

(Слайд 1: Титульный)

Добрый вечер, меня зовут Сюй Хаоюй, группа ИУ5И-32М. Сегодня я расскажу на тему «Современные методологии разработки программного обеспечения».

(Слайд 3: От Waterfall до Agile)

Классическая каскадная модель, предполагает строгую последовательность этапов. Её основные проблемы — это невозможность вернуться на предыдущий этап и позднее обнаружение ошибок.

Agile делает акцент на гибкости, быстром реагировании на изменения и тесном взаимодействии с заказчиком, *что стало ответом на недостатки Waterfall*.

(Слайд 4: Манифест Agile: четыре ценности)

Манифест Agile основан на четырёх ключевых ценностях:

1. Люди и взаимодействие важнее процессов и инструментов.
2. Работующий продукт важнее исчерпывающей документации.
3. Сотрудничество с заказчиком важнее условий контракта.
4. Готовность к изменениям важнее следования первоначальному плану.

(Слайд 5: Scrum)

Scrum — это лёгкий фреймворк, основанный на эмпирическом контроле процесса. Его три столпа — это прозрачность, проверка и адаптация. *Scrum помогает командам совместно решать сложные адаптивные задачи*.

(Слайд 6: Пять точек контроля)

Процесс в Scrum строится вокруг пяти основных церемоний или точек контроля:

1. Спринт — это фиксированный по времени цикл работы.
2. Планирование спринта — где определяется цель и задачи на спринт.
3. Ежедневный скрам — короткая встреча для синхронизации команды.

4. Обзор спринта — для демонстрации результата заказчику.

5. Ретроспектива спринта — для анализа и улучшения процессов команды.

(Слайд 8: Роли в Scrum)

Теперь рассмотрим три ключевые роли в Scrum.

В Scrum традиционная роль менеджера проекта упраздняется. Его обязанности распределяются между тремя ролями, *что позволяет уменьшить командование и контроль, повысив гибкость и ответственность команды*.

В Scrum есть три ключевые роли:

Владелец продукта — это визионер, отвечающий за ценность продукта.

Скрам-мастер — это слуга-лидер, фасилитатор и коуч команды.

Команда разработки — это самоорганизующаяся, кроссфункциональная единица, *которая сама решает, как достичь цели спринта* и несёт коллективную ответственность за результат.

(Слайд 10: Лидерство и Фасилитация)

В Scrum мы видим три аспекта лидерства:

Продуктовое лидерство через видение и приоритизацию

Процессное лидерство через устранение препятствий

Техническое лидерство через самоорганизацию и мастерство

Скрам-мастер выступает главным фасилитатором, обеспечивая эффективное проведение всех событий Scrum. Ценность фасилитации в том, что каждый голос слышен, *что ведёт к обоснованным решениям и повышению эффективности*.

(Слайд 11: Kanban)

Kanban — это метод, пришедший из японского производства. Он основан на визуализации рабочего процесса на доске с колонками, *что повышает прозрачность и контроль*.

(Слайд 12: Преимущества Kanban)

Основные преимущества Kanban:

Гибкость: Позволяет оперативно добавлять новые задачи.

Эффективность: Ограничение работы в прогрессе снижает перегрузки.

Прозрачность: Клиенты видят прогресс в реальном времени.

Простота внедрения: Не требует радикальных изменений процессов.

(Слайд 13: DevOps)

DevOps — это культура и набор практик, объединяющих разработку и эксплуатацию.

Его основная цель — ускорить выпуск программного обеспечения, *сохраняя его стабильность и надёжность*.

(Слайд 14: CI/CD)

Основу технической реализации DevOps составляют:

Непрерывная интеграция (CI) — автоматическая сборка и тестирование кода.

Непрерывное развёртывание (CD) — автоматизация вывода продукта в продакшен.

Но главное в DevOps — это культура сотрудничества между командами.