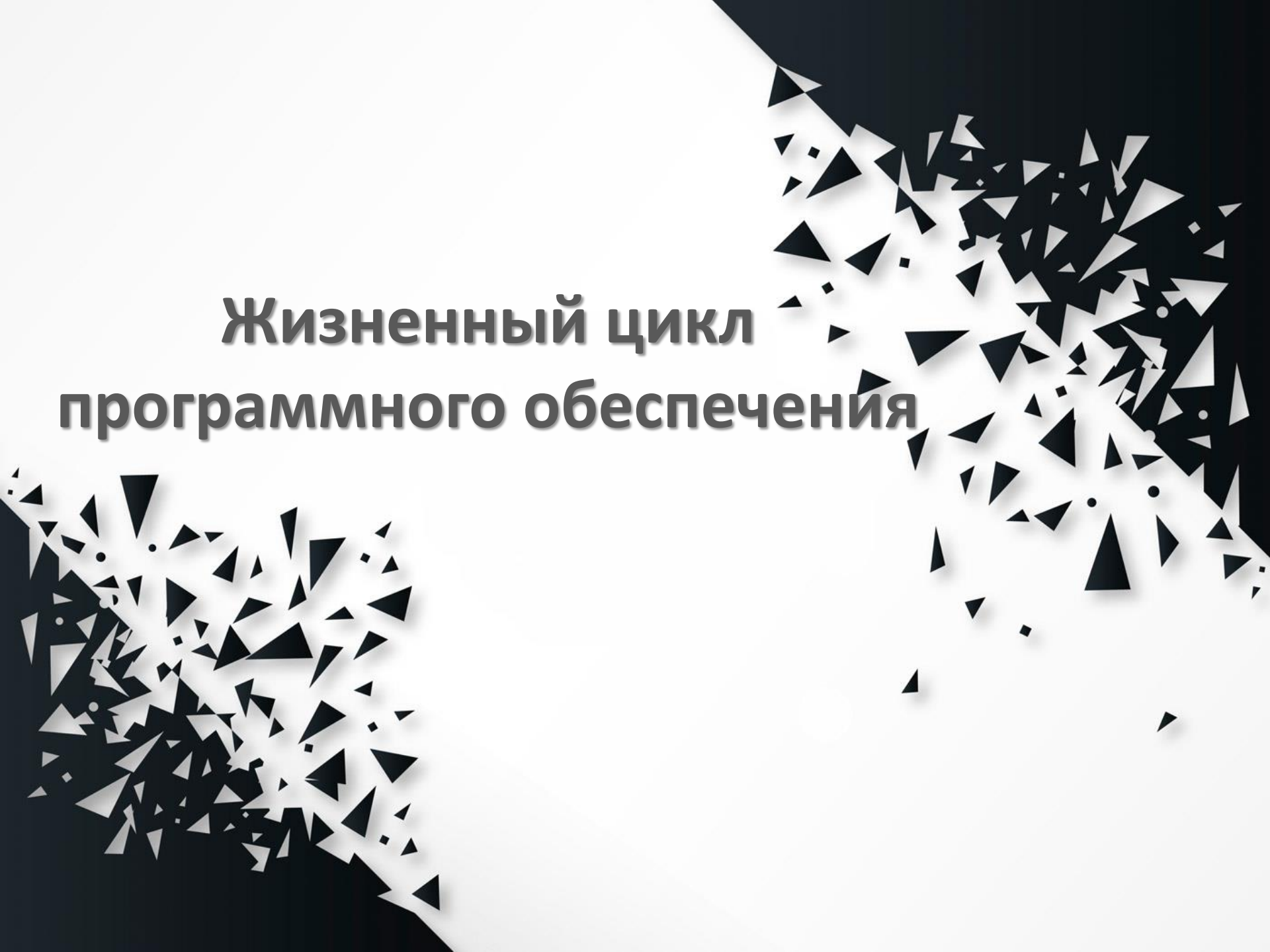


Жизненный цикл программного обеспечения



План:

1. Понятие жизненного цикла.
2. Основные этапы.
3. Модели жизненного цикла.
4. Дополнительные модели жизненного цикла.

ЖИЗНЕННЫЙ ЦИКЛ ПО

Жизненный цикл ПО (software lifecycle) – это период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

ОСНОВНЫЕ ЭТАПЫ

Обычно к этапам жизненного цикла относят:

1. Анализ требований
2. Проектирование
3. Программирование
4. Тестирование и отладку
5. Эксплуатацию, сопровождение и поддержку

Модели жизненного цикла ПО

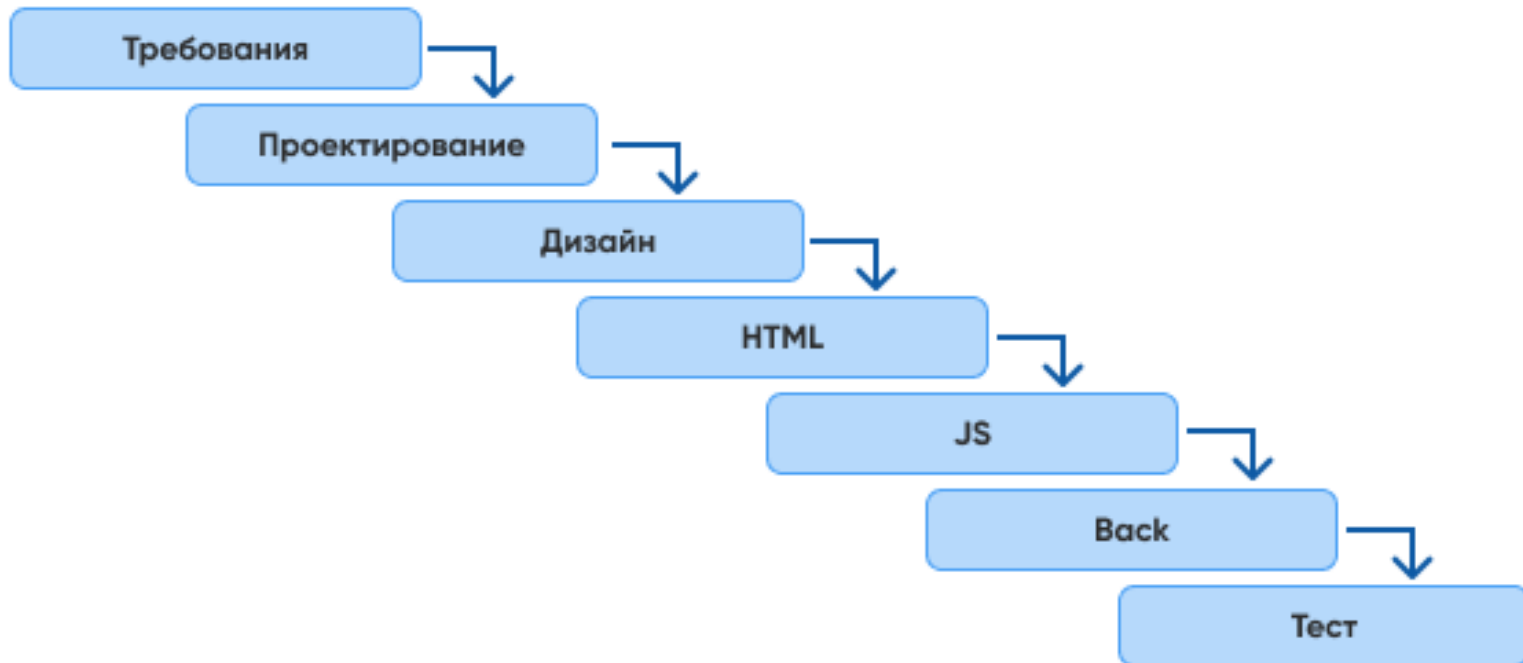
- По большому счету все модели можно разделить на две больших группы:
- последовательные
- итерационные модели

Waterfall (каскадная модель)

- Основная суть модели **Waterfall** в том, что этапы зависят друг от друга и следующий начинается, когда закончен предыдущий, образуя таким образом поступательное (каскадное) движение вперед.
- Параллелизм этапов в каскадной модели, хоть и ограничен, но возможен для абсолютно независимых между собой работ. При этом интеграция параллельных кусков все равно происходит на каком-то следующем этапе, а не в рамках одного.
- Команды разных этапов между собой не коммуницируют, каждая команда отвечает четко за свой этап.

Каскадная модель

Как работает изнутри



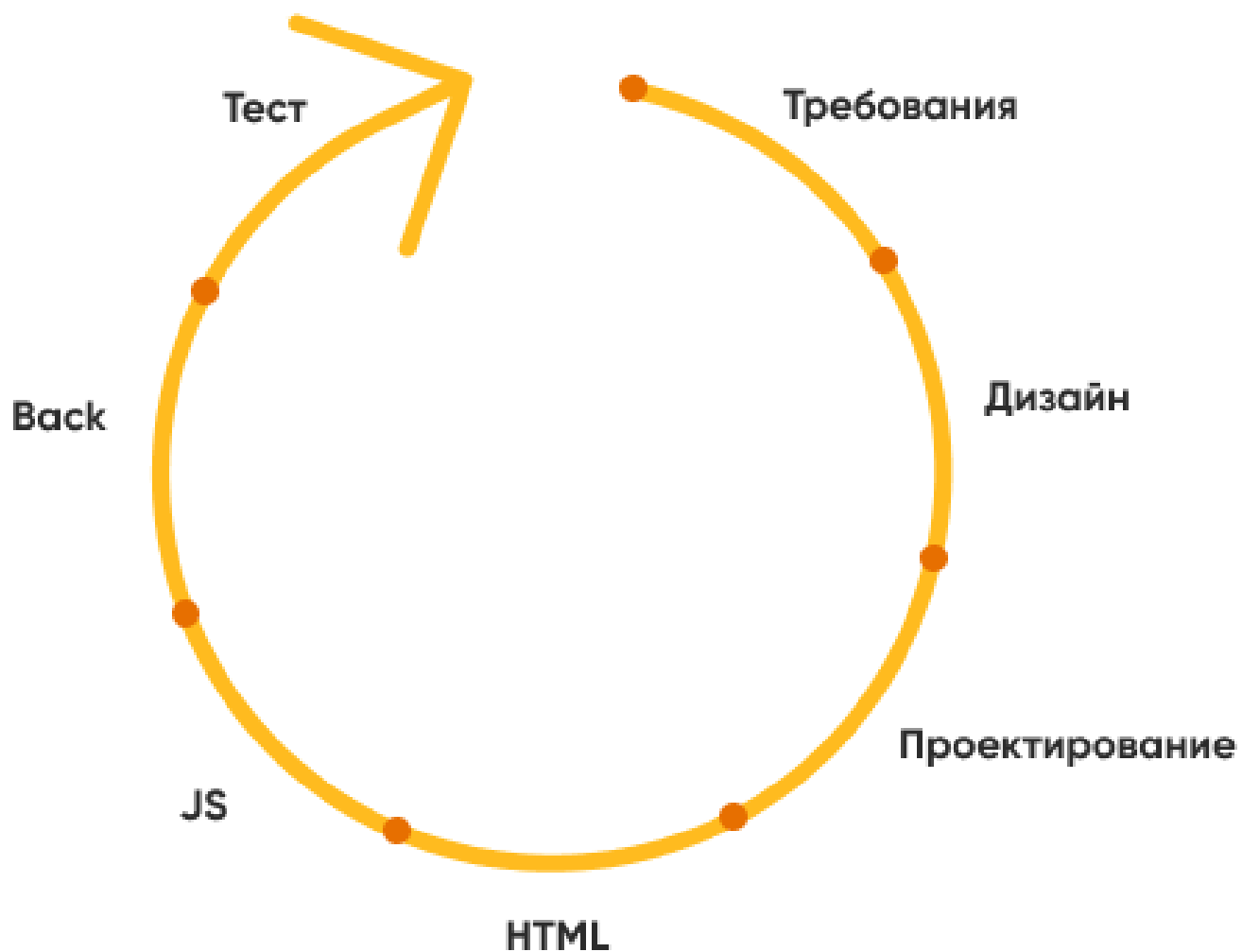
Как видит клиент



Итерационная, спиральная и инкрементная модели

- **Итерационная модель** предполагает разбиение проекта на части (этапы, итерации) и прохождение этапов жизненного цикла на каждом из них. Каждый этап является законченным сам по себе, совокупность этапов формирует конечный результат.

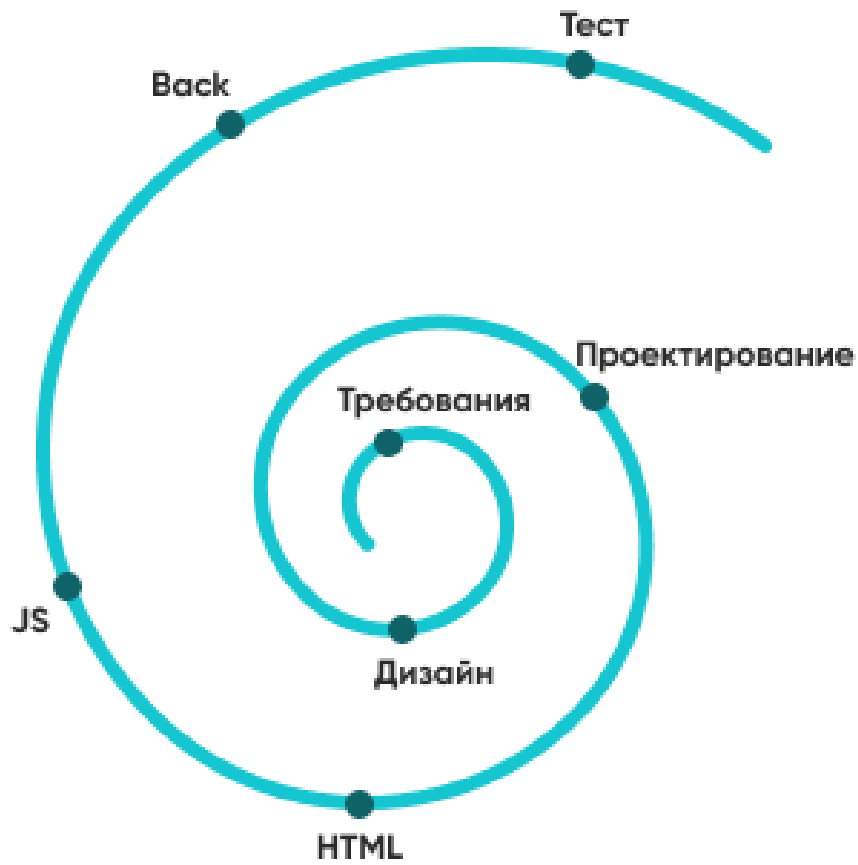
Итерационная модель



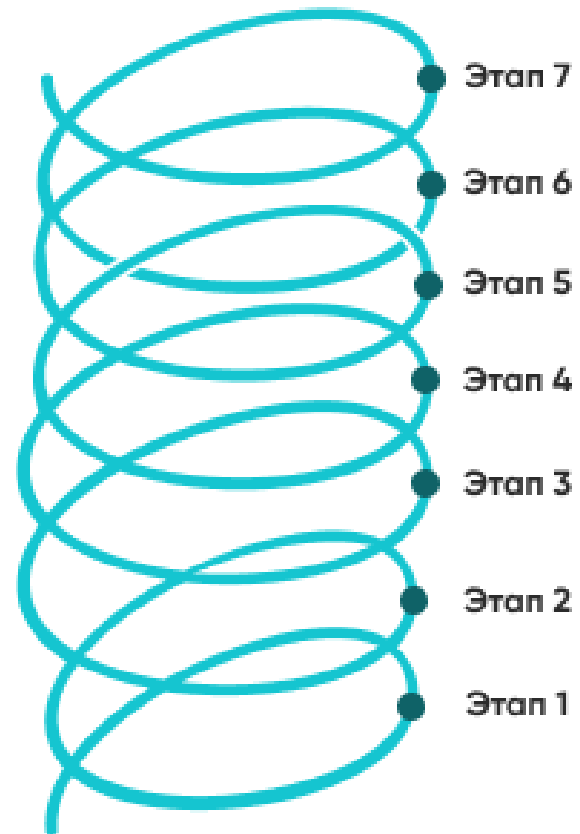
Спиральная модель

- Спиральная и инкрементная модели являются видами итерационной модели жизненного цикла.
- Все этапы жизненного цикла при спиральной модели идут витками, на каждом из которых происходят проектирование, кодирование, дизайн, тестирование и т. д. Такой процесс отображает суть названия: поднимаясь, проходится один виток (цикл) спирали для достижения конечного результата.

Спиральная модель



Как работает изнутри



Как видит клиент

Инкрементной модели

- Принцип, который лежит в основе инкрементной модели, подразумевает расширение возможностей, достраивание модулей и функций приложения. Буквальный перевод слова инкремент: «увеличение на один». Это «увеличение на один» применяется в том числе для обозначения версий продукта.

Дополнительные модели ЖЦ

- RAD Model
- Agile Model
- Iterative Model

«RAD Model» (rapid application development model)

- RAD-модель — разновидность инкрементной модели. В RAD-модели компоненты или функции разрабатываются несколькими высококвалифицированными командами параллельно, будто несколько мини-проектов. Временные рамки одного цикла жестко ограничены. Созданные модули затем интегрируются в один рабочий прототип. Синергия позволяет очень быстро предоставить клиенту для обозрения что-то рабочее с целью получения обратной связи и внесения изменений.

«Agile Model» (гибкая методология разработки)

- В «гибкой» методологии разработки после каждой итерации заказчик может наблюдать результат и понимать, удовлетворяет он его или нет. Это одно из преимуществ гибкой модели. К ее недостаткам относят то, что из-за отсутствия конкретных формулировок результатов сложно оценить трудозатраты и стоимость, требуемые на разработку. Экстремальное программирование (XP) является одним из наиболее известных применений гибкой модели на практике.
- В основе такого типа — непродолжительные ежедневные встречи — «Scrum» и регулярно повторяющиеся собрания (раз в неделю, раз в две недели или раз в месяц), которые называются «Sprint».

«Iterative Model» (итеративная или итерационная модель)

- Итерационная модель жизненного цикла не требует для начала полной спецификации требований. Вместо этого, создание начинается с реализации части функционала, становящейся базой для определения дальнейших требований. Этот процесс повторяется. Версия может быть неидеальна, главное, чтобы она работала. Понимая конечную цель, мы стремимся к ней так, чтобы каждый шаг был результативен, а каждая версия — работоспособна.
- Примером итерационной разработки может служить распознавание голоса. Первые исследования и подготовка научного аппарата начались давно, в начале — в мыслях, затем — на бумаге. С каждой новой итерацией качество распознавания улучшалось. Тем не менее, идеальное распознавание еще не достигнуто, следовательно, задача еще не решена полностью.



Спасибо за внимание!