

Jan Vraný

8.3.2006

eXtrémní Programování

Jan Vraný, vranyj@fel.cvut.cz

Co to je XP?

XP je agilní technika vývoje softwaru, která upřednostňuje:

- **teamovou spolupráci a interakci před formálními procesy a nástroji,**
- **fungující software před obsáhlou, komplexní dokumentací,**
- **spolupráci mezi zákazníkem a vývojáři před specifikacemi, zadáními, dohodnutými kontrakty,**
- **rychlou adaptaci na změny v zadání před dodržováním předem stanoveného plánu.**

12 technik používaných v XP

- **Planning game**
- **Small releases**
- **Metaphor**
- **Simple design**
- **Pair programming**
- **Testing**
- **Refactoring**
- **Collective code ownership**
- **Continuous integration**
- **40-hour week**
- **On-site customer**
- **Coding standards**

12 technik používaných v XP

Planning game

Software se vyvíjí v malých iteracích. Vždy se vyvíjí jen to, co zákazník v tu danou chvíli potřebuje.

Iterace se sestává ze:

- **sepsání user stories (zákazník)**
- **rozdělení user stories na úkoly (vývojáři)**
- **casových odhadů na jednotlivé úkoly (vývojáři)**
- **určení priorit (zákazník)**
- **vlastního programování (vývojáři)**
- **akceptace naprogramovaných user stories (zákazník)**

Small Releases

na konci každé iterace (naprogramování user story) se vytvoří nová verze (release). To znamená:

- **zákazník dostává nová verze poměrně často (jednou za 14 dní, jednou týdně, každý den),**
- **existuje okamžitá zpětná vazba, neboť zákazník může software neustále testovat a připomínkovat,**
- **zákazník má k dispozici funkční část SW velmi rychle a může ho začít používat.**

Metaphor

Metafora je jednoduchá paralela nebo příběh, který popisuje budovaný systém slovy a pojmy, kteří všichni zúčastnění (vývojáři i zákazník) intuitivně chápou.

Slouží především k lepší komunikaci a porozumění mezi členy teamu.

Simple Design

Co přesně má software dělat a jak má vypadat víte v nejlepším případě až když ho potřetí přepisujete celý znova. Scott W. Ambler.

Proto je vhodné udržovat návrh tak jednoduchý, jak jen to jde. Použít to nejjednodušší, co ještě splní akceptační testy.

Neprogramovat pro budoucnost, netvořit frameworky, pokud to není bezpodmínečně nutné.

Pair Programming

Každá řádka kódu je napsána dvěma programátory u jednoho počítače s jedním monitorem, s jednou klávesnicí.

To zajišťuje:

- **vyšší spolehlivost kódu, neboť programátoři se kontrolují navzájem,**
- **vyšší čitelnost a pochopitelnost kódu, neboť programátoři se kontrolují navzájem,**
- **všeobecné povědomí všech vývojářů o celém kódu**
- **a v neposlední řadě postupné vzájemné učení se vývojářů od sebe.**

Testing

Neustálé testování je jedna z fundamentálních technik XP, bez testování XP nemůže fungovat.

Vývojáři píší jednotkové testy na každou komponentu vyvíjeného systému, dokonce na každou třídu, tyto testy se spouští neustále.

Pokud neprochází všechny jednotkové a akceptační testy user stories, není možné pokračovat ve vývoji.

Refactoring

Refactoring je technika zlepšování stávajícího kódu beze změny dosavadního chování. Refaktoruje kvůli:

- **zjednodušení kódu**
- **optimalizaci kódu**
- **lepší čitelnosti, pochopitelnosti kódu,**
- **budoucí přidání nějaké další funkcionality**

Jistotu, že refaktorováním se nezmění stávající chování nám dávají jednotkové testy.

Collective Code Ownership

V XP teamu, kterýkoliv vývojář je oprávněn kdykoli modifikovat kterýkoliv kus kódu, pokud po modifikaci procházejí testy.

Continuous Integration

Veškerý kód je zpřístupněn všem vývojářům v centrálním repozitáři.

Kdykoli je úkol dokončen (naprogramován) a otestován, je uložen do centrálního repozitáře.

40-hour week

Psychicky nebo intelektuálně unavený vývojář je:

- **nevýkonný,**
- **chybující**
- **velmi naštvaný**

Proto je nutné vývojovému teamu zajistit klid na práci, důstojné pracovní prostředí a hlavně je nepřetěžovat.

On-site customer

On-site customer je člověk od zákazníka, který představuje typického uživatele vyvíjeného software a která je *neustále* k dispozici vývojářům aby:

- **zodpovídal otázky vývojářů během práce**
- **tvořil akceptační testy pro user stories**
- **testoval a podával zpětnou vazbu**

Coding Standards

Je nutné, aby *všichni* programátoři dodržovali stejné "coding standards", tedy:

- **stejně zarovnání kódu,**
- **stejně názvové konvence,**
- **stejně nastavené vývojové prostředí.**

Kdy má cenu použít XP?

- **Když je limitovaný čas nebo nebo financování.**
- **Když zákazník neví přesně co chce nebo není schopen dodat dostatečně přesné zadání.**
- **Když zákazník chce spolupracovat na projektu.**
- **Když se ví, že systém se bude měnit.**
- **Když členové teamu spolu dobře vycházejí a jsou schopni spolupracovat.**

Kdy nebude XP fungovat?

- **Je nutný početný (>15) team vývojářů?**
- **Jak dlouho trvá cyklus uprava kódu - překlad, linkování - spuštění? Více než 5 minut? Více než hodinu?**
- **Jak dlouho běží testy? Minuty? Hodiny?**
- **Je nemožné dostat zákazníka na pracoviště?**
- **Nesouhlasí zákazník s XP? Nechápe dobře jeho principy?**

Pokud alespoň na jednu otázku odpovíte ano, XP zřejmě nebude nejlepší volba...

eXtrémní Programování

- **Představuje alternativu ke klasickému vodopádovému modelu vývoje SW.**
- **XP není nekoordinované hackování softwaru zdivočelými programátory - hackery.**
- **Aby XP fungovalo, team musí mít jisté zkušenosti a musí bezpodmínečně dodržovat *všech* 12 technik.**
- **XP není všelék, hodí se jen někdy. Pokud se ale správně nasadí, mívá lepší výsledky za kratší čas.**