

NÁSKOK
DÍKY
ZNALOSTEM

PROFINIT

Softwarový proces

Kolektiv autorů

Únor 2019



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

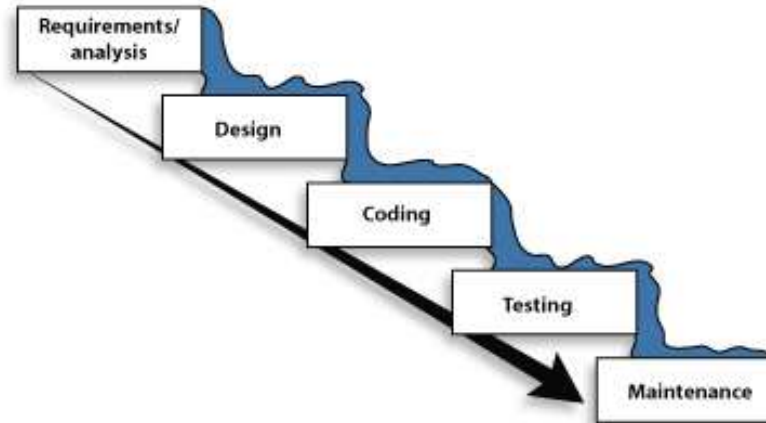
› Výhody

- 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

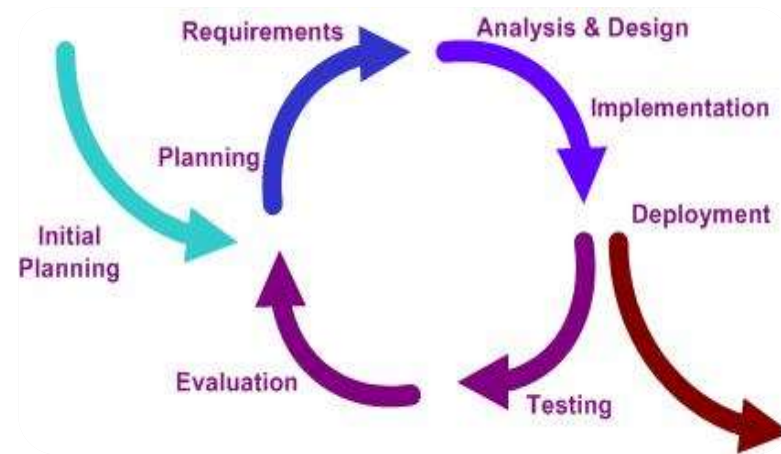
- Několik verzí systému
- Jednotlivé verze se dělají vodopádem

› 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)~~

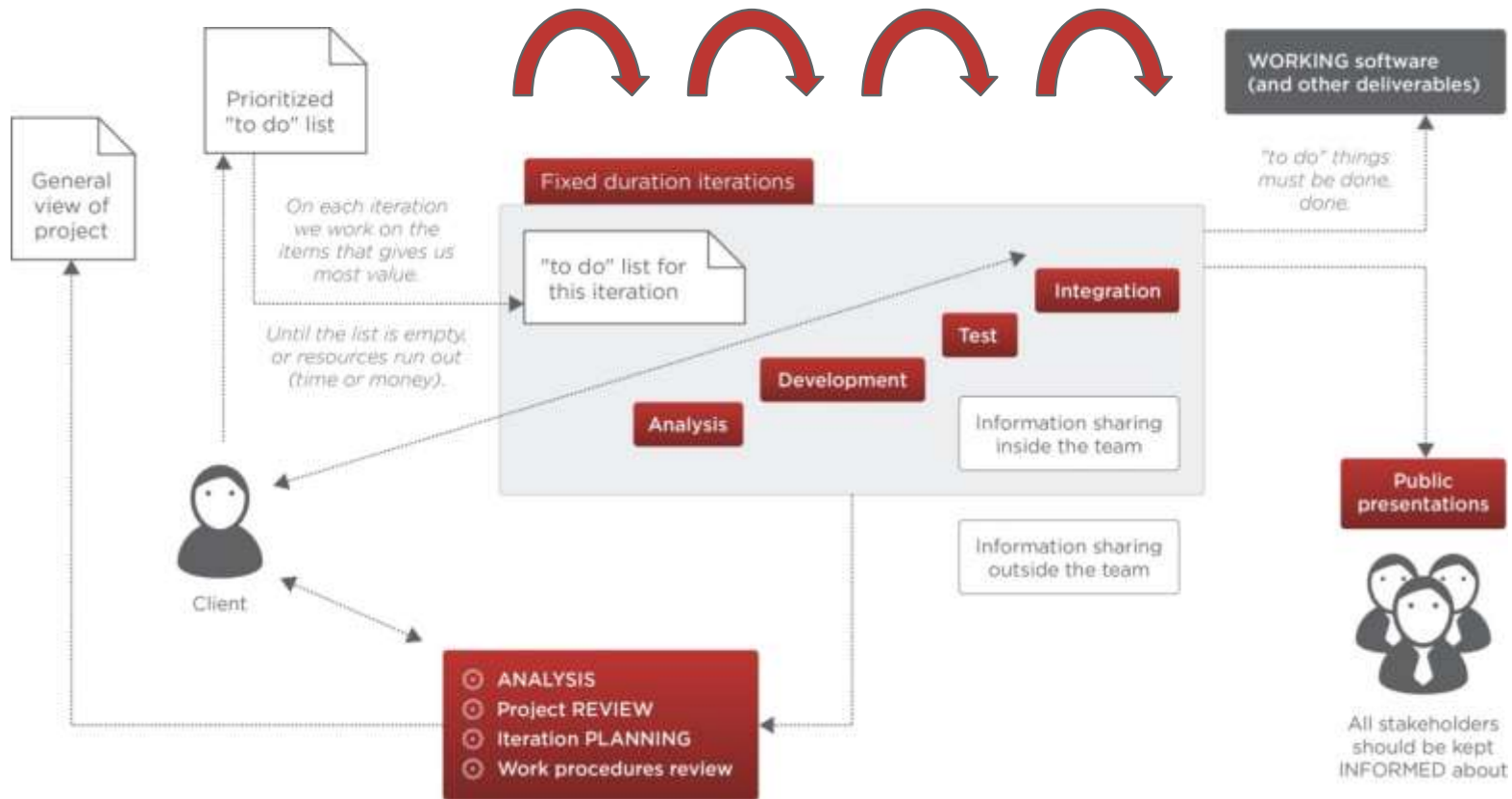


Agilní

- › **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 of the slide is a complex, abstract composition of numerous overlapping, semi-transparent geometric shapes. These shapes, which include various polygons and rectangular forms, are rendered in shades of light gray and white, creating a sense of depth and movement. The shapes are scattered across the frame, with some appearing more prominent than others, and they seem to be layered on top of each other, giving the overall appearance of a dynamic, crystalline or architectural structure.

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

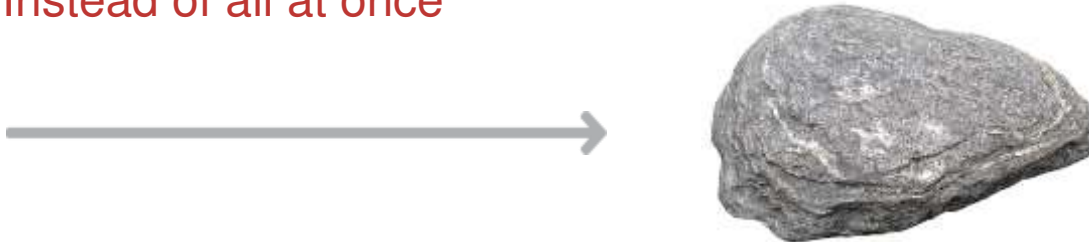
Proč agile?

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

Incrementally



Instead of all at once



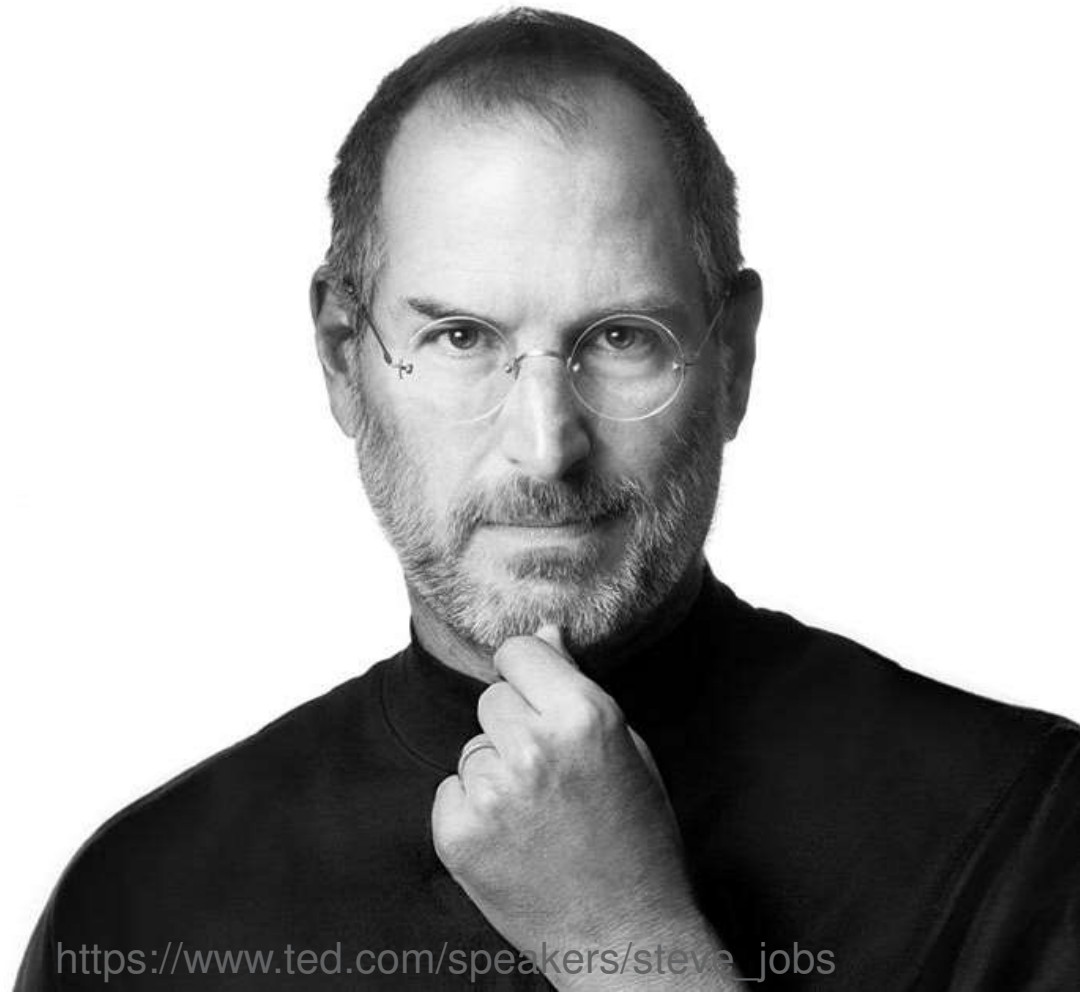
1. Business + IT

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



2. Silný product owner

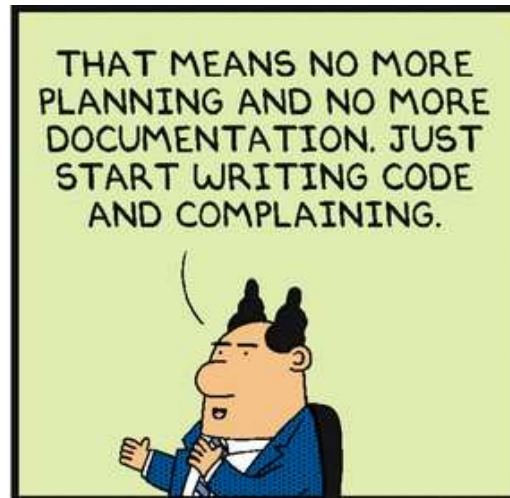
- › Má vizi produktu
- › Má čas se tomu věnovat
- › Je schopný dělat rozhodnutí



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**



Agile & FTFP

Agile & FTFP?



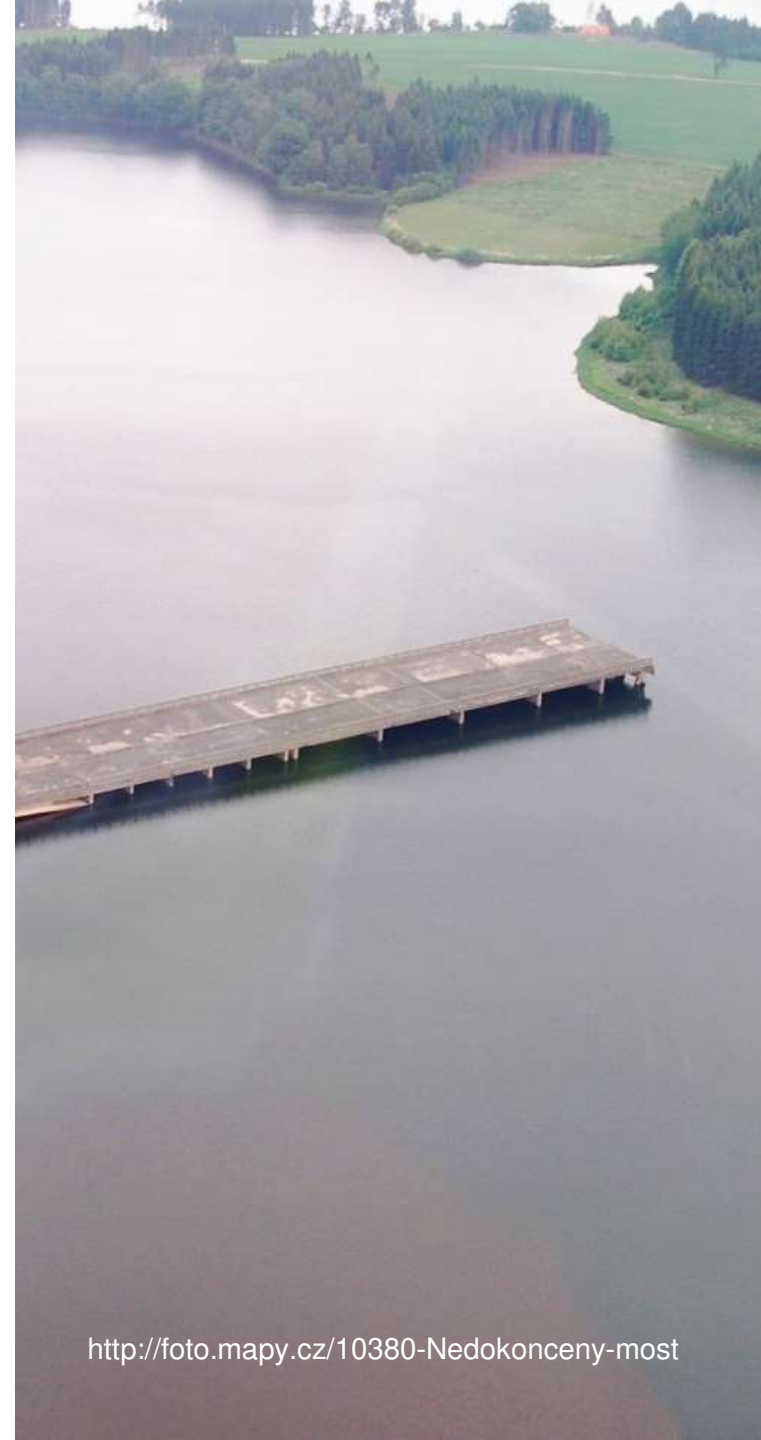
Fixed Time, Fixed Price, Variable Scope

- › 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

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