

Группа: 20п-3
Сделали: Ахтарьянов Динислам,
Гаджиев Асим

Жизненный цикл программного обеспечения

Понятие жизненного цикла основные этапы

Модели жизненного цикла

Дополнительные модели жизненного цикла

Жизненный цикл программного обеспечения

Жизненный цикл программного обеспечения (ПО) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.



Стандарты жизненного цикла ПО

- ГОСТ 34.601-90
- ISO/IEC 15288:2015 Systems and software engineering — System life cycle processes
- ISO/IEC/IEEE 12207:2017 System and software engineering — Software life cycle processes



Понятие жизненного цикла основные этапы

Жизненный цикл изделия (жизненный цикл продукции) — совокупность явлений и процессов, повторяющаяся с периодичностью, определяемой временем существования типовой конструкции изделия, от её замысла до утилизации или конкретного экземпляра изделия от момента завершения его производства до утилизации (ГОСТ Р 56136-2014)[1]. Является частным случаем жизненного цикла системы применительно к изделиям промышленного производства.

Типовая модель жизненного цикла по стандарту ISO/IEC 15288

В 2002 году Международная организация по стандартизации и Международная электротехническая комиссия выпустили результат многолетней работы — стандарт ISO/IEC 15288:2002 (см. русскоязычный аналог ГОСТ Р ИСО МЭК 15288-2005).

Согласно стандарту, процессы и действия жизненного цикла определяются, соответствующим образом настраиваются и используются в течение стадии жизненного цикла, для полного удовлетворения целей и результатов на этой стадии. В различных стадиях жизненного цикла могут принимать участие разные организации. Не существует единой универсальной модели жизненных циклов систем. Те или иные стадии жизненного цикла могут отсутствовать или присутствовать в зависимости от каждого конкретного случая разработки системы.

Типовая модель жизненного цикла по стандарту ISO/IEC 15288

В стандарте в качестве примера были приведены следующие стадии жизненного цикла:

- Замысел.
- Разработка.
- Производство.
- Применение.
- Поддержка применения.
- Прекращение применения и списание.
- В версии стандарта от 2008 года (ISO/IEC 15288:2008) и в последующих версиях примеры стадий жизненного цикла отсутствуют[8].

Типовая модель жизненного цикла по версии Министерства обороны США

Для управления рисками в области применения передовых технологий, и сведения к минимуму дорогостоящих технических или управленческих ошибок, МО США разработало руководство, содержащее все необходимые принципы разработки систем. Эти принципы вошли в специальный перечень директив — DoD 5000.

Модель жизненного цикла системы управления материально-техническим обеспечением по версии МО США состоит из пяти стадий:

- Анализ.
- Разработка технологии.
- Инженерная и производственная разработка.
- Производство и развертывание.
- Функционирование и поддержка.

Типовая модель жизненного цикла системы Национального общества профессиональных инженеров (NSPE)

Этот вариант модели жизненного цикла NSPE адаптирован для коммерческих систем и направлен на развитие новых продуктов, обычно являющихся результатом технического прогресса. Жизненный цикл по модели NSPE разбивается на шесть стадий:

- Концепция.
- Техническая реализация.
- Разработка.
- Коммерческая валидация и подготовка производства.
- Полномасштабное производство.
- Поддержка конечного продукта.

Типовая модель жизненного цикла продукции по Р 50-605-80-93

В руководящем документе Р 50-605-80-93 рассматривается жизненный цикл промышленного изделия, в том числе — военной техники.

Для промышленной продукции гражданского назначения предложены следующие стадии:

- Исследование и проектирование.
- Изготовление.
- Обращение и реализация.
- Эксплуатация или потребление.
- В рамках жизненного цикла промышленной продукции гражданского назначения предложено рассматривать 73 вида работ и 23 типа стейкхолдеров («участников работ» по терминологии документа).

Типовая модель жизненного цикла продукции по Р 50-605-80-93

Для промышленной продукции военного назначения предложены следующие стадии:

- Исследование и обоснование разработки.
- Разработка.
- Производство.
- Эксплуатация.
- Капитальный ремонт.

В рамках жизненного цикла промышленной продукции гражданского назначения предложено рассматривать 73 вида работ и 23 типа стейкхолдеров («участников работ» по терминологии документа)

Типовая модель жизненного цикла продукции по Р 50-605-80-93

Для промышленной продукции военного назначения предложены следующие стадии:

- Исследование и обоснование разработки.
- Разработка.
- Производство.
- Эксплуатация.
- Капитальный ремонт.

В рамках жизненного цикла промышленной продукции военного назначения предложено рассматривать 25 видов работ и 7 типов стейкхолдеров (участников работ).

Типовая модель жизненного цикла программного обеспечения

Стадии жизненного цикла системы и их составные фазы, представленных на рисунке «Модель жизненного цикла системы», относятся к большинству сложных систем, в том числе к тем, которые содержат программное обеспечение со значительным объемом функциональных возможностей на уровне компонентов. В программно-интенсивных системах, в которых программное обеспечение выполняет практически все функции (как например в современных финансовых системах, в системах бронирования авиабилетов, в глобальной сети интернет, и в др.), как правило жизненные циклы схожи по содержанию, но часто усложняются итерационными процессами и прототипированием.

Модели жизненного цикла

Модель жизненного цикла программного обеспечения — структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.

Эти модели можно разделить на 3 основных группы:

Инженерный подход

С учетом специфики задачи

Современные технологии быстрой разработки

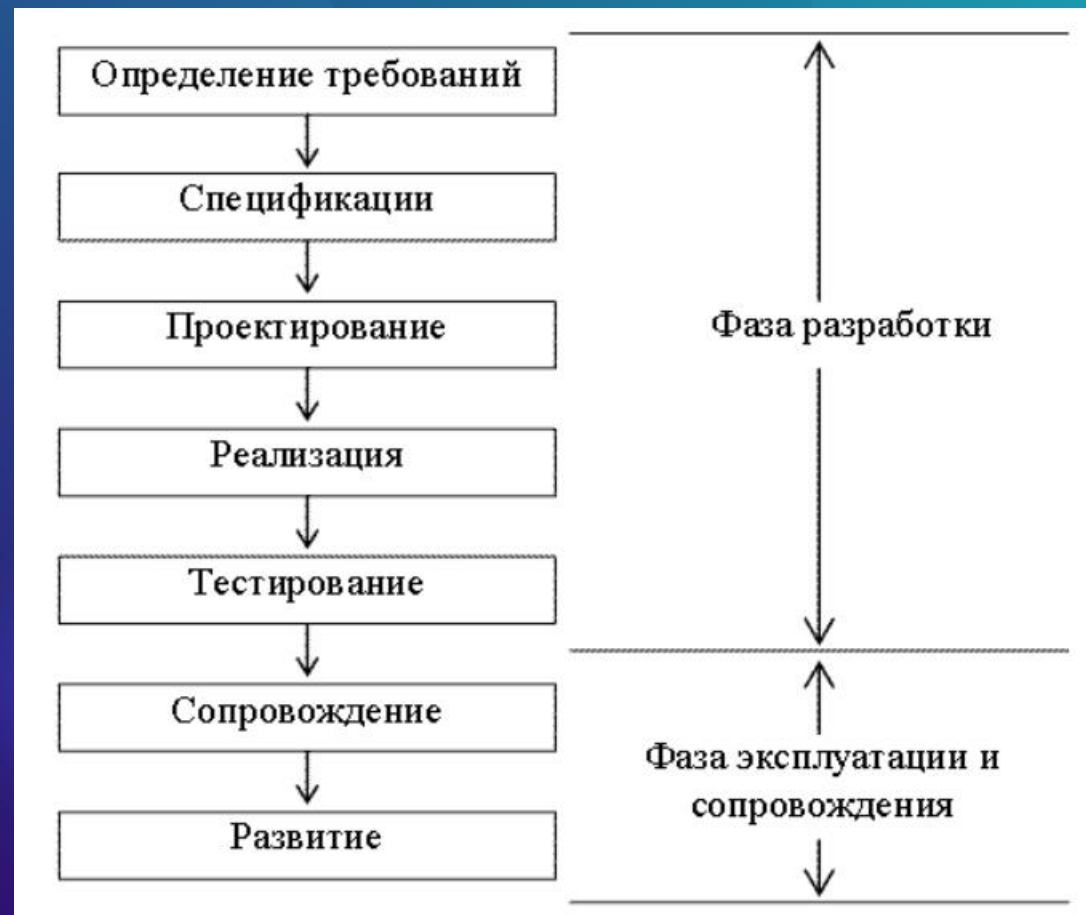


Общепринятая модель

Вероятно, самым распространенным мотивом обращения к понятию жизненного цикла является потребность в систематизации работ в соответствии с технологическим процессом. Этому назначению хорошо соответствует так называемая общепринятая модель жизненного цикла программного обеспечения, согласно которой программные системы проходят в своем развитии две фазы :

- разработка,
- сопровождение.

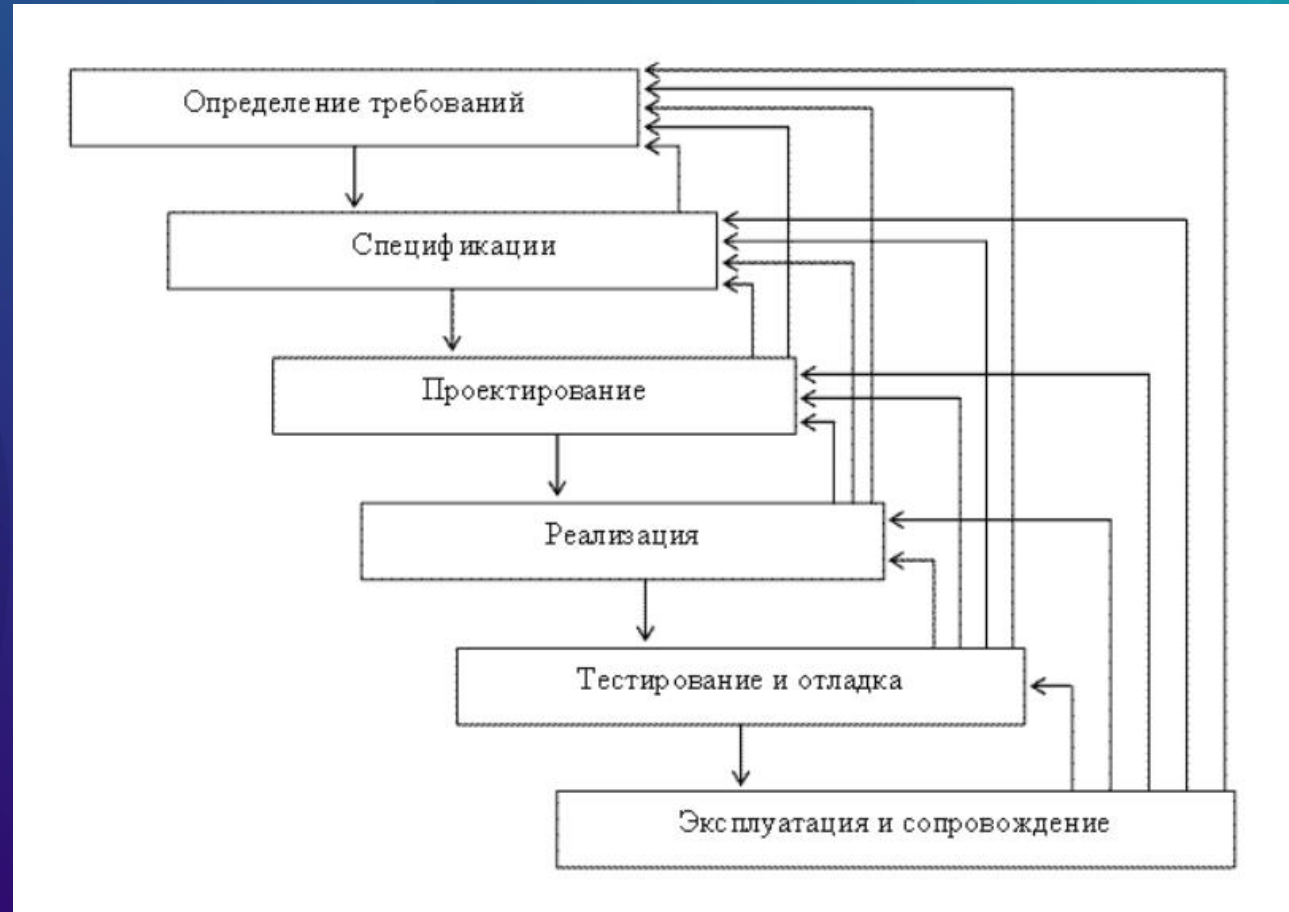
Общепринятая модель



Классическая итерационная модель

Общепринятая модель жизненного цикла является идеальной, так как только очень простые задачи проходят все этапы без каких-либо итераций — возвратов на предыдущие шаги технологического процесса. При программировании, например, может обнаружиться, что реализация некоторой функции очень громоздка, неэффективна и вступает в противоречие с требуемой от системы производительностью. В этом случае требуется перепроектирование, а может быть, и переделка спецификаций. При разработке больших нетрадиционных систем необходимость в итерациях возникает регулярно на любом этапе жизненного цикла как из-за допущенных на предыдущих шагах ошибок и неточностей, так и из-за изменений внешних требований к условиям эксплуатации системы.

Классическая итерационная модель



Каскадная модель

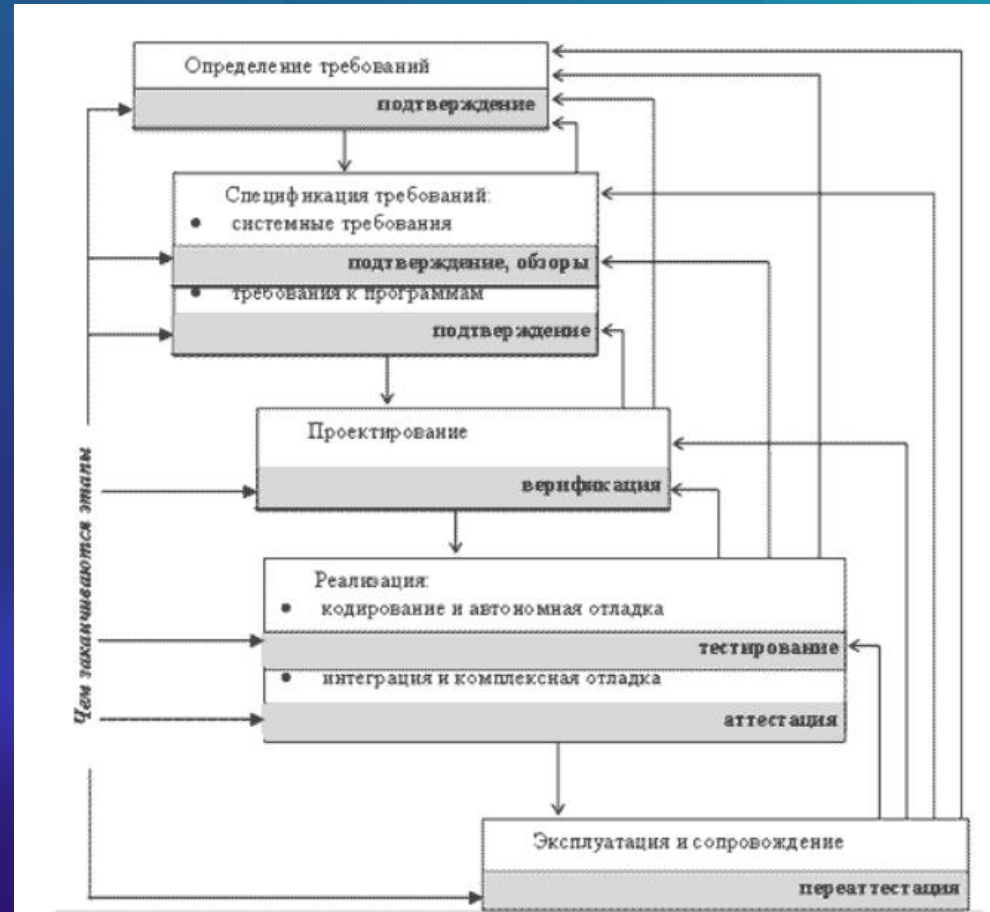
Некоторой более строгой разновидностью классической модели является так называемая каскадная модель, которую можно рассматривать в качестве показательного примера того, какими методами можно минимизировать возвраты.

Характерные черты каскадной модели:

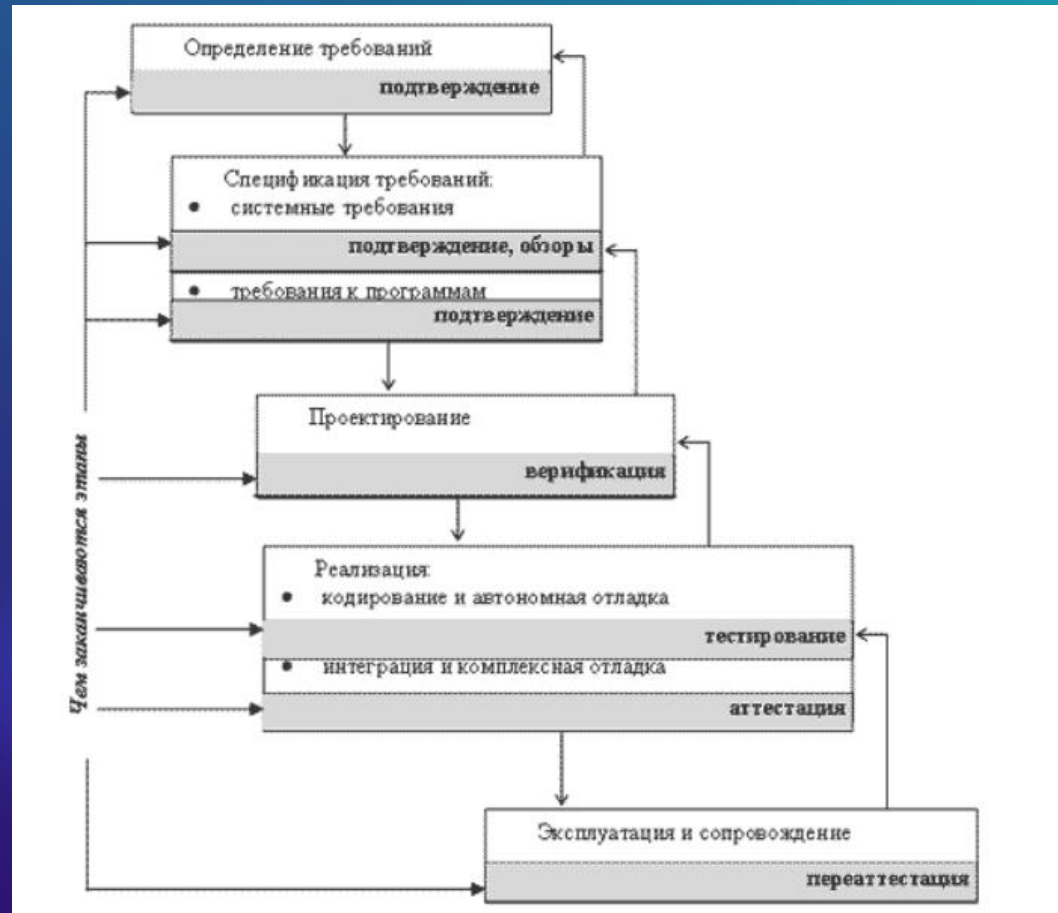
- завершение каждого этапа (они почти те же, что и в классической модели) проверкой полученных результатов с целью устранить как можно большее число проблем, связанных с разработкой изделия;
- циклическое повторение пройденных этапов (как в классической модели).

Мотивация каскадной модели связана с так называемым управлением качеством программного обеспечения. В связи с ней уточняются понятия этапов, некоторые из них структурируются (спецификация требований и реализация).

Каскадная модель



Строгая каскадная модель



Модель кодирования и устранения ошибок

Совершенно простая модель, характерная для студентов ВУЗов. Именно по этой модели большинство студентов разрабатывают, ну скажем лабораторные работы.

- Данная модель имеет следующий алгоритм:
- Постановка задачи
- Выполнение
- Проверка результата
- При необходимости переход к первому пункту

Модель также ужасно устаревшая. Характерна для 1960-1970 гг., по-этому преимуществ перед следующими моделями в нашем обзоре практически не имеет, а недостатки на лицо. Относится к первой группе моделей.

Каскадная модель жизненного цикла программного обеспечения (водопад)

Алгоритм данного метода, который я привожу на схеме, имеет ряд преимуществ перед алгоритмом предыдущей модели, но также имеет и ряд весомых недостатков.

Преимущества:

- Последовательное выполнение этапов проекта в строгом фиксированном порядке
- Позволяет оценивать качество продукта на каждом этапе

Недостатки:

- Отсутствие обратных связей между этапами
- Не соответствует реальным условиям разработки программного продукта
- Относится к первой группе моделей.

Каскадная модель жизненного цикла программного обеспечения (водопад)



Каскадная модель с промежуточным контролем (водоворот)

Данная модель является почти эквивалентной по алгоритму предыдущей модели, однако при этом имеет обратные связи с каждым этапом жизненного цикла, при этом порождает очень весомый недостаток: 10-ти кратное увеличение затрат на разработку. Относится к первой группе моделей.

V модель (разработка через тестирование)

Данная модель имеет более приближенный к современным методам алгоритм, однако все еще имеет ряд недостатков. Является одной из основных практик экстремального программирования.



Модель на основе разработки прототипа

Данная модель основывается на разработки прототипов и прототипирования продукта.

Прототипирование используется на ранних стадиях жизненного цикла программного обеспечения:

- Прояснить не ясные требования (прототип UI)
- Выбрать одно из ряда концептуальных решений (реализация сценариев)
- Проанализировать осуществимость проекта

Классификация прототипов:

- Горизонтальные и вертикальные
- Одноразовые и эволюционные
- бумажные и раскладовки

Модель на основе разработки прототипа

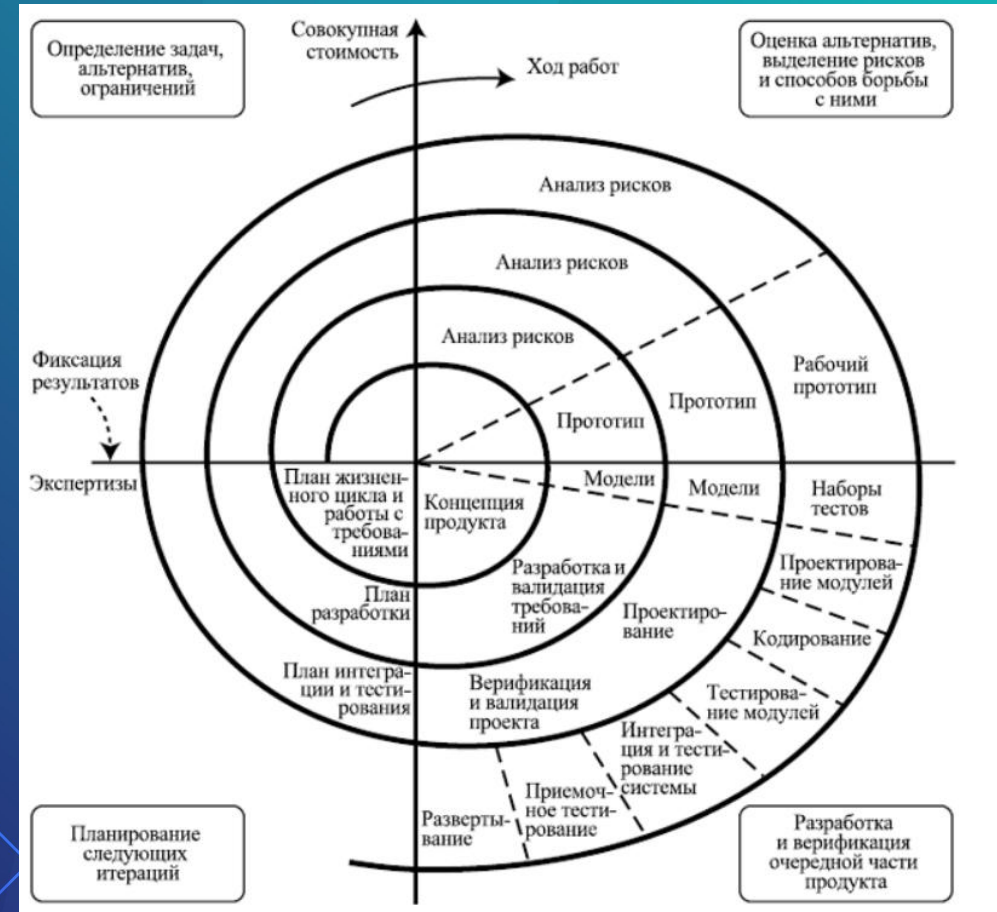
Горизонтальные прототипы — моделирует исключительно UI не затрагивая логику обработки и базу данных.

Вертикальные прототипы — проверка архитектурных решений.

Одноразовые прототипы — для быстрой разработки.

Эволюционные прототипы — первое приближение эволюционной системы.

Модель принадлежит второй группе.



Спиральная модель жизненного цикла программного обеспечения

Преимущества:

- Быстрое получение результата
- Повышение конкурентоспособности
- Изменяющиеся требования — не проблема

Недостатки:

- Отсутствие регламентации стадий

Третьей группе принадлежат такие модели как экстремальное программирование (XP), SCRUM, инкрементальная модель (RUP), но о них я бы хотел рассказать в отдельном топике.