



Semestrální práce ke kurzu  
**4IT421 Zlepšování procesů budování IS**

# **User story driven documentation**

Autoři:

Tomáš Jirásek (jirt00)  
Martin Pešek (xpesm27)  
Lucie Růžičková (xruzl13)

Zimní semestr 2015/2016  
Datum odevzdání: 19. 12. 2015

## Abstrakt

Tato seminární práce se zabývá způsobem tvorby dokumentace pomocí uživatelských příběhů. V první části je čtenář seznámen s dokumentací obecně. Dále jsou popsány nevýhody tvorby dokumentace uživatelského rozhraní a její vliv na uživatele i vývojáře. Další část stručně pojednává o pojmu uživatelské příběhy. V poslední části je podrobně popsána tvorba dokumentace pomocí uživatelských příběhů a konkrétní příklad.

## Klíčová slova

Dokumentace, uživatelská dokumentace, uživatelské příběhy, uživatelské rozhraní

# Obsah

1. Úvod.....	1
1.1. Tvorba dokumentace pomocí uživatelských příběhů.....	1
2. Dokumentace a vztah k aplikaci a jejímu životnímu cyklu .....	2
2.1. Dokumentace obecně .....	2
2.2. Uživatelská dokumentace .....	2
3. Dokumentace UI, její nevýhody a jak poznat, že je vytvářena .....	3
3.1. Nevýhody dokumentace UI .....	3
3.2. Jak poznat nevhodnou dokumentaci .....	4
4. Vliv UI dokumentace na uživatele a vývojáře .....	4
4.1. Vliv dokumentace uživatelského rozhraní na uživatele .....	4
4.2. Vliv dokumentace uživatelského rozhraní na vývojáře .....	5
5. Co to je user story a k čemu slouží, uvedení příkladů.....	6
5.1. Uživatelské příběhy .....	6
5.2. Psaní uživatelských příběhů.....	7
5.3. Základní atributy uživatelských příběhů .....	7
5.4. Příklady uživatelských příběhů .....	8
6. Alternativa k dokumentaci uživatelského rozhraní - dokumentace řízená uživatelskými příběhy .....	9
6.1. K čemu slouží dokumentace řízená uživatelskými příběhy .....	11
6.2. Hlavní srovnání dokumentace řízené uživatelskými příběhy a dokumentace uživatelského rozhraní .....	11
6.3. Výhody dokumentace řízené uživatelskými příběhy.....	12
6.4. Příklad dokumentace řízené uživatelskými příběhy.....	12
7. Proces tvorby dokumentace v OutSystems .....	14
7.1. Současný proces tvorby dokumentace.....	14
8. Závěr .....	16

# 1. Úvod

Cílem této semestrální práce je popsat přístup k tvorbě dokumentace pomocí uživatelských příběhů (User Story Driven Documentation), která se zabývá zpracováním uživatelské dokumentace softwarového produktu. Dále tento přístup srovnat se současnou praxí, tedy vytvářením dokumentace softwarového produktu dle uživatelského rozhraní (UI = user interface) a vyzdvihnout primární výhody. Na závěr uvést příklady a popsat procesy při tvorbě dokumentace pomocí uživatelských příběhů ve společnosti OutSystems.

## 1.1. Tvorba dokumentace pomocí uživatelských příběhů

Tento přístup k tvorbě dokumentace využitý ve společnosti OutSystems popsal ve svém článku Joao Fernandes (2015). Podrobně se v něm zabývá důvody, které společnost vedly ke změně přístupu k tvorbě dokumentace. Jedním z hlavních důvodů bylo zpřehlednění a zpřístupnění dokumentace reálným uživatelům vyvíjeného systému. Namísto zdlouhavých popisů funkcí jednotlivých komponent totiž přístup pomocí uživatelských příběhů dává přesný návod pro vykonání požadované akce krok za krokem. Uživatel tak nemusí přečíst několik stran dokumentace, aby například zjistil, jak používat funkci pro vyhledávání.

## 2. Dokumentace a vztah k aplikaci a jejímu životnímu cyklu

V této kapitole je podrobněji popsán pojem dokumentace a její vztah k životnímu cyklu aplikace. Vytvoření dokumentace je nezbytnou součástí při vývoji softwaru. V různých fázích životního cyklu aplikace na ní může být kladen různý důraz. Rozeznáváme několik typů dokumentace. Zde bude podrobněji rozebrána pouze uživatelská dokumentace.

### 2.1. Dokumentace obecně

Dokumentace poskytuje informace o aplikaci. V průběhu životního cyklu aplikace jsou vytvářeny různé druhy dokumentace pro různé uživatele. První dokumentace by měla vznikat ještě před zahájením vývoje. Taková dokumentace obsahuje záměr projektu. Další fází cyklu je návrh, zde by měla dokumentace obsahovat koncepty pro jednotlivá řešení a důvod jejich výběru. Při samotném vývoji vzniká dokumentace samotného kódu, testovací dokumentace a také již průběžně uživatelská dokumentace. Po dokončení vývoje je potřeba jednotlivé dokumentace, hlavně tedy tu uživatelskou, udržovat aktuální. Podrobněji se pojmu dokumentace věnuje například Martina Černá(2011). ve své bakalářské práci

### 2.2. Uživatelská dokumentace

Uživatelská dokumentace popisuje funkce programu a slouží uživateli jako příručka. Pro koncového uživatele je to nejdůležitější část dokumentace aplikace. Jsou u ní kladeny velké nároky na konzistenci, aktuálnost a přehlednost. Měla by posloužit uživateli, který se systémem teprve začíná pracovat, ale i pokročilým uživatelům či správcům systému.

Z toho důvodu by měla pokrývat následující oblasti:

- funkční popis, který uvádí systémové požadavky a stručně popisuje funkcionalitu softwarového systému,

- návod k instalaci systému, který poskytuje správcům systému detailní informace o instalaci softwarového systému v konkrétních podmínkách,
- úvodní informace, tedy příručka, která umožní začínajícímu uživateli zorientovat se v softwarovém systému a poskytne mu informace o jeho běžném použití a řešení běžných problémů,
- referenční příručka, která by měla popisovat všechny funkce softwarového systému a jejich použití a obsahovat seznam všech známých chybových hlášení a způsob řešení chyb,
- příručka pro správce systému, která popisuje zprávy generované během interakcí softwarového systému s jinými systémy a jak na ně reagovat. (Černá, 2011)

Kromě klasické papírové dokumentace se ve většině případů již setkáme s online dokumentací například ve formě nápovědy k aplikaci. Tato dokumentace pak obsahuje křížové odkazy a funkci vyhledávání a může tak uživateli velmi ušetřit čas při hledání řešení na konkrétní problém.

## 3. Dokumentace UI, její nevýhody a jak poznat, že je vytvářena

### 3.1. Nevýhody dokumentace UI

Dokumentace UI sice může mít své výhody, protože obsahuje veškerou funkcionalitu daného prvku, nicméně pro uživatele je tento druh dokumentace nevhodný. Není pro něj snadné spojit si funkcionalitu jednotlivých tlačítek tak, aby dosáhl svého cíle. Hledáním v dokumentaci stráví plno času a ne vždy je úspěšný. Rovněž pro vývojáře je takováto dokumentace časově náročná na údržbu. S tím jak se mění funkcionalita systému, dokumentace stále narůstá a je těžké ji udržovat konzistentní. Namísto několika kroků, které uživateli ukážou, jak s danou funkcionalitou zacházet, tak vzniká i několika stránkový popis jednoho tlačítka, který nakonec ani není využit, protože neobsahuje to, co uživatel potřebuje.

## 3.2. Jak poznat nevhodnou dokumentaci

Největším problémem jak rozpoznat, že je vytvářena dokumentace UI namísto dokumentování pomocí uživatelských příběhů je, že to sám uživatel nerozpozná. Pro tvůrce dokumentace je tak obtížné identifikovat chyby a jejich řešení.

Firma OutSystems sepsala několik svých zkušeností, které mohou napovědět:

- zákazníci si stěžují, že s poskytnutou dokumentací nejsou schopni propojit jednotlivé funkce UI, tak aby mohli požadovanou funkcionalitu využít
- mnoho stran dokumentace obsahuje to samé s jen nepatrnými rozdíly
- V případě monitorování přístupu na jednotlivé stránky dokumentace se objevují strany s malým počtem shlednutí
- Údržba dokumentace je velmi náročná

Dále se v OutSystems zmiňují, že všechny výše zmíněné problémy považovali za oddělené. Jak se později ukázalo stačilo pouze změnit přístup k tvorbě dokumentace.

## 4. Vliv UI dokumentace na uživatele a vývojáře

J., Fernandes (2015) popisuje vliv dokumentace uživatelského rozhraní na uživatele a tým vývojářů. Hlavním vlivem je, že uživatel ztrácí možnost porozumění samotnému produktu, jehož funkcionalita je popsána pomocí často složité dokumentace uživatelského rozhraní. Negativní vliv tohoto druhu dokumentace se projevuje i u vývojářů zejména v nárůstu času potřebného pro její údržbu.

### 4.1. Vliv dokumentace uživatelského rozhraní na uživatele

Uživatele využívající dokumentaci uživatelského rozhraní, lze rozdělit na začátečníky a pokročilé uživatele. Tyto dvě skupiny uživatelů mají do jisté míry rozdílný přístup k využívání tohoto druhu dokumentace vyplývající z jejich znalostí.

Ke zdokumentování malé části funkcionality aplikace pomocí dokumentace uživatelského rozhraní může být obvykle vytvořeno i několik stran. Takto vzniklá dokumentace se stává rozsáhlá a nepřehledná. Přestože si začátečník vše přečte,

nemusí pro něho být vše plně vysvětlující, protože mu schází celkový kontext mezi dílčími úkony potřebnými pro vykonání jednoduché operace pomocí aplikace.

Na druhé straně pokročilí uživatelé mohou vědět, jak daná část aplikace funguje (mají kontextovou znalost). Přesto mají-li využít rozsáhlou dokumentaci uživatelského rozhraní k využití jednoduché funkcionality aplikace, mohou se v ní i pokročilí uživatelé ztrácet. Důsledkem toho tito uživatelé ztrácí zájem o dokumentaci a snaží se daný problém vyřešit experimentování přímo v aplikaci.

## 4.2. Vliv dokumentace uživatelského rozhraní na vývojáře

Jelikož vývojáři se podílejí na vytváření dokumentace pro uživatele, snaží se, aby byla dokonalá (dokumentující všechny možnosti aplikace). Z tohoto důvodu vznikají rozsáhle dokumentace uživatelského rozhraní. Pro vývojáře je obtížné stanovit priority pro jednotlivé části funkcionality, které mají být zdokumentovány.

Vliv dokumentace uživatelského rozhraní na vývojáře má zejména tyto dopady:

### **Na práci zejména v důležitých etapách či milnících.**

Před důležitým milníkem, kdy má být poskytnuta již kompletní dokumentace, je často kladen tlak na dokončení dosud nekompletní dokumentace. V této fázi je potřeba dobře stanovit, jaké části aplikace by měly být ještě zdokumentovány a které se již z časového omezení nepodaří do dokumentace zahrnout. Do dokumentace by tak měly být doplněny ty části zdokumentované funkcionality aplikace, které jsou důležité pro budoucí uživatele. Protože pro vývojáře je obtížné stanovit, jaké části aplikace jsou pro uživatele nejdůležitější, může se tak stát, že některé části nebudou v dokumentaci zahrnuty.

### **Na údržbu a aktualizace dokumentace.**

V dokumentaci uživatelského rozhraní jsou jednotlivé informace poměrně silně provázány (tento druh dokumentace obsahuje mnoho referencí). Právě tato vlastnost dokumentace se stává problematická v době její údržby a aktualizace. Změní-li vývojáři například některou komponentu uživatelského rozhraní v aplikaci (název tlačítka, pozice v okně aplikace), je potřeba zaznamenat tuto změnu na všech místech v



dokumentaci. Čím je tedy dokumentace rozsáhlejší a obsahuje více referencí, tím více se stává údržba a aktualizace dokumentace pro vývojáře problematičtější.

Dokumentace uživatelského rozhraní má tedy negativní vliv na vývojáře, kteří se tak následně začnou více soustřeďovat na tvorbu a údržbu dokumentace. Právě tento problém řeší dokumentace řízená uživatelskými příběhy.

## 5.Co to je user story a k čemu slouží, uvedení příkladů

Cílem této kapitoly je popsat user stories (uživatelské příběhy). V této kapitole je popsáno, co se rozumí pod uživatelským příběhem, jaké mají základní vlastnosti a jakým způsobem by měly být vytvářeny. Na závěr jsou uvedeny příklady uživatelských příběhů, na kterých jsou popsány některé z chyb vznikajících při jejich zapisování.

S uživatelskými příběhy se lze setkat zejména v agilním vývoji softwaru. Uživatelské příběhy tvoří významnou část v metodikách agilního vývoje, jako je například XP (Extreme programming), Scrum, aj. Pomocí uživatelských příběhů jsou obvykle definovány business požadavky na systém. V XP definuje zákazník své požadavky skrze uživatelské příběhy a s určením jejich priority stanovuje práci pro vývojáře.

### 5.1. Uživatelské příběhy

M., Cohn (2004) definuje uživatelský příběh jako popis funkcionality, která bude přínosná pro uživatele či odběratele kupující daný systém nebo software. Dále uvádí tři základní aspekty, na kterých jsou uživatelské příběhy založeny:

- Psaný popis příběhu použitelný pro plánování a jako připomenutí.
- Konverzace ohledně příběhu, jenž slouží k rozvinutí detailů tohoto příběhu.
- Testy, které sdělují a dokumentují detaily a mohou tak být použity k rozhodnutí, kdy je příběh hotový.

Podstatnou stránkou uživatelských příběhů je jejich zapisování. Obvykle je zapisuje zákazník, který popisuje funkcionality systému či softwaru z pohledu business požadavků. Zákazník tedy určuje skrze uživatelské příběhy množinu funkcionality,

kerou by měl daný produkt poskytovat. Dále je potřeba, aby zákazník stanovil priority jednotlivých uživatelských příběhů pro určení, které příběhy mají být realizovány nejdříve (v které iteraci agilního vývoje).

## 5.2. Psaní uživatelských příběhů

Pro psaní uživatelského příběhu M., Cohn (2008) doporučuje používat následující šablonu: “As a <type of user>, I want <some goal> so that <some reason>.” Psaní uživatelských příběhů pomocí této šablony má pozitivní vliv na jejich kvalitu.

První část šablony zajišťuje lepší porozumění požadavkům stanoveným skrze daný uživatelský příběh. Je-li příběh stanoven z pohledu konkrétního uživatele, je následně pro vývojáře, který má daný příběh implementovat, snazší porozumět danému požadavku. Pro uživatelské příběhy je tedy důležité vědět, z pohledu jakého uživatele jsou napsány a pro koho slouží, co je podstatou příběhu a za jakým účelem je to vyžadováno. Navíc psaní uživatelských příběhů podle této šablony zajistí jejich konzistentnost a následně snazší stanovení priority příběhů.

## 5.3. Základní atributy uživatelských příběhů

Pro psaní kvalitních uživatelských příběhů klade M., Cohn (2004) důraz na šest základních atributů. Tyto atributy jsou označovány jako INVEST podle jejich akronymů a udávají základní vlastnosti uživatelských příběhů, které lze neustále zlepšovat.

- Independent (nezávislé)
- Negotiable (vyjednatelné)
- Valuable to users or customers (přínosné pro uživatele nebo zákazníky)
- Estimatable (odhadnutelné)
- Small (malé)
- Testable (testovatelné)

Uživatelské příběhy by měly být navzájem nezávislé. Tato vlastnost je důležitá zejména pro stanovování priorit dílčích příběhů a k jejich plánování. Například při závislosti významného příběhu na příběhu s malým významem by docházelo k problémům, jak stanovit dobu potřebnou k jejich implementaci. Tato vlastnost je důležitá

pro tvorbu dokumentace pomocí uživatelských příběhů, kdy je cílem snižovat počet referencí, které mají negativní vliv na dokumentaci.

Druhá vlastnost je spojena se smluvní nezávazností příběhů. Je kladen důraz na možnost dojednat detaily příběhu během komunikace zákazníka a vývojářského týmu. S touto vlastností je spojen problém, kolik informací by měl daný příběh obsahovat. Může se tak stát, že zákazník do příběhu zapíše příliš mnoho business požadavků případně vývojář mnoho technologických detailů a příběh se tak stává nekvalitním.

Dále by uživatelské příběhy měly být psány tak, aby byly přínosné pro uživatele nebo zákazníky. Nemělo by se tedy stát, že by příběh byl psán z pohledu vývojářů s technologickými detaily. Takový příběh by byl nesrozumitelný pro uživatele nebo zákazníka. Tato vlastnost pomáhá při tvorbě dokumentace pomocí uživatelských příběhů, kdy se dokumentace stává přehlednější pro uživatele a nevzbuzuje tak dojem, že je vytvářena spíše pro samotné vývojáře místo pro uživatele či zákazníky.

Odhadnutelnost uživatelského příběhu je spojena zejména se schopností vývojářů odhadnout časovou náročnost příběhu na jeho implementaci. Ke špatné odhadnutelnosti náročnosti příběhu přispívá převážně složitě napsaný příběh, kdy vývojář nerozumí, co se má v rámci příběhu implementovat respektive nerozumí celkovému kontextu příběhu. Dalším vlivem mohou být technické neznalosti vývojářů, které jim brání k implementaci příběhu a dále nadměrná velikost příběhu.

Předposlední vlastností je velikost uživatelského příběhu, kdy je potřeba stanovit jeho optimální velikost. Příběh by tedy neměl být příliš veliký, aby ho nebylo zapotřebí dělit na dílčí příběhy a na druhé straně by neměl být příliš malý a nebylo nutno kombinovat dílčí příběhy v jeden souhrnný. Tato vlastnost tedy navazuje na nezávaznost příběhů a také do jisté míry ovlivňuje odhadnutelnost příběhů.

Nakonec uživatelské příběhy by měly být psány tak, aby je bylo možno otestovat a určit tak kompletnost implementace příběhu. Příběhy by tedy neměly být psány příliš obecně, ale tím způsobem, aby byly měřitelné.

## 5.4. Příklady uživatelských příběhů

Cílem této kapitoly je uvést pár příkladů uživatelských příběhů a zhodnotit jejich kvalitu respektive poukázat na některé chyby při psaní uživatelských příběhů. Výhodou

uživatelských příběhů je, že se dají psát s rozdílným množstvím detailů a také souhrnně pro větší část funkcionality systému. Na druhou stranu, aby byly uživatelské příběhy kvalitní, je potřeba stanovit optimální množství detailů a jejich optimální velikost.

Následující příklady ukazují jednak správné uživatelské příběhy, ale i ty, které obsahují některé z možných chyb.

**Uživatel může vyexportovat data do XML.**

Tento příběh je kvalitní. Jedinou nevýhodou je, že není napsán dle základní šablony.

**Uživatel se může rychle naučit pracovat se systémem.**

Tento uživatelský příběh je příliš obecný. Rychle se naučit pracovat se systémem není definováno.

**Systém bude používat Log4J k logování všech chybových hlášení do souboru.**

Tento uživatelský příběh je psán s technologickými detaily. Logovací mechanismus Log4J by neměl být uváděn.

**Uživatel může hromadně přidávat, editovat a mazat životopisy.**

Tento uživatelský příběh je příliš obsáhlý, měl by tedy být rozdělen do více dílčích příběhů.

**Jako kupující na internetových stránkách, chci mít možnost platit kreditní kartou, abych mohl okamžitě potvrdit můj nákup.**

Tento uživatelský příběh je kvalitní a navíc je napsán pomocí základní šablony.

Tyto příklady uživatelských příběhů nastínily některé z problémů spojených s psaním uživatelských příběhů. Tyto problémy jsou převážně spjaté s nedodržováním pravidel základních atributů uživatelských příběhů. Navíc by měla být dodržována základní šablona, kde by bylo možné výše uvedené příklady doplnit o jejich účel (například u prvního příkladu by mohlo být uvedeno, pro jaký účel uživatel potřebuje data v XML).

## 6. Alternativa k dokumentaci uživatelského rozhraní - dokumentace řízená uživatelskými příběhy

Tato kapitola se zabývá popisem vytváření dokumentace softwarových produktů pomocí přístupu zvaného "Dokumentace řízená uživatelskými příběhy". Dále srovnává tento přístup s přístupem dokumentace uživatelského rozhraní. Vyzdvihuje výhody

dokumentace řízené uživatelskými příběhy oproti dokumentaci uživatelského rozhraní a ukazuje příklad použití dokumentace řízené uživatelskými příběhy.

## 6.1. K čemu slouží dokumentace řízená uživatelskými příběhy

Dokumentace řízená uživatelskými příběhy vychází z uživatelských příběhů (user stories), které jsou využívány v agilních metodikách pro rozdělování úkolů na menší a zadávání práce. Více u uživatelských příběhů viz kapitola 5. Na tomto místě je také důležité zdůraznit, že tento přístup k dokumentaci se využívá k vytváření uživatelských příruček pro softwarové produkty, tedy tento typ dokumentace je určen především uživatelům, ne však samotným vývojářům softwarových produktů.

## 6.2. Hlavní srovnání dokumentace řízené uživatelskými příběhy a dokumentace uživatelského rozhraní

Dokumentace řízená uživatelskými příběhy stejně jako dokumentace uživatelského rozhraní se zabývá tvorbou uživatelských příruček. Rozdíly mezi těmito přístupy jsou v tvorbě dokumentace. Zde dle Fernandese(2015) jsou základní znaky těchto přístupů:

Dokumentace uživatelského rozhraní:

- Rozdělí softwarový produkt na mnoho na sobě nezávislých komponent.
- Popisuje vlastnosti komponent.
- Popis vlastností je velmi detailní.

Dokumentace řízená uživatelskými příběhy:

- Nejdříve se stanoví typické operace, které uživatelé budou vykonávat a napíší se na ně uživatelské příběhy. (Je-li software vyvíjen agilní metodikou s využitím uživatelských příběhů, nemusí se při tvorbě dokumentace stanovovat uživatelské příběhy, jelikož jsou již vytvořeny.)
- Dokumentace se vytvoří tak, že ke každému uživatelskému příběhu je uveden jasný, stručný a přehledný postup, který je doprovázen výseky obrazovek (screenshoty).
- Nejsou zde detailní popisy vlastností jednotlivých komponent, pouze jen ty základní.

## 6.3. Výhody dokumentace řízené uživatelskými příběhy

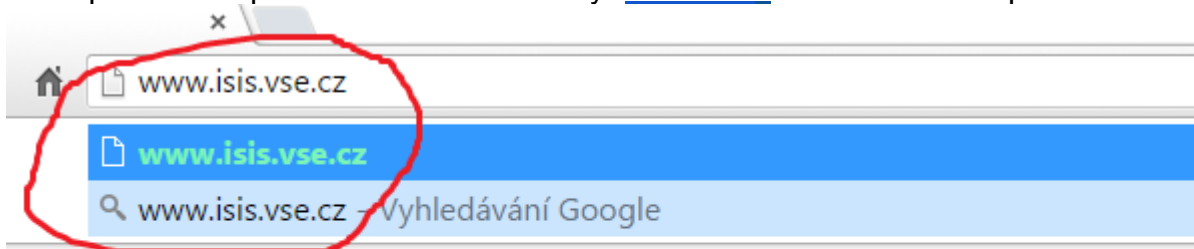
Dokumentace řízená uživatelskými příběhy dle Fernandese(2015) má oproti dokumentaci uživatelského rozhraní tyto výhody:

- Jednodušší údržba dokumentace při změně produktu.
- Dokumentace je stručná a přehledná z důvodu, že nepopisuje každý detail.
- Uživatelé si rychle osvojí práci se softwarem.

## 6.4. Příklad dokumentace řízené uživatelskými příběhy

Na tomto jednoduchém příkladu je vidět, jak by mohla vypadat dokumentace k systému ISIS pro uživatelský příběh “Já, jako student, se chci registrovat na obor vedlejší specializace.”

1. Vstoupíme do aplikace zadáním adresy [www.isis.vse.cz](http://www.isis.vse.cz) do webového prohlížeče.



2. Klikneme na odkaz “Přihlášení do osobní administrativy ISIS”.



### Osobní administrativa

- [Přihlášení do osobní administrativy ISIS](#)
- [Návod k prvnímu přihlášení do ISIS](#)
- [Systémová integ](#)

3. Zadáme přístupové údaje a přihlásíme se.

Přihlašovací jméno: abc00  
Heslo: .....

**Přihlásit se**

Odhlásit po 1 dni neaktivity ([změnit](#)).

4. Klikneme na odkaz "Portál studenta".

**Moje studium**

- Portál studenta**
- Statistiky studentských registrací
- Hodnocení úspěšnosti předmětů
- Evaluace předmětů studenty
- Elektronická přihláška ke studiu

5. Klikneme na tlačítko "Registrace/Zápisy" na VS".

Zvolit	Fakulta	Program - Obor	Stav studia	Přihlašování na zkoušky	Reg./Zápisy
	FIS	<b>N-AI Aplikovaná informatika Informační systémy a technologie</b> [prezenční]	řádně zapsán		

6. Přejdeme na stránku s výběrem předmětů kliknutím na odkaz "Registrace".

<a href="#">Katalog</a>	<a href="#">Registrace</a>	<a href="#">Prohlížení registrací</a>	<a href="#">Statistika registrací</a>	<a href="#">Výběr</a>
-------------------------	----------------------------	---------------------------------------	---------------------------------------	-----------------------

7. Zaškrtneme námi zvolený předmět.

<input type="checkbox"/>	3CA	Consulting	LS 2015/2016
<input type="checkbox"/>	1CO	Controlling	LS 2015/2016
<input type="checkbox"/>	2CS	Čínská studia	LS 2015/2016
<input checked="" type="checkbox"/>	1DB	Daně v podnikání	LS 2015/2016
<input type="checkbox"/>	5MV	Dějiny mezinárodních vztahů	LS 2015/2016
<input type="checkbox"/>	4DS	Demografie a sociální analýza	LS 2015/2016



8. Potvrdíme kliknutím na tlačítko "Registrovat".

<input type="checkbox"/>	1UC	Účetnictví a finanční řízení podniku	LS 2015/2016
<input type="checkbox"/>	1UP	Učitelství odborných ekonomických předmětů pro střední školy	LS 2015/2016
<input type="checkbox"/>	5VS	Veřejná správa	LS 2015/2016

**Registrovat**

9. Hotovo. Předmět byl registrován.

**Operace byla úspěšně provedena.**

V systému máte registrovány následující vedlejší specializace. Pokud zrušit, označte ji a klikněte na tlačítko Odebrat. Odebírat lze pouze sp a neabsolvovali jste některou část výběrového řízení. Priority u jednotliv

Ozn.	Priorita	Posun	Specializace	Období
<input type="checkbox"/>	1.	↓	4IQ, Inteligentní systémy	LS 2015/2016 - FIS
<input type="checkbox"/>	2.	↑	1DB, Daně v podnikání	LS 2015/2016 - FFU

**Odebrat**

Pokud bychom vytvářeli dokumentaci pomocí přístupu uživatelského rozhraní, popisovali bychom, co který prvek (odkaz, či tlačítko na stránce) dělá a jaké vlastnosti má (co se stane po kliknutí na něj). Taková dokumentace by byla nepřehledná a zjistit, jak se zaregistrovat do vedlejšího oboru, by znamenalo nastudovat celou dokumentaci.

## 7. Proces tvorby dokumentace v OutSystems

Tato kapitola se zabývá firmou OutSystems, která přišla první s nápadem na tvorbu dokumentace řízenou uživatelskými příběhy. Kapitola shrnuje současný proces tvorby dokumentace ve firmě OutSystems.

### 7.1. Současný proces tvorby dokumentace

Současný proces tvorby dokumentace v OutSystems se skládá z pěti základních kroků (Fernandes,2015):

#### **Start.**

Před začátkem nového projektu se sejde vývojový tým a tým, který má na starosti dokumentaci a diskutují spolu na téma projektu. Poté hledají vhodné uživatelské příběhy, které jsou následně konzultovány s managementem.

### **Zjištění, zda je pravý čas na vytváření dokumentace a školících materiálů.**

Jelikož vývojový tým používá mix Kanbanu a Scrumu, uvádí po každé iteraci funkční demo. Tým vytvářející dokumentaci demo verze posoudí, zda-li je dostatečně vyžralá a je čas začít tvořit dokumentaci. Toto načasování je velmi důležité. Pokud se začne tvořit dokumentace příliš brzy, je velmi pravděpodobné, že se dokumentace bude muset z větší části přepracovat z důvodu změny v softwaru. Pokud se začne později, dokumentace nebude včas hotová a uživatelé se nebudou mít podle čeho řídit.

### **Seznámení se s aplikací.**

Když je čas začít tvořit dokumentaci, spustí si tým (tvořící dokumentaci) demo verzi aplikace a snaží se ji pochopit a naučit ovládat, aby do uživatelských příběhů mohla předat co nejkvalitnější zkušenosti.

### **Vytvoření dokumentace.**

Následuje vytvoření dokumentace pro každý uživatelský příběh. Důraz se klade na to, aby se pokryl uživatelský příběh a nic víc. Tento krok je prováděn v několika kolech, kdy je po každém kole poskytnuta zpětná vazba na dokumentaci od vývojového týmu.

### **Vytvoření grafické části.**

Když už jsou všechny uživatelské příběhy zdokumentovány a odsouhlaseny, dojde pouze ke grafické úpravě dokumentace a tvorbě výseků obrazovek (screenshotů).

## 8. Závěr

Tvorba dokumentace řízené uživatelskými příběhy nás velmi oslovila svou jednoduchostí a přesto možností velmi kvalitně popsat softwarový produkt. Podle našeho názoru těmito vlastnostmi odsunula dokumentaci vytvářenou podle uživatelského rozhraní na “druhou kolej”, kde může sloužit jako referenční příručka, či nemusí existovat vůbec. Velmi silné spojení pak vznikne zkombinujeme-li tento způsob vytváření dokumentace s vývojem softwaru pomocí agilních metodik za využití uživatelských příběhů, kdy pak samotná tvorba dokumentace se stává jednoduchou záležitostí. I přesto, že dokumentace řízená uživatelskými příběhy je oficiálně používána ve firmě OutSystems, která tento způsob vytváření dokumentace vytvořila a prosazuje ho, očekáváme její nárůst použití i u ostatních firem, zabývajících se vývojem vlastních softwarových produktů.

## Zdroje:

-.The Easy Way to Writing Good User Stories [online]. 2008. Dostupné z: <http://codesqueeze.com/the-easy-way-to-writing-good-user-stories/>

COHN, Mike. User Stories Applied: For Agile Software Development. Boston: Addison-Wesley Professional, 2004. ISBN 978-0-321-20568-1.

COHN, Mike. Advantages of the “As a user, I want” user story template. Mountain Goat Software [online]. 2008. Dostupné z: <https://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template>

ČERNÁ, Martina. Dokumentace při vývoji softwaru. Bakalářská práce. Praha : Vysoká škola ekonomická, 2011.

FERNANDES, Joao. User Story Driven Docs. InfoQ [online]. 2015. Dostupné z: <http://www.infoq.com/articles/story-driven-docs>

KNESL, Jiří. User Story [online]. Dostupné z: <https://www.youtube.com/watch?v=wlbQXL0Nu8s&feature=youtu.be>

NAZZARO, William a Charles SUSCHECK. New to User Stories. Scrum Alliance [online]. 2010. Dostupné z: <https://www.scrumalliance.org/community/articles/2010/april/new-to-user-stories>