



profinit.
PROFESSIONALS IN IT

QA & Dokumentace



Agenda

- Docházka
- Návrat k minulému praktickému cvičení
- Zápočtové práce
- QA opakování
- Dokumentace
 - Co, jak a proč dokumentovat
- Dotazy, přání, stížnosti



Kde je chyba?

```
public static StringBuilder sbd  
    = new StringBuilder();
```

@Test

```
public void test() {  
    sbd.clear();  
    sbd.append("ab");  
    Assert.assertEquals(sbd.toString(), "ab");  
}
```

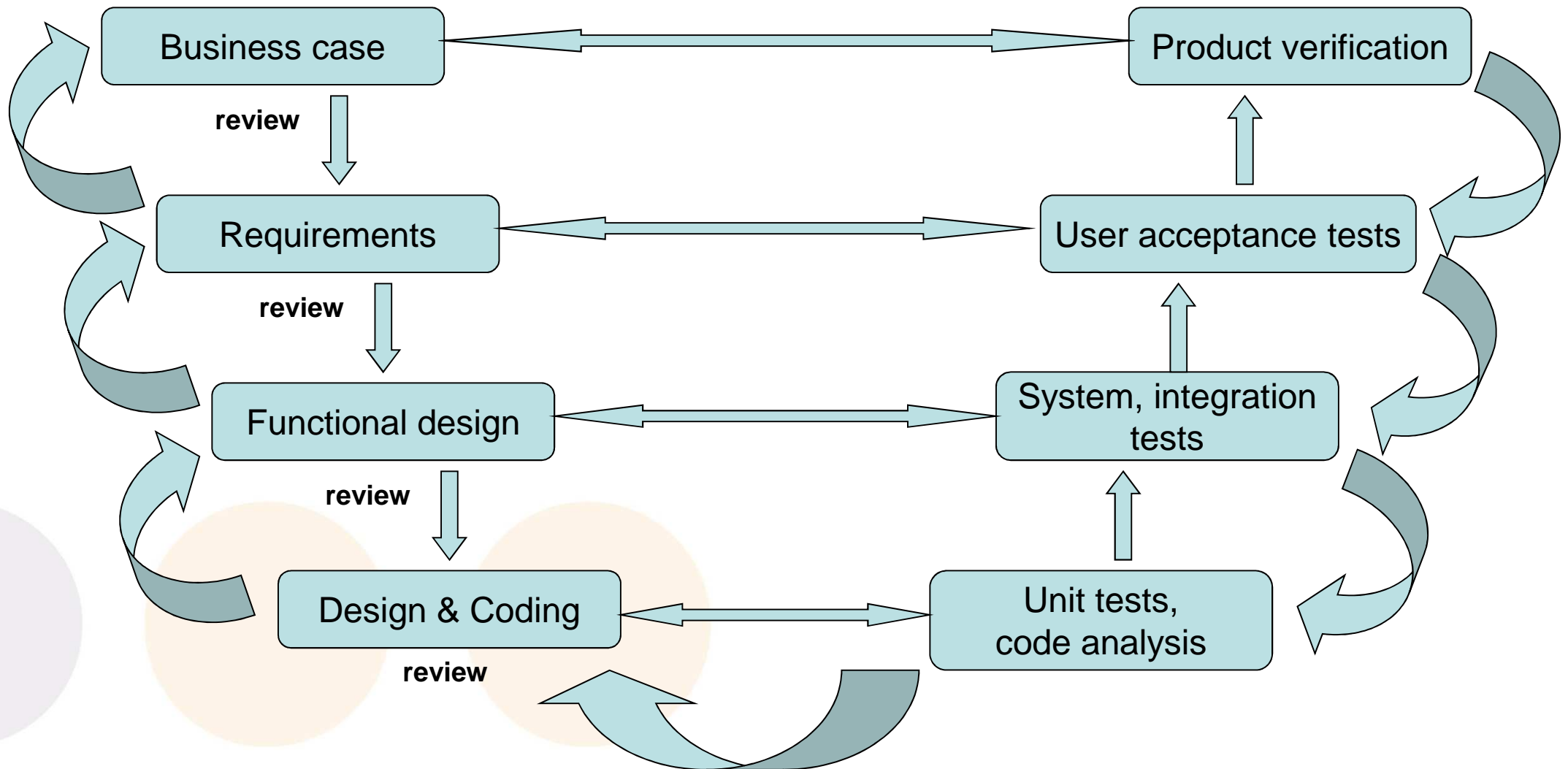


QA

- Obecně zajišťování kvality
 - V průmyslu – cíl „Six Sigma“
- Kvalita vs. cena
 - QA stojí peníze
 - Chyby stojí peníze
 - Pragmatismus
- V&V – Verifikace a Validace
 - Testování
 - Přezkoumání



V - model





Doporučení

- Klíčové dokumenty (nabídka, specifikace, design, plán testů, projektový plán, ...) někdo zkontroluje.
 - Někdo jiný než autor :-)
- Chyby se evidují (např. v Bugzille)
 - Nejen chyby v kódu
- Opravy chyb se před dodávkou přetestovávají
 - Někdo jiný než autor
- Dodržování domluvených postupů se kontroluje
 - Pragmatismus (vyhnout se byrokracií)



Dokumentace

- Druhy
 - Vývojářská
 - Projektová
 - Uživatelská
 - Administrátorská (instalační)
- Mýtus - Dobrý kód se dokumentuje sám
 - To platí nejvýš pro zápočťáky.
 - Nicméně špatnému kódu ani dobrá dokumentace nepomůže



Dokumentace - forma

- Není příliš důležitá
- Musí informace zobrazovat přehledně
- Dobré je, pokud podporuje efektivní vyhledávání
- Wiki
- Forrest
- Word (Writer,...)
- Textové soubory
- Anotace či speciální komentáře v kódu
 - Generování dokumentace

Project Profis 2 - Doxygen - manual - Windows Internet Explorer

http://212.24.144.117/wiki/Doxygen%20-%20manual.aspx

Live Search

Favorites

Detail přijaté faktury

Project Profis 2 - Doxyge... x

Page

Safety


Tools

Project Profis 2

Welcome Martin Hlavatý

This Site

Home



Project Profis 2 > Wiki > Doxygen - manual

Doxygen - manual

History | Incoming Links

View All Site Content

Documents

- Shared Documents
- Wiki

Lists

- Calendar
- Project Tasks
- Contacts
- Tasks
- Images

Discussions

People and Groups

Recent Changes

- Zápis ze schůzky - 2.11.2009
- zápisy ze schůzek
- Implementace_Hinty
- Home
- Orade DB

View All Pages

Doxygen manual

Misto SandCastle jsme se nakonec rozhodli používat Doxygen, protože umožňuje jednoduchou automatizaci generování dokumentace.

Stručný návod:

Doxygen je vlastně aplikace spuštěná z příkazové řádky a řídí se podle konfiguračního souboru. Celá konfigurace se provádí buď ručně, nebo přes velmi praktický frontend GUI.

Toto GUI je samovysvětlující.

Na záložce Wizzard

- podzáložka Project : * vybrat zdrojový adresář a také místo, kam hotovou dokumentaci exportovat. Je třeba zaškrtnout rekurzivní scan.
- podzáložka Mode * vybrat optimalizaci pro C#
- podzáložka Output * zvolit výstupní formát (pro LaTeX je potřeba nainstalovat, jinak vypadnou jen zdrojové soubory a nezkompilují se)

Na záložce Expert

- Project - nastavit jazyk
- Build - potřeba zvážit, co zaškrtnout...
- HTML - lze nastavit leccos, ale default look je použitelný
- LaTeX - nastavení cest

Na záložce Run



Domů

Analýza

Implementace

Testování

Management

Last Published: 11/09/2009 19:39:35

▼ Analýza

- Špecifikácia -
 - Podpora a vykazovanie práce
 - Špecifikácia - Obchodný cyklus
 - Školní specifikace
 - Business process model
 - Use case

Školská špecifikácia

[PDF](#)

- [Zadávateľ](#)
- [Motivácia](#)
- [Ciele projektu](#)
- [Náväznosť na predchádzajúci projekt](#)
- [Negatívne vymedzenie](#)
- [Užívatelia navrhovaného systému](#)
- [Architektúra](#)
- [Výstup projektu](#)
- [Vývojový tím](#)
- [Trvanie projektu](#)

Zadávateľ

Zadávateľom projektu je firma Profinit, s.r.o., ktorá je softwarovou spoločnosťou s desaťročnou históriou na stredoeurópskom trhu IT. Jej sídlo je v Prahe, ale pobočku má taktiež i na Slovensku v Bratislave. Portfólio firmy pokrýva hlavne vývoj softwaru na zakázku, dátové sklady a business intelligence, integrácia softwarových systémov a data mangement. Profinit poskytuje svoje IT riešenia prevažne v oblasti financií, telekomunikácií a dopravy.

Motivácia

Spoločnosti Profinit vďaka profesionálnemu rastu za posledné roky, s ktorým súvisí hlavne rozšírenie portfólia poskytovaných služieb, prestal dostačovať stávajúci informačný systém. Ten je nevyhovujúci predovšetkým z hľadiska slabej podpory reportov nutných pre fungovanie firmy, chýbajúcej podpory pre realizáciu obchodného cyklu a taktiež i pre nie príliš príjemnému užívateľskému rozhranie. Z tohto i z dôvodu, že žiadny z existujúcich systémov dostatočne nepokrýva špecifiká interných

9. Kliknout na OK

10. V emulátoru kliknout na ikonku Connectivity (dvě vodorovné šipky)



11. Kliknout na Settings

12. V záložce Connections kliknout na Connections

13. V záložce Advanced kliknout na Select Networks

14. Do obou políček zadat My Work Network





Cvičení

- Máte za úkol napsat specifikaci – jaký formát zvolíte a proč?
- Kdy by se mělo dokumentovat?
- Projekt trvá 6 měsíců a má rozsah 200 MD, kolik času by mělo být naplánováno na dokumentaci?
- Jak dokumentovat automatické testy?
- Je lepší mít 60 desetistránkových dokumentů nebo 6 stostránkových?



Uživatelská dokumentace

- Forma
 - Kontextová nápověda
 - Příručka
 - Jiné
- Ideální, pokud ji píše někdo z businessu ve spolupráci s vývojáři.
 - Jak se v programu udělá to, co uživatel potřebuje
 - Většinou utopie.



Administrátorská dok.

- Instalační
 - Potřebný SW a HW
 - Včetně verzí
 - Co musí být splněno
 - Hodnoty DB options
 - Postup instalace
 - V jakém pořadí nasazovat uloženy, ...
- Troubleshooting
 - Většinou se neřeší papírově, nýbrž telefonicky
- Techničtější než uživatelská
 - Pozor! Administrátoři nejsou vývojáři. Jsou pro ně důležité jiné věci než pro vás.



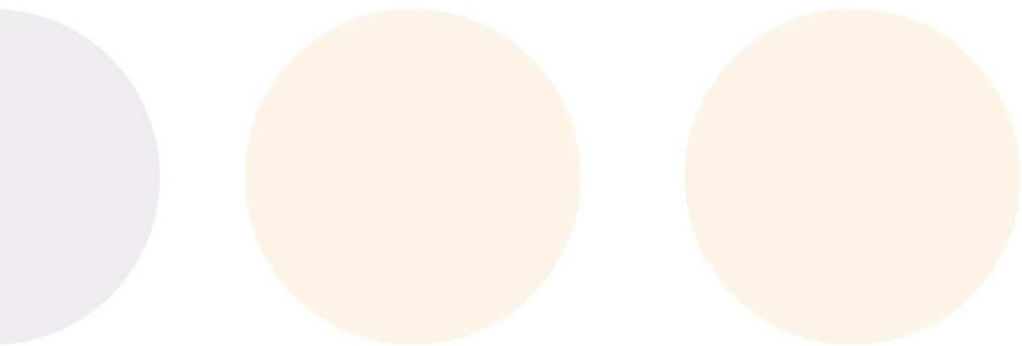
Vývojářská dokumentace

- Design
 - High level pohled na aplikaci.
- Kuchařky, návody
 - Jak se co udělá
- Dokumentace testů a postupů
- Slovníček pojmů (jak rozumět zákazníkům)
- JavaDoc
- Přítel na telefonu :-)



Cvičení

- Co by mělo být součástí JavaDoc třídy?
- Dokumentovat gettery a settery?
- Je třeba psát JavaDoc privátním atributům?
- Je třeba psát JavaDoc implementacím metod z
interfaců?





Cvičení – napište JavaDoc

```
public class BinTree<T> {  
    private BinTree left;  
    private BinTree right;  
    private T data;  
  
    public BinTree(BinTree left, BinTree r, T data) {  
        this.left=left;  
        this.right=r;  
        this.data=data;  
    }  
    // Gettery  
}
```

- Je to dobrý nápad, nedělat tam žádné settery?
- Jaké další problémy má tento kód?



Cvičení – napište JavaDoc

```
public int
```

```
VynasobZadanaCislaAVratVysledekPokudAniJednoZCiselNeniZap  
orneJinakVyhodVyjimku (int prvni, int druhe) throws  
Exception {
```

```
    if (prvni * druhe >= prvni + druhe - 1 && druhe > 0) {  
        log.fatal(„Nektere cislo je zaporne“);  
        throw new Exception(„Spatny vstup!!!!“);  
    } else {  
        return prvni * druhe;  
    }  
}
```

- Jaké problémy má tento kód?

Doporučení pro JavaDoc

- Popisujte, CO metoda dělá a ne jak. Popisujte, k čemu je daná třída a ne co obsahuje.
- Vždy dokumentujte metody interfaců
- První věta končí tečkou.
- Dodržujte konvence:
 - <http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>



Doporučení pro komentáře

- Popisujte své předpoklady
 - Pokud jsou špatné (situace se od napsání kódu mohla změnit), snáze se na to přijde z komentáře než z kódu.
- Popisujte netriviální obraty v kódu
- Popisujte, co se dělá ne jak.
 - NE: *„Když je částka větší než 1000 tak odečti pět procent částky“*. Lepší je třeba *„Aplikace pásmové slevy“*.
- Mě se osvědčilo – komentovat bloky kódu
 - Rychlejší orientace v kódu



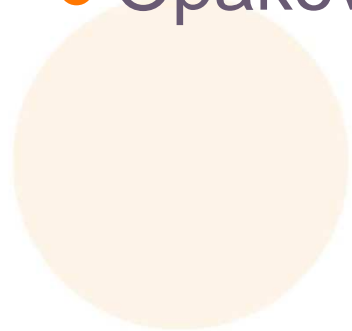
Projektová dokumentace

- Projektový plán
 - Forma: MS Project, Excel, linkovaný sešit ...
 - Není důležitá (pokud plní svoji funkci)
 - Obsah
 - Nikdy nebude obsahovat vše
 - Podrobný rozpad úkolů na nejbližší dny až týdny
 - Může mít PM v hlavě
 - Milestones a strategické úkoly
- Detailněji bude probrán v některém z následujících cvičení.



Projektová dokumentace (2)

- Historie projektu
 - Dělá se zpravidla na začátku fáze údržby.
 - Měření – práce, řádek kódu
 - Důležité pro budoucí odhady a pro vylepšování procesů.
- Plán testů
 - Opakování – co by měl obsahovat.





Dokumentace - údržba

- Svět se mění – dokumenty zastarávají.
 - Zastaralá dokumentace je občas horší než žádná.
 - => Na dokumentaci se nikdy nedá 100% spolehnout.
- Které dokumenty udržovat?
 - Všechny, u kterých se to vyplatí
 - Projektový plán, design, kuchařky, ...
 - Vyplatí se to, pokud dokument šetří čas (i se započítaným časem na údržbu).
- Největší žrout času – diagramy.
 - Text je snadné změnit, u obrázku už je to horší.



Automatizace dokumentace

- Generování dokumentů z metadat
- Princip DRY
- JavaDoc
 - Osvědčený
 - Doporučuji psaní komentářů neodkládat
- Generování UML ze zdrojových kódů
 - Vhodné, pokud se spojí s JavaDocem
 - Přínos není tak velký



Diskuse

- Komentáře
- Otázky
- Připomínky
- Upřesnění
- Poznámky
- ...

