



ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР  
ПРОГРАММИРОВАНИЯ И ВЫСОКИХ  
ТЕХНОЛОГИЙ

Функциональное тестирование ПО

# ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ



ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

## В ЭТОМ РАЗДЕЛЕ:



- [УРОВНИ ТЕСТИРОВАНИЯ](#)
- [ВИДЫ И НАПРАВЛЕНИЯ ТЕСТИРОВАНИЯ](#)
- [МЕТОДЫ ТЕСТИРОВАНИЯ](#)



# УРОВНИ ТЕСТИРОВАНИЯ

(по степени детализации приложения)



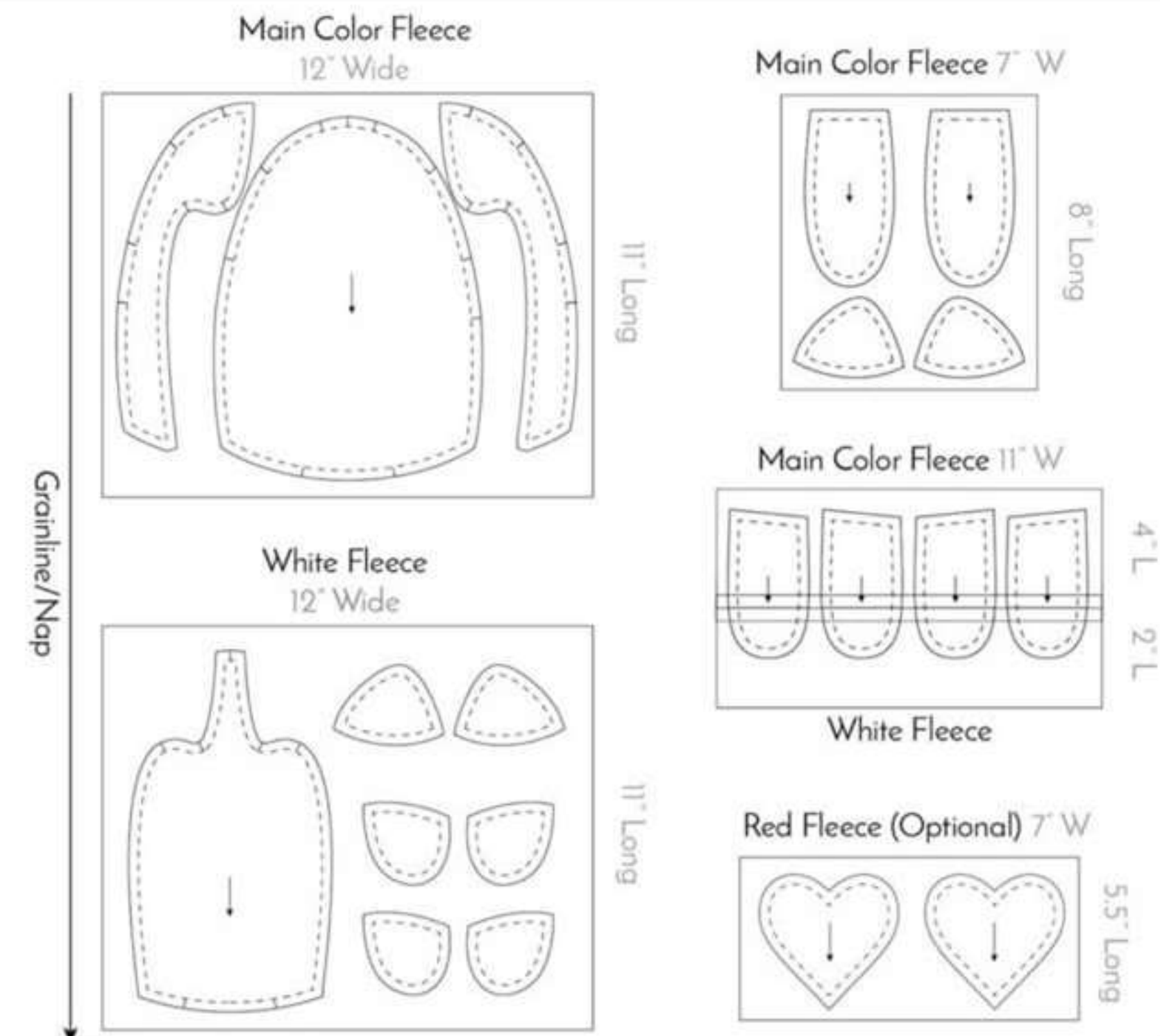
- ① КОМПОНЕНТНОЕ ТЕСТИРОВАНИЕ  
(component testing, unit testing) –  
тестирование отдельного модуля
- ② ИНТЕГРАЦИОННОЕ ТЕСТИРОВАНИЕ  
(integration testing) – проверка  
взаимодействия модулей
- ③ СИСТЕМНОЕ ТЕСТИРОВАНИЕ  
(system testing) – полная проверка  
приложения



# КОМПОНЕНТНОЕ ТЕСТИРОВАНИЕ

- **КОМПОНЕНТНОЕ ТЕСТИРОВАНИЕ**  
(Component Testing/Unit testing/  
Module testing ) - тестирование  
отдельных компонентов  
программного обеспечения
- **КОМПОНЕНТ**  
(Component) - наименьший  
элемент программного  
обеспечения, который может быть  
протестирован отдельно

Мягкая игрушка Кот своими руками  
(выкройка)







# КОМПОНЕНТНОЕ ТЕСТИРОВАНИЕ



Компоненты для тестирования:

- отдельный метод/функция в коде программы
- отдельный компонент в программе (модуль логина, аккаунт пользователя)
- отдельная программа из сложной интеграционной системы



# ИНТЕГРАЦИОННОЕ ТЕСТИРОВАНИЕ

## ➤ ИНТЕГРАЦИОННОЕ ТЕСТИРОВАНИЕ

(Integration Testing) - тестирование части системы, состоящей из двух и более частей.

Основная задача – поиск дефектов, связанных с ошибками в реализации и интерпретации взаимодействия между модулями.

Так же, как и модульное тестирование, оперирует интерфейсами модулей и подсистем и требует создания тестового окружения.







ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

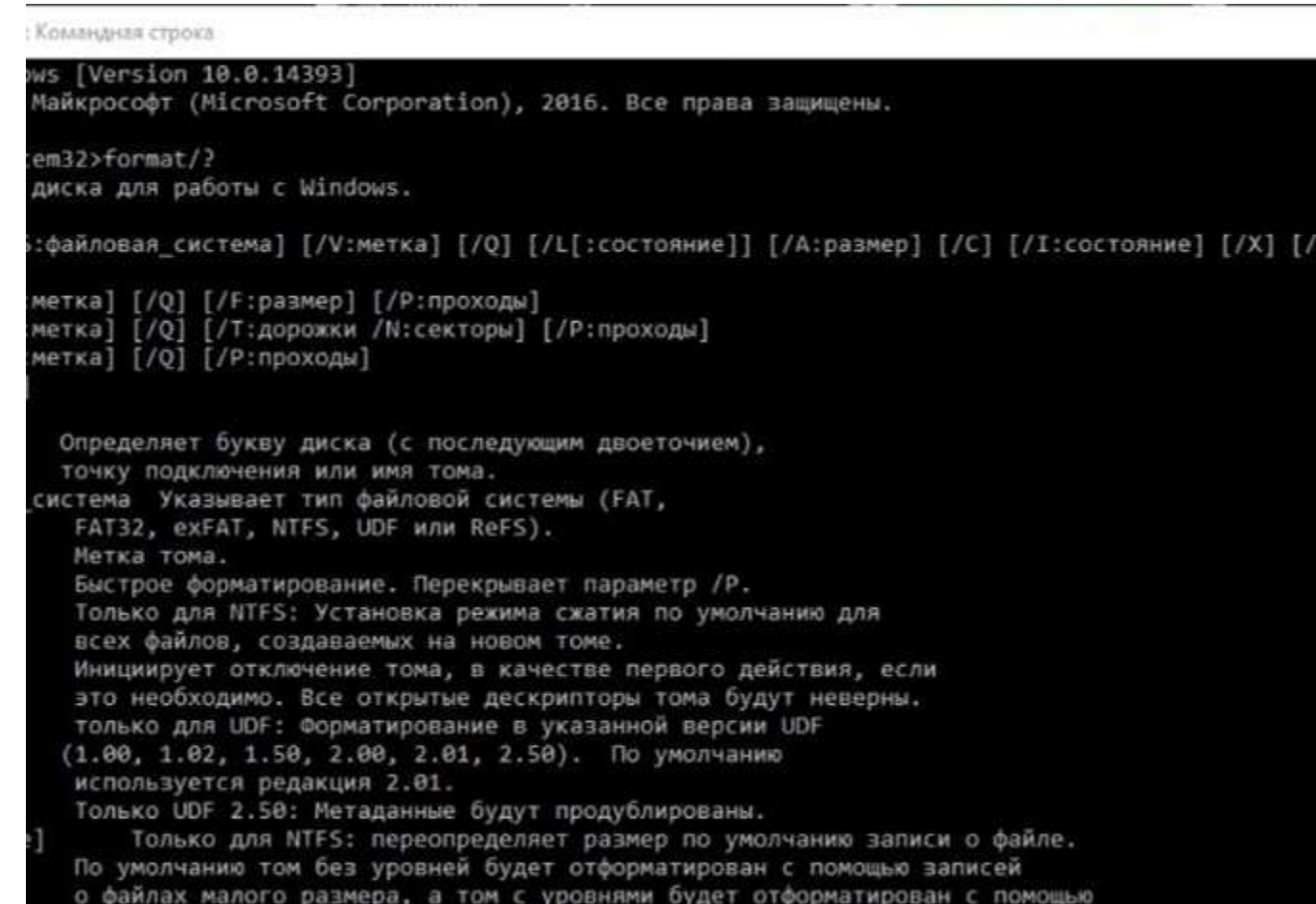
# ИНТЕГРАЦИОННОЕ ТЕСТИРОВАНИЕ

**Интерфейс** (interface) — совокупность возможностей, способов и методов взаимодействия двух информационных систем, устройств или программ.



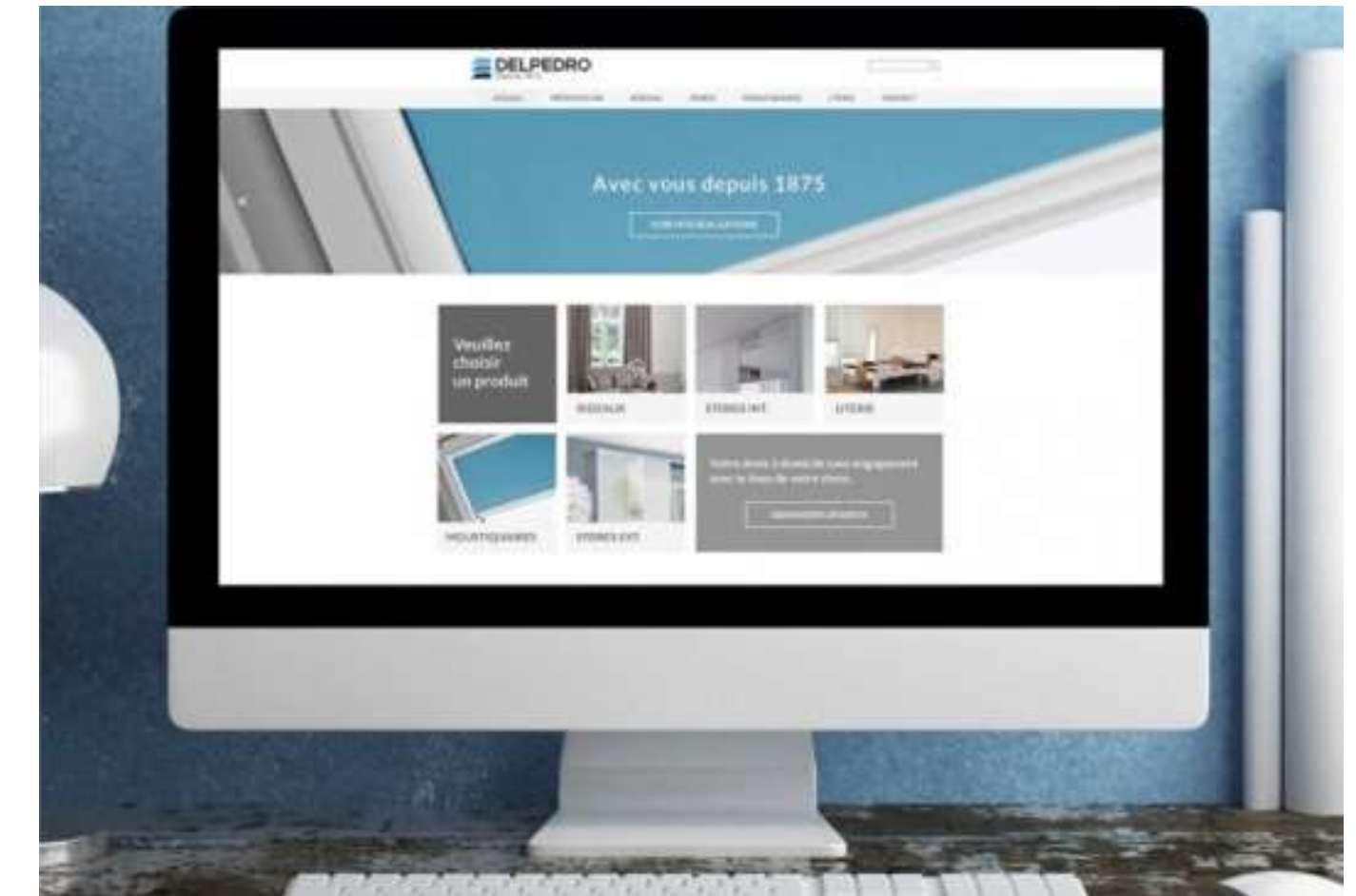
## Интерфейс программирования приложений (API)

набор методов, которые можно использовать для доступа к функциональности другой программы.



## Интерфейс командной строки (CLI)

инструкции компьютеру даются путём ввода с клавиатуры текстовых строк (команд)



## Графический интерфейс пользователя (GUI)

программные функции представляются графическими элементами экрана



# ИНТЕГРАЦИОННОЕ ТЕСТИРОВАНИЕ

Driver -> SUT -> Stub

**Заглушки** (stab) и **драйверы** (driver):

- Используются для эмуляции недостающих компонентов:
  - внешние компоненты/системы (регистрация из соц. сетей без подключения к ним)
  - подсистемы/неготовые модули (регистрация без БД)
- Может понадобиться специально написать их для тестируемой системы (**SUT**).





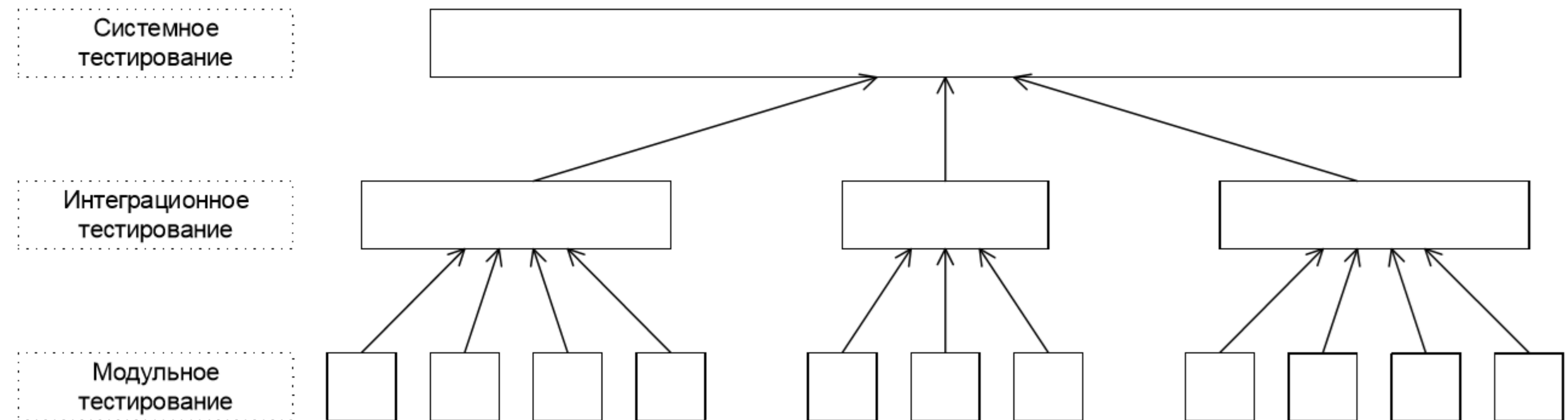
# ИНТЕГРАЦИОННОЕ ТЕСТИРОВАНИЕ

**Тестирование интеграции компонентов** (component integration testing):  
тестирование, взаимодействия между несколькими интегрированными компонентами **одного приложения**.

Bottom Up Integration

Top Down Integration

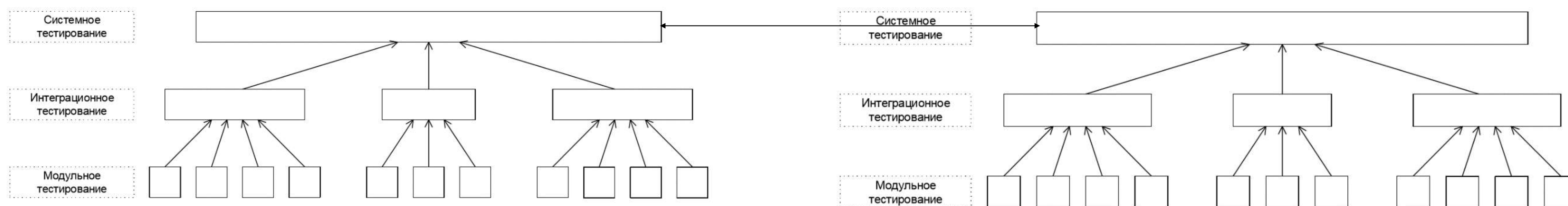
"Big Bang" Integration





# СИСТЕМНО - ИНТЕГРАЦИОННОЕ ТЕСТИРОВАНИЕ

**Системное интеграционное тестирование** (system integration testing): тестирование, взаимодействия между всеми компонентами системы, между различными приложениями объединенными в систему, **интерфейсами** связи с внешними системами (интернет и т.д.).





# СИСТЕМНОЕ ТЕСТИРОВАНИЕ

- **СИСТЕМНОЕ ТЕСТИРОВАНИЕ**  
(System Testing) - процесс тестирования системы в целом с целью проверки того, что она соответствует установленным требованиям
- **ТЕСТИРОВАНИЕ ПОЛНОЙ СИСТЕМЫ**
  - может быть последним шагом в интеграционном тестировании в узком смысле
  - может быть первый раз, когда из компонентов появляется рабочая система

В идеале – проводится независимой тестовой командой.

Мягкая игрушка Кот своими руками







# ПРИЕМОЧНОЕ ТЕСТИРОВАНИЕ

## Приёмочное тестирование

(acceptance testing):

тестирование по отношению к потребностям и требованиям пользователя, проводимое с целью дать возможность пользователям, заказчикам определить, принимать систему или нет.

Как правило заключительный этап тестирования, осуществляемый перед передачей продукта заказчику и/или конечным пользователям.

Типичные **формы** приемочного тестирования:

- **пользовательское** приемочное тестирование (UAT)
- **эксплуатационное** приемочное тестирование (OAT)
- альфа- и бета- тестирование



# ПРИЕМОЧНОЕ ТЕСТИРОВАНИЕ

## Пользовательское приемочное тестирование

(User acceptance testing):

- тестирование конечного продукта **проводят пользователи**
- может проходить как на оборудовании производителя или пользователей

## Эксплуатационное приемочное тестирование

(Operational acceptance testing):

- тестирование резервного копирования/восстановление
- аварийное восстановление
- задачи **технической поддержки**
- периодические проверки уязвимостей безопасности



# ПРИЕМОЧНОЕ ТЕСТИРОВАНИЕ

## Альфа-тестирование

(alpha testing):

тестирование потенциальными пользователями/заказчиками или независимой командой тестирования **на стороне разработчиков**, но вне разрабатывающей организации.

## Бета-тестирование (beta testing):

тестирование потенциальными и/или существующими клиентами/заказчиками **на внешней стороне никак не связанными с разработчиками.**







ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# УРОВНИ ТЕСТИРОВАНИЯ

## Резюме

- Существует много разных подходов к **компонентному** тестированию.
- Интеграционное тестирование проверяет как компоненты и/или системы **взаимодействуют** друг с другом.
- Системное тестирование позволяет проверить **систему в целом** согласно спецификаций, требований и бизнес процессов.
- Участие **пользователя** в приемочном тестировании является критически важным.

A man with a beard and mustache, wearing a dark bow tie and a light-colored shirt, is smiling and gesturing with both hands raised. He is in a workshop or office environment with computer monitors and papers visible in the background. The entire image has a blue color overlay.

Есть вопросы?  
Давайте обсудим!





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# ВИДЫ ТЕСТИРОВАНИЯ

## SOFTWARE TESTING TYPES



Исторически так сложилось, что как минимум «тип тестирования» (testing type) и «вид тестирования» (testing kind) давно стали синонимами.

С. Куликов





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# ВИДЫ ТЕСТИРОВАНИЯ



Виды/направления тестирования  
по целям и задачам



# ФУНКЦИОНАЛЬНОЕ И НЕФУНКЦИОНАЛЬНОЕ ТЕСТИРОВАНИЕ



## ФУНКЦИОНАЛЬНОЕ

(functional testing): тестирование, основанное на анализе функциональности компонента или системы.

Функции – это то, **ЧТО ДЕЛАЕТ** система;



## НЕФУНКЦИОНАЛЬНОЕ

(non-functional testing): тестирование атрибутов компонента или системы, не относящихся к функциональности, то есть: **надежность, внешний вид, практичность, устойчивость..**

Проверяет **КАК** система **РАБОТАЕТ**.





# ИНСТАЛЛЯЦИОННОЕ ТЕСТИРОВАНИЕ



**Инсталляционное тестирование** (installation testing, installability testing) — тестирование, направленное на выявление дефектов, влияющих на протекание стадии инсталляции (установки) приложения.





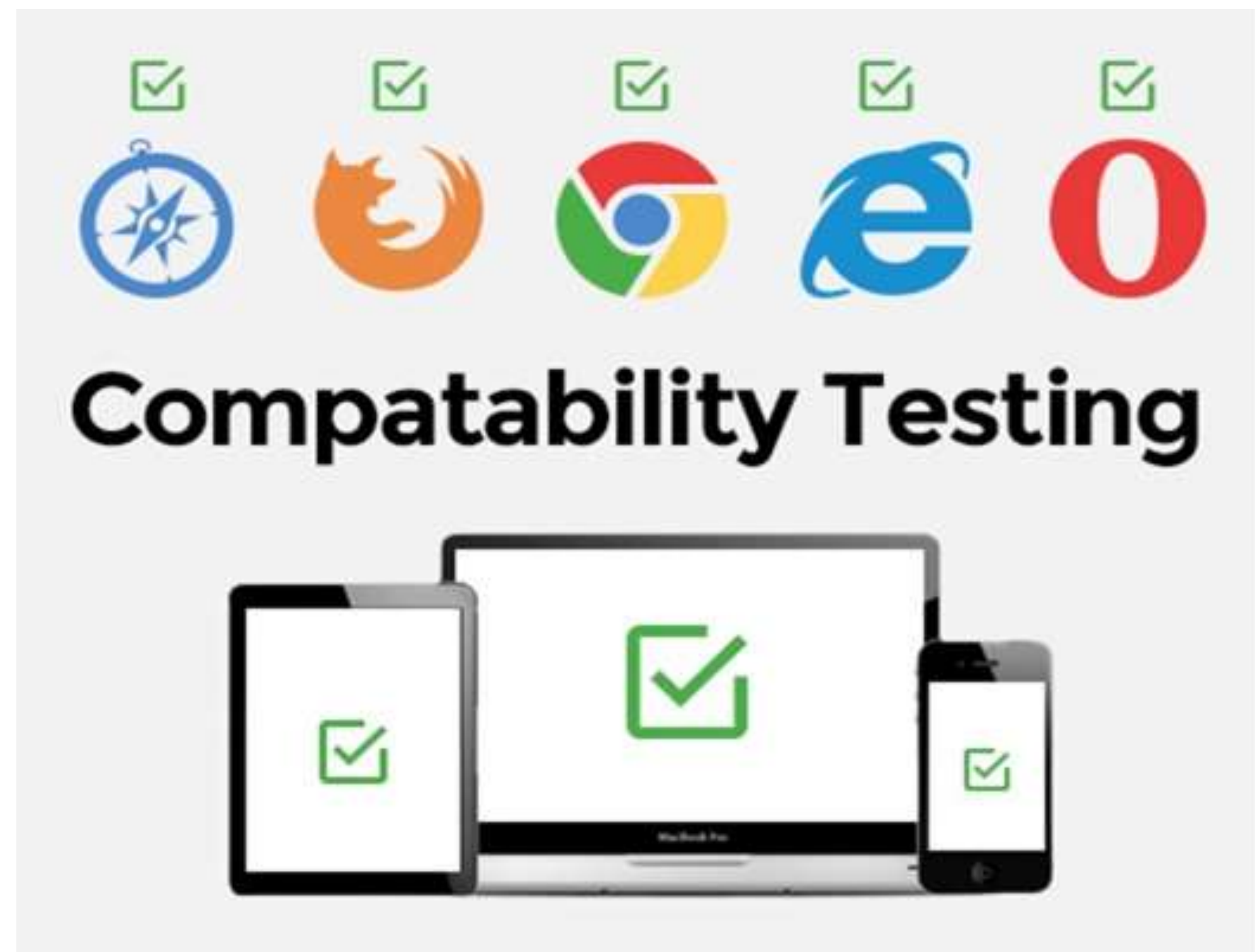
# КОНФИГУРАЦИОННОЕ ТЕСТИРОВАНИЕ



**Конфигурационное тестирование** (Configuration Testing) — вид тестирования, направленный на проверку работы программного обеспечения при различных конфигурациях системы (заявленных платформах, поддерживаемых драйверах, при различных конфигурациях компьютеров и т.д.)



# ТЕСТИРОВАНИЕ СОВМЕСТИМОСТИ



## Тестирование совместимости

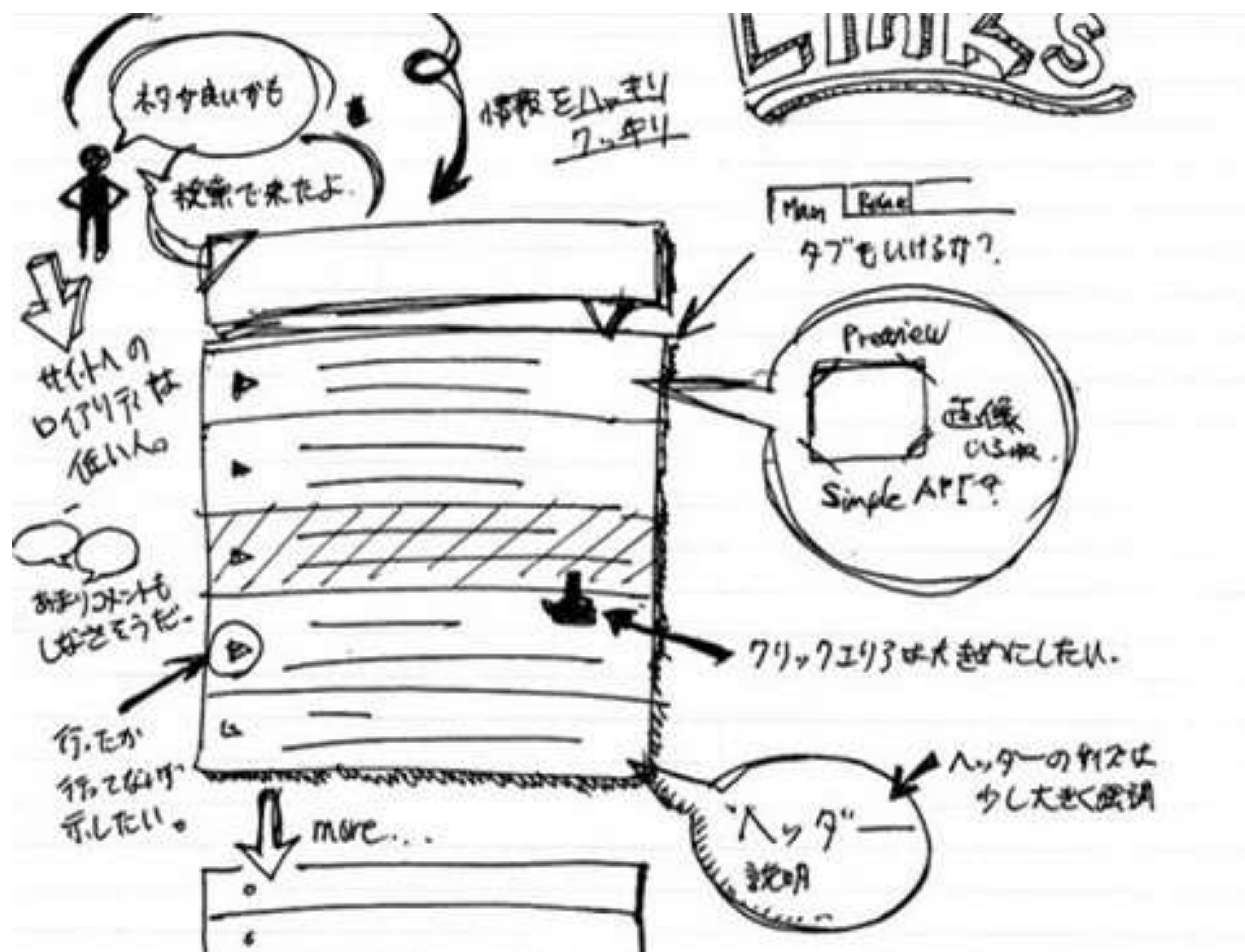
(compatibility testing) — тестирование, направленное на проверку способности приложения работать в указанном окружении

- кросс-платформенное тестирование
- кросс-браузерное тестирование
- мобильное тестирование





# ТЕСТИРОВАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА



## Тестирование графического интерфейса

(GUI testing) – анализ соответствия графического пользовательского интерфейса программы спецификациям.





# ТЕСТИРОВАНИЕ УДОБСТВА ИСПОЛЬЗОВАНИЯ



**Тестирование удобства использования** (usability testing) — тестирование, направленное на исследование того, насколько конечному пользователю понятно, как работать с продуктом, а также на то, насколько ему нравится использовать продукт



# ТЕСТИРОВАНИЕ ИНТЕРНАЦИОНАЛИЗАЦИИ



**Тестирование интернационализации** (internationalization testing, i18n) — тестирование, направленное на проверку **ГОТОВНОСТИ** продукта к работе с использованием различных языков и с учётом различных национальных и культурных особенностей.





# ТЕСТИРОВАНИЕ ЛОКАЛИЗАЦИИ



**Тестирование локализации** (localization testing, l10n) — тестирование, направленное на проверку корректности и качества адаптации продукта к использованию на том или ином языке с учётом национальных и культурных особенностей.





# ТЕСТИРОВАНИЕ БЕЗОПАСНОСТИ



**Тестирование безопасности**  
(security testing) - тестирование с целью оценить защищенность программного продукта.



# ТЕСТИРОВАНИЕ ДОСТУПНОСТИ



## Тестирование доступности

(accessibility testing) - тестирование, направленное на исследование пригодности продукта к использованию людьми с ограниченными возможностями (слабым зрением и т.д.)





# ТЕСТИРОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ



**Тестирование производительности** (performance test) - тест, проводимый с целью оценить поведение системы под нагрузкой





# НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ

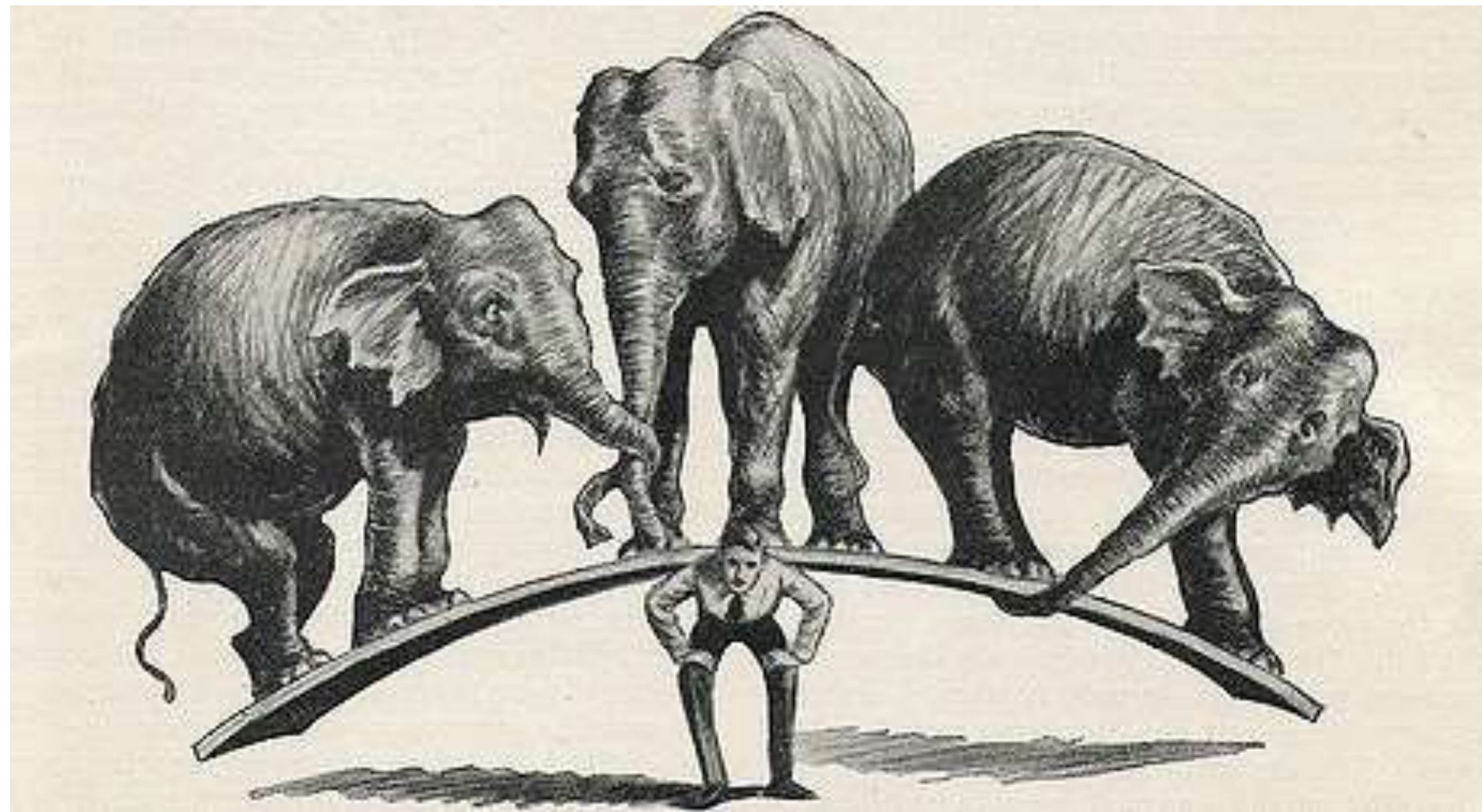


## **Нагрузочное тестирование**

(load testing) — исследование способности приложения сохранять заданные показатели качества при нагрузке в допустимых пределах и некотором превышении этих пределов (определение «запаса прочности»)



# СТРЕСС ТЕСТИРОВАНИЕ



## **Стресс тестирование**

(stress testing) — исследование поведения приложения при нештатных изменениях нагрузки, значительно превышающих расчётный уровень, или в ситуациях недоступности значительной части необходимых приложению ресурсов



A man with a beard and mustache, wearing a dark bow tie and a light-colored shirt, is smiling and gesturing with both hands raised. He is in a workshop or office environment with computer monitors and papers visible in the background. The entire image has a blue color overlay.

Есть вопросы?  
Давайте обсудим!



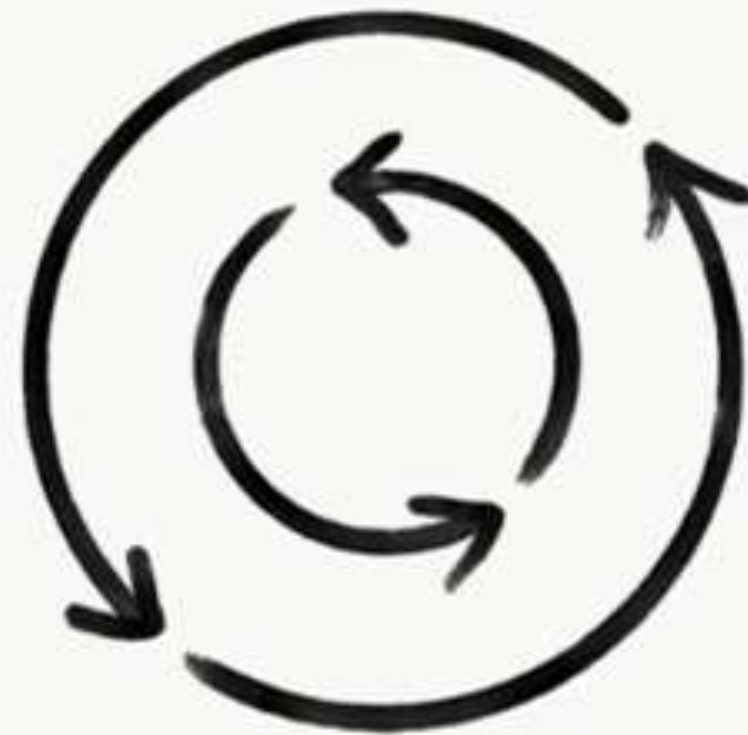


ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# ВИДЫ ТЕСТИРОВАНИЯ

По «тестированию изменений»

REPEAT





# ТЕСТИРОВАНИЕ НОВОГО ФУНКЦИОНАЛА



**Тестирование нового функционала**  
(new feature testing)





# ПОВТОРНОЕ ТЕСТИРОВАНИЕ



## Повторное тестирование

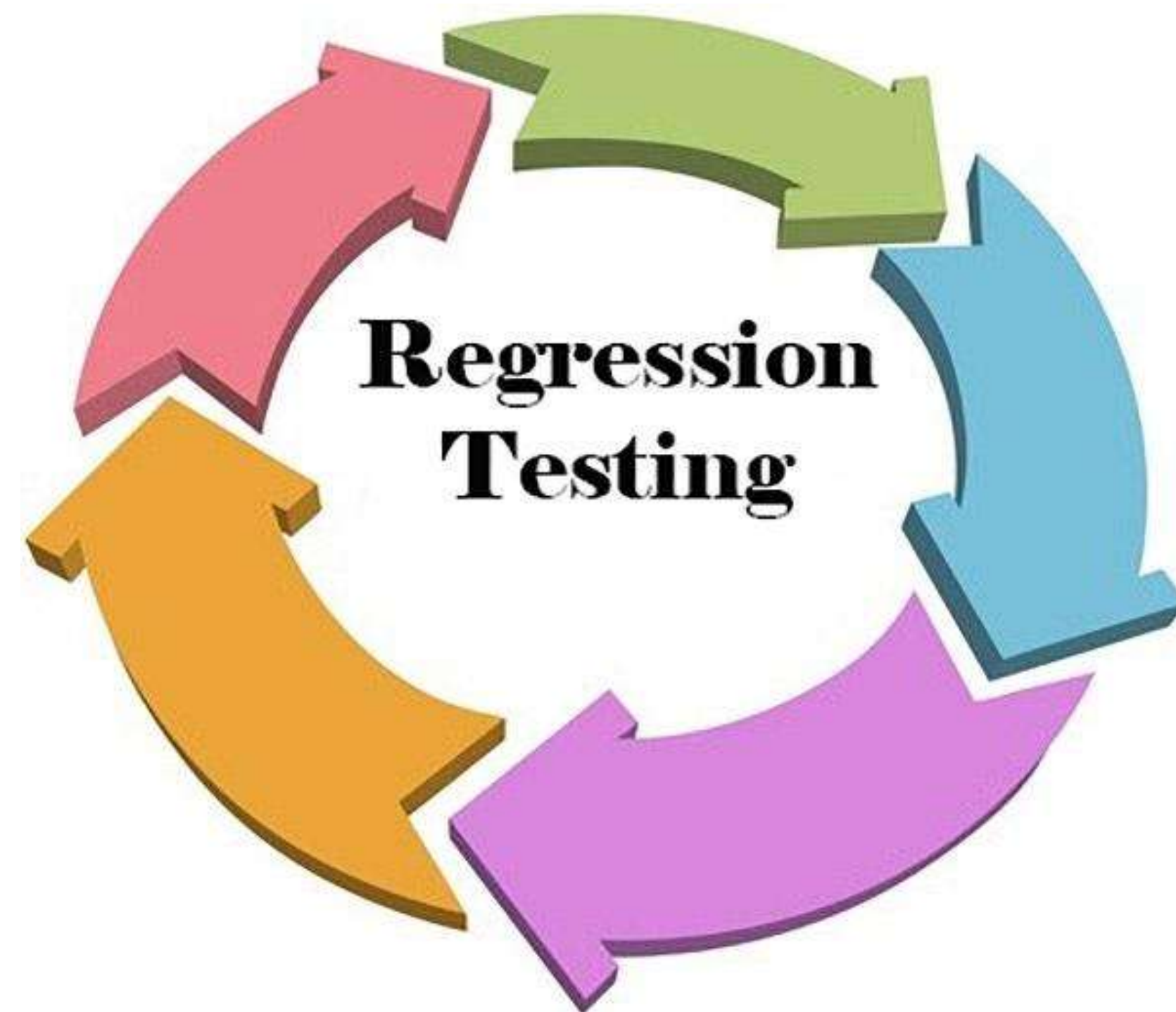
(re-testing, confirmation testing) — выполнение тест-кейсов, которые ранее обнаружили дефекты, с целью подтверждения устранения дефектов







# РЕГРЕССИОННОЕ ТЕСТИРОВАНИЕ



## Регрессионное тестирование

(regression testing) — тестирование, направленное на проверку того факта, что в ранее работоспособной функциональности не появились ошибки, вызванные изменениями в приложении или среде его функционирования

Мы применяем регрессионное тестирование, т.к. код сложный, и часто работает по принципу: "Купил хлеба — перестала открываться балконная дверь по вторникам"







# ТЕСТИРОВАНИЕ ИЗМЕНЕНИЙ

## Резюме

- **Исправлена ошибка** – программа должна быть протестирована повторно.
- **Исправления старых** – могут появляться новые.
- **Обещали исправить – не факт**, что исправили.
- Тесты должны разрабатываться с целью их повторного использования.
- **Повторное тестирование** – это перезапуск «провалившихся» тестов после исправления ошибок, чтобы убедиться, что исправленное работает.
- **Регрессионное тестирование** - это запуск более широкого набора тестов, чтобы проверить, не появились ли неожиданные ошибки в коде, который не меняли.

A man with a beard and mustache, wearing a dark bow tie and a light-colored shirt, is smiling and gesturing with both hands raised. He is in a workshop or office environment with computer monitors and papers visible in the background. The entire image has a blue color overlay.

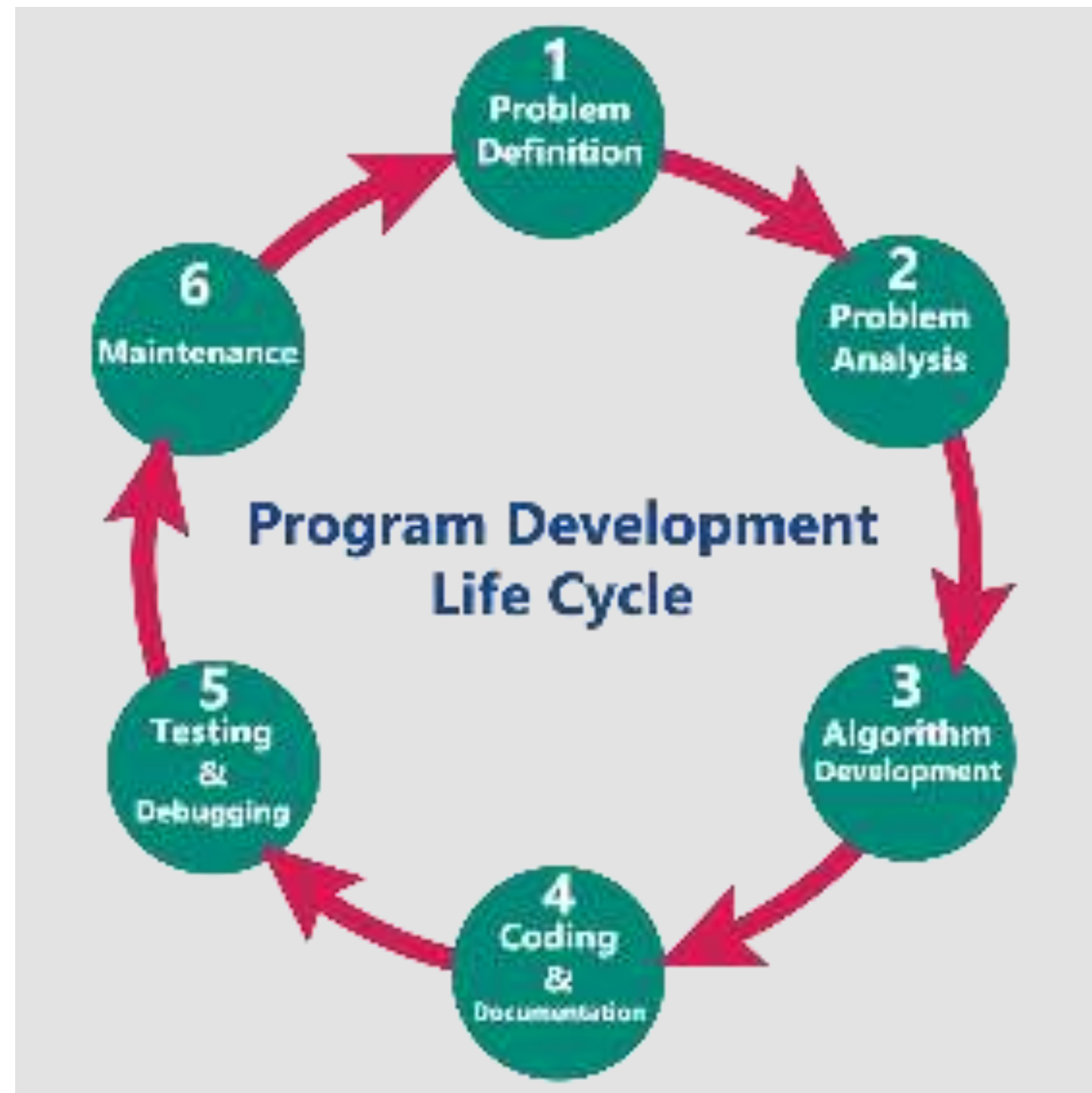
Есть вопросы?  
Давайте обсудим!





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# ВИДЫ ТЕСТИРОВАНИЯ



Типы тестирования в зависимости от фазы разработки



# ВИДЫ ТЕСТИРОВАНИЯ

## **Тестирование разработки**

(development testing) -

тестирование, проводимое во время разработки системы, обычно в рабочей среде разработчиков.

**Основная цель** – вызвать максимально возможное количество отказов, чтобы они были **обнаружены и исправлены**.

## **Приёмочное тестирование**

(acceptance testing) -

тестирование по отношению к потребностям пользователя или заказчика.

**Основная цель** – подтвердить, что система работает, как ожидается, чтобы **решить принимать или нет**.





# ТЕСТИРОВАНИЕ В ПЕРИОД СОПРОВОЖДЕНИЯ

## Тестирование в период сопровождения

(maintenance testing) - тестирование изменений в действующей системе или влияния изменений в окружении на действующую систему.

**Основная цель** – проверить, что при разработке изменений **не было произведено новых дефектов.**





# ТЕСТИРОВАНИЕ В ПЕРИОД СОПРОВОЖДЕНИЯ

- **Изменение** программных продуктов. Включает запланированные улучшения, поправки и аварийные изменения, изменения окружения (обновление БД, патчи для уязвимостей ОС)
- **Миграция** программных продуктов (из одной программы в другую)
- **Вывод** из эксплуатации ПО или систем. Может включать тестирование миграции данных или архивирования (если требуется длительное хранение данных)

## Что делать:

- **Регрессионное тестирование для частей системы**, которые подверглись изменениям

Объем тестов связан с риском изменения, размером существующей системы и размером изменения.





# ТЕСТИРОВАНИЕ В ПЕРИОД СОПРОВОЖДЕНИЯ

## Особенности:

- изменения в приложении - как правило незначительные
- система должна быть протестирована **быстро и эффективно**, так как:
  - ранняя остановка тестирования – ошибки могут быть пропущены
  - затянутое тестирование – упущенные возможности бизнеса
- эти системы могут быть установлены и работать годами:
  - жизненно важные для бизнеса
  - могут быть в использовании **24/7**

## Основные сложности для тестировщика:

- **отсутствие документации**/требований к приложению
- **старая/не** актуальная документация

## Выход:

- уже **имеющаяся** система (для эталонного тестирования)
- **руководство** пользователя
- профессиональные знания **специалиста**

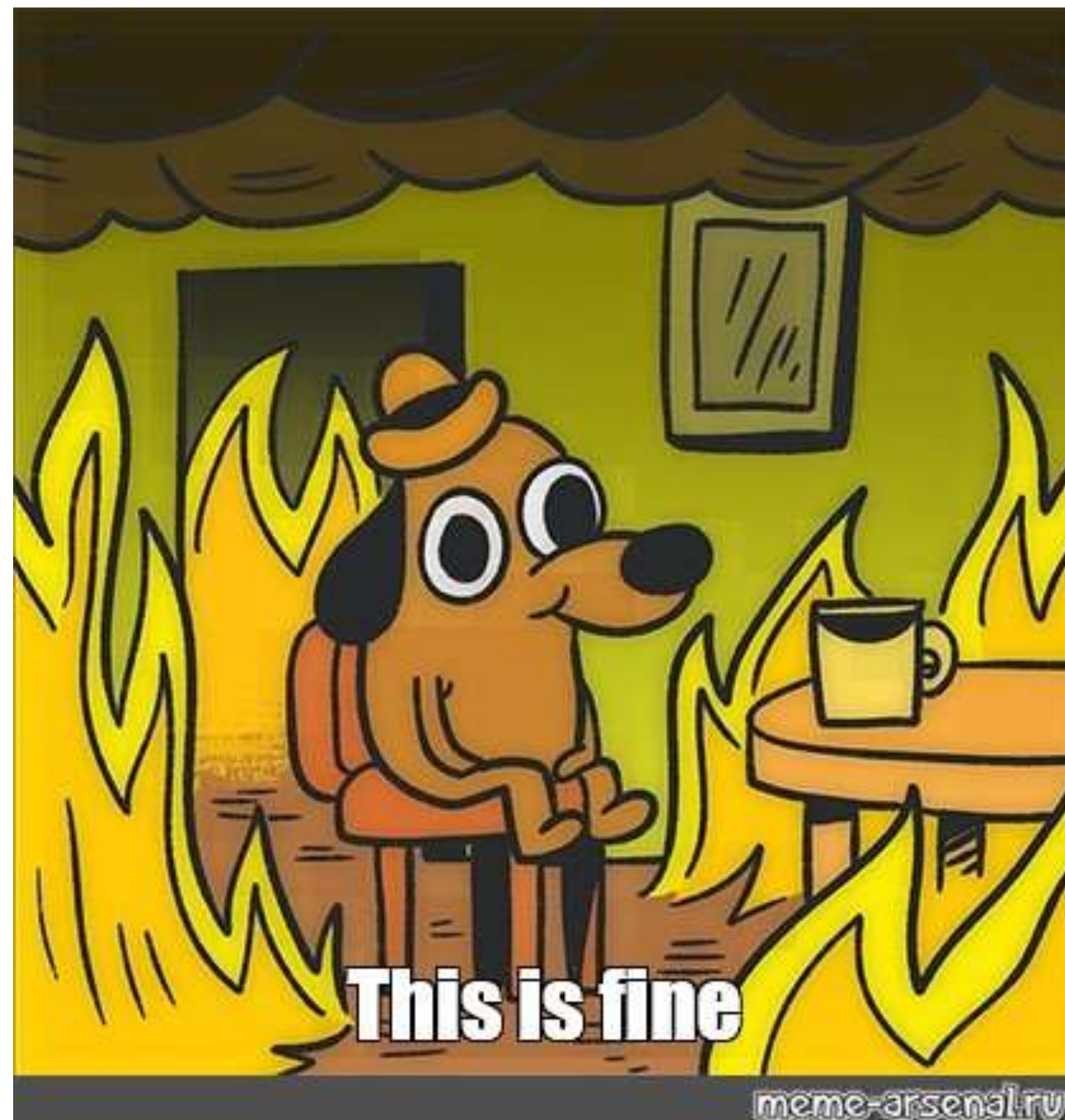
A man with a beard and mustache, wearing a dark bow tie and a light-colored shirt, is smiling and gesturing with both hands raised. He is in a workshop or office environment with computer monitors and papers visible in the background. The entire image has a blue color overlay.

Есть вопросы?  
Давайте обсудим!





# ВИДЫ ТЕСТИРОВАНИЯ



По степени важности тестируемых функций и приоритету выполнения тестов





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# ПРИОРИТЕТ ВЫПОЛНЕНИЯ



## Дымовое тестирование

(**smoke test**) направлено на проверку самой главной, важной, ключевой функциональности, неработоспособность которой делает бессмысленной саму идею использования приложения



## Тестирование критического пути

(**critical path test**) направлено на исследование функциональности, используемой типичными пользователями в типичной повседневной деятельности



## Расширенное тестирование

(**extended test**) направлено на исследование всей заявленной в требованиях функциональности — даже той, которая низко проранжирована по степени важности





# ВОЗМОЖНА ПУТАНИЦА!

Внимание!

Единой классификации не существует, и две категории имеют в обиходе профессионалов похожие названия:

**Уровни тестирования** (по степени детализации приложения)  
**= компонентное, интеграционное, системное.**

**Уровни функционального тестирования** (по важности функций)  
**= smoke, critical path, extended.**

A man with a beard and mustache, wearing a dark shirt and a bow tie, is smiling and gesturing with both hands raised. He is in a workshop or office environment with computer monitors and papers visible in the background. The entire image has a blue color overlay.

Есть вопросы?  
Давайте обсудим!





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# ВИДЫ ТЕСТИРОВАНИЯ



По принципу работы с приложением



## ПО ПРИНЦИПУ РАБОТЫ С ПРИЛОЖЕНИЕМ

### **Позитивное тестирование**

(positive testing) направлено на исследование приложения в ситуации, когда все действия выполняются строго по инструкции без каких бы то ни было ошибок, отклонений, ввода неверных данных и т.д.

### **Негативное тестирование**

(negative testing) направлено на исследование работы приложения в ситуациях, когда с ним выполняются (некорректные) операции и/или используются данные, потенциально приводящие к ошибкам (деление на ноль)





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# ВИДЫ ТЕСТИРОВАНИЯ

```
try {  
    // race condition fix  
    Thread.Sleep(2000);  
    // Нет времени объяснять, тут нужна 2 (костыль)  
    int arg = Calculate(2);  
    Send(((arg & 7) << 2) + 345);  
    // TODO: после сдачи проекта переписать  
} catch {}  
if (!IsSuccess(true, false, false))  
    throw new Exception("Impossible");
```

По запуску кода на исполнение



# ПО ЗАПУСКУ КОДА НА ИСПОЛНЕНИЕ



## **Статическое тестирование**

(static testing) -  
тестирование без запуска  
кода на исполнение.



## **Динамическое тестирование**

(dynamic testing) -  
тестирование с запуском  
кода на исполнение.





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# ВИДЫ ТЕСТИРОВАНИЯ

По доступу к коду





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# МЕТОДЫ ТЕСТИРОВАНИЯ



## Метод белого ящика

(**white box testing**) - у тестировщика есть доступ к внутренней структуре и коду приложения



## Метод чёрного ящика

(**black box testing**) - у тестировщика либо нет доступа к внутренней структуре и коду приложения, либо недостаточно знаний для их понимания, либо он не обращается к ним в процессе тестирования



## Метод серого ящика

(**gray box testing**) - комбинация методов белого ящика и чёрного ящика, состоящая в том, что к части кода и архитектуры у тестировщика доступ есть, а к части — нет





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# МЕТОДЫ ТЕСТИРОВАНИЯ

Какие плюсы и минусы есть  
у каждого из этих методов  
тестирования?



A man with a beard and mustache, wearing a dark bow tie and a light-colored shirt, is smiling and gesturing with both hands raised. He is in a workshop or office environment with computer monitors and papers visible in the background. The entire image has a blue color overlay.

Есть вопросы?  
Давайте обсудим!





ВИДЫ И МЕТОДЫ ТЕСТИРОВАНИЯ

# ВИДЫ ТЕСТИРОВАНИЯ



По уровню формализации



## ПО УРОВНЮ ФОРМАЛИЗАЦИИ

**На основе тест-кейсов** (scripted testing, test case based testing) – тестирование производится на основе заранее подготовленных тест-кейсов.

**На основе чек-листов** (checklist-based testing) – тестирование производится на основе чек-листов.

**Исследовательское тестирование** (exploratory testing) – по тест-кейсу/сценарию, который дорабатывается в процессе выполнения самих тестов.

**Свободное (интуитивное) тестирование** (ad hoc testing) – ни тест-кейсов, ни чек-листов, ни сценариев, полностью опираемся на свой профессионализм и интуицию.





## ПО УРОВНЮ ФОРМАЛИЗАЦИИ

### Свободное тестирование

(ad hoc testing) -тестирование, выполняемое **неформально**: без формальной подготовки тестов, формальных методов проектирования тестов, определения ожидаемых результатов и руководства по выполнению тестирования.

### Ad Hoc Testing



No Documentation



No Test Case



No Test Design





## ПО УРОВНЮ ФОРМАЛИЗАЦИИ

### Исследовательское тестирование

(exploratory testing) - неформальный метод проектирования тестов, при котором тестировщик активно продолжает **проектирование тестов** **в то время**, как эти тесты **выполняются**, и использует полученную во время тестирования информацию для проектирования новых и улучшенных тестов.







## ПО УРОВНЮ ФОРМАЛИЗАЦИИ

**Хаотическое тестирование**  
(monkey testing) - тестирование  
**случайным выбором** из  
большого диапазона входов,  
**случайным нажатием кнопок**,  
без соотнесения с тем, как в  
реальности будет использоваться  
система.





A man with a beard and mustache, wearing a dark bow tie and a light-colored shirt, is smiling and gesturing with both hands raised. He is in a workshop or office environment with computer monitors and papers visible in the background. The entire image has a blue color overlay.

Есть вопросы?  
Давайте обсудим!





## ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Мы разрабатываем новое мобильное приложение – **месенджер**.

Приведите примеры (3-5 идей) smoke, critical path and extended проверок.





## ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Приведите примеры (3-5 идей) smoke, critical path and extended проверок

User Story #1

Summary: As an end user, I want to send files via chat.

Description:

- There is possibility to send \*.jpg and \*.doc format;
- User can click on “Browse” button and select file in file system;
- User can drag-and-drop file into chat field;
- In case of slow internet connection (< 1 Mbps), sending is canceled, user is notified with “File can not be sent due to slow Internet. Please, try later”.