нетология

Жизненный цикл ПО

Алексей
Метляков

Алексей Метляков

DevOps Engineer

OpenWay

Алексей Метляков

План занятия

1. Что такое жизненный цикл ПО?
2. Waterfall
3. Agile и Lean
4. Итоги
5. Домашнее задание

Что такое жизненный цикл ПО?

Что такое жизненный цикл ПО?

Жизненный цикл ПО – ряд событий (этапов), происходящих с системой от ее создания до дальнейшего использования и изъятия из эксплуатации.

Для чего нужно знать жизненный цикл ПО?

- для понимания общей картины развития продукта
- для определения стратегических и тактических целей команды
- для лучшего управления продуктом
- для обеспечения согласования требований заказчика

Что такое жизненный цикл ПО?

Жизненный цикл ПО состоит из следующих этапов:

- разработка;
- тестирование;
- внедрение;
- сопровождение.

Что такое жизненный цикл ПО?

Разработка состоит из следующих этапов:

- составление требований;
- проектирование;
- дизайн;
- разработка.

Что такое жизненный цикл ПО?

Тестирование состоит из следующих этапов:

- функциональное тестирование;
- интеграционное тестирование;
- нагрузочное тестирование;
- показ.

Что такое жизненный цикл ПО?

Внедрение состоит из следующих этапов:

- выделение инфраструктуры;
- настройка рабочего окружения;
- внедрение продукта;
- тестирование работоспособности.

Что такое жизненный цикл ПО?

Сопровождение состоит из следующих этапов:

- мониторинг работоспособности приложения;
- работа с инцидентами;
- вывод из эксплуатации.

Что такое жизненный цикл ПО?

Жизненный цикл ПО состоит из следующих этапов:

Разработка	Тестирование	Внедрение	Сопровождение
Составление требований	Функциональное тестирование	Выделение инфраструктуры	Мониторинг работоспособности
Проектирование	Интеграционное тестирование	Настройка рабочего окружения	Работа с инцидентами
Дизайн	Нагрузочное тестирование	Внедрение продукта	Вывод из эксплуатации
Разработка	Показ	Тестирование работоспособности	

Waterfall

Waterfall

Waterfall – последовательная методология разработки и тестирования ПО. Концепция заключается в последовательном выполнении всех шагов для получения конечного продукта.

К плюсам можно отнести:

- стабильность и предсказуемость результата;
- простота отслеживания текущего состояния разработки;
- полноценная документация.

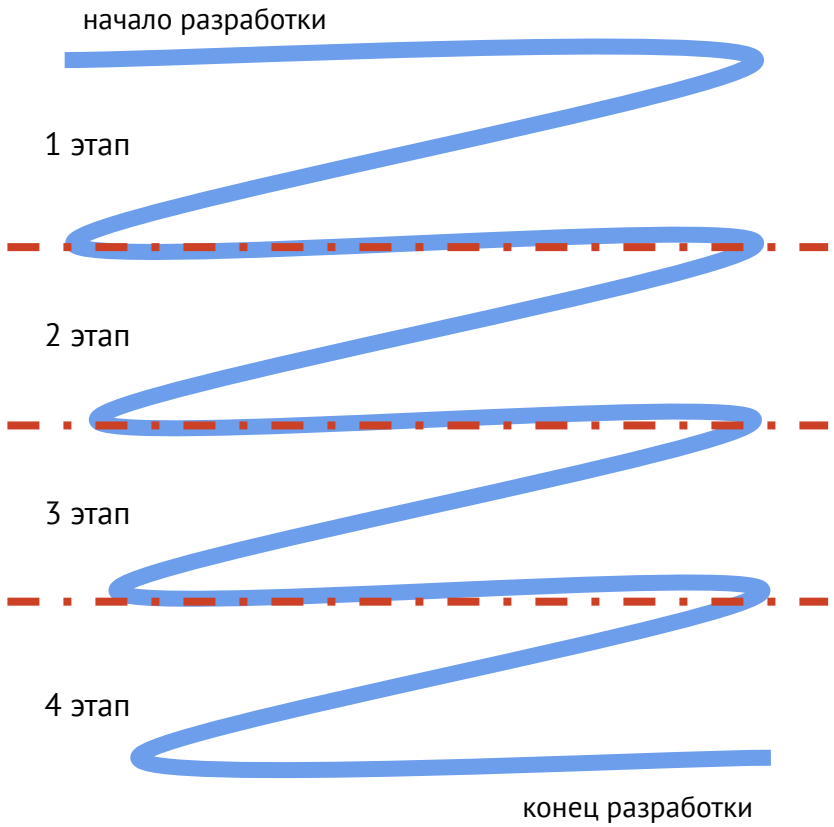
Минусы:

- медленная разработка;
- выявление ошибок разработки на поздних этапах.

Waterfall

Спиральная модель – последовательная методология разработки и тестирования ПО. Концепция заключается в последовательном выполнении шагов, которые разделены на несколько этапов.

К плюсам добавляется возможность отслеживать промежуточный результат разработки в момент завершения очередного витка итерации.



Waterfall

Инкрементная модель – последовательная методология разработки и тестирования ПО. Основная концепция waterfall дополняется версионированием. Иными словами, у вас появляется возможность создавать релизы (минорные и мажорные) и патчи на баги.

Agile и Lean

Agile

Agile – набор из 12 принципов гибкой разработки ПО:

- Приоритет команды в удовлетворении заказчика;
- Приветствуется постоянное изменение требований;
- Рабочий продукт следует выпускать как можно чаще;
- На протяжении всего проекта команда должна работать вместе;
- Над проектом должны работать мотивированные профессионалы;
- Непосредственное общение;
- Работающий продукт - показатель прогресса;
- Вся команда должна иметь возможность поддерживать постоянный рабочий ритм;
- Постоянное внимание к совершенству и качеству повышает гибкость;
- Простота необходима;
- Самые лучшие требования рождаются у самоорганизующихся команд;
- Команда должна систематически анализировать возможности улучшения эффективности работы и корректировать свой стиль.

Agile

На «бумаге» **Agile** выглядит как решение всех проблем:

- гибкость жизненного цикла ПО;
- лёгкость в принятии изменений;
- уменьшение бюрократического процесса в разработке;
- быстрота доставки новых версий ПО;
- улучшение качества кода без потерь в наращивании функционала.

Agile

В жизни с **Agile** часто возникают проблемы:

- документация не успевает за кодовой базой проекта;
- повышается сложность сопровождения продукта;
- количество задач на разработку постоянно растёт;
- большинство разработчиков занимаются просмотром PR;
- в маленьких командах на разработчиков ставят задачи тестирования;
- времени на создание окружения автотестирования не хватает;
- в рамках agile у продукта гибкость требований влечёт за собой гибкость окружения;

Lean

Исторически, **бережливое производство** пришло из промышленности. Первая компания – **toyota**.

Постулаты:

- устранение отходов;
- расширение прав и возможностей работников;
- уменьшение запасов;
- повышение производительности.

Фактически они стремились заключать контракты и собирать ровно столько машин, сколько у них заказали, не создавая сверх меры.

Lean

Lean – принцип разработки, основанный на бережливом производстве, смысл которой в устранении помех производству.

Устранение происходит в два этапа:

- **Анализ** (Проводится анализ всех процессов в алгоритме работы команды, определяются уязвимости, траты ресурсов и времени, «бутылочные горлышки»);
- **Внесение изменений** (Предлагается альтернатива процессам алгоритма. Альтернатива не обязательно должна быть инновационной, он должна просто улучшить конечный результат. Все альтернативы считаются возможными и исследуются на практике. На основе практического подхода выбирается, какая из альтернатив лучшая, и используются в постоянном алгоритме разработки).

Lean

Муда, мура, мури – три концепции управления «отходами» разработки.

Муда – отходы, иными словами, последствия ошибок. Примеры:

- бесполезные данные, которые отправляются из приложения в БД;
- несоответствие ТЗ;
- безразличие к рабочему процессу;
- непрошеная самодеятельность в готовом проекте;
- накопленные запросы без движения.

Lean

Мура – причины появления муда (мусора). Например, неравномерная по времени нагрузка на команду.

Мури – необоснованные сложности в работе (нецелесообразность). Примеры:

- разработчик занимается тестированием;
- у команды нет необходимых инструментов для разработки;
- плохо поставленные задачи в ТЗ;
- для выполнения задачи требуется сделать множество звонков и согласований.

Основной смысл **Lean** – убрать все три категории, при этом нужно бороться с причиной, а не следствием.

Lean

Основные принципы **Lean**:

- ликвидировать мусор;
- создавать только то, что нужно;
- все задачи должны исходить из желаний потребителя;
- сделать правильно с первого раза;
- расширять возможности команды;
- создать культуру постоянного совершенствования.

Lean

Кайдзен – (**кай** – изменения, **дзен** – хорошо) метод, позволяющий команде непрерывно предлагать и тестировать идеи по улучшению своей работы, работы всей команды и программного продукта.

Пять характеристик кайдзен:

- равноправное **взаимодействие** всех участников команд и прямая **коммуникация** между ними;
- индивидуальная **дисциплина;**
- здоровое **моральное состояние** всей команды и каждого участника;
- кружки **качества;**
- **предложения** по улучшению всего в рамках работы.

Гибкие методологии

Глоссарий

- **Product Manager** - тот, кто владеет продуктом;
- **Project Manager** - тот, кто управляет кастомизацией продукта для одного клиента;
- **Team Lead** - программист, занимающийся управлением командой;
- **QA** - тестировщик;
- **Front** - разработчик интерфейсов;
- **Back** - разработчик «подкапотного» кода продукта
- **Ops** - сопровождение продукта
- **Delivery** - тот, кто отвечает за доставку продукта до клиентов
- **Bug** - найденное отклонение от ожидаемого поведения продукта
- **Task** - задача на изменение продукта
- **Feature** - задача на разработку нового функционала продукта
- **Epic** - верхнеуровневое описание какой-то общей цели
- **Stories** - описание результата, который хочется достичь
- **Workflow** - движение задач разных видов по статусам
- **Backlog** - общий пул задач

Kanban

Kanban

Kanban – один из способов управления потоком работы.

Концептуально он разделяет все задачи по столбцам статусов, в рамках которых команда понимает, какие задачи сейчас:

- находятся в разработке;
- нужно выполнить в ближайшее время;
- уже сделаны;
- невозможно выполнить.

Kanban

При использовании **Kanban** могут возникнуть трудности:

- неявный переход между статусами;
- неуправляемый поток задач;
- централизованное управление;
- неясный план;
- пренебрежение качеством.

Scrum

Scrum

Scrum – ещё один способ управления рабочим процессом команды. В рамках данной идеологии, все задачи выполняются в рамках **sprint**.

Sprint – промежуток времени, за который команда гарантирует выполнение определенного набора задач. Обычно **sprint** завершается создание релиза продукта.

- задачи для **sprint** выбирает команда на планировании;
- задачи выбираются исходя из **сложности;**
- сложность вычисляется **опытным** путем;
- первое завершение **sprint** у новой команды обычно не соответствует ожиданиям.

Scrum

При использовании **scrum** тоже могут возникнуть трудности:

- происходит накопление задач в **backlog;**
- задачи переносится из одного **sprint** в следующий и не выполняются;
- график **выгорания** задач стремится к нулю.

Итоги

Итоги

Сегодня мы узнали, что:

- **Жизненный цикл ПО** – перечисление этапов проектирования, разработки, тестирования, сопровождения ПО до процесса вывода из эксплуатации;
- существует несколько принципов разработки ПО: **Watefall**, **Agile**, **Lean;**
- на основе принципов **Agile** и **Lean** созданы методы ведения разработки и сопровождения:
  - **Kanban;**
  - **Scrum.**

Домашнее задание

Давайте посмотрим ваше домашнее задание.

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

нетология

Задавайте вопросы и пишите отзыв о лекции!

Алексей Метляков

Алексей Метляков