

# **Введение**

**Первый кризис программирования** был отмечен в конце 60-х годов прошлого века (стоимость ПО вплотную приблизилась к стоимости компьютерной техники)

Термин **Программная инженерия** впервые был озвучен 1968 году на конференции подкомитета НАТО по науке и технике

В том же году была предложена концепция **жизненного цикла ПО**

# **Предпосылки и история развития ТПРО. Модульное программирование**

Использование при создании новых программ ранее написанных фрагментов сулило существенное снижение сроков и стоимости разработки

**Модульное программирование.** Главный принцип модульного программирования состоит в выделении таких фрагментов и оформлении их в виде модулей

Каждый модуль снабжается описанием, в котором устанавливались правила его использования – **интерфейс модуля**

# Структурное программирование

Проблема. Переход от разработки относительно простых программ к разработке **сложных программных комплексов**.

**Сложность** оценивалась следующими показателями:

- Большой объем кода (миллионы строк)
- Большое количество связей между элементами кода
- Большое количество разработчиков (сотни человек)
- Большое количество пользователей (сотни и тысячи)
- Длительное время использования

Основная часть стоимости приходится на **внедрение и эксплуатацию**.

# Принципы структурного программирования

- **Нисходящее проектирование**, при котором в системе выделяются основные функциональные подсистемы, которые потом разбиваются на подсистемы и т.д.
- **Применение специальных языков проектирования** и средств автоматизации использования этих языков (CASE-системы)
- **Согласованность процессов проектирования и разработки**: планирование и документирование проекта, поддержка соответствие кода проектной документации
- Структурное кодирование **без goto**

# Объектно-ориентированное проектирование и программирование

Проблема. Изменения требований стали вноситься не только на стадии сопровождения, но и на стадии проектирования и программирования.

Вводится понятие **класса** как развития понятия модуля.

**Принципы ООП:**

- **Инкапсуляция** – объединение в классе данных и методов
- **Наследование** – возможность вывода нового класса из старого с частичным изменением свойств и методов
- **Полиморфизм** – определение свойств и методов объекта по контексту

# Кризис программирования продолжается

	2004	2006	2008	2010	2012	2013
УСПЕШНЫЕ	29%	35%	32%	37%	39%	36%
ПРОВАЛЬНЫЕ	18%	19%	24%	21%	18%	16%
СПОРНЫЕ	53%	46%	44%	42%	43%	48%

**Успешные** – завершены в полном объёме во время и в рамках бюджета

**Спорные** – завершены не в полном объёме и/или с нарушением сроков и/или перерасходом бюджета

**Провальные** – не доведены до конца (перерасход средств, низкое качество)

# **Кризис программирования продолжается**

Основные причины неудач IT-проектов:

- Нереалистичные временные рамки проекта
- Недостаток количества исполнителей
- Размытые границы проекта
- Недостаток финансирования
- Нехватка квалифицированных кадров

# Определение

**Технология разработки программного обеспечения** – это система инженерных принципов для создания экономичного ПО с заданными характеристиками качества.

Любая технология разработки ПО базируется на некоторой **методологии**.

Под **методологией** понимается система принципов и способов организации процесса разработки программ.



# Основы разработки программного обеспечения

**ЦЕЛЬ ДИСЦИПЛИНЫ:** рассмотрение вопросов проектирования, разработки и испытаний больших программных систем с точки зрения требований разработчика

**Задачами дисциплины является изучение:**

- методов разработки программного обеспечения
- способов создания функциональных спецификаций
- методов проектирования программных комплексов
- доказательства правильности программ
- организации тестов
- сопровождения программных комплексов

# Программа и программное обеспечение

**Программа = задача + модель + алгоритм + структура данных**

**Определение 1:** Программа – это записанное на понятном некоторому вычислителю языке решение стоящей перед нами задачи.

**Определение 2:** Компьютерная программа - набор определенных команд, выполняющихся по заданному алгоритму.

**Определение 3:** Программа - данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определённого алгоритма. (ГОСТ 19781—90. ЕСПД. Термины и определения)

**Определение 4:** Программное обеспечение - совокупность программ системы обработки информации и программных документов, необходимых для их эксплуатации. (ГОСТ 19781—90. ЕСПД. Термины и определения)

# Свойства ПО

- Необходимость документирования.
- Эффективность.
- Надежность. В том числе:
  - тестирование программы при всех допустимых спецификациях входных данных
  - защита от неправильных действий пользователя
  - защита от взлома - пользователи должны иметь возможность взаимодействия с ПО только через легальные интерфейсы
- Возможность сопровождения.

# Отличие программной инженерии от других инженерий

- Компьютерная программа не материальный объект (**следствие**: стоимость разработки ПО – основной элемент его стоимости)
- Компьютерная программа – искусственный объект, который не подчиняется объективным законам природы (**следствие**: тестирование – единственный способ убедиться в работоспособности ПО)
- Программная инженерия – молодая дисциплина

# Определение жизненного цикла

**ЖЦ ИС** включает в себя следующие обобщённые понятия:

- стадии и этапы работ;
- результаты выполнения работ на каждой стадии;
- ключевые события – точки завершения работ и принятия решений.

**Стадия** – часть процесса создания ИС, ограниченная определенными временными рамками и заканчивающаяся выпуском конкретного продукта (моделей, программных компонентов, документации), определяемого заданными для данной стадии требованиями.

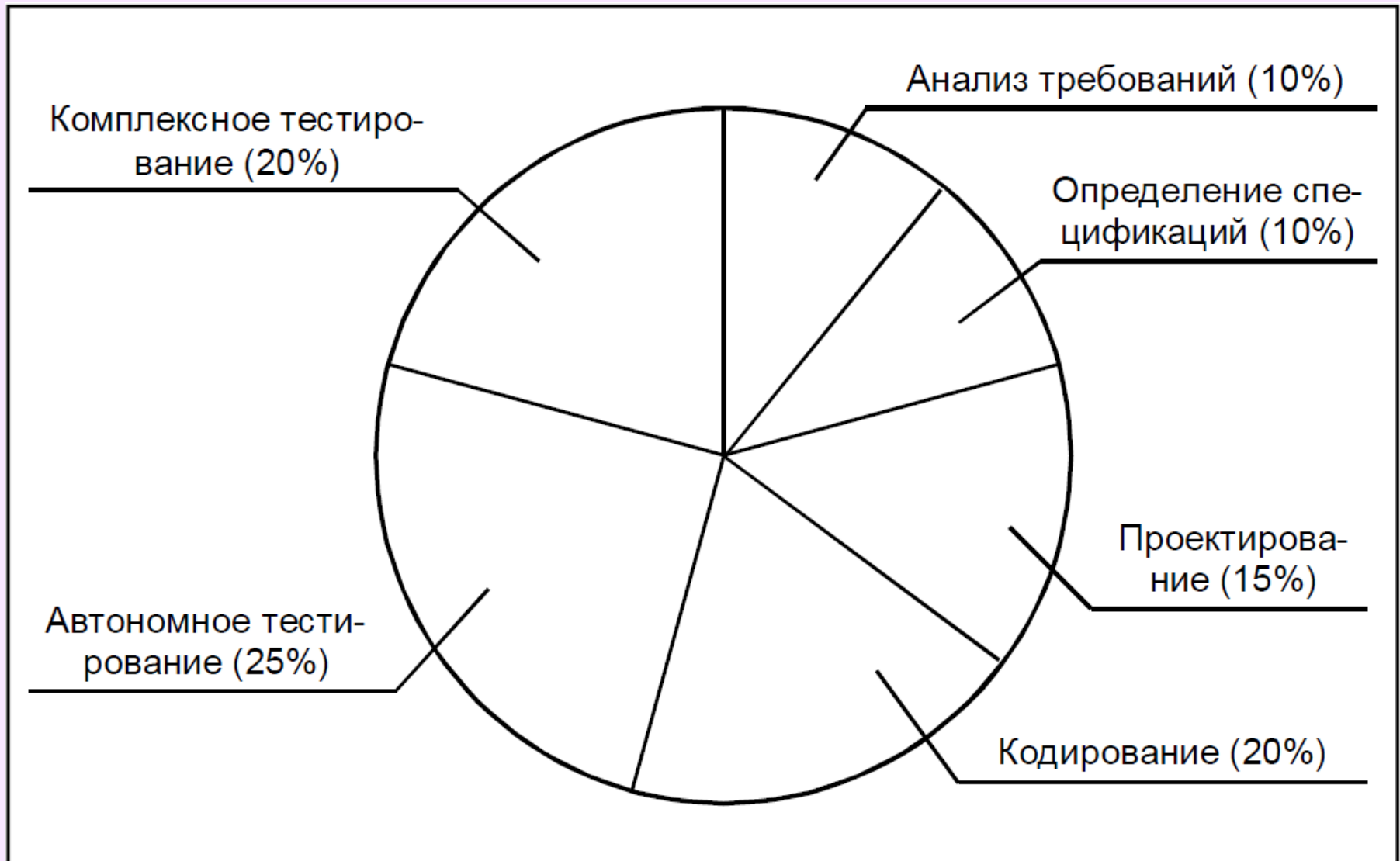
# Стадии жизненного цикла

Выделяют следующие **стадии** жизненного цикла ИС:

- **анализ, составление требований к продукту**  
(концепции) на основе анализа предметной области
- **определение спецификаций**
- **проектирование**
- **кодирование**
- **тестирование**
- **сопровождение и эксплуатация**

Завершается жизненный цикл информационной системы **выводом её из эксплуатации.**

# Распределение затрат по этапам разработки



# Анализ требований, предъявляемых к системе

На этом этапе формулируется *целевое* назначение и основные *свойства* разрабатываемой программной системы.

Базовые требования для программных систем:

- Время обработки (работы) программы;
- Стоимость обработки;
- Вероятность ошибки;
- Реакция на непредсказуемые действия оператора (защита от дурака) и др.



# Определение спецификаций

На этапе определения спецификаций:

- осуществляется точное описание функций, реализуемых ЭВМ;
- задается структура входных и выходных данных, методы и средства их размещения;
- определяются алгоритмы обработки данных.

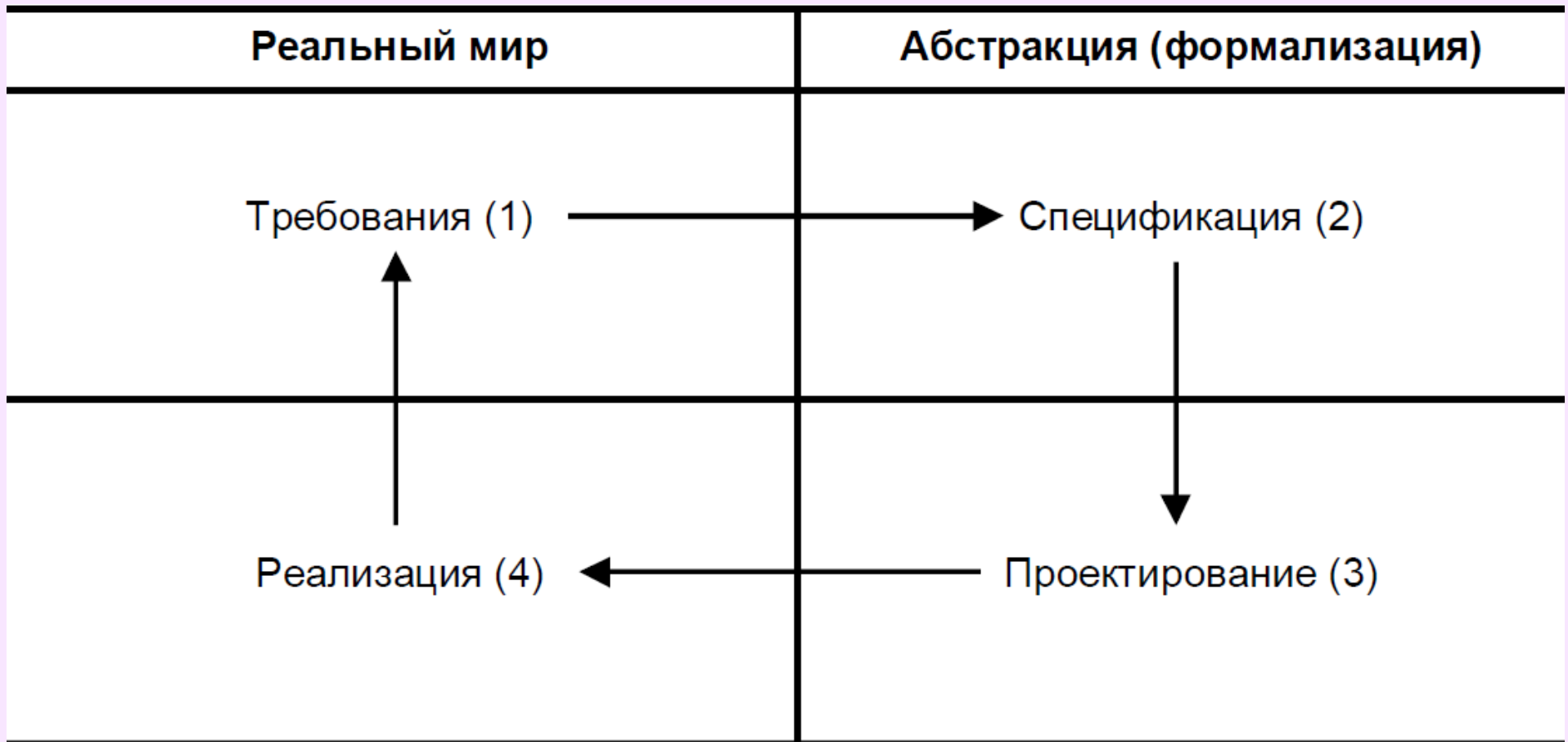
# Проектирование

На стадии проектирования разрабатываются алгоритмы, задаваемые спецификациями, и формируется общая структура вычислительной системы.

При этом система разбивается на составные части.

Для каждого определенного таким образом модуля системы должны быть сформированы предъявляемые к нему требования.

# Схема проектирования программных систем



# Стадии кодирования и тестирования

На стадии **кодирования** пишется реальный код программы, как правило, на языке высокого уровня.

На стадии **тестирования** осуществляется поиск и исправление ошибок.

Данная стадия включает в себя следующие этапы:

- кодирование и документирование каждого компонента ПО и базы данных, а также подготовку тестовых процедур и данных для их тестирования
- тестирование каждого компонента ПО и БД на соответствие предъявляемым к ним требованиям с последующим документированием результатов тестирования
- обновление документации (при необходимости)
- обновление плана интеграции ПО

# Тестирование

Стадии тестирования:

- Автономное (модуль проверяется с помощью подготовленных программистом данных).
- Комплексное (совместная проверка групп программных компонент).
- Системное (проверка системы в целом с помощью независимых тестов).

# Критерии правильности выполнения программы

- Каждый оператор должен быть выполнен, по крайней мере, один раз для заданного набора тестов.
- Каждая ветвь программы должна быть опробована.
- Каждый путь в программе должен быть испытан хотя бы один раз с использованием набора тестовых данных.
- Для каждой спецификации программы необходимо располагать набором тестовых данных, позволяющих установить, что программа правильно реализует данную спецификацию.

# **Аттестация, испытание и верификация**

**Аттестация** – общий процесс создания правильных программ с помощью процедур испытания и верификации.

**Испытание** системы осуществляется посредством тестирования. Цель такой проверки – показать, что система функционирует в соответствии с разработанными на нее спецификациями.

**Верификация** заключается в выполнении доказательств, что программа удовлетворяет своим спецификациям.

# Отклонения от нормальной работы системы

**Сбой** системы – это явление, связанное с нарушением системой установленных на нее спецификаций.

**Выброс** – данные, при обработке которых правильными алгоритмами системы происходит сбой.

**Ошибка** – это алгоритмический дефект, который создает выброс (программная ошибка).



# Правильность и надёжность

**Правильная программа** – это та, что удовлетворяет своим спецификациям.

**Надежная программа** выдает приемлемый результат даже в том случае, когда входные данные либо условия ее использования не удовлетворяют принятым допущениям.

**В правильной системе** нет ошибок, а ее внутренние данные не содержат выбросов.

Система называется **надежной**, если, несмотря на сбои, она продолжает удовлетворительно функционировать.

# Эксплуатация и сопровождение

**Сопровождение** – это внесение изменений в эксплуатируемое ПО

Цели изменений:

- исправление ошибок
- адаптация к изменениям внешней для ПО среды
- усовершенствование ПО по требованиям заказчика

# Эксплуатация и сопровождение



# Причины изменения системы в процессе эксплуатации

- Заказчик обычно не может четко сформулировать свои требования, редко бывает удовлетворен созданной системой и поэтому настаивает на внесении изменений в готовую систему.
- Могут быть обнаружены ошибки, пропущенные при тестировании.
- Могут потребоваться специальные модификации системы для частных условий функционирования, связанные с различными применениями.
- Сопровождение многочисленных компонентов системы.

# Процессы жизненного цикла

**Процесс** – набор взаимосвязанных действий, которые преобразуют входные данные в выходные

Согласно основному международному стандарту **ISO/IEC 12207**, все процессы жизненного цикла ИС делятся на три группы:

- **основные** процессы ЖЦ (приобретение, разработка, эксплуатация, сопровождение)
- **вспомогательные** процессы, обеспечивающие выполнение основных процессов
- **организационные** процессы

# Основные процессы ЖЦ

- **Заказ** – действия и задачи заказчика, приобретающего ИС
- **Поставка** – действия и задачи поставщика, предоставляющего заказчику программный продукт или услугу
- **Разработка** – действия и задачи, выполняемые разработчиком: создание программного обеспечения, создание проектной и эксплуатационной документации, подготовка тестовых и обучающих материалов и т. д.
- **Эксплуатация** – действия и задачи оператора - организации, эксплуатирующей систему
- **Сопровождение** деятельность и задачи, выполняемые сопровождающей организацией, т.е. есть службой сопровождения

# Вспомогательные процессы ЖЦ

- Документирование
- Управление конфигурацией
- Обеспечение качества
- Верификация
- Аттестация
- Совместная оценка
- Аудит
- Решение проблем

# Организационные процессы ЖЦ

**Управление** – действия и задачи, выполняемые любой стороной, которая управляет своими процессами

**Создание инфраструктуры проекта** – выбор и поддержка технологий, стандартов и инструментов, выбор и развертывание аппаратного и программного обеспечения, используемого для разработки, поддержки или сопровождения программного обеспечения

**Усовершенствование** – улучшение процессов оценки, измерения и контроля жизненного цикла

**Обучение** – начальное обучение и последующее непрерывное повышение квалификации персонала



# Обновления системы

A
B
C

I

A
B
C

II

A
B
C

III

A
B
C

I

A'
B
C

II

A
B
C

III

A''
B
C

I

A'''
B
C

II

A
B
C

III