

# ALT+F4

## Přednáška č. 2 – Iterativní vývoj softwaru

### Software má N rozměrů

► **Vývoj software má N rozměrů**

► ... běžná aktivita v informační společnosti

Na zakázku  
Interní projekt  
Krabicový software  
„Pro radost“

Closed source  
Open source (+ reuse)

Utilita  
Systémová komponenta  
Mission-critical software

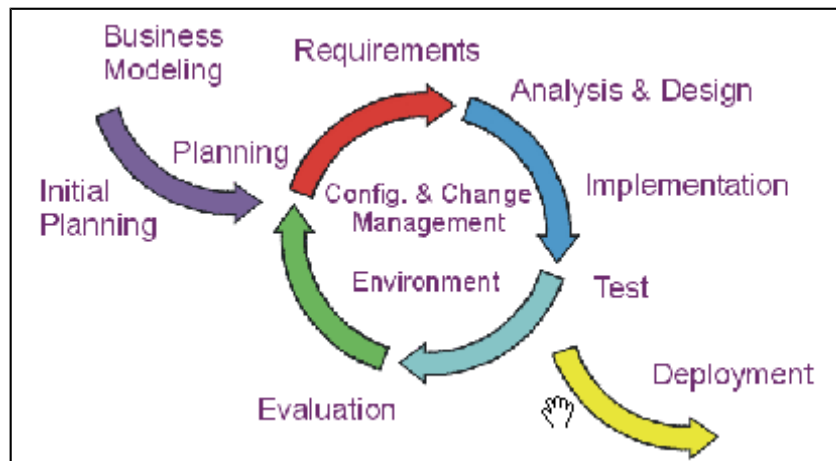
Na zelené louce (green field)  
Rozvoj existujícího produktu  
Integrační projekt

Komerční zákazník  
Státní sféra  
Vertikály (utility, banky, telco, ...)

Ukládejte a sdílejte soubory ve službě Document Cloud  
Další informace

### Průběh iterace

- plánování cíle iterace (zejména funkční část)
- doplnění nebo zpřesnění požadavků (plán projektu, vize, předchozí feedback)
- (úprava návrhu)
- implementace přírůstku funkčnosti
  - jednoduchá cesta – práce na iteraci
  - lepší cesta – práce na jednotky (itemy)
- integrace přírůstku do stávajícího projektu (ověření a otestování)
- (možnost předání do provozu) – validace zákazníkem
- zhodnocení iterace



## Milníky globálního plánování

- LCO (LifeCycle Objectives) – definování vize produktu
- LCA (LifeCycle Architecture)
  - určení způsobu řešení (architektura technického řešení)
  - ověření (modely, testy, technické prototypy)
- IOC (Initial Operational Capability)
  - schopnost efektivně vyrobit řešení – výstupem je beta verze, která má všechny featury
  - jednotkové testy a funkční testy
- GA (General Availability)
  - uvedení produktu do běžného provozu – dokončený produkt
  - v provozu support tým

## **Přednáška č. 3 – Fáze Inception (zahájení)**

### Milník fáze Inception

- kdy je fáze dosažena
- LCO
  - srozumění s obsahem, cenou a harmonogramem
- Artefakty
  - Vize produktu
  - Bussiness case
  - Seznam rizik a strategie, jak je vyřešit
  - Slovník pojmů a přehled klíčových požadavků
  - Koncept technického řešení – architektura a prototypy
  - Plán projektu
  - Popis procesu a infrastruktury

### Postup práce s požadavky

- požadavky vývoje
  -
- požadavky managementu
- **DODĚLEJ**

### Typy požadavků

- Bussiness požadavky
  - Vize projektu
  - Rozsah projektu
- Funkční (user) požadavky
- Bussiness pravidla, omezení
- Extra funkcionality
  - vlastnosti
- Systémové požadavky
- Licenční, legální požadavky

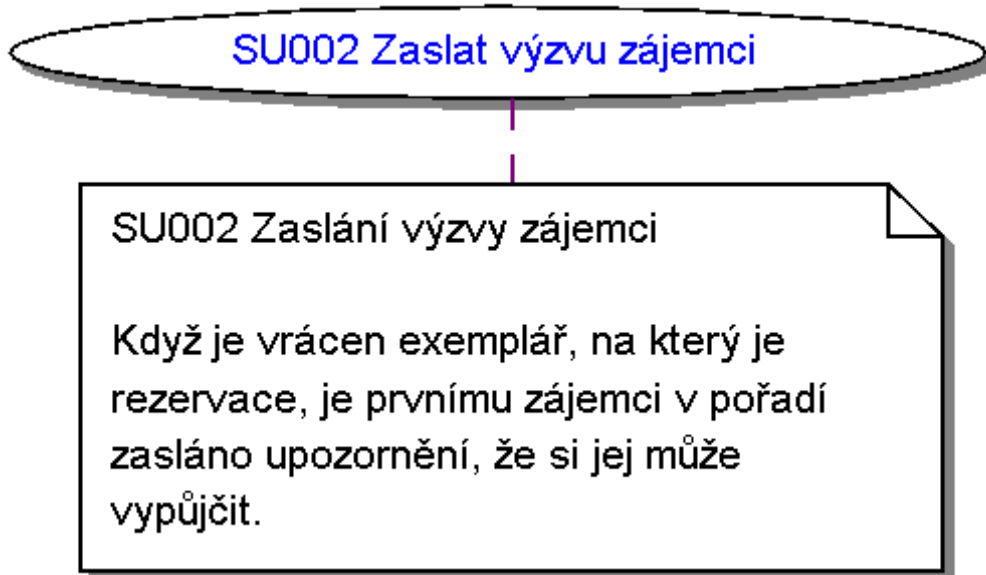
## Kostra vize produktu

- Popis problému a účelu
  - smysl a účel cílového produktu
  - obchodní příležitost, důvod ekonomické návratnosti
- Přehled stakeholderů
  - kdo jsou zájemci o systém, kdo to bude používat, typy uživatelů
  - potenciální konkurence
- Přehled očekávaných schopností a funkcí produktu
  - popis, kvantitativní charakteristiky, priority
  - stručný výčet bez detailů
- Omezení, standardy, závislosti
  - vztahující se k projektu
- Rámec plánu projektu – časový rozsah, plánované verze, plánovaná vydání

## Základní popis případu užití

- při shromažďování požadavků – základní popis dané funkce
- Název (možno přidat ID)
- 1 věta o tom, co vlastně dělá
- Případě
  - základní kroky postupu pro klíčové případy užití
  - odkazy na zdrojové informace

## Obrysy požadavků s případy užití



- charakteristiky aktérů
- model užití
- jeden dokument, diagram jako obsah nebo jako informace v UML nástroji
- Obrázek (02b – požadavky – 24. slide)

## Forma User story

- co uživatel od systému očekává a proč
- obsahuje název, stručný popis, důležitost
- zapisuje se pomocí karty a nebo položky v ALM nástroji

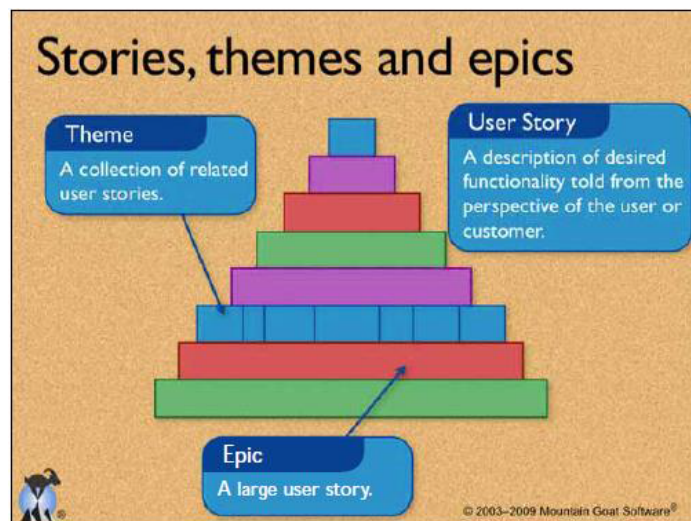
173

As a student I want to purchase  
a parking pass so that I can  
drive to school

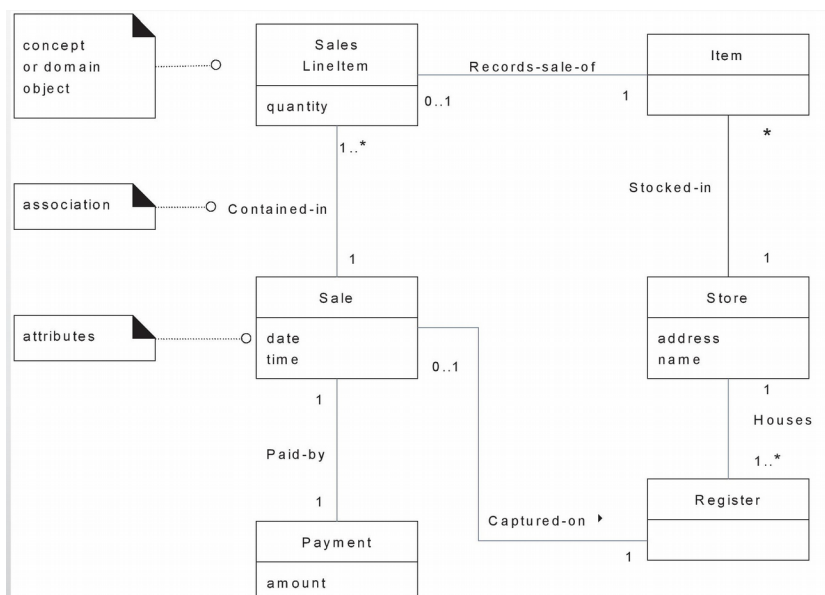
Priority: ~~High~~ Should  
Estimate: 4

## Obrysy požadavků agilně – backlog

- product backlog = základní struktura
  - obsahuje epics a user stories
  - just-in-time (zpřesňování v další iteraci)
  - nejen požadavky



## Obrysy struktur s UML – doménový diagram



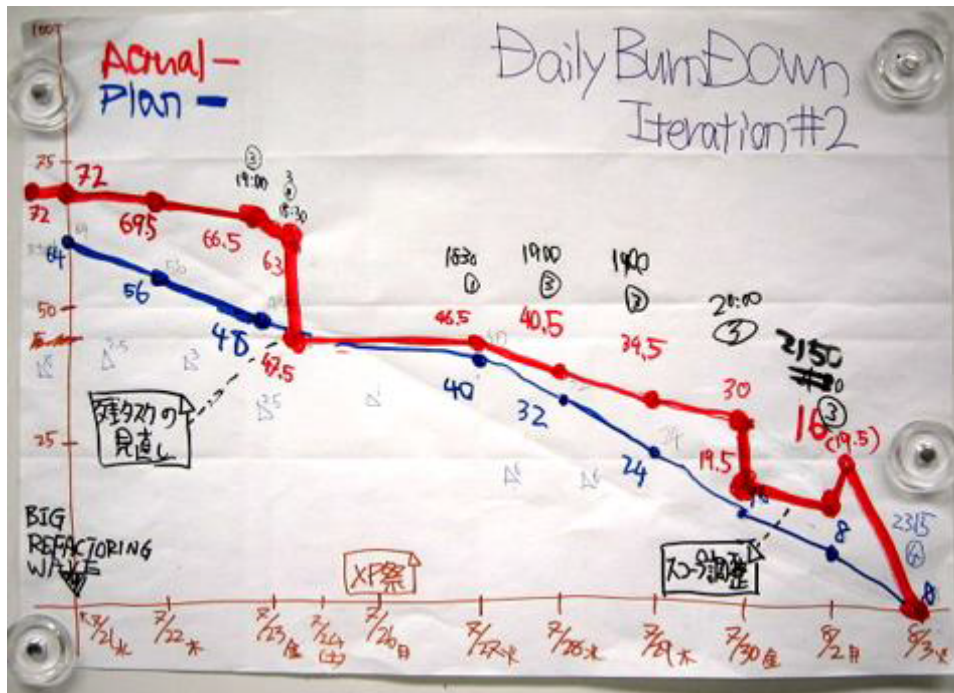
## Stupně volnosti při plánování

- klasické – čas, zdroje (cena), kvalita
  - obtížně měnitelné a odhadované
  - kvalita je obtížně říditelná
    - bude to v daném termínu, za danou cenu a samozřejmě kvalitní produkt
    - většinou – pozdě dostaneš hroznej software
- agilně – přidává se funkčnost
  - nejlepší faktor pro řízení projektu
    - první tři pevné, funkčnost nejsnáze měnitelná
  - vhodná granularita (kouskování) snadné a přesné odhady

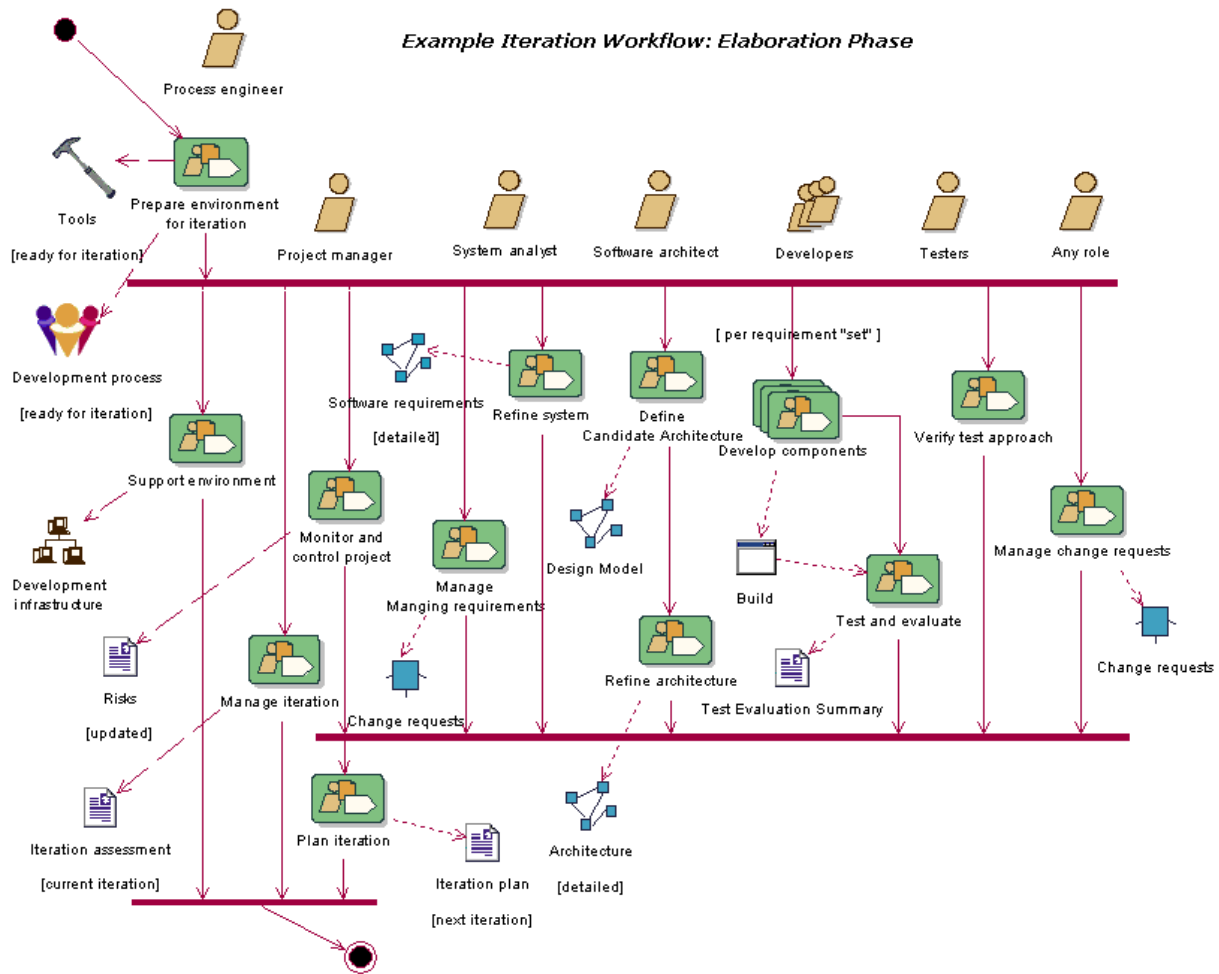
## Backlog jako plán

- Požadavky, priority, odhadování, plánování, průběh
- Product backlog
  - epics a user stories → požadavky na produkt
  - priority a rozpracovanost → pořadí implementace
- Iterační backlog
  - stories a tasky → plán iterace
- Aktivity související s plánování
  - Product – backlog grooming, dot Votic
  - Iterační – planning meeting, daily standup

## Burndown chart



# Přednáška č.3 – fáze projektu Elaboration (projektování)



## Milník

- kdy jsou cíle fáze dosaženy
- LCA (LifeCycle Architecture)
  - vize a klíčové požadavky jsou stabilní
  - testy ověřily, že architektura řeší rizikové faktory a požadavky
  - přesnější odhady pracnosti, na nich postavené plány
  - nástroje a postupy pro realizaci jsou v provozu
  - stakeholderi – vize realizovatelná, spotřebované zdroje adekvátní
- artefakty
  - aktualizace vize produktu, specifikace požadavků
  - aktualizace seznamu rizik a strategie řešení
  - popis architektury a validační testy
  - plán projektu a popis infrastruktury

## Validace architektury

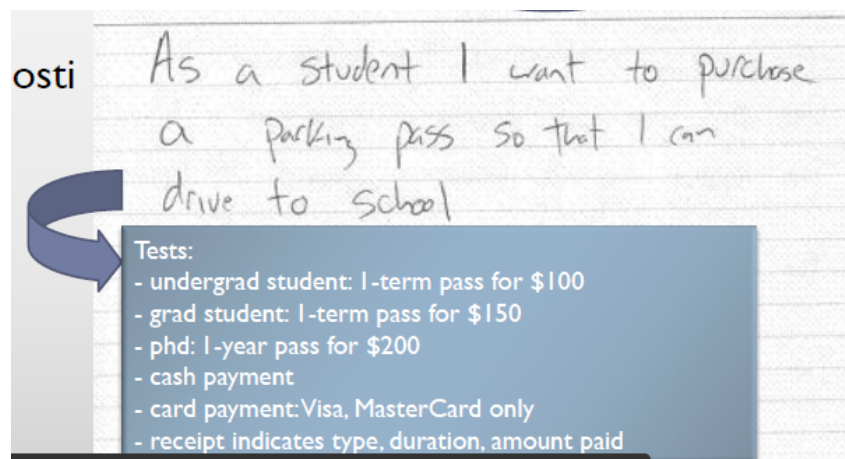
- Bude IOC a REL v tomto postavený splňovat LCO?
  - executable architecture a její validace
  - architektonicky důležitá funkčnost (případy užití)
- mechanismy
  - návrh na základě klíčových případů užití a mimofunkčních požadavků
  - referenční architektura
  - implementace proof of koncept
  - oponentura

## Podrobný popis případů užití

- detailní rozbor komunikace aktér – systém
- standartní průběh
  - nejčastější sled akcí
  - bez chyb a různých možností
- vstupní a výstupní podmínky
  - co potřebujeme pro standartní průběh
- chybové stavy a alternativní průběhy
  - určení míst výskytu, příčin a následků
  - popis alternativních a chybových akcí

## Obsah User stories

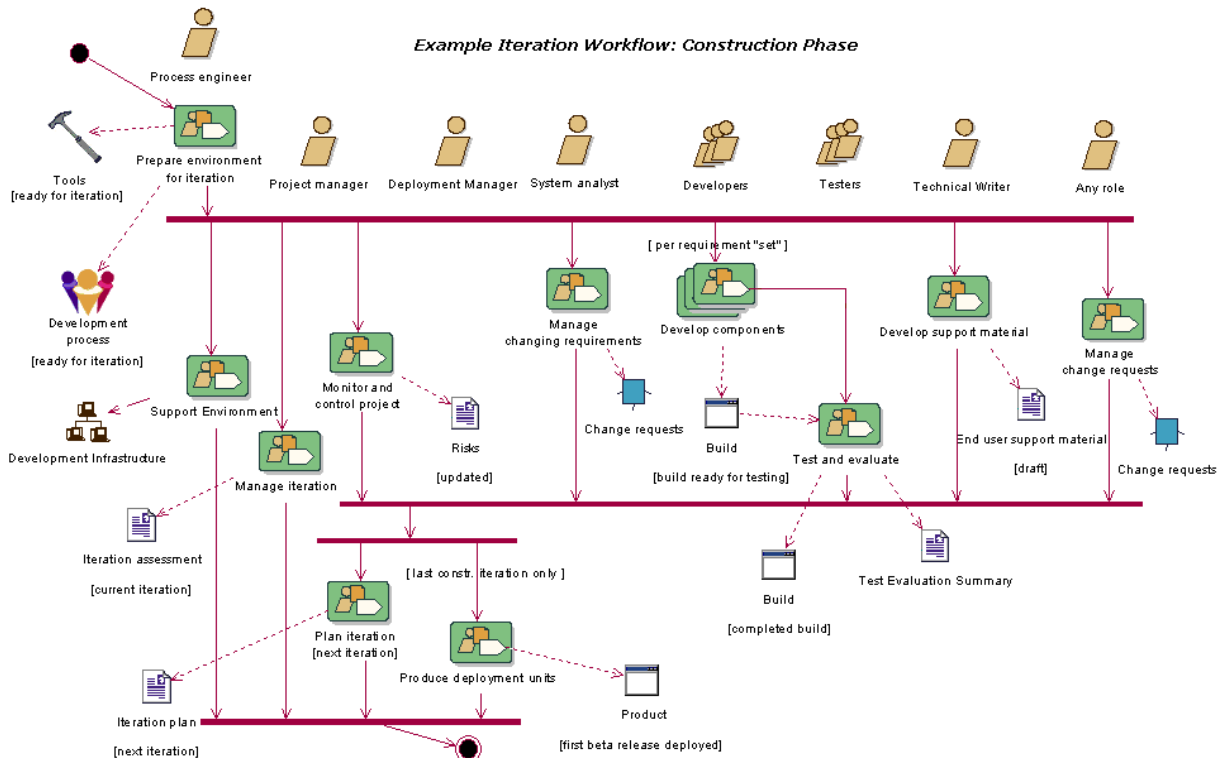
- popis jedné funkčnosti z pohledu uživatele
  - bussiness value
  - terminologie
  - Jako .... chci ... (takže)...
- hlavní vlastnosti
  - stručnost
  - ověřovací kritéria



## Reprezentace vlastností

- účelem je možnost ověřit splnění v implementaci
- měřitelný způsob (numericky)
  - obvyklá hodnota, povolené odchylky, četnost, přírůstky
  - zvážení realizovatelnosti funkčních požadavků
  - vhodná reprezentace pro neměřitelné
- popis vlastností
  - u případů užití
  - u tříd problémové oblasti
  - vázané na celý systém
- příklad CoCoME

# Přednáška 4. Fáze Constuction (konstrukce)



## Milník

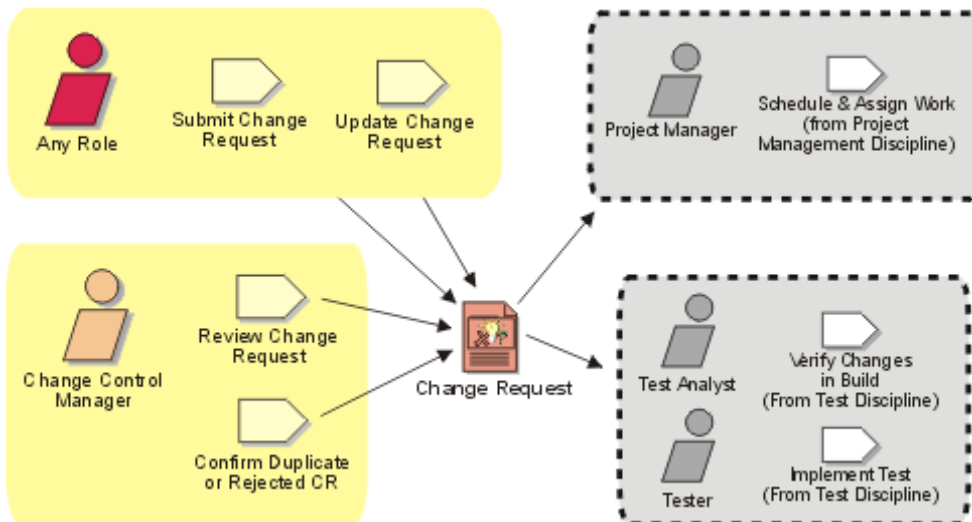
- kdy jsou cíle fáze konstrukce dosaženy
- IOC
  - hotová beta verze produktu
- Artefakty
  - architektura aktualizovaná, popisy implementace
  - testovací sady a reporty
  - první verze plánu nasazení
  - uživatelská příručka a podpůrné materiály (draft)

## Cyklus správy změn

- požadavek na změnu – stavy
  - vytvořený
  - schválený
  - přiřazení
  - vyřešený
  - ověřený
  - uzavřený
  - znovuotevřený
  - odmítnutý
- během provádění
  - znovuotevření projektu
  - vygenerování nových hlášení

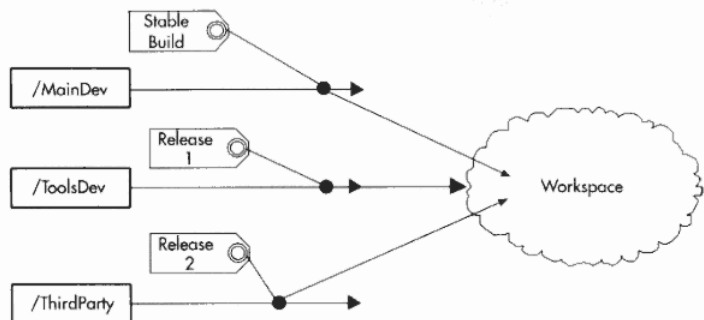


## Role ve správě změn



## Codeline

- vývojová linie
- série podob (verzí) množiny prvků konfigurace tak, jak se mění v čase
- extenzionální verzování podle stavu obsahuje commity a nebo changesety, větve
- vrchol obsahuje nejčerstvější verzi
- má přiřazena pravidla práce na codeline
- konfigurace se může skládat z více codelines
  - vlastní projekt
  - knihovní codeline



## Vzory pro verzování

- hlavní vývojová větev (mainline) a pravidla linie (codeline policy)
  - jedna codeline pro průběžný vývoj
- stabilizační období (code freeze)
  - pravidla před release
- větev pro release a jeho přípravu (release line)
  - místo code freeze mít samostatnou větev pro release
- kód třetích stran na větev (third party codeline)
  - vlastní větev pro každý kód od externího dodavatele
- privátní verze (private versions)
  - soukromé úložiště, častější check-in
- větev pro úkol (task branch)
  - složitější úpravy s většími následky dělat na větvi
  - vazba na správu změn
- check-in pro každý úkol (task-level commit)
  - minimum nutného
  - po ukočení práce na jednom úkolu udělat check-in změn

## Vlastnosti sestavení

- jedinečnost a identifikovatelnost
  - PROJEKT\_v2\_build2134\_20041220T1954
  - identifikátor jednoznačný, identifikovatelný
  - stvořitelný a zpracovatelný automaticky
- úplnost
  - tvoří kompletní systém, obsahuje všechny komponenty
- konzistence
  - vzniklo ze správných verzí správných komponent
- opakovatelnost
  - možno opakovat build daného sestavení kdykoliv v budoucnu
- dodržuje pravidla vývojové linie
  - build odpovídající baseline
  - zejména release má striktní pravidla

## Integrační sestavení

- cílem je spolehlivě ověřit, že produkt jde sestavit
  - soukromý build nestačí
  - složité závislosti, specifika ve workspace, zjednodušení pro zrychlení
  - úplné sestavení trvá dlouho → nemůže provádět vývojář
- celý produkt sestaven centrálně, automatizovaným a opakovatelným způsobem
  - postup co nejpodobnější sestavení pro release
    - vždy na zelené louce
  - maximální automatizace – typicky běží přes noc
  - spolehlivé mechanismy, zaznamenání chyb a informování o nich
    - emailové notifikace, začátek, konec výsledek
    - web s přehledy a detaily
  - úspěšné sestavení může být označováno ve verzovacím systému

## Daily build a smoke test

- integrační sestavení + zkouška těsnosti
  - pravidelně 1x denně, někdy nočně
  - výsledky okamžitě známy a reflektovány
    - nová hlášení problémů
    - opravy ihned zpracovány do kodu
  - check-in kódu, který vede k chybám, je neslušné chování
    - lehká sankce vhodná
- výhody
  - malé množství změn během denních check-in
  - zvladatelné množství oprav, včas detekce problému, analýza změn kódu místo ladění – diskuze o sestavení
  - pravidelný obecně známý rytmus projektu
  - lepší morálka týmu
  - cenou je trocha disciplíny a automatizace

## **GQM – Goal-Question-metric**

- přístup k definování metrik
- GOAL – problém a cíl měřícího programu
  - zlepšit spravedlivost v oceňování práce na projektu
- Question – měřené objekty a způsob měření
  - kolik práce odvádí jednotliví členové týmu?
- Metric – konkretizují získávaná data
  - počet řádek uložených v SVN, váha uzavřených tasků v bug trackeru

## **Přednáška 5 – fáze Transition (předání produktu)**

### **Milník**

- GA – product release
  - výsledky zákaznickova přezkoumání a přijetí výstupného projektu
- Artefakty
  - release produktu
  - podpůrné materiály (Uživatelská dokumentace)
  - baseline kompletní konfigurace release
  - ... dle povahy produktu