

ISO/IEC 15288:2008

Системная инженерия - процессы жизненного цикла систем

ISO/IEC 15288

«Системная инженерия - процессы жизненного цикла систем»

- Дать возможность организациям (внешним и внутренним контракторам) договориться о совмещении замыслов, процессов проектирования, создания, эксплуатации и вывода из эксплуатации самых разных рукотворных систем – от зубочисток до атомных станций, от систем стандартизации до корпораций
- Внедрить в практику организации ряд ключевых идей системной инженерии:
 - системного подхода
 - жизненного цикла
 - инжиниринга требований
 - архитектурного дизайна
 - процессного подхода
 - проектного подхода
 - культуры контрактации

Универсальность

- Применим к любым рукотворным системам любой области человеческой деятельности (включая организации, сервисы, сами системы стандартизации).
- Охватывает полный цикл жизни (например: замысел, разработка, производство, использование, поддержка и вывод из эксплуатации)
- Учитывает необходимость контрактации (приобретения и поставки продуктов и услуг)
- Охватывает использование внутри организаций и между организациями (в «расширенной организации» проекта)
- Включает в процессы людей, оборудование, компьютеры, софт (ссылается на связанный стандарт ISO 12207 – жизненный цикл софта)
- Применяется **параллельно, итеративно и рекурсивно** для различных частей системы
- Учитывает особенности композиции любых систем – встроенных, автономных, интегрированных и любых других, сложных и простых

Происхождение

- Совместная разработка ISO и IEC, активное участие INCOSE
- Начало работ в 1996, версии в 2002, 2005 (ГОСТ Р ИСО/МЭК 15288-2005), 2008
- Призван гармонизировать так называемое «болото стандартов» системной инженерии (многочисленные стандарты, принятые различными военными ведомствами, государствами, отраслевыми организациями стандартизации)

Жизненный цикл

Аббревиатура русск: **ЖЦ**

Аббревиатура англ: **LC (Life Cycle)**

ОПРЕДЕЛЕНИЯ

1. Развитие системы, продукта, услуги, проекта или другого объекта, созданного человеком от концепции до изъятия из обращения. ISO/IEC 12207, ISO/IEC 15288.
2. Конечный набор основных фаз и шагов, которые система проходит на протяжении всей истории своего существования. ISO 15704.

Жизненный цикл системы

System Life Cycle

1. Эволюция во времени целевой системы от концепции до утилизации. ISO/IEC 15288.
2. Ход связанных с развитием изменений, через которые проходит система, начиная от ее замысла и до прекращения использования. ISO/IEC 2382-20.

Стадия

Stage, во всех стандартах системной инженерии.

Период времени в течение ЖЦ объекта, который относится к определенному состоянию его реализации или описания.

Стадии описывают основные контрольные точки продвижения и успехов системы по ходу жизненного цикла. Такие сегменты дают упорядоченное продвижение системы через установленные пересмотры выделения ресурсов, что снижает риски и обеспечивает удовлетворительное продвижение. Основной причиной применения описаний жизненного цикла является потребность в принятии решений по определенным критериям до продвижения системы на следующую стадию.

Стадии жизненного цикла

Стадии жизненного цикла (жёстко стандартом не предписываются)

Стадии могут сосуществовать или возобновляться после завершения

Стадии определяют основные точки принятия решений, вехи проекта (decision gates)

Понятие зафиксированного состояния (baseline): после формального закрепления принятых по проекту решений, изменение которых можно провести только через формальную процедуру

Стадии создания систем (ISO/IEC 15288)

№ п./п	Стадия	Описание
1	Формирование концепции	Анализ потребностей, выбор концепции и проектных решений
2	Разработка	Проектирование системы
3	Реализация	Изготовление системы
4	Эксплуатация	Ввод в эксплуатацию и использование системы
5	Поддержка	Обеспечение функционирования системы
6	Снятие с эксплуатации	Прекращение использования, демонтаж, архивирование системы

Представления жизненного цикла системы

У системы есть два основных представления: целевое (архитектурное, чаще всего структурное в своей основе, плюс процессы времени эксплуатации системы) и жизненного цикла (развертка во времени жизненного цикла - процессы обеспечивающих систем).

Языков представления жизненного цикла и текстовых и графических нотаций для этих языков много, ограничимся для примера лишь следующими:

- «Нарезанная колбаска»
- V-диаграмма

«Нарезанная колбаска»

Просто перечисление стадий жизненного цикла их названиями, для выразительности названия упакованы в отрезки "колбаски"

Замысел	Разработка	Производство	Использование	Поддержка	Прекращение использования
---------	------------	--------------	---------------	-----------	---------------------------

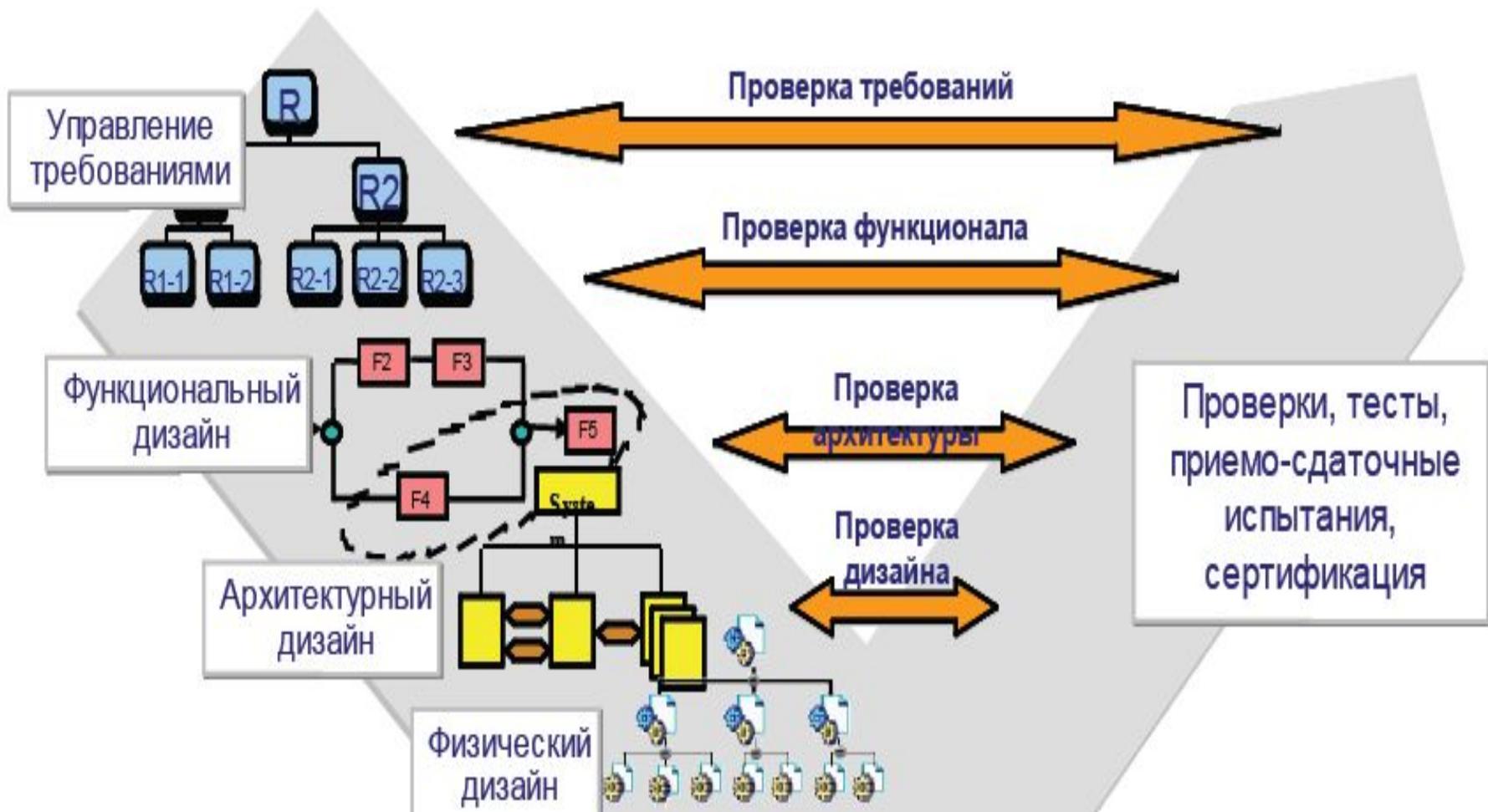
Разнообразие жизненных циклов

Софт	Концепция	Разработка	Поддержка	Списание		
Оборудование	Идея	Проектирование	Изготовление	Эксплуатация и поддержка	Списание	
Персонал	Определение требуемых компетенций	Приобретение	Обучение	Использование и рост	Отставка	
Здание	Визуализация	Проектирование сооружения и площадки	Согласование	Строительство	Эксплуатация и поддержка	Разборка
Природный ресурс	Приобретение	Разработка	Эксплуатация	Рекультивация		
Процесс	Определение выхода	Графическое представление	Описание	Пилотное внедрение	Использование и совершенствование	Ликвидация
Система	Идея	Разработка	Изготовление	Использование	Поддержка	Списание

V – модель



Пример V-модели



Информационные системы

- В *широком смысле* информационная система есть совокупность технического программного, и организационного обеспечения, а также персонала, предназначенная для того, чтобы своевременно обеспечивать надлежащих людей надлежащей информацией.
- Одно из наиболее широких определений ИС дал М.Р. Когаловский: «информационной системой называется комплекс, включающий вычислительное и коммуникационное оборудование, программное обеспечение, лингвистические средства и информационные ресурсы, а также системный персонал и обеспечивающий поддержку динамической информационной модели некоторой части реального мира для удовлетворения информационных потребностей пользователей»
- В *узком смысле* информационной системой называют только подмножество компонентов ИС в широком смысле, включающее базы данных, СУБД и специализированные прикладные программы.
- ИС в узком смысле рассматривают как программно-аппаратную систему, предназначенную для автоматизации целенаправленной деятельности конечных пользователей, обеспечивающую, в соответствии с заложенной в нее логикой обработки, возможность получения, модификации и хранения информации

Жизненный цикл информационной системы

Совокупность стадий и этапов, которые проходит ИС в своем развитии от момента принятия решения о создании системы до момента прекращения функционирования системы, называется жизненным циклом ИС.

Стадии жизненного цикла информационной системы

1. Планирование и анализ требований (предпроектная стадия) — системный анализ. Проводится исследование и анализ существующей информационной системы, определяются требования к создаваемой ИС, формируются технико-экономическое обоснование (ТЭО) и техническое задание (ТЗ) на разработку ИС;
2. Проектирование (техническое и логическое проектирование). В соответствии с требованиями формируются состав автоматизируемых функций (функциональная архитектура) и состав обеспечивающих подсистем (системная архитектура), проводится оформление технического проекта ИС;
3. Реализация (рабочее и физическое проектирование, кодирование). Разработка и настройка программ, формирование и наполнение баз данных, формулировка рабочих инструкций для персонала, оформление рабочего проекта;
4. Внедрение (опытная эксплуатация). Комплексная отладка подсистем ИС, обучение персонала, поэтапное внедрение ИС в эксплуатацию по подразделениям организации, оформление акта о приемо-сдаточных испытаниях ИС;
5. Эксплуатация ИС (сопровождение, модернизация). Сбор рекламаций и статистики о функционировании ИС, исправление недоработок и ошибок, оформление требований к модернизации ИС и ее выполнение (повторение стадий 2-5).

Модели жизненного цикла ИС

- каскадная модель (до 70-х годов) — последовательный переход на следующий этап после завершения предыдущего;
- итерационная модель (70-80-е годы) — с итерационными возвратами на предыдущие этапы после выполнения очередного этапа;
- спиральная модель (80-90-е годы) — прототипная модель, предполагающая постепенное расширение прототипа ИС.

Каскадная модель (водопадная, предложена Винстоном Ройсом)



АНАЛИЗ КАСКАДНОЙ МОДЕЛИ

Достоинства

Последовательное выполнение этапов проекта в строгом фиксированном порядке

Позволяет оценивать качество продукта на каждом этапе

Недостатки

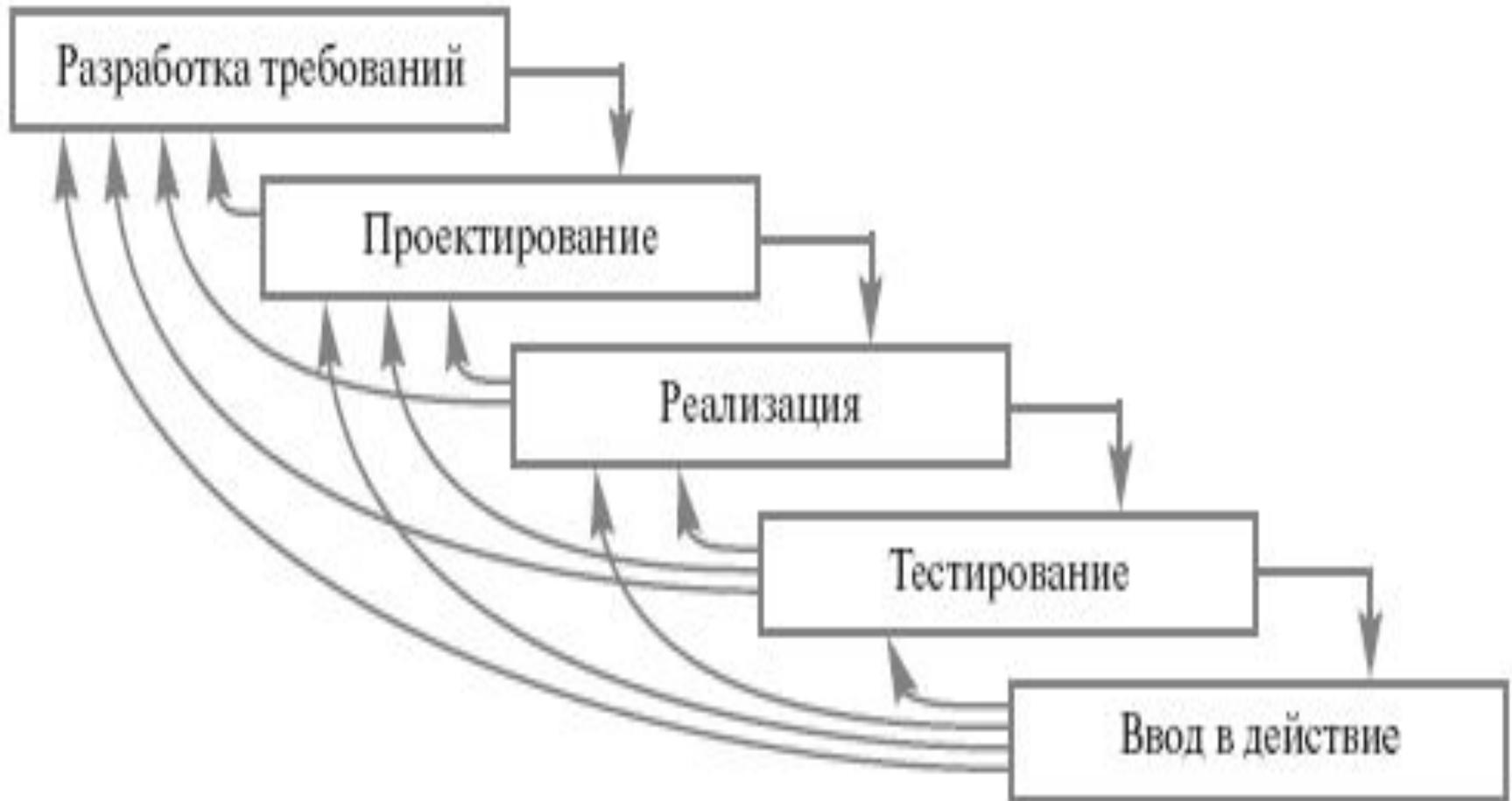
Отсутствие обратных связей между этапами

Недостатки каскадной модели особо остро проявляются в случае, когда трудно (или невозможно) сформулировать требования или требования могут меняться в процессе выполнения проекта. В этом случае разработка ПО имеет принципиально циклический характер.

Каскадная модель актуальна в следующих случаях

- Требования и их реализация максимально четко определены и понятны
используется неизменяемое определение продукта и вполне понятные
технические методики. Это задачи типа:
 - ✓ научно-вычислительного характера (пакеты и библиотеки научных программ типа расчета несущих конструкций зданий, мостов, ...);
 - ✓ операционные системы и компиляторы;
 - ✓ системы реального времени управления конкретными объектами;
- Повторная разработка типового продукта (автоматизированного Бухгалтерского учета, начисления зарплаты, ...).
- Выпуск новой версии уже существующего продукта, если вносимые изменения вполне определены и управляемы (перенос уже существующего продукта на новую платформу).
- И наконец, принципы каскадной модели находят применение как элементы моделей других типов, о чем речь пойдет ниже.

Итерационная модель



Анализ итерационной модели

В ходе разработки всегда выявляются дополнительные требования или изменяются выявленные ранее. Также появляются новые ограничения, связанные с принятыми техническими решениями. В наиболее полной мере их удастся учесть именно в итерационной разработке, поскольку именно при таком подходе руководство проекта в полной мере готово к изменениям.

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.*

Эволюционная разработка – один из основных факторов, определяющих создание современного программного обеспечения

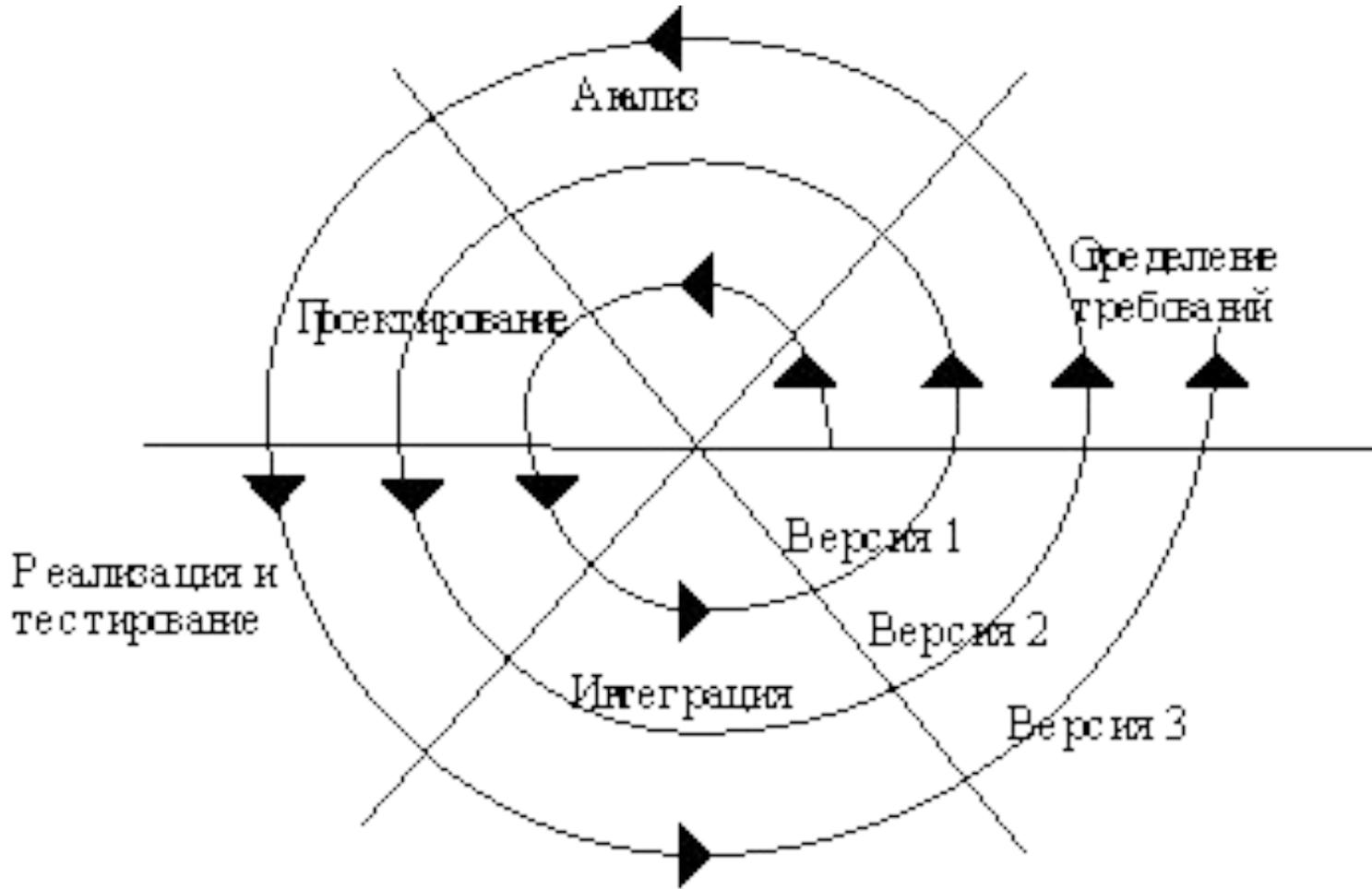
Окончание очередного витка эволюционного цикла разработки версии продукта, приводит к получению кода, написанного на некотором языке программирования. **Новый виток** спирали требует расширения и модификации этого кода, то есть его **эволюции**.

Спиральная модель (описана Б. Боэмом в 1988 году)

Разработка вариантов продукта представляется как набор циклов раскручивающейся спирали. Каждому циклу спирали соответствует такое же количество стадий, как и в модели каскадного процесса. При этом, начальные стадии, связанные с анализом и планированием представлены более подробно с добавлением новых элементов. В каждом цикле выделяются четыре базовые фазы:

- определение целей, альтернативных вариантов и ограничений.
- оценка альтернативных вариантов, идентификация и разрешение рисков.
- разработка продукта следующего уровня.
- планирование следующей фазы.

Представление спиральной модели



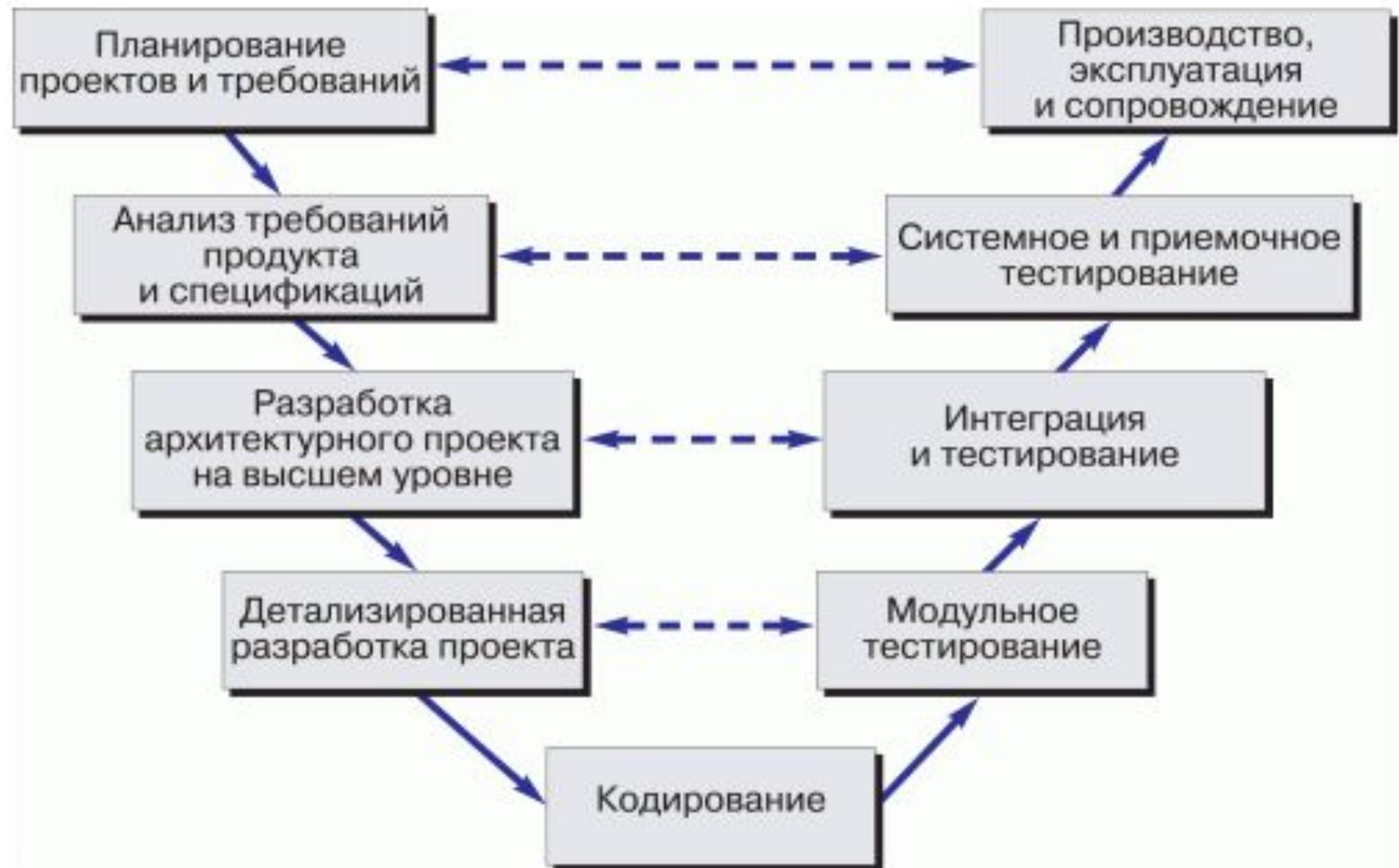
Достоинства спиральной модели

- Более тщательное проектирование (несколько начальных итераций) с оценкой результатов проектирования, что позволяет выявить ошибки проектирования на более ранних стадиях.
- Поэтапное уточнение требований в процессе выполнения итераций, что позволяет более точно удовлетворить требованиям заказчика
- Участие заказчика в выполнении проекта с использованием прототипов программы. Заказчик видит, что и как создается, не выдвигает необоснованных требований, оценивает реальные объемы финансирования.
- Планирование и управление рисками при переходе на следующие итерации позволяет разумно планировать использование ресурсов и обосновывать финансирование работ.
- Возможность разработки сложного проекта «по частям», выделяя на первых этапах наиболее значимые требования.

Недостатки спиральной модели

- Сложность анализа и оценки рисков при выборе вариантов.
- Сложность поддержания версий продукта (хранение версий, возврат к ранним версиям, комбинация версий)
- Сложность оценки точки перехода на следующий цикл
- Бесконечность модели – на каждой витке заказчик может выдвигать новые требования, которые приводят к необходимости следующего цикла разработки.

V модель (разработка через тестирование)



Анализ V-образной модели

Целями итераций в этой модели является обеспечение процесса тестирования.

Тестирование продукта обсуждается, проектируется и планируется на ранних этапах жизненного цикла разработки. План испытания приемки заказчиком разрабатывается на этапе планирования, а компоновочного испытания системы - на фазах анализа, разработки проекта и т.д. Этот процесс разработки планов испытания обозначен пунктирной линией между прямоугольниками V-образной модели. Помимо планов, на ранних этапах разрабатываются также и тесты, которые будут выполняться при завершении параллельных этапов.

Выбор приемлемой модели жизненного цикла ПО ИС

Следует проанализировать следующие отличительные категории проекта:

- Требования.
- Команда разработчиков.
- Коллектив пользователей.
- Тип проекта и риски.

Выбор модели производят с помощью набора матриц, представляющих характеристики требований, участников команды разработчиков, коллектива пользователей, типа проектов и рисков.

Матрица выбора модели жизненного цикла на основе характеристик требований

Требования	Каскад- Ная	V-образ- ная	Протопа- пирован	Спираль- ная	RAD	Инкре- ментная
Являются ли требования легко определяемыми и/или хорошо известными?	Да	Да	Нет	Нет	Да	Нет
Могут ли требования заранее определяться в цикле?	Да	Да	Нет	Нет	Да	Да
Часто ли будут изменяться требования в цикле?	Нет	Нет	Да	Да	Нет	Нет
Нужно ли демонстрировать требования с целью определения?	Нет	Нет	Да	Да	Да	Нет
Требуются ли для демонстрации возможности проверки концепции?	Нет	Нет	Да	Да	Да	Нет
Будут ли требования отражать сложность системы?	Нет	Нет	Да	Да	Нет	Да
Обладает ли требование функциональными свойствами на раннем этапе?	Нет	Нет	Да	Да	Да	Да

Вывод в результате анализа матрицы требований

Если требования не могут быть заранее определены, а в ходе работ будут часто изменяться, то наиболее подходящими являются модель прототипирования и спиральная модель

ЗАКЛЮЧЕНИЕ

Системный Инжиниринг отвечает за **всю картину в целом**, обеспечивая выполнение требований в течение всего жизненного цикла изделия.

- Повышается вероятность успеха создания Системы
 - ▶ Понимание природы Системы и ее поведения в окружающей среде
 - ▶ Определение характеристик Системы с точки зрения пользователя
- Уменьшается вероятность принятия неправильных решений
 - ▶ Управление рисками
 - Идентификация и оценка рисков и возможных проблем
 - Формирование планов решения проблем (*парирование*)
 - Мониторинг условий возникновения и динамики развития рисков
 - ▶ Поиск неопределенностей и изменяемых параметров
 - ▶ Учет требований нормативных документов и общих ограничений по Программе
 - Контроль за расходом бюджета (*общего, по проекту, па пакету работ, по работе*)
 - Мониторинг ресурсов (*квалифицированный персонал, производственные мощности*)
- Уменьшение общей стоимости жизненного цикла изделия
 - ▶ Улучшение процесса принятия решений в планировании, разработке, эксплуатации
- Такой подход начинается с понимания **потребностей заказчика**, определения **функциональности изделия** и обязательных **запланированных проверок** (*аттестаций, приемочных испытаний, контроля*) на самых ранних стадиях жизненного цикла создания изделия

