

MODELOVAT! – PROČ?

***„Kdyby architekti stavěli domy tak, jak
programátoři dělají programy, dokázal by
jeden jediný datel zbořit celá města.“***

(Joachim Graf, Murphyho zákony)

Z předmluvy knihy „Z kodéra vývojářem“

Je něco zvláštního na vývoji softwaru, něco mystického, co způsobuje, že si všichni lidé myslí, že ví, jak to dělat. Bez jakýkoliv zkušeností a znalosti jsou přesvědčeni, že ví, jak řídit softwarové týmy. To je podivné, protože si nikdo z nich nemyslí, že ví, jak odstranit slepé střevo nebo opravit auto, dokud nebude skutečně vědět, jak to udělat.

(Zkráceno)

Proč modelovat?

1. Vyhnout se chybným metodám řízení projektu
2. Plnit termíny
3. Optimalizovat vývojové náklady
4. Využít plně zdrojů, ale nepřetěžovat
5. Využívat předchozích výsledků vývoje
6. Minimalizovat údržbu

Co je to tunel?

- Chybí koncepce vývoje
 - žádné opakovatelné postupy
 - projekt je řízen intuicí a zkušenostmi
 - každý projekt má různorodé a nesourodé výsledky
- Nelze použít predikci, tj. nelze odhadnout
 - jak bude projekt pokračovat
 - jaká bude pracnost
- Neví se kdy má co kdo dělat
 - vše je v kompetenci vedoucího projektu
 - jako hlavní metoda řízení jsou operativní zásahy

Další vlastnosti tunelu

- Nízká znovupoužitelnost vyvinutého kódu
- Nízká transparentnost
- Velká složitost
- Redundance evidovaných informací
- Odmítání nových požadavků
- Synergická nestabilita (malé změny – velká pracnost)
- Příliš velký počet chyb v nasazeném programu
- Vývoj doprovází hektičnost prací, shon a nervozita
- Souhrn negativních vlastností se projeví ve zvýšených nákladech

Byrokratická metoda

Snaha odstranit nedostatky tunelu byrokratickým způsobem:

- vytvářením směrnic a postupů bez zkušeností
- příliš velká očekávání od jejich zavedení
- zavedení byrokracie zpomalí vývoj
- aplikací se většinou zvýší náklady a dojde ke ztrátám

Zhodnocení předchozích přístupů

- Obě dvě metody ukazují co nechceme
- Moderní metodiky především odstraňují největší nevýhody předchozích přístupů
- Cílem tohoto předmětu je se s těmito metodikami seznámit.

Odpověď na to „Proč modelovat?“

- Modelování používá dobré metodiky tvorby softwaru
- Předchozí tvrzení platí i obráceně – z toho se vlastně vyšlo
- Je si nutné stále uvědomovat proč se modeluje, jinak se sklouzne mezi neefektivní postupy

Příčný řez

- Příčný řez vzniká, když je práce rozdělena mezi pracovníky po jednotlivých oblastech tj. agendách.
- Analýzu, návrh a kódování si provádí každý sám.
- Nevznikají dokumenty na jednotlivých úrovních abstrakce.

Jak se nedostat do tunelu

Čtyři základní pilíře:

1. Úrovně abstrakce
2. Objektový přístup
3. Dobrý jazyk
4. Použití vzorů

Když jsou pilíře vzájemně propojeny, potom je možné se vyhnout tunelu i příčnému řezu.

Úrovně abstrakce

- Minimálně 3 úrovně
 - Analýza
 - Návrh
 - Kód
- Jen tak lze překlenout problém sémantické mezery tj. že zadavatel a realizátor nepoužívají stejný slovník

Objektový přístup

- Je to určitý náhled na systém
- Bude vysvětleno v přednášce „Základy objektové orientace“.

Dobrý jazyk

- Dobrým jazykem se jeví UML
 - je určen pro objektový přístup
 - je normalizován viz www.omg.org nebo www.uml.org
 - definiční dokumenty používejte jako referenční příručku

Použití vzorů

- Použití je primitivní:
 - Problémy neřešíme vždy znova od počátku, ale snažíme řešení nalézt v katalogu vzorů
- Princip:
 - Nevynalézáme kolo, protože už bylo vynalezeno.
- Existují různé kategorie vzorů a i tzv. antivzorů.
- Vývojáři (organizace) by si měli budovat vlastní katalog vzorů – což je jejich základní know-how.