

# Лекции 7-8

## **«Научный подход в программировании»**

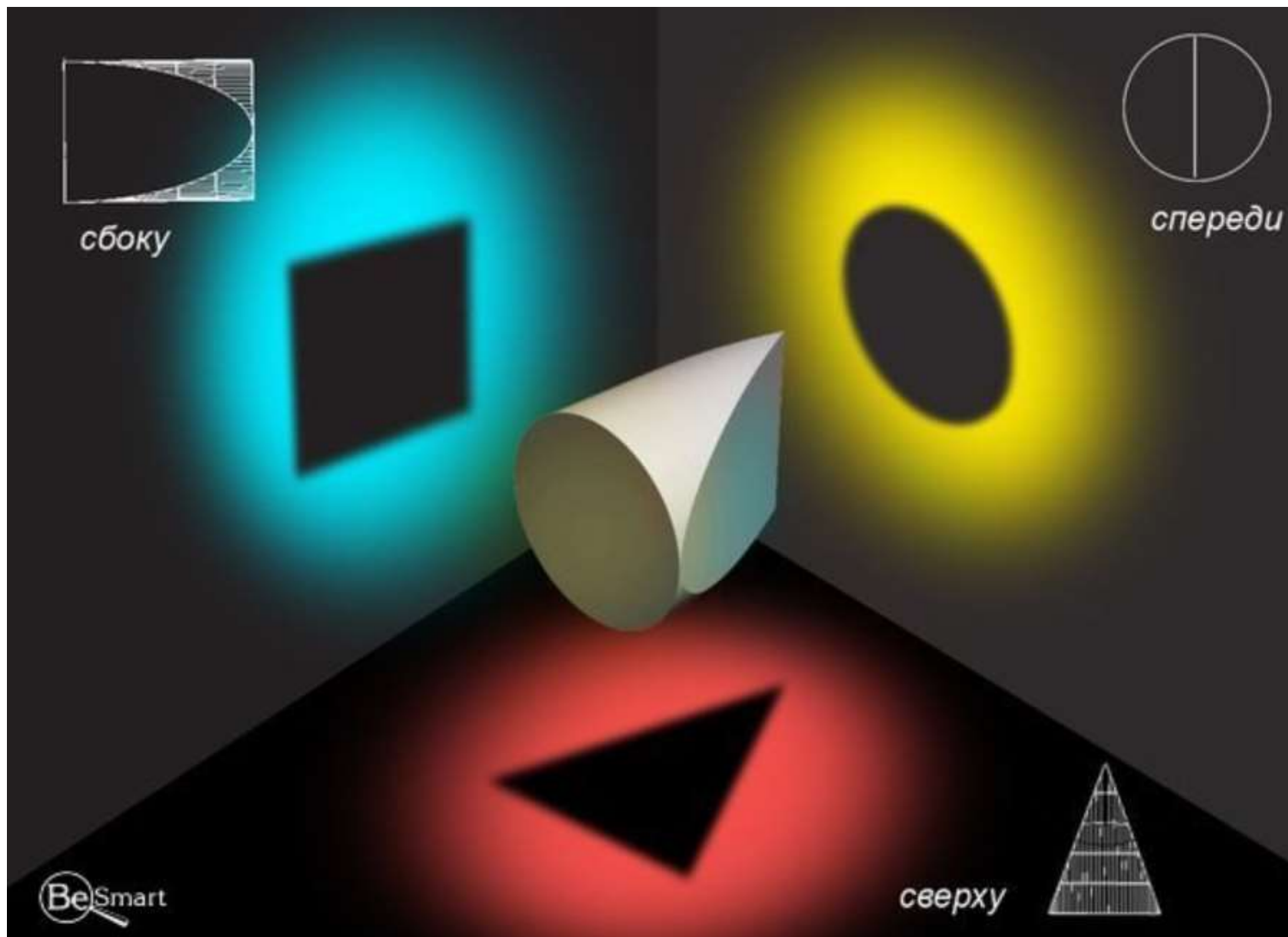
Овчинников П.Е.  
МГТУ «СТАНКИН»,  
ст.преподаватель кафедры ИС

# Лекция 7

## **«Программирование как научная деятельность»**

Овчинников П.Е.  
МГТУ «СТАНКИН»,  
ст.преподаватель кафедры ИС

# Точка зрения определяет все



# Объектно-ориентированный подход (инженерия знаний)

ГОСТ 7.0-99 СИБИД. Информационно-библиотечная деятельность, библиография. Термины и определения

**Информация - сведения**, воспринимаемые человеком и (или) специальными устройствами как отражение фактов материального или духовного мира в процессе коммуникации

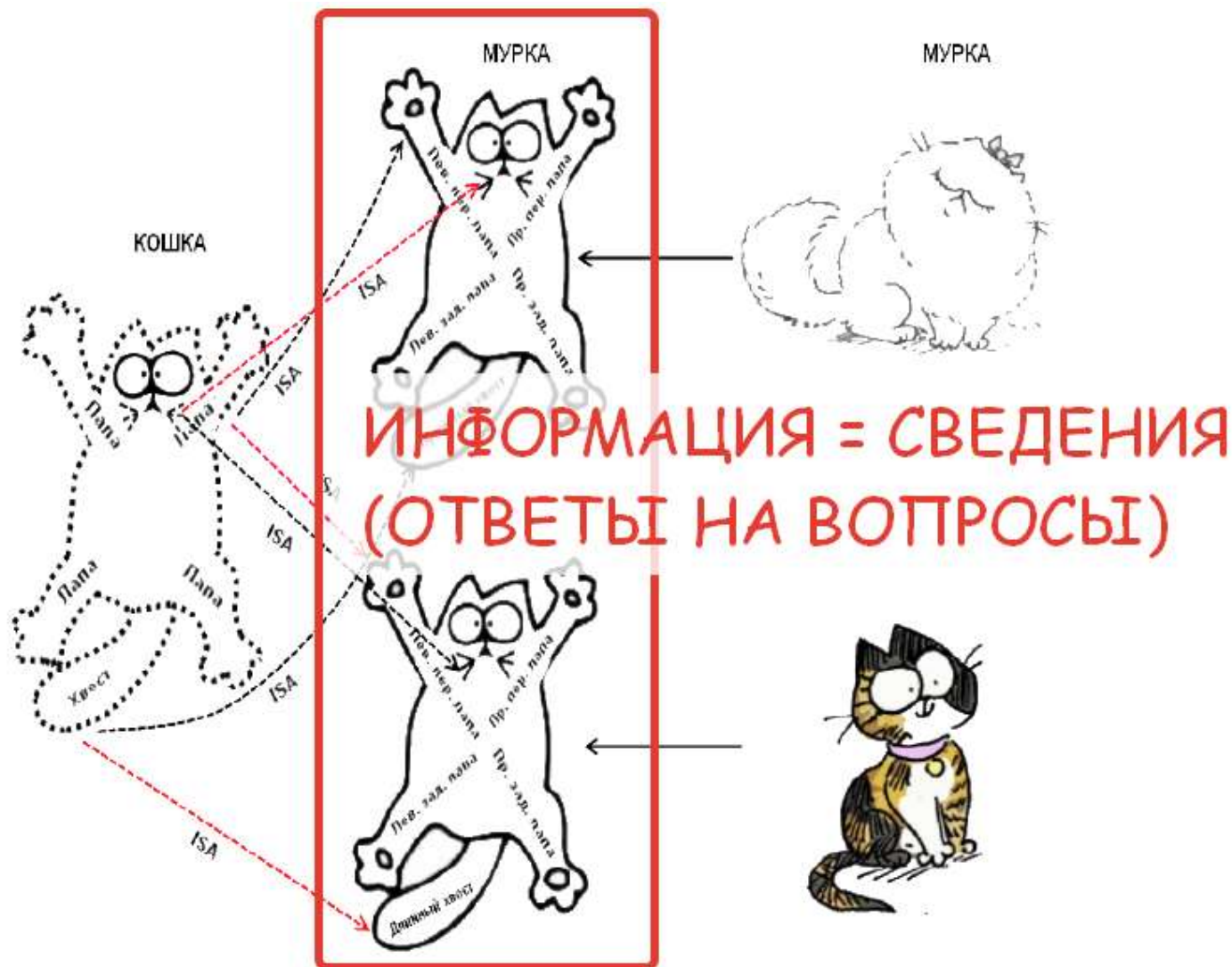
Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 21.07.2014) «Об информации, информационных технологиях и о защите информации»

**Информация - сведения (сообщения, данные)** независимо от формы их представления

ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными

**Данные - информация**, представленная в формализованном виде, пригодном для передачи, интерпретации или обработки с участием человека или автоматическими средствами

# Объектно-ориентированный подход (инженерия знаний)



# Терминология: программирование

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

**Программа (Program)**

**Данные**, предназначенные **для управления** конкретными **компонентами** системы обработки информации в целях реализации определенного алгоритма

**Программное обеспечение**

Совокупность **программ** системы обработки информации и **программных документов**, необходимых для эксплуатации этих программ

**Программирование (Programming)**

**Научная** и практическая **деятельность** по созданию программ

ГОСТ 19.004-80 Единая система программной документации (ЕСПД).  
Термины и определения

**программное изделие (program product)**

программа на носителе данных, являющаяся продуктом **промышленного производства**

# Терминология: программный продукт

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

программное **изделие** (program product) в автоматизированной системе

программное средство:

- **изготовленное**,
- прошедшее **испытания** установленного вида и
- **поставляемое** как продукция производственно-технического назначения **для применения** в АС

ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология (ИТ). Системная и программная инженерия. Процессы жизненного цикла программных средств

программный **продукт** (software product)

совокупность **компьютерных программ, процедур** и, возможно, связанных с ними **документации** и **данных**

# Объектно-ориентированный подход (инженерия знаний)

**Знания - структурированная информация**, связанная причинно-следственными отношениями и образующая системы

**Научные знания** могут быть:

- **эмпирическими** (на основе **опыта** или **наблюдения**)
- **теоретическими** (на основе анализа **абстрактных моделей**)

Научные знания в любом случае должны быть **обоснованными** на эмпирической или теоретической доказательной основе.

Теоретические знания:

- **абстракции**
- **анalogии**
- **схемы**

отображающие

- **структуру** и
- **природу**

процессов **изменения объектов**, протекающих в **предметной области**.

Эти знания **объясняют явления** и могут использоваться для **прогнозирования поведения объектов**



# Терминология: познание

Познание, когнция — совокупность

- процессов,
- процедур и
- методов

**приобретения** знаний о явлениях и закономерностях объективного мира

Познание является основным предметом гносеологии (теории познания)

Устанавливая сущность познания, его формы и принципы, теория познания стремится ответить на вопросы:

- как **возникает знание** и
- как оно **соотносится с действительностью**

**Знание** — результат процесса познавательной деятельности

обычно под знанием подразумевают только тот результат познания, который может быть логически или фактически обоснован и допускает эмпирическую или практическую проверку

# Элементы научного метода

**Теория** ([др.-греч.](#) θεωρία «рассмотрение, исследование») — **система знаний**, обладающая **предсказательной силой** в отношении какого-либо явления

Теории формулируются, разрабатываются и проверяются в соответствии с научным методом

**Гипóтеза** (от [др.-греч.](#) ὑπόθεσις — «основание», «предположение») — недоказанное утверждение, предположение или догадка.

Как правило, гипотеза высказывается на основе ряда **подтверждающих** её **наблюдений** (примеров) и поэтому выглядит правдоподобно. Гипотезу впоследствии или **доказывают**, превращая её в установленный факт (см. теорема, теория), или же **опровергают** (например, указывая контрпример), переводя в разряд ложных утверждений.

**Закóн** — вербальное и/или математически сформулированное утверждение, которое описывает соотношения, связи между различными научными понятиями, предложенное в качестве объяснения фактов и признанное на данном этапе научным сообществом согласующимся с экспериментальными данными. **Непроверенное** научное утверждение называют **гипотезой**

# Элементы научного метода

**Моделирование** — это изучение объекта посредством моделей с переносом полученных знаний на оригинал.

**Предметное** моделирование — создание моделей уменьшенных копий с определёнными свойствами, дублирующими оригинальные.

**Мысленное** моделирование — с использованием мысленных образов.

**Знаковое** или **символическое** — представляет собой использование формул, чертежей.

**Компьютерное** — компьютер является и средством, и объектом изучения, моделью является компьютерная программа.

**Эксперимент** (от [лат.](#) *experimentum* — проба, опыт) в научном методе — набор действий и наблюдений, выполняемых для **проверки** (истинности или ложности) **гипотезы** или научного исследования причинных связей между феноменами.

Эксперимент является краеугольным камнем эмпирического подхода к знанию. [Критерий Поппера](#) выдвигает в качестве главного отличия научной теории от псевдонаучной возможность постановки эксперимента, прежде всего такого, который может дать опровергающий эту теорию результат. Одно из главных требований к эксперименту — его [воспроизводимость](#).

# Терминология: защита авторских прав

**Интеллектуальная собственность** — в широком понимании термин означает закреплённое законом временное исключительное право, а также личные неимущественные права авторов на результат интеллектуальной деятельности или средства индивидуализации

Законодательство, которое определяет права на интеллектуальную собственность, устанавливает монополию авторов на определённые **формы использования** результатов своей интеллектуальной, творческой деятельности, которые, таким образом, могут **использоваться другими лицами** лишь с **разрешения первых**

**Открытый контент** (англ. *Open content*) — неологизм, придуманный по аналогии с открытым ПО (англ. *open source*), описывает любое творческое произведение или контент, опубликованный по лицензии, явно разрешающей копирование и изменение этой информации кем угодно, а не только закрытой организацией, фирмой или частным лицом. Открытый контент — это альтернативная парадигма использованию копирайта для создания монополий.

Открытый контент способствует целям демократизации знаний. Крупнейшим open content проектом является Википедия.

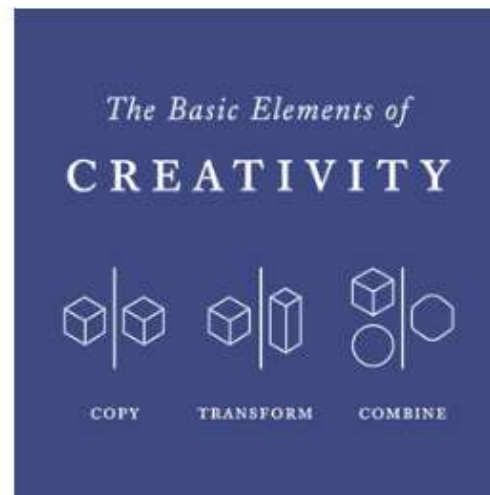
# Проблематика: генерация идей

## Источники идей

Существует множество способов генерации идей (например, метод мозгового штурма), но на практике, как правило, все проще.

Если посмотреть на появляющиеся на рынке стартапы, то можно заметить, что лишь малая часть из них является уникальной, а остальные созданы одним из трех методов, приведенных на рисунке ниже:

- **Копирование** (клонирование): как правило, копируют идею (не продукт)
- **Трансформация**: дополнение существующей идеи
- **Комбинирование**: объединение нескольких идей



# Продукт-ориентированные методологии (PDM)

PDM часто используют в уже действующих компаниях для запуска нового продукта. Что при этом происходит?

**Этап 1: концепция.** На данном этапе авторы проекта формируют и формализуют концепцию своего проекта, т.н. видение. Сначала в виде краткого описания проекта, потом в виде технико-экономического обоснования (ТЭО), после чего создают бизнес-план проекта, где отражены концепция продукта, целевая аудитория, *бизнес-модель* и стратегия продаж, финансовый прогноз денежных потоков, инвестиционная привлекательность проекта и т.д. Чаще всего, бизнес-план делается авторами проекта для того, чтобы получить инвестиции.

Большинство проектов никогда не переходят на следующую стадию, так как постоянно переписывают бизнес-план и ищут инвестора, готового вложиться на столь ранней стадии.

На практике, очень мало частных или институциональных инвесторов готовы вложить в проект, у которого нет даже **прототипа**.

# Продукт-ориентированные методологии (PDM)

PDM часто используют в уже действующих компаниях для запуска нового продукта. Что при этом происходит?

**Этап 2: разработка продукта.** На следующей стадии стартап собирает и анализирует информацию о рынке, спросе на нём, основных игроках и конкурентах. Как правило, эта *информация* берётся из интернета и печатных изданий.

В лучшем случае, проводится несколько интервью с потенциальными клиентами (т.н. фокус-группы)

**Этап 3: тестирование.** На данном этапе сначала проводится alpha-тестирование продукта, т.е. соответствие функциональности продукта заявленной спецификации, исправляются наиболее грубые недочёты и ошибки.

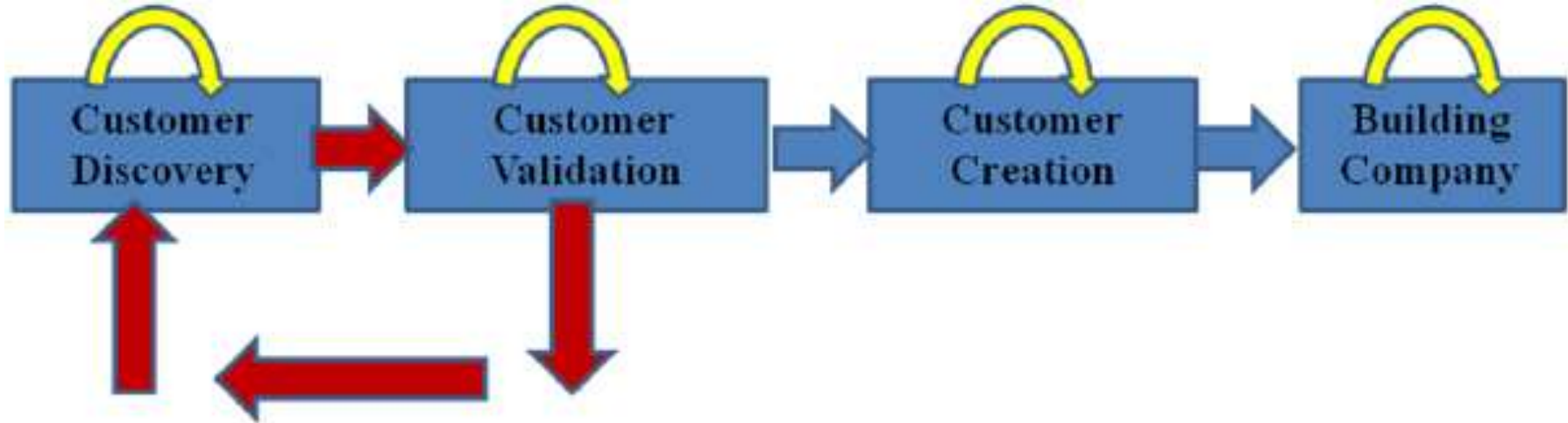
Обычно оно проходит внутри компании силами разработчиков и тестеров. После этого продукт дорабатывается и проходит уже beta-тестирование: ошибки в работе продукта ищет ограниченный круг "потенциальных пользователей".

# Продукт-ориентированные методологии (PDM)

Основная идея *Customer Development Methodology* - как можно раньше и как можно чаще проверять все свои идеи и предположения о будущем продукте на будущих клиентах.

Эта модель состоит из 4 стадий развития:

- **Customer discovery** – выявление потребителя
- **Customer validation** – верификация потребителей
- **Customer creation** – создание спроса
- **Company building** – построение компании





# Продукт-ориентированные методологии (PDM)

**Работа с гипотезами** - действия на первом этапе *CDM - Customer Discovery*.

Этот этап является итеративным и состоит из 4-х последовательных фаз или процессов.

- Фаза 1: **постановка гипотез** (state hypotheses)
- Фаза 2: **проверка гипотез** (test & qualify hypotheses)
- Фаза 3: проверка **концепции продукта** (test & qualify the product concept)
- Фаза 4: **анализ** (verify)

## **Фаза 1: постановка гипотез**

Когда компания подходит к фазе 1, на которой формулируются и записываются все первоначальные предположения (гипотезы), команде требуется написать *по* 1-2 страницы своих предположений о следующих элементах:

- **Продукт**
- **Потребитель и его проблема**
- **Канал продаж и ценовая политика**
- **Создание спроса**
- **Тип рынка**
- **Конкуренция**

# Продукт-ориентированные методологии (PDM)

## *Фаза 2: тестирование гипотез*

В этой фазе вы тестируете и квалифицируете гипотезы из фазы 1.

После получения обратной связи от потребителей гипотезы остаются в неизменном виде в очень редких случаях.

Поэтому вы будете **не только проверять** свои предположения, но и **изменять** их на основе новой полученной информации.

Помните, что вы идете к потребителям **не** для того, чтобы выяснить, **нравится ли им ваш продукт**. Ваша цель – проверка предположений, в том числе – о проблемах потребителей.

Если **предположения** в корне **ошибочны** – не имеет значения, насколько хорош сам продукт – его просто никто не купит, так он никому не нужен.

# Продукт-ориентированные методологии (PDM)

## *Фаза 3: тестирование концепции продукта*

В третьей фазе вы переходите к тестированию гипотез о продукте на потенциальных потребителях.

Цель данного этапа – не продать продукт, а получить обратную связь.

Используйте схемы и графики, чтобы формализовать и обобщить полученную Вами информацию. Сравните эти описания с вашими первоначальными гипотезами:

- **Какие проблемы** на самом деле **есть у потребителей?**
- Насколько они серьезны для них?
- Как сейчас их решают?
- **Опишите работу потребителей с вашим продуктом и без него. Насколько велика разница?**
- Готовы ли потребители за это платить?
- Что нового вы узнали о потребителях?
- Что было самым удивительным?
- Что вас разочаровало?

# Продукт-ориентированные методологии (PDM)

Если ваш продукт хотя бы отчасти решает проблемы потребителей, рассмотрите внимательнее **гипотезы** и **список свойств**.

Расположите свойства продукта в порядке **убывания их важности** для потребителей.

Помните, что цель – найти **минимум** необходимых свойств продукта, а **не** бесконечно **дополнять их список**

Шкала APDEX

№ п. /п.	Значение <u>от</u>	Значение <u>до</u>	Оценка
1.	0.00	0.50	неприемлемо
1.	0.50	0.70	очень плохо
2.	0.70	0.85	плохо
3.	0.85	0.94	хорошо
4.	0.94	1.00	отлично

APDEX рассчитывается по формуле:

$$APDEX = (NS + NT/2)/N,$$

где:

- 1) N – общее количество выполнений операции;
- 2) NS – количество выполнений с временем отклика от 0 до T;
- 3) NT – количество выполнений с временем отклика от T до 4T;
- 4) T – целевое время выполнения операции.

# Прототипирование в программировании

## Классификация прототипов

### по назначению:

- **горизонтальные** или **поведенческие** (horizontal, behavioral)
- **вертикальные** или **структурные** (vertical, structural)

### по времени жизни:

- **одноразовые** или **исследовательские** (throwaway, exploratory)
- **эволюционные**

### по способу реализации:

- бумажные (paper)
- **электронные**
- раскადровки

# Прототипирование в программировании

## Одноразовый прототип

Одноразовый или исследовательский прототип (*throwaway prototype*, *exploratory prototype*) создается, когда нужно быстро промакетировать те или иные аспекты и компоненты системы

Целям создания исследовательских прототипов служит технология *RAD* (*rapid application development*) - быстрая разработка приложений, см. ["Выявление требований"](#)

Одноразовый прототип **должен создаваться быстро**. При его разработке **не следует** уделять внимание вопросам:

- повторного использования кода,
- качества,
- быстродействия,
- технологичности и т.п.

В результате получается "сырой" код, который может содержать **значительное количество дефектов**. Необходимо принять меры к тому, чтобы фрагменты кода, реализующие такого рода прототипы, не стали частью целевой системы.

# Прототипирование в программировании



На рисунке присутствует новое, не раскрытое ранее понятие: "**карта диалога**", говорят также "схема диалога". Прежде, чем создавать одноразовый прототип, необходимо определиться:

- какие **основные экраны** будут присутствовать
- какие **окна** будут открываться
- какие **правила перехода** между ними будут поддерживаться.

Информация такого рода хорошо ложится на модель **диаграммы состояний**, см. "[Расширенный анализ требований. Моделирование](#)", где разным экранам (окнам) сопоставляются состояния, а активным элементам управления, вызывающим закрытие одних интерфейсных элементов и открытие других - переходы.

# Прототипирование в программировании

← → ↻ [https://jhupanen.github.io/#filter/\[\\*cuisine":\["английский","русский"\],"ingredients":\["поэзия"\]\]](https://jhupanen.github.io/#filter/[*cuisine)

## Домашняя художественная библиотека

### Язык

- ☒ Русский
- ☒ Английский
- ☐ Немецкий
- ☐ Французский
- ☐ Японский

### Жанр

- ☐ Фольклор
- ☐ Российская проза
- ☐ Зарубежная проза
- ☐ Афоризмы. Цитаты
- ☒ Поэзия
- ☐ Историческая литература
- ☐ Приключенческая литература
- ☐ Сентиментальная литература
- ☐ Детектив
- ☐ Фантастика



### Стихотворения. Поэмы

Язык: русский

Жанр: Поэзия Автор: Сергей Есенин

[Подробнее](#)



### Сонеты

Язык: русский

Жанр: Поэзия

Автор: Уильям Шекспир

[Подробнее](#)



# Прототипирование в программировании

← → ↺ <https://stankin.github.io/oop-2018/?db=idb-14-12-db.json#> ☆ 4 P ⋮

🔍

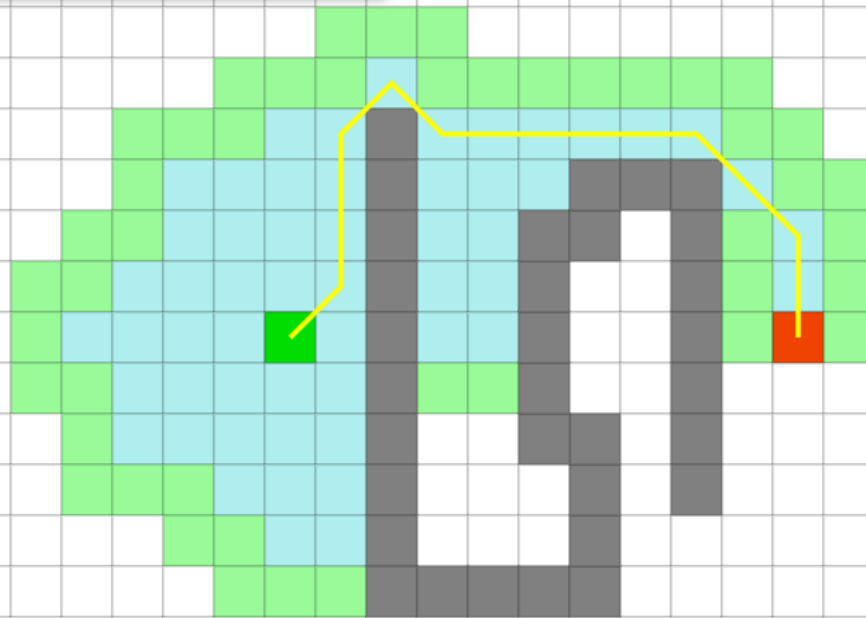
Ф.И.О. ▼	Module #1	Module #2	
[blurred]	[blurred]	[blurred]	⋮
[blurred]	[blurred]		⋮
[blurred]	[blurred]	[blurred]	⋮
[blurred]	[blurred]	[blurred]	⋮
[blurred]	[blurred]	[blurred]	⋮
[blurred]	[blurred]	[blurred]	⋮

# Прототипирование в программировании

## Instructions

hide

Click within the white grid and drag your mouse to draw obstacles.  
Drag the **green** node to set the start position.  
Drag the **red** node to set the end position.  
Choose an algorithm from the right-hand panel.  
Click Start Search in the lower-right corner to start the animation.



length: 17.07  
time: 14.7500ms  
operations: 157

# Исследовательские испытания

ГОСТ 24026-80 Исследовательские испытания. Планирование эксперимента. Термины и определения

## Планирование эксперимента

Выбор плана эксперимента, удовлетворяющего заданным требованиям

## План эксперимента

Совокупность данных, определяющих:

- **число**,
- **условия** и
- **порядок** реализации опытов

## Опыт

**Воспроизведение** исследуемого явления **в определенных условиях** проведения эксперимента при **возможности регистрации** его результатов

## Фактор

Переменная величина, по предположению **влияющая на результаты** эксперимента

# Лекция 8

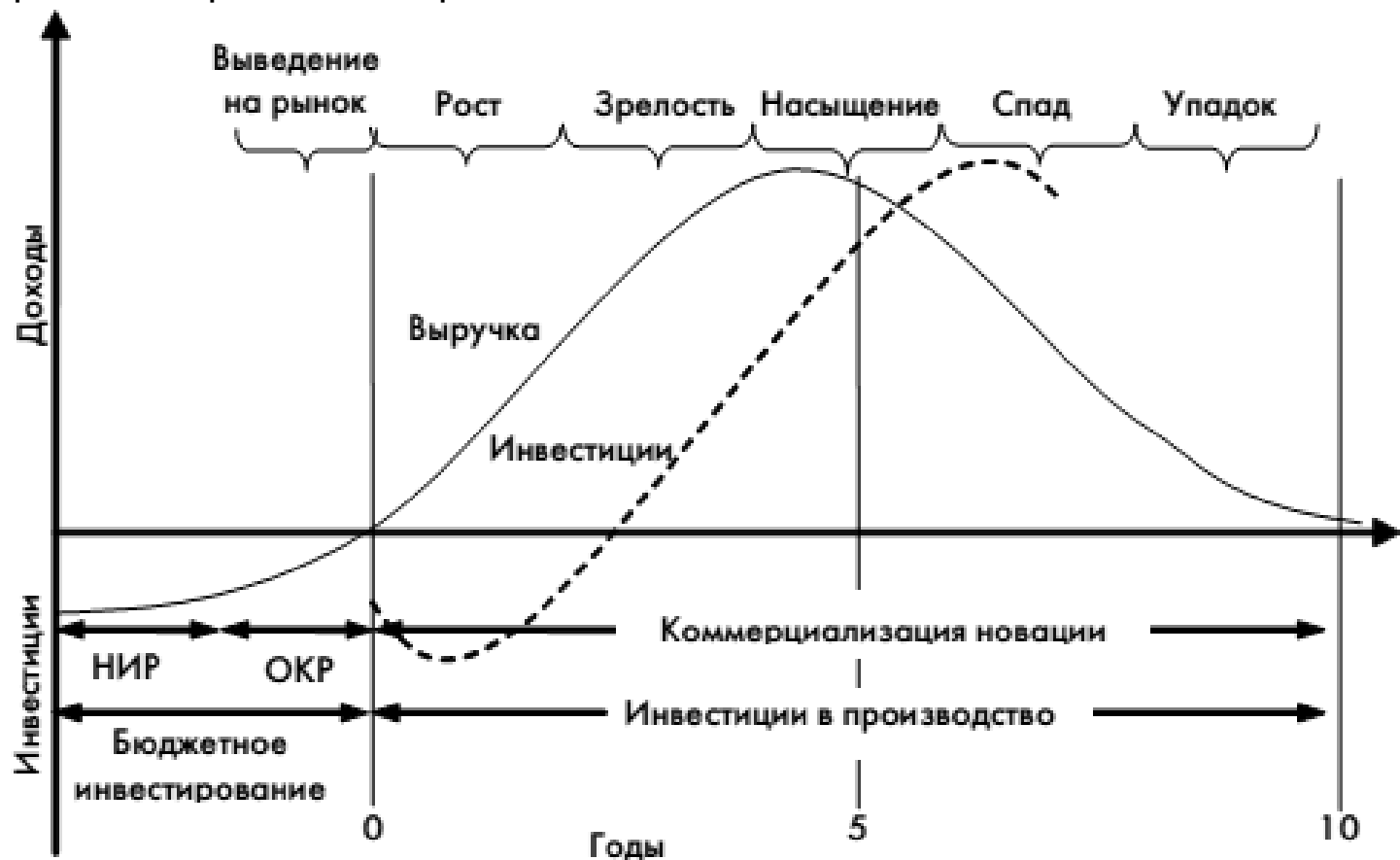
## **«Тестирование программных средств»**

Овчинников П.Е.  
МГТУ «СТАНКИН»,  
ст.преподаватель кафедры ИС

# Терминология: инновации

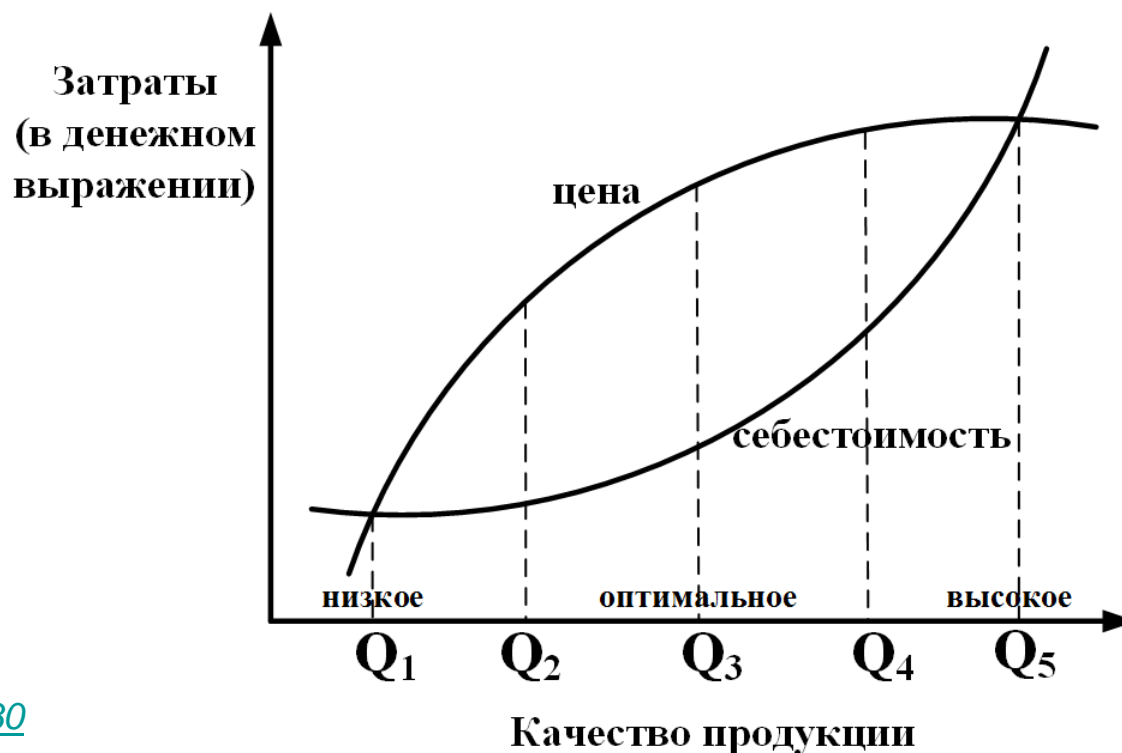
Федеральный закон "О науке и государственной научно-технической политике"  
от 23.08.1996 N 127-ФЗ

**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: испытание программы

8. Проверка программы <i>Program check-out</i>	Проверка правильности реализации заданного алгоритма путем выполнения программы на вычислительной машине
9. Отладка программы <i>Program debug</i>	Обнаружение, локализация и устранение ошибок в программе вычислительной машины
10. Испытание программы <i>Program test</i>	Установление соответствия программы вычислительной машины заданным требованиям и программным документам



# 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: оператор

Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 29.07.2017) «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: пользователь

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

Примечание 1 - **Восприятие пользователем** системы включает все

- эмоции,
  - убеждения,
  - предпочтения,
  - ощущения,
  - физические и
  - психологические реакции пользователя,
  - поведение и
  - достижения,
- которые возникают
- до,
  - во время и
  - после
- использования.

# Терминология: пользователь

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

Примечание 2 - Восприятие пользователем системы формирует сочетание образа торговой марки, способа представления, функциональности и производительности системы, ее основных и вспомогательных возможностей, а также **физического и психологического состояния** пользователя, являющегося результатом:

- предшествующего **опыта**,
- **привычек**,
- **навыков** и
- **индивидуальности**, а также
- **условий** использования

Примечание 3 - Пригодность использования, если ее интерпретируют с точки зрения **личных целей пользователей**, может включать аспекты восприятия и эмоциональные аспекты, обычно связанные с опытом пользователя.

Для оценки аспектов восприятия пользователем системы могут быть использованы критерии **пригодности использования**.

# 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

# Терминология: пользовательские истории

Пользовательская история – это легковесный инструмент для документации пользовательских требований к разработке продукта. История описывает функциональность системы с точки зрения пользователя с определенной ролью и целью этой системы.

**Как <роль/персона юзера> я <что-то хочу получить> <с такой-то целью>.**

# 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

**Эпик**—это **большая история**

Это требование, слишком большое чтобы реализовать его за один спринт.

Эпики нужно разбивать на более мелкие работы (истории).



# 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».



# BDD

In [software engineering](#), **behavior-driven development (BDD)** is a [software development process](#) that emerged from [test-driven development](#) (TDD).

Behavior-driven development combines the general techniques and principles of TDD with ideas from [domain-driven design](#) and [object-oriented analysis and design](#) to provide software development and management teams with shared tools and a shared process to collaborate on software development.

Although BDD is principally an idea about how software development should be managed by both **business interests** and **technical insight**, the practice of BDD does assume the use of specialized software tools to support the development process

Although these tools are often developed specifically for use in BDD projects, they can be seen as specialized forms of the tooling that supports test-driven development. The tools serve to add automation to the [ubiquitous language](#) that is a central theme of BDD.

# DDD

**Предметно-ориентированное проектирование** (реже проблемно-ориентированное, [англ. Domain-driven design, DDD](#)) — это набор принципов и схем, направленных на создание оптимальных систем объектов

Сводится к созданию программных абстракций, которые называются моделями [предметных областей](#). В эти модели входит [бизнес-логика](#), устанавливающая связь между реальными условиями области применения продукта и кодом

Предметно-ориентированное проектирование не является какой-либо конкретной технологией или методологией. DDD — это набор правил, которые позволяют принимать правильные проектные решения

## Тестирование

Неотъемлемой частью разработки является [тестирование](#). Оно позволяет быть уверенным в работоспособности разрабатываемого продукта на всех этапах разработки. Практичнее производить разработку посредством **автотестов** по технике [TDD](#)

# DDD

**Предметно-ориентированное проектирование** (реже проблемно-ориентированное, [англ. Domain-driven design, DDD](#)) — это набор принципов и схем, направленных на создание оптимальных систем объектов

Сводится к созданию программных абстракций, которые называются моделями [предметных областей](#). В эти модели входит [бизнес-логика](#), устанавливающая связь между реальными условиями области применения продукта и кодом

Предметно-ориентированное проектирование не является какой-либо конкретной технологией или методологией. DDD — это набор правил, которые позволяют принимать правильные проектные решения

## Тестирование

Неотъемлемой частью разработки является [тестирование](#). Оно позволяет быть уверенным в работоспособности разрабатываемого продукта на всех этапах разработки. Практичнее производить разработку посредством **автотестов** по технике [TDD](#)

# Регрессионное тестирование

**Регресси́онное тести́рование** ([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 ([Покрывтие кода](#)): для создания набора тестов используется знание внутреннего устройства программы
- **Структура входных данных**: используются логические особенности входных данных
- **Требования** ([Покрывтие 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

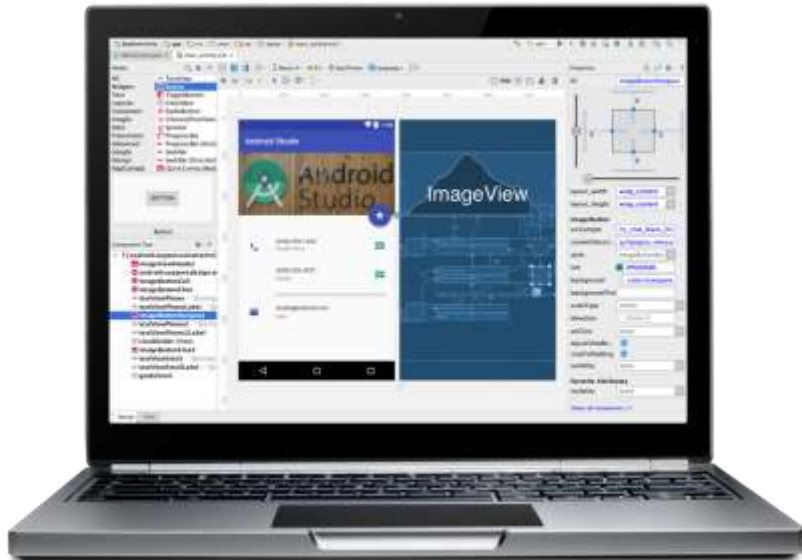
Сам по себе плохой код не всегда является техническим долгом, так как ущерб ("проценты по долгу") появляются из-за необходимости изменения кода со временем

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

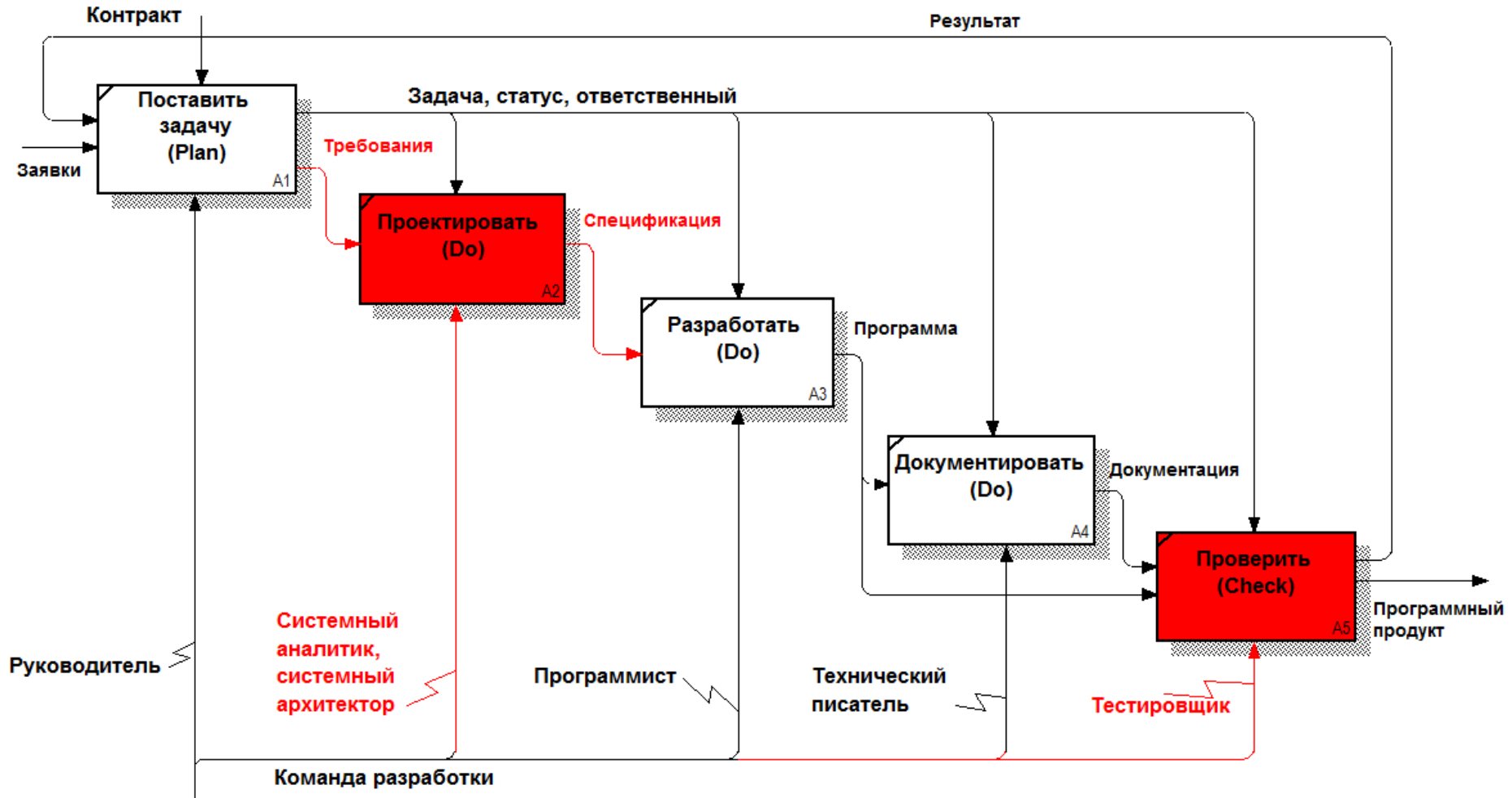
**Симулятор** — **имитатор** (обычно механический или компьютерный), задача которого состоит в имитации управления каким-либо процессом, аппаратом или транспортным средством

**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