

Testování webových aplikací

Václav Sobotka, Zdeněk Věcek

A series of horizontal stripes in various colors (yellow, green, blue, dark blue, orange, red, white, light green) spanning the width of the page at the bottom.

Co vás čeká

- Co je testování?
- Co všechno lze testovat?
- Cíle testování
- Proč testovat
- Přínosy
- Testování obecně
- Etapy testování
- Metody testování
- User eXperience, příklad
- Test Driven Development
- junit
- Selenium
- Mockování
- Nástroje pro testování

Co je testování?

- Běh testů je pouze částí
- Výzkum kvality testovaného produktu
- Může nahradit zajištění kvality
- Odhaluje přítomnost chyb, ne jejich nepřítomnost!
- **Promítá se do celého vývoje!**

Co všechno lze testovat?

- Fšecho...
- Návrh, dokumentace
- Metody, skripty
- Funkcionality, řešení, konečný produkt
- Bezpečnost, upgrade
- UX, lokalizovatelnost, ...

Cíle testování

- Bezchybný produkt není cílem
- Hledání chyb
- Měření kvality
- Sběr informací pro rozhodování
- Prevence vzniku chyb

Proč testovat

- Člověk je omylný
- Možné škody
 - ztráta peněz (bankovníctví)
 - ztráta času (crash)
 - špatná reputace (Ride to Hell: Retribution)
 - zranění (automatizace)
 - smrt (zdravotnictví)
 - <http://www.cs.tau.ac.il/~nachumd/horror.html>

Přínosy

- snížení rizika výskytu chyb
- zvýšení kvality produktu
- snadnější údržba
- spokojení uživatelé

Testování obecně

- Testování = debugging ?
- Nope.

TESTING	DEBUGGING
a) Finding and locating of a defect	a) Fixing that defect
b) Done by Testing Team	b) Done by Development team
c) Intention behind is to find as many defect as possible	c) Intention is to remove those defects

© ianswer4u.com

Testing V/s Debugging

Testování obecně

- Testujte “tak akorát”
- Čím dříve se chyba odhalí, tím levnější je oprava

Cost to fix a defect		Time detected				
		Requirements	Architecture	Construction	System test	Post-release
Time introduced	Requirements	1×	3×	5–10×	10×	10–100×
	Architecture	–	1×	10×	15×	25–100×
	Construction	–	–	1×	10×	10–25×

Testování obecně

- Nevyhnutelné “omezení shora”
- Plán postupného testování lze stanovit již nad dokumentací
- ISTQB - Certified Tester:
 - <http://www.istqb.org/downloads/finish/16/15.html>

Etapy - dle stádia projektu

- Zahajovací fáze
 - dokumentace, bez funkčního kódu
- Produkční fáze
 - nové funkcionality, řešení dle popisu chování
- Dokončení, předávání
 - celý produkt, instalační balíček, upgrade

Etapy - dle stádia vývoje

- Developer's Unit testing - metody objektů, fce skriptů
- Feature testing - úzce zaměřené scénáře (uživatel)
- Integration testing - kolize nových modulů se starými
- Solution Set testing - nově přidané řešení
- System testing - automatizované zpětné ověření
- Acceptance testing - klient, projekt vyhověl představám
- Pilot testing - napojení na reálné prostředí

Metody

- Statické testování
 - nevyžaduje běh sw
 - lze provádět před samotným psaním kódu
 - (přesnější odhady na čas a zdroje)
- Dynamické testování
 - poskytování vstupů a posuzování výstupů

Metody

- Černá skříňka
 - závislost výstupů na vstupech
 - bez kontextu implementace
- Bílá skříňka
 - testy na základě znalosti kódu
- “Šedá” skříňka
 - známe například matematické principy

Metody

- Hledání úspěchu
 - test-to-pass
 - u prvních verzí k testování
 - za úspěšnou je považována taková funkcionálnita, k jejíž splnění vede alespoň jedna úspěšná cesta

Metody

- Hledání neúspěchu
 - test-to-fail, ke konci vývoje
 - analogicky k předchozí
 - je třeba stanovit časový rámec, chyby lze opravovat věčně
- Procházka po okraji
 - test-to-fail, hledání hraničních hodnot

Metody

- Automatické testování
 - stroj
 - opakované spouštění velkého množství testů
 - zátěžové testy
 - užití generovaných dat
 - **detekuje změnu chování - (ne)chtěnou**

Metody

- Manuální testování
 - člověk
 - je třeba lidský úsudek, hodnocení
 - rozličné přístupy

User eXperience - bonus

- Testovat samotný návrh přímo s uživateli
- Uživatelské testování
 - kvalitativní - více kol s malým počtem testerů
 - dochází k přeceňování výběru uživatelů
 - série úkolů, záznam chování, think aloud
- Alternativou je vzdálené testování
 - [OpenHallway.com](https://www.openhallway.com), [UserTesting.com](https://www.usertesting.com), ...

User eXperience - bonus

- A/B testování
 - rozhoduje, který návrh je lepší, neříká proč
 - dva identické weby s rozdílným formulářem
 - vhodné zejména pro testování layoutů
- User feedback
 - cenné informace přímo od cílové skupiny
 - kdo to dělá...

User eXperience - bonus

- Eye Tracking
 - nákladné
 - marketingový nástroj
- Cliskstream analysis
 - sledování chování uživatelů
 - Myx.cz, CrazyEgg.com, ...

Test Driven Development - TDD

Napiš testy, poté vyvíjej..

- Návrh rozhraní a spolupráce tříd
- Implementace je druhořadá
- Čistý a udržitelný kód

Způsoby psaní testů

- Keep It Simple Stupid
- You Aren't Gonna Need It
- Fake It Till You Make It
- as If You Meant It
- Outside-In

Bottom-Up

vs

Outside-In

Bottom-Up



User Interface

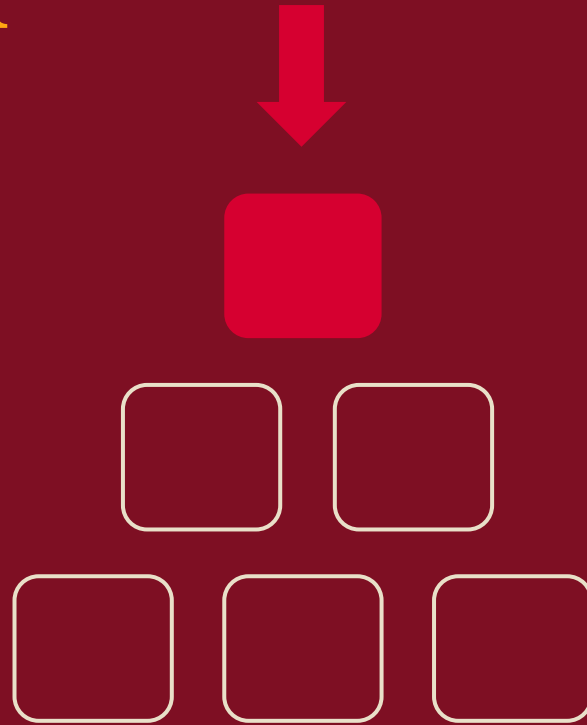
Controllers

Domain model

Data access

DB schema

Outside-In



User Interface

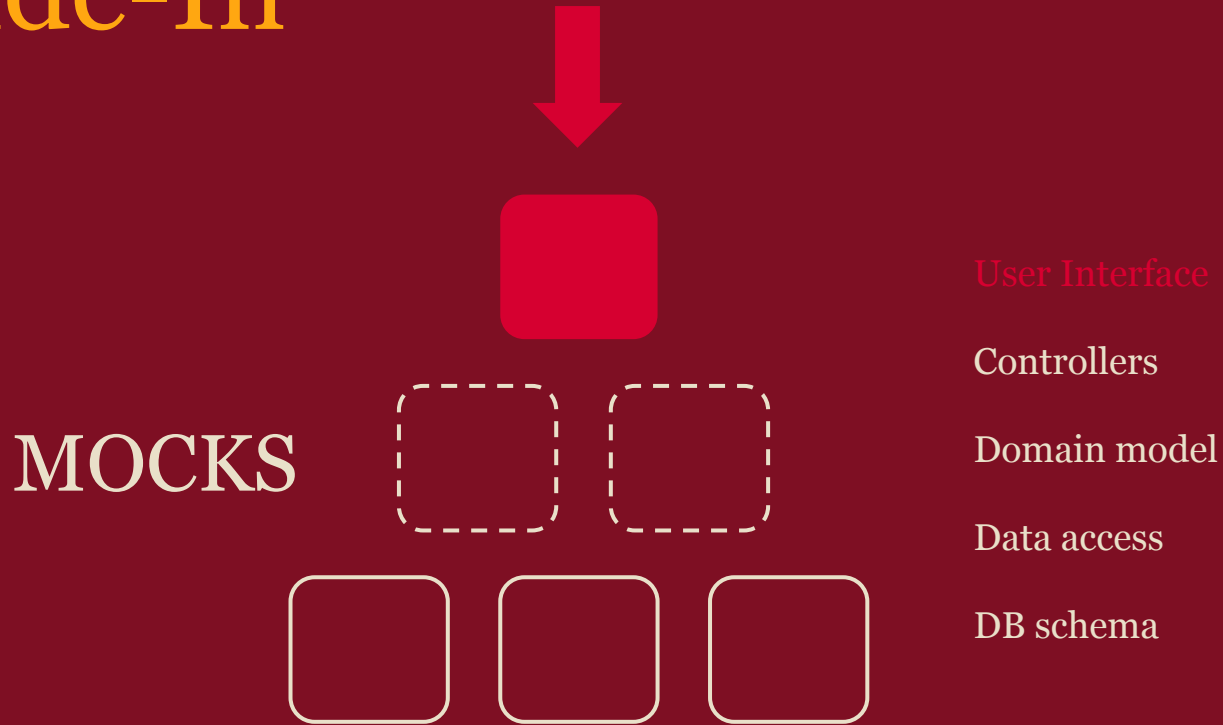
Controllers

Domain model

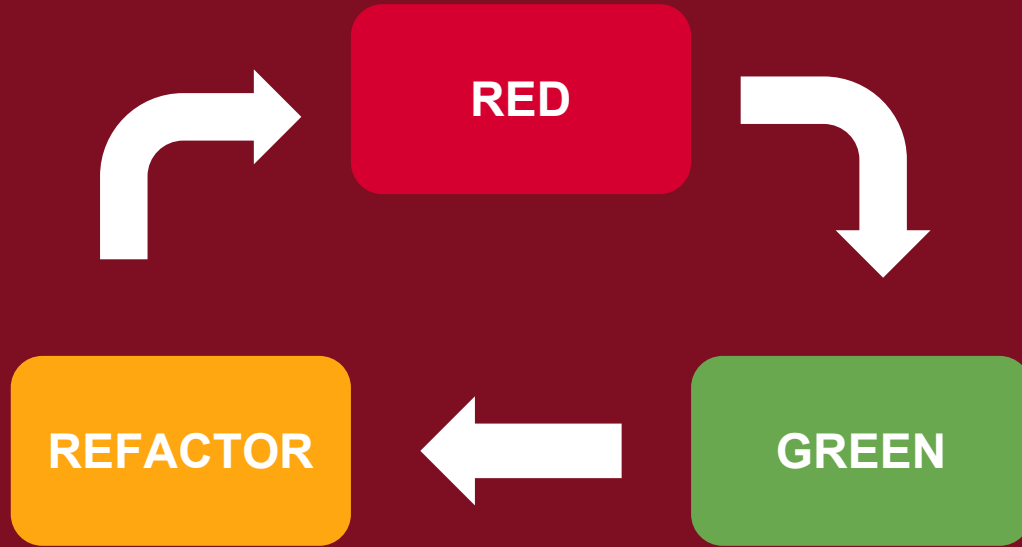
Data access

DB schema

Outside-In



Psaní testu začíná červenou..



Postup

1. Přidej metodu
2. Zkontroluj, že neprojde
3. Zprovozni všechny testy
4. Proved' refactor
5. Vše opakuj

Externí kvalita
VS
Interní kvalita

Externí kvalita

Jak moc dobře systém splňuje požadavky svých zákazníků a uživatelů

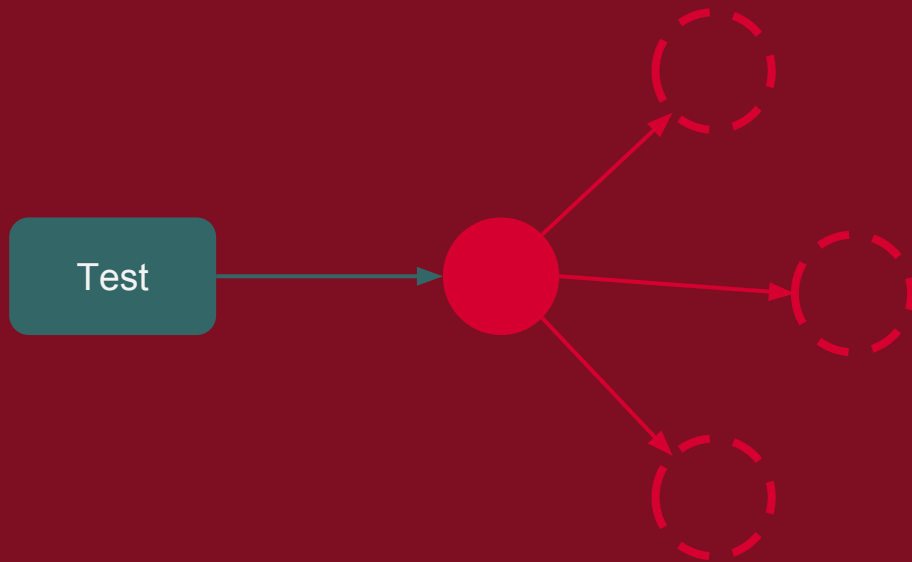
- Funkčnost
- Důvěryhodnost
- Spolehlivost
- Rychlost
- Responzivní

Interní kvalita

Jak moc dobře systém vyhovuje vývojářům a administrátorům

- Snadný na pochopení
- Lehké úpravy

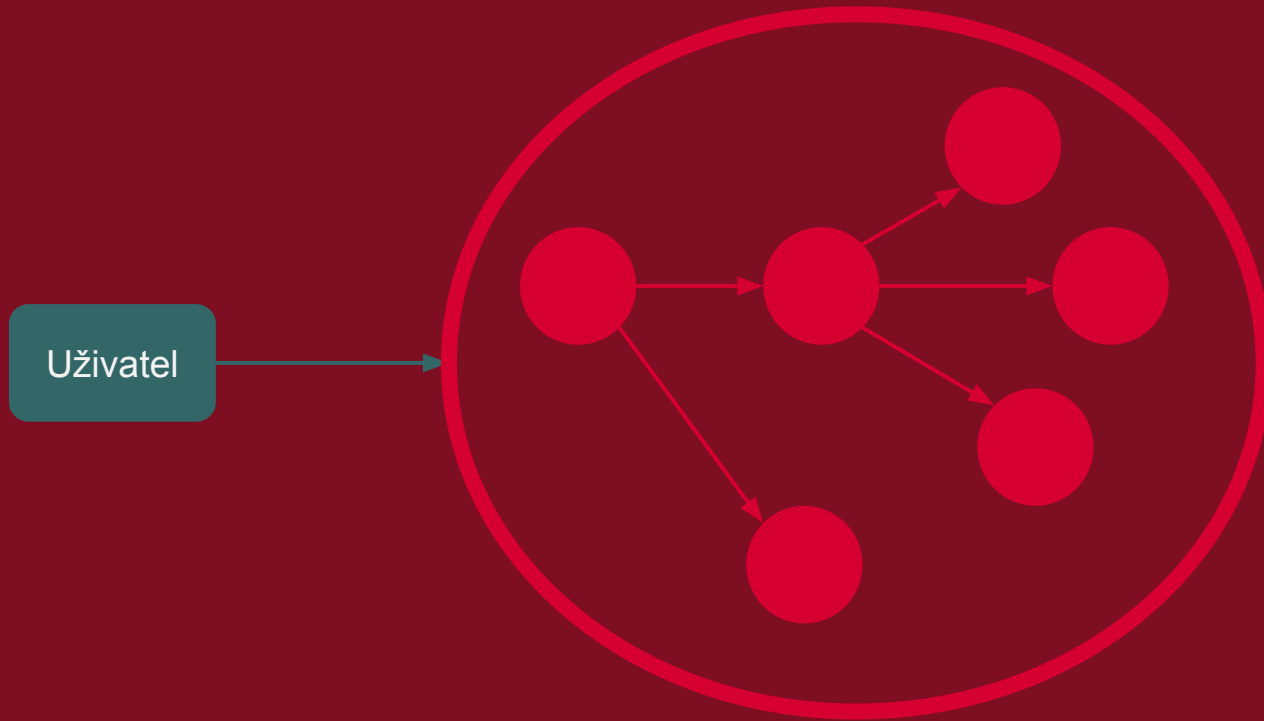
Unit test



Unit test ověří kvalitu jedné jednotky v izolaci namockováním všech závislostí.

Dobrým ukazatelem interní kvality.

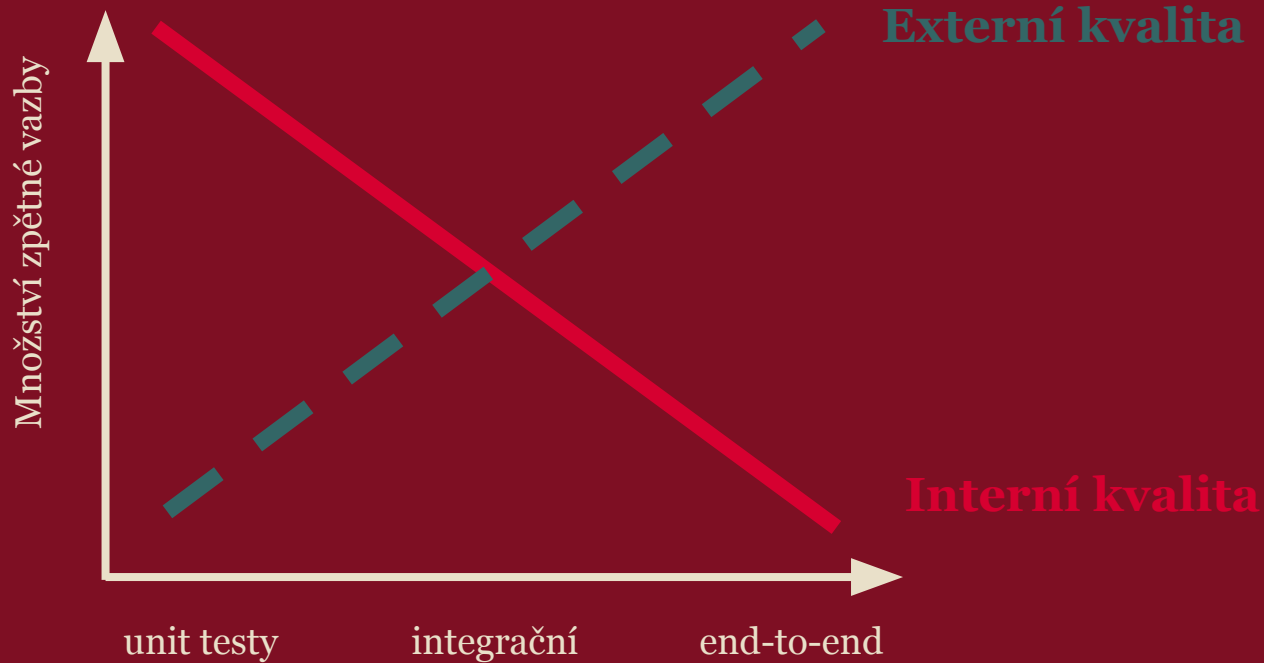
End-to-End test



Uživatel vnímá systém jako celek.

End-to-End testy jsou dobrým ukazatelem externí kvality.

Externí a interní kvalita



Unit testy - jUnit

- Jednotkový framework v jazyce JAVA
- Z rodiny xUnit testů
- Standardní nástroj pro TDD
- Součástí většiny IDE

Výhody

- Neopakují tutéž chybu znovu
- Psaní lepších rozhraní
- Snadnější refactoring
- Lepší pochopení kódu
- Rychlejší psaní
- Snazší odhalení chyby

Nevýhody

- Více napsaného kódu
- Špatně se píše do již existující aplikace

Příklad 1

```
public class BowlTest extends TestCase {  
    public void testEmptyList() {  
        Bowl emptyBowl = new Bowl();  
        List emptyList = new ArrayList();  
        assertEquals("Size of an empty list should be zero.",0, emptyList.size());  
        assertTrue("An empty bowl should report empty.", emptyBowl.isEmpty());  
    }  
}
```

Příklad 2

```
@RunWith(SpringJUnit4ClassRunner.class)
```

```
@ContextConfiguration(locations = { "classpath:/META-INF/spring/applicationContext.xml" })
```

```
public class BowlTest{
```

```
    @Test
```

```
    public void testEmptyList() {
```

```
        Bowl emptyBowl = new Bowl();
```

```
        List emptyList = new ArrayList();
```

```
        Assert.assertEquals("Size of an empty list should be zero.",0, emptyList.size());
```

```
        Assert.assertTrue("An empty bowl should report empty.", emptyBowl.isEmpty());
```

```
    }
```

```
}
```


Metody třídy Assert

- `assertTrue(String message, Boolean test)`
- `assertFalse(String message, Boolean test)`
- `assertNull(String message, Object object)`
- `assertNotNull(String message, Object object)`
- `assertEquals(String message, Object expected, Object actual)` (equals method)
- `assertSame(String message, Object expected, Object actual)` (uses == operator)
- `assertNotSame(String message, Object expected, Object actual)`

Selenium

- Nástroj automatického testování
- Skládá se z několika částí
 - IDE, RC, WebDriver, Grid
- Multiplatformní

Selenium IDE

- Tvorba maker podobně jako v MS Office
- Firefox plugin v prohlížeči
- Selensense
 - Specifikace příkazu
 - Argument
 - Argument

Selenium RC, WebDriver

- Psaní testů v mnoha programovacích jazycích (Java, C#,...)
- Více prohlížečů
- Využití jUnit

Test2.html - Selenium IDE 2.5.0

Soubor (F) Upravit Actions Options Nápověda

Base URL <http://www.csfd.cz/>

Fast Slow

Test Case

Test2

Runs: 1

Failures: 0

Table Source

Command	Target	Value
open	/film/257071-zmizeni/	
type	id=mainSearch	Prisoners
clickAndWait	css=input.submit	
clickAndWait	link=Zmizení	
type	id=mainSearch	K-Pax
clickAndWait	css=input.submit	
clickAndWait	link=Svět podle Prota	

Command: open

Target: /film/257071-zmizeni/

Value:

Log Reference UI-Element Rollup Info Clear

[info] Executing: |open | /film/257071-zmizeni/ | |

[info] Executing: |type | id=mainSearch | Prisoners |

[info] Executing: |clickAndWait | css=input.submit | |

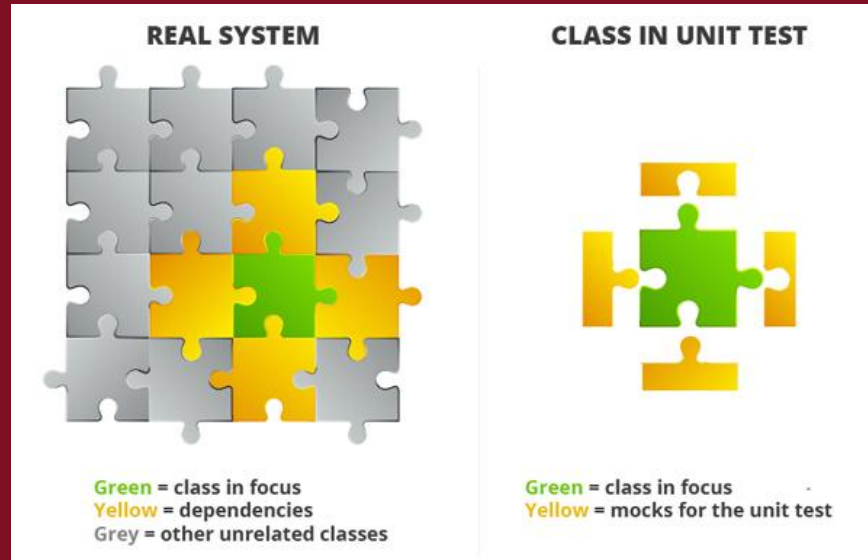
[info] Executing: |clickAndWait | link=Zmizení | |

[info] Executing: |type | id=mainSearch | K-Pax |

[info] Executing: |clickAndWait | css=input.submit | |

Mockování

- Simulace objektů a chování např. dat z DB, závislostí, dat z view



Nástroje pro testování

Jednotkové testy

- junit
- jTest
- Cactus
- TestNG

Funkční testování

- Selenium
- Windmill
- Watin
- Watir
- Sahi

Mockování

- jMock
- EasyMock
- mockito

Zdroje

- <http://www.slideshare.net/danielkolman1/tdd-outsidein>
- <http://blog.kolman.cz/2012/03/tdd-outside-in.html>
- <http://blog.kolman.cz/2012/04/tdd-as-if-you-meant-it.html>
- <http://www.agiledata.org/essays/tdd.html>
- <http://www.root.cz/clanky/test-driven-development-v-jazyce-java/>
- <http://www.infoq.com/presentations/TDD-as-if-You-Meant-It>
- <http://www.tomas-dvorak.cz/clanky/k-cemu-dohaje-nejaky-junit-testy>
- http://cs.wikipedia.org/wiki/Selenium_%28software%29
- <http://zeroturnaround.com/rebellabs/how-to-mock-up-your-unit-test-environment-to-create-alternate-realities/>
- http://en.wikipedia.org/wiki/Software_testing#Testing_levels
- http://cs.wikipedia.org/wiki/Testov%C3%A1n%C3%AD_softwaru
- https://docs.google.com/file/d/oB6-XMUrK_hQLR2ltXzRNdTVtUoE/edit?pli=1
- <http://www.istqb.org/downloads/finish/16/15.html>
- <http://www.ianswer4u.com/2012/06/testing-and-debugging.html>
- <http://www.cs.tau.ac.il/~nachumd/horror.html>

Hezký den!!

