

Václav Dedík

KONTAKTNÍ INFORMACE

Rudíkovy 93
Třemešná
79382 Česká Republika

Mobil: +420 773 450 960
E-mail: vaclav.dedik@gmail.com
Web: fi.muni.cz/~xdedik1/PB029

VZDĚLÁNÍ

Masarykova Univerzita, Brno, Česká Republika

Bakalářské studium, Fakulta Informatiky, 2010 - ...

- Obor: Aplikovaná informatika

Střední škola průmyslová, Krnov, Česká Republika

Maturitní studium, 2006 - 2010

- Obor: Elektronické počítačové systémy

Základní škola, Třemešná, Česká Republika

1997 - 2006

POČÍTAČOVÉ DOVEDNOSTI

Programovací jazyky, značkovací jazyky a jiné technologie

Velmi pokročilá znalost jazyka **Java**.

- zkušenost s vývojem na platformách Java SE a Java EE
- zkušenost s technologiemi: Swing, JDBC, JSP, JSTL, Java Servlet Technology, Java Portlet Technology, EJB, JSF, JAX-RS, JPA, Spring (AOP, IoC, Security, MVC, JDBC), Maven, Tomcat, Glassfish, JodaTime, JSR 303, Apache Tiles, SLF4J, LOG4J, JUnit, Mockito

Základní znalost jazyka **C**.

Základní znalost funkcionálního paradigma a jazyka **Haskell**.

Pokročilá znalost jazyka **HTML**.

Základní znalost jazyka **JavaScript**.

- zkušenost s technologiemi: jQuery, AJAX, JSON

Základní znalost jazyka **CSS**.

- jak CSS2 tak CSS3
- zkušenost s preprocesorem LESS

Pokročilá znalost značkovacího meta-jazyka **XML**.

- zkušenost s technologiemi: XPath, XQuery, XML Schema, XSLT
- zkušenost s XML databází eXist

Pokročilá znalost dotazovacího jazyka **SQL**.

- zkušenost s databázemi PostgreSQL, MySQL

Ostatní: DTD, DOM, Docbook, Bash

Sázecí systémy

Základní znalost sázecího systému **Latex**.

Operační systémy

Pokročilá znalost OS **GNU/Linux**.

Pokročilá znalost OS **UNIX**.

Pokročilá znalost OS **Microsoft Windows**.

Vývojové nástroje: IntelliJ IDEA, NetBeans, gedit, vim

KURZY **Software Summer Camp** pořádán firmou IBA.cz, podzim 2011

JAZYKY **Angličtina** pokročilý
 Němčina začátečník
 Čeština rodilý mluvčí

KONÍČKY Programovací jazyky, filmy, seriály, hudba.

Testování software

Václav Dedík

###

Testování softwaru je obecně chápáno jako technická kontrola funkčnosti testovaného produktu nebo služby. Rozděluje se na ruční a automatizované. Automatizované testování je nedílnou součástí některých agilních metodik vývoje, například Test-driven developmentu (dále TDD) nebo Extreme programmingu. Rozlišujeme několik typů testování podle účelu [1]:

1. Jednotkové testování
2. Integrační testování
3. Funkční testování
4. Akceptační testování
5. Testy uživatelské přívětivosti
6. Bezpečnostní testování
7. Výkonostní testování

1 Jednotkové testování

Jednotkové testování se nachází při vývoji softwaru na prvním místě. Cílem jednotkového testování je otestovat funkčnost programu na nejnižší možné úrovni, nejčastěji na úrovni tříd a metod.

V případě TDD je jednou z nejdůležitější součástí vývoje, vývojář se totiž napsáním jednotkového testu ujistí, že přesně chápe funkcionalitu a požadavky na onu testovanou komponentu, čímž se při psaní samotného kódu eliminuje, odchýlení se od původního záměru a cíle [2]. Jednotkové testování také velmi napomáhá k zajištění lepší dekompozice programu [3].

2 Integrační testování

Integrační testování je fáze vývoje, která většinou následuje po jednotkovém testování. Integrační testování slouží k otestování vzájemné kooperace více testovaných komponent, které jsou otestované jednotkovými testy.

3 Funkční testování

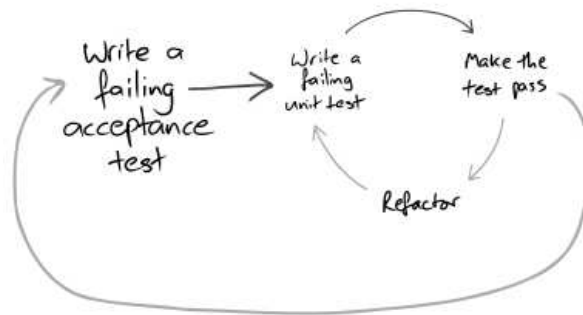
Funkční testování je podobné integračnímu testování, liší se tím, že jeho cílem je otestovat funkčnost celé aplikace bez jakékoliv znalosti vnitřní implementace. A navíc vyžaduje nasazení aplikace v produkčním prostředí.

4 Akceptační testování

Akceptační testování se od funkčního testování liší účelem, nejčastěji je prováděno zákazníkem a slouží k ověření, že vyvíjená aplikace souhlasí se specifikací a naplňuje požadavky zákazníka.

Akceptační testování nelze moc dobře provádět automatizovaně, zákazník musí většinou zajistit otestování například svými zaměstnanci.

V případě TDD se akceptační testy píšou podobně jako funkční testy, jejich účel je ale podobný jednotkovým testům, slouží vývojáři ke správnému pochopení specifikace vyvíjené aplikace [3].



Cyklus vývoje metodikou TDD [3].

5 Testy uživatelské přívětivosti

Účelem testů uživatelské přívětivosti je zajistit jednoduchou a produktivní práci s vyvíjeným softwarem koncovým uživatelem. Nelze provádět automatizovaně a je velmi drahé, jelikož často nelze jednoduše analyzovat chybu, která způsobuje špatnou práci s produktem.

6 Bezpečnostní testování

Bezpečnostní testování je velmi důležitou součástí vývoje softwaru, jeho cílem je odhalit bezpečnostní chyby, které při vývoji vznikají. Příkladem bezpečnostních chyb může být XSS zranitelnost, SQL injection nebo code injection. Testování bezpečnostních chyb se dá provádět buď ručně, což je velmi nespolehlivé, nebo automatizovaně např. pomocí různých utilit třetích stran.

7 Výkonostní testování

Výkonostní testování má za úkol zajistit škálovatelnost a přípustnou latenci vyvíjené aplikace. Škálovatelností se rozumí stejné chování aplikace při nízkém i vysokém vytížení aplikace. Latencí se rozumí doba, kterou je nutné čekat na zpracování požadavku odeslaného klientem.

Reference

- [1] Tomáš Pitner. Testování aplikací, jednotkové testy, junit. <http://goo.gl/eaWx6>.
- [2] Wikipedia. Software testing. http://en.wikipedia.org/wiki/Software_testing.
- [3] Steeve Freeman and Nat Pryce. *Growing Object-Oriented Software, Guided by Tests*. Addison-Wesley Professional, 2009.

Rejstřík

A

Akceptační testování, 3

B

Bezpečnostní testování, 3, 4

F

Funkční testování, 3

I

Integrační testování, 3

J

Jednotkové testování, 3

M

Metodiky vývoje, 3

Extreme programing, 3

Test-driven development, 3

T

Testy uživatelské přívětivosti, 3, 4

V

Výkonostní testování, 3, 4