# ISKB12 INFORMAČNÍ SYSTÉMY

Ing. Mgr. Pavel Synek

19.5.2023

# ÚKOL


WHEN YOU FORGET TO
DO YOUR HOMEWORK...
AND THE TEACHER IS WALKING AROUND
imgflip.com

# HAPPY HOUR...

Kdo nestihl úkol odevzdat, může tak učinit dnes do 11:00...


# TERMÍNY ZKOUŠKY!

## S1 Coexistence / Symbiosis


- ✓ Retain both IT systems.
- ✓ Build portal above current systems to aggregate information.
- ➢ Speed to integrate is fast but savings and synergies are low.

## S2 Absorption / Takeover


- ✓ Dominant IT organizational form will be sustained through a full integration and consolidation process.
- ➢ Speed to integrate is fast but savings and synergies are high.

## S3 Best of breed / Standardization


- ✓ When the strategic intent of the merger is to add value through capitalizing on merger synergies in the two organizations.
- ➢ Speed to integrate slow, savings-low, synergies-medium.

## S4 Transformation


- ✓ Involves the installation of entirely new computing platforms and IT infrastructure.
- ➢ Speed to integrate slow and savings & synergies are medium.

# ÚROVEŇ INTEGRACE

S1—Koexistence/symbióza Zachování obou IT systémů. Vybudování portálu nad stávajícími systémy pro shromažďování informací. Zaměřit se na standardizaci obchodních procesů a konsolidaci kmenových dat. To je nutné pro všechny stávající systémy zvlášť a vyžaduje další úsilí. Rychlost integrace je rychlá, ale úspory a synergie jsou nízké.

S2—Absorpce/převzetí Dominantní organizační forma IT se udrží prostřednictvím procesu úplné integrace a konsolidace. Na jeho základě se vytvoří šablona ERP, která se implementuje ve všech obchodních jednotkách. Bude dosaženo všech tří primárních strategických cílů ERP. To však vyžaduje velké úsilí o reengineering podnikových procesů ve všech podnikových oblastech, v nichž je nový systém implementován, v kombinaci s odpovídajícím školením a řízením organizačních změn . Rychlost integrace je rychlá, ale úspory a synergie jsou nízké.

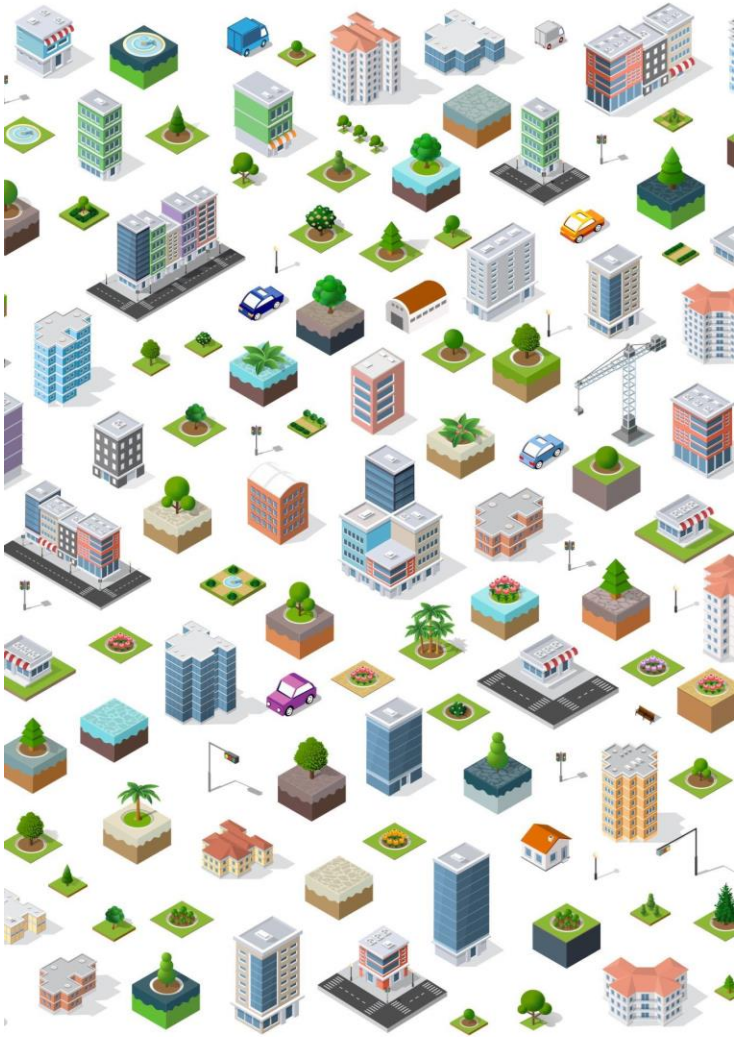
S3—Best of Breed/Standardizace Realizuje strategický záměr přidat hodnotu fúze prostřednictvím využití synergií ve všech organizačních jednotkách. Výsledkem je také šablona ERP, která však představuje synergii nejlepších postupů všech organizačních jednotek, a proto vyžaduje reengineering podnikových procesů a řízení organizačních změn na všech úrovních a ve všech organizacích. Rychlost integrace je pomalá, úspory jsou nízké při středních synergiích.

S4—Transformace Zavedení zcela nové IT platformy a infrastruktury při současném nahrazení stávajících systémů je nejkomplexnější a nejpropracovanější ze všech čtyř strategií. Výsledkem je rovněž šablona ERP. Tato strategie má smysl, pokud jsou stávající systémy zastaralé nebo nemohou splnit nové požadavky. Rychlost integrace je pomalá a úspory a synergie jsou střední.

# 1. ERP

Obchodní struktury, procesy, a IT infrastruktura jsou silně provázané v ERP a lze je tedy standardizovat při migraci do v hodného ERP řešení.

ERP - systém podporující všechny business procesy ve firmě. Zpravidla nabízí moduly pro podporu sales, produkce, správy majetku, financí, účetnictví, aj.



# 1. ERP

3 základní strategické cíle:

1. Vytvoření architektury společných podnikových procesů
2. Standardizace interních a externích dat
3. Standardizace architektury informačních systémů

# 1. ERP – STANDARDIZACE PODNIKOVÝCH PROCESŮ

- Podnikový proces – soubor vzájemně provázaných úkonů, které mají jeden či více vstupů a konkrétní výstup pro uživatele
- Standardizace procesů umožňuje vytvoření definovaného a vzájemně sdíleného rámce pro lepší kontrolu nad unifikovanými a integrovanými procesy.
- Výsledkem lepší výměna informací a služeb napříč jednotlivými odděleními a pobočkami stejně tak jako směrem ven

# 1. ERP – STANDARDIZACE PODNIKOVÝCH PROCESŮ

Standardizace procesů nutná. Vzniká provázanost mezi organizacemi, kterou nutno reflektovat a kde lze nalézat synergii

Kde ji hledat:

- Produkce
- Sales
- Logistika
- Finance
- Účetnictví,...

# 1. ERP – STANDARDIZACE PODNIKOVÝCH PROCESŮ

- Lokalizace standardů?
  - Máme kulturní rozdíly mezi firmami?
    - Příklad: pivo v arabských zemích
    - Příklad: McDonalds – velikosti porcí v USA vs. Evropě, halal food v arabsky mluvících zemích, vepřové v Turecku

# 1. ERP – STANDARDIZACE PODNIKOVÝCH PROCESŮ

- Standardizace master dat
  - Chceme-li konsolidovat základní data o firmě (např. Nákup/prodej), je nutné začít zde…
  - Standardní formát
  - Datový standard
  - Standard ve shromažďování dat
  - Definice celého procesu zpracování dat
  - Právní standardizace

# 1. ERP – STANDARDIZACE PODNIKOVÝCH PROCESŮ

- Standardizace master dat
- Kvalita dat a její hlavní aspekty
  - Přesnost
  - Kompletnost
  - Aktuálnost
  - Konsistence
  - Redundance
  - Dostupnost

# 1. ERP – STANDARDIZACE PODNIKOVÝCH PROCESŮ

- Standardizace IT infrastruktury
  - Microsoft vs. Google?
  - AWS, Snowflake, Teradata, Oracle,…?
  - On-prem / cloud?

# 1. ERP – STANDARDIZACE PODNIKOVÝCH PROCESŮ

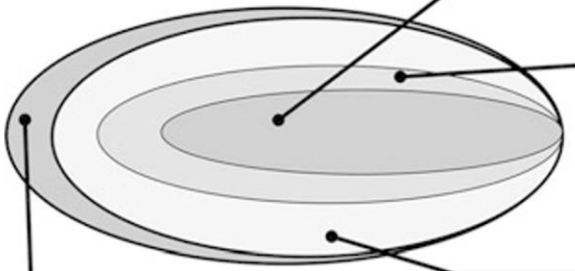
- Total Cost of Ownership
  - popisuje náklady po celý životní cyklus ve společnosti
  - celkové náklady se dělí na akvizici (sw, hw), provoz (servery, síť), technickou podporu (maintenance, trénink, asistence) a uživatelské aktivity
  - **Direct costs** (cca 60 procent)
    - kapitálové náklady, administrace, technická podpora. Fixní náklady na provoz a údržbu. Přímé náklady lze plánovat.
  - **Indirect costs** (cca 40 procent)
    - náklady vytvářené koncovými uživateli a systémovými změnami.
    - Mohou vzniknout kvalitou uživatelské podpory, je těžké je plánovat a předvídat (trénink, dostupnost systémů, response time systému, atp)

# 1. ERP – STANDARDIZACE PODNIKOVÝCH PROCESŮ

- základní idea ERP je set standardizovaných business process modulů, které lze nastavit vcelku libovolně tak, aby vyhovovaly celkem jakékoliv reálné situaci a business modelu bez nutnosti programovat, ale pouze za pomoci úpravy parametrů a master dat.
- Tyto moduly jsou nastavené v závislosti na centrální jádro ve smyslu HW A SW
- V reálu ale nelze nastavit na 100 procent procesů  za pomoci jen standardní konfigurace
- procesnímu reengineeringu mohou často vadit vyšší okolnosti (právní, obchodní, kulturní, finanční, organizační, bezpečnostní, …)
- z toho vyplývá konfigurace vs. customizace

# 1. ERP – STANDARDIZACE PODNIKOVÝCH PROCESŮ

- ERP Template
  - Konfigurace vs. Customizace
  - Přílišná customizace je cestou do pekel



**Kernel**
The Kernel defines all globally applicable system structures and data standards, such as: global Chart of Accounts, global parameters (currencies,...), SAP hierarchies, key master data standards.

**Process Clusters**
This repository of business process scenarios is the key to the success and speed of the roll-outs.
The goal will be to cover 90 % of the global common business scenarios, which are fully integrated. It includes common interfaces, reports and bolt-ons.

**Reference System**
Pre-configured system based on Kernel & Process Clusters.

**ERP Template Lokalisierung**

Regional / Business Unit variations
Market Specific Customizations

**ERP Template**

Localization effort increases

- Modify the original software code
- Create an interface to a legacy system
- Create a custom developed component
- Configure specific settings in the template
- Execute a manual task
- Do nothing

Podmínkou pro úspěšné nasazení nového ERP systému je připravenost organizace na změnu

# ÚROVEŇ KONSOLIDACE

Má smysl slučování produktů?

Portfolio nabízených produktů každé z poboček se mírně liší

Pobočka 1 - nabízí především sudy

Pobočka 2 – orientuje se na běžného zákazníka - zaměření na malá balení

Pobočka 3 – chce nabízet vše

Pobočka 4 – orientuje se na prémiovou klientelu

# ERP : CHANGE MANAGEMENT

Tři fáze:

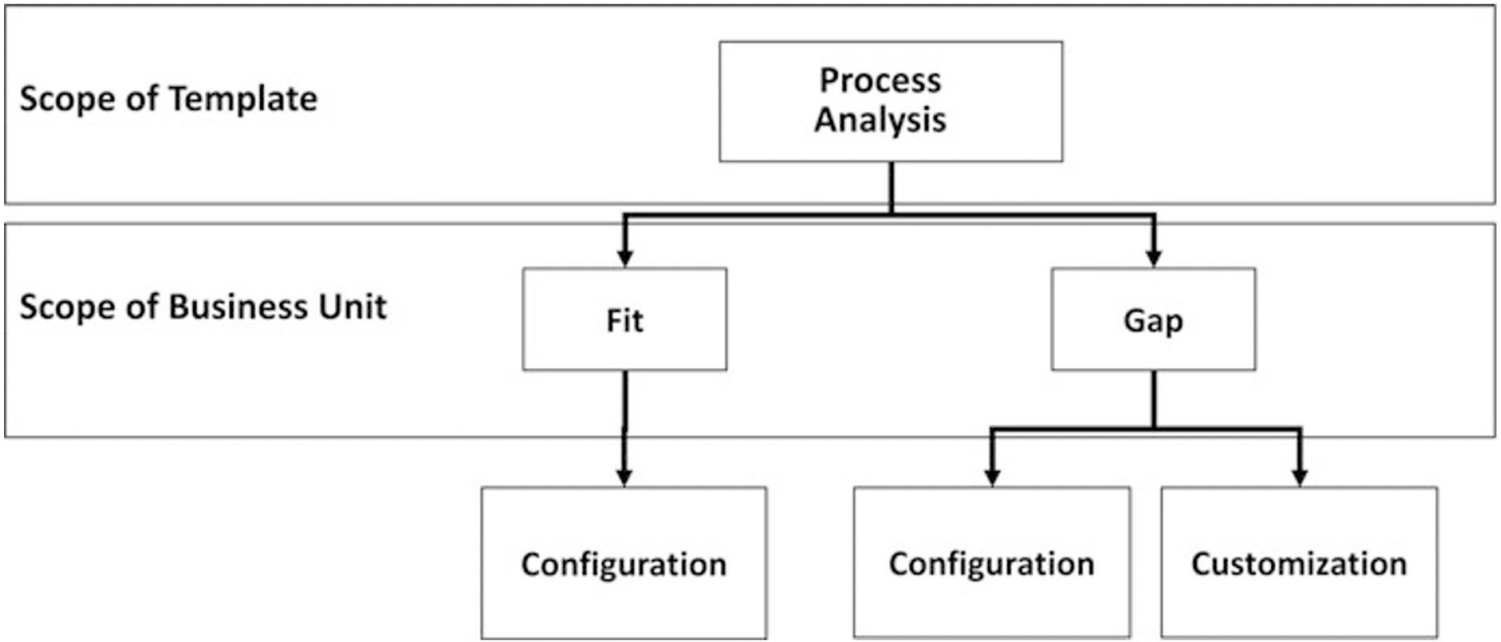
- příprava
- implementace
- integrace do organizace

Cíle:

- Připravit lidi na co, co bude
- Zrychlit adaptační období, zajistit co nejmenší pokles efektivity práce
- Integrovat a ladit

# ERP : CUSTOMIZACE

- nutno prioritizovat a domluvit rozumnou hranici toho, co bude odbaveno a co ne.
- body ke customizaci:
  - modifikace originálního zdrojového kódu dodaného sw - časovaná bomba!!
  - UI k původnímu systému
  - custom komponenta
  - vytvoření specifických parametrů
  - manuální úkoly
- customizace jsou hrozbou pro stabilitu standardního řešení.
- fit-gap analýza - nutno vždy zjistit, zda daná funkcionalita nahradí tu chybějící a nakolik zasahuje do stávající struktury standardizovaného řešení


Scope of Template
Process Analysis
Scope of Business Unit
Fit
Gap
Configuration
Configuration
Customization

# PROJEKTOVÉ ŘÍZENÍ V IT

# VLASTNOSTI PROJEKTŮ IS/ICT

- Unikátnost
- Snaha o přesné určení cílů a výstupů projektu
- Dané termíny (realizace projektu ve stanoveném čase)
- Daná cena a omezené zdroje
- Interdisciplinární charakter a dynamické projektové týmy
- Projekty se musí vyrovnávat s riziky a faktorem neznámého

# ODLIŠNOSTI ICT PROJEKTŮ OD TRADIČNÍCH PROJEKTŮ

- Jejich produkty (ICT) jsou převážně nehmotné povahy, tedy obtížněji definovatelné a mentálně uchopitelné. Tato vlastnost může být i jedním z důvodů stále nepříliš velkého procenta úspěšných projektů ICT.
- Nemají většinou jasně definované zadání, cíle, uživatelské požadavky, obsah jednotlivých výstupů. Tyto obsahové náležitosti projektu se objasňují až v průběhu projektu.
- Podporují podnikové procesy a jsou jedním z nástrojů dosahování reálných potenciálů zlepšení podnikových procesů. Tato skutečnost ale současně významně ovlivňuje množství projektových změn, které musí být v průběhu projektu zpracovávány a integrovány do řešení.
- Terminologie, metodiky, metody a techniky související s řízením projektů ICT a budováním ICT nejsou jednotné.

# PODNĚTY PRO VZNIK PROJEKTŮ IS/ICT

Z existující a udržované Informační strategie,

Ze zhodnocení průběhu vykonávaných hlavních, podpůrných a řídících procesů podniku,

Z podnětů a požadavků uživatelů IS/ICT.

# VYMEZENÍ POJMŮ

- **Metoda** je návod či postup získávání poznatků.
  - Odpovídá na otázku: JAK dosáhnout vytyčeného cíle? Zaměřuje se na určitou etapu vývoje SW.
- **Metodika** je nauka o metodě nebo pracovní postup. Jedná se o doporučený souhrn etap, přístupů, zásad, postupů, pravidel, metod, technik, nástrojů, dokumentů, metod řízení, atd.
  - Odpovídá na otázky typu CO, KDY KDO, PROČ. Ve vývoji software představuje metodika souhrn doporučených praktik a postupů, pokrývajících celý životní cyklus vytvářené aplikace.

# VYMEZENÍ POJMŮ

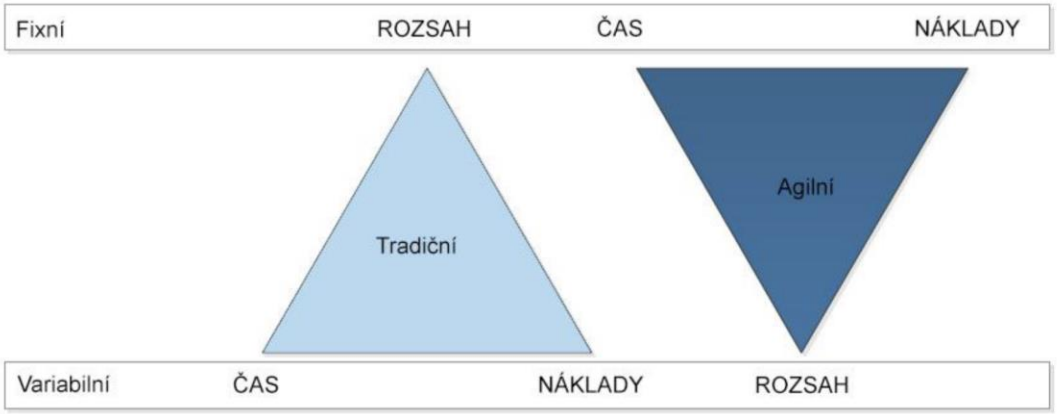
- **Technika** zahrnuje postupy kroků, jak se dobrat k výsledku (např. normalizace datového modelu, prototypování, aj.).
- **Nástroj** je prostředek k uskutečnění určité činnosti, resp. k vyjádření výsledku dané činnosti (formalizuje vyjádření výsledku). Může být svázán s konkrétní technikou, např. CASE nástroje, modely IS (datový, funkční, stavový diagram).

# VYMEZENÍ POJMŮ

- **Projekt** je „časově ohraničené úsilí, směřující k vytvoření unikátního produktu nebo služby“. V této obecně přijímané definici jsou klíčové zejména omezení projektu v čase a jedinečnost jeho výstupů, protože právě tyto charakteristiky ho odlišují od procesu.
- V každém projektu je nutné uvažovat nad třemi základními konstantami - čas, náklady a rozsah projektu (zadání). Některé z nich jsou fixní (neměnné), jiné se naopak mění během projektu.

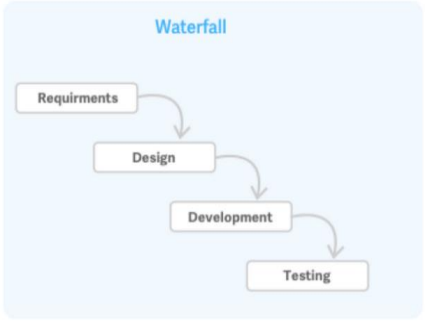
# VYMEZENÍ POJMŮ

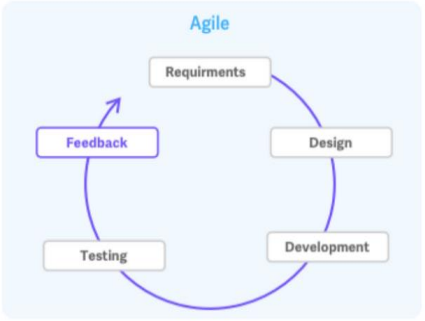
- **Projektový trojimperativ** nebo také projektový trojúhelník znamená, že nelze zvyšovat kvalitu výstupů (CO?), aniž by to mělo dopad na cenu (ZA KOLIK?) nebo čas (KDY?) a analogicky to platí i opačnými směry.
- Proto je v projektu nutné vymezit i tzv. určující osu, která vymezuje jedinou hlavní prioritu v rámci cílů projektu (kvalita/cena/čas).

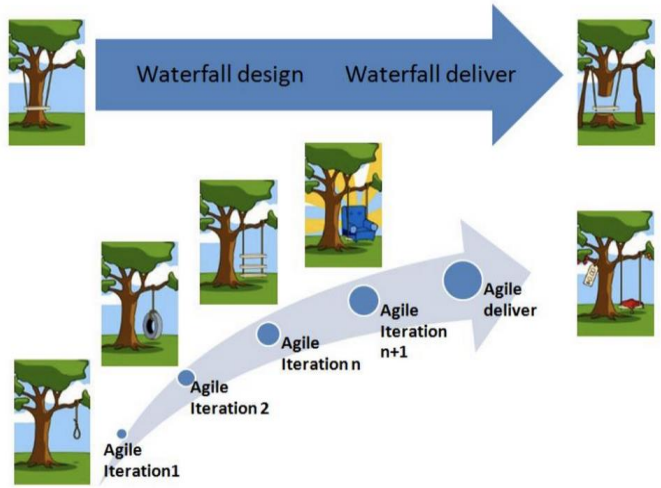


# RIGORÓZNÍ (TRADIČNÍ) VS AGILNÍ METODIKY VÝVOJE

| | Rigorózní metodiky | Agilní metodiky |
|---|---|---|
| Nástroje metodiky | Procesy se zaměřují na formálnost a dokumentovatelnost, lidé jsou sekundární faktor | Praktiky vychází ze znalostí jednotlivců, lidé jsou klíčovým faktorem úspěchu |
| Podrobnost metodiky | Podrobný popis činností a procesů | Pouze základní struktura |
| Kvalita | Zaměření na kvalitu procesů, které povedou ke kvalitnímu výsledku | Zaměření na priority zákazníka |
| Předvídatelnost | Sběr požadavků a plánování předem | Přírůstkové shromažďování požadavků, plánování po iteracích |
| Změny | Snaha změny minimalizovat a řízení změn | Snaha změny umožnit s ohledem na nové znalosti |
| Participace zákazníka na projektu | Jen v počátečních a koncových fázích | Po celou dobu projektu |
| Specializace lidí | Vyžaduje specializaci pro týmové role | Sdílení znalostí a spolupráce v týmu |
| Dokumentace | Rozsáhlá dokumentace | Minimální dokumentace, podstatné je pochopení |
| Způsob vývoje | Vodopádový, iterativní s dlouhými iteracemi | Přírůstkový vývoj s velmi krátkými iteracemi |
| Forma komunikace | Převážně písemná | Osobní |



# RIGORÓZNÍ METODIKY VÝVOJE

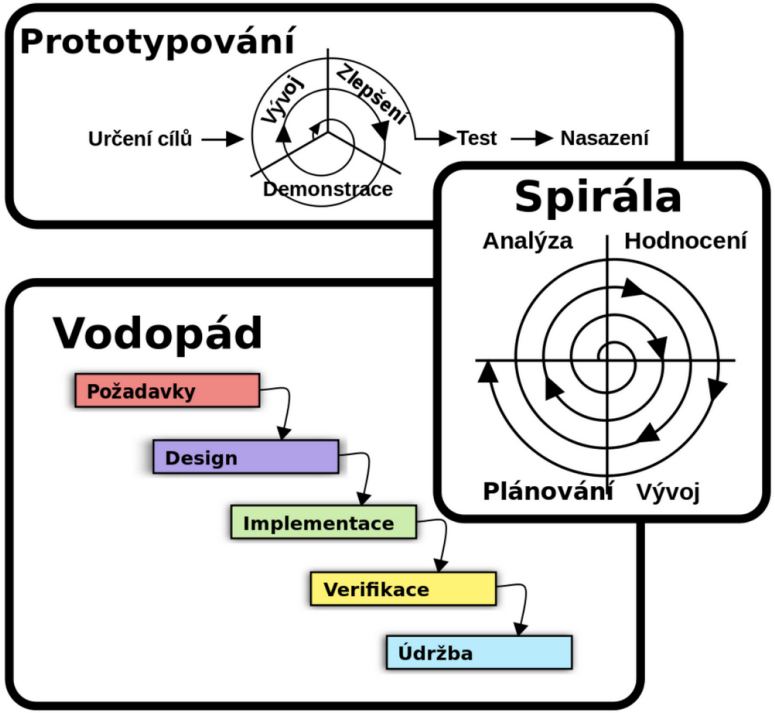
- Jedná se historicky o největší skupinu metodik, které se zabývají podrobným popisem procesu vývoje software.
- Tradiční (rigorózní) metodiky kladou velký důraz na objemnou dokumentaci.
- Celý proces je charakterizován tzv. vodopádovým modelem – definice posloupnosti jednotlivých činností během vývojového cyklu.
- Vývoj software je jasně definovaný proces, od kterého se nemůžeme odchýlit.

# RIGORÓZNÍ METODIKY VÝVOJE

- Rigorózní metodiky jsou považovány za tzv. "těžké" - jsou velmi podrobné, obsahují hodně formalit a jsou specifické direktivním řízením.
- Rigorózní metodiky předpokládají, že zákazník vlastně neví co chce. Proto je na programátorech, aby vytvořili takový produkt, který by zákazník mohl v budoucnu potřebovat.
- Procesy jsou opakovatelné a předpokládají definování všech požadavků na řešení předem.

# RIGORÓZNÍ METODIKY VÝVOJE

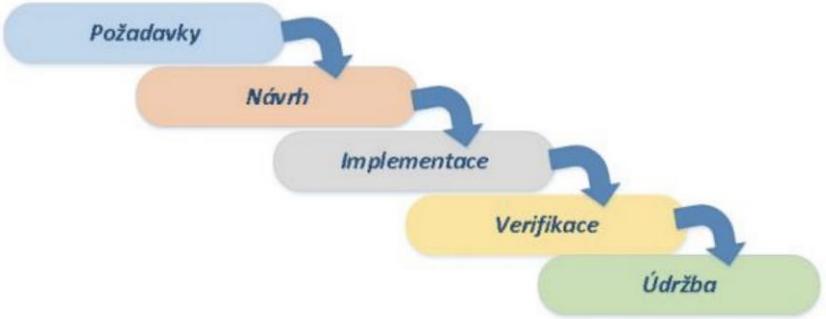
- Do rigorózních metodik patří:
  - Vodopád,
  - Prototyp,
  - Výzkumník,
  - Spirálový model,
  - Unified Process,
  - Rational Unified Process,
  - Enterprise Unified Process, aj.



# MODEL WATERFALL

- Znázorňuje určitý idealizovaný stav – posloupnost na sebe navazujících etap bez cyklických návratů zpět.
- V praxi by bylo vhodné jej dodržovat, většinou to však není možné, proto má tento model význam spíše teoretický a slouží jako základní myšlenkový postup pro studium etap životního cyklu.

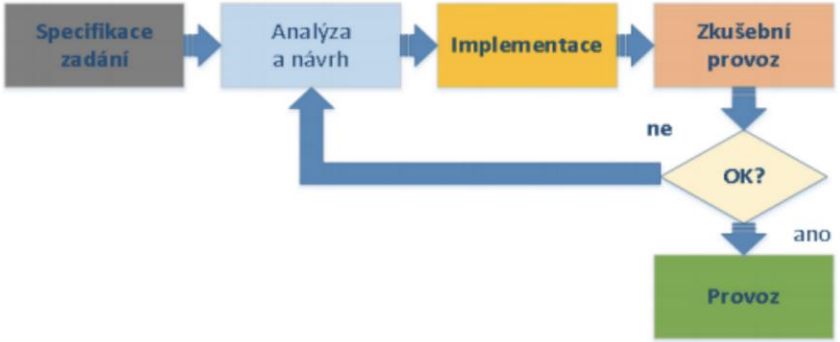
# MODEL WATERFALL




- Výhody:
  - jednoduchý z hlediska řízení
  - při stálých požadavcích: nejlepší struktura výsledného produktu
- Nevýhody:
  - zákazník není schopen přesně stanovit veškeré požadavky předem
  - při změnách požadavků má tento model dlouhou dobu realizace
  - zákazník vidí spustitelnou verzi až v závěrečných fázích projektu, z čehož vyplývá, že např. nedostatky jsou odhaleny příliš pozdě (fáze verifikace) a jejich odstranění vede k navýšení čerpání zdrojů.

# MODEL VÝZKUMNÍK

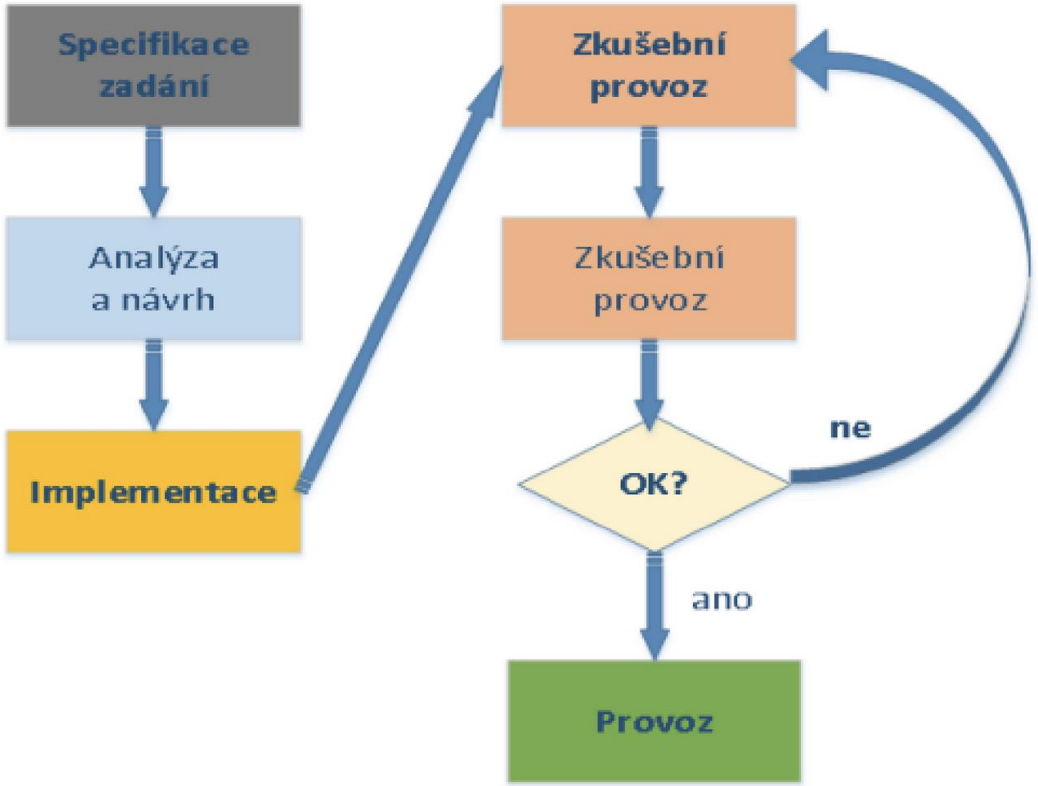
- je uváděn spíše jako negativní případ přístupu k vývoji IS.
- Jeho použití svědčí o tom, že řešitelský tým neovládá dobře danou problematiku, získává postupně zkušenosti v oblasti, pro kterou je IS určen.
- Za takovýchto okolností je doba etap těžce plánovatelná.
- U rozsáhlejších IS lze takový přístup použít (bez negativních následků) jen stěží.



# MODEL PROTOTYP

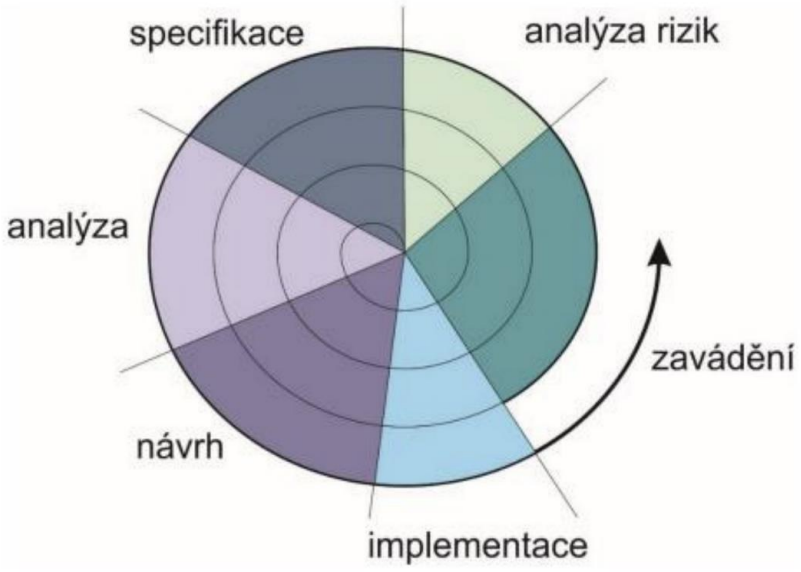
- Jde o aplikaci plánovaného a řízeného inkrementálního přístupu k vývoji IS.
- Pod pojmem „prototyp“ rozumíme částečnou implementaci produktu (části IS) v logické nebo fyzické formě, která by měla reprezentovat všechna vnější rozhraní systému.
- Uživatelé IS se s prototypem seznamují a testují jej.
- Na základě jejich připomínek je upřesňována specifikace požadavků na systém, prototyp je upravován a dále doplňován až do podoby výsledného systému.
- Na rozdíl od typu „výzkumník“ zde vycházíme z řádné analýzy a návrhu systému, jehož součástí je i rozsah prototypu v jednotlivých stupních vývoje.

# MODEL PROTOTYP



# SPIRÁLOVÝ MODEL

- Spirálový model je odvozen od vodopádového modelu. Zásadním způsobem však mění přístup a oproti vodopádovému modelu má dvě odlišné vlastnosti:
  - iterativní přístup,
  - podrobná analýza rizik.



# METODIKA UP – UNIFIED PROCESS

- Metodika vychází z průmyslového standardu SEP – Software Engineering Process.
- UP je jen zkrácené označení průmyslového standardu USDP – Unified Software Development Process (unifikovaný proces vývoje software).
- Jednoduše řečeno, jedná se o generický proces pro jazyk UML – Unified Modeling Language (unifikovaný modelovací jazyk).
- Jde pouze o základní, generickou a otevřenou metodiku, kterou je možné uzpůsobit jakémukoli rozsahu projektu.
- Metodika UP musí být přizpůsobena cílům a podmínkám daného vývoje, tzn. používaným normám, šablonám, nástrojům, atd.

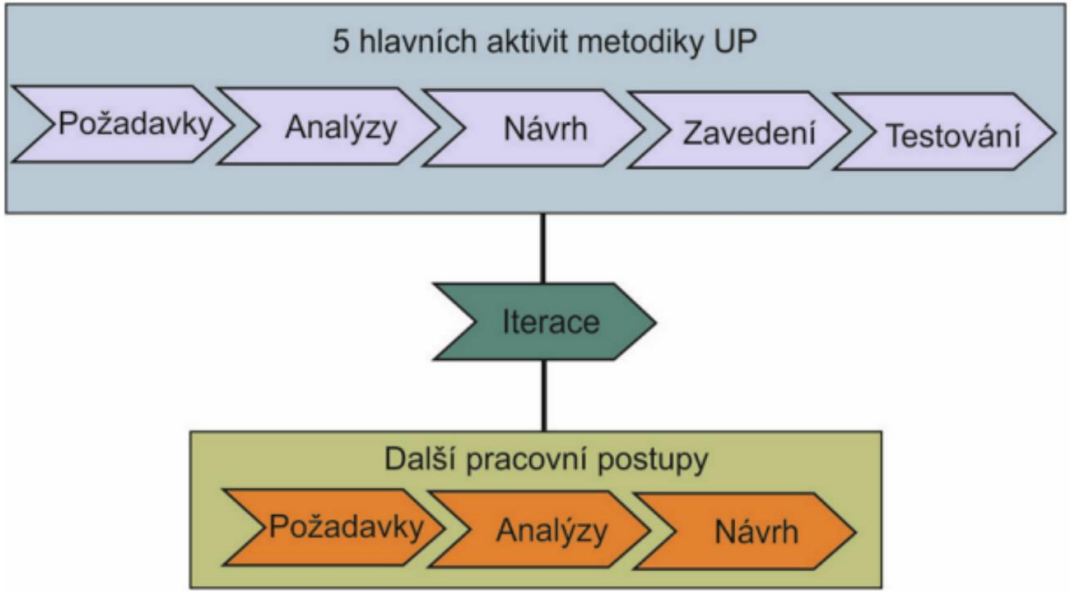
# METODIKA UP – UNIFIED PROCESS

- Unifikovaný proces (UP) znamená:
  - řízení požadavky a případy užití,
  - řízení rizikem,
  - základ na architektuře,
  - iterativní a přírůstkový proces vývoje SW produktu.

- UP je rozdělen do jednotlivých iterací, z nichž každá prochází pěti základními pracovními procesy:
  - stanovení požadavků,
  - analýza,
  - návrh,
  - implementace,
  - testování,
  - iterace v UP.

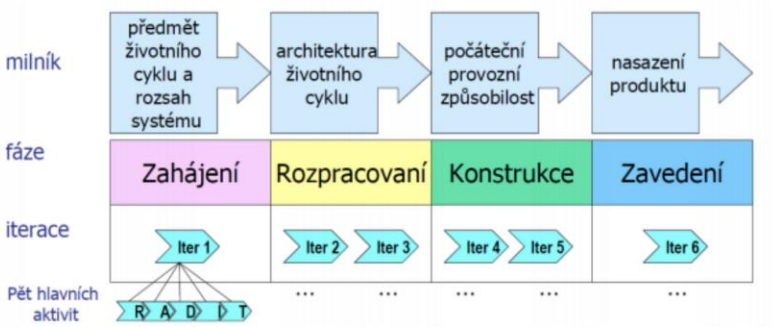
# METODIKA UP – UNIFIED PROCESS

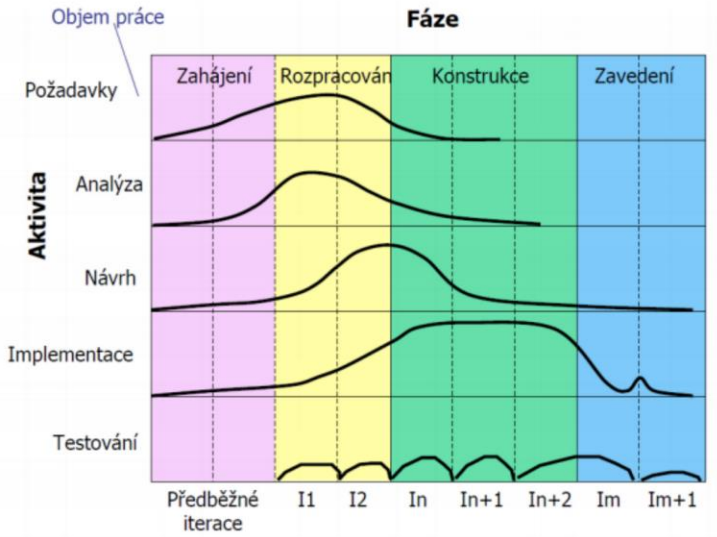
- Iterace je klíčovým prvkem UP.
- Každou iteraci v UP můžeme chápat jako samostatný dílčí projekt, který má své etapy (jako každý projekt).
- Etapy iterace v UP jsou tvořeny následujícími činnostmi(aktivitami):
  - plánování
  - analýza a návrh
  - implementace
  - integrace a testování
  - interní nebo externí uvedení



# METODIKA UP – UNIFIED PROCESS

- Každá iterace v UP vytváří tzv. base line, tj. základní linii.
- Základní linie představuje souhrn schválených a revidovaných prvků a představuje určitý základ pro následné přezkoumání a vývoj.
- Iteraci je možné měnit pouze prostřednictvím formálních metod správy změn a konfigurace.
- Přírůstky jsou rozdílem mezi dvěma následnými základními liniemi.
- Proto se metodice UP říká také „metodika iterací a přírůstků“




# METODIKA UP – UNIFIED PROCESS

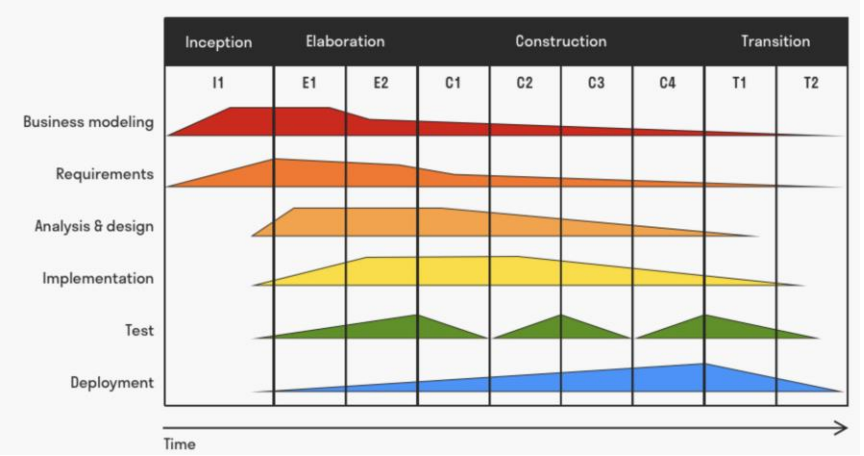
# RUP (RATIONAL UNIFIED PROCESS)

- RUP je běžně dostupnou verzí UP, přičemž metodika RUP obsahuje veškeré normy obsažené v UP.
- Její součástí je i bohaté uživatelské prostředí, přičemž RUP rozšiřuje UP v mnoha významných parametrech
- I když mají obě metodiky mnoho společného, najdeme hlavní rozdíly v jednotlivých detailech a úplnosti metodik, než v samotné sémantice.

# RUP (RATIONAL UNIFIED PROCESS)

- Metoda RUP využívá během vývoje následující ověřené postupy:
- iterativní vývoj (postupné zjemňování, přidávání vlastností),
- správa uživatelských požadavků(doplňování, třídění, hodnocení),
- použití architektury komponent (využití stávajících vzorů, komponent částečných řešení),
- vizuální modelování (vytváření modelů reality),
- sledování kvality.

- Podobně jako UP, RUP má 4 etapy vývoje:
  - zahájení,
  - rozpracování,
  - realizace,
  - zavedení.

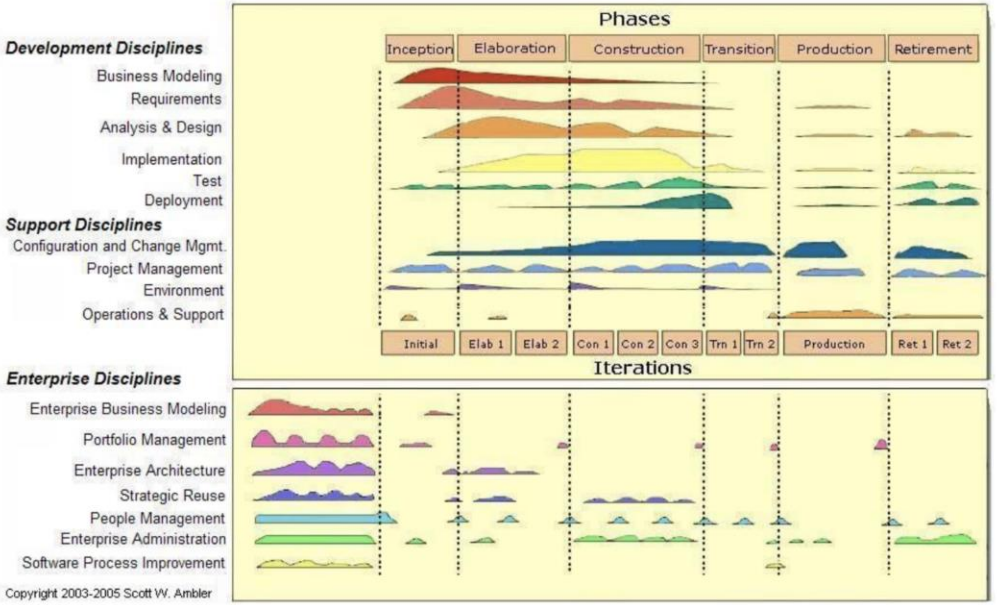


# EUP (ENTERPRISE UNIFIED PROCESS)

- Metodika EUP je další komerční verzí vycházející z RUP.
- Podobně, jako RUP rozšiřuje a obohacuje UP, stejně tak EUP rozšiřuje a obohacuje procesní Framework RUP.
- Toto rozšíření se podrobuje normě ISO 12207.
- Metodika EUP usiluje o pokrytí celého životního cyklu vývoje software.

# EUP (ENTERPRISE UNIFIED PROCESS)

- Základ EUP rozšiřuje RUP o dvě fáze:
- fáze produkční (Production),
- fáze stažení z provozu (Retirement).
- Součástí metodiky EUP jsou rovněž nově zavedené součásti:
  - provoz a podpora (Operations andSupport),
  - modelování podnikových procesů (Enterprise Business Modelling),
  - správa portfolia (Portfolio Management),
  - podniková architektura (Enterprise Architecture),
  - strategie znovupoužitelných komponent, postupů a šablon (Strategic Reuse),
  - zlepšování softwarového procesu (Software Process Improvement).



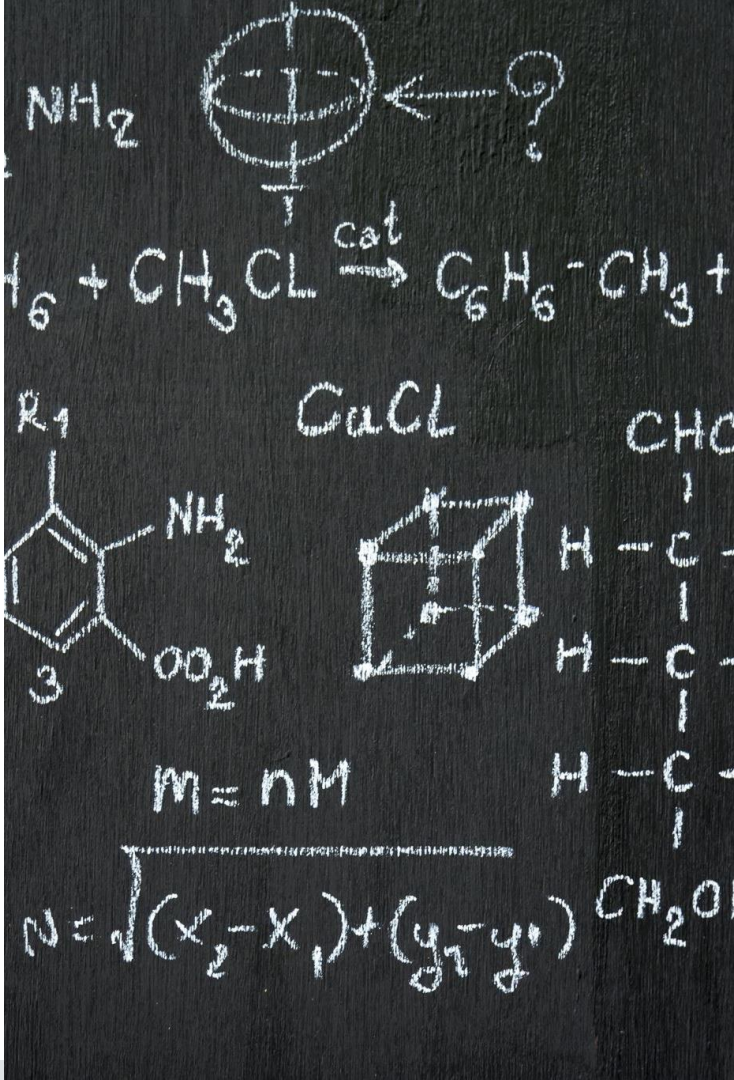

# EUP (ENTERPRISE UNIFIED PROCESS)

# AGILNÍ METODIKY VÝVOJE

- Agilní metodiky kladou (narozdíl od rigorózních) velký důraz na zákazníka.
- Software musí dodávat hodnotu pro zákazníka, a nikdo jiný, než zákazník sám nemůže tuto hodnotu určit.
- Agilní metodiky představují velkou změnu paradigmatu - nejedná se o proces, ale spíše o přístup k vývoji software.
- Agilní metodiky upřednostňují neustálou interakci a kooperaci v rámci týmu, před "standardizací" lidí v organizaci a jejich neustálou kontrolou. Jsou více flexibilní ke změnám a nesledují tolik plán.

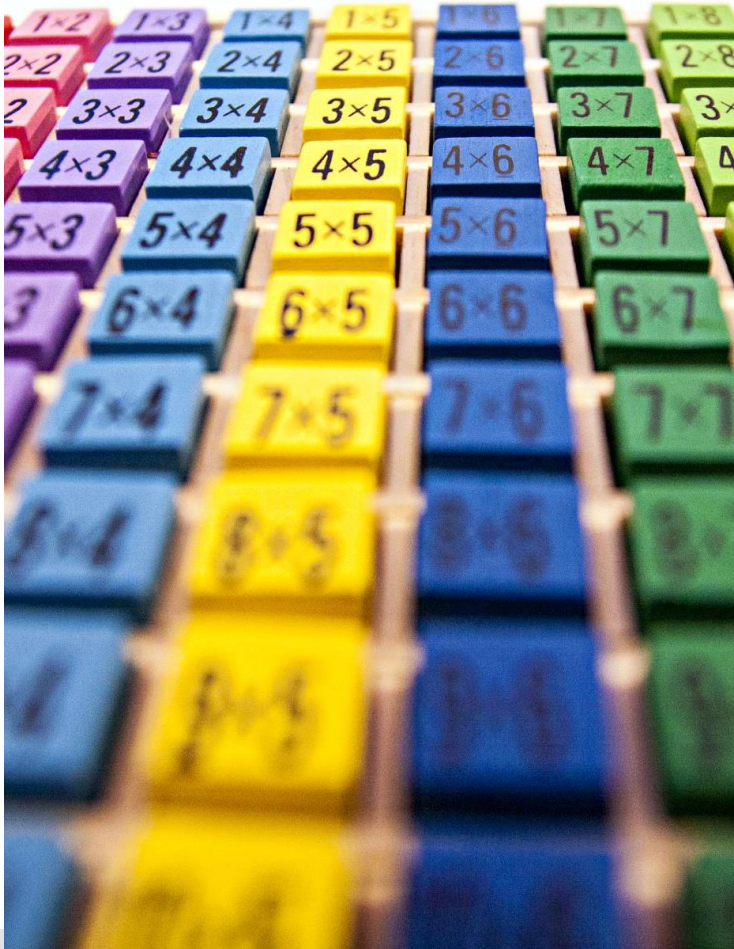
# AGILNÍ METODIKY VÝVOJE

- Agilní metodiky jsou skupiny metod původně určených pro vyvíjení SW založené na iterativním a inkrementálním vývoji.
- Umožňují rychlý vývoj softwaru a zároveň dokáží reagovat na změnu požadavků v průběhu vývojového cyklu.
- Podle těchto metodik se správnost systému ověří jedině pomocí rychlého vývoje, předložení zákazníkovi a následných úprav dle zpětné vazby.



# AGILNÍ METODIKA A TECHNIKA

- **Agilní metodika** je způsob rozvržení a ověřování práce. Cílem Agilní metodiky bývá lepší organizace práce. Odlišné Agilní metodiky vedou k odlišným produktům, protože považují jiné aspekty produktu za klíčové. Liší se i přístup ke změně zadání. Možnost změny zadání je ale určující podmínkou.
- **Agilní technika** je naproti tomu konkrétní postup, který se většinou týká samotných vývojářů. Cílem těchto technik bývá zvýšení produktivity práce, odstranění chyb, kvalitnější software nebo přesnější dodržení specifikace. Agilní techniky jsou od metodik oddělitelné. Cílem Agilní techniky bývá kvalitnější produkt.



# ROZDÍLY MEZI AGILNÍ METODIKOU A TECHNIKOU

- Agilní metodika se zaměřuje na organizaci práce bez ohledu na to, zda se ve firmě vyvíjí software, nebo se jedná o jinou oblast, kde lze agilní řízení uplatnit.
- V agilní metodice je kladen důraz na možnost změny, agilní technika nic takového nevyžaduje.
- Agilní metodika se týká nejen vývojářů, ale i managementu.
- Agilní technika je konkrétní činnost (především vývojáře, může se ale týkat i například klienta).

# PŘÍNOS AGILNÍ METODIKY A TECHNIKY

- Je předem známo, kolik práce je tým schopen udělat (v časovém rozsahu jedné iterace). To proto, že z minulosti je známo, jaká je rychlost v jednotlivých iteracích. Protože se zároveň dělají časové odhady na co nejkratší úkoly, je možné řídit rychlost týmu.
- Změny zadání za běhu (což je například pro klasický vodopádový model naprosto neřešitelný problém).
- Zrychlení doručování produktu od vývojáře k zákazníkovi (protože klientovi produkt doručujete po každém běhu – zákazník se navíc často účastní běhu firmy).

# PRIORITY AGILNÍHO PROGRAMOVÁNÍ

- Lidé a jejich spolupráce před procesy a nástroji.
- Fungující software před obsáhlou dokumentací.
- Spolupráce se zákazníkem před sjednáváním smluv.
- Reakce na změnu před dodržováním plánu.

# PRINCIPY AGILNÍHO PROGRAMOVÁNÍ

1. Nejvyšší prioritou je uspokojit zákazníka skrz rychlé a průběžné dodávání kvalitního software.
2. Změnové požadavky jsou vítány, dokonce i v průběhu vývoje. Agilní procesy je zpracují tak, aby zákazníkovi přinášely konkurenční výhody.
3. Dodávejte fungující software často, v intervalech týdnů až měsíců. Upřednostňujte kratší intervaly dodání.
4. Lidé z businessu a vývojáři musí spolupracovat každý den během celého projektu.
5. Pro práci na projektu vybírejte motivované jedince. Dejte jim prostředí a podporu, kterou potřebují, a důvěřujte jim, že práci dokončí.
6. Nejúčinnější metoda sdílení informací vývojářskému týmu (i uvnitř tohoto týmu) je osobní setkání.

# PRINCIPY AGILNÍHO PROGRAMOVÁNÍ

7. Fungující software je hlavním měřítkem postupu vývoje.

8. Agilní procesy podporují udržitelný vývoj. Sponzoři, vývojáři i uživatelé by měli být schopní dodržovat stálý výkon dokud je třeba.

9. Průběžná pozornost věnovaná technické dokonalosti a dobrému návrhu posiluje agilní přístup.

10. Základem je jednoduchost – umění co nejvíce práce vůbec nedělat.

11. Nejlepší architektury, požadavky a návrhy vznikají v týmech, které se samy organizují.

12. Tým v pravidelných intervalech vyhodnocuje svou práci a upravuje své postupy tak, aby byl co nejefektivnější.

# FÁZE CELÉHO PROJEKTU V AGILNÍCH METODIKÁCH

1. **Nultá iterace** – první krátká analýza a naprogramování nějaké základní činnosti. V agilním týmu nejde příliš o to, co se v této části bude implementovat. Podmínkou je, aby na konci byl hotový kousek aplikace, který se dá předvést (tedy i klientovi).
2. **Analýza změny** (výběr toho, co se bude implementovat, analýza a designování změn).
3. Samotná **implementace požadované vlastnosti** (či vlastností).

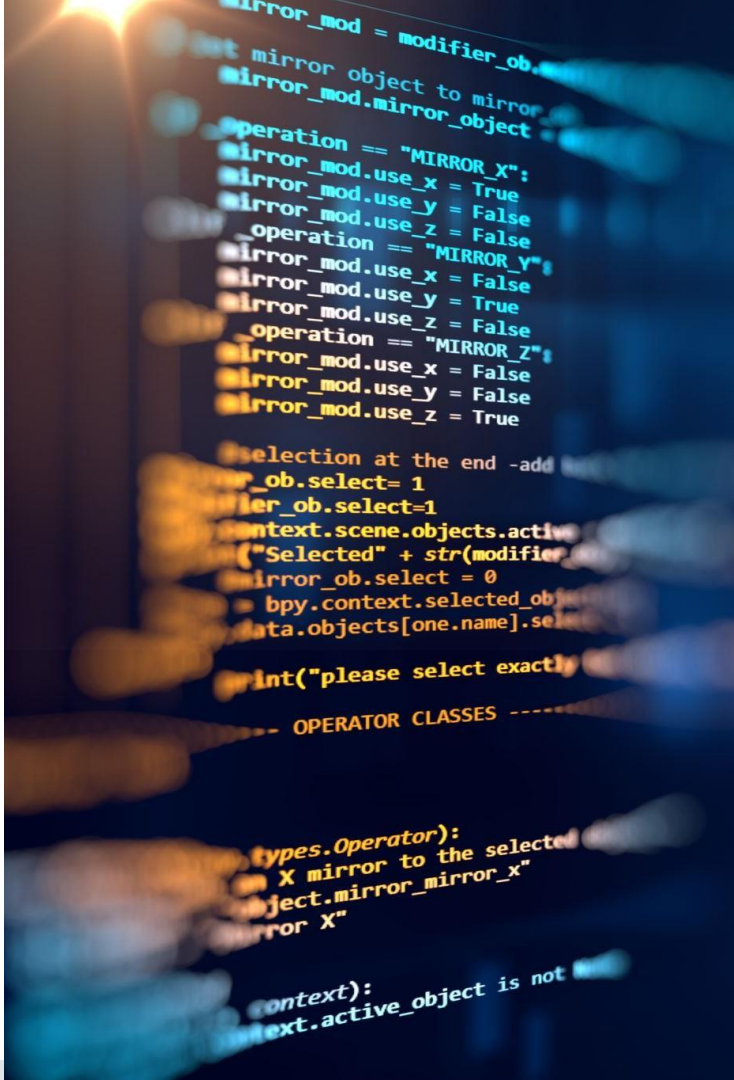
# FÁZE CELÉHO PROJEKTU V AGILNÍCH METODIKÁCH

- 4. **Předvedení klientovi.**
- 5. **Pokud není produkt hotov**, zpět na bod 2.
- 6. Pokud ano, následuje **údržba**, **rozvoj** (rovněž v iteracích).

- Body 2 – 4 se označují jako **jedna iterace**. Tyto body se opakují tak dlouho, dokud není vývoj ukončen (úspěchem, odložením projektu, proto, že dojdou peníze, změnou priorit klienta).

# AGILNÍ METODIKY VÝVOJE

- Mezi agilní metodiky patří například:
  - SCRUM,
  - Extrémní programování (XP),
  - Feature-Driven Development (FDD),
  - Test Driven Development (TDD),
  - Lean Software Development,
  - Crystal metodiky,
  - Kanban,
  - Adaptive Software Development (ASD), aj.



# SCRUM – METODA ŘÍZENÍ AGILNÍHO VÝVOJE

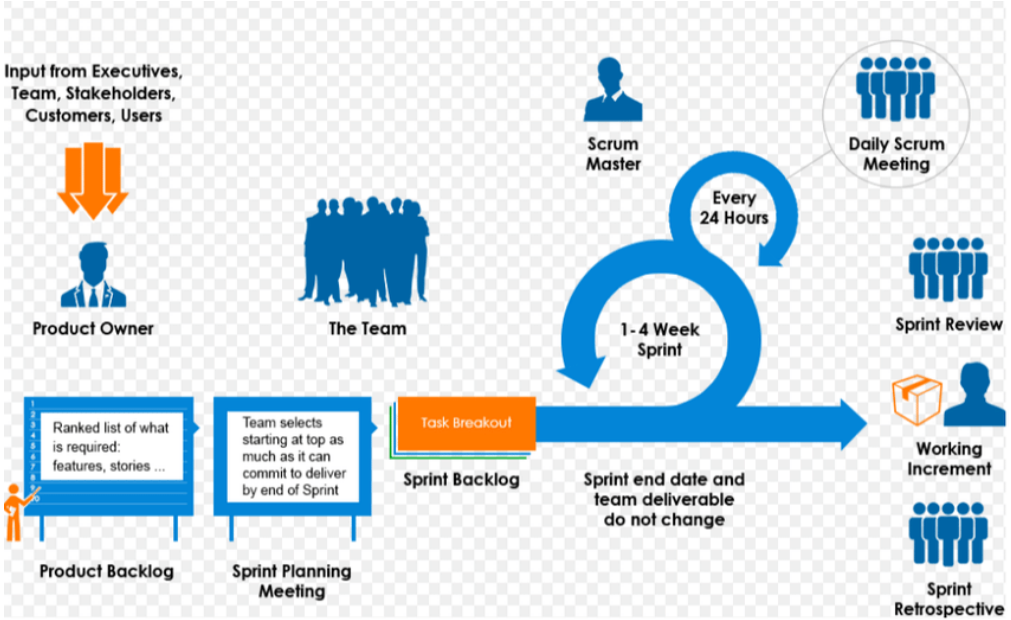
- V programování SCRUM (česky mlýn, skrumáž) je iterativní a inkrementální metodologie agilního vývoje softwaru používaná na řízení produktového vývoje.
- Definuje flexibilní, holistickou strategii produktového vývoje, kde vývojový team pracuje jako jednotka na dosažení společného cíle", zpochybňuje předpoklady "tradičního, sekvenčního přístupu" k vývoji produktu, a umožňuje týmům se samo organizovat podpořením fyzické kolokace nebo blízké online spolupráce všech členů týmu, stejně jako denní ústní komunikaci všech členů týmu a disciplín v projektu.
- Více o SCRUM a jeho součástech: https://www.atlassian.com/agile/scrum

# SCRUM – METODA ŘÍZENÍ AGILNÍHO VÝVOJE

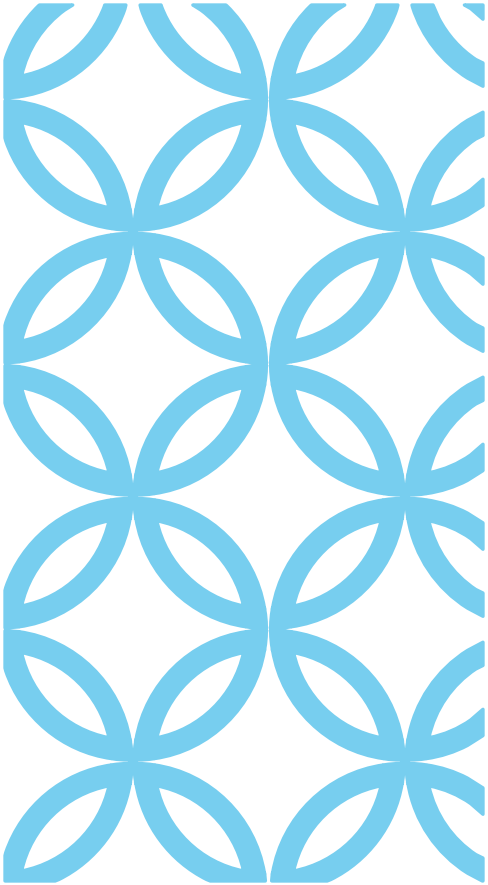
- Klíčový princip SCRUM je jeho pochopení, že během projektu mohou zákazníci změnit názor o tom, co chtějí a potřebují (často zvané "souhrn požadavků") a že nepředvídané úkoly nelze jednoduše řešit tradičním předvídáním a plánováním.
- SCRUM používá empirický přístup, podle kterého problém nelze zcela pochopit nebo definovat a proto se soustředí na maximální schopnost týmu rychle dodat a reagovat na nové požadavky.

# SCRUM – METODA ŘÍZENÍ AGILNÍHO VÝVOJE

- Základem SCRUMu je **Sprint** - periodická činnost (s periodou obvykle uváděnou v týdnech), která je neustále kontrolována (např. každých 24 hod, 48 hod).
- Vstupními parametry Sprintu jsou tzv. **Backlog items** (seznam návrhů na vylepšení stávajícího produktu, příp. chyby) a výstupním parametrem je **Produkt** s novou funkcionalitou, příp. bez zjištěných chyb.
- Více o backlogu - https://www.atlassian.com/agile/scrum/backlogs




# SCRUM PROCES



**Product Owner** - osoba zodpovědná za výsledný produkt, přiřazuje priority jednotlivým Backlog Items (vytváří Selected Backlog).

**SCRUM Master** - osoba, jejíž úkolem je zajistit hladký průběh sprintu, organizovat schůzky, řešit případné administrativní problémy.

**Team** - programátoři.

**Stakeholder(s)** – uživatelé produktu, manažeři, osoby, kterých se problém dotýká, ale produkt (systém či SW) nevytvářejí.

# ROLE VE SCRUM

# ROLE VE SCRUM

- **Vlastník produktu** (Product Owner) reprezentuje zainteresované subjekty a je hlasem zákazníka.
- Je odpovědný za ujištění, že team do byznysu přidá hodnotu.
- Vlastník produktu píše články pro zákazníky (typicky zkušenosti uživatelů), hodnotí a přiřazuje jim priority, a přidává je do produktového Backlogu.
- SCRUM týmy mají mít jednoho vlastníka produktu, tato role se nemá spojit se SCRUM masterem.
- Product Owner má být na obchodní straně projektu, a nikdy nemá interagovat s členy týmu o technických aspektech vývoje.
- Tato role je stejná jako role Customer Representative (reprezentant klientů) v jiných agilních frameworcích.

# ROLE VE SCRUM

- Náplň role **Vlastníka produktu**:
  - Demonstrovat řešeni pro klíčové zainteresované osoby, které nebyly na iteračním demu.
  - Ohlašovat uvedení nových verzí.
  - Komunikovat stav týmu.
  - Organizovat milníkové přehledy.
  - Vzdělávat v procesu vývoje.
  - Dohadovat priority, rozsah, financování a rozvrh.
  - Ujistit se, že produktové testy jsou viditelné, transparentní a jasné.

# ROLE VE SCRUM

- Vývojový tým
  - Je odpovědný za dodání Potenciálně použitelných inkrementů (Potentially Shippable Increments - PSIs) produktu na konci každého sprintu (cíl sprintu).
  - Team je složen z 3-9 jednotlivců, kteří dělají aktuální práci (analýza, design, vývoj, test, technická komunikace, dokument, atd.).
  - Vývojové týmy jsou vícefunkční, se všemi dovednostmi vytvořit produktový inkrement.
  - Vývojový team v SCRUM je sebe organizující, i když tady může být nějaký stupeň interakce s projektovým managementem (Project Management Offices - PMOs).

# ROLE VE SCRUM

**SCRUM master**

SCRUM je usnadněný SCRUM masterem (mistrem), který je odpovědný za odstranění překážek teamu na dodání produktových cílů.

SCRUM master není tradiční team leader nebo projektový manažer, ale koná jako prostředník mezi teamem a jakýmikoli negativními vlivy.

SCRUM master zajišťuje, že SCRUM proces je použit jako bylo plánováno a členové týmu dodržují dohodnuté procesy.

Často organizuje schůzky a povzbuzuje tým k zlepšení. Tato role se někdy označuje jako "team facilitátor„.

# EVENTY (UDÁLOSTI)

**Sprint** (nebo iterace) je základní jednotka vývoje ve SCRUM. Sprint je časově omezená snaha, tedy je omezen na specifický čas. Doba je určena dopředu pro každý sprint a obvykle je to jeden týden až jeden měsíc, nejčastěji 2 týdny.

Každý sprint začíná událostí **plánování sprintu**, cílem události je definovat úkoly sprintu, kde je definována práce sprintu a odhadnut závazek pro cíl sprintu.

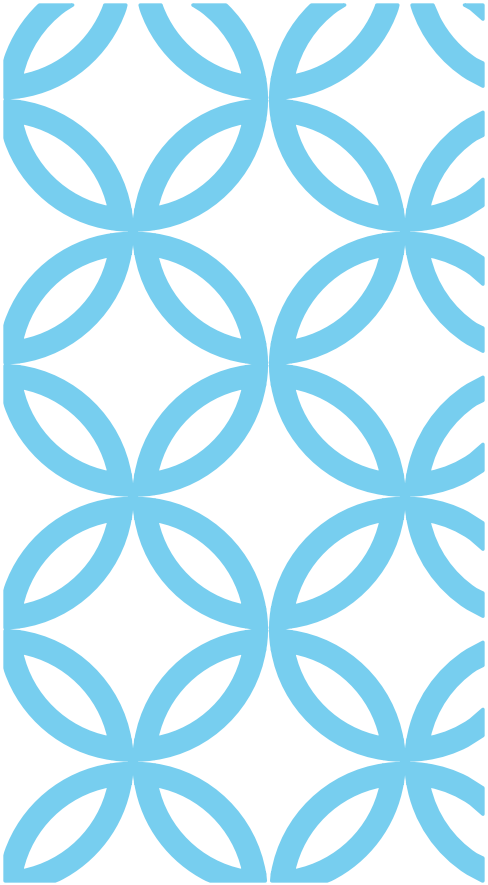
Každý sprint končí **recenzí sprintu** (Sprint Review) a retrospektivou, ve které je reportován progres pro zainteresované osoby a definují se úlohy na zlepšení pro další sprinty.

SCRUM zdůrazňuje zpracovaný produkt na konci sprintu, který je opravdu hotový, v případě software to může znamenat, že software byl integrován, kompletně testován, zdokumentován a potenciálně může být dodán.

# PLÁNOVÁNI SPRINTU

Na začátku sprintu, tým organizuje Sprint Planning Event - plánovací událost, která má následující cíle:

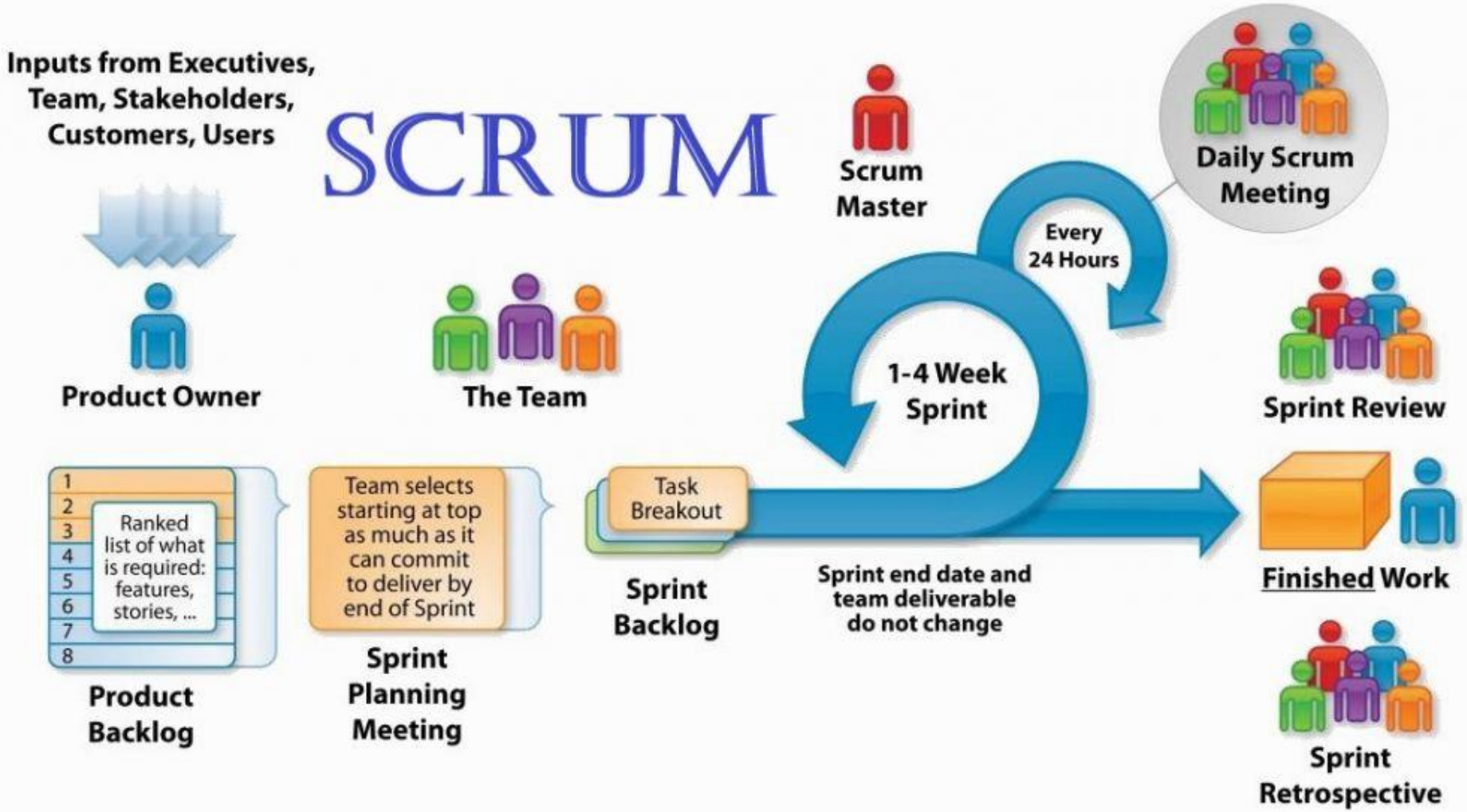
- vybrat jaká práce se udělá
- připravit úkoly sprintu, které určí čas na úkoly pro celý tým
- definovat a diskutovat kolik práce je třeba udělat během sprintu
- 4-hodinový limit pro 2-týdenní sprint, poměrně pro jinou délku
  - v první půli se celý tým (vývojový tým, SCRUM Master a Product Owner) dohodnou, které produktové položky (Backlog) budou v daném sprintu uvažovat
  - v druhé půli vývojový team určí práci (úlohy) potřebné na dodání položek Backlogu, co rezultuje do sprintového backlogu.



# DENNÍ SCRUM

Každý den během sprintu team organizuje Denní SCRUM (nebo Stand-Up) se specifickými zásadami:

- Všichni členové softwarového týmu přijdou připraveni.
- Denní SCRUM začne přesně načas i když někteří členové chybí.
- Denní SCRUM má proběhnout každý den na stejném místě a ve stejný čas.
- Délka denního SCRUM je omezena na 15 minut.
- Každý je vítán, i když obvykle mluví jen teamové role SCRUM.

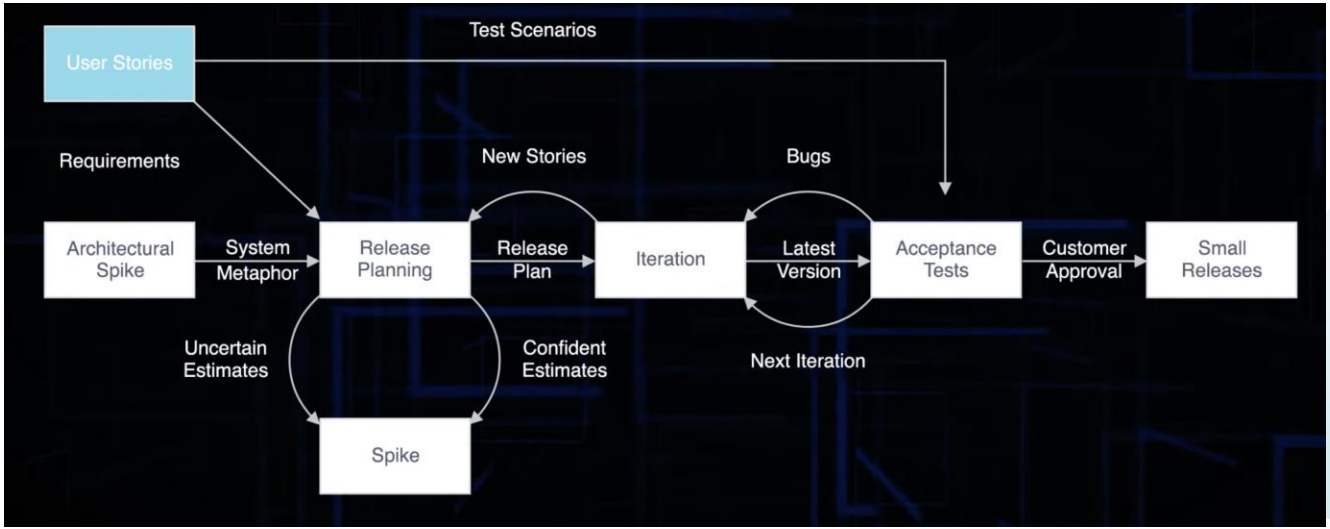

Inputs from Executives, Team, Stakeholders, Customers, Users
SCRUM
Scrum Master
Daily Scrum Meeting
Every 24 Hours
Product Owner
The Team
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Team selects starting at top as much as it can commit to deliver by end of Sprint
Task Breakout
Sprint end date and team deliverable do not change
Finished Work
Product Backlog
Sprint Planning Meeting
Sprint Backlog
Sprint Retrospective

# KANBAN

- Metoda vznikla ve 40. letech 20. století (Taiichi Ohno) v továrnách japonské automobilky Toyota
- Cíl: Zjednodušit a zeštíhlit výrobu a zároveň umožnit lepší vizibilitu do časů v jednotlivých fázích výrobního procesu.
- Více o Kanban: https://www.atlassian.com/agile/kanban
- Rozdíl mezi Agile a Kanban: https://www.guru99.com/agile-vs-kanban.html
  - Totéž, pro audiovizuálně založené: https://www.youtube.com/watch?v=pxxmSLJj8FQ

# EXTRÉMNÍ PROGRAMOVÁNÍ (XP)

- XP je pravděpodobně nejznámější agilní metodikou.
- Propaguje časté dodávky software v krátkých vývojových cyklech.
- Kromě toho navrhuje párové programování, jednotkové testování celého kódu (nejdříve se vytvoří test, až pak samotný kód), programovat jen to, co je v danou chvíli nezbytné, jednoduchý a jasný kód.
- Mezi další praktiky patří společné vlastnictví kódu a neustálý refaktoring.
- Vývojáři by měli počítat se změnami požadavků v průběhu času. Důležitá je častá komunikace se zákazníkem i mezi programátory.
- Více informací k XP: https://www.youtube.com/watch?v=PRYmsmMdlko
- Více informací k XP: https://www.agilealliance.org/glossary/xp

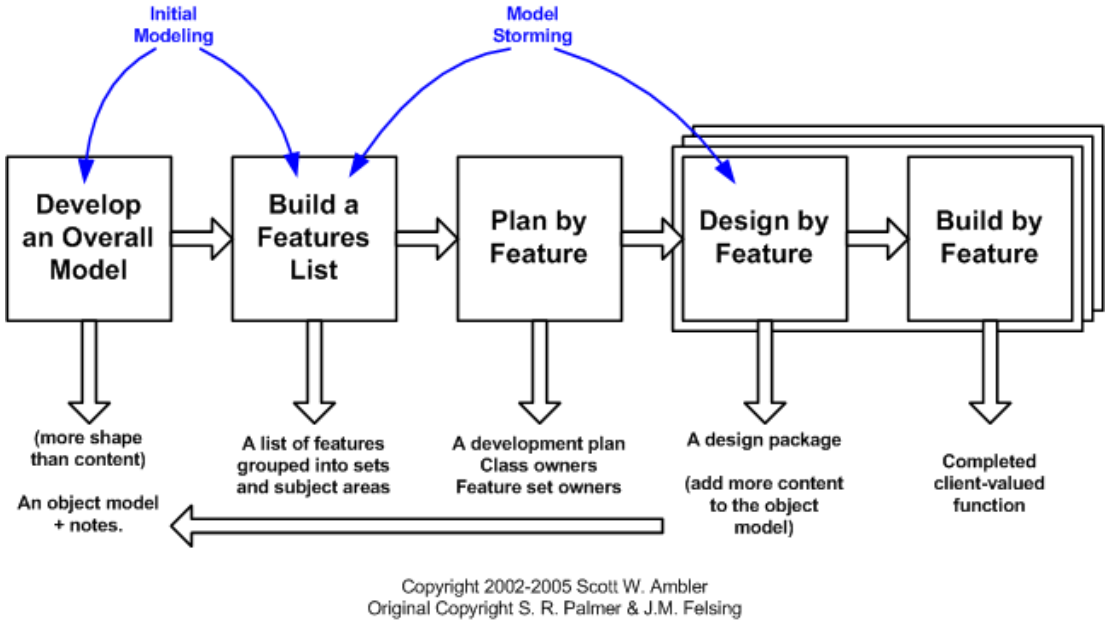




# EXTRÉMNÍ PROGRAMOVÁNÍ (XP)

# VÝVOJ ŘÍZENÝ VLASTNOSTMI (FDD – FEATURE DRIVEN DEVELOPMENT)

- FDD začíná vytvořením doménového modelu popisujícího celý systém.
- Ten se převede do seznamu vlastností (elementární funkcionality, které přináší hodnotu uživateli).
- Vývoj má celkem pět fází (první tři sekvenční, další dvě iterativní) Iterace trvá většinou dva týdny.
- Během každé iterace se implementují konkrétní užitné vlastnosti systému.
- Zákazník průběžně dostává mezivýsledky a nové verze produktu.
- Na rozdíl od XP nebo SCRUM je jednotlivým programátorům práce přidělena – nevybírají si ji sami.
- Více informací k FDD: https://www.youtube.com/watch?v=lYV0q0X64lQ



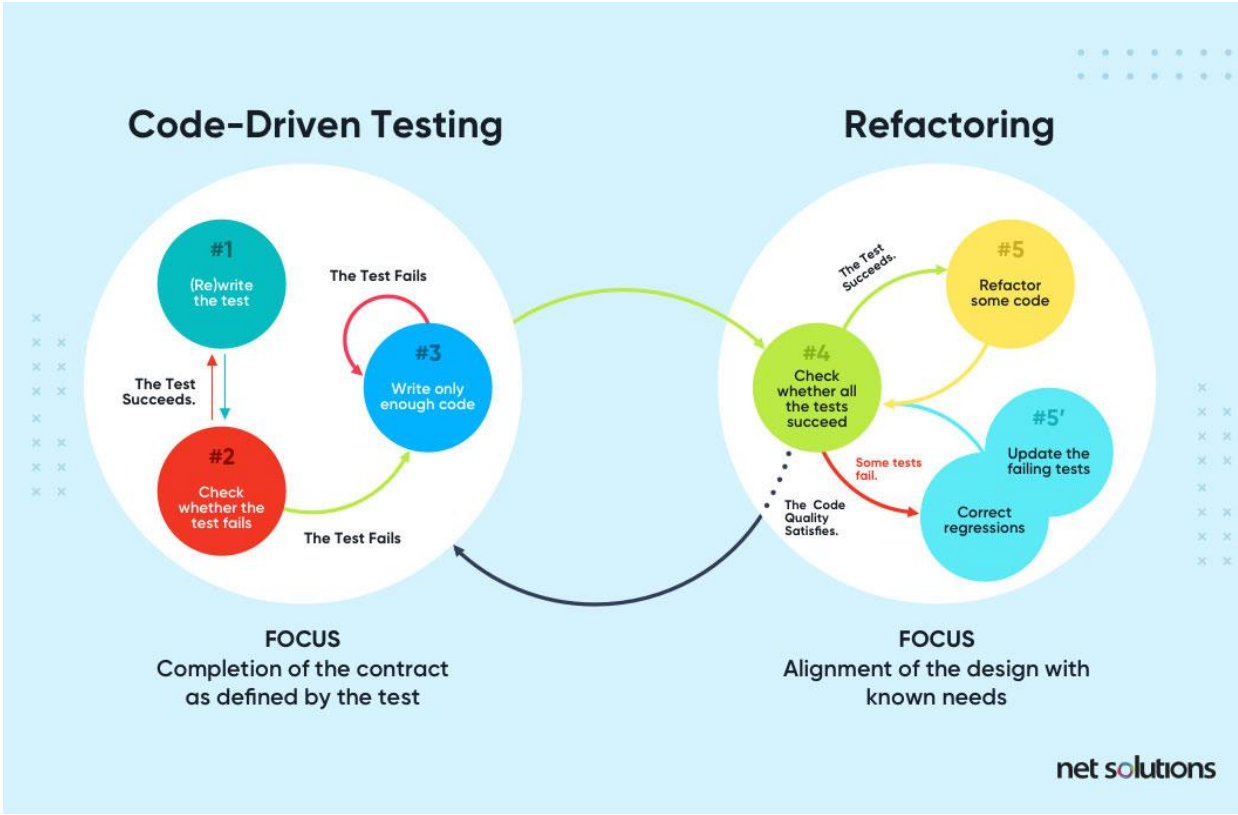

# VÝVOJ ŘÍZENÝ VLASTNOSTMI (FDD – FEATURE DRIVEN DEVELOPMENT)

# VÝVOJ ŘÍZENÝ TESTY (TDD – TEST DRIVEN DEVELOPMENT)

- TDD navrhuje psaní testů před samotným kódem a následně naprogramovat samotný kód.
- Implementuje se přesně takové množství kódu, jaké dokáže projít testem.

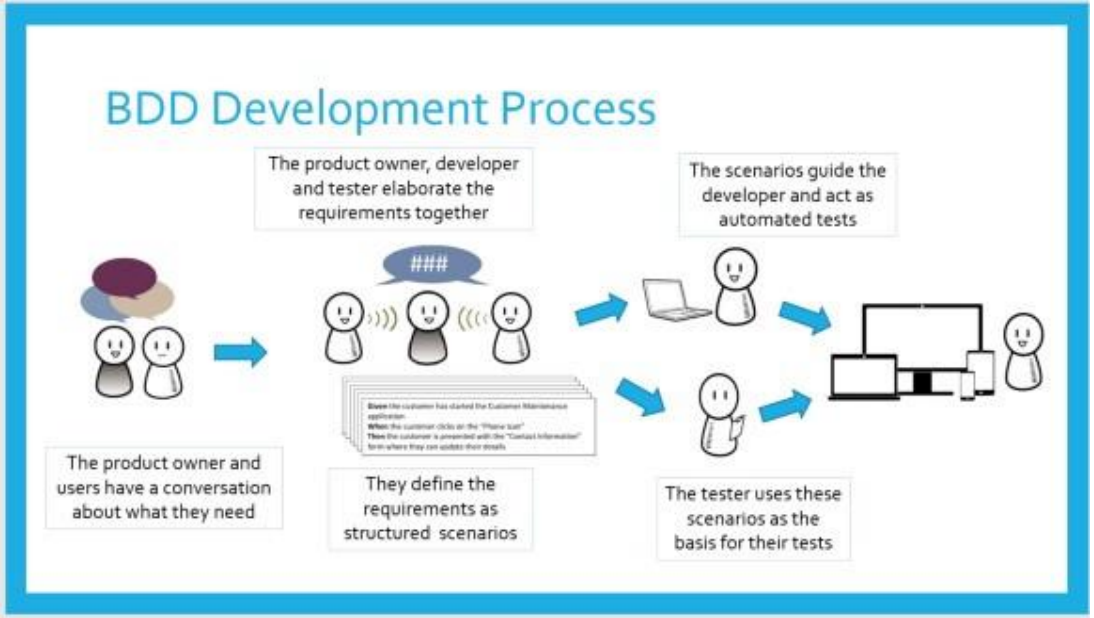
- Více informací k tématu: https://www.youtube.com/watch?v=uGaNkTahrIw

# VÝVOJ ŘÍZENÝ TESTY (TDD – TEST DRIVEN DEVELOPMENT)



# VÝVOJ ŘÍZENÝ CHOVÁNÍM (BDD - BEHAVIOR-DRIVEN DEVELOPMENT)

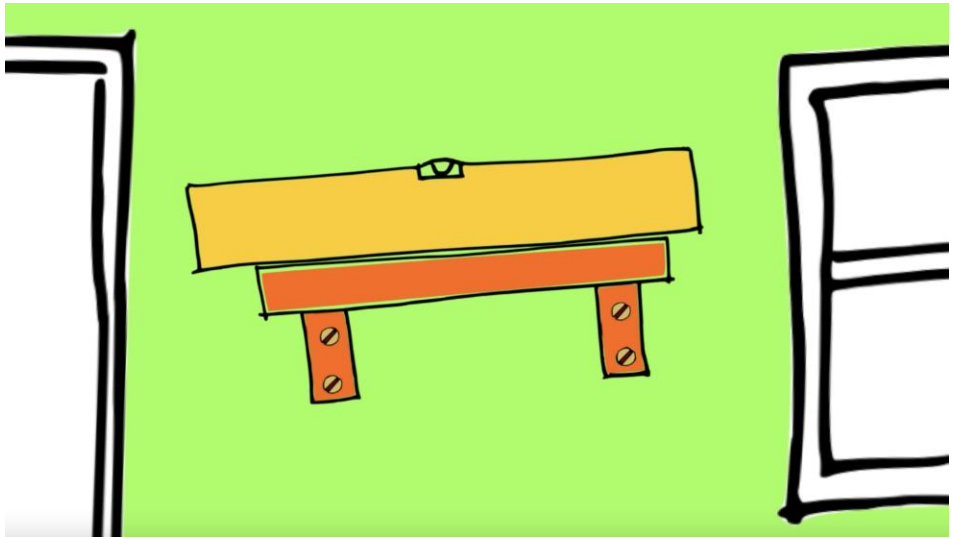
- Metoda částečně doplňuje TDD
- Důraz je na specifikaci chování systému prostřednictvím přirozeného jazyka a následné převedení do "řeči IT"




# VÝVOJ ŘÍZENÝ CHOVÁNÍM (BDD - BEHAVIOR-DRIVEN DEVELOPMENT)

# ROZDÍL MEZI WATERFALL, TDD A BDD

- https://www.youtube.com/watch?v=4QFYTQy47yA



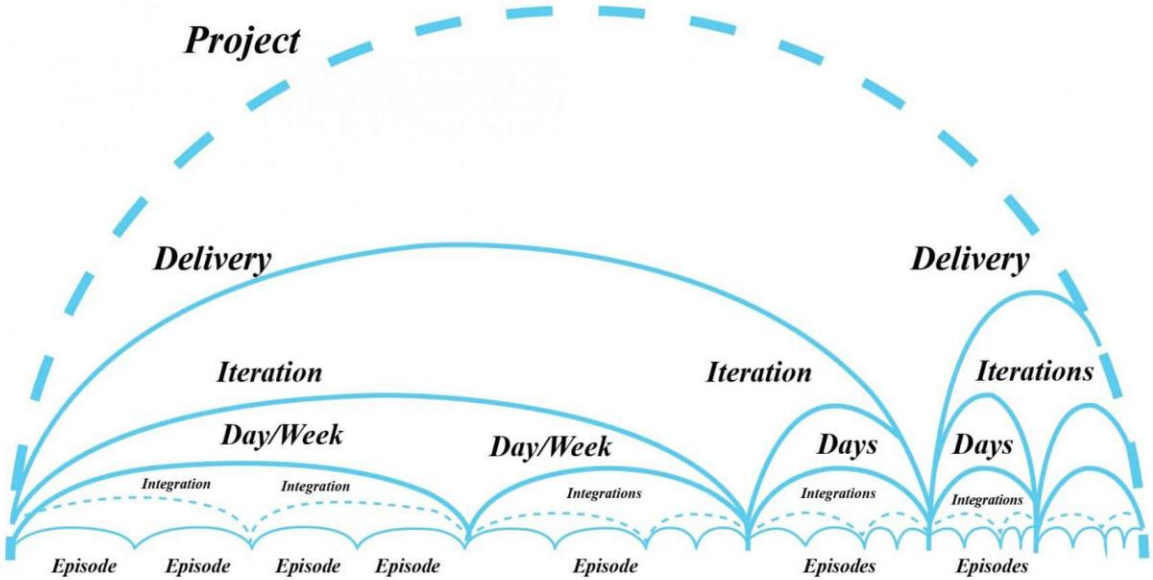
# CRYSTAL METODIKY (CRYSTAL METHOD)

- Vytv. v 90. letech v IBM prostředí
- Nejde jen o jednu metodiku.
- Důraz na lidi a jejich spolupráci spíše než na nástroje a procesy
- Hlavní myšlenkou je to, že je lepší metodiku přizpůsobit danému projektu, žádná metodika nebude vyhovovat každému projektu.
- Vytvoření individuální a účelové metodiky pro konkrétní projekt je první fází vývoje.
- Pro vytvořenou metodiku je rozhodující například velikost projektu a vývojového týmu.

# CRYSTAL METODIKY (CRYSTAL METHOD)

- Dále se dělí na další metodiky:


- Základní rozdíly:
  - Velikost týmu
  - Prioritizace
  - Vážnost (Criticality)

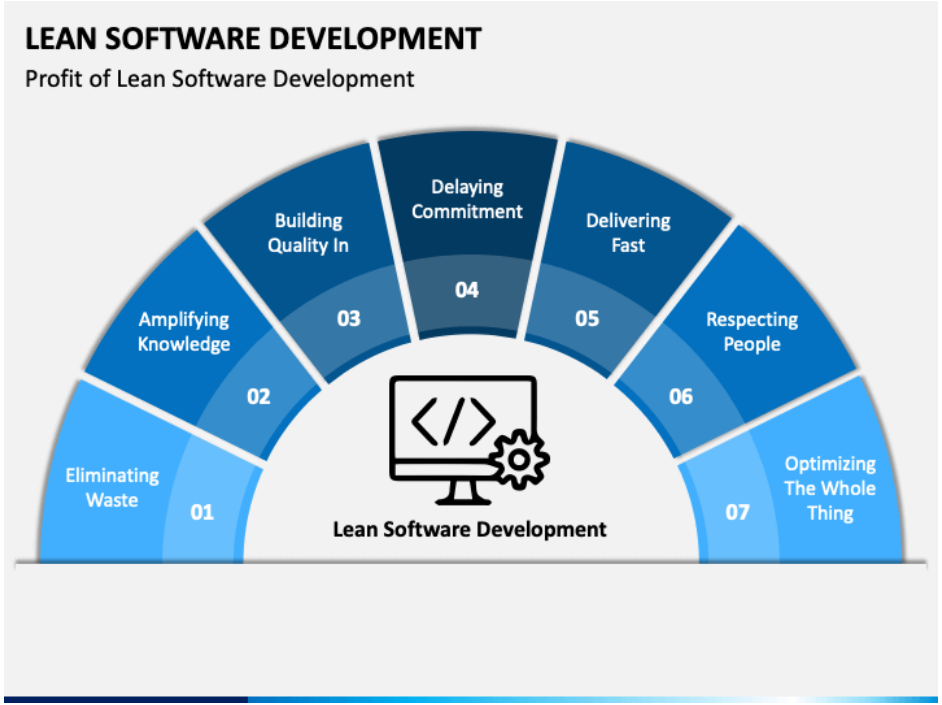



# CRYSTAL METODIKY (CRYSTAL METHOD)

Více ke studiu zde: https://www.toolsqa.com/agile/crystal-method/ a zde: https://www.youtube.com/watch?v=8M-iNHErYvw

# LEAN SOFTWARE DEVELOPMENT

- Lean Development je spíš než metodikou souhrnem pravidel, jejichž používání by mělo zefektivnit a zrychlit vývojový proces.
- Tato pravidla zní: eliminovat zbytečné (to, co nepřináší zákazníkovi žádnou hodnotu), zdůraznit proces učení, rozhodovat se tak rychle a pozdě, jak je možné, posílit odpovědnost týmu, zabudovat integritu a vidět systém jako celek.

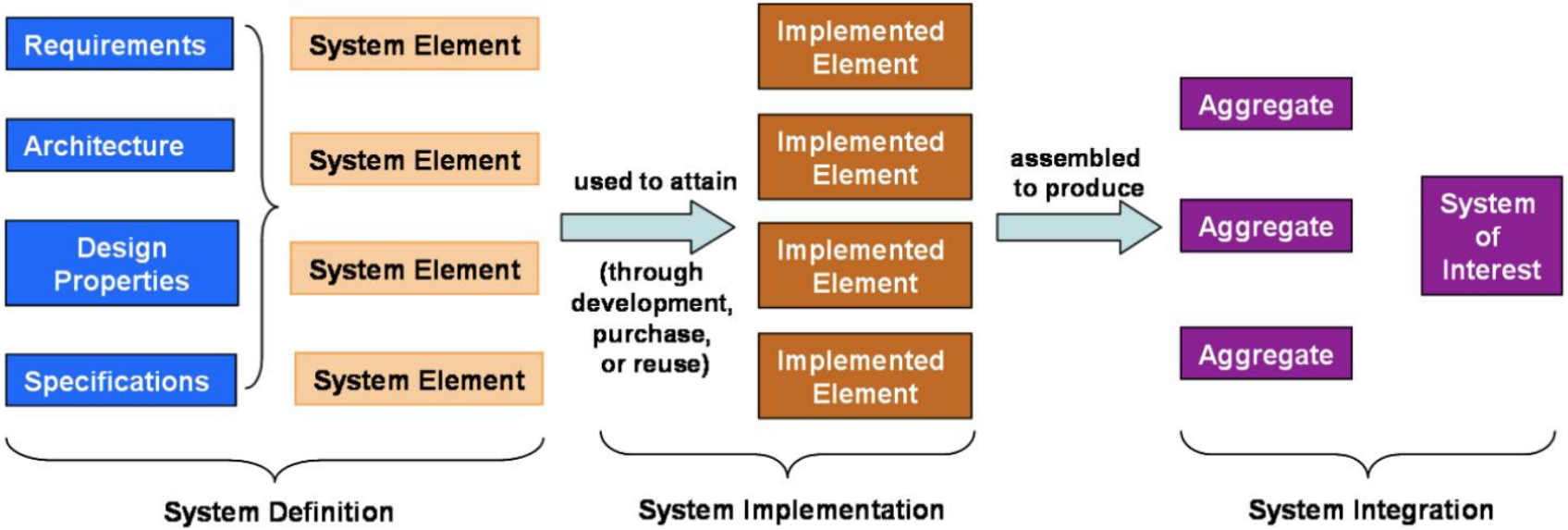



# LEAN SOFTWARE DEVELOPMENT

# LEAN SOFTWARE DEVELOPMENT

7 Wastes in software development

1. Partially done work
2. Extra features
3. Relearning
4. Task switching
5. Waiting
6. Handoffs
7. Defects
8. Management activities




# GUIDE TO THE SYSTEMS ENGINEERING BODY OF KNOWLEDGE (SEBOK)

# GUIDE TO THE SYSTEMS ENGINEERING BODY OF KNOWLEDGE (SEBOK)

- Part 1 SEBoK Introduction
- Part 2 Foundations of Systems Engineering
- Part 3 Systems Engineering and Management
- Part 4 Applications of Systems Engineering
- Part 5 Enabling Systems Engineering
- Part 6 Related Disciplines
- Part 7 Systems Engineering Implementation Examples
- Part 8 Emerging Knowledge

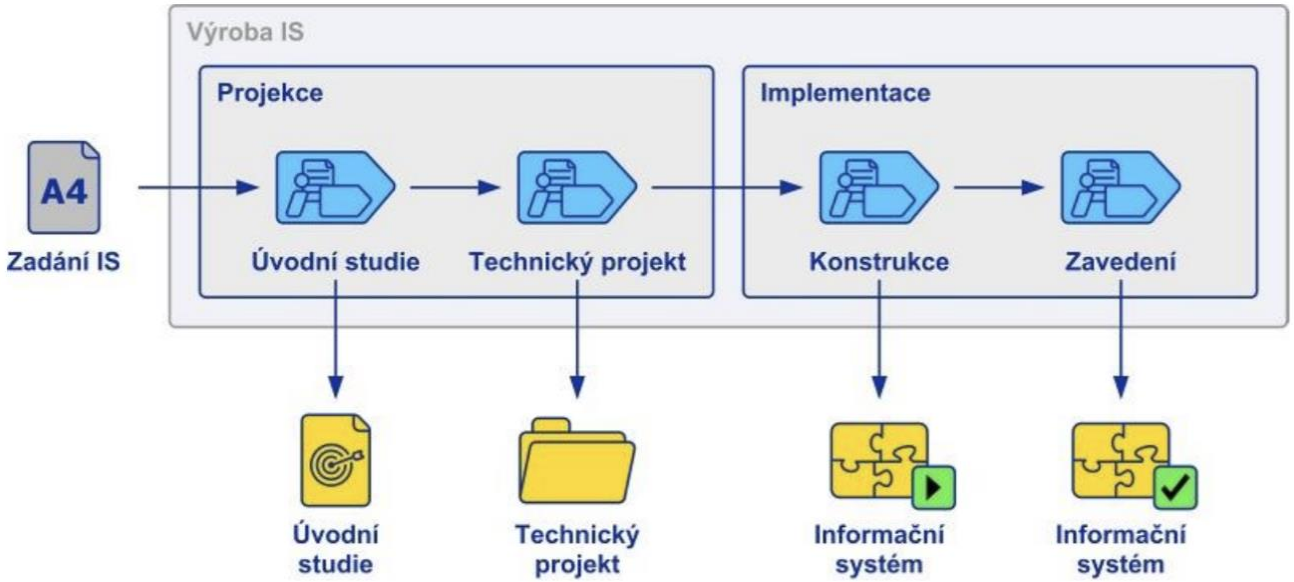
# BACK TO BASICS

# IMPLEMENTACE IS

- Výběr vhodného informačního systému zahrnuje řadu kroků; výchozím bodem je nepochybně analýza situace organizace (firmy), kde má být nasazen, i jejího předpokládaného vývoje.

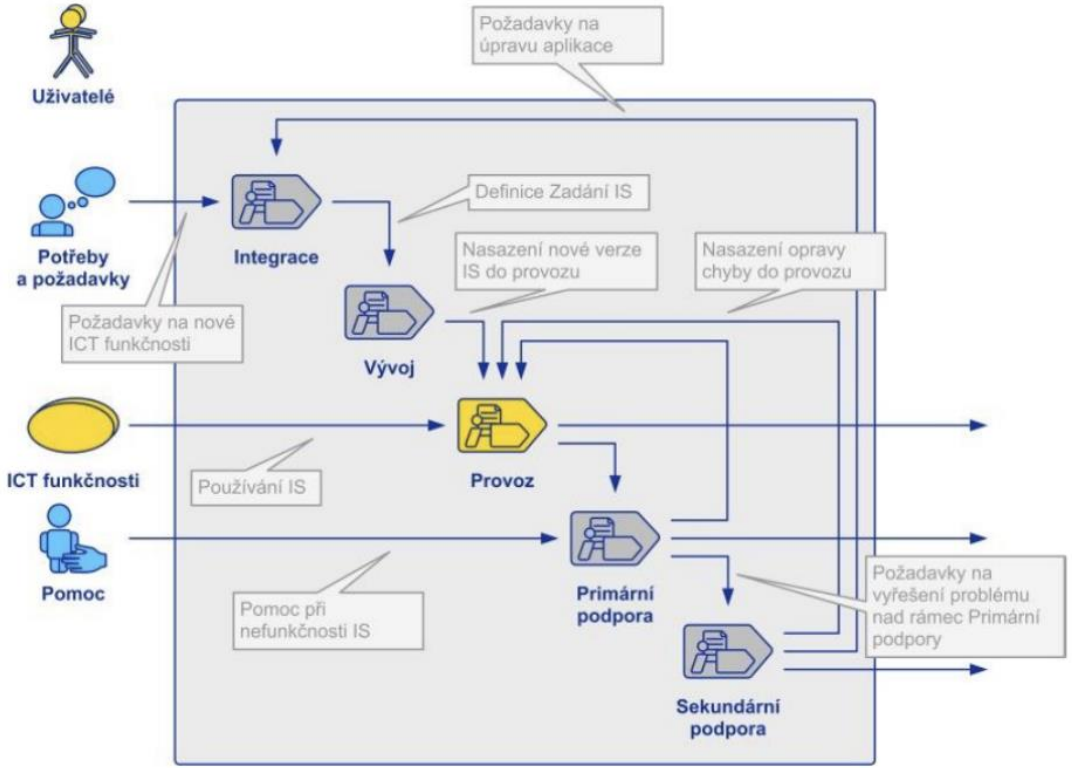
# PROCES IMPLEMENTACE

- **Schůzka a analýza** - veškeré procesy ve firmě a důkladně je popsat. Na základě toho vzniká úvodní analýza, která zahrnuje popis nejen toho, jak je to teď, ale jsou tam zahrnuty i klientovy požadavky na to, jak by to mělo být v budoucnu.
- **Sestavení systému** - řešení na míru, které zahrnuje podrobný popis toho, jak to bude v budoucnu fungovat včetně přístupových práv, převodů dat a dalších a dalších věcí.
- **Nasazení a testování** - možnost zjistit, které procesy jsou nadbytečné anebo která data duplicitní a na základě toho něco zjednodušit či pročistit.




# PROCES VÝROBY INFORMAČNÍHO SYSTÉMU

# PROCESY EFEKTIVNÍHO FUNGOVÁNÍ IS



# IMPLEMENTACE IS

- Úvahy o implementaci systému zahrnují následující oblasti:
  - funkcionalitu systému
  - rozšiřitelnost
  - přístupy k rozhraní IS (konvertibilita dat)
  - vývoj systému
  - poskytované služby dodavatele IS
  - odvětvová IS
  - konkurence – široká nabídka produktů
  - provoz vlastními silami nebo outsourcing?

# PREFERENCE DLE VELIKOSTI ORGANIZACE

- Menší firmy kladou důraz na rychlost a snadnost pořizování dat na úkor šíře, komplexnosti a kvality.
- Čím menší firma, tím nižší nároky na zpětné vyhodnocení dat, naproti tomu má vyšší nároky na rychlost a jednoduchost pořízení vstupních dat. To je v rozporu s původním požadavkem na kvalitu, šíři a komplexnost dat.
- Proto menší firmy nedávají přednost komplexním velkým balíkům, kde je důraz na komplexnost, kvalitu a špičkové analytické informace; pro malou firmu se implementace takového balíku může stát komplikací.

# MODULY IS

- **Zaměstnanci:** nábor, docházkový systém (sledování přesčasů, nastavení přístupových práv), výkazy práce, mzdy, zaměstnanecké výhody, školení, sledování výkonu, sledování výdajů zaměstnance, hodnocení zaměstnanců, sledování kariéry, přehled know-how zaměstnance, porady, samoobslužný informační portál pro zaměstnance, personální plánování, výkazy potřebné pro státní instituce…
- **Dodavatelé a nákup:** přehled nákupů a dodavatelů, přehled komunikace, sdílení dokumentů s dodavateli, hodnocení nabídek, hodnocení dodavatelů, kombinování zdrojů (různých dodavatelů), objednávky (vytváření, schvalování, sledování, elektronické zaslání dodavateli)…

# MODULY IS

- **Logistika**: doprava (plánování, objednávání u dodavatelů, sledování vlastních vozů, evidence a provozní deníky, knihy jízd, komunikace s čerpacími stanicemi, silniční daň), sklady (evidence zásob, správa skladovacích míst, balení, operace příjmu a výdeje, podpora čárových kódů a RFID, automatické generování objednávek), celnice (celní sklady, celní režimy)...
- **Výroba**: tvorba prognóz, plánování (se zřetelem např. na kapacitu pracovníků a strojů, dostupnost nástrojů, materiálu a komponent, kapacitu skladů, externí kooperace apod.), správa technických podkladů (popisy výrobků, výkresy, postupy), podpora výroby/úpravy na zakázku, projektové výroby a výroby na sklad, řízení a synchronizace výrobních procesů i v různých lokalitách, konfigurátory výrobků, kalkulace, sledování průběhu výroby, řízení jakosti, údržba výrobních kapacit...

# MODULY IS

- **Projekty**: projektová dokumentace, řízení projektů – termíny, činnosti, zdroje, subdodávky, sledování vytížení/volné kapacity zdrojů, sledování postupu projektu, finanční řízení projektu, řízení rizik, sledování projektů ve více firmách…
- **Prodej**: distribuční systém, maloobchod (propagace, doplňování zboží, analýza prodeje, pokladní terminály), e-shop, mobilní prodej, prodejní dokumenty, cenové kalkulace/slevy, rezervace, přehled nabídek, sledování prodejních týmů, sledování servisních smluv…

# MODULY IS

- **Marketing**: segmentace trhu, marketingové akce (a analýza akcí), direct mailing, podpora tvorby katalogů produktů, sledování konkurence, analýza příležitostí...
- **Zákazníci**: analýza chování zákazníků (spokojenost, potenciál pro nákup dalších produktů), získávání zákazníků, podpora marketingu, plánování/sledování kontaktů s klienty, správa odpovídajících dokumentů, kontaktní centrum, servis...
- **Účetnictví**: vnitropodnikové, daňové, faktury, celní deklarace, DPH, Intrastat (celní správa), cizí měny, přístup k internet bankingu, tisk platebních poukázek...

# MODULY IS

- **Majetek:** krátkodobý a dlouhodobý, umístění a inventarizace majetku (včetně podpory čárových kódů), odpisy, analýzy…
- **Správa dokumentů:** příjem (v elektronické i papírové podobě/skenování) a archivace dokumentů, vyhledávání, možnost opatřit papírové dokumenty čárovými kódy, správa oficiálních šablon dokumentů…

# SPECIALIZOVANÉ MODULY IS

- Oborová řešení často zahrnují modely řešící specifické požadavky různých odvětví podnikání, např.:
  - správa IT (správa událostí, správa konfigurací, řešení problémů, řízení změn),
  - správa portfolia projektů (analýza, zajišťování zdrojů, synchronizace),
  - řízení shody a rizik (audit souladu s právními rámci a určenými standardy, analýza externích rizik, analýza bezpečnosti dat),
  - komunikace s dalším softwarem (EDI, propojení s emailovým systémem, s kancelářským balíkem), konstrukční systémy CAD, elektronické publikační systémy..

# FAKTORY OVLIVŇUJÍCÍ CENU IMPLEMENTACE ERP

- Počet uživatelů systému
- Složitost podnikových procesů
- Nevhodná „velikost“ systému
- Nekvalitní přípravná (analytická) fáze
- Špatná součinnost pracovníků zákazníka
- Neochota se změnit a používat standardní procesy
- Nedostatečně zkušení pracovníci dodavatele

# FAKTORY OVLIVŇUJÍCÍ CENU IMPLEMENTACE ERP

- Programové úpravy
- Nedostatečné školení uživatelů
- Školení koncových uživatelů nedostatečně vyškolenými klíčovými uživateli
- Nutnost integrace ERP s jinými systémy
- Organizační a další změny během projektu, nejasná strategie firmy
- Objednání modulů či funkčností, které pak nebudou využity
- Posun termínu spuštění

# CO OVLIVŇUJE CENU SYSTÉMU PŘI PROVOZU?

# KRITÉRIA VÝBĚRU ERP

- Jaké ERP potřebuji? (výrobní společnost)
- Jaké funkční řešení požaduji? (potravinářská výroba, zemědělská výroba)
- Jaké funkce má ERP systém mít? (moduly)
- Jakou společnost na implementaci preferujete? (výrobce, lokální partner, certifikovaný partner)
- Bude ERP určeno pro jednu nebo více společností? (jedna firma, více subjektů)

# KRITÉRIA VÝBĚRU ERP

6. Jsou jiné země mimo ČR, kde budeme ERP používat?

7. Jak chcete provozovat ERP? (cloud, vlastní server)

8. Jaký máme obrat (25 - 150 mil Kč)

9. Kolik uživatelů s ERP bude pracovat? (11 - 25)

10. Jaký rozpočet na projekt ERP mám? (do 50 tis. Kč, do 250 tis. Kč, do 500 tis Kč, atd.)

11. Požaduji napojení na stávající systém? (na jaký)

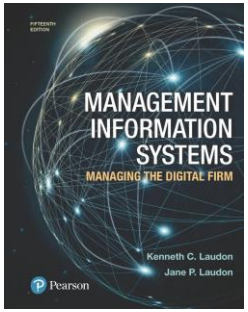

That's all Folks!

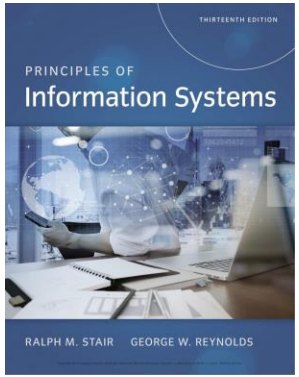
# CO KE ZKOUŠCE?

- Syllabus a obsah předmětu:
  1. Úvod do informačních systémů.
  2. Důvody potřeby IS pro řízení, jejich cíle.
  3. Tvorba informačních systémů. Životní cyklus IS.
  4. Analýza a návrh systému
  5. UML
  6. Příklady IS
  7. Podnikové informační systémy
  8. Aplikace prostředků CASE
  9. Řízení projektů
  10. Úvod do databázových systémů
  11. Analytické nástroje. OLAP – Online Analytical Processing. Data mining. Aplikace pro databáze.

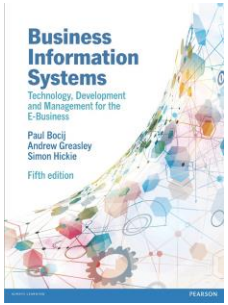
# CO KE ZKOUŠCE?

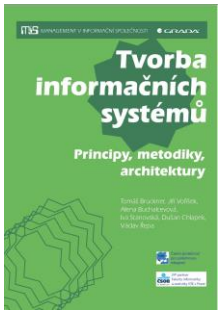
- Doporučená literatura:








# DOPORUČENÁ LITERATURA:


- LAUDON, Kenneth C. a Jane Price LAUDON. *Management information systems: managing the digital firm / Kenneth C. Laudon, Jane P. Laudon*. 2022. ISBN 9781292403281. https://ezproxy.muni.cz/login?url=https://search.ebscohost.com/login.aspx?authtype=ip&custid=s8431878&lang=cs&profile=eds&direct=true&db=cat02515a&AN=muc.MUB01006490575
- STAIR, Ralph M. et al. *Principles of information systems*. Fourteenth. Singapore;United States;Brazil;Mexico;United Kingdom;Australia;: CENGAGE, 2021. ISBN 9780357112410;0357112415
- PILONE, Dan; MILES, Russ. Head First Software Development : A Brain-Friendly Guide. 1 edition. Sebastopol (California) : O'Reilly Media, 2008. 496 s. ISBN 978-0596527358.
- BRUCKNER, Tomáš, Jiří VOŘÍŠEK, Alena BUCHALCEVOVÁ, Iva STANOVSKÁ, Dušan CHLAPEK a Václav ŘEPA. *Tvorba informačních systémů: principy, metodiky, architektury / Tomáš Bruckner, Jiří Voříšek, Alena Buchalcevová, Iva Stanovská, Dušan Chlapek, Václav Řepa*. 2012. ISBN 9788024741536.
- KANISOVÁ, Hana a Miroslav MÜLLER. *UML srozumitelně / Hana Kanisová, Miroslav Müller*. 2007. ISBN 8025110834.

# HODNĚ ŠTĚSTÍ U ZKOUŠEK!