

Proč raději agilně?

Agenda

- 01 Neduhý vodopádového přístupu
- 02 Co znamená agilně?
- 03 Co to obnáší pro zadavatele?
- 04 Co je to Scrum?



01

Neduhy vodopádového přístupu

... aneb ne vše lze dopředu naplánovat

Vodopádový přístup

- Základní charakteristika:
 - Rozdělit projektové aktivity do fází, kde jedna striktně navazuje na druhou.
 - Po dokončení poslední fáze je produkt hotový.
- Původně vznikl v odvětvích průmyslové výroby a stavebnictví.



Požadavky

Sběr veškerých požadavků na funkce nového systému.

Analýza a návrh

Na základě požadavků se provede kompletní analýza a konkrétní návrh řešení.

Vývoj a testování

Výstupy z analýzy a návrhu se „převědou na kód“.

Nasazení

Vyvinutý aplikační kód se nasadí na veřejně dostupné úložiště, server, apod.

Údržba

Po dokončení vývoje probíhá již jen údržba.

Na papíře to zní dobře, jenže ...

- Vývoj software je do značné míry **kreativní činnost**. Rozdíl mezi stavbou domu a výrobou softwaru je zásadní.
- Odhadnout na začátku veškeré detaily návrhu aplikace **je v u větších systémů prakticky nemožné**.
- Měnit požadavky v průběhu vodopádového projektu je **velmi obtížné a v zásadě nežádoucí**.
- Řada problémů se typicky **objeví až na konci**, kdy probíhá nasazení a testování.

Negativní důsledky vodopádového přístupu

- **Dlouhý time to market.**
 - Významná konkurenční nevýhoda.
 - Řada předpokladů v době dokončení již nemusí platit.
 - Jenže s vodopádem produkt neumíme ven dostat dřív než je hotový celý.
- **Zbytečně vysoká cena.**
 - Nejistota ohledně skutečné časové náročnosti nutí dodavatele nadsazovat nabídkovou cenu, aby na projektu neprodělal.
 - Drahá je též výroba nesmyslně podrobné specifikace.
- **Zmařené úsilí na nepotřebných funkcích.**
 - Produkt se navrhuje na začátku a nelze tak využít průběžnou zpětnou vazbu.
 - Řada funkcí, které původně vypadaly nezbytné, nakonec potřeba není.
 - Naopak jsou mnohdy potřeba funkce úplně jiné.



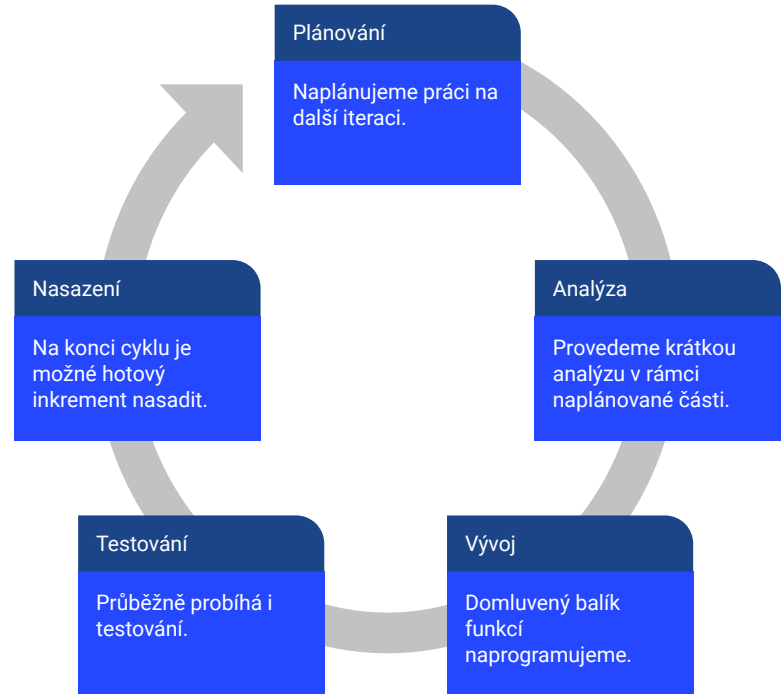
02

Co znamená agilně?

„Embrace the change“

Agilní = flexibilní, připravený na změny v průběhu

- Agilně řízený projekt **dopředu předpokládá změny požadavků v průběhu vývoje.**
- Místo snahy vše dopředu naplánovat **pracujeme inkrementálně**, v krátkých cyklech (~ iteracích).
- Rizika potlačujeme tím, že jim **čelíme ihned**. V každém cyklu **výsledek testujeme a také nasazujeme**.
- Krátké cykly umožňují **velmi rychlou zpětnou vazbu**.
- K popularizaci tohoto přístupu došlo v roce 2001 v tzv. Manifestu agilního softwarového vývoje.



Agile neznamená, že nevím co chci

- Poměrně rozšířený pohled na agile je, že se hodí jen tehdy, **když nevíme, jaké funkce má výsledný software plnit.**
- Ve skutečnosti je mylný spíše dojem, že **přesně víme co chceme.**
- Agile nijak nekomplikuje vývoj funkcí, které umíme dobře definovat.
- Umožňuje však snadno reagovat na situaci, kdy si uvědomíme, že ne všechno bylo stoprocentně domyšlené. Nikdy to totiž nejde zcela promyslet dopředu.

Agile má řadu výhod

- **Místo předpokladů vše rovnou ověřujeme s koncovými uživateli.**
 - Výsledek je uživatelům dostupný prakticky ihned.
 - Rychlá zpětná vazba zajistí, že výsledek dává smysl.
- **Možnost kdykoliv říct „stop, takhle mi to už stačí“.**
 - V každém okamžiku přesně víme, jaký kus celého produktu je již hotový.
 - Místo objednávky produktu kupujeme tým zkušených vývojářů.
 - Výsledná cena řešení skutečně odráží jeho náročnost.
- **Věnujeme se tomu podstatnému.**
 - Vývoj probíhá na základě prioritizace funkcí.
 - Nevytváříme věci do šuplíku.
 - Pokud dojdou peníze, nehrozí totální fiasko. To podstatné jsme řešili na začátku.

Není to žádná experimentální záležitost

- Agilně pracujeme již řadu let, je to **ověřeno dlouhodobou praxí**.
- Adopce agilních přístupů **je v kurzu** i v řadě těžkotonážních institucí, např. ČS, KB.
- Agilní přístup pro veřejnou správu **kombinujeme s metodikou Best Value Approach**, abychom se perfektně **vešli do české legislativy**.

Agile klade větší nároky na zadavatele

- V kontrastu s vodopádovým přístupem je zapojení zadavatele **nutné po celou dobu vývoje**.
- Zadavatel musí být ve **velmi těsném kontaktu s dodavatelem** a směřování projektu průběžně usměrňovat ~ **nastavovat priority**.
- Výrazná **časová alokace** odpovědné osoby na straně zadavatele **je nezbytná**.

03

Na co se připravit?

Co všechno obnáší řízení
agilního projektu z pohledu
zadavatele

Product owner

- Klíčová je existence tzv. **product ownera**.
 - Agreguje požadavky směrem od úřadu.
 - Plánuje práci a stanovuje priority jednotlivých úkolů.
 - Ve spolupráci s týmem vývojářů průběžně připravuje a upřesňuje finální zadání.
 - Nemusí mít (a ideálně by ani mít neměl) vhléd do technického řešení!
- Product owner je **práce na full-time**. Je zcela nezbytné, aby byl vývojářskému týmu vždy k dispozici pokud se objeví nějaké nejasnosti.
- Pro jeden agilní projekt by měl být **pouze jeden product owner**.
 - Nemusí vědět všechno, musí ale fungovat jako styčný důstojník.
 - Přítomnost více product ownerů vede k chaosu - hlavně s ohledem na priority.



Kompetence učinit rozhodnutí

- Aby vše dobře fungovalo, product owner musí mít **dostatečné kompetence**.
- Mnohdy je totiž nutné o podobě řešení **rozhodnout rychle**. Pokud pro dané rozhodnutí product owner musí vždy získat deset razítek, rychlost vývoje se tím dramaticky sníží.
- To neznamená, že si může dělat co chce.
 - **Mantinely pro rozhodování mu připravuje širší pracovní skupina (nadřízení, kolegové, ...)** tím, že mu vysvětlí jejich očekávání.
 - Očekávání se stanovují skrze formulaci **strategických záměrů**. Např. *chci, aby vyplnění elektronického formuláře zvládlo i pětileté dítě*. Jednotlivým záměrům stanoví patřičnou prioritu.
 - Konkrétní podoba realizace požadavků je již v kompetenci product ownera.

Co tedy bude nutné?

- Vybrat osobu, která roli product ownera zastane.
- Svěřit mu/jí dostatečné pravomoce.
- Alokovat dostatečné množství času na projekt (full time).

- Vhodné vlastnosti product ownera:
 - Komunikativnost
 - Znalost problematiky, kterou vznikající software má řešit (ne technicky, ale funkčně)
 - Zodpovědnost a kompetentnost
 - Chuť naučit se něco nového

04

Scrum

Nejrozšířenější agilní framework

Základní přehled a principy

- Scrum je sada receptů, která popisují, jak řídit agilní projekt.
- Definuje **tři hlavní role**, které se projektu účastní.
 - Scrum Master (externista, z řad ASWA, případně od dodavatele řešení)
 - Product Owner (z řad Celní správy)
 - Vývojářský tým
- Kombinuje **čtyři formální schůzky**, které probíhají během každého „Sprintu“.
 - Daily Scrum
 - Sprint Planning
 - Sprint Review
 - Sprint Retrospective
- Sprint trvá zpravidla **2 týdny** a na jeho konci je vždy **funkční produkt**, který potenciálně lze veřejně nasadit.



Role ve Scrumu

■ Scrum Master

- Pomáhá ostatním se ve Scrumu zorientovat a správně jej praktikovat.
- Nejedná se o projektového manažera, neúkoluje!

■ Product Owner

- Již jsme představili - zodpovídá za zadání a správný směr projektu.
- Udržuje tzv. Product Backlog - evidenci požadavků na vznikající software. Průběžně jej rozpracovává a prioritizuje.

■ Vývojářský tým

- Zodpovídá za vlastní vývoj.
- Zadání získává od Product Ownera dle priorit, ale vlastní práci si již organizuje uvnitř.



Scrumové schůzky

■ Daily Scrum

- Krátká každodenní schůzka, slouží k synchronizaci práce.

■ Sprint Planning

- Probíhá jednou až dvakrát za Sprint.
- Slouží k finální přípravě zadání a naplánování práce na následující Sprint.

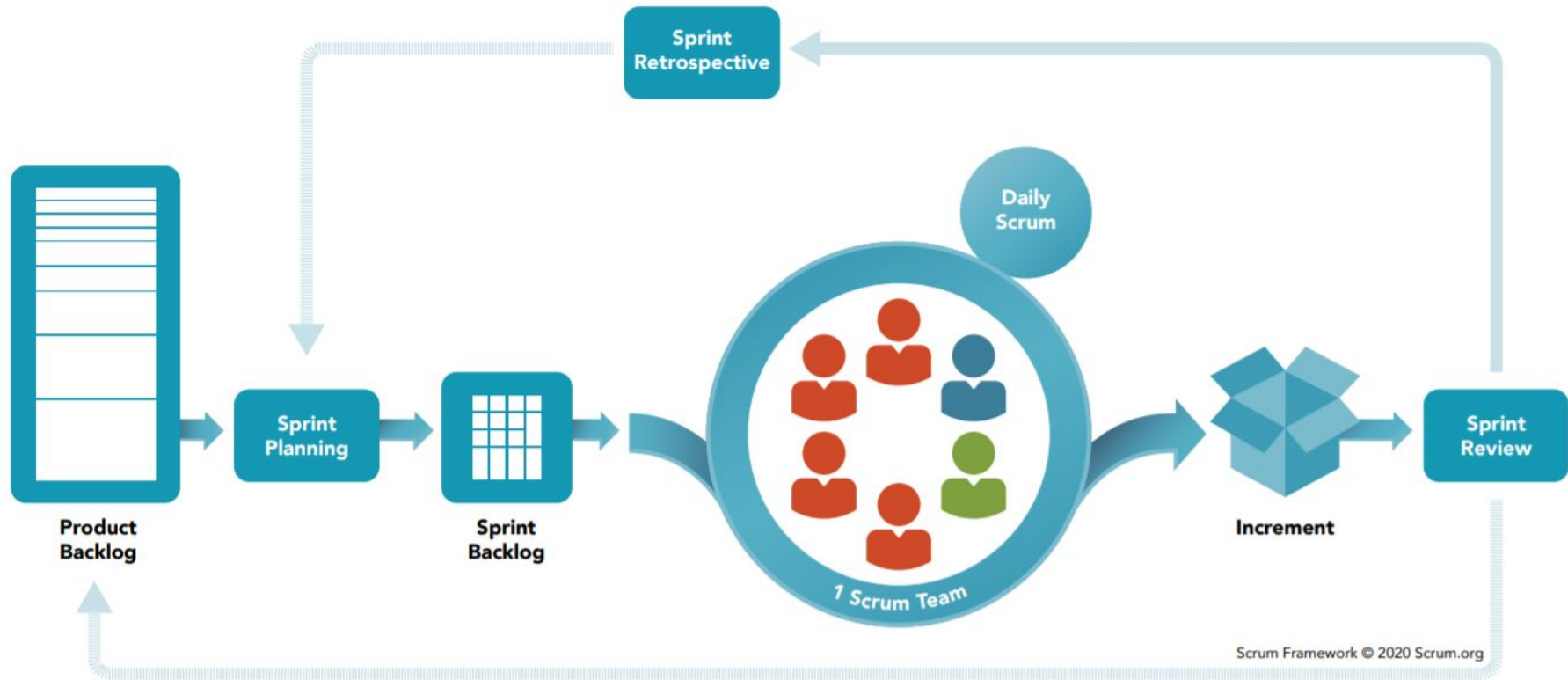
■ Sprint Review

- Slouží ke zhodnocení výsledku aktuálního Sprintu a koordinaci na vyšší úrovni.
- Je vhodné, aby byli přítomní i stakeholderi!

■ Sprint Retrospective

- Cílem je zvýšit efektivitu práce v následujícím Sprintu.
- Rozebereme co funguje a co nikoliv (např. komunikace).
- Navrhujeme opatření, jak to do budoucna dělat lépe.





Zdroj: scrum.org

A

Děkuji za pozornost

ASWA | ASOCIACE
SOFTWAREVÝCH
AGENTUR

Filip Vařecha

Fragaria s.r.o., člen ASWA

+420 777 883 133

filip.varecha@fragaria.cz