

Softwarový proces Iterativní vývoj software

KIV/ASWI 2007/2008



Vývoj software

... běžná aktivita v informační společnosti

- Na zakázku
 - » komerční zákazník
 - » státní sféra
- Interní projekt
- Krabicový software
- „Pro radost“

- Closed vs Open Source

Cíle zákazníka





Cíle dodavatele software

- Vytvořit aplikaci co možná
 - nejefektivněji (zdroje)
 - nejrychleji
- Minimalizovat přepracování
 - zadání, re-use
- Snížit rizika
 - plynoucí z neznámého: funkčnost, technologie



Cíle studenta

■ Comments?

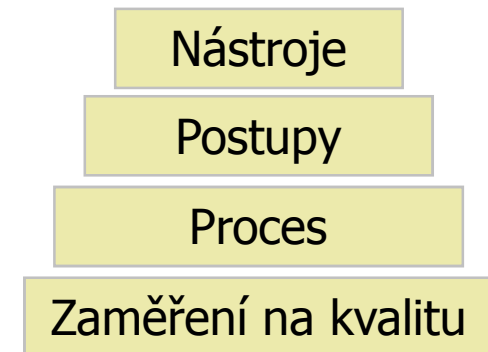
Softwarový proces

- **Proces:** *systematická série akcí vedoucí k určitému výsledku*

[Random House Unabridged Dictionary, 2006]

- **Softwarový**

- výsledek = kvalitní software
- členění: fáze, aktivity; produkty
- meta-proces, ŽC
 - » varianty uspořádání aktivit, produktů





Typické aktivity sw procesu

- **Technické**

- Komunikace
- Plánování
- Modelování
- Konstrukce
- Nasazení

- **Podpůrné**

- Řízení
- Kontrola kvality
- Správa konfigurace
- Dokumentace



Role lidí v procesu

■ Technické

- analytik (konzultant)
- architekt, návrhář
- vývojář
- „buildovač“ a správce konfigurace
- tester
- databázista
- poradce, kouč

■ Manažerské

- team leader
- technický vedoucí projektu
- šéf vývojářů
- šéf projektů
- CEO (příp. CIO)

■ Podpůrné

- lektor
- uživ. podpora
- dokumentace



Artefakty a jejich role

- **Technické**
 - » specifikace
 - » dokumentace
 - » kód, data
 - » testy
 - » modely
- **Komunikační**
 - » specifikace
 - » plán
- **Obchodní**
 - » plán
 - » rozpočet
 - » produkt
- **Účel**
 - Popis - dokumentace
 - Kontrakt
- **Vlastnictví**
- **Výsledek/vstup aktivity**
 - » podpora – CASE



Postup řešení „dle učebnice“

Návod

- Preskriptivní modely procesu / ŽC
 - » přehlednost a kontrolovatelnost
 - » vodopád, V-model
- Dodávka celého systému najednou
 - » začátek: kompletní specifikace požadavků

Realita života

- Změna je součástí podnikání
 - » zákazník neví co chce
 - » dodavatel neví jak na to
- Napoprvé se to nepovede
 - » velký třesk
 - » všechna (špatná) překvapení na konci

*Fast.
Cheap.
Good.
Choose any two.*



Varianty procesu

- Společná snaha = snížení rizika chaotického postupu
- Řízené plánem
 - » typicky sekvenční – vodopád, V-model
- Řízené riziky
 - » průzkumník/prototypování, spirála
- Řízené změnou
 - » iterativní, agilní



Sekvenční postup

- Hlavní technické aktivity lineárně po sobě
 - vztažené na celý produkt → „velký třesk“
 - naplánované pro celý projekt
 - oddělené meziprodukty
- Vodopádový model (v běžném podání)

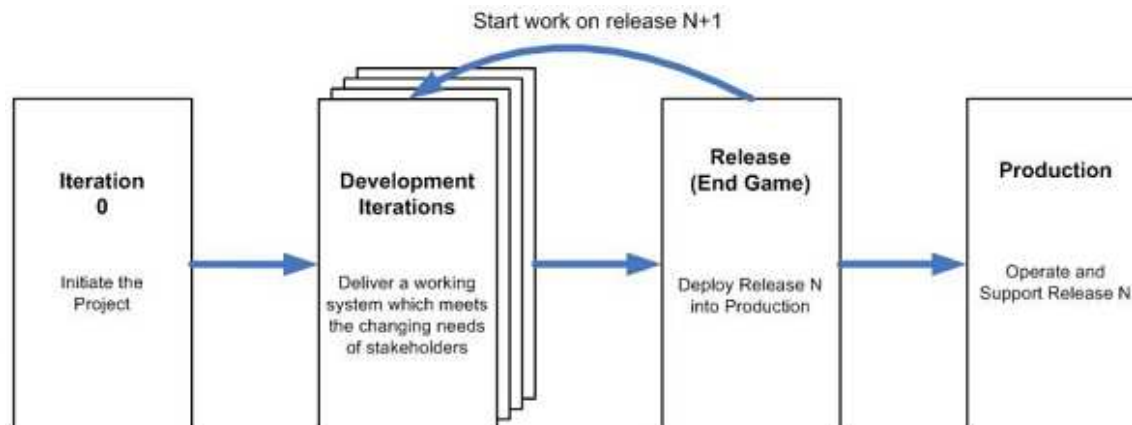


Cyklický postup

- *„Když sekvenční postup funguje pro malé projekty s malou mírou neznáma, proč nerozbit velký projekt do řady malých?“*
– P.Kruchten
- **Opakování technických aktivit**
 - » obsah podle sekvenční fáze, znalosti detailů
- **Produkt postupně „roste“**
 - » znalost, funkcionalita, kvalita, ...

Alternativy dodávek funkčnosti

- Velký třesk
 - malé projekty, jasné požadavky
 - Přírůstkově
 - » určení přírůstků -> plán -> postupné dodávky
 - zpětná vazba, ale úpravy projektu obtížné
- + iterativně ⇒ určování a plán průběžné, nutná disciplína





Životní cyklus, metodika

- **ŽC** = proces od zahájení vývoje až po vyřazení z provozu
- **Metodika** = definovaný proces pro konkrétní účel, tj. fáze, aktivity, role, artefakty, milníky atd. jsou dobře popsány
 - » Booch method
 - » SSADM, Rational Unified Process, SCRUM
- UML není metodika!



Jakou zvolit metodiku?



Software: Mýty vs Realita

- *Software není automobil*
 - *Změna je život*
- *Dinosauri vyhynuli, myši nikoli*
- *Sjíždět vodopád je nebezpečné*



Pár čísel

Standish Group
„Chaos Report“,
1995

FAILURE RECORD

In the United States, we spend more than \$250 billion each year on IT application development of approximately 175,000 projects. The average cost of a development project for a large company is \$2 322 000; for a medium company, it is \$1 331 000; and for a small company, it is manufacturing, retail, wholesale, health care, insurance, services, and local, state, and federal organizations. The total sample size was 365 respondents and represented 8,380 applications. In addition, The Standish Group conducted four focus groups and numerous personal interviews to provide qualitative context for the survey results.

For purposes of the study, projects were classified into three resolution types: -

- Resolution Type 1, or project success: The project is completed on-time and on-budget, with all features and functions as initially specified.
- Resolution Type 2, or project challenged: The project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified.
- Resolution Type 3, or project impaired: The project is cancelled at some point during the development cycle.

Overall, the success rate was only 16.2%, while challenged projects accounted for 52.7%, and impaired (cancelled) for 31.1%.



Realita stavu SWI





Kolik je \$17 mld?

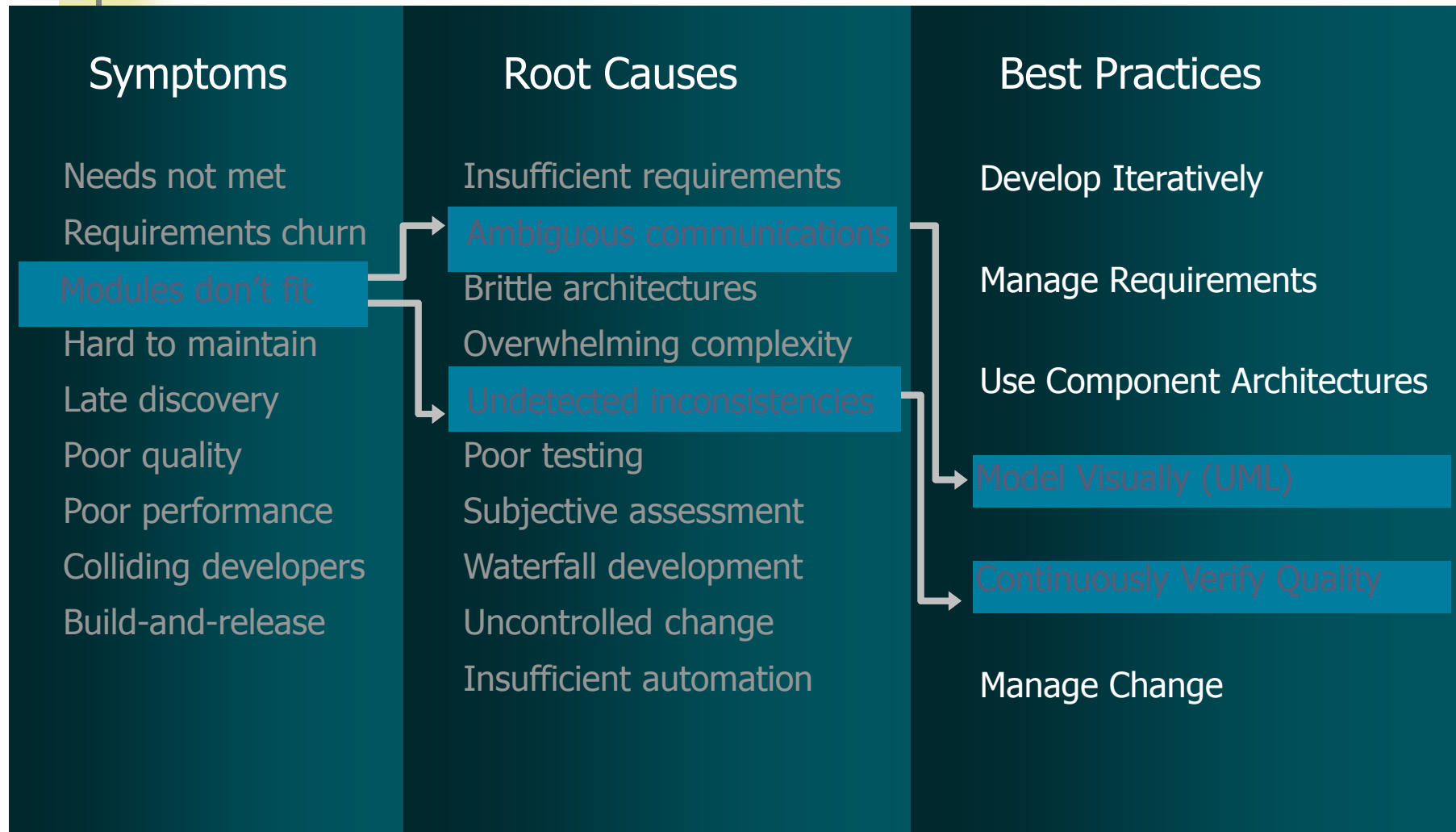
- tucet komerčních letů na Měsíc [google „project apollo cost“]
- 3x cena majority v ČTc
- výše dotace EU do zemědělství na jeden rok [google „14 miliard EUR“]
- 3/4 nákladů na přechod ČR od centrálně plánované ekonomiky na ekonomiku tržní [MFČR]
- cca 1/2 celkové ceny lunárního programu Apollo [google „project apollo cost“]



Discussion: Symptoms and Root Causes

- What are some symptoms of software development problems?
 - To what root causes can these symptoms be traced?
-

Trace Symptoms to Root Causes





Mýty softwarových projektů

- Zákazník ví, co chce
 - pevné vlastnosti produktu
 - předem známý cílový stav
- Dodavatel ví, jak na to
 - predikovatelný postup, náklady, kvalita
 - lineární škálování složitosti projektu
- Tyto předpoklady – platné pro sériovou výrobu – používá příliš mnoho softwarových procesů (a manažerů a klientů)
 - založené na zjednodušeném, a idealizovaném, vodopádovém modelu



Tvorba SW není sériová výroba

Software není automobil

- Sériová výroba
 - CDčka, koloběžky, pračky, mobily, auta, paneláky
 - pevné a předem známé specifikace, známý cíl
 - známý výrobní postup, přesné odhady na začátku
 - malá míra variability a nutnost reakce na změny
 - problémem je logistika a ekonomie výroby kopií
- Tvorba software nemá (ve většině případů) charakter předvídatelného projektu a/nebo sériové výroby. Naopak: jde o vývoj nového (typu) produktu.
 - studie vozu, ekologický dům, raketoplán
 - produkt a projekt jedinečný, bez vzoru a modelu



Změny jsou pravidlem, ne výjimkou

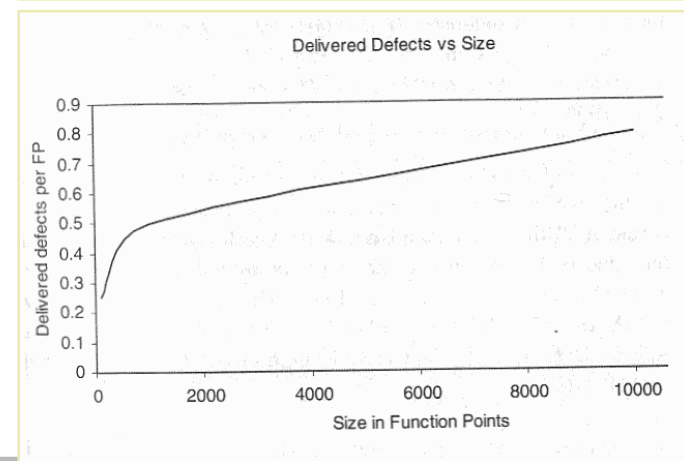
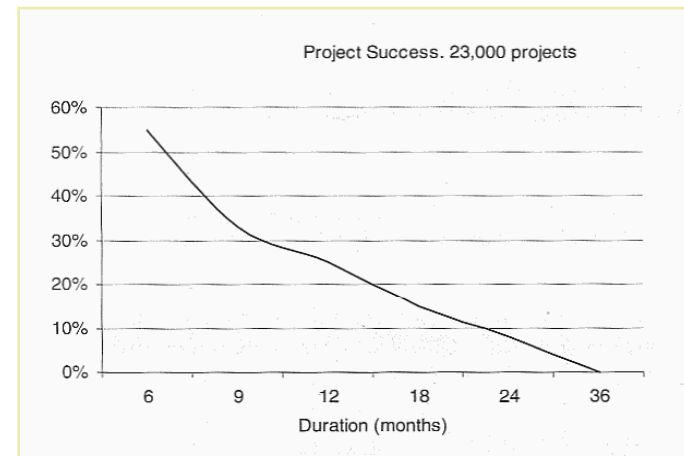
Změna je život

- Změny požadavků: spekulativní funkce/vlastnosti, fenomén IKIWISI, nedostatečná komunikace, ...
 - » „Zákazník neví, co chce, neumí to říct, ale chce to“
 - » typicky se změny zhruba 25% specifikovaných požadavků [Boehm88]
- Změny prostředí: legislativní rámec, akvizice firmy, upgrade systémů zákazníka, nové technologie, ...
- Změny postupu: fluktuace v týmu, chybná architektonická rozhodnutí, změny nástrojů, ...

Velikost pracuje proti nám

Dinosaurři vyhynuli, myši nikoli

- Úspěšnost
 - velké plány, velká zklamání
 - přes 1/2 velkých zrušeno
 - potvrzováno teorií systémů
- Produktivita
 - nepřímá úměra k velikosti produktu
 - » větší pro malé přírůstky, týmy
- Četnost změn
 - 10% malé projekty (100 FP)
 - 35% velké (10000 FP)
- Kvalita
 - nepřímá úměra k velikosti produktu
 - prototyp se 40% → 20% funkčnosti
 - ⇒ pokles chybovosti o 10/měs/MLOC



Zjednodušené modely nefungují

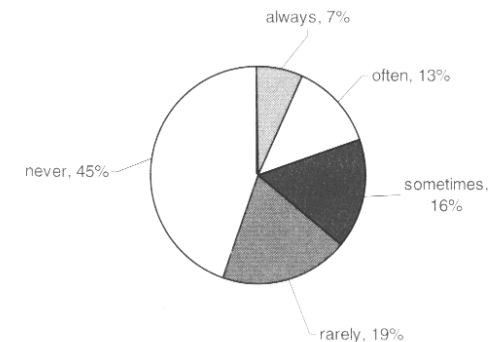
Sjíždět vodopád je nebezpečné


- „Pro každý složitý problém existuje řešení, které je jednoduché, elegantní, a špatné“ [H.Mencken] – například vodopádový model v „klasickém“ vydání ...
 - 4 z 5 faktorů neúspěchu projektů jsou spojeny s VM [Jones95]
 - použití VM nejvíce přispívá ke krachu projektů v 80% [Thomas01]
 - expertní doporučení je vyhnout se VM [Brooks87]

■ Například

- studie DoD 1995: ze systémů za celkem \$37 mld, vyvíjených podle DOD-STD-2167, jich 46% nebylo nikdy nepoužito
- US ATC projekt 1983-1994: vodopád, velký třesk, \$2.6 mld, zrušeno
- Johnson02: využití předem specifikovaných požadavků

Kolik je \$17 mld?





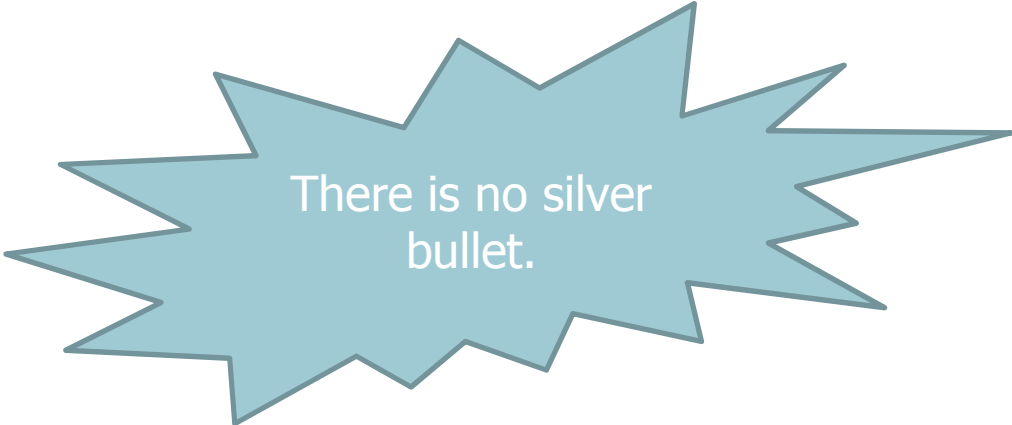
„Jestliže se čtení webu podobá prohlížení billboardů, pak navrhujte web tak, jako byste navrhovali billboard.“

Steve Krug



Čili: Jakou zvolit metodiku?

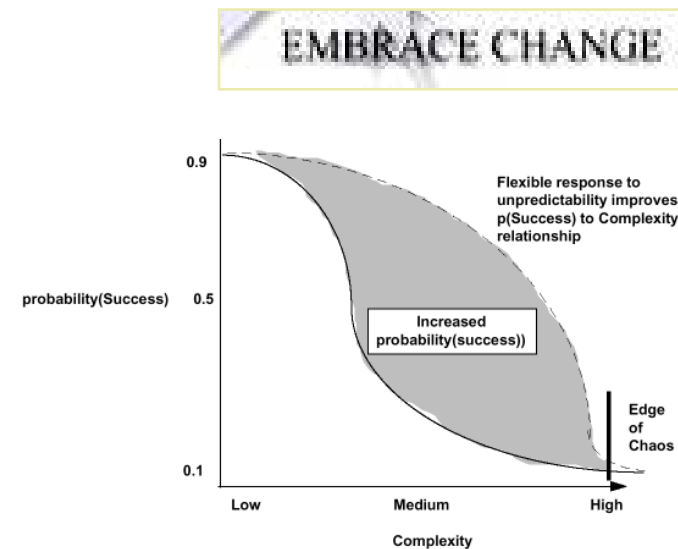
- Ad-hoc (neřízený proces)
- Sekvenční
- Iterativní
- Agilní



There is no silver
bullet.

Řešení

- Přivítat změnu
 - » opustit to, co nefunguje – přístup vodopádu
- Iterativní a evoluční vývoj
- Adaptivní plánování
- Agilní přístup





Iterativní vývoj software

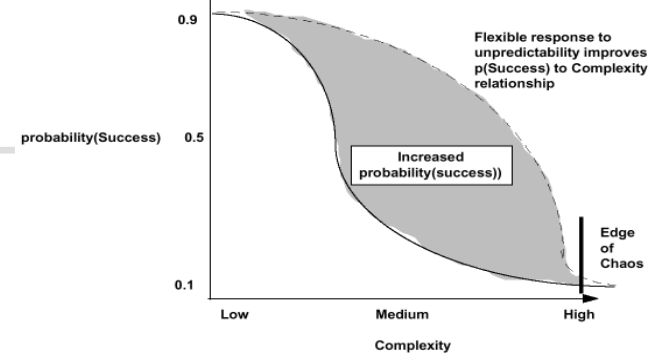
Q: Jaké můžeme v nejbližší době čekat nové, vzrušující a slibné myšlenky nebo techniky v oblasti software?

A: Myslím, že [nejslibnější myšlenky] jsou už léta známy, jen nejsou správně používány.

– David Parnas

Přehled

EMBRACE CHANGE

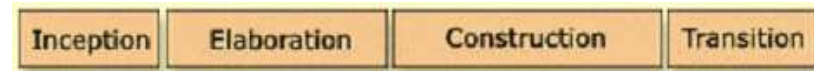


- Iterativní vývoj
 - včasná reakce na problémy při vývoji
- Evoluční (adaptivní) dodávky a plánování
 - podchycení změn požadavků
- Empirický proces
 - adaptace na změny týmu a postupu

Iterativní vývoj

- Rámcový plán životního cyklu

- milníky – např. RUP



- Řetězec vývojových iterací

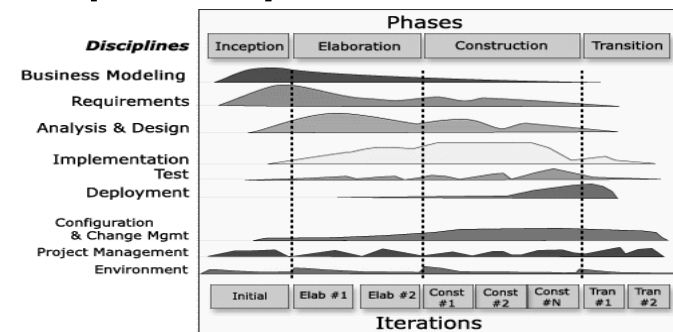
- miniaturní úplný projekt – cca vodopádový model

- cíl: iterační release (interní)

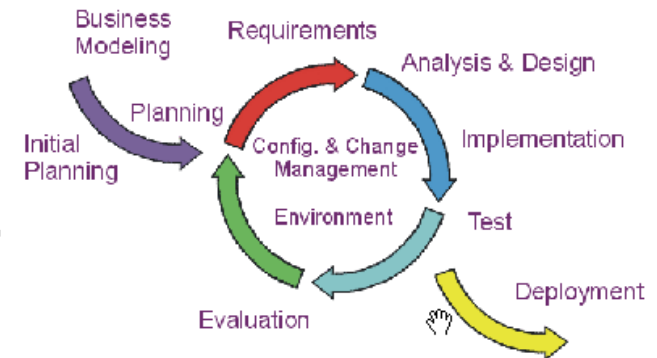
- produkt funkčně neúplný
- ale otestovaný a funkční

- vede na přírůstkový vývoj

- nevyklučuje kompletní počáteční specifikaci požadavků



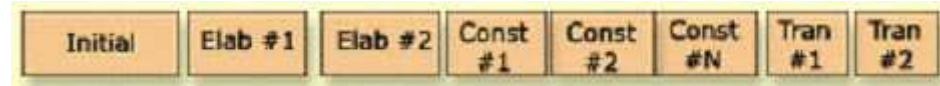
Průběh iterace



1. Plánování cíle iterace (funkčnost)
2. Doplnění / zpřesnění požadavků
3. Dotváření návrhu
4. Implementace funkčností
5. Integrace přírůstku a otestování
6. Nasazení do provozu
 - » release interní / externí

... vodopád v malém

Kolikrát iterovat?



- Délka iterací
 - pevně daná (SCRUM)
 - variabilní v okamžiku plánování (XP, UP, ...)
 - malá – blízký cíl, menší složitost/riziko, rychlá adaptace
 - » 1-4 týdny pro malé, 3-6 týdnů velké projekty, někdy měsíce
- Počet
 - dle potřeby, obvykle alespoň 3
- Běžící iterace uzavřená změnám zvenčí
 - » nutné pro stabilitu projektu
 - možný tlak na změnu: čas, funkčnost, postup
 - neakceptovat ani od šéfů (viz SCRUM)



Timeboxing iterací

- Délka \Rightarrow datum ukončení pevné
 - omezení plánované funkčnosti možné
 - viz dále plánování
 - nehotový release, změna datumu neakceptovatelné
 - nesmí být tlak na přesčas
- Výhody
 - zacílení iterace
 - lidé si pamatují překročené termíny, ne opuštěné vlastnosti
 - nutí včas k těžkým rozhodnutím a kompromisům
 - vysoká produktivita: 80 vs 25 FP/měs

SCRUM: 30 dní
XP: 1-2 týdny



Globální řízení iterativního vývoje

- Oddělené sekvenční fáze

- » analogie „klasických“ inženýrských disciplin
- » jasné rozdělení cílů a výsledků

- milníky

- » po stupních přesnosti, míře rizika
- » vodopád: po činnostech 💣

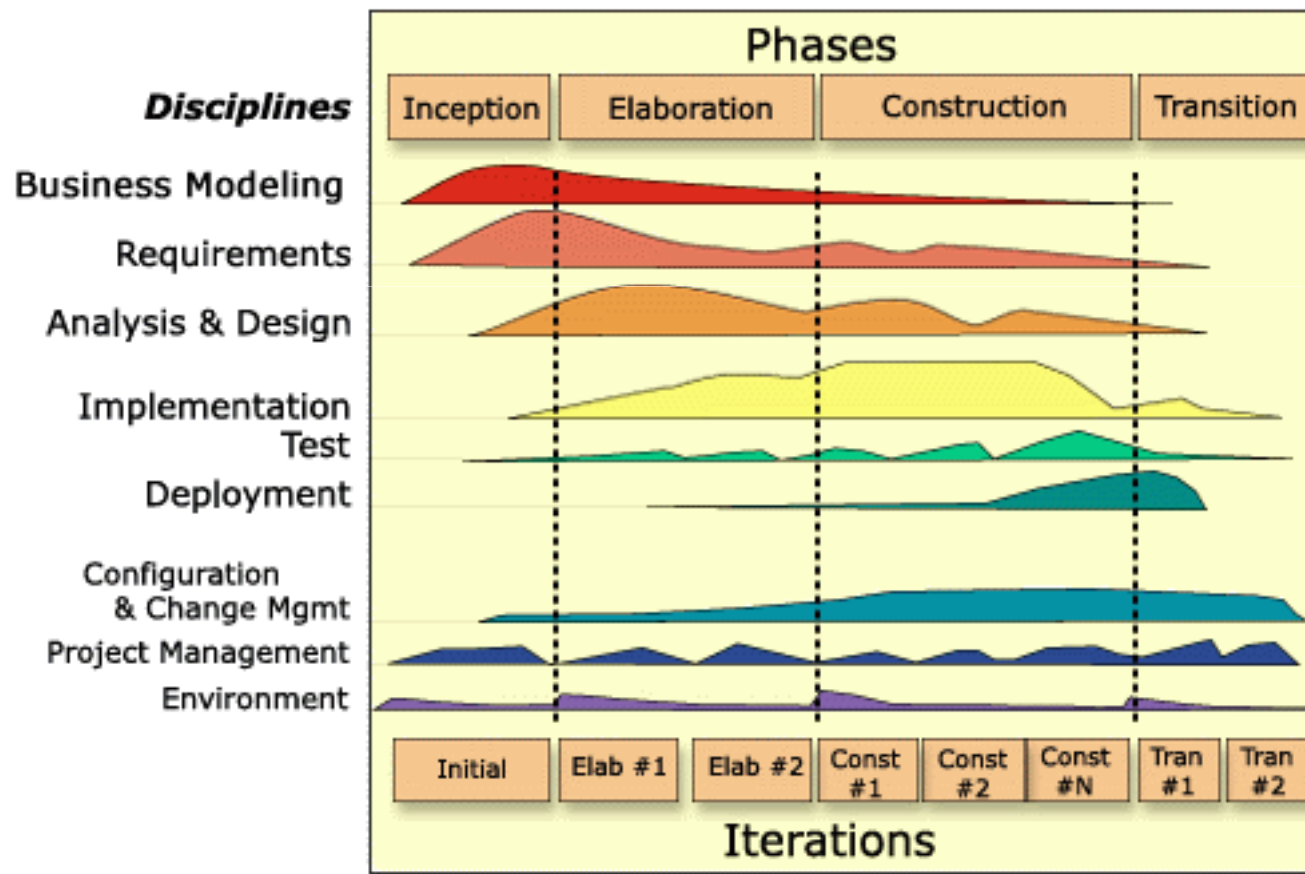
- 1 fáze = 1..N iterací

- Inicializace projektu

Barry Boehm (1995): *Anchoring the Software Process*

- LCO - Lifecycle Objectives
- LCA - Lifecycle Architecture
- IOC - Initial Operational Capability
- REL - Product Release

Fáze vývoje: příklad RUP





Evoluční a adaptivní vývoj

... jeden z 4 nejčastějších faktorů úspěchu sw projektů

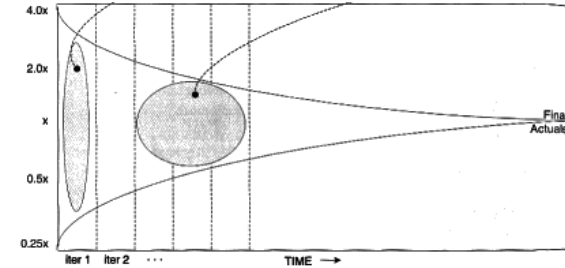
■ Evoluční vývoj

- dotažení iterativního přístupu
- znalosti o požadavcích, návrhu, odhadech a plánu se vyvíjejí/zpřesňují v průběhu projektu
 - » žádné kompletní, dále neměnné specifikace na začátku (20-80)
 - » míra změny obvykle klesá s postupujícími iteracemi
- „don't develop software, grow it“

■ Adaptivní

- zdůraznění zpětné vazby v evolučním vývoji
 - » analogie řízení auta
- zejména evoluční dodávky – zpětná vazba od uživatelů

Adaptivní plánování

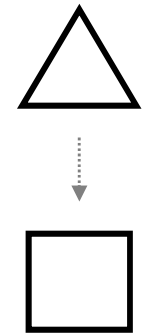


- Prediktivní plánování: velká míra nejistoty
 - » „plan work, work plan“
 - neznalost odhadů v době, kdy jsou potřeba
 - měnící se požadavky \Rightarrow rozsah projektu
- Řešení
 - přesnější odhady a plán až po několika iteracích
 - detailně plánovat jen na co máme rozumně přesná data
 - » obvykle nejvýše pro následující iteraci
 - » plus hrubé milníky (dodávky zákazníkovi)

Stupně volnosti při plánování

*Cheap. Fast. Good.
Choose any two.*

- Klasicky: čas, zdroje (cena), kvalita
 - obtížně měnitelné, odhadované
 - kvalita obtížně říditelná
 - » typický požadavek: „bude to v termínu, s daným rozpočtem, a v bezchybné kvalitě jako vždy“ ... „you get crappy SW late“
- Agilně: +funkčnost
 - nejlepší faktor pro řízení projektu
 - » první tři pevné, funkčnost nejsnáze měnitelná
 - vhodná granularita ⇒ snadné a přesné odhady



Iteration Plan

- Shows timeframes and resources by discipline

Outline of an Iteration Plan

1. Objectives
2. Scope
3. References
4. Plan
 - 4.1 Iteration Activities
 - 4.2 Iteration Schedule
 - 4.3 Iteration Deliverables
5. Resources
 - 5.1 Staffing Resources
 - 5.2 Financial Resources
 - 5.3 Equipment & Facilities Resources
6. Use Cases
7. Evaluation Criteria

Iteration Schedule section for Requirements discipline

Requirements	40 days	Tue 12/1/98	Mon 1/25/99	
Develop Vision	25 days	Tue 12/1/98	Mon 1/4/99	System Analyst
Elicit Stakeholder Requests	4 days	Tue 1/5/99	Fri 1/8/99	System Analyst
Manage Dependencies	26 days	Tue 12/1/98	Tue 1/5/99	System Analyst
Capture a Common Vocabulary	10 days	Wed 12/23/98	Tue 1/5/99	System Analyst

Riziky a klientem řízený vývoj

Již z dob spirálového modelu
(1986)

» Kontext: plán iterace (výběr funkčnosti)

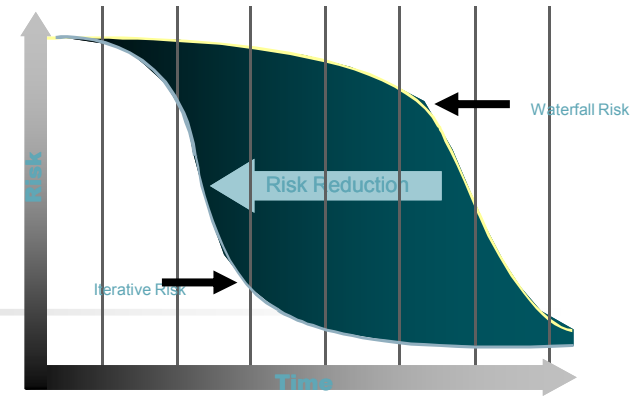
■ Řízení riziky

- vyhodnotit rizikové faktory projektu
 - » designová/architektonická rizika, obchodní, legislativní, neznámá funkčnost, použitelnost, ...
- začít s částmi funkčnosti/designu s největší mírou rizika

■ Řízení prioritami klienta

- výběr funkčnosti je na zákazníkovi
 - » množství funkcí omezeno délkou iterace
- umožňuje pružně reagovat na aktuální potřeby

Risk Terms



- **Direct risk** - the project has a large degree of control
 - **Indirect risk** - the project has little or no control

 - **Risk Magnitude** is used for ranking risks. It is a combination of:
 - Probability of occurrence
 - Impact on the project (severity) e.g. project delays

 - Týmové rozhodování: dot voting
-



Výsledek: Empirický proces

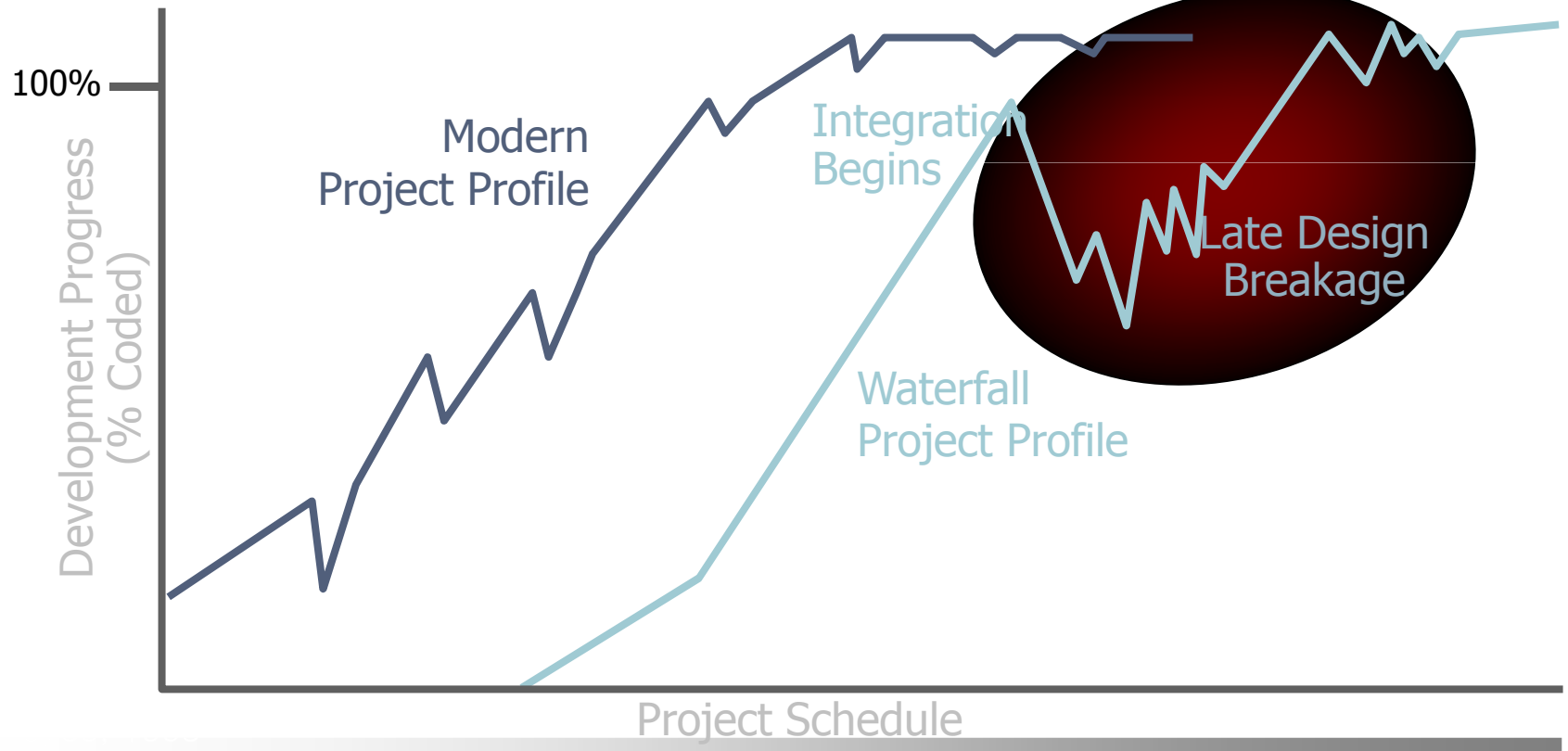
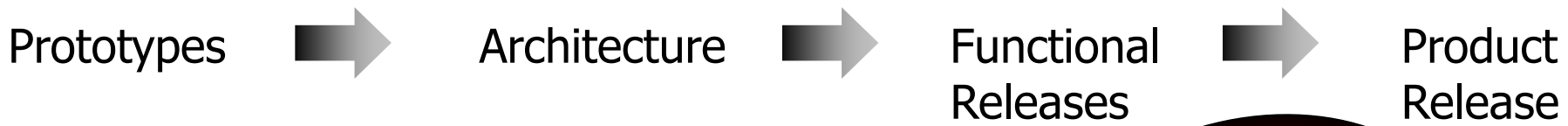
- Definovaný proces („rule-based“)
 - předem známé/dané aktivity, jejich návaznost
 - PERT diagram
- Empirický („principle-based“)
 - » uznání, že vývoj software není sériová výroba
 - sada jednoduchých aktivit
 - časté měření procesu a zpětná vazba
 - dynamická adaptace na změny a události
 - » emergent behaviour, samoorganizující tým
 - přizpůsobení typu (závažnosti) projektu

SCRUM: denní setkání týmu, „empowered team“

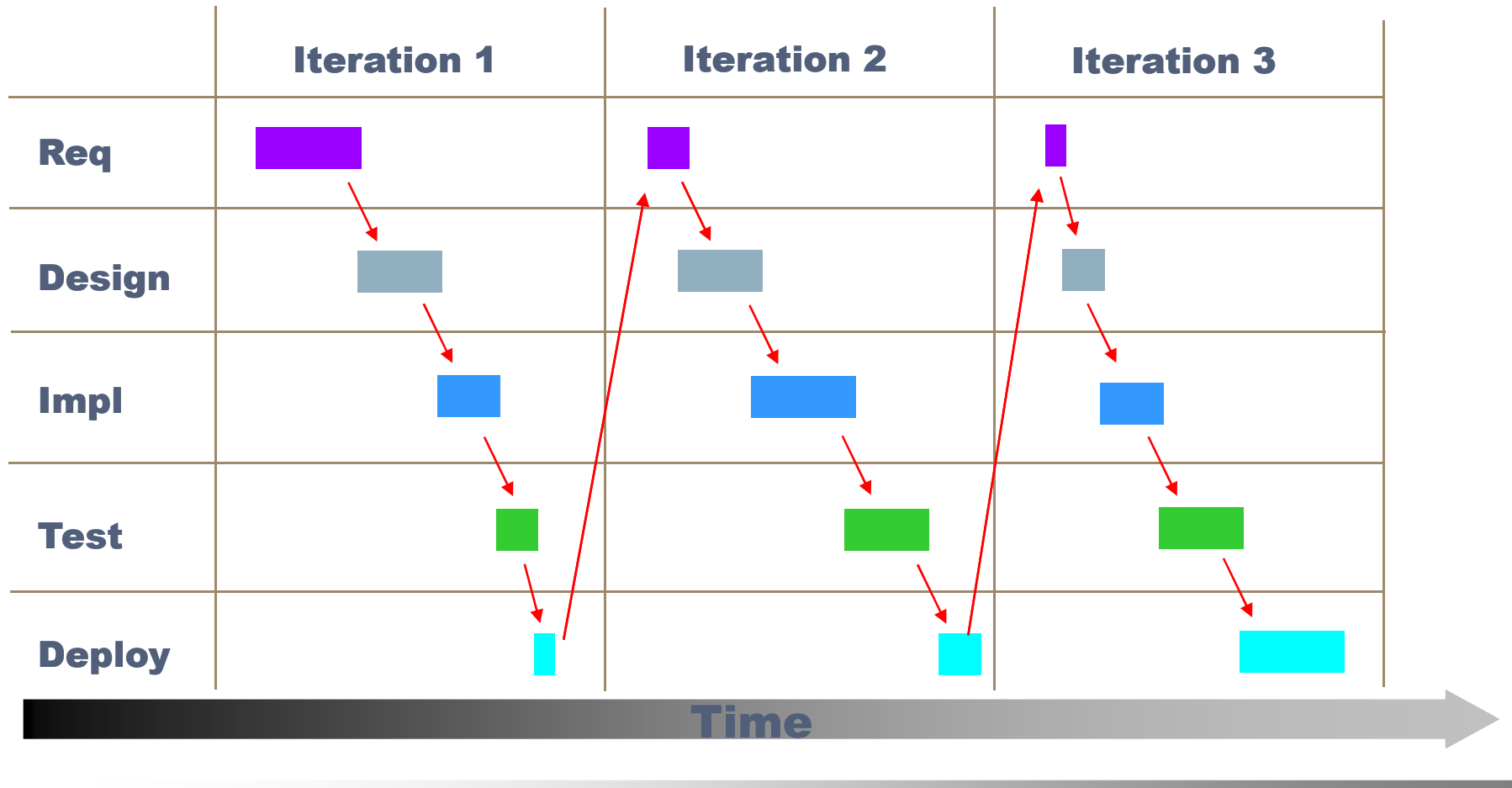
XP: denní setkání týmu, role „tracker“, plánovací hra



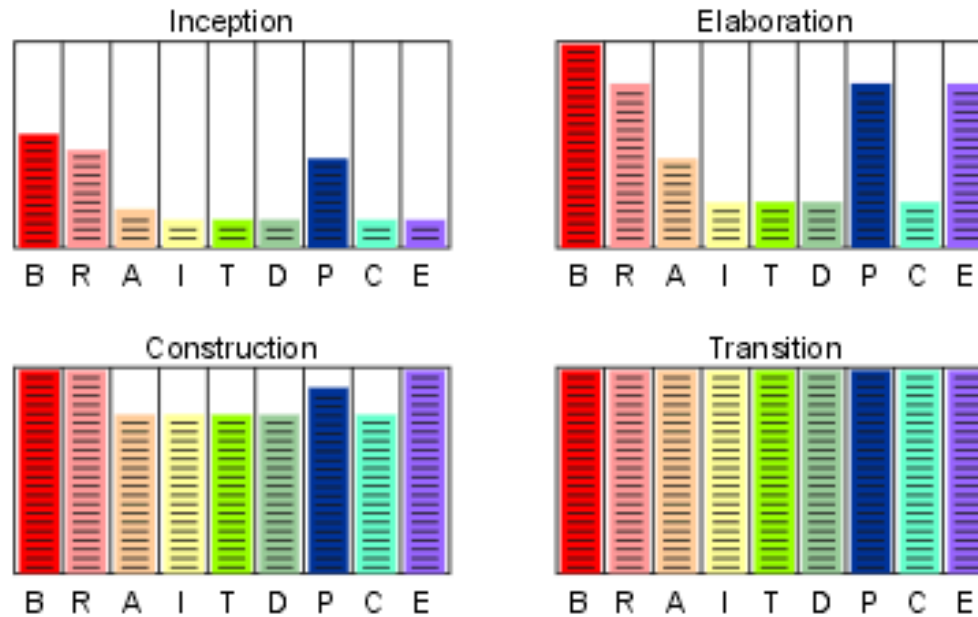
Better Progress Profile



Changing Focus Over Time



Lifecycle Evolution of Artifacts

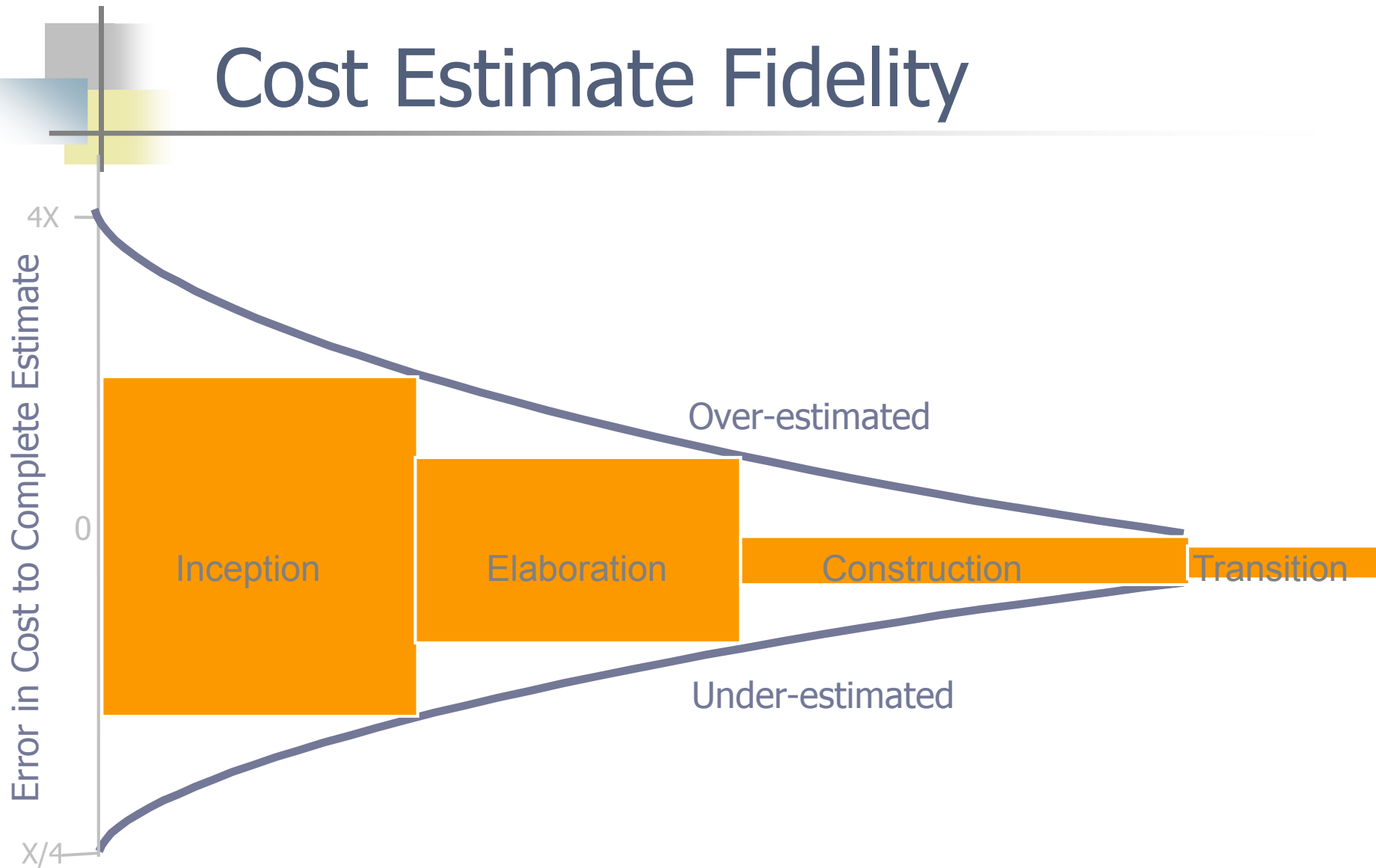


- Artifact sets mature over time.

■ B : Business Modeling Set	■ D : Deployment Set
■ R : Requirements Set	■ P : Project Management Set
■ A : Analysis & Design Set	■ C : Configuration & Change Management Set
■ I : Implementation Set	■ E : Environment Set
■ T : Test Set	

Information set evolution over the development phases.

Cost Estimate Fidelity



Shrnutí

Achieving Key Practices: Through a Process

- Iterative approach preferred
- Guidance for activities and artifacts
- Process focus on architecture
- Use cases which drive design and implementation
- Models which abstract the system

