



IT-Academy

ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР
ПРОГРАММИРОВАНИЯ И ВЫСОКИХ
ТЕХНОЛОГИЙ

Функциональное тестирование ПО

ОСНОВЫ ТЕСТИРОВАНИЯ ПО ВВЕДЕНИЕ



В ЭТОМ РАЗДЕЛЕ:



- НЕМНОГО ИСТОРИИ
- ПСИХОЛОГИЯ ТЕСТИРОВАНИЯ
- ПОЧЕМУ ТЕСТИРОВАНИЕ НЕОБХОДИМО?
- ПРИНЦИПЫ ТЕСТИРОВАНИЯ
- ВВЕДЕНИЕ В ОСНОВНУЮ ТЕРМИНОЛОГИЮ



НЕМНОГО ИСТОРИИ

> 60-е ГОДЫ

В 60-х годах прошлого века основное внимание уделялось т.н. «исчерпывающему тестированию» - проверке всех возможных путей выполнения кода со всеми возможными входными данными.

> 80-е ГОДЫ

В 80-х годах тестирование ПО расширилось таким понятием, как предупреждение дефектов.

> СОВРЕМЕННЫЙ ЭТАП

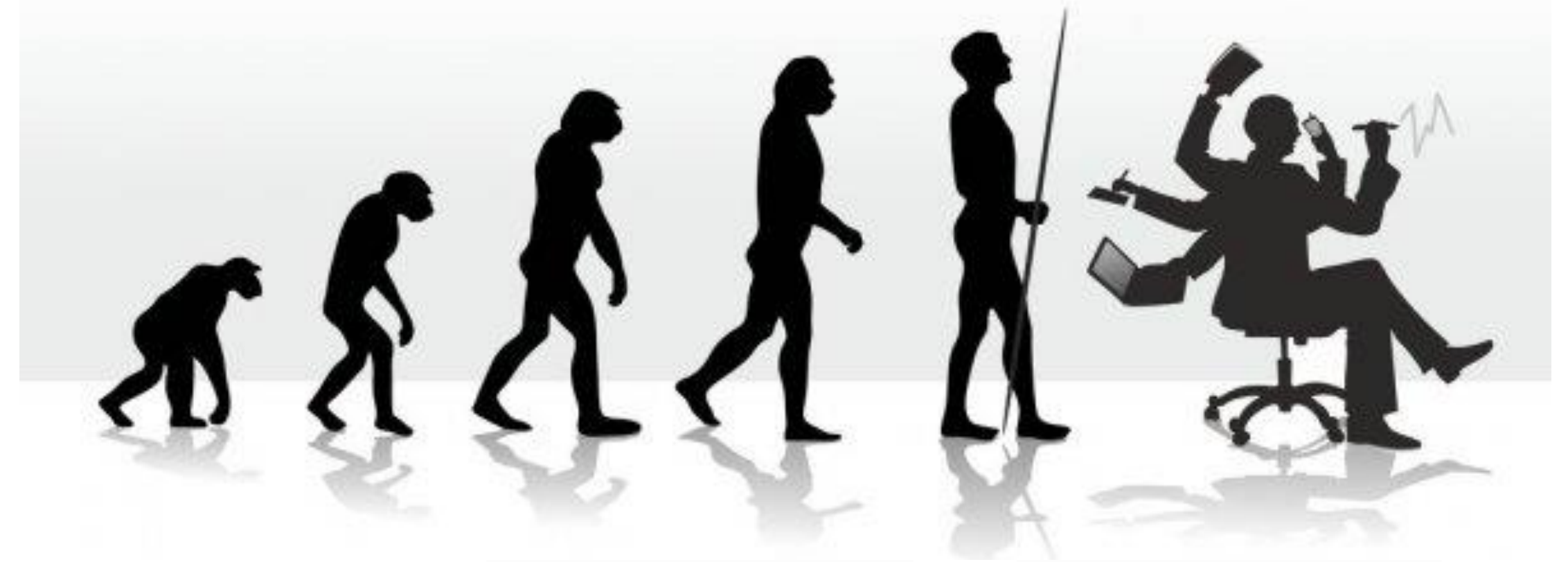
«гибкие методологии, тесная интеграция с разработкой, автоматизация».

> 70-е ГОДЫ

В начале 70-х тестирование ПО обозначалось как «процесс, направленный на демонстрацию корректности продукта» или как «деятельность по подтверждению правильности работы ПО».

> 90-е – 20-е ГОДЫ

В понятие «тестирование» стали включать планирование, проектирование, создание, поддержку и выполнение тестов и тестовых окружений.

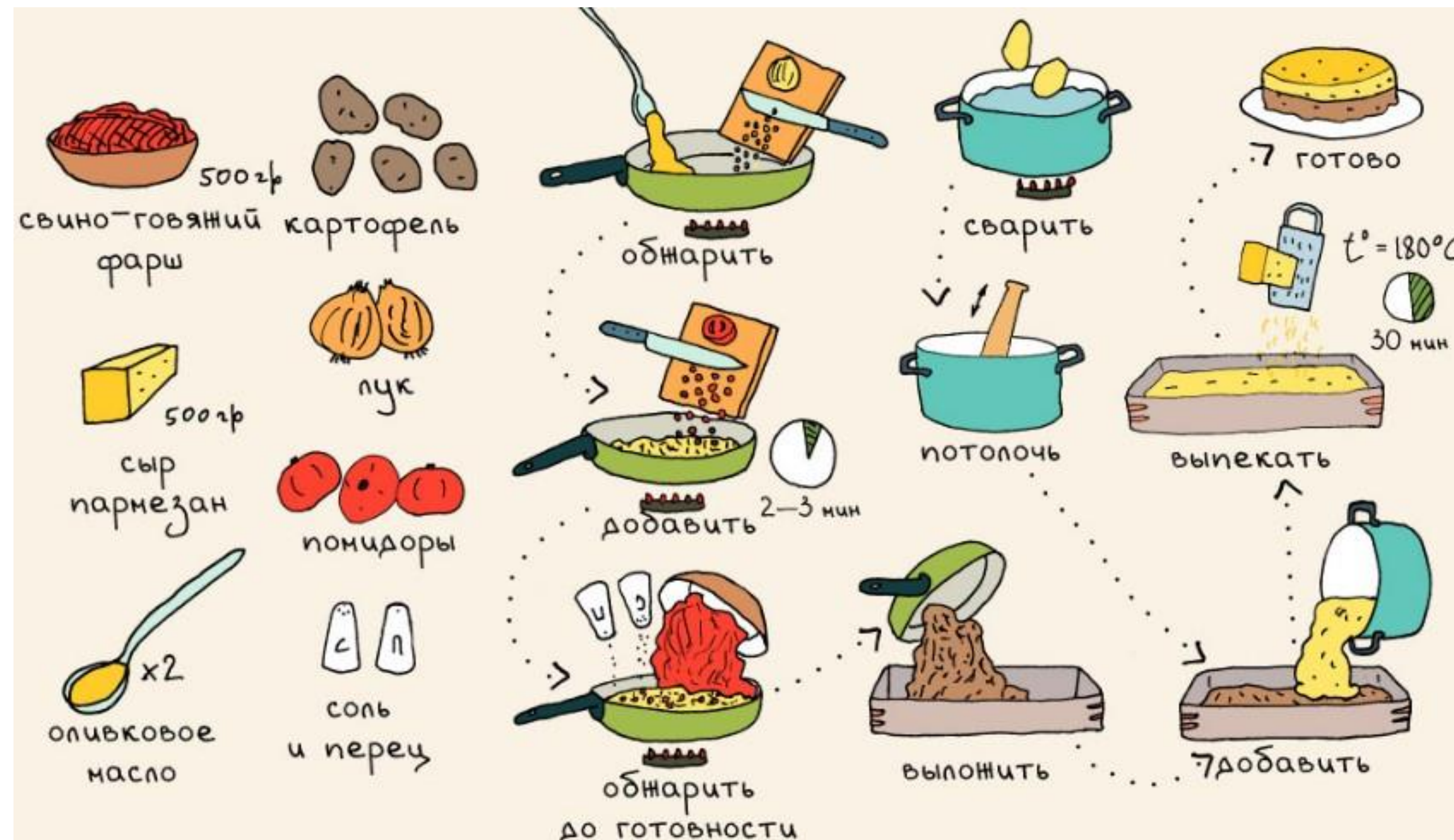




ВВЕДЕНИЕ

Пара слов о методологиях

Методология/модель/процесс разработки ПО – это структура (набор правил), согласно которой построена разработка.





ВВЕДЕНИЕ

Каскадный процесс

Анализ требований

Спецификация (документ)

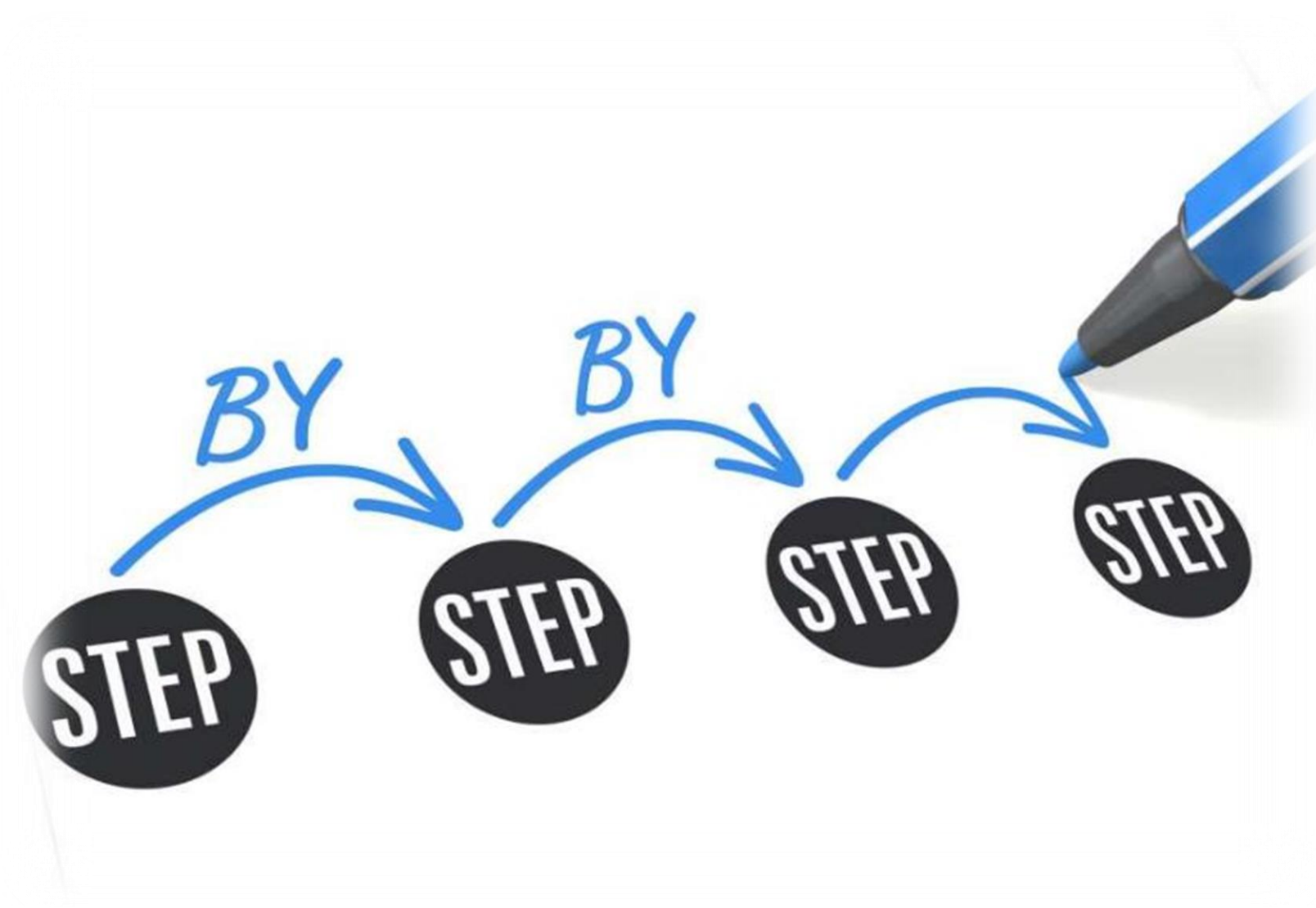
Проектирование ПО

Программирование

Тестирование

Внедрение (установка)

Сопровождение





Agile методологии (Scrum, Kanban...)

- Agile Manifesto разработан и принят 11-13 февраля 2001 года на лыжном курорте The Lodge at Snowbird в горах Юты.
- Манифест подписали представители следующих методологий:

Extreme programming

Scrum

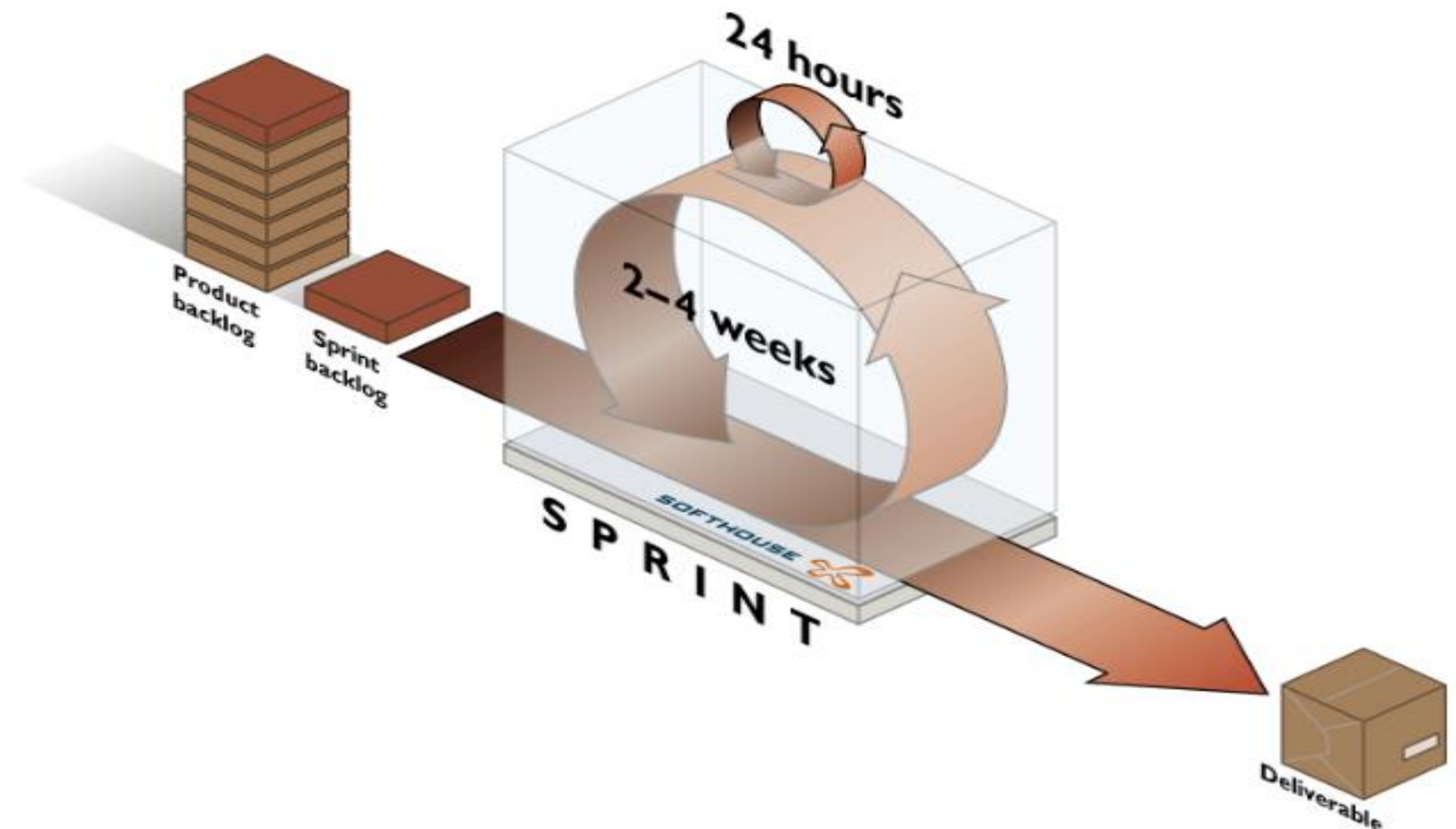
DSDM

Adaptive Software Development

Crystal Clear

Feature-Driven Development

Pragmatic Programming.





ВВЕДЕНИЕ

ПСИХОЛОГИЯ ТЕСТИРОВАНИЯ

- ХОРОШИЕ КОММУНИКАТИВНЫЕ НАВЫКИ
- СПОСОБНОСТЬ ЯСНО, БЫСТРО, ЧЕТКО ВЫРАЖАТЬ СВОИ МЫСЛИ
- ИСПОЛНИТЕЛЬНОСТЬ
- ОТВЕТСТВЕННОСТЬ
- ТЕРПЕНИЕ, ВНИМАТЕЛЬНОСТЬ К ДЕТАЛЯМ, НАБЛЮДАТЕЛЬНОСТЬ
- ГИБКОЕ МЫШЛЕНИЕ, ХОРОШАЯ СПОСОБНОСТЬ К ОБУЧЕНИЮ
- ХОРОШЕЕ АБСТРАКТНОЕ И АНАЛИТИЧЕСКОЕ МЫШЛЕНИЕ
- СПОСОБНОСТЬ СТАВВИТЬ НЕСТАНДАРТНЫЕ ЭКСПЕРИМЕНТЫ
- СКЛОННОСТЬ К ИСЛЕДОВАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ



КАКИЕ ЗНАНИЯ НУЖНЫ

- ① Общение с СУБД: SQL
- ① Основы Web (Client-Server, HTTP, HTML, WS)
- ① Администрирование ОС: Windows, Sun Solaris, Free-BSD, Linux
- ① Сетевое администрирование: TCP/IP, IPX/SPX, NetBIOS
- ① Программирование: C/C++/C#, Java, PHP, Object Pascal, HTML

- ① АНГЛИЙСКИЙ ЯЗЫК



**“ Первоначально важно хотя бы
"Умение излагать мысли и
замечания на родном языке", а
потом уже "Английский для
тестировщиков" ! ”**

Народная мудрость

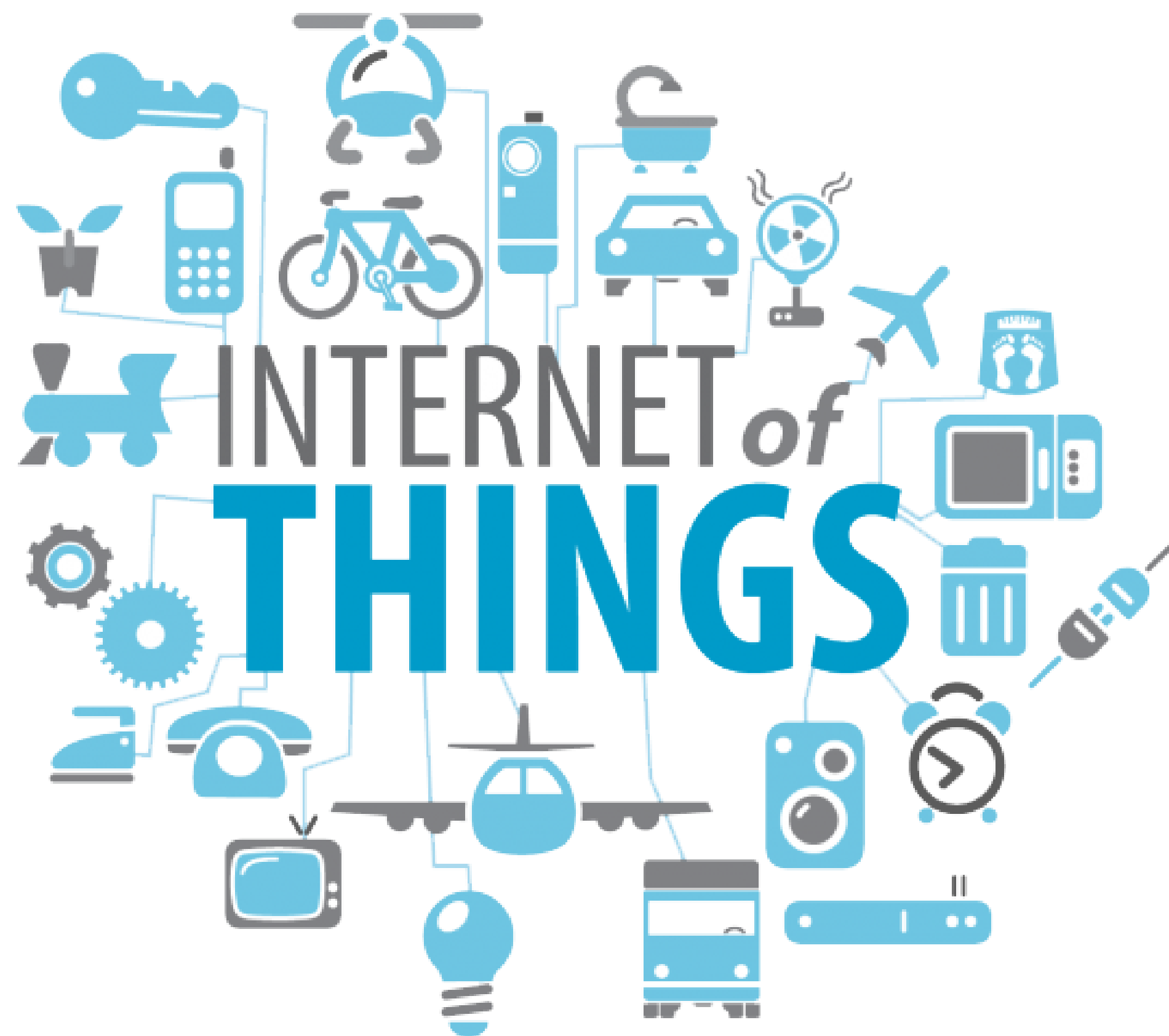


ПОЧЕМУ ТЕСИРОВАНИЕ НЕОБХОДИМО?

- **БИЗНЕС**
Пользователи склонны пользоваться качественными продуктами (даже если они дороже)
- **ДАННЫЕ**
Пользователи: «лучше не рисковать личными данными, деньгами и т.п.»
- **БЕЗАПАСТНОСТЬ**
Все: «Мы не хотим рисковать!»



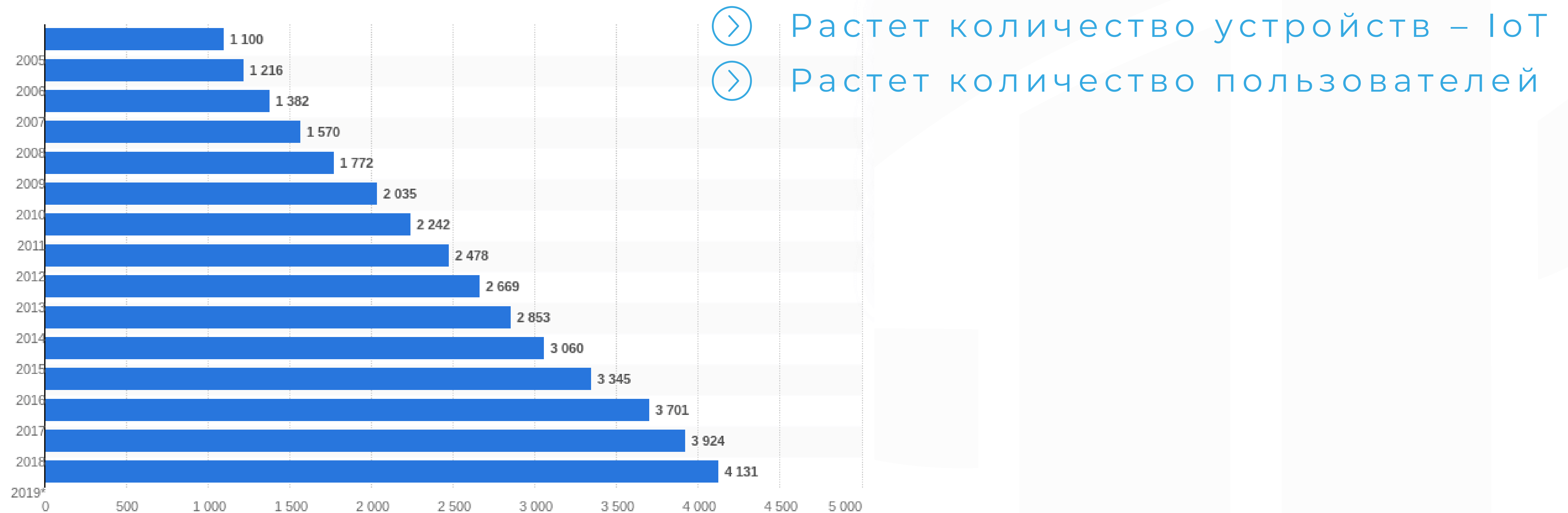
ПОЧЕМУ ТЕСИРОВАНИЕ НЕОБХОДИМО?



➤ Растет количество устройств – IoT



ПОЧЕМУ ТЕСИРОВАНИЕ НЕОБХОДИМО?



ВВЕДЕНИЕ



ПОЧЕМУ ТЕСИРОВАНИЕ НЕОБХОДИМО?



- Растет количество устройств – IoT
- Растет количество пользователей
- Растет сложность ПО



ПОЧЕМУ ТЕСИРОВАНИЕ НЕОБХОДИМО?



- Растет количество устройств – IoT
- Растет количество пользователей
- Растет сложность ПО
- Скорость выхода на рынок является ключевым конкурентным преимуществом сегодня

ВВЕДЕНИЕ



ПОЧЕМУ ТЕСИРОВАНИЕ НЕОБХОДИМО?



Никто не совершенен!

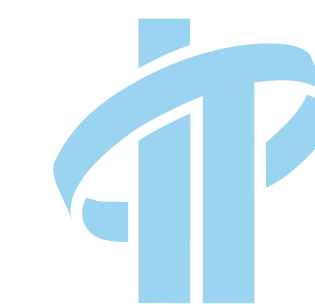
Чем большее давление на нас оказывается, тем более мы склонны делать ошибки.

В ИТ-разработке мы должны соблюдать временные сроки и бюджет.

Требования определены нечетко или плохо документированы.

Спецификации данных не завершены.

ПРЕДПОЛОЖЕНИЯ!



“

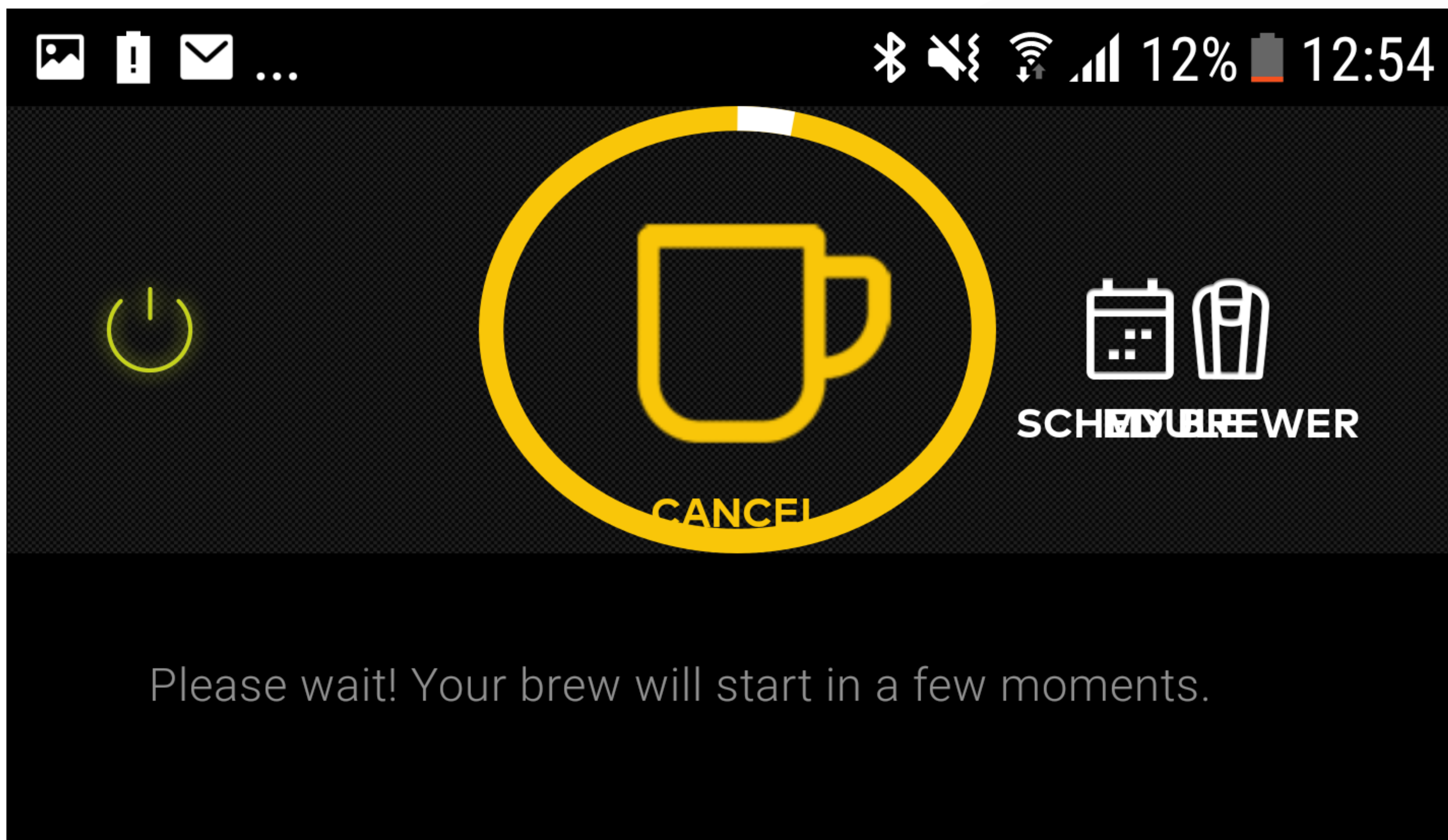
**Приведите примеры «багов ПО»
из жизни**

”

ВВЕДЕНИЕ



Примеры «багов ПО» из жизни



1 Right click to add/remove planogram rows and columns

Right click to configure new / edit existing planogram

2019-02-18 11:44 PM

[illegible]

2 Click **CHOOSE PRODUCTS** to add products to POG

CHOOSE PRODUCTS



Click, Drag and drop the product in the desired planogram column

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!



СЕМЬ ПРИНЦИПОВ ТЕСТИРОВАНИЯ

➤ Принцип 1 – Тестирование демонстрирует наличие дефектов

Тестирование может показать, что дефекты в программном обеспечении есть, но не может доказать, что никаких дефектов нет. Тестирование снижает вероятность того, что в программном обеспечении остались необнаруженные дефекты, но, даже если никаких дефектов не обнаружено, это не доказательство правильности работы программы.



СЕМЬ ПРИНЦИПОВ ТЕСТИРОВАНИЯ

- ① Принцип 1 – Тестирование демонстрирует наличие дефектов
- ② Принцип 2 – Исчерпывающее тестирование невозможно

Протестировать абсолютно все (все комбинации входов и условий) не представляется возможным, за исключением тривиальных случаев.

Вместо исчерпывающего тестирования, мы используем риски и приоритеты для эффективного сосредоточения усилий тестирования.



СЕМЬ ПРИНЦИПОВ ТЕСТИРОВАНИЯ

- ① Принцип 1 – Тестирование демонстрирует наличие дефектов
- ② Принцип 2 – Исчерпывающее тестирование невозможно
- ③ Принцип 3 – Раннее тестирование

Тестовые активности должны начинаться как можно раньше в цикле разработки программного обеспечения или системы, и должны быть направлены на достижение определенных целей.



СЕМЬ ПРИНЦИПОВ ТЕСТИРОВАНИЯ

- ① Принцип 1 – Тестирование демонстрирует наличие дефектов
- ② Принцип 2 – Исчерпывающее тестирование невозможно
- ③ Принцип 3 – Раннее тестирование
- ④ Принцип 4 – Скопление дефектов

Небольшое количество модулей содержат большинство дефектов, выявленных в ходе тестирования, или демонстрируют наибольшее количество операционных сбоев.

Это еще одно проявление правила Парето 80/20 – 80% дефектов находятся в 20% функций.



СЕМЬ ПРИНЦИПОВ ТЕСТИРОВАНИЯ

- ① Принцип 1 – Тестирование демонстрирует наличие дефектов
- ② Принцип 2 – Исчерпывающее тестирование невозможно
- ③ Принцип 3 – Раннее тестирование
- ④ Принцип 4 – Скопление дефектов
- ⑤ Принцип 5 – «Парадокс пестицида» (DDT paradox)

Если одни и те же тесты повторяются снова и снова, в конце концов с их помощью вы перестанете находить дефекты.



СЕМЬ ПРИНЦИПОВ ТЕСТИРОВАНИЯ

- ① Принцип 1 – Тестирование демонстрирует наличие дефектов
- ① Принцип 2 – Исчерпывающее тестирование невозможно
- ① Принцип 3 – Раннее тестирование
- ① Принцип 4 – Скопление дефектов
- ① Принцип 5 – «Парадокс пестицида» (DDT paradox)
- ① Принцип 6 – Тестирование зависит от контекста

Тестирование проводится по-разному в различных контекстах.

Контекст включает: тип продукта, его цели, связанные риски, доступные инструменты, ресурсы и время, опыт команды и т.д.



СЕМЬ ПРИНЦИПОВ ТЕСТИРОВАНИЯ

- ① Принцип 1 – Тестирование демонстрирует наличие дефектов
- ② Принцип 2 – Исчерпывающее тестирование невозможно
- ③ Принцип 3 – Раннее тестирование
- ④ Принцип 4 – Скопление дефектов
- ⑤ Принцип 5 – «Парадокс пестицида» (DDT paradox)
- ⑥ Принцип 6 – Тестирование зависит от контекста
- ⑦ Принцип 7 – Заблуждение об отсутствии ошибок

Нахождение и исправление дефектов не поможет, если разработанная система не удовлетворяет нуждам и ожиданиям пользователей.



Тестирование программного обеспечения (software testing) – это

- процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта
- выявления дефектов и повышения качества продукта





QA ≠ QC ≠ Testing

- **Обеспечение качества** (Quality Assurance) – совокупность мероприятий, охватывающих все технологические этапы разработки, выпуска и эксплуатации ПО информационных систем, предпринимаемых на разных стадиях жизненного цикла ПО, для обеспечения качества выпускаемого продукта.
- **Контроль качества** (Quality Control) – совокупность мероприятий проводимых в процессе разработки, для постоянного получения исчерпывающей информации о соответствии объекта тестирования поставленным требованиям.
- **Тестирование ПО** (Testing) – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.



Верификация vs Валидация

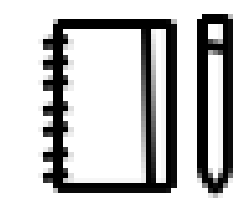
- **Верификация** (verification) – это процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа.
- **Валидация** (validation) – это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе.



План тестирования (test plan) – это

Документ, описывающий цели, подходы, ресурсы и график запланированных тестовых активностей и определяющий:

- что тестировать
- что не нужно тестировать
- кто будет тестировать
- где это нужно тестировать и на каком оборудовании
- методы и подходы для проектирования тестов
- критерии для начала и окончания тестирования, а также любые риски, требующие планирования на случай чрезвычайных обстоятельств



Чек-лист (check-list) – высокоуровневый список

- список, содержит пункты
- список того, что мы собрались сделать
- список того, что хотим не забыть
- список того, что будем проверять
- набор идей тестов



Тест-кейс (test case) – это

Документ в котором есть

**входные данные,
условия выполнения и
ожидаемые результаты,**

разработанный с целью проверки того или иного свойства или поведения программного средства.



Набор тестов (test suite) – это

Набор тестов (тест-кейсов), собранных в последовательность для достижения некоторой цели.

Хороший тестовый сценарий всегда следует некоторой логике, например: типичному использованию приложения, удобству тестирования, распределению функций по модулям и т.д.

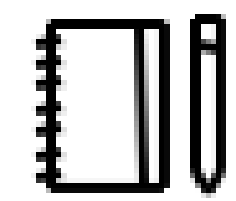




Дефект (defect, bug) – это

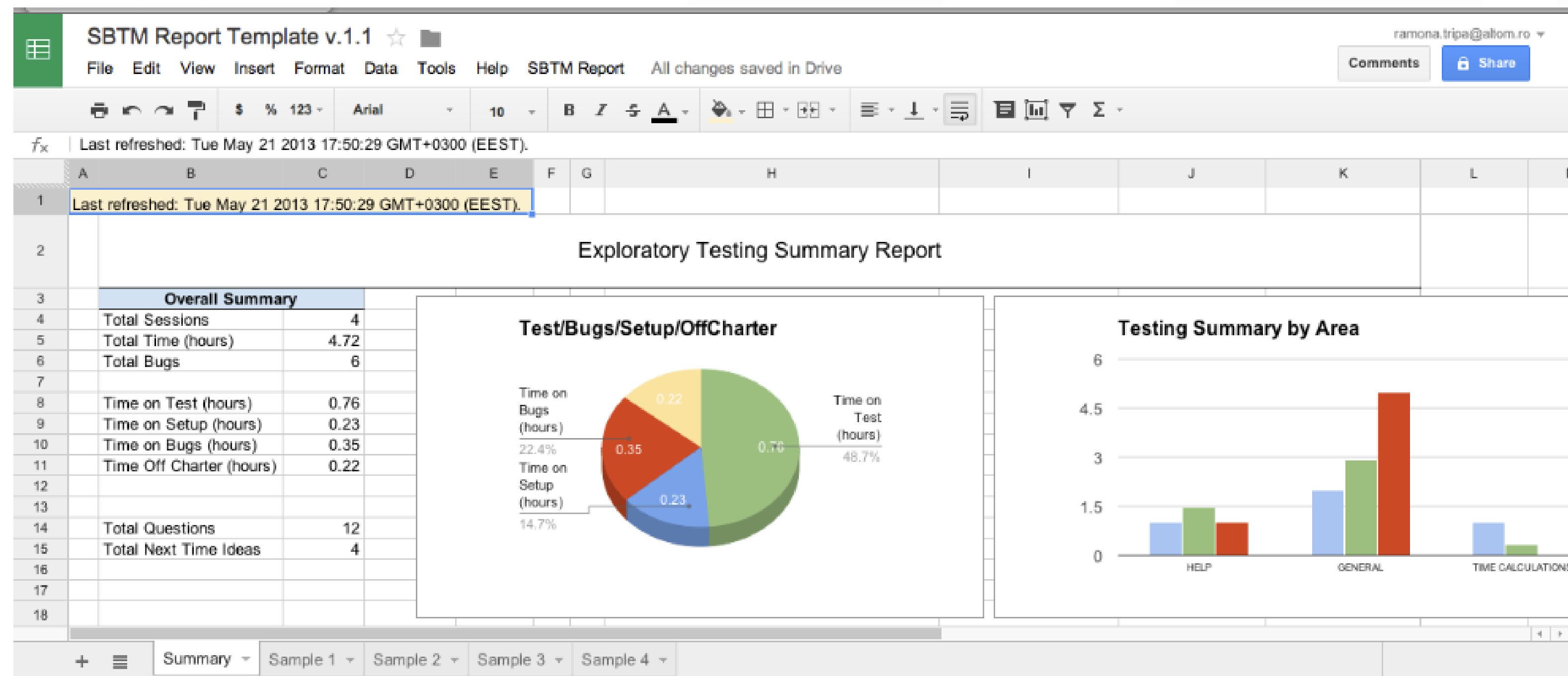
Любое несоответствие фактического и ожидаемого результата (согласно требованиям или здравому смыслу).

Исъян в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию, например неверный оператор или определение данных.



Отчет о тестировании (test result report, TRR) – это

Документ, подводящий итог проделанной работы в ходе тестирования, а также содержащий оценку состояния качества программы.



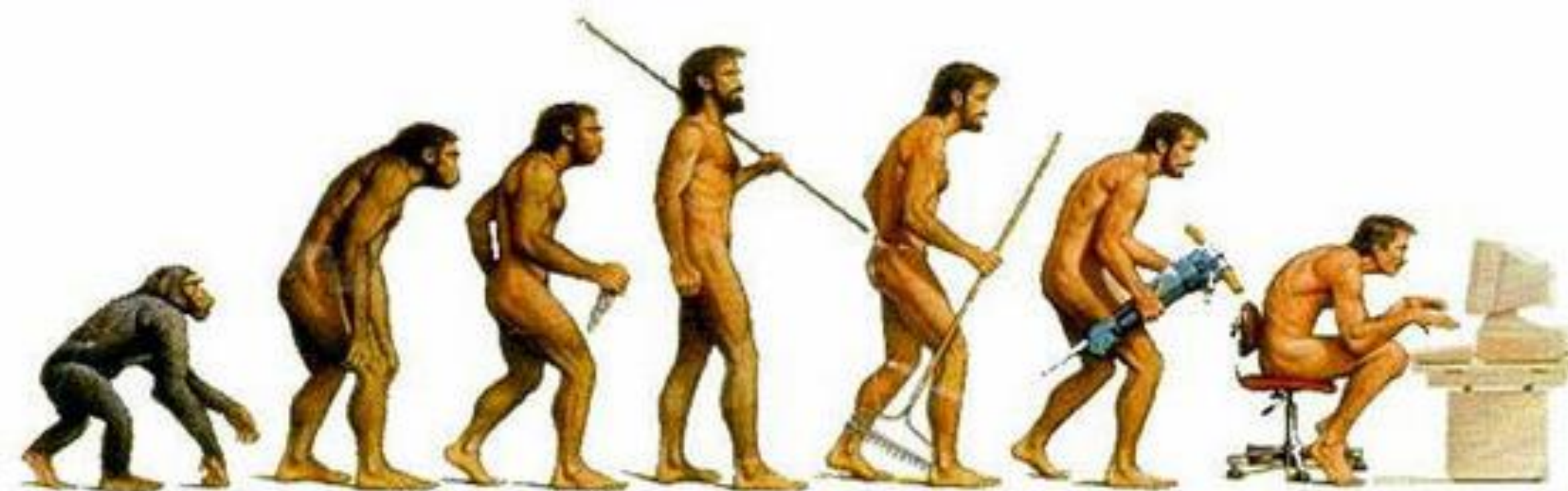


Билд («сборка», build) – это

Очередная версия программы.

Финальный билд – часто называют Релизом (Release) – то, что уходит пользователям/заказчику.

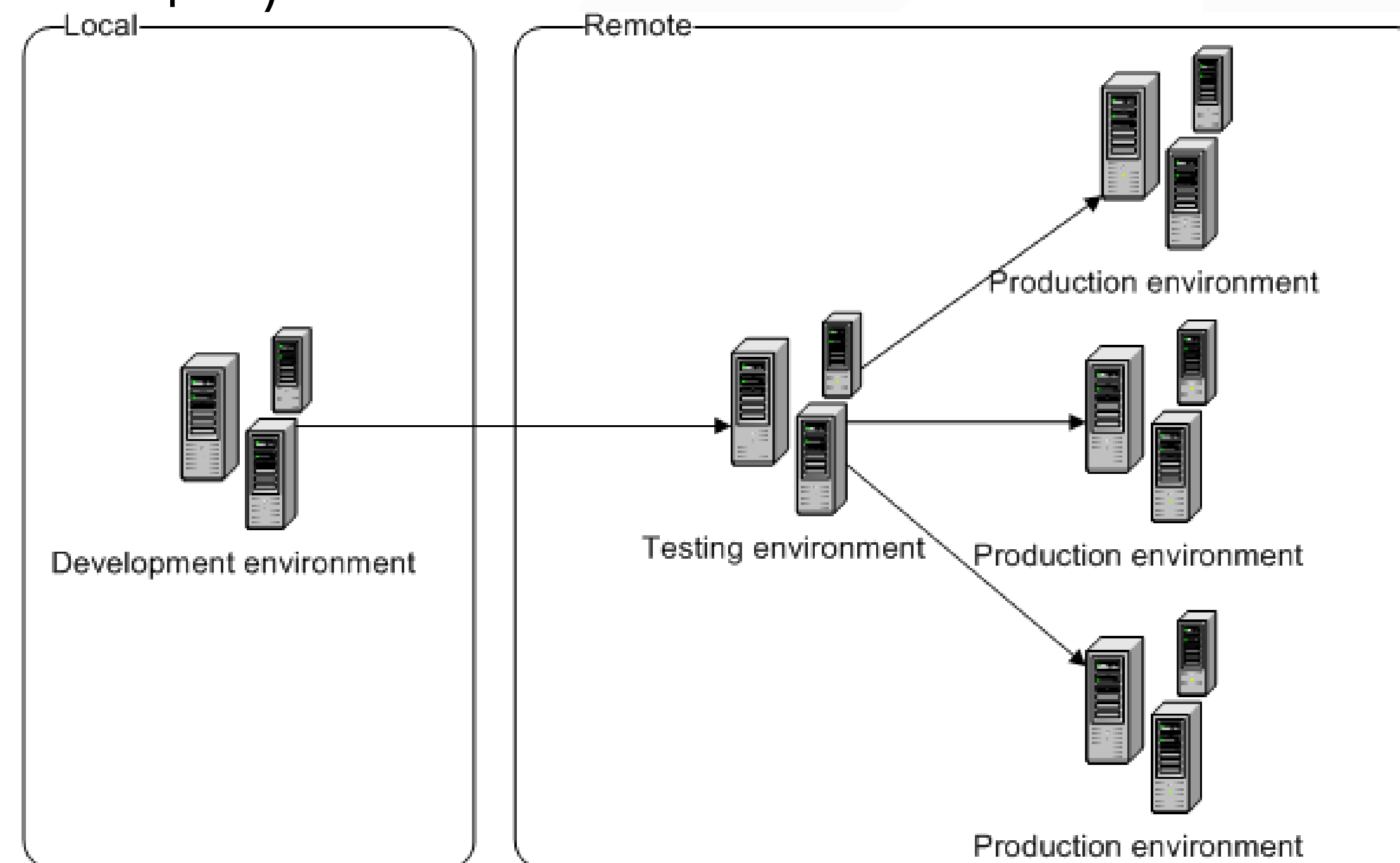
Ежедневная сборка (daily build) – действия, в ходе которых система ежедневно (обычно ночью) компилируется и собирается целиком, так что целостная система доступна в любое время, включая все последние изменения.





Тестовое окружение (test environment) – это

Аппаратура (по сути компьютер/смартфон и установленное на нем ПО) и инструментарий, необходимые для проведения теста (которыми пользуется тестировщик).





Отладка (Debugging) – это

Процесс поиска, анализа и устранения причин отказов в программном обеспечении.





Качество (Quality) – это

Степень, с которой компонент, система или процесс соответствует зафиксированным требованиям и/или ожиданиям и нуждам пользователя или заказчика.

- если заказчик доволен продуктом – продукт качественный
- если продукт соответствует требованиям – продукт качественный
- у качественного продукта всегда есть преимущества и нет серьёзных недостатков
- хорошее качество – низкий риск потерь (денег, времени, репутации...)
- заказчик должен быть удовлетворен



Метрика (metric) – это

Шкала измерений и метод, используемый для измерений [ISO 14598].

Варианты метрик:

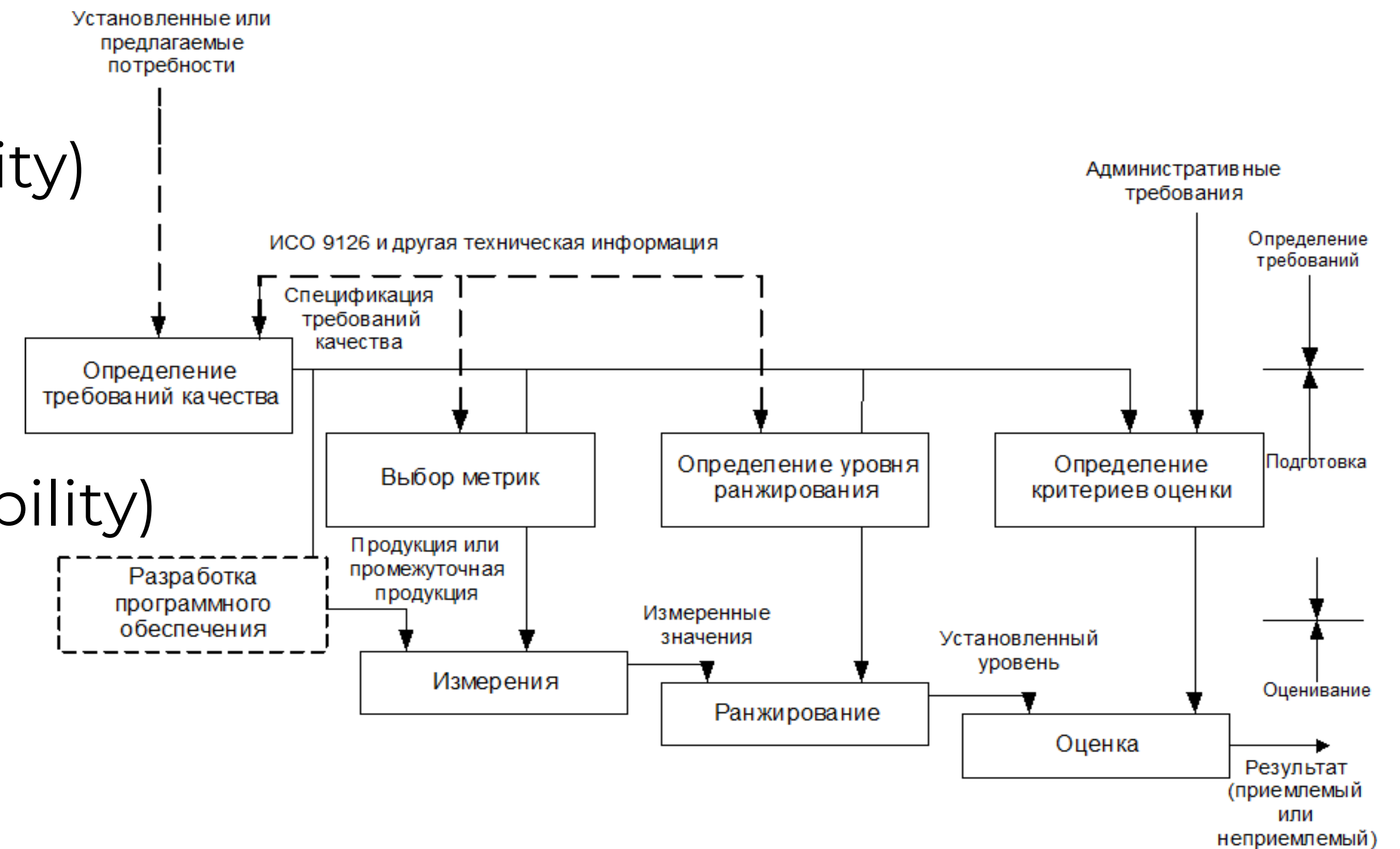
- покрытие требований тестами – не менее 80%
- плотность покрытия – не менее 3
- закрыто 100% известных критических дефектов,
90% дефектов средней критичности,
50% остальных дефектов.
- общий показатель прохождения тестов – не менее некоторого значения:

$$X = (\text{Passed}/\text{Executed}) * 100\%$$



Качество ПО включает характеристики:

- Функциональная пригодность (Functional suitability)
- Производительность (Performance efficiency)
- Совместимость (Compatibility)
- Удобство использования (Usability)
- Надежность (Reliability)
- Безопасность (Security)
- Ремонтопригодность (Maintainability)
- Переносимость (Portability)





ВВЕДЕНИЕ

5 МИФОВ О ТЕСТИРОВАНИИ

- Миф первый: тестирование — это скучно.





ВВЕДЕНИЕ

5 МИФОВ О ТЕСТИРОВАНИИ

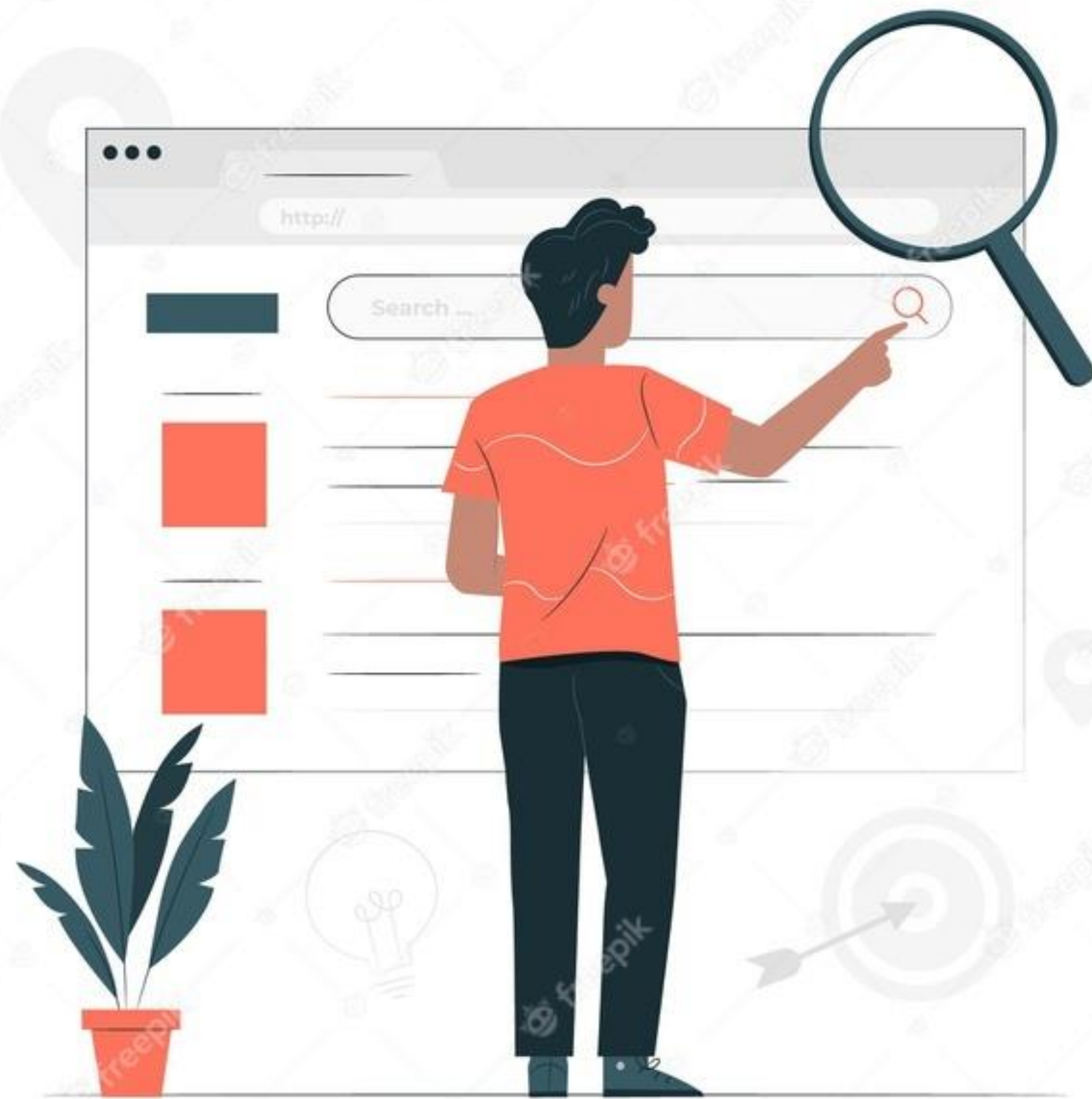


- Миф первый: тестирование — это скучно.
- Миф второй: тестирование — это просто.



ВВЕДЕНИЕ

5 МИФОВ О ТЕСТИРОВАНИИ

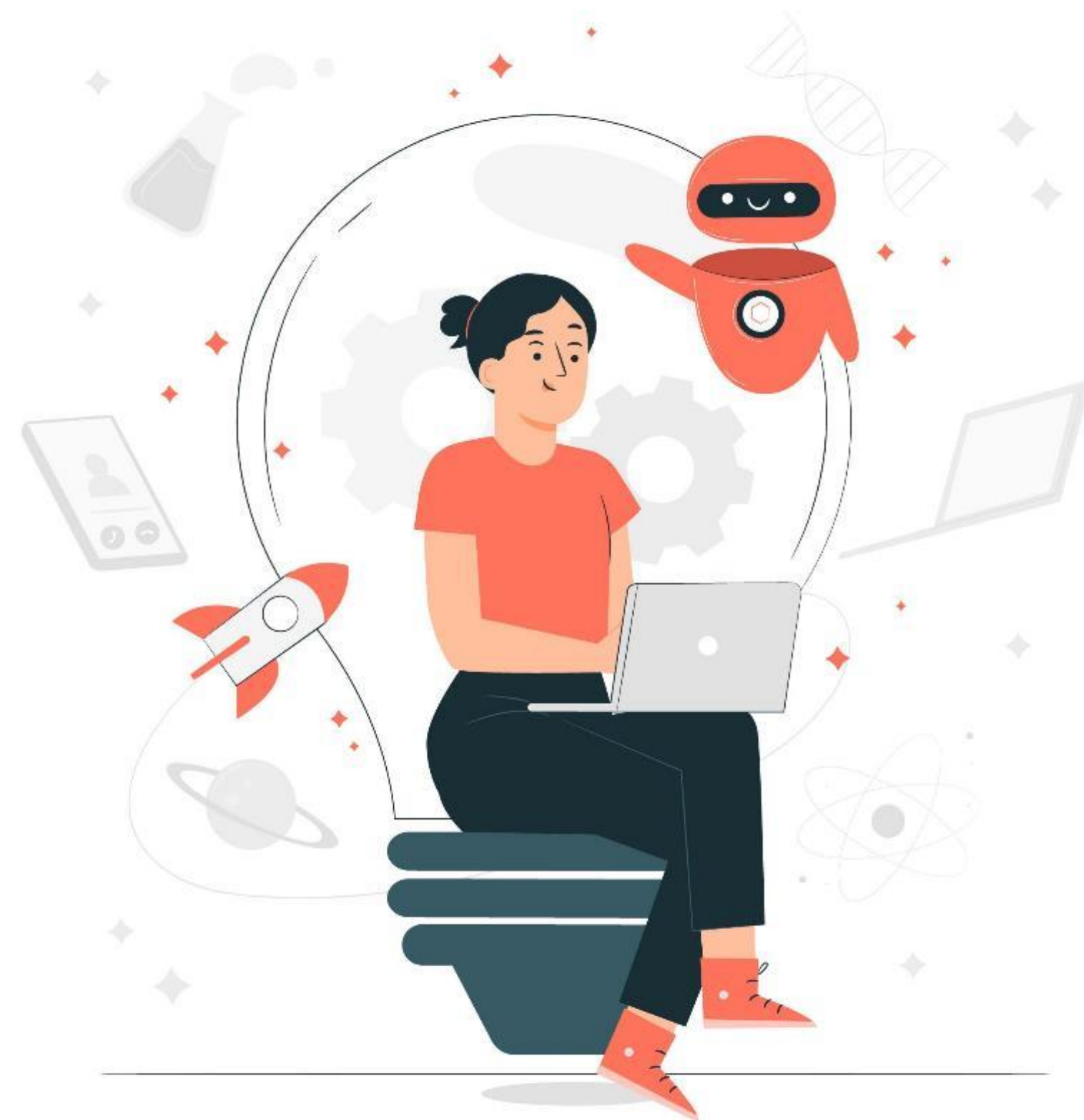


- Миф первый: тестирование — это скучно.
- Миф второй: тестирование — это просто.
- Миф третий: тестировщики всего лишь ищут ошибки.



ВВЕДЕНИЕ

5 МИФОВ О ТЕСТИРОВАНИИ

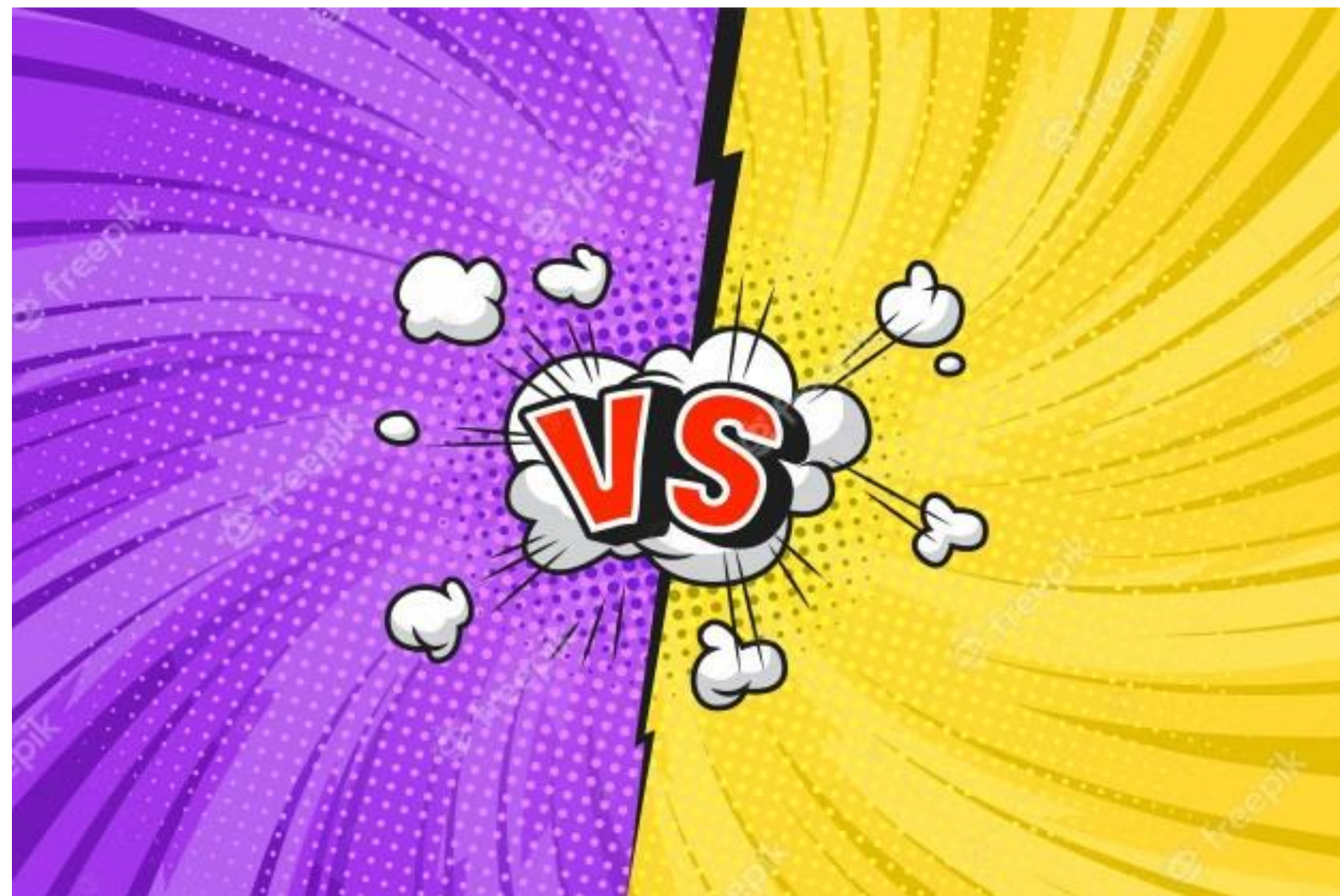


- Миф первый: тестирование — это скучно.
- Миф второй: тестирование — это просто.
- Миф третий: тестировщики всего лишь ищут ошибки.
- Миф четвертый: машины заменят тестировщиков, и они станут ненужными.



ВВЕДЕНИЕ

5 МИФОВ О ТЕСТИРОВАНИИ



- Миф первый: тестирование — это скучно.
- Миф второй: тестирование — это просто.
- Миф третий: тестировщики всего лишь ищут ошибки.
- Миф четвертый: машины заменят тестировщиков, и они станут ненужными.
- Миф пятый: тестировщики не ладят с разработчиками.



Рекомендуемые ресурсы

- <https://www.w3schools.com> – множество простой информации по целой серии технологий
- http://www.sql-ex.ru/learn_exercises.php – множество практических заданий по SQL
- <https://youtu.be/Z-a7MNStFQs> – простой полуторачасовой видеокурс по основам компьютерных сетей
- <https://htmlacademy.ru> – серия бесплатных курсов по HTML / CSS / JS / PHP
- <http://software-testing.ru> – большой портал, на котором есть как профессиональные материалы, так и небольшие статьи, понятные и полезные начинающим в помощь





Когда день тестировщика?



Photo # NH 96566-KN (Color) First Computer "Bug", 1947

9/2

9/9

0800 Andam started { 1.2700 9.037 847 025
 1000 " stopped - andam ✓ 9.037 846 995 correct
 1300 (032) MP-MC 1.58264000 4.615925059(-2)
 (033) PRO 2 2.130476415
 correct 2.130676415
 Relays 6-2 in 033 failed special speed test
 in relay .. 11.00 test.

1100 Relays changed
 Started Cosine Tape (Sine check)
 1525 Started Multi-Adder Test.

1545 Relay #70 Panel F
 (moth) in relay.

First actual case of bug being found.

1630 Andam started.
 1700 closed down.

Relay 3145
 Relay 3376



ВВЕДЕНИЕ

Давайте поищем баги!

Payment Details

☐ Remember

CARD NUMBER

Valid Card Numbe

MM

EXPIRY DATA

CV CODE

YY

CV

Final Payment

\$ €4200

Day




ВВЕДЕНИЕ

Давайте поищем баги!

Payment Details ☐ Remember

CARD NUMBER

Valid Card Numbe 

EXPIRY DATA

MM YY

CV CODE

CV

Final Payment \$ €4200

Day

A man with a beard and a bow tie is smiling and gesturing with his hands raised, as if in a discussion or presentation. The image is overlaid with a blue filter. The text "Есть вопросы? Давайте обсудим!" is written in white, bold, sans-serif font across the center of the image.

Есть вопросы?
Давайте обсудим!