

Раздел 6. Классические методологии разработки ПО

Каскадная модель (англ. waterfall model, иногда переводят, как модель «Водопад») — модель процесса разработки программного обеспечения, в которой процесс разработки выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки.

Следуя каскадной модели, разработчик переходит от одной стадии к другой **строго последовательно**. Сначала полностью завершается этап «определение требований», в результате чего получается список требований к ПО. После того как требования полностью определены, происходит переход к проектированию, в ходе которого создаются документы, подробно описывающие для программистов способ и план реализации указанных требований. После того как проектирование полностью выполнено, программистами выполняется реализация полученного проекта. На следующей стадии процесса происходит интеграция отдельных компонентов, разрабатываемых различными командами программистов. После того как реализация и интеграция завершены, производится тестирование и отладка продукта; на этой стадии устраняются все недочёты, появившиеся на предыдущих стадиях разработки. После этого программный продукт внедряется и обеспечивается его поддержка — внесение новой функциональности и устранение ошибок.

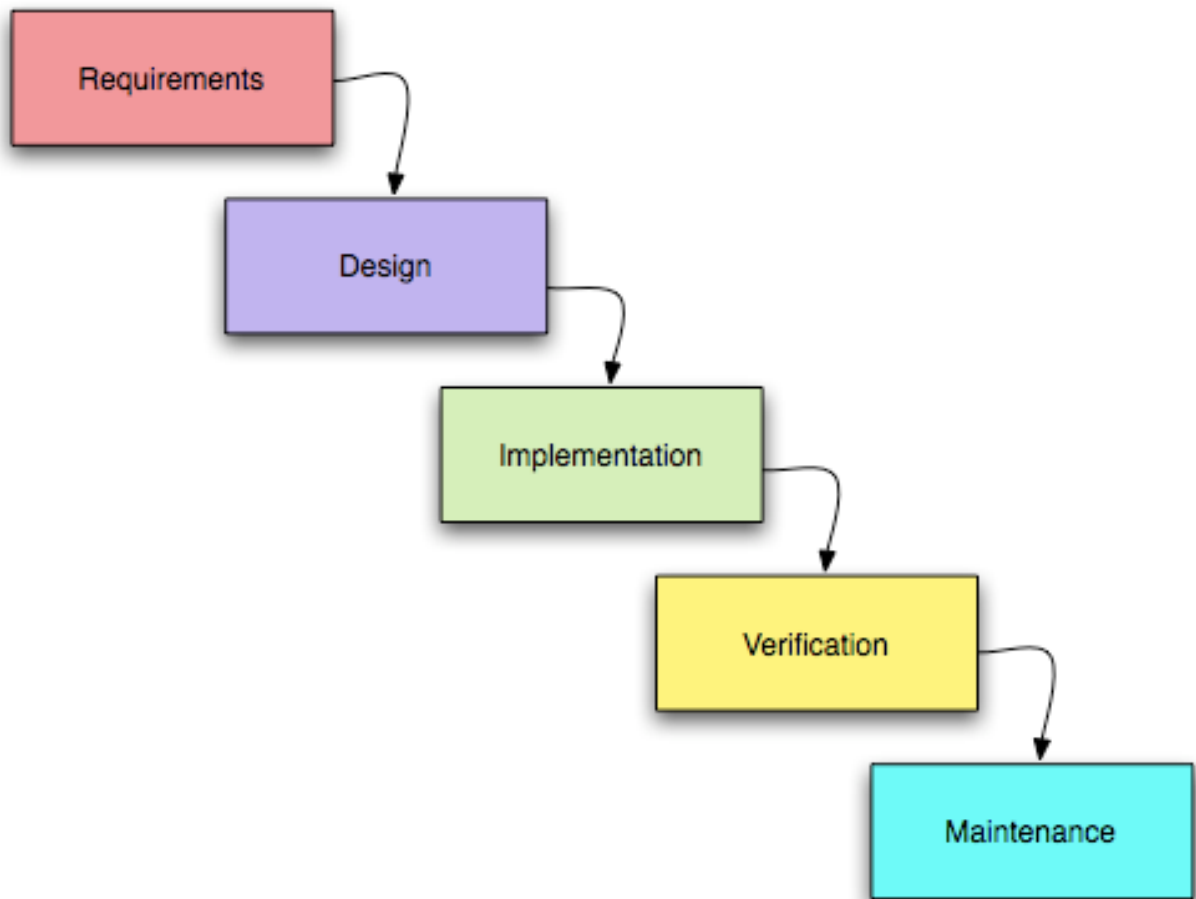


Рис. 6.1 Waterfall

Для классической модели разработки программного обеспечения выделяют следующие этапы:

1. *Анализ требований проекта.* Определяются программные требования для информационной предметной области системы.
2. *Проектирование.* Разрабатывается и формулируется логически последовательная техническая характеристика программной системы. Детализация системы.
3. *Реализация ПО.* Воплощение полноценного проекта.
4. *Тестирование продукта.* Тестовая эксплуатация продукта
5. *Интеграция системы.* Включает установку и официальную приёмку продукта.
6. *Поддержка.* Предоставление технической помощи по продукту после запуска в коммерческую эксплуатацию.

Эта модель подразумевает строго последовательное и однократное выполнение каждой фазы проекта. Переход от одной фазы к другой возможен только после успешного

завершения предыдущего этапа. Каждый этап подразумевает детальное планирование и полную корректность результата этапа.

Такие жёсткие ограничения последовательности позволяет построить процесс разработки, который максимально прозрачен и удобен для Заказчика.

6.1 ПРОЕКТ И ДОКУМЕНТАЦИЯ

Обратная сторона «медали» данного метода, это необходимость поддержки и постоянной актуализации документации разработки продукта. Любое изменение необходимо обязательно согласовывать с Заказчиком. А не достаточный уровень проработки требований несёт за собой увеличение бюджета и сроков проекта, которые довольно сложно оценить.

На сегодняшний день водопадная модель разработки ПО практически не используется из-за малой гибкости модели. Однако её продолжают использовать из-за высокой прозрачности разработки. Благодаря высокому уровню формализации, управлять таким проектом значительно проще. Принято считать, что каскадная модель разработки снижает риски и вносит ясность в процесс разработки, когда над проектом работает несколько десятком человек.

Каскадная модель подходит при разработке сложных и больших проектов и систем со строго определённой функциональностью. Использовать при разработке больших гос.заказов или научных разработках. Использовать данную методологию для разработки бизнес-приложений крайне нежелательно.

Мета информация

- Количество процессов — 6
- Команда — десятки человек

Минусы

- Watrefall проект должен постоянно иметь актуальную документацию.

Обязательная актуализация проектной документации. Избыточная документация

- Очень негибкая методологии
- Может создать ошибочное впечатление о работе над проектом (например, фраза «45% выполнено» не несёт за собой никакой полезной информации, а является всего лишь инструментов для менеджера проекта)

- У Заказчика нет возможности ознакомиться с системой заранее и даже с «Пилотом» системы

- У Пользователя нет возможности привыкать к продукту постепенно
- Все требования должны быть известны в начале жизненного цикла проекта
- Возникает необходимость в жёстком управлении и регулярном контроле, иначе проект быстро выйдет из графиков

- Отсутствует возможность учесть переделку, весь проект делается за один раз

Плюсы

- Высокая прозрачность разработки и фаз проекта
- Чёткая последовательность
- Стабильность требований
- Строгий контроль менеджмента проекта
- Облегчает работу по составлению плана проекта и сбора команды проекта
- Хорошо определяет процедуру по контролю качества

Применение

- Для средних и больших проектов, где задействованы десятки программистов и несколько разных команд проекта.

- Проекты, в которых требования и границы прозрачны и точно известны в начале жизненного цикла проекта.

- Тем самым, каскадная модель подразумевает, что **переход от одной фазы разработки к другой происходит только после полного и успешного завершения предыдущей фазы, и что переходов назад либо вперёд или перекрытия фаз — не происходит.**

RUP методология разработки

RUP (Rational Unified Process) – методология разработки ПО, созданная компанией Rational Software

Основные принципы:

1 — ранняя **идентификация и непрерывное (до окончания проекта) устранение основных рисков**

2 — **концентрация на выполнении требований заказчиков к исполняемой программе**

3 — **ожидание изменений в требованиях, проектных решениях и реализации** в процессе разработки

4 — **компонентная архитектура**, реализуемая и тестируемая на ранних стадиях проекта

5 — **постоянное обеспечение качества** на всех этапах разработки проекта (продукта)

6 — работа над проектом в **сплочённой команде**, ключевая роль в которой принадлежит **архитекторам**

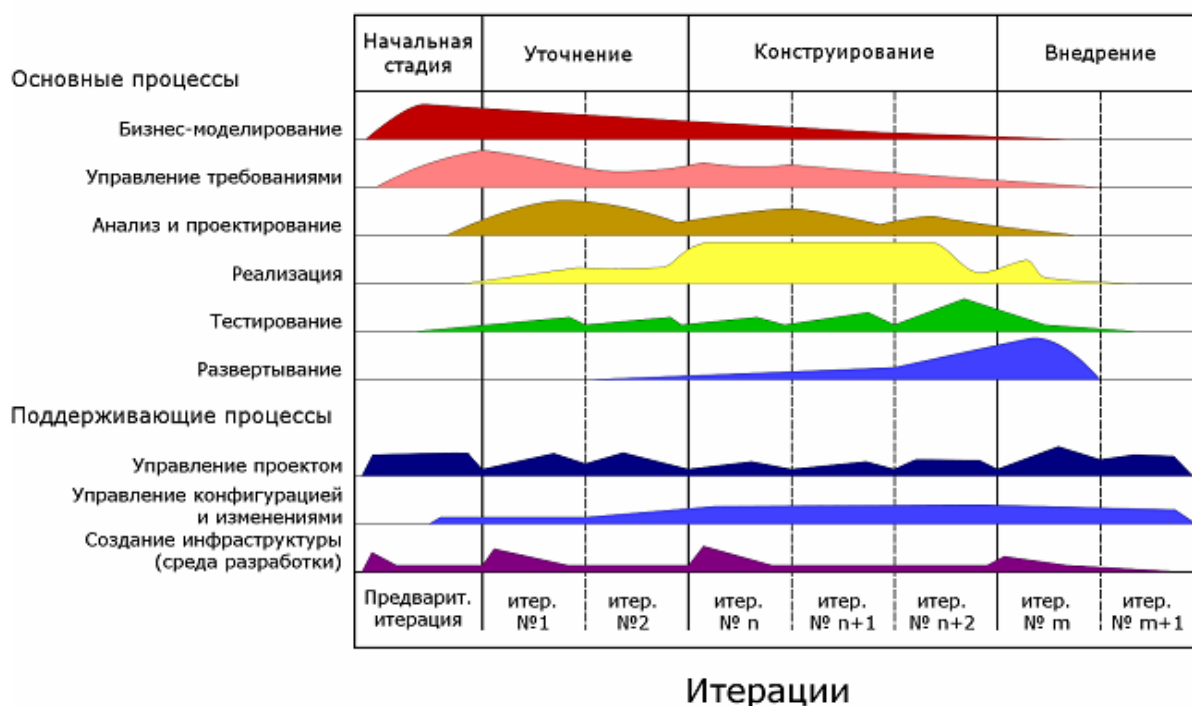
Процессы и стадии RUP

RUP использует **итеративную** модель разработки. В конце каждой итерации (в идеале продолжающейся от 2 до 6 недель) проектная команда должна достичь запланированных на данную итерацию целей, создать или доработать проектные артефакты и **получить промежуточную, но функциональную версию конечного продукта**.

Полный жизненный цикл разработки ПО состоит из 4 этапов:

Рабочие процессы

Стадии



Методология RUP

1. Начальная стадия (Insertion)

В фазе начальной стадии:

Формируются **видение и границы проекта**.

Создается **экономическое обоснование** (business case).

Определяются *основные требования, ограничения и ключевая функциональность продукта.*

Создается *базовая версия модели прецедентов.*

(Прецедент соответствует отдельной функциональности системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе)

Оцениваются риски.

При завершении начальной фазы *оценивается достижение этапа жизненного цикла цели*, которое предполагает соглашение заинтересованных сторон о продолжении проекта.

2. Уточнение (Elaboration)

В фазе «Уточнение» производится **анализ предметной области и построение исполняемой архитектуры**. Это включает в себя:

Документирование требований (включая детальное описание для большинства прецедентов).

Спроектированную, реализованную и оттестированную исполняемую архитектуру.

Обновленное экономическое обоснование и более точные оценки сроков и стоимости.

Сниженные основные риски.

Успешное выполнение фазы разработки означает *достижение этапа жизненного цикла архитектуры*

3. Построение (Construction)

В фазе «Построение» происходит *реализация большей части функциональности продукта*. Фаза Построение завершается *первым внешним релизом системы и вехой начальной функциональной готовности.*

4. Внедрение (Transition)

В фазе «Внедрение» создается *финальная версия продукта* и передается от разработчика к заказчику. Это включает в себя программу *бета-тестирования*, обучение пользователей, а также *определение качества продукта*. В случае, если качество не соответствует ожиданиям пользователей или критериям, установленным в фазе Начало, фаза Внедрение повторяется снова. Выполнение всех целей означает достижение *вехи готового продукта* и завершение полного цикла разработки

