

Agilní metodiky vývoje Software

ČVUT-FEL

Tomáš Borovička

Osnova

- Metodiky a životní cykly
- Agilní metodiky vývoje
 - Extreme Programming
 - SCRUM Development Process

Proč metodiky

- SW krize
- Týmový vývoj
- Rozsáhlé projekty
- Kvalita, rychlost
- Podpora, údržba

- Efektivita

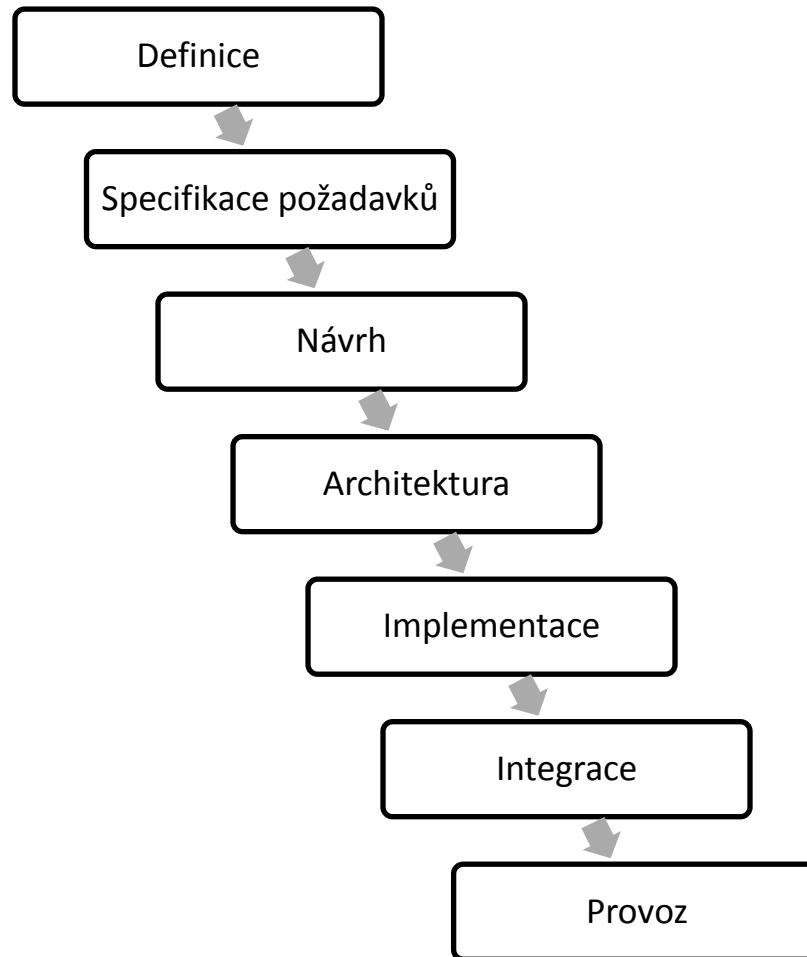
Životní cykly

- Code and Fix
- Stagerwise
- Waterfall
- Spiral

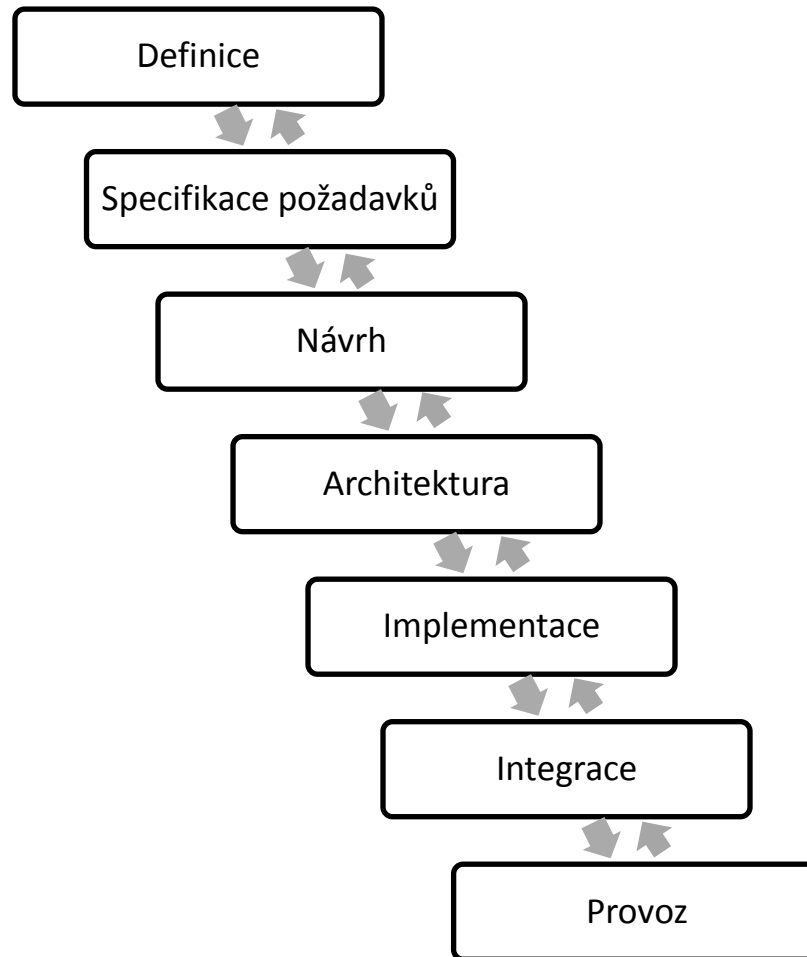
Code and Fix



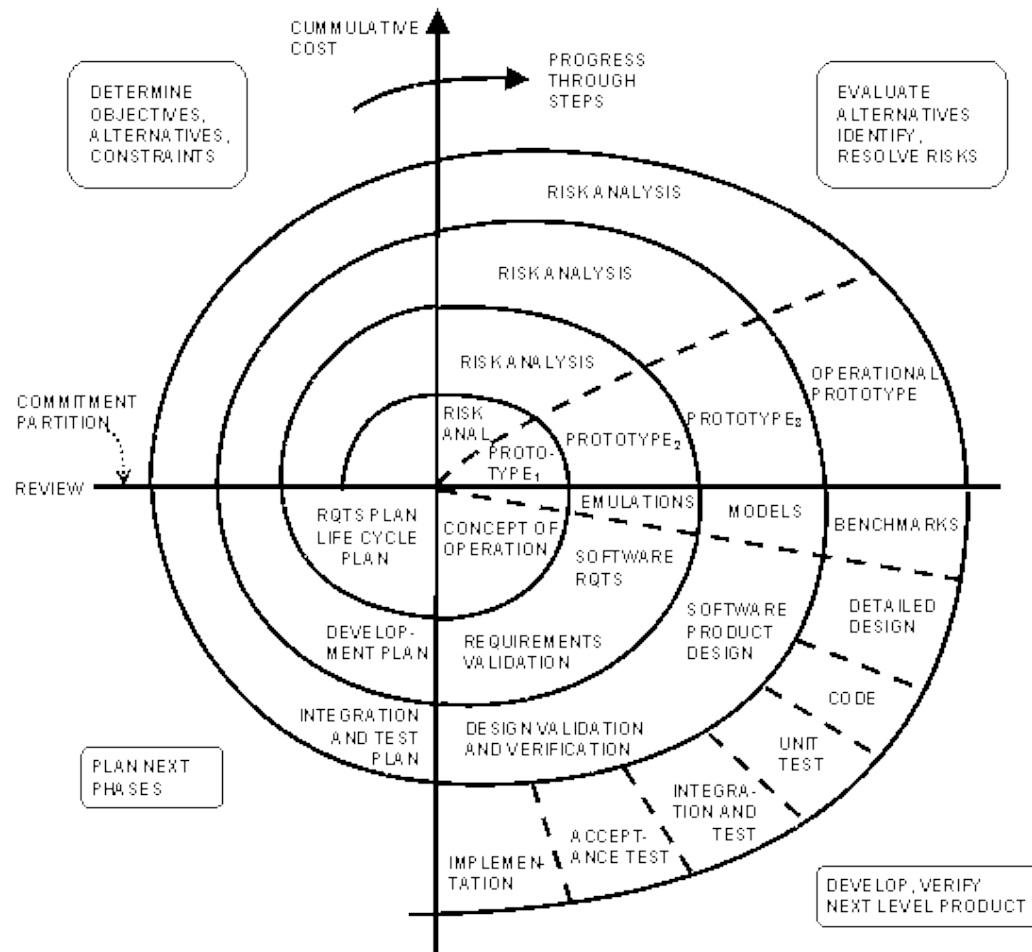
Stagewise



Waterfall



Spiral



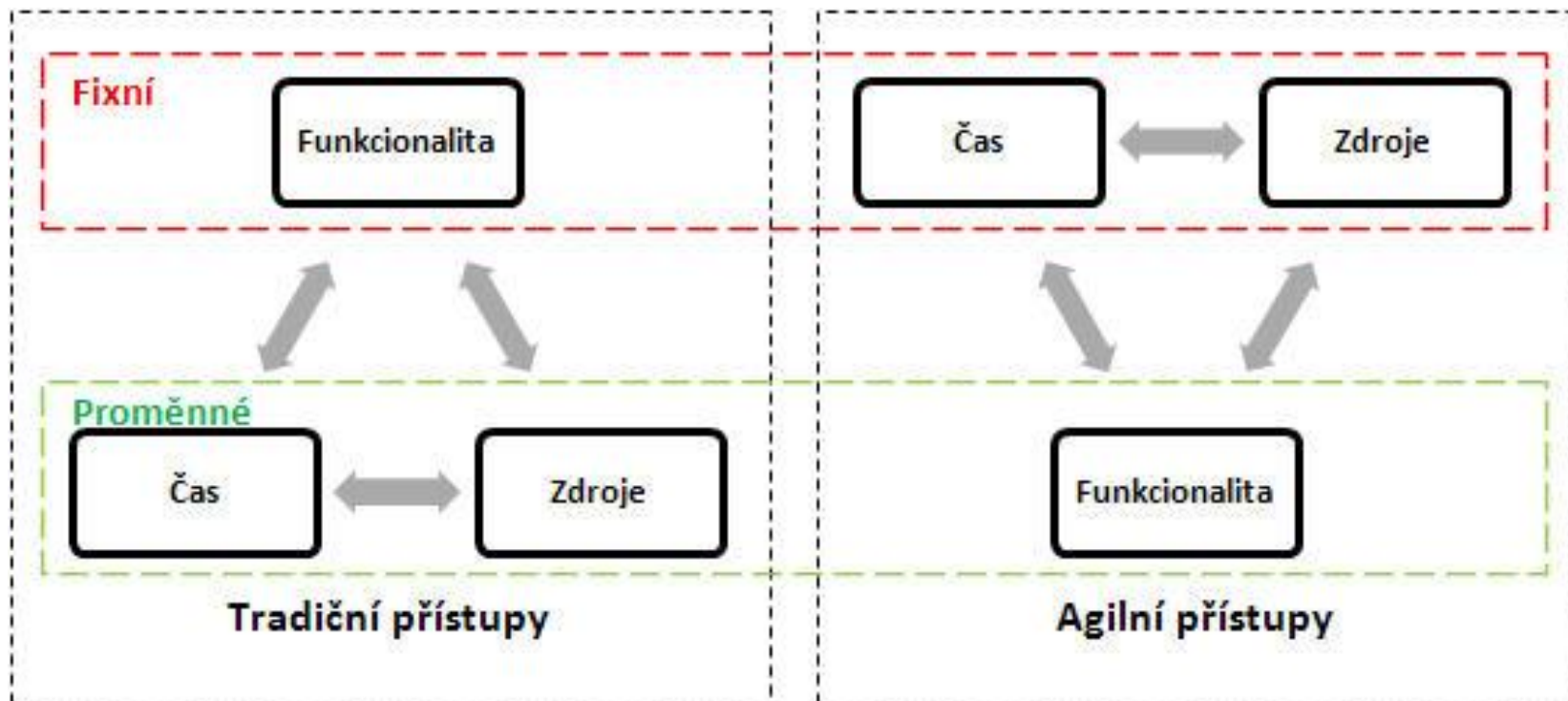
RUP

- Rational Software
- Komerční produkt
- Objektově orientovaná iterativní metodika
- Základní element USE CASE
- Fáze
 - Zahájení
 - Projektování
 - Realizace
 - Předání
- Free alternativa USDP (UP)

Agilní metodiky

- Agile - rychlý, hbitý, čilý svižný.
- Agile Alliance - aliance pro agilní vývoj
- Agile manifesto - manifest agilního vývoje

Agilní metodiky



Iterace

- Iterativní a inkrementální vývoj s velmi krátkými iteracemi.
- Nové funkce se dodávají často (i denně).
- Zákazník pořád vidí v jakém je projekt stavu

Komunikace v týmu

- Množství problémů vychází ze špatné komunikace
- Integrace komunikace přímo do procesu

Komunikace se zadavatelem

- Člen vývojového týmu
- Komunikace s vývojáři
- Návrh
- Testy

Testování

- Časté změny – nutnost ověřování správnosti
- Automatizované testy
- Zachování kvality aplikace

The Agile manifesto

- Základní pilíř agilních metodik.
- Vychází ze dvou základních tezí:
 - 1)Přijmout a umožnit změnu je mnohem efektivnější, než pokoušet se jí zabránit.
 - 2)Je třeba být připraven reagovat na na nepředvídatelné události.

Agilní metodiky

- Užitná hodnota pro zákazníka
- Změny jsou výhodou
- Časté dodávky
- Zákazníci spolupracují s týmem
- Motivace je klíčová
- Vzájemná konverzace
- Úspěch posuzujeme podle fungování
- Udržitelný vývoj
- Perfektní návrh, perfektní řešení
- Zásadní je jednoduchost

Agilní metodiky

- **Extreme Programming**
- **SCRUM Development Process**
- Test-driven Development
- Lean Development
- Feature-Driven Development
- Adaptive Software Development
- Crystal Methodologies

Extreme Programming

- Kent Beck (1999)
- Nejrozšířenější a nejznámější
- Malé a střední týmy (2-10)
- Zábavnější vývojový proces
- Životní styl – ne pro každého

Extreme Programming

Víte jak se pozná dobrý extrémní programátor?

Zapisuje přímo binární kód do souboru *.exe.

Víte jak se pozná skvělý extrémní programátor?

Píše rovnou komprimovaný kód do souboru *.zip

Extreme Programming

„Jediným exaktním, jednoznačným,
změřitelným, ověřitelným a
nezpochybnitelným zdrojem informací je
zdrojový kód.“

Proč extrémní?

- Používá běžně známé postupy a myšlenky, ale jejich používání dotahuje do extrému.
- **Jednoduchost**
- **Revize a kontroly zdrojového kódu**
- **Návrh**
- **Architektura**
- **Testování**
- **Krátké iterace**

Proměnné vývojového cyklu XP

- Kvalita
- Čas
- Náklady
- Šíře zadání (nová proměnná, kterou XP přidává)

Ideový rámec XP

- Komunikace
- Jednoduchost
- Zpětná vazba
- Odvaha

Základní postupy XP

- Plánovací hra
- Malé verze
- Metafora
- Jednoduchý návrh
- Testování
- RefaktORIZACE
- Párové programování
- Společné vlastnictví
- Nepřetržitá integrace
- Žádné přesčasy
- Zákazník na pracovišti
- Standardy pro psaní kódu

Párové programování

- Všechny zdrojový kód píší dvojice programátorů. Každá dvojice sdílí jeden monitor, klávesnici a počítač.
- Dvě role:
 - Lokální
 - Globální

RefaktORIZACE

- RefaktORIZACE je restrukturalizace systému beze změn jeho chování.
- Kent Beck - metafora klobouků

„Kód, kterému rozumí jen počítač, dokáže napsat každý. Dobří programátoři píší kód, kterému rozumí lidé.“

Činnosti XP

- Testování
- Psaní zdrojového kódu
- Poslouchání
- Navrhování

Životní cyklus XP

- Průzkum
- Plánování
- Iterace
- Zprovozňování
- Údržba
- Smrt

Použití XP

- 2-10 členů
- Nehodí se pro velké týmy
- Ne každý je vhodný pro XP
- Párové programování
- Kolektivní vlastnictví kódu
- Odvaha

SCRUM



SCRUM

- Ken Schwaber, Mike Beedle – 1995
- Cíl: vyvinout flexibilní metodiku, která se dokáže vyrovnat s měnícími se požadavky.
- Iterativní přístup

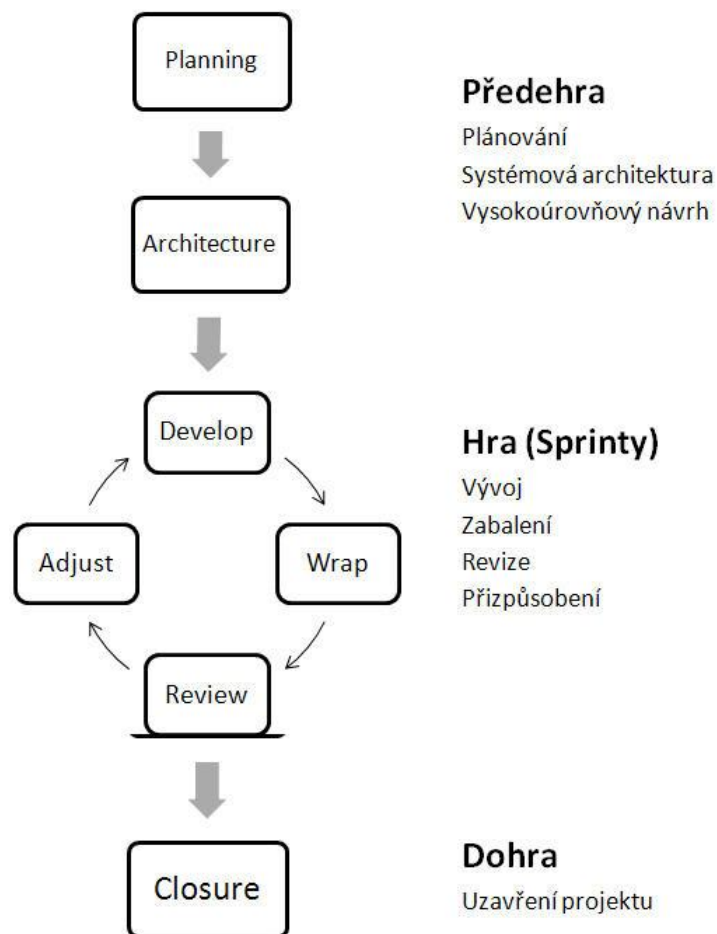
SCRUM

- Malé týmy 3-6 členů
- Každý vývojář odpovídá za množinu objektů
- 3-8 iterací, v každé se vytvoří funkční demo
- Spolupráce – denní schůzky

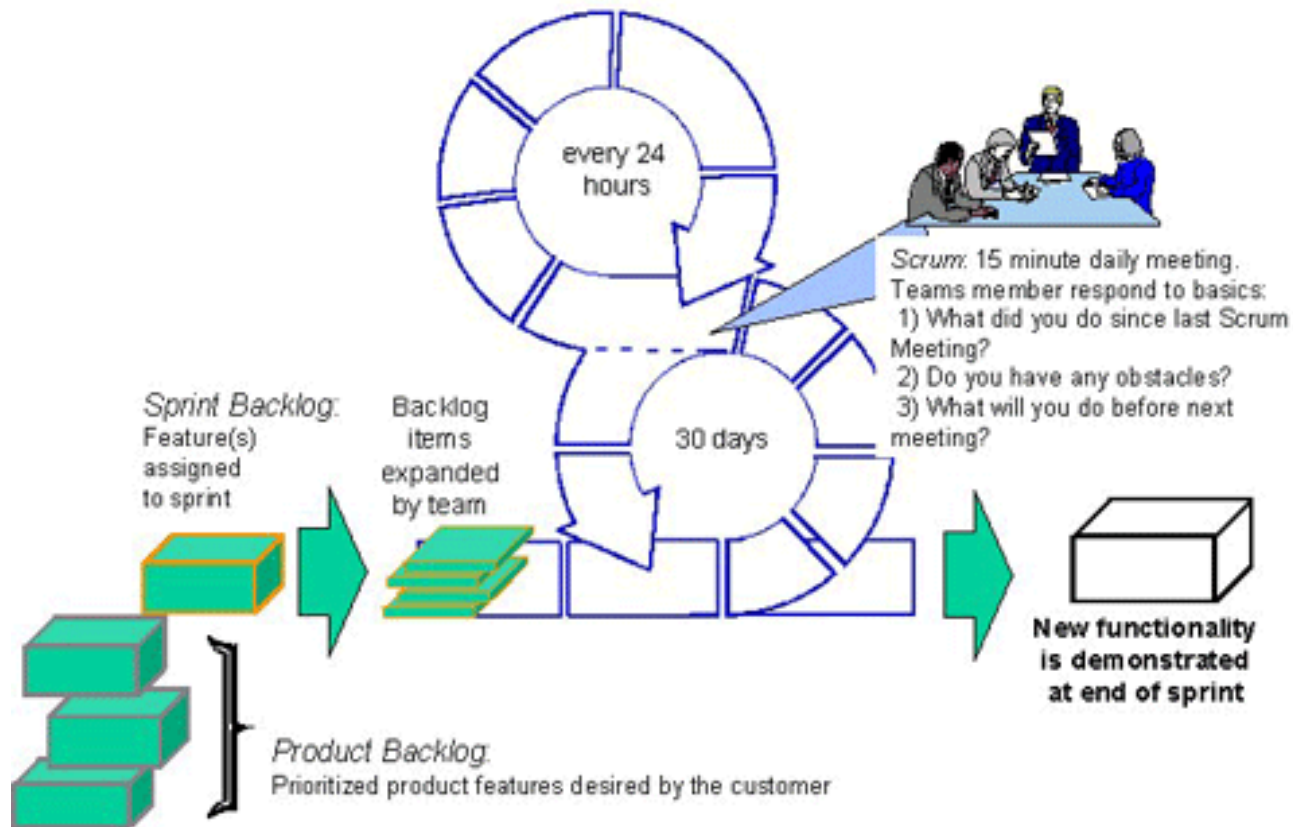
SCRUM - pojmy

- Backlog
- Riziko
- Sprint
- Scrum meeting

SCRUM - životní cyklus



SCRUM meetings



Feature Driven Development

- Devadesátá léta
- Postavena na vlastnostech produktu
- Krátké iterace – přidání feature
- Dobrá kontrola
- Globální model
- Seznam vlastností
- Plánování , návrh, implementace

Lean development

- Převzato z výrobních odvětví
- Absolutní odstranění všeho zbytečného
- Vyvíjet SW za třetinu obvyklého času
- Vystačit s třetinou obvyklého rozpočtu
- Snížit četnost chyb na třetinu

Test Driven Development

- Nejprve je nutné napsat test, který dokáže dokonale ověřit funkcionalitu
 1. Napíšeme test
 2. Otestujeme dosavadní aplikaci (test selhává)
 3. Napíšeme modul
 4. Znovu otestujeme

Děkuji za pozornost