

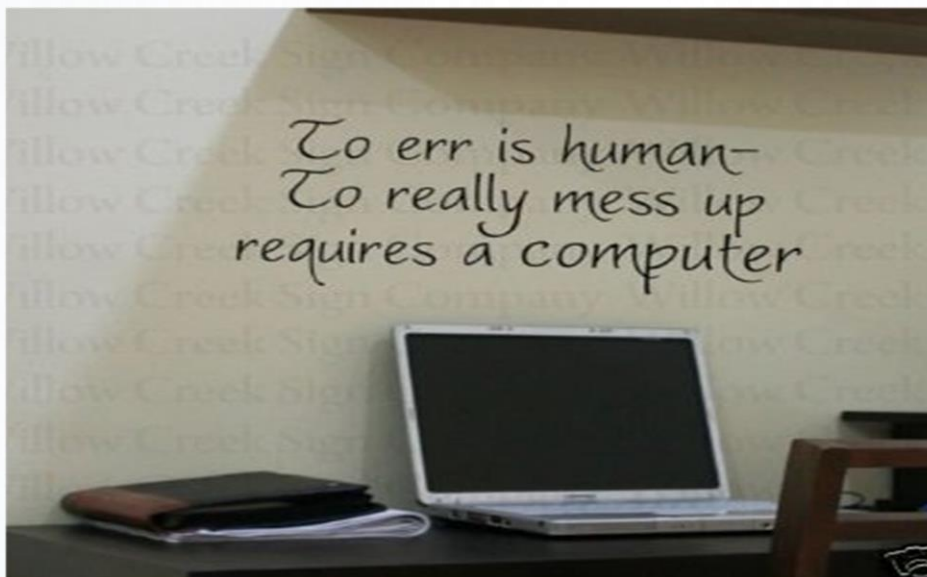
# ОСНОВНИ ПРИНЦИПИ И ПОНЯТИЯ НА СОФТУЕРНОТО ТЕСТВАНЕ

# Съдържание

- ◎ Защо има нужда от софтуерно тестване (software testing)?
  - Контекст на софтуерни системи (Software systems context)
  - Защо се случват софтуерни дефекти (software defects)?
  - Роля на тестването в разработването, поддържането и работата на софтуера
  - Тестване и качество на софтуера
  - Има ли такова нещо като достатъчно тестване?
- ◎ Какво е софтуерно тестване (software testing)?
- ◎ Основни принципи на софтуерното тестване (software testing)

# Защо тестването на софтуера е важно?

<https://youtu.be/TDynSmrzpXw>



# Защо е необходимо тестване на софтуера ?

- Тестването на софтуера (software testing) е необходимо, защото съществуването на грешки в софтуера е неизбежно. Освен намирането на грешка (fault-detection), модерното разбиране за тестването се състои в това, че превенцията на грешки (fault-prevention) (например- ранно откриване на грешки/премахване от изискванията (requirements), дизайни (designs) и т.н. чрез статичните тестове (static tests)) е поне толкова важна колкото намирането на грешки в софтуера чрез изпълнението на динамични тестове (dynamic tests).



# Контекст на софтуерни системи (Software Systems Context)

◎ Софтуерното тестване е зависимо от контекста:

- Бизнес
- Образование
- Медицина
- Военни
- Наука
- Забавление
- Онлайн залагания



# Защо се случват софтуерни дефекти? (1)

- ◎ Всички хора правят грешки
- ◎ Грешките раждат дефекти (Errors produce defects)
  - Дефектите (defects) са повреди (faults / bugs) в програмния код (program code), или в документ (изискване (requirement))
  - Човешката грешка причинява повреди във всеки доставен проект. Повредите може да бъдат причинени от грешки в изискванията, дизайна или кода (requirements, design or coding errors).
- ◎ Ако един дефект в кода е изпълнен(executed), това може да причини провал/неуспех (failure):
  - Не успява (fail) да направи това, което трябва да направи
  - Прави нещо, което не трябва

# Защо се случват софтуерни дефекти? (2)

## ◎ Човешкият фактор

- Хората правят грешки
- Слабо обучение
- Натиск на времето
- Сложност на кода
- Сложност на инфраструктурата
- Променящи се технологии

# Защо се случват софтуерни дефекти? (3)

## ⦿ Организационни фактори

- Неефикасна комуникация
- Неясно дефинирани изисквания (requirements)

## ⦿ Фактори на околната среда

- Радиация, Магнетизъм, Електрически полета, Замърсяване и т.н.
- Тези могат да променят хардуерните условия (hardware conditions)



# Определения

## ⦿ Аномалия (Anomaly)

- Всяко състояние, което се отклонява от очакванията, базирани на requirements specifications, design documents, user documents, standards, и т.н. или от възприятието или опита на някого.

## ⦿ Грешка (Error (mistake))

- човешко действие, което довежда до неправилен резултат

# Определения (2)

- ⦿ Дефект (Bug/Defect/Fault/Problem)
  - Пропуск в един компонент или система, който може да причини, компонентът или системата да не успее да изпълни функцията, която се изисква от него
  - Ако се е срещнал случайно по време на изпълнението (execution) дефект, то той може да причини провал на компонента или системата
- ⦿ Провал/неуспех (Failure)
  - е отклонение на софтуера от неговата очаквана доставка или очаквана услуга
- ⦿ Маскиране на дефект/ повреда (Defect/fault masking)
  - Когато един дефект спира откриването на друг

# Какво е качество на софтуера (Software Quality)? (1)

- ◎ Определение намерено в IEEE (Institute of Electrical and Electronics Engineers) Standard Glossary Of Software Engineering Terminology
  - Степента, до която една система, компонент или процес срещат вече уточнените изисквания (requirements)
  - Степента, до която една система, компонент или процес срещат клиентските или потребителските очаквания

# Какво е качество на софтуера (Software Quality)? (2)

- ◎ Софтуерното функционално качество отразява колко добре то се съобразява с или съответства на един даден дизайн, основан на функционалните изисквания или спецификации. Този атрибут може също да бъде описан, като колко софтуерът отговаря на предназначението или как той се сравнява с конкурентите на пазара, като един продукт, който си заслужава
- ◎ Софтуерното структурно качество се отнася до това как то среща нефункционалните изисквания, които подкрепят доставката на функционалните изисквания, такива като устойчивост и възможност за поддръжка, степента до която софтуера е бил изработен правилно.



# Ролята на тестването

- ◎ Роля на тестването в разработването, поддръжката и работата на софтуера (development, maintenance and operations)
  - Намалява риска от проблеми
  - Намалява в дългосрочно време разходите свързани с дефекти
  - Допринася за качеството на софтуера
  - Помага да срещне стандартите:
    - Договорните или законови изисквания
    - Стандарти специфични за индустрията

# Тестване и качество на софтуера (Testing and Software Quality)

- ⦿ Тестването може да даде увереност в качеството на софтуера, ако намери малко или никакви дефекти
- ⦿ Ако се открият дефекти, качеството се повишава, когато тези дефекти се поправят
- ⦿ Уроците научени от предишни грешки подобряват бъдещото изпълнение

# Колко тестване е достатъчно?

- ◎ Колко тестване трябва да бъде направено е въпрос на риск:
  - Много тестване може да забави пускането на продукта и да повиши неговата цена
  - Недостатъчното тестване крие рискове от грешки в крайния продукт



ILLUSTRATION BY SEGUE TECHNOLOGIES

# И така какво е софтуерно тестване?

**SOFTWARE**  
**TESTING**





# Какво е тестване?

## ⦿ Какво е тестване?

- Процесът на упражняване на софтуер
  - Да провери, че той задоволява специфичните изискванията и да открие грешки
- Процесът на анализиране на софтуерна част
  - Да открие разликите между съществуващите и желаните състояние (това са bugs)
  - Да оцени характеристиките на софтуерната част

# Какво е тестване? (2)

## ⦿ Какво е тестване?

- Процесът на работа на една система или компонент при специфични условия
  - Наблюдавайки или записвайки резултатите
  - Правейки оценка на някой аспект на системата или компонента

# Какво е тестване? (3)

- ◎ Софтуерното тестване е процес, който се използва да идентифицира правилността, пълнотата и качеството на разработен компютърен софтуер.
- ◎ Софтуерното тестване е дейност за проверка дали действителните резултати съвпадат с очакваните и да осигури, че софтуерната система е без дефекти.

# Какво е тестване? (4)

- ◉ Testing is the process of establishing confidence that a program or system does what it is supposed to (Hetzel, 1973).
- ◉ Testing is the process of executing a program or system with the intent of finding errors (Myers, 1979).
- ◉ Software Testing is the process of exercising or evaluating a system by manual or automatic means to verify that it satisfies specified requirements or to identify differences between actual and expected results (IEEE 610.12, 1990)
- ◉ Testing is any activity aimed at evaluating an attribute or capability of a program or system. Testing is the measurement of software quality (Hetzel, 1984)
- ◉ Testing is demonstrating that a system is fit for purpose (Evans, et al, 1996)
- ◉ Testing is a process of planning, preparation and execution to establish the characteristics of a software product and to determine the difference between the actual and required status (Pol and Van Veenendaal, 1996)
- ◉ Software Testing is a process consisting of all life cycle activities concerned with checking software and software-related work products (Gelperin and Hetzel, 1988)
- ◉ Testing is the process of exercising software to verify that it satisfies specified requirements and to detect errors (BS7925-1, 1998)
- ◉ As the objective of a test should be to detect faults, a “successful” test is one that does detect a fault (ISEB, 1999)



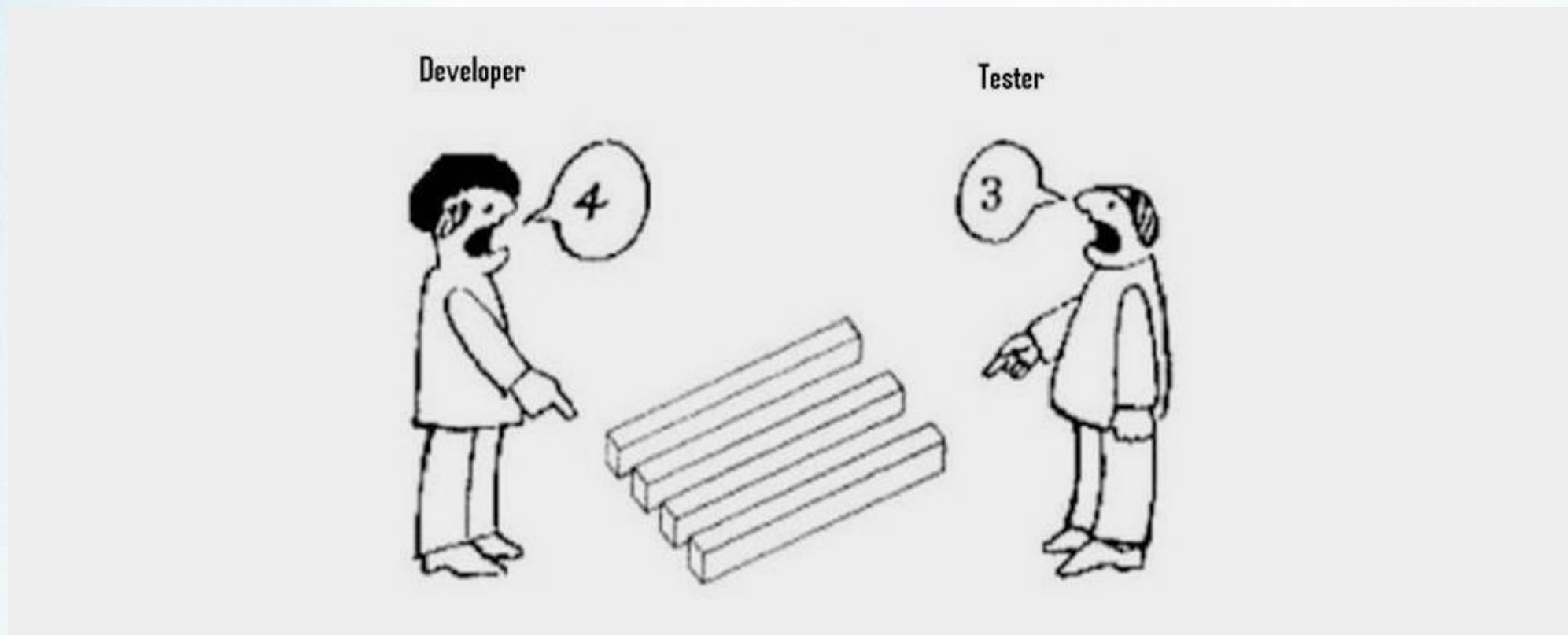
# Главни тестови дейности

- ◎ Тестването не е само пускане на тестове, но също така:
  - Планиране и контрол (Planning and control)
  - Избиране на условия на тестване (Choosing test conditions)
  - Проектиране и изпълнение на тестове (Designing and executing test cases)
  - Проверяване на резултатите (Checking results)
  - Оценяване на критериите за край (Evaluating exit criteria)
  - Докладване на тестовия процес и системата, която се тества (Reporting on the testing process and system under test)
  - Завършване на дейностите по приключване (Completing closure activities)

# Основен фокус в софтуерното тестване

- ◎ Тестването преследва няколко цели:
  - Намиране на дефекти
  - Печелене на увереност относно нивото на качество
  - Предоставяне на информация за вземане на решение
  - Предотвратяване на дефекти

# Различни гледни точки



# Различни гледни точки (2)

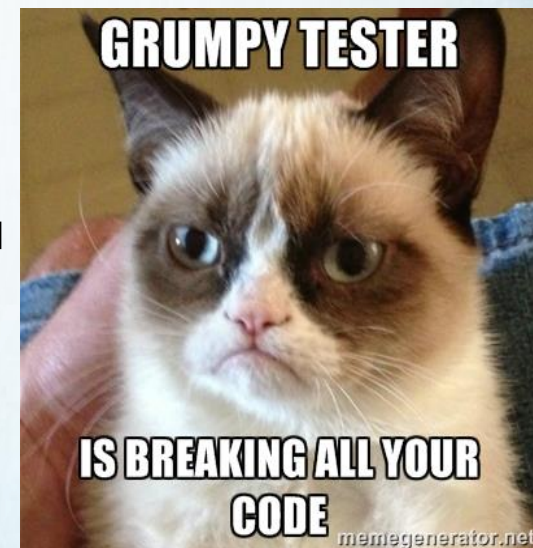
- ◎ Фокусът на софтуерното тестване се различава според гледната точка:
  - От гледна точка на програмистите:
    - Се възприемат като много креативни (градивни)— те пишат код, без който няма да има система
    - Те са рядко добри комуникатори
    - Могат често да специализират в само едно или две умения (e.g. VB, C++, JAVA, SQL)





# Различни гледни точки (2)

- ◎ Фокусът на софтуерното тестване се различава според гледната точка:
  - От гледна точка на QA инженерите:
    - Се възприемат като разрушителни- щастливи са само, когато са намерили повреди!
    - Обикновено се изискват добри комуникационни умения, такт и дипломатичност.
    - Обикновено е нужно да бъде multi-talented (технически умения и умения за тестване, работа в екип).



# Debugging VS Testing



- ◎ Процес на софтуерно тестване
  - Дейност, която първоначално намира повреди в софтуерната част
- ◎ Отстраняване на грешки (Debugging)
  - Дейност по разработка, която намира, анализира и премахва причината за повредата
  - Отстраняването на грешки (Debugging) е дейност, която подпомага тестването, но не може да го замени
- ◎ Последващо повторно тестване от тестер
  - Гарантира, че поправката наистина е оправила повредата

# Верификация и валидация (Verification and Validation)

- Верификацията е проверка или тестване на части, включително софтуер, за съответствие и съгласуваност със свързана спецификация.

Verification is the checking or testing of items, including software, for conformance and consistency with an associated specification. (Are we doing the job right?)

- Валидацията е процес на проверка, че това което е било зададено е това което всъщност потребителя желае.

Validation is the process of checking that what has been specified is what the user actually wanted. (Are we doing the right job?)

# Седемте основни принципа на тестването (The Basic Seven Testing Principles)

- Тестването показва наличието на дефекти
- Пълно тестване е невъзможно (Exhaustive testing is impossible)
- Ранно тестване (Early Testing)
- Групиране на дефекти (Defect Clustering)
- Парадокса на повтаряемост на дефекти (Pesticide Paradox)
- Тестването зависи от контекста (Testing is context dependent)
- Заблудата за липса на грешки



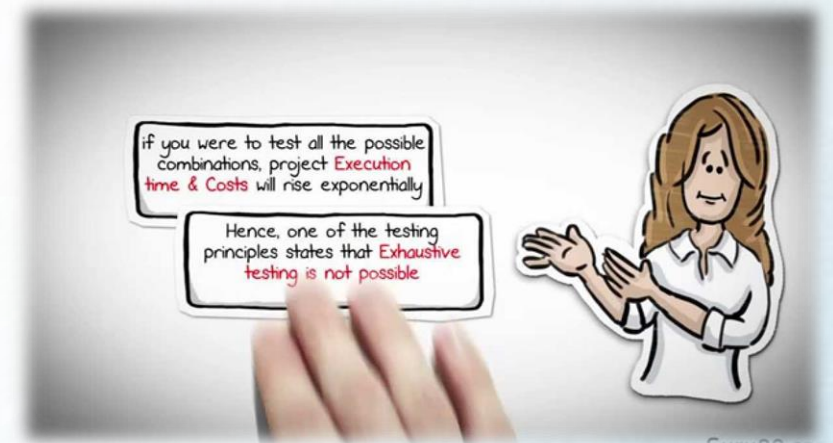


# Тестването показва наличието на дефекти

- Тестването може да покаже, че дефектите присъстват
- Не може да докаже, че няма дефекти
- Подходящото тестване намалява вероятността за дефекти

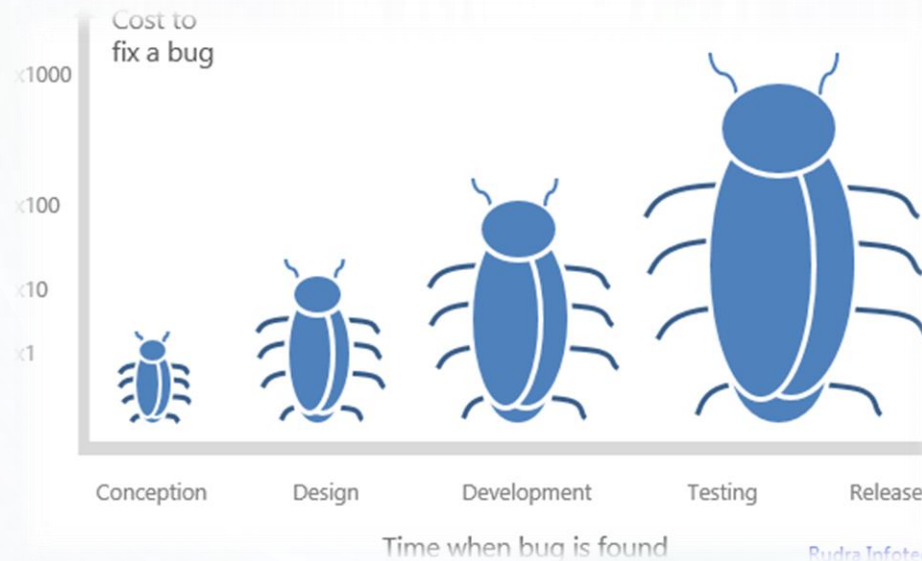
# Пълно тестване е невъзможно

- Всички комбинации на входни данни и предварителни условия са обикновено почти безкраен брой
- Не е осъществимо тестване на всичко
  - С изключение на тривиални случаи
- Анализът на риска и приоритетите трябва да се използват за фокусиране усилията за тестване



# Ранно тестване

- Дейностите по тестване трябва да започнат възможно най-рано
  - И трябва да бъдат фокусирани върху определени цели
- Колкото по-късно се открие един bug, толкова по-скъпо струва той!



# Групиране на дефекти

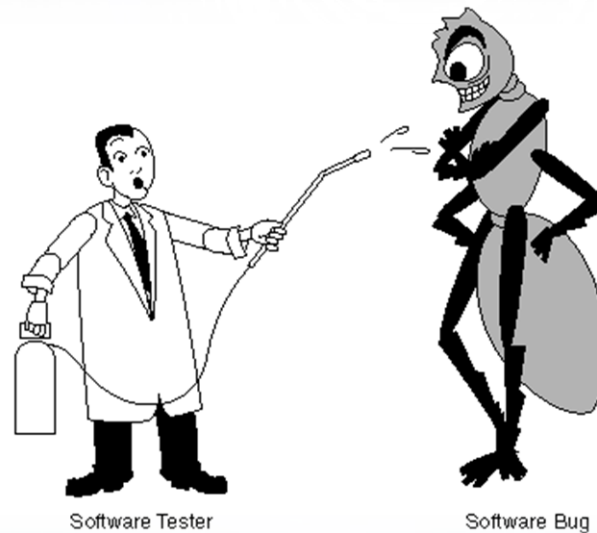
## Групиране на дефекти

- Усилието за тестване трябва да бъде фокусирано пропорционално
  - Към очакваните и последащи дефекти в модулите
- Един малък брой от модули обикновено съдържат повечето от откритите дефекти
  - Отговорни са за повечето оперативни повреди



# Парадокса на повтаряемост на дефекти (Pesticide paradox)

- Ако едни и същи тестове се повтарят отново и отново то те са склонни да загубят тяхната ефективност
  - Дефектите, които не са били открити преди остават неразкрити
- Трябва да бъдат разработени нови и модифицирани тестове



## Заблудата за липса на грешки (Absence of errors – fallacy)

- Намирането и поправянето на дефекти не помага в следните случаи:
  - Създадената система не е използвана
  - Не изпълнява нуждите и очакванията на потребителите



VS.



# Липсата на грешки не бива да ни заблуждава

- Откриването и отстраняването на проблеми и дефекти не помага в следните ситуации:
  - Конструираната система е нестабилна
  - Не покрива нуждите на потребителя и неговите очаквания



# Допълнителни тестови принципи

- ◎ Тестването трябва да бъде направено от независима страна
- ◎ Възложете на най-добрите служители задачата
- ◎ Тествайте за невалидни и неочаквани входни данни както и валидни условия
- ◎ Дръжте софтуера статичен по време на тестването.
- ◎ Осигурете очаквани тестови резултати, ако е възможно.



Колко сериозно един QA приема своята  
работа😊



# Интересни линкове

- ◎ <http://www.devtopics.com/20-famous-software-disasters/>
- ◎ <http://www.computerworlduk.com/galleries/infrastructure/top-10-software-failures-of-2014-3599618/#23>
- ◎ Когато липсват QA като част от процеса на разработване:  
<https://youtu.be/xtpyjPrpyX8?t=50>

