

# Нива и видове тестване

# Нива на тестване

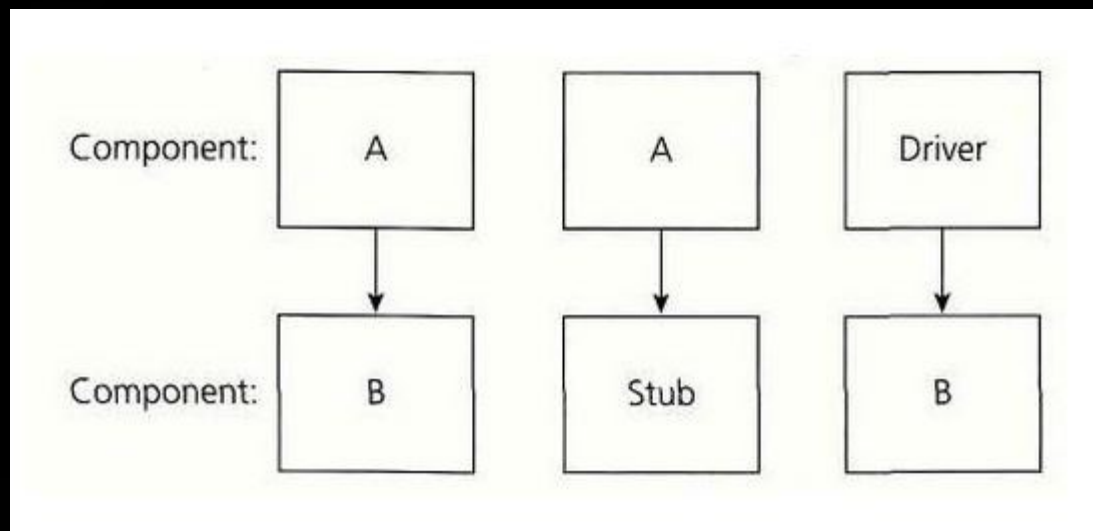
- | Component testing
- | Integration testing
- | System testing
- | Acceptance testing

# Component testing

- | Тестване на всеки един компонент поотделно.
- | Не е нужно всички компоненти от системата да са завършени.
- | Предотварява външно влияние над тествания компонент.

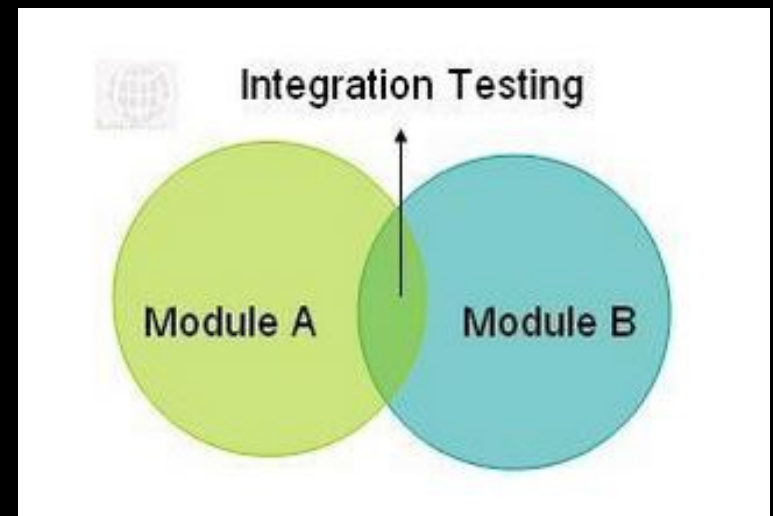
# Component testing

- Stub – Извиква се от компонента който ще бъде тестван (“A” извиква “Stub”).
- Driver – Извиква компонента който ще бъде тестван (“Driver” извиква “B”).



# Integration testing

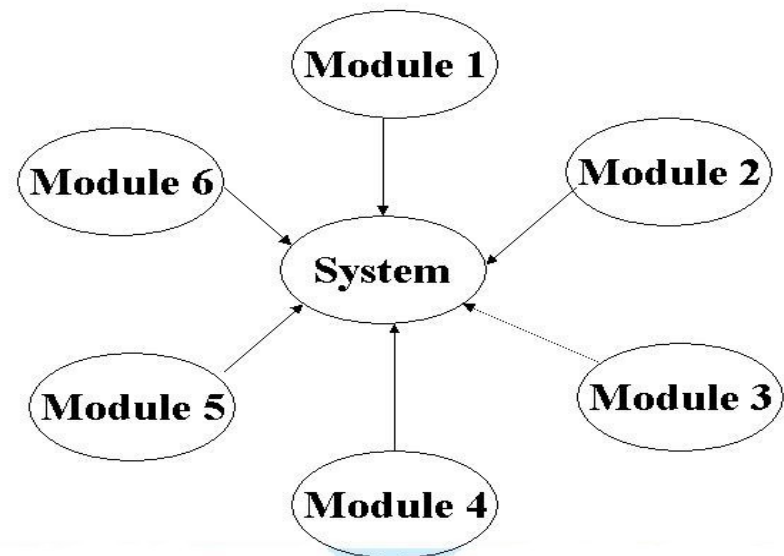
- Тестване на взаимната работа на отделните компоненти.
- Тестване на интерфейсите и взаимоотношенията между компонентите



# The Big Bang integration testing

- | The Big Bang approach – всички компоненти са интегрирани едновременно и тестването започва след това.
- | Всички компоненти се разглеждат като едно.

**Big Bang Integration Testing**



# System testing

- | Проверяваме дали поведението на системата отговаря на системните изисквания.
- | Целта е да се уверим че произведения продукт отговаря на очакванията и изискванията на клиента / потребителя.

# System testing

- † Системното тестване се извършва в среда максимално близа до клиентската среда.
- † Системното тестване не трябва да се извършва директно на клиентската среда (UAT) – по време на тестването клиентските данни и среда могат да се повредят.



# Acceptance testing

- Тестване от гледната точка на потребителя.
- Често клиентът се включва в тази фаза на тестването:
  - Лесни за разбиране от клиента тестове.
- Тестването се извършва на среда близка до клиентската (UAT) или направо на нея.



# Acceptance testing

- ▮ User Acceptance test – тестване на функционалността и използваемостта на сайта.
- ▮ Operational Acceptance test – проверява се дали системата отговаря на операционните изисквания (бек-ъп на бази данни, възстановяване на данни, възстановяване на сайта и др.)

## Видове тестване

- | Risk-Based testing
- | Functional testing
- | Non-functional testing
- | Re-testing and Regression testing

# Risk-based testing

- Риск – вероятността да се появи какъвто и да е неблагоприятен или нежелан ефект.
- Риск-базираното тестване се извършва в на базата на приоритизация – определят се по-важните тестове и се изпълняват по-напред.

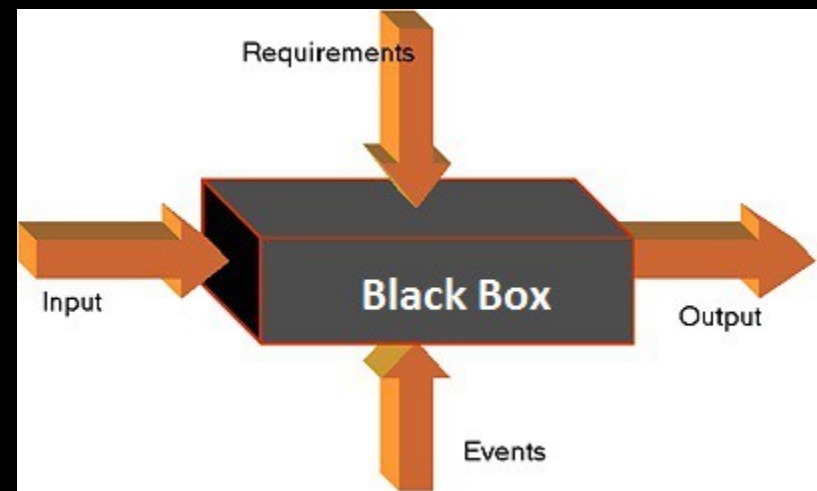


## Risk-based приоритизиране

- Product Quality Risk – проблеми свързани с качеството на системата.
- Project Planing Risk – проблеми които могат да доведат до забавяне на проекта.
- Фактори за приоритизация:
  - Вероятността да се появяват проблеми.
  - Въздействието което биха оказали проявилите се проблеми.

# Functional testing

- Тестване на входно-изходните характеристики на продукта.
- Black box техниката се ползва за функционално тестване.



# Functional testing

- † Функционалните тестове са базирани на функционалните изисквания зададени от клиента.
- † Software Requirements Specifications (SRS) – документът в който са записани функционалните изисквания.
- † Добре е да има поне по 1 тест за всяко изискване.



# Non-functional Testing

- Performance test – проверява се бързодействието и времето за отговор.
- Load test – проверява се поведението на сайта при повишаване на натоварването:
  - Повишаване на потребителите.
  - Повишаване на броя на извършваните задачи.



# Non-functional Testing

- Stress test - проверява се поведението на сайта при свръх натоварване. Системата обикновено се натоварва докато блокира.
- Security test – търсят се потенциални ниши през които могат да се извършат хакерски атаки.

# Non-functional Testing

- ▮ Stability test – средно време на проваляне на системата, честота на проваляне.
- ▮ Compatibility test – проверяваме за съвместимост с дадени системи (ОС, браузъри и др.)
- ▮ Usability test – проверяваме лесно ли се използва системата, лесно ли се научава и лесно ли се разбира.

# Non-functional Testing

- Accessibility testing – проверява се дали системата е използвана от потребители със специфични увреждания.



## Re-test и Regression test

- ┌ След като е намерен и оправен даден проблем, системата трябва да се ретества:
  - ┌ За да се провери дали наистина е отстранен проблема.

## Re-test и Regression test

- Regression test – ретестване на системата, за да се провери дали не са се появили нови бъгове следствие на:
  - Промени по сиситемата.
  - Поправяне на бъгове.
- Извършва се на всички тестови нива.

## Regression test cases

- ▮ Регрешън тестовете могат да бъдат изпълнявани много често.
- ▮ Регрешън тестове се правят всеки път когато има:
  - ▮ Поправяне на бър.
  - ▮ Промяна по кода.
  - ▮ Нова функционалност.

# Въпроси





## Задачи за домашно

Напишете Regression test сюита (не повече от 10 теста), който има за цел бързо да се изтества най-основната функционалност на сайта <http://qa.soft-intellect.com/>

- ▮ Pages.
- ▮ Posts.
- ▮ Media.