

Technická univerzita v Košiciach  
Fakulta elektrotechniky a informatiky, Katedra počítačov a informatiky

Agílné metódy programovania  
Extrémne programovanie

Lukáš Vavrek  
B3  
2015/2016

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Definícia</b>	<b>2</b>
<b>3</b>	<b>Základné princípy</b>	<b>2</b>
3.1	Jednoduchosť . . . . .	2
3.2	Komunikácia . . . . .	2
3.3	Spätná väzba . . . . .	2
3.4	Odvaha . . . . .	3
<b>4</b>	<b>Základné praktiky</b>	<b>3</b>
4.1	Business praktiky . . . . .	3
4.1.1	Plánovacia hra . . . . .	3
4.1.2	Zákazník na pracovisku . . . . .	3
4.1.3	Vydávanie malých verzií . . . . .	3
4.1.4	Metafora . . . . .	3
4.2	Tímové praktiky . . . . .	3
4.2.1	Párové programovanie . . . . .	3
4.2.2	Spoločné vlastníctvo kódu . . . . .	4
4.2.3	Štandardy kódu . . . . .	4
4.2.4	Udržateľné tempo . . . . .	4
4.3	Programovacie praktiky . . . . .	4
4.3.1	Priebežná integrácia . . . . .	4
4.3.2	Jednoduchý návrh . . . . .	4
4.3.3	Refraktorizácia kódu . . . . .	4
4.3.4	Testovanie . . . . .	5
<b>5</b>	<b>Záver</b>	<b>6</b>
<b>6</b>	<b>Použité zdroje</b>	<b>7</b>

# 1 Úvod

Agilné metódy programovania sa čoraz častejšie objavujú ako alternatíva ku klasickým metódam vývoja softvéru. Tieto metódy kladú dôraz na iteratívny prístup riešenia problému, na programovanie, na rýchle dodávanie fungujúceho softvéru a na rýchle zapracovanie zmien v dôsledku meniacich sa požiadaviek.

Jednou z najznámejších agilných metód je Extrémne programovanie, ktorému sa venuje táto práca.

## 2 Definícia

Extrémne programovanie (označované tiež skratkou **XP**) je metóda agilného programovania, ktorá bola vytvorená začiatkom 90. rokov. Táto metóda predpisuje určité činnosti všetkým účastníkom vývojového procesu. Jedná sa o tradičné činnosti, v “extrémnej” forme. Vďaka týmto vlastnostiam je extrémne programovanie schopné prispôbovať sa a dodávať vysoko kvalitný softvér. Táto metodológia je vhodná pre malé a stredne veľké tímy.

## 3 Základné princípy

### 3.1 Jednoduchosť

V XP sa vytvára kód, ktorý spĺňa tzv. minimálnu funkčnú implementáciu. To znamená, že sa tvorí iba nutná časť kódu potrebná pre splnenie požiadaviek a nič viac. Nikdy sa nevytvára kód ktorý je iba predpokladom pre využitie do budúcnosti, pretože XP predpokladá, že tieto požiadavky sa môžu meniť každým dňom.

### 3.2 Komunikácia

Kvalitná komunikácia medzi všetkými zainteresovanými subjektami je dôležitá. Komunikácia medzi programátorom a zákazníkom zabezpečí softvér ktorý bude zákazníkovi vyhovovať a ktorý bude navrhnutý na mieru. Programátor musí tiež komunikovať s manažérom, aby sa zabezpečilo predanie všetkých dôležitých informácií o stave projektu a zjednodušilo sa tak rozhodovanie. Komunikácia programátor - programátor je najdôležitejšou a venujú sa jej viaceré princípy XP.

### 3.3 Spätná väzba

Spätná väzba má v XP viaceré podoby. Pre programátorov sú ňou jednotkové a integračné testy vykonávané v dostatočne krátkych časových intervaloch. (V podstate sa jedná o vývoj riadený testami - test driven development). Tieto testy slúžia na overenie funkcionality kódu a ako overenie toho, že zmeny ktoré boli v kóde urobené nenarušujú funkcionality iných častí programu. Taktiež podporujú ďalšie praktiky, ako je napríklad refraktoring.

## 3.4 Odvaha

“Odvaha” programátorov sa v XP metodológii spomína na viacerých miestach. Je bežnou praktikou že oprava určitého komplexného problému spôsobí ďalšie súvisiace chyby. Pustiť sa do takýchto významných zmien vyžaduje odvahu a dôveru v dobre napísané jednotkové testy. Tak tiež je v XP “bežnou praktikou” zahodenie kódu u ktorého je problém s integráciou, pretože sa verí, že následné nájdenie optimálneho riešenia bude oveľa jednoduchšie.

# 4 Základné praktiky

## 4.1 Business praktiky

### 4.1.1 Plánovacia hra

Na procese spolupracujú zákazník a vývojový tím a spoločne plánujú vývojový proces jednotlivých iterácií. Jednotlivé zákaznícke požiadavky popisuje tzv. “užívateľský scenár”. Vývojový tím tieto požiadavky rozoberie a odhadne implementačné detaily (čas, cena) na základe čoho si zákazník určuje prioritu jednotlivých požiadaviek.

### 4.1.2 Zákazník na pracovisku

Na projekte pracuje aj osoba (alebo tím) reprezentujúci zákazníka, ktorý úzko spolupracuje na riešeníach, určovaní priorít a objasňovaní požiadaviek.

### 4.1.3 Vydávanie malých verzií

Nová verzia programu sa vydáva tak často, ako je to len možné. Jednotlivé funkcionality sa po naprogramovaní a otestovaní okamžite integrujú do produkčného kódu. Získava sa tak rýchla spätná väzba od zákazníka.

### 4.1.4 Metafora

Pri komunikovaní v tíme sa používajú jednoducho zapamätateľné metafory ktoré pomáhajú k jednoduchšiemu pochopeniu problému.

## 4.2 Tímové praktiky

### 4.2.1 Párové programovanie

Párové programovanie je charakteristická vlastnosť XP. V jednom momente pracujú na kóde dvaja programátori. Šofér, programátor ktorý píše kód, a navigátor, prísediaci programátor, spolu analyzujú riešené problémy. Hľadajú najlepšie riešenie a o optimalizácií a implementačných detailoch spolu komunikujú čím sa vytvára spätná väzba už pri vytváraní kódu. Programátori si tieto role striedajú.

## 4.2.2 Spoločné vlastníctvo kódu

V XP neexistuje rozdelenie zodpovednosti za jednotlivé časti kódu. Každý programátor môže meniť ktorúkoľvek časť kódu. Jediná potrebná vec je držať sa jednotkových testov.

## 4.2.3 Štandardy kódu

Tím programátorov sa dohodne na vopred určených pravidlách týkajúcich sa čistoty a kvality kódu. Všetky aspekty písania kódu ako napr. odsadzovanie, umiestnenie zátvorky, menné konvencie a pod. sú schválené každým členom tímu. Táto dohoda zabezpečí jednotnosť kódu tvoreného viacerými programátormi, čo v praxi znamená že programátori nebudú mať problém pracovať na kóde niekoho iného, pretože si nebudú musieť zvykať na jeho štýl programovania.

## 4.2.4 Udržateľné tempo

Extrémne programovanie sa opiera o tvrdenie, že pokiaľ sú programátori unavení, tvoria menej kvalitný kód a produkujú viacero chýb. Preto je potrebné už zo začiatku nastaviť také tempo postupu, ktoré bude možné udržať počas celého projektu. Približná týždenná pracovná doba programátora je 40 hodín a nadčasy sú nežiadaným javom.

# 4.3 Programovacie praktiky

## 4.3.1 Priebežná integrácia

Novo naprogramované funkcie sú do produkčného kódu integrované čo najskôr, minimálne raz denne. Pridanie každej funkcie nasleduje overenie funkčnosti a stability kódu pomocou jednotkových a integračných testov.

## 4.3.2 Jednoduchý návrh

Extrémne programovanie predpokladá zmenu požiadaviek zákazníka. Táto myšlienka tvrdí, že navrhovanie funkcie do budúcnosti je neefektívne. Programátori by sa pri vývoji mali sústrediť len na implementáciu funkcionality pre konkrétnu iteráciu.

## 4.3.3 Refraktorizácia kódu

Neustále vylepšovanie kvality kódu ktoré sa opiera o kvalitne navrhnuté a napísané testy. Tie umožňujú uskutočniť rozsiahlejšie zmeny bez obáv o stabilitu a narušenie funkcionality. Eliminujú sa duplicity, zvyšuje čitateľnosť, zlepšuje vnútorný návrh systému. Vďaka spoločnému vlastníctvu je každý programátor oprávnený opravovať ktorúkoľvek kód.

#### 4.3.4 Testovanie

Pri XP sa využíva tzv. programovanie riadené testami - test driven development. Znamená to, že ako prvé programátori píšú jednotkové testy a až potom sa zaoberajú implementáciou riešenia ktorá vyhovuje prechádza všetkými pripravenými testami.

## 5 Záver

Extrémne programovanie sa svojími vlastnosťami radí medzi agilne metódy programovania. Poskytuje kvalitné a overené spôsoby vývoja softvéru ktoré sú dotiahnuté do najmenších detailov a najväčšej efektivity. Táto metodológia ponúka zaujímavé pohľady na tvorbu softvéru a zefektívny prácu v tíme.

## 6 Použité zdroje

- Berta Igor - Agilné metódy vývoja softvéru. Škálovanie agilných metód. STU FIT Bratislava [online] [cit: 5.4.2016] Dostupné na:  
<http://www2.fiit.stuba.sk/bielik/courses/msi-slov/kniha/2006/groupd/essay/berta.pdf>
- Extrémní programování [online] [cit: 5.4.2016] Dostupné na:  
<https://cs.wikipedia.org/wiki/Extr>