

DĚLAT
DOBRÝ SOFTWARE
NÁS BAVÍ

PROFINIT

Softwarový proces

Bohumír Zoubek, Tomáš Krátký

1

Úvod

Základní pojmy

- › Softwarový proces / Model životního cyklu vývoje software (SDLC, Software Development Lifecycle)
 - Množina aktivit nutných k tomu, aby software vznikl, jejich souslednost, opakování, vstupy a výstupy jednotlivých aktivit a nároky na jejich provedení
- › Vždy je potřeba udělat
 - Specifikaci – co bude systém dělat
 - Architekturu a design – z jakých „kostek“ a jak se bude systém skládat
 - Implementaci – vlastní výroba systému
 - Validaci – ověření, že systém dělá co má
 - Další rozvoj – úpravy systému na základě měnících se požadavků
- › Software process improvement
 - „Ladění“ a vylepšování procesu

Dle plánu nebo agilně?

- › Plánovaný softwarový proces (plan-driven)
 - Aktivita jsou plánovány dopředu
 - Pokrok je měřen porovnáním stavu a plánu
 - Větší režie v případě změn
- › Agilní softwarový proces
 - Plánování je jen po malých částech
 - Snadné změny „kurzu“ v případě změn požadavků zákazníků
- › Typický proces v Profinitu obsahuje
 - Plánované aktivity
 - Prvky agilního přístupu

2

Modely softwarového procesu

Waterfall

○ Oddělené fáze

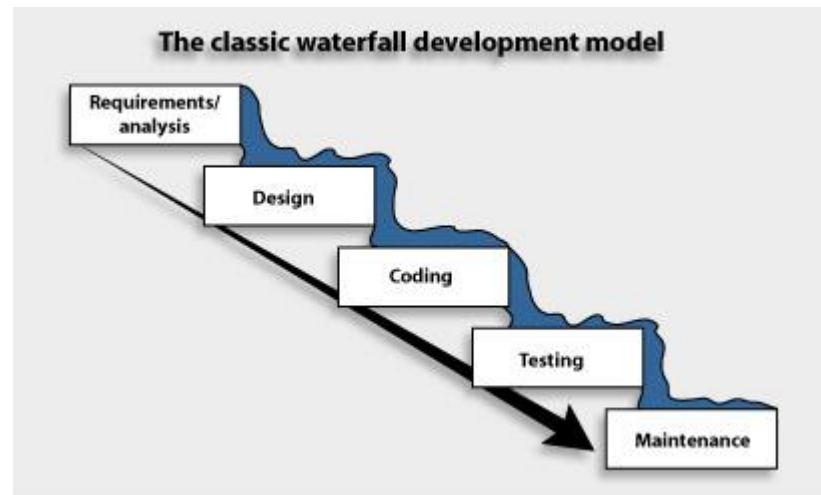
- Analýza požadavků
- Design
- Implementace
- Testování
- Provoz a údržba

○ Výhody

- Jasně definovaný plán
- Predikovatelnost (čas, rozsah, cena)
- Snadná koordinace práce

○ Nevýhody

- Nutno chápat co se chce již na začátku
- Reakce na změny (požadavků, termínů, ...)
- Rychlost dodávky (kdy zákazník něco uvidí od zadání)



Iterativní

○ Změny oproti vodopádu

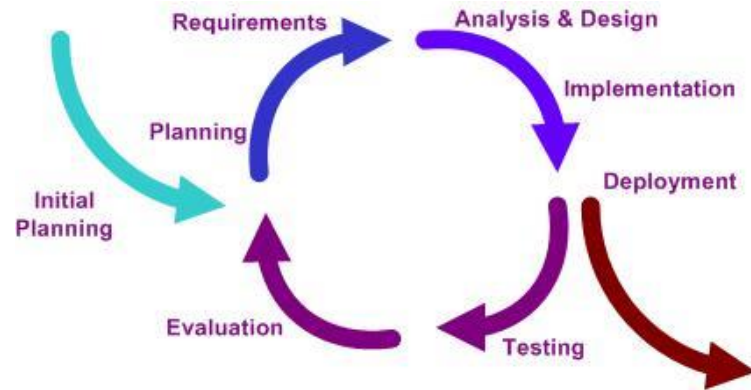
- Několik verzí systému
- Jednotlivé verze se dělají vodopádem

○ Výhody

- Jasně definovaný plán
- Predikovatelnost (čas, rozsah, cena)
- Snadná koordinace práce
- Zákazník má přístup k verzím/prototypům, čili vidí co dostane

○ Nevýhody

- Nutno chápat co se chce na již začátku, možné změny do dalších verzí
- ~~Reakce na změny (požadavků, termínů, ...)~~
 - Lepší než vodopád, ale změny nejsou dodány rychle
- ~~Rychlost dodávky (kdy zákazník něco uvidí od zadání)~~



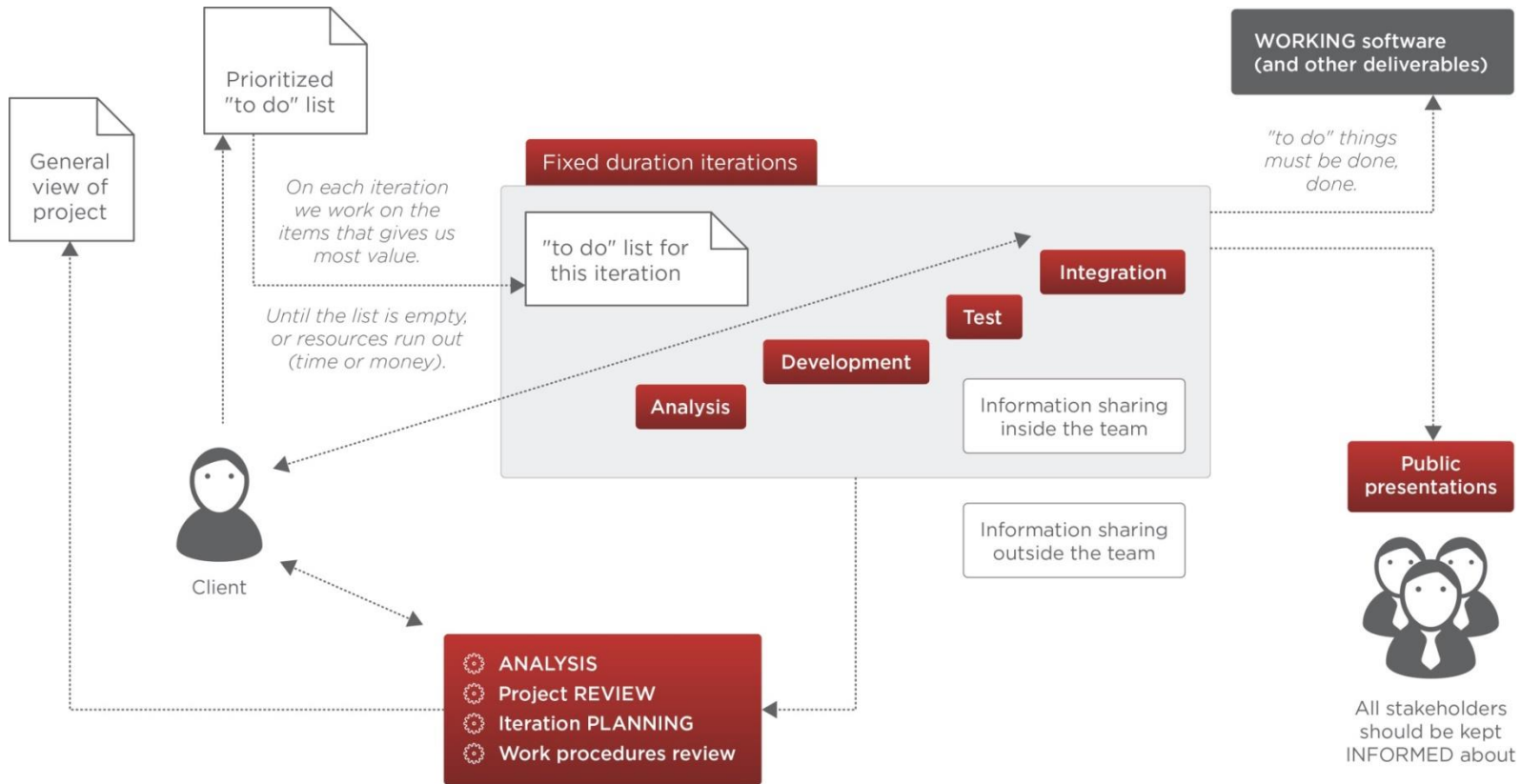
Agilní (Scrum apod.)

AGILE

Manifesto for Agile Software Dev.

- INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

Princip



2

Modely z různých pohledů

Flexibilita

- › Flexibilita
 - Reakce na změny
 - Rychlost a náklady
- › Vodopád
 - Velmi nepružný
 - Vysoké náklady
- › Iterativní
 - Lze zakomponovat změny do další iterace
 - Náklady nižší než u vodopádu
- › Agilní
 - Snadné, očekávají změny
 - Náklady na změny velmi nízké



Predikovatelnost

- › Predikovatelnost
 - Vím, co dostanu
 - Kdy
 - Kolik mě to bude stát
- › Vodopád
 - Vysoká, jedeme dle plánu
- › Iterativní
 - Vysoká, máme plán
- › Agilní
 - Nízká, plán jen na krátké období
 - Víme tedy jen co dostaneme např. během dalšího sprintu



Architektura a design

- › Architektura a design
 - Dobře navržený systém
 - Konzistentní dodržování principů návrhu
- › Vodopád
 - Vysoká
- › Iterativní
 - Vysoká
 - Riziko zanesení problémů při dalších iteracích
- › Agilní
 - Nízká
 - Riziko zanesení problémů při každém sprintu



Implementace

- › Implementace
 - Prostor pro dodání kvalitního díla
 - Požadavky na programátory
- › Vodopád
 - Kvalitní, dost prostoru pro QA
 - Revize, coding standards, ...
- › Iterativní
 - Kvalitní, dost prostoru pro QA
 - Revize, coding standards, ...
 - Riziko zanesení problémů při dalších iteracích
- › Agilní
 - Nutný kvalitní tým, může být problematická
 - Riziko nekvalitní práce, když není prostor na revize



Dokumentace

- › Dokumentace
 - Mám kvalitní a konzistentní dokumentaci systému
- › Vodopád
 - Typicky OK
- › Iterativní
 - Nutno dodržet napříč verzemi (jedna pravda či více verzí?)
- › Agilní
 - Nízká
 - Obtížné udržovat napříč sprinty



Spolupráce se zákazníkem

- › Požadavky na součinnost
 - Jak moc a často se musí zákazník podílet na projektu
- › Vodopád nebo iterativní
 - V přesně definovaných okamžicích
 - Lze dobře plánovat
- › Agilní
 - V průběhu celého projektu
 - Velké riziko selhání, pokud nebude



Smlouva na dodávku

- › Smlouva na dodávku díla
 - Lze model využít pro dodávku FTFP?
- › Vodopád
 - Ano
- › Iterativní
 - Ano, nutno ošetřit rozsah verzí
- › Agilní
 - Ne, požadavky nejsou dopředu známy
 - Aktuální modely fungování nákupu tomu nejsou nakloněny



3

Který model vybrat?

Který vybrat?

- › Vodopád
 - Téměř nepoužíváme pro již uvedené typy projektů (jen pro menší změny)
- › Iterativní
 - Nejčastěji používaný pro většinu systémů našich zákazníků
 - Jak iniciální rozvoj tak údržba
 - Predikovatelnost důležitější než flexibilita
 - Lze na velké systémy
- › Agilní (vývoj nebo dodávky?)
 - Produktový vývoj
 - Vývoj v rámci firmy, kde je jasná ochota lidí se na tom podílet
 - Spíše na menší projekty nebo na některé části procesu (např. jen vývoj)



Věci ke zvážení

- › **Systém**
 - Typ a velikost
 - Očekávaná životnost systému
- › **Tým dodavatele**
 - Znalosti a zkušenosti týmu
 - Schopnost pracovat agilně
- › **Agilně**
 - Ne vždy a ne všechno
 - **Nesmí být zástěrkou pro chaos**



4

Příklady procesu

Iterativní

- › Core systém
 - Iterativní proces dodávky systému na zelené louce (3 verze)
 - Iterativní proces údržby
 - Ukázka harmonogramu
- › Online systém sjednání pojištění
 - Iterativní
 - Ukázka harmonogramu
- › Internetové bankovníctví
 - Iterativní
 - 4 release ročně
 - Paralelně práce na až 3 release
 - Ukázka

Ne-agilní

- › Vývoj internetového bankovníctví
 - 2 problematické projekty
 - Nejasné/nepřipravené zadání
 - Nedělají se odhady
 - Neexistující dokumentace
 - Spotřeba MD mnohem větší, než bylo třeba (2-5x)
- › Core systém pro pojišťovnu
 - Aplikace agilního přístupu i na analýzu v rámci údržby
 - Omezení na délku Scrum sprintu, když nejde rozdělit na menší části

Volba správného procesu



<http://foto.mapy.cz/10380-Nedokonceny-most>

MATERIÁLY SWENG

PROJECT AND PROCESS

ČLÁNKY

- [Anchoring the Software Process](#) – klasický článek od Barry Boehma, ve kterém diskutuje existenci a význam 3 klíčových milníků v životě softwarového systému/projektu, které jsou nezávislé na konkrétním zvolném životním cyklu (SDLC, Software Development Life Cycle)
- [A Proposal for a Recursive Object Oriented Life-Cycle](#) – článek diskutující ideu rekurzivního SDLC (životního cyklu)
- [The Joel Test: 12 Steps to Better Code](#) – populárně naučný článek od zajímavého člověka se zajímavou minulostí, který se snaží vyjmenovat několik základních a elementárních kroků k lepšímu a lépe řízenému projektu.
- [Manager's Handbook for Software Development](#) – příručka z roku 1990 vytvořená v SEL NASA (Software Engineering Laboratory), která nabízí přehled metod a nástrojů software managementu, informace o tom, čeho lze použitím uvedených metod dosáhnout, kdy a jak mají být aplikovány a samozřejmě, kde lze nalézt informace další.
- [Software Management Guidebook](#) – aktualizovaná a přepracovaná Manager's Handbook for Software Development z roku 1996
- [Software Process Improvement Guidebook](#) – příručka pro implementaci Software Process Improvement (SPI) programu; popisuje základní koncepty programu SPI a detailní informace o tom, jak program úspěšně definovat a implementovat
- [Profinit Software Engineering Best Practices](#) – starší, rozsáhlejší, verze minimálních nároků na dobrou praxi softwarového inženýra v Profinitu; je velmi užitečná, neboť obsahuje popis jednotlivých nároků a praktik ve větší míře detailu a s mnoha odbornými zdroji
- [Profinit Software Engineering Minimal Practices](#) – aktuální, stručnější, verze minimálních nároků na dobrou praxi softwarového inženýra v Profinitu; základ firemního procesu
- [Software Process Improvement](#) – obecné uvedení do tématu SPI inspirované zaváděním SPI programu v Profinitu

Všechny odkazované materiály jsou poskytnuty výhradně za účelem výuky softwarového inženýrství.

© Of Respective Parties 2007-2009

OBLASTI SWENG

› Project and process

› Project management

› Documentation

› Quality assurance

› Configuration management

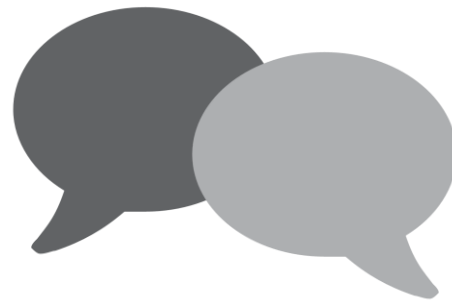
› Testing

› Construction

› Architecture and design

› Requirements

› Obecné, přehledové



Diskuze

Děkujeme za pozornost

PROFINIT

Profinit, s.r.o.
Tychonova 2, 160 00 Praha 6



Telefon
+ 420 224 316 016



Web
www.profinit.eu



LinkedIn
linkedin.com/company/profinit



Twitter
twitter.com/Profinit_EU