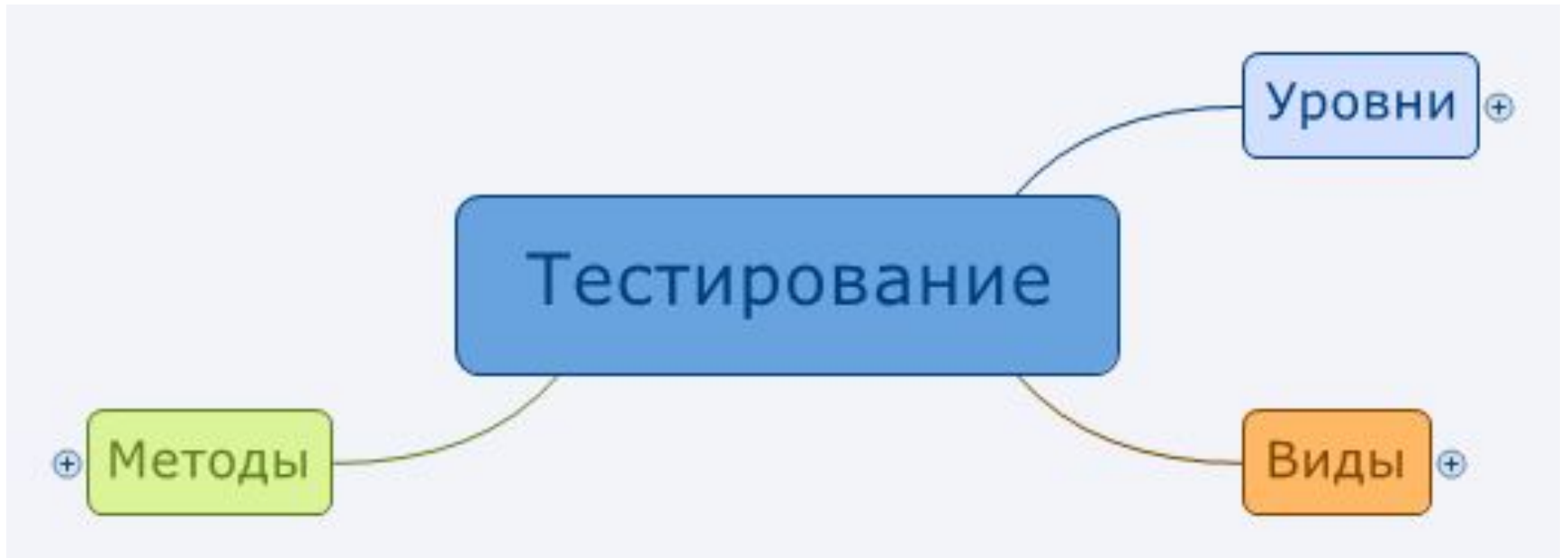


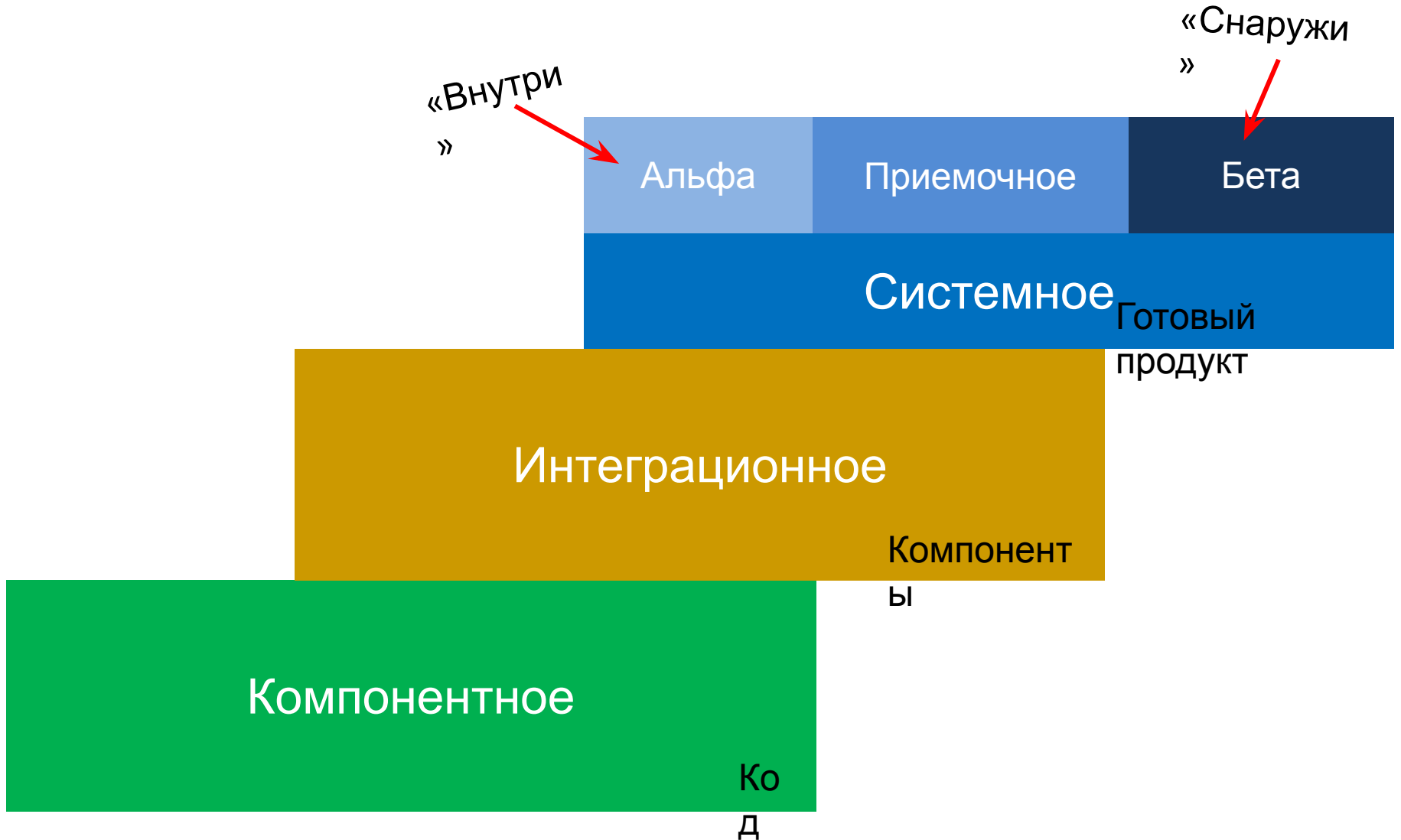
# Тестирование

Какое бывает тестирование

# Варианты классификации



# Уровни



# Компонентное

- Тестирование функциональности отдельных изолированных компонентов системы.
- Для проверки могут использоваться драйвера и заглушки.

# Интеграционное

- Может быть компонентным или системным.
- Проверяет **взаимодействие** уже протестированных отдельных компонентов.

# Системное

- Проверяет работу системы как единого целого.

# Приемочное

Приемка продукта заказчиком.

Конкретный заказчик есть не всегда: если продукт «коробочный» или SaaS, то выделенного приемочного тестирования может не быть.

# Beta!

Раздача почти готового продукта всем заинтересованным лицам.

Как правило, продукт уже хорошо оттестирован. Но бета-тестирование — хороший способ проверить на большом количестве разных конфигураций.



# Виды тестирования



# Функциональное

Что делает приложение?

- ✓ Достаточно ли функций?
- ✓ Делают ли они какую-то полезную пользователю работу?
- ✓ Соответствует ли их реализация тому, что ждет пользователь?

# Smoke

«Дымовое» тестирование позволяет определить, работоспособен ли продукт в принципе.

Пример типичных smoke-тестов:

- Установка приложения;
- Открытие/сохранение файлов;
- Загрузка сайта;
- Регистрация/вход на сайт;
- Базовая функциональность приложения;

# Sanity

Выполняется после получения нового билда с незначительными изменениями, чтобы удостовериться, что эти изменения присутствуют в билде, а также убедиться в общей «адекватности» билда.

Данный вид тестирования позволяет не тратить зря время на тестирование новой функциональности, если новый билд собран неверно или исправление содержит критические ошибки.

# New Features

Что именно тестировать?

Как тестировать?

# Подтверждающее

Подтверждающее тестирование позволяет установить, исправлен ли найденный ранее баг.

Затрагивает измененные участки кода / функциональности.

# Регрессионное тестирование

**Регрессионное тестирование** — тестирование уже протестированной программы, проводящееся после модификации для уверенности в том, что процесс модификации не внес или не активизировал ошибки в областях, не подвергавшихся изменениям. Проводится после изменений в коде программного продукта или его окружении.

# Нефункциональное

- С помощью этих видов оцениваются нефункциональные характеристики продукта. Как правило, их можно оценить в некоторых единицах.



# Производительности

- Нагрузочное — нормальная нагрузка
  - Потребление ресурсов процессора
  - Потребление оперативной памяти
  - Потребление сетевых ресурсов
  - Работа с дисковой подсистемой
  - Время выполнения запроса
  - ...
- Стресс
  - Нагрузка, сильно превышающая «обычную»
- Стабильности
  - Поведение системы при *длительной* постоянной нагрузке

# Юзабилити и интерфейса

В чем разница?

# Безопасности

- Конфиденциальность
  - доступ должны получать только те, кто имеет нужные права
- Целостность:
  - изменения вносятся только теми, кто имеет доступ;
  - процедуры восстановления повреждённых данных;
- Доступность:
  - информация, сервис и прочие ресурсы всегда доступны, когда это требуется;

# Интернационализации

- Поддержка unicode, RTL
- Использование различных разделителей
- Перевод интерфейса на остальные языки

# Совместимости

Аппаратное и программное обеспечение

# Доступности

Доступность приложения для людей с ограниченными возможностями

# Структурное

- Структурное тестирование – тестирование архитектуры или структуры приложения. К нему относят все white-box тестирование.

# Сопровождающее

- Тестирование готового продукта после передачи его заказчику.
- Может включать в себя:
  - Проверку установки обновлений, патчей
  - Проверку работы на новом аппаратном и программном окружении
  - Проверку миграции на новые версии
  - ....

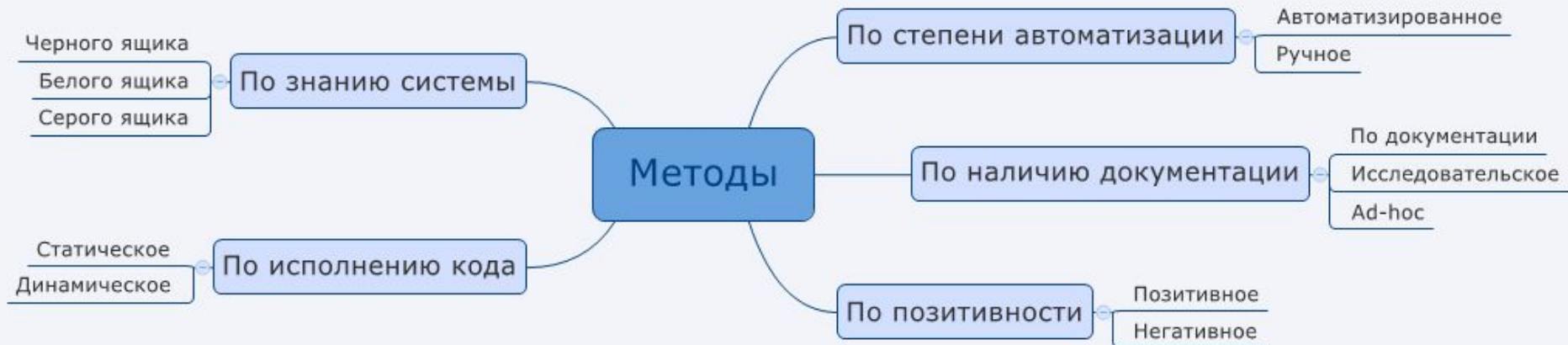


# Верификация и валидация

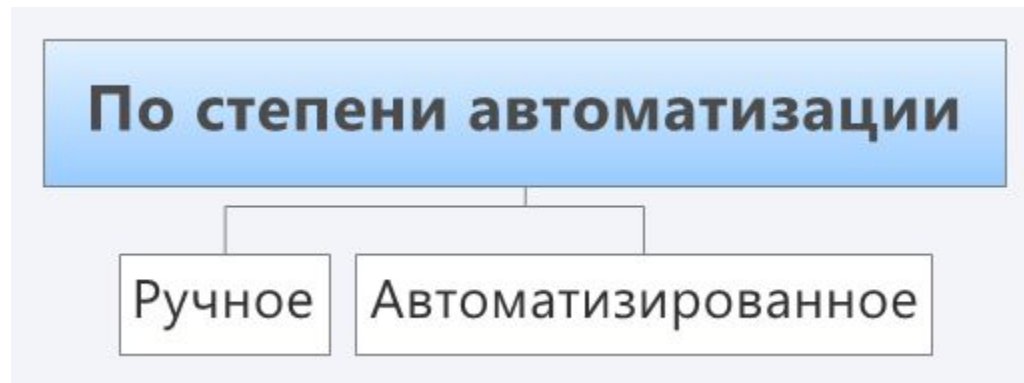
**Верификация** – проверка продукта на соответствие входным данным, правилам, стандартам.

**Валидация** – проверка продукта на соответствие потребностям пользователя

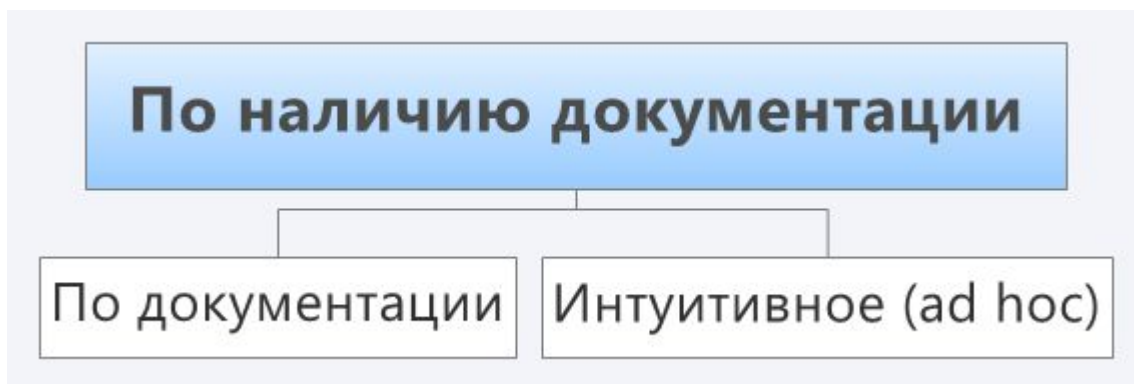
# Методы



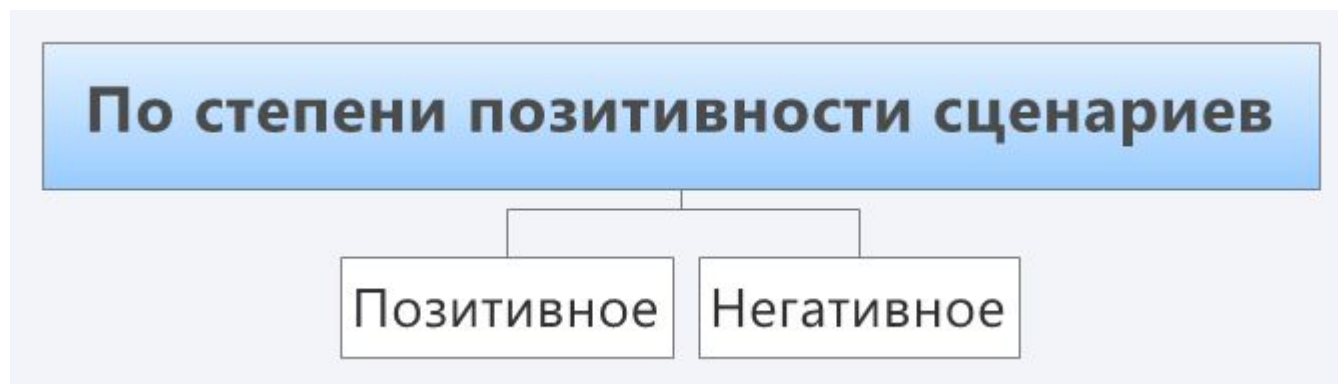
# По степени автоматизации



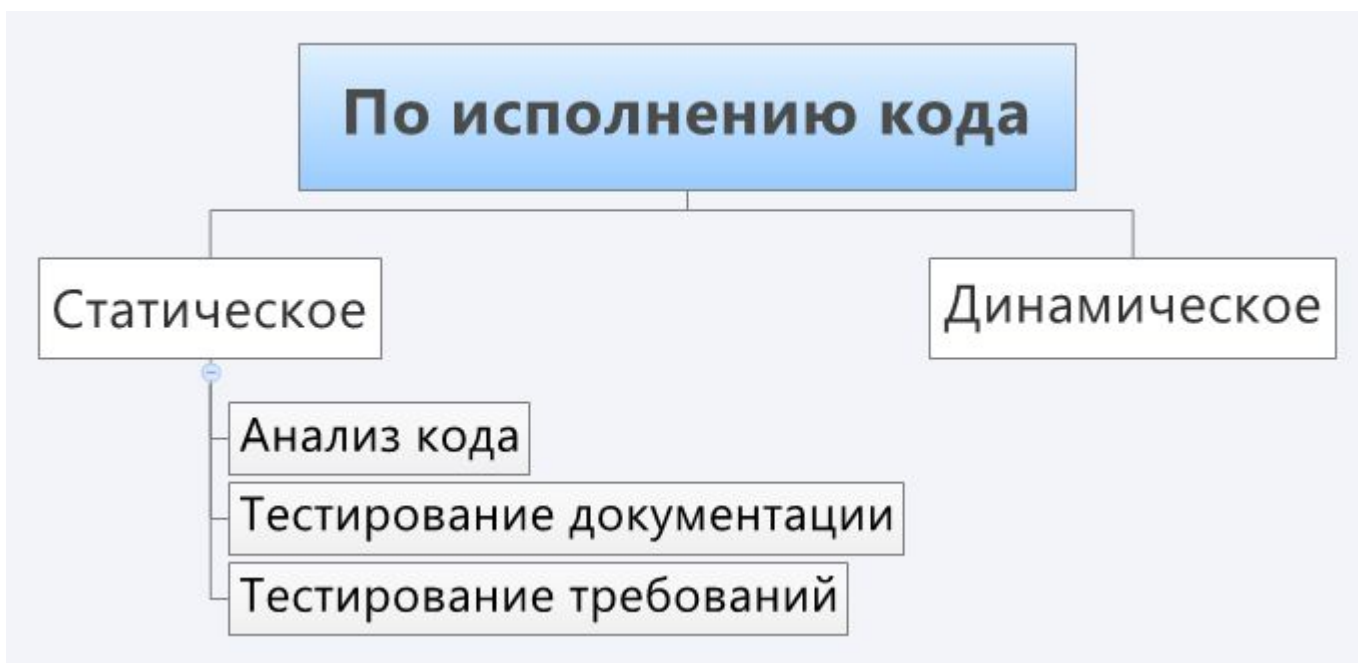
# По степени подготовленности



# По признаку позитивности



# По исполнению кода



# По знанию системы

