Модели жизненного цикла ПО.

Модели жизненного цикла ПО.

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

Методики построения моделей можно разделить на 3 основных направления:

- Инженерный подход
- С учетом специфики задачи
- Современные технологии быстрой разработки

Теперь рассмотрим непосредственно существующие модели (подклассы) и оценим их преимущества и недостатки.

Классификация моделей ЖЦ

Стандарт ISO/IEC 12207 не предлагает конкретную модель ЖЦ и методы разработки ПО (под моделью ЖЦ понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении ЖЦ. Модель ЖЦ зависит от специфики ИС и условий, в которых последняя создается и функционирует). Его регламенты являются общими для любых моделей ЖЦ, методологий и технологий разработки. Стандарт ISO/IEC 12207 описывает структуру процессов ЖЦ ПО, но не конкретизирует в деталях, как реализовать или выполнить действия и задачи, включенные в эти процессы.

- Наиболее часто говорят о следующих **моделях жизненного цикла:**
  - Каскадная (водопадная) или последовательная
  - Итеративная и инкрементальная – эволюционная (гибридная, смешанная)
  - Спиральная (spiral) или модель Боэма.

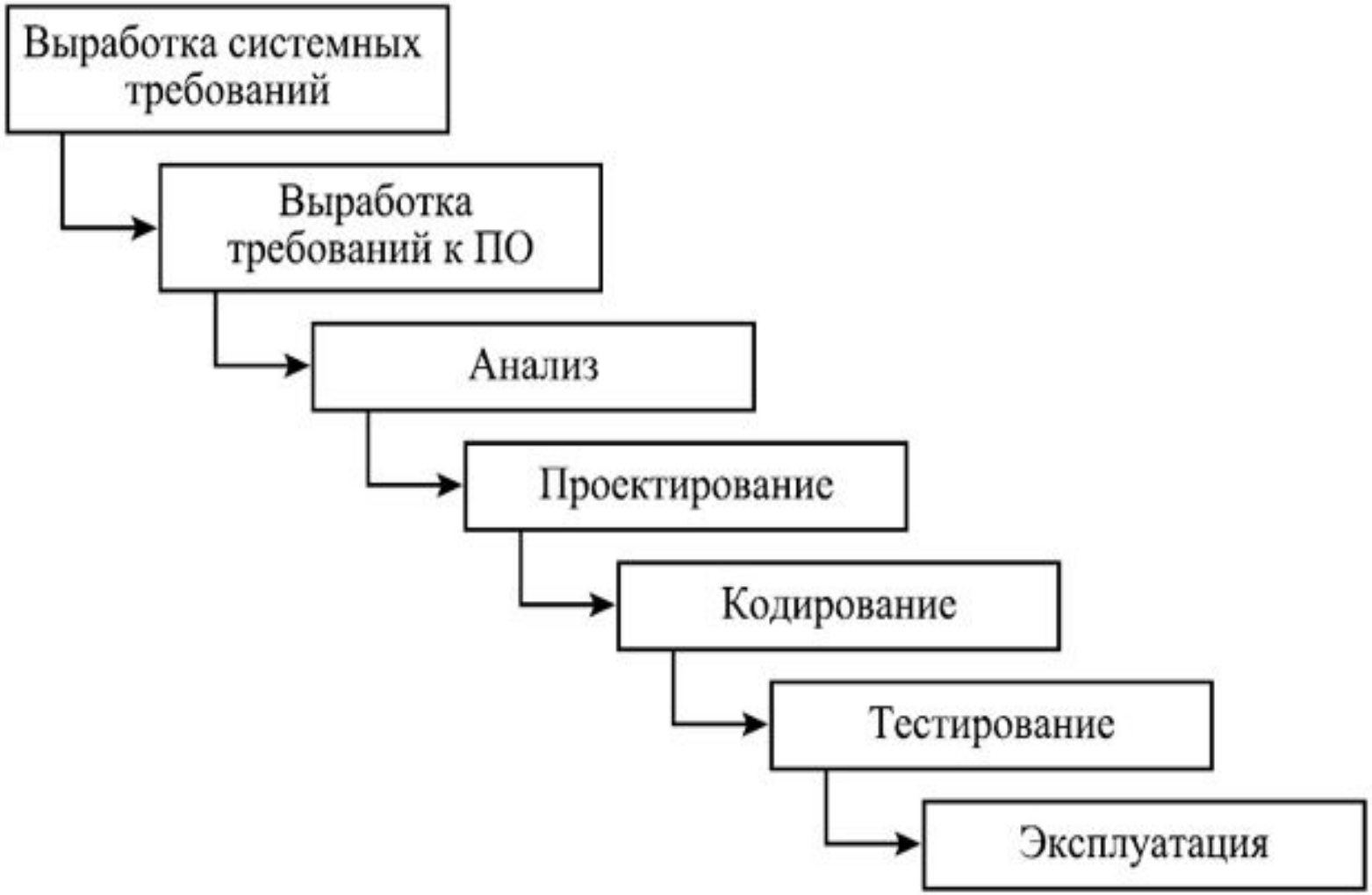
Каскадная схема разработки ПО

- В изначально существовавших однородных ИС каждое приложение представляло собой единое целое. Для разработки такого типа приложений применялся **каскадный** способ.
- Водопадная модель жизненного цикла (англ. *waterfall model*) была предложена в 1970 г. Уинстоном Ройсом.
  Она предусматривает последовательное выполнение всех этапов проекта в строгофиксированном порядке.
- Ее основной характеристикой является разбиение всей разработки на этапы, причем переход с одного этапа на следующий происходит только после того, как будет полностью завершена работа на текущем. Каждый этап завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка могла быть продолжена другой командой разработчиков.

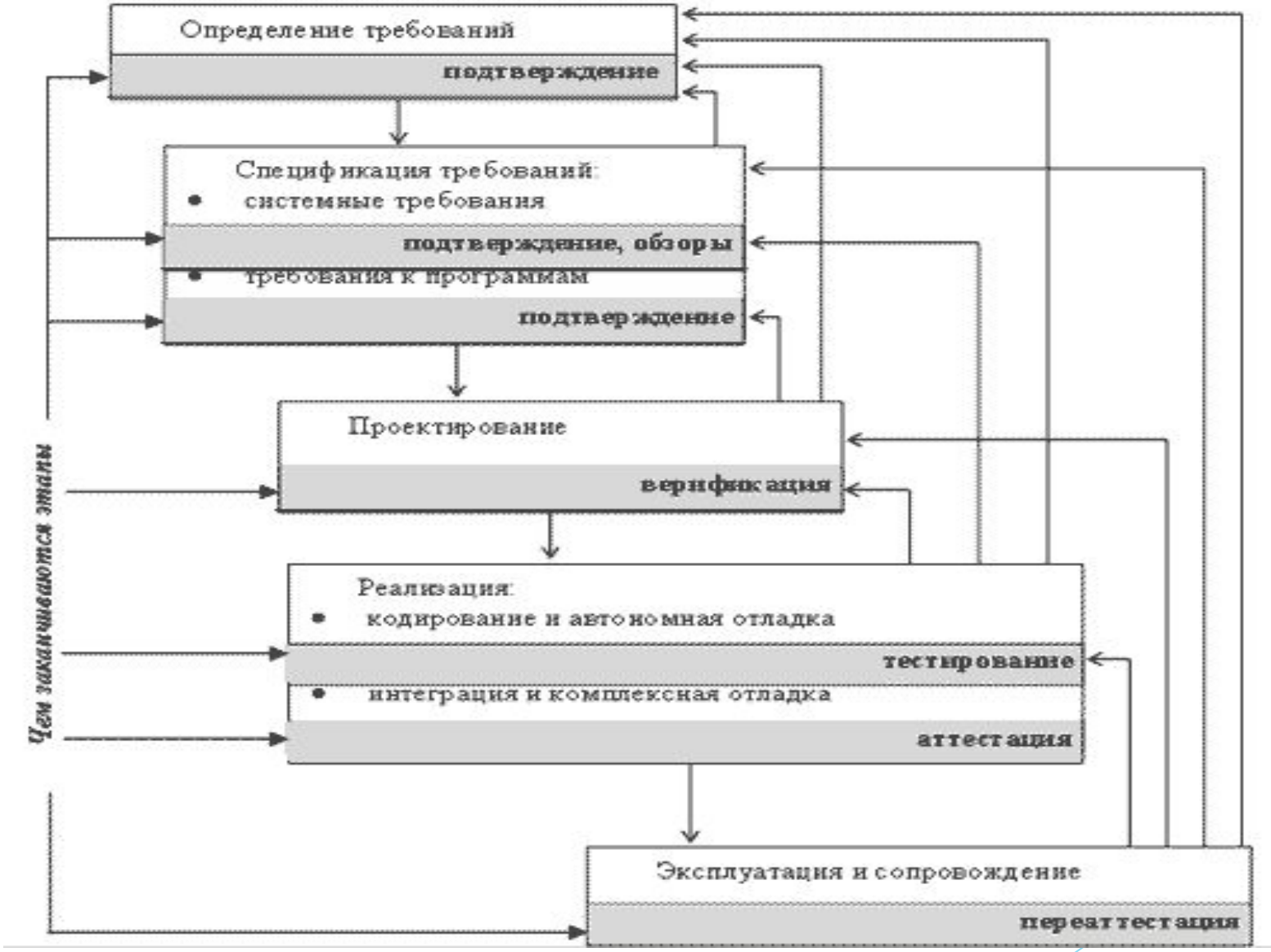
Этапы проекта в соответствии с каскадной моделью:

- Формирование требований;
- Проектирование;
- Реализация;
- Тестирование;
- Внедрение;
- Эксплуатация и сопровождение.

Каскадная схема разработки ПО



Современная каскадная модель



Преимущества современной каскадной модели

- Положительные стороны применения каскадного подхода заключаются в следующем:
  - на каждом этапе формируется законченный набор проектной документации, отвечающий критериям полноты и согласованности;
  - выполняемые в логичной последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.

Недостатки каскадной модели

- Реальный процесс создания ПО никогда полностью не укладывался в жесткую схему. В процессе создания ПО постоянно возникала потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений.
- Основное заблуждение каскадной модели состоит в предположениях, что проект проходит через весь процесс один раз, архитектура хороша и проста в использовании, проект осуществления разумен, а ошибки в реализации устраняются по мере тестирования. Иными словами, каскадная модель исходит из того, что все ошибки будут сосредоточены в реализации, а потому их устранение происходит равномерно во время тестирования компонентов и системы.

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.

Классическая итерационная модель

Преимущества итеративного подхода:

Альтернативой последовательной (водопадной) модели является так называемая модель итеративной и инкрементальной разработки *(IID)*, получившей также от Т. Гилба в 70-е гг. название *эволюционной модели*. Также эту модель называют *итеративной моделью* и *инкрементальной моделью.*

- снижение воздействия серьёзных рисков на ранних стадиях проекта, что ведет к минимизации затрат на их устранение;
- организация эффективной обратной связи проектной команды с потребителем (а также заказчиками) и создание продукта, реально отвечающего его потребностям;
- акцент усилий на наиболее важные и критичные направления проекта;
- непрерывное итеративное тестирование, позволяющее оценить успешность всего проекта в целом;
- раннее обнаружение конфликтов между требованиями, моделями и реализацией проекта;
- более равномерная загрузка участников проекта;
- эффективное использование накопленного опыта;
- реальная оценка текущего состояния проекта и, как следствие, большая уверенность заказчиков и непосредственных участников в его успешном завершении;
- затраты распределяются по всему проекту, а не группируются в его конце.

Различные варианты итерационного подхода реализованы в большинстве современных технологий и методов: Rational Unified Process (RUP), Microsoft Solutions Framework (MSF) и Extreme Programming (XP).

RUP предлагает итеративную модель разработки, вклю чающую четыре фазы: начало, исследование, построение и внедрение. Прохождение через четыре основные фазы называется циклом разработки. Используется объектно-ориентированный анализ, объектно-ориентированное программирование.

MSF сходна с RUP, так же включает четыре фазы: анализ, проектирование, разработка, стабилизация, является итерационной, предполагает использование объектно-ориентированного моделирования.

Экстремальное программирование(XP) является самым новым среди рассматриваемых методологий.

В основе лежит командная работа, эффективная коммуникация между заказчиком и исполнителем в течении всего проекта по разработке ИС, а разработка ведется с использованием последовательно разрабатываемых прототипов.

Спиральная модель

- Спиральная модель (англ. *spiral model*) была разработана в середине 1980-х годов Барри Боэмом.
- При использовании этой модели ПО создается в несколько итераций (витков спирали) методом прототипирования.

На каждой итерации оцениваются:

- риск превышения сроков и стоимости проекта;
- необходимость выполнения ещё одной итерации;
- степень полноты и точности понимания требований к системе;
- целесообразность прекращения проекта.

Основная **проблема** спирального цикла – определение момента перехода на следующий этап.

Отличительной особенностью спиральной модели является специальное внимание, уделяемое рискам, влияющим на организацию жизненного цикла, и контрольным точкам.

Особенности спиральной модели

Боэм формулирует 10 наиболее распространённых рисков:

- Дефицит специалистов.
- Нереалистичные сроки и бюджет.
- Реализация несоответствующей функциональности.
- Разработка неправильного пользовательского интерфейса.
- Перфекционизм, ненужная оптимизация и оттачивание деталей.
- Непрекращающийся поток изменений.
- Нехватка информации о внешних компонентах, определяющих окружение си стемы или вовлеченных в интеграцию.
- Недостатки в работах, выполняемых внешними (по отношению к проекту) ре сурсами.
- Недостаточная производительность получаемой системы.
- Разрыв в квалификации специалистов разных областей.

Преимущества спиральной модели

- Модель уделяет специальное внимание раннему анализу возможностей повторного использования.
- Модель предполагает возможность эволюции жизненного цикла, развитие и изменение программного продукта.
- Модель предоставляет механизмы достижения необходимых параметров качества как составную часть процесса разработки программного продукта.
- Модель уделяет специальное внимание предотвращению ошибок и отбрасыванию ненужных, необоснованных или неудовлетворительных альтернатив на ранних этапах проекта.

Преимущества спиральной модели

- Модель позволяет контролировать источники проектных работ и соответствующих затрат.
- Модель, ориентированная на риски, позволяет в контексте конкретного проекта решить задачу приложения адекватного уровня усилий, определяемого уровнем рисков, связанных с недостаточным выполнением тех или иных работ.
- Модель не проводит различий между разработкой нового продукта и расширением (или сопровождением) существующего.
- Модель позволяет решать интегрированные задачи системной разработки, охватывающие и программную и аппаратную составляющие создаваемого продукта.

Особенности реализации спиральной модели

- Каждый этап реализуется CASE-средствами. Содержание этапов совпадает с аналогичными в каскадной модели, но в отличие от нее, этапы реализуются с помощью CASE-средств (Computer Aided Software System Design) – (автоматизированная технология создания программных информационных систем) – использование этих средств позволяет существенно снизить время реализации витка спирали проектирования подсистем (в этом состоит **основное преимущество спиральной модели**), но это профессиональные средства, непредназначенные для конечных пользователей.
- С помощью CASE-средств можно быстро сгенерировать проект хотя бы на уровне экранных форм и показать его конечному пользователю, который высказывает свои предложения и замечания, и на следующем витке реализуются они. Когда спецификация на уровне экранных форм будет согласована, то проектировщик может начинать детальную реализацию.
- Описанная схема разработки называют визуальным проектированием. Витков может быть много, разделение на этапы условно. Работы могут выполняться параллельно или в комплексной итерации на любом витке спирали.

Модель фазы—функции

Важным мотивом развития моделей жизненного цикла программного обеспечения является потребность в подходящем средстве для комплексного управления проектом. По существу, это утверждение указывает на то, что модель должна служить основой организации взаимоотношений между разработчиками, и, таким образом, одной из ее целей является поддержка функций менеджера.

Это приводит к необходимости наложения на модель контрольных точек и функций, задающих организационно-временные рамки проекта.

Наиболее последовательно такое дополнение классической схемы реализовано в модели Гантера в виде матрицы «фазы—функции». Уже из упоминания о матрице следует, что модель Гантера имеет два измерения:

- *фазовое* , отражающее этапы выполнения проекта и сопутствующие им события;
- *функциональное* , показывающее, какие организационные функции выполняются в ходе развития проекта и какова их интенсивность на каждом из этапов.

В модели Гантера отражено то, что выполнение функции на одном этапе может продолжаться и на следующем.

Фазовое измерение модели фазы–функции

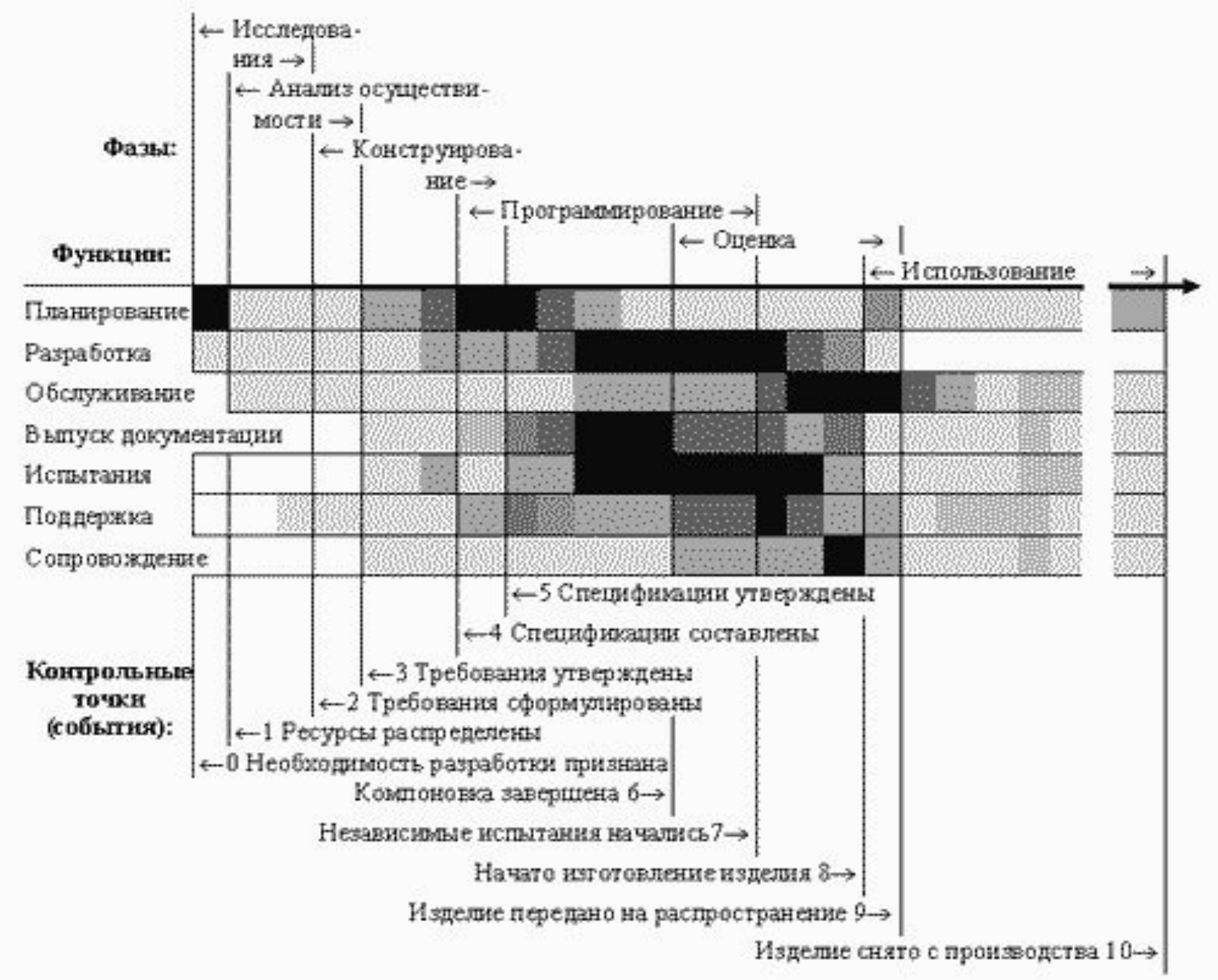
Этапы жизненного цикла

- ***Этап исследования*** — начинается, когда необходимость *разработки* признана руководством проекта (контрольная точка 0), и заключается в том, что для проекта обосновываются необходимые ресурсы (контрольная точка 1) и формулируются требования к разрабатываемому изделию (контрольная точка 2).
- ***Анализ осуществимости*** — начинается на *этапе исследования*, когда определены исполнители проекта (контрольная точка 1), и завершается утверждением требований (контрольная точка 3). Цель *этапа* — определить возможность *конструирования* изделия с технической точки зрения (достаточно ли ресурсов, квалификации и т.п.), будет ли изделие удобно для практического *использования* ; решение вопросов экономической и коммерческой эффективности.
- ***Конструирование*** — начинается обычно на *этапе анализа осуществимости*, как только документально зафиксированы предварительные цели проекта (контрольная точка 2), и заканчивается утверждением проектных решений в виде официальной спецификации на *разработку* (контрольная точка 5).

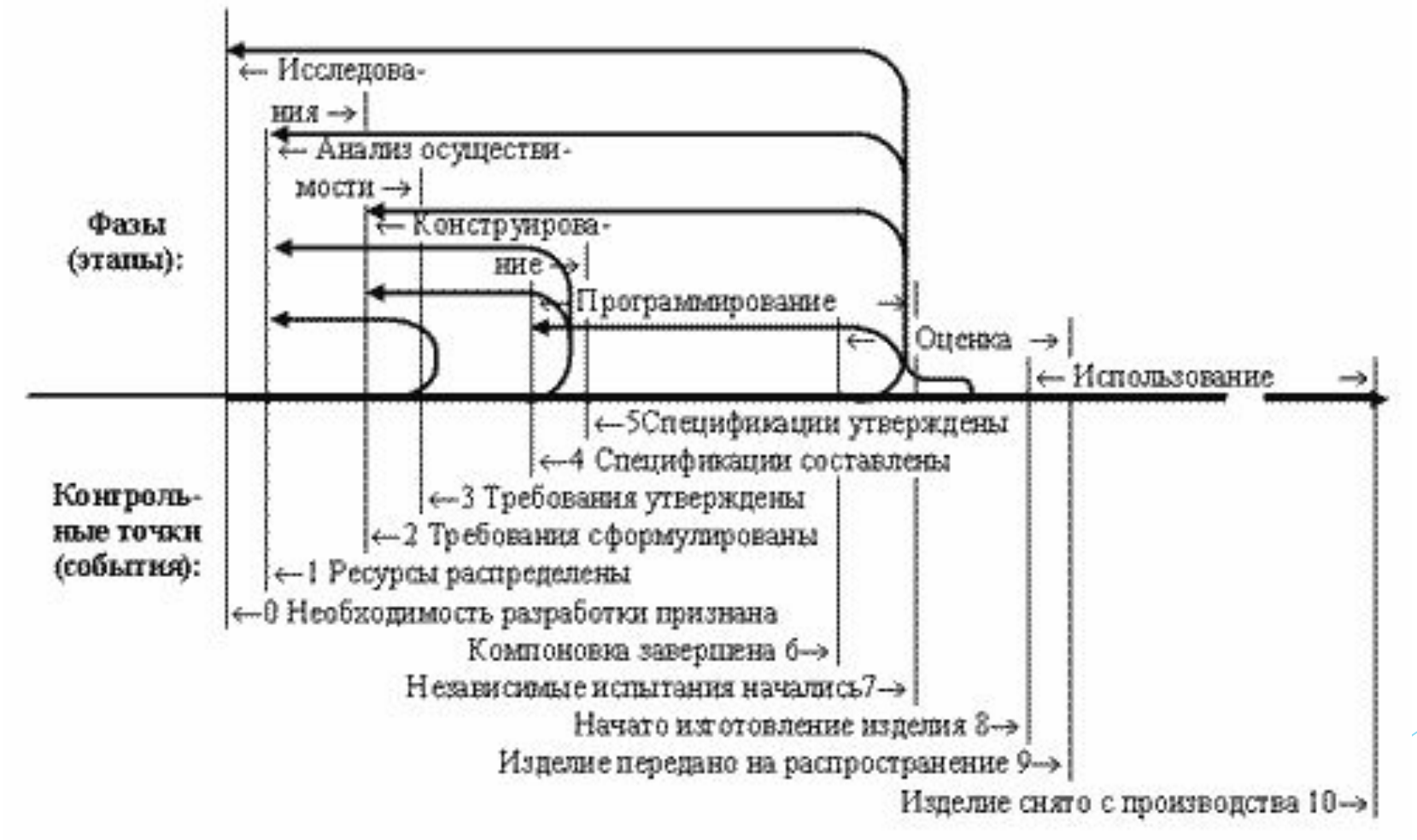
Этапы жизненного цикла

- ***Программирование*** — начинается на *этапе конструирования*, когда становятся доступными основные спецификации на отдельные компоненты изделия (контрольная точка 4), но не ранее утверждения соглашения о требованиях (контрольная точка 3). Совмещение данной *фазы* с заключительным *этапом конструирования* обеспечивает оперативную проверку проектных решений и некоторых ключевых вопросов *разработки*. Цель *этапа* — реализация программ компонентов с последующей сборкой изделия. Он завершается, когда разработчики заканчивают документирование, отладку и компоновку и передают изделие службе, выполняющей независимую *оценку* результатов работы (независимые *испытания* начались — контрольная точка 7).
- ***Оценка*** — является буферной зоной между началом *испытаний* и практическим *использованием* изделия. *Этап* начинается, как только проведены внутренние (силами разработчиков) *испытания* изделия (контрольная точка 6) и заканчивается, когда подтверждается готовность изделия к эксплуатации (контрольная точка 9).
- ***Использование*** — начинается ближе к концу *этапа оценки*, когда готовность изделия к эксплуатации проверена и может организовываться передача изделия на распространение (контрольная точка 8). *Этап* продолжается, пока изделие находится в действии и интенсивно эксплуатируется. Он связан с внедрением, обучением, настройкой и *сопровождением*, возможно, с модернизацией изделия. *Этап* заканчивается, когда разработчики прекращают систематическую деятельность по *сопровождению* и *поддержке* данного программного изделия (контрольная точка 10).

Матрица фазы—функции модели Гантера



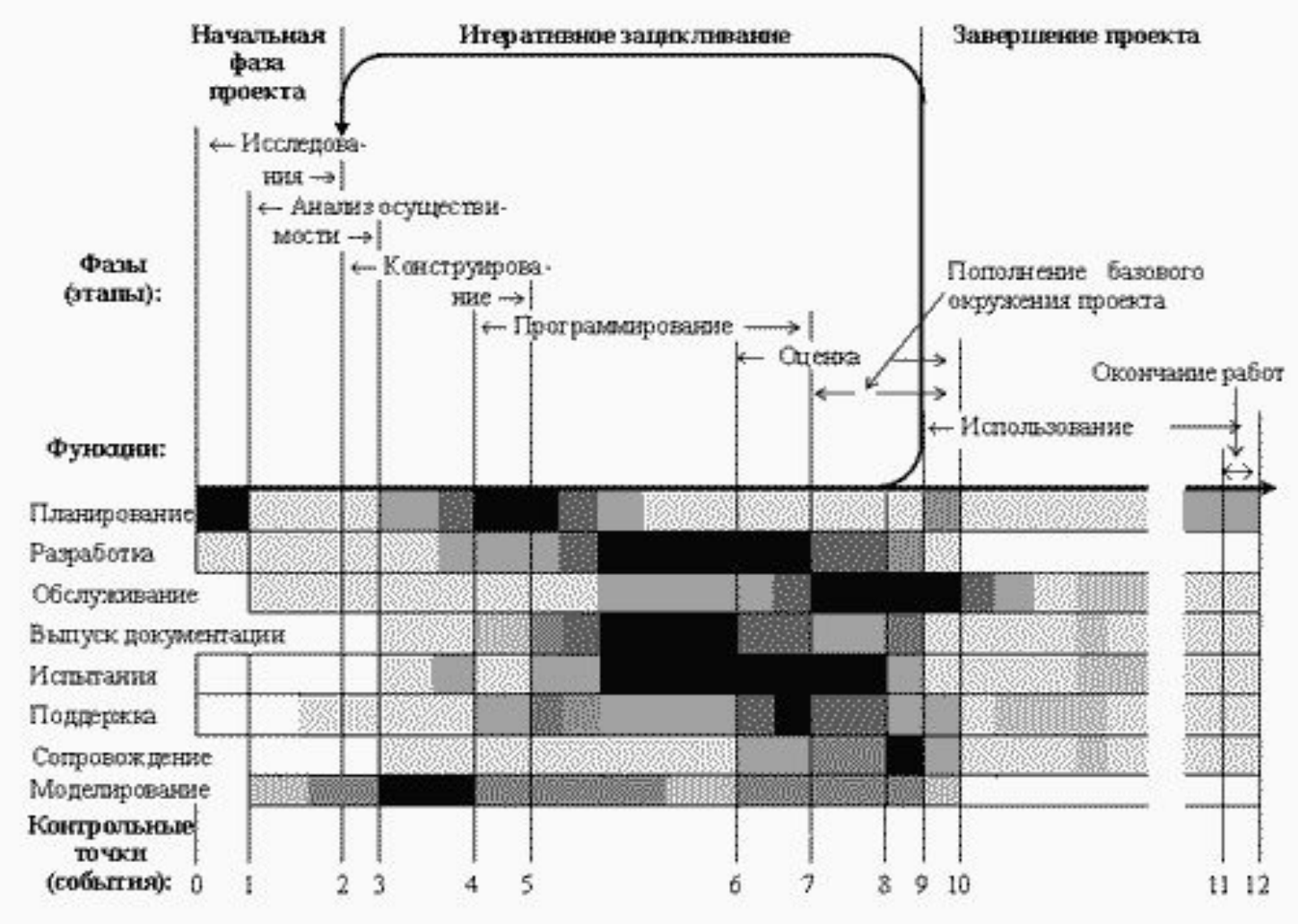
Учет итеративности в модели фазы—функции (фазовое измерение, показаны лишь некоторые возвраты)



Принципы ООП в развитии проектов

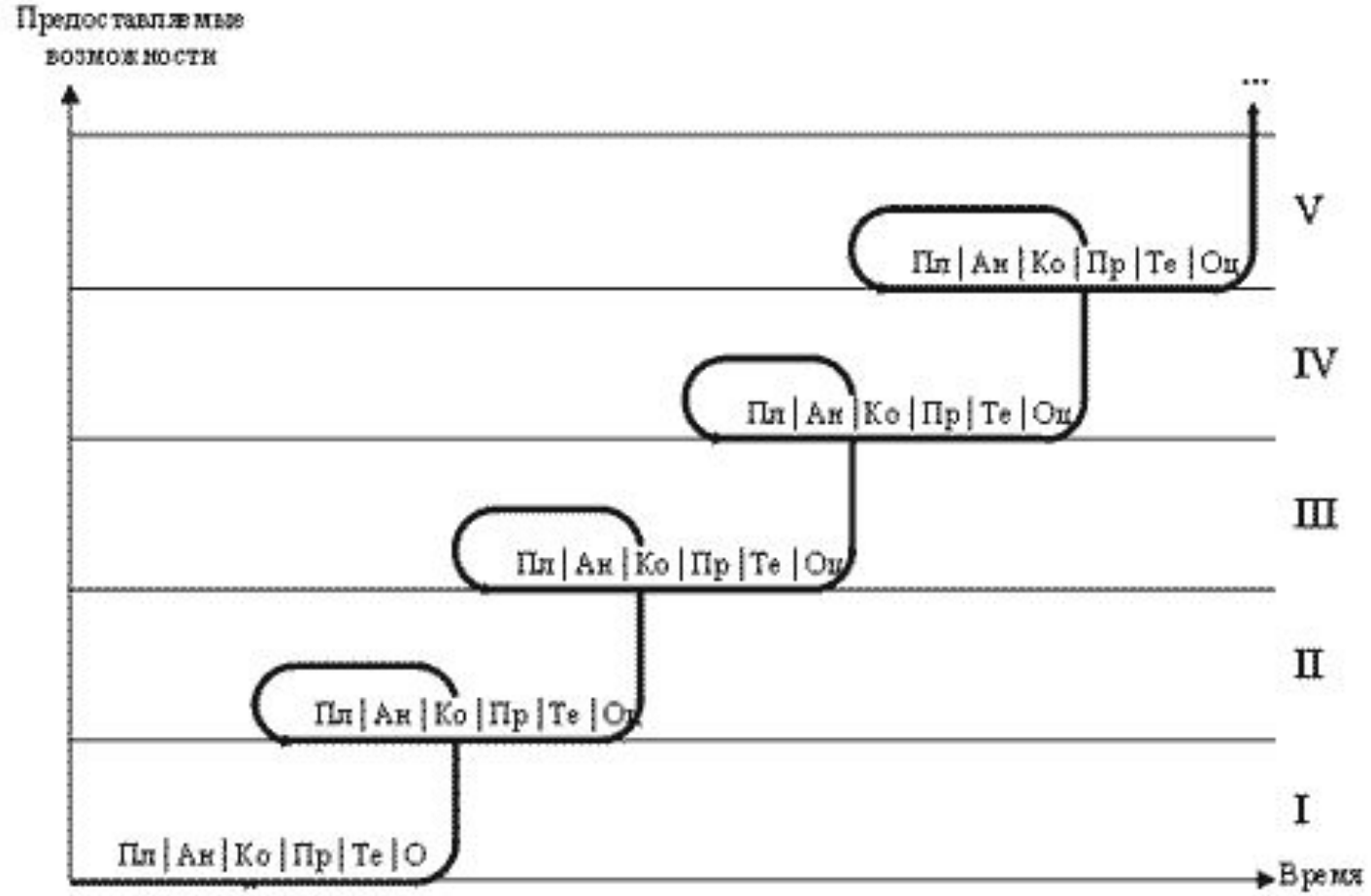
- **Итеративность развития.** Начиная с фазы анализа и до завершения реализации, процесс объектно-ориентированного проектирования в противоположность последовательному развитию строится как серия итераций, которой возможно предшествует определенный период последовательного изучения предметной области и задач проекта в целом (этапы *определения требований* и *начального планирования*).
- **Наращивание функциональности в соответствии со сценариями.** Наращивание функциональности проектируемого изделия представляется как развитие сценариев, которые соответствуют описаниям (диаграммам) взаимодействия объектов и отражают отдельные стороны функционирования.
- **Ничто не делается однократно.** Последовательный подход предполагает, что анализ завершен перед конструированием, завершение которого предшествует программированию.
- **Оперирование на размножающихся фазах подобно.** Как в начале проектирования, на последующих итерациях анализ предшествует конструированию, за которым следует программирование, тестирование и другие виды работ.

Модель фазы—функции, модифицированной для объектно-ориентированного развития проекта



Моделирование итеративного наращивания возможностей системы

В предыдущих моделях жизненного цикла объектно-ориентированного программного обеспечения не был наглядно выделен важный аспект подхода: постепенное наращивание возможностей системы по мере развития проекта. Для его отражения можно предложить представление жизненного цикла в виде *спирали развития*



Модель расширения охвата прикладной области объектно-ориентированной

Постепенное наращивание возможностей системы по мере развития проекта часто изображают в виде спирали, раскручивающейся на плоскости от центра. В соответствии с этой простой (грубой) моделью развитие проекта описывается как постепенный охват все более расширяющейся области плоскости по мере перехода проекта от этапа к этапу и от итерации к итерации.

В отличие от предыдущих моделей, обе спиралевидные модели никак не отражают тот факт, что у проекта есть фаза завершения. Как следствие, они предполагают, что все модификации какой-либо версии программной системы, которые требуются после ее выпуска, будут относиться к одной из следующих версий.

Спиральная модель ЖЦ

2.3. Содержание и организация проектирования

- Под проектированием автоматизированных ИС понимается процесс разработки технической документации, связанный с организацией системы получения и преобразования исходной информации в результатную, Т.е. с организацией автоматизированной информационной технологии.
- Документ, полученный в результате проектирования, носит название *проект*.
- Целью проектирования является подбор технического и формирование информационного, математического, про граммного и организационно-правового обеспечения.

Основными задачами проектирования являются:

- Оказание влияния на улучшение организации учетной, плановой и аналитической работы;
- Выбор оборудования и разработка рациональной технологии решения задач и получения результатной информации;
- Составление графиков прохождения информации как внутри производственных и функциональных подразделений, так и между ними;
- Создание БД, обеспечивающей оптимальное использование информации, касающейся планирования, учета и анализа хозяйственной деятельности;
- Создание нормативно-справочной информации.

2.3.1. Каноническое проектирование ИС

- Организация *канонического проектирования* ИС ориентирована на использование главным образом каскадной модели жизненного цикла ИС.
- В зависимости от сложности объекта, стадии и этапы работ могут иметь различную трудоемкость. Допускается объединять последовательные этапы и даже исключать некоторые из них на любой стадии проекта, а также начинать выполнение работ следующей стадии до окончания предыдущей.

Стадии и этапы создания ИС:

- Стадия 1. Формирование требований к ИС:
  - обследование объекта и обоснование необходимости создания ИС.
  - Формирование требования пользователей к ИС.
  - Оформление отчета о выполненной работе и тактико-технического задания на разработку
- Стадия 2. Разработка концепции ИС:
  - Изучение объекта автоматизации.
  - Проведение научно-исследовательских работ.
  - Разработка вариантов концепции ИС, удовлетворяющих требованиям пользователей
  - Оформление отчета и утверждения концепции

- Стадия 3. Техническое задание:
  - Разработка и утверждение тех. Задания на создание ИС
- Стадия 4. Эскизный проект:
  - Разработка предварительных проектных решений по системе и её частям
  - Разработка эскизной документации на ИС и её части
- Стадия 5. Технический проект:
  - Разработка проектных решений по системе и её частям
  - Разработка документации на ИС и её части
  - Разработка и оформление документации на постановку комплектующих изделий
  - Разработка заданий на проектирование в смежных частях проекта

- Стадия 6. Рабочая документация:
  - Разработка рабочей документации на ИС и её части
  - Разработка и адаптация программ
- Стадия 7. Ввод в действие:
  - Подготовка объекта автоматизации
  - Подготовка персонала
  - Комплектация ИС поставляемыми изделиями(Программно техническими средствами, комплексами, информационными изделиями)
  - Строительно-монтажные и пусконаладочные работы
  - Предварительные испытания
  - Опытная эксплуатация
  - Приемочные испытания

- Стадия 8. Сопровождение ИС:
  - Выполнение работ в соответствии с гарантийными обязательствами
  - Послегарантийное обслуживание