

# **Методы и технологии испытаний информационных систем**

## **Лекция 15**

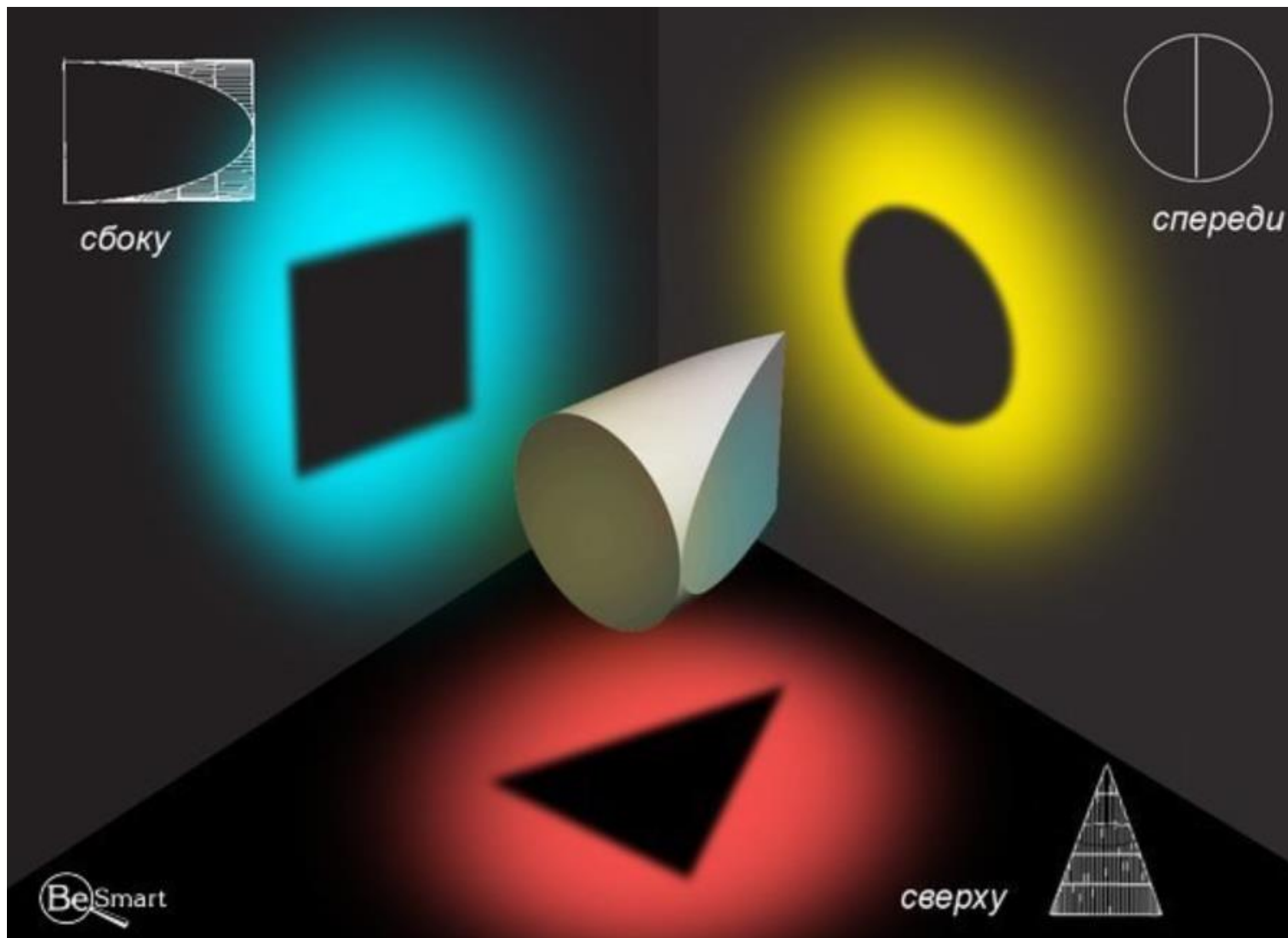
### **Тестирование программных средств**

Овчинников П.Е.

МГТУ «СТАНКИН»,

ст.преподаватель кафедры ИС

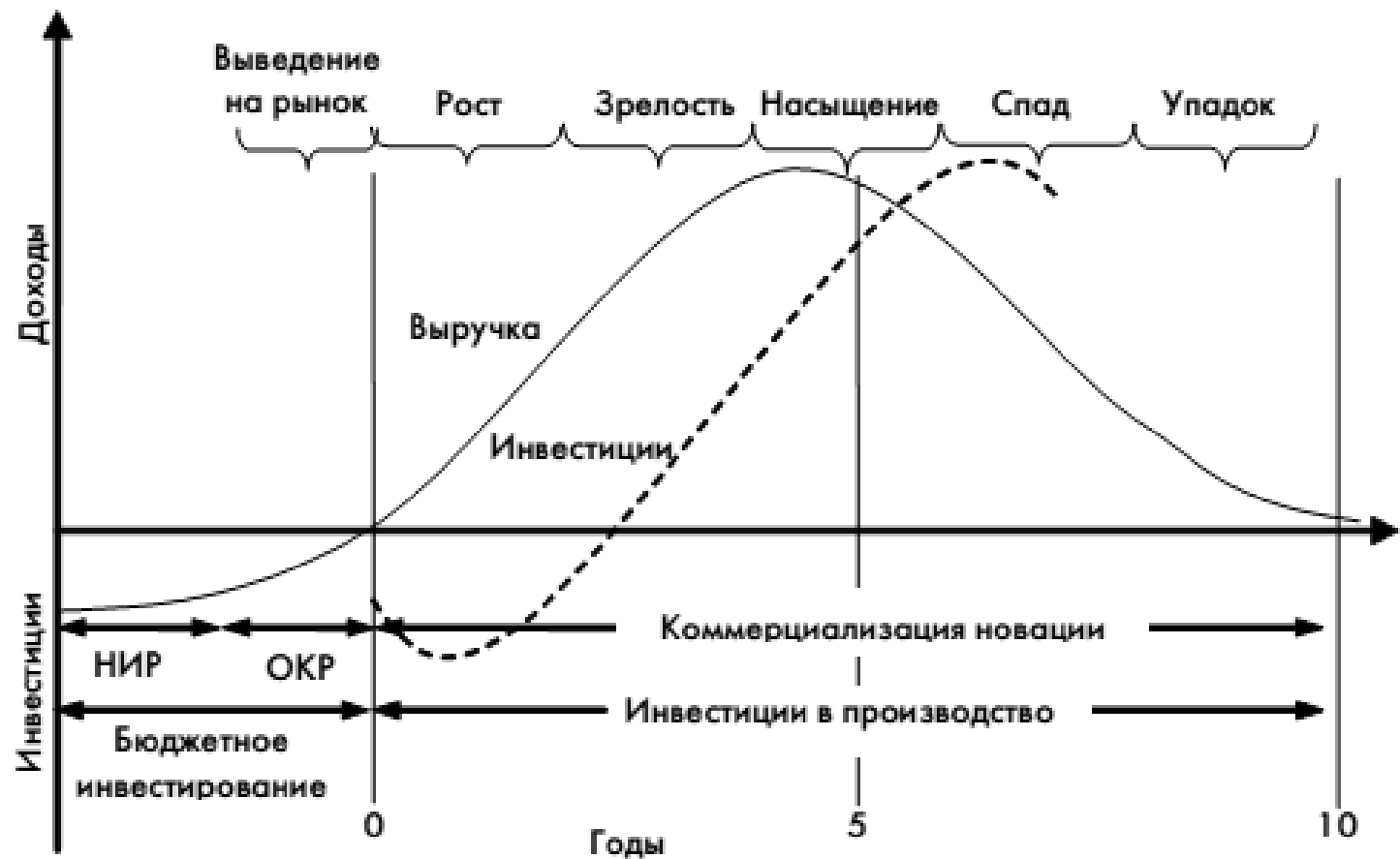
# Точка зрения определяет все



# Терминология: инновации

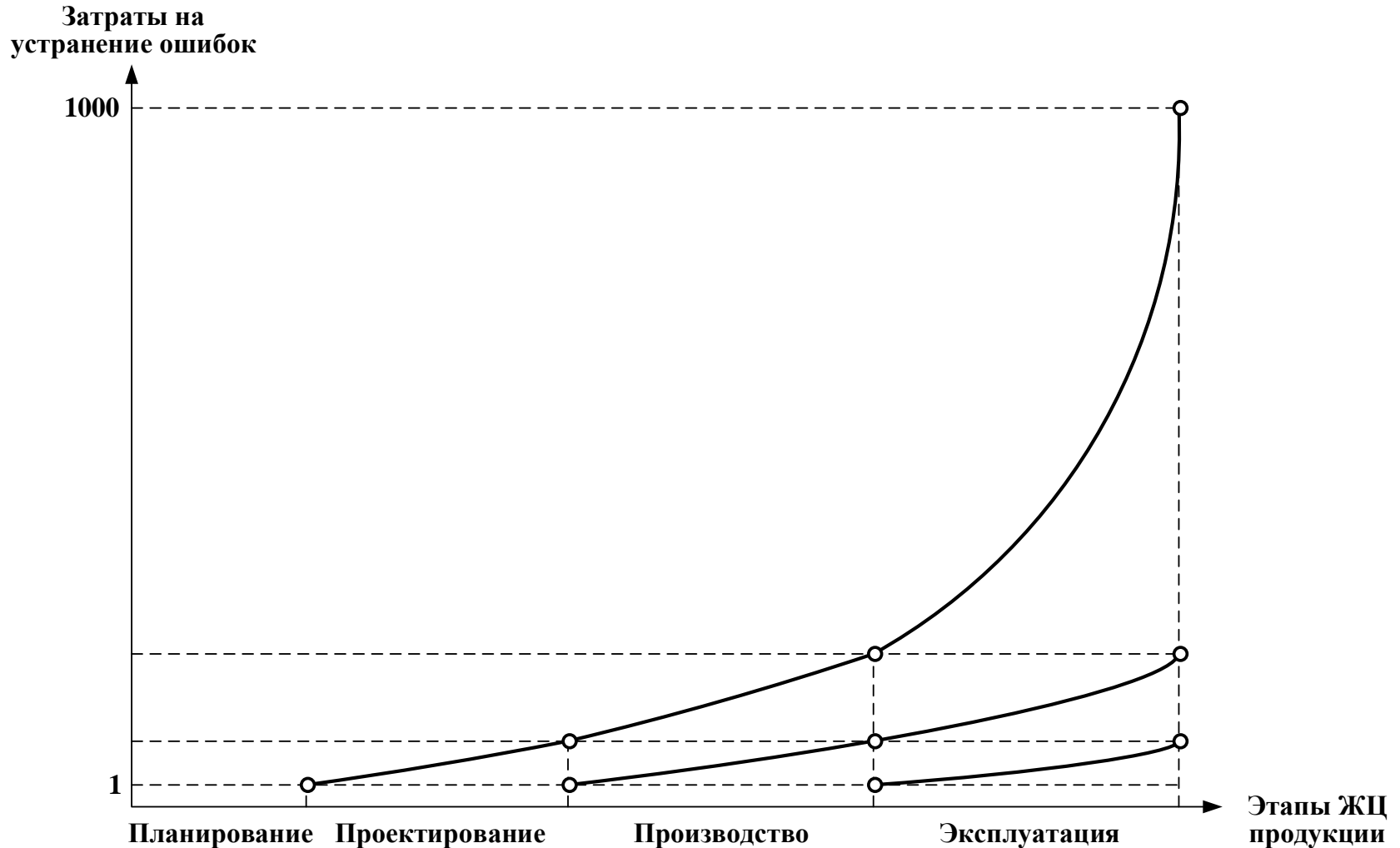
Федеральный закон "О науке и государственной научно-технической политике" от 23.08.1996 N [127-ФЗ](#)

**Инновации** - введенный в употребление **новый** или значительно **улучшенный продукт (товар, услуга)** или **процесс**, новый **метод продаж** или новый **организационный метод** в деловой практике, организации рабочих мест **или во внешних связях**.



# Терминология: ошибки проектирования

Если ошибка, допущенная на этапе **проектирования**, обнаружена на этапе **эксплуатации**, то затраты на ее устранение вырастут **в разы** по сравнению с затратами на ее устранение на этапе проектирования



# Терминология: верификация и валидация

ГОСТ Р ИСО 9000-2015 Системы менеджмента качества. Основные положения и словарь

**проектирование и разработка** (design and development)

совокупность **процессов**, преобразующих **требования** к объекту в **более детальные требования** к этому объекту.

**верификация** (verification):

**подтверждение**, посредством представления *объективных свидетельств*, того, что установленные **требования** были **выполнены**.

**валидация** (validation):

**подтверждение**, посредством представления *объективных свидетельств*, того, что **требования**, предназначенные для конкретного **использования или применения**, выполнены.

**объективное свидетельство** (objective evidence)

*данные*, подтверждающие **наличие** или **истинность** чего-либо.

Объективное свидетельство может быть получено путем наблюдения, измерения, испытания или другим способом.

**испытание** (test):

**определение соответствия** требованиям для конкретного предполагаемого **использования или применения**.

# Терминология: верификация программы

ГОСТ 19781-90 Обеспечение систем обработки информации программное.  
Термины и определения

Технология программирования и отладки программ

**Спецификация программы (Specification)**

формализованное представление **требований**, предъявляемых к программе, которые должны быть удовлетворены при ее разработке, а также **описание задачи**, условия и эффекта действия **без указания способа** его достижения

**Верификация программы (Program verification)**

доказательство того, что **поведение** программы **соответствует спецификации** на эту программу

**Поиск ошибок в программе (Error detection)**

деятельность, в результате которой **выявляются ошибки** в программе с целью их последующего **исправления**

# Терминология: испытание программы

[ГОСТ 19.004-80](#) Единая система программной документации (ЕСПД).

## Термины и определения

### Проверка программы (Program check-out)

проверка **правильности реализации** заданного **алгоритма** путем выполнения программы на вычислительной машине

### Отладка программы (Program debug)

обнаружение, локализация и устранение **ошибок** в программе вычислительной машины

### Испытание программы (Program test)

установление **соответствия** программы вычислительной машины заданным **требованиям** и программным **документам**

### Программный документ (Program document)

документ, содержащий сведения, необходимые для **разработки**, **изготовления**, **эксплуатации** и **сопровождения** программного изделия

### Сопровождение программного изделия (Program product maintenance)

**процесс модификации** существующей программы вычислительной машины, обусловленный необходимостью **устранения** выявленных в ней **ошибок** и (или) **изменения** ее функциональных **возможностей**

# Терминология: программа и методика испытаний

ГОСТ 19.301-79 ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению

Документ "Программа и методика испытаний" должен содержать следующие разделы:

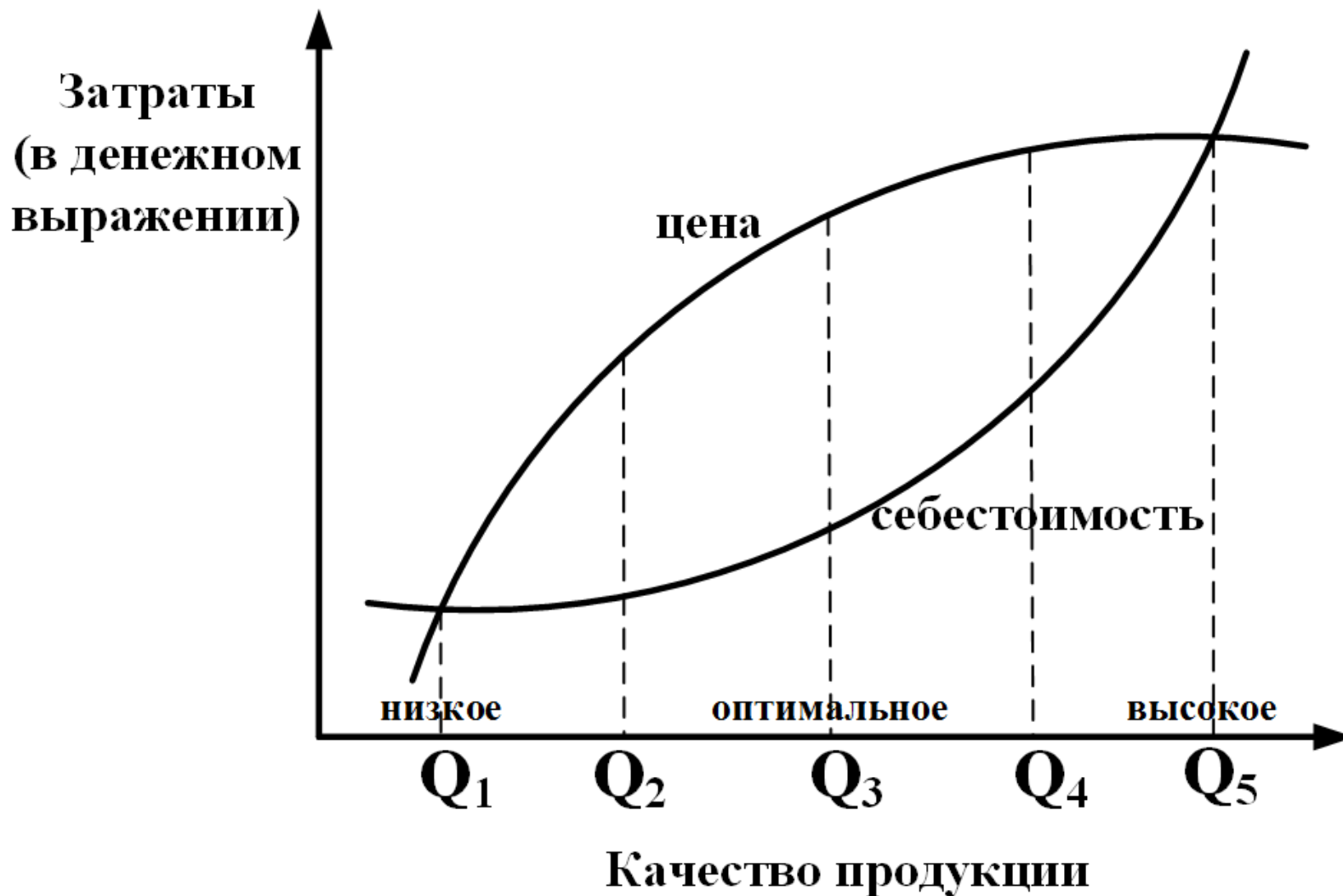
- **объект** испытаний;
- **цель** испытаний;
- требования к **программе**;
- требования к программной **документации**;
- **средства** и **порядок** испытаний;
- **методы** испытаний.

В разделе "**Методы испытаний**" должны быть приведены описания используемых методов испытаний. Методы испытаний рекомендуется по отдельным показателям располагать в последовательности, в которой эти показатели расположены в разделах "Требования к программе" и "Требования к программной документации".

В методах испытаний должны быть приведены **описания проверок** с указанием **результатов** проведения испытаний (перечней тестовых примеров, контрольных распечаток тестовых примеров и т.п.).



# Терминология: управление качеством



# Терминология: испытания АС

## ГОСТ 34.603-92 Информационная технология (ИТ). Виды испытаний автоматизированных систем

1.1. Испытания АС проводят на стадии "Ввода в действие" по ГОСТ 34.601 с целью проверки **соответствия** создаваемой АС **требованиям** технического задания

1.2. Испытания АС представляют собой процесс проверки:

- **выполнения** заданных **функций** системы
- определения и проверки **соответствия требованиям** ТЗ количественных и (или) качественных **характеристик** системы
- **выявления** и **устранения** недостатков:
  - в **действиях** системы
  - в разработанной **документации**

1.3. Для АС устанавливают следующие основные виды испытаний:

1. **предварительные**
2. **опытная эксплуатация**
3. **приемочные**

# Терминология: испытания АС

## ГОСТ 34.603-92 Информационная технология (ИТ). Виды испытаний автоматизированных систем

1.4. В зависимости от взаимосвязей испытываемых в АС объектов испытания могут быть **автономные** или **комплексные**.

- Автономные испытания охватывают **части АС**. Их проводят по мере готовности частей АС к сдаче в опытную эксплуатацию.
- Комплексные испытания проводят для **групп** взаимосвязанных **частей** АС или для АС в целом.

1.5. Для планирования проведения всех видов испытаний разрабатывают документ "**Программа и методика испытаний**". Разработчик документа устанавливается в договоре или ТЗ.

1.6. Программа и методика испытаний должны устанавливать **необходимый и достаточный объем** испытаний, обеспечивающий заданную достоверность получаемых результатов.

1.7. Программа и методика испытаний может разрабатываться на АС в целом, на части АС. В качестве приложения могут включаться **тесты (контрольные примеры)**.

# Терминология: испытания АС

ГОСТ 34.603-92 Информационная технология (ИТ). Виды испытаний автоматизированных систем

1.8. **Предварительные** испытания АС проводят для **определения** ее **работоспособности** и решения вопроса о **возможности приемки** АС в опытную эксплуатацию.

1.9. Предварительные испытания следует выполнять **после** проведения разработчиком **отладки** и **тестирования** поставляемых программных и технических средств системы и представления им соответствующих **документов** о их готовности к испытаниям, а также после ознакомления персонала АС с эксплуатационной документацией.

1.10. **Опытную эксплуатацию** АС проводят с целью **определения** фактических значений **количественных** и **качественных характеристик** АС и **готовности персонала** к работе в условиях функционирования АС, определения **фактической эффективности** АС, корректировке (при необходимости) документации.

1.11. **Приемочные** испытания АС проводят для определения **соответствия** АС **техническому заданию**, оценки качества опытной эксплуатации и решения вопроса о возможности приемки АС в **постоянную эксплуатацию**.

# Терминология: испытания АС

## ГОСТ 34.603-92 Информационная технология (ИТ). Виды испытаний автоматизированных систем

1.13. В зависимости от вида требований, предъявляемых к АС на испытаниях, **проверке** или **аттестации** в ней подвергают:

- 1) **комплекс** программных и технических средств
- 2) **персонал**;
- 3) эксплуатационную **документацию**, регламентирующую деятельность персонала при функционировании АС;
- 4) АС в целом.

2.2.2. В программе автономных испытаний указывают:

- 1) **перечень функций**, подлежащих испытаниям;
- 2) описание **взаимосвязей** объекта испытаний с другими частями АС;
- 3) **условия, порядок** и **методы** проведения испытаний и обработки результатов;
- 4) **критерии приемки** частей по результатам испытаний.

К **программе** автономных испытаний следует прилагать **график** проведения автономных испытаний.

# Терминология: испытания АС

## Нагрузочное тестирование (англ. *load testing*)

подвид тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству).

В общем случае под нагрузочным тестированием понимается практика моделирования ожидаемого использования приложения с помощью **эмуляции** работы нескольких **пользователей** одновременно.

Таким образом, подобное тестирование больше всего подходит для многопользовательских систем, чаще — использующих клиент-серверную архитектуру (например, веб-серверов). Однако и другие типы систем ПО могут быть протестированы подобным способом. Например, текстовый или графический редактор можно заставить прочесть очень большой документ; а финансовый пакет — сгенерировать отчёт на основе данных за несколько лет. Наиболее адекватно спроектированный нагрузочный тест даёт более точные результаты.

# Терминология: испытания АС

Для исследования времени отклика системы на высоких или пиковых нагрузках производится **стресс-тестирование**, при котором создаваемая на систему нагрузка превышает нормальные сценарии её использования.

Не существует чёткой границы между **нагрузочным** и **стресс-тестированием**, однако эти понятия не стоит смешивать, так как эти виды тестирования отвечают на разные бизнес-вопросы и используют различную методологию

В идеальном случае в качестве критериев успешности нагрузочного тестирования выступают требования к **производительности** системы, которые формулируются и документируются на стадии разработки функциональных требований к системе до начала программирования основных архитектурных решений. Однако часто бывает так, что такие требования не были четко сформулированы или не были сформулированы вовсе. В этом случае первое нагрузочное тестирование будет являться *пробным* ([англ. exploratory load testing](#)) и основываться на разумных **предположениях** об ожидаемой **нагрузке** и потреблении **аппаратной части** ресурсов.

Одним из оптимальных подходов в использовании нагрузочного тестирования для измерений производительности системы является тестирование на стадии ранней разработки. Нагрузочное тестирование на первых стадиях готовности архитектурного решения с целью определить его состоятельность называется 'proof-of-concept' тестированием.

# Терминология: оператор

Федеральный закон от 27.07.2006 N [149-ФЗ](#) (ред. от 08.06.2020) «Об информации, информационных технологиях и о защите информации»

**оператор** информационной системы

**гражданин** или **юридическое лицо**, осуществляющие деятельность по **эксплуатации** информационной системы, в том числе по обработке информации, содержащейся в ее **базах данных**

**сайт** в сети "Интернет"

совокупность программ для электронных вычислительных машин и иной информации, содержащейся в информационной системе, доступ к которой обеспечивается посредством информационно-телекоммуникационной сети "Интернет" по доменным именам и (или) по сетевым адресам, позволяющим идентифицировать сайты в сети "Интернет"

**страница сайта** в сети "Интернет" (далее также - интернет-страница)

часть сайта в сети "Интернет", доступ к которой осуществляется по указателю, состоящему из доменного имени и символов, определенных владельцем сайта в сети "Интернет"

**владелец сайта** в сети "Интернет"

лицо, самостоятельно и по своему усмотрению определяющее порядок использования сайта в сети "Интернет", в том числе порядок размещения информации на таком сайте



# Терминология: пользователь

**ГОСТ 34.003-90 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения**

**пользователь автоматизированной системы**

**лицо**, участвующее в функционировании АС или использующее результаты ее функционирования

**ГОСТ Р ИСО 9241-210-2016 Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем**

**пользователь (user)**

**человек**, взаимодействующий с продукцией [ИСО 9241-11:1998]

**восприятие пользователем (системы, продукции, услуги) (user experience)**

**впечатления** пользователя, возникающие **в результате** использования и/или **предстоящего использования** продукции, системы или услуги

# Терминология: пользовательские истории

**Пользовательские истории** ([англ. User Story](#)) — способ описания требований к разрабатываемой системе, сформулированных как одно или более предложений на **повседневном** или **деловом** языке **пользователя**

Пользовательские истории используются [гибкими методологиями разработки](#) программного обеспечения для **спецификации требований** вместе с [приёмочными испытаниями](#)

Каждая пользовательская история ограничена в **размере** и **сложности**. Часто история пишется на маленькой бумажной карточке. Это гарантирует, что она не станет слишком большой

В [экстремальном программировании](#) пользовательские истории **пишутся пользователями (заказчиками)** системы.

В методологии [SCRUM](#) — **пишутся** либо **одобряются** ролью **владельца продукта** ([англ. Product Owner](#))

Для заказчиков (пользователей) пользовательские истории являются основным инструментом влияния на разработку программного обеспечения

# Приемочные испытания (acceptance testing)

**Аттестация или Валидация (Validation)** —

- подтверждение на основе представления объективных свидетельств того, что элемент (элемент системы, система, документ, услуга, задача, требование и т.д.) соответствует его назначению и функциям, описанным в требованиях к нему ([SEBoK](#))
- подтверждение на основе представления объективных свидетельств того, что требования, предназначенные для конкретного использования или применения, выполнены ([ISO 9000](#))

**Пользовательская история** остается **неофициальным** определением **требований**, пока отсутствует процедура **приемочного тестирования**

**Прежде** чем **реализовывать** пользовательскую историю, клиент должен определить соответствующую **приемочную процедуру**, чтобы гарантировать, что **цели** пользовательской истории были **достигнуты**

# Терминология: пользовательские истории

Хотя пользовательские истории (User Story) и [сценарии использования](#) (Use Case) служат единой цели документирования пользовательских требований, с точки зрения взаимодействия между пользователем и системой, между ними есть различия

**Пользовательские истории** — это небольшое и удобное в работе представление информации. Они сформулированы на повседневном языке пользователя и содержат небольшие детали, таким образом оставаясь **открытыми для интерпретации**. Они помогают читателю понимать что должна делать система

**Сценарии использования**, в отличие от пользовательских историй, описывают **процесс** и его **шаги** подробно, и могут быть сформулированы с точки зрения формальной модели

**Сценарий самодостаточен**. Он обеспечивает всю необходимую информацию и детали для понимания. Сценарий описывается как «обобщенное описание ряда взаимодействий между системой и одним или более агентами, где агент — пользователь или другая система»

# Регрессионное тестирование

**Регрессионное тестирование** ([англ.](#) *regression testing*, от [лат.](#) *regressio* — движение назад) — собирательное название для всех видов [тестирования программного обеспечения](#), направленных на обнаружение ошибок в уже протестированных участках [исходного кода](#)

Такие ошибки — когда после внесения изменений в программу перестает работать то, что должно было продолжать работать — называют *регрессионными ошибками* ([англ.](#) *regression bugs*)

Регрессионное тестирование включает

- ***new bug-fix*** — проверка исправления вновь найденного дефекта,
- ***old bug-fix*** — проверка, что исправленный ранее и верифицированный дефект не воспроизводится в системе снова, а также
- ***side-effect*** — проверка того, что не нарушилась работоспособность работающей ранее функциональности.

Обычно используемые методы регрессионного тестирования включают **повторные прогоны** предыдущих тестов, а также проверки, не попали ли регрессионные ошибки в очередную версию в результате слияния кода.

# Автоматизированное тестирование

Автоматизированное тестирование программного обеспечения — часть процесса [тестирования](#) на этапе [контроля качества](#) в процессе разработки [программного обеспечения](#)

Оно использует **программные средства** для **выполнения тестов** и **проверки результатов** выполнения, что помогает сократить время тестирования и упростить его процесс

Одной из главных проблем автоматизированного тестирования является его **трудоемкость**: несмотря на то, что оно позволяет устранить часть рутинных операций и ускорить выполнение тестов, **большие ресурсы** могут тратиться на обновление самих тестов

При [рефакторинге](#) часто бывает необходимо обновить и модульные тесты, а **изменение кода тестов** может занять **столько же времени**, сколько и **изменение основного кода**

С другой стороны, при изменении интерфейса приложения необходимо заново переписать все тесты, которые связаны с обновленными окнами, что при большом количестве тестов может отнять значительные ресурсы

# Тестовое покрытие

**Критерий тестового покрытия** — это [метрика](#) для оценки качества [тестирования](#): исполнения программы с целью обнаружения ошибок

Критерий покрытия измеряет долю **классов ситуаций**, представители которых попали в тестовый набор. Чем больше уровень тестового покрытия, тем больше классов ситуаций покрыто, тем больше ошибок можно обнаружить

Источники информации о **поведении программы**:

- **исходный код** программы ([Покрытие кода](#)): для создания набора тестов используется знание внутреннего устройства программы
- **структура входных данных**: используются логические особенности входных данных
- **требования** ([Покрытие требований](#)): используется предположение о том, что ошибка в реализации требования проявляется при любой проверке этого требования
- **модели**: используется формальная модель или спецификация поведения или структуры системы. Классы тестовых ситуаций, извлечённые из модели, часто уточняют классы ситуаций, определённые на основании входных данных или требований

# Технический долг

**Технический долг** (также известный как **долг кодинга**) — это метафора программной инженерии, обозначающая **накопленные** в программном коде или архитектуре **проблемы**, связанные с пренебрежением к качеству при разработке программного обеспечения и вызывающие **дополнительные затраты** труда **в будущем**

Технический долг обычно незаметен для конечных пользователей продукта, а связан с недостатками в:

- сопровождаемости
- **тестируемости**
- понятности
- модифицируемости
- переносимости

Сам по себе плохой код не всегда является техническим долгом, так как ущерб ("проценты по долгу") появляются из-за необходимости изменения кода со временем

Термин **технический долг** используется в первую очередь по отношению к **разработке** программного обеспечения, но он также может быть применён и к другим **сферам проектирования**



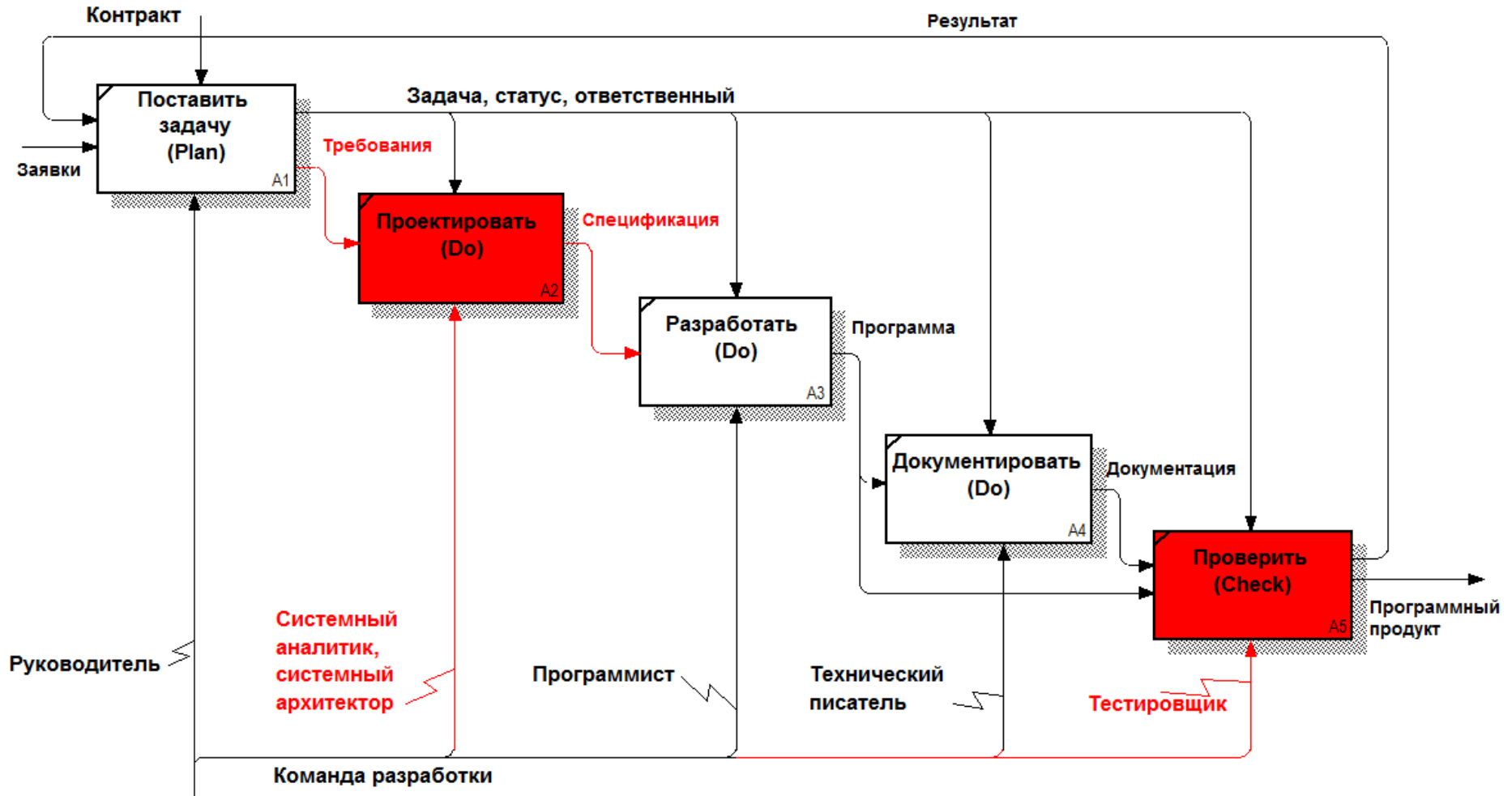
# Симуляторы и эмуляторы

**Симулятор** — **имитатор** (обычно механический или компьютерный), задача которого состоит в имитации управления каким-либо процессом, аппаратом или транспортным средством

**Эмуляция** ([англ. emulation](#)) в [вычислительной технике](#) — комплекс программных, аппаратных средств или их сочетание, предназначенное для **копирования** (или **эмулирования**) **функций** одной вычислительной системы (*гостя*) на другой, отличной от первой, вычислительной системе (*хосте*) таким образом, чтобы эмулированное поведение как можно ближе соответствовало поведению оригинальной системы (*гостя*)



# Модель TDD/BDD и команда



# ПРОФЕССИОНАЛЬНЫЙ СТАНДАРТ

## Специалист по тестированию в области информационных технологий

68

Регистрационный  
номер

### I. Общие сведения

Разработка и тестирование программного обеспечения

06.004

(наименование вида профессиональной деятельности)

Код

Основная цель вида профессиональной деятельности:

Оценка качества разрабатываемого программного обеспечения путем проверки соответствия продукта заявленным требованиям, сбора и передачи информации о несоответствиях

Группа занятий:

2131	Разработчики и аналитики компьютерных систем	2139	Специалисты по компьютерам, не вошедшие в другие группы
(код ОКЗ <sup>1</sup> )	(наименование)	(код ОКЗ)	(наименование)

Обобщенные трудовые функции			Трудовые функции	
код	наименование	уровень квалификации	наименование	код
А	Подготовка тестовых данных и выполнение тестовых процедур	4	Подготовка выполнения рабочего задания	А/01.4
			Подготовка тестовых данных в соответствии с рабочим заданием	А/02.4
			Выполнение процесса тестирования	А/03.4
			Регистрация дефектов в системе контроля (базах данных)	А/04.4
			Тестирование сопроводительной документации на соответствие требованиям заказчика	А/05.4
В	Разработка тестовых случаев, проведение тестирования и исследование результатов	5	Определение и описание тестовых случаев, включая разработку автотестов	В/01.5
			Проведение тестирования по разработанным тестовым случаям	В/02.5
			Восстановление тестов после сбоев, повлекших за собой нарушение работы системы	В/03.5
			Анализ результатов тестирования	В/04.5
			Проверка исправленных дефектов в порядке их приоритета	В/05.5
			Предоставление результатов тестирования руководителю группы (отдела) тестировщиков	В/06.5
			Деятельность по обучению младших тестировщиков	В/07.5

# ПРОФЕССИОНАЛЬНЫЙ СТАНДАРТ

## Программист

4

Регистрационный номер

### I. Общие сведения

Разработка программного обеспечения

06.001

(наименование вида профессиональной деятельности)

Код

Основная цель вида профессиональной деятельности:

Разработка, отладка, проверка работоспособности, модификация программного обеспечения

Группа занятий:

2132	Программисты	2131	Разработчики и аналитики компьютерных систем
(код ОКЗ <sup>1</sup> )	(наименование)	(код ОКЗ)	(наименование)

Обобщенные трудовые функции			Трудовые функции	
код	наименование	уровень квалификации	наименование	код
А	Разработка и отладка программного кода	3	Формализация и алгоритмизация поставленных задач	A/01.3
			Написание программного кода с использованием языков программирования, определения и манипулирования данными	A/02.3
			Оформление программного кода в соответствии с установленными требованиями	A/03.3
			Работа с системой контроля версий	A/04.3
			Проверка и отладка программного кода	A/05.3
В	Проверка работоспособности и рефакторинг кода программного обеспечения	4	Разработка процедур проверки работоспособности и измерения характеристик программного обеспечения	B/01.4
			Разработка тестовых наборов данных	B/02.4
			Проверка работоспособности программного обеспечения	B/03.4
			Рефакторинг и оптимизация программного кода	B/04.4
			Исправление дефектов, зафиксированных в базе	B/04.5

С	Интеграция программных модулей и компонент и верификация выпусков программного продукта	5	Разработка процедур интеграции программных модулей	C/01.5
			Осуществление интеграции программных модулей и компонент и верификации выпусков программного продукта	C/02.5
D	Разработка требований и проектирование программного обеспечения	6	Анализ требований к программному обеспечению	D/01.6
			Разработка технических спецификаций на программные компоненты и их взаимодействие	D/02.6
			Проектирование программного обеспечения	D/03.6