



Технология программирования

Литература

- Цилькер Б.Я., Орлов С. А. Технологии разработки программного обеспечения. СПб.: Питер, 2012. – 608.
- Соммервилл И. Инженерия программного обеспечения. – М.: Вильямс, 2002. – 624 с.
- Буч Г., Рамбо Д., Якобсон И. Введение в UML от создателей языка. 2-е изд. – М.: ДМК Пресс, 2011. – 496 с.

Литература

- Фредерик П. Брукс. Проектирование процесса проектирования: записки компьютерного эксперта.: Пер. с англ. - М.: ООО "И.Д. Вильямс", 2012. – 464 с.
- Эванс Э. Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем.: Пер. с англ. - М.: ООО "И.Д.Вильямс", 2011. – 448 с.
- Гудлиф П. Ремесло программиста. Практика написания хорошего кода. – Пер. с англ. – СПб.: СимволПлюс, 2009. – 704 с.
- Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. – СПб.: Питер, 2010. – 464 с.
- Тидвелл ДЖ. Разработка пользовательских интерфейсов. 2-е изд. – СПб.: Питер, 2011. – 480 с.
- Резник С., Бьерк А. Scrum с Team Foundation Server 2010. Профессиональный подход. – М.: ЭКОМ, 2012 – 416 с.

Литература

- Вигерс К. Разработка требований к программному обеспечению/Пер. с англ. – М.: Русская Редакция, 2012. – 576 с
- ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств. Гостандарт.:М, 2000. – 56 с.
- <http://ood.asf.ru/> - объектно-ориентированный анализ, проектирование и программирование,
- <http://www.implementingscrum.com> – Scrum
- Сазерленд Д. Принципы и значение гибкой разработки <http://msdn.microsoft.com/ru-ru/library/dd997578.aspx>

Литература

- Тидвелл ДЖ. Разработка пользовательских интерфейсов. 2-е изд. – СПб.: Питер, 2011. – 480 с.
- Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. – СПб.: Питер, 2010. – 464 с.
- Вигерс К. Разработка требований к программному обеспечению/Пер. с англ. – М.: Русская Редакция, 2012. – 576 с
- ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств. Гостандарт.: М, 2000. – 56 с.
- <http://ood.asf.ru/> - объектно-ориентированный анализ, проектирование и программирование,
- <http://www.implementingscrum.com> – Scrum
- Сазерленд Д. Принципы и значение гибкой разработки <http://msdn.microsoft.com/ru-ru/library/dd997578.aspx>



- Руководство по разработке качественных требований к программному обеспечению.
- Описаны приемы выявления, формулирования, разработки, проверки, утверждения и тестирования требований, которые помогут создать эффективное ПО.
- Издание дополнено новыми приемами, посвященными разработке требований в проектах гибкой разработки (agile).

ИАН СОММЕРВИЛЛ

Инженерия программного обеспечения

6-е издание



- Руководство по разработке качественных требований к программному обеспечению.
- Описаны приемы выявления, формулирования, разработки, проверки, утверждения и тестирования требований, которые помогут создать эффективное ПО.
- Издание дополнено новыми приемами, посвященными разработке требований в проектах гибкой разработки (agile).



Введение в программирование

Что будем делать?

- Простые «учебные» программы
- Новые системы, создаваемые с чистого листа
- Расширения существующих программ
- Сопровождение старой базы кода

Программирование – это искусство?

- Чем руководствуетесь – инстинктом или планом?
- Как понять, что хочет заказчик?

Ваша программа



Как ее видит пользователь



МАГИЯ ПРОГРАМИРОВАНИЯ
не понятна 95% населения

Программное обеспечение

- Программное обеспечение – совокупность всех программ, хранящихся на всех устройствах долговременной памяти компьютера, а также сопутствующая документация и конфигурационные данные

```
graph TD; A[Программное обеспечение] --> B[Системное]; A --> C[Прикладное]; A --> D["Инструментальное (системы программирования)"]
```

Системное

Прикладное

Инструментальное
(системы
программирования)

Программирование

- Цель программирования – описание процессов обработки данных
- Данные – это представление фактов и идей в формализованном виде, пригодном для передачи и переработке в некоем процессе.
- Информация – это смысл, который придается данным при их представлении.



Программирование

- Обработка данных – выполнение систематической последовательности действий с данными.
 - Данные представляются и хранятся на носителях данных.
- Совокупность носителей данных, используемых при какой-либо обработке данных – информационная среда
- Набор данных, содержащихся в какой-либо момент в информационной среде – состояние этой информационной среды

Программирование

- Процесс – последовательность сменяющих друг друга состояний некоторой информационной среды
- Описать процесс, значит, определить последовательность состояний заданной информационной среды
- Чтобы по заданному описанию требуемый процесс порождался автоматически на каком-либо компьютере, необходимо, чтобы это описание было формализованным – такое описание называется программой

Программирование

- Программное средство – программа или логически связанная совокупность программ на носителях данных, снабженная программной документацией

Программный продукт

- Общие программные продукты – автономные программные системы, которые созданы для продажи на открытом рынке программных продуктов любому потребителю
 - Примеры: системы управления базами данных, текстовые процессоры, графические пакеты, средства управления проектами, планировщики задач и т.п.
 - Распространяются как обычный товар «в коробке» или по Интернету
 - Клиент – массовый потребитель
 - Стандартные операции установки на машину клиента

Программный продукт

- Программные продукты, созданные на заказ - программные системы, которые создаются по заказу определенного потребителя
 - Примеры: системы поддержки определенных производственных или бизнес-процессов и т.п.
 - Разрабатываются специально для данного потребителя согласно заключенному контракту
 - Распространяются как внедрение в аппаратно-программной системе клиента – «решение»
 - Клиент – «заказчик» - организация (в широком смысле) со своими особыми требованиями
 - Развертывание и установка с учетом особенностей клиента

Системный программный продукт



Разработка программного обеспечения

- Хаотическая деятельность – "code and fix" ("пишем и правим")
 - единого плана не существует,
 - общий проект представляет собой просто смесь краткосрочных решений
- Технология превращает создание программного продукта в упорядоченный процесс
 - работа программиста более прогнозируемая и эффективна
 - создается детальное описание процесса создания системы, особое место в котором занимает планирование (аналогично другим инженерным дисциплинам)
- Облегчённые (lightweight) или гибкие (agile) технологии
 - меньшая ориентация на документацию
 - ориентированность на код

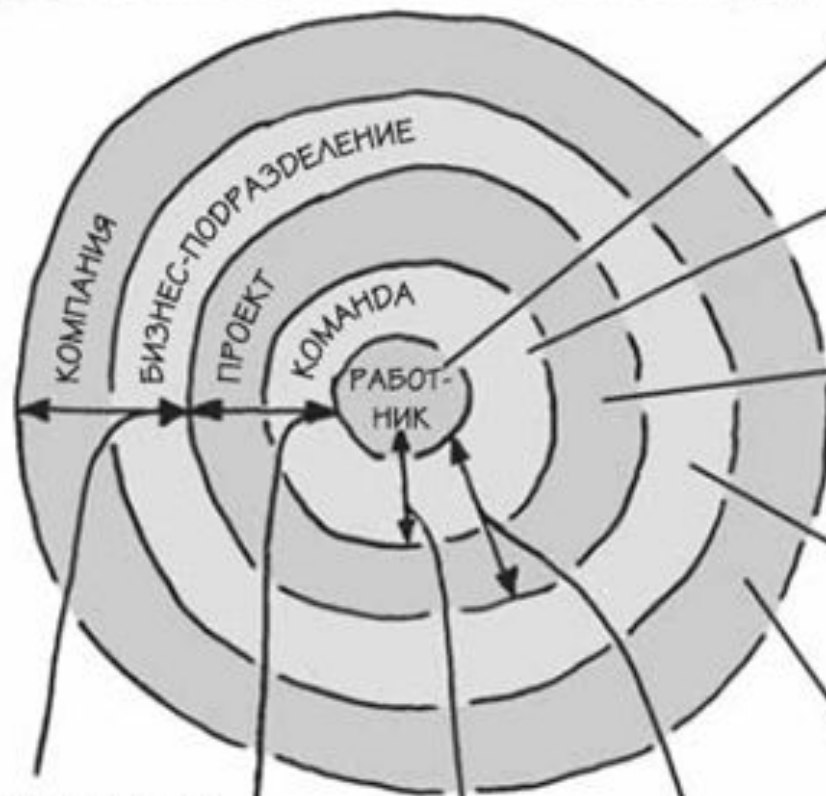
Командная разработка

- Работа в команде является необходимым условием успешности проекта
- Умение работать в команде – важное качество высококвалифицированного разработчика программного обеспечения

Уровни команд

- Отдельный программный компонент, входящий в более крупный проект;
- Компонент должен войти в более общий продукт.
- Разработка нескольких проектов одновременно

УРОВНИ ГРУППОВОЙ РАБОТЫ



Динамика организации
(верхний уровень
групповой работы)

Динамика группы
(нижний уровень
групповой работы)

Взаимодействие
внутри команды

Взаимодействие
между командами

Требует:

- умения разрабатывать ПО
- умения мыслить
- обучения
- мотивации

Вводит:

- социальные навыки
- динамику внутри команды

Объединяет несколько маленьких команд, вводя динамику между ними; большие усилия по координации. Особого внимания требуют:

- обмен данными
- планирование
- управление ресурсами

Каждый проект должен решать задачи бизнес-подразделения. На этом уровне на работу влияют:

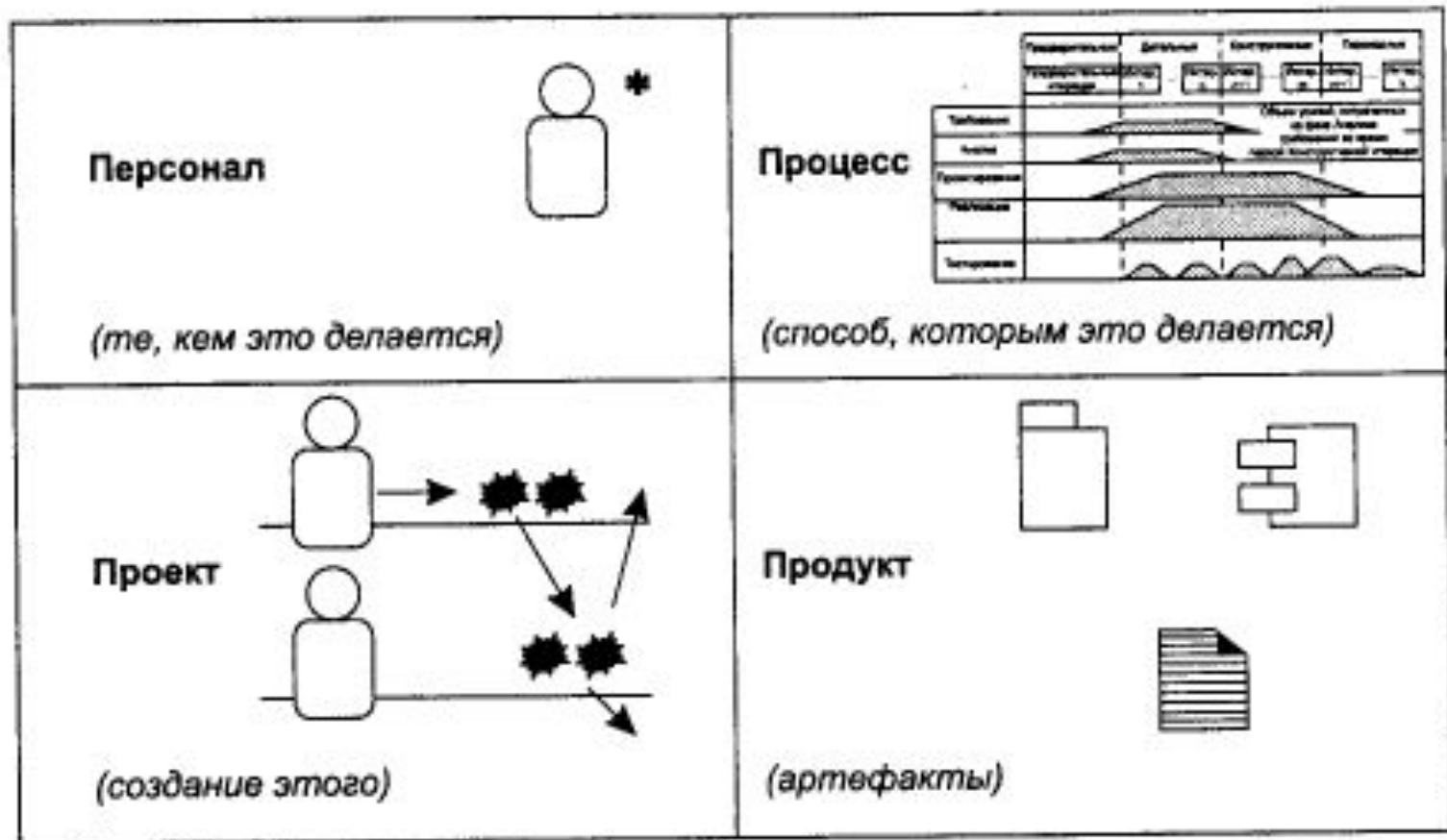
- политика корпорации
- культура
- процедуры

Корпоративный контекст:

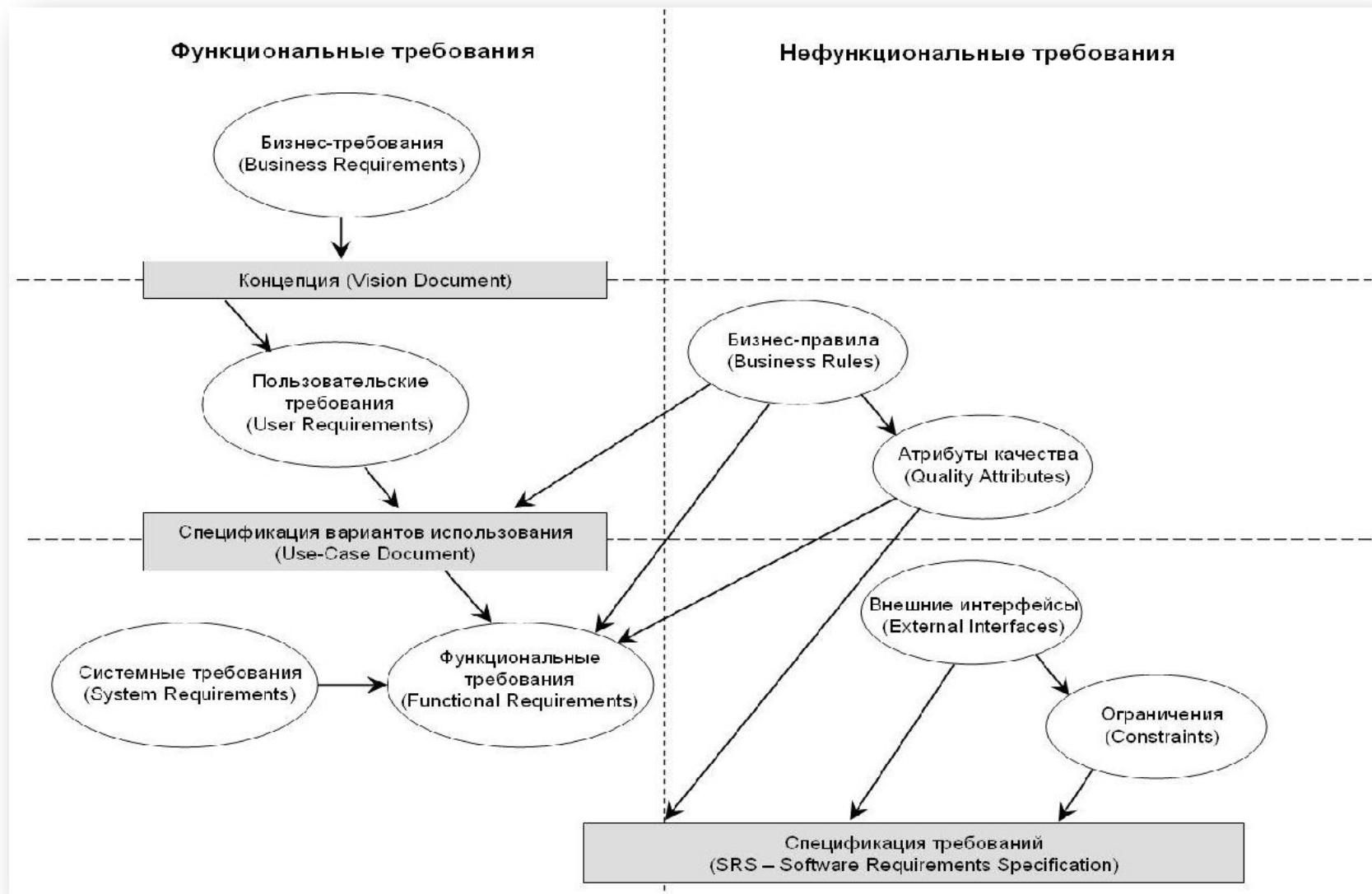
- взаимодействие с другими компаниями (клиентами, поставщиками)
- бизнес-стратегия
- схемы стимулирования

Система разработки ПО

- Система разработки программного обеспечения включает в себя *персонал, процесс, проект* и *продукт*



Функциональные и нефункциональные требования к программному средству



- *Бизнес-требования* (business requirements) содержат высокоуровневые цели организации или заказчиков системы
- *Требования пользователей* (user requirements) описывают цели и задачи, которые пользователям позволит решить система
- *Функциональные требования* (functional requirements) определяют функциональность ПО, которую разработчики должны построить, чтобы пользователи смогли выполнить свои задачи в рамках бизнес-требований
- *Системные требования* (system requirements) определяют высокоуровневые требования к продукту, которые содержат многие подсистемы
- *Бизнес-правила* (business rules) включают корпоративные политики, правительственные постановления, промышленные стандарты и вычислительные алгоритмы
- *Атрибуты качества* (quality attributes) представляют собой дополнительное описание функций продукта, выраженное через описание его характеристик, важных для пользователей или разработчиков

Рамки решения и рамки проекта

- Требования к системе («рамки решения») и соответствующие им задачи разработчика («рамки проекта») требования удобно свести в таблицу

URS ID	Описание «рамки решения»	FRS ID	Описание «рамки проекта»
Функция: <i>Название функции</i>			
1	Система обеспечивает возможность пользователям.....	101	Создание меню.....
		102	Реализация просмотра результатов....

Процесс и стадии создания ПО

Процесс создания ПО – совокупность мероприятий, целью которых является создание или модернизация ПО.

- Выделяют 4 основных мероприятия:
 - **Спецификация:** формулирование спецификаций определяет основные требования к ПО (что должна делать система).
 - **Разработка:** создание ПО в соответствии со спецификациями.
 - **Аттестация:** проверка ПО на соответствие потребностям заказчика.
 - **Модернизация:** развитие ПО в соответствии с изменившимися потребностями заказчика.
- **Постановка задачи** – составление точного и понятного словесного описания того, как должна работать будущая программа, что должен делать пользователь в процессе ее работы.
- **Разработка проекта системы** – создание модели, отражающей основные функциональные требования, предъявляемые к программе.
- **Программирование** – создание программного кода на выбранном языке программирования.
- **Тестирование и отладка программы** – проверка правильности ее работы.
- **Создание документации.**

Стандартизация проектирования ПО

- ГОСТ 34.601-90 - распространяется на автоматизированные системы и устанавливает стадии и этапы их создания
- ISO/IEC 12207 - стандарт на процессы и организацию *жизненного цикла*. Распространяется на все виды заказного ПО
- ГОСТ Р ИСО/МЭК 15288 — 2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем
- Custom Development Method (методика Oracle)
- Rational Unified Process (RUP)
- Microsoft Solution Framework (MSF)

ISO/IEC 12207

- Международный стандарт ISO/IEC 12207
 - (ISO - International Organization of Standardization - Международная организация по стандартизации,
 - IEC - International Electrotechnical Commission - Международная комиссия по электротехнике).
- ГОСТ Р ИСО/МЭК 2207-99 содержит полный аутентичный текст международного стандарта ISO/IEC12207

Процессы, определенные в стандарте, образуют множество общего назначения.

Конкретная организация, в зависимости от своих целей, может выбрать соответствующее подмножество процессов для выполнения своих конкретных задач (адаптировать для конкретной организации, проекта или приложения).

Жизненный цикл программного обеспечения

- *Жизненный цикл* - это непрерывный процесс, который начинается
 - с момента принятия решения о необходимости его создания
 - и заканчивается в момент его полного изъятия из эксплуатации.

Структура ЖЦ ПО

Жизненный цикл ПО базируется на трех группах процессов:

- **основные** процессы

- реализуются под управлением основных сторон (заказчик, поставщик, разработчик, оператор и персонал сопровождения), вовлеченных в жизненный цикл программных средств

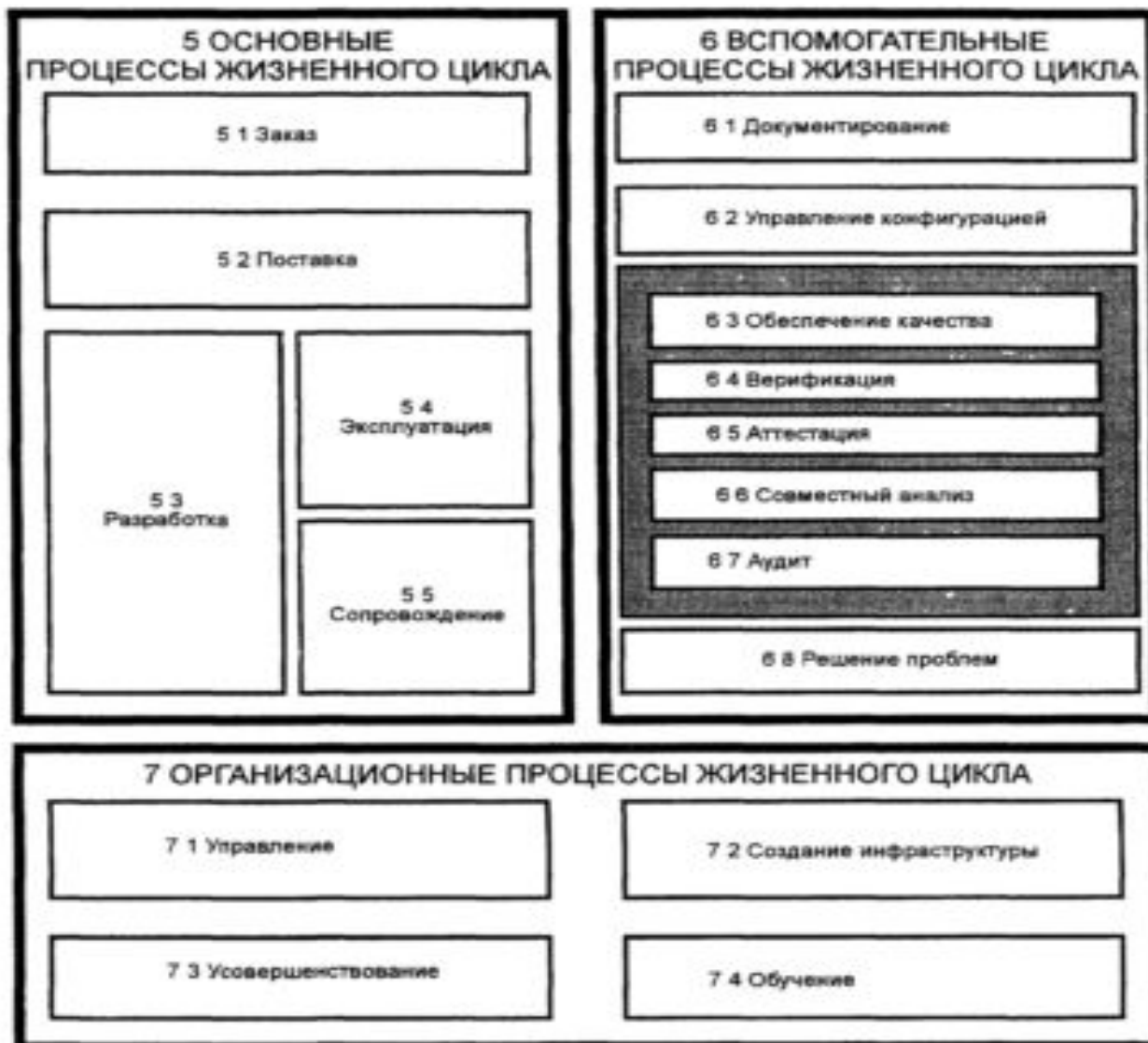
- **вспомогательные** процессы

- обеспечивают выполнение основных процессов;

- **организационные** процессы

- применяются для создания, реализации и постоянного совершенствования основной структуры, охватывающей взаимосвязанные процессы жизненного цикла и персонал.

Процессы жизненного цикла



Основные процессы

- Процесс заказа.
 - Определяет работы заказчика.
- Процесс поставки.
 - Определяет работы поставщика.
- Процесс разработки.
 - Определяет работы разработчика.
- Процесс эксплуатации.
 - Определяет работы оператора.
- Процесс сопровождения.
 - Определяет работы персонала сопровождения.
 - Охватывает снятие с эксплуатации программного продукта



Вспомогательные процессы

- Процесс документирования.
- Процесс управления конфигурацией.
- Процесс обеспечения качества
- Процесс верификации.
- Процесс аттестации.
- Процесс совместного анализа.
- Процесс аудита.
- Процесс решения проблемы.



Организационные процессы

- Процесс управления.
- Процесс создания инфраструктуры.
- Процесс усовершенствования.
- Процесс обучения.

Модель жизненного цикла

- ГОСТ Р ИСО/МЭК 15288 — 2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем
- **Модель жизненного цикла** (life cycle model): Структурная основа процессов и действий, относящихся к жизненному циклу, которая также служит в качестве общей ссылки для установления связей и взаимопонимания сторон.
- **Процесс** (process) — совокупность взаимосвязанных и взаимодействующих видов деятельности, преобразующих входы в выходы



Стадии жизненного цикла

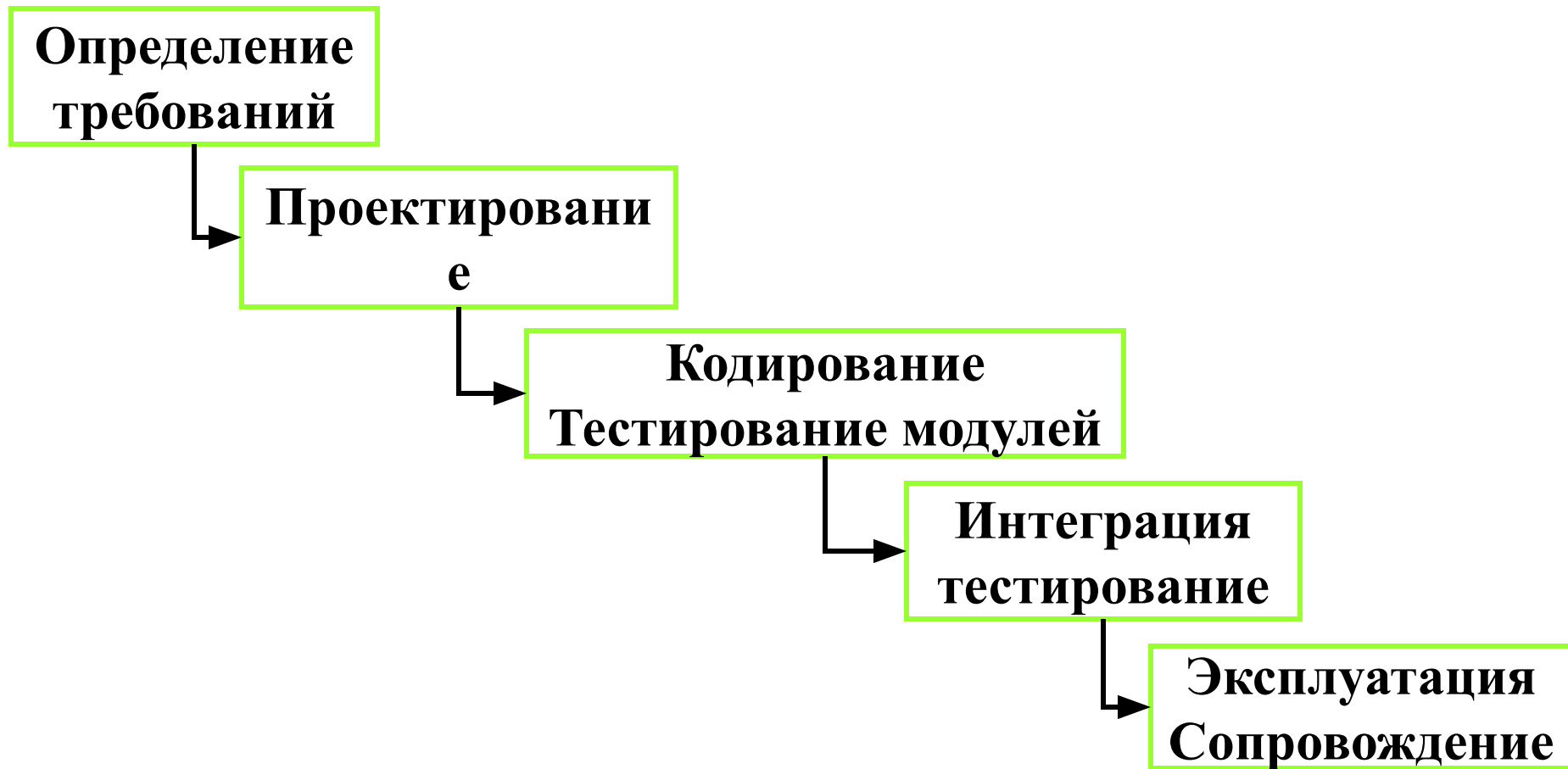
ГОСТ Р ИСО/МЭК 15288 — 2005 (Приложение В)

- Стадия замысла
- Стадия разработки
- Стадия производства
- Стадия применения
- Стадия поддержки применения
- Стадия прекращения применения и списания

Модели процесса

- **Классические модели процесса разработки ПО:**
 - **Каскадная модель (Waterfall model)**
фазы выполняются по порядку
 - **Эволюционная модель (Evolutionary development)**
фазы выполняются по порядку, процесс повторяется

Каскадная модель





■ Каскадная модель:

- Фиксированный набор стадий
- Каждая стадия -> законченный результат
- Стадия начинается, когда закончилась предыдущая.

■ Недостатки: негибкость

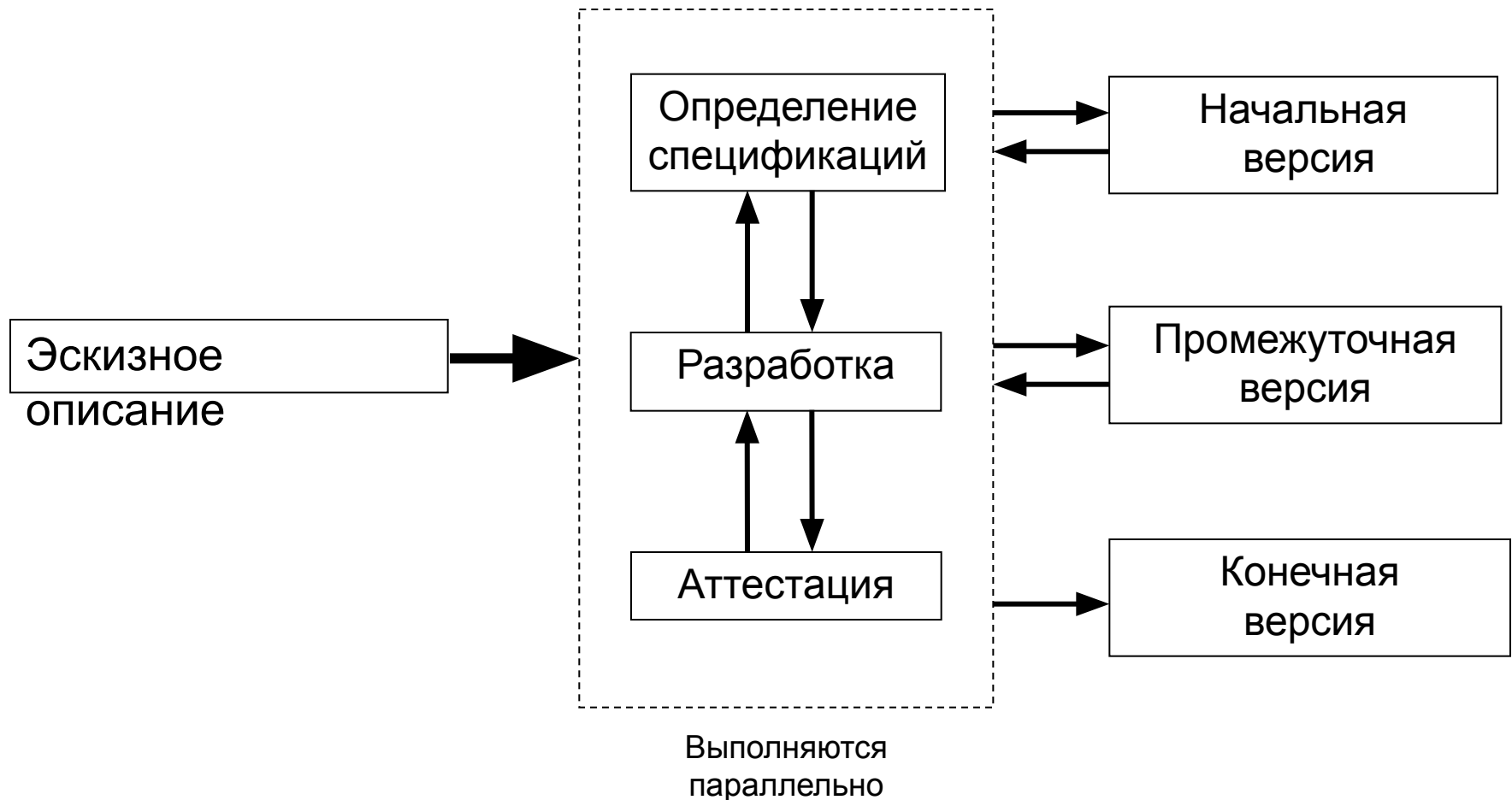
- фаза д.б. закончена, прежде чем приступить к следующей
- Набор фаз фиксирован
- Тяжело реагировать на изменения требований

■ Использование: там, где требования хорошо понятны и стабильны.

Эволюционная модель

- Стадии повторяются неоднократно.
 - для плохо сформулированных требований выполняется весь цикл работ по созданию работающего прототипа
 - уточняются требования
 - и все повторяется...
- На выходе – продукт, отвечающий потребностям пользователей.
- Недостатки:
 - Система часто плохо структурирована
 - Проект «не прозрачен»
 - Требуются средства для быстрой разработки
 - Подходит для малых и средних проектов

Эволюционная модель

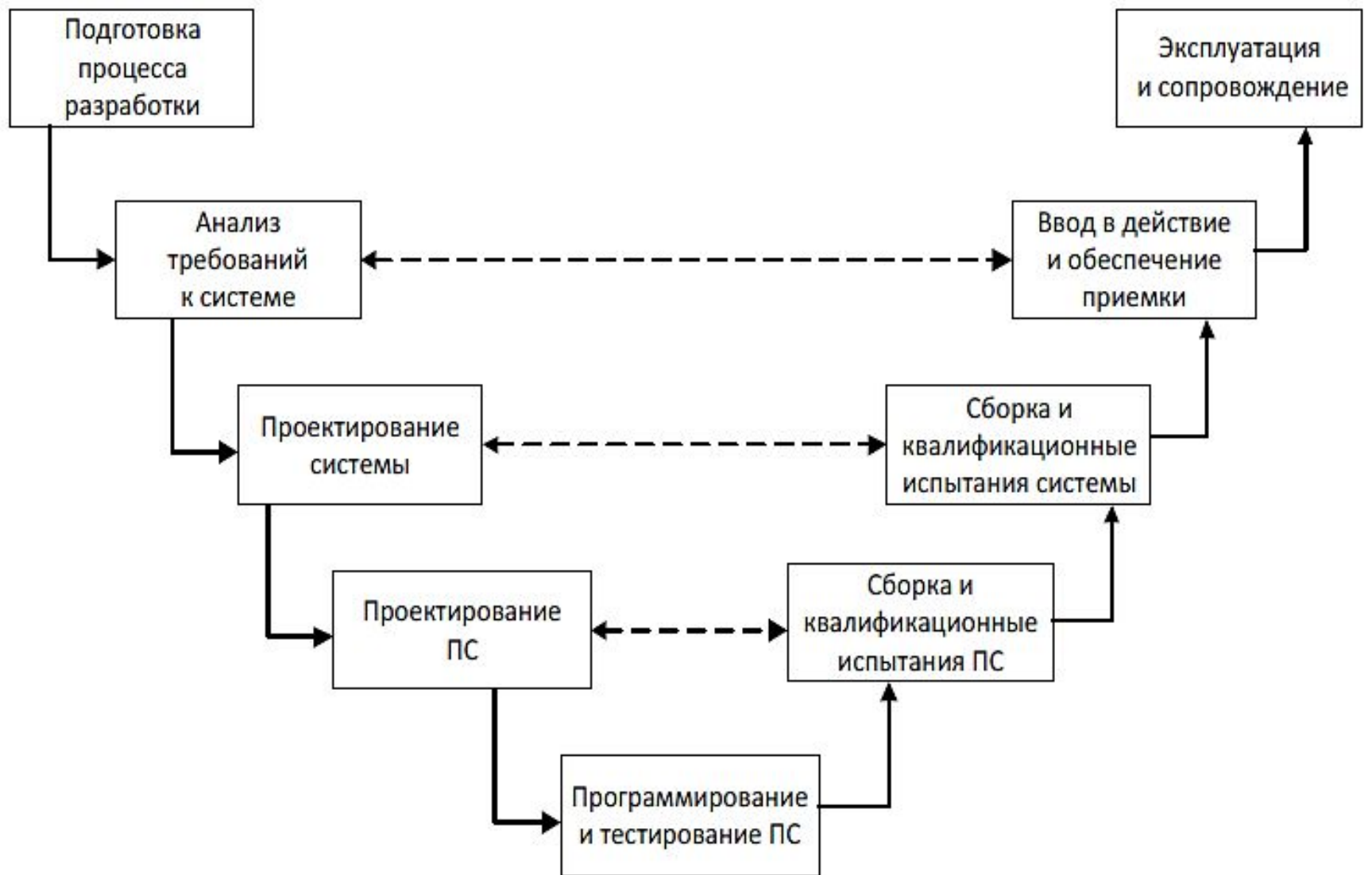


Прототип – действующий программный модуль, реализующий отдельные функции создаваемого ПО

Модель эволюционного прототипирования



V-образная модель



V-образная модель

- Основное назначение V-образной модели – обеспечение планирования тестирования программного средства на ранних стадиях проекта.
- Данная модель поддерживает каскадную стратегию однократного выполнения этапов процесса разработки ПС и базируется на предварительном полном формировании требований.
- В классической V-образной модели каждый шаг начинается после завершения предыдущего шага.
- Отличием V-образной модели от каскадной является то, что в ней выделены связи между шагами, предшествующими программированию, и соответствующими видами тестирования

V-образная модель

Достоинства

- планированием тестирования на ранних стадиях разработки программного средства;
- упрощение аттестации и верификации промежуточных результатов разработки;
- упрощение управления и контроля хода процесса разработки.

Недостатки

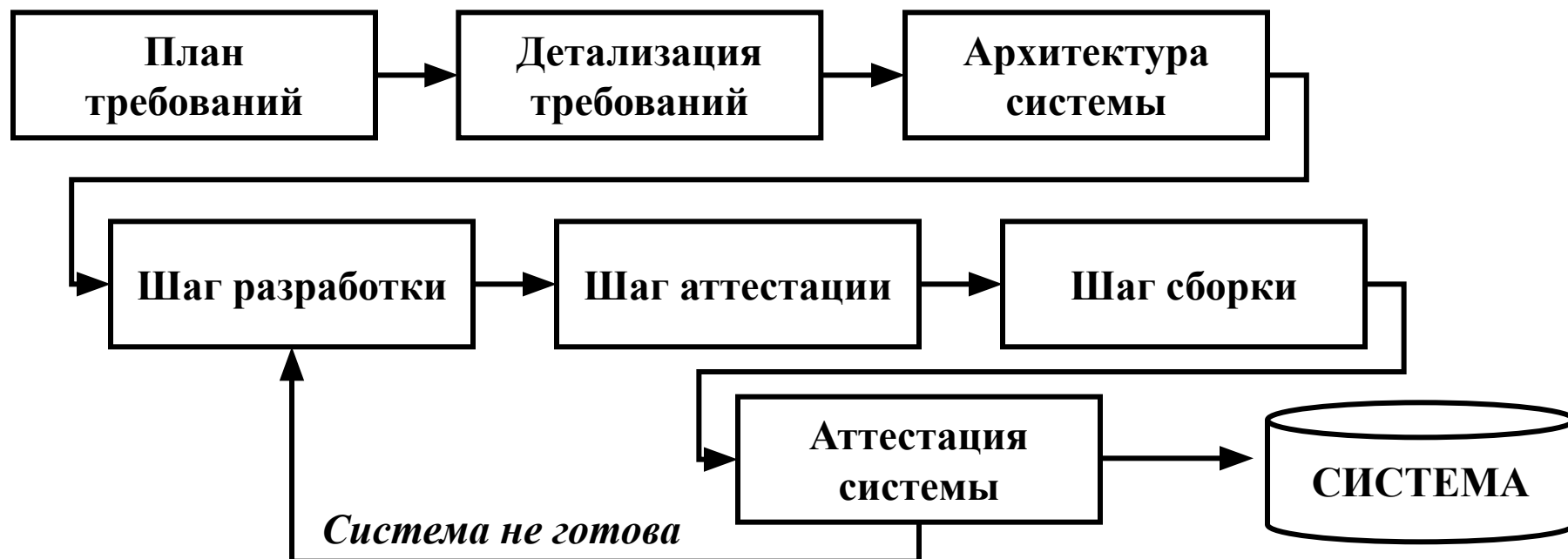
- поздние сроки тестирования требований в жизненном цикле, что оказывает существенное влияние на график выполнения проекта при необходимости изменения требований;
- отсутствие, как и в остальных каскадных моделях, действий, направленных на анализ рисков

Итерационный подход

- Часто подходы, перечисленные ранее, используется **в совокупности**.
- **Требования** всегда меняются в ходе разработки.
- К каждой из предыдущих моделей можно применить **итерации**.
- Следовательно, важна возможность выполнения итераций, результатом которых является прототип продукта с частичной функциональностью.
- Это достигается в итерационных моделях.
 - **Модель пошаговой разработки**
 - **Спиральная модель разработки**

Модель пошаговой разработки

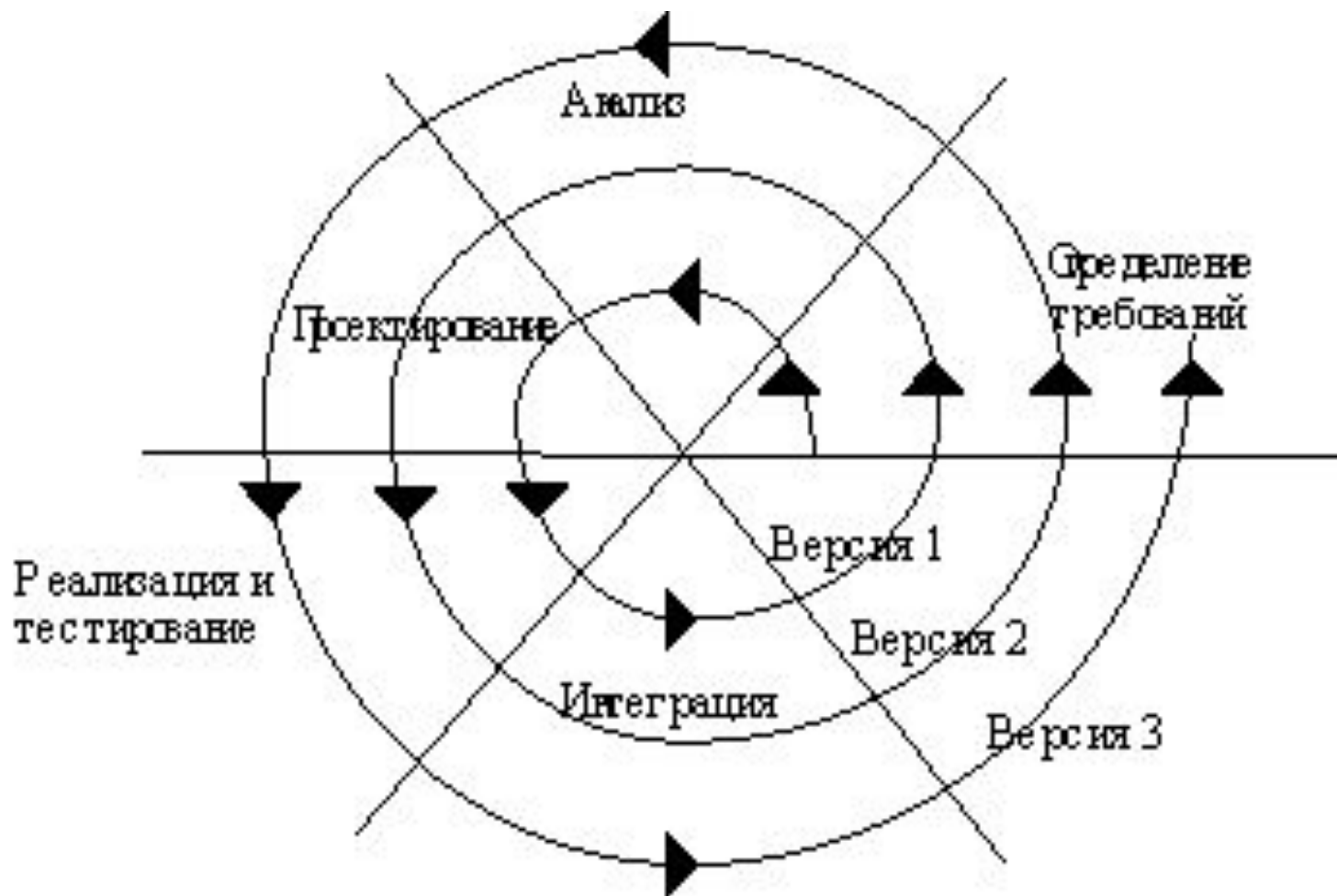
- Шаги. Каждый шаг – работающий прототип.
- Наиболее важные для заказчика компоненты – в начале.
- Требования фиксированы во время шага.
- Для шага можно применять каскадную или эволюционную модель.



Спиральная модель

- Вместо действий с обратной связью – спираль.
- Каждый виток спирали соответствует 1 итерации.
- Нет заранее фиксированных фаз. В зависимости от потребностей.
- Каждый виток разбит на 4 сектора:
 - Определение целей
 - Оценка и разрешение рисков
 - Разработка и тестирование
 - Планирование
- **Главное отличие: акцент на анализ и преодоление рисков.**
- На каждом витке могут применяться разные модели процесса разработки ПО.

Спиральная модель



Характеристики успешного проекта

- проект ориентирован на потребности пользователей,
- команда создает высокоуровневый план для реализации проекта,
- разработка ведется последовательно с регулярным уточнением плана,
- команда обладает эффективными средствами для адаптации к неизбежным изменениям.



Программная документация. Стандарты на разработку прикладных программных средств

- ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
- ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
- ГОСТ 34.603-92. Информационная технология. Виды испытаний автоматизированных систем.
- ГОСТ 19. Единая система программной документации.

ГОСТ 19.101-77 “Виды программ и программных документов”

Вид программного документа	Содержание программного документа
Спецификация	Состав программы и документации на нее
Ведомость держателей подлинников	Перечень предприятий, на которых хранят подлинники программных документов
Текст программы	Запись программы с необходимыми комментариями
Описание программы	Сведения о логической структуре и функционировании программы
Программа и методика испытаний	Требования, подлежащие проверке при испытании программы, а также порядок и методы их контроля
Техническое задание	Назначение и область применения программы, технические, технико-экономические и специальные требования, предъявляемые к программе, необходимые стадии и сроки разработки, виды испытаний
Пояснительная записка	Схема алгоритма, общее описание алгоритма и (или) функционирования программы, а также обоснование принятых технических и технико-экономических решений
Эксплуатационные документы	Сведения для обеспечения функционирования и эксплуатации программы

Эксплуатационные документы

Вид эксплуатационного документа	Содержание эксплуатационного документа
Ведомость эксплуатационных документов	Перечень эксплуатационных документов на программу
Формуляр	Основные характеристики программы, комплектность и сведения об эксплуатации программы
Описание применения	Сведения о назначении программы, области применения, применяемых методах, классе решаемых задач, ограничениях для применения, минимальной конфигурации технических средств
Руководство системного программиста	Сведения для проверки, обеспечения функционирования и настройки программы на условия конкретного применения
Руководство программиста	Сведения для эксплуатации программы
Руководство оператора	Сведения для обеспечения процедуры общения оператора с вычислительной системой в процессе выполнения программы
Описание языка	Описание синтаксиса и семантики языка
Руководство по техническому	Сведения для применения тестовых и диагностических

ГОСТ 19.102-77 “Стадии разработки”

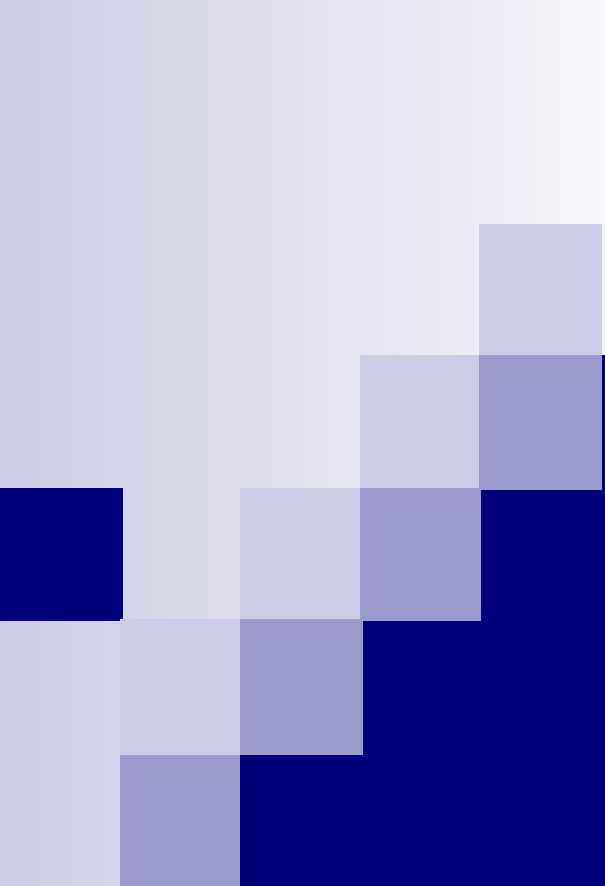
Стадии разработки	Этапы работ
1. Техническое задание	Обоснование необходимости разработки программы
	Научно-исследовательские работы
	Разработка и утверждение технического задания
2. Эскизный проект	Разработка эскизного проекта
	Утверждение эскизного проекта
3. Технический проект	Разработка технического проекта
	Утверждение технического проекта
4. Рабочий проект	Разработка программы
	Разработка программной документации
	Испытания программы
5. Внедрение	Подготовка и передача программы

Техническое задание должно содержать следующие разделы:

- Введение
- Основания для разработки
- Назначение разработки
- Требования к программе или программному изделию
- Требования к программной документации
- Технико-экономические показатели
- Стадии и этапы разработки
- Порядок контроля и приемки

Техническое задание должно содержать следующие разделы:

- Аннотация
- Общие положения
- Общие требования к системе
- Требования к обслуживанию
- Требования к аппаратному обеспечению
- Требования к программному обеспечению
- Требования к обеспечению конфиденциальности и защиты от несанкционированного доступа
- Требования обслуживанию



Практическое занятие 1

Спецификация требований
и создание вариантов
использования

Практическое занятие 1

■ ***Цель***

- Уяснить принципы документирования требований к ПО

■ ***Порядок выполнения***

- Уяснить правила программной системы (игры «Угадай Чет-Нечет»)
- Выявить требования к программе
- Разработать спецификацию требований и варианты использования
- Оформить отчет