Основные понятия программной инженерии

Международный терминологический стандарт ISO/IEC 2382/1-93

- Программная инженерия – систематическое применение научных и технологических знаний, методов и практического опыта к проектированию, реализации, тестированию и документированию программного обеспечения в целях оптимизации его производства, поддержки и качества.

- Различают методы, средства и процессы программной инженерии.

Методы

Обеспечивают решение широкого спектра технических задач.

Например:

- Планирование и оценка программного проекта;
- Тестирование и т.д.

Средства

- Утилиты программной инженерии обеспечивают автоматизированную или автоматическую поддержку методов.
- Утилиты могут объединяться всистемы автоматизированной разработки ПО. Такие системы принято называть CASE-системами.

Процессы

- Это набор ваимосвязанных работ, которые преобразуют исходные данные в выходные результаты.
- Соединяют методы и утилиты для обеспечения непрерывной разработки.

- Классификацию процессов программной инженерии задают международный стандарт ISO/IEC 12207-95 «Information Technology – Software Life Cycle Processes» и его российский аналог ГОСТ Р ИСО/МЭК 12207-99.

- Жизненный цикл программного обеспечения определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

Работы, выполняемые в ЖЦ ПО

- 5 основных
- 8 вспомогательных
- 4 организационных процесса

Основные процессы ЖЦ

- 1. Процесс заказа (acquisition process).
- 2. Процесс постановки (supply process).
- 3. Процесс разработки(development process).
- 4. Процесс эксплуатации (operation process).
- 5. Процесс сопровождения (maintenance process).

Вспомогательные процессы ЖЦ

- 1. Процесс документирования (documentation process).
- 2. Процесс управления конфигурацией (configuration management process).
- 3. Процесс обеспечения качества (quality assurance process).
- 4. Процесс верификации (verification process).
- 5. Процесс аттестации (validation process).
- 6. Процесс совместной проверки (joint review process).
- 7. Процесс аудита (audit process).
- 8. Процесс решения проблемы (problem resolution process).

Организационные процессы ЖЦ

- 1. Процесс управления (management process).
- 2. Процесс создания инфраструктуры (infrastructure process).
- 3. Процесс усовершенствования (improvement process).
- 4. Процесс обучения (training process).

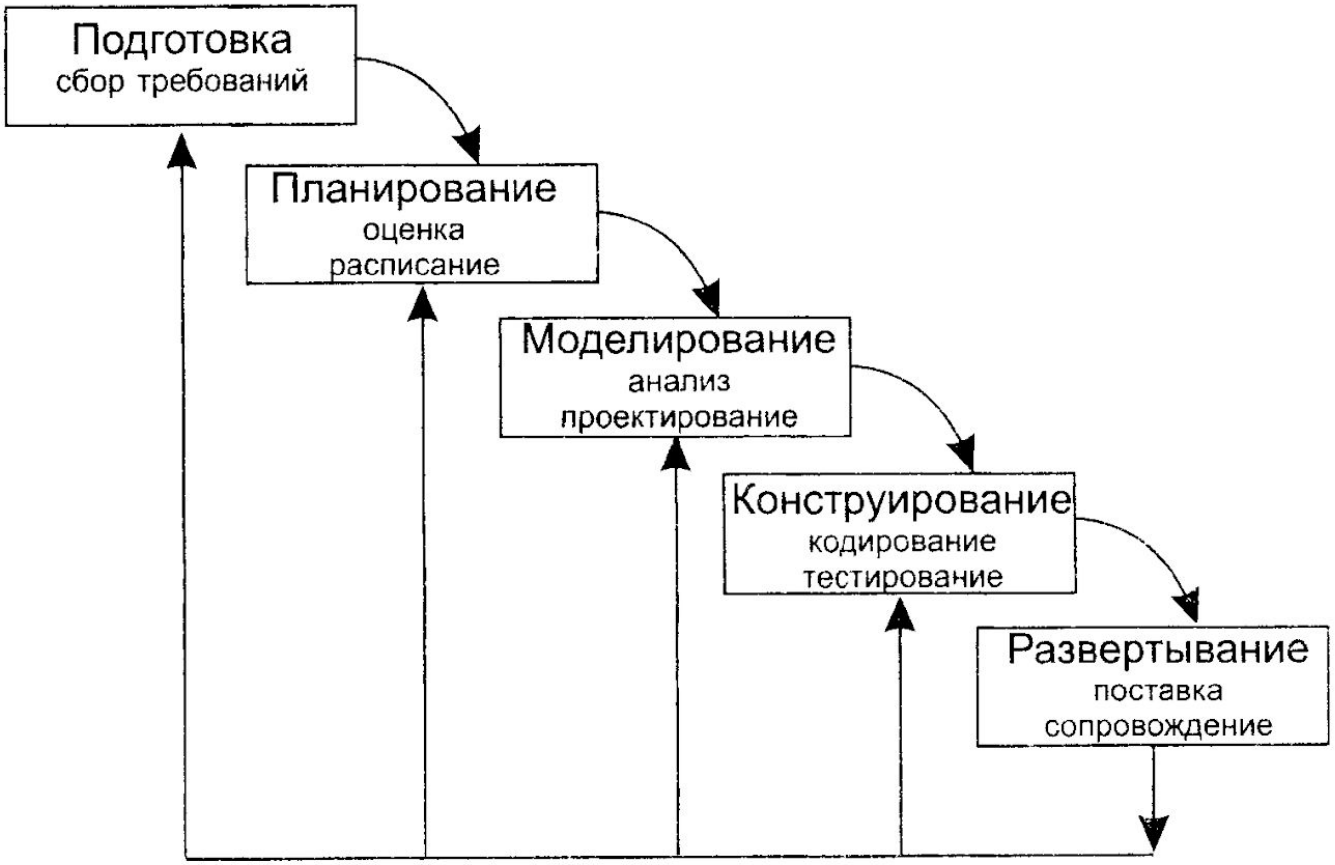
Стратегии конструирования ПО

- Существуют 3 стратегии конструирования ПО:
- *однократный проход* (водопадная стратегия) — линейная последовательность этапов конструирования;

- *инкрементная стратегия.* В начале процесса определяются все пользовательские и системные требования, оставшаяся часть конструирования выполняется в виде последовательности версий.
- Первая версия реализует часть запланированных возможностей, следующая версия реализует дополнительные возможности и т. д., пока не будет получена полная система;

- *эволюционная стратегия.* Система также строится в виде последовательности версий, но в начале процесса определены не все требования. Требования уточняются в результате разработки версий.

Классический жизненный цикл

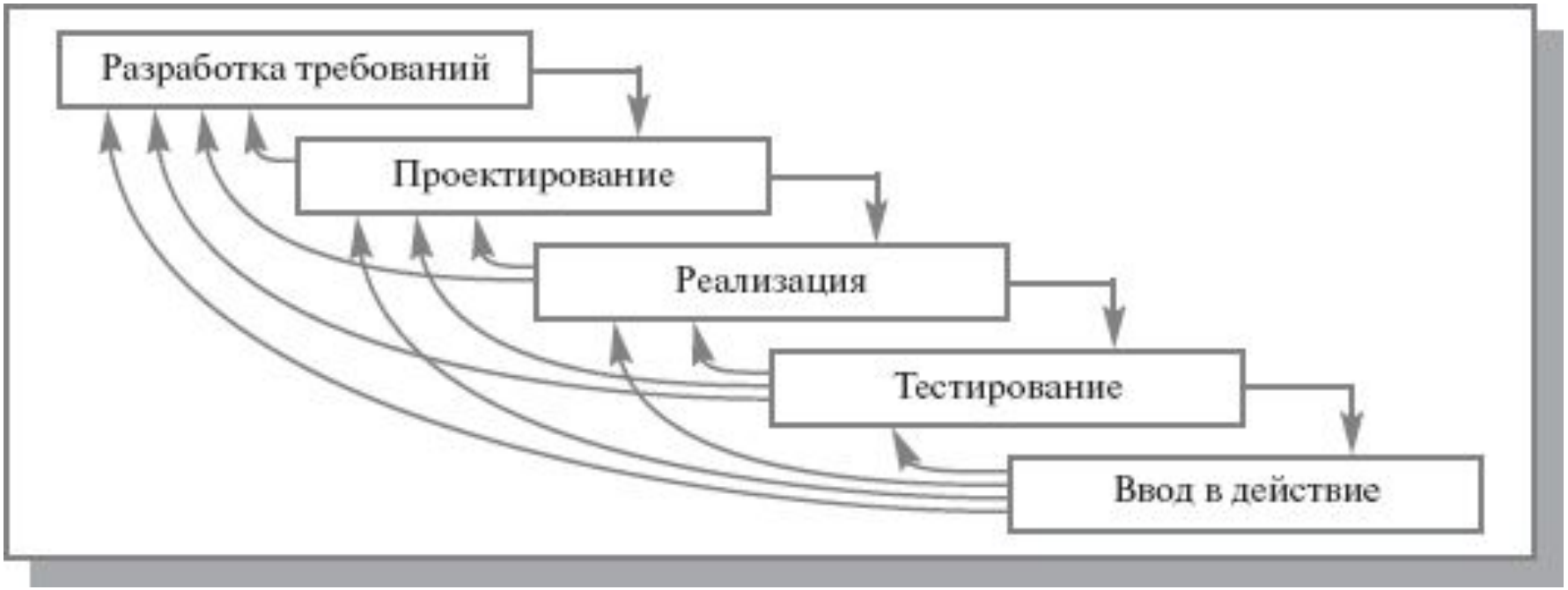


- предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе

- Как и любая инженерная схема, модель классического жизненного цикла имеет достоинства и недостатки.
- Достоинства: дает план и временной график по всем этапам проекта, упорядочивает ход разработки.

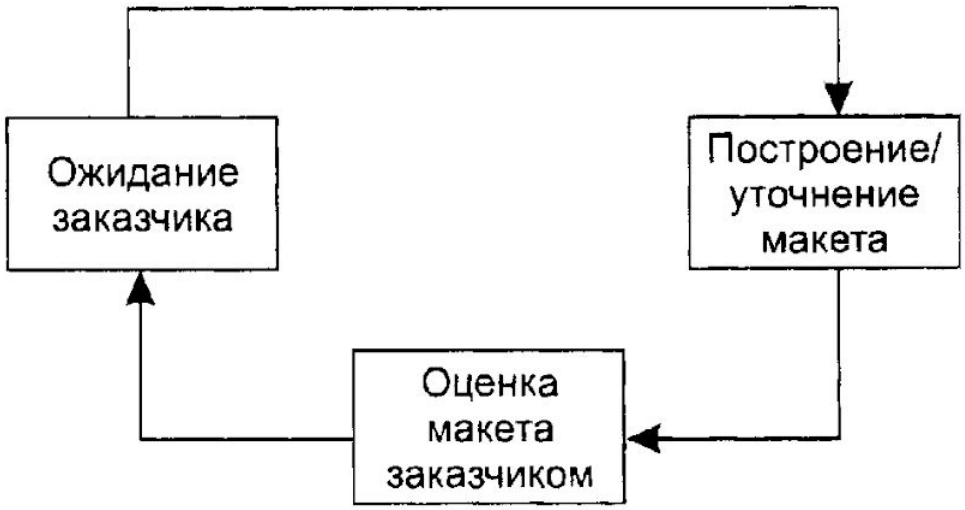
- Недостатки:
- 1) реальные проекты часто требуют отклонения от стандартной последовательности шагов;
- 2) ЦИКЛ основан на точной формулировке исходных требований к ПО (реально в начале проекта требования заказчика определены лишь частично);
- 3) результаты проекта доступны заказчику только в конце работы.

Поэтапная модель с промежуточным контролем



- ***Поэтапная модель с промежуточным контролем***. Разработка ИС ведется итерациями с циклами обратной связи между этапами. Межэтапные корректировки позволяют учитывать реально существующее взаимовлияние результатов разработки на различных этапах; время жизни каждого из этапов растягивается на весь период разработки.

Макетирование



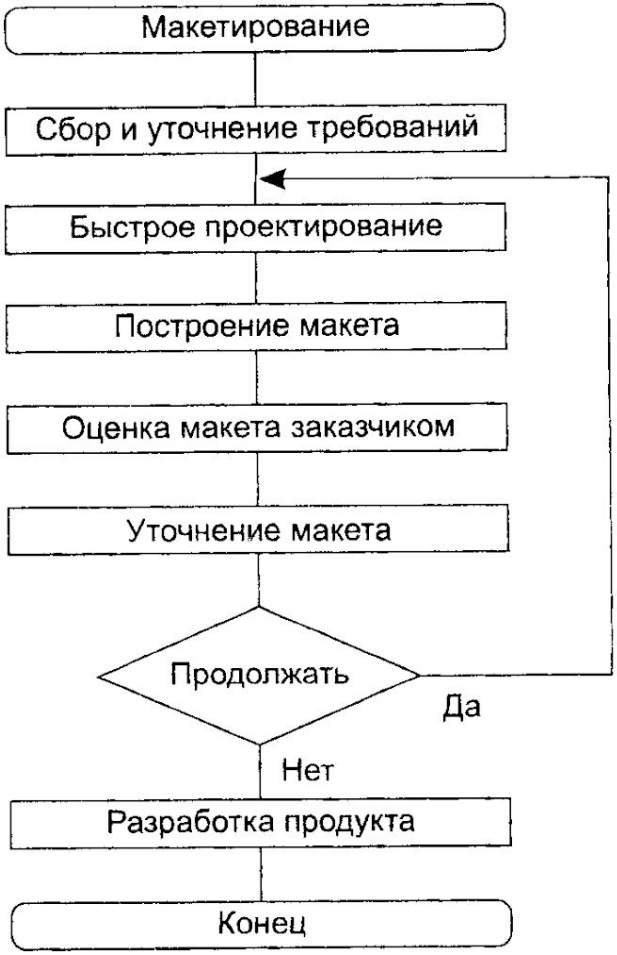
- Основная цель макетирования: снять неопределенности в требованиях заказчика.
- Макетирование (прототипирование) это процесс создания модели требуемого программного продукта.
- Модель может принимать одну из трех форм:

1) бумажный макет или макет на основе ПК (изображает или рисует человекомашинный диалог);
2) работающий макет (выполняет некоторую часть требуемых функций);
3) существующая программа (характеристики которой затем должны быть улучшены).

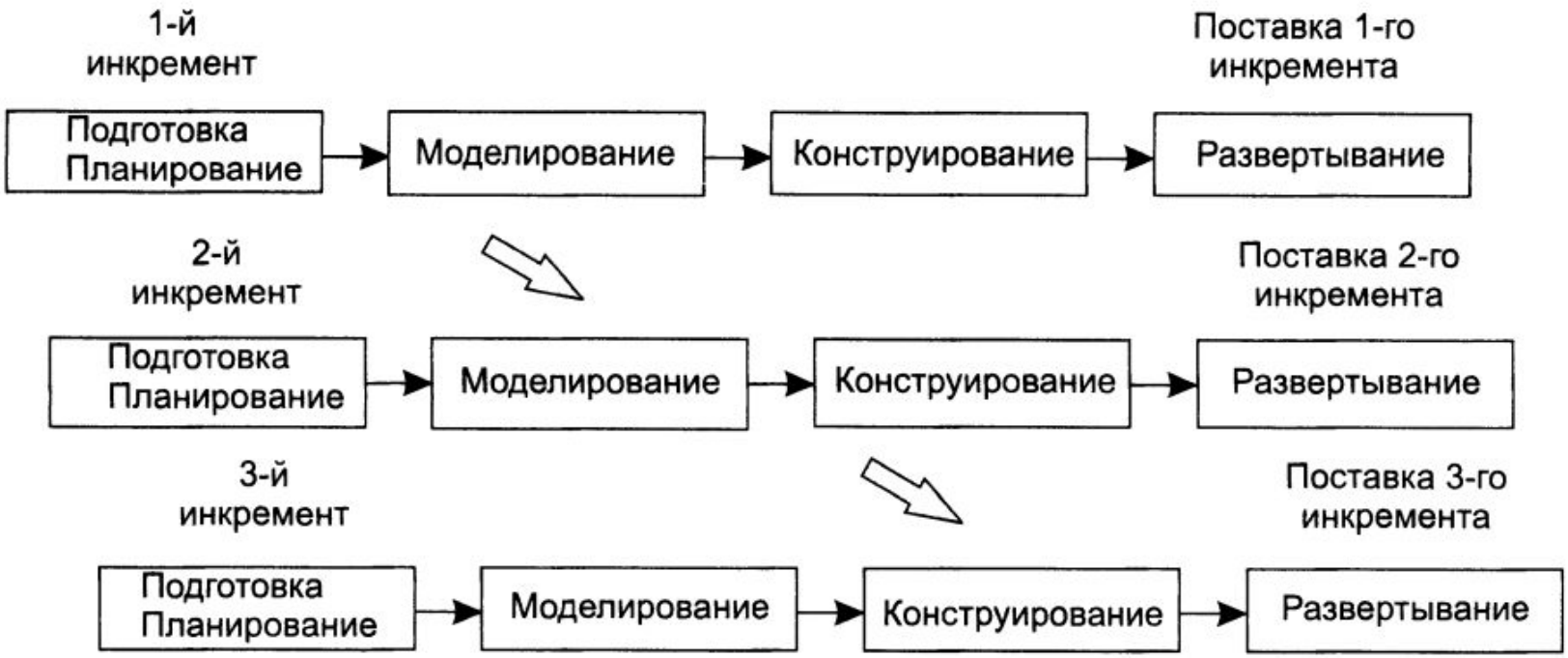
- Макетирование начинается со сбора и уточнения требований к создаваемому ПО. Разработчик и заказчик встречаются и определяют все цели ПО, устанавливают, какие требования известны, а какие предстоит доопределить.
- Затем выполняется быстрое проектирование. В нем внимание сосредотачивается на тех характеристиках ПО, которые должны быть видимы пользователю.

- Достоинство макетирования: обеспечивает определение полных требований к ПО.
- Недостатки макетирования:
- заказчик может принять макет за продукт;
- разработчик может принять макет за продукт.

Последовательность действий при макетировании

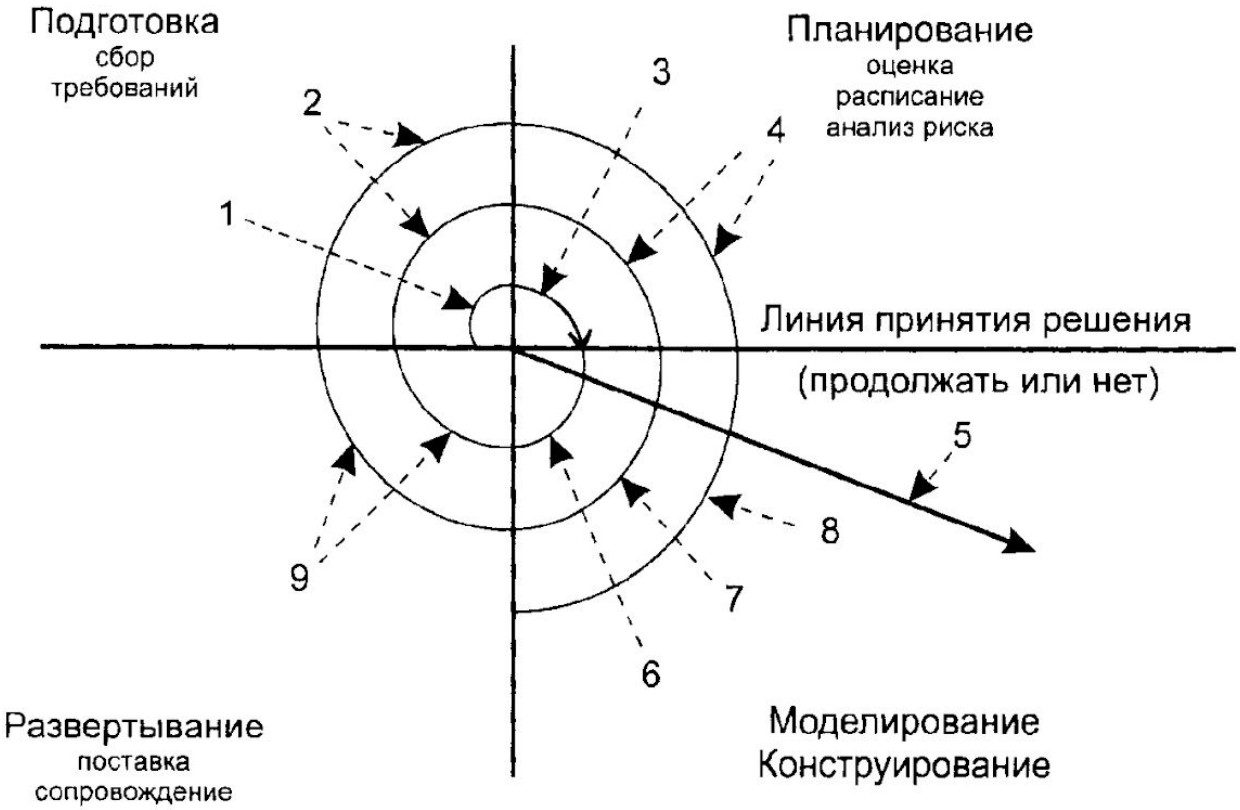


Инкрементная модель



- в отличие от макетирования инкрементная модель обеспечивает на каждом инкременте работающий продукт.

Спиральная модель



Достоинства спиральной модели:

- наиболее реально (в виде эволюции) отображает разработку программного обеспечения;
- позволяет явно учитывать риск на каждом витке эволюции разработки;
- включает возможность оценки системы в итерационную структуру разработки;
- использует моделирование для уменьшения риска и ПП.

Недостатки спиральной модели:

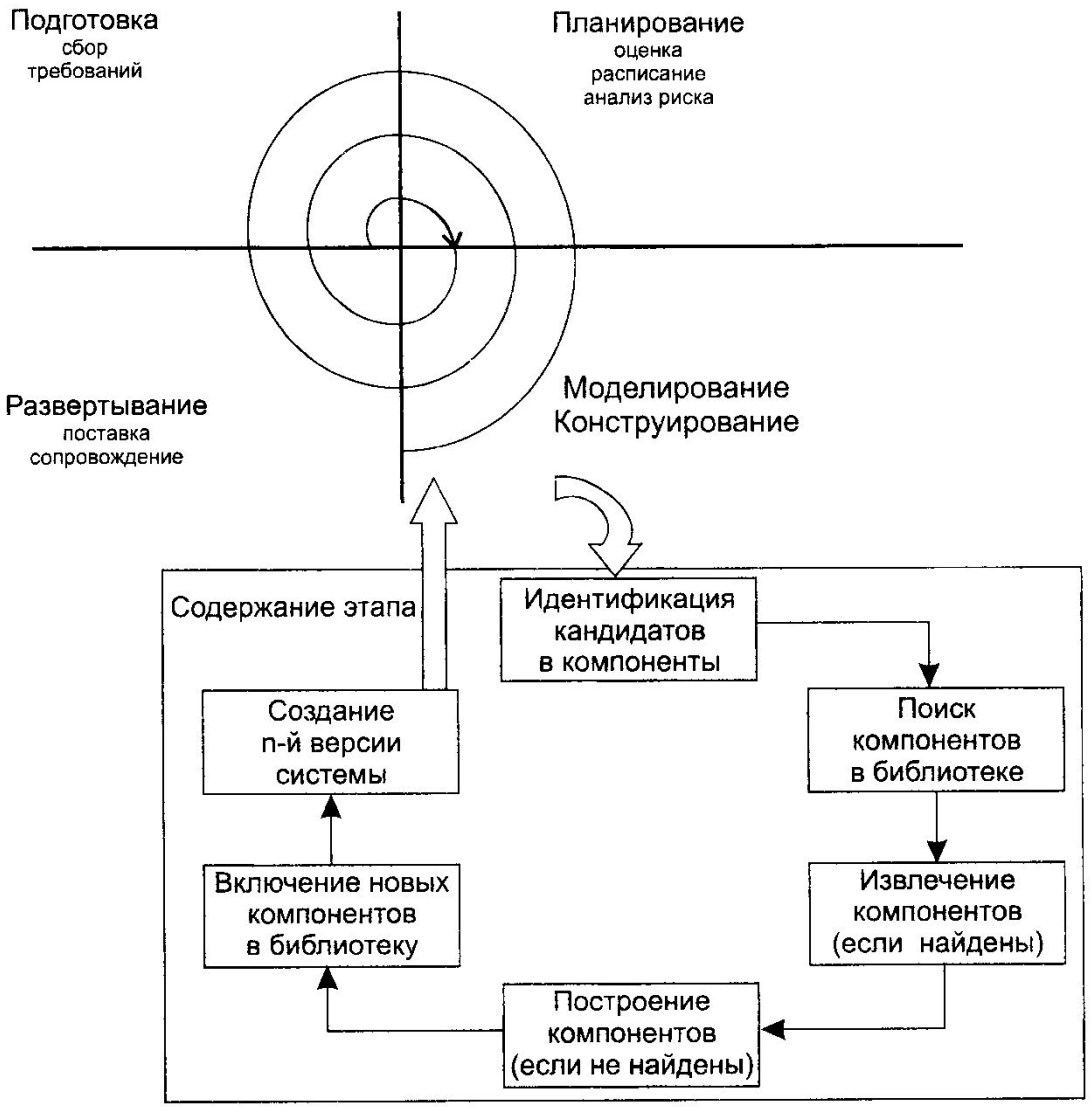
- повышенные требования к заказчику;
- трудности контроля и управления временем разработки.

Компонентно-ориентированная модель

- В этой модели модифицируется содержание квадранта моделирования-конструирования.

Достоинства компонентно-ориентированной модели :

- 1) уменьшает на 3О% время разработки программного продукта;
- 2) уменьшает стоимость программной разработки до 70%;
- 3) увеличивает в полтора раза производительность разработки.


Подготовка
сбор
требований
Планирование
оценка
расписание
анализ риска
Развертывание
поставка
сопровождение
Моделирование
Конструирование
Содержание этапа
Идентификация
кандидатов
в компоненты
Поиск
компонентов
в библиотеке
Извлечение
компонентов
(если найдены)
Построение
компонентов
(если не найдены)
Включение новых
компонентов
в библиотеку
Создание
n-й версии
системы

- ГОСТ 34.601-90 - распространяется на автоматизированные системы и устанавливает стадии и этапы их создания. Кроме того, в стандарте содержится описание содержания работ на каждом этапе. Стадии и этапы работы, закрепленные в стандарте, в большей степени соответствуют *каскадной модели* жизненного цикла.
- ISO/IEC 12207:1995 - стандарт на процессы и организацию *жизненного цикла*. Распространяется на все виды заказного ПО. Стандарт не содержит описания фаз, стадий и этапов .

- Custom Development Method (методика Oracle) по разработке прикладных информационных систем - технологический материал, детализированный до уровня заготовок проектных документов, рассчитанных на использование в проектах с применением Oracle. Применяется CDM для классической модели ЖЦ (предусмотрены все работы/задачи и этапы), а также для технологий "быстрой разработки" (Fast Track) или "облегченного подхода", рекомендуемых в случае малых проектов.

- Rational Unified Process (RUP) предлагает итеративную модель разработки, включающую четыре фазы: начало, исследование, построение и внедрение. Каждая фаза может быть разбита на этапы (итерации), в результате которых выпускается версия для внутреннего или внешнего использования.

- Прохождение через четыре основные фазы называется циклом разработки, каждый цикл завершается генерацией версии системы.

- Если после этого работа над проектом не прекращается, то полученный продукт продолжает развиваться и снова минует те же фазы. Суть работы в рамках RUP - это создание и сопровождение моделей на базе UML.

- Microsoft Solution Framework (MSF) сходна с RUP, так же включает четыре фазы: анализ, проектирование, разработка, стабилизация, является итерационной, предполагает использование объектно-ориентированного моделирования.
- MSF в сравнении с RUP в большей степени ориентирована на разработку бизнес-приложений.

- Extreme Programming (XP). Экстремальное программирование (самая новая среди рассматриваемых методологий) сформировалось в 1996 году.
- В основе методологии командная работа, эффективная коммуникация между заказчиком и исполнителем в течение всего проекта по разработке ИС, а разработка ведется с использованием последовательно дорабатываемых прототипов.

- В соответствии с базовым международным стандартом ISO/IEC 12207 все *процессы ЖЦ ПО* делятся на три группы:
- **Основные процессы:**
  - приобретение;
  - поставка;
  - разработка;
  - эксплуатация;
  - сопровождение.

- **Вспомогательные процессы:**
  - документирование;
  - управление конфигурацией;
  - обеспечение качества;
  - разрешение проблем;
  - аудит;
  - аттестация;
  - совместная оценка;
  - верификация.

- **Организационные процессы:**
  - создание инфраструктуры;
  - управление;
  - обучение;
  - усовершенствование.