помощью которого клиенты компании будут создавать заказы на перевозку грузов. У заказчика есть следующие требования:

1. Рассчитать бюджет всего проекта в самом начале – это необходимо для корректного бюджетирования деятельности компании. Рассчитать бюджет можно примерно, оценка бюджета может уточняться по мере разработки продукта;
2. Заказчик готов участвовать в процессе разработки;
3. У заказчика есть только идея, но нет чёткого представления, как сформировать техническое задание на разработку – он хочет обсудить это с Вами.

---

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

### Ход выполнения промежуточной аттестации:

В контексте разработки мобильных приложений методология Agile неопределима, поскольку она решает проблемы, связанные с быстро меняющимися технологиями, ожиданиями пользователей и возможностями устройств.

Используя методологию Agile, команда может сосредоточиться на создании минимального жизнеспособного продукта (MVP) с основными функциями. По мере того как пользователи будут работать с приложением, команда сможет собирать отзывы и итеративно улучшать существующие функции, одновременно работая над новыми возможностями. Такой подход ускоряет выход на рынок и обеспечивает соответствие конечного продукта потребностям пользователей и тенденциям рынка, что позволяет создать более конкурентоспособное мобильное приложение.

Agile разбивает проекты на более мелкие, управляемые итерации или спринты, позволяя командам разрабатывать, тестировать и совершенствовать свою работу постепенно. Такой итерационный подход позволяет командам постоянно совершенствовать продукт, учитывать обратную связь и быстро адаптироваться к изменениям требований или рыночных условий.

#### Scrum

Scrum - это широко используемая Agile-система для управления сложными проектами в разработке программного обеспечения. Это итеративный и инкрементальный подход, подчеркивающий сотрудничество, прозрачность и адаптивность. Scrum предоставляет структурированный процесс для организации работы, управления прогрессом и обеспечения эффективной коммуникации между членами команды и заинтересованными сторонами.

#### Особенности Scrum-подхода

- Возможность разработать новый продукт. Скрам хорошо подходит, когда неизвестно, нужен ли пользователям такой продукт. Вы выпускаете пробный вариант, обучаетесь с помощью обратной связи и делаете следующую версию.
- Фокусировка на пользе для потребителя. Методология Scrum и agile-принципы заточены на то, чтобы изучать потребности пользователей, улучшать их опыт использования продукта. А не на то, чтобы сделать продукт в срок и согласовать его с заказчиком.

- Возможность менять продукт. Работая короткими циклами, можно предлагать пользователям продукты почти по требованию и быстро адаптироваться к новым условиям.
- Снижение риска за счет тестирования. Скрам позволяет быстро выпустить тестовый вариант продукта, оценить его работоспособность и затем приступить к доработке. Это снижает риски провала по сравнению с подходом, когда продукт бесконечно шлифуется до запуска.
- Вовлеченные сотрудники. Команда вовлекается в работу, потому что она работает только над одним проектом и видит результат от своих усилий.

#### События:

- Планирование спринта: Встреча в начале каждого спринта, на которой команда выбирает элементы из бэклога продукта для разработки и создает план доставки.
- Ежедневный скрам: Короткое ежедневное совещание, во время которого команда разработчиков делится достигнутым прогрессом, обсуждает препятствия и координирует усилия.
- Обзор спринта: Встреча в конце каждого спринта, на которой команда демонстрирует выполненную работу заинтересованным сторонам, собирает отзывы и при необходимости обновляет бэклог продукта.
- Ретроспектива спринта: Встреча после обзора спринта, на которой команда анализирует спринт, выявляет области для улучшения и обсуждает практические шаги по улучшению процесса разработки в последующих спринтах.

Итеративный подход Scrum позволяет командам предоставлять рабочие части продукта через регулярные промежутки времени, обычно каждые 2-4 недели. Это позволяет командам собирать обратную связь, корректировать и быстро адаптироваться к изменяющимся требованиям или условиям рынка, что приводит к более эффективному управлению проектом и более качественному программному обеспечению.

Для улучшения рабочего процесса можно также дополнительно воспользоваться методом **Kanban**, который поможет визуализировать процесс работы, помогая определить проект на разных этапах производства.

С помощью досок Kanban можно добавить обратную связь с клиентами в рамках процесса, что повышает эффективность и продуктивность всего рабочего процесса, генерируя необходимые результаты и держа все заинтересованные стороны в курсе событий.

---

Ссылка на репозиторий:

[https://github.com/olgashenkel/GeekBrains-specialization-ELECTIVES/tree/main/10.%20Flexible%20methodologies%20\(Agile%2C%20SCRUM%2C%20Kanban%20and%20others\)](https://github.com/olgashenkel/GeekBrains-specialization-ELECTIVES/tree/main/10.%20Flexible%20methodologies%20(Agile%2C%20SCRUM%2C%20Kanban%20and%20others))