МЕТОДЫ И СРЕДСТВА ИСИТ 2024

ЛЕКЦИЯ 3

Классические модели (методологии) разработки программного обеспечения.

СОДЕРЖАНИЕ ЛЕКЦИИ

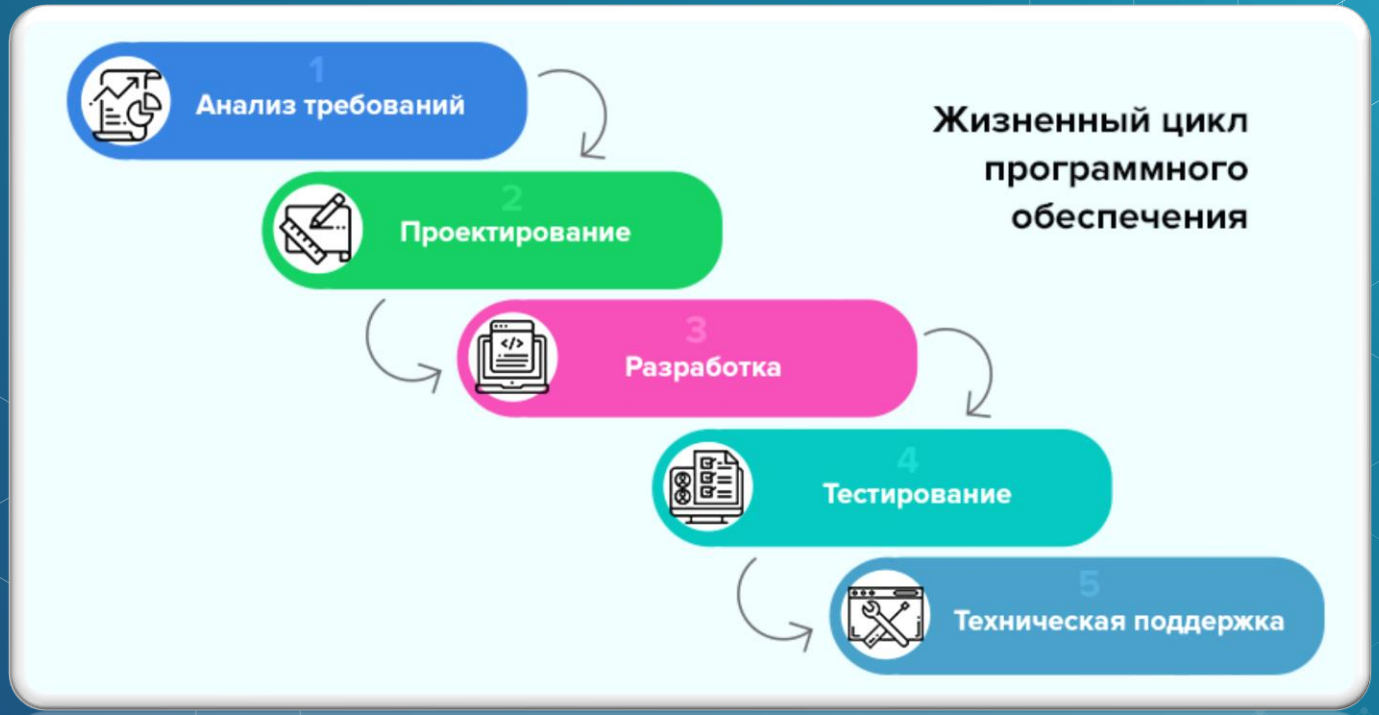
- Стадии разработки программного средства.
- Основные модели разработки ПО.
- Водопадная модель, стадии водопадной модели разработки ПО.
- Модель с верификацией, инкрементная модель разработки ПО.
- Итеративная и спиральная модель разработки ПО.

ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (В ОБЩЕМ ВИДЕ)

МОДЕЛИ ОБЕСПЕЧЕНИЯ ЖИЗНЕННОГО ЦИКЛА (МЕТОДОЛОГИИ)

- Waterfall model (водопадная модель).
  - V-model (модель с верификацией).
  - Incremental model (инкрементная модель).
  - Iterative model (итеративная модель).
  - Spiral model (спиральная модель).
- Agile model (гибкая модель) – тема следующей лекции

МОДЕЛЬ ВОДОПАДА



ПРИНЦИПЫ МОДЕЛИ ВОДОПАДА

- Документы и инструкции - это важно, всё должно быть зафиксировано.
- Следующий этап работы не начинается, пока не закончится предыдущий.
- Пропускать этапы нельзя.
- Если требования к продукту изменились после согласования - переписываем ТЗ.
- Нельзя возвращаться на предыдущий этап, чтобы что-то изменить.
- Нет итераций, есть один общий процесс создания продукта.
- Выявлять и исправлять ошибки - только на этапе тестирования.
- Клиент не участвует в создании продукта после постановки ТЗ.

ПРИМЕНИМОСТЬ МОДЕЛИ ВОДОПАДА

Когда использовать водопадную модель?

- Требования известны, понятны и зафиксированы. Противоречивых требований не имеется.
- В проектах с критичностью групп C и D.
- Нет проблем с доступностью программистов нужной квалификации.
- В относительно небольших проектах.

МОДЕЛЬ, ОСНОВАННАЯ НА ТЕСТИРОВАНИИ (V-model)

ПОЯСНЕНИЯ ПО ЭТАПАМ ТЕСТИРОВАНИЯ

- ***Модульное тестирование*** (юнит-тестирование) - процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы, наборы из одного или более программных модулей вместе с соответствующими управляющими данными, процедурами использования и обработки. Примеры утилит тестирования: QTest, QSignalSpy.
- ***Интеграционное тестирование (I&T)*** - Интеграционное тестирование в качестве входных данных использует модули, над которыми было проведено модульное тестирование, группирует их в более крупные множества, выполняет тесты, определённые в плане тестирования для этих множеств, и представляет их в качестве выходных данных и входных для последующего системного тестирования. Пример ПО для тестирования: VectorCAST/C++.

ПОЯСНЕНИЯ ПО ЭТАПАМ ТЕСТИРОВАНИЯ

- ***Функциональное тестирование*** **-** это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает. Пример ПО тестирования: Ranorex.
- ***Приемо-сдаточное тестирование (UAT)*** **-** это тип тестирования, выполняемый конечным пользователем или клиентом для проверки / принятия системы программного обеспечения перед перемещением приложения в производственную среду. UAT выполняется на заключительном этапе тестирования после выполнения функциональных, интеграционных и системных испытаний.

ПРИМЕНИМОСТЬ V-МОДЕЛИ

1. Особенностью модели можно считать то, что она направлена на тщательную проверку и тестирование продукта, находящегося уже на первоначальных стадиях проектирования.

2. Стадия тестирования проводится одновременно с соответствующей стадией разработки, например, во время кодирования пишутся модульные тесты.

Когда использовать V-модель?

- В проектах с критичностью групп C и D.
- Если требуется тщательное тестирование продукта.
- Для малых и средних проектов, где требования четко определены и фиксированы.

ИНКРЕМЕНТНАЯ МОДЕЛЬ

КЕЙС ИНКРЕМЕНТНОЙ МОДЕЛИ

Заказчик решил, что хочет запустить соцсеть, и написал подробное техническое задание. Программисты предложили реализовать основные функции - страницу с личной информацией и чат. А затем протестировать на пользователях, «взлетит или нет».

Команда разработки показывает продукт заказчику и выпускает его на рынок. Если и заказчику, и пользователям социальная сеть нравится, работа над ней продолжается, но уже по частям.

Программисты параллельно создают функциональность для загрузки фотографий, обмена документами, прослушивания музыки и других действий, согласованных с заказчиком. Инкремент за инкрементом они совершенствуют продукт, приближаясь к описанному в техническом задании.

ПРИМЕНИМОСТЬ ИНКРЕМЕНТНОЙ МОДЕЛИ

1. В инкрементной модели полные требования к системе делятся на различные сборки.

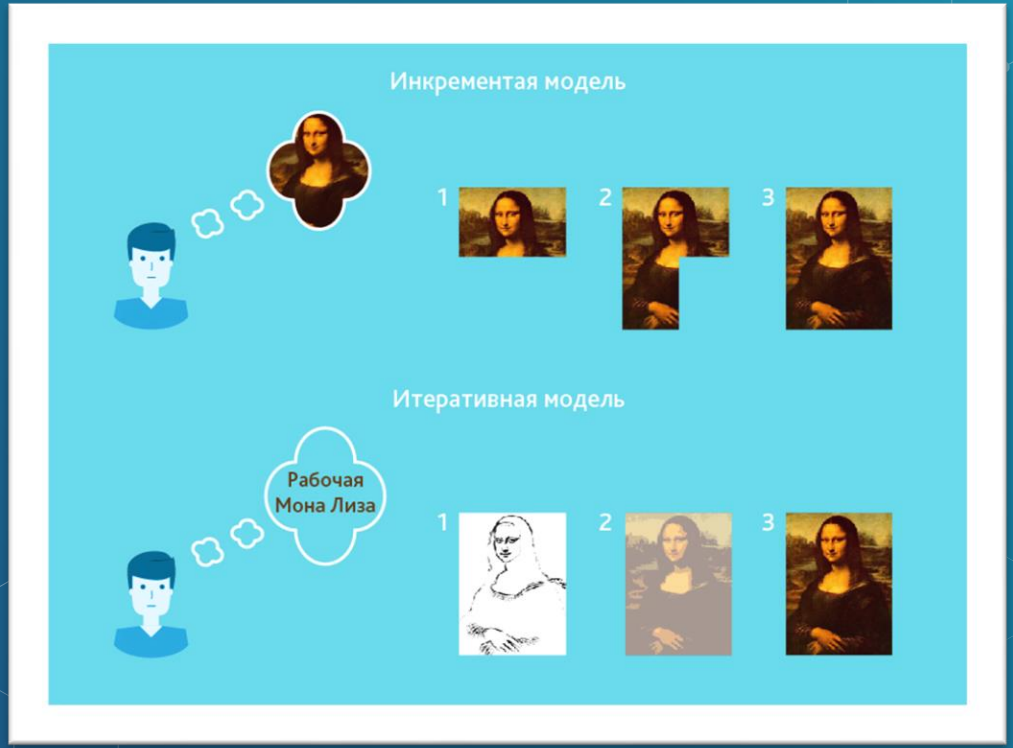
2. Имеют место несколько циклов разработки, и вместе они составляют жизненный цикл «мульти-водопад».

3. Процедура разработки по инкрементной модели предполагает выпуск на первом большом этапе продукта в базовой функциональности, а затем уже последовательное добавление новых функций, так называемых «инкрементов».

Когда использовать инкрементную модель?

- Когда основные требования к системе четко определены и понятны. В то же время некоторые детали могут дорабатываться с течением времени.
- Требуется ранний вывод продукта на рынок.
- Есть несколько рисковых feature или целей.

ИТЕРАТИВНАЯ МОДЕЛЬ



КЕЙС ИТЕРАТИВНОЙ МОДЕЛИ

Заказчик решил, что хочет создать мессенджер. Разработчики сделали приложение, в котором можно добавить друга и запустить чат на двоих.

Мессенджер «выкатили» в магазин приложений, пользователи начали его скачивать и активно использовать. Заказчик понял, что продукт пользуется популярностью, и решил его доработать.

Программисты добавили в мессенджер возможность просмотра видео, загрузки фотографий, записи аудиосообщений. Они постепенно улучшают функциональность приложения, адаптируют его к требованиям рынка.

ПРИМЕНИМОСТЬ ИТЕРАТИВНОЙ МОДЕЛИ

1. Модель не требует для начала полной спецификации требований.

2. Создание начинается с реализации части функционала, становящейся базой для определения дальнейших требований. Этот процесс повторяется.

Когда оптимально использовать итеративную модель?

- Общие требования к конечной системе в целом понятны.
- Проект большой или очень большой.
- Есть несколько рисковых feature или целей.

СПИРАЛЬНАЯ МОДЕЛЬ

КЕЙС СПИРАЛЬНОЙ МОДЕЛИ

Заказчик решил, что хочет сделать такую систему, и заказал программистам реализовать управление чайником с телефона. Они начали действовать по модели «водопад»: выслушали идею, провели анализ предложений на рынке, обсудили с заказчиком архитектуру системы, решили, как будут её реализовывать, разработали, протестировали и «выкатили» конечный продукт.

Заказчик оценил результат и риски: насколько нужна пользователям следующая версия продукта — уже с управлением телевизором. Рассчитал сроки, бюджет и заказал разработку. Программисты действовали по водопадной модели и представили заказчику более сложный продукт, разработанный на базе первого. Заказчик подумал, что пора создать функциональность для управления холодильником с телефона. Но, анализируя риски, понял, что в холодильник сложно встроить Wi-Fi-модуль, да и производители не заинтересованы в сотрудничестве по этому вопросу. Следовательно, риски превышают потенциальную выгоду. На основе полученных данных заказчик решил прекратить разработку и совершенствовать имеющуюся функциональность, чтобы со временем понять, как развивать систему «Умный дом».

ПРИМЕНИМОСТЬ СПИРАЛЬНОЙ МОДЕЛИ

1. Артефактом каждого витка спирали является новая версия продукта.

2. После каждого витка обязательно оценивается целесообразность продолжения поддержки продукта.

Когда оптимально использовать спиральную модель?

- Требует особое внимание хеджированию рисков проекта.
- Планы по постепенному расширению функционала продукта после релиза.

СПАСИБО!

ВАШИ ВОПРОСЫ, ПОЖАЛУЙСТА?