

Основы тестирования мобильных приложений



Содержание

Основные виды тестирования

Уровни тестирования

Функциональное тестирование

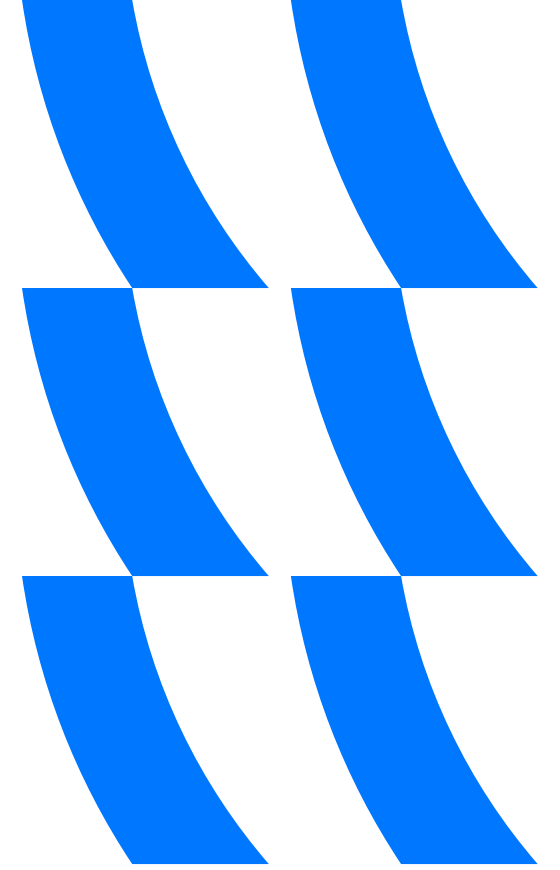
Нефункциональное тестирование

Связанное с изменениями тестирование

Требования: виды требований, зачем они нужны, примеры

Основные виды тестирования

Зачем знать типы и
виды тестирования?



Зачем знать типы и виды тестирования?

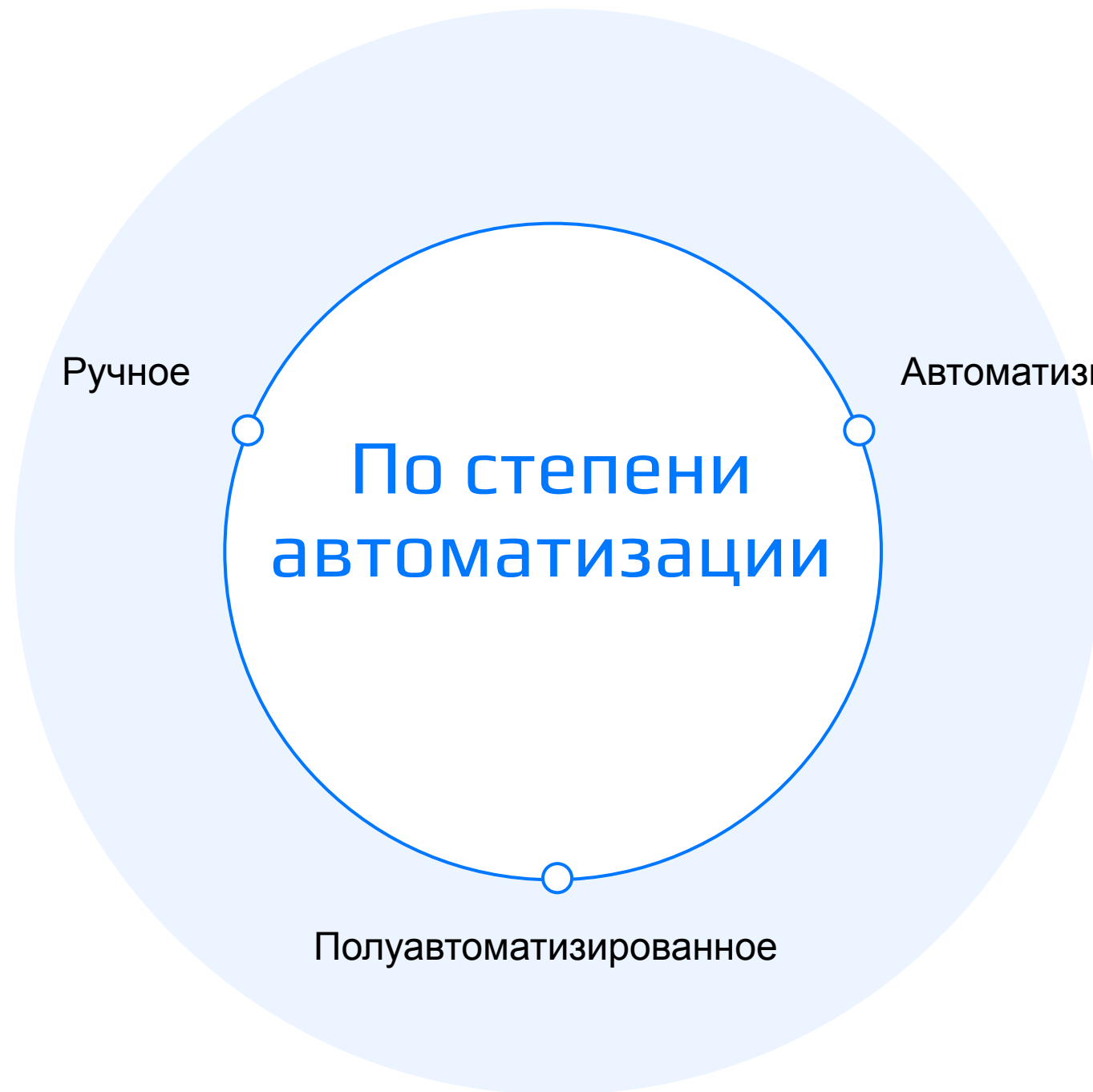
1. Необходимы для понимания подходов к тестированию.
2. Зная особенности тестируемого продукта, мы делаем меньше усилий для лучшего покрытия фичи проверками.
3. Экономим свои ресурсы, можем спрогнозировать время на задачу
4. Знаем, что ожидать от функциональности, документации, требований
5. Пригодится на собеседовании.

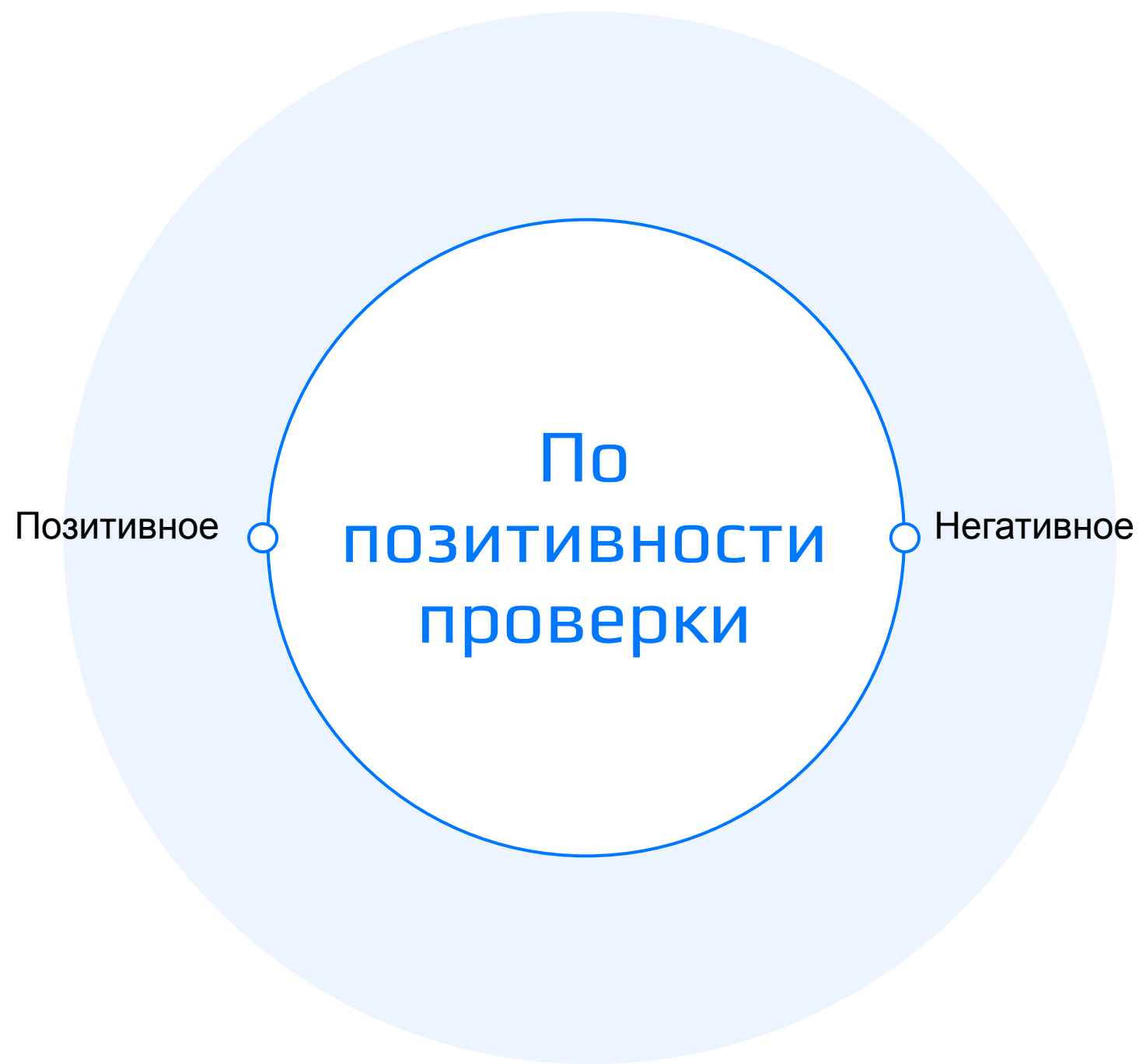


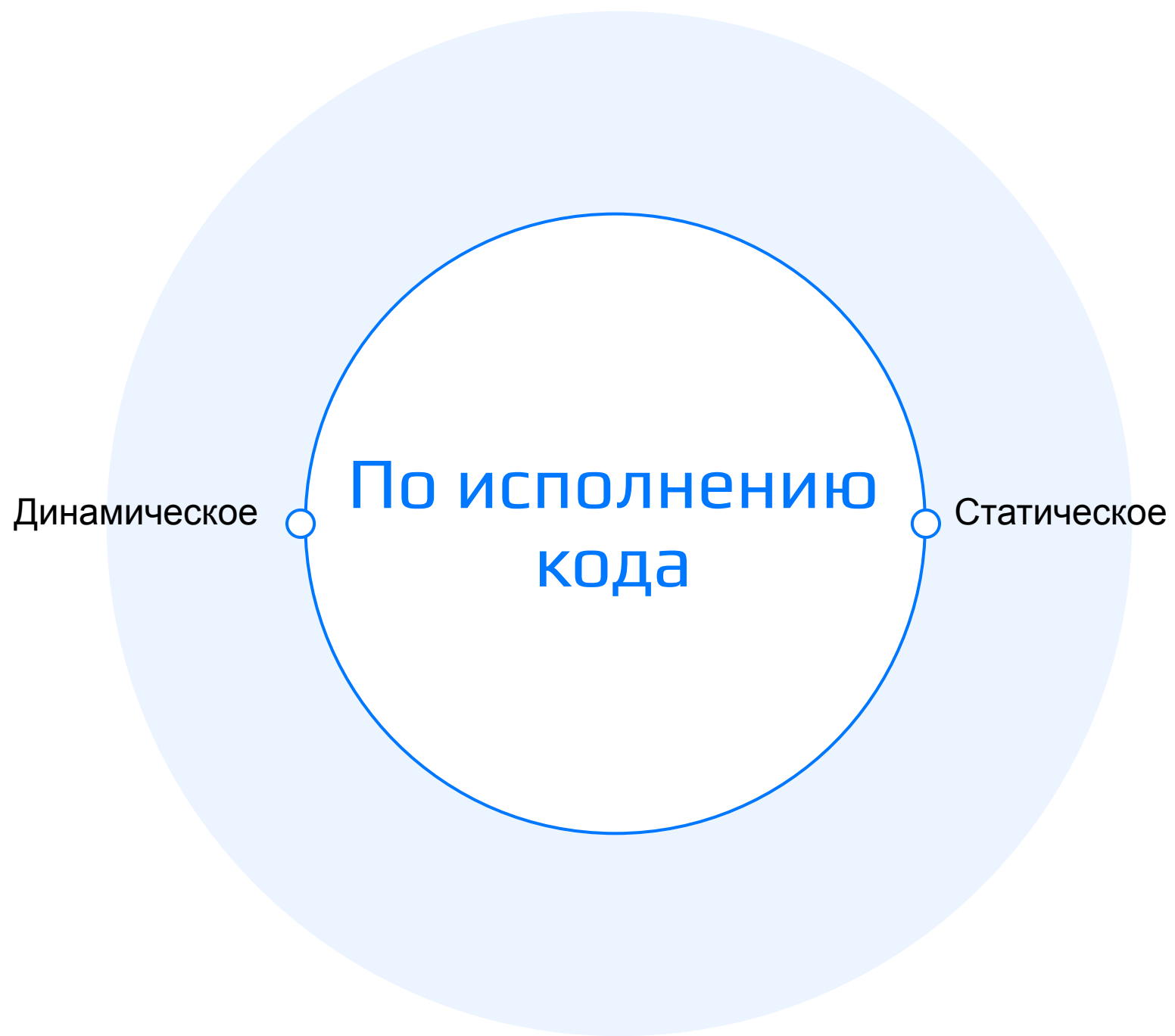












На основе модели

На основе требований

**По разработке
тестовых
сценариев**

На основе вариантов
использования

Уровни тестирования

Уровни тестирования

Уровень тестирования определяет то, **над чем** производятся тесты: над отдельным модулем, группой модулей или системой в целом.

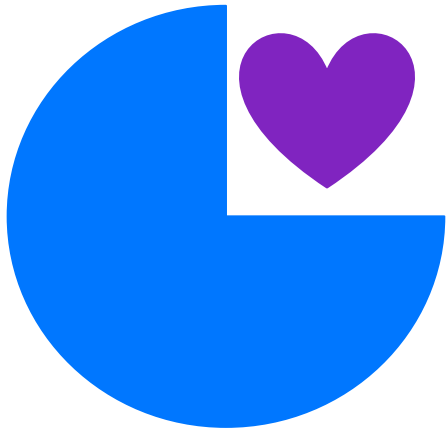
Модульное тестирование
(Unit Testing)

Интеграционное тестирование
(Integration Testing)

Системное тестирование
(System Testing)

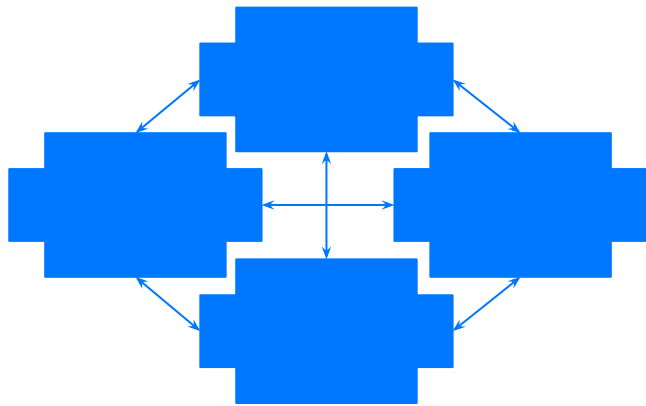
Приемочное тестирование (Acceptance
Testing)

Модульное тестирование



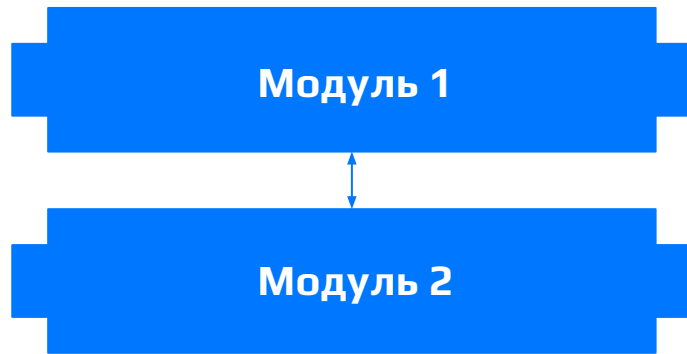
Проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по отдельности (модули программ, объекты, классы, функции и т.д.).

Интеграционное тестирование

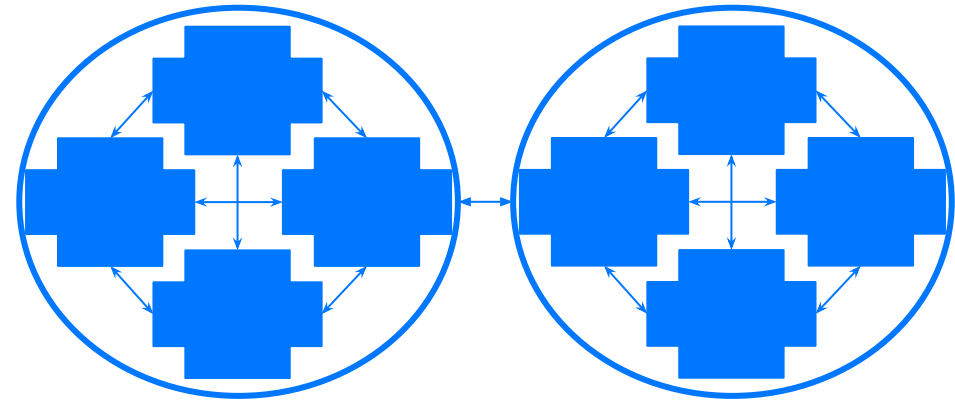


Проверяет объединение модулей, их взаимодействие между собой, а также интеграция подсистем в единую систему. Отдельно проверенные модули составляют множество.

Уровни интеграционного тестирования

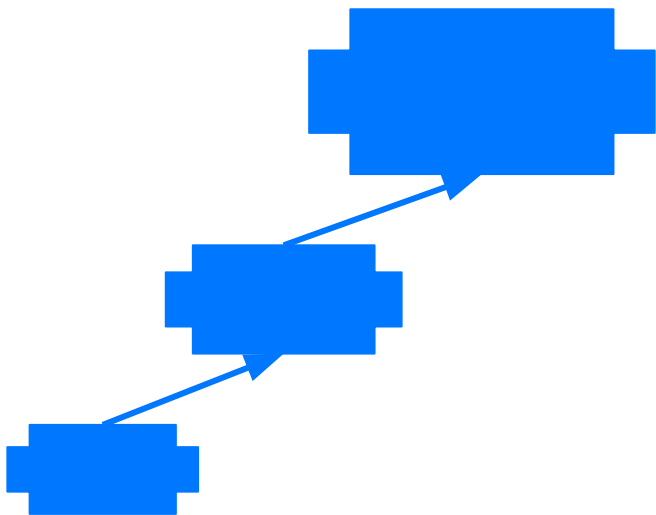


Компонентный интеграционный уровень



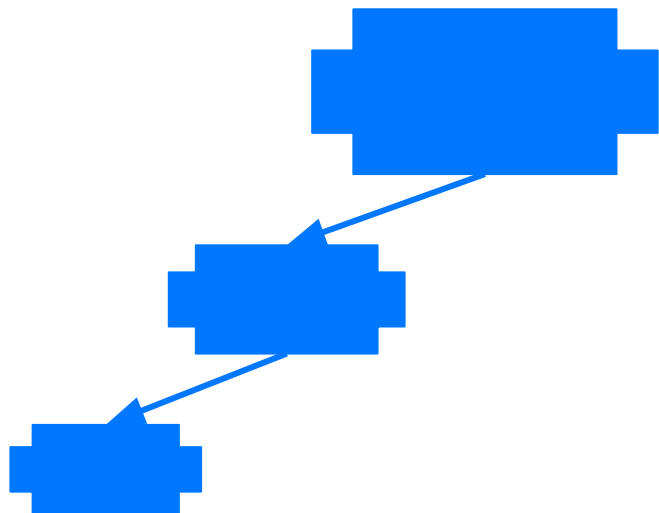
Системный интеграционный уровень

Подходы к интеграционному тестированию



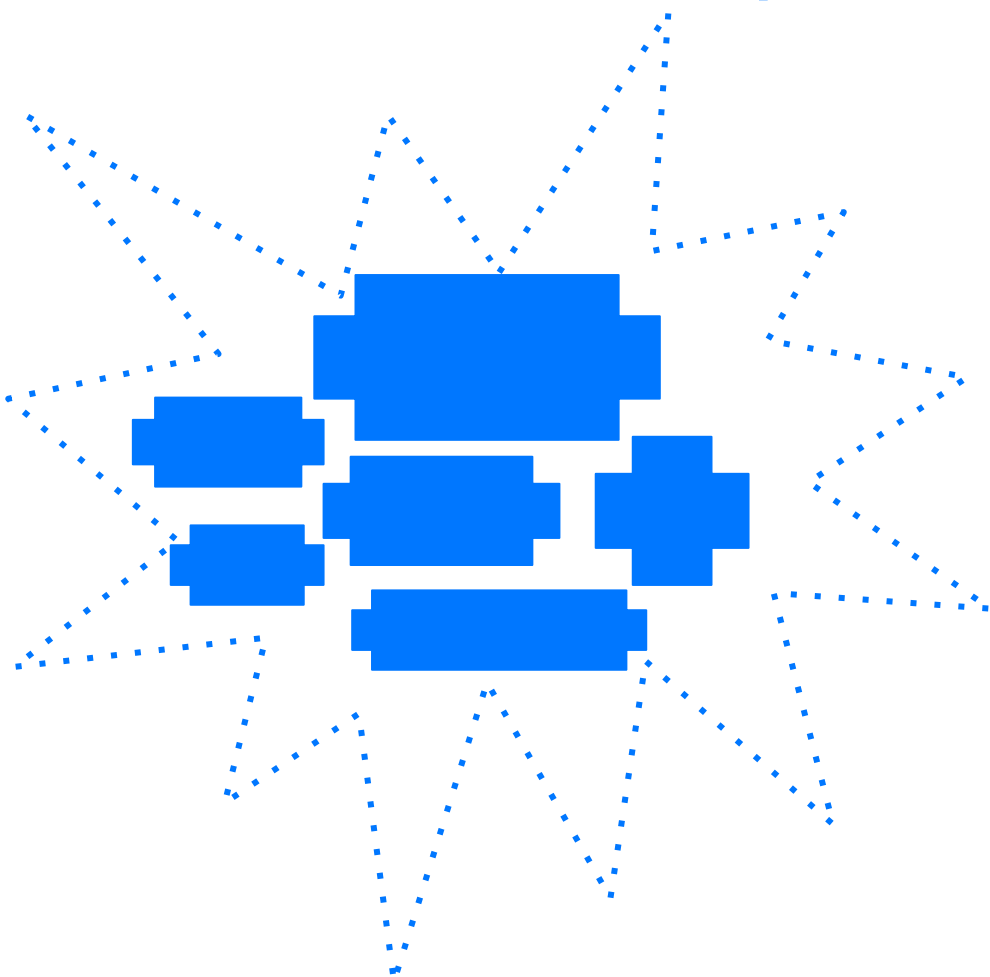
“Снизу вверх” – низкоуровневые модули объединяются и тестируются. Потом берется следующий уровень модулей и тестируется. Этот подход помогает понять готовность приложения.

Подходы к интеграционному тестированию



“Сверху вниз” – сначала тестируем высокоуровневое объединение уровней, затем к нему добавляем низкоуровневые. Модули более низкого уровня на этом этапе можно заменить “заглушками” и “моками”.

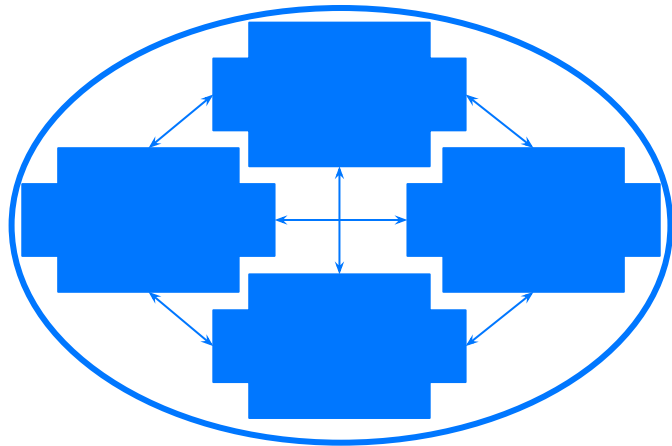
Подходы к интеграционному тестированию



“Big Bang” – все модули собираем вместе и наблюдаем, как все ломается. Можно сказать, что это “черновик” рабочей системы. Такой подход помогает сохранить время, но без тестовой документации такой подход неуместный.



Системное тестирование



Тестирование выполняется на полностью интегрированной системе с целью проверки соответствия системы исходным требованиям, как функциональным, так и не функциональным.

Подходы к системному тестированию

1

На основании требований



Требую кофе! Завариваю кофе.

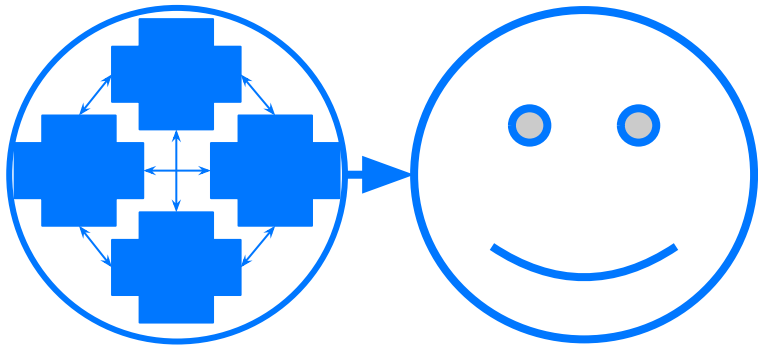
2

Интуитивное



Требую кофе! Ищу по запаху.

Приемочное тестирование




Формальный процесс тестирования, который проверяет соответствие системы требованиям и проводится с целью определения удовлетворения системой приемочным критериям и вынесения решения заказчиком или другим уполномоченным лицом принятия приложения.

Вопросы



Функциональное тестирование

Подробнее о функциональном тестировании



1

Проверяем
соответствие
требований и того, как
они реализованы

2

Подтверждаем, что
система обладает тем
функционалом, что
ожидает заказчик

3

На основании
спецификации
составляются тест-кейсы

Подробнее о функциональном тестировании

1

Проверяем соответствие требований и того, как они реализованы

2

Подтверждаем, что система обладает тем функционалом, что ожидает заказчик

3

На основании спецификации составляются тест-кейсы

4

На основании бизнес-процессов составляются юзер-кейсы

5

Имитируем реальное физическое использование системы

6

Минусы: вероятность избыточного тестирования, допущение логических багов

Тестирование безопасности

1

Конфиденциальность – ожидается сокрытие определенных ресурсов или информации.

2

Целостность – ожидается, что ресурс может быть изменен только определенной группой пользователей; проверки на повреждение и восстановление.

3

Доступность – требования о том, что ресурсы должны быть доступны авторизованному пользователю, внутреннему объекту или устройству

Примеры функциональных тестов



Тестируем ручку :)



Нефункциональное тестирование

Подробнее о нефункциональном тестировании

Проверяем свойства, которые не относятся к функциональности системы

1

Надежность – система реагирует на непредвиденные ситуации

2

Производительность – работоспособность системы под разными нагрузками

3

Удобство – удобность работы системы с т.з. пользователя

Подробнее о нефункциональном тестировании

Проверяем свойства, которые не относятся к функциональности системы

1

Надежность – система реагирует на непредвиденные ситуации

2

Производительность – работоспособность системы под разными нагрузками

3

Удобство – удобность работы системы с т.з. пользователя

4

Портируемость – переносимость приложения на различные платформы

5

Масштабирование – требования к вертикальному и горизонтальному масштабированию приложения

6

Безопасность – защищенность пользовательских данных

Нефункциональные виды тестирования

1

Тестирование
производительности

1. Нагрузочное
2. Стрессовое
3. Объемное
4. Стабильности и надежности

2

Тестирование
установки

3

Юзабилити
тестирование
(удобства
пользования)

4

Тестирование на
отказ и
восстановление

5

Конфигурационное
тестирование

Примеры нефункциональных тестов



Тестируем ручку :)



Связанное с
изменениями
тестирование

Виды связанного с изменениями тестирования



Дымовое (смоук)
тестирование



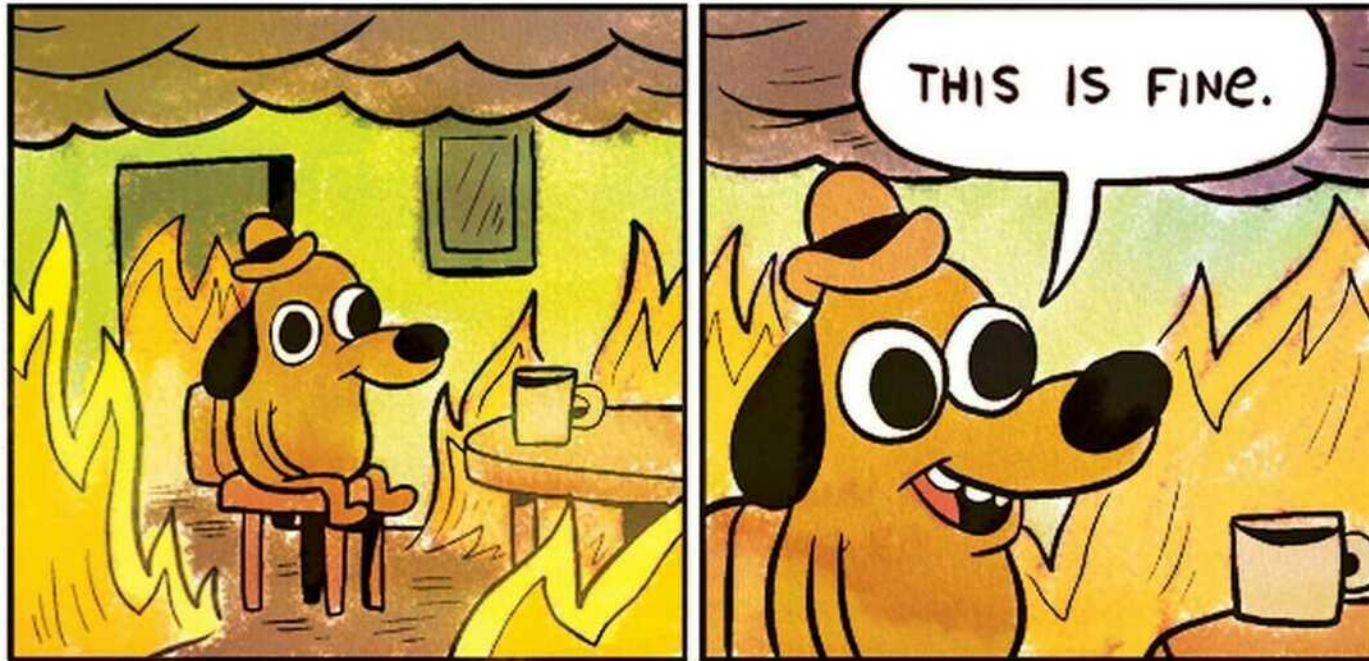
Регрессионное
тестирование



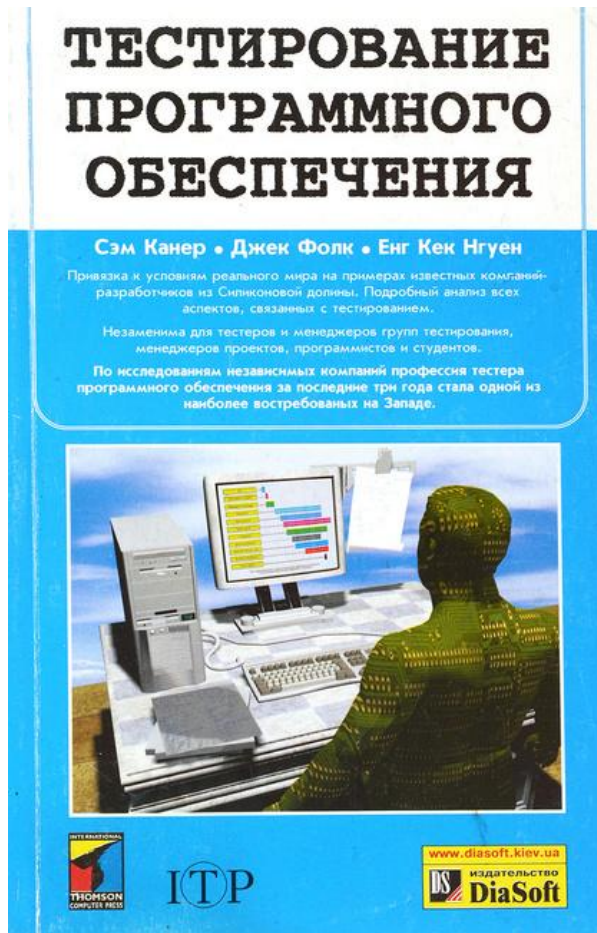
Санитарное
(санити)
тестирование

Смоук (дымовое) тестирование

"При вводе в эксплуатацию нового оборудования тестирование прошло удачно, т.к. из установки не пошел дым." (с) Инженерная среда



Регрессионное тестирование



3 типа регрессионного тестирования по Сэму Канеру:

Регрессия багов (Bug regression) - попытка доказать, что исправленная ошибка на самом деле не исправлена.

Регрессия старых багов (Old bugs regression) - попытка доказать, что недавнее изменение кода или данных сломало исправление старых ошибок, т.е. старые баги стали снова воспроизводиться.

Регрессия побочного эффекта (Side effect regression) - попытка доказать, что недавнее изменение кода или данных сломало другие части разрабатываемого приложения.

Санитарное тестирование



Санитарное тестирование - это максимально узконаправленное тестирование, достаточное для доказательства того, что *конкретная* функция работает согласно заявленным в спецификации требованиям.

Отличия дымового от санити



Дымовое (смоук)
тестирование

VS



Санитарное
(санити)
тестирование

Требования: виды
требований, зачем они
нужны, примеры

Цели тестирования

1. Выявить все наиболее серьезные последствия проблемы
2. Найти простейший и кратчайший путь ее воспроизведения.
3. Найти альтернативные действия, приводящие к такому же результату.
4. Выявить связанные проблемы.

А как?

1. Выявить все наиболее серьезные последствия проблемы

Какой основной функционал и приоритеты?

2. Найти простейший и кратчайший путь ее воспроизведения.

Какие шаги воспроизведения есть еще? Какие точки входа?

3. Найти альтернативные действия, приводящие к такому же результату.

Можно ли упростить воспроизведение дефекта?

4. Выявить связанные проблемы.

А что еще могло «пострадать»?



Требования

Какие функции

Требования

A diagram consisting of a large light blue circle. Inside it is a smaller white circle with a blue outline. The word "Требования" is written in blue in the center of the white circle. Four small white circles with blue outlines are positioned at the top-left, top-right, bottom-left, and bottom-right of the white circle's perimeter. To the left of the white circle, the text "Какие функции" is written in black.

Какие функции

С соблюдением
каких внешних
условий

Требования



Какие функции

С соблюдением
каких внешних
условий

Требования

В процессе
решения
полезной
задачи



Требования - это

Условия или возможности,
необходимые пользователю для
решения той или иной проблемы
или для достижения определённых
целей

**Когда в процессе реализации твой
проект изменился до неузнаваемости:**



все зашло слишком далеко
но мне даже нравится

Что важно для тестирования?

Требования к программному обеспечению — совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации.

... набор параметров

Зачем нужны?

- Реализовать все функции, которые ожидает увидеть в продукте заказчик
- Реализованные функции соответствовали ожиданиям
- Тестировщики могли проверить, что программа делает именно то, что должна делать
- Технические писатели написали хелп, руководство пользователя или другую документацию для конечного пользователя
- Оценить затраты на разработку и время разработки



Три уровня функциональных требований :



Бизнес-требования:
высокоуровневые цели
организации или
заказчиков продукта



Требования пользователей:
цели и задачи, которые
решает пользователь

- Пользовательские истории
- Сценарии взаимодействия



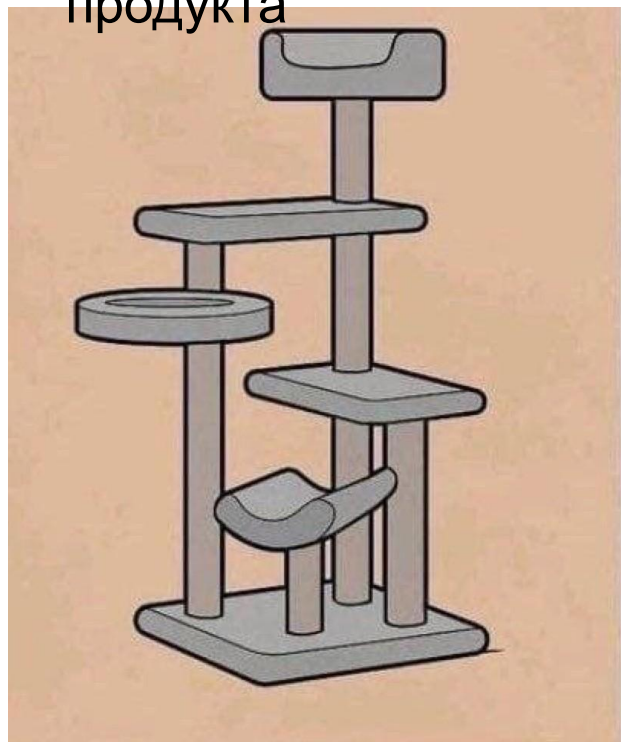
Функциональные требования:
какой функционал реализуют
разработчики, чтобы
пользователи смогли
выполнить свои задачи в
рамках бизнес-требований

- Спецификация требований*
- Системные требования

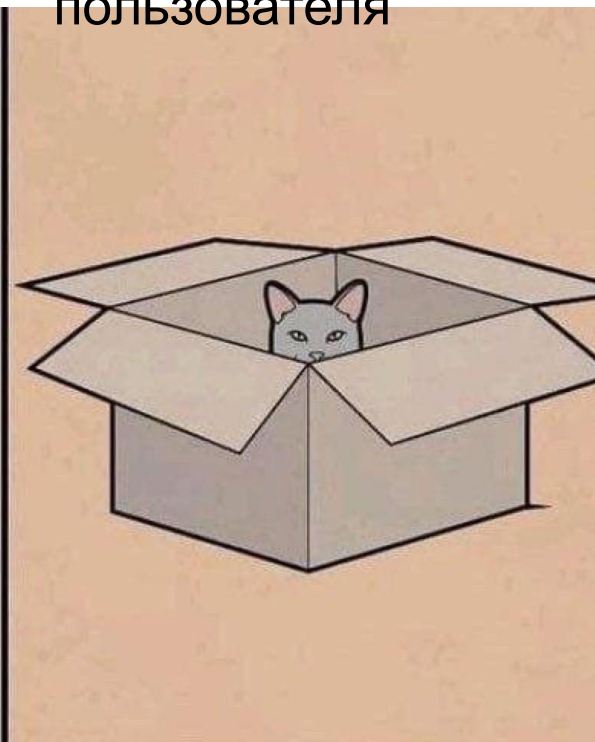
* Спецификация требований – законченное описание поведения программы, которую требуется разработать.



Функционал
продукта



Нужды
пользователя



Нефункциональные требования

- «Бизнес-правила»: пользовательские соглашения, ссылки на законодательство, внутренние правила заказчика
- Внешние интерфейсы: интерфейсы пользователя, протоколы взаимодействия с другой системой, UI Guidelines
- Атрибуты качества: ... ?

Атрибуты качества

- Легкость и простота использования (usability)
- Производительность (performance)
- Удобство эксплуатации и технического обслуживания (maintainability)
- Надежность и устойчивость к сбоям (reliability)
- Взаимодействия системы с внешним миром (interfaces)
- Масштабируемость (scalability)
- Требования к пользовательским и программным интерфейсам (user and software interface).

Примеры требований к ПО



Шариковая
ручка



Шариковая ручка

ГОСТ 28937-91 Ручки шариковые
автоматические (Дата введения
1992-01-01)



Домашнее задание

1. Составить майндкарту с видами тестирования
2. Подготовиться к следующему заданию и изучить информацию о тест-дизайне и методиках тест-дизайна



Не забудьте отметить
на портале и оставить
обратную связь о
занятии :)