Гусев Виталий Евгеньевич

09-335

Waterfall

**Слайд 2.** Waterfall (или каскадная модель) — это модель для разработки продуктов, которая позволяет решать задачи по принципу последовательного плана без возврата на предыдущие этапы.

Графически ее можно представить в виде водопада.

Американский ученый-информатик Уинстон Уокер Ройс придумал и описал ее еще в 1970 году, а в 1976 году ученые Томас Белл и Томас Тэйер дали ей название.

**Слайд 3.** Основным принципом работы является то, что каскадная модель разработки ПО состоит из последовательных циклов, порядок которых нельзя менять. Также нельзя начинать новый до завершения предыдущего.

Отсюда и сравнение с водопадом: каждый этап создания продукта, словно поток воды, продолжает предыдущий и не может начаться, пока прошлый не завершился.

 А также важно, чтобы все этапы работы были задокументированы.

**Слайд 4.** Процесс разработки при использовании каскадной водопадной модели выглядит как поток, последовательно проходящий следующие фазы:

1. Анализ и определение требований проекта

Команда собирает требования к будущему продукту, после чего необходимо составить подробное техническое задание. На данном шаге также планируется график работ и происходит оценка возможных рисков.

2. Проектирование

На этом этапе готовят документы, в которых подробно описывается для программистов способ и план реализации сформулированных ранее требований. На этой стадии команда создает прототип и дизайн-макеты, а когда они будут готовы, подключаются разработчики.

3. Реализация

После завершения проектирования, программистами выполняется воплощение полноценного проекта. На этом этапе разработчики пишут код продукта согласно утвержденному плану, макетам и требованиям, работая четко по ТЗ.

4. Тестирование продукта

Когда код готов, начинается тестирование и отладка ПО. На этой стадии устраняют все недочеты, которые появлялись на предыдущих стадиях разработки.

5. Эксплуатация и поддержка

На заключительном этапе проект передают заказчику, продукт запускается в коммерческую эксплуатацию и обеспечивается его поддержка, включающая внесение новой функциональности и устранение ошибок.

**Слайд 5.** Последовательность процессов, соблюдение сроков, выполнение задач в каскадной модели лучше всего отображает диаграмма Ганта (a Gantt Chart) или горизонтальная гистограмма. Она состоит из блоков, расположенных на двух осях. По горизонтали — задачи, по вертикали — время, затраченное на их выполнение. На диаграмме можно проследить, какие задачи входят в проект и кто за них отвечает, а также продолжительность каждого этапа.

**Слайд 6.** Допустим, вы строите быстровозводимый дом ― дачу в Подмосковье, чтобы выбираться туда на лето. Времени мало, максимальный бюджет — три миллиона рублей. Земля в вашей собственности, все документы в порядке. Срок строительства двухэтажного коттеджа, как сообщает застройщик, — от 25 дней. Все этапы известны и определены, а материалы закуплены.

Для начала перечислим каждый этап, затем дату начала и завершения. Первые две задачи офисные специалисты делают только в рабочие дни, далее работа переходит к строительной бригаде, которая трудится каждый день. Срок проекта — 28 дней. Чтобы показать весь проект на нашей диаграмме, представим, что этап поддержки длится неделю. В жизни [срок обнаружения ненадлежащего качества работ гораздо больше](http://www.consultant.ru/document/cons_doc_LAW_9027/0a4ea608362f47bcede62d4c1f35777b1b99493e/).

**Слайд 7.** Затем построим диаграмму Ганта. Мы использовали [smartsheet](https://ru.smartsheet.com/" \t "_blank), но это можно делать в Excel или просто на бумаге.

По горизонтали перечисляем этапы строительства, по вертикали указываем начало и конец каждого. Теперь диаграмма иллюстрирует принцип Waterfall: этапы идут один за другим, следующий начинается только тогда, когда заканчивается предыдущий. Это логично: невозможно возвести хороший фундамент и покрыть крышу без инженерно-исследовательских работ и четкого плана дома. Мы также видим, из каких этапов состоит проект, какие задачи входят в каждый этап и сколько времени они занимают.

**Слайд 8.** Безусловно, в этой методологии есть свои плюсы и минусы.

**Плюсы:**

- Наличие инструкций и правил по всему процессу. Работа начинается с подробного анализа требований заказчика и того, как будет реализован проект. Планы, этапы и процессы утверждаются заранее, фиксируются в документах и вопросов не вызывают. Исполнителю нужно просто им следовать.

- Определенность в сроках и бюджете**.** Стоимость продукта и сроки сдачи проекта рассчитаны и утверждены в самом начале и не меняются в процессе.

- Отсутствие дополнительных трат на коммуникацию в команде**.** Даже если придет новый разработчик или тестировщик, понять задачу и приступить к работе получится быстро: для всех процессов есть описанные правила.

**Минусы:**

- Отсутствие гибкости**.** Невозможно предусмотреть все проблемы в проекте заранее. Из-за жесткой последовательности этапов недочеты станут известны только в конце проекта, придется делать дополнительные итерации и начинать работу заново, а это новые затраты и лишние рабочие часы.

- Заказчик не допускается до разработки и тестирования. Он не может комментировать макеты или прототипы и видит результат только в конце проекта. Если изменились требования или условия, заранее учесть это невозможно.

- Проблемы всплывают только при тестировании. Сделать часть работы и сразу протестировать или совместить разработку и тестирование, чтобы найти уязвимости, нельзя. Тестирование начинается после окончания разработки, поэтому часто недостатки обнаруживаются слишком поздно.

**Слайд 9.** Сначала Waterfall использовали в создании любого программного обеспечения, но потом появилась модель Agile и водопад стал применяться реже. Команда, работающая по [Agile](https://practicum.yandex.ru/blog/metodology-agile/), двигается к цели спринтами: каждую неделю подводит итоги и показывает промежуточные варианты. В Waterfall есть чёткие сроки и техническое задание, в которые нужно вписать проект.

**Слайд 10.** В этой таблице мы собрали их основные отличия гибких методологий от каскадной модели.

Разберëм разницу между моделями на примере создания веб-приложения для книжного интернет-магазина.

**Waterfall**

В начале работы клиент прописывает свои ожидания от дизайна и функциональности. Например, хочется отдельный раздел с акциями, который будет открываться при входе в приложение, личный кабинет, где покупатель сможет отслеживать заказ, и блог — электронный журнал о книжных новинках.

Все требования учитывают в техническом задании, которое клиент согласовывает на старте, и формируют финальную документацию для работы над проектом. После этого оценивают объём работ и сроки.

**Возможные проблемы:**

- Нужно что-то доработать. Например, клиент на старте не учёл в ТЗ программу лояльности, чтобы начислялись баллы за покупку. Приходится переделывать личный кабинет, из-за чего срок проекта увеличивается.

- На финальных стадиях разработки клиент просит добавить функцию как на сайте конкурента. Это возможно только в следующей версии, потому что в начале проекта это не зафиксировали.

**Agile**

Каждый спринт начинается с планирования — например, на первом этапе команда будет делать макет будущего приложения. Глобальные задачи — их ещё называют эпики — на этот спринт будут следующие: нарисовать mind map с функциями, затем [сделать прототип](https://practicum.yandex.ru/blog/kak-sozdat-maket-sayta/) в Figma, согласовать с клиентом. Задачи на спринт участники команды определяют вместе с [проджект-менеджером](https://practicum.yandex.ru/blog/professiya-project-manager/).

Промежуточным результатом для клиента будет макет будущего приложения. Чтобы команда могла двигаться дальше, клиент комментирует, что нравится, а что хочется улучшить. Эти задачи будут решать в следующем спринте.

**Возможные проблемы:**

- Если клиент будет давать слишком много пожеланий для улучшения продукта на каждом этапе, есть риск затянуть сроки проекта из-за того, что все удлинившиеся спринты соберутся в итоге в большой снежный ком и серьёзно увеличат сроки сдачи проекта.

**Слайд 11**. Как понять, что проекту подойдет Waterfall?

Несмотря на то, что в наши дни данная модель используется реже, сбрасывать со счетов каскадную систему не стоит. Она прошла проверку временем и может свободно применяться во многих отраслях производства. Каскадная система подойдет, если:

- есть четкое понимание конечного результата и на него ничего не повлияет в процессе работы;

- финансовых ресурсов и времени достаточно для реализации одного большого проекта;

- требуется подробная и детальная документация по всему проекту и его этапам;

-требуется строгое соблюдение последовательного плана;

-выполнение значительной части работы по проекту связано с аутсорсом.