

ŽIVOTNÍ CYKLUS VÝVOJE SW

Jaroslav Žáček
jaroslav.zacek@osu.cz
<http://www1.osu.cz/~zacek/>

PROČ POTŘEBUJEME DEFINOVAT PROCES VÝVOJE

- Při vývoji SW nemáme tvrdá fakta, jako v jiných vědách (fyzika, chemie, ...), proto musíme stavět na zkušenostech.
- Proces říká co, jak a kdy dělat při vývoji SW
 - Ověřené činnosti (best practices)
 - Role, aktivity, artefakty, techniky, nástroje
- Jaké výstupy produkovat (doc, modely, kód, testy)



Vývoj Software

```
graph TD; A[Vývoj Software] --> B[Vývoj bez přímé spolupráce s budoucími uživateli pro hromadný prodej OS]; A --> C[Specifikace požadavků ve spolupráci s budoucími uživateli IS]; C --> D[Vývoj na míru]; C --> E[Customizace]; D --> F[Transformace úspěšné verze na konfigurovatelný SW]; F --> E;
```

Vývoj bez přímé spolupráce s budoucími uživateli pro hromadný prodej (OS)

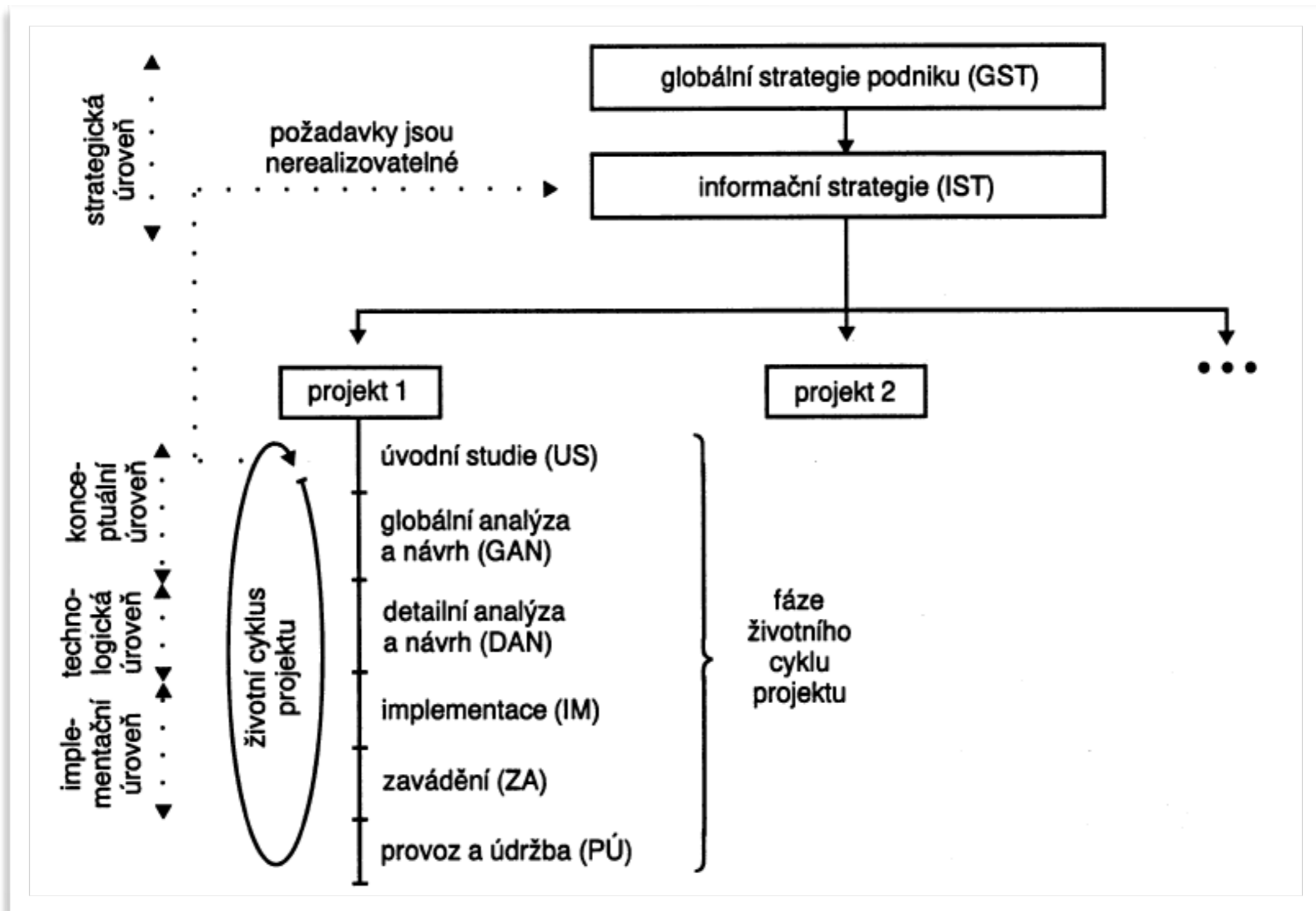
Specifikace požadavků ve spolupráci s budoucími uživateli (IS)

Vývoj na míru

Customizace

Transformace úspěšné verze na konfigurovatelný SW

GST/IST/PROJEKTY



INFORMAČNÍ STRATEGIE

- Vychází z globální podnikové strategie, plánů a zjištěných nedostatků v informační podpoře,
- IS podporují strategické cíle a záměry organizace,
- analýza stavu informačních systémů, stavu IT, vývojových trendů,
- určují se architektury IS (funkční, datová, technologická),
- analýza dopadů IS na org. strukturu firmy (zaměstnanci, kvalifikace), ekonomické a technologické dopady nákupu/vytvoření IS.
- vypracování projektů – obsah, harmonogram, ...

ÚVODNÍ STUDIE

- Posouzení realizovatelnosti jednoho vybraného systému,
- určení, zda lze dosáhnout očekávaných výsledků,
- rozhodnutí, zda ve vývoji pokračovat či nikoliv,
- odhad nákladů, přínosů, hrubý návrh funkcí, vstupů, výstupů, datového modelu systému, vymezení procesů a hranic systému, hrubý návrh technologického řešení – výběr ASW, ZSW a HW.

ANALÝZA A NÁVRH – GLOBÁLNÍ

- Zpodrobnění základních požadavků, rozdělení na podsystémy a vymezení subprojektů,
- návrh hrubého modelu funkcí a dat pro každý subsystém a návrh rozhraní systému.
- úplná specifikace všech hlavních funkčních požadavků, datových, snaha nalézt odvozené požadavky.
- určení priority všech požadavků, struktura systému.
- východisko:
 - cíle IS/IT organizace a schválený plán jejich vývoje.

ANALÝZA A NÁVRH – DETAILNÍ

- analýza, definice požadavků a návrh systému až na úroveň, kdy je možné začít navržený systém implementovat.
- zpodrobňujeme funkce, požadavky a modely z předchozí fáze Globální analýzy a návrhu.
- detailní návrh se týká technologické architektury, datové základny, výstupů systému a také organizační struktury.
- možnost prototypu systému (např. návrh UI).
 - určení dat, kterých se to týká,
 - realizace
 - nezbytné ověření prototypu

IMPLEMENTACE

- Vytvoření fungujícího systému, který realizuje návrh vytvořený v předchozích etapách, realizace v daném implementačním prostředí.
- Systém musí bezchybně fungovat a musí mít implementovány všechny stanovené požadavky.
- Vytvořená uživatelská a provozní dokumentace a popis pracovních procesů.

TESTOVÁNÍ

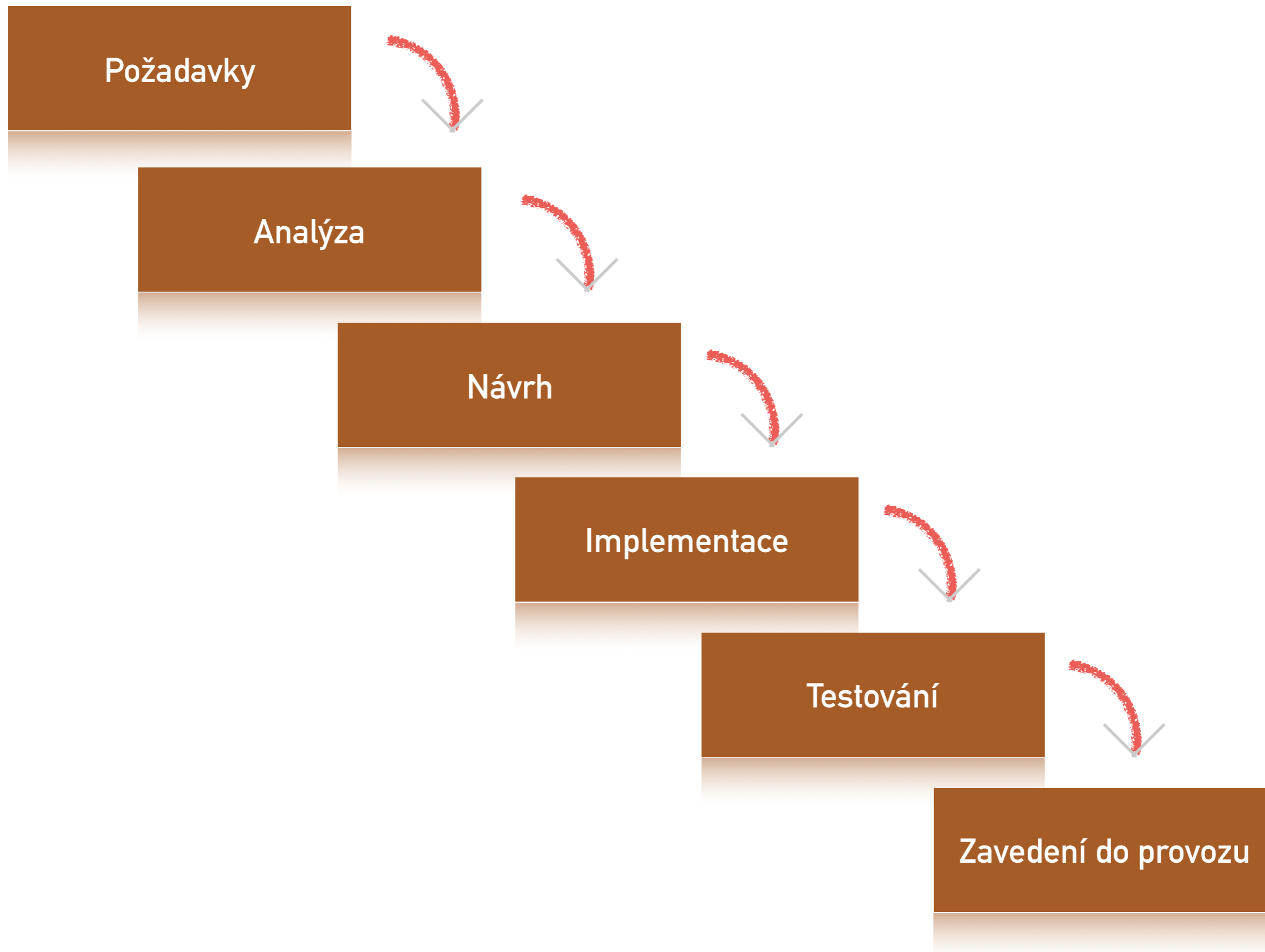
- Zajišťujeme kvalitu produktu...
 - ...vytvořením unitových testů.
 - ...vytvořením testovacích skriptů pro databázi.
 - ...zaváděním nástrojů pro automatizované testování.
 - ...komplexním otestování celého systému z pohledu programátora/zákazníka.

ZAVÁDĚNÍ DO PROVOZU

- instalace technického a programového vybavení,
- školení uživatelů – vedoucích pracovníků, administrátorů systému a běžných uživatelů,
- vytvoření a úpravy databáze, integrační a zátěžové testy.
- přechod na nový systém nesmí neomezovat běžný pracovní režim uživatelů a je také třeba zajistit, aby měli uživatelé čas si na nový systém zvyknout.
- počáteční podpora systému, která zahrnuje pomoc uživatelům, sledování zkušebního provozu a opravy chyb.

PROVOZ, ÚDRŽBA, PODPORA

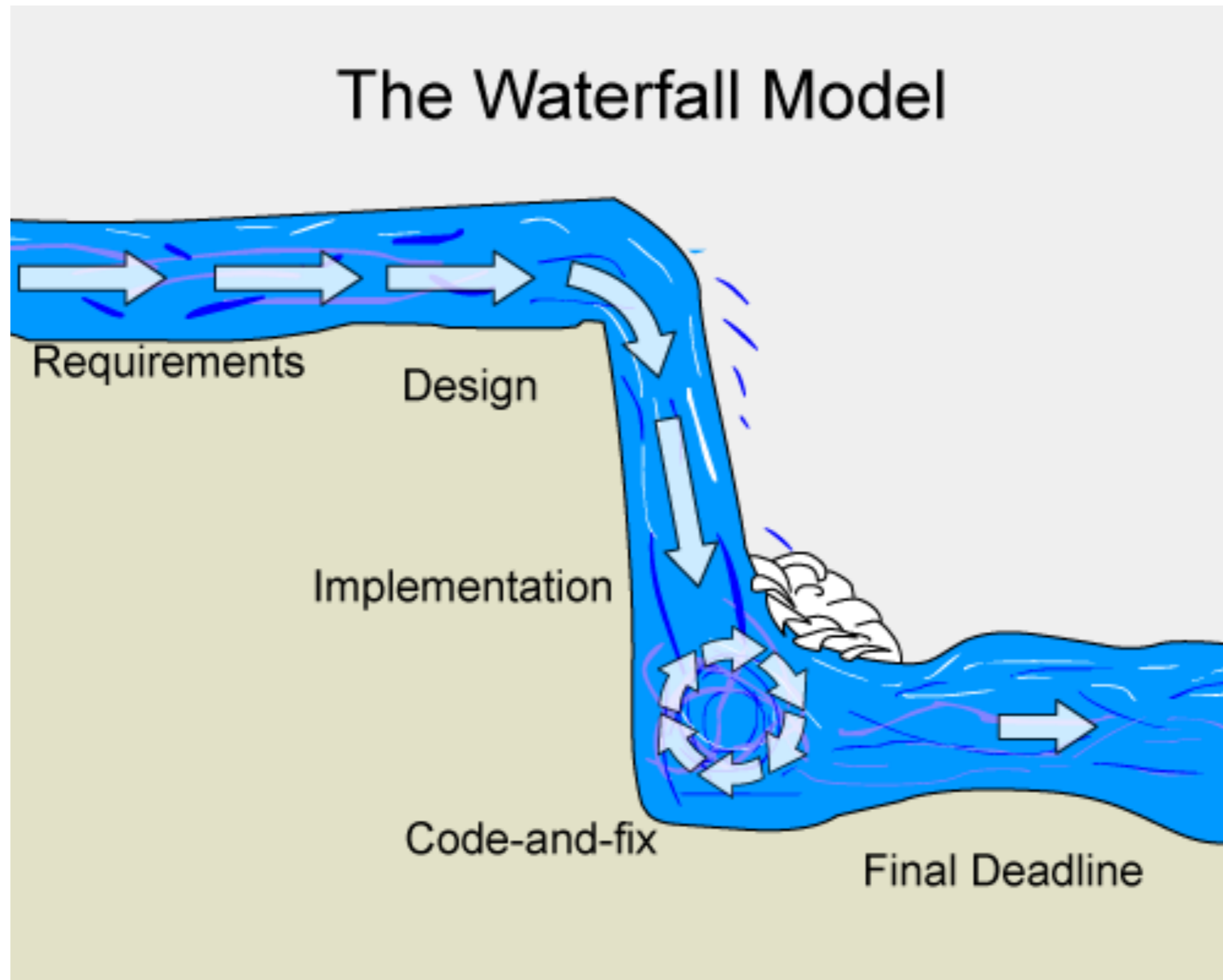
- zajištění provozu systému, jeho údržbu a rozvoj vzhledem k novým uživatelským požadavkům, které jsou v souladu se záměry a cíli organizace.
- monitorování provozu z důvodu optimalizace procesů, zjištění využití aplikací a provozních chyb.
- návrhy změn a úprav mohou být funkčního, provozního nebo organizačního charakteru.



ISO/IEC 12207



PROBLÉMY VODOPÁDU

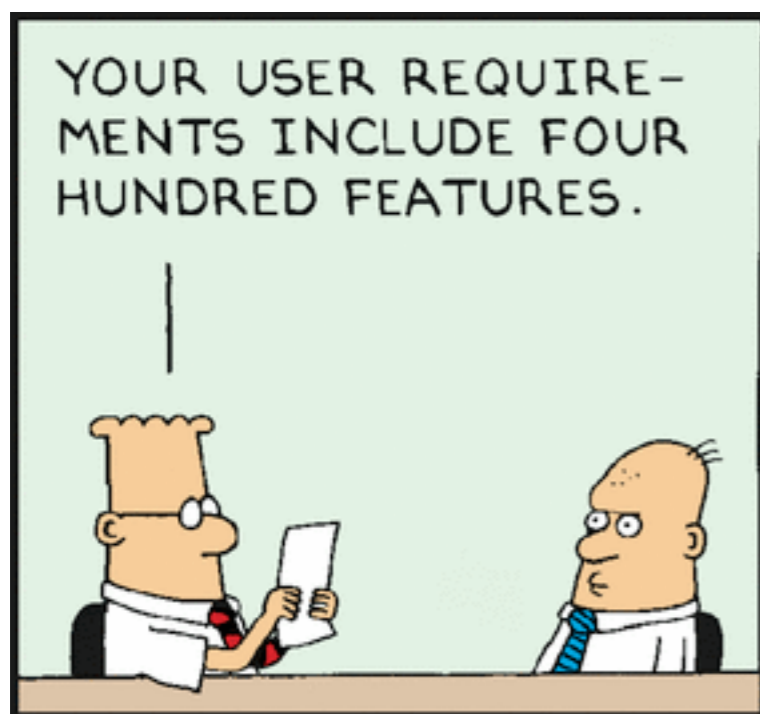
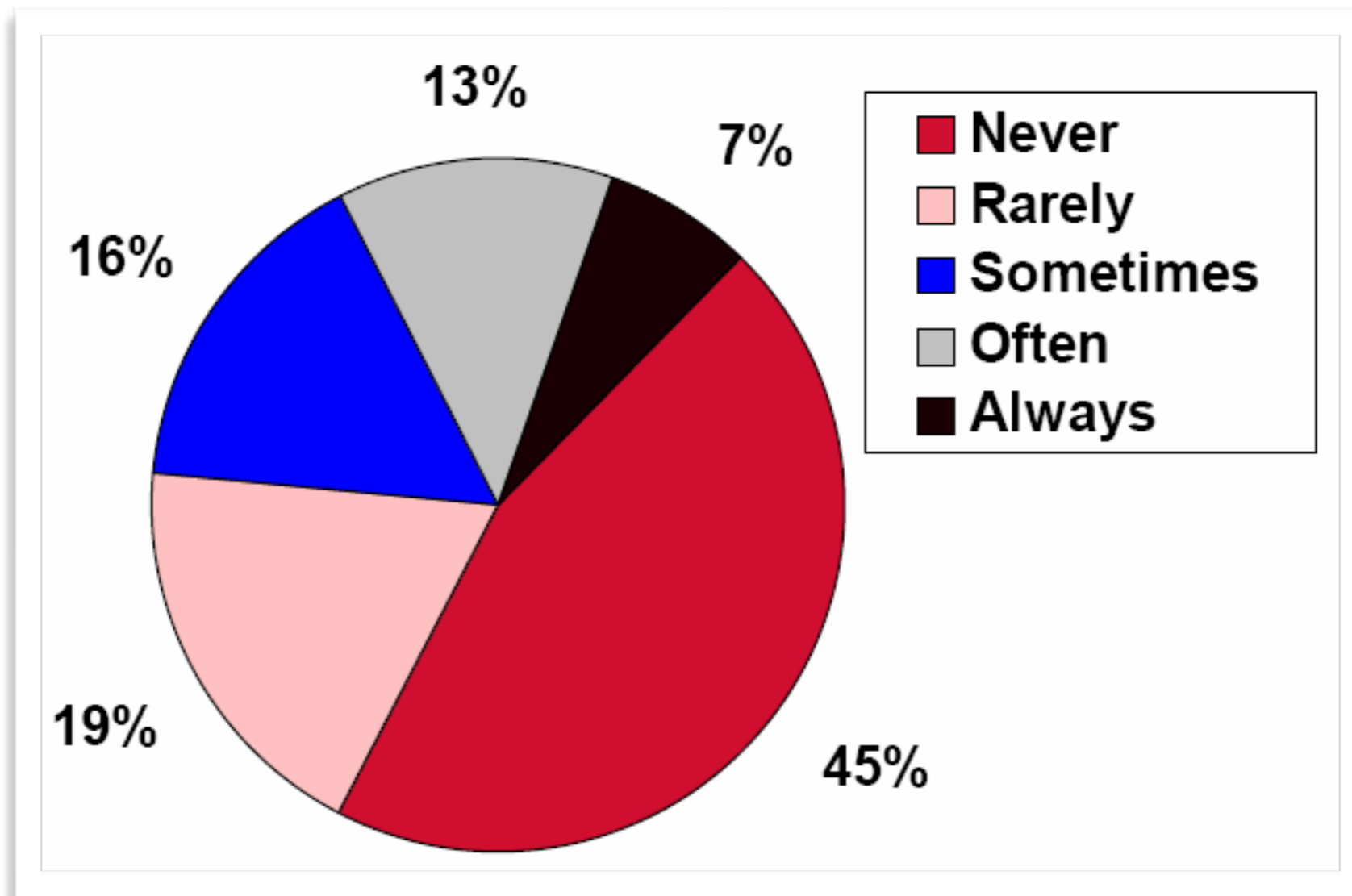


PROBLÉMY KLASICKÝCH METOD

- Nepochopení klíčovými potřebami zákazníka (core needs).
- Neschopnost práce s měnícími se požadavky.
- Nemožnost či obtížná integrace SW modulů.
- Obtížná a drahá údržba či rozšíření SW systémů.
- Pozdní odhalení chyb, problémů.
- Špatná kvalita, výkonnost SW produktů.
- Problémy s buildy a releases

PŘÍČINA PROBLÉMŮ

- Nevhodná, špatná specifikace požadavků.
- Nejasná a slabá komunikace (se zákazníkem, mezi členy týmu).
- Nevhodná či nedefinovaná architektura.
- Přílišná složitost systémů.
- Nekonzistence v požadavcích či v návrhu.
- Slabé a nedostatečné testování.
- Problém s riziky a neočekávanými událostmi.
- Nekontrolované a nestandardní změny.
- Nedostatečná automatizace.



www.dilbert.com scottadams@aol.com

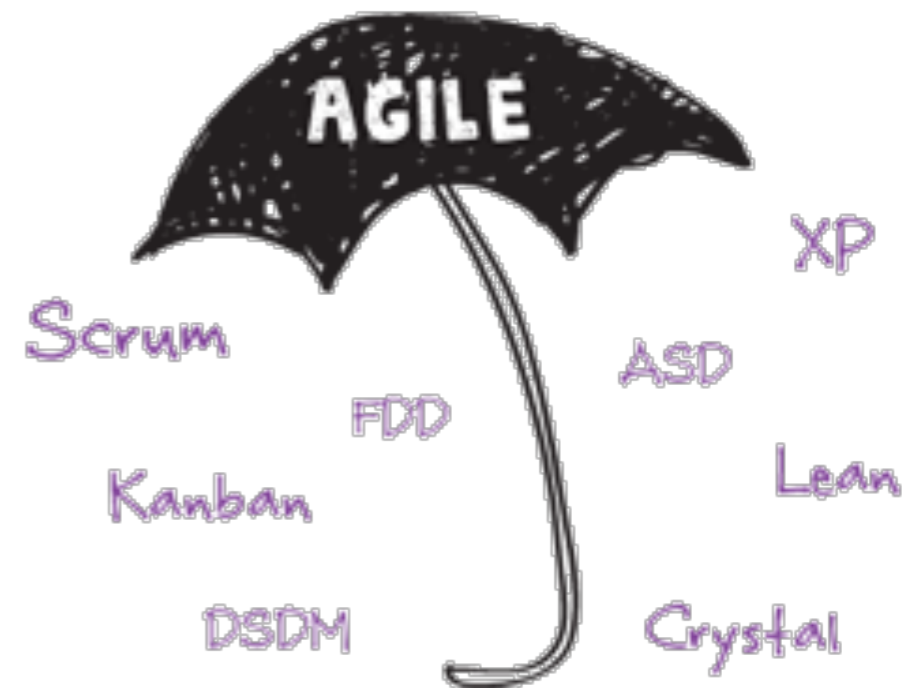
4/14/01 © 2001 United Feature Syndicate, Inc.

DŮSLEDKY

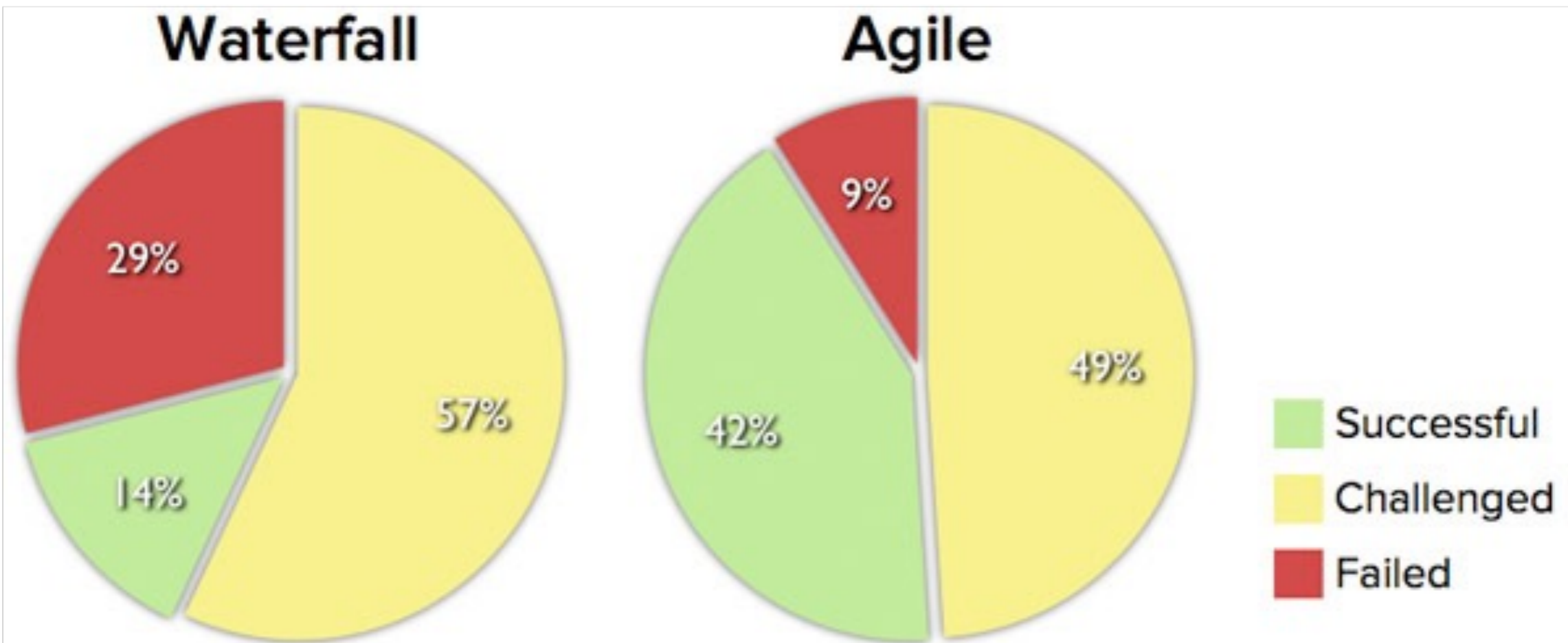
- 20% rysů softwarových aplikací je vždy nebo velmi často používáno.
- = klíčové potřeby zákazníka (core needs).
- 80% požadavků – problém v produktivitě týmů?
- 80% snahy analytiků, návrhářů, programátorů, testerů mohlo být soustředěno jinam (na větší kvalitu produktu, na jiné projekty) a také, že tuto práci zákazník „zbytečně“ platí!

AGILNÍ METODIKY

- Impulsem bylo Agile Manifesto (www.agilemanifesto.org)
- 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

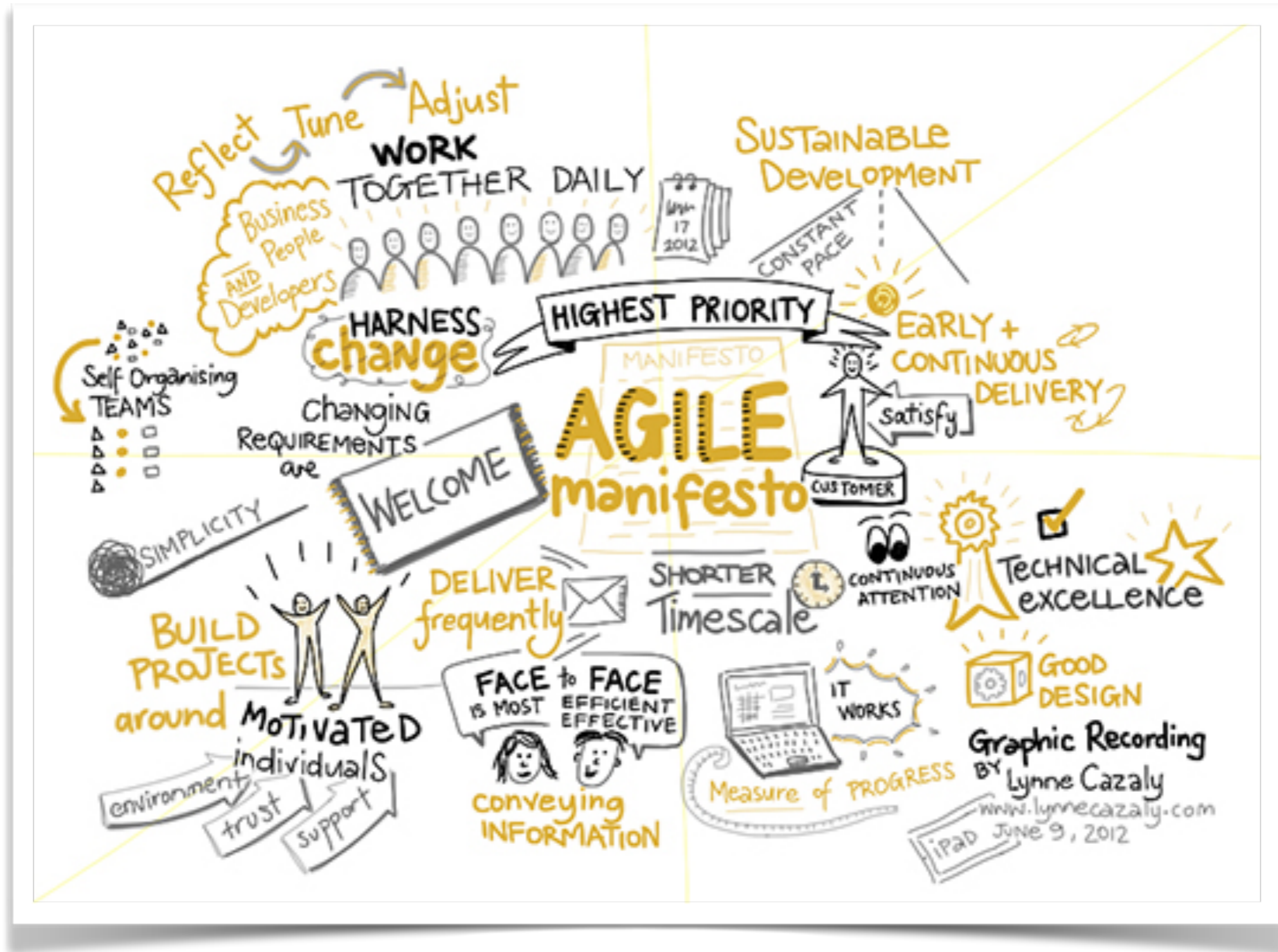


PROČ UVAŽOVAT O AGILNÍCH METODIKÁCH



Source: The CHAOS Manifesto, The Standish Group, 2012.

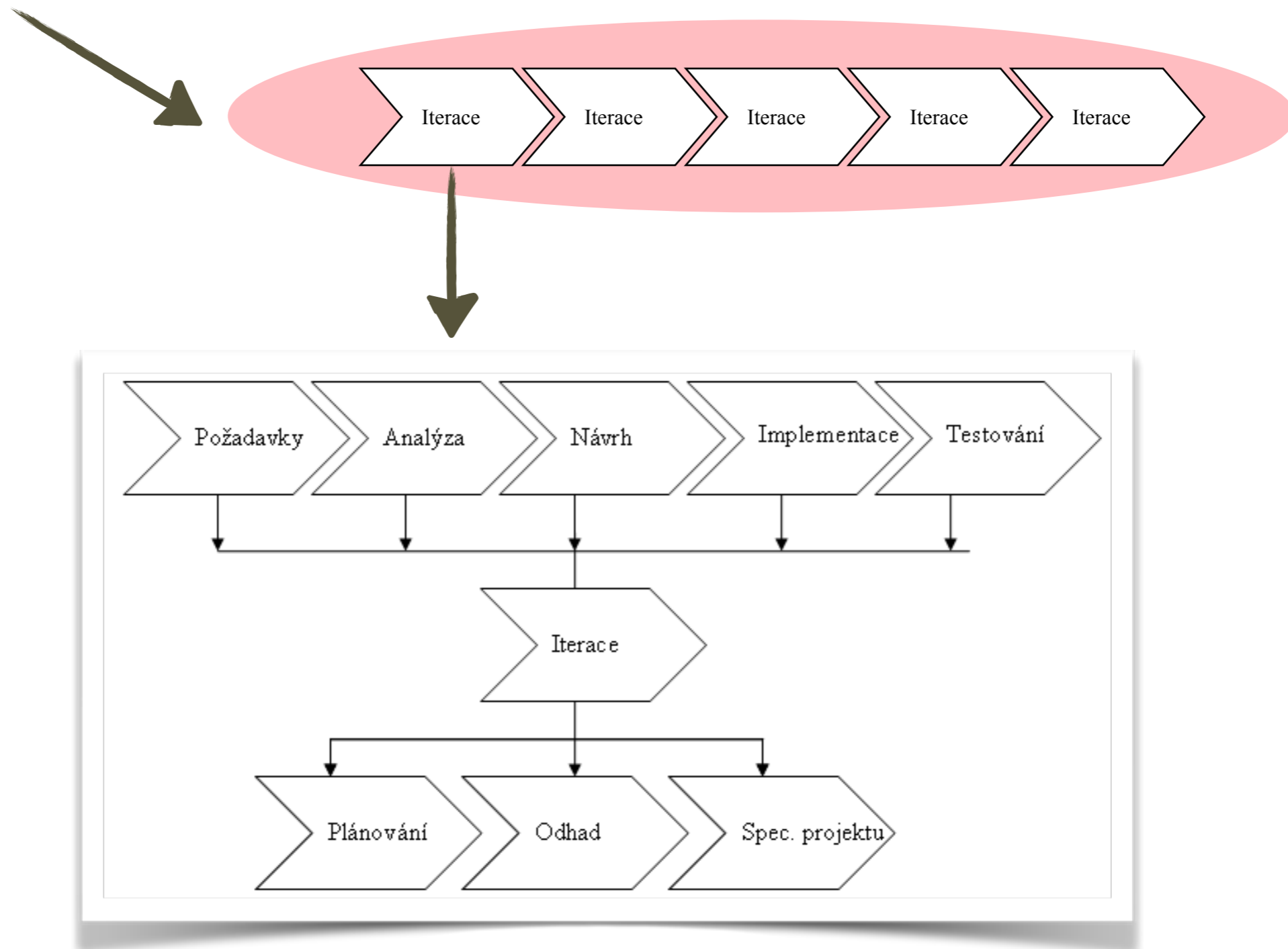
AGILE NENÍ PŘEDPIS, JE TO ZPŮSOB MYŠLENÍ



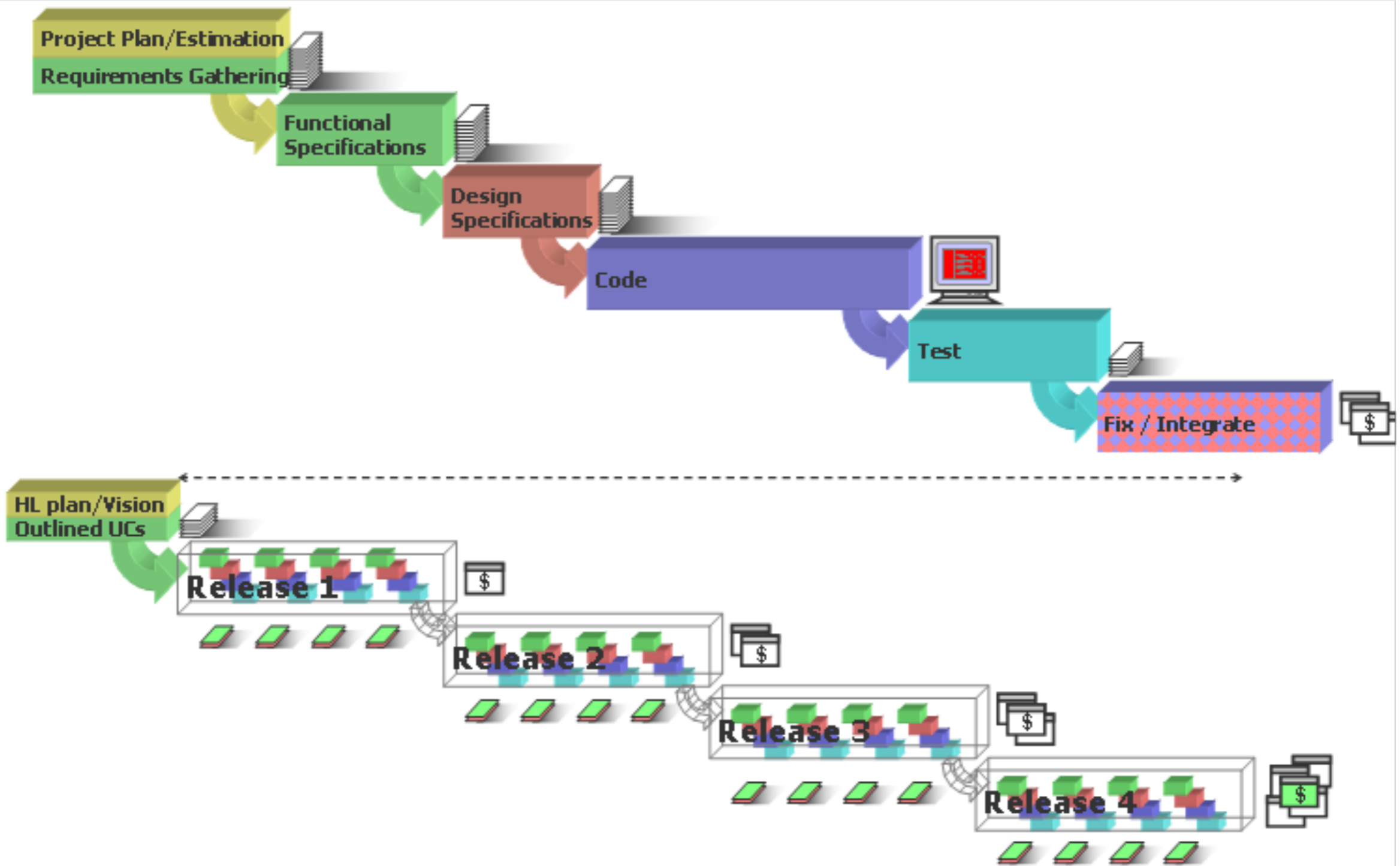
Vodopádové principy	Iterativní (agilní principy)
Zaměřen na procesy, předpokládá jejich opakovatelnost.	Zaměřen na lidi – motivace, komunikace prvořadá.
Pevné, podrobné plány definovány na úvod, kdy je spousta nejasností.	Pro celý projekt pouze road map. Podrobné plány jen pro iterace (kratší časové úseky, max. 2 měsíce).
Rizika jsou často překvapení, přináší problémy.	Řízen riziky – nejrizikovější věci řešíme nejdříve.
Integrace a testování až na konci.	Průběžná integrace a testování.
Změny nejsou vítány.	Počítá se změnami, přijímá je.
Často zaměřen na tvorbu dokumentů bez přidané hodnoty a jejich revize.	Zaměřen na fungující SW (hodnota pro zákazníka).
Buildy a testy až na konci, často přeskočeno nefunkční testování.	Automatizované buildy a testy.
Za kvalitu odpovědní pouze testeři, QA manažeři nebo často nikdo.	Všichni (celý tým) odpovědní za kvalitu produktu.

ITERATIVNÍ PŘÍSTUP

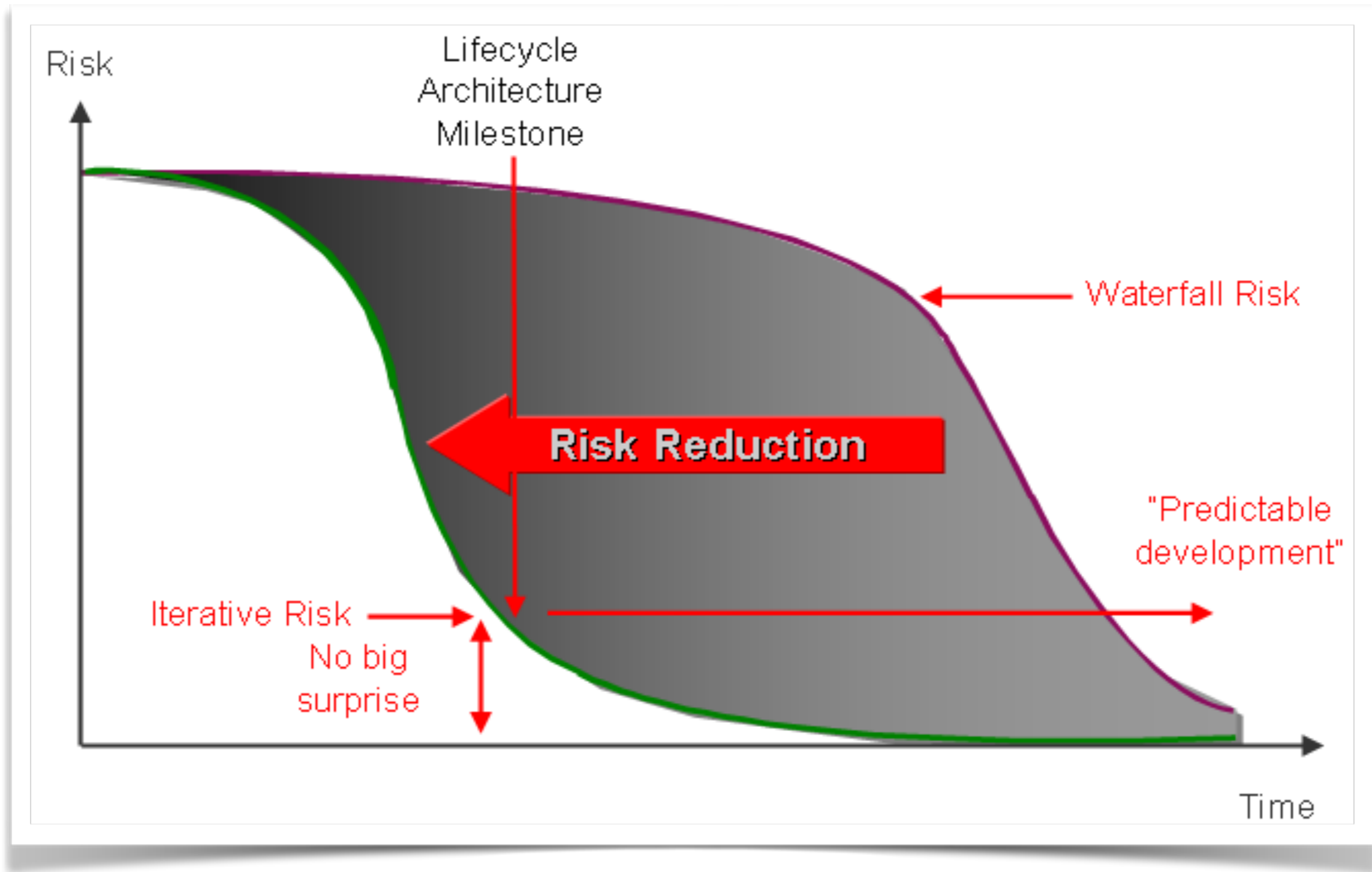
Projekt



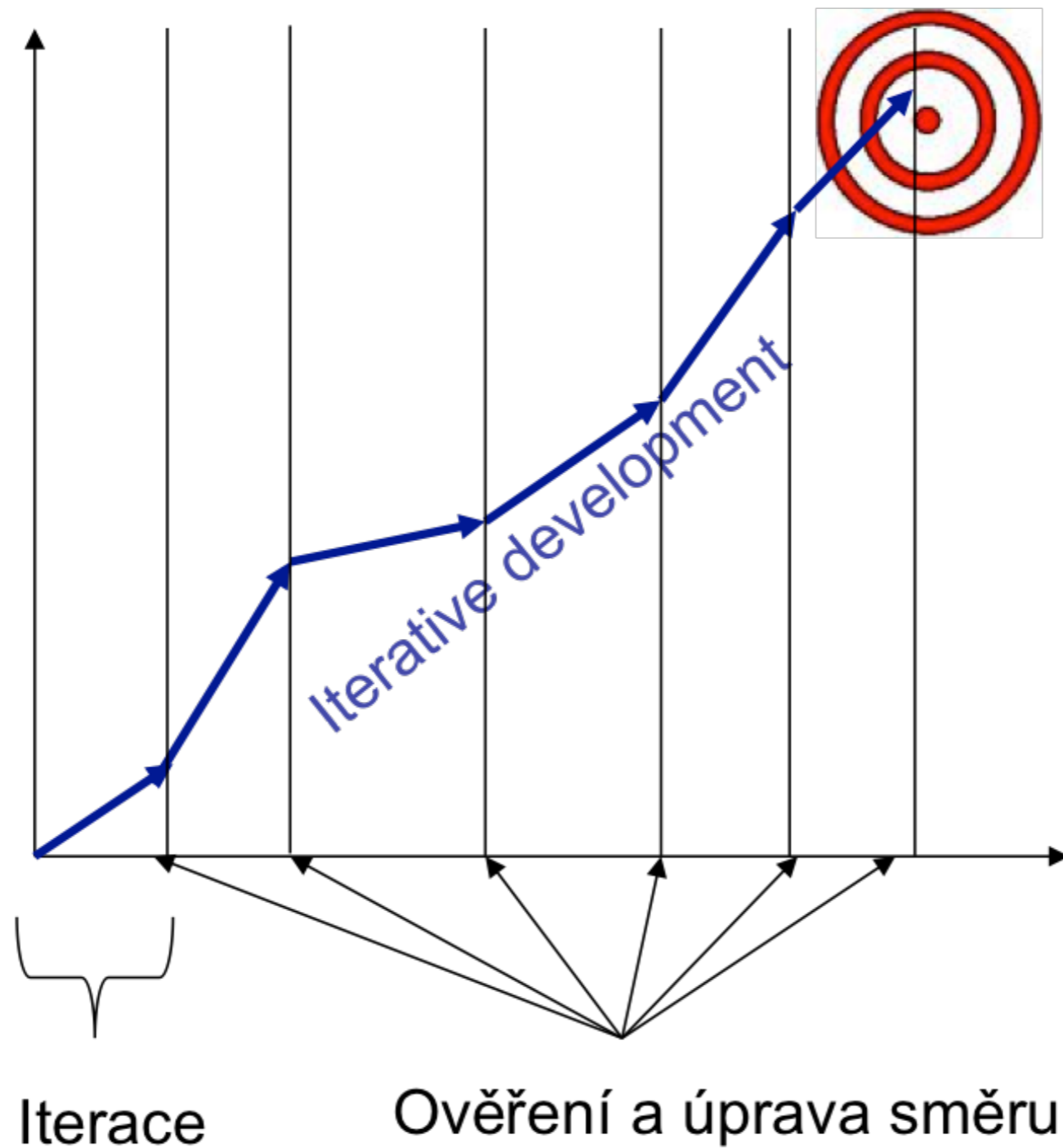
VODOPÁDOVÝ VS. ITERATIVNÍ PŘÍSTUP



ŘÍZENÍ RIZIK OBOU PŘÍSTUPŮ



OVĚŘENÍ SPRÁVNOSTI ZADÁNÍ



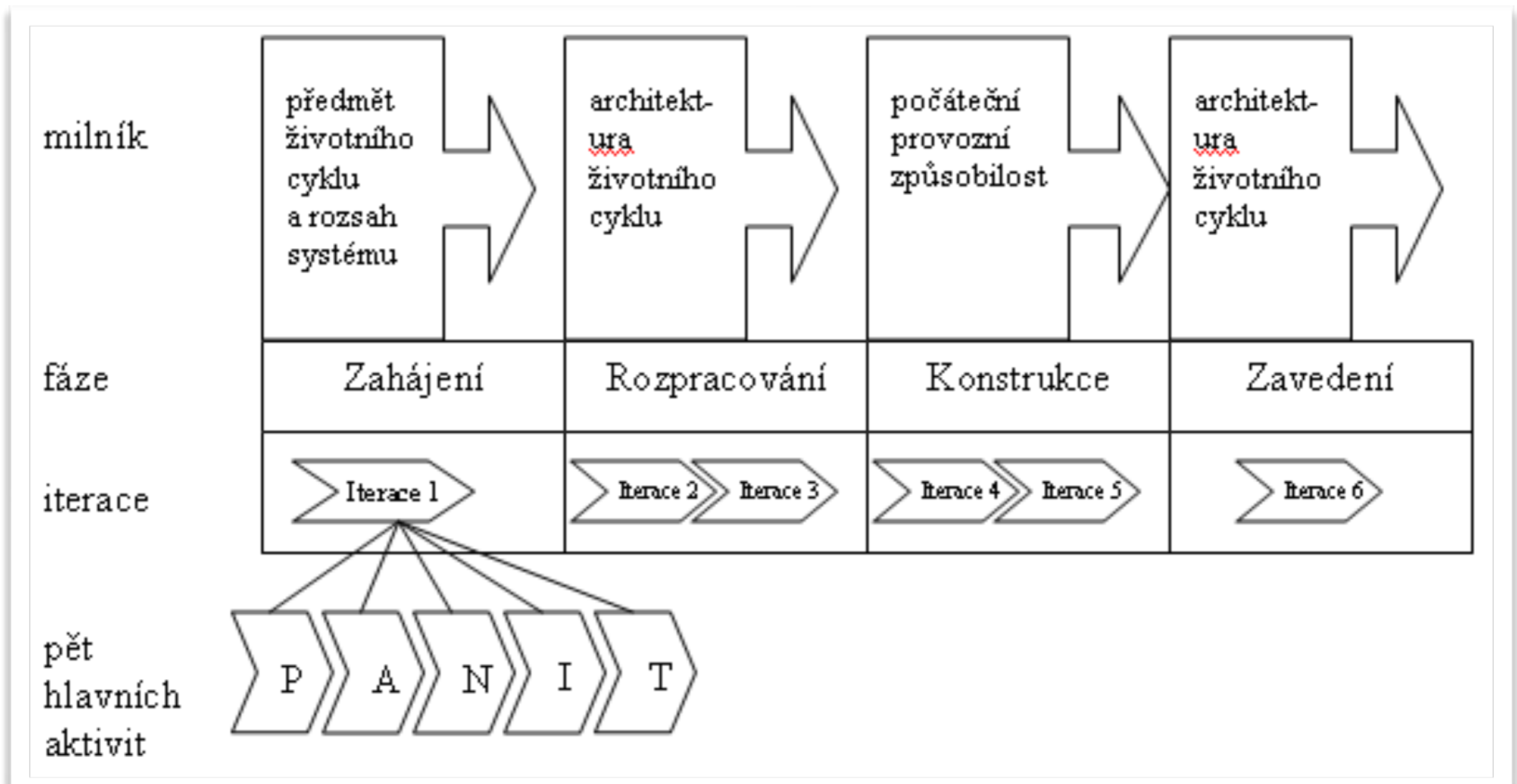
UNIFIED PROCESS A JEHO VARIANTY

- **UP** – Unified Process Software Development, iterativní přístup, UC driven, iterace řízeny riziky
- **RUP** – Rational Unified Process (IBM), rozšířené a propracované UP, web aplikace, příklady, šablony, guidelines, IBM nástroje, ...
- **Agilní přístupy** – XP, Scrum, Lean, ...

MILNÍKY, FÁZE, ITERACE, PŘÍRŮSTKY

- **Milník** (milestone) – konkrétní ověřovací krok např. ve formě checklistu, schválení cíle a rozpočtu, demonstrace produktu, architektury (spustitelná aplikace, ne dokumentace), akceptační testování, ...
- **Fáze** (phase) – vývoj projektu v čase, zaměřena na konkr. aspekt vývoje (inicializace projektu, odstranění rizik, doručení produktu zákazníkovi) končí definovaným milníkem, nejedná se o fázi vodopádu! Obsahuje několik iterací (ve kterých vykonáváme všechny disciplíny).
- **Iterace** (iteration) – mini projekt, provádíme všechny disciplíny, celý tým spolupracuje, výstupem je build (spustitelný kód).
- **Přírůstek** (release) – nově implementované funkčnosti aplikace (další scénáře) od posledního releasu, celý release funguje bez chyb jako výsledná aplikace.

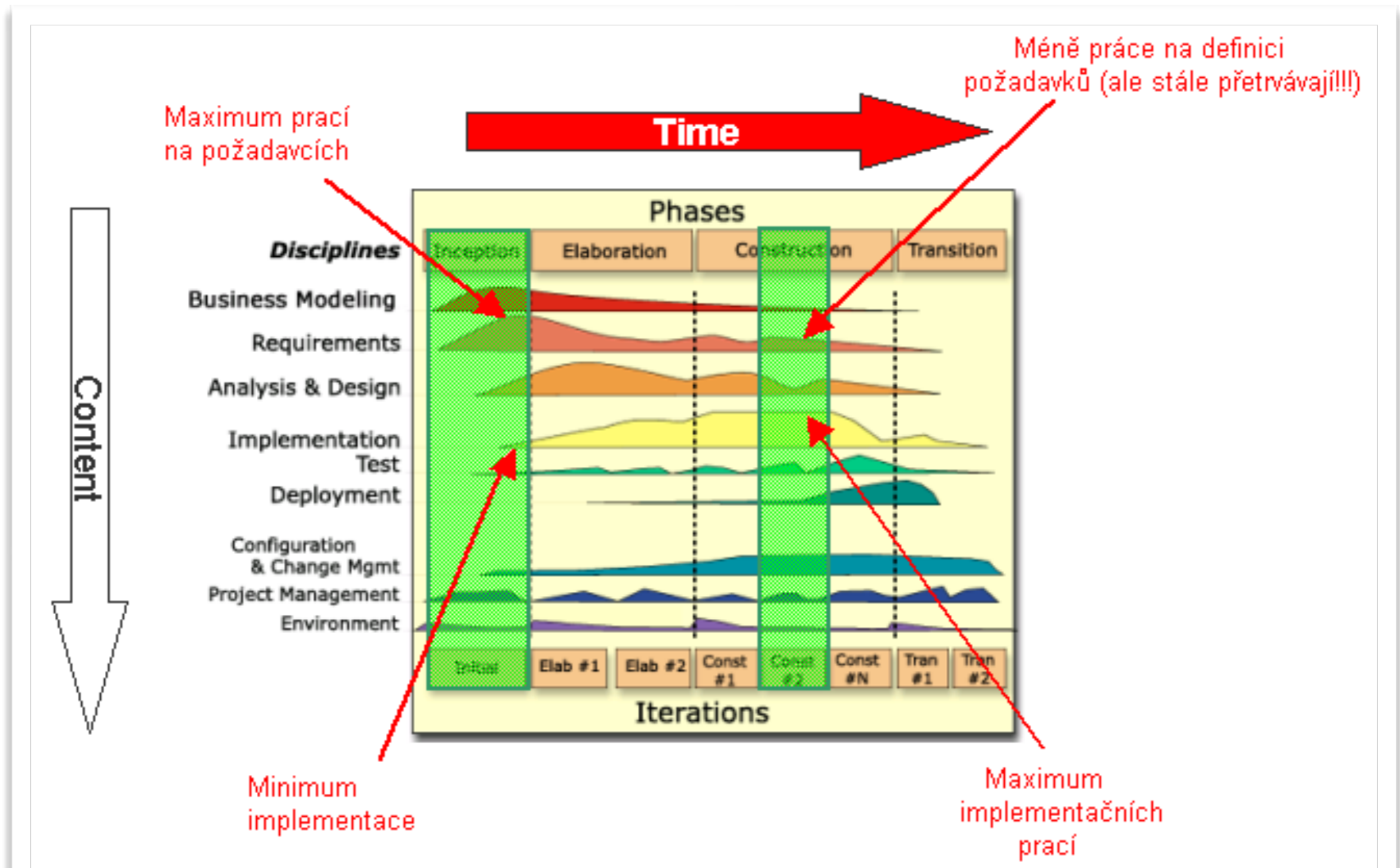
ŽIVOTNÍ CYKLUS UP

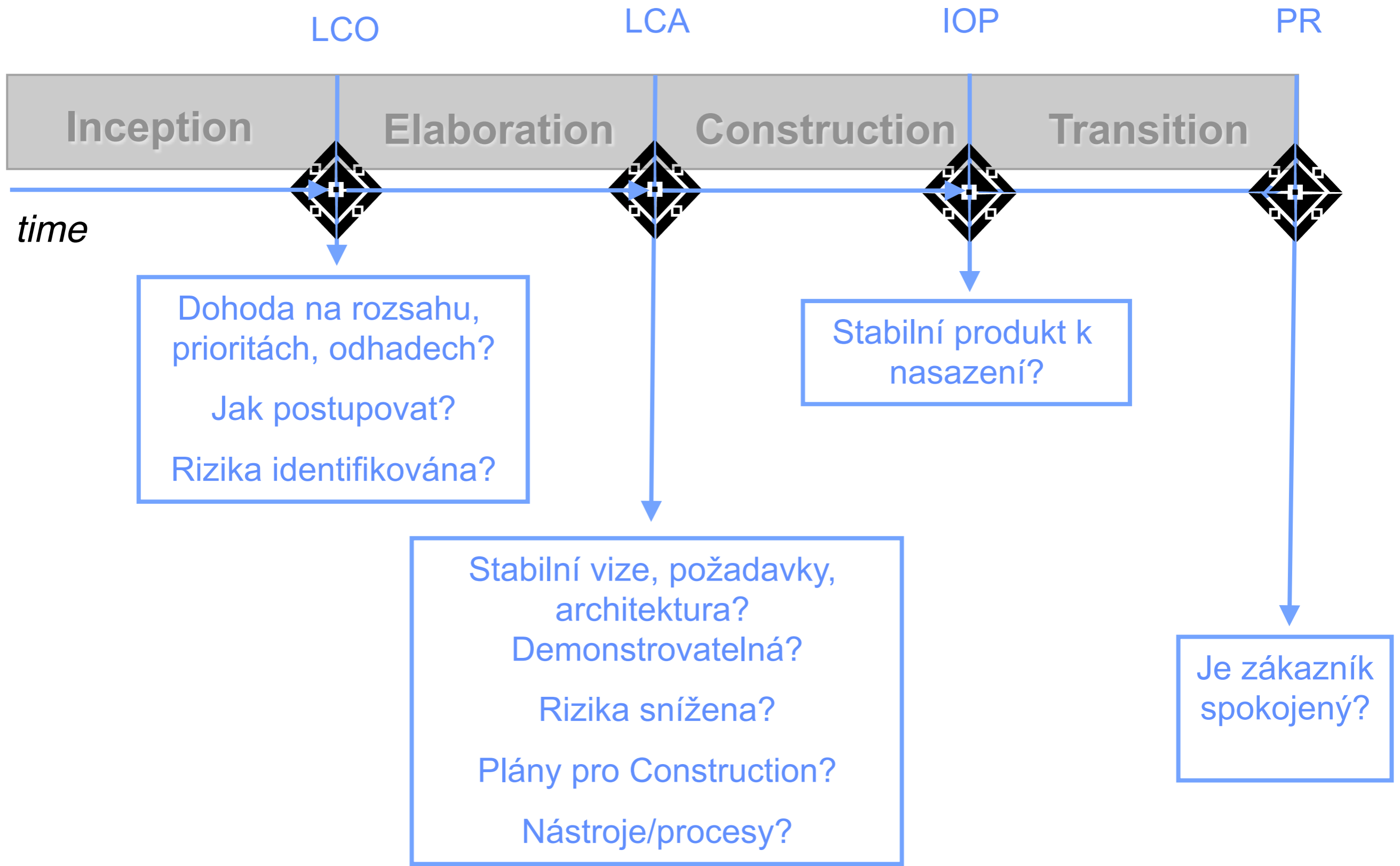


RUP A JEHO PRINCIPY

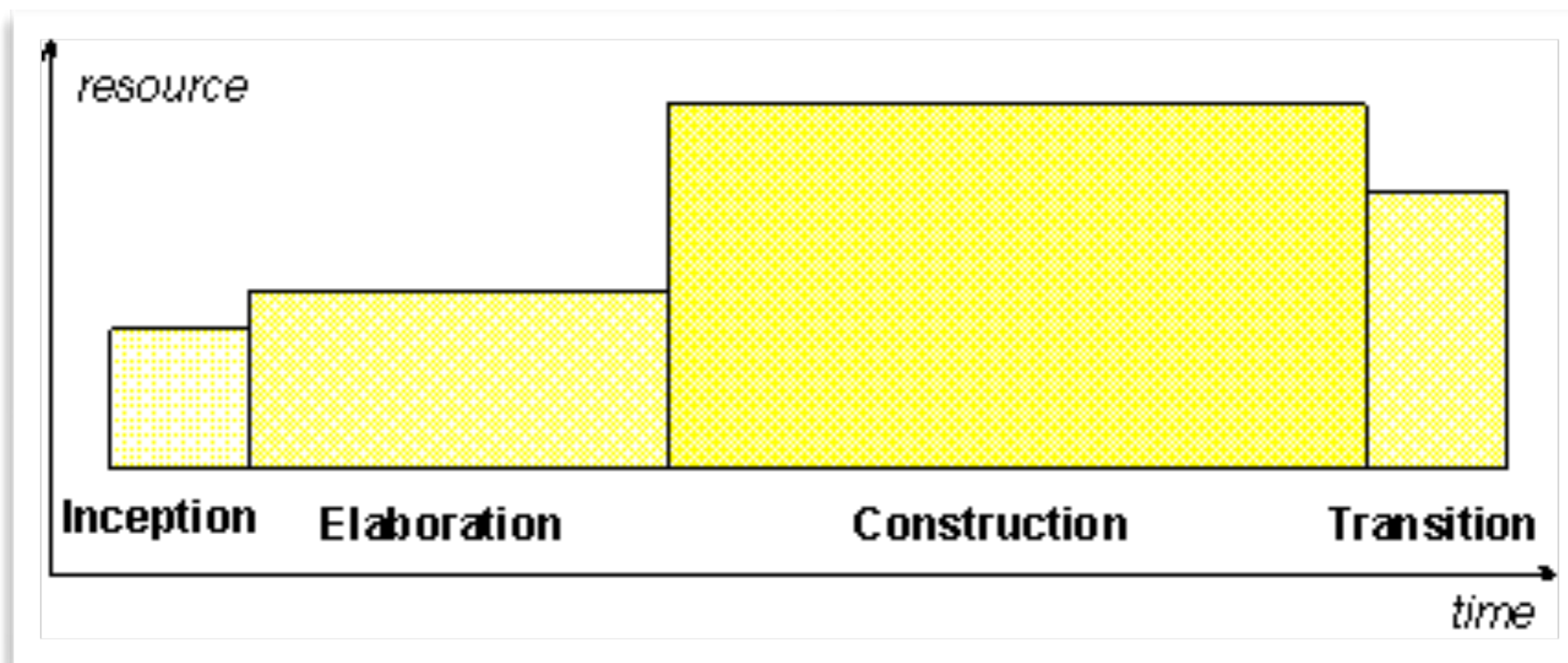
- Adapt the process
- Balance Competing Stakeholder priorities
- Collaborate across teams
- Demonstrate value iteratively
- Elevate the level of abstraction
- Focus on quality

DISCIPLÍNY, FÁZE





DÉLKA FÁZE, OBJEMY PRACÍ



Objem prací a časové trvání jednotlivých fází RUP

	Inception	Elaboration	Construction	Transition
Pracnost	5 %	20 %	65 %	10 %
Plán	10 %	30 %	50 %	10 %

FÁZE INCEPTION

- Porozumění tomu, co vytvořit – vytvoření vize, definice rozsahu systému, jeho hranic; definice toho, kdo chce vytvářený systém a co mu to přinese.
- Identifikace klíčových funkcionalit systému – identifikace nejkritičtějších Use Casů.
- Návrh alespoň jednoho možného řešení (architektury).
- Srozumění s náklady (ROI), plánem projektu, riziky.
- Definice/úprava procesu, výběr a nastavení nástrojů.

VIZE

Osobní časovač: vize

Problém: Gary není schopný sbírat konsistentní časové údaje od vývojářů reprezentující čas strávený na různých projektech. Není tedy možné monitorovat a porovnat postup oproti plánům, fakturovat řádné časy, platit externí spolupracovníky a samozřejmě také na základě těchto dat dělat věrné odhady dalších iterací

Souhrn vize: Osobní časovač (OČ) měří čas strávený na projektech, shromažďuje a ukládá tato data pro pozdější zobrazení (stylem Post-it poznámek), aby mohl Gary systematicky organizovat a hodnotit projekty, sledovat aktuální postup prací a ty porovnávat s plánovanými odhady pro jednotlivé projekty

Zainteresované strany

- jednotlivý vývojáři
- pracovníci administrativy
- projektový manažer

Use Case

- Změř čas aktivity
- Sesbírej týdenní data
- Sluč / konsoliduj data pro každý projekt
- Nastav nástroj a databázi pro projekt

MILNÍK LCO

- Shoda účastníků projektu (samozřejmě včetně zákazníka) na rozsahu projektu, počátečních nákladech a odhadu plánu, který bude dále upřesňován.
- Shoda na identifikaci správných požadavků a porozumění jim.
- Shoda na věrnosti odhadů nákladů a plánu, prioritách, rizicích a vývojovém procesu.
- Shoda na tom, že počáteční rizika byla identifikována a existují strategie na jejich snížení.

FÁZE ELABORATION

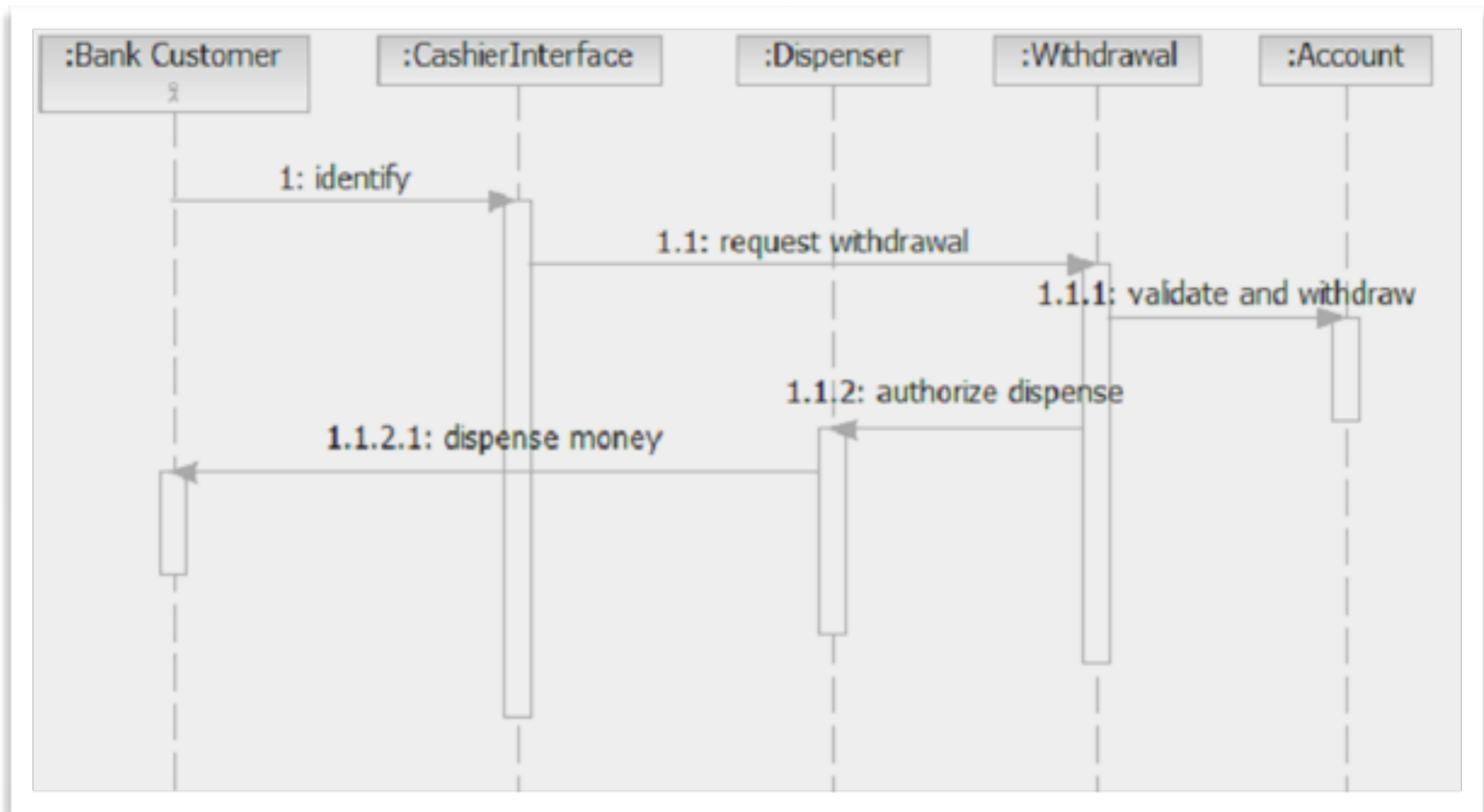
Cílem fáze hlavně snížení či odstranění rizik, 4 hlavní oblasti:

- Rizika spojená s požadavky na systém (Vyvíjíme správnou aplikaci?).
- Rizika architektonická (Poskytujeme správné řešení?).
- Rizika spojená s náklady a plány.
- Rizika procesní a spojená s prostředím a nástroji (Máme správný proces a nástroje?)

CÍLE ELABORATION

- Podrobnější pochopení požadavků (detailnější popis UC, další UML diagramy pro upřesnění)
- Návrh, implementace a ověření architektury (technologie, rozhraní, stavební bloky, prototyp, ...)
- Vytvoření přesnějšího plánu a odhad nákladů
- Implementace cca 20% scénářů

FORMÁLNÍ ROZPRACOVÁNÍ SCÉNÁŘŮ



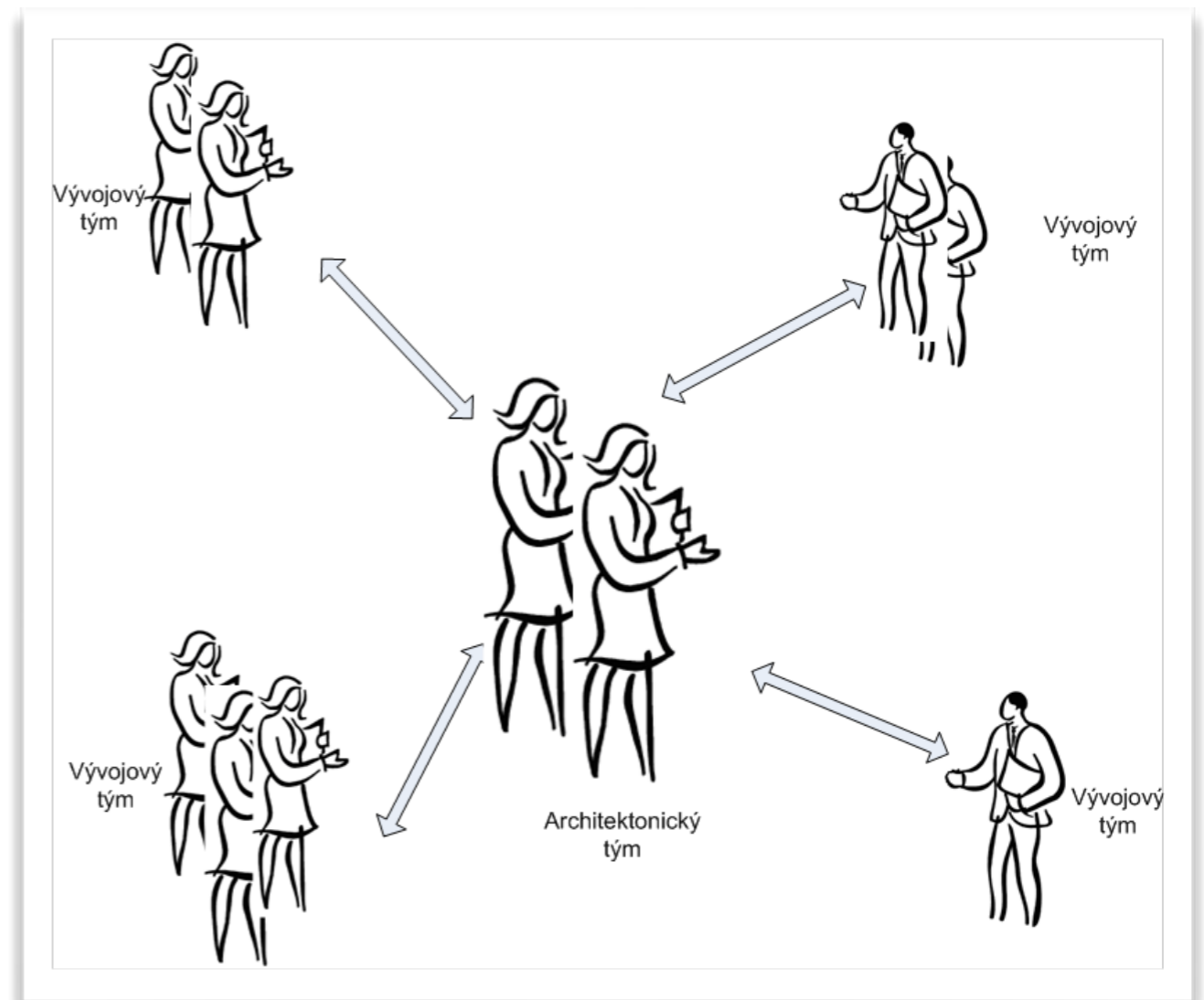
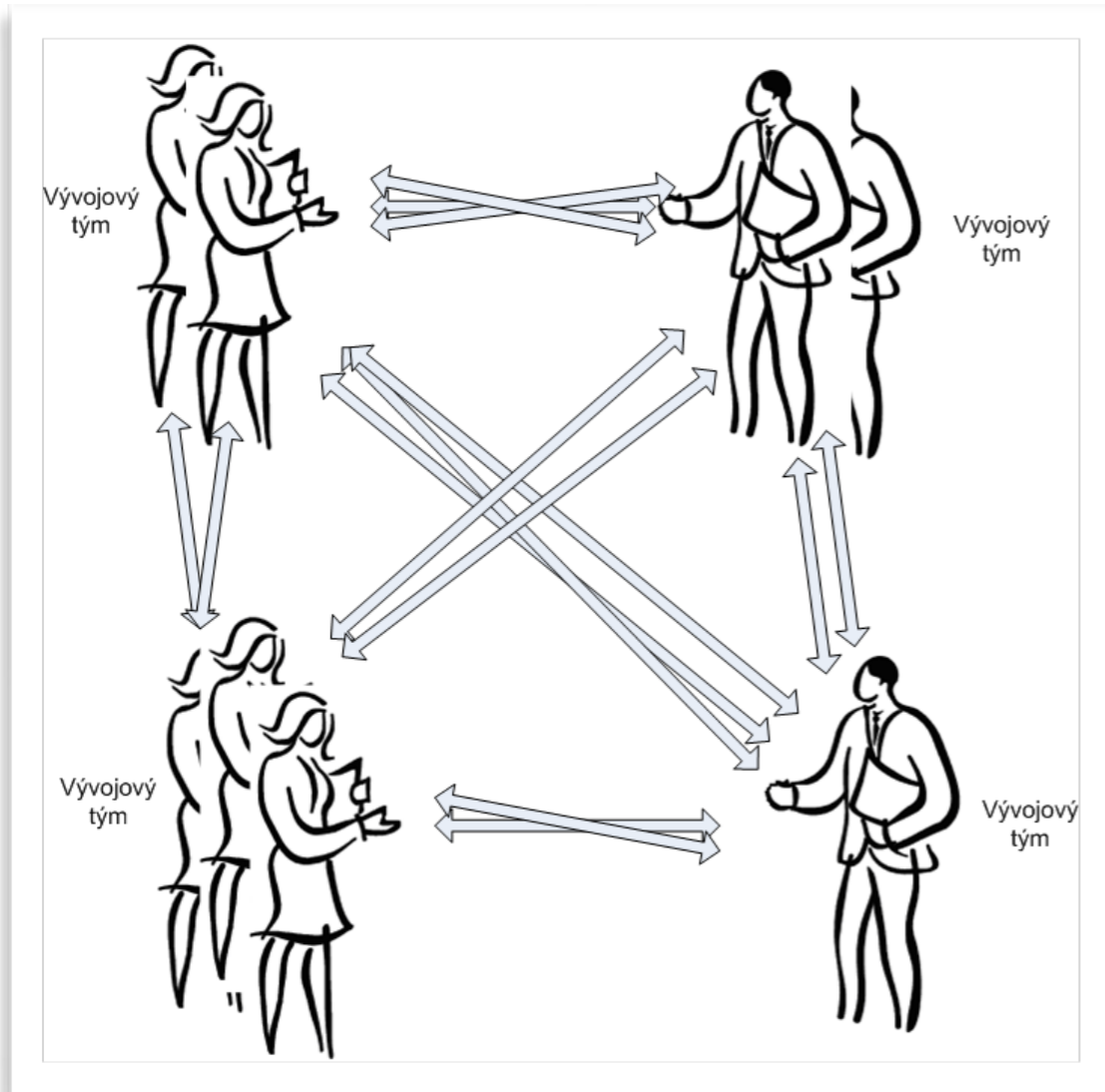
MILNÍK LCA

- Je vize produktu stabilní, jsou stabilní požadavky?
- Máme stabilní architekturu?
- Jsou klíčové postupy a přístupy, které budeme používat, otestovány a je dokázána jejich použitelnost?
- Ukázalo testování spustitelného prototypu, že jsou klíčová rizika identifikována a vyřešena?
- Máme definovány plány iterací pro následující Construction fázi v náležitých podrobnostech, abychom byli schopni podle nich postupovat?
- Jsou tyto plány podpořeny důvěryhodnými odhady?
- Naplněním plánu s použitím definované architektury dosáhneme cílů shrnutých ve vizi?
- Jsou aktuální náklady akceptovatelné vůči plánovaným?

CONSTRUCTION

- Minimalizace nákladů na vývoj.
- Dosažení určitého stupně paralelního vývoje (pro efektivnější využití zdrojů).
- Iterativní vývoj kompletního produktu, který bude připravený k doručení uživatelské komunitě.
- Iterativní vývoj celého produktu
- Beta release – první funkční verze aplikace.

ORGANIZACE KOLEM ARCHITEKTURY



SPRÁVA KONFIGURACÍ

- CM – configuration management
- Správa a řízení složitých softwarových projektů
- Vytvořen v Inception, doladěn v Elaboration
- Evidence a verzování komponent (SW, dokumentace, modely, ...)
- Sestavení buildů ze správných verzí
- Umožnění paralelní práce programátorů
 - Repository
 - Možnost připojení z více míst

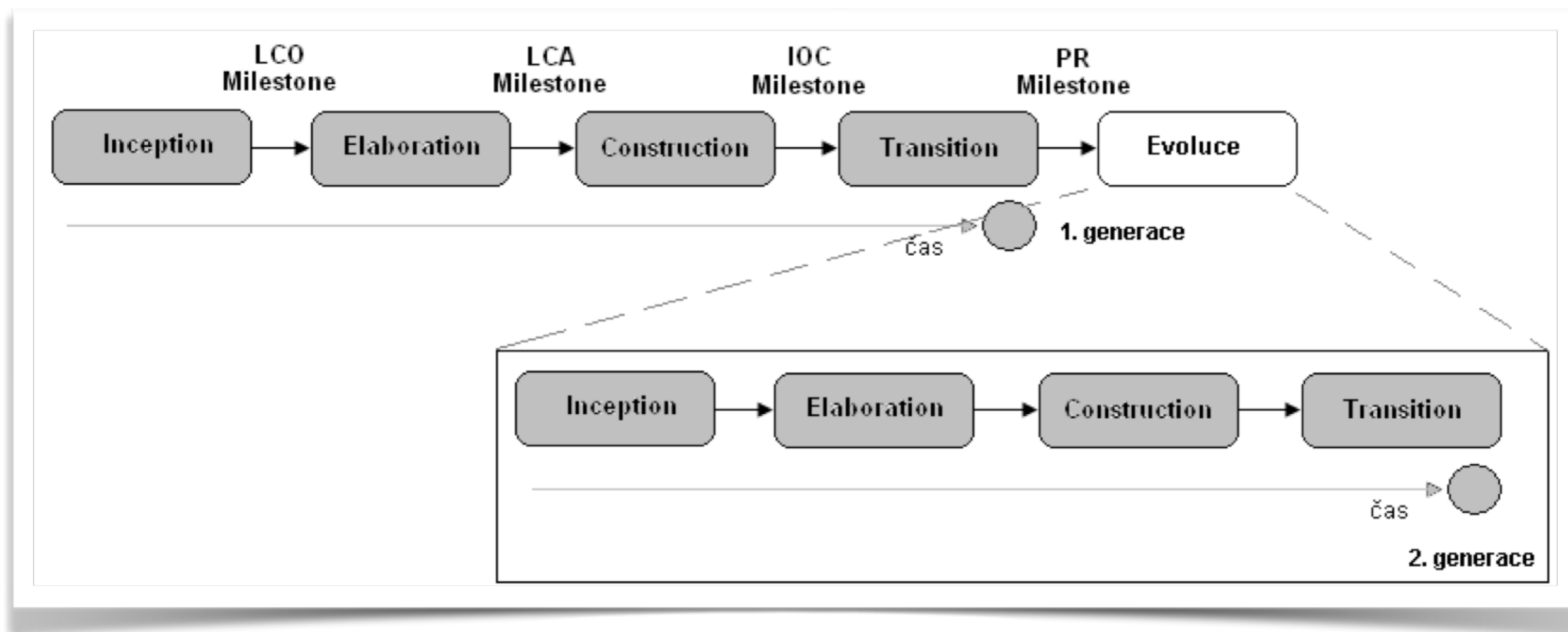
MILNÍK IOP

- Je produkt dostatečně stabilní a vyzrálý, aby mohl být rozeslán mezi komunitu uživatelů?
- Jsou aktuální výdaje na zdroje oproti plánovaným stále akceptovatelné?

TRANSITION

- Beta-testování – zjištění, zda jsme naplnili očekávání uživatelů.
- Školení uživatelů a správců aplikace.
- Příprava prostředí a dat.
- Příprava dalších aktivit zahrnujících marketing, distribuci, prodej – tvorba letáků s popisem produktu, white papers, technical papers, case study, demo nahrávek, zpráv pro tisk.
- Dosažení souhlasu uživatelů, že aplikace splňuje jejich představy zachycené v dokumentu Vize (Vision).
- Zlepšení průběhu budoucích projektů díky ponaučením z tohoto projektu tzv. lessons learnt.

VÝVOJOVÝ CYKLUS VERZÍ



MILNÍK PR

- Jsou uživatelé spokojeni?
- Jsou aktuální výdaje versus plánované akceptovatelné; pokud ne, jaké akce mohou být v příštích projektech provedeny, abychom tomuto problému předešli?