

**SFT.EXT.01**

# **ВВЕДЕНИЕ В ТЕСТИРОВАНИЕ**

Модели жизненного цикла ПО. Жизненный цикл тестирования. Основная терминология.

**Author: Svyatoslav Kulikov**  
**Training and Education Manager**  
**[svyatoslav\\_kulikov@epam.com](mailto:svyatoslav_kulikov@epam.com)**

## Содержание

1. История развития тестирования.
2. Важность тестирования.
3. Модели жизненного цикла ПО.
4. Жизненный цикл тестирования.
5. Технические навыки и личностные качества тестировщика.
6. Основная терминология тестирования.

# ИСТОРИЯ РАЗВИТИЯ ТЕСТИРОВАНИЯ

60-е годы –  
«исчерпывающее  
тестирование»



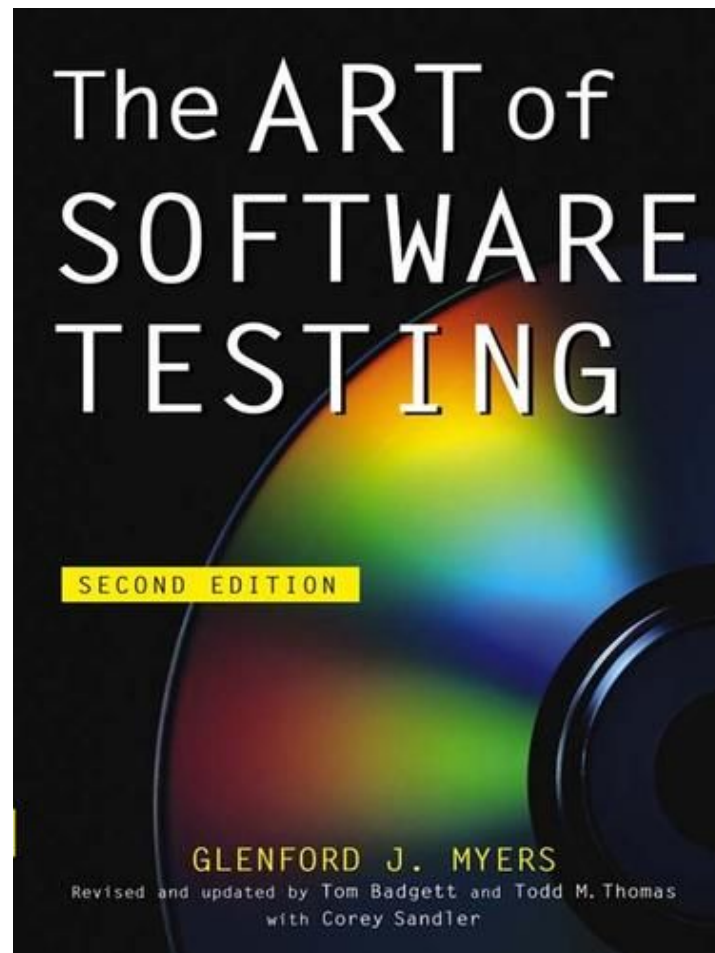
## История развития тестирования 60-е годы

60-е годы –  
«исчерпывающее тестирование»

**НЕВОЗМОЖНО**



20 вложенных  
операторов if =>  
**1'048'576** ветвей  
выполнения



# 70-е годы – «поиск дефектов»



70-е годы –  
«поиск дефицита»

НЕЭФФЕКТИВНО





## История развития тестирования, 70-е годы

60-е годы –  
«программа  
работает»



VS



70-е годы –  
«программа  
НЕ работает»

80-е годы –  
«предупреждение  
дефектов»

**ЭТО СРАБОТАЛО**



90-е годы –  
«обеспечение  
качества»



0-е ☺ годы –  
«тотальное  
обеспечение  
качества»



Современный этап – «гибкие методологии, тесная интеграция с разработкой, автоматизация»



# ВАЖНОСТЬ ТЕСТИРОВАНИЯ

## Важность тестирования

Бизнес: «пользователи склонны пользоваться качественными продуктами (даже если они дороже)»



Пользователи:  
«лучше не рисковать  
личными данными,  
деньгами и т.п.»





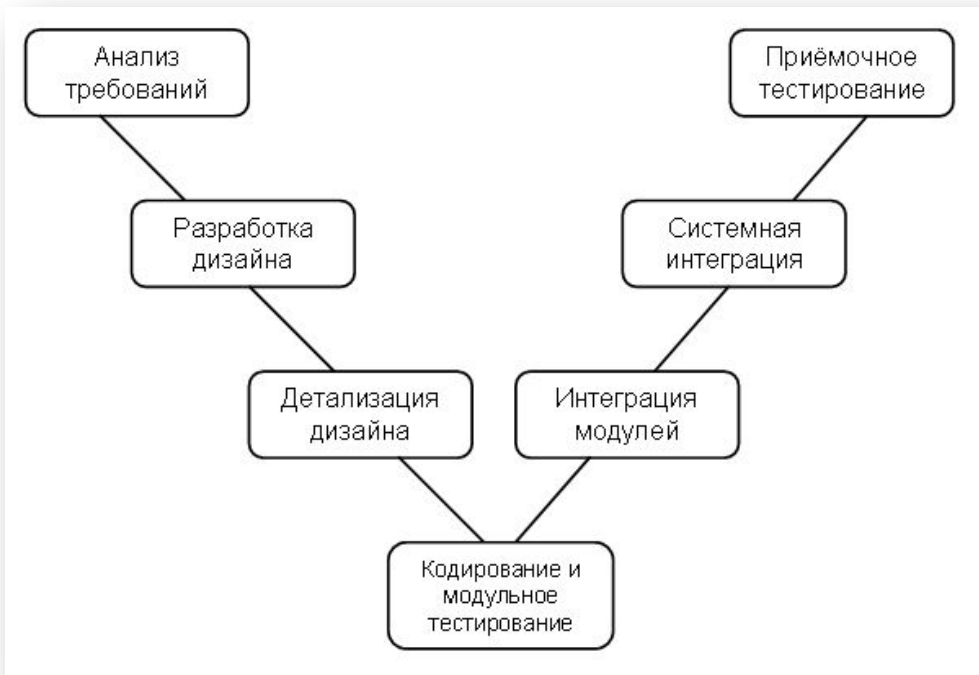
## Важность тестирования

Все: «мы не хотим рисковать»



# МОДЕЛИ ЖИЗНЕННОГО ЦИКЛА ПО

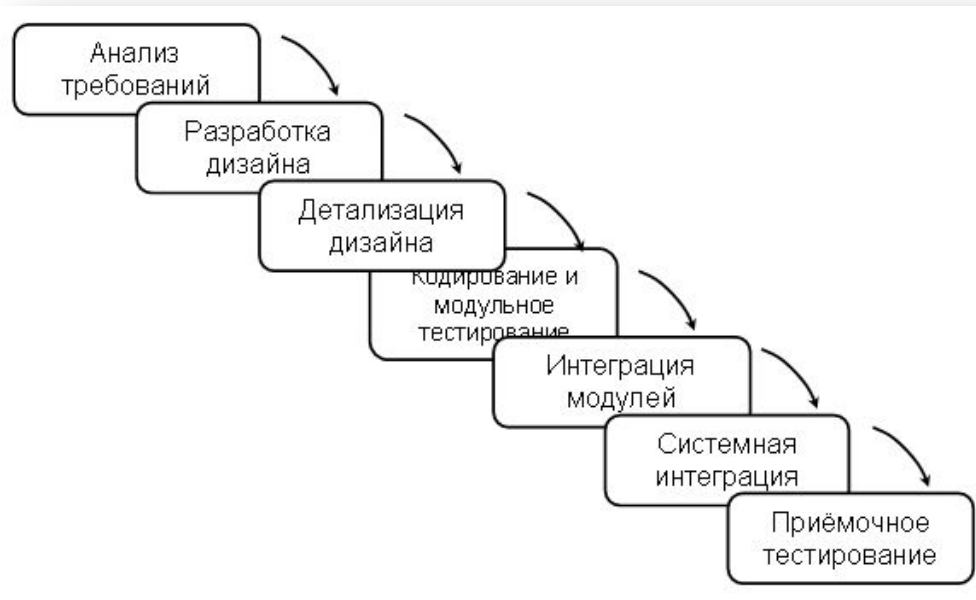
## Модели жизненного цикла ПО, v-образная модель



- Простая.
- На каждой фазе свои очевидные артефакты.
- Хорошо работает для мелких проектов.

- Негибкая.
- Нет раннего прототипирования.
- Неочевидны решения проблем, обнаруженных на поздних стадиях.

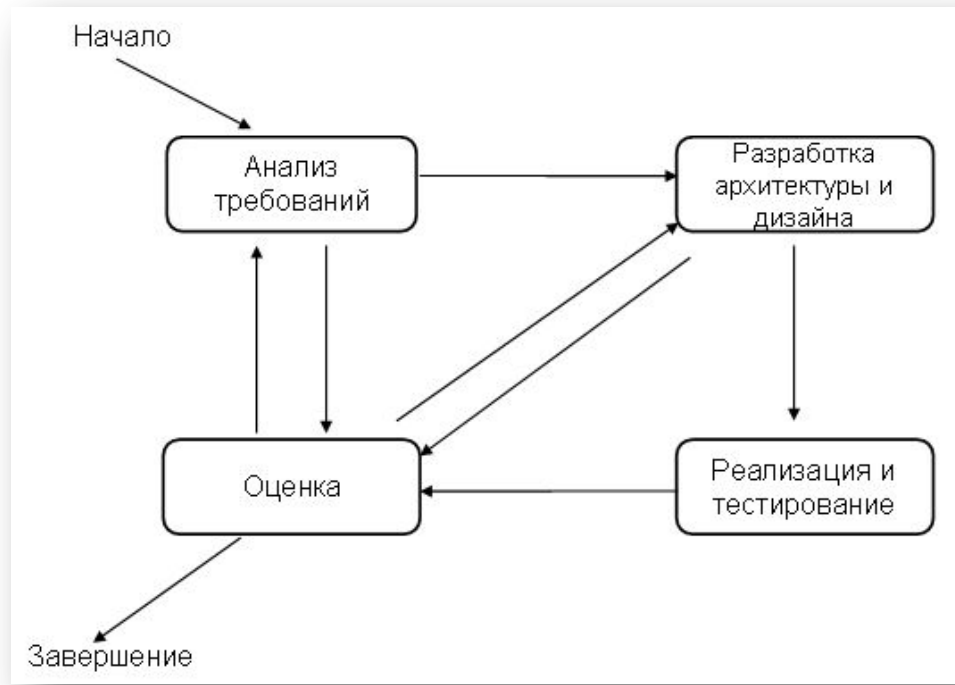
## Модели жизненного цикла ПО, водопадная модель



- Простая.
- Пошаговая.
- Очевидные артефакты и действия на всех стадиях.

- Неадаптивная.
- Нет раннего прототипирования.
- Сложно управлять рисками.
- Неочевидны решения проблем, обнаруженных на поздних стадиях.
- Не подходит для сложных проектов.

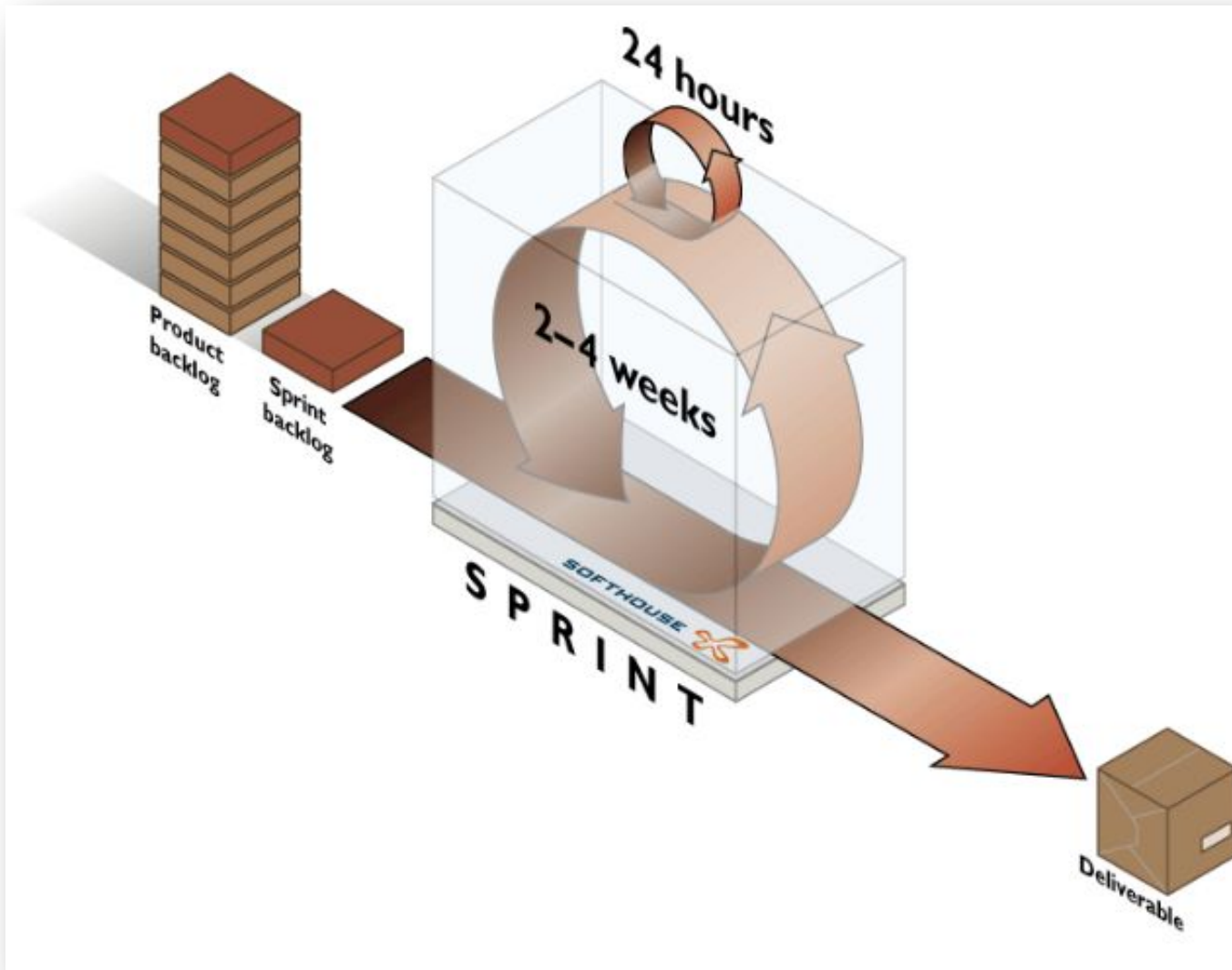
## Модели жизненного цикла ПО, итерационная модель



- Много анализа требований.
- Подходит для больших важных проектов.
- Раннее прототипирование, ранние поставки продукта.

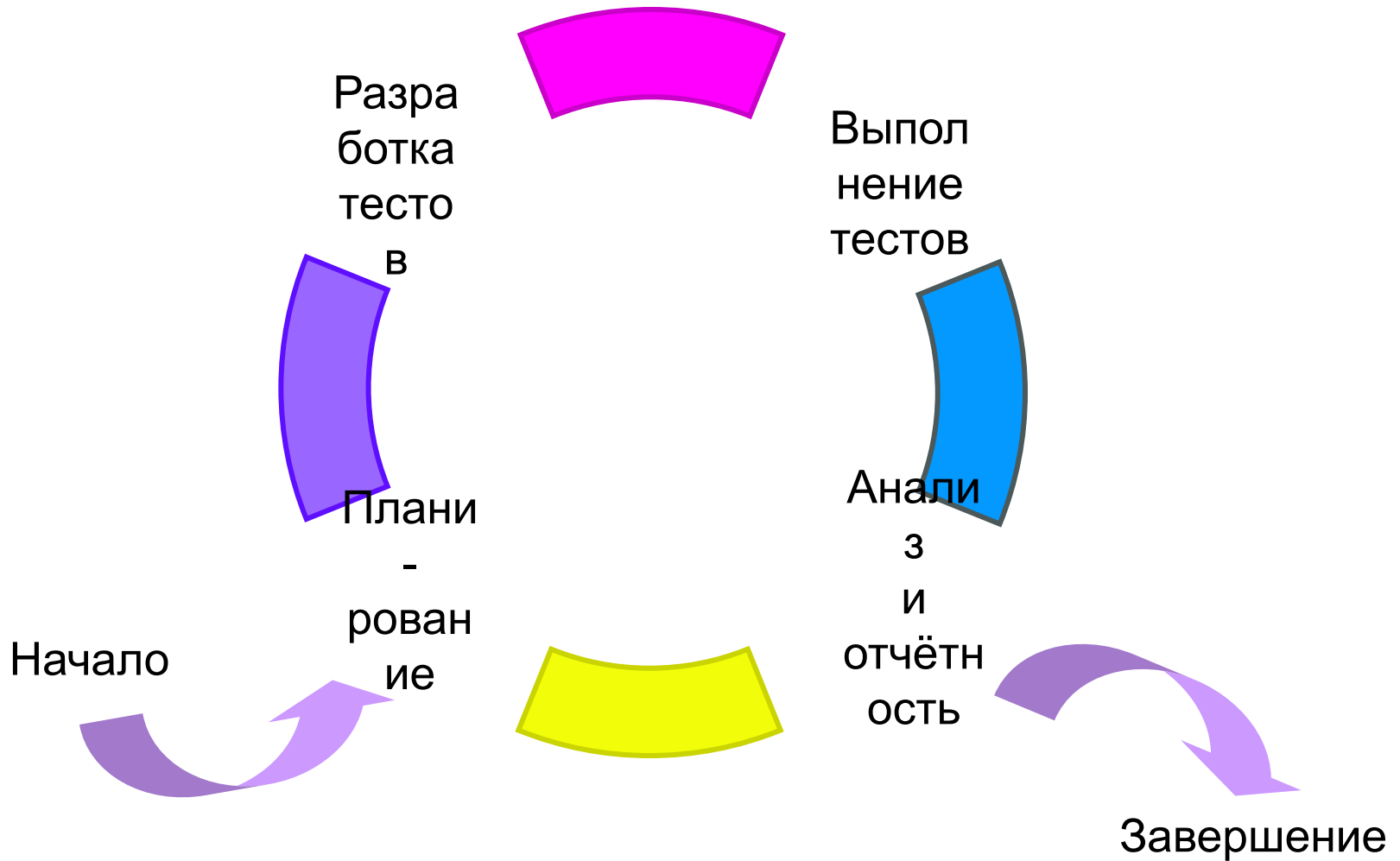
- Дорогая.
- Не работает для мелких проектов.

# Модели жизненного цикла ПО, гибкие методологии: Scrum, Agile и т.п.



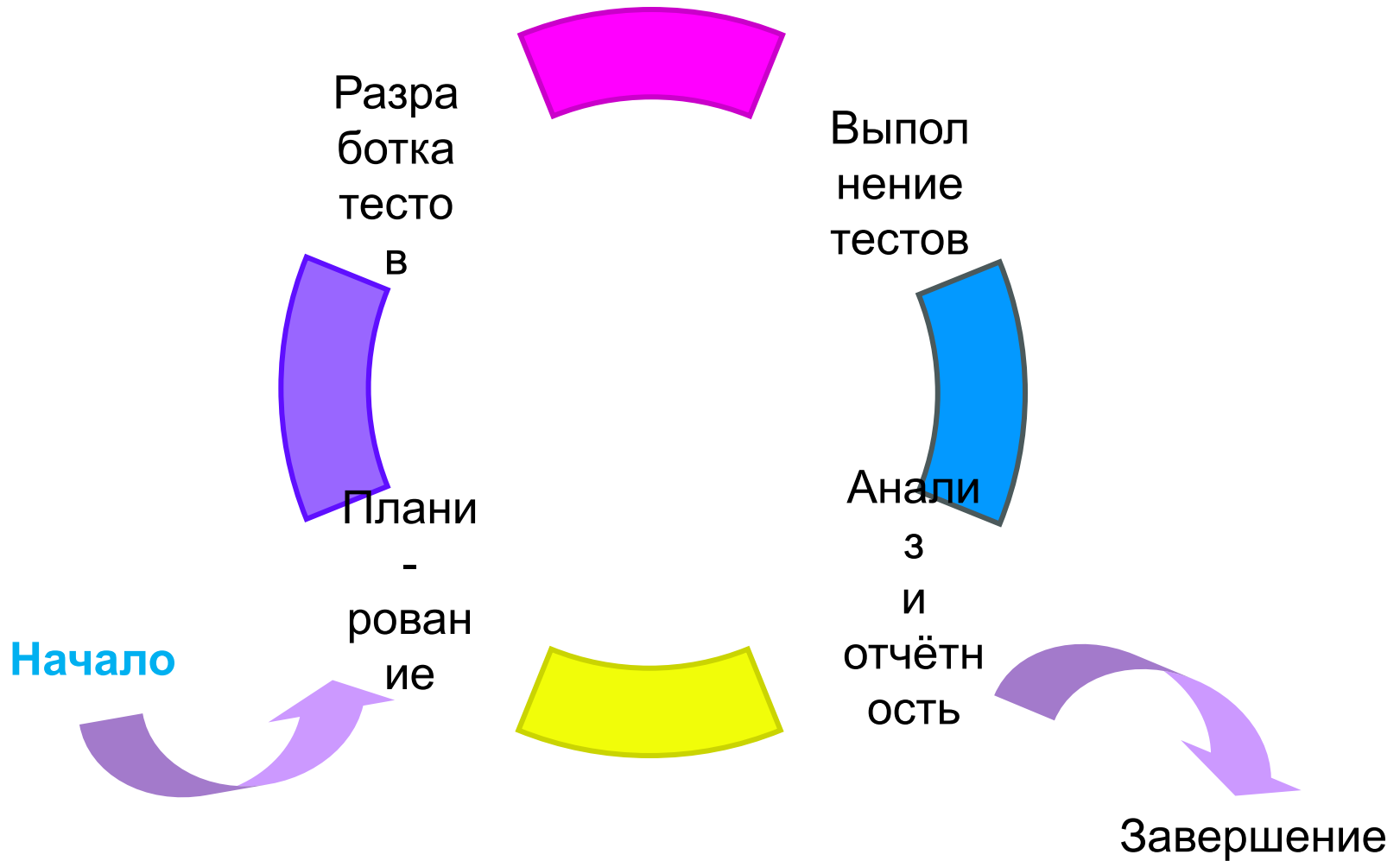
# ЖИЗНЕННЫЙ ЦИКЛ ТЕСТИРОВАНИЯ

# Жизненный цикл тестирования





# Жизненный цикл тестирования, начало



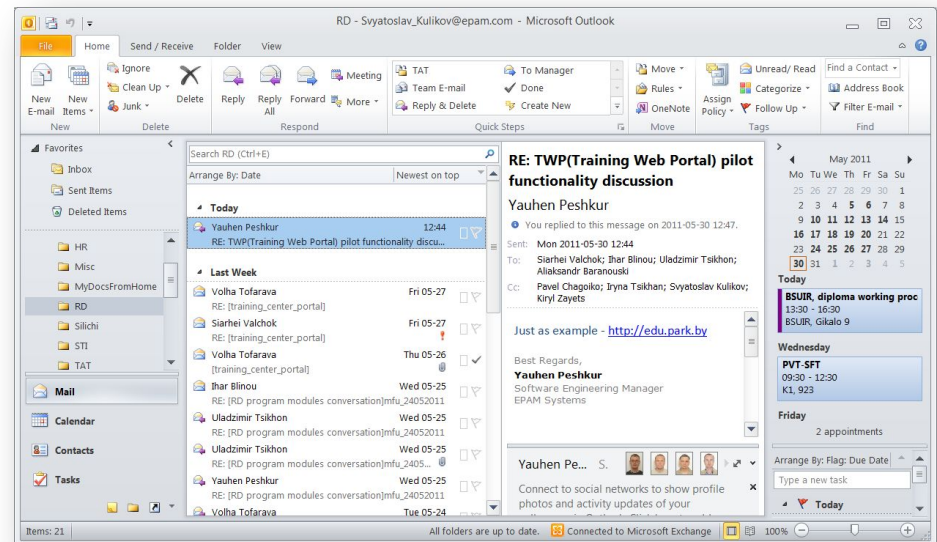
# Жизненный цикл тестирования, начало

## Действия:

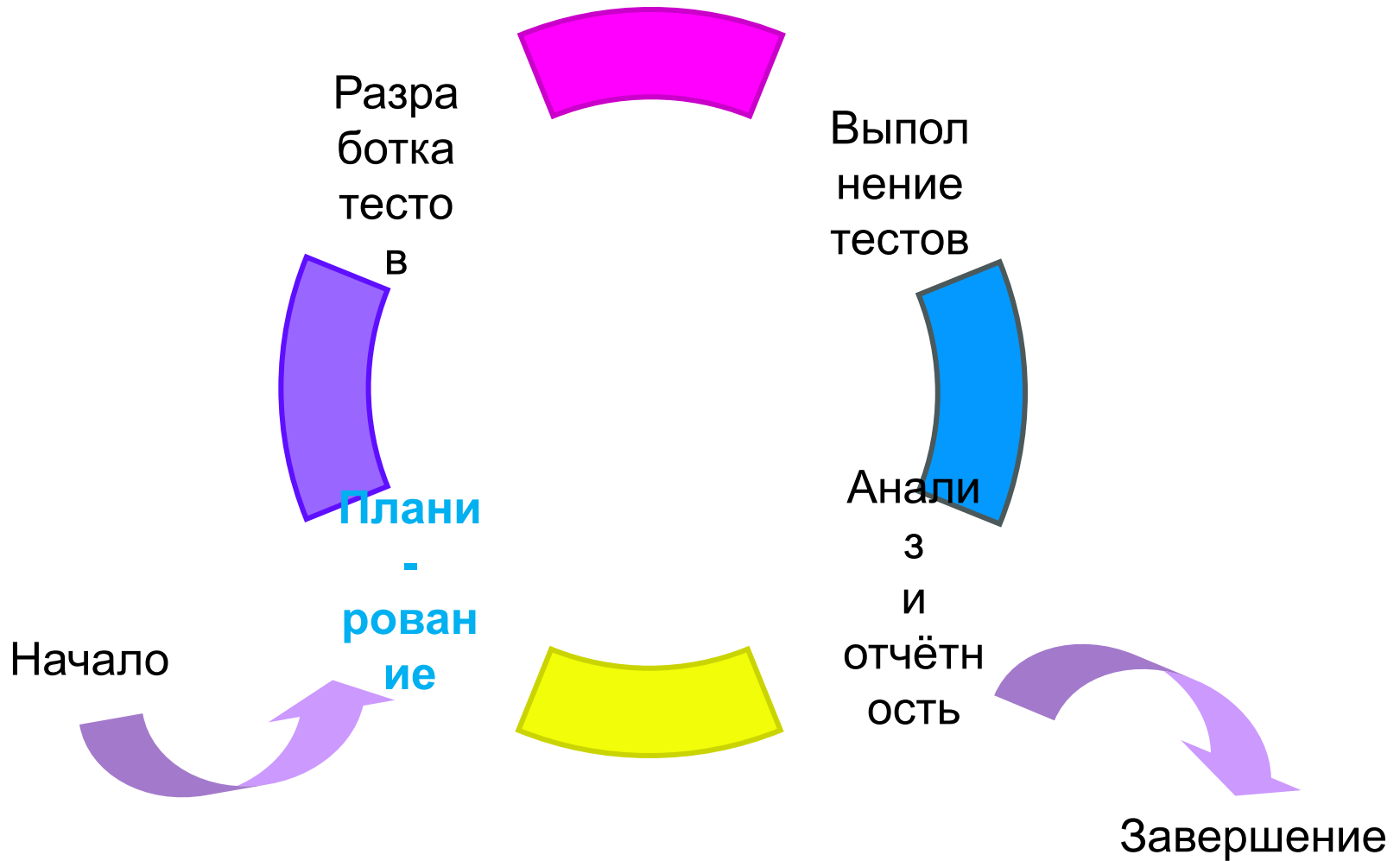
- Краткие обсуждения.
- Распределение обязанностей.
- Изучение списка требований.

## Артефакты:

- Запросы на выделение ресурсов.
- Письма с заданиями.
- Отчёты об анализе требований.



# Жизненный цикл тестирования, планирование



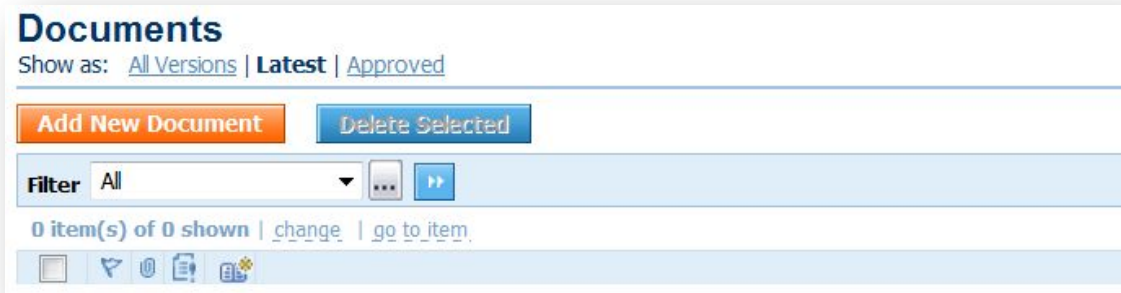
# Жизненный цикл тестирования, планирование

## Действия:

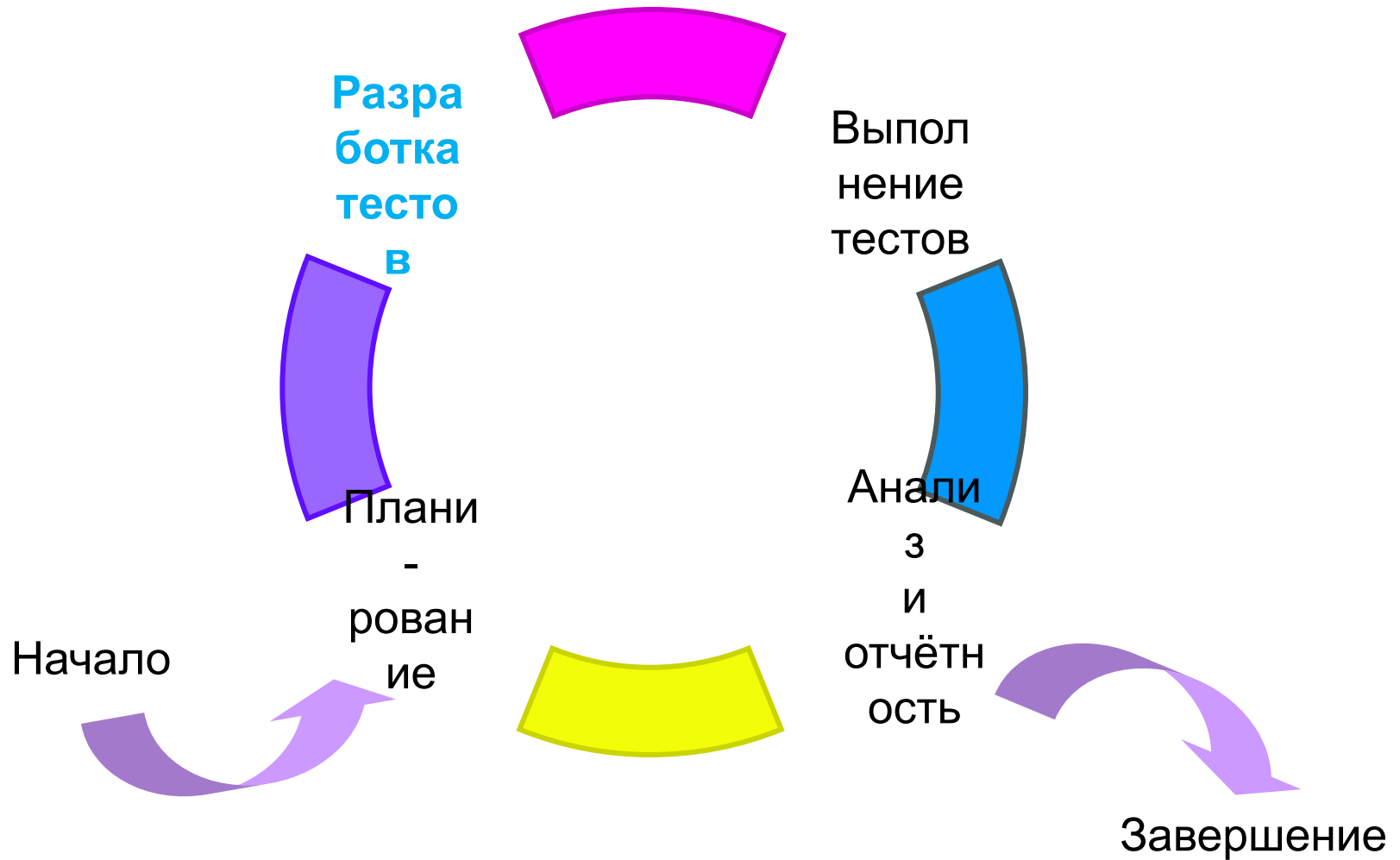
- Глубокое изучение требований к продукту.
- Определение и обсуждение рисков.
- Формирование, утверждение и публикация плана тестирования.
- Подготовка тестового окружения.

## Артефакты:

- План тестирования.
- Запросы на выделение ресурсов.



# Жизненный цикл тестирования, разработка тестов



# Жизненный цикл тестирования, разработка тестов

## Действия:

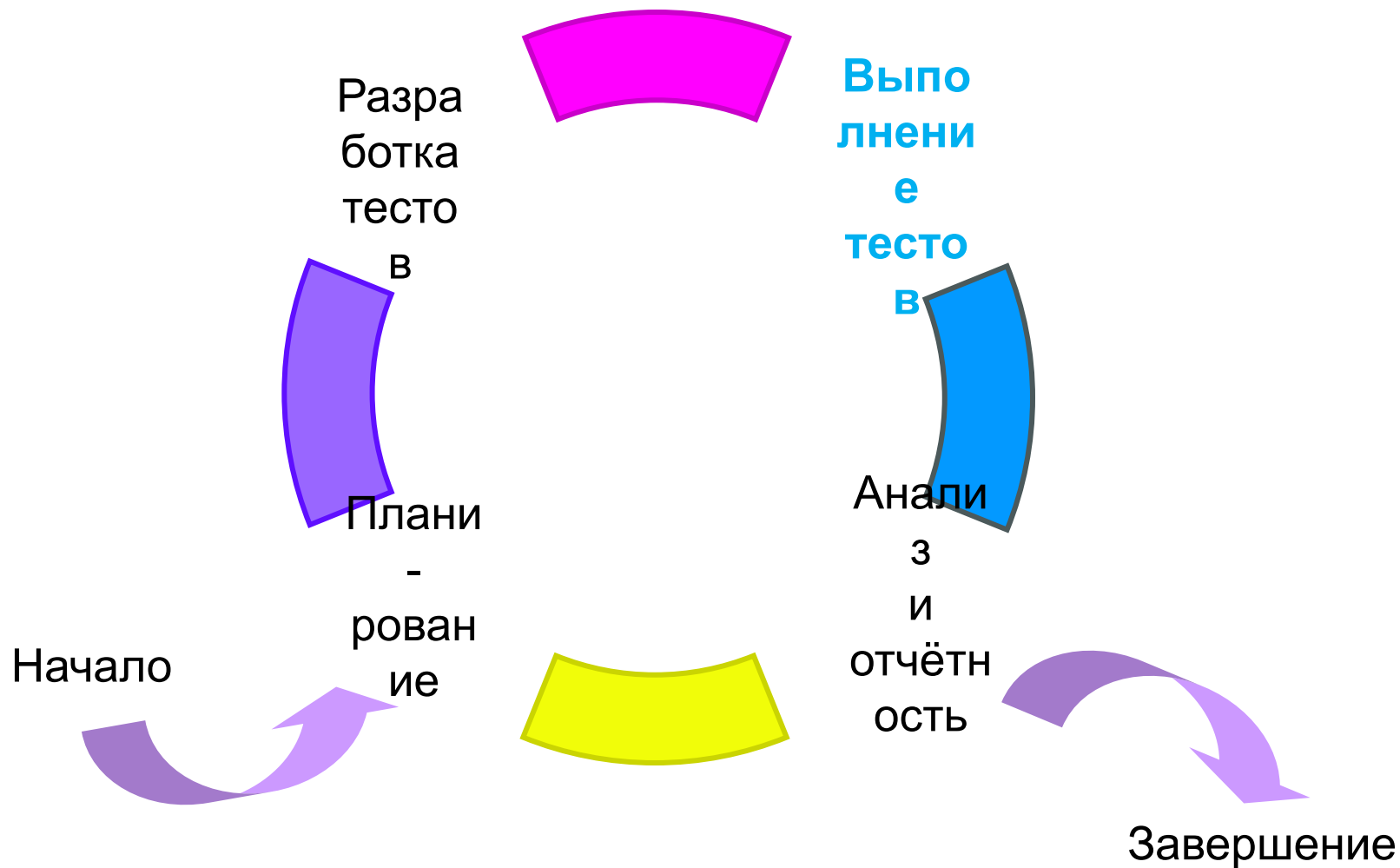
- Разработка тестовых случаев и тестовых сценариев.
- Разработка скриптов для автоматизированного тестирования.



## Артефакты:

- Тестовые случаи.
- Тестовые сценарии.
- Скрипты для автоматизированного тестирования.

# Жизненный цикл тестирования, выполнение тестов



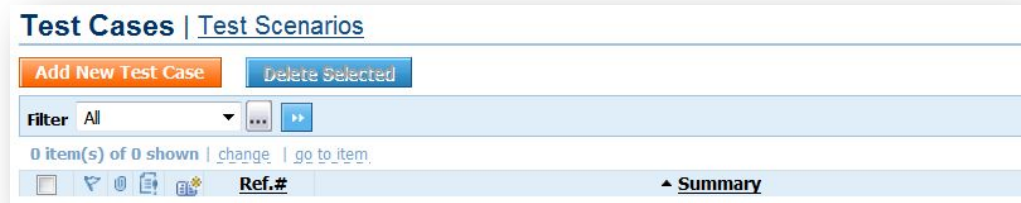
## Жизненный цикл тестирования, выполнение тестов

### Действия:

- Получение уведомления о выходе билда.
- Изучение сопроводительной документации.
- Установка билда.
- Запуск смоук-теста и принятие решения о дальнейшем тестировании.
- Тест критического пути и расширенный тест.
- Написание отчётов об ошибках.

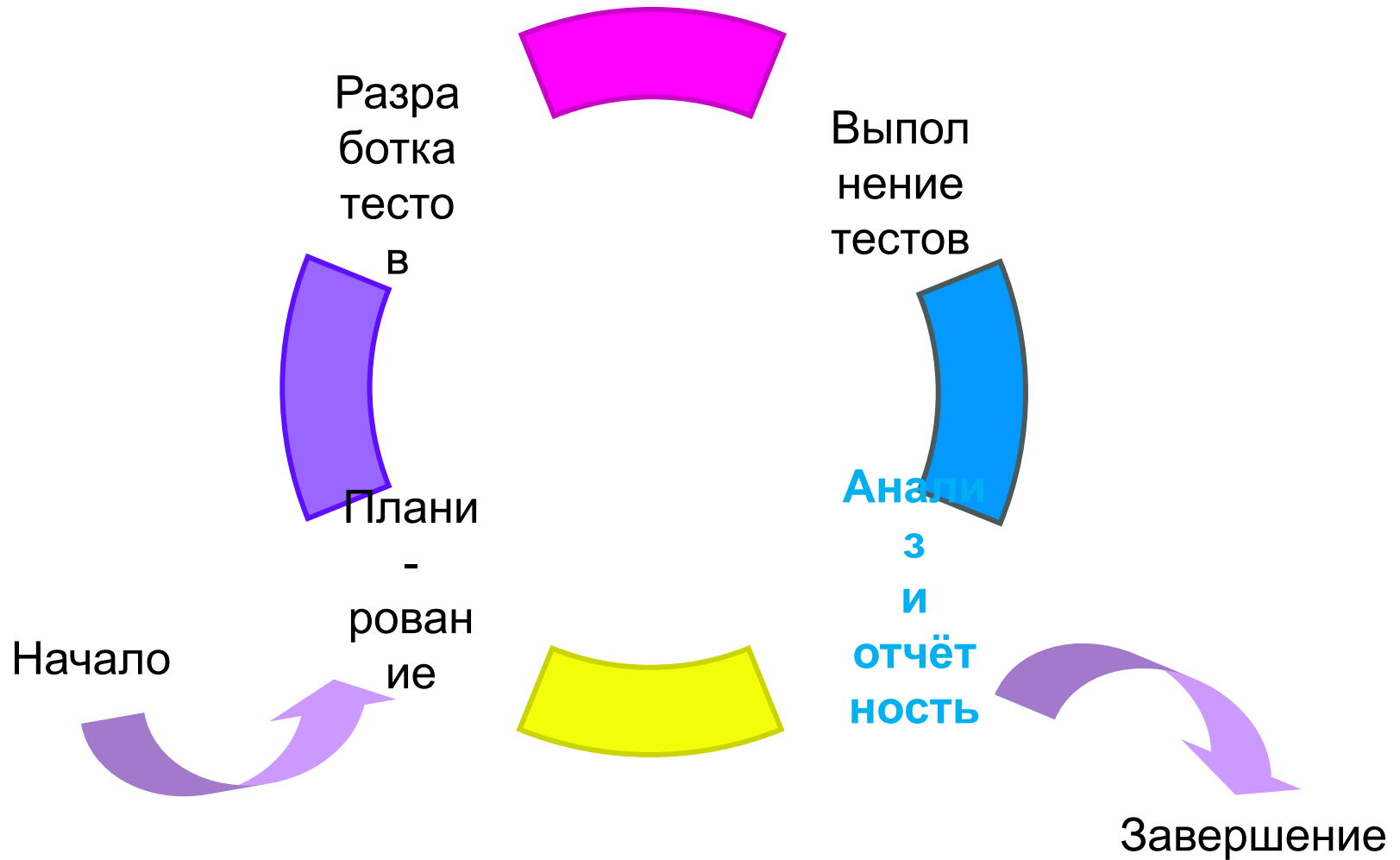
### Артефакты:

- Уведомления.
- Отчёты об ошибках.
- Тесты.
- Скрипты для автоматизированного тестирования.





# Жизненный цикл тестирования, анализ и отчётность



# Жизненный цикл тестирования, анализ и отчётность

## Действия:

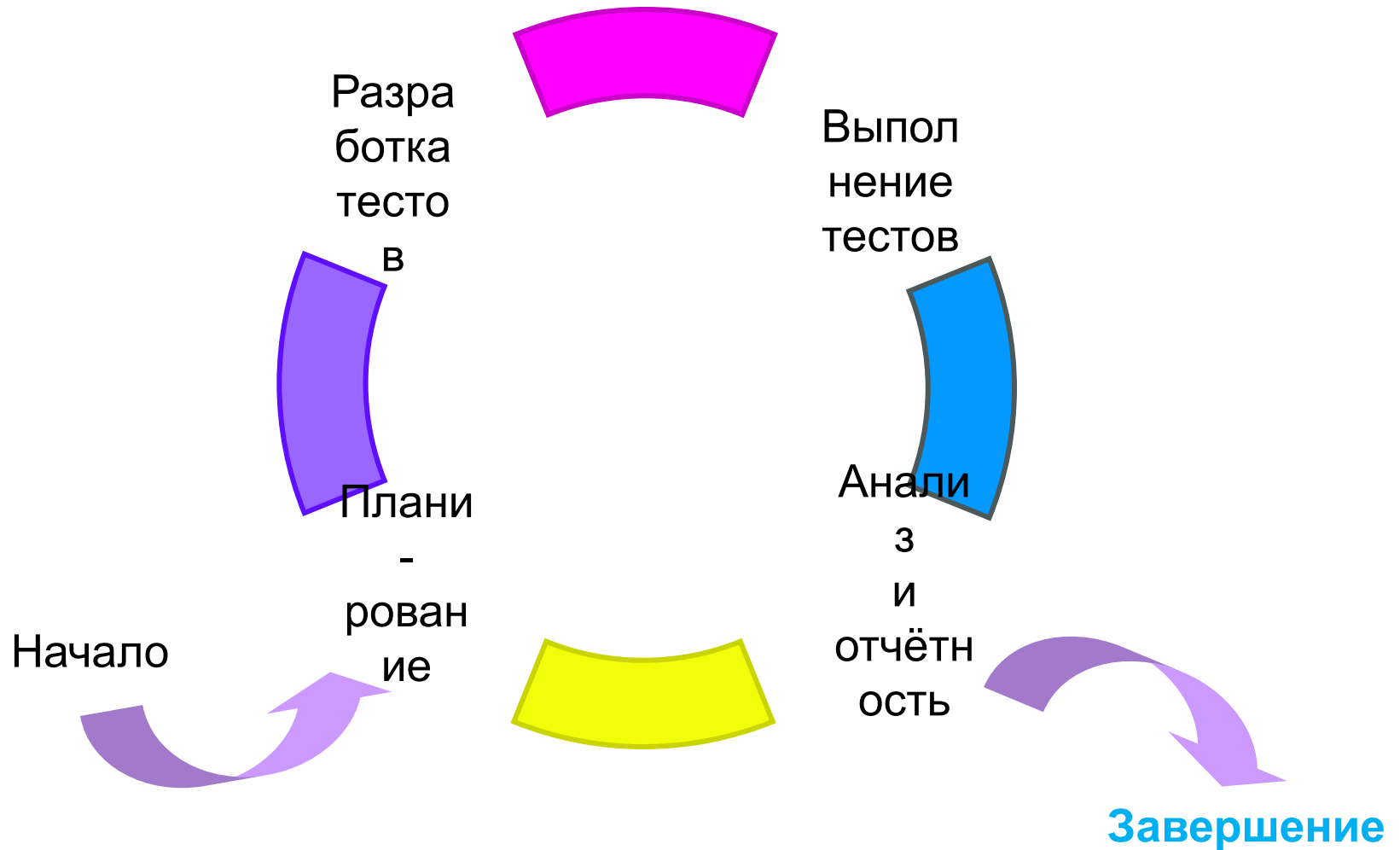
- Оценка качества продукта.
- Использование метрик.
- Уведомление руководства.
- Написание отчёта о результатах тестирования.

## Артефакты:

- Метрики.
- Отчёт о результатах тестирования.

				Ref.#	Summary
<input type="checkbox"/>					"Add new Address" form do not modifies by cursor guidance after adding one A
<input type="checkbox"/>					"Invalid schedule duration" logs can't be opened
<input type="checkbox"/>					'Login to ATS', 'Add/Edit User' pages: text fields are validated differently from sp
<input type="checkbox"/>					'View Topic' page: GUI errors
<input type="checkbox"/>					Account Information: Unnecessary message appears after changing account ty
<input type="checkbox"/>					Admin, User DB: User's subscription period is incorrectly displayed, days and m
<input type="checkbox"/>					After adding new country user can't add new city until he leaves Add/Edit Orgar
<input type="checkbox"/>					After creating country/region/city by student user is redirected to empty page
<input type="checkbox"/>					After creating new city page Add/Edit training group is displayed incorrectly
<input type="checkbox"/>					After filling in all necessary fields for trainer and add teaching preferences all er
<input type="checkbox"/>					Auswertung2 doesn't work
<input type="checkbox"/>					Auswertung3: Does not function
<input type="checkbox"/>					Calendar isn't fully visible on Test List page
<input type="checkbox"/>					CC info screen: if fields for billing address not filled and then checked autofill - e
<input type="checkbox"/>					Checkboxes are not cleaned on 'List of Tets' and 'List of Probationers' pages

# Жизненный цикл тестирования, завершение



## Жизненный цикл тестирования, завершение

### Действия:

- Рекомендация билда к выпуску.
- Финальная оценка качества продукта и процесса его разработки.
- Организация финального собрания проектной группы.

### Артефакты:

- Итоговый отчёт о результатах тестирования.
- Отчёт о финальном собрании.



Project <ATS>

### Test Results Report example

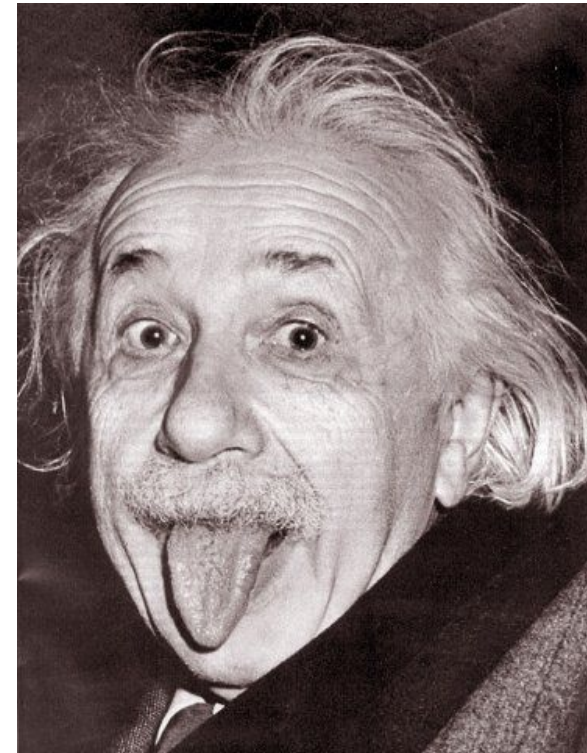
Project-related Documentation

Background	Some metrics are needed to estimate the testing process efficiency and current state of given application.
Purpose	To represent the testing results over a period of time covered in report.
Scope	Testing process description, Testing results, Overall bugs status, Recommendations.
Audience	Management staff, QA team, project team.
File	TRR_Example.doc

# ТЕХНИЧЕСКИЕ НАВЫКИ И ЛИЧНОСТНЫЕ КАЧЕСТВА ТЕСТИРОВЩИКА

## Технические навыки, необходимые тестировщику

- Знание **иностраннЫХ языков**.
- **Программирование:** C/C++/C#, Java, PHP, Object Pascal, Visual Basic, JavaScript, HTML, .NET, **«веб-разработка вообще»**.
- Администрирование СУБД: Oracle, MS SQL, MySQL.
- Администрирование ОС: Windows, Sun Solaris, HP-UX, Free-BSD, Linux.
- Сетевое администрирование: TCP/IP, IPX/SPX, NetBIOS.
- Автоматизированное тестирование: Silk\*, Rational\*, Mercury Interactive \*, JUnit, HTTP/HTML-Unit.



## Личностные качества хорошего тестировщика



- Повышенная ответственность.
- Хорошие коммуникативные навыки.
- Способность ясно, быстро, чётко выражать свои мысли.
- Исполнительность.
- Терпение, усидчивость, внимательность к деталям, наблюдательность.
- Гибкое мышление, хорошая способность к обучению.
- Хорошее **абстрактное и аналитическое мышление**.
- Способность ставить нестандартные эксперименты.
- Склонность к исследовательской деятельности.

# ОСНОВНАЯ ТЕРМИНОЛОГИЯ ТЕСТИРОВАНИЯ





**Обеспечение качества**  
(«профилактика» и «здоровый образ жизни»).

## Контроль качества

(«а всё ли идёт так, как надо?»,  
«или есть проблемы?»)



Фактически, «тестирование ПО» – это «диагностика» и «помощь в лечении»

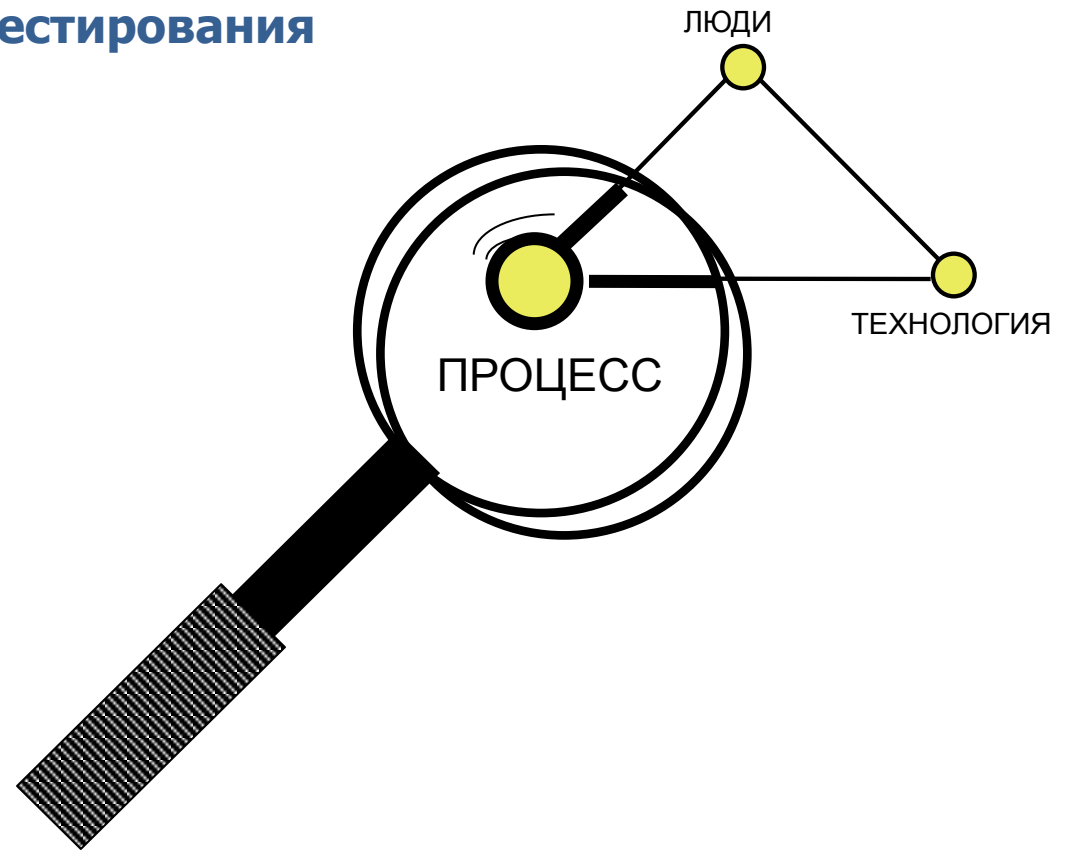
программного средства как такового и всего проекта в целом.



**Тестирование программного обеспечения** (software testing) – процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта.



## Основная терминология тестирования



Даже лучшие работники не смогут выполнить поставленную задачу, если процесс не организован.

**Дефект** (баг, глюк; defect, bug) – любое несоответствие фактического и ожидаемого результата (согласно требованиям или здравому смыслу).





**Ожидаемый результат** (expected result) – такое поведение программного средства, которое мы ожидаем в ответ на наши действия.

## Основная терминология тестирования

**Тест-план** (test plan) – часть проектной документации, описывающая и регламентирующая процесс тестирования.



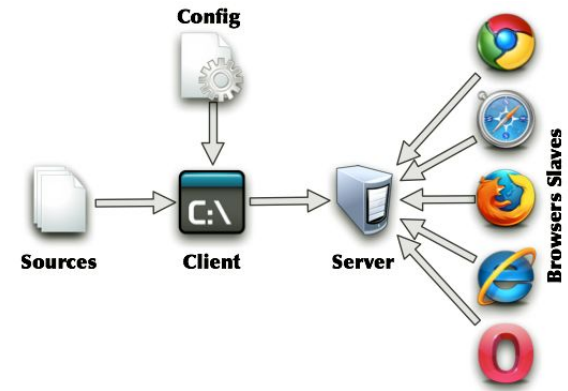


**Чек-лист** (check-list) –  
набор **идей тестов**.



## Основная терминология тестирования

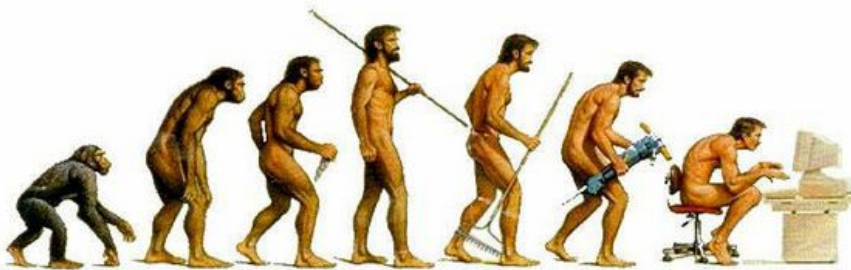
**Тест-кейс** (test case) – набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.



**Тестовый сценарий, тест-сьют** (test scenario, test-suite) – набор тест-кейсов, собранных в группу (последовательность) **для достижения некоторой цели.**



**Билд («сборка»)** (build) – промежуточная версия программного средства (финальный билд часто называют релизом (release)).



**Качество** (quality) – показатель степени соответствия продукта его требованиям.



**Метрика качества** (quality metric) – числовое значение некоторого показателя качества.

*Может определяться расчётным способом или по некоторой формуле.*

$$\begin{aligned} J_{\lambda}(x_2, y_2, s_2) &= \iint I_{\lambda}(x_2, y_2) \cdot \left| m_{\lambda} \left( \frac{x_2 - x_0}{\lambda \cdot s_2}, \frac{y_2 - y_0}{\lambda \cdot s_2} \right) \right|^2 dx_0 dy_0 = \\ &= I_{\lambda}(x_2, y_2) \otimes \left| m_{\lambda} \left( \frac{x_2}{\lambda \cdot s_2}, \frac{y_2}{\lambda \cdot s_2} \right) \right|^2 \end{aligned}$$

# СПАСИБО ЗА ВНИМАНИЕ!

## ВОПРОСЫ?

Введение в тестирование

**Author: Svyatoslav Kulikov**  
**Training And Education Manager**  
**svyatoslav\_kulikov@epam.com**