

Компонентно тестване

(Unit Testing)



Учителски екип

Обучение за ИТ кариера

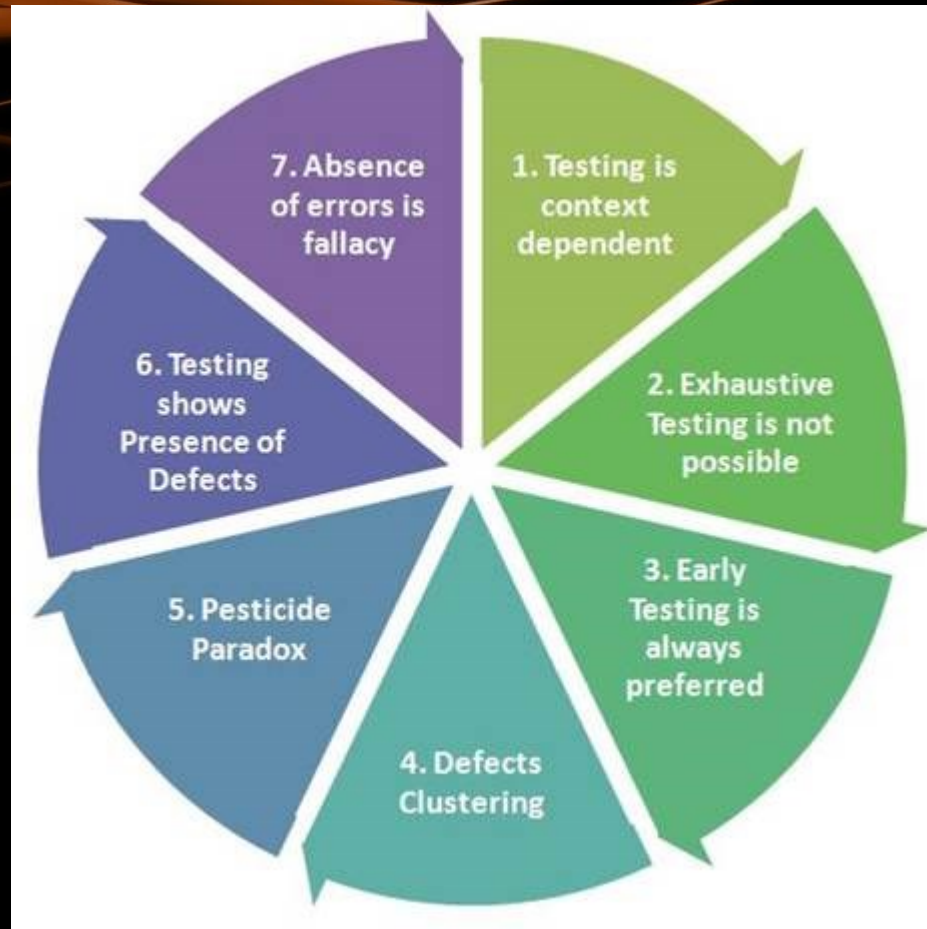
<https://it-kariera.mon.bg/e-learning/>



Съдържание

1. Седемте принципа на тестването
2. Какво е компонентното тестване?

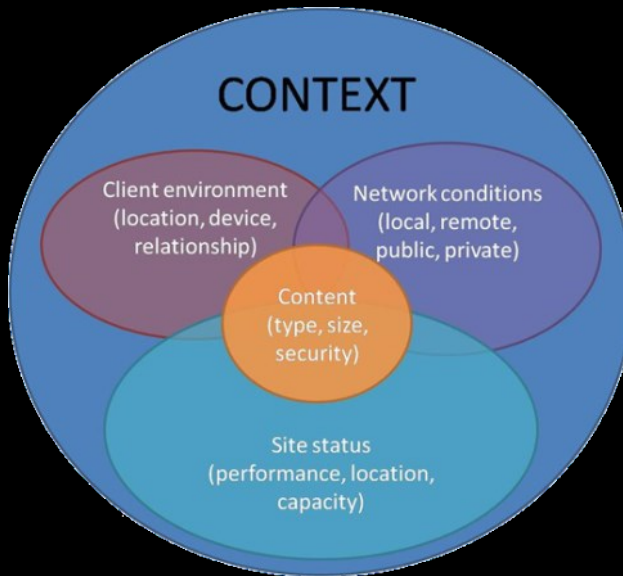




Седемте принципа на тестването

Седемте принципа на тестването

- Тестването е зависимо от **контекста**
 - Тестването се извършва по различни **контексти**
 - Пример: банков софтуер и текстов редактор се тестват по различен начин



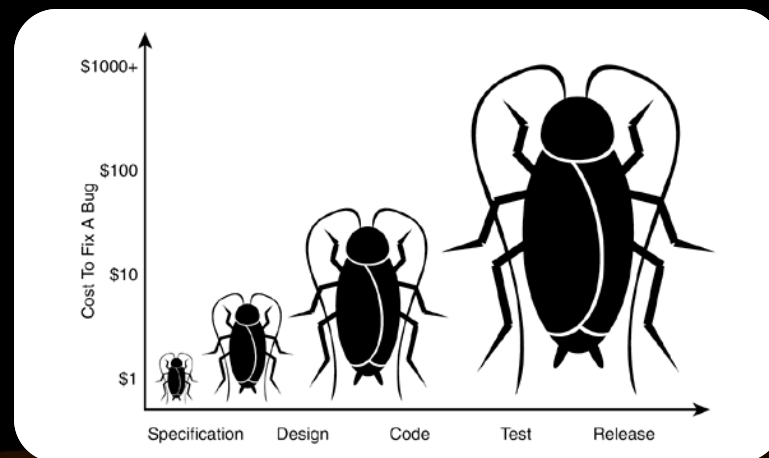
Седемте принципа на тестването (2)

- Изчерпващото тестване е **практически невъзможно**
 - Всичките възможни комбинации от входни данни и начални условия обикновено се неизборимо много
 - Тестването на всичко **не е възможно**
 - Освен на основните случаи
 - Анализът на риска и приоритетите на софтуера показват къде трябва да насочим усилията при тестване



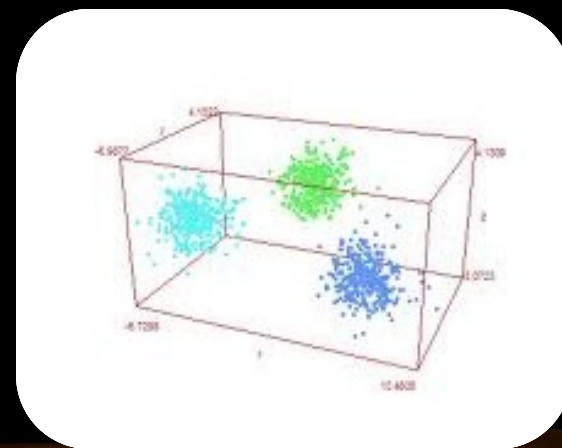
Седемте принципа на тестването (3)

- По-добре рано, отколкото късно!
 - Дейностите по тестването трябва да започнат възможно **най-рано**
 - И трябва да са насочени към конкретни цели
 - Колкото по-късно се открие даден бърк – толкова по-зле



Седемте принципа на тестването (4)

- Групиране на грешките
 - Тестването трябва да е **пропорционално**
 - На очакваната и по-късно наблюдавана честота на дефектите на модулите
 - Обикновено малък брой модули съдържат по-големия брой дефекти



Седемте принципа на тестването (5)

- Парадоксът на пестицидите
 - Едни и същи тестове, които се повтарят отново и отново, **губят** своята ефективност
 - Незасечените досега дефекти остават неоткрити
- Трябва да се разработват нови и модифицирани тестове



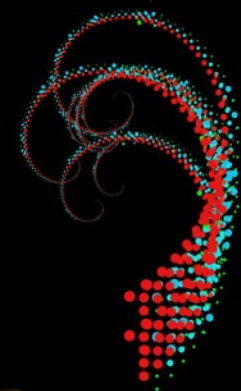
Седемте принципа на тестването (6)

- Тестването показва **наличието** на дефекти, но не доказва **отсъствието** им!
 - Тестването показва съществуващи дефекти
 - Тестването **не доказва** отсъствието на други дефекти
 - Подходящото тестване намаля вероятността от дефекти



Седемте принципа на тестването (7)

- Заблудата за липса на грешки
 - Намирането и оправянето на дефекти само по себе си **не е достатъчно** в тези случаи:
 - Продуктът е направен неудобно, неизползваемо и неоптимално
 - Продуктът не покрива изискванията, нуждите и очакванията на потребителите





Какво е компонентното тестване?

Софтуер, който тества софтуер

Ръчно тестване

- Не е **структурирано**
- Не е **повторимо**
- Не **покрива** целия код
- **Не е толкова лесно** колкото би трябвало да бъде

```
void testSum() {  
    if (this.sum(1, 2) != 3) {  
        throw new Exception("1 + 2 != 3");  
    }  
}
```

Ръчно тестване (2)

- Имаме нужда от **структуриран подход**, който:
 - Позволява **рефакториране**
 - Намаля **цената за промяна**
 - **Намаля** броя на **дефектите** в кода
- Бонус:
 - Подобрява **проектирането**

Автоматично тестване

- Системни тестове
- Интеграционни тестове
- Компонентни тестове



Компонентно тестване



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

