

Scrum

Základní charakteristika

- první varianta představena v roce 1995
- název odkazuje na důležitost týmové práce (odvozeno z hry *rugby*)
- dá se kombinovat s programovacími praktikami XP
- tři základní fáze
 - *pre-game*
 - *development (game)*
 - *post-game*

Reference

- <http://www.controlchaos.com>
- Ken Schwaber, Mike Beedle. *Agile Software Development with SCRUM*.

Úvod do softwarového inženýrství

IUS 2019/2020

11. přednáška

Ing. Radek Kočí, Ph.D.
Ing. Bohuslav Křena, Ph.D.

9. a 13. prosince 2019

3 / 61

Scrum Proces: Pre-game

Plánování

- počáteční seznam požadavků *Product Backlog*, seřazený podle priorit (*backlog*) – nedodělánky, nevyřízené objednávky)
- analýza rizik
- odhad času, zdrojů, ...
- formování týmů (*scrum teams*)
 - jeden tým má 5 až 10 členů
 - každý člen má jinou specializaci
 - *Scrum Master* – vedoucí, zajišťuje správné používání *Scrum* praktik

Architektonický návrh

- analýza problémové domény na základě *backlog*;
tvorba doménových modelů, prototypů, ...
- definice architektury systému
- úprava požadavků (*backlog*) podle navržené architektury

Téma přednášky

- **Agilní metodiky – dokončení**
- **Jiný úhel pohledu na vývoj softwaru**
 - Management SW projektů
 - Řízení kvality softwaru
 - Měření v SW inženýrství
 - *Softwarový tým*
 - *Motivace lidí*

Scrum Development: Sprint Development

Popis fáze

- analýza, návrh a implementace požadavků plynoucích z cíle *Sprintu* a úloh definovaných v *Sprint backlog*
- pro efektivní řízení aktivit se konají setkání týmů

Scrum Meeting

- každodenní, 15 minutová setkání týmu
- účastní se členové týmu, *Scrum Master*, management
- základní otázky
 - co bylo uděláno od posledního setkání
 - jaké překážky se objevily
 - co bude uděláno do příští schůzky

Scrum Proces: Development

Popis fáze

- probíhá v iteracích; iterace se nazývá *Sprint*
- typická délka iterace je 30 dnů
- výsledkem je funkční část (inkrement) odpovídající *Sprint Backlog* (podmnožina *Product Backlog*)
- každá iterace obsahuje
 - plánování (*planning*)
 - vývoj (*development*)
 - posouzení (*review*)

7 / 61

5 / 61

Scrum Development: Sprint Review

Popis fáze

- na konci každého *Sprintu*
- demonstruje se výsledný produkt (*inkrement*)
- vyhodnocení dosažených výsledků ve srovnání s cílem *Sprintu*
- úprava *Product backlog*
 - plně implementované požadavky jsou označeny
 - nutné úpravy (opravy chyb nebo vylepšení) jsou přidány
 - změny či nové požadavky jsou začleneny
- vyhodnocení úsilí, splnění cíle, možné změny architektury systému

Scrum Development: Sprint Planning

Setkání

- na začátku každého *Sprintu*
- účastníci: vývojový tým, uživatelé, zákazníci, management, *Scrum Masters*, ...
- definuje se cíl *Sprintu*

Sprint Backlog

- vývojový tým definuje *Sprint backlog*
- seznam úloh nutných pro dosažení cíle
- je implementačně orientovaná, rozšířená podmnožina *Product backlog*
- jednotlivé položky jsou rozděleny mezi týmy

8 / 61

6 / 61

Crystal

Základní charakteristika

- rodina metodologií
 - různé projekty potřebují různé přístupy
- základní přístupy a techniky sdílí s XP
 - ale na rozdíl od XP není tak přísně disciplinovaný proces
- obecně je méně produktivní než XP
 - ale více lidí je schopno tento proces lépe akceptovat

Reference

- <http://alistair.cockburn.us/Crystal+methodologies>
- Alistair Cockburn. *Agile Software Development: The Cooperative Game*.
- Alistair Cockburn. *Crystal Clear: A Human-Powered Methodology for Small Teams*.

11 / 61

Crystal

Kategorie projektů

- projekty jsou rozděleny do kategorií podle kritičnosti, důležitosti (*criticality* /*jaké jsou ztráty, pokud vývoj selže*/) a velikosti
- kritičnost (důležitost)
 - Comfort (C)
 - Discretionary /*volné uvážení*/ Money (D)
 - Essential /*zásadní, nezbytný*/ Money (E)
 - Life (L)
- velikost – maximální počet lidí zapojených do vývoje

Složitost projektů

- větší projekt potřebuje komplexnější metodiku a lepší koordinaci
- kritičtější projekt (*criticality*) potřebuje rigoróznější (preciznější) přístup

12 / 61

Scrum Proces: Post-Game

Popis fáze

- integrace výsledků jednotlivých *Sprintů* (inkrementů)
- testování celého systému
- příprava dokumentace
- zaškolování uživatelů
- akceptační testování

9 / 61

Scrum: Vyhodnocení

Silné stránky

- iterativní inkrementální proces
- časté uvolňování verzí (inkrementů)
- architektura systému je navržena před procesem vývoje
- požadavky se *ladí* během celého vývoje
- zapojení uživatelů
- jednoduchý proces

Slabé stránky

- nedefinuje přesný postup úloh
- integrace až po vytvoření všech inkrementů
- předpoklad, že přímá komunikace je vhodná pro všechny typy projektů
- nepředepisuje modely pro návrh, často se od *Project backlog* přechází na implementaci

10 / 61

Je RUP agilní metodikou?

Základní charakteristika

- základní vyjadřovací prostředek je UML
- pracuje v iteracích
- definuje obsah každé iterace
- definuje pracovní rámec (framework)

Je RUP agilní metodikou?

Základní charakteristika

- základní vyjadřovací prostředek je UML
- pracuje v iteracích
- definuje obsah každé iterace
- definuje pracovní rámec (framework)

Použití RUP

- klasický heavyweight proces
- agilní proces

Crystal

Kategorie metodik

- metodiky patřící do rodiny *Crystal* jsou přiřazeny do kategorií podle své velikosti
- kategorie metodik se označují barvou (seřazeno od nejméně náročné)
 - *Clear* – C6, D6
 - *Yellow* – C20, D20, E20
 - *Orange* – C40, D40, E40
 - *Red* – C80, D80, E80
 - *Maroon, Blue, Violet*
 - další mohou být přidány

Crystal: Vyhodnocení

Silné stránky

- iterativní inkrementální proces
- časté uvolňování verzí
- proces se *ladí* na základě zpětné vazby
- požadavky se *ladí* během celého vývoje
- zapojení uživatelů
- průběžná integrace

Slabé stránky

- nedefinuje jasný společný proces
- není vhodný pro vysoce kritické projekty
- příliš velká závislost na přímé komunikaci, nepodporuje distribuované týmy

Prediktivní či agilní metodika?

Kdy použít agilní metodiku?

- neurčité nebo měnící se požadavky
- menší nebo neurčitý rozpočet
- odpovědní a dobře motivovaní vývojáři
- menší až středně velký vývojový tým (do cca 80 lidí)
- zákazník, který je ochoten zapojit se do vývoje
- ...

Kdy použít prediktivní metodiku?

- známé a stabilní požadavky
- dostatečný rozpočet
- velký vývojový tým (více jak 100 lidí)
- pevný rozsah projektu
- ...

Metodiky v praxi

- většina metodik může být vytvořena (použita) tak, aby pracovala v nějakém projektu
- libovolná metodika může vést nějaký projekt k neúspěchu
- úspěšné týmy používají inkrementální vývoj
- *heavy* procesy bývají úspěšné
- *light* procesy jsou častěji úspěšné

Je RUP agilní metodikou?

Agile Unified Process

- zjednodušená verze RUP (blíží se UP)
- hlavní modelovací jazyk je UML, ale není omezeno
- modelování bez limitů může ohrozit *agilnost* metodiky
- Ambler, S. W., *The Agile Unified Process (AUP)*., Ambysoft Corp., 2006.
<http://www.ambysoft.com/unifiedprocess/agileUP.html>

dX proces

- minimální RUP proces
- považuje UML za jeden z možných pomocných prostředků

Jména spojená s agilními metodikami

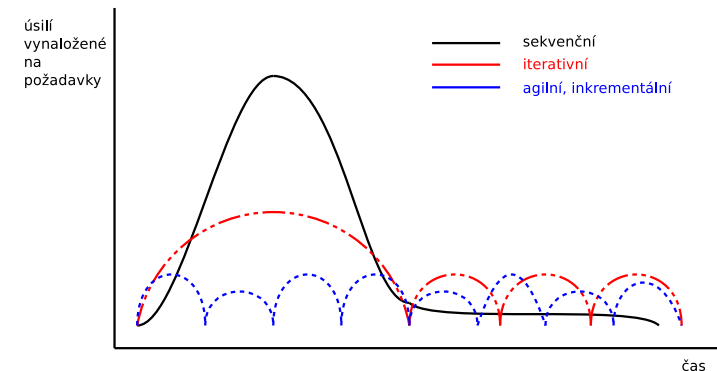
- Ward Cunningham, Kent Beck, Craig Larman, Ron Jeffries, ...

19 / 61

Srovnání přístupů z pohledu požadavků

Jednotlivé typy přístupů k procesu vývoje

- se příliš neliší v celkovém úsilí věnovanému tvorbě požadavků
- se liší v rozložení tohoto úsilí v čase



Převzato z *Software Requirements*, Microsoft Press, 2014

20 / 61

17 / 61

18 / 61

Úvod do managementu SW projektů

Co je to **management**?

Management je proces koordinace činností skupiny lidí, který realizuje jednotlivec nebo skupina lidí za účelem dosažení stanovených cílů.

Tyto cíle se nedají dosáhnout jenom prací jednotlivce.

Management se uskutečňuje v rámci projektů.

Proto se zde soustředíme na management softwarových projektů.

Projekt je ...

... časově ohraničené úsilí, které se vyvíjí s cílem vytvořit jedinečný výsledek (např. výrobek nebo službu).

- **časově ohraničené (úsilí)** – každý projekt má jednoznačný začátek a konec. Konec projektu je dosažen tehdy, když jsou dosaženy stanovené cíle projektu nebo když se ukáže, že těchto cílů dosáhnout nelze.
- **jedinečný (výsledek)** – výsledek projektu se nějak liší od výsledků podobných projektů.

SW inženýr a metodiky

Existuje pro náš projekt 100% správná metodika?

- Ne!

Co musí umět dobrý SW inženýr?

- vybrat vhodnou metodiku nebo
- na základě metodik vytvořit scénář vývoje softwaru tak, aby projekt úspěšně dosáhl stanoveného cíle,
- stanovit cíle splnitelné v daném prostředí
 - cena
 - termín dokončení
 - rozsah
 - kvalita

a to s ohledem na vývojový tým, který má k dispozici.

Jiný úhel pohledu na vývoj softwaru

- **Management SW projektů**
- Řízení kvality softwaru
- Měření v SW inženýrství
- Softwarový tým
- Motivace lidí

Inicializace

- Rozpoznání, že projekt může začít a získání všech relevantních informací potřebných pro plánování projektu, např.
 - časový a cenový horizont
 - základní koncepce projektu
 - potenciální rizika
- trvá několik dní až měsíců
- v některých organizacích je projekt formálně inicializovaný až po ukončení studie vhodnosti, předběžného plánu nebo jiné formy analýzy
- zdroj nebo stimulace inicializace projektu může být:
 - poptávka na trhu
 - požadavek zákazníka
 - z důvodu prestiže
 - výhody technologie
 - požadavky legislativy

Plánování

Vytvoření a udržování plánu pro zabezpečení chodu projektu.

- Definují se požadavky na zdroje, požadavky na práci a definuje se kvalita a kvantita práce.
- Plánování by mělo být tak podrobné, jak je to nezbytné a ne tak, jak je to možné.
- Plánování je intenzivní hlavně v počátečních etapách projektu, v průběhu provádění a řízení se plány upravují podle potřeby.
- Vytvořený plán musí schválit všechny skupiny zapojené do projektu.
- Nedostatky v plánování představují pro projekt značné riziko.

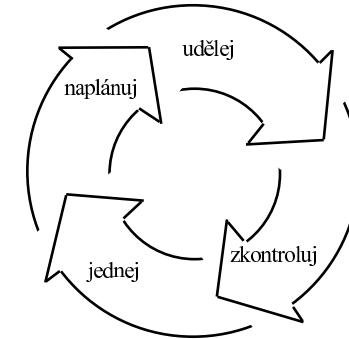
Důvody pro plánování

- snížit neurčitost (výsledku projektu)
- dosáhnout cenovou efektivitu
- zajistit lepší pochopení cílů projektu
- vytvořit základnu pro sledování a řízení práce

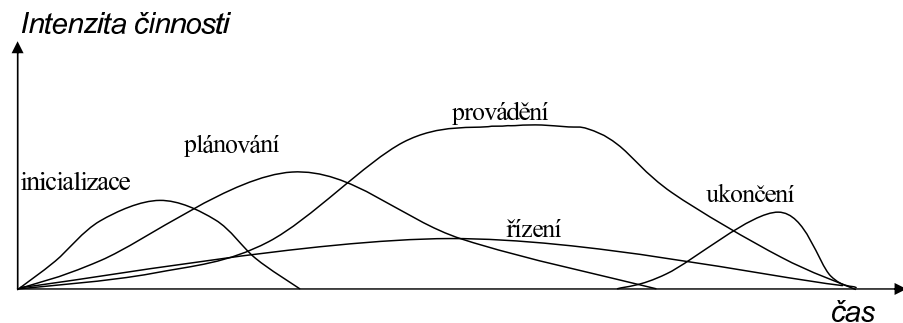
Demingův manažerský cyklus (PDCA)

Manažerské procesy by měly probíhat v této nekonečné smyčce:

- **Plánování (Plan)** – naplánování zamýšleného zlepšení
- **Zavádění (Do)** – realizace plánu
- **Ověření (Check)** – zhodnocení dosažených výsledků
- **Jednání (Act)** – rozhodnutí, jaké další změny provést pro další zlepšení procesu řízení



Procesy managementu projektu



Řízení

- kontrola a řízení na základě naměřených výkonů (na základě výsledků práce a požadavků na změny)
- preventivní činnosti s cílem předcházet problémům
- shromažďují a rozšiřují se informace
 - o stavu projektu (kde se projekt momentálně nachází v porovnání s plánem)
 - o postupu projektu (co se dosud udělalo)
 - o budoucím stavu a postupu projektu (předpověď vývoje projektu)
- sleduje se stav projektu, porovnává a posuzuje se
 - postup dosažený v posledním období (týden, měsíc, ...)
 - a v obdobích předcházejících
 - dosažené výsledky s tím, co je ještě potřeba udělat
 - odhady a skutečné hodnoty; předpověď budoucích hodnot
 - přiřazení zdrojů (lidé, počítače, ...)
 - poměr dosažených výsledků a času, který uplynul; rozhoduje se, či je dosažení cílů projektu reálné
- zajišťuje se řízení změn

Provádění

- spotřebuje nejvíce času (úsilí) a peněz u SW projektů jenom při špatném plánu
- realizace plánu projektu
- manažer projektu koordinuje a usměrňuje provádění činností z plánu
 - přidělování úkolů
 - stanovení priorit
 - rozdělování pravomocí
 - sledování postupu prací na projektu
 - rozhodování o umístění důležitých zdrojů
- provádění projektu nejvíce ovlivňuje problémová doména (oblast), technické parametry řešení a model vývoje
- vytváří se výsledek (výrobek, služba)

Projektový plán

Projektový plán obsahuje:

- definici cílů, úloh a odpovědností
co je potřeba udělat,
pro koho je to potřeba udělat
- definici požadavků na zdroje
kdo to má udělat
kolik to bude stát
- techniky, prostředky, zdroje pro vykonávání plánu
jak to udělat
- kontrolní body
kdy je to potřeba udělat
- definici kvality, základ pro měření postupu projektu
- stanovení rizik projektu
co když nastane určitá situace

31 / 61

29 / 61

Plánování

Plánování staví mosty mezi tím,
kde jsme, a kam chceme jít.

H. Koontz, H. Weihrich

32 / 61

30 / 61

Řízení kvality SW projektů

Obvyklý postup při tvorbě softwaru spočívá v

- co nejrychlejší **implementaci** programu a
- rychlém **testování** s cílem najít a
- **odstranit chyby** a nedostatky.

V žádném jiném technickém oboru se nevytvářejí výrobky **nekontrolovatelné kvality**, přičemž by se spoléhalo na **testování**.

Kvalita

- *The totality of features and characteristics of a product or service that bear on its ability to meet stated or implied needs.* (ISO 8402-1986)
- Souhrn vlastností nebo charakteristik produktu či služby, které souvisí s jeho či její schopností splnit explicitně uvedené či implicitně předpokládané potřeby.
- Kvalita není definovaná jako absolutní míra, ale jako stupeň splnění požadavků či potřeb.
- Kvalita je . . .
 - míra stupně dokonalosti (Oxfordský slovník)
 - splnění požadavků (Crosby)
 - vhodnost k danému účelu (ISO 9001)
 - schopnost produktu nebo služby plnit dané potřeby (BS 4778)

35 / 61

Ukončení

- zaznamenají se nové poznatky, zkušenosti a poučení pro budoucí projekty
- ukončí se kontrakty s dodavateli a dodávky potřebných výrobků a služeb v rámci projektu, vyřeší se také všechny otevřené problémy (závazky a pohledávky) spojené s dodávkami

33 / 61

Jiný úhel pohledu na vývoj softwaru

- *Management SW projektů*
- **Řízení kvality softwaru**
- Měření v SW inženýrství
- Softwarový tým
- Motivace lidí

36 / 61

34 / 61

Parametry softwarového projektu

V každém projektu existují čtyři základní parametry

- **cena**
 - nízké náklady: klesá kvalita i rozsah
 - příliš nízké náklady: zadání nelze splnit
 - příliš vysoké náklady: kvalita se nezvýší, čas se nesníží
- **čas**
 - málo času: snižuje se kvalita a rozsah zadání, roste cena
 - příliš mnoho času: oddaluje se zpětná vazba od systému v provozu
- **kvalita**
 - vyšší kvalita: zvyšuje počáteční náklady, do budoucna je snižuje
 - nízká kvalita: nízké počáteční náklady, do budoucna zvyšuje enormě náklady (finanční i lidské)
- **rozsah**
 - menší rozsah zadání: možnost vyvíjet rychle, levně a s lepší kvalitou

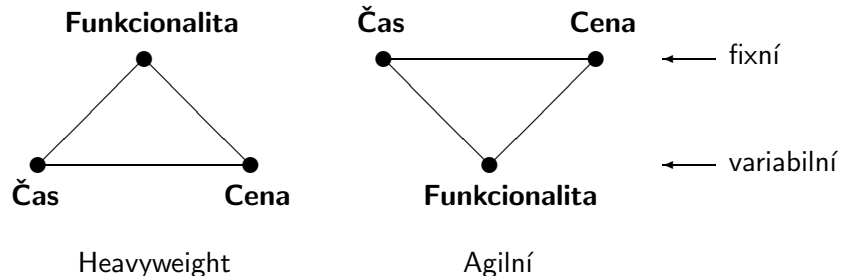
39 / 61

Parametry softwarového projektu

- žádný účastník nemůže definovat *všechny* parametry
- pro heavyweight metodiky platí, že zákazník často volí rozsah a případně kvalitu, vývojový tým pak určí cenu a čas
- pro agilní metodiky platí, že zákazník často volí cenu, čas a kvalitu, vývojový tým pak určí rozsah

Alternativní pohled ($funkcionalita = rozsah + kvalita$)

- čas, cena a funkcionalita (rozsah + kvalita)



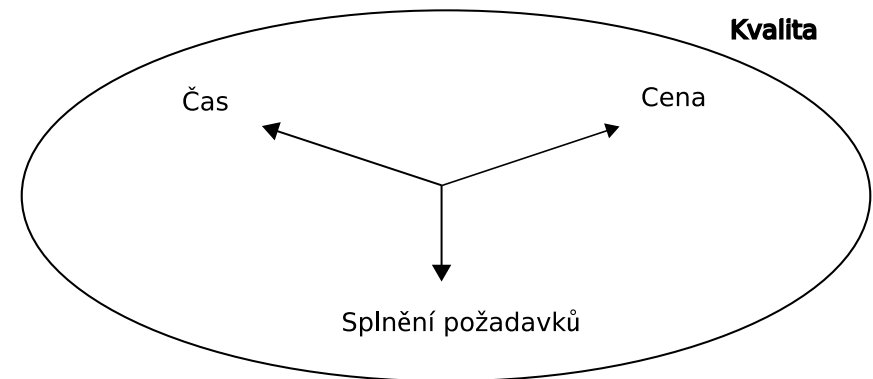
40 / 61

Různé pohledy na kvalitu

- Z pohledu **uživatele** může kvalita odpovídat jednoduchosti obsluhy systému, spolehlivému a efektivnímu provádění jednotlivých funkcí systému.
- Z pohledu **provozu** systému kvalita obvykle znamená dobrou provozní dokumentaci a efektivní využití výpočetních prostředků.
- Z pohledu **tvůrce a údržbáře** systému kvalita odpovídá čitelným a modifikovatelným programům a srozumitelné a přesné dokumentaci.
- Z pohledu **manažera** kvalita obvykle odpovídá dodání výrobku včas, v rámci rozpočtu a dohodnutých požadavků.

37 / 61

Kvalita SW produktu



38 / 61

ISO 9000

- ISO – Mezinárodní organizace pro normy (International Standards Organisation), viz <http://www.iso.ch/>
- ISO 9000 – soustava norem pro řízení a zajištění kvality (1979, 1987, 1994, 2000, 2008)
- slučuje standardy 9001, 9002, 9003 (rok 2000) a 9004 (rok 2008)
- mezinárodní měřítko kvality
- Zavádí zpětnou vazbu do business procesů.
- Umožňuje nezávislé posuzování kvality třetí stranou.
- Vychází z vnitřních norem britského ministerstva obrany pro muniční závody z druhé světové války.
- Primárně byly tyto normy vytvořeny pro hromadnou výrobu (ne pro softwarový průmysl). Obsahují však všeobecné požadavky platné pro libovolnou oblast výroby.
- ISO 9000 definuje body, které musí systém výroby splňovat, aby vyhověl této normě.

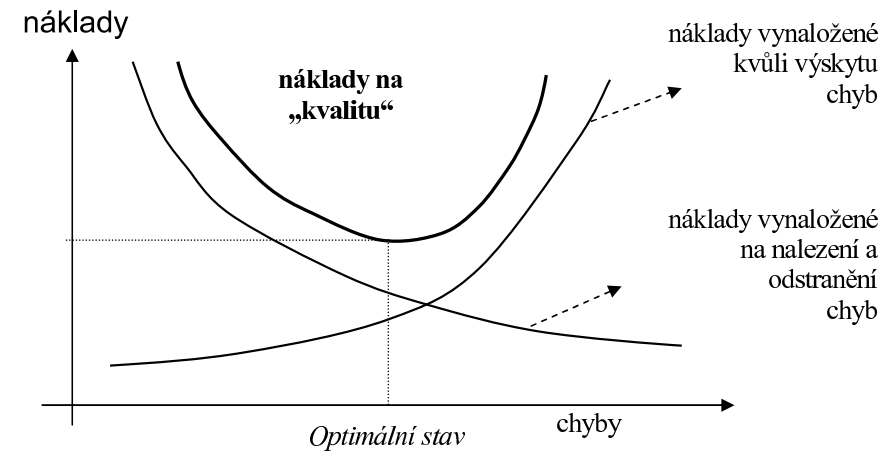
Zhodnocení ISO 9000

- + Dává slušný základ pro dobře fungující výrobu, který se dá dále rozvíjet.
- + Zákazník získá určitou představu o organizaci.
- + Zvyšuje konkurenceschopnost a zlepšuje jméno organizace (dokud certifikát nezískají všichni).
- Snadno sklouzne k byrokratickému přístupu.
- Důraz je kladen na kontrolu a ne na samotnou kvalitu.
- Zavedení ISO 9000 je poměrně vysoká investice. Zatím neexistuje přizpůsobení normy malým organizacím.
- Poměrně dlouhé zavádění normy (zisk certifikátu).
- Norma nestanovuje vhodný postup pro jejich zavedení.

Funkcionalita, chyby a kvalita

Nepleťte si kvalitu s funkcionalitou!

Systém s řadou funkcí může mít nízkou kvalitu (např. příliš chyb) a naopak. software bez chyb \neq kvalitní software



43 / 61

Normy pro systém zajištění kvality

Pro softwarové produkty se vychází z následujícího předpokladu:

Pokud má organizace *kvalitní* proces tvorby výrobku (softwaru), budou i její výrobky kvalitní.

Tento přístup se používá hlavně proto, že v softwarovém inženýrství není jednoduché měřit kvalitu programů pomocí nějaké výstupní kontroly.

kvalitní proces \Rightarrow kvalitní výrobek

41 / 61

Capability Maturity Model – CMM

Úrovně

- 0 *Neexistující řízení. Procesy a jejich řízení je zcela chaotické.*
- 1 Počáteční (Initial). Procesy jsou realizovány ad hoc, organizace je schopna použít nové či nedokumentované procesy.
- 2 Opakovatelné (Repeatable). Procesy jsou dostatečně dokumentované a umožňují opakování stejných kroků.
- 3 Definované (Defined). Procesy jsou definovány a potvrzeny jako standardní procesy.
- 4 Řízené (Managed). Procesy jsou vyhodnocovány na základě předem stanovených metrik.
- 5 Optimalizující (Optimizing). Řízení procesů zahrnuje i inovační cyklus, lze optimalizovat a zlepšovat procesy.

47 / 61

Normy pro systém zajištění kvality

Capability Maturity Model Integration (CMMI)

- Řeší problém s nasazením CMM pro tvorbu softwaru – Aplikace různorodých modelů, které nejsou integrální součástí procesů vývoje softwaru, zvyšují náklady spojené se školením, posuzováním apod.
- Určený pro vývojové týmy
- CMM se stává obecným teoretickým modelem

ISO/IEC 15504

- IEC – International Electrotechnical Commission
- Soustava technických standardů pro vývoj počítačového softwaru
- Odvozen od ISO/IEC 12207 a modelů zralosti CMM, Bootstrap a Trillium
- Další související normy: ISO/IEC 33001 (rodina norem 330xx má nahradit 155xx)
- *Není dostupné zdarma*

48 / 61

Capability Maturity Model – CMM

Účel

- vyhodnocení schopnosti (U.S.) vládních dodavatelů splnit softwarové projekty
- zaměřuje se na *procesy*
- Humphrey, W. S. *Managing the Software Process*. SEI series in software engineering. Addison-Wesley. 1989
- *Capability Maturity Model for Software*. Technická zpráva. Software Engineering Institute, Carnegie Mellon University. 1993

Maturity Model

- nástroje popisující, jak dobře jsou nastaveny praktiky, procesy a chování organizace; jak kvalitně mohou dosáhnout požadovaných výstupů
- lze použít jako prostředek pro srovnání procesů různých organizací a pro porozumění těmto procesům

45 / 61

Capability Maturity Model – CMM

Struktura modelu

- Úrovně zralosti (*Maturity Levels*) – 5 úrovní, nejvyšší stupeň reprezentuje ideální stav
- Klíčové oblasti (*Key Process Areas*) – soubory souvisejících aktivit pro dosažení stanovených cílů
- Cíle (*Goals*) – cíle definují rozsah, omezení a záměry klíčových oblastí
- Vlastnosti (*Common Features*) – praktiky pro začlenění klíčových oblastí do procesů organizace
- Klíčové praktiky (*Key Practices*) – popisují praktiky a elementy infrastruktury, které přispívají k efektivnímu začlenění oblastí

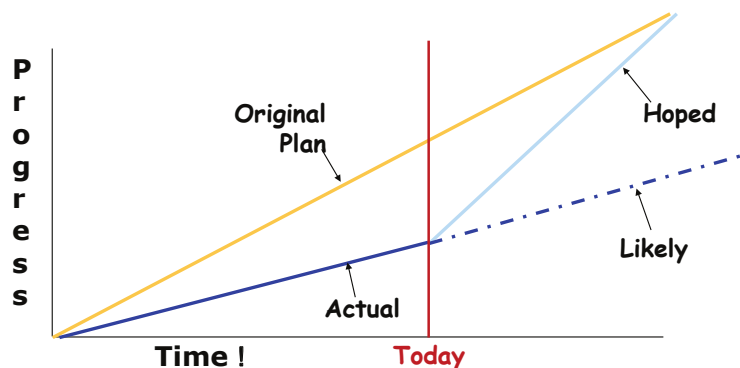
46 / 61

Měření v softwarovém inženýrství



Graphs Can Be Very Useful

Very often, a picture is much easier to interpret than raw data



Copyright 2018, Dennis J. Frailey

Fundamentals of Measurement and Data Analysis for Software Engineers
Part 4 - Analysis Techniques Especially Suited to Software

Jiný úhel pohledu na vývoj softwaru

- Management SW projektů
- Řízení kvality softwaru
- Měření v SW inženýrství
- Softwarový tým
- Motivace lidí

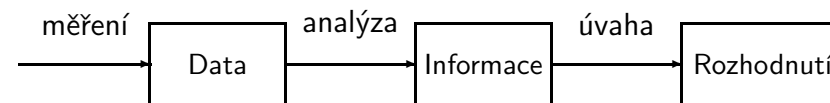
51 / 61

Měření v softwarovém inženýrství

- Typy měření
 - **Přímé měření:** přímé získání hodnoty sledovaného atributu např. počet řádků programu
 - **Nepřímé měření:** odvození z jiných atributů, které lze měřit přímo např. udržitelnost lze určit jako čas potřebný pro odstranění chyby
 - jsou možné různé interpretace naměřených hodnot např. počet chyb nalezených za jednotku času reprezentuje kvalitu testování nebo (ne)spolehlivost programu
- Pro úspěch projektu je důležitá dohoda na kritériích přijetí projektu. Musejí být měřitelná.
 - Uživatelské rozhraní musí být přátelské.
 - + Operátoři musí být schopni začít pracovat s libovolnou funkcí programu do 30 sekund od usednutí k terminálu.

Měření v softwarovém inženýrství

- Měření je proces přiřazování hodnot k vlastnostem entit reálného světa.
- Měření zvyšuje pravděpodobnost, že i přes nejistotu uděláme **dobré rozhodnutí**.
- Každé měření musí mít svůj účel (cíl).



Bez měření nelze řídit.
ale
Dostaneme, co měříme.

52 / 61

49 / 61

50 / 61

Metriky pro výrobek – pokračování

- složitost
 - počet souborů
 - velikost programu (počet příkazů, řádků, ...)
 - počet větvení (příkazy IF)
 - hloubka zanoření řídicích struktur
 - počet cyklů
 - průměrná délka věty v dokumentaci
- chyby
 - počet chyb a nedostatků
 - chybovost = počet chyb / kLOC
 - klasifikace chyb a nedostatků a frekvence jejich výskytu
- udržitelnost
 - střední doba potřebná na opravu chyby
 - střední doba potřebná na pochopení logiky modulu
 - střední doba na nalezení příslušné informace v dokumentaci

Další používané metriky

Metriky pro proces

- úsilí – čas vynaložený na vývoj systému (člověko-měsíce)
- změny požadavků (odráží kvalitu specifikace požadavků)
 - počet změn požadavků
 - střední doba od dokončení specifikace do požadavku na změnu
- náklady a čas
 - začátky a konce činností
 - délka trvání činností
 - náklady na provedení jednotlivých činností

Metriky pro zdroje

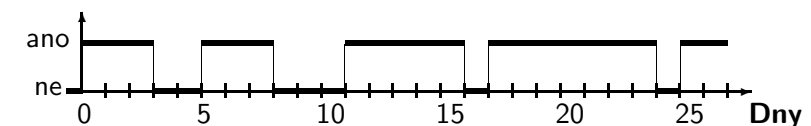
- charakteristiky personálu
 - produktivita
 - velikost týmu
 - rozsah a způsob komunikace
 - zkušenosti
- zatížení sítě, ...

Metriky pro výrobek

- velikost, rozsah (pro odhad času a nákladů a měření produktivity)
 - počet řádků zdrojového textu programu (LOC)
 - počet modulů
 - průměrný počet LOC na modul
 - rozsah dokumentace (počet stran)
- modularita
 - svázanost modulů (počet toků údajů a řízení mezi moduly a počet globálních datových struktur)
 - soudržnost modulů
- spolehlivost
 - střední doba mezi výpadky systému (MTBF)
 $MTBF = MTTF + MTTR$
MTTF – střední doba do následujícího výpadku
MTTR – střední doba opravy
- dostupnost
 - pravděpodobnost, že v daném čase program pracuje správně
 $dostupnost = 100 * MTTF / MTBF [\%]$

Určení spolehlivosti a dostupnosti

Funkce



$$MTTF = (3 + 3 + 5 + 7) / 4 = 4,5 \text{ dne}$$

$$MTTR = (2 + 3 + 1 + 1) / 4 = 1,75 \text{ dne}$$

$$MTBF = MTTF + MTTR = 25 / 4 = 6,25 \text{ dne}$$

$$dostupnost = MTTF / MTBF = 18 / 25 = 72 \%$$

Poznámka: Plánované odstávky z důvodu údržby se do výpadků nepočítají.

Studijní koutek – Důvody ukončení studia

Studijní důvody

- alespoň 15 kreditů v 1. semestru studia
- alespoň 30 kreditů za každý rok studia nebo nejméně polovina zapsaných kreditů
- opakovaný povinný předmět
- státní závěrečná zkouška
Státní závěrečnou zkoušku nebo kteroukoli její část lze jednou opakovat.
- překročení maximální doby studia (SZŘ VUT, čl. 4)
- (opakovaná neomluvená neúčast v kontrolované výuce)

Kázeňské důvody

- vyloučení ze studia za závažný nebo opakovaný disciplinární přestupek

Formální důvody

- nezapsání se do dalšího ročníku

Zanechání studia písemným oznámením

59 / 61

Studijní koutek – Poplatek za studium

§ 58 odst. 3 Zákona č. 111/1998 O Vysokých školách (...)

- *Studuje-li student ve studijním programu déle, než je standardní doba studia zvětšená o jeden rok v bakalářském nebo magisterském studijním programu, stanoví mu veřejná vysoká škola poplatek za studium, který činí za každých dalších započatých šest měsíců studia nejméně jedenapůlnásobek základu; do doby studia se započtou též doby všech předchozích studií v bakalářských a magisterských studijních programech, které byly ukončeny jinak než řádně podle § 45 odst. 3 nebo § 46 odst. 3, nejde-li o předchozí studium, po jehož ukončení student řádně ukončil studijní program stejného typu. Období, ve kterém student studoval v takovýchto studijních programech, nebo v takovýchto studijních programech a v aktuálním studijním programu souběžně, se do doby studia započítávají pouze jednou. Od celkové doby studia vypočtené podle tohoto odstavce se však nejdříve odečte uznaná doba rodičovství.*

Příloha č. 4, článek 2, odst. 2 Statutu VUT

- *Výše poplatků za prodlouženou dobu studia za každých započatých 6 měsíců studia činí: a) trojnásobek základu v prvním roce, b) šestinásobek základu ve druhém roce, c) dvanáctinásobek základu ve třetím a dalších akademických rocích.*

60 / 61

Zkouška – Variantní termíny

- 1. termín: **čtvrtek 2. 1. 2020, 12:00** (výsledky do 9. 1. 2020)
 - **270 míst** (D105, D0206, D0207, E112)
- 2. termín: **pátek 10. 1. 2020, 9:00** (výsledky do 17. 1. 2020)
 - **270 míst** (D105, D0206, D0207, E112)
- 3. termín: **pondělí 20. 1. 2020, 15:00** (výsledky do 27. 1. 2020)
 - **180 míst** (D105, D0206)
- 4. termín: **čtvrtek 23. 1. 2020, 15:00** (výsledky do 30. 1. 2020)
 - **180 míst** (D105, D0206)
- 5. termín: **pátek 31. 1. 2020, 16:00** (výsledky do 7. 2. 2020)
 - **326 míst** (D105, D0206, D0207, E112, E104, E105)
- Celkem vysláno **1 226** míst pro **775** zapsaných studentů.
- Na studenta je **1,58** místa (tedy více než minimum 1,50).

57 / 61

Zkouška – Průběh

- Přihlašování na termíny zkoušky
 - student v IS FIT (přihlašování začíná 18. 12. 2019 ve 20:04)
 - podmínkou je získání zápočtu
 - lze se přihlásit nejvýše na jeden termín současně
 - na další termín až po získání neuspokojivého výsledku
 - přihlašování/odhlašování končí **24 hodin** před zkouškou
- Rozsazení do místností (ob 2): v IS FIT
- Nutné: průkaz studenta, propisovací tužka
- Povoleno: nealkoholické nápoje a klasické hodinky
- Zakázáno: vše ostatní
- Čistá doba trvání: **90 minut**
- Maximální počet bodů: **60**
- Minimum bodů pro průchod: **30**

58 / 61

Studijní koutek – Poplatek za studium

Výše základu vyhlášeného MŠMT pro akademický rok 2019/2020

- 4.090 Kč

Výše poplatku za každých započatých 6 měsíců studia po dobu 12 měsíců pro akademický rok 2019/20:

- 12.300 Kč pokud studium přesahuje standardní dobu zvětšenou o 1 rok
- 24.600 Kč pokud studium přesahuje standardní dobu zvětšenou o 2 roky
- 49.200 Kč pokud studium přesahuje standardní dobu zvětšenou o 3 a více let

Vizte Přílohu č. 1 Rozhodnutí rektora č. 6/2017.