

Жизненный цикл IT-проекта

Лектор – Скороход С. В.

Жизненный цикл проекта по PMBok

Жизненный цикл проекта – это совокупность фаз проекта, которые связывают его начало и окончание.

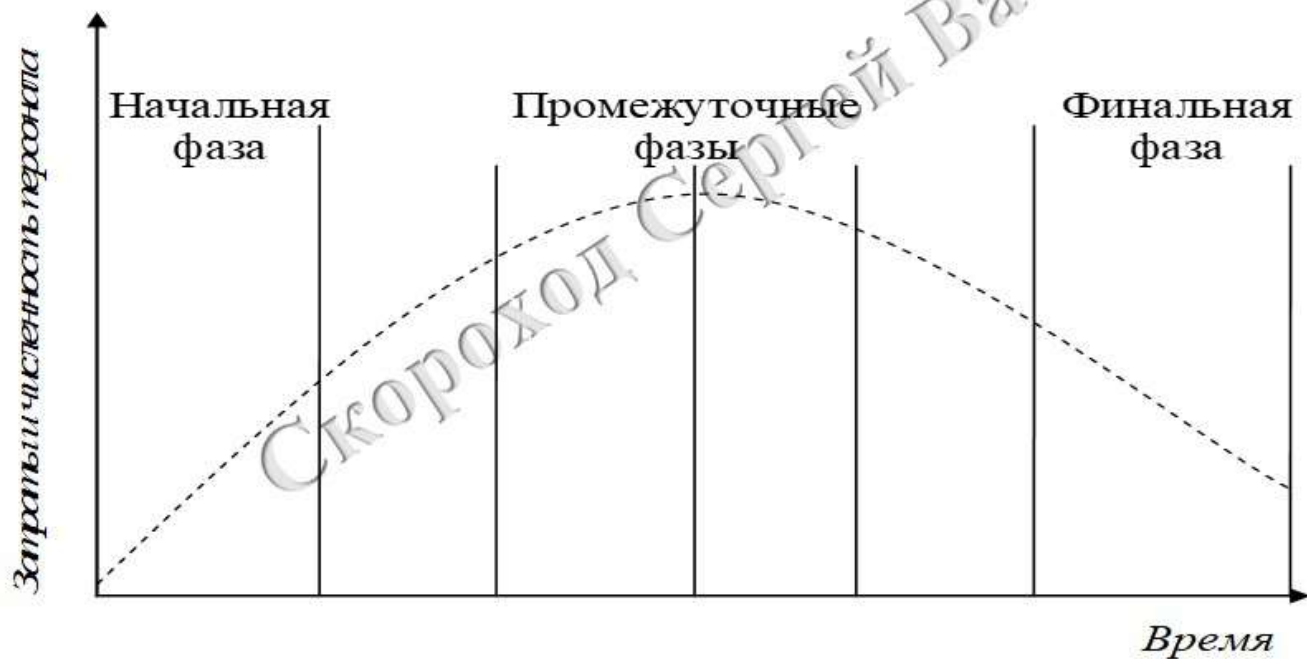
- Переход из одной фазы в другую подразумевает некую форму технической передачи или сдачи результатов, и именно это указывает на переход от одной фазы к другой.
- Результаты фазы проверяются на предмет завершенности и точности и проходят процедуру одобрения, прежде чем начнутся работы следующей фазы.

Жизненный цикл проекта определяет:

- Какие технические работы должны быть проведены в каждой фазе.
- В какой момент каждой фазы должны быть получены конкретные результаты и как проходит проверка и подтверждение каждого результата.
- Кто участвует в каждой фазе.
- Как контролировать и подтверждать каждую фазу.

Свойства жизненного цикла проекта

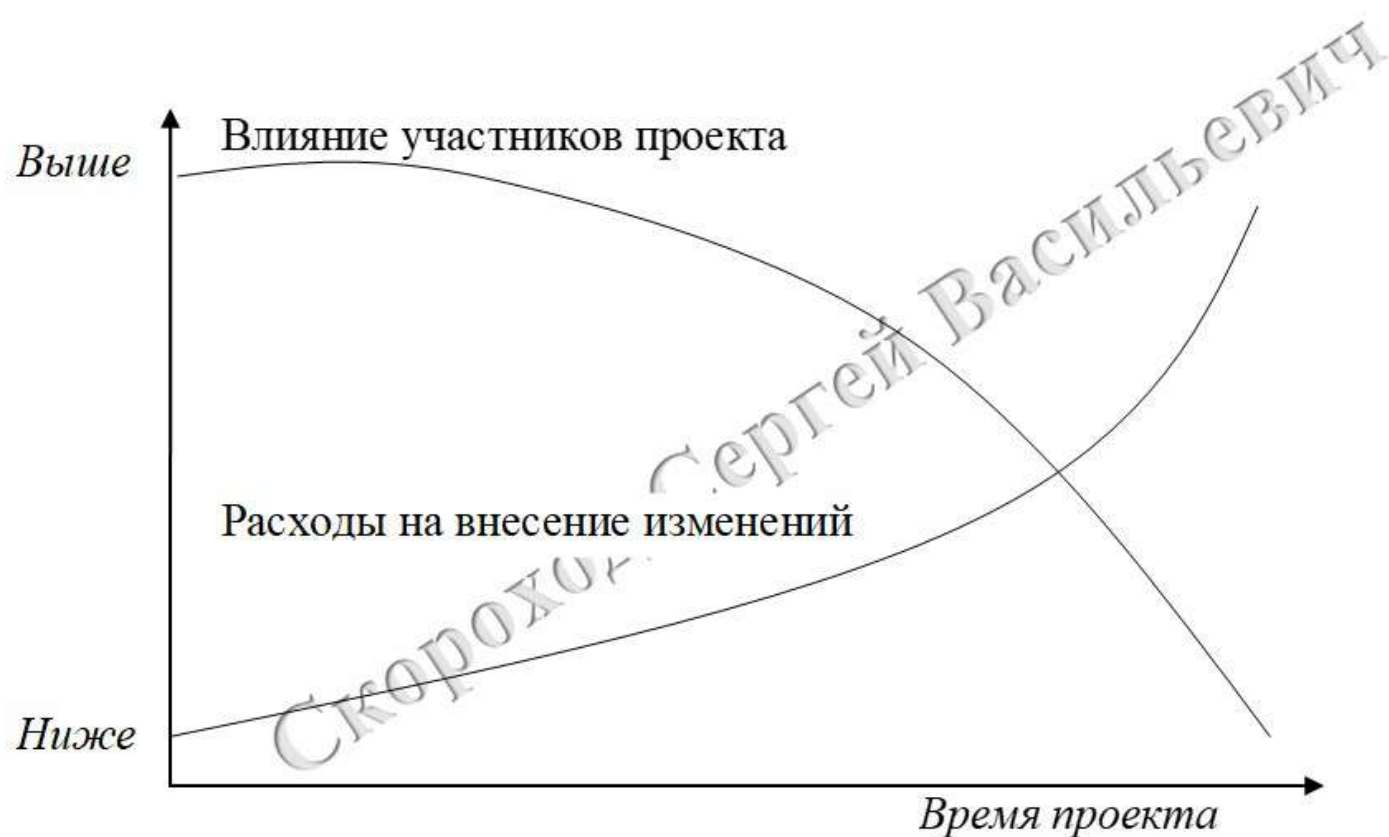
Уровень затрат и численность задействованного персонала невелики в начале, увеличиваются по ходу выполнения проекта и быстро падают на завершающем этапе проекта.



Свойства жизненного цикла проекта

- Уровень неуверенности и риск недостижения целей наиболее велики в начале проекта.
- Уверенность в завершении проекта увеличивается по ходу выполнения проекта.
- Способность участников проекта повлиять на конечные характеристики продукта проекта и окончательную стоимость проекта максимальны в начале проекта и уменьшаются по ходу выполнения проекта.
- Стоимость внесения изменений в проект и исправления ошибок возрастает по ходу выполнения проекта.

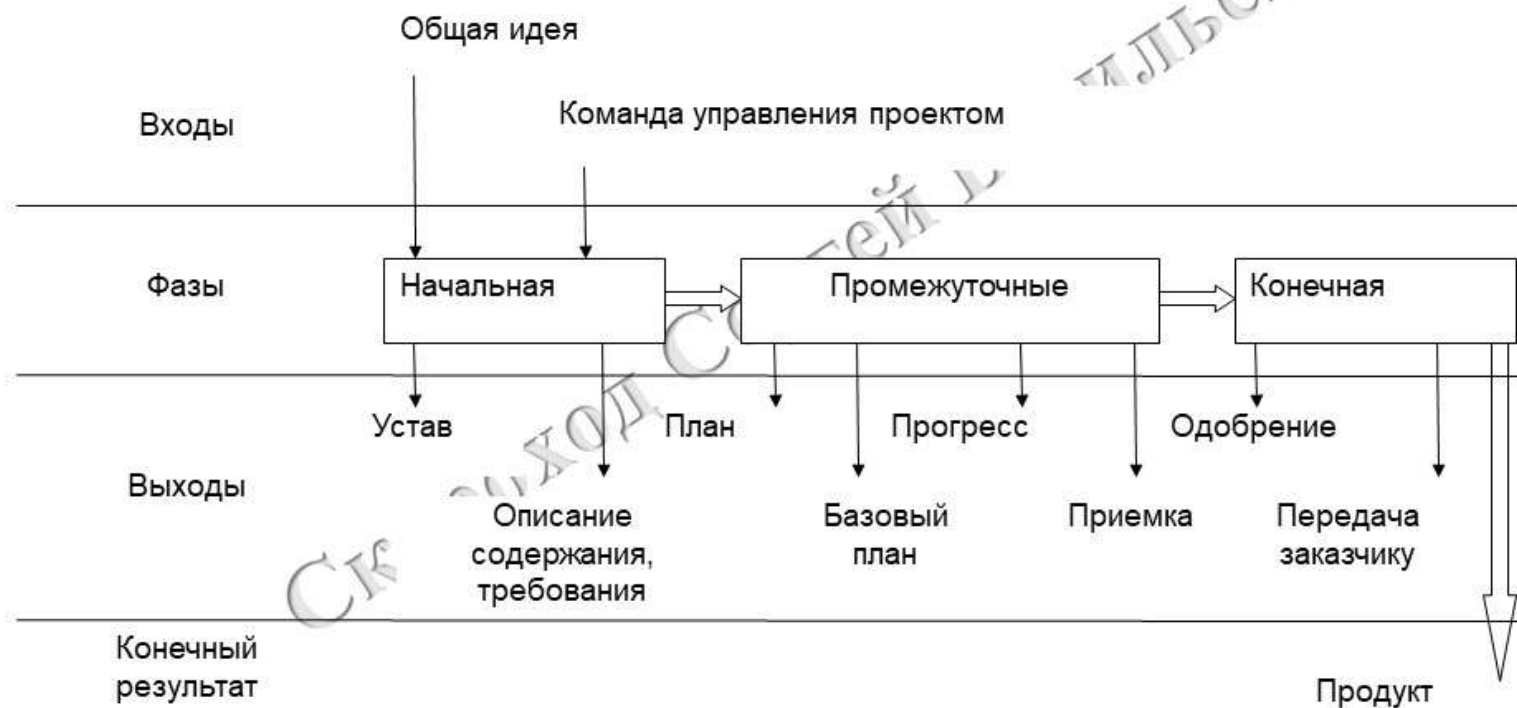
Влияние участников на проект



Фаза проекта

- Характеризуется завершением и одобрением одного или нескольких её результатов.
- Результат фазы –измеримый, проверяемый продукт работы (спецификация, отчет, план, библиотека, модуль, таблица).
- Фазы могут разбиваться на подфазы из соображений размера, сложности, уровня риска и ограничений на финансирование.
- Фаза завершается изучением проделанной работы и её результатов, чтобы определить, насколько они приемлемы, и решить, необходимы ли еще дополнительные работы или фазу можно считать закрытой.
- Выходы предыдущей фазы являются входами для последующих фаз.

Последовательность фаз в жизненном цикле проекта по PMBok

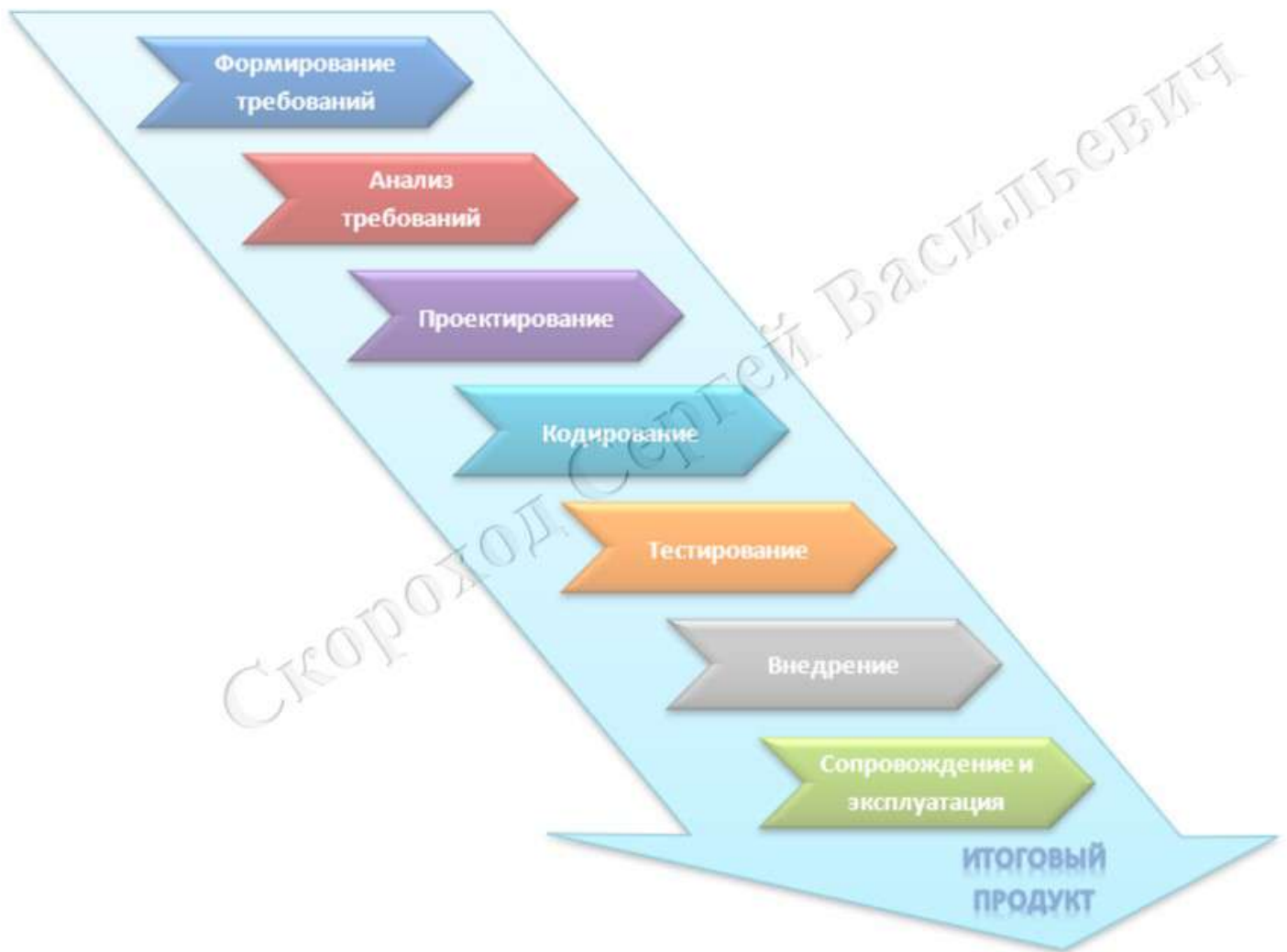


Жизненный цикл продукта и проекта

Жизненный цикл продукта намного шире и включает в себя жизненный цикл проекта.



Каскадная модель ЖЦ ИС



Каскадная модель ЖЦ ИС

- Переход на следующую стадию осуществляется только после того, как будет полностью завершена работа на текущей стадии.
- Возвратов на пройденные стадии не предусматривается.
- Каждая стадия завершается получением некоторых результатов. При этом они служат в качестве исходных данных для других стадий.
- Требования к разрабатываемому ПО (ИС) строго документируются в виде технического задания, которое неизменно на все время разработки.
- Каждая стадия завершается выпуском полного комплекта документации.

Преимущества каскадной модели:

- На каждой стадии формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
- выполняемые последовательности работ позволяют планировать сроки завершения всех работ и затраты по ним.

Недостатки каскадной модели:

- реальный процесс никогда не укладывается в жесткую схему;
- возникает необходимость в возврате к предыдущим стадиям;
- существенное запаздывание реальных результатов;
- получение в итоге ПО (ИС) с параметрами существенно отличающимися от ожидаемых.

Каскадная модель ЖЦ ИС

Дает **хорошие результаты**.

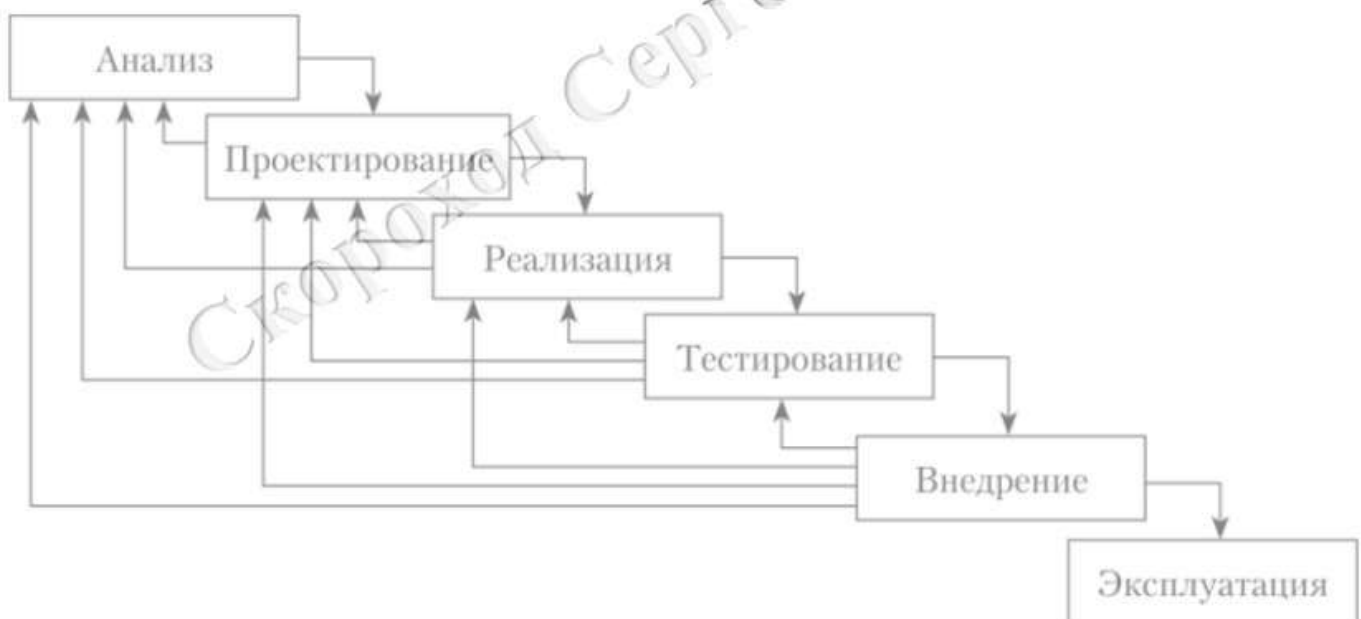
- Для реализации проектов, где каждое приложение представляет собой некое целое, единый независимый блок. В этом случае каждый этап завершается после полного выполнения и документального оформления всех предусмотренных работ.
- При построении относительно простых ИС, для которых в самом начале разработки можно точно и полно сформулировать, если не все, то большинство требований к системе (расчетные системы, системы реального времени).
- При создании и выпуске новой версии уже существующего продукта, если вносимые изменения вполне определены и управляемы (перенос существующего продукта на новую платформу).
- При выполнении крупных проектов с несколькими большими командами разработчиков.

Дает **плохие результаты**.

- При разработке систем, для которых невозможно четко и однозначно определить требования до начала работ по реализации (для новых систем).
- При разработке систем с быстрой изменчивостью в процессе эксплуатации системы.

Итеративная (инкрементная, эволюционная) модель ЖЦ ИС

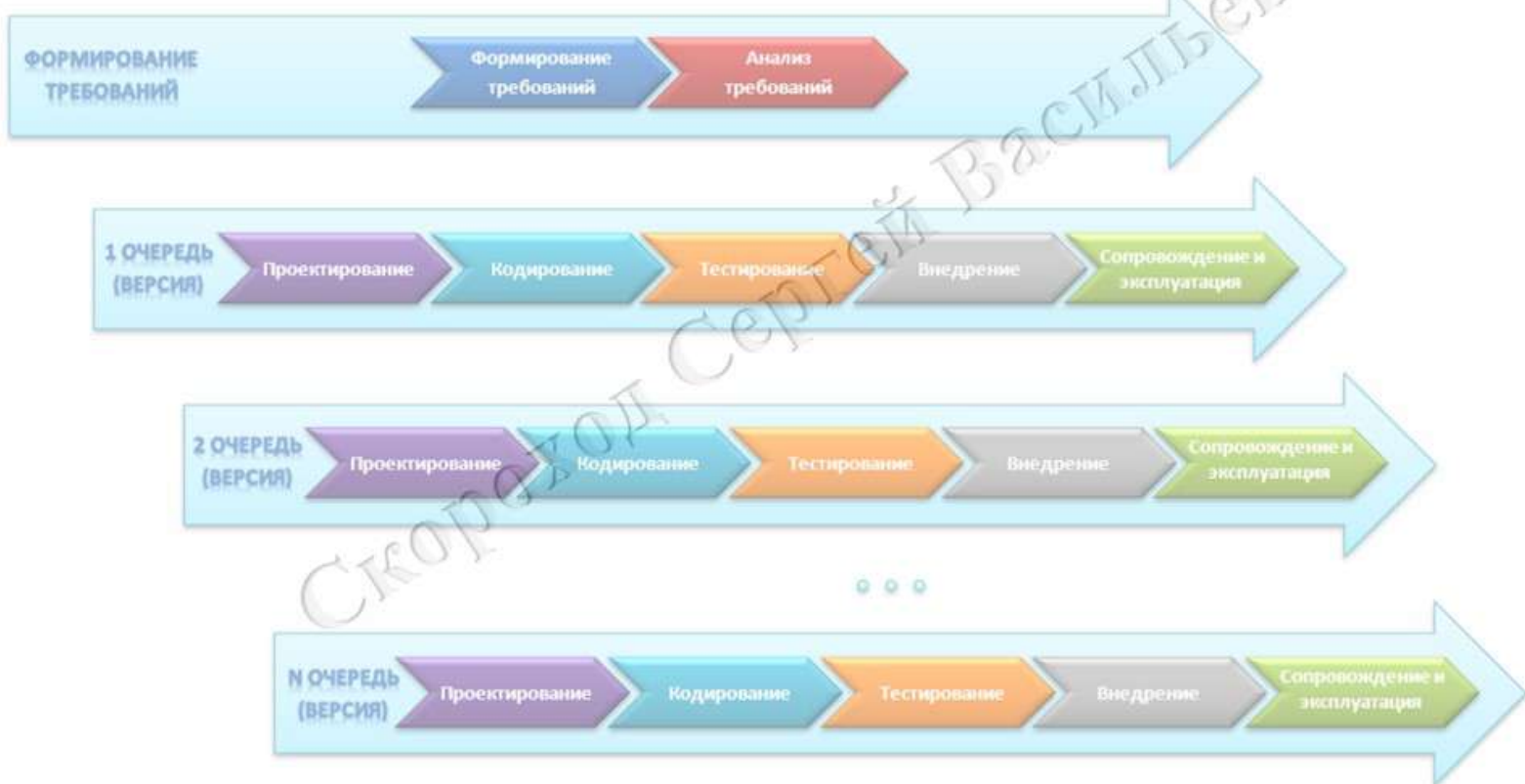
- Результат доработки каскадной модели с учетом взаимозависимости этапов и необходимости возврата на предыдущие ступени по причине неполноты требований или ошибок в формировании задания.
- Являлась основой для многих проектов.
- Разработка ИС ведется итерациями с циклами обратной связи между этапами.



Итеративная (инкрементная, эволюционная) модель ЖЦ ИС

- Первоначально инкрементная модель понималась как построение системы в виде последовательности сборок (релизов) по заранее подготовленному плану и заданным и неизменным требованиям.
- Сегодня под ней понимают постепенное наращивание функциональности создаваемого продукта. Такую модель называют итеративной.
- Итеративная модель предполагает разбиение жизненного цикла проекта на последовательность итераций, каждая из которых напоминает «мини-проект», создающий небольшой законченный фрагмент функциональности.
- Цель каждой итерации – получение работающей версии системы, включающей функциональность всех предыдущих и текущей итерации. Результат финальной итерации содержит полную функциональность продукта.
- Эволюционная модель подразумевает не только сборку работающей версии системы, но и её развертывание в реальных условиях с анализом откликов пользователей для планирования следующей итерации.

Итеративная (инкрементная, эволюционная) модель ЖЦ ИС



Итеративная (инкрементная, эволюционная) модель ЖЦ ИС

Достоинства модели.

- Возможность тестирования пользователями получаемого программного продукта уже на ранних стадиях разработки.
- Отсутствие необходимости заранее вовлекать большое количество финансовых средств для реализации всего проекта.
- Снижаются затраты на первоначальную поставку программного продукта.
- Снижение неопределенности и рисков с завершением каждой итерации.

Недостатки модели.

- Согласование результатов разработки с пользователями производится только в точках, планируемых после завершения каждого этапа работ, а общие требования к ИС зафиксированы в виде ТЗ на все время ее создания.
- Не предусмотрены итерации в рамках каждого инкремента.
- Определение полной функциональной системы должно осуществляться в начале ЖЦ.
- Затягивание проекта за счет переноса решений трудных проблем на будущее.

Итеративная (инкрементная, эволюционная) модель ЖЦ ИС

Область применения модели – для следующих проектов.

- Где большинство требований можно сформулировать заранее, но их появление ожидается через определенный период времени.
- Когда существует потребность быстро поставить на достаточно «узкий» рыночный сегмент продукт, имеющий базовый функционал.
- Для которых предусмотрен большой период времени разработки.
- Где при рассмотрении риска, финансирования, графика выполнения проекта, размера программы, ее сложности или необходимости в реализации на ранних фазах оказывается, что самым оптимальным вариантом является применение принципа пофазовой разработки.
- Выполнение которых проводится с применением новой технологии, что позволяет пользователю адаптироваться к системе путем выполнения более мелких инкрементных шагов, без резкого перехода к применению основного нового продукта.

V-образная модель

Создана с целью помочь работающей над проектом команде в планировании дальнейшего тестирования системы



V-образная модель

- Позволяет гораздо лучше контролировать результат на предмет его соответствия ожиданиям, поскольку сфокусирована на тестировании.
- Дает возможность значительно повысить качество ПО за счет своей ориентации на тестирование.
- Во многом разрешила проблему соответствия созданного продукта выдвигаемым требованиям благодаря разработке процедур верификации и аттестации на ранних стадиях ЖЦ (пунктирные линии на рисунке указывают на зависимость этапов планирования/постановки задачи и тестирования/приемки).
- Является всего лишь модификацией каскадной модели и обладает многими ее недостатками.
- Слабо приспособлена к возможным изменениям требований заказчика.
- Если процесс разработки занимает продолжительное время (иногда до нескольких лет), то полученный в результате продукт может оказаться фактически ненужным заказчику, поскольку его потребности существенно изменились.

V-образная модель

Достоинства модели.

- Планирование верификации и тестирования разрабатываемого продукта на ранних стадиях его разработки.
- Предусматриваются аттестация и верификация всех внешних и внутренних полученных данных, а не только самого программного продукта.
- Определение требований выполняется перед разработкой проекта системы.
- Определяются продукты, которые должны быть получены в результате разработки, причем каждые полученные данные должны подвергаться тестированию.
- Модель проста в использовании.

Недостатки модели.

- Нет учета существующих между фазами итераций.
- Не предусмотрено внесение изменений на разных этапах ЖЦ.
- Тестирование требований происходит слишком поздно, вследствие чего невозможно внести изменения, не повлияв на график выполнения проекта.

V-образная модель

Сфера применения.

- Проекты, в которых вся информация о требованиях доступна заранее.
- Для систем, требующих высокой надежности, например прикладные программы для наблюдения за пациентами в клиниках, а также встроенное ПО для устройств управления аварийными подушками безопасности в автомобилях.

Спиральная модель

Подразумевает разработку в виде последовательности версий, но в начале проекта определены не все требования. Требования уточняются в результате разработки версий.

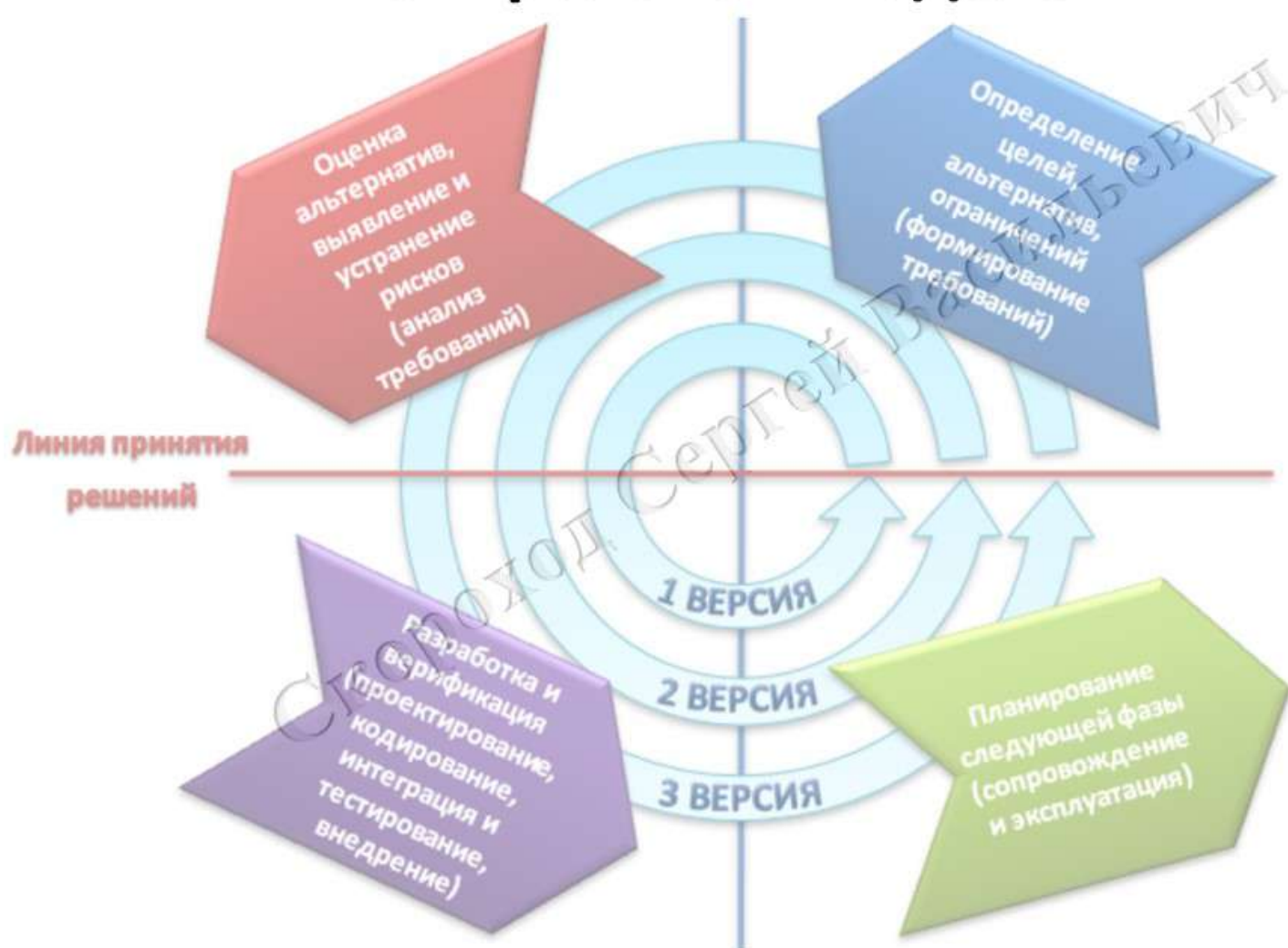
На каждом витке спирали.

- Выполняется создание очередной версии продукта.
- Уточняются требования проекта.
- Определяется его качество.
- Планируются работы следующего витка.

Особое внимание уделяется начальным этапам разработки – анализу и проектированию, где реализуемость тех или иных технических решений проверяется и обосновывается посредством создания прототипов (макетирования).

- Прикладное ПО (ИС) создается не сразу, как в случае каскадной схемы, а по частям, с использованием метода прототипирования. Под прототипом понимается программный компонент, реализующий отдельные функции и внешние интерфейсы, разрабатываемого ПО.
- Создание ПО (ИС) осуществляется не сразу, а за несколько витков спирали.
- На каждом витке производится тщательная проверка риска превышения сроков и стоимости проекта.

Спиральная модель



Спиральная модель

Преимущества спиральной модели.

- Позволяет быстрее показать пользователям системы работоспособный продукт, активизируя процесс уточнения и дополнения требований.
- Допускает изменение требований при разработке ИС.
- Обеспечивает большую гибкость в управлении проектом.
- Позволяет получить более надежную и устойчивую систему. По мере развития системы ошибки и слабые места обнаруживаются и исправляются на каждой итерации.
- Позволяет совершенствовать процесс разработки – анализ, проводимый в каждой итерации, позволяет проводить оценку того, что должно быть изменено и улучшено в организации разработки не следующем витке.
- Уменьшаются риски заказчика. Заказчик может с минимальными финансовыми потерями завершить развитие неперспективного проекта.

Недостатки:

- Увеличивается неопределенность у разработчика в перспективах развития проекта.
- Затруднены операции временного и ресурсного планирования проекта в целом. Нужно ввести временные ограничения на каждую из стадий ЖЦ. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа выполнена.

Сравнение моделей ЖЦ

Характеристика	Каскадная	Инкрементная	Спиральная
Новизна разработки и обеспеченность ресурсами	Типовой. Хорошо проработаны технология и методы решения задачи		Нетиповой (новаторский). Нетрадиционный для разработчика
	Ресурсов хватает для реализации в сжатые сроки	Ресурсов не хватает для реализации в сжатые сроки	
Масштаб проекта	Малые и средние	Средние и крупные	Любые
Сроки выполнения проекта	До года	До нескольких лет. Разработка одной версии может занимать срок от нескольких недель до года	
Заключение отдельных договоров на отдельные версии	Заключается один договор. Версия и есть итоговый результат проекта		На отдельную версию или несколько последовательных версий обычно заключается отдельный договор

Сравнение моделей ЖЦ

Характеристика	Каскадная	Инкрементная	Спиральная
Определение основных требований в начале проекта	Да	Да	Нет
Изменение требований по мере развития проекта	Нет	Незначительное	Да
Разработка итерациями (версиями)	Нет	Да	Да
Распространение промежуточного ПО	Нет	Может быть	Да