

**Жизненный цикл  
разработки ПО.  
Управление проектами.  
Стандарты тестирования.  
Виды тестирования.**



# ПЛАН ЗАНЯТИЯ

- Знакомство с жизненным циклом разработки ПО
- Управление проектами
  - Каскадная модель
  - V-модель
  - Инкрементная модель
  - итеративная модель
  - спиральная модель
  - Agile методология
- Понятие качества
- Стандарты тестирования

1

# Повторение изученного

# Повторение

## Вопрос 1

**Что из перечисленного  
входит в обязанности QA?**

Разработка технической документации

Разработка тестовой документации

Исправление ошибок

Локализация ошибок

# Повторение

## Вопрос 1

Что из перечисленного  
входит в обязанности QA?



Разработка технической документации

Разработка тестовой документации

Исправление ошибок



Локализация ошибок

# 2

# ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПО

# Жизненный цикл разработки программного обеспечения (SDLC — Software development life cycle)

это непрерывный процесс, который начинается с момента принятия решения о создании программного обеспечения и заканчивается в момент его выпуска и дальнейшей поддержки ПО.

# Жизненный цикл ПО



**Security Development Lifecycle**



# Оценка, планирование, аналитика

## Evaluation, planning, analytics

“Идея” принимает конкретный вид

Бизнес-аналитики (БА) прорабатывают полученную информацию

БА детализируют информацию

БА преобразовывают идею в технические требования к системе  
(Software Requirement Specification - SRS)



# Проектирование и дизайн

## Design

Создается “скелет” проекта, основываясь на требованиях

Контекст проекта и организации определяют используемые технологии, инструменты, рабочие процессы, взаимосвязи между разными частями проекта, структуры баз данных, потоки данных и т.д.

В итоге определяется спецификация по дизайну (Design Document Specification - DDS) с описанием того, что и как нужно делать с технической точки зрения

# Разработка Development

Разработчики знакомятся с требованиями, архитектурой и начинают писать продукт

Первая разработка до тех пор, пока не появится на выходе запускаемое ПО (build), которое приблизительно похоже на заявленный проект и готовое к тестированию



# Тестирование

## Testing

Тестирование продукта/модуля/компонента с последующим предоставлением информации о результатах в виде:

баг-репортов (bug-report)

метрики  
(metrics)



# Развертывание (внедрение)

## Deployment

Иногда развертывание продукта происходит поэтапно в соответствии с бизнес-стратегией организации.

Он может быть сначала выпущен в ограниченном сегменте и протестирован в реальной бизнес-среде, а затем выпущен с улучшениями



# Поддержка

## Support

После внедрения проекта наступает его поддержка. На данном этапе обрабатываются и принимаются обращения пользователей, дорабатывается продукт.



# Экспресс-опрос

- **Вопрос 1.**

В каком виде может предоставляться информация о результатах после тестирования?

- **Вопрос 2.**

Назовите стадии разработки ПО?



# ВОПРОСЫ



# 3

## УПРАВЛЕНИЕ ПРОЕКТАМИ

# Модель жизненного цикла ПО

это описание, как проходит процесс разработки.

# Модели жизненного цикла ПО

**Модель** — это, прежде всего, этапы разработки и их последовательность.

**Существующие модели:**

- Каскадная модель, или «водопад»
- V-модель
- Инкрементная модель
- Итеративная модель
- Спиральная модель



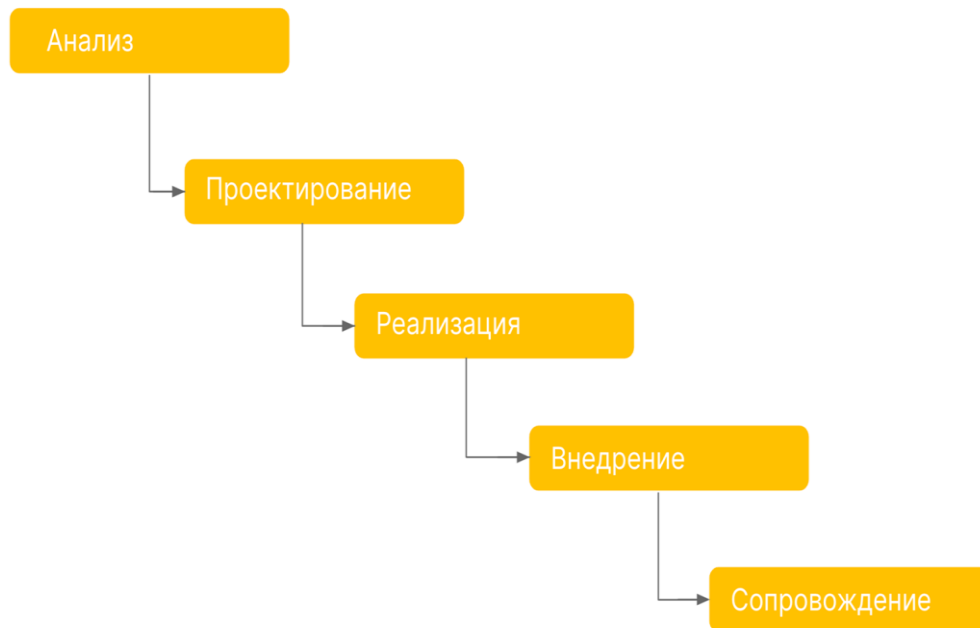
# Каскадная модель, или «водопад»

## Преимущества:

1. Проста и понятна заказчикам.
2. Проста и удобна в применении.
3. На каждом этапе формируется законченный набор проектной документации.
4. Планирование сроков завершения всех работ и соответствующие затраты.
5. Достаточно точная и полная документация.

## Недостатки:

1. Позднее получение результатов.
2. Ошибки и недоработки сложнее исправить, так как процесс возврата на предыдущий этап практически невозможен.
3. Заказчик не допускается до разработки и тестирования.



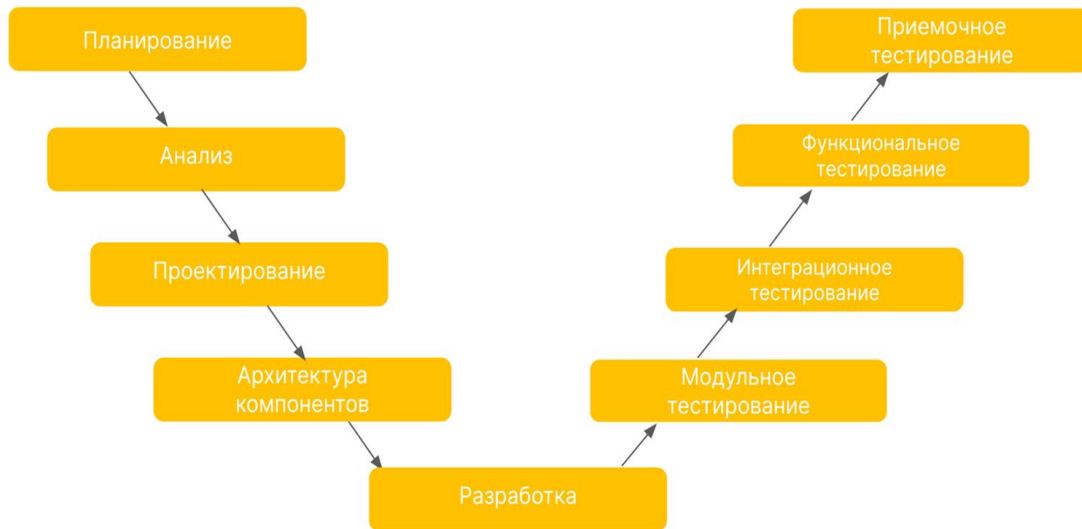
# V-модель

## Преимущества

1. На каждом этапе формируется законченный набор проектной документации.
2. Тестирование каждого этапа разработки.
3. Контролируемый каждый этап разработки ПО.
4. Сроки и бюджет проекта заранее известны.

## Недостатки

1. Недоработки сложнее исправить, так как процесс возврата на предыдущий этап практически невозможен.



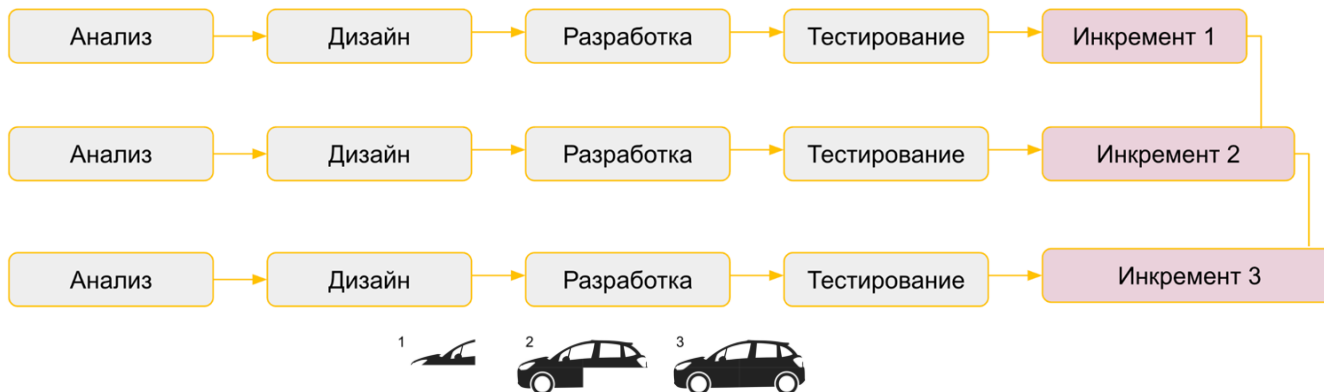
# Инкрементная модель

## Преимущества

1. Программное обеспечение создается быстрее и раньше поставляется пользователям.
2. Не так затратно вносить изменения в программное обеспечение на стадии разработки, как например в водопадной модели.
3. Управление рисками.

## Недостатки

1. Разработка требований до начала стадии разработки программного обеспечения.



# Итеративная модель

## Преимущества

1. Раннее обнаружение дефектов и быстрая обратная связь от пользователей.
2. Быстрый выпуск минимально работающего продукта (MVP) на рынок.

## Недостатки

1. Изменяющиеся требования влекут за собой увеличение бюджета и сроков разработки продукта.



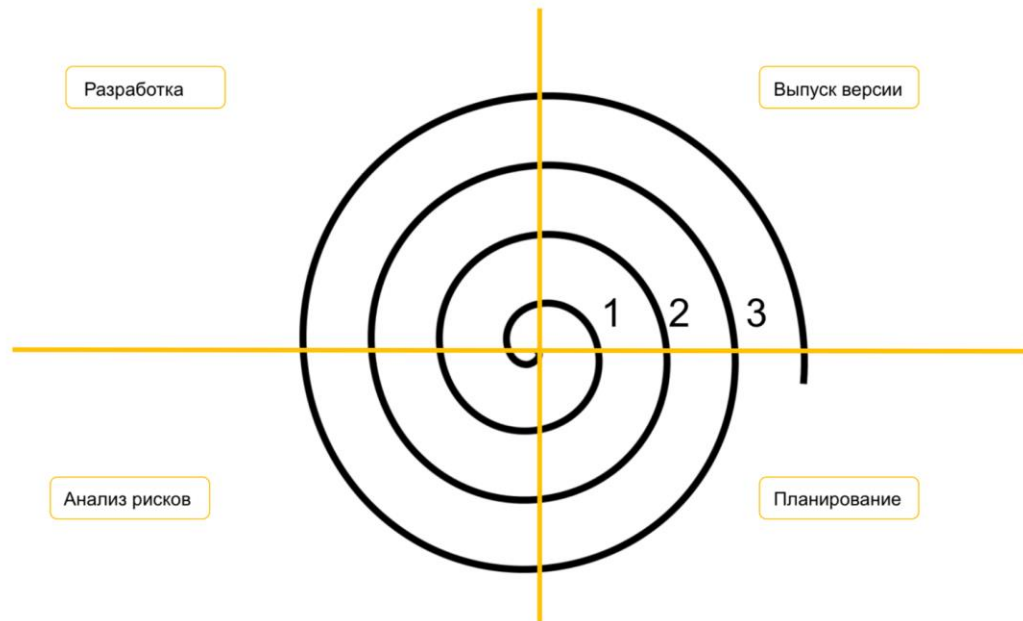
# Спиральная модель

## Преимущества

1. Анализ рисков.

## Недостатки

1. Изменяющиеся требования влекут за собой увеличение бюджета и сроков разработки продукта.





# Экспресс-опрос

- Вопрос 1.

Как Вы своими словами опишете смысл V-модели?

- Вопрос 2.

Чем отличается Каскадная модель от Итеративной модели?



# Методология разработки

набор методов по управлению разработкой ПО, подборка практических правил и техник разработки.

Или по-другому, методология — это детализированный набор правил, практик и принципов, применяемый при реализации той или иной модели.

# Гибкие методологии разработки

**Agile** — не модель и не методология. Agile объединяет в себе разные гибкие методологии разработки.

Это философия разработки, на которую опираются гибкие методологии.

Существующие методологии разработки:

- **SCRUM**
- **Kanban**
- **Lean**
- **и другие**



# Agile

Гибкая методология разработки

ЛЮДИ И ОБЩЕНИЕ

РАБОТАЮЩИЙ ПРОДУКТ

СОТРУДНИЧЕСТВО С ЗАКАЗЧИКОМ

ГОТОВНОСТЬ К ИЗМЕНЕНИЯМ

ВАЖНЕЕ

ПРОЦЕССОВ И ИНСТРУМЕНТОВ

ИСЧЕРПЫВАЮЩЕЙ ДОКУМЕНТАЦИИ

СОСТАВЛЕНИЯ КОНТРАКТОВ

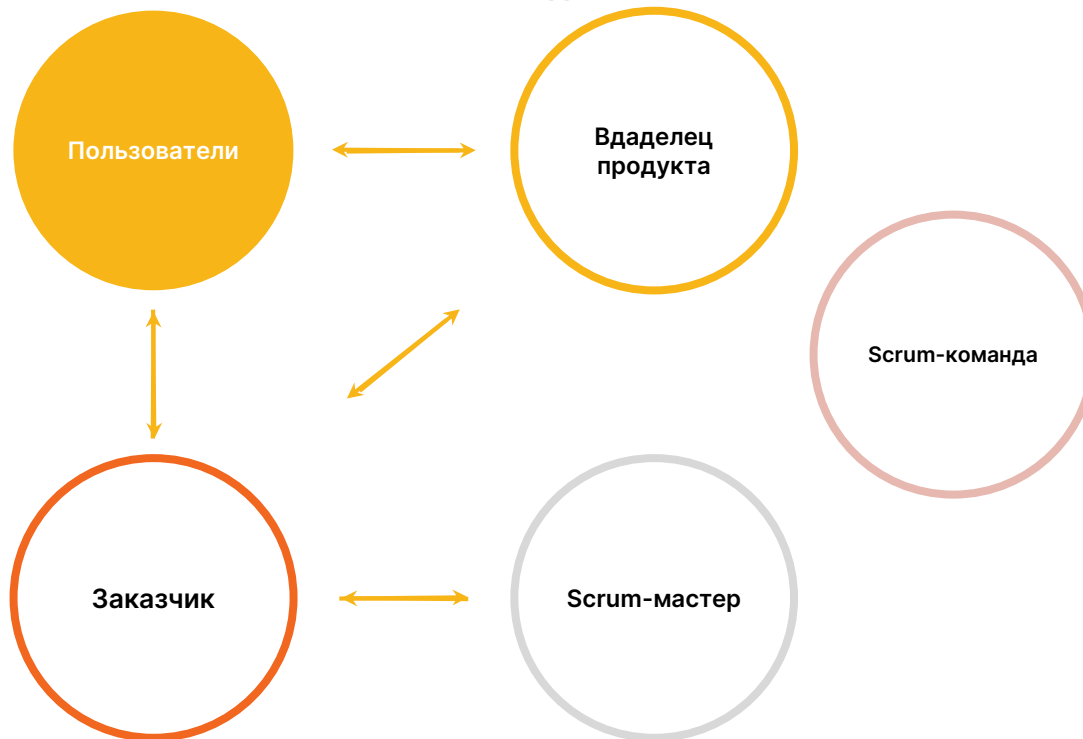
СЛЕДОВАНИЯ ПЛАНУ

# 12 принципов Agile

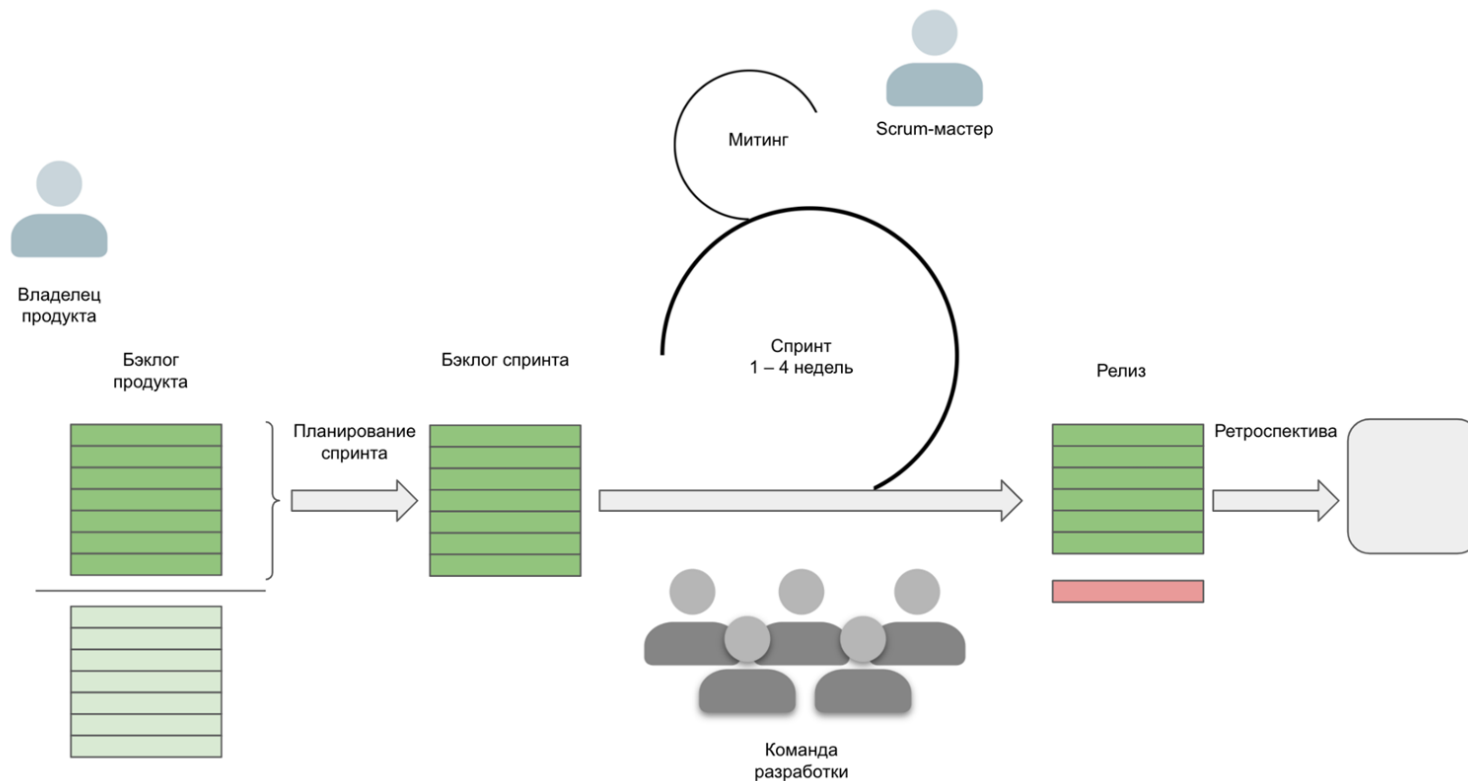
1. Задача высшего приоритета — регулярно и как можно раньше удовлетворять потребности заказчика, предоставляя ему программное обеспечение.
2. Учитывать, что требования могут измениться на любом этапе разработки. Если изменения быстро вносятся в проект, заказчик может получить конкурентные преимущества.
3. Выпускать версии готовой программы как можно чаще — с промежутком от двух недель до двух месяцев.
4. Ежедневно вместе работать над проектом — разработчикам и заказчикам.
5. Поручить работу мотивированным профессионалам. Обеспечить поддержку и условия, довериться им — и работа будет сделана.
6. Общаться напрямую — это самый эффективный способ взаимодействия внутри команды и вне ее.
7. Считать главным показателем прогресса работающий продукт.
8. Поддерживать постоянный ритм работы — касается и разработчиков, и заказчиков.
9. Уделять пристальное внимание техническому совершенству и качеству проектирования — это повышает гибкость проекта.
10. Минимизировать лишнюю работу.
11. Стремиться к самоорганизующейся команде — в ней рождаются наиболее эффективные и качественные решения.
12. Всем участникам команды — постоянно искать способы повышать эффективность работы.

# SCRUM

Основной состав Scrum-команды (обычно не более 10 человек):



# SCRUM



# SCRUM

Название спринта

- Test Project  
Software project
- Roadmap
- Backlog
- Board**
- Code
- Project pages
- Add shortcut
- Project settings

TP Sprint 1

Q \$ + Type ▾

TO DO 2 ISSUES

Реализовать профиль пользователя  
TP-1

Неактивна кнопка "подписаться" на карточке товара  
TP-4

IN PROGRESS 1 ISSUE

Не отображается бейдж "продано" на карточке проданного товара  
TP-3

DONE 1 ISSUE ✓

Изменить цвет кнопки "Купить"  
TP-2

+

Доска с задачами

⚡ ☆ ⌚ 5 days remaining Complete sprint ...

GROUP BY None ▾ Insights



# KANBAN



The screenshot displays a Kanban board within a software project management application. On the left, a sidebar lists navigation options: 'kanban' (Software project), 'Roadmap', 'Board' (selected), 'Code', 'Project pages', 'Add shortcut', and 'Project settings'. The main area is titled 'Projects / kanban' and 'KAN board'. It features three columns: 'TO DO 2 ISSUES', 'IN PROGRESS 1 ISSUE', and 'DONE ✓'. The 'TO DO' column contains two tasks: 'Реализация профиля пользователя' (KAN-6) and 'Поправить цвет шрифта' (KAN-8), each with a checkbox. The 'IN PROGRESS' column contains one task: 'Изменить цвет кнопки "Продать" на экране карточки товара' (KAN-7), also with a checkbox. The 'DONE' column is empty except for a link to 'See all Done issues'. A search bar and user avatars are at the top of the board. A 'GROUP BY' dropdown is set to 'None'. A yellow rounded rectangle highlights the board area, with an arrow pointing to it from the text 'Доска с задачами'.

kanban  
Software project

Roadmap

Board

Code

Project pages

Add shortcut

Project settings

Projects / kanban

KAN board

TO DO 2 ISSUES

Реализация профиля пользователя

✓ KAN-6

Поправить цвет шрифта

✓ KAN-8

+ Create issue

IN PROGRESS 1 ISSUE

Изменить цвет кнопки "Продать" на экране карточки товара

✓ KAN-7

DONE ✓

See all Done issues

GROUP BY None

Доска с задачами

# Экспресс-опрос

- Вопрос 1.

Как Вы поняли основные принципы Agile?

- Вопрос 2.

Назовите Scrum roles?



# ВОПРОСЫ

4

# ПОНЯТИЕ КАЧЕСТВА

# Качество программного обеспечения

способность программного продукта при заданных условиях удовлетворять установленным или предполагаемым потребностям.

# Качество ПО

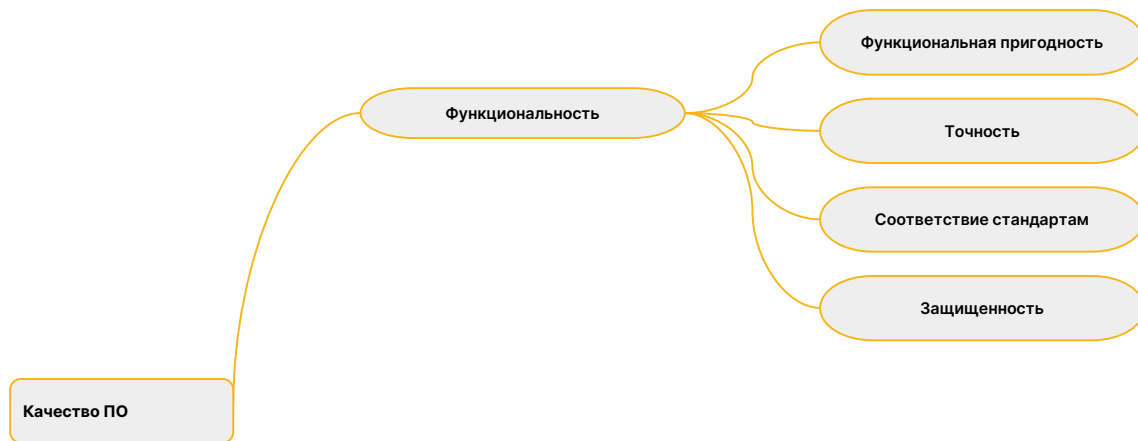
**Как мы можем проверить, что требования к продукту определены достаточно полно, описывают весь функционал?**

**Это можно сделать, проследив, все ли необходимые аспекты качества ПО отражены в них. Понятие качественного ПО соответствует представлению о том, что программа достаточно успешно справляется со всеми возложенными на нее задачами.**



# Качество ПО

Выделяют следующие характеристики и атрибуты качества ПО согласно стандарту ISO 9126:

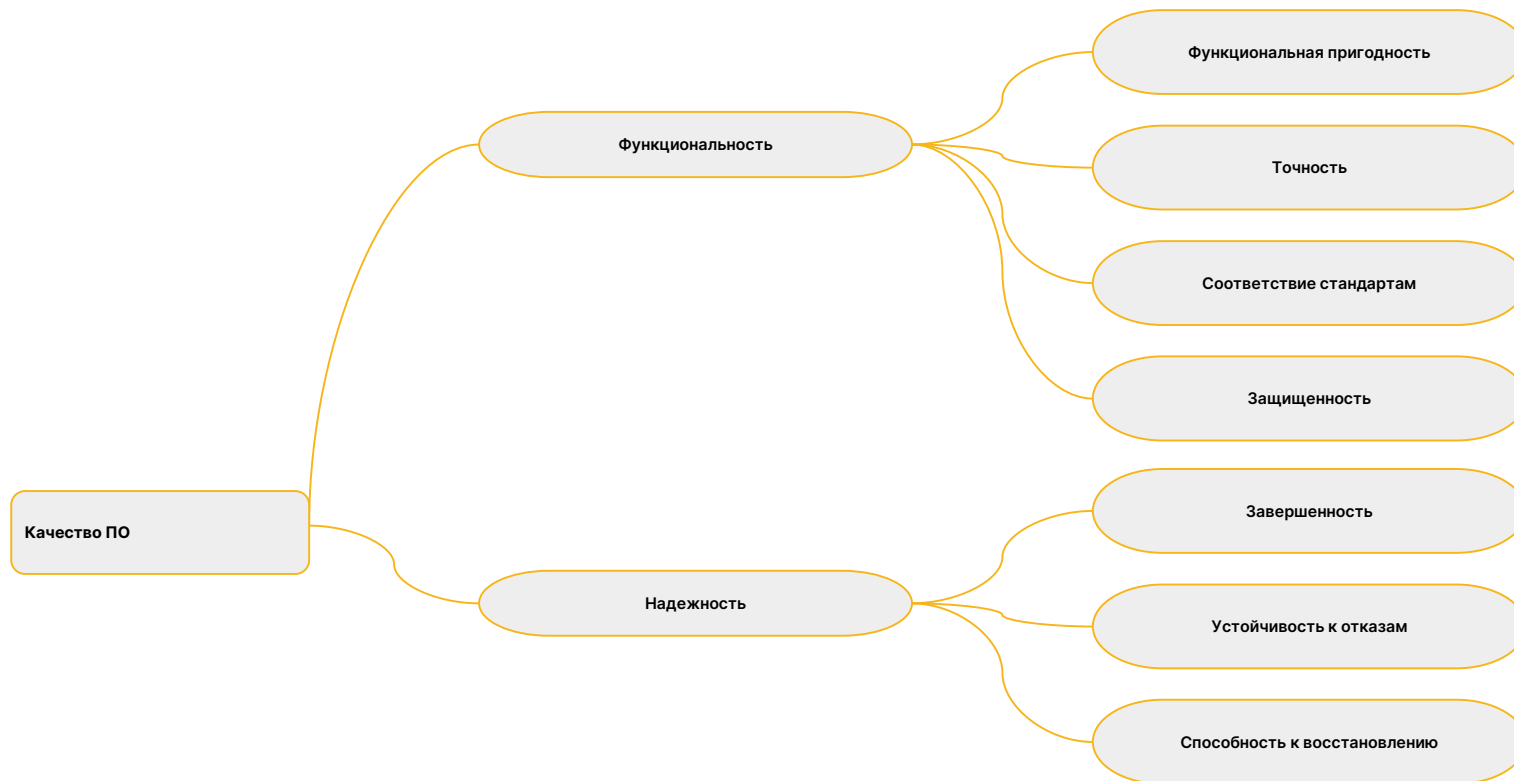


ISO/IEC 9126 — международный стандарт, определяющий оценочные характеристики качества программного обеспечения.

В 2011 году был заменён стандартом ISO/IEC 25010:2011. Состоит из 4 частей, описывающие следующие аспекты:

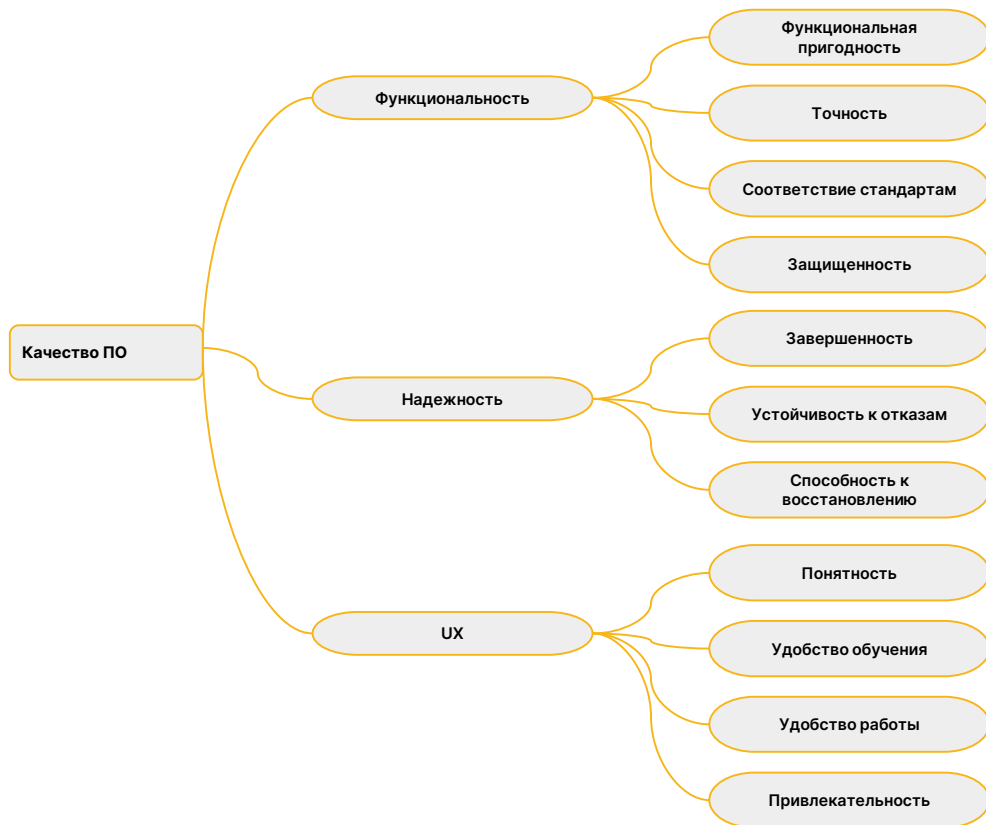
- модель качества;
- внешние метрики качества;
- внутренние метрики качества;
- метрики качества в использовании.

# Качество ПО



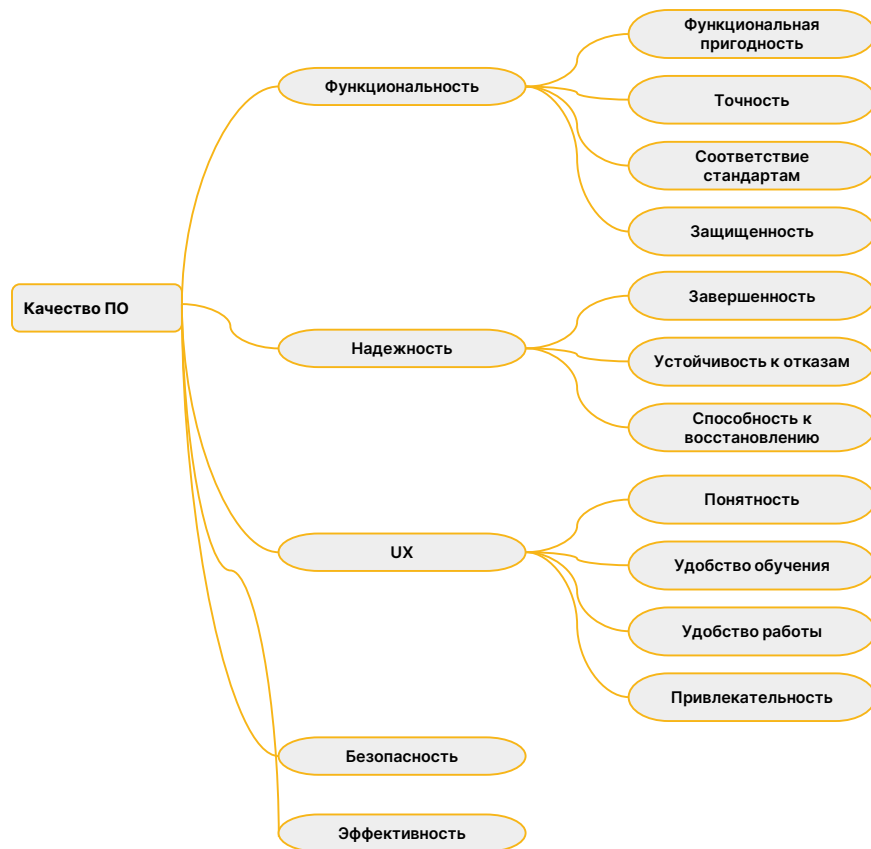


# Качество ПО

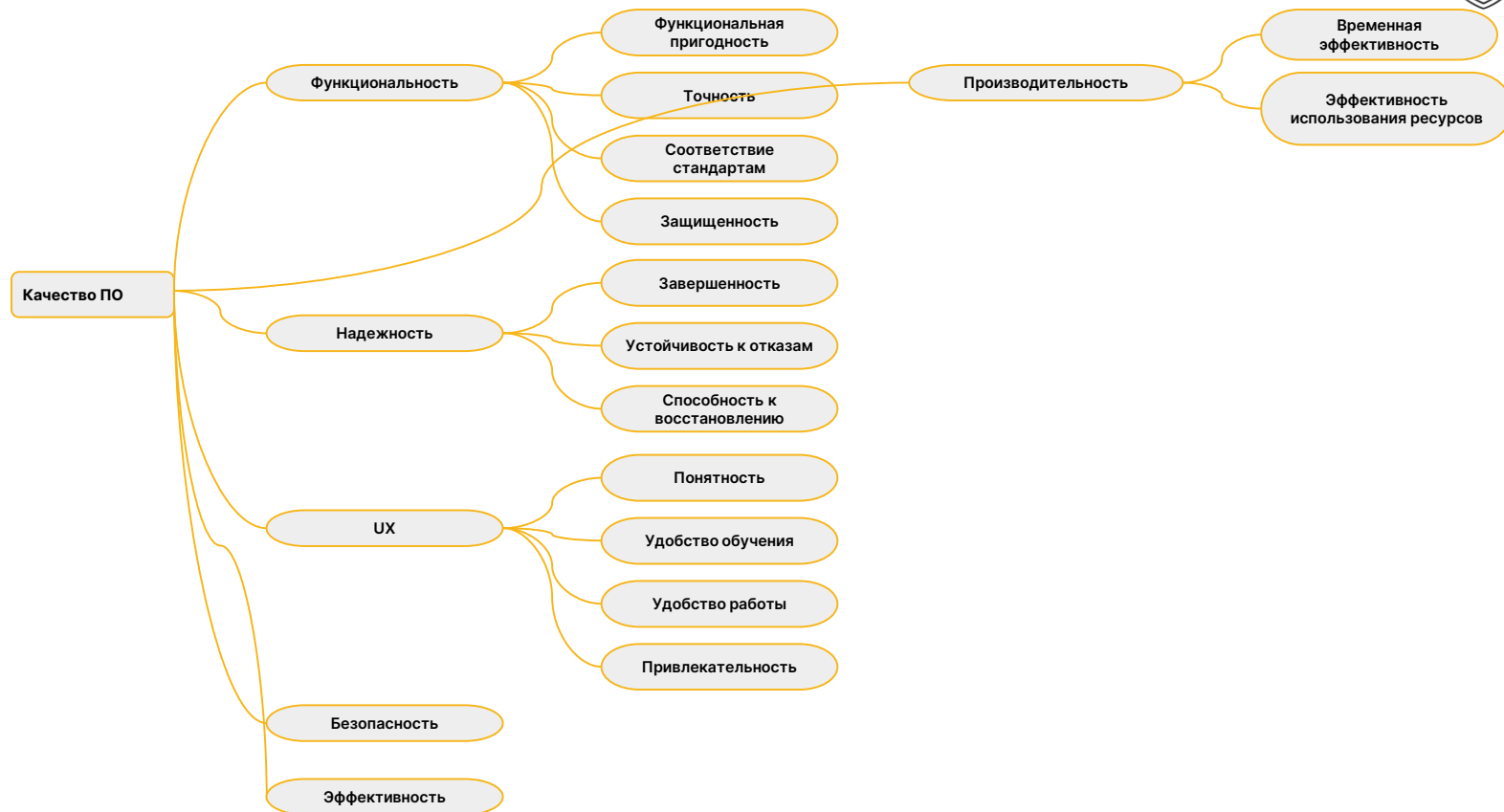


Временная  
эффективность

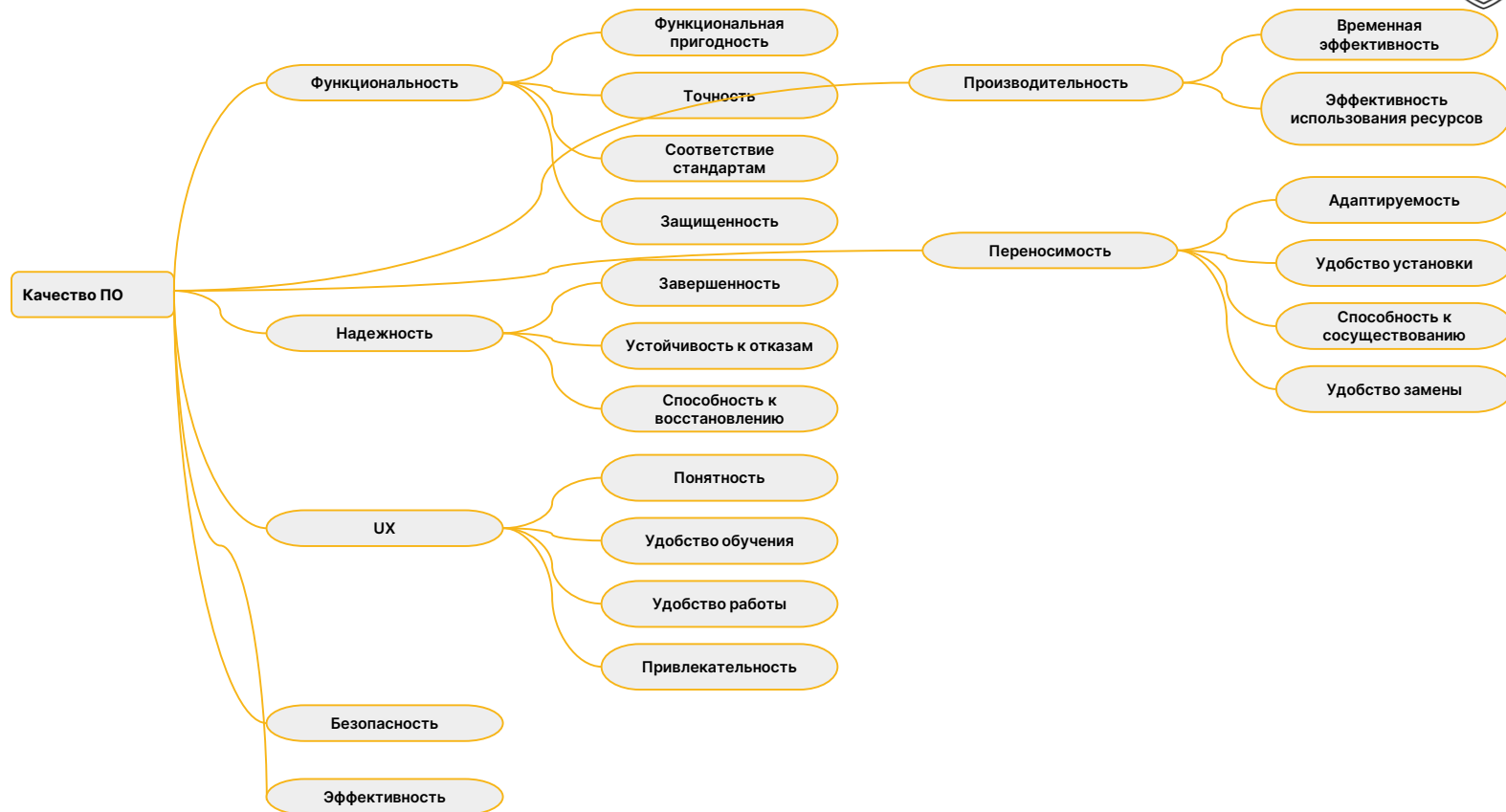
# Качество ПО



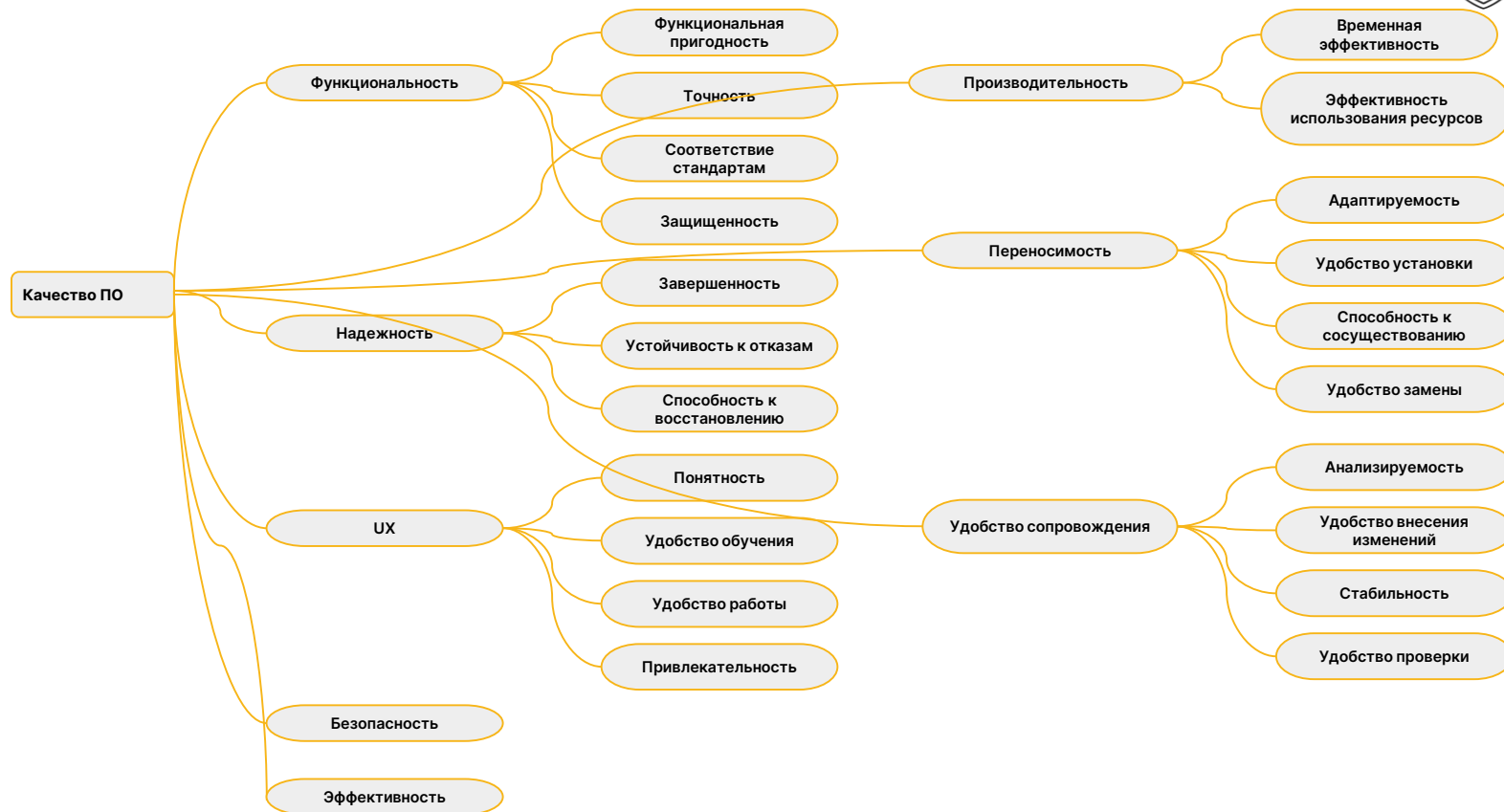
# Качество ПО



# Качество ПО



# Качество ПО



# ВОПРОСЫ

5

# СТАНДАРТЫ ТЕСТИРОВАНИЯ

# Стандарты

Многие организации во всем мире разрабатывают и внедряют различные стандарты для повышения качества своих программ.

## ISO / IEC 9126

В этом стандарте рассматриваются следующие аспекты для определения качества программного приложения:

- Модель качества
- Внешние показатели
- Внутренние показатели
- Показатели качества в использовании

## ISO / IEC 25000: 2005

Широко известен как стандарт, который содержит рекомендации по требованиям и оценке качества программного обеспечения (SQuaRE). Этот стандарт помогает в организации и улучшении процесса, связанного с требованиями к качеству программного обеспечения и их оценками. В действительности ISO-25000 заменяет два старых стандарта ISO, то есть ISO-9126 и ISO-14598.

## ISO / IEC 12119

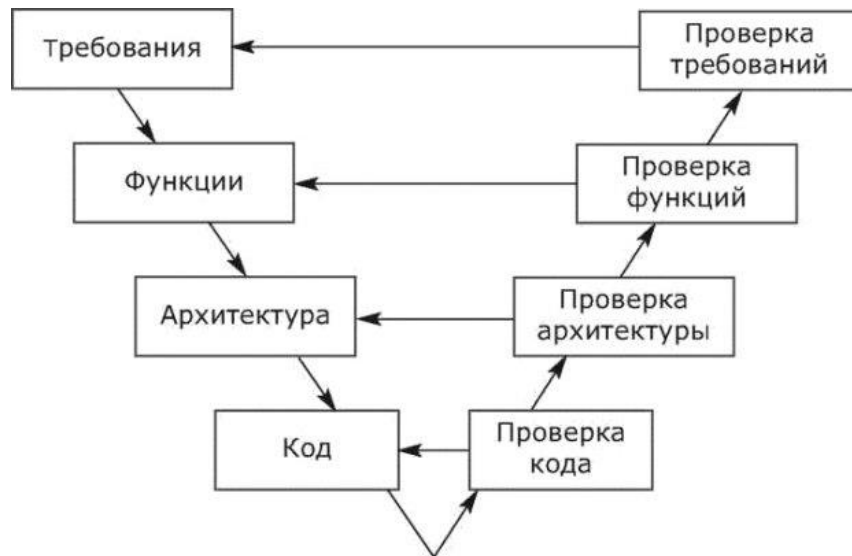
Этот стандарт касается пакетов программного обеспечения, предоставляемых клиенту. Он не фокусируется или не обрабатывает производственный процесс клиентов. Основное содержание связано со следующими пунктами:



# ВОПРОСЫ

# Экспресс-опрос

1. Схема какой модели жизненного цикла программного обеспечения приведена на рисунке?



- a. каскадной модели
- b. V-образной модели
- c. спиральной модели

# Экспресс-опрос

2. Схема какой модели жизненного цикла программного обеспечения приведена на рисунке?



- a. каскадной модели
- b. V-образной модели
- c. спиральной модели

# Экспресс-опрос

3. В каких случаях целесообразно использование V-образной модели жизненного цикла программного обеспечения?
- a. при невозможности четко и полно сформулировать требования на начальном этапе
  - b. при повышенных требованиях к качеству результатов разработки
  - c. при отсутствии критериев завершения разработки

# Немного мемов

Тестировщик пишет свой  
первый автотест под  
присмотром программиста



Тестирование в паре



Тестирование после  
глобального рефакторинга



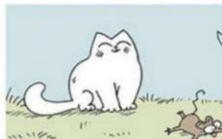
Неуловимый баг



Начинающий тестировщик и баг



Опытный тестировщик и баг



Есть что потестить?



# ОСТАВШИЕСЯ ВОПРОСЫ

# Домашнее задание

**Задание 1.** У вас команда будет релизить web-приложение “Магазин НЛО” (интернет-магазин по продаже летающих тарелок). Подумайте и выпишите, какие виды тестирования вы будете проводить при тестировании проекта.

Шаблон для выполнения дз: [Домашние задания](#)

# Полезные ссылки

[Manifesto for Agile Software Development](#)

[What is Agile? | Atlassian](#)

[Что такое agile? | Atlassian](#)

[Waterfall Methodology: The Ultimate Guide to the Waterfall Model](#)

[Waterfall методология разработки - QA evolution](#)



# Вывод

Сегодня мы с вами познакомились с понятием качества, а также узнали по каким моделям работают команды.

**КТО МОЛОДЦЫ?**

