

Sekvenční a iterativní přístup k vývoji software, výhody, nevýhody, důsledky, způsoby dodávky produktu.

Wednesday, May 29, 2013 4:50 PM

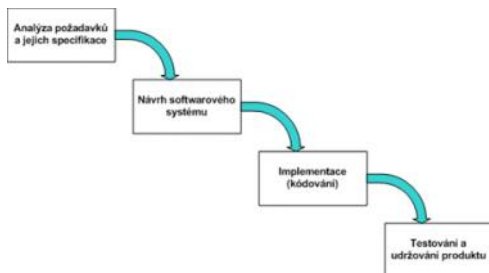
Sekvenční přístup

- vztahené na celý produkt - velký třesk
- naplánované pro celý projekt, oddělené meziprodukty
- vhodné pouze pro malé projekty s malou mírou neznáma (change management, náklady na změny, ...)
- předání až na konci celého projektu



Vodopádový model

- Životní cyklus projektu je rozdělen na 4 základní fáze:
 - Analýza požadavků a jejich specifikace
 - Návrh softwarového systému
 - Implementace (kódování)
 - Testování a udržování vytvořeného produktu
- Následující množina činností spjatá s danou fází nemůže započít dříve než skončí předchozí



Výhody:

- snadné k pochopení
- dobrá možnost řízení a sledování postupu řešení (milníky (milestones) – voleny po činnostech)
- klade důraz na dokumentaci - specifikace, design, analýza

Nevýhody:

- vyžaduje mít na počátku přesně a úplně definované požadavky (uživatel často nedokáže stanovit předem)
- provozuschopnost verze vidí zákazník až v závěrečných fázích řešení, případné závažné nedostatky jsou odhaleny velmi pozdě.
- během vývoje se mohou měnit požadavky a výsledkem je, že dodaný produkt není to, co zákazník chtěl
- během implementace se zjistí, že design není v pořádku a je třeba ho změnit

Iterativní přístup

- Vývoj rozdělen na malé části, miniaturní úplné projekty s cca vodopádovým modelem, tzv. Iterace. (Charakteristika a vlastnosti / průběh iterace viz. Otázka 1.1.4.)
- Inkrementální rozšiřování produktu z původní hrubé formy do výsledné podoby -> Umožňuje postupné upřesňování požadavků na cílový produkt
- Řízení rizik a priority uživatele na funkčnost produktu
- Zaměření na architekturu produktu
- Požadavky mají zásadní vliv na návrh a implementaci produktu
- Řízení a plánování iterativně vedeného softwarového projektu viz. Otázka 1.1.5.

Výhody:

- Menší časové úseky v dodávkách produktu -> zvýšení úspěšnosti produktu díky zpětné vazbě
- Snížení rizik
- Snazší řízení změn na základě zpětné vazby uživatele
- Vyšší míra znovuvyužitelnosti
- Projektový tým se může učit během procesu vývoje
- Vyšší kvalita produktu

Nevýhody:

- Průběžné změny mohou způsobit porušení původní systémové struktury což vede k náročnější údržbě softwaru
- Vyžaduje přísnější management

Způsoby dodávky produktu

Velký třesk

- malé projekty, jasné požadavky

Přirůstkově

- určení přirůstků -> plán -> postupné dodávky
- zpětná vazba, ale úpravy projektu obtížné

Evolučně

- cyklus: určení cíle -> dodávka -> zpřesnění ("growing sw")

Sekvenční a iterativní přístup k vývoji software, výhody, nevýhody, důsledky, způsoby dodávky produktu.

Sekvenční a iterativní přístup k vývoji software

Sekvenční

- Skládá se z několika po sobě jdoucích fází, typický příklad takového procesu je waterfall – inception, elaboration, construction, transition

Výhody:

- Jednoduché na pochopení a implementaci
- Vhodné pro velmi malé projekty s jednoznačným zadáním

Nevýhody:

- U větších projektů riziko, že výsledek nebude v souladu s tím, co si zákazník představoval
- **Nutné přesně definovat požadavky**

Iterativní:

- Iterativní přístup rozdělí celý proces vývoje software na menší celky, tzv. iterace, kdy v rámci dané iterace probíhá vývoj sekvenčně a na konci je třeba mít funkční výstup.

Výhody:

- Na konci každé iterace je funkční výstup
- Konzultace se zákazníkem, odchýlení se od požadavků se projeví včas a ne až na konci
- Zákazník vidí progress
- **Postupné zpřesňování požadavků**

Nevýhody:

- Složitější na implementaci než vodopád
- Náročnější údržba software díky změnám v požadavcích

Způsoby dodávky produktu

Velký třesk

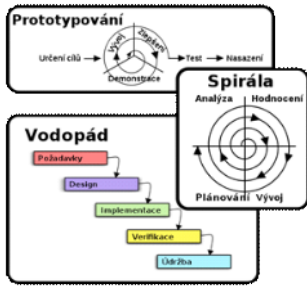
- Typicky u vodopádu, zákazník netuší až do konce jak produkt vypadá

Přirůstkově

- určení přirůstků, plán, úpravy složitě

Evolučně

- Postupně dodáváme software ve funkčním stavu a přidáváme funkcionalitu do nových verzí – určení cíle, dodávka, zpřesnění



From <<https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>> w