



Автоматизарно тестване на уеб базирани приложения с JAVA

Радостина Стоилова

Радостина Стоилова

Проксиад България



r.stoilova@proxiad.com



0882 66 01 01



[/radostina-stoilova-61011393](#)



[/rstoilova](#)

Програма

Автоматизираното тестване е неделима част от съвременния процес за ефективна разработка на софтуерни продукти. Целта на дисциплината е да запознае студентите с принципите, основите, процесите и инструментите за автоматизирано тестване на уеб базиран софтуер. По време на обучението се използват и демонстрират едни най-популярните инструменти за автоматизирано тестване в Java света (JUnit и Selenium/WebDriver), приложими както в класически Java Enterprise приложения, така и в съвременни JavaScript потребителски интерфейси (ReactJS, AngularJS, Vue.JS и др.). Обучението включва и запознаване с добри практики, често използвани инструменти за управление на дефекти (JIRA, Mantis) и ръчно тестване (Swagger, SoapUI, Selenium IDE). Придобитите умения и знания ще бъдат полезни както за софтуерни разработчици, така и за специалисти по софтуерно тестване.



1

Въведение в тестването на софтуерни продукти (3 лекции)

2

JUnit, Eclipse, Harmcrest, Maven/CI, CI (2 лекции)

3

Selenium/WebDriver (Page Object). Client/Server rendering (3 лекции)

4

Grey Box testing (1 лекция)

5

QA engineer armors – Jira, Mantis, Swagger/Soap UI (1 лекция)

Whiteboard

- <https://app.ziteboard.com/team/5bbb585d-5728-4855-8225-fc7553c1caf6>
- <https://goo.gl/rwYdr6>



**“Software quality engineers don’t
break software, it is already broken”**



Цената на грешките



Грешки, които струват малко или нищо

DD-MM-YY vs. DD/MM/YYYY

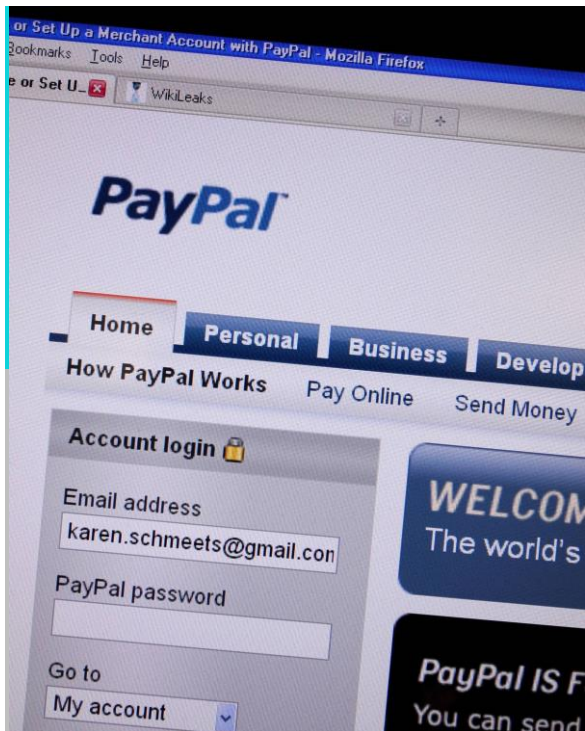
Цената на грешките



Грешки, които струват милиони

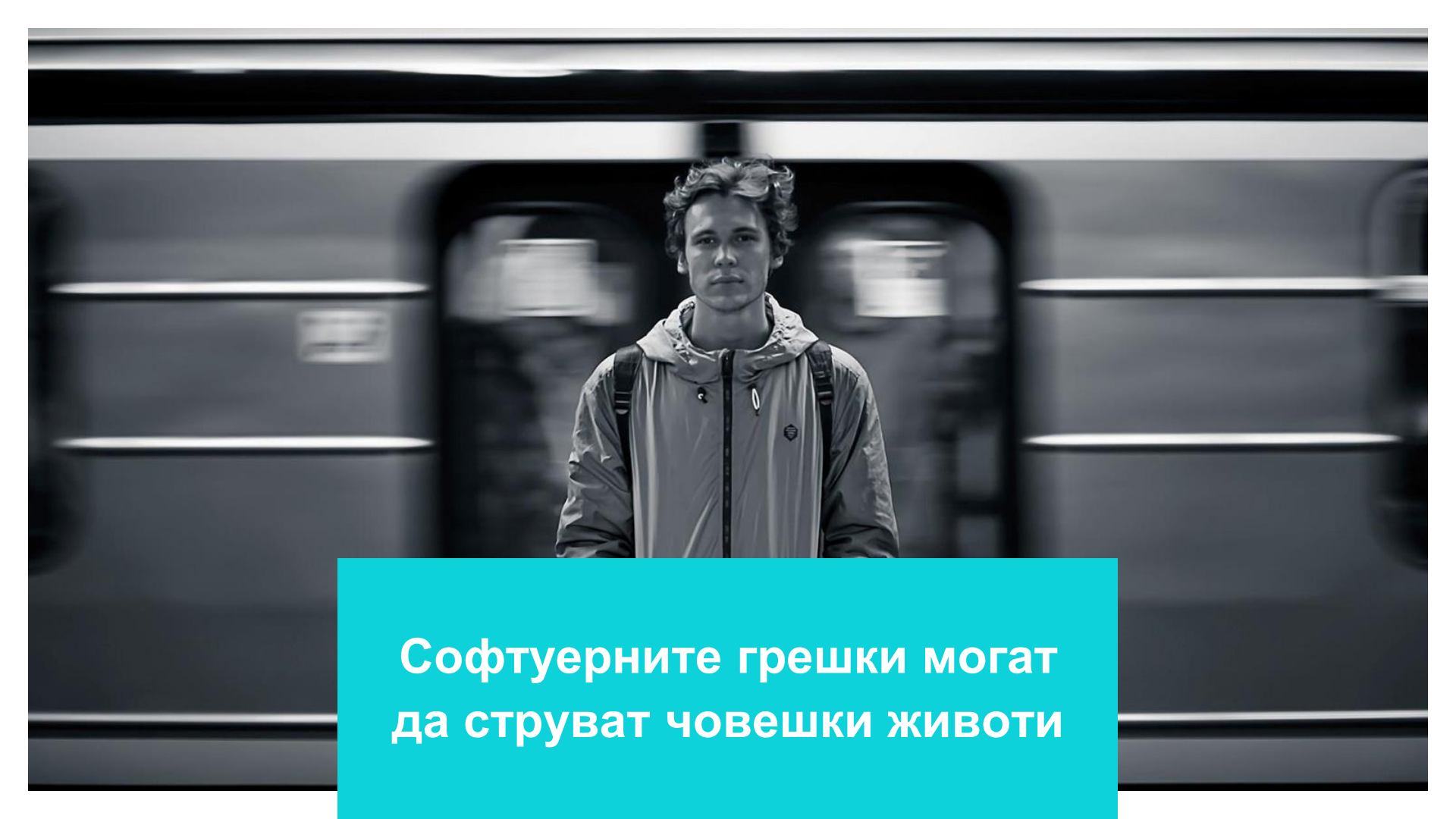
През юни 1995 Европейската космическа агенция изстрелва космическата ракета Ariane 5. Грешка в контролния софтуер, написан на програмният език Ada, причинява саморазрушаване на ракетата 37 секунди след изстрелването

Цената на грешките



Грешки, които влияят на имиджа на компанията

PayPal - \$92 Квадрилиона депозит. През 2013 Крис Рейнълдс се събужда като най-богатия човек на света. В сметката му има \$92,233,720,368,547,800. PayPal признава, че това е причинено от софтуерна грешка и предлага да дари голяма сума пари за кауза, определена от Рейнадлс, за да изчисти имиджа си.



**Софтуерните грешки могат
да струват човешки животи**

Цената на грешките



Грешки, които убиват

През 1980, 5 човека умират след силно радиоактивно облъчване, причинено от бърк в софтуера на машина за радиотерапия Therac-25. Грешката е била допусната при имплементирането на нишки, които се опитват да правят едни и същи неща по едно и също време. Подобна грешка остава без осветление Северна Америка през 2003 за повече от 2 денонощия, 55 милиона човека от 8 щата за засегнати. 12 от тях загиват от коствени обстоятелства около липсата на захранване.

Цената за отстраняване на грешките



Цената за отстраняване на грешките расте експоненциално във времето. За това колко по-рано започне тестването, толкова по-голяма е вероятност грешките да бъдат открити по-рано и съответно толкова по-ниска е цената за тяхното отстраняване.

Nobody is perfect! Are you?



Whiteboard

- <https://app.ziteboard.com/team/5bbb585d-5728-4855-8225-fc7553c1caf6>
- <https://goo.gl/rwYdr6>



Защо допускаме грешки?

Допускане на грешки е естествен процес. Тестването помага да открием тези грешки .
Процесът по осигуряване на качеството (Quality assurance) помага да се учим от тези грешки и
да не ги допускаме в бъдеще.

Причините

- Никой не е перфектен
- Всички правим грешки или пропуски
- Лошо или липсващо обучение
- Лоша комуникация
- Неясни изисквания
- Промяна в изискванията или лошо документиране на изискванията
- ПРЕДПОЛОЖЕНИЯ
-



**„Тестването е сравнение на това какво е с това,
което трябва да бъде“**



Софтуерното тестване е ...

Съществуват много и различни дефиниции на понятието „Софтуерно тестване“

„Процесът на изпълнение на програмата с цел да се удостовери нейното качество“ Милс

В тази дефиниция, не отразява фактът, че тестването открива грешки.

„Процесът на изпълнение на програмата с цел да се открият грешки в нея

Тази дефиниция не отразява фактът, че тестването проверява придържането към изискванията

BS7925-1:

„Процесът на изпълнение на програма, с цел да се провери, че тя удовлетворява изискванията (бизнес и технически) и за да се открият грешки в нея. „

ISTQB:

„Тестването на софтуер е процес на изпълнение на програма или приложение с цел да се намерят програмни грешки. Това е процес на валидиране и проверка дали дадена софтуерна програма, приложение или продукт, отговаря на бизнес и технически изисквания, които са довели до нейното проектиране и разработка.“

IEEE Standard Glossary of Software Engineering Terminology: *“Тестването е процес на работата с дадена система или неин компонент под предварително определни условия, наблюдение и записване на резултатите от работата и извършване на оценка на някои аспекти на системата или компонента.”*



Нива на зрялост на тестването Борис Бейзър



Ниво 0

Няма разлика между дебъгване и тестване.
Тестването няма смисъл раличен от подпомагане на дебъгването



Ниво 1

Целта на тестването е да покаже се софтуерът работи



Ниво 2

Целта на тестването е да докаже, че софтуерът не работи



Ниво 3

Целта на тестването е не да докаже нещо, а да сведе риска от неработеща програма до приемливи нива“.



Ниво 4

Тестването не е акт, то е „състояние на ума“

Принципи на софтуерно тестване



1

Принцип 1

Тестването показва наличието на дефекти

2

Принцип 2:

Пълно (exhaustive) тестване е невъзможно

3

Принцип 3

Ранно тестване

4

Принцип 4

Клъстериране на дефектите

5

Принцип 5

Ако един и същи тест се повтаря отново и отново , в даден момент той ще спре да открива дефекти

6

Принцип 6

Тестването е зависимо от контекста

7

Принцип 7

Заблудата за липса на дефекти

Фундаментален тестов процес



Планиране и контрол

„A goal without a plan is just a wish”



Анализиране и дизайн

„ Analysis gain better understanding”



Имплементация и изпълнение

„Ideas are easy. Implementation is hard”



Оценка и репортинг

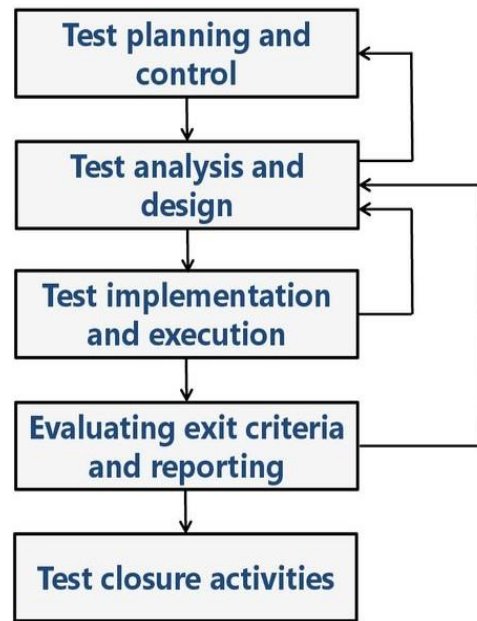
“Evaluation- Learning tool to Improve”



Финализиране

“Closure – Way to start new beginning”

Fundamental Test Process





Фундаментален тестов процес - дейности

| Планиране и контрол | Анализ и дизайн | Имплементация и изпълнение | Оценка и репортинг | Финализиране |
|---|---|---|---|---|
| Дефиниране на стратегия, подход и обхват | Анализ на съществуващите артефакти | Имплементация на тест условията (test conditions) | Документиране на резултатите от тестовите (най-добре паралелно с изпълнението им) | Финализиране на тест кейсовете, скриптовете и данните |
| Определяне на целите и рисковете | Идентифициране условията за тестване | Създаване на тестови пакети (Test suites) | Анализ на инцидентите с цел да се разбере дали тестът е грешен или наистина има открит дефект | Проверка дали всички логнати дефекти са острани и затворени |
| Планиране на ресурсите, тестови среди, инструменти за тестване | Проектиране на тестови сценарии | Изпълнение на тест кейсовете и тест сценариите, използвайки предварително подготвените данни | Взимане на решение дали да се добавят или премахнат някои тестове | Lessons learned |
| Изготвяне на график на времето за тест анализ, проектиране, внедряване и затваряне. | Създаване на среда за тестване на дизайна | Имплементация и изпълнение на автоматизираните тестове в различни среди | | Предаване на всички артефакти към съпорт и поддръжка |
| Планиране бюджета за тестване | Писане на тестови скриптове и данни за тестовите | Повторно изпълнение на тестовите, които са открили бъгове, за да се провери дали дефектите са били отстранени | | |
| Мониторинг на прогреса | Анализ на тестовите инструменти, които са избрани по време на планирането | Анализ на получените резултати по отношение на очакваните | | |
| Измерване и анализ на резултатите | Проектиране на автоматизирането на тестването | Рапортуване на разкритите в резултатите и инцидентите | | |

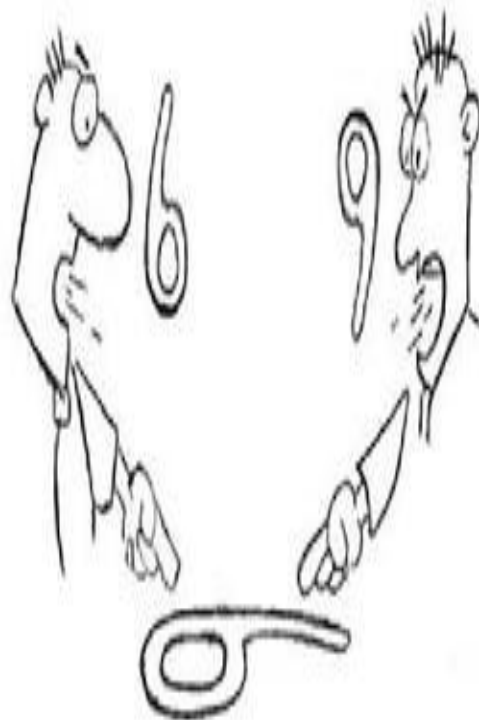
„Defect reports needs to be raised against the software under test, not against the individual who made the mistake “



Developer V/s Tester

Независима гледна точка

Извършване на дейностите по тестването от различен от разработчиците човек или екип запазва независимата гледна точка върху системата . Чрез този подход се запазва безпристрастността и се постига по-високо ниво на ефективност при откриването на дефектите в приложението . ..



Степени на независимост
на тестването

- 1.Тестването се извършва от човека, който пише кода
- 2.Тестването се извършва от човек в същия екип като друг програмист/разработчик
- 3.Тестване от човек от друг екип, например екип от тестери
- 4.Тестването се извършва от друга организация (Test outsourcing, Certification)



Умения на добрия тестер

Комуникация

Сътрудничество вместо битки
Обсъждане на приложението, не на хората
Общ език

Методичност и систематичност

Тактичност и дипломатичност

Скептичност, особено по отношение на
предположенията. Винаги търси доказателства

Внимание към детайлите

Отлични умения за писане и вербална
комуникация (за да обясни грешките ясно и с
нужните детайли)

Желание да експериментира, да опитва различни
неща и да наблюдава и анализира резултите от
действията си

Технически умения в целия
технологичен стек на приложението

Задълбочени познания в Business domain на
клиента.

Умение да приоритизира и организира

Умение да се адаптира и учи