

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

---

# **Prispôsobiteľný Widget**

**(dokumentácia k 1. šprintu)**

**Tím č.10 : the 6\_p@ck**

**Bc. Michal Immer**

**Bc. Jakub Korch**

**Bc. Jozef Macho**

**Bc. Peter Petrilák**

**Bc. Igor Repka**

**Bc. Ján Sivulka**

---

Študijný program: Softvérové inžinierstvo/Informačné systémy

Ročník: 1.ročník inžinierskeho štúdia

Predmet: Tímový projekt

Vedúci projektu: Ing. Tomáš Kuzár

Ak. rok: 2010/11

## Metodika riešenia

Požiadavkou zadávateľa projektu (v našom prípade vedúceho projektu) bolo, aby sme vývoj softvéru realizovali metodikou Scrum. Scrum patrí medzi agilné metodiky vývoja projektov (nie len softvérových). Pre vývoj softvéru touto metodikou je potrebné mať identifikované roly jednotlivcov v tíme. Títo jednotlivci vytvárajú zoznam vlastností, požiadaviek a funkcionality produktu – tzv. „product backlog“. Tu môže byť zahrnuté všetko, o čom si členovia tímu myslia, že je dobré a potrebné pre daný produkt.

Zadávateľ projektu, tzv. „product owner“, do zoznamu vlastností pridáva vlastnosti a požiadavky hlavne na základe pohľadu záujemcov o tento produkt a jeho používateľov. Pomáha určovať smerovanie projektu. Ďalšou rolou v tíme je „scrum master“, ktorý sa stará o to, aby projekt napredoval a boli splnené časové požiadavky na jednotlivé fázy projektu. Organizuje stretnutia k projektu, manažuje projekt a ľudí pracujúcich na ňom. Nasledujúcimi rolami sú vývojári a testeria produktu, ktorí samotný produkt vytvárajú a testujú a nakoniec zákazníci a podporný manažment.

Zo zoznamu „product backlog“ sa následne vyberú tie položky, ktoré budú súčasťou samotného vyvíjaného produktu a zaradia sa do tzv. „release backlog“ zoznamu. Položky z „release backlog“ sa ohodnotia počtom hodín, teda odhadom, koľko približne potrvá jednotlivé položky zrealizovať.

Pomocou tejto metodiky je vývoj softvéru rozdelený do niekoľkých etáp, ktoré sa nazývajú šprinty. Trvanie šprintov môže byť v rámci vývoja jednotlivých projektov rozličné, v závislosti od typu a mohutnosti projektu (týždeň, 2 týždne až mesiac), v rámci jedného projektu však musia mať vždy rovnakú dĺžku. V metodike Scrum je okrem stretnutí kvôli šprintom potrebné realizovať aj krátke (asi 15 minútové) denné stretnutia, na ktorých sa predostrú zrealizované úlohy a objasní sa plán na daný deň pre každého člena. Každý šprint je tvorený niekoľkými položkami zo zoznamu „release backlog“, ktoré sú už presnejšie špecifikované. Nazývajú sa príbehy - „user stories“ a predstavujú opis nejakej funkcionality, ktorá bude prospešná pre používateľa, resp. vlastníka softvéru. Tieto príbehy sa v šprintoch rozbiť na ešte menšie úlohy, za ktoré budú zodpovední členovia vývojového tímu a na konci šprintu vznikne prototyp s implementovanými a otestovanými príbehmi a príslušnou dokumentáciou.

V našom prípade sme museli túto metodiku mierne pozmeniť a prispôbiť študijnému prostrediu, v ktorom pracujeme. Stretnutia k projektu realizujeme v týždňových intervaloch

a šprinty majú dĺžku 2 týždne. Denné stretnutia celého tímu nemáme, čiastočne ich však nahrádza komunikácia prostredníctvom e-mailu alebo stretnutia niektorých členov tímu pri riešení osobitných problémov. Počas týždňových stretnutí sa teda oznamuje aktuálny pokrok v riešení úloh, prípadne každý druhý týždeň sa robí plán na šprint nasledujúci(a vyhodnotí sa šprint predchádzajúci).

Na určovanie náročnosti úloh(časovej) používame odhad jednotlivých členov vývojového tímu, kde každý člen povie číslo(v hodinách) a po dohode sa diskutovanej úlohe priradí dohodnutá hodnota.

Počas každého stretnutia je určený jeden člen vývojového tímu, ktorý moderuje dané stretnutie a jeden člen, ktorý zapisuje diskutované témy a vytvára zápisnicu.

## **Podporné prostriedky**

Pri implementácii softvéru bolo možné vybrať si z jazykov PHP a Java, po diskusii v rámci tímu a po konzultácii so zadávateľom sme sa rozhodli pre použitie PHP. Táto voľba bola spravená vzhľadom na to, PHP má oproti Jave výhodu v rýchlejšom nasadení na web a väčšej rozšíriteľnosti v prípade úspechu produktu.

Na čítanie údajov používame Google Spreadsheet API a následné uloženie je už vykonávané v jazyku PHP. Vytvorenie samotného widgetu je realizované pomocou skriptovacieho jazyka JavaScript. Manažment vývoja softvéru riešime pomocou nástroja Agilo, ktorý podporuje agilný vývoj softvéru metódou Scrum. Tento nástroj je nadstavbou nástroja Trac. Na verzionovanie softvéru používame SVN server.

Tímovú komunikáciu, mimo osobných stretnutí, riešime pomocou vytvorenej skupiny na Google Groups, kde sme riešili potrebné záležitosti už od vytvorenia tímu. Vytvárame tu témy, nahrávame vytvorené súbory a upozorňujeme na nové záležitosti týkajúce sa tímového projektu. Zvyšnú komunikáciu medzi sebou riešime pomocou e-mailov alebo ICQ.

## Prvý šprint

Prvý šprint začínal 4. októbra 2010 a jeho koniec bol stanovený na 18. októbra 2010. Počas tohto šprintu bolo základnou úlohou vytvoriť jednoduchý prototyp widgetu, sfunkčniť tímovú stránku, podporné nástroje a dohodnúť sa na roliach Scrum master, zapisovateľ a moderátor.

### *„User story“*

Zadávatel' chce mať možnosť aplikovať jednoduchý filter na dáta, ktoré sa načítajú z dátového úložiska. Ďalej nech je možné vygenerovať kód, ktorý keď sa vloží do zdrojového kódu stránky, tak na jeho mieste sa zobrazia filtrované dáta z daného dátového úložiska.

Dátovým úložiskom bude pre začiatok použitý Google Spreadsheet.

## Analýza

V dnešnej dobe existuje veľa aplikácií a webových doplnkov, ktoré ponúkajú návštevníkom webových stránok obrovské množstvo informácií o rôznych podujatiach. Často je však prístup k týmto informáciám náročný a zdĺhavý. Problémom je i to, že informácie sú roztrúsené na veľkom počte stránok, čo značne komplikuje výber a aj samotné vyhľadanie žiadanej udalosti. Pre názornosť je možné uviesť napríklad skupinu ľudí, ktorá sa zaoberá nejakou spoločnou doménou na blogu a autor blogu chce poskytovať jeho čitateľom, ale aj sebe, informácie o aktuálnych podujatiach vrámci tejto domény.

Z tohto dôvodu by bolo vhodné mať taký webový doplnok, ktorý informácie o udalostiach získa z určitých zdrojov a spracuje ich tak, aby boli v prehľadnej forme zobraziteľné na stránke, a aby sa tento zoznam udalostí aj aktualizoval.

Pre tento typ úlohy je vhodný widget. Po vytvorení takéhoto widgetu by mal byť schopný pripojiť sa k zdroju údajov a zobrazíť žiadané informácie na stránke, kde je umiestnený.

Z hľadiska použitia technológií sme sa rozhodovali medzi programovacím jazykom Java a jazykom PHP. Nakoniec sme zvolili jazyk PHP z dôvodu, že takto vytvorený widget bude možné jednoduchšie nasadiť na stránky.

## Návrh

Základnou úlohou je získať údaje z úložiska a zobrazit' ich po prechode filtrom na stránke.

Aplikácia bude poskytovať používateľovi možnosť použitia jednoduchého filtra a to tak, že na stránke bude uvedený zoznam hodnôt jedného atribútu udalosti. Používateľ bude môcť vybrať jednu hodnotu a potvrdiť ju tlačidlom umiestneným vedľa tohto zoznamu. Po potvrdení hodnoty systém vygeneruje kód, ktorý po vložení na stránku používateľa a jej spustení spôsobí zobrazenie filtrovaných údajov na tejto stránke.

## Implementácia

Načítanie údajov z dátového úložiska, v našom prípade Google Spreadsheet sme realizovali pomocou jazyka PHP a Google Spreadsheet API. Pomocou Google Spreadsheet API sme získali údaje vo forme XML, ktoré sme v jazyku PHP následne rozparsovali, a tak sme získali informácie o udalostiach uložených na Google Spreadsheet.

Pre jednoduchosť tohoto prvého prototypu sme na stránke umiestnili len jeden comboBox a jedno tlačidlo. V comboBoxe sa zobrazujú kategórie, ktoré sú získané z údajov načítaných z XML(Príloha – obr. 1). Po výbere žiadanej kategórie a potvrdení výberu stlačením tlačidla sa na stránke zobrazí vygenerovaný kód v JavaScripte(Príloha – obr. 2). Tento kód po pridaní do zdrojového kódu stránky spôsobí zobrazenie filtrovaných dát na stránke.

Implementácia bola rozdelená do niekoľkých úloh(podľa rozdelenia pri stretnutiach, ktoré sú uvedené v zápisniciach). Za jednotlivé úlohy boli zodpovední poverení členovia tímu. Niektoré úlohy boli riešené jednotlivo a iné aj spoločne viacerými členmi.

## Testovanie

Vzhľadom na jednoduchosť prvého prototypu bolo zo začiatku testovanie vykonávané hlavne autormi jednotlivých podúloh. Pri spojení týchto podúloh do jedného celku bolo vykonané iba triviálne porovnanie či načítané kategórie zodpovedajú kategóriám v úložisku údajov a po zobrazení už samotných udalostí na stránke podľa vygenerovaného kódu, bolo vykonané porovnanie či sa zhodujú s udalosťami v úložisku údajov podľa filtrácie.

**Tabuľka č.1:** Akceptačný test pre získanie kategórií z Google Spreadsheet

<b>ID</b>	1	<b>Názov</b>	Získanie kategórií z Google Spreadsheet		
<b>Úroveň splnenia testu</b>		Musí – <del>Ma</del> by – <del>Mohol</del> by	<b>Autor</b>	Jozef Macho	
<b>Rozhranie</b>	Hlavná stránka				
<b>Účel</b>	Overenie správnosti funkcionality získavania kategórií.				
<b>Vstupné podmienky</b>	Existujúci dátový zdroj na Google Spreadsheet.				
<b>Výstupné podmienky</b>	Zobrazenie prvej získanej kategórie v ComboBoxe a po rozrolovaní komponentu všetkých kategórií.				
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná reakcia</b>		<b>Skutočná reakcia</b>	
1	Otvorenie stránky	Zobrazenie stránky s ComboBoxom, zobrazená kategória – získaná z prvého záznamu v Google Spreadsheet		Rovnaká ako očakávaná.	
2	Rozrolovanie ComboBoxu	Zobrazenie všetkých kategórií získaných zo záznamov v Google Spreadsheet		Rovnaká ako očakávaná. Ale Kategórie sú aj duplicitné.	

**Tabuľka č.2:** Akceptačný test pre zobrazenie vygenerovaného kódu v jazyku JavaScript

<b>ID</b>	2	<b>Názov</b>	Vygenerovanie kódu v JavaScripte		
<b>Úroveň splnenia testu</b>		Musí – <del>Ma</del> by – <del>Mohol</del> by	<b>Autor</b>	Jozef Macho	
<b>Rozhranie</b>	Hlavná stránka				
<b>Účel</b>	Overenie správnosti funkcionality generovania kódu v JavaScripte.				
<b>Vstupné podmienky</b>	Označená kategória v ComboBoxe				
<b>Výstupné podmienky</b>	Zobrazené okno na stránke s JavaScriptom				
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná reakcia</b>		<b>Skutočná reakcia</b>	
1	Stlačenie tlačidla odoslať	Zobrazenie vygenerovaného JavaScript kódu.		Rovnaká ako očakávaná.	

**Tabuľka č.3:** Akceptačný test pre zobrazenie widgetu z JavaScript kódu

<b>ID</b>	3	<b>Názov</b>	Zobrazenie widgetu z JavaScript kódu		
<b>Úroveň splnenia testu</b>	Musí – <del>Mal by</del> – <del>Mohol by</del>		<b>Autor</b>	Jozef Macho	
<b>Rozhranie</b>	Stránka používateľa widgetu				
<b>Účel</b>	Overenie správnosti vygenerovaného JavaScriptu pre zobrazenie widgetu na stránke				
<b>Vstupné podmienky</b>	Vygenerovaný JavaScript				
<b>Výstupné podmienky</b>	Zobrazené okno widgetu				
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná reakcia</b>		<b>Skutočná reakcia</b>	
1	Vloženie vygenerovaného kódu do zdrojového kódu stránky a jej spustenie	Zobrazenie vygenerovaného widgetu na stránke.		Rovnaká ako očakávaná.	

## Príloha

Obrázok č. 1: Zobrazenie kategórií z načítaných údajov

VITAJTE V NAŠOM INFORMAČNOM SYSTÉME "WIDGETIZÉRI"

Vyberte si kategóriu podujatia:

výskum	▼	Odoslať
výskum		
technológie		
podnikanie		
obchod		

Obrázok č.2: Zobrazenie vygenerovaného kódu po potvrdení výberu kategórie

VITAJTE V NAŠOM INFORMAČNOM SYSTÉME "WIDGETIZÉRI"

Skopírujte zobrazený text do Vašej stránky:

```
<script language="javascript" type="text/javascript"> var category = "vyskum";</script> <script language="javascript" type="text/javascript" src="testjs.js" > </script><div id="result"></div>
```