

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

---

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

2011

Martin Kašný

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 5. května 2011

.....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla. Děkuji Fakultě FEI VŠB-TUO za možnost absolvovat odbornou praxi ve zvolené společnosti. Děkuji společnosti Tieto Czech s.r.o. za poskytnutí pracovních příležitostí na praxi právě u ní. Dále děkuji vedoucí bakalářské práce Mgr. Pavle Dráždilové za poskytnutí pomoci a orientaci při tvorbě bakalářské práce. A také děkuji mému tutorovi Bc. Pavlovi Marešovi za podporu a odbornou pomoc při vykonávání praxe.

## **Abstrakt**

Tato práce je zaměřena na odbornou praxi ve společnosti Tieto Czech s.r.o. v oddělení MOSS Testing na pozici tester (Test Engineer). Jedná se o testování MOSS webových aplikací pro zákazníky. Hlavní náplní praxe byla spolupráce na testování aspx webových aplikací, které mají společnou platformu SharePoint Server založenou na ASP.NET. Součástí mé bakalářské práce bylo seznámení s prostředím MOSS, testovacími postupy ve firmě Tieto a prostudování nástrojů a prostředků, které se k testování používají.

**Klíčová slova:** MOSS; SharePoint Server; webová aplikace; ASP.NET; Selenium; WatiN; Testování; Testovací scénář; Testovací případ; problémy a chyby; dokumentace; bakalářská práce

## **Abstract**

This thesis is focused to the Professional Practice in the Company Tieto Czech s.r.o. in Department MOSS Testing to position Test Engineer. It is testing MOSS web applications for Customers. The main scope of practical experience was cooperation on testing aspx web applications which they have together platform SharePoint Server based on ASP.NET. Part of my thesis was introduction with a working environment MOSS, Test Procedures in the Company Tieto and I studied tools and resources which used for testing.

**Keywords:** MOSS; SharePoint Server; Web application; ASP.NET; Selenium; WatiN; Testing; Test Scenarios; Test case; Bugs and Issues; Documentation; Thesis

## Seznam použitých zkratk a symbolů

MOSS	– Microsoft Office SharePoint Server
TC	– Test case (Testovací případ)
VS	– Visual Studio
ASP.NET	– je součástí .NET Frameworku pro tvorbu webových aplikací a služeb
WSS	– Windows SharePoint Services
SPS	– SharePoint Portal Server
XML	– Extensible Markup Language
XHTML	– eXtensible HyperText Markup Language
IT	– Informační technologie
IE	– Internet Explorer
SAP	– Informační podnikový systém
BI	– Business Intelligence - obchodní operace, systém pro podporu rozhodování
UIT	– Úvod do informačních technologií
PJ II	– Programovací jazyky 2
AJ	– Anglický jazyk
VIA	– Vývoj internetových aplikací
DAIS	– Databázové a informační systémy
ISEP	– Informační systémy elektronického podnikání
VŠB-TUO	– Vysoká škola Báňská - Technická univerzita Ostrava
MS	– Microsoft

## Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Popis odborného zaměření firmy a popis pracovního zařazení studenta</b>	<b>4</b>
2.1	O společnosti Tieto Czech s.r.o. . . . . .	4
2.2	Pracovní zařazení studenta . . . . .	5
2.3	Systém a jeho technologie . . . . .	7
<b>3</b>	<b>Zadané úkoly v průběhu odborné praxe</b>	<b>11</b>
3.1	Seznámení se službou MOSS a technologií SharePoint Server . . . . .	11
3.2	Testování webových aplikací . . . . .	11
3.3	Testování, kontrolní testy a tvorba prezentací . . . . .	11
3.4	Automatické testování . . . . .	11
<b>4</b>	<b>Postup řešení zadaných úkolů</b>	<b>12</b>
4.1	Seznámení se s MOSS a testování webové aplikace . . . . .	12
4.2	Reportování chyb a testování webových aplikací . . . . .	13
4.3	Testování a vytvoření prezentací o MOSS . . . . .	14
4.4	Seznámení se s automatickým testováním . . . . .	15
<b>5</b>	<b>Znalosti a dovednosti získané v průběhu studia a uplatněné v průběhu odborné praxe</b>	<b>16</b>
<b>6</b>	<b>Znalosti či dovednosti scházející studentovi v průběhu odborné praxe</b>	<b>17</b>
<b>7</b>	<b>Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení</b>	<b>18</b>
<b>8</b>	<b>Reference</b>	<b>19</b>
	<b>Přílohy</b>	<b>19</b>
<b>A</b>	<b>Přílohy</b>	<b>20</b>

---

## Seznam obrázků

1	Ukázka MOSS webové aplikace . . . . .	5
2	Architektura služby MOSS 2010 . . . . .	8
3	Funkce služby MOSS 2010 . . . . .	9
4	Prezentace - Seznámení s MOSS . . . . .	21
5	Prezentace - Testování v MOSS . . . . .	21

# 1 Úvod

Testování je dnes velmi důležitou součástí koloběhu vývoje aplikací, ať už po stránce funkčnosti nebo bezpečnosti. Cílem mé práce je popsat mou činnost ve společnosti Tieto na pozici tester.

V kapitole 2. uvádím základní informace o společnosti a jejím zaměření na testování. V sekci 2.2. uvádím mé pracovní zařazení ve společnosti s popisem prvních dvou dnů ve společnosti a v týmu. Obsahem sekce je také seznámení s testováním, konkrétně funkčním testováním, které je prováděno tzv. „z pohledu uživatele“ a testujeme tedy funkčnost a funkce systému, jak pro správné přístupy uživatelů, tak i ošetření nesprávných přístupů a výjimek s ohlášením nějaké chyby. V sekci 2.3. popisuji platformu, serverovou aplikaci a službu MOSS, se kterou jsem pracoval, a která sjednocuje všechny intranetové, extranetové a webové aplikace, v celém podniku, do jedné integrované platformy a nahrazuje tím využívání oddělených fragmentovaných systémů. Výsledkem je velice komplexní webová aplikace, kterou je třeba otestovat. A to bylo i jedním z mých úkolů, což rozepisuji v následující kapitole 3.

Pro funkční testování jsou zapotřebí také nástroje, aplikace a nezbytné prostředky, jako webový prohlížeč, místo pro psaní reportů a vytváření TC, což je opět webová aplikace fungující na platformě MOSS, nebo nástroje pro nahrání obsahu na web, aby jsme mohli rychle přejít k testování funkcí. Tyto nástroje a aplikace jsou popsány v sekci 2.3. a také v kapitole 4.

Hlavní úkoly byly, seznámení a nastudování informací o MOSS, uvedeno v sekci 3.1. a testování aplikací v sekci 3.2. Nebo vytvoření prezentací o MOSS pro vedoucí oddělení a nově příchozí zaměstnance. Úkol popsáný v sekci 3.4. , se zaměřuje na seznámení s automatickým ověřením funkčnosti aplikací a příprava frameworků. Následně řešení těchto úkolů v kapitole 4. a průběžné zlepšení schopností v této oblasti.

Před výsledným zhodnocením uvádím, jaké dřívější vědomosti jsem uplatnil a s čím jsem se seznámil. V kapitole 5. popisuji, jaké znalosti získané při studiu na VŠB-TUO mi pomohly na této praxi. V kapitole 6. uvádím, s jakými nedostatky jsem se musel potýkat. V kapitole 7. zhodnocuji průběh celé odborné praxe u společnosti Tieto.



## 2 Popis odborného zaměření firmy a popis pracovního zařazení studenta

### 2.1 O společnosti Tieto Czech s.r.o.

Tieto Czech je přední severoevropská společnost poskytující služby v oblasti IT a produktového inženýrství. Vysoce specializovaná IT řešení a služby doplněné silnou technologickou základnou, vytvářejí hmatatelné obchodní výhody pro jejich místní i celosvětové zákazníky. Hlavními trhy, na které se společnost zaměřuje, jsou země severní Evropy, Německo, Rusko a Polsko. Poskytuje také služby ve vybraných oblastech zákazníkům na celém světě s orientací na velké a středně velké organizace. Se svými zákazníky celosvětově spolupracují v oblasti digitálních technologií a služeb, telekomunikací, lesním, ropném a plynárenském průmyslu.

Se svými 17 000 odborníky je jedním z největších poskytovatelů IT služeb v severní Evropě a je přední světovou společností ve vybraných odvětvích, která se neustále rozrůstá, viz [3].

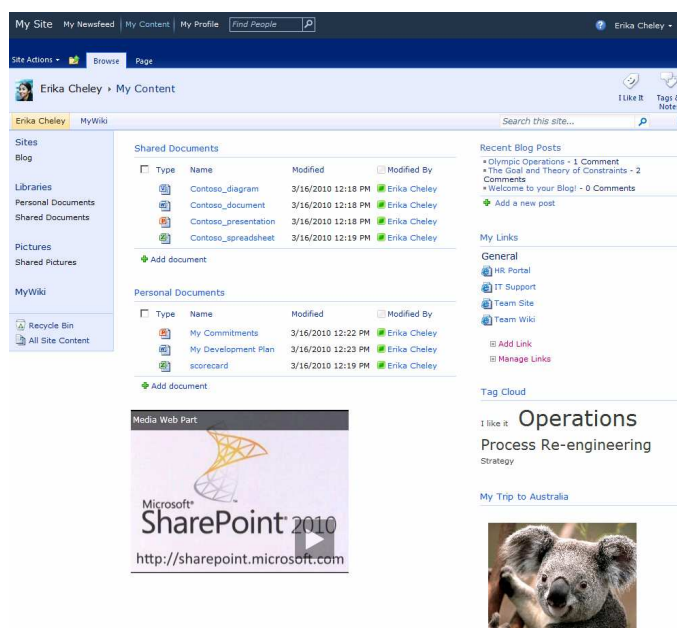
#### 2.1.1 Odvětví testování aplikací

Mezi poskytované IT služby patří také odvětví *Testování aplikací*. Na konci roku 2008 bylo v Ostravě založeno samostatné testovací oddělení, tzv. "Testing practice". V současné době sdružuje cca 50 zkušených testerů, kteří spolupracují na významných projektech pro zákazníky Tieto po celém světě. Testing practice v Ostravě je vedena Ivou Brandejsovou, do jejíž kompetence spadá i testování v Bělorusku a Litvě.

Testovací praktika nabízí následující služby:

- Funkční testování
- Automatizované testování
- Zátěžové testování
- Test konzulting
- Akceptační testování
- Analýzu testování včetně přípravy testovacích scénářů
- Konzultace a vedení testování
- Příprava testovacích metodik

Samozřejmostí jsou standardizované výstupy podle vnitřních metodik Tieto nebo metodik zákazníka. Jako mezinárodní společnost je Testing oddělení schopno výstupy dodat v českém i anglickém jazyce. Pro řízení testovacích projektů jsou použity jak standardní metodiky (Waterfall), tak i moderní agilní metodologie (SCRUM), viz [3].



Obrázek 1: Ukázka MOSS webové aplikace

## 2.2 Pracovní zařazení studenta

Ve společnosti Tieto jsem se ucházel o pozice zabývající se databázemi nebo testováním. A protože pozice na databáze byla již obsazena, ucházel jsem se o pracovní zařazení v oddělení testovacích praktik. Po absolvování pohovoru, s vedoucí tohoto oddělení Ivou Brandejsovou, jsem byl přijat na oddělení MOSS Testing na pozici tester.

První dva dny, kdy probíhalo školení nových zaměstnanců, jsem se dozvěděl základní informace o běhu společnosti, přístupu k zaměstnancům, povinnostech a výhodách pro zaměstnance, bezpečnosti jak fyzické, tak informací, politice firmy a o pravidlech, řádech a systému s prací ve společnosti spojeném. Poté jsem byl nadřízeným managerem přidělen k týmu, který se zabývá funkčním testováním webových aplikací postavených na technologii ASP.NET a běžící na serverové platformě Microsoft Office SharePoint Server, neboli MOSS. Samozřejmě se tyto aplikace ve společnosti Tieto i vyvíjejí, takže v některých projektech není problém s úpravami a komunikací s vývojovým oddělením. Ovšem zákazník může chtít i pouze otestování již své hotové aplikace, ale vždy jsou zapotřebí znalosti specifikace a funkčnosti těchto aplikací a zkušenosti s testováním.

Tým pracoval na několika projektech pro různé zákazníky, ale vždy se využívala stejná technologie i platforma. MOSS je vhodný pro podnikové weby jako je intranet, extranet, atd. a to pro střední nebo větší firmy, ale není problémem použít tuto platformu i pro malé firmy, i když se nevyužijí všechny možnosti tohoto systému. Ukázkový příklad, jak taková webová aplikace vypadá, můžete vidět na obrázku 1.

Funkci testera jsem vykonával v části týmu, který se zabývá automatickým a manuálním testováním MOSS aplikací. A má práce měla za cíl zvýšit kvalitu testovaných aplikací tím, že zjistím, popíšu a nahlásím případné designové nebo funkční chyby, zkontroluji změny v systému, pomohu s vytvářením a úpravou automatických testů, či testovacích scénářů, TC a frameworků.

### 2.2.1 Funkční testování

Testování aplikací můžeme definovat jako proces, při kterém se ověřuje, zda reálné vlastnosti aplikací odpovídají vlastnostem očekávaným a požadovaným. Smyslem je analyzovat chování aplikace, vzhledem k očekávaným vlastnostem tak, jak ho vidí uživatel. Tento typ testování se označuje jako testování černé skříňky (black box), při kterém se zaměřujeme na vstupy a výstupy programu bez znalosti, jak je naimplementován. Produkt je černou skříňkou, do které se nelze podívat. Vidíme jen jak vypadá a jak se chová navenek. Existuje také testování bílé skříňky (white box), kdy naopak testujeme produkt, aplikaci na základě jejich zdrojového kódu. Ovšem ztrácíme tím pohled uživatele, jak je uvedeno např. v [4] a [5].

Vlastnosti, které se v rámci testování ověřují, lze rozdělit do dvou skupin a to na vlastnosti Funkční a Nefunkční:

**Funkční vlastnosti** se týkají samotného účelu testované aplikace. Testování funkčních vlastností má ověřit, že aplikace správně vykonává úkoly, pro které byla vytvořena. To ale neznamená pouze to, že při korektním zacházení vykoná odpovídající operace. Testuje se také to, že aplikace je schopná vypořádat se i s nekorektním chováním ze strany uživatele. Jinak řečeno, že obsahuje správně implementované validace uživatelských vstupů a že případný nevalidní vstup vyvolá předem definovaný chybový stav.

**Nefunkční vlastnosti** aplikace můžeme označit všechny ty, které se týkají instalace, výkonu, dostupnosti a zabezpečení. Nejčastěji se testuje výkonnost. Tedy přesněji rychlost s jakou aplikace provádí své operace.

Další je zátěžové testování. To má ověřit, zda nárůstem objemu zpracovávaných dat není negativně ovlivněn právě výkon. V poslední době na důležitosti získávají bezpečnostní testy, jejichž úkolem je ověřit odolnost aplikace proti vnějším útokům a ochránit tím uložená data na serveru, blíže popsáno v [5].

Služba Funkční testování má ještě podkategorie, které se již zabývají konkrétním typem testování:

- Funkční testování
- Multi-Site testování
- Automatické testování

### **Automatické a manuální testování**

Podle toho, zda jsou testy prováděny člověkem nebo softwarem, se rozlišuje manuální a automatické testování. Pokud test vyžaduje lidské ohodnocení a úsudek nebo rozličné přístupy, které není třeba zaznamenat a pravidelně opakovat, je vhodnější manuální testování. Pro opakované spouštění velkého množství testů nebo testu s velkým množstvím generovaných dat, stejně jako pro zátěžové testování, je dobré použít automatické testy.

Při vývoji nových aplikací se ovšem neustále řeší nové případy, proto jsou tyto případy testovány především manuálně. Dříve nebyl vytvořen žádný testovací scénář, z kterého bychom mohli vytvořit testovací rámec (framework), proto také ještě nemohly být automatizovány. Automatizované testování využíváme zpravidla při opakovaném navštěvování stejného scénáře, např. při pokusech o verifikaci údajně opraveného defektu. Jde o situace, kdy je potřeba nově upravený systém rychle uvést do požadovaného stavu pro začátek manuálního testu.

Nebo v situaci, kdy pro automatizaci samotného testování, je potřeba splnit sadu předpokladů a tedy mít:

- testovací nástroj,
- vstupní data a výstupní data,
- testovaný program, který už automatizaci podporuje skrze svá rozhraní.

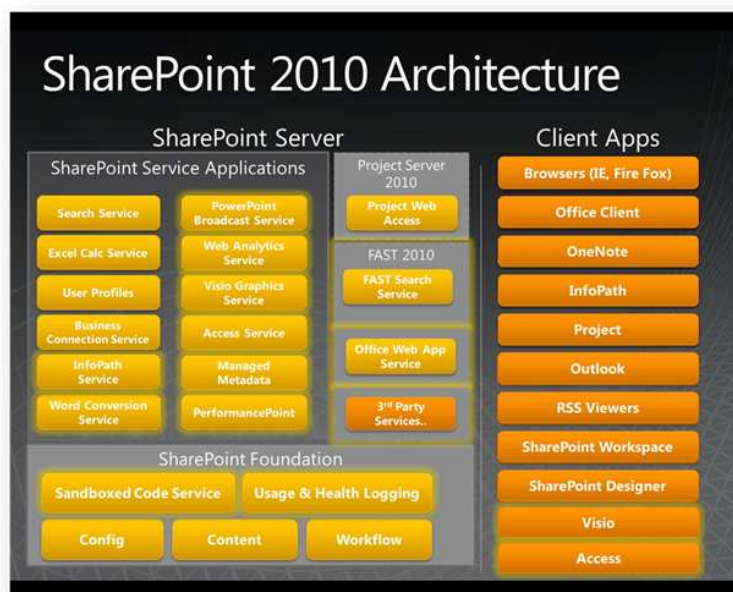
Kompletní automatický test se tedy skládá z jednotlivých testovaných případů (TC), vytvořených během vývoje nových funkcionalit (feature test). TC se dále seskupují do testovacích scénářů, ze kterých se pak může vytvořit framework, tedy celý automatický test. Testovací scénář může obsahovat dvacet nebo sto TC a je použitelný pro manuální i automatické testování, viz [4].

## **2.3 Systém a jeho technologie**

### **Microsoft Office SharePoint Server**

Jedná se o serverovou aplikaci, kterou mohou organizace použít ke správě podnikového obsahu, k vyhledávání, pro pracovní postupy a ke spolupráci. Tato aplikace je součástí systému Microsoft Office, nejlépe od verze 2007 a všechny funkce, které jsou k dispozici ve službě Microsoft Windows SharePoint Services 3.0 (WSS 3.0), jsou dostupné také na serveru SharePoint Server. MOSS však nabízí i další funkce a možnosti, které jsou určeny pro velké organizace vyžadující skutečnou platformu pro správu obsahu. Pokud se organizace potřebuje připojit k jiným podnikovým aplikacím, například SAP a vyhledávat v nich, implementovat složité obchodní procesy a pracovní postupy, jako například BI a zajistit zaměstnancům nebo uživatelům vlastní webové stránky nebo osobní weby, pak MOSS tyto funkce poskytne, podrobně popsáno v [1].

MOSS je novější verze SPS 2003 (SharePoint Portal Server) a rozšíření WSS 3.0 (Windows SharePoint Services). Rozšiřuje možnosti služby WSS tím, že nabízí funkce, jako například vyhledávání, robustní zásady pro ukládání a audit. Flexibilní nástroje pro správu organizace.



Obrázek 2: Architektura služby MOSS 2010

### Webová Služba WSS

Škálovatelná a zabezpečená platforma, která umožňuje organizacím spolupráci, ukládání a načítání informací, vytváření vlastních aplikací a správu obsahu. Služba Windows SharePoint Services je podpůrná technologie, která je součástí serveru Microsoft Windows Server 2003.

### Server SPS

Škálovatelný portálový server propojující osoby, týmy a znalosti v rámci podnikových procesů a umožňující tak efektivnější práci. Integruje informace z různých systémů do jednoho zabezpečeného řešení prostřednictvím funkcí jednotného přihlášení a integrace s podnikovými aplikacemi. Poskytuje flexibilní nástroje pro zavedení a správu a usnadňuje spolupráci koncových systémů prostřednictvím slučování, uspořádání a vyhledávání dat.

### MOSS

Spojuje funkce a výhody SPS a WSS do jednoho celku. Má zatím dvě edice a to z roku 2007 a nyní z poloviny roku 2010 s rozdělením do několika verzí, jako je Basic, Enterprise a další. Architektura MOSS je také velice komplexní a zajímavá, jak ukazuje obrázek 2. , kde je vidět začlenění služeb WSS a portálového serveru SPS spolu s podporou klientských aplikací a dalších microsoft aplikací a systémů jako je FAST Search - služba rychlého vyhledávání obsahu. Tyto informace jsou dále k dispozici v [1] a [2].



Obrázek 3: Funkce služby MOSS 2010

Cíle technologie SharePoint jsou především:

- usnadnit spolupráci mezi lidmi a pracovními týmy
- zajistit sdílení znalostí
- poskytnout nástroje pro správu dokumentů a webového obsahu
- umožnit uživatelům přístup k informacím, které potřebují pro svou práci
- umožnit další vývoj aplikací využívajících funkčnosti zabudované v technologii SharePoint (například obousměrná komunikace s podnikovými aplikacemi, nastavení automatizovaných procesů, vyhledávání informací nebo nástroje pro reportování)

Sjednocuje všechny intranetové, extranetové a webové aplikace v celém podniku do jedné integrované platformy a nahrazuje tím využívání oddělených fragmentovaných systémů. Vše je centralizované a pod jednotnou infrastrukturou.

Schopnosti a možnosti technologie SharePoint jsou pro přehlednost rozděleny do šesti oblastí tak, jak je ukázáno na obrázku 3. Mne, jako testera funkčního testování, nejvíce zajímají funkce:

- Content - správa obsahu,
- Sites - práce s weby,
- Search - vyhledávání různého obsahu na webu.

Všechny funkce jsou popsány v [2].

Se vstupem Visual Studia do procesu vývoje a s využitím technologie .NET Framework, s kterou je SharePoint úzce spjat, se otevírají téměř neomezené možnosti rozšíření o další funkcionality.

### 2.3.1 Nástroje a aplikace mnou využívané při testování

- MOSS 2007 nebo 2010 - viz sekce 2.3
- Visual Studio 2008 - 2010
- MS Office 2007 nebo 2010 (Outlook, Excel, Word, . . .)
- Webové prohlížeče (Internet Explorer, Mozilla Firefox, Google Chrome, Opera, Safari, . . .)
- Selenium - nástroj pro automatické testy, také utilita v prohlížeči Firefox
- WatiN - balík pro prostředí VS, který implementuje funkce pro možnosti automatického testování v IE

**Microsoft Visual Studio** - Vývojové prostředí od Microsoftu. Využívá se pro vývoj konzolových aplikací a aplikací s grafickým rozhraním. Je možné přidávat různá rozšíření, která vylepšují funkčnost. Umožňuje s využitím Windows.Forms vývoj webových stránek, webových aplikací a služeb.

**Microsoft Office** - Balík kancelářských aplikací od Microsoftu. Obsahuje aplikace pro tvorbu dokumentů, emailového klienta, tvorbu prezentací, databázový program a také serverové aplikace jako je SharePoint Server. Propojení těchto aplikací pro spolupráci využívá služba MOSS.

**Webový prohlížeč (browser)** - Počítačový program, který slouží k prohlížení World Wide Webu (WWW). Program umožňuje komunikaci s HTTP serverem a zpracování přijatého kódu (HTML, XHTML, XML apod.), který zpracuje a zobrazí jako webovou stránku. Pro zobrazení Flash animací nebo Java appletů, je třeba prohlížeč doplnit o specializované zásuvné moduly.

**Selenium** - Nástroj pro automatizované testování webových aplikací. Funguje tak, že spustí webový prohlížeč a v něm provede sadu příkazů testu, které jsme si předtím nahráli. To může být otevření stránky, kliknutí na odkaz, nějaké ověření textu, atd. viz v sekci 4.4.

**WatiN** - Nástroj pro automatizované testování webových aplikací. Primárně je určen pro automatizaci pomocí Visual Studia a využívá jazyk C# a frameworky, viz v sekci 4.4.

## **3 Zadané úkoly v průběhu odborné praxe**

### **3.1 Seznámení se službou MOSS a technologií SharePoint Server**

Seznámil jsem se s náplní mé práce na pracovní pozici Test Engineer, se zaměřením na funkční testování MOSS webových aplikací. Po nástupu na pozici tester a vyřešení technických záležitostí, abych mohl pracovat a podílet se na testování, jsem od svého tutora, Bc. Pavla Mareše, dostal podklady k této problematice. Jedná se o knihy, jako jsou Beginning SharePoint 2007, SharePoint Services and MOSS, Mistrovství ve Windows SharePoint Services 3.0, prezentace a internetové tutoriály. Tyto kurzy slouží k zaškolení a seznámení se s funkcemi služby a platformy MOSS, ve kterých je názorně předvedena správa a funkcionalita MOSS aplikací. Také jsem byl obeznámen se způsobem testování a dokumentací, která je k tomuto účelu zapotřebí. Mým úkolem bylo projít si podklady k SharePointu, dokumentaci k testování a začít testovat webovou aplikaci jedné finské společnosti.

### **3.2 Testování webových aplikací**

Mým hlavním úkolem bylo zdokonalit se v praktikách testování, pomáhat při hledání chyb a problémů, a reportovat nalezené chyby. Také projít testovací případy (TC), které byly vytvořeny již dříve a zkontrolovat, zda stále odpovídají svým postupem testované aplikaci. Dále jsem i vytvářel nové TC. V této části jsem testoval tři různé webové aplikace nejmenovaných firem, jejichž názvy neuvádím, z důvodu ochrany firemních dat.

### **3.3 Testování, kontrolní testy a tvorba prezentací**

V této části jsem kromě testování a kontrolního testování, za účelem ověřit starší funkce po přidání nových funkcí, dostal za úkol vytvořit prezentace o MOSS a testování v MOSS pro firemní účely. Prezentace měla mít zaměření i pro manažerské účely, a protože bylo třeba seznámit s poměrně novou službou, nově příchozí zaměstnance, vytvořil jsem tuto prezentaci „na míru“ dané problematice. Kromě těchto hlavních úkolů jsem dostával ještě i drobnější úkoly, které sice nevyžadovaly tolik úsilí a času, ale byly nutnou součástí odborné praxe.

### **3.4 Automatické testování**

V posledním oddílu popisuji seznámení s automatickým testováním pomocí nástrojů Selenium a WatiN. Během tohoto zkoušení a seznámení jsem i nadále testoval extranet Tieto.com a další weby, dle potřeb mých kolegů. Ve spolupráci s kolegy jsme zkoušeli automatické testy, jenž by ulehčily testování. Automatické testy jsou rychlejší, precizní, uspoří čas a eliminují potřebu nudného opakování stejných testů. Mohli jsme se tak zaměřit na důležitější oblasti testování.



## 4 Postup řešení zadaných úkolů

### 4.1 Seznámení se s MOSS a testování webové aplikace

Nejprve jsem začal se studiem starší verze MOSS 2007, ve které byla také testována první webová aplikace založená na MOSS, se kterou jsem se seznámil a o které jsem věděl, že využívá platformu SharePoint Server. V poměrně obsáhlých knihách, které jsou zmíněny v sekci 3.1. , jsem zjistil na čem je MOSS vlastně postaven, z jakých částí se skládá a k čemu slouží. Ale pro dobré a rychlé pochopení postupů a odzkoušení funkcností, se jako nejlepší volba jevíly internetové kurzy (tutoriály). Tutoriály se vyskytují ve verzi pro MOSS 2007 i pro MOSS 2010, jak mezi tutoriály, tak i v samotných aplikacích jsou rozdíly vizuální, v uspořádání a množství dostupných funkcionalit.

MOSS 2010 má lepší přístup k jednotlivým funkcionalitám, nejdůležitější ovšem je rozšíření o nové funkce. Také více přizpůsobuje své ovládání, aby vypadalo a fungovalo jako v aplikacích MS Office 2007-2010. Což je hlavně využití Ribbon, neboli kontextový pás karet služby SharePoint, který obsahuje některé stejné funkce jako v MS Office, uživatel pak nemá takový problém přejít na používání aplikací MOSS, protože mu prostředí této služby přijde známé a přehledné. Samozřejmě pokud přejde uživatel z verze MOSS 2007 na 2010 může si ponechat ovládání aplikace jako ve starší verzi, protože tato funkcionalita je elegantně řešena jako featura v nastavení aplikace, kde si můžeme nový vzhled a ovládání aktivovat nebo deaktivovat a na funkce nemá vliv.

Při procházení aplikace jsem nejprve hledal chyby a problémy stylového a vizuálního rázu, které jsem zapisoval a poté konzultoval s mým tutorem. Později jsem již postupoval podle dokumentace zákazníka, která se musí vytvořit pro účely vývoje a testování. Tato dokumentace specifikovala funkcionalitu oblastí testované aplikace a proto se jí také říká funkční specifikace, ve zkratce „funcspec“.

Další nezbytná aplikace je firemní extranet, který funguje také pod službou MOSS a kde se provádí reportování chyb, vytváření a úprava TC i celých testovacích scénářů. Přístup do testovaných projektů je podle týmů a projektů, ke kterým je tester nebo vývojář přidělen. Report je možné provádět i přes VS 2008, kde jsou k tomu patřičné funkce a stačí se připojit k serveru na síti. Všechna dokumentace i aplikace byly v anglickém jazyce, takže rozhodně nebylo lehké vše pochopit a využil jsem tedy i možnosti komunikace a spolupráce s kolegy, kteří mají již potřebné zkušenosti s testováním a orientací v systému. Chyby či problémy, které jsem nahlásil (reportoval), byly předány odpovědnému programátorovi nebo jiné odpovědné osobě a začalo se podle priority s jejich řešením.

## 4.2 Reportování chyb a testování webových aplikací

Od této části odborné praxe jsem prováděl většinou systémové a integrační testy, které nám zajistí, aby požadované funkce portálového webu a rozhraní pro externí přístup podporovaly obchodní procesy definované v dokumentaci daného projektu. Integrační testování zahrnuje testování integrace aspektů funkcionality, která byla naimplementována. Všechny funkce v rámci projektu nebo verze jsou testovány systémovými a integračními testy. Cílem je ověřit, že systém funguje tak, jak byl navržen (koncepte, scénáře, případy užití) a bez vad. Testování je založeno na TC a kontrolních listech chyb, které mají být testovány. Výsledky testů jsou zdokumentovány, reportovány. Pokud zjistíme vady při testování opravených chyb, musí být opraveny a znovu otestovány.

Jedním z testovaných webů byl i nový extranet Tieto.com, který jsem testoval v podstatě po celou dobu odborné praxe. A na konci praxe byl uveden do ostrého provozu na internetu. Report chyby by měl v základu obsahovat:

- Vhodně zvolený název problému, aby bylo jasné o jakou část dané testovací oblasti se jedná.
- Danou testovací oblast.
- Kdo má tento druh problému na starosti.
- Priorita závažnosti chyby - korekce stylů není tak důležitá jako problém s funkčností.
- Popis chyby a přiložené odkazy a screenshoty - velmi důležité pro správné pochopení podstaty chyby.
- V jakém prohlížeči se chyba vyskytla a pokud je jiná zareportovaná chyba podobná, tak uvést její odkaz.

Vytvořený TC by měl v základu obsahovat:

- Název testovacího případu, který bude jedinečný.
- Popis TC, kterým získáme přehled o testu.
- Očekávaný výsledek - popisuje, co se má stát na konci testu. Co bylo očekáváno.
- Testovací kroky - seznam kroků, které nás provedou testem.
- Stav TC (status) - jestli byl úspěšný, neúspěšný nebo vůbec není funkční.
- Testovací oblast - zdroj TC a požadavků nebo případ užití, manuál.

V případě nalezení neaktuálních a tedy již nesprávně pasujících TC, jsem tyto případy, buď upravil přeepsáním nevyhovujících částí nebo byly po shodě s vedoucím odstraněny. Dobře vysvětlit nalezené chyby a tím zajistit jejich opravu, je velmi důležitá část práce testera. Také netradiční přístup k testované aplikaci, trpělivost a schopnost objevit chybu, je cesta k úspěchu. Část se lze naučit, část lze odpozorovat a pochopit od zkušenějších, ale důležité je mít vlastní přístup a pohled na věc, tedy na testování. Výše zmíněné úkony mě provázely celou odbornou praxí.

### 4.3 Testování a vytvoření prezentací o MOSS

Průběžně jsem tedy testoval firemní extranet, z pohledu různých oprávnění uživatelů. Četl jsem funkční specifikaci projektu, testovací scénáře a jednotlivé TC, podle kterých jsem testoval, zda funkce splňují zadání. Aplikoval jsem také různé neobvyklé způsoby použití, abych ověřil, že jsou ošetřeny všechny uživatelské přístupy. Testování jsme prováděli iteračním způsobem, za použití agilní metodiky scrum. Kdy testování aplikace bylo rozděleno na kratší etapy tzv. Sprints, na které bylo vyhrazeno období, kdy se měla daná etapa otestovat a skončit úspěšným otestováním. V dalších fázích se tedy množství testování zvyšovalo a při každém Sprintu se kladl důraz na zaměřenou část aplikace s menším důrazem na starší části, jejichž funkce měly být již zajištěny. Ovšem ne vždy tomu tak bylo a proto je potřeba se v závěru testovací (vývojové) etapy ohlédnout a provést regresní testování, to znamená prověřit širokou škálu testů i u méně používaných funkcí.

Úkol, zadaný manažerem oddělení MOSS testing Ivou Brandejsovou, abych vytvořil prezentace o SharePointu, vypadal jednoduše, ale problémů bylo hned několik. Protože do tohoto úkolu bylo zainteresováno více lidí, zpočátku nebylo jasné, pro koho je prezentace určena, jak by měla vypadat nebo co obsahovat. Začal jsem tvořit prezentaci podle přibližných instrukcí a svých představ o obsahu. Vystihnout a přiblížit rozsáhlou službu a serverovou aplikaci MOSS, ve stručných bodech, nebylo jednoduché a většinu informací jsem čerpal z internetových článků. Druhá část prezentace, zaměřená na testování, byla jednodušší, jelikož jsem vycházel z toho, co jsem sám prošel a vyzkoušel. Ve spolupráci s kolegy se tedy vydařila dobrá práce, která prošla ještě kritikou drobných změn nebo přidavků od manažera. Druhá prezentace se zaměřovala více na obchodní problematiku. Musela zaujmout zákazníka. Z podkladů a instrukcí jsem vypracoval prezentaci, která obsahovala problém testování a řešení z obchodního pohledu, přístup k testování a možnosti pro Tieto, zkušenosti, co může Tieto nabídnout, co zákazník získá a reference s jinými zakázkami. Opět je potřeba doladit nějaké části pro daný účel, protože nejsem tak zběhlý v práci z pohledu manažera či obchodníka. Samozřejmostí bylo, že prezentace budou v anglickém jazyce. Prezentace jsou přiloženy k bc. práci na CD a jejich ukázkou si lze prohlédnout v příloze A.

Další úkol byl nahrát na web, nejmenované finské společnosti, obsah a strukturu stránek, protože při upgradování webové aplikace se vždy smazala struktura a obsah, který je potřeba pro testování a stále dokola vytvářet strukturu webu manuálně, je časově náročné. Struktura webu je popsána v dokumentaci projektu. MOSS je kvalitní produkt, který obsahuje featuru (funkci) Dummy Content Creator. Tato featura má vstupní formát soubory XML a umí vytvořit obsah webu získaný z XML souboru. Tedy jsem dostal za úkol dopsat v XML dokumentu obsah pro testovaný web. Rodičovské elementy byly hlavní stránky a potomci byli podstránky s popisem.

Také jsem se začal seznamovat s automatickým testováním a jedním z úkolů bylo sepsat TC, které by byly univerzální a tím pádem použitelné pro všechny web. aplikace postavené na MOSS platformě (foundation). Tento testing framework jsem psal jako dokument v programu MS Office Excel a na konci činil asi čtyřicet použitelných TC pro potřeby opakovatelného testování, ať už manuálního nebo automatického.

## 4.4 Seznámení se s automatickým testováním

Nástroje, které jsem použil, byly Selenium IDE a WatiN. Zjednodušeně řečeno, jde o technologii, která sama „kliká“ na webové stránce, otevřené ve webovém prohlížeči. A protože jako všechny monotonné úkoly i klikání provádí počítač rychleji a přesněji než člověk. Je to velká úspora práce, která se musí udělat.

Nejprve jsem se seznámil s nástrojem Selenium IDE. Je implementován jako rozšíření Firefoxu, pomocí kterého lze nahrávat, editovat a také spouštět testy. Takto nahrané testy jsou uloženy ve formátu XHTML jako soubory na disk. Ale je možné je převést i do kódování nějakého z programovacích jazyků jako Java, C# nebo skriptovací jazyk PHP. Metodou nahrávání kroků TC, udělaných při testování webové aplikace v prohlížeči, jsem získal několik testů, které všechny bylo možné spustit v listu a provést znovu. Problém byl s některými dynamickými částmi aplikace, které měnily své označení nebo adresu. Což mělo za následek nefunkčnost části testu. Také rychlost provádění testu je vcelku pomalá, i když rychlejší než ruční testování.

Další nástroj, se kterým jsem se seznámil, byl WatiN. Jedná se o klon známějšího projektu WATIR, který vyšel z Ruby komunity. Pro nahrávání testů se instaluje jako samostatná utilita, kterou spustíme. Starší verze podporuje pouze prohlížeč IE a nahrané testy jsou ve formátu kódování C#. A právě pro IE je WatiN stabilní a rychlý. Pokud není potřeba testování ve více prohlížečích, výhoda Selenia, a pro desetitisíce uživatelských přístupů, tak jde o dobrý nástroj. Úpravy a rozšíření testů na složitější a celková práce probíhá ve VS, kde přidáme balíček WatiN, aktuálně WatiN 3.5, obsahující jádro s dalšími třídami a metodami připravenými pro testování funkcí v IE.

Vyzkoušel jsem si jednoduché testy, které procházely některé s funkcí MOSS, s cílem dalšího využití a implementování pro opakovatelné frameworky, zmíněné výše v sekci 4.3.

Ukázka jednoduchého kódu ve VS pomocí WatiN (nutno použít WatiN.Core a MbUnit.Framework reference):

```
public static void TestWeb()
{
    IE webIE = new IE();
    webIE.GoTo("http://www.tieto.cz");
    webIE.Link(Find.ByClass("down-nav over")).Click();
    webIE.TextField(Find.ByClass("q")).TypeText("tieto");
    Assert.AreEqual("tieto", webIE.TextField(Find.ByClass("q")).Text);
    webIE.Button(Find.Id("generalSearchButton")).Click();
    Assert.IsTrue(webIE.ContainsText("tieto"));
}
```

Popis: Vytvoření objektu, instance pro práci s Internet Explorerem. Otevření stránky [www.tieto.cz](http://www.tieto.cz) v novém okně webového prohlížeče Internet Explorer, kliknutí na odkaz "Co nabízíme", zadání textu "tieto" do pole vyhledávání, kontrola, zda je text správně, jak jsme chtěli. Kliknutí na tlačítko search a kontrola, zda opravdu bylo vyhledáno podle slova "tieto".

## 5 Znalosti a dovednosti získané v průběhu studia a uplatněné v průběhu odborné praxe

V prvním ročníku jsem absolvoval předmět UIT, kde jsme probírali a pracovali v HTML,  $\text{\LaTeX}$ u a MS Office. Získané dovednosti jsem využil při testování webových aplikací, k určení chyb způsobených právě špatným HTML kódem a úpravě některých testovaných případů nebo při analýzování prvků na stránce webu, abych mohl například správně použít elementy pro automatické testy, jako je button, link, table, textfield, div a další. Práce v  $\text{\LaTeX}$ u jsem využil při tvorbě této bakalářské práce, která je také psána v  $\text{\LaTeX}$ u. V aplikacích MS Office Word, Excel, PowerPoint, Outlook jsem pracoval po celou dobu odborné praxe.

V předmětu AJ, který byl vyučován v prvním a druhém ročníku studia, jsem absolvoval základy anglického jazyka, bez kterého bych se neobešel na škole ani v odborné praxi. Také jsem si prohloubil znalosti anglického jazyka a lépe pochopil jeho strukturu.

V druhém ročníku jsem v předmětu PJ II absolvoval základy programovacího jazyka C#, kdy jsem u automatického testování tento jazyk používal pro úpravu a spouštění testů.

Přínosem pro praxi v Tietu bylo pro mne také studium třetího ročníku VŠB-TUO v předmětech DAIS, VIA, ISEP, kde se probíralo vytváření databáze, tvorba internetových aplikací v ASP.NET a obchodní stránka informačních systémů. Kdy jsem v praxi využil znalost, jak pracují aktivní, dynamické webové aplikace aspx, protože jsem pracoval také s aspx aplikacemi. Povědomí o databázích bylo také užitečné, protože hlavní činnost na stránkách je práce s různými uživateli a dokumenty uloženými na serveru. Předmět ISEP se také pohyboval okolo HTML a prezentování vytvořené práce, což mi pomohlo při tvorbě prezentací o MOSS.

## 6 Znalosti či dovednosti scházející studentovi v průběhu odborné praxe

Odbornou praxi jsem absolvoval v týmu, který testoval MOSS webové aplikace. S touto službou a platformou jsem se před nástupem do firmy nesetkal. Takže vše o SharePointu, neboli MOSS bylo pro mne nové. Také teoretické a praktické znalosti testování byly pro mne téměř novou zkušeností. Už jsem měl možnost testovat jinou aplikaci u jiné firmy, ale jednalo se spíše o náhodné zjištění chyb. Testování nebylo mým úkolem. Z teoretických znalostí to jsou např. metodiky testování a z praktických, testovací postupy a report chyb.

Jelikož jsem anglický jazyk začal studovat až na vysoké škole, kde jsem prošel výukou pro začátečníky, neměl jsem požadované znalosti anglických technických výrazů, používaných kolegy při testování a v dokumentaci projektů.

S nástroji pro automatické testy jsem se dříve také nesetkal, ale výhodou byla praxe ze školy z předmětů orientovaných na programování. Jedná se hlavně o předmět PJ II, kde se probíraly základy programovacího jazyka C#, který je základem i v testovacích nástrojích používaných na praxi a také webových aplikací a platformy MOSS.

## 7 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Testováním jednotlivých webových aplikací jsem měl zjistit co nejvíce chyb nebo ověřit, že již je vše z hlediska funkčnosti v pořádku. Pomáhal jsem tak již dost vytíženým kolegům a hlavně mému tutorovi. Objevené chyby byly poté co nejdříve opraveny nebo zdůvodněny. Takže jsem pomáhal svou prací i zvyšovat kvalitu produktu, aby byl zákazník co nejvíce spokojen. Dále jsem svou činností ulehčoval a zrychloval práci testerů. Testerům jsem pomohl shromáždit univerzální TC, aby mohli podle nějaké struktury testovat tu funkčnost, která je pro všechny webové aplikace stejná a zlepšit tak práci s MOSS.

Odbornou praxi jsem si vybral proto, abych si mohl vyzkoušet práci ve velké mezinárodní společnosti, která má zaměření na oblast IT. Chtěl jsem zjistit, jakým přínosem pro mne bude praxe v této společnosti a zkušenosti při ní získané, spolu s využitím anglického jazyka. A abych se lépe orientoval na trhu IT s možností uplatnění. Nebo jestli pro mne bude praxe motivací k dalšímu studiu na VŠB. Samozřejmě nejdůležitějším rozhodnutím, pro mne bylo, že jsem chtěl poznat změnu oproti práci ve škole.

Odborná praxe ve společnosti Tieto byla zajímavá a dozvěděl jsem se, jak probíhá vývoj a testování rozsáhlé aplikace postavené na platformě MOSS a kde hledat a jak formulovat chyby. A protože se MOSS pomalu rozšiřuje mezi různé podniky, určitě nebude na škodu mít zkušenosti s touto službou a platformou. Na funkčním testování mne nejvíce zaujala jeho rozmanitost a možnost seznámit se s aplikacemi různých společností, které používají stejnou platformu MOSS. I když je základní MOSS funkcionality aplikací stejná, tak se testování, až na některé případy, rozhodně jednotvárností nevyznačovalo. Názor na testování aplikací MOSS a testování obecně, mám po zkušenostech rozhodně pozitivní, i když mnoho neznalých lidí testování považuje za zbytečné, není tomu tak. Protože zjistit a opravovat chyby až za provozu aplikace, je velmi nákladné a nebezpečné, jak z důvodu možných útoků, tak ztráty zákazníků a zájmu o aplikaci z příčiny její nefunkčnosti.

Studentům, kteří uvažují o odborné praxi, mohu společnost Tieto jen doporučit. Studenti zde mají šanci, po absolvování praxe, nastoupit na danou nebo jinou pozici ve společnosti, jako zaměstnanci. Kolektiv je plný odborníků z různých odvětví, s kterými se dobře spolupracovalo. Můj celkový dojem z práce v Tietu je velmi pozitivní, obzvláště přátelský přístup k zaměstnancům je velmi motivující. Přesto praxe úplně nenaplnila mé očekávání, jelikož práce jednotlivce je velmi separátní. Představoval jsem si, že lépe pochopím návaznost jednotlivých procesů ve společnosti. Na druhou stranu má tak člověk šanci stát se odborníkem na danou problematiku a postupovat v kariéře jako specialista a zvyšovat si tak své zkušenosti v oboru. Vedení také iniciuje své zkušené zaměstnance k vytváření školicích kurzů, zaměřených na jednotlivé profesní oblasti zaměstnanců. Pro testovací oddělení to jsou například automatické testování, základy práce s MOSS, testovací metodiky, práce s firemními aplikacemi a weby.

## 8 Reference

- [1] Errin O'Connor, *Mistrovství ve Windows SharePoint Services 3.0*, Brno: Computer Press, a.s., 2008, Vydání první, [citace 20.4.2011].
- [2] *Microsoft SharePoint* [on-line],  
URL: <<http://sharepoint.microsoft.com/cs-cz/Pages/default.aspx>>, [citace 20.4.2011].
- [3] *Tieto Czech s.r.o.* [on-line], URL: <<http://www.tieto.cz>>, [citace 20.4.2011].
- [4] Mgr. Anna Borovcová, *Testování softwaru* [on-line],  
URL: <<http://testovanisoftwaru.blogspot.com/2009/08/zaklady-testovani.html>>, [citace 20.4.2011].
- [5] David Herbst, *Testing software* [on-line], URL: <<http://www.swtestovani.cz>>, [citace 20.4.2011].



## A Přílohy

### Prezentace

- Seznámení s MOSS, obrázek 4 - IntroductionMOSS.ppsx
- Testování v MOSS, obrázek 5 - TestingMOSS.ppsx

Prezentace jsou přiloženy na CD ve složce Příloha.



## Microsoft® Office SharePoint® Server



### Agenda:

- Introduction to MOSS
- What is Microsoft Office SharePoint Server ?
- What is MOSS about ?
- MOSS functionality in brief
- MOSS 2010
- What is new in 2010 version?
- What is so special about MOSS
- SharePoint applications in real life
- Open Source Alternatives
- Testing MOSS applications
- Test approach
- Sample Moss Test case
- What can go wrong in SharePoint application?
- Experience with testing in MOSS

Obrázek 4: Prezentace - Seznámení s MOSS



## Microsoft® SharePoint®



### Agenda:

- Problem statement
- Test approach
- Functional testing
- Automated testing
- Performance testing
- Sample testing tools
- Sample Test Case
- What can go wrong in SharePoint application?
- Experience with testing in MOSS
- What Customer gets
- References in real life

Obrázek 5: Prezentace - Testování v MOSS