

# ÚVOD DO SOFTWAREVÉHO INŽENÝRSTVÍ

Předmětem softwarového inženýrství jsou metodiky pro řízení vývoje softwaru.

Proč potřebujeme tyto metodiky?

Čím je vývoje softwaru specifický oproti jiným odvětvím?

## SOFTWAREVÉ INŽENÝRSTVÍ

Softwarové inženýrství je **zavedení a používání inženýrských principů tak, abychom dosáhli ekonomické tvorby softwaru.**

Takto vytvořený software je spolehlivý a pracuje na dostupných výpočetních prostředcích.

## PODMÍNKY ÚSPĚCHU VE VÝVOJI SW

- Vhodné sestavení vývojového týmu
- Volba správných nástrojů
- Úvaha koupit/vyvíjet
- Nalézt společnou řeč se zadavatelem
- Řešení budoucí údržby a rozšiřování

## CO JE TO METODIKA?

Metodika vývoje SW zahrnuje všechny etapy řešení, měla by odpovídat na otázky

Proč? Kdo? Kdy? Co?

Metodika je tedy souhrn postupů vedoucích k dodání funkčního software.

## KLASIFIKACE METODIK

### Tradiční metodiky

- Model „napiš – oprav“ (Build and Fix)
- Striktní posloupnost fází (Stagewise)
- Vodopádový model (Waterfall)
- Spirálový model
- Další metodiky: RUP, USDP, ...

### Agilní metodiky

- Extrémní programování
- Crystal
- SCRUM
- Aspect Oriented Programming
- Test Driven Development

# TRADIČNÍ METODIKY

## MODEL „NAPIŠ – OPRAV“ (BUILD AND FIX)

Implementace -> Dodání -> Opravy chyb

## STAGEWISE MODEL

Tento model vývoje software byl definován v roce 1957

Založen na striktní posloupnosti fází:

- Definice problému
- Analýza
- Specifikace požadavků
- Návrh
- Architektura
- Implementace (+testování)
- Provoz

Tato metoda vývoje software je založena na následujících skutečnostech:

Absence zpětné vazby, neprovádí se revize žádné fáze, nerevidují se požadavky ani se nehledají rizika.

## MODEL VODOPÁD (WATERSTAGE)

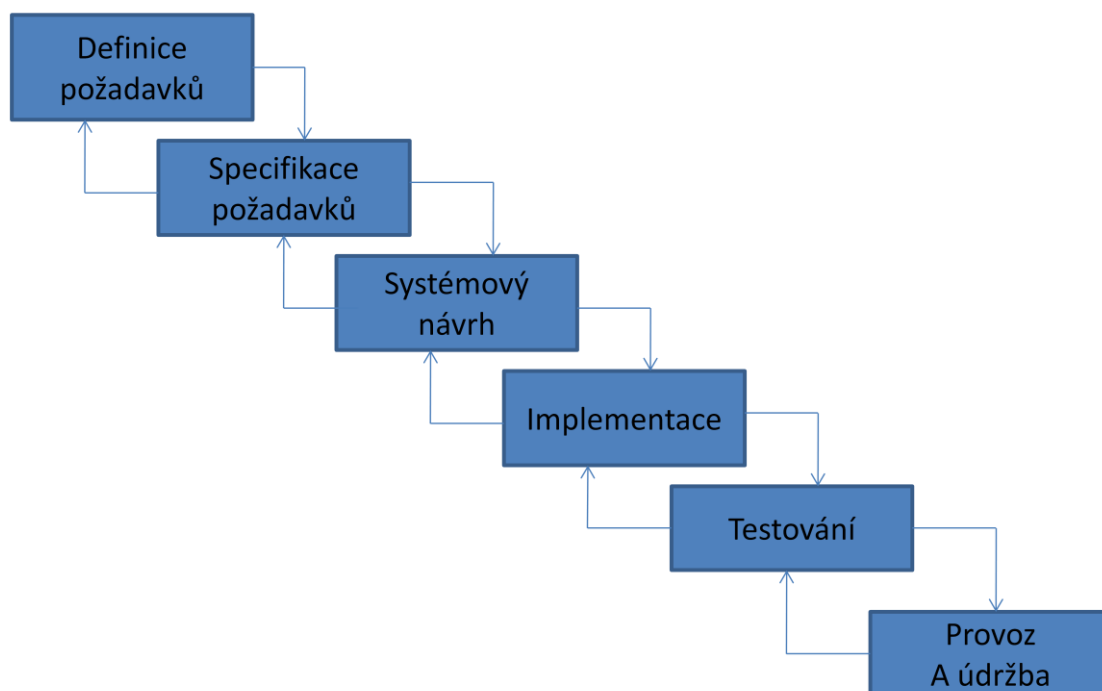
- 1970 Winston Royce
- Každá etapa má stanovený přesný cíl a dokumenty, které musí v jejím průběhu vzniknout
- Na konci každé etapy dochází k jejímu vyhodnocení a případně přepracování nebo opravení

- Možnost vrátit se zpět do předchozí etapy
- Pokračuje se teprve tehdy, je-li etapa zcela dokončena a schválena (pak již návrat není možný)

#### Výhody:

- Jednoduchý
- Ideální pro řízení
- Vnáší disciplínu do vývoje

## Model „vodopád“



#### Nevýhody:

- Dodávka formou „velkého třesku“
- Určitá nepružnost
- V době mezi analýzou a nasazením se mohou změnit požadavky

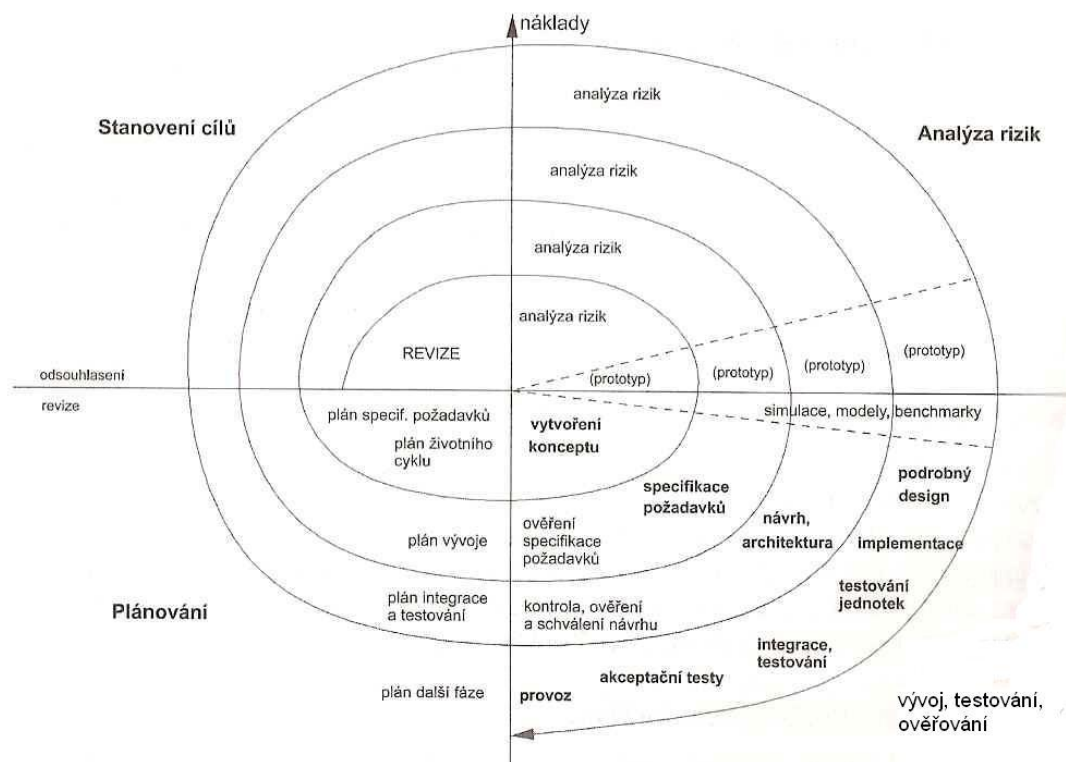
„Začnu-li padat, nezastavím se dříve, než se rozbiji o kámen zvaný předvedení“

## SPIRÁLOVÝ MODEL

- 1985, Barry Boehm
- Zavádí iterativní přístup a opakovanou (důslednou) analýzu rizik

Rizika – situace nebo události, které mohou způsobit nesplnění cílů projektu.

Vychází se z předpokladu, že na začátku je obtížné nebo až nemožné přesně specifikovat všechny funkce.



Stanoví se tedy jen obecný rámec.

Vyvine se část aplikace -> konzultace se zákazníkem -> pokračuje se dalším krokem.

Vývoj probíhá ve spirále, v postupných iteracích.

## Výhody

- Vytváří prostředí pro vývoj znovupoužitelných komponent
- Je komplexní a vhodný i pro složité projekty (díky důrazu na plánování)
- Včasné vyloučení nevhodných řešení

## Nevýhody

- Celková komplikovanost
- Software není uvolněn před dokončením posledního cyklu
- Změna požadavků je možná pouze po dokončení cyklu
- Pro nové druhy aplikací (např. internetové) je nepružný

Je vhodnější pro rozsáhlé projekty!

## RUP – RATIONAL UNIFIED PROCESS

- Vývoj probíhá v iteracích; dělí se na 4 fáze: zahájení, projektování, realizace, předání (zákazníkovi nebo do další fáze vývoje).
- Robustní, propracovaná metodika, vhodná pro větší projekty a rozsáhlejší týmy.
- Komerční produkt
- Je založen na UML

## Klíčové principy:

- Iterativní vývoj softwaru (vychází ze spirálového modelu, průběžná detekce rizik)
- Správa a řízení požadavků (požadavky se v čase mění)
- Použití komponentové architektury
- Vizuální modelování softwaru (za účelem porozumění systému; UML)
- Průběžné zajišťování a ověřování kvality (po předání je nalezení problému dražší)
- Řízení změn (počítáme s tím, že změny nastanou, neřízení změn vede k chaosu)

#### Výhody RUP:

- Obecnost a mohutnost
- Iterativní přístup – včasné odhalení rizik
- Snazší správa změn
- Provázanost s notací UML, dokumentace
- Výrobce průběžně pracuje na zlepšování metodiky
- Existence doplňkových nástrojů

#### Nevýhody RUP:

- Komerční, tj. placený produkt
- Rozsáhlost RUP může být na škodu u malých týmů – tým stráví spoustu času implementací metodiky
- Její použití vyžaduje hluboké studium, týká se i projektových manažerů

Klasické metodiky vývoje SW jsou tedy založeny na **striktní definici postupů**.