

Модели жизненного цикла программного продукта

Тема 1.2

Введение

Жизненный цикл программного обеспечения охватывает промежуток времени с момента возникновения потребности в ПО до вывода его из эксплуатации

В зависимости от используемой модели жизненный цикл протекать может по-разному

Модели отличаются между собой по таким параметрам как:

- этапность (фазы, стадии, этапы)
- последовательность прохождения этапов (линейная или циклическая)
- гибкость (возможность подстраивать процесс под конкретные условия)
- связь с определёнными методологиями разработки ПО
- использование специализированных инструментальных средств
- другие

Основные стандарты на ЖЦ ПО

- 1985 (уточнен в 1988 г.) DOD-STD-2167 A – Разработка программных средств для систем военного назначения.
- 1994г. MIL-STD-498. Разработка и документирование программного обеспечения.
- 1995г. IEEE 1074. Процессы жизненного цикла для развития программного обеспечения.
- Стандарт ISO/IEC 12207. Процессы жизненного цикла ПП.

Проблемы разработки и внедрения стандартов

- Внедрение стандартов требовало вложения значительных средств, что не всегда окупалось.
- Было неясно, все ли требуемые процессы надо выполнять и в какой мере.
- Различные типы ПО (ИС, реального времени, бизнес системы), различные требования.
- Высокая динамика отрасли и устаревание стандартов.
- Терминологическая неоднозначность различных государственных и корпоративных стандартов.
- Во многих случаях применение стандартов было вызвано только требованиями заказчиков, хотя на практике часто тормозило выполнение проектов.

Стандарт ISO/IEC 12207 –

процессы жизненного цикла

ПП

- Был принят в 1995 году ISO совместно с IEC (International Electrotechnical Commission - Международная электротехническая комиссия)
- В 2000 г. он был принят как ГОСТ (ИСО/МЭК) 12207 - Процессы жизненного цикла программных средств
- Определяет организацию ЖЦ программного продукта как совокупность процессов, каждый из которых разбит на действия, состоящие из отдельных задач; устанавливает структуру (архитектуру) ЖЦ программного продукта в виде перечня процессов, действий и задач

Процессы ЖЦ в соответствии со стандартом ISO/IEC 12207

● Основные

- Заказа. Определяет работы заказчика, то есть организации, которая приобретает систему, программный продукт или программную услугу.
- Поставки. Определяет работы поставщика, то есть организации, которая поставляет систему, программный продукт или программную услугу заказчику.
- Разработки. Определяет работы разработчика, то есть организации, которая проектирует и разрабатывает программный продукт.
- Эксплуатации. Определяет работы оператора, то есть организации, которая обеспечивает эксплуатационное обслуживание вычислительной системы в заданных условиях в интересах пользователей.
- Сопровождения. Определяет работы персонала сопровождения, то есть организации, которая предоставляет услуги по сопровождению программного продукта, состоящие в контролируемом изменении программного продукта с целью сохранения его исходного состояния и функциональных возможностей. Данный процесс охватывает перенос и снятие с эксплуатации программного продукта.

● Вспомогательные

- Документирования. Определяет работы по описанию информации, выдаваемой в процессе жизненного цикла.
- Управления конфигурацией. Определяет работы по управлению конфигурацией.
- Обеспечения качества. Определяет работы по объективному обеспечению того, чтобы программные продукты и процессы соответствовали требованиям, установленным для них, и реализовывались в рамках утвержденных планов. Совместные анализы, аудиторские проверки, верификация и аттестация могут использоваться в качестве методов обеспечения качества.

● Организационные

- Управления. Определяет основные работы по управлению, включая управление проектом, при реализации процессов жизненного цикла.
- Создания инфраструктуры. Определяет основные работы по созданию основной структуры процесса жизненного цикла.
- Усовершенствования. Определяет основные работы, которые организация (заказчика, поставщика, разработчика, оператора, персонала сопровождения или администратора другого процесса) выполняет при создании, оценке, контроле и усовершенствовании выбранных процессов жизненного цикла.
- Обучения. Определяет работы по соответствующему обучению персонала.

ISO/IEC 15504

- В 1998 г. стандарт ISO/IEC TR 15504: Information Technology - Software Process Assessment (Оценка процессов разработки ПО)
- Пять категорий процессов:
- Основные процессы: категория CUS: Потребитель-поставщик и категория ENG: Инженерная
- вспомогательные процессы: категория SUP: вспомогательная
- Организационные процессы: категория MAN: Управленческая и категория ORG: Организационная

Категория Потребитель- Поставщик

- CUS.1 Процесс приобретения (Acquisition process)
 - CUS.1.1 Процесс подготовки приобретения (Acquisition preparation process)
 - CUS.1.2 Процесс выбора поставщика (Supplier selection process)
 - CUS.1.3 Процесс мониторинга поставщика (Supplier Monitoring process)
 - CUS.1.4 Процесс приемки (Customer Acceptance process)
- CUS.2 Поставки (Supply process)
- CUS.3 Процесс выявления требований (Requirements process)
- CUS.4 Эксплуатации (Operation process)
 - CUS.4.1 Процесс эксплуатационного использования (Operational use process)
 - CUS.4.2 Процесс поддержки потребителя (Customer support process)

Инженерная категория

- ENG.1 Процесс разработки (Development process)
 - ENG.1.1 Процесс анализа требований и разработки системы (System requirements analysis and design process)
 - ENG.1.2 Процесс анализа требований к программным средствам (Software requirements analysis process)
 - ENG.1.3 Процесс проектирования программных средств (Software design process)
 - ENG.1.4 Процесс конструирования программных средств (Software construction process)
 - ENG.1.5 Процесс интеграции программных средств (Software integration process)
 - ENG.1.6 Процесс тестирования программных средств (Software testing process)
 - ENG.1.7 Процесс интеграции и тестирования системы (System integration and testing process)
- ENG.2 Процесс сопровождения системы и программных средств (System and software maintenance process)

Вспомогательная категория

- SUP.1 Процесс документирования (Documentation process)
- SUP.2 Процесс управления конфигурацией (Configuration management process)
- SUP.3 Процесс обеспечения качества (Quality assurance process)
- SUP.4 Процесс верификации (Verification process)
- SUP.5 Процесс проверки соответствия (Validation process)
- SUP.6 Процесс совместных проверок (Joint review process)
- SUP.7 Процесс аудита (Audit process)
- SUP.8 Процесс разрешения проблем (Problem resolution process)

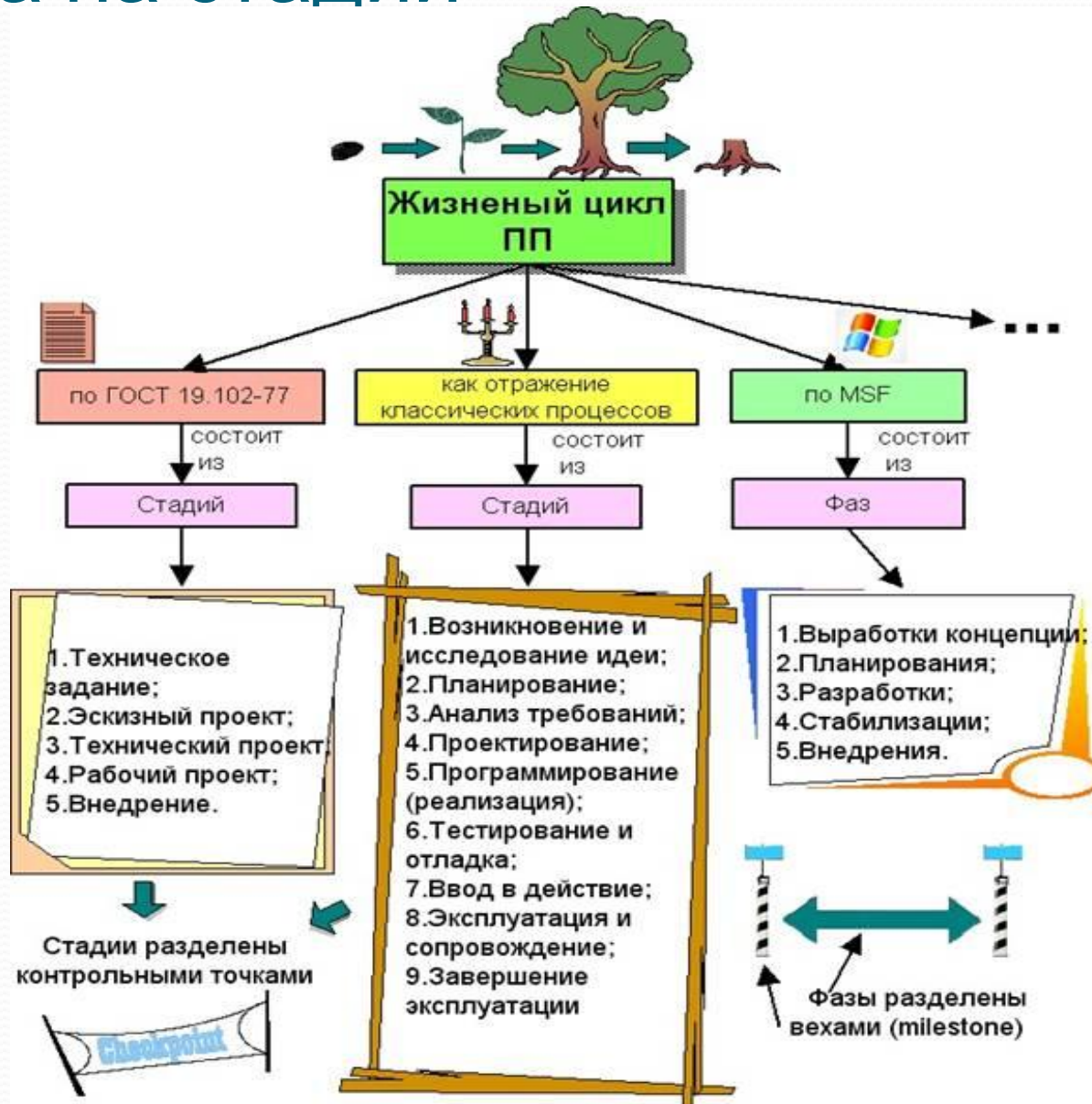
Управленческая категория

- MAN.1 Процесс административного управления (Management process)
- MAN.2 Процесс управления проектами (Project management process)
- MAN.3 Процесс управления качеством (Quality Management process)
- MAN.4 Процесс управления рисками (Risk Management process)

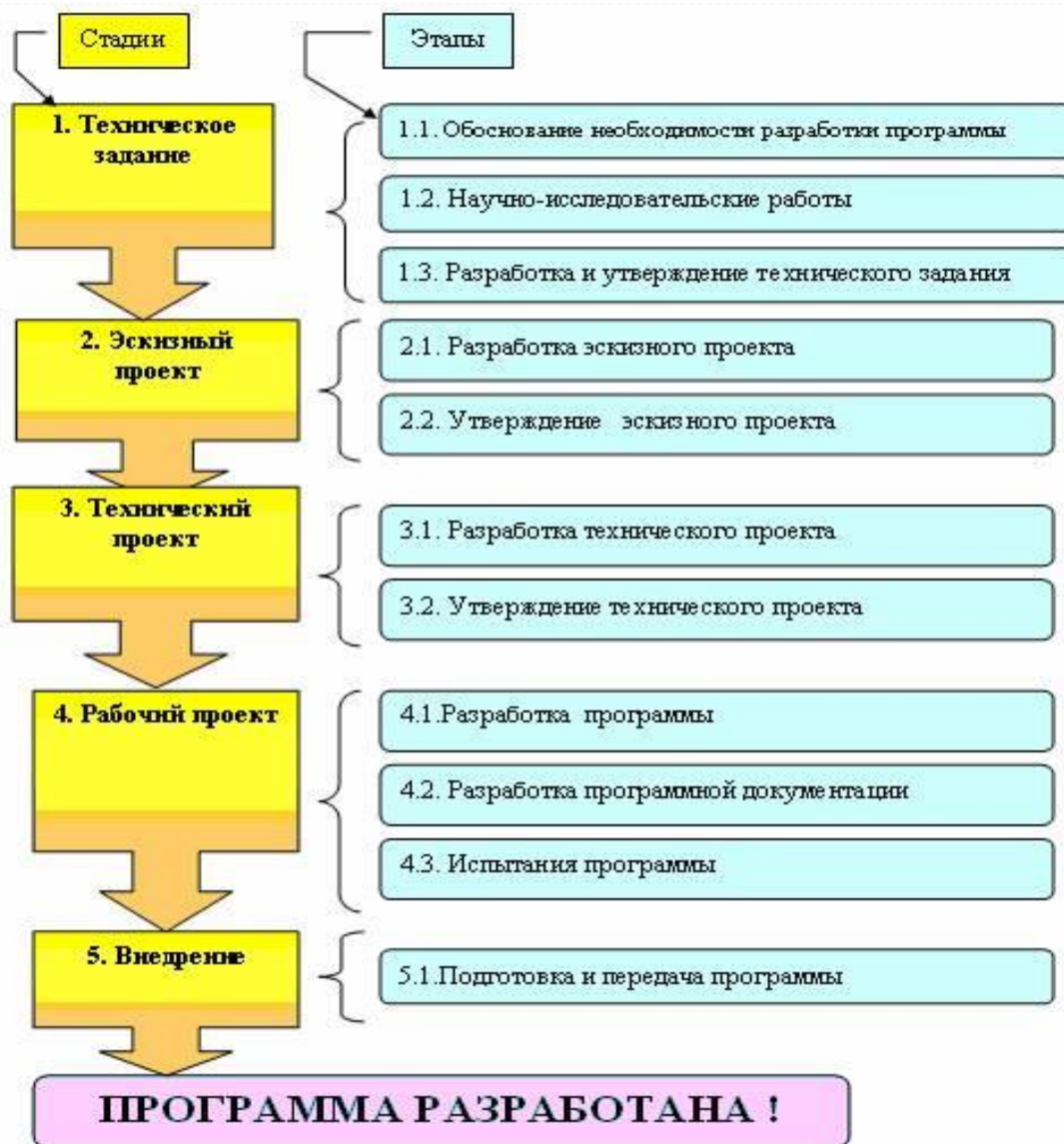
Организационная категория

- ORG.1 Процесс организационных установок (Organizational alignment process)
- ORG.2 Процесс усовершенствования (Improvement process)
 - ORG.2.1 Процесс создания процессов (Process establishment process)
 - ORG.2.2 Процесс аттестации процессов (Process assessment process)
 - ORG.2.3 Процесс усовершенствования процессов (Process improvement process)
- ORG.3 Процесс административного управления кадрами (Human resource management process)
- ORG.4 Процесс создания инфраструктуры (Infrastructure process)
- ORG.5 Процесс измерения (Measurement process)
- ORG.6 Процесс повторного использования (Reuse process)

Варианты разделения жизненного цикла на стадии



Содержание стадий создания программы по ГОСТ 19.102-77



Жизненный цикл ПО согласно методологии MSF

Популярная методология разработки программных средств MSF (Microsoft Solutions Framework) компании Microsoft предполагает разбиение жизненного цикла ПО на фазы:

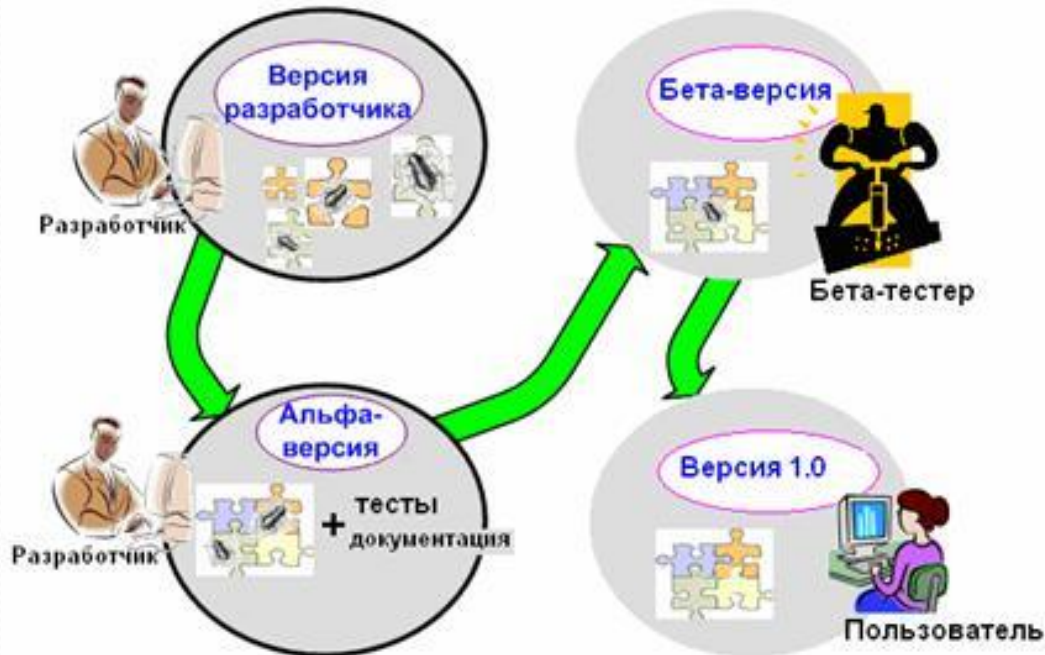
- выработки концепции
- планирования
- разработки
- стабилизации
- внедрения

Каждая фаза завершается контрольной точкой, называемой вехой – milestone . Название вех звучит примерно следующим образом: «Концепция утверждена», «План проекта утверждён».



Последовательность создания версий программного продукта

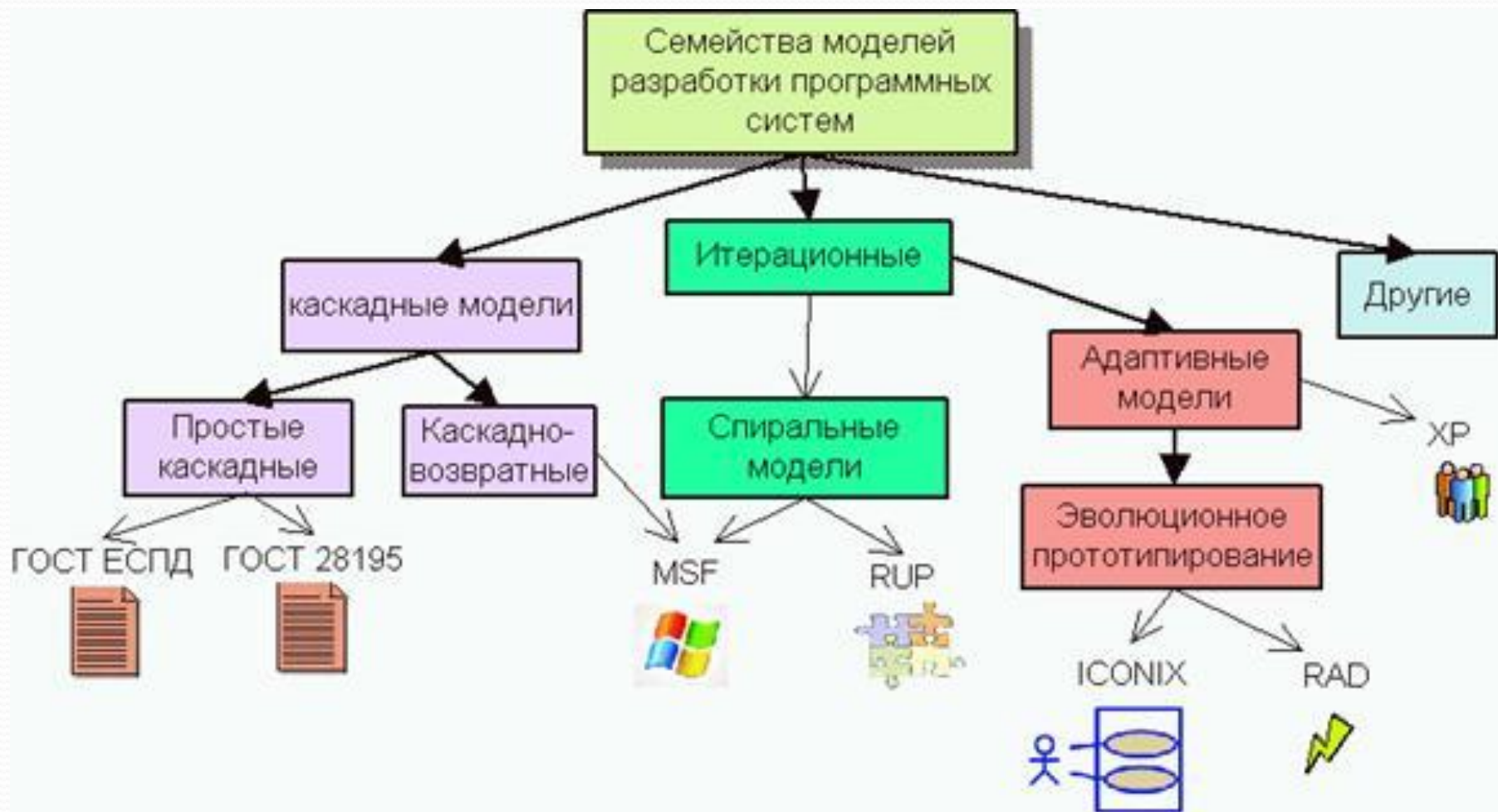
- При использовании итерационных моделей в качестве контрольных точек часто могут использоваться выпуски версий продукта. Тестируя каждую из выпущенных версий, можно сформировать достаточно полную картину того, какие из намеченных целей достигнуты, и какие изменения необходимо внести.
- Иногда выпускаемые версии обозначают так, как изображено на рисунке. Однако приведённые названия версий не являются обязательными, и разработчики могут вовсе не использовать их, а, например, просто присваивать версиям номера.



Версии программного продукта

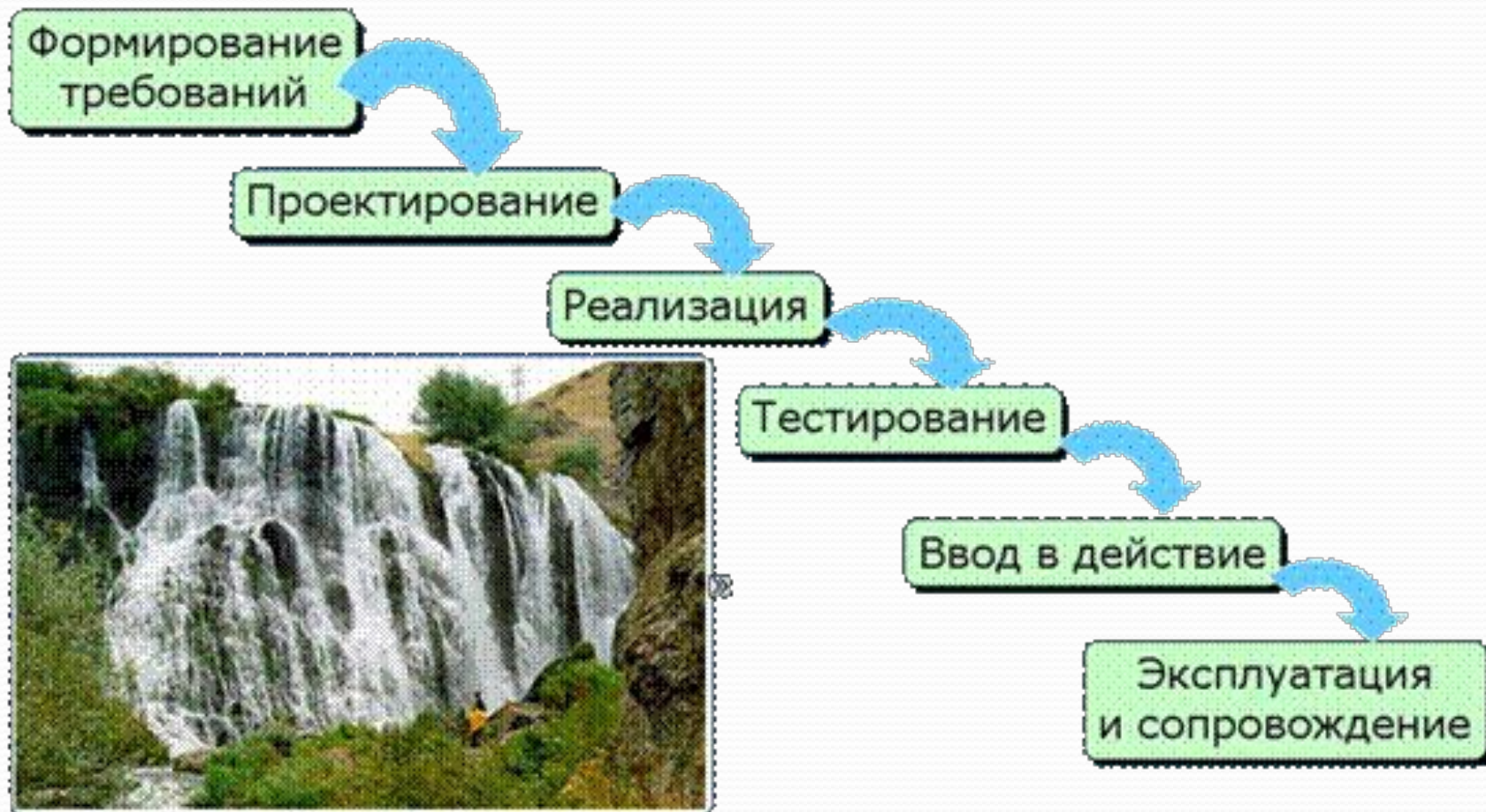
- *Версия разработчика*, как правило, содержит большое количество ошибок, неполную функциональность, не имеет достаточного количества документации. Часто бывает, что на стадии выпуска этой версии не произведена интеграция, и некоторые модули программы работают по отдельности. Данную версию не выпускают за пределы круга разработчиков.
- *Альфа-версия* обычно имеет уже полный или почти полный функционал, документацию, все компоненты интегрированы, но для внешнего тестирования обычно альфа-версия не передаётся.
- *Бета-версия* содержит меньшее количество ошибок и может предоставляться тестировщикам. Это делается из-за того, что разработчики зачастую не могут выявить некоторые ошибки из-за психологического фактора (объективно оценить результат своей деятельности достаточно сложно).
- На этапе первой *поставки пользователю* программный продукт должен содержать минимум ошибок и выполнять поставленные заранее «критерии первой поставки», которые оговариваются с заказчиком.

Модели разработки ПО



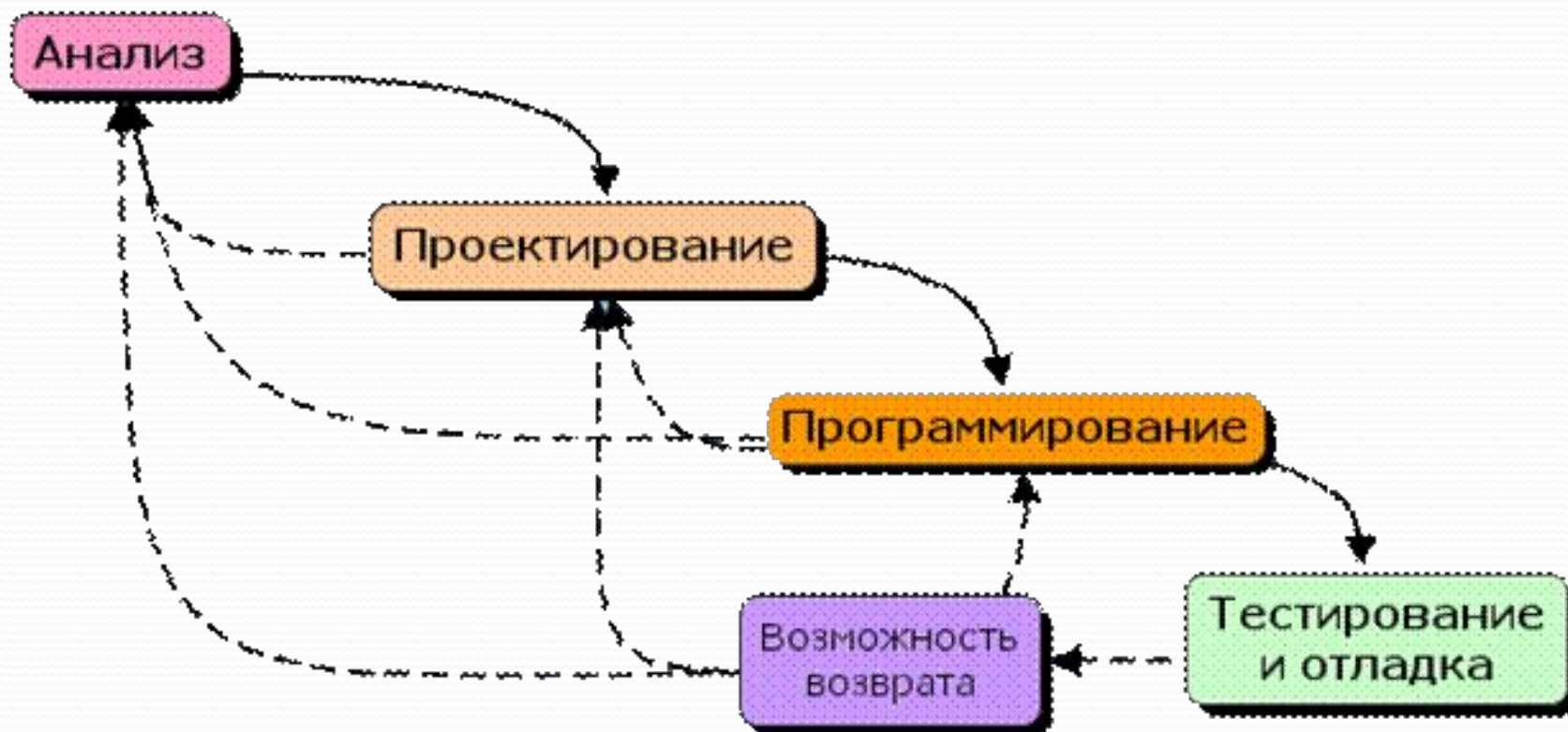
Семейство каскадных

моделей: простая каскадная модель



Семейство каскадных

моделей: каскадно - возвратная модель



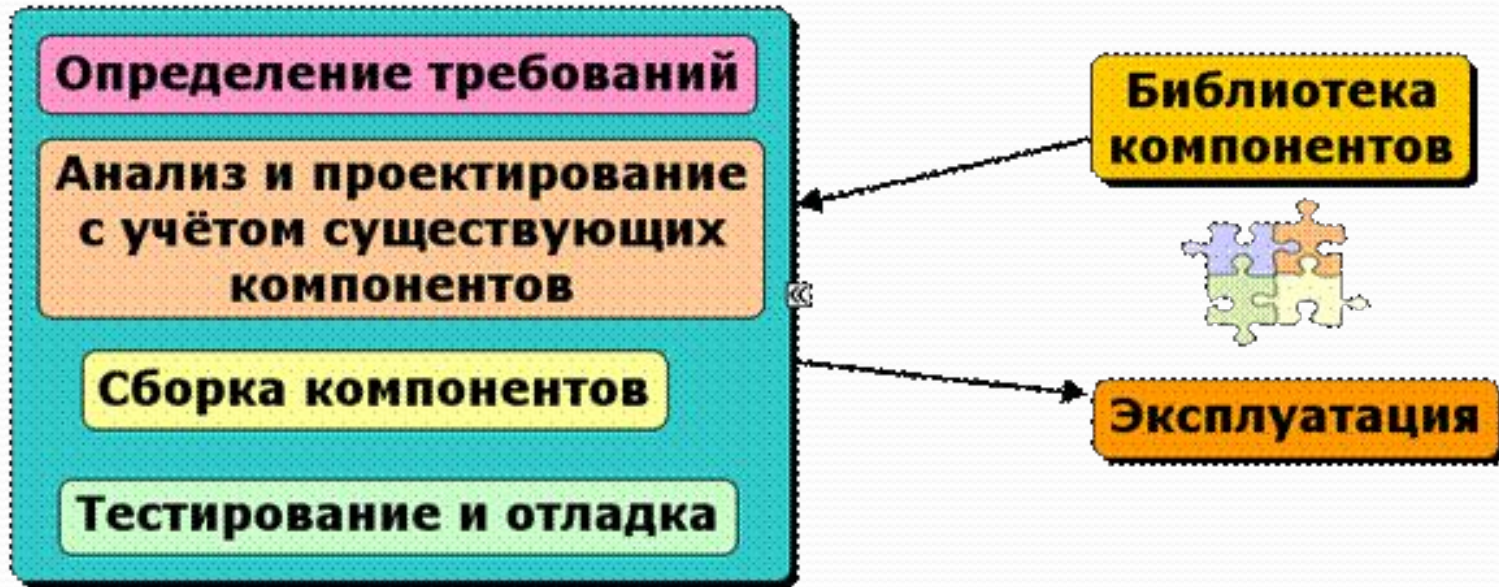
Семейство итерационных моделей: спиральные модели



Семейство итерационных моделей: каркасная модель разработки

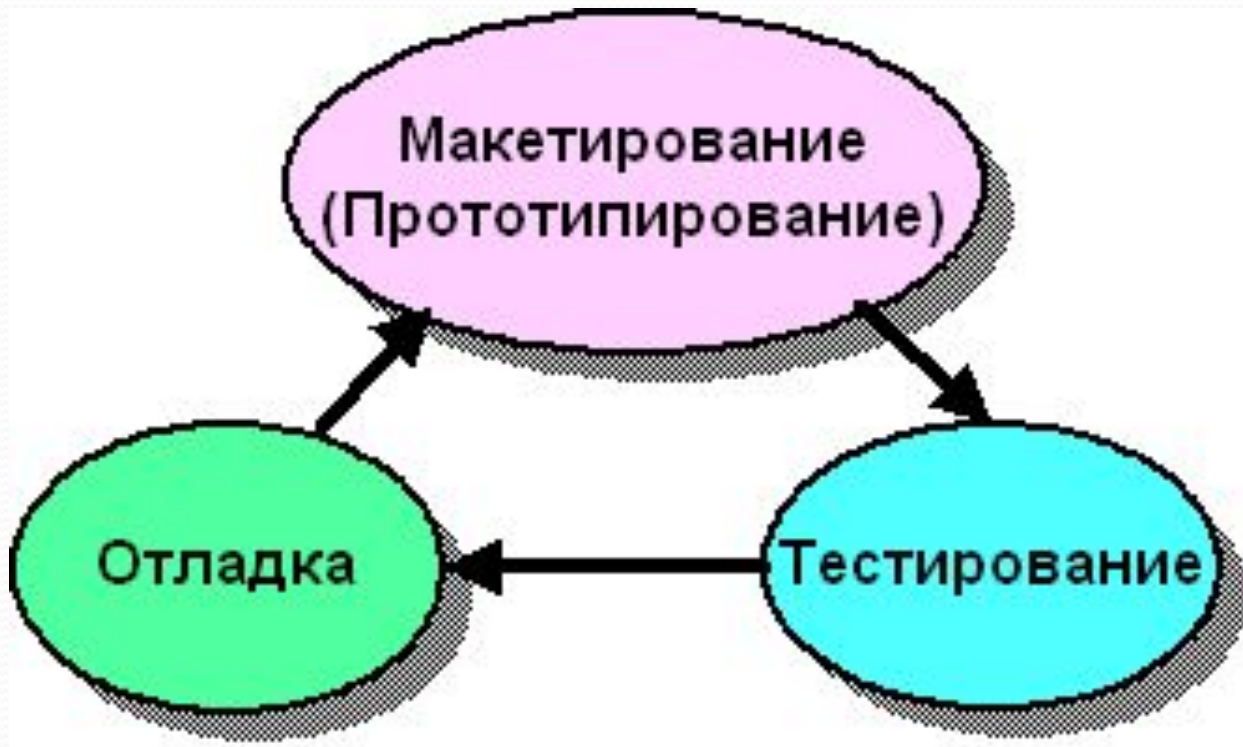


Другие модели: сборочное программирование



Другие модели:

исследовательское программирование



Заключение

- Модель жизненного цикла программного продукта связана с моделью разработки программного продукта
- Итерационные модели разработки являются более гибкими
- При создании очень надежных систем рекомендуется использовать каскадные методы разработки