

# Úvod do softwarového inženýrství

## IUS 2024/2025

### 9. přednáška

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

15. a 18. listopadu 2024

# Organizace následujících přednášek

- **10. přednáška – Ochrana intelektuálního vlastnictví, etický kodex**
  - pondělí 25. 11. 2024
  - pátek 22. 11. 2024
- **11. přednáška – Provoz a servis IT**
  - **Ing. Stanislav Vaněk, Innogy (dříve Kydryl/IBM)**
  - pondělí 2. 12. 2024
  - pátek 6. 12. 2024
- **12. přednáška – Řízení softwarových projektů**
  - **Ing. Dana Brhelová, Artysis, s.r.o.**
  - pondělí 9. 12. 2024
  - pátek 29. 11. 2024

# 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

## 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.*

# 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

# 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*)

# 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 podmnožina *Product backlog*
- jednotlivé položky jsou rozděleny mezi týmy

# 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 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 Proces: Post-Game

## Popis fáze

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

# Scrum: Vyhodnocení

## Silné stránky

- iterativní inkrementální proces
- časté uvolňování verzí (inkrementů)
- architektura systému je navržena před procesem vývoje
- požadavky se *ladí* během celého vývoje
- sledovatelnost požadavků (*Product Backlogs*)
- 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

# DevOps: Development + Operations

## Hlavní myšlenka

- propojení týmů pro vývoj a pro provoz softwaru  
Odstrašující příklad: Nemůže za to můj kód, ale tvůj stroj.

## Nejdůležitější vlastnosti

- velice časté uvolňování a nasazování nových verzí
- nezbytná nástrojová podpora pro dosažení vysoké míry automatizace
- nejenom průběžná integrace ale i průběžné nasazování

# 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

# 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

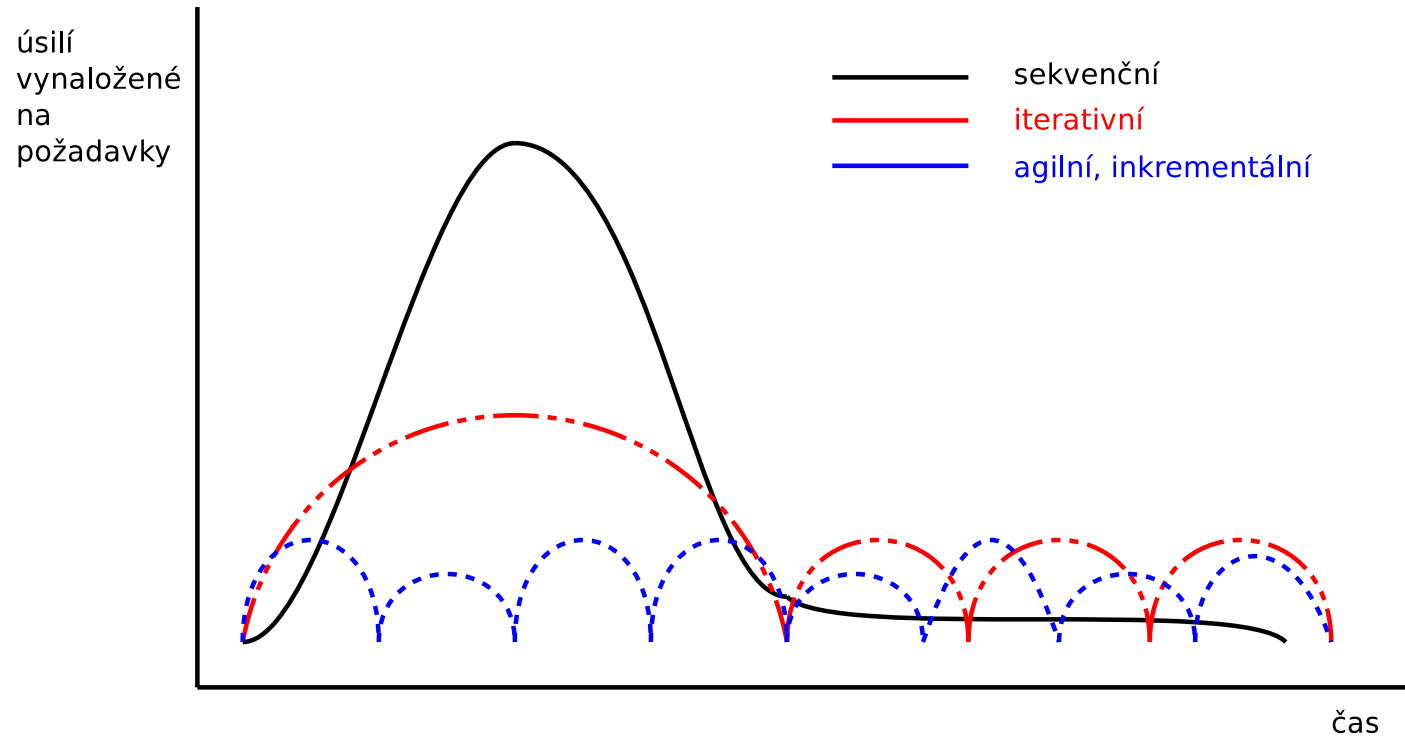
## Jména spojená s agilními metodikami

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

# Srovnání přístupů z pohledu požadavků

Jednotlivé typy přístupů k procesu vývoje

- se příliš neliší v celkovém úsilí věnovanému tvorbě požadavků,
- liší se však v rozložení tohoto úsilí v čase
- a tedy v možnostech provádět průběžné úpravy specifikace



Převzato z *Software Requirements*, Microsoft Press, 2014



# 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

# 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

# Úvod do managementu SW projektů

## Co je to management?

Definice managementu na skoušce nebude

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 ...

Definice projektu na skoušce taky nebude

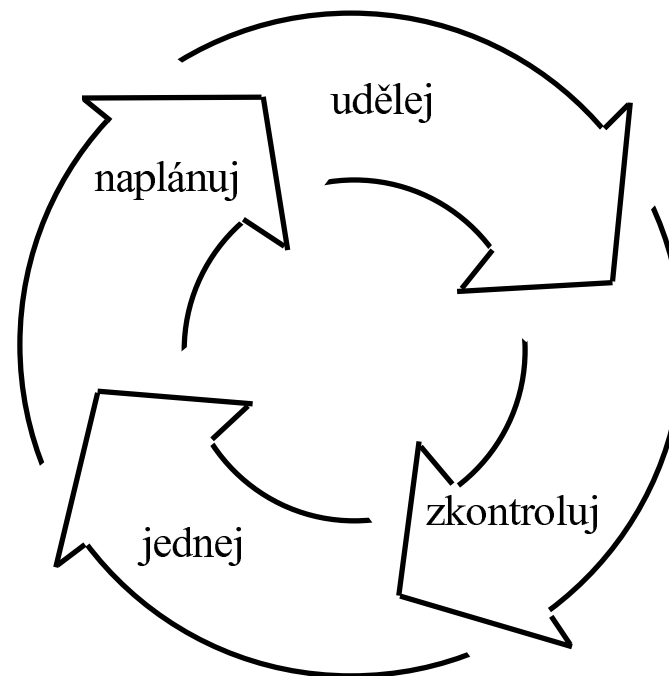
... časově ohraničené úsilí, které se vyvíjí s cílem vytvořit jedinečný výsledek (např. výrobek nebo službu).

- **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.

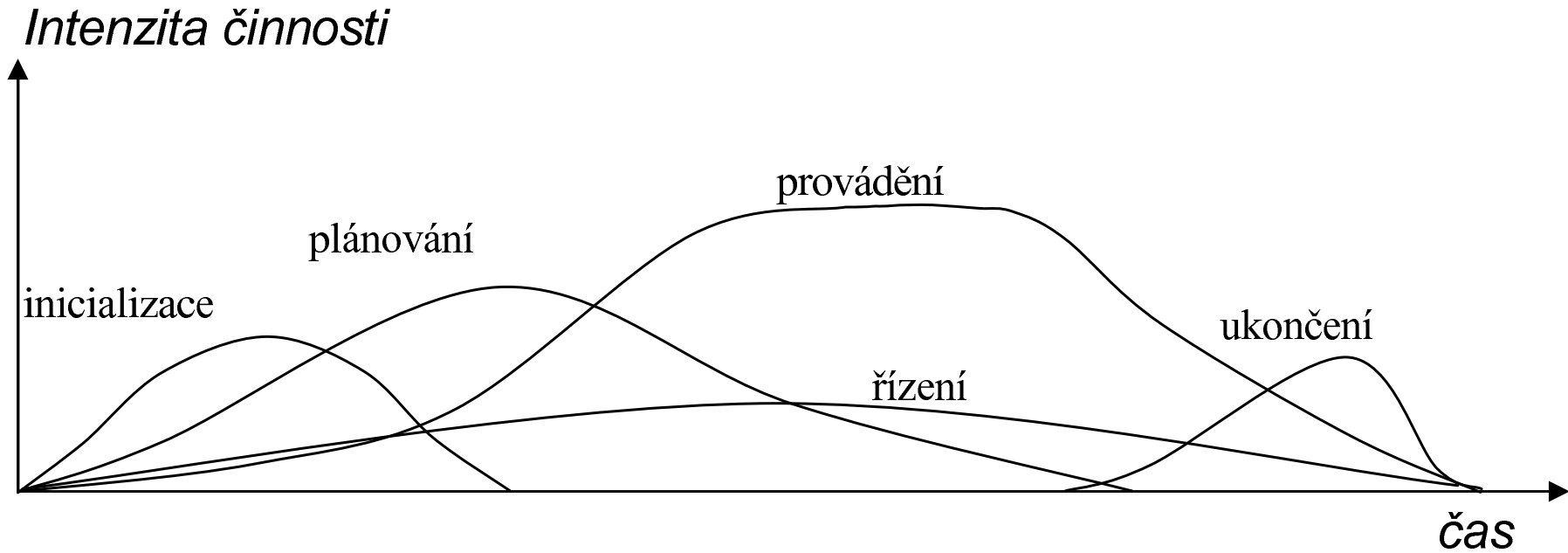
# Demingův manažerský cyklus (PDCA)

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





# 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

# 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

# Plánování

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

H. Koontz, H. Weihrich

# 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)

# 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

# Jiný úhel pohledu na vývoj softwaru

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



# 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**.

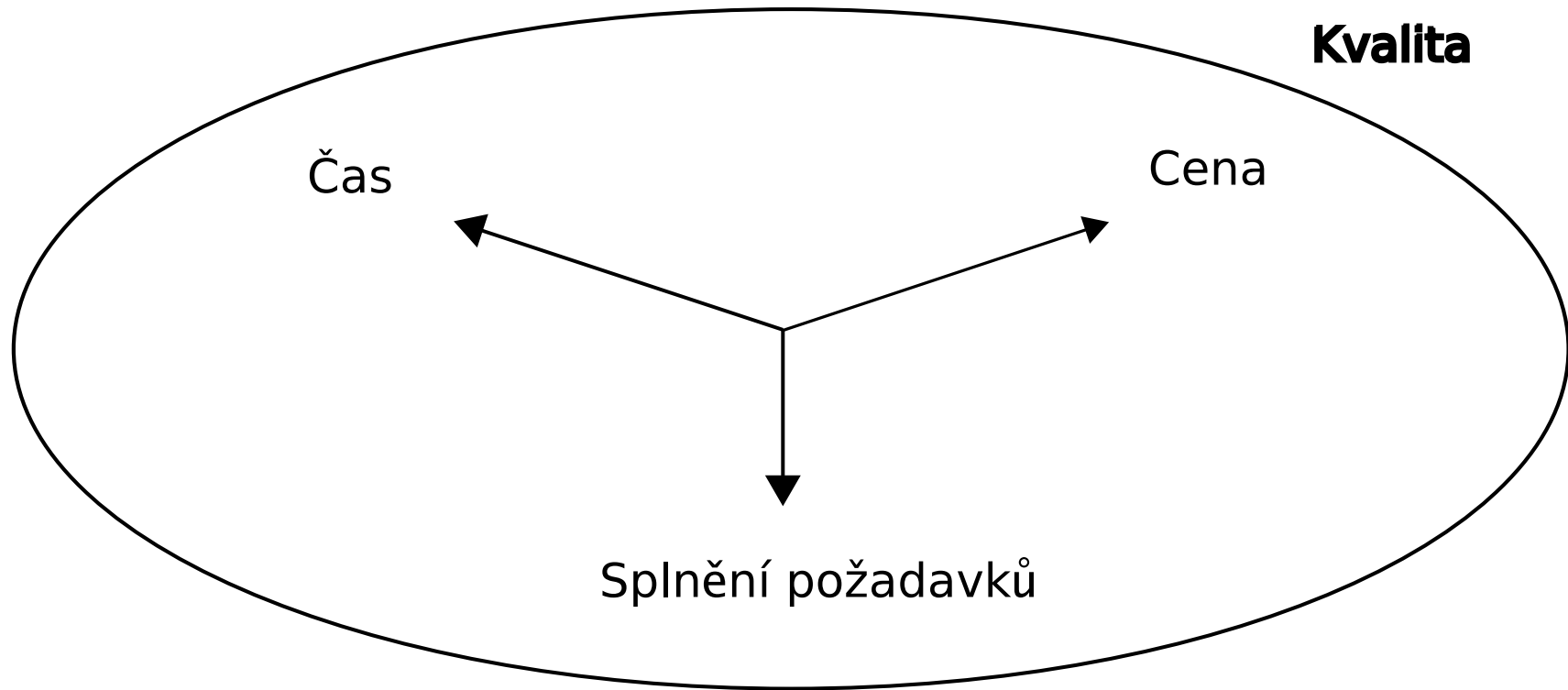
# Kvalita

- *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)

# 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.

# Kvalita SW produktu



# 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**

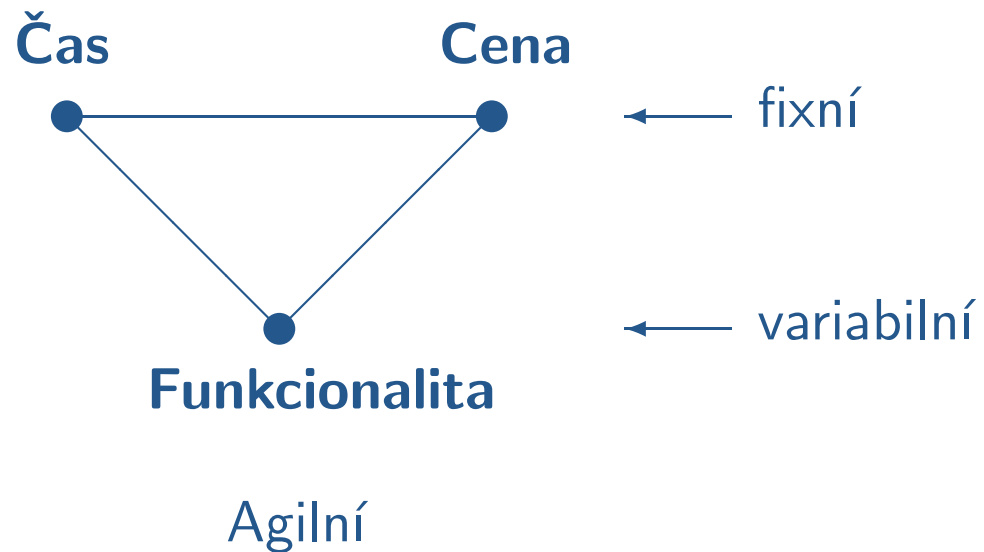
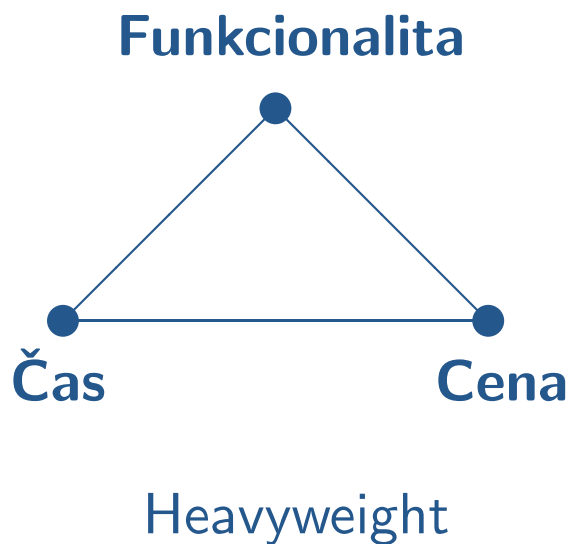
- menší rozsah zadání: možnost vyvíjet rychle, levně a s lepší kvalitou

# 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)

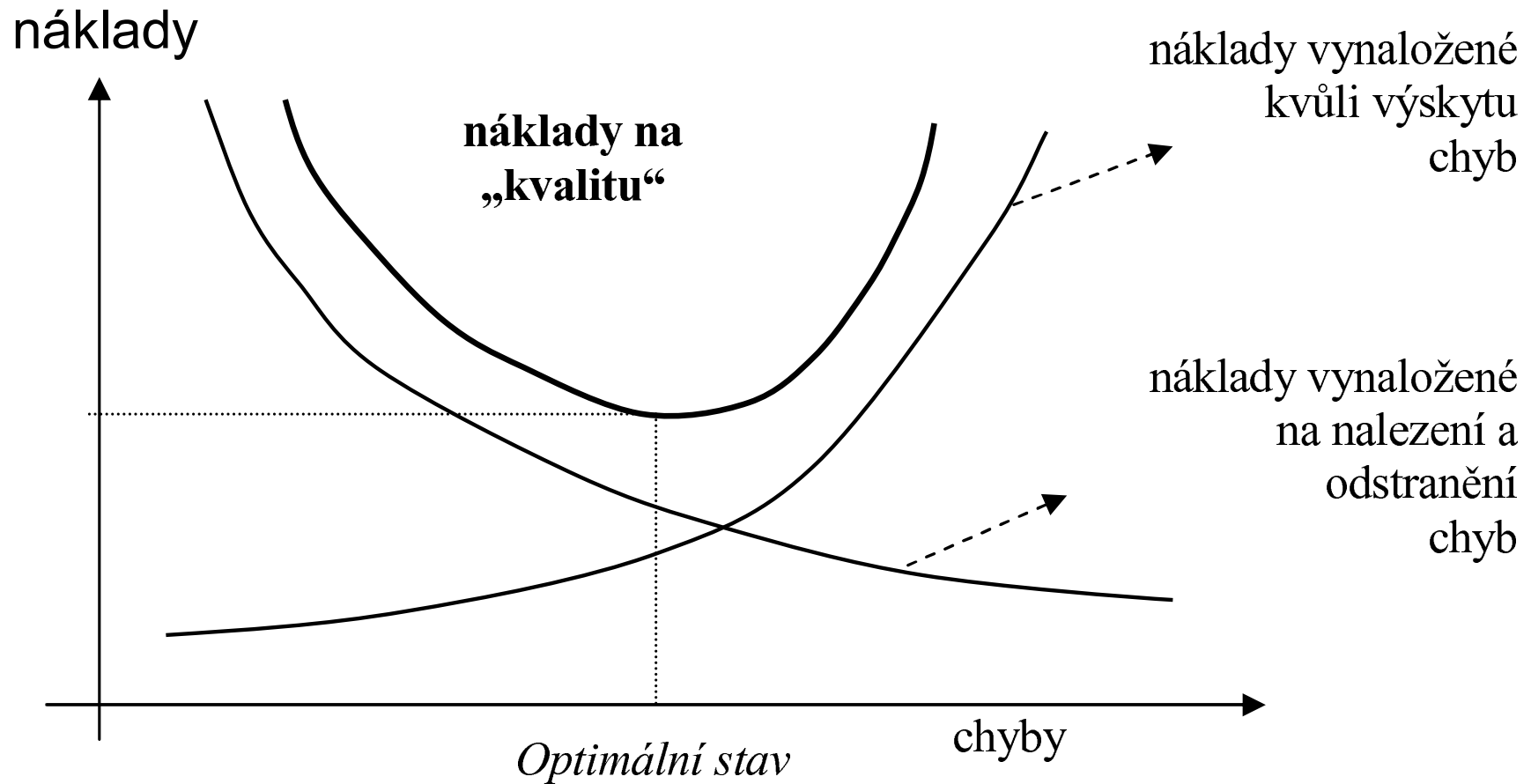


# Funkcionalita, chyby a kvalita

Nepleťte si kvalitu s funkcionalitou!

System s řadou funkcí může mít nízkou kvalitu (např. příliš chyb) a naopak.

software bez chyb  $\neq$  kvalitní software



# 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



# 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. Lidé díky definovaným procesům ví, co a jak mají dělat.
- + Zákazník získá určitou představu o organizaci.
- + Zvyšuje konkurenceschopnost a zlepšuje jméno organizace (dokud certifikát nezískají všichni).
- Soustředí se na procesy a jejich kontrolu a ne na samotnou kvalitu.
- Snadno sklouzne k byrokratickému přístupu.
- Zavedení ISO 9000 je poměrně vysoká investice. Zejména malým organizacím se tak nemusí vyplatit.
- Poměrně dlouhé zavádění normy (zisk certifikátu).
- Norma nestanovuje vhodný postup pro její zavedení.

# 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

# 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

# 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.

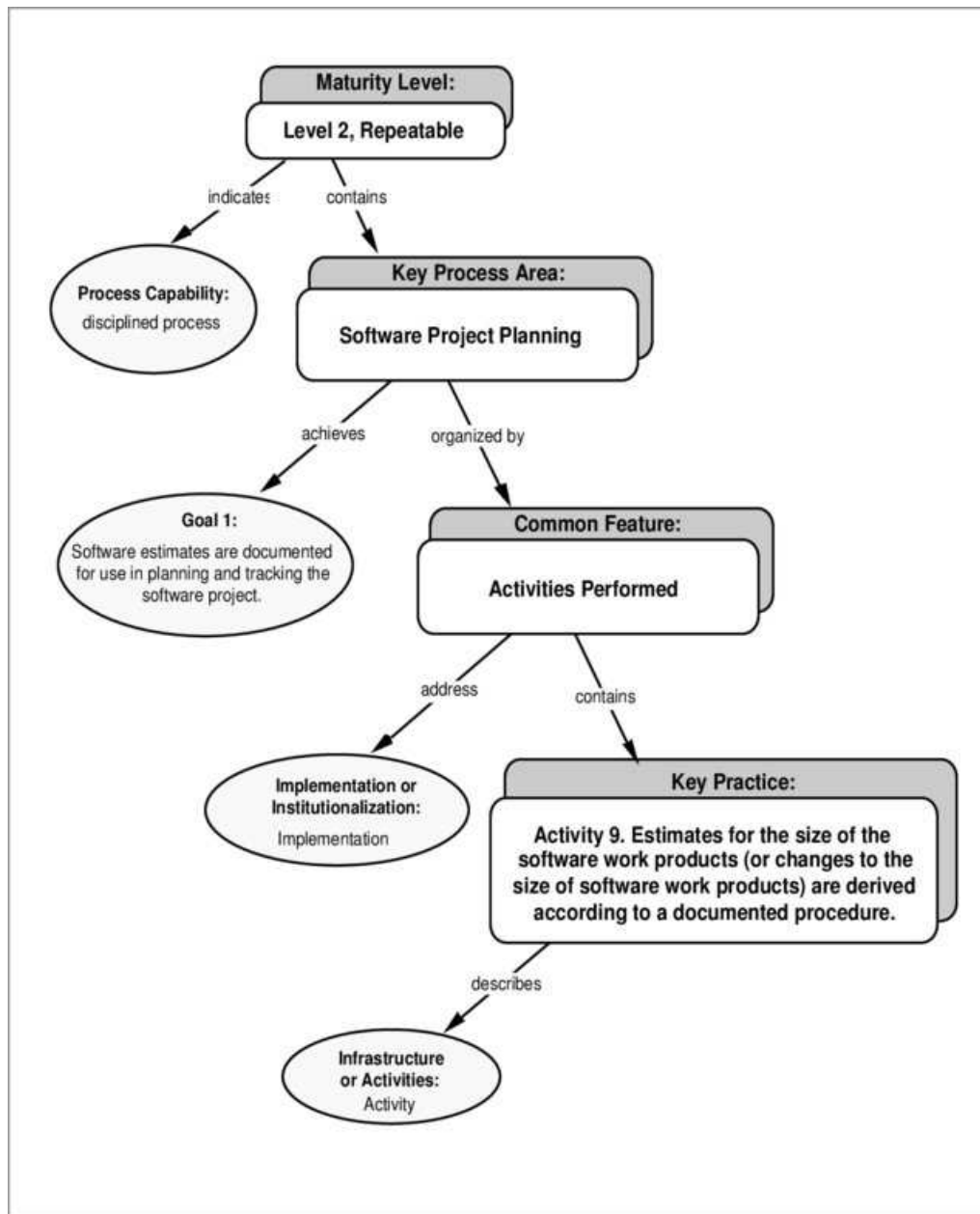
# Capability Maturity Model – CMM

Příklad klíčových oblastí a určení úrovně zralosti.

Levels	Key Process Areas
5 – Optimizing	Technology Change Management Process Change Management
4 – Managed	Quantitative Process Management Software Quality Management
3 – Defined	Integrated Software Management Organization Process Definition Training Program Peer Review
2 – Repeatable	Requirements Management Software Project Planning Software Quality Assurance

Zdroj: Capability Maturity Model, Version 1.1, IEEE Software 10(4):18-27

# CMM – Příklad oblasti



# 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
- *Různé deriváty pro různé oblasti použití* – vývoj, služby, řízení dodavatelského řetězce (outsourcing, akvizice, ...)

## ISO/IEC 33001

- IEC – International Electrotechnical Commission
- Soustava technických standardů pro vývoj počítačového softwaru
- Rodina norem 330xx nahrazuje předchozí rodinu 155xx (**ISO/IEC 15504**), která je odvozena od ISO/IEC 12207 a modelů zralosti CMM, Bootstrap a Trillium
- *Není dostupné zdarma*

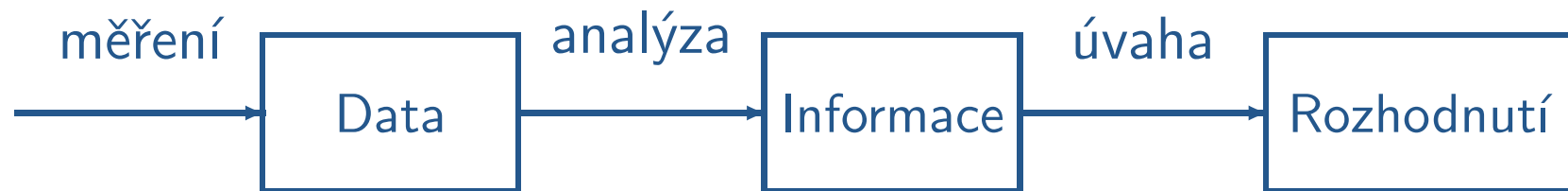


# Jiný úhel pohledu na vývoj softwaru

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

# 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

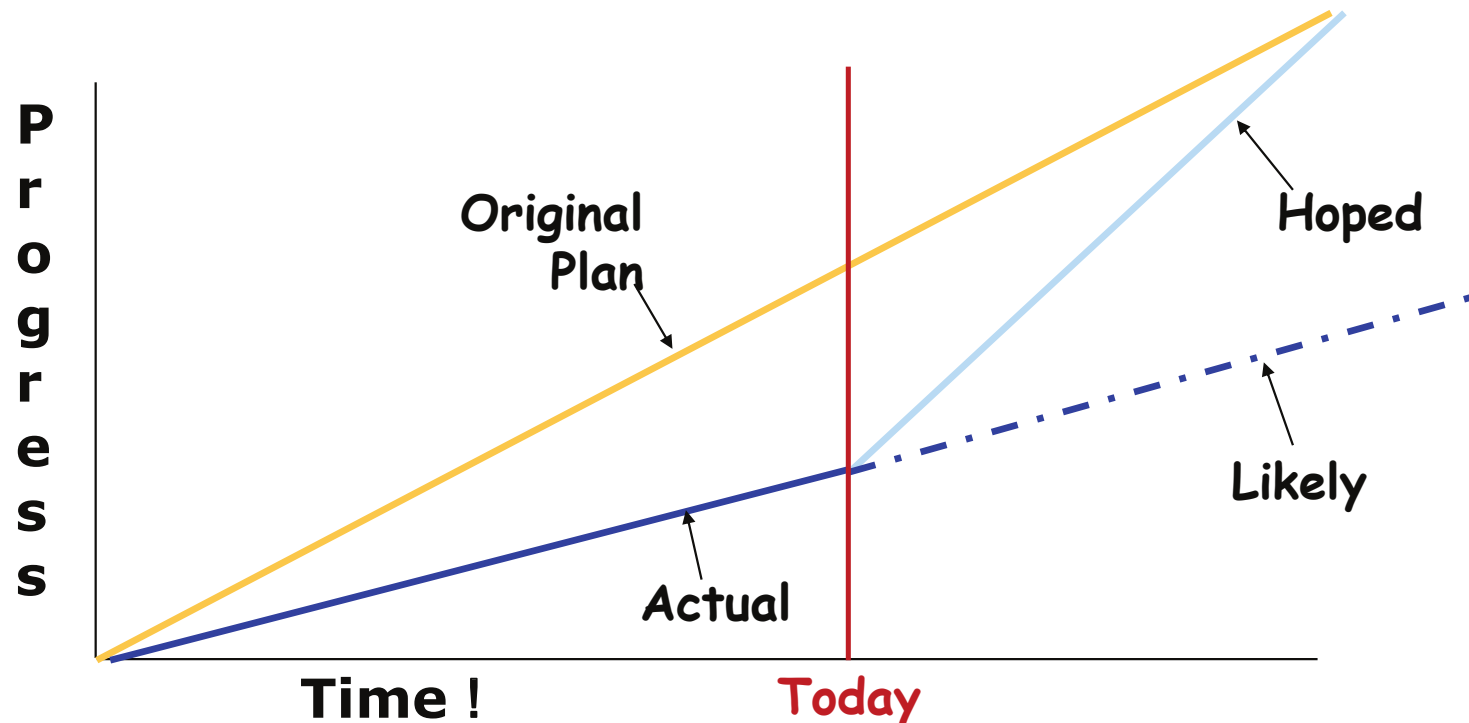
Dostaneme, co měříme.

# 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

# Měření v softwarovém inženýrství

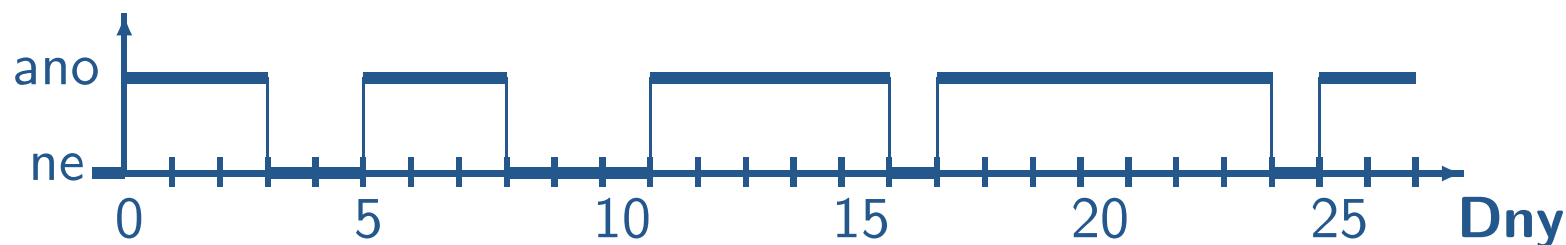
- Typy měření
  - **Přímé měření:** přímé získání hodnoty sledovaného atributu  
např. počet řádků programu
  - **Nepřímé měření:** odvození z jiných atributů, které lze měřit přímo  
např. udržitelnost lze určit jako čas potřebný pro odstranění chyby
  - jsou možné různé interpretace naměřených hodnot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

# 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



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

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

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

$$\text{dostupnost} = \text{MTTF} / \text{MTBF} = 18 / 25 = 72\%$$

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

# 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
- udržovatelnost
  - 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, ...



# Studijní koutek – IT a studium (1/4)

## Naše postřehy

- Úroveň přijatých studentů na FIT (měřeno percentilem Národních srovnávacích zkoušek od SCIO) v posledních letech roste.
- Přesto se studijní výsledky našich studentů spíše zhoršují.
- **To nedává smysl! Někde musí být chyba! Ale kde?**

## Práce při studiu

- Jistě, když jsou studenti v práci, nemohou být na přednáškách.
- Apelujeme na studenty, aby rozumně vyvážili práci a studium. Nejen prostřednictvím projektové praxe nabízíme studentům možnost vydělat si v našich projektech.
- Naši firemní partneři se musí zavázat, že budou studentům poskytovat dostatečný prostor pro studium. Ostatní firmy ke studentům nepouštíme.
- **Je to ale skutečná příčina? I dříve studenti při studiu pracovali či podnikali a na studijních výsledcích se to neprojevovalo.**

# Studijní koutek – IT a studium (2/4)

## Digitální média

- Sociální sítě vytvářejí závislost prostřednictvím dopaminové vazby.  
Čím je uživatel déle připojen, tím víc reklamy se prodá.
- Mladí lidé ztrácejí schopnost navazovat reálné vztahy.  
Mají podstatně méně sexu než předchozí generace.
- Klesá doba, po kterou je mozek schopen se soustředit.
- Průměrná inteligence přestala růst a začala klesat.
- Zato roste počet lidí s úzkostmi a depresemi.
- Na lékařské fakulty přichází gramlaví medicí.
- Závislost na moderních médiích vznikne snáze než závislost na alkoholu.

# Studijní koutek – IT a studium (3/4)

## Co s tím dělá společnost?

- osvěta (např. MUDr. Martin Jan Stránský)
- zákazy mobilních telefonů ve školách (např. Francie)
- úprava Mezinárodní statistické klasifikace nemocí
- budování center pro léčení závislosti na internetu (např. Německo)
  - počítačové (zejména on-line) hry
  - on-line komunikace (např. sociální sítě)
  - stránky s pornografickým obsahem
  - získávání co největšího množství informací

# Studijní koutek – IT a studium (4/4)

## A co s tím můžete dělat vy?

- Dbejte na dostatek spánku.  
Je důležitý pro utřídění informací a zahození „smetí“.
- Využívejte fyzická média (papír).  
Zlepšují učení (rámování). Aktivují více center v mozku současně.
- Nenoste si mobily a notebooky na přednášky.  
Rozptylují Vaši pozornost (i vypnuté).
- Probíranou látku si zopakujte.  
Různé úhly pohledu na téma pomáhají dlouhodobějšímu zapamatování.
- Bojujte se svými závislostmi.  
Vydržíte třeba dva týdny bez internetu a mobilu (detox)?