

Оптимизация программных средств

РАЗДЕЛ 1. «ОРГАНИЗАЦИЯ РАЗРАБОТКИ ПРОГРАММНЫХ СИСТЕМ»

ТЕМА 1. История развития
методологий разработки
сложных систем

Особенностью современных масштабных проектов является необходимость участия в них большого количества исполнителей, выполняющих творческую, интеллектуальную работу.

Противоречие.

С одной стороны, необходимо получить сложное и принципиально новое изделие со строго заданными характеристиками в заданные сроки.

С другой стороны, заранее не известны методы решения задач, поставленных перед исполнителями.

Характерные особенности успешных крупных проектов середины XX в.:

1) заказчиком проектов выступала политическая элита государств, ясно и четко определяющая цель проектов и готовая потратить на их осуществление значительные ресурсы при жестких ограничениях на сроки реализации;

2) создавался коллектив разработчиков, нацеленный на конечный результат;

3) было организовано эффективное взаимодействие между разработчиками: сбор разработчиков в одном месте и обеспечение возможности непосредственного общения между ними (Арзамас-16, Челябинск-70, Лос-Аламос);

4) несмотря на наличие жестких сроков, существовала обратная связь между разработчиками и высшим руководством, которая позволяла корректировать принятые решения и вовремя ставить новые задачи;

5) большое значение имел человеческий фактор. К разработке привлекались наиболее квалифицированные специалисты.

ТЕМА 2. Жизненный цикл программной системы

Под **системой** в данном случае понимается совокупность взаимодействующих компонентов и взаимосвязей между ними, направленная на достижение определенной цели.

Под **проектом** будем понимать временное предприятие, осуществляемое в целях создания уникального продукта или услуги.

Программной системой (ПС) будем считать комплекс связанных между собой программ (программный комплекс), снабженный необходимой документацией, ориентированный на сбор, хранение, поиск, обработку, передачу и отображение информации в некоторой предметной области.

Термины информационная система и автоматизированная система во многих источниках используются для обозначения более широкого понятия, в которое кроме программной системы входят технические средства и люди, обеспечивающие работу программной системы.

Под жизненным циклом программной системы обычно понимается непрерывный процесс, который начинается с момента принятия решения о необходимости создания системы и заканчивается в момент ее полного изъятия из эксплуатации.

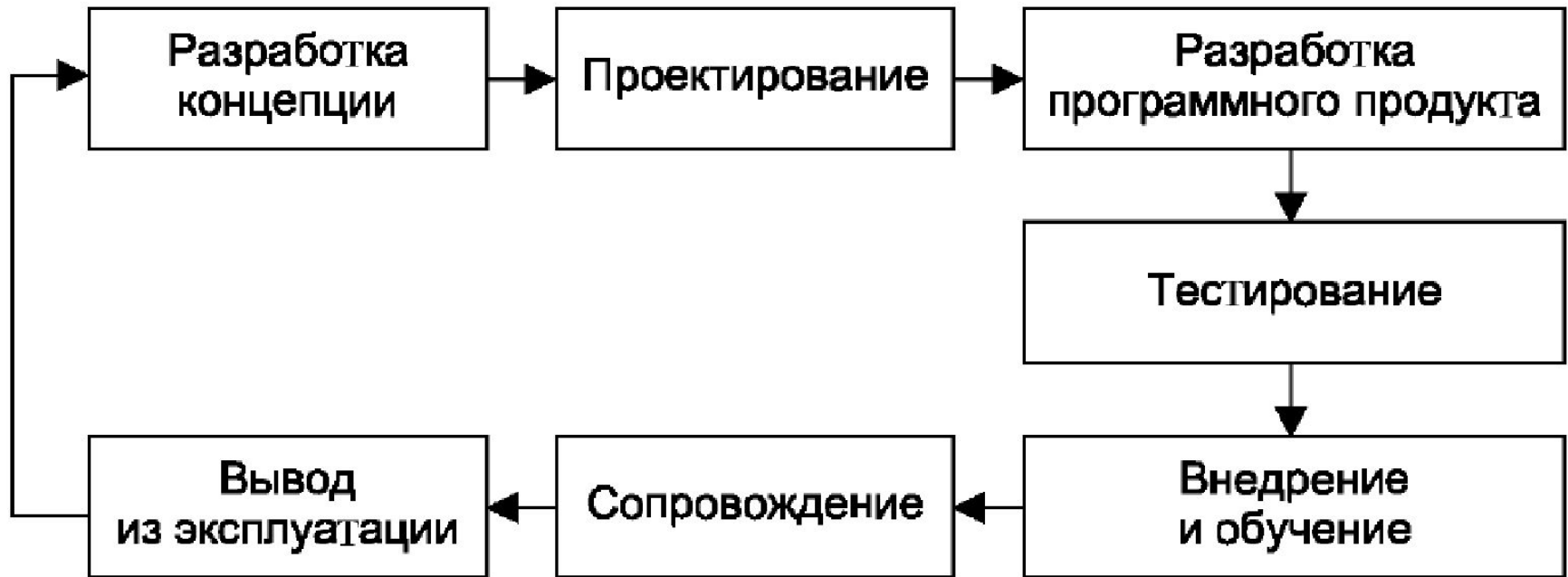
ТЕМА 3. **Международные и
национальные стандарты
методологий разработки
программных систем**

**Отечественные и международные стандарты,
регламентирующие этапы и процессы жизненного цикла ИС**

Стандарт	Название	Комментарий
ГОСТ 34.601-90 [7]	Информационная технология. КСАС. Автоматизированные системы. Стадии создания	Данный стандарт разрабатывался в СССР во времена плановой экономики и государственной собственности на средства производства, когда не было речи о коммерческих программных продуктах. Стандарт отражает сложившуюся на то время в России практику разработки программных продуктов. Учитывает российскую специфику взаимоотношений разработчика и заказчика
ГОСТ 19.102-77 [4]	Единая система программной документации (ЕСПД). Стадии разработки	Первый Российский стандарт, разработанный в 70-е годы прошлого века. Длительное применение этой системы стандартов сформировало определенный стиль взаимодействия заказчика и разработчика, основанный на каскадной модели ЖЦ. Несмотря на то, что этот стандарт разработан достаточно давно, некоторые его положения остаются полезными до настоящего времени

Стандарт	Название	Комментарий
ГОСТ Р ИСО/МЭК 12207-99 [9]	Информационная технология. Процессы жизненного цикла программных средств	Этот стандарт является переводом международного стандарта ISO/IEC 12207:1995 «Software Life Cycle Processes» [13], разработанного в недрах Министерства обороны США. В нем обобщен опыт управления разработками крупных программных проектов по заказу упомянутого министерства
ГОСТ Р ИСО/МЭК 15288-2005 [10]	Системная инженерия – процессы жизненного цикла систем	Российский аналог международного стандарта ISO/IEC 15288 System life cycle processes [18]. Устанавливает последовательность этапов и перечень процессов ЖЦ разработки любых крупных проектов, в том числе и ПС. Введен в России с 1 января 2007г. Наиболее прогрессивный стандарт, в котором учтен международный опыт разработки сложных систем

ISO/IEC 15504 [14]	Information Technology-Software Process Assessment	Международный стандарт, определяющий порядок оценки процессов жизненного цикла ПС
ISO/IEC 9000-3-2002 [17]	Системная и программная инженерия – руководство по применению стандарта ISO 9001-2000 к программному обеспечению	Международный стандарт по управлению качеством разработки ПС



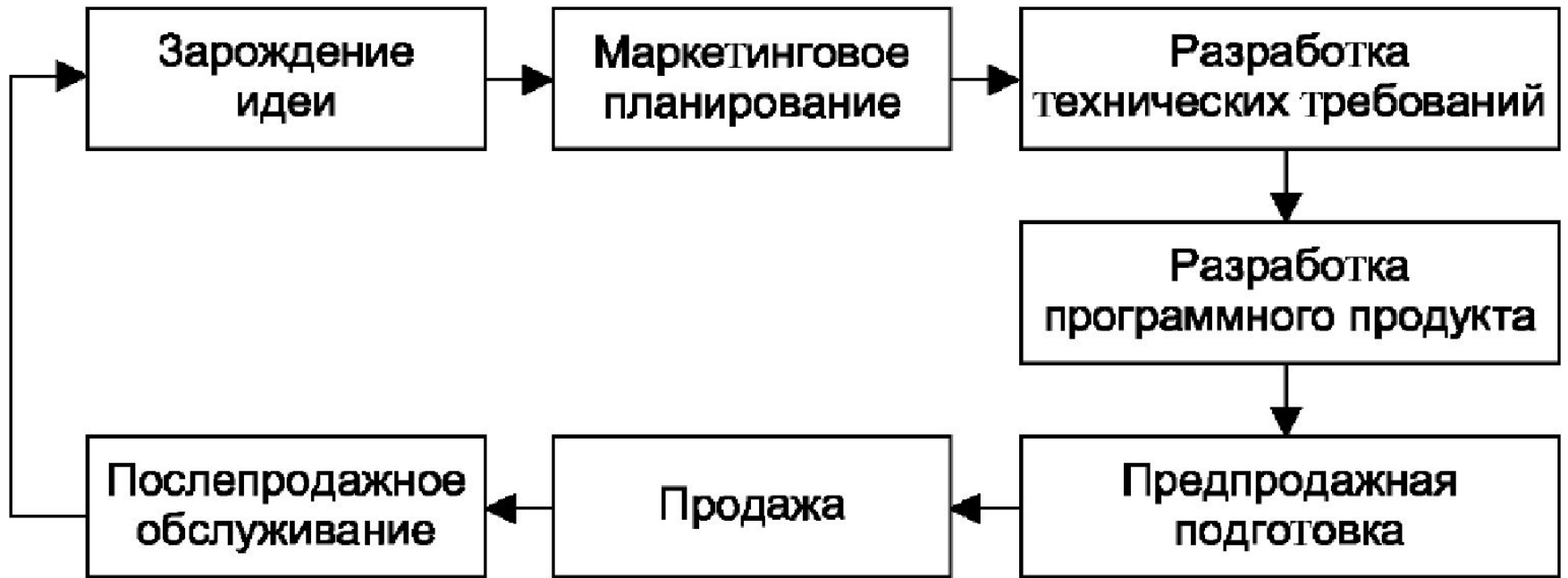
Обобщенная схема этапов жизненного цикла программной системы

Цели и критерии качества этапов жизненного цикла изделия, разрабатываемого по заказу

Этап	Цель	Критерий качества
Разработка концепции	<p>Определение цели программы. Выбор возможных направлений реализации. Оценка целесообразности разработки и реализуемости программы</p>	<p>Единое понимание заказчиком и разработчиком целей и задач программы. Согласованные начальные оценки стоимости затрат и времени разработки</p>
Проектирование. Разработка и анализ технических требований	<p>Анализ бизнес-процессов предприятия, для которого разрабатывается программа. Выделение процессов, подлежащих автоматизации. Формирование системы пользовательских требований. Формирование системы технических требований к изделию. Выбор средств реализации. Планирование разработки и выпуска</p>	<p>Полнота и согласованность потребительских требований, их соответствие концепции программы.</p> <p>Технические требования должны:</p> <ul style="list-style-type: none"> • однозначно пониматься всеми участниками проекта; • быть реализуемыми; • максимально учитывать мнения потенциальных потребителей

Этап	Цель	Критерии качества
Разработка программного продукта	Разработка и изготовление продукта с требуемыми характеристиками, в требуемом объеме и в требуемые сроки	<p>Продукт должен соответствовать техническим требованиям.</p> <p>Сроки и стоимость разработки не должны превышать установленных в плане</p>
Тестирование	<p>Проверка соответствия разработанной программы техническим требованиям:</p> <ul style="list-style-type: none"> • тестирование; • сертификация. <p>Регистрация программы как объекта интеллектуальной собственности</p>	<p>Программа должна соответствовать предъявленным к ней требованиям.</p> <p>Заказчик должен быть убежден в том, что программа:</p> <ul style="list-style-type: none"> • работает правильно; • защищена как объект интеллектуальной собственности

Этап	Цель	Критерий качества
Внедрение и обучение	<p>Подготовка производства и персонала к использованию программы.</p> <p>Адаптация программы к особенностям ее использования</p>	<p>Готовность производства к эксплуатации программы.</p> <p>Минимум сбоев и несоответствий в период эксплуатации</p>
Сопровождение	<p>Помощь пользователям, выявление и разрешение проблем, связанных с использованием программы, накопление опыта, который будет использован при реализации будущих проектов</p>	<p>Качество программы определяется количеством и сложностью проблем, связанных с ее использованием, а также с частотой их появления. Чем меньше эти характеристики, тем выше качество, тем выше репутация разработчиков</p>



Этапы жизненного цикла программного продукта, реализуемого на свободном рынке (в соответствии с международными стандартами серии ISO 9000)

**Цели и критерии качества этапов жизненного цикла изделия,
выпускаемого на продажу**

Этап	Цель	Критерий качества
Зарождение идеи	Идея нового изделия, которое может создать фирма	Решает реальные задачи потребителя. Выгодно отличается от существующих изделий
Маркетинговое планирование	<p>Определение круга потребителей, изучение их вкусов и предпочтений. Анализ аналогов и конкурентов.</p> <p>Оценка доли рынка, которую может «захватить» изделие.</p> <p>Маркетинговый план</p>	Целостность, согласованность и надежность результатов исследований и оценок
Разработка технических требований	<p>Формирование четких технических требований к изделию.</p> <p>Планирование разработки и выпуска</p>	<p>Технические требования должны:</p> <ul style="list-style-type: none"> • однозначно пониматься всеми участниками проекта; • быть реализуемыми; • максимально учитывать мнения потенциальных потребителей

<p>Разработка продукта</p>	<p>Разработка и изготовление продукта с требуемыми характеристиками, в требуемом объеме и в требуемые сроки</p>	<p>Продукт должен соответствовать техническим требованиям. Сроки и стоимость разработки не должны превышать установленных в плане</p>
<p>Предпродажная подготовка</p>	<p>Превращение продукта в товар:</p> <ul style="list-style-type: none"> • регистрация продукта как объекта интеллектуальной собственности; • тестирование и сертификация; • определение способов продвижения 	<p>Товар должен быть:</p> <ul style="list-style-type: none"> • известен на рынке; • достаточно надежным, чтобы его продажи не повредили репутации фирмы; • защищен от несанкционированного использования (пиратства)
<p>Продажа</p>	<p>Продажа товара максимальному числу покупателей, нуждающихся в этом товаре</p>	<p>Приобретен максимумом пользователей, которым он нужен. Репутация фирмы на рынке повысилась</p>

Этап	Цель	Критерий качества
Послепродажное обслуживание	<p>Помощь пользователям.</p> <p>Поддержание интереса к товару как можно на более длительный срок.</p> <p>Выявление проблем, связанных с использованием товара</p>	<p>Фирма имеет представление о том, что нужно пользователю, в каком направлении развивать товар.</p> <p>Репутация фирмы на рынке повысилась</p>

Процессы жизненного цикла систем. Стандарт ГОСТ Р ИСО/МЭК 15288-2005.

Процессом называется совокупность взаимосвязанных действий, преобразующих некоторые входные объекты или данные в выходные.

Перечень процессов и составляющих их действий по ГОСТ 15288.

Процессы соглашения:

- 1) процесс приобретения (используется организациями для приобретения продукции или получения услуг);
- 2) процесс поставки (используется организациями для поставок продукции или оказания услуг).

Процессы предприятия:

- 1) управления предприятием;
- 2) управления инвестициями;
- 3) управления жизненным циклом системы;
- 4) управления ресурсами;
- 5) управления качеством.

Перечень процессов и составляющих их действий по ГОСТ 15288.

Процессы проекта:

- 1) планирования проекта;
- 2) оценки проекта;
- 3) контроля проекта;
- 4) принятия решений;
- 5) управления рисками;
- 6) управления конфигурацией;
- 7) управления информацией.

Перечень процессов и составляющих их действий по ГОСТ 15288.

Технические процессы:

- 1) определения требований заказчика;
- 2) анализа требований;
- 3) проектирования архитектуры;
- 4) реализации проекта;
- 5) комплексирования;
- 6) верификации;
- 7) передачи заказчику;
- 8) валидации;
- 9) функционирования;
- 10) сопровождения;
- 11) списания.

Процессы жизненного цикла программных средств. Стандарт ГОСТ Р ИСО/МЭК 12207-99.

Все процессы, в отличие от стандарта
ГОСТ 15288, сгруппированы в три класса:

1. Основные процессы: приобретение, поставка,
разработка, эксплуатация, сопровождение.

2. Вспомогательные процессы: документирование,
управление конфигурацией, обеспечение
качества, разрешение проблем,
аудит, аттестация, совместная оценка,
верификация.

3. Организационные процессы: создание
инфраструктуры, управление; обучение,
усовершенствование.

ТЕМА 4. Модели жизненного цикла программных систем

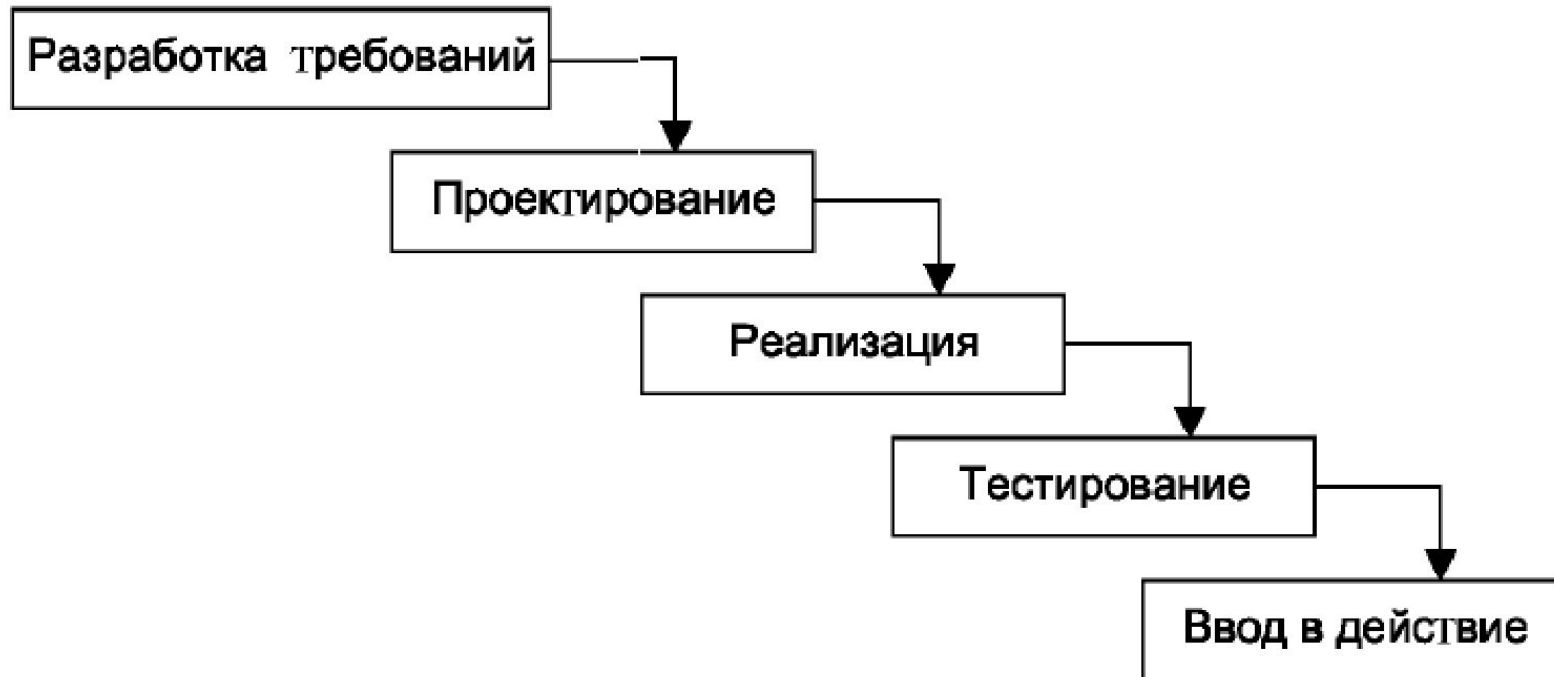
Модель жизненного цикла — структура, содержащая процессы, действия и задачи, которые осуществляются в ходе разработки, функционирования и сопровождения программного продукта.

В настоящее время известны и используются следующие **модели жизненного цикла**:

- **каскадная (водопадная);**
- **поэтапная с промежуточным контролем;**
- **спиральная (эволюционная);**
- **инкрементная.**

Каскадная (водопадная) модель

Каскадная (водопадная) модель ЖЦ рассматривает жизненный цикл ПС как последовательность стадий.



Типовые стадии создания автоматизированной информационной системы согласно стандарту ГОСТ 34.601-90:

Стадия 1. Формирование требований к ПС:

- обследование объекта и обоснование необходимости создания ПС;
- формирование требований пользователей к ПС;
- оформление отчета о выполненной работе и тактико-технического задания на разработку.

Стадия 2. Разработка концепции ПС:

- изучение объекта автоматизации;
- проведение необходимых научно-исследовательских работ;
- разработка вариантов концепции ПС, удовлетворяющих требованиям пользователей;
- оформление отчета и утверждение концепции.

Стадия 3. Техническое задание:

- разработка и утверждение технического задания на создание ПС.

Стадия 4. Эскизный проект:

- разработка предварительных проектных решений по системе и ее частям;
- разработка эскизной документации на ПС и ее части.

Стадия 5. Технический проект:

- разработка проектных решений по системе и ее частям;
- разработка документации на ПС и ее части;
- разработка и оформление документации на поставку комплектующих изделий;
- разработка заданий на проектирование в смежных частях проекта.

Стадия 6. Рабочая документация:

- разработка рабочей документации на ПС и ее части;
- разработка и адаптация программ.

Стадия 7. **Ввод в действие:**

- подготовка объекта автоматизации;
- подготовка персонала;
- комплектация ПС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями);
- строительно-монтажные работы;
- пусконаладочные работы;
- проведение предварительных испытаний;
- проведение опытной эксплуатации;
- проведение приемочных испытаний.

Стадия 8. **Сопровождение ПС:**

- выполнение работ в соответствии с гарантийными обязательствами;
- послегарантийное обслуживание.

Каскадная модель

«+»

- явное описание всех этапов работы и определение последовательности их реализации
- выполняемые в логической последовательности этапы работ позволяют планировать сроки завершения всех работ и соответствующие затраты.
- имеется возможность сопоставлять и оценивать различные варианты их реализации, а также накапливать статистические данные о фактической трудоемкости и затратах на реализацию каждого этапа.

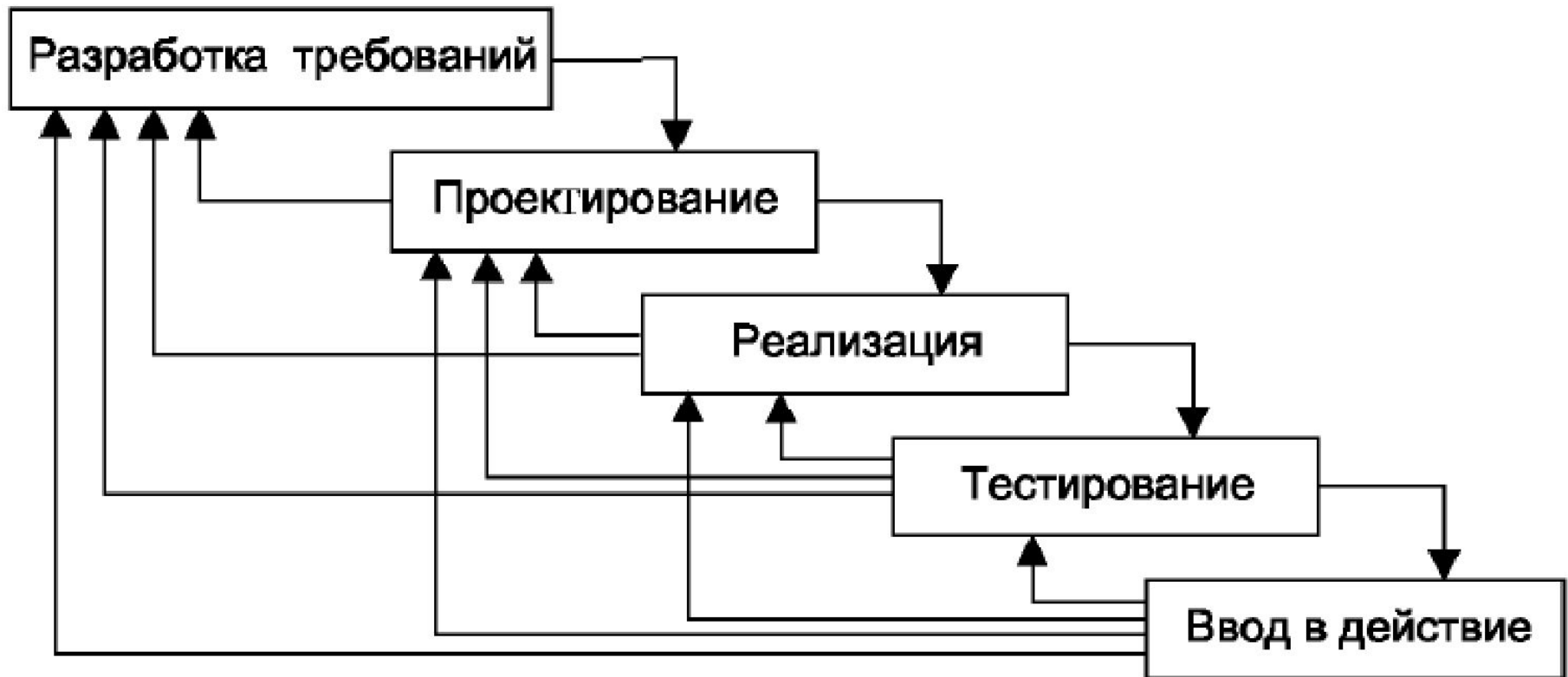
Каскадная модель

«-»

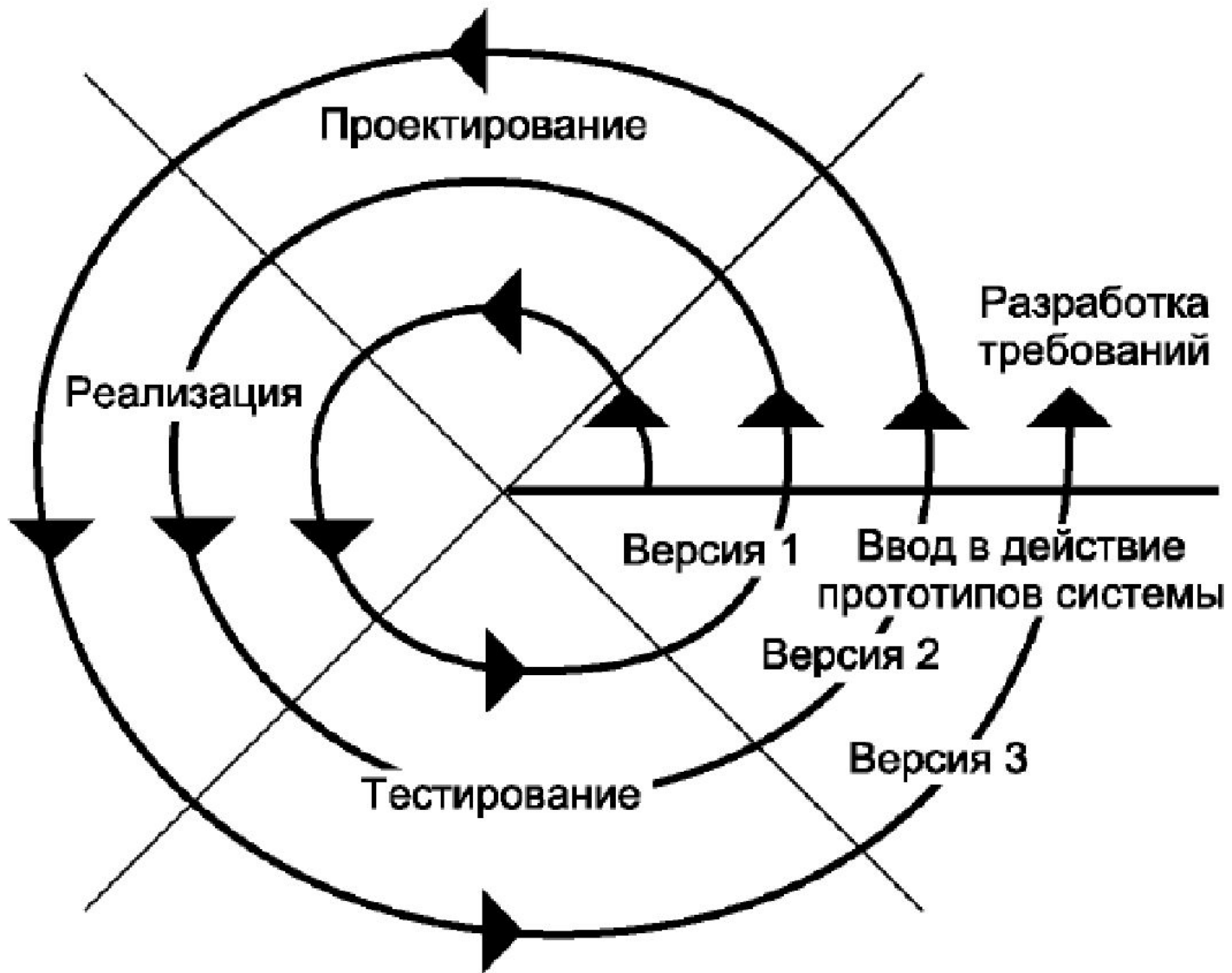
- слабо отражены процессы управления
- реальный процесс создания системы никогда полностью не укладывается в жесткую каскадную схему, постоянно возникает потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений

Поэтапная модель с промежуточным контролем

Модификация предыдущей модели. Разработка ПС в этой модели ведется итерациями с циклами обратной связи между этапами.



Спиральная модель



Спиральная модель

Прототип программы — версия программы, предназначенная для демонстрации ее основных свойств.

«+»

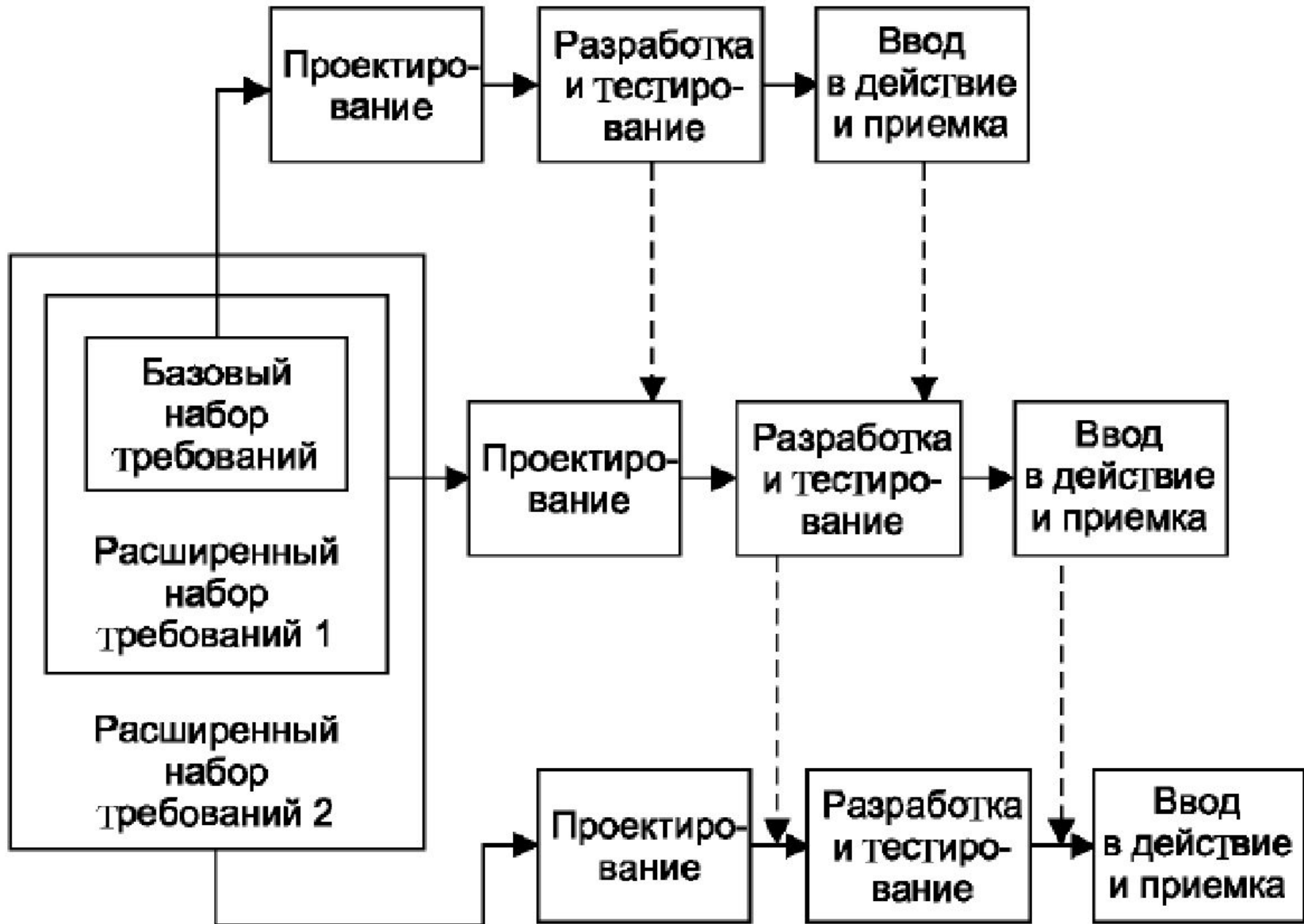
- до реализации доводится обоснованный окончательный вариант ПС, который удовлетворяет действительным требованиям заказчика
- ускорение разработки ПС, обусловленное более активным привлечением заказчика к формированию требований на основе анализа работы прототипов, а также интенсификация труда разработчиков ПС

Спиральная модель

«-»

- сложность планирования работ, оценки затрат, сроков и рисков выполнения проекта
- проблема определение момента перехода на следующий этап

Инкрементная модель



ТЕМА 5. **Документальное
сопровождение этапов
жизненного цикла
программной системы**

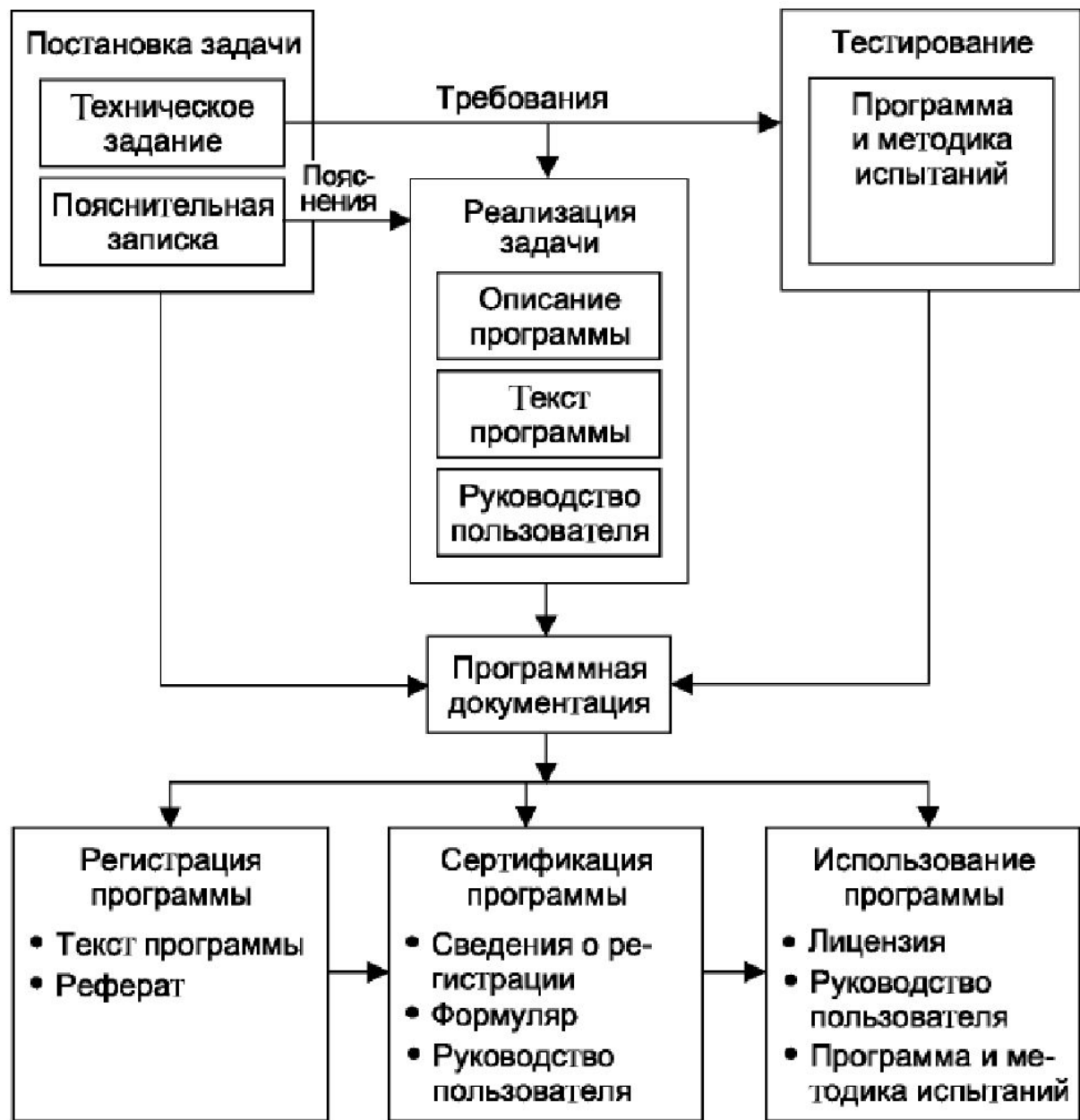


Схема создания и использования программной документации

В России действуют два стандарта на состав программной документации — «ГОСТ 19.101-77 Виды программ и программных документов» (далее ГОСТ 19.101) и «ГОСТ 34.201-89 Виды, комплектность и обозначение документов при создании автоматизированных систем» (далее ГОСТ 34.201)

**Перечень документов, разрабатываемых на этапах ЖЦ
согласно стандарту ГОСТ 34.201-89**

Стадия создания ПС	Предъявляемые документы	Комментарий
Разработка концепции ПС	Отчет о НИР	Обычно НИР выполняется перед разработкой принципиально новой ПС
Формирование требований к ПС	Отчет об обследовании объекта компьютеризации	Бизнес-модель компьютеризованного процесса с указанием места и роли ПС
Техническое задание (ТЗ)	Техническое задание	Требования к содержанию, структуре и оформлению ТЗ определены в ГОСТ 34.602-89 [6]
Эскизный проект	Ведомость эскизного проекта	Перечень документов эскизного проекта
	Пояснительная записка к эскизному проекту	Обоснование принятых проектных решений

Стадия создания ПС	Предъявляемые документы	Комментарий
	Схема организационной структуры	Схема взаимодействия участников компьютеризируемого процесса
	Схема функциональной структуры	Иерархический перечень функций ПС с указанием взаимосвязей между функциями
Технический проект	Ведомость технического проекта	Документы технического проекта отличаются более глубокой проработкой и наличием программного кода, реализующего указанные требования
	Схема структурная комплекса технических средств	
	Пояснительная записка к техническому проекту	
	Ведомость оборудования и материалов	
	Предварительная смета затрат	Окончательная смета затрат может появиться только после завершения проекта

Рабочая документация	Ведомость эксплуатационных документов	Требования к содержанию документа приведены в РД 50-34.698-90, ГОСТ 34.201
	Паспорт или Формуляр	Разрабатывается только один документ – или «Паспорт», или «Формуляр»
	Общее описание системы	Документ не разрабатывается, если на стадии «Технический проект» разрабатывается документ «Пояснительная записка к техническому проекту»
	Программа и методика испытаний	Документ должен содержать описания методов испытания для контроля выполнения всех требований, записанных в ТЗ. Допускаются ссылки на стандартные методы испытаний

Стадия создания ПС	Предъявляемые документы	Комментарий
	Спецификация оборудования	Перечень требований к оборудованию ПС. Требования к содержанию документов приведены в ГОСТ 21.110
	Инструкция по эксплуатации комплекса технических средств	Требования к содержанию документов приведены в РД 50-34.698-90 [11]
	Схемы соединения и подключения внешних устройств	
	Руководство пользователя	Описание методов работы с ПС, ориентированное на пользователя
	Руководство системного администратора	Документ, регламентирующий права и обязанности персонала ПС, процедуры организации доступа, обеспечения работоспособности и безопасности ПС
	Регламент штатного обслуживания	Документ, регламентирующий условия обслуживания ПС в штатном режиме, проведение мониторинга и резервного копирования, организацию антивирусной защиты и других штатных операций
	Регламент аварийного обслуживания	Документ, классифицирующий основные типы сбоев, описывающий методы восстановления работоспособности ПС и ее подсистем после сбоев указанных типов
	Регламент опытной эксплуатации	Документ, определяющий цели, продолжительность и порядок проведения опытной эксплуатации. Содержит перечни характеристик, контролируемых в ходе опытной эксплуатации, порядок и методы их контроля

Стадия создания ПС	Предъявляемые документы	Комментарий
	Инструкция по формированию и ведению базы данных	Документ, определяющий источники пополнения информации ПС, процедуры получения и обновления необходимой информации, методы обеспечения целостности и актуальности информационной базы ПС
Ввод в промышленную эксплуатацию	Программа ввода ПС в промышленную эксплуатацию	Документ, устанавливающий последовательность этапов ввода и перечни мероприятий на каждом этапе
	Программа обучения персонала	Документ, классифицирующий персонал, определяющий цели, требования и учебные планы для каждой категории персонала

Следование стандартам — дело добровольное.

Выбор стандартов является результатом договоренностей заказчика и разработчика.

ТЕМА 6. **Фирменные
(корпоративные) технологии
разработки программной
системы**

**Наиболее известные «фирменные» (корпоративные)
методологии разработки ПС**

Название методологии	Инструментальное средство автоматизированной разработки	Комментарий
Microsoft Solutions Framework (MSF) – разработка компании Microsoft	Microsoft Visual Studio	Стандарт, ориентированный на спиральную и инкрементную модели разработки. Включил в себя опыт Microsoft по реализации различных программных проектов
Rational Unified Process (RUP) – разработка компании Rational Software, входящей в состав корпорации IBM	<p>Rational Rose – проектирование и моделирование информационных систем и бизнес-процессов.</p> <p>Rational SoDa – автоматизация процесса создания документации.</p> <p>Rational RequisitePro – управление требованиями.</p> <p>Rational ClearQuest – управление изменениями в проекте. Интегрируется с Rational RequisitePro, Microsoft Project, Test Manager, ClearCase.</p> <p>Rational ClearCase – инструмент управления проектами. Поддерживает версию документов, моделей и т.д.</p> <p>Rational XDE Professional v2002: Microsoft.NET Edition – интегрируется в Microsoft Visual.NET, позволяя работать в одной среде. Дает возможность как генерировать код согласно построенной модели, так и создавать шаблоны проектов для использования в будущем</p>	Объемный стандарт, ориентированный на спиральную и инкрементную модели разработки, а также на объектно-ориентированное проектирование (ООП) с помощью CASE-средств и языка UML. Претендует на роль мирового корпоративного стандарта

Название	Инструментальное средство автоматизированной разработки	Комментарий
<p>ICONIX – разработка компании ICONIX Software Engineering, Inc</p>	<p>Предлагается использование CASE-средств разработки компании Rational Software</p>	<p>Упрощенный вариант методологии RUP. Особое внимание в стандарте уделено анализу требований и начальным стадиям проектирования. Для моделирования используется упрощенное подмножество языка UML</p>
<p>Oracle Method – разработка компании Oracle</p>	<p>Oracle Developer Suite</p>	<p>Предусматривает каскадную и итерационные модели разработки</p>

Стандарты и модели жизненного цикла ПС корпорации Microsoft.

Корпорация Microsoft разработала две связанные и дополняющие друг друга фирменные методологии: Microsoft Solutions Framework (**MSF**) — для процессов разработки ПС и Microsoft Operations Framework (**MOF**) — для процессов сопровождения и эксплуатации ПС.

Модель **MSF** основана на спиральной модели разработки и состоит из пяти этапов:

- создания общей картины приложения (разработка концепции);
- планирования;
- разработки;
- стабилизации;
- развертывания.

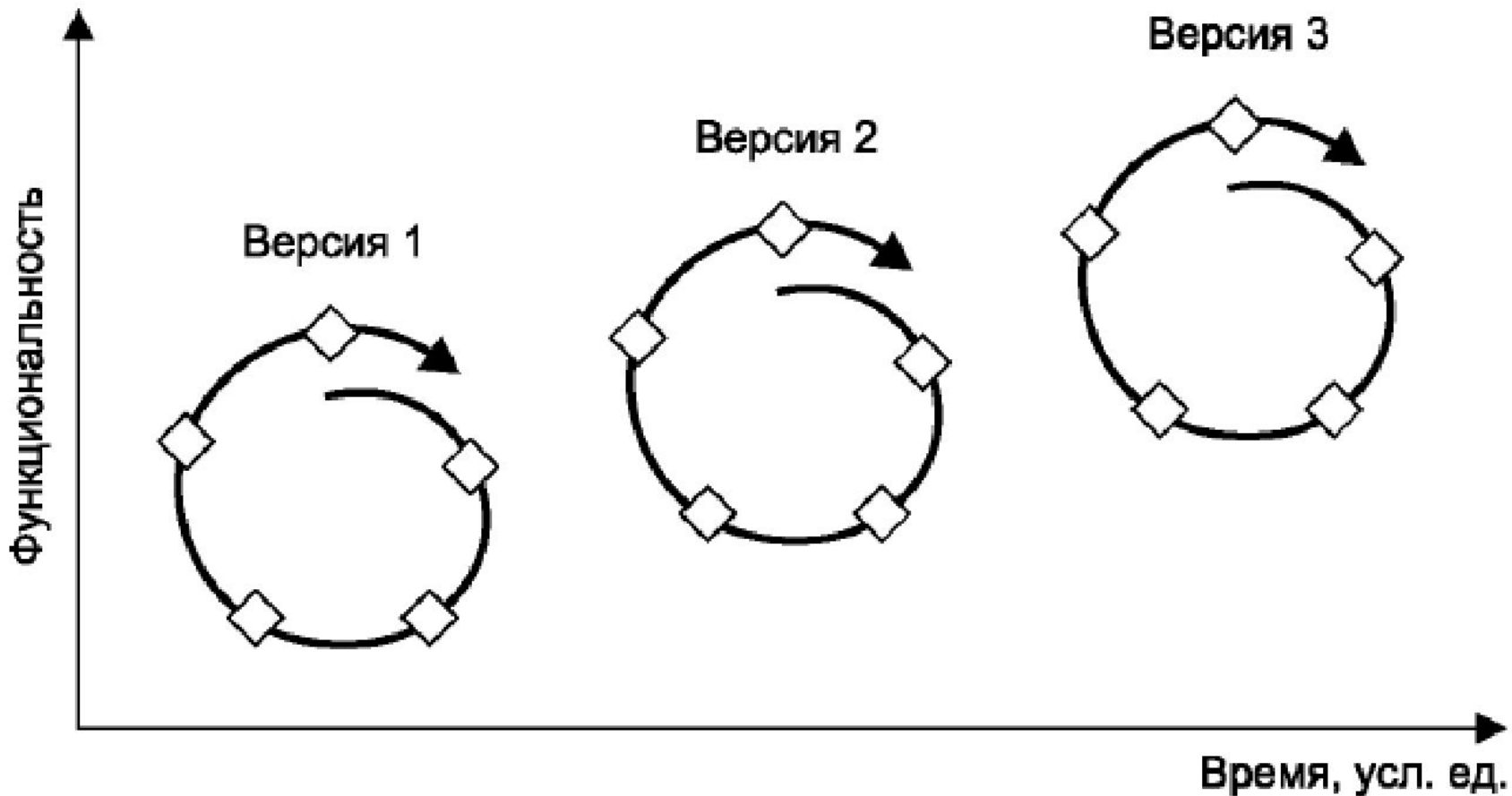


Этапы и контрольные точки модели MSF

Для достижения поставленной цели на четвертом этапе (этап стабилизации) необходима тесная взаимосвязь тестировщиков и разработчиков. Задача первых — выявление как можно большего количества ошибок, задача вторых на этом этапе — исправление найденных ошибок. Для контроля процесса стабилизации Microsoft предлагает две количественные характеристики — так называемую **точку конвергенции** и **точку достижения нуля**.

Точка конвергенции — это момент времени, когда скорость обнаружения ошибок сравнивается со скоростью их исправления.

Точка достижения нуля — это момент времени, когда имеющийся в наличии набор тестов не выявляет ни одной ошибки.



Функциональность версий

Фактически данный подход соответствует инкрементной модели ЖЦ стандарта ГОСТ Р 15288

Наряду с моделью процессов в технологии MSF предусмотрена **модель команд (MSF Team Model)**, которая применяется для организации проектных команд.

В составе **команды разработчиков** выделяют следующие роли:

- **Менеджер продукта** (product management) — отвечает за управление связями с клиентом;
- **Менеджер программы** (program management) — несет ответственность за разработку и поставку решения заказчику в полном соответствии с ограничениями проекта;
- **Разработчик** (development) — обеспечивает разработку технологического решения в соответствии со спецификациями, предоставленными менеджерами продукта и программы;

- **Тестировщик (testing)** — отвечает за выявление и устранение всех неполадок и проблем с качеством продукта и дает окончательное добро на выпуск и поставку решения;
- **Менеджер по выпуску (release management)** — отвечает за развертывание и работу продукта;
- **Специалист по удобству использования продукта (user experience)** — анализирует потребности и проблемы, возникающие у пользователей, и оценивает продукт на предмет соответствия таким потребностям.

Таблица совместимости ролей согласно стандарту MSF

	Менеджер продукта	Менеджер программы	Разработчик	Тестировщик	Менеджер по выпуску	Специалист по удобству использования
Менеджер продукта		::	::	+	-	+
Менеджер программы	::		::	-	+	-
Разработчик	::	::		::	::	::
Тестировщик	+	-	::		+	+
Менеджер по выпуску	-	+	::	+		-
Специалист по удобству использования	+	-	::	+	-	

ТЕМА 7. Методы «быстрой»
разработки программной системы

Альтернативой тяжелым методологиям в области программной инженерии в последние годы стали облегченные (Light) методологии.

Одной из популярных облегченных методологий сегодня стала методология Agile (дословный перевод этого термина — быстрый, подвижный, живой).

Общепризнанного стандарта методологии Agile не существует.

Выдержки из «Манифеста Agile», декларирующего основные принципы этой методологии:

«...Мы находим лучшие подходы к разработке проекта, непосредственно участвуя в процессе разработки и помогая другим.»

«В процессе работы мы пришли к тому, что для нас важнее:

- люди и их взаимодействие, чем процессы и средства;
- работающее ПС, чем исчерпывающая документация;
- сотрудничество с заказчиком, чем обсуждение условий контракта;
- реагирование на изменения, чем следование плану.
- Наивысшим приоритетом для нас является удовлетворенность заказчика ранними и периодическими поставками ценного для заказчика ПС.
- Мы приветствуем изменения требований даже на поздних этапах разработки. Agile-процессы готовы к таким изменениям ради достижения заказчиком конкурентного преимущества.»

Основной упор в методе Agile делается на создание сплоченной команды единомышленников-профессионалов и тесное сотрудничество с будущими пользователями.

Крайним проявлением облегченных методологий является методология экстремального программирования (XP).

ТЕМА 8. Выбор и адаптация методологии разработки

На сегодняшний день подавляющее большинство проектов ПС внедряется с нарушениями требований к качеству, срокам или сметы. Почти треть проектов информационных систем прекращают свое существование, оставшись незавершенными.

Основными факторами, влияющими на выбор методологии, являются два: **масштаб и критичность проекта.**

Масштаб проекта определяется его сложностью.

Параметры влияющие на сложность программной системы:

Параметр сложности	Комментарий
Количество функций и вариантов их использования	Количество функциональных требований и вариантов поведения системы в зависимости от внешней обстановки напрямую связано со сложностью реализации
Размерность задачи	Количество объектов в системе
Связанность объектов	Количество связей между объектами в системе
Сложность методов и алгоритмов	Сложность взаимодействий между объектами в системе
Некорректность решаемых задач	Отсутствие необходимой для решения информации
Отсутствие методов решения	Приводит к необходимости разрабатывать новые методы решения или использовать не строгие и приближенные методы
Интеллектуальный интерфейс	Глубина проработки запросов пользователя и реакции программы, степень «интеллектуальности» программы
Качество	Показатели качества по ГОСТ 9126. Чем выше требования к качеству, тем сложнее разработка
Автономность/системность	Необходимость встраивания системы в другую систему более высокого уровня может приводить к росту сложности
Доля готовых решений	Чем больше в программе можно использовать унифицированных или готовых решений, тем она проще

Следующий фактор — критичность, он определяет уровень риска (размер возможного ущерба), связанный с ошибками в системе. Часто используется шкала, предложенная американским специалистом в области программной инженерии Алистэром Коуберном, на которой выделяют **четыре уровня критичности**:

- потеря комфорта,
- потеря части денег,
- потеря «больших» (всех) денег,
- потеря жизни.

Шкала критичности ПС согласно ISO/IEC 14598

Уровень критичности	Аспекты оценки			
	безопасности	экономичности	конфиденциальности	экологичности
<i>A</i>	Множество людей погибнет	Финансовая катастрофа	Защита данных и служб стратегического назначения	Невосстановимые разрушения окружающей среды
<i>B</i>	Угроза человеческим жизням	Большие экономические потери	Защита критических данных и служб	Восстановимые разрушения окружающей среды
<i>C</i>	Угроза собственности, некоторые люди могут получить повреждения	Существенные экономические потери	Защита от возможных ошибок	Локальные загрязнения
<i>D</i>	Небольшая угроза собственности, отсутствие риска для людей	Незначительные экономические потери	Конкретные опасности не определены	Отсутствие опасности для окружающей среды

Масштаб, соотношение параметров сложности и критичность проекта определяют количество и квалификацию разработчиков, привлекаемых для реализации проекта, необходимые сроки и бюджет проекта. Эти же факторы определяют выбор методологии разработки и управления проектом. Чем выше критичность и сложность проекта, тем более «тяжелая» методология должна быть использована для его реализации.



Области эффективности различных методологий управления проектом

Стоимость вычислительной техники и другого оборудования по отношению к их характеристикам постоянно снижается, становится очевидной тенденция: стоимость фирмы-разработчика ПС все больше определяется квалификацией ее коллектива, эффективностью бизнес-процессов и репутацией фирмы, заработанной ею в предыдущих проектах.

По оценкам западных аналитиков, доля перечисленных факторов в капитализации ведущих программистских фирм достигает 80%.

Характерной особенностью программного обеспечения как товара, является его относительно низкая стоимость по отношению к выгодам или потерям, связанным с его применением.

Например, в сравнении с внедрением оборудования, удачное внедрение ПС даст значительно больший эффект на рубль инвестиций.

Контрольные вопросы

1. Почему при создании сложных программных проектов трудоемкость их разработки и риск провала проекта растут гораздо быстрее увеличения функциональности проекта?
2. Что такое жизненный цикл программного продукта, как это понятие соотносится с понятиями «Жизненный цикл разработки программы» и «Жизненный цикл товара»?
3. Чем отличаются жизненные циклы программ, разрабатываемых по конкретному заказу, и программных продуктов, предназначенных для широкой продажи на рынке?
4. Как соотносятся этапы и процессы жизненного цикла программных систем?
5. Перечислите российские и международные стандарты, регламентирующие этапы и процессы жизненного цикла программных продуктов.

Контрольные вопросы

6. Какие группы процессов жизненного цикла устанавливают стандарты ГОСТ Р ИСО/МЭК15288-2005 и ГОСТ Р ИСО/ МЭК 12207-99?
7. В чем состоит отличие основных и вспомогательных процессов жизненного цикла программных средств согласно стандарту ГОСТ Р ИСО/МЭК 12207-99?
8. Что такое модель жизненного цикла программной системы?
9. Какие преимущества и недостатки имеет каскадная модель жизненного цикла?
10. Какие преимущества дает применение спиральной и инкрементной моделей жизненного цикла? Какие требования накладывает на разработчика использование этих моделей?

Контрольные вопросы

11. Какие документы создаются в процессе разработки программной системы. На каких этапах жизненного цикла следует начинать и заканчивать разработку этих документов?

12. Перечислите наиболее известные «фирменные» методологии и инструментальные средства разработки программных систем.

13. Какую модель жизненного цикла использует методология Microsoft Solutions Framework MSF?

14. Каким этапам жизненного цикла классической каскадной модели соответствуют этапы «стабилизация» и «развертывание», используемые в модели MSF?

15. Перечислите роли, которые играют разработчики в рамках модели MSF Team Model. Какие из этих ролей могут быть совмещены?

Контрольные вопросы

16. Что такое «точка конвергенции» и «точка достижения нуля», используемые в модели MSF? По каким признакам они определяются и как используются?

17. Перечислите основные принципы, положенные в основу методологии разработки Agile. Что обеспечивает выигрыш во времени разработки при применении этой методологии?

18. Возможно ли применение быстрых методов программирования при разработке новой операционной системы уровня Windows 7? Обоснуйте свой ответ.

