

NÁSKOK
DÍKY
ZNALOSTEM

PROFINIT

Softwarový proces

Kolektiv autorů

Únor 2020



Úvod


Základní pojmy

- › Softwarový proces / Model životního cyklu vývoje software (SDLC, Software Development Lifecycle)
 - Množina aktivit nutných k tomu, aby software vznikl.
 - Jejich souslednost, opakování, vstupy a výstupy jednotlivých aktivit a nároky na jejich provedení

- › Vždy je potřeba udělat
 - Specifikaci – co bude systém dělat, jak se změní
 - Architekturu a design – z jakých „kostek“ a jak se bude systém skládat
 - Implementaci – vlastní výroba systému
 - Validaci – ověření, že systém dělá co má



PROJECT MANAGEMENT / QUALITY ASSURANCE / DOCUMENTATION / CONFIGURATION MANAGEMENT / RELEASE MANAGEMENT / DEVOPS



Modely softwarového procesu

Waterfall

› Oddělené fáze

- Analýza požadavků
- Design
- Implementace
- Testování
- Provoz a údržba

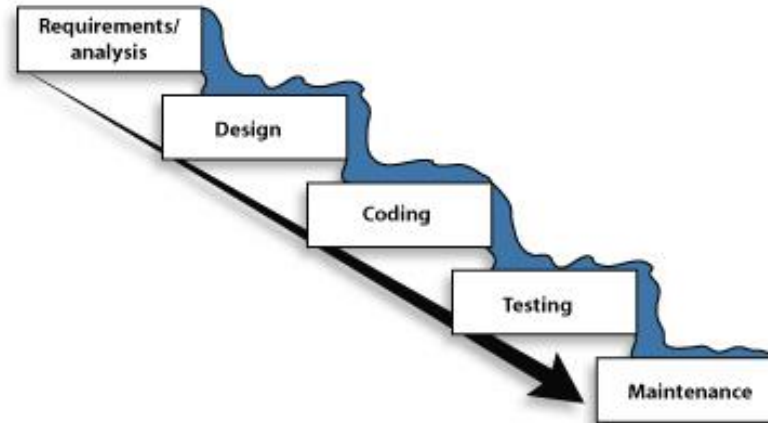
› Výhody

- Jasně definovaný plán
- Predikovatelnost (čas, rozsah, cena)
- Snadná koordinace práce

› Nevýhody

- Nutno chápat, co se chce již na začátku
- Reakce na změny (požadavků, termínů, ...)
- Rychlost dodávky (kdy zákazník něco uvidí od zadání)
- Integrace více systémů

The classic waterfall development model



Iterativní

› Změny oproti vodopádu

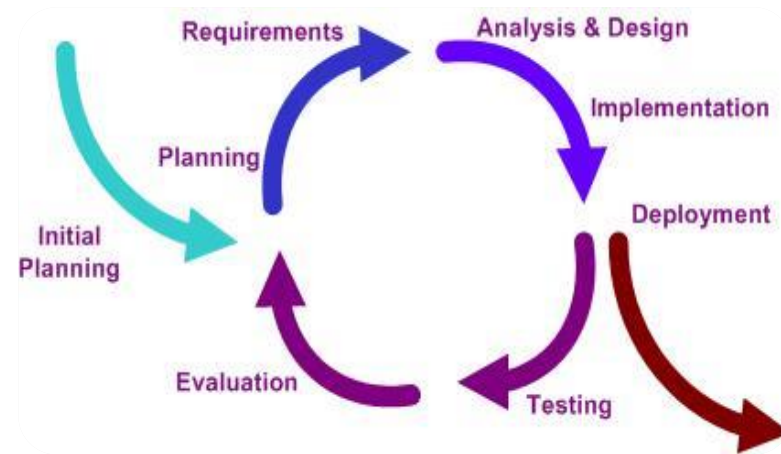
- Několik verzí systému
- Jednotlivé verze se dělají vodopádem

› Výhody

- Jasně definovaný plán
- Predikovatelnost (čas, rozsah, cena)
- Snadná koordinace práce
- Zákazník má přístup k verzím/prototypům, čili vidí co dostane

› Nevýhody

- Nutno chápat, co se chce na již začátku, možné změny do dalších verzí
- ~~– Reakce na změny (požadavků, termínů, ...)~~
 - Lepší než vodopád, ale změny nejsou dodány rychle
- ~~– Rychlost dodávky (kdy zákazník něco uvidí od zadání)~~

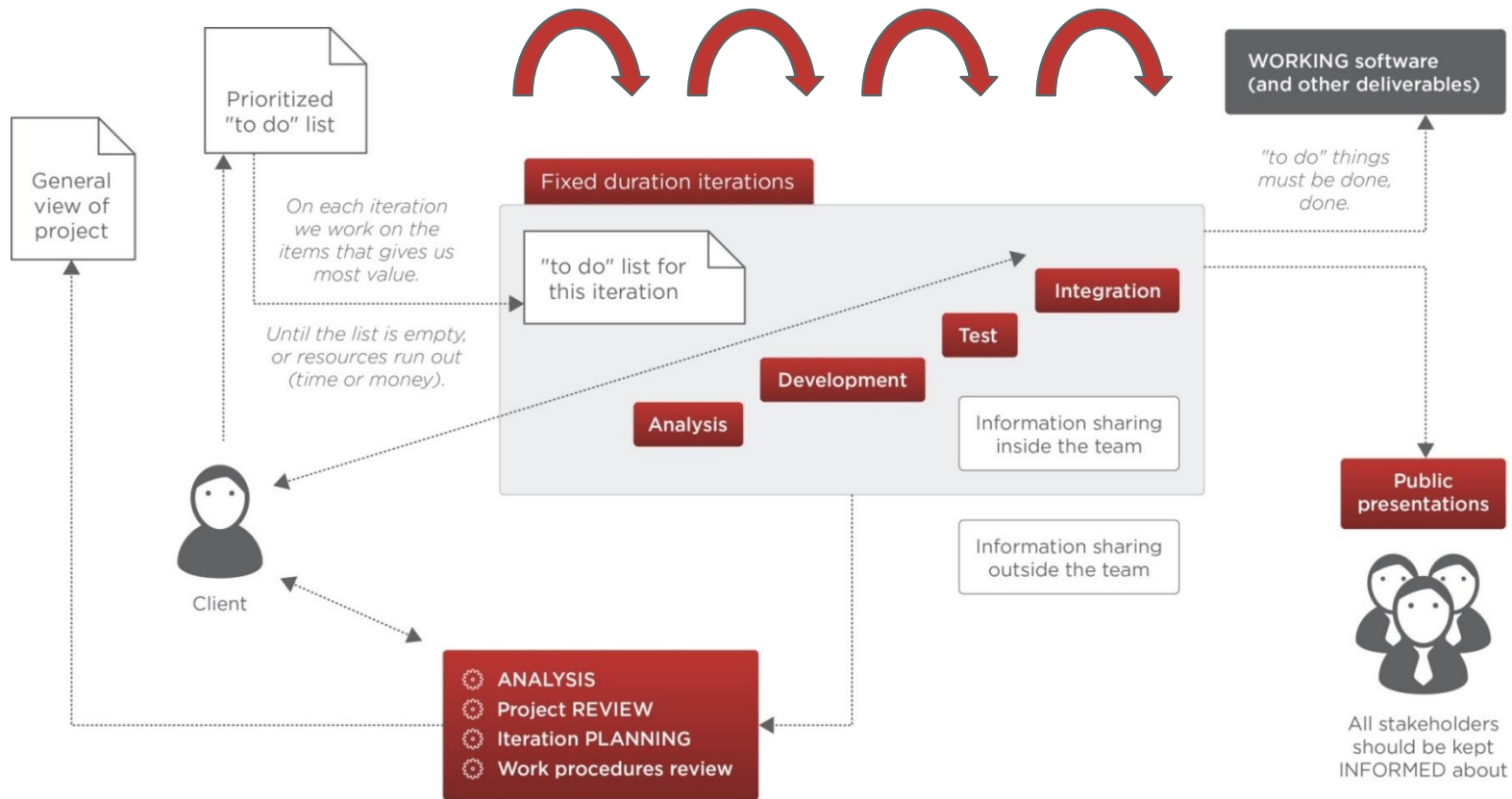


Agilní

- › **Změny oproti iterativnímu**
 - Mnohem kratší iterace
 - Jednotlivé verze ne vždy produkční
 - Velké nároky na celý tým
 - Změna myšlení
- › **Výhody**
 - Rychlé
 - Nové verze mám brzy, zpětná vazba
 - ~~Predikovatelnost (čas, rozsah, cena)~~
- › **Nevýhody**
 - Nutné kontinuální zapojení všech členů týmu
 - Nutný silný business vlastník




Princip agilního vývoje



Scrum



The background consists of numerous overlapping, semi-transparent, light gray geometric shapes, primarily cubes and rectangular prisms, creating a complex, layered, and three-dimensional effect. The shapes are scattered across the white background, with some appearing more prominent than others due to their position and opacity.

7 kroků,
jak agilní přístup
nepohrbit

Agilní (Scrum apod.)

AGILE

Manifesto for Agile Software Dev.

- INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

12 principů agilního vývoje

- › Customer satisfaction by **early and continuous delivery** of valuable software
- › **Welcome changing requirements**, even in late development
- › **Working software** is delivered **frequently** (weeks rather than months)
- › Close, **daily cooperation** between business people and developers
- › Projects are built around **motivated individuals**, who should be trusted
- › Face-to-face conversation is the best form of communication (co-location)
- › Working software is the primary measure of progress
- › **Sustainable development**, able to maintain a constant pace
- › **Continuous attention to technical excellence and good design**
- › **Simplicity**—the art of maximizing the amount of work not done—is **essential**
- › Best architectures, requirements, and designs emerge from self-organizing teams
- › **Regularly, the team reflects** on how to become more effective, and adjusts accordingly

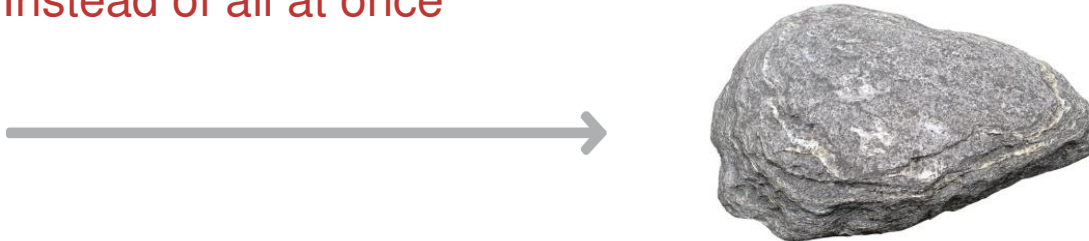
Proč agile?

- › Možnost reagovat na změny
- › Možnost pracovat s vizí místo pevného zadání
- › Možnost průběžně si zkusit nápady i řešení

Incrementally



Instead of all at once



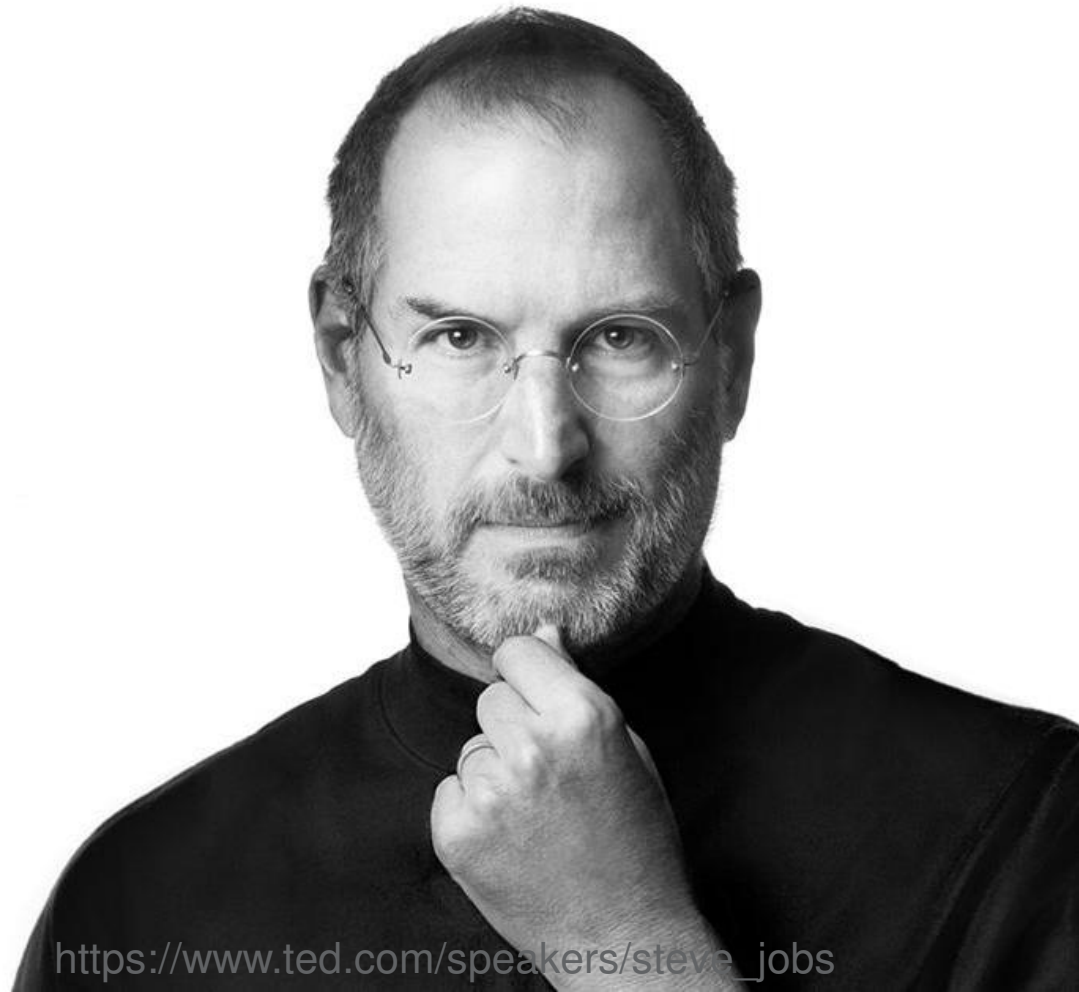
1. Business + IT

- › Business je součástí procesu tvorby software
- › Není to nákup auta



2. Silný product owner

- › Má vizi produktu
- › Má čas se tomu věnovat
- › Je schopný dělat rozhodnutí



https://www.ted.com/speakers/steve_jobs

3. Omluva pro absenci procesu

- › Neznamená to...
 - nemít žádný plán a žádný proces
 - nedodržovat termíny
 - nepsat dokumentaci



4. Doing agile vs being agile

- › Koncentrace na praktiky místo na podstatu
- › Cílem je dodávat **funkční software**, ne dodržovat metodiku
- › Napodobování věcí, které fungují jinde..., ale víme proč?



4. Doing agile vs being agile



5. Agile jako silver-bullet

- › Nepomůže vyřešit všechny problémy, které organizace má
- › Může rovněž skončit neúspěchem
- › Nepomůže, pokud není jasná vize
- › Nehodí se pro všechny situace

NO



7 častých problémů

1. Business + IT
2. Silný product owner
3. Omluva pro absenci procesu
4. Doing agile vs being agile
5. Agile jako silver-bullet
- 6. Marketing do firmy**
- 7. Agile a fixování rozsahu**

The background of the slide is a complex, abstract composition of numerous overlapping, translucent gray geometric shapes. These shapes, which include various polygons and rectangular forms, are scattered across the white background, creating a sense of depth and movement. The shapes are semi-transparent, allowing the ones behind them to be visible, which adds to the layered, architectural feel of the design.

Agile & FTFP

Agile & FTFP?



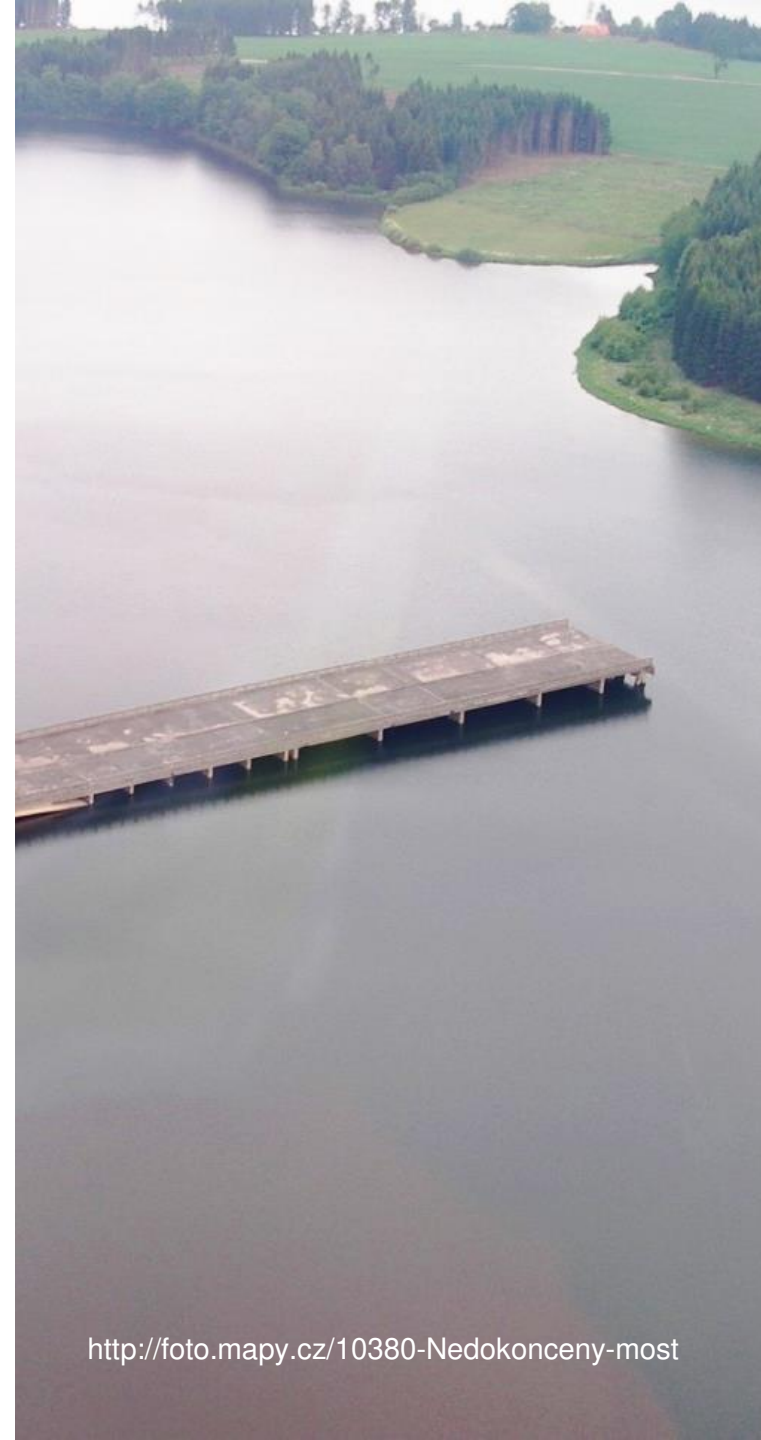
Fixed Time, Fixed Price, Variable Scope

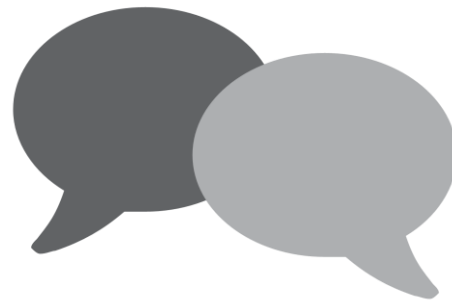
- › S rozsahem se bude pracovat!
- › User stories
- › Minimum Viable Product



Jak z toho ven?

- › Používat zdravý selský rozum
- › Definovat si, co agile znamená pro vaši organizaci/projekt
- › Vysvětlit to celému týmu





Diskuze

NÁSKOK
DÍKY
ZNALOSTEM

PROFINIT

Softwarový proces a jeho zlepšování

Martin Hlavatý

Únor 2020

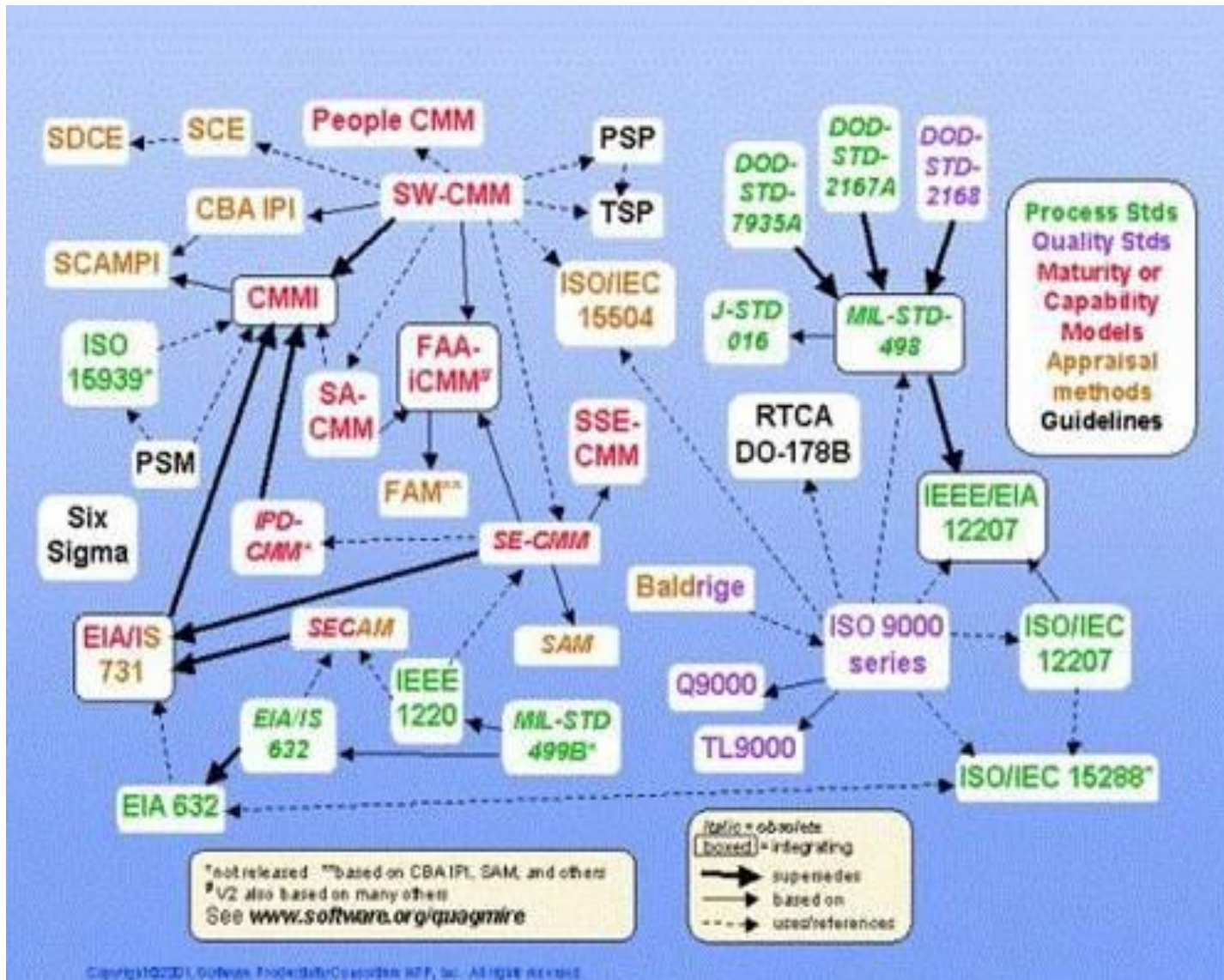
Základní koncept SPI

PDCA model (Deming cycle)

- › **Plan**
 - Prověřit současnou výkonnost
 - Posoudit problémy, omezení
 - Navrhnout řešení
 - Naplánovat provedení
- › **Do**
 - Otestovat účinnost řešení
- › **Check**
 - Zhodnotit výsledky testu
 - Posoudit dosažení výsledků
 - Zaměřit se na překážky bránící zlepšení
- › **Act**
 - Rozpracovat konečné řešení, aby bylo kdekoli použitelným přístupem



Mnoho přístupů



Základní přístupy

- › Systematický, dlouhodobý přístup
 - ISO, CMM, CMMI, ... (prescriptive)
 - Preskriptivní přístupy vychází z premisy, že správné praktiky vývoje SW jsou známy, a tyto předepisují.
 - SEL/NASA, ... (inductive)
 - Induktivní vycházejí z premisy, že zlepšovat se má to, co odpovídá stavu, cílům, problémům, atd. konkrétní dané organizace.
- › Best practices

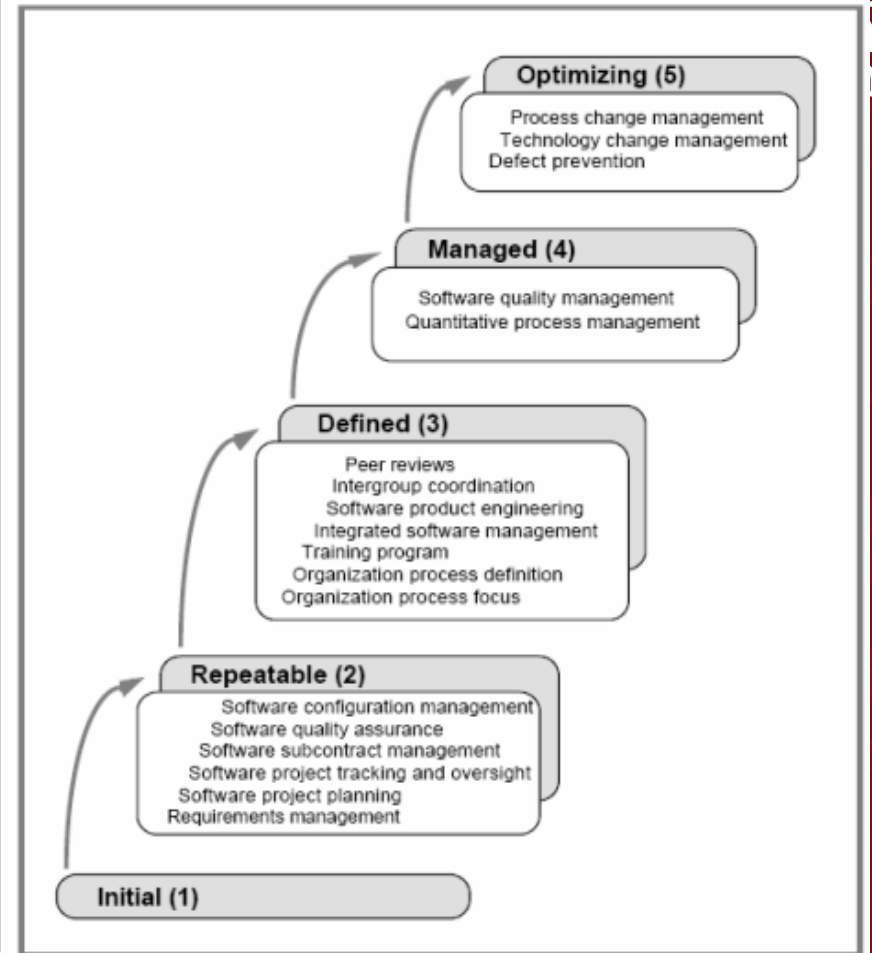
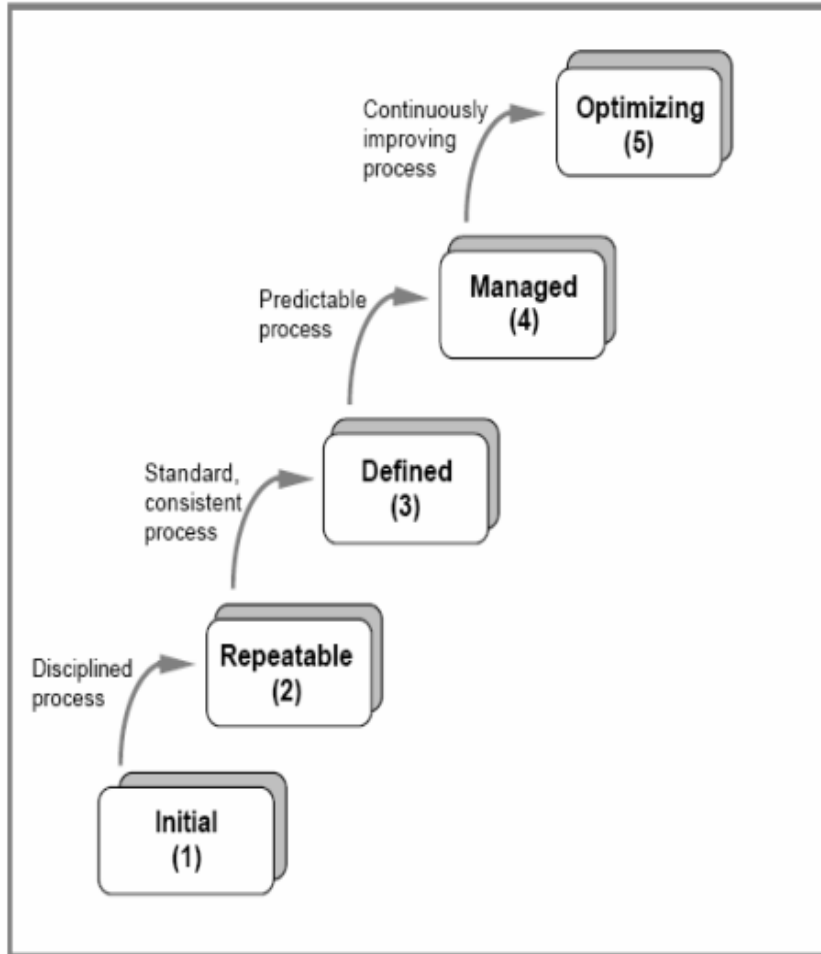
Rodina ISO 9000

ISO 9001:2008

- › Požadavky na Quality Management System
- › Standard

ISO/IEC 90003:2004

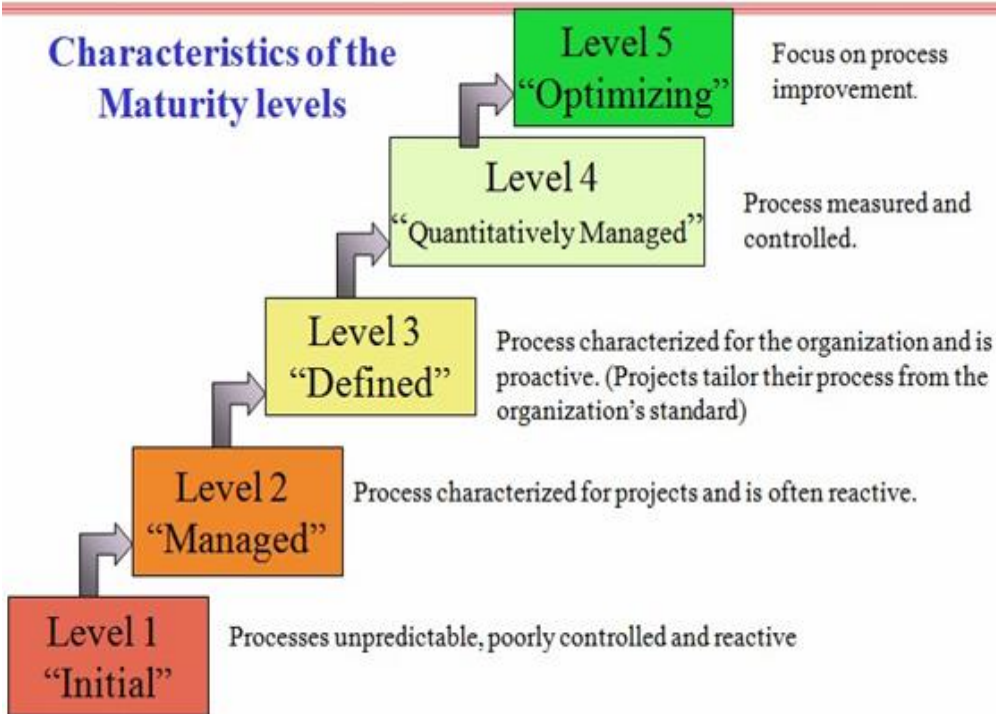
- › Návod pro aplikaci ISO 9001 na SW



CMMI

- › Nástupce staršího CMM
- › 5 úrovní, 22 key process areas (KPA)
- › Staged vs. Continuous
- › 3 různé modely – Development, Acquisition, Services

Characteristics of the Maturity levels



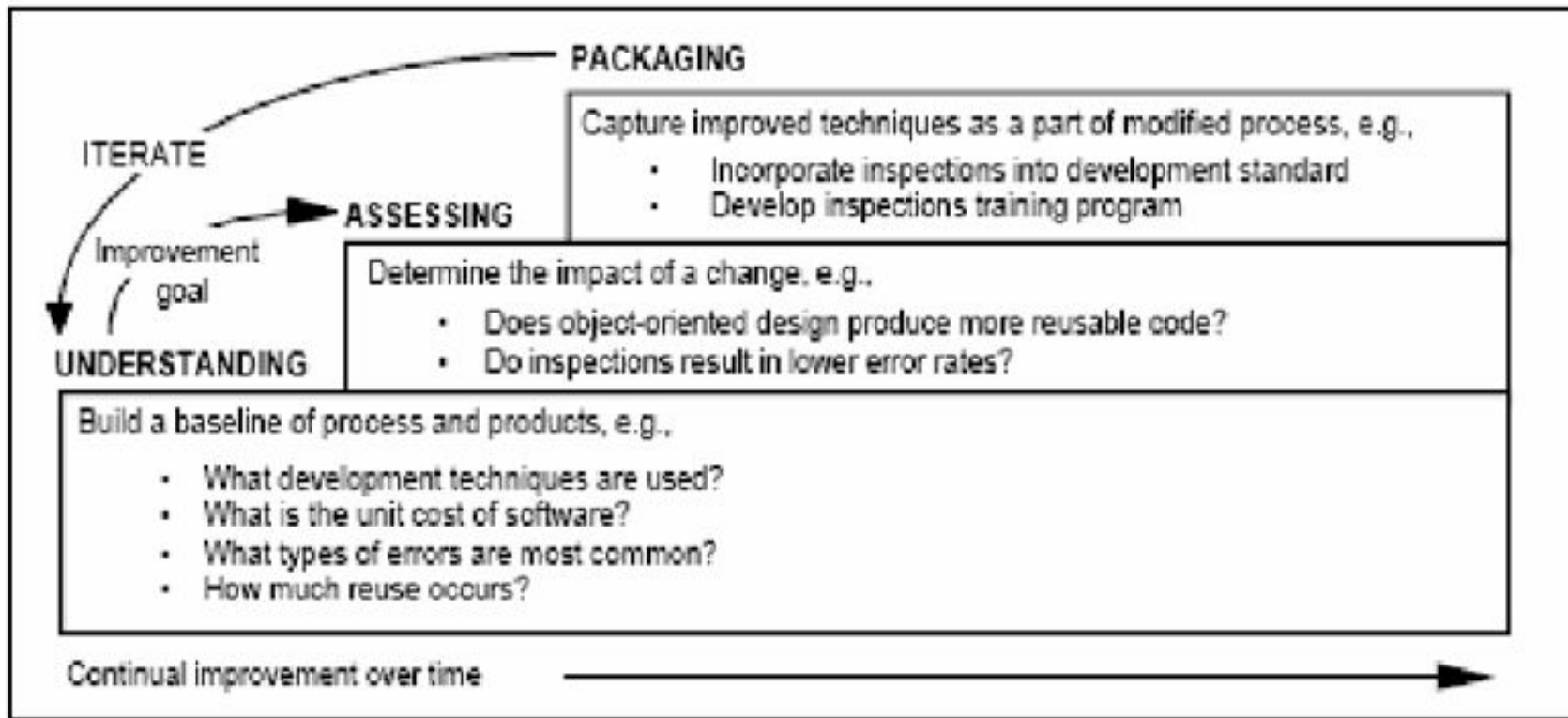
Level	Focus	Process Areas	Result
5 Optimizing	<i>Continuous process improvement</i>	Organizational Innovation & Deployment Causal Analysis and Resolution	Productivity & Quality
4 Quantitatively Managed	<i>Quantitative management</i>	Organizational Process Performance Quantitative Project Management	
3 Defined	<i>Process standardization</i>	Requirements Development Technical Solution Product Integration Verification Validation Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management Risk Management Decision Analysis and Resolution	
2 Managed	<i>Basic project management</i>	Requirements Management Project Planning Project Monitoring & Control Supplier Agreement Management Measurement and Analysis Process & Product Quality Assurance Configuration Management	
1 Initial	<i>Competent people and heroics</i>		

SEL / NASA

Základní premisa

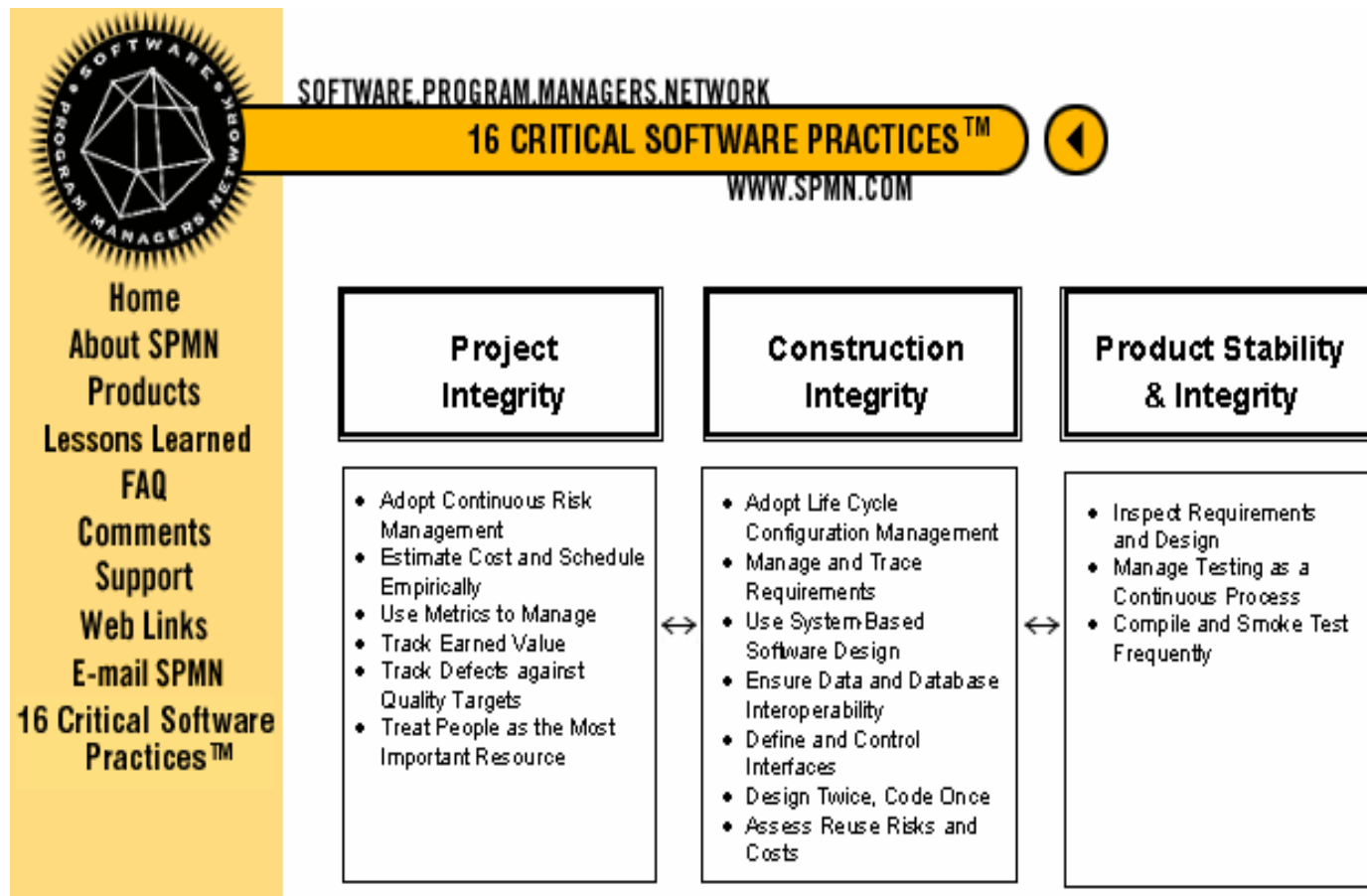
Vývojová organizace musí úsilí o zkvalitňování zaměřovat na

- zamezení minulých problémů a
- opakování minulých úspěchů.



SPMN best practices

- Reakce na problémy se zvládáním velkých projektů
- Přímou použitelné strategie, techniky a praktiky
- „SPMN books“ (nutná registrace)



Shrnutí

Je třeba:

- › znát současný stav vlastního procesu vývoje a jeho charakteristiky
- › znát problémy s ekonomickým projevem a jejich vážnost
- › mít názor, které problémy je nutné a možné odstranit
- › mít názor, jak modifikovat proces vývoje
- › mít prostředky, jak tuto modifikaci prosadit

... a znovu na začátek ...

Děkujeme za pozornost

PROFINIT

NÁSKOK DÍKY ZNALOSTEM

Profinit EU, s.r.o.
Tychonova 2, 160 00 Praha 6



Telefon
+ 420 224 316 016



Web
www.profinit.eu



LinkedIn
linkedin.com/company/profinit



Twitter
twitter.com/Profinit_EU

The background consists of a dense, overlapping field of translucent, light gray geometric shapes, primarily rectangular and polygonal. These shapes are layered and oriented in various directions, creating a complex, three-dimensional effect. The overall composition is abstract and modern, with a clean, minimalist aesthetic.

Backup
slides



Modely
z různých
pohledů

Flexibilita

- › Flexibilita
 - Reakce na změny
 - Rychlost a náklady
- › Vodopád
 - Velmi nepružný
 - Vysoké náklady
- › Iterativní
 - Lze zakomponovat změny do další iterace
 - Náklady nižší než u vodopádu
- › Agilní
 - Snadné, očekávají změny
 - Náklady na změny velmi nízké



Predikovatelnost

- › Predikovatelnost
 - Vím, co dostanu
 - Kdy
 - Kolik mě to bude stát
- › Vodopád
 - Vysoká, jedeme dle plánu
- › Iterativní
 - Vysoká, máme plán
- › Agilní
 - Nízká, plán jen na krátké období
 - Víme tedy jen co dostaneme např. během dalšího sprintu



Architektura a design

- › Architektura a design
 - Dobře navržený systém
 - Konzistentní dodržování principů návrhu
- › Vodopád
 - Vysoká
- › Iterativní
 - Vysoká
 - Riziko zanesení problémů při dalších iteracích
- › Agilní
 - Nízká
 - Riziko zanesení problémů při každém sprintu



Implementace

- › Implementace
 - Prostor pro dodání kvalitního díla
 - Požadavky na programátory
- › Vodopád
 - Kvalitní, dost prostoru pro QA
 - Revize, coding standards, ...
- › Iterativní
 - Kvalitní, dost prostoru pro QA
 - Revize, coding standards, ...
 - Riziko zanesení problémů při dalších iteracích
- › Agilní
 - Nutný kvalitní tým, může být problematická
 - Riziko nekvalitní práce, když není prostor na revize



Dokumentace

- › Dokumentace
 - Mám kvalitní a konzistentní dokumentaci systému
- › Vodopád
 - Typicky OK
- › Iterativní
 - Nutno dodržet napříč verzemi (jedna pravda či více verzí?)
- › Agilní
 - Nízká
 - Obtížné udržovat napříč sprinty



Spolupráce s klientem

- › Požadavky na součinnost
 - Jak moc a často se musí zákazník podílet na projektu
- › Vodopád nebo iterativní
 - V přesně definovaných okamžicích
 - Lze dobře plánovat
- › Agilní
 - V průběhu celého projektu
 - Velké riziko selhání, pokud nebude



Smlouva na dodávku

- › Smlouva na dodávku díla
 - Lze model využít pro dodávku FTFP?
- › Vodopád
 - Ano
- › Iterativní
 - Ano, nutno ošetřit rozsah verzí
- › Agilní
 - Ne, požadavky nejsou dopředu známy
 - Aktuální modely fungování nákupu tomu nejsou nakloněny



The background consists of numerous overlapping, semi-transparent, light gray geometric shapes, primarily cubes and rectangular prisms, scattered across the white background. These shapes are oriented in various directions, creating a complex, layered, and three-dimensional effect.

Který
model
vybrat?

Který vybrat?

- › Vodopád
 - Téměř nepoužíváme pro již uvedené typy projektů (jen pro menší změny)
- › Iterativní
 - Nejčastěji používaný pro většinu systémů našich zákazníků
 - Jak iniciální rozvoj tak údržba
 - Predikovatelnost důležitější než flexibilita
 - Lze na velké systémy
- › Agilní (vývoj nebo dodávky?)
 - Produktový vývoj
 - Vývoj v rámci firmy, kde je jasná ochota lidí se na tom podílet
 - Spíše na menší projekty nebo na některé části procesu (např. jen vývoj)



Věci ke zvážení

- › System
 - Typ a velikost
 - Očekávaná životnost systému
- › Tým dodavatele
 - Znalosti a zkušenosti týmu
 - Schopnost pracovat agilně
- › Agilně
 - Ne vždy a ne všechno
 - **Nesmí být zástěrkou pro chaos**



Zákazník a dodavatel na jedné lodi

- › Silný Product Owner...
- › ... závislý na dokončení projektu



Faktory úspěchu

- › Interní vývoj
- › Nebo team-lease...
- › ... se zapojením interních lidí
- › On-site
- › Prostředí **kontrolované důvěry**

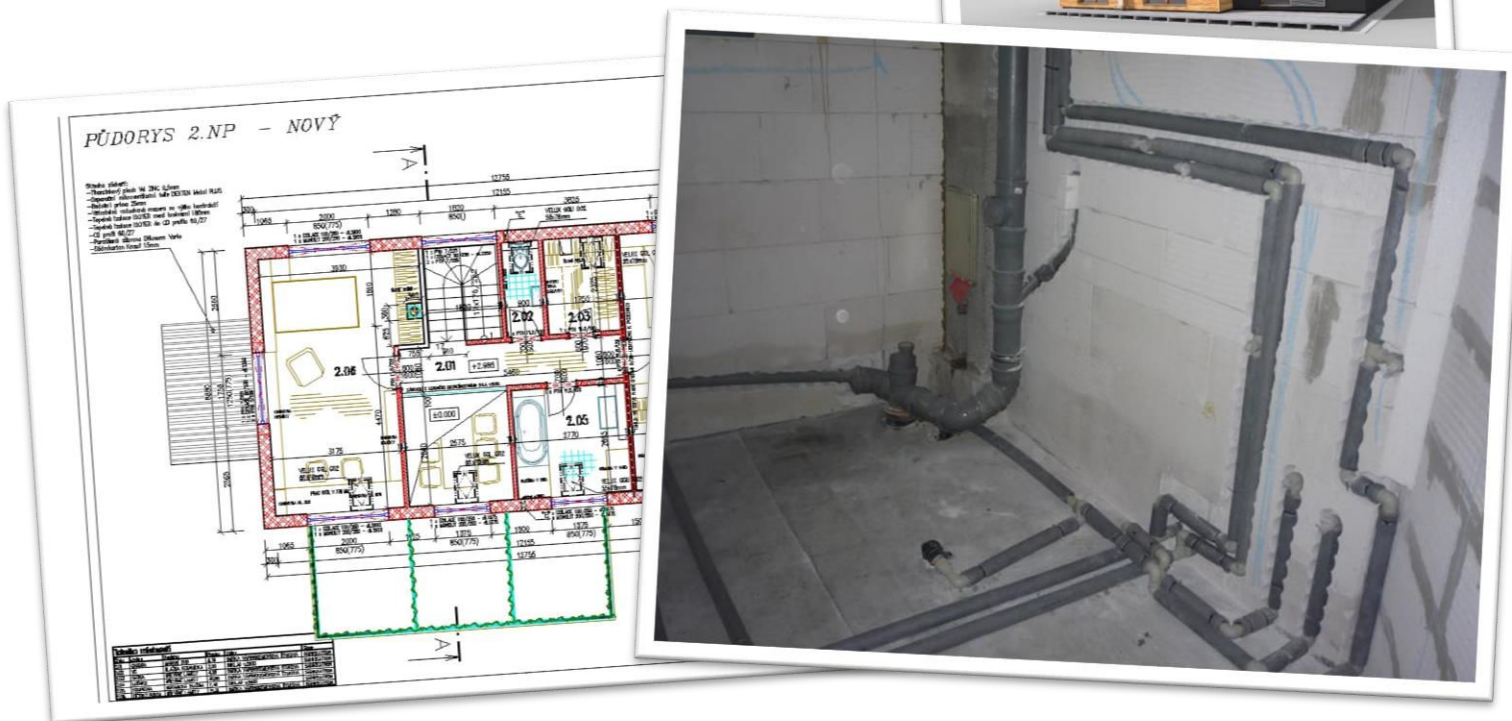


"Trust but Verify."

- Ronald Wilson Reagan

Dokumentace

- › Analytické sprinty
- › Koncepce a architektura
- › User stories a backlog
- › Specifikace během vývoje



Agile & FTFP done right

