

Agilní přístupy k vývoji SW

Jaroslav Žáček

jaroslav.zacek@osu.cz

<http://www1.osu.cz/~zacek/>

<http://www.agilemanifesto.org/>

Manifest Agilního vývoje software

Objevujeme lepší způsoby vývoje software tím,
že jej tvoříme a pomáháme při jeho tvorbě ostatním.

Při této práci jsme dospěli k těmto hodnotám:

Jednotlivci a interakce před procesy a nástroji

Fungující software před vyčerpávající dokumentací

Spolupráce se zákazníkem před vyjednáváním o smlouvě

Reagování na změny před dodržováním plánu

Jakkoliv jsou body napravo hodnotné,
bodů nalevo si ceníme více.

Principy 1/4

- Naší nejvyšší prioritou je vyhovět zákazníkovi včasným a průběžným dodáváním hodnotného softwaru.
- Vítáme změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.
- Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.

Principy 2/4

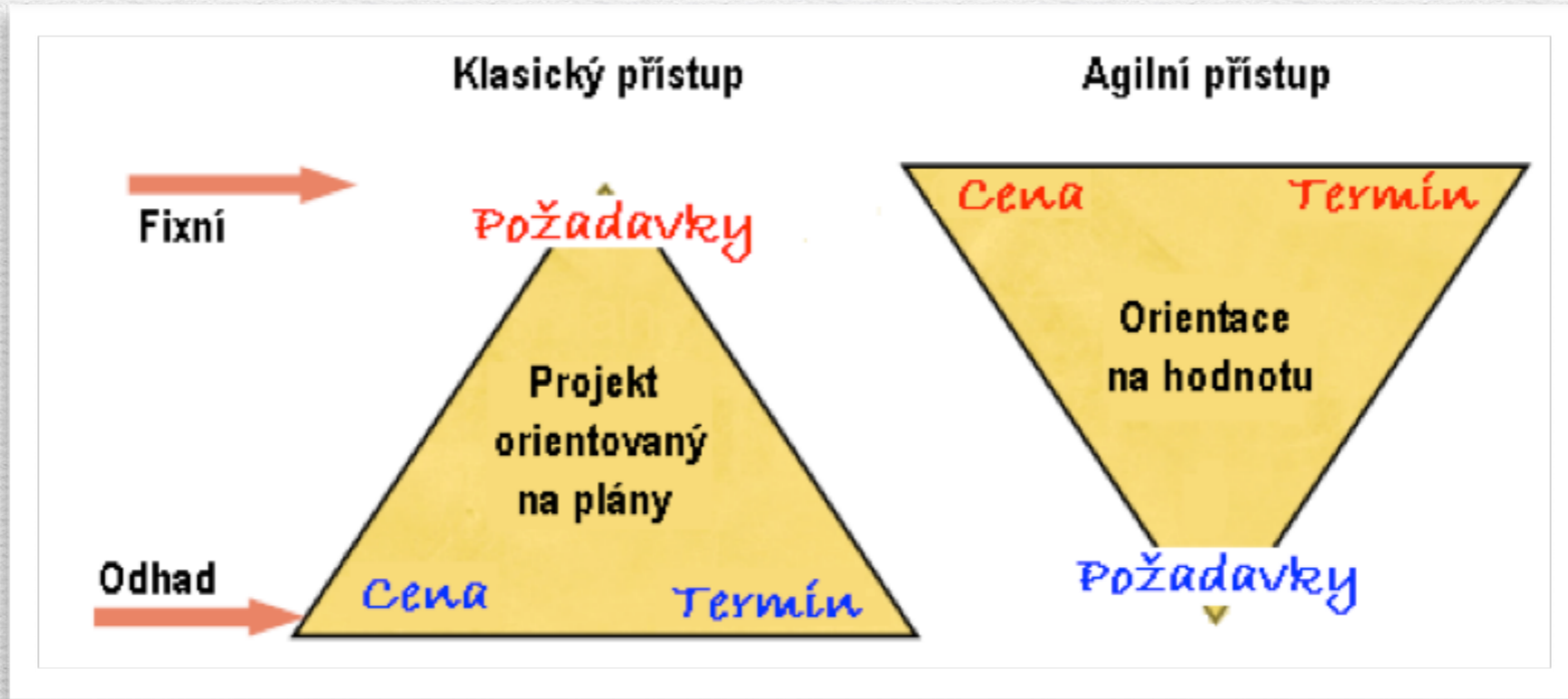
- Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.
- Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.
- Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.

Principy 3/4

- Hlavním měřítkem pokroku je fungující software.
- Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.
- Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.

Principy 4/4

- Jednoduchost-umění maximalizovat množství nevykonané práce je klíčová.
- Nejlepší architektury, požadavky a návrhy vzejdou ze samo-organizujících se týmů.
- Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším, a následně koriguje a přizpůsobuje své chování a zvyklosti.



Scrum

- Označení mlýnice v rugby
- Představen Schwabberem a Beedlem v roce 1995
- Framework pro projektové řízení
- Neobsahuje žádné inženýrské praktiky



Jak na Scrum



- Agilní projektové řízení
- Založený na součinnosti lidí
- Iterativní a inkrementální
- Vždy doručuje hodnotu zákazníkovi

Základní pojmy

- Sprint - cílem je doručení fungující kód
- Product backlog - seznam požadavků
- User stories - scénář, který reprezentuje požadavek
- Daily meetings (daily scrum)
- **Self-managed týmy**

User Stories



Role ve Scrum

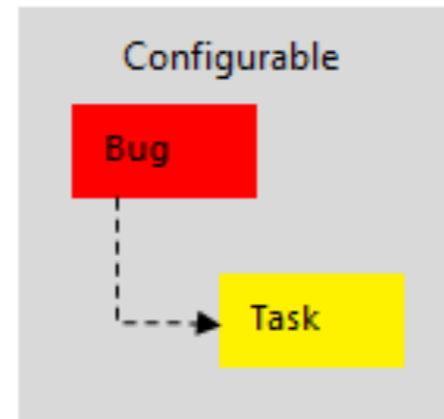
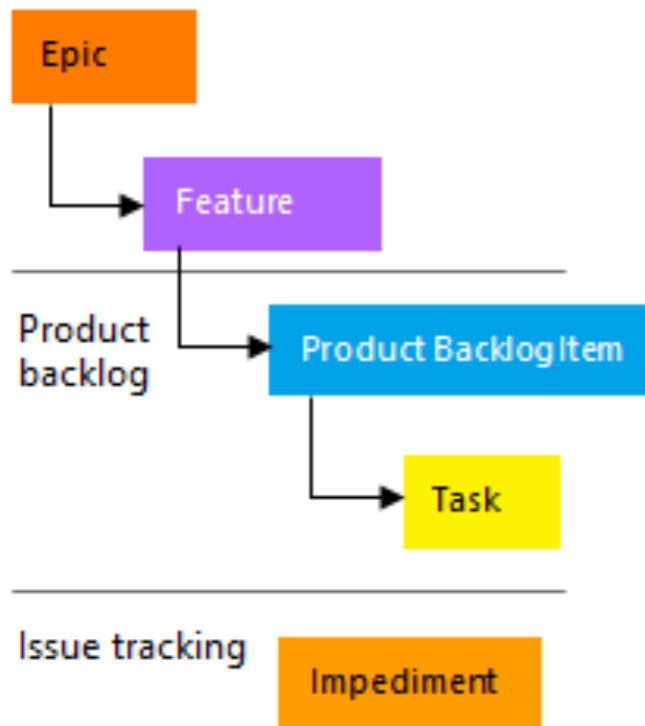
- **Product Owner (PO)** - člověk, který reprezentuje zájmy na projektu, vytváří backlog, definuje cíle, vytváří roadmap
- **Scrum Master (SM)** - odpovědný za proces, jeho implementaci, odstraňuje problémy v týmu
- **Člen týmu (T)** - transformuje položky backlogu na funkcionality systému

Artefakty Scrum

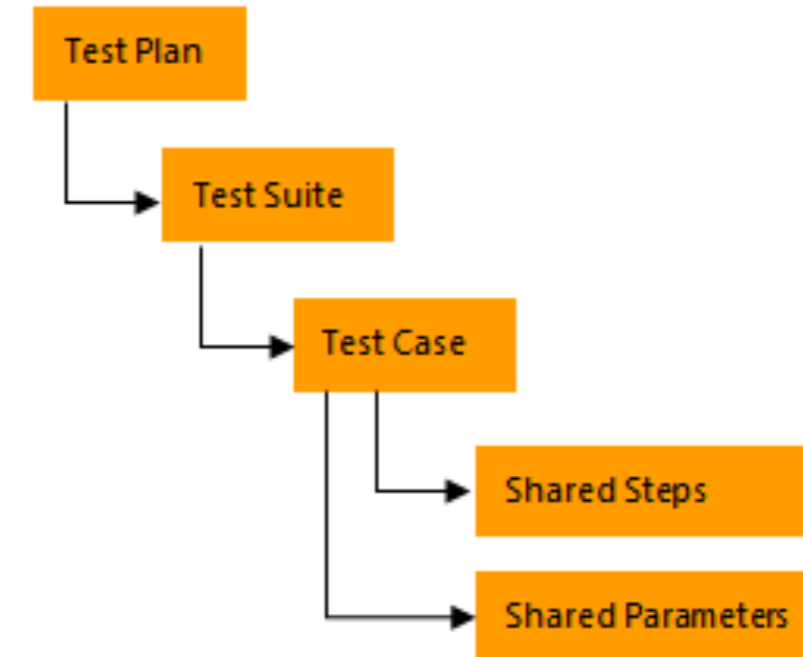
- **Product backlog** - seznam požadavků seřazený podle priorit, obsahuje odhad komplexnosti,
- **Sprint backlog** - práce a úkoly pro aktuální sprint, tým definuje úkoly, které je potřeba vykonat k doručení vybraných scénářů
- **Potenciálně doručitelný produkt** - výstup sprintu, který je možno zákazníkovi do provozu

Artefakty Scrum

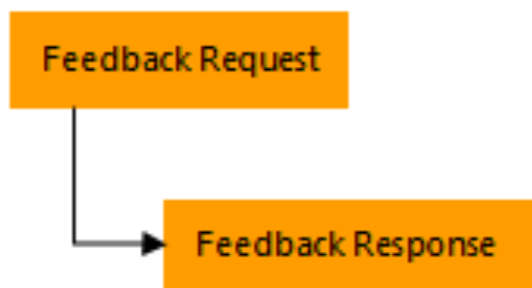
Portfolio backlog



Test



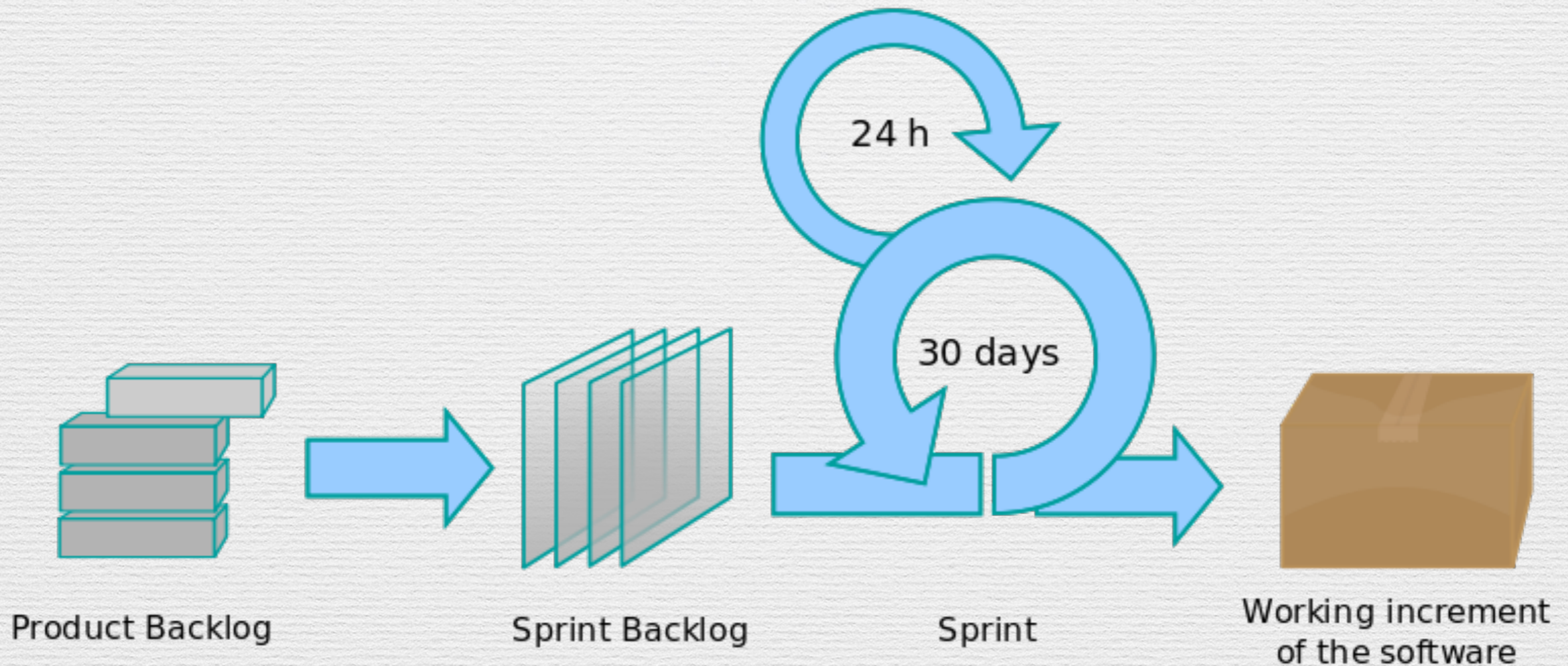
Feedback



Code review

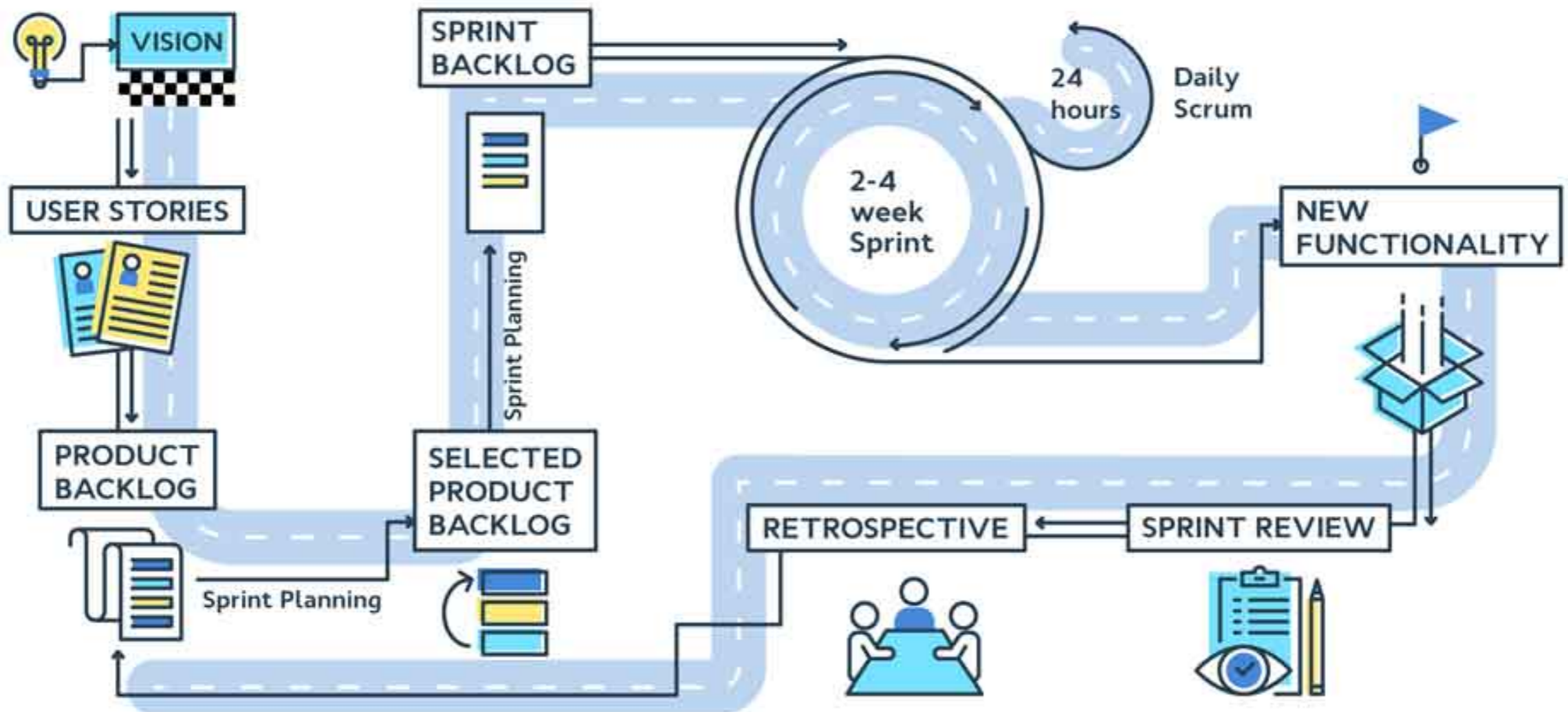


Proces řízení projektu

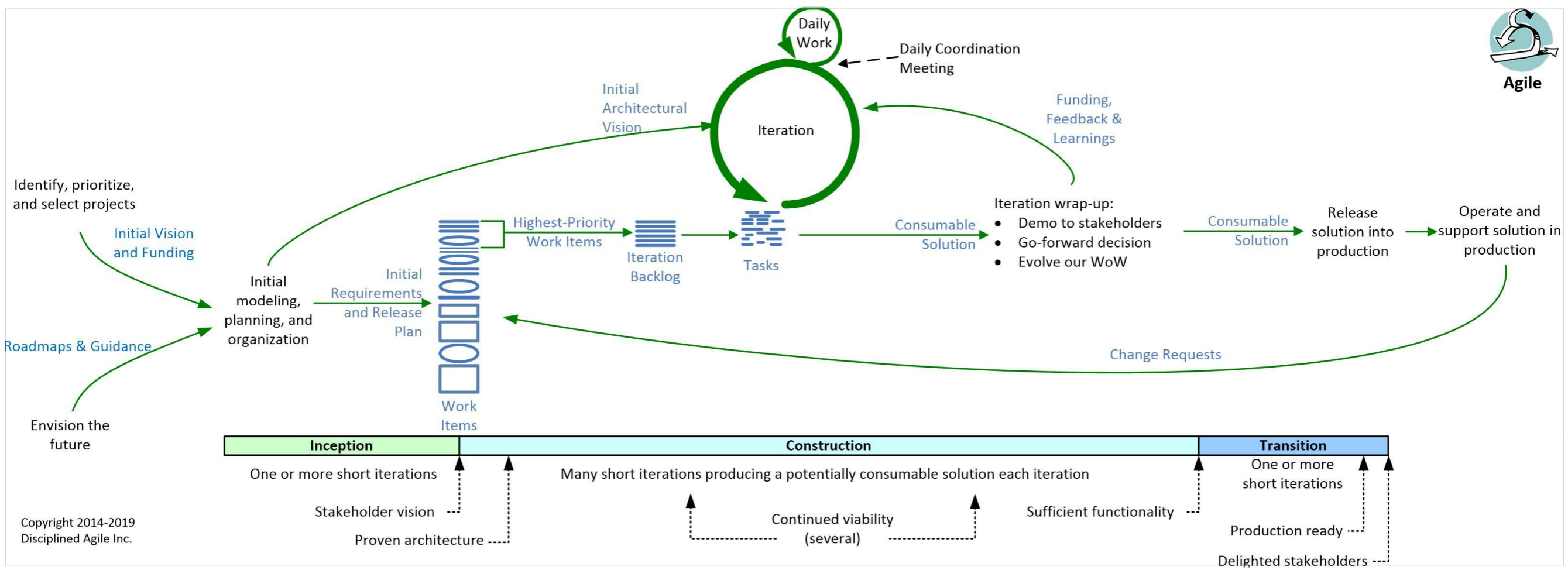


Proces řízení projektu

SCRUM PROCESS



DAD se Scrumem



Sprint

Sprint 1



Sprint 2



...

Sprint n



Scrum ceremonie

- Daily scrum
- Sprint planning
- Sprint review
- Sprint retrospective

Daily Scrum

- Co jste udělali od té doby, co jsme se naposledy viděli?
- Co uděláte do příště?
- Je něco, co vás zpomaluje, nebo vám stojí v cestě?



Kanban-style

Dělá věci jednodušší:

- Jaké překážky mi brání v tom, abych doručoval hodnotu?
- Jaký je pokrok?
- (potřebujete Kanban board)



Plánování sprintu

Připraveno

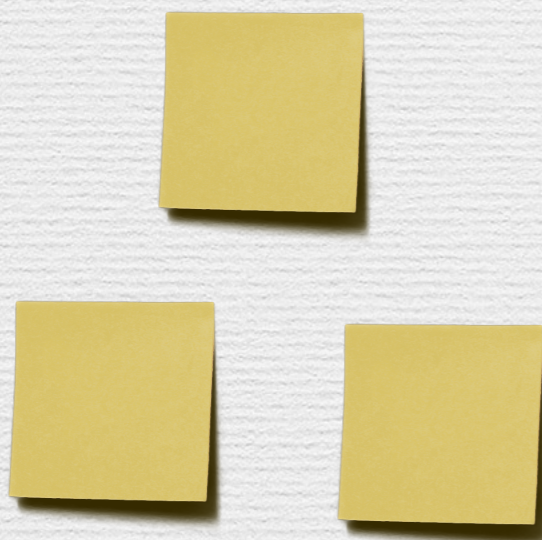


Probíhá

Dokončeno

Plánování sprintu

Připraveno



Probíhá



Dokončeno

Plánování sprintu

Připraveno

Probíhá

Dokončeno



Plánování sprintu

Připraveno

Probíhá

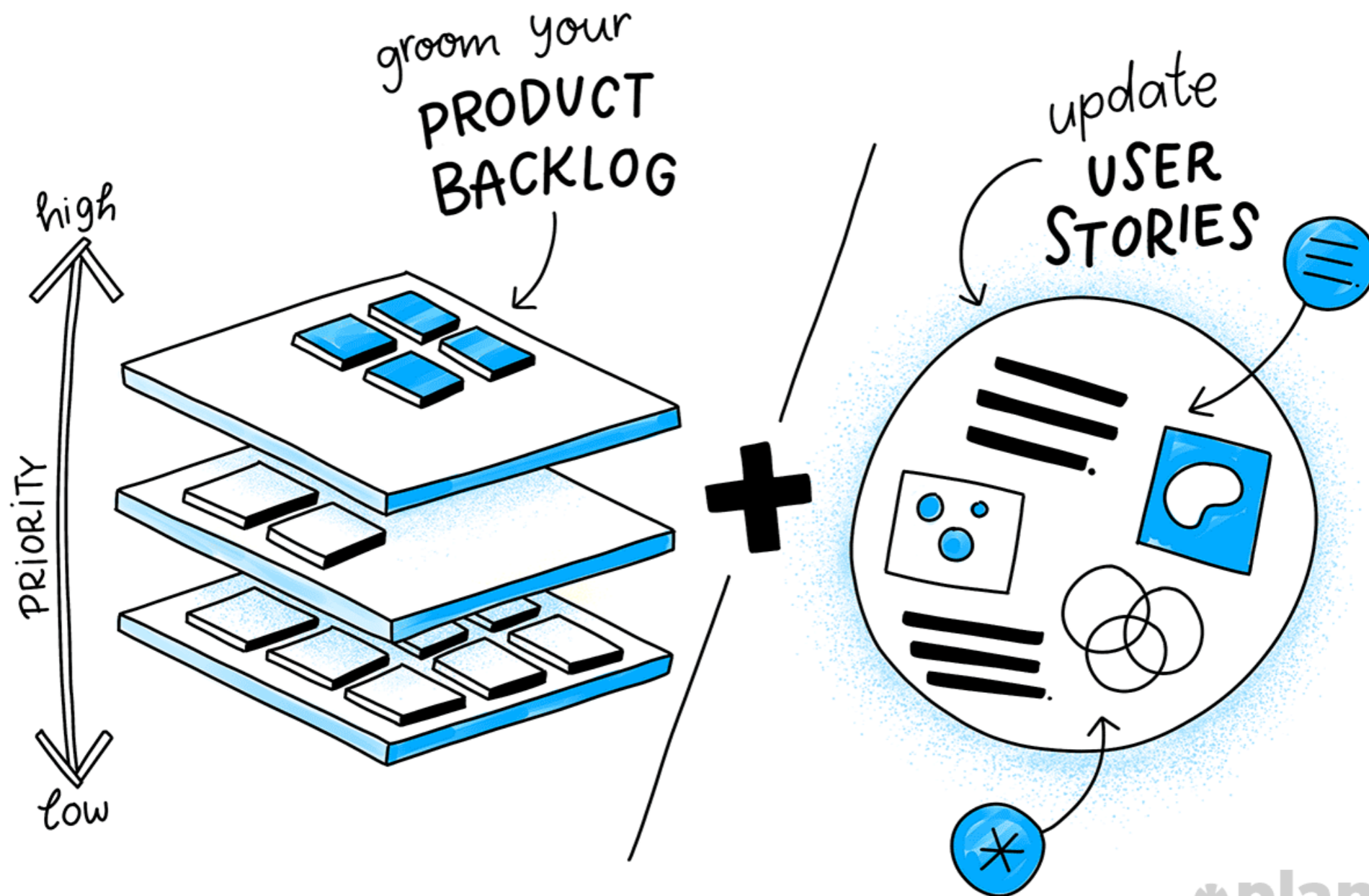
Dokončeno



Sprint planning

- Zkontrolovat roadmapu projektu
- Odhadněte produktový backlog - grooming
- Navrhněte cíl/smysl sprintu (sprint goal)
- Použijte data a zkušenosti pro zlepšení plánování sprintu
- Projděte všechny story a derivujte z nich jednotlivé tasky (úkol pro vývojáře/minimální jednotka práce)

Grooming session



Grooming session

- Nemělo by být delší, než 30 minut.
- U user stories hlídáme:
 - Priority - jsou storky správně seřazené
 - Je storka připravená na implementaci? Rozumí všichni zadání?
 - Je popis aktuální vzhledem k roadmapě? Můžeme stanovit nějaký odhad?

Grooming session



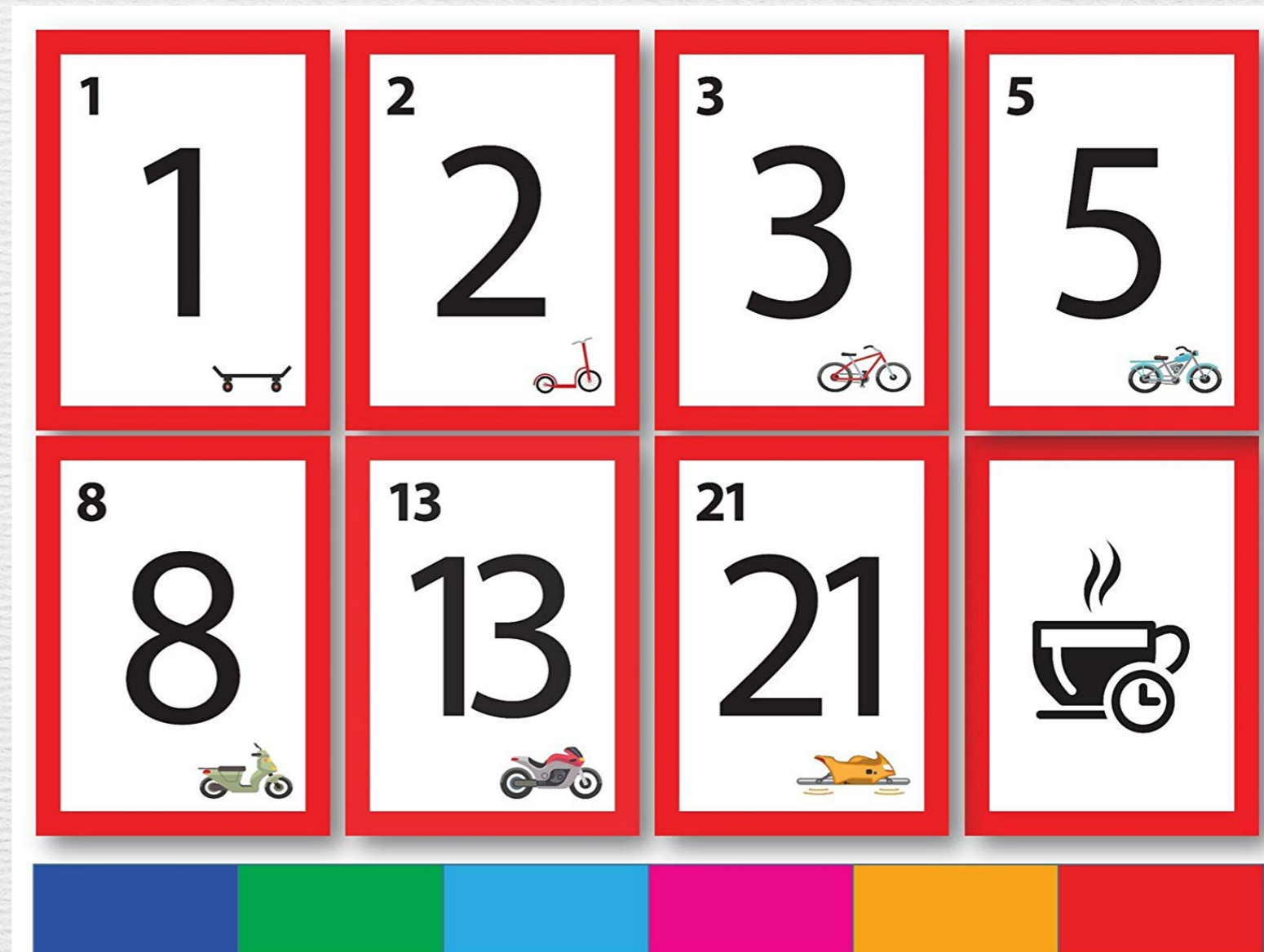
Can Do Quickly
No Testing Needed



More Time Needed
Only Item in a Sprint

Grooming session

- Fibonacci 0, 1, 2, 3, 5, 8, 13, 21, 34, 55
- Agile cards
- Hodiny

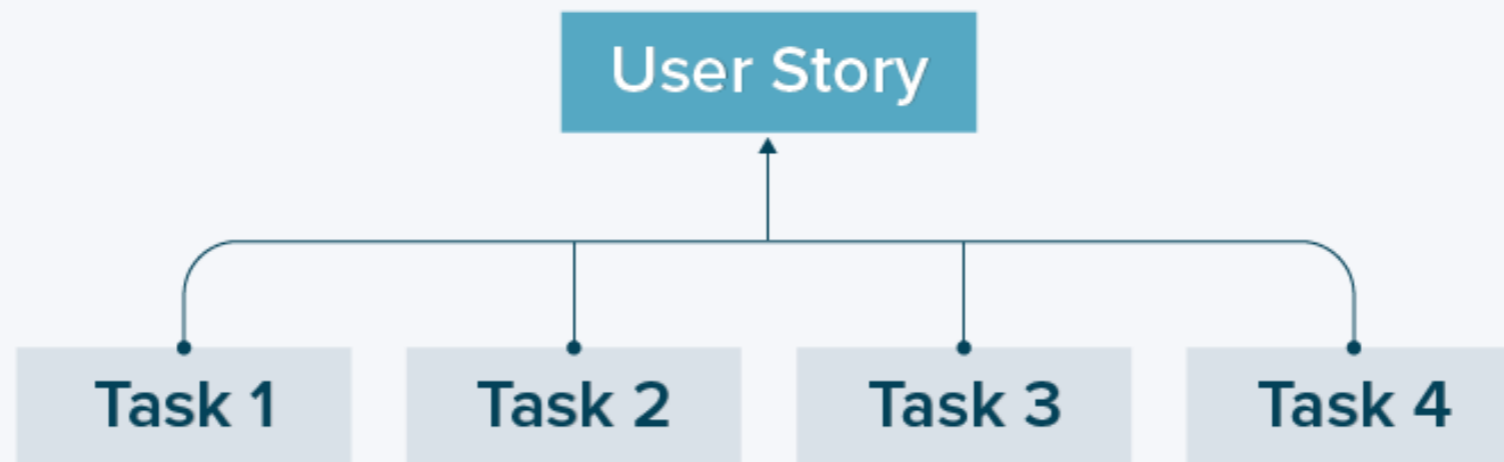


Kdo by měl být přítomen na plánování sprintu

- Product owner
- Scrum master
- Všichni členi týmu

Co se má při plánování dělat?

- Rozpad user stories na tasky:



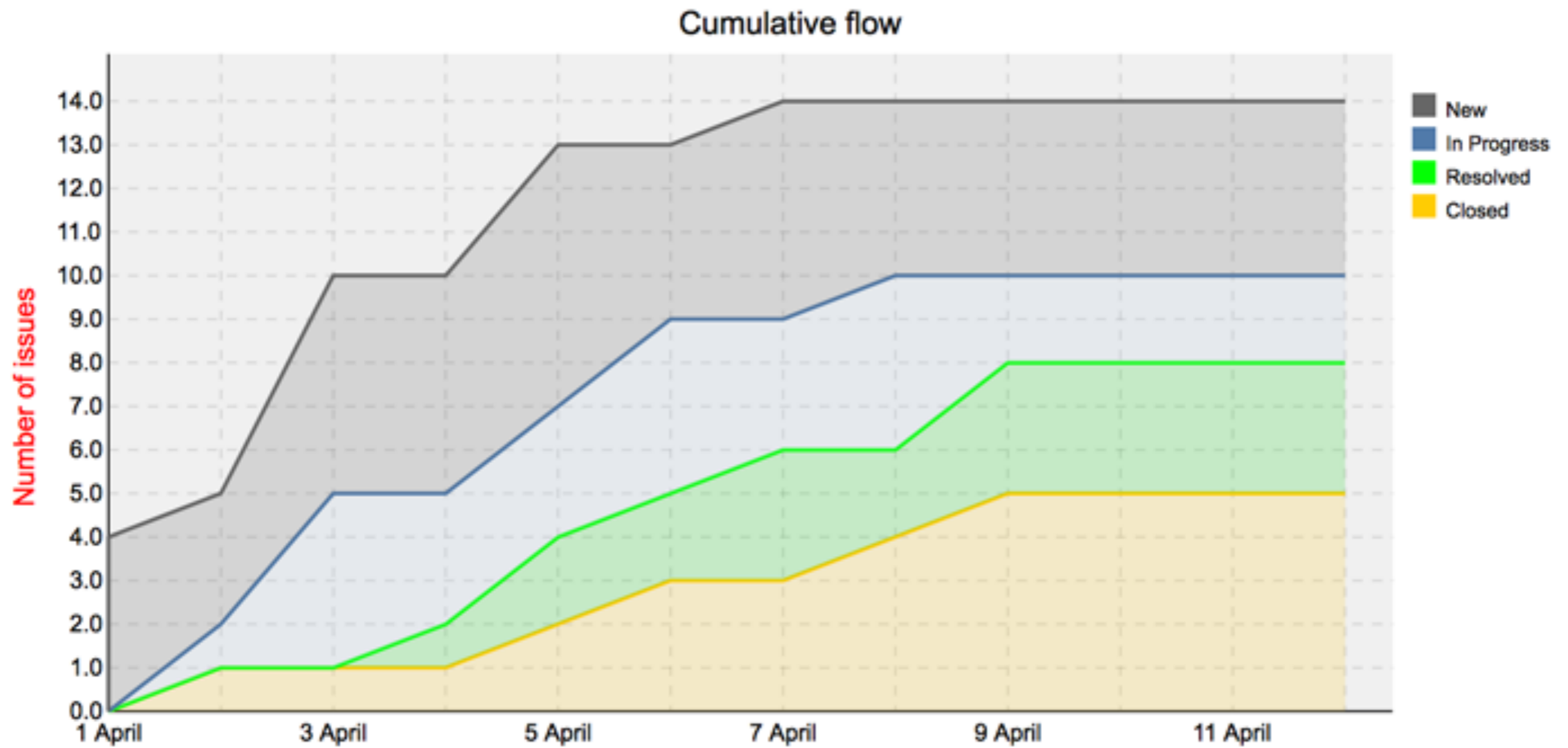
Co se má při plánování dělat?

- Kontrola definitions of done (DoD)
- Stanovení akceptačních kritérií
- Tým odsouhlasí jejich kapacitu pro nadcházející sprint (dovolená, státní svátky,...)
- Je zodpovědností členů týmu, co si dají do nadcházejícího sprintu -> vytváří **commitment**.

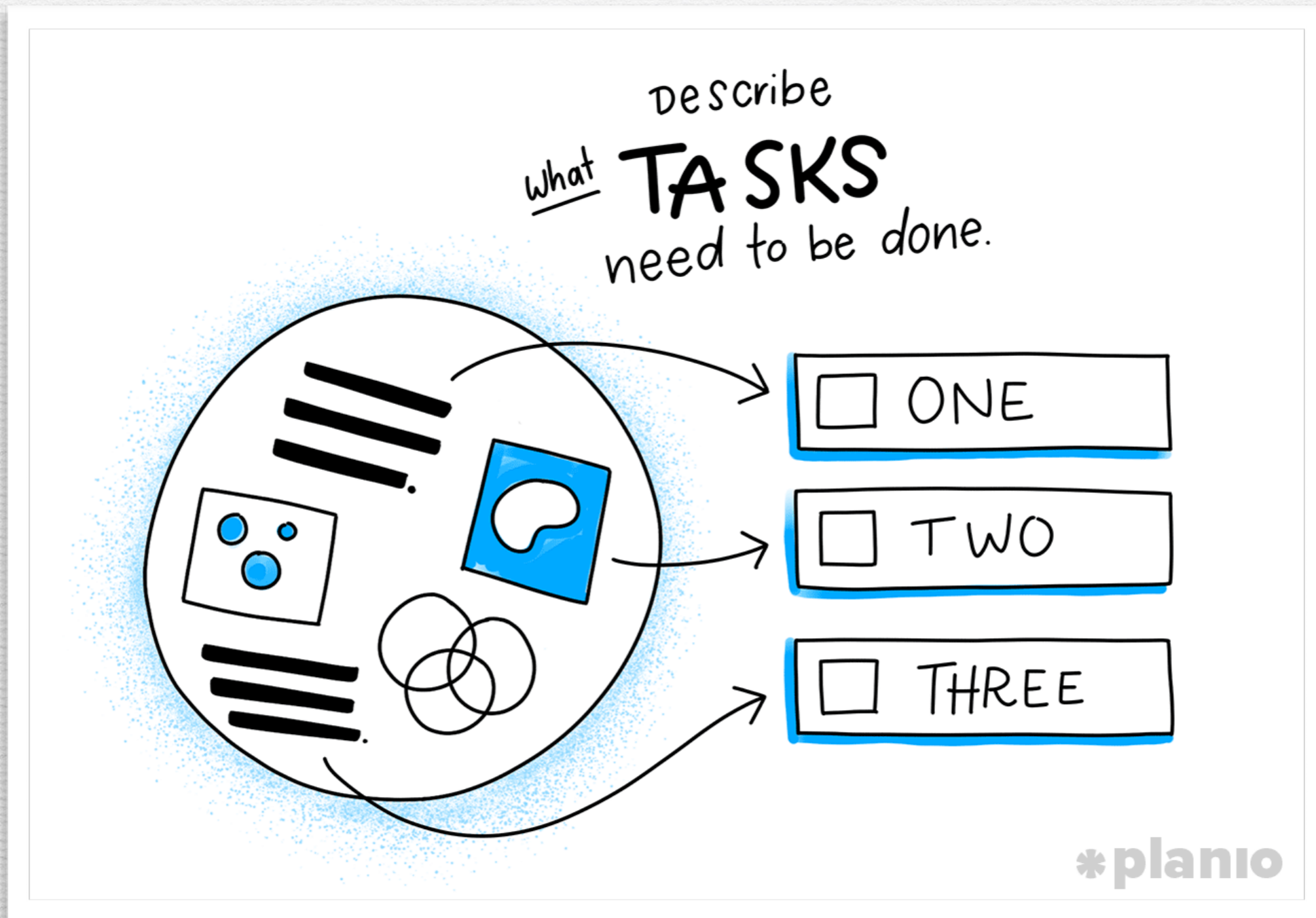
Kolik by toho mělo být ve sprintu?

- V průběhu vývoje může dojít k neplánovaným událostem...
 - špatný odhad, neočekávaná chyba, limitace architektury, opravy předešlých chyb.
- Většina týmů vyčlení 20-40% času pro tyto případy, většinou hodnotu hlídá scrum master a upravuje směrem k realitě na základě předchozích zkušeností.

Hlídání pokroku



Příprava úkolů z user stories



Příprava úkolů z user stories

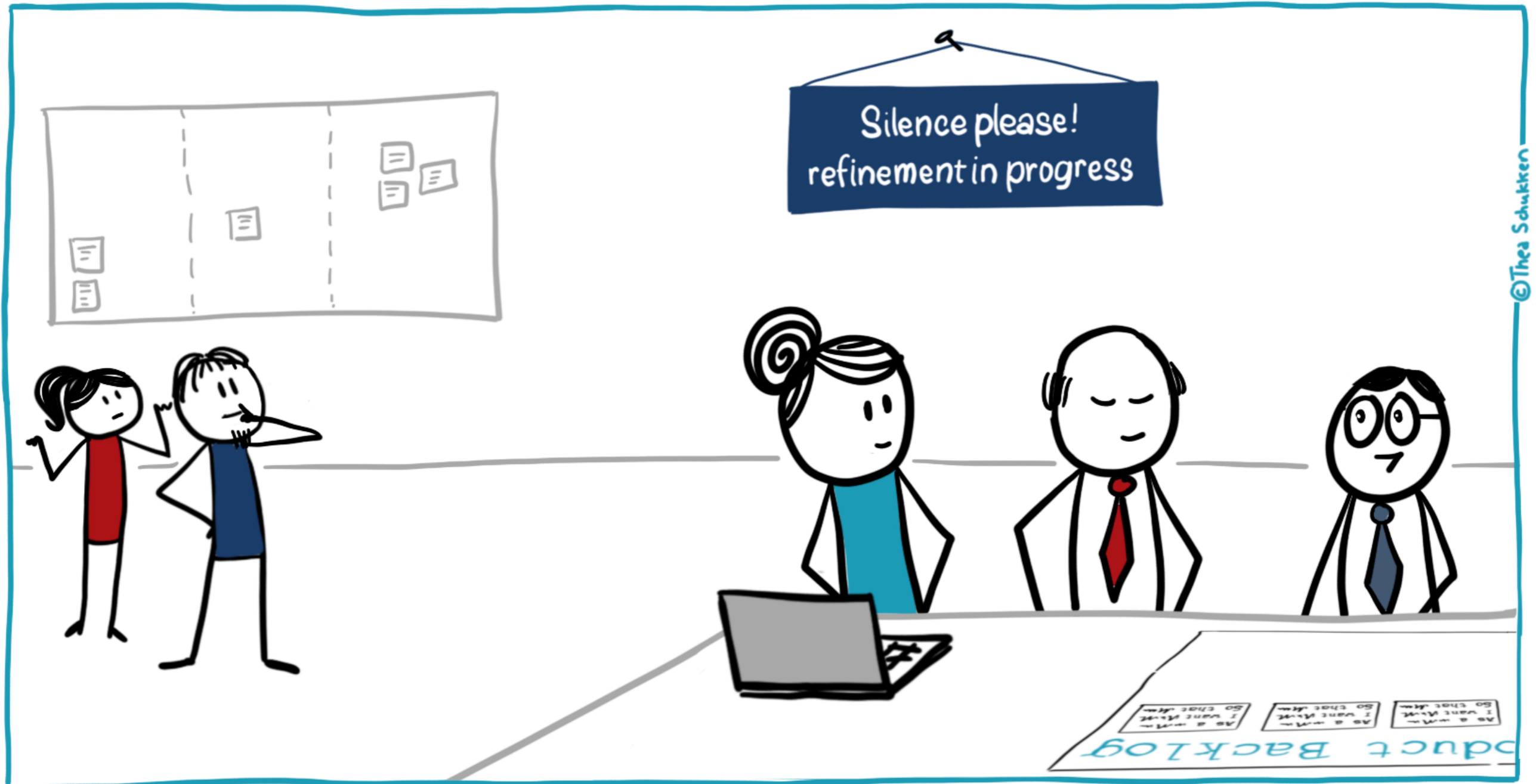
- Změnilo se něco od vytvoření user story?
- Je odhad pořád podložený reálným očekáváním?
- Jsou nějaké závislosti mezi user stories?
- Jak budeme testovat? Dá se to automatizovat?
- Umíme tuto story udělat/dokončit?
- Jaký bude mít dopad doručení této story na celý produkt?

Možné problémy Scrumu

- Scrum nezmiňuje práci s riziky
- Není zaměřen na architekturu
- Používá User Stories místo Use-Case driven přístupu
- Porušování pravidel Scrumu
- Scrum of scrums (více Scrum týmů)
- Zpravidla se kombinuje s XP, RUP, OpenUP

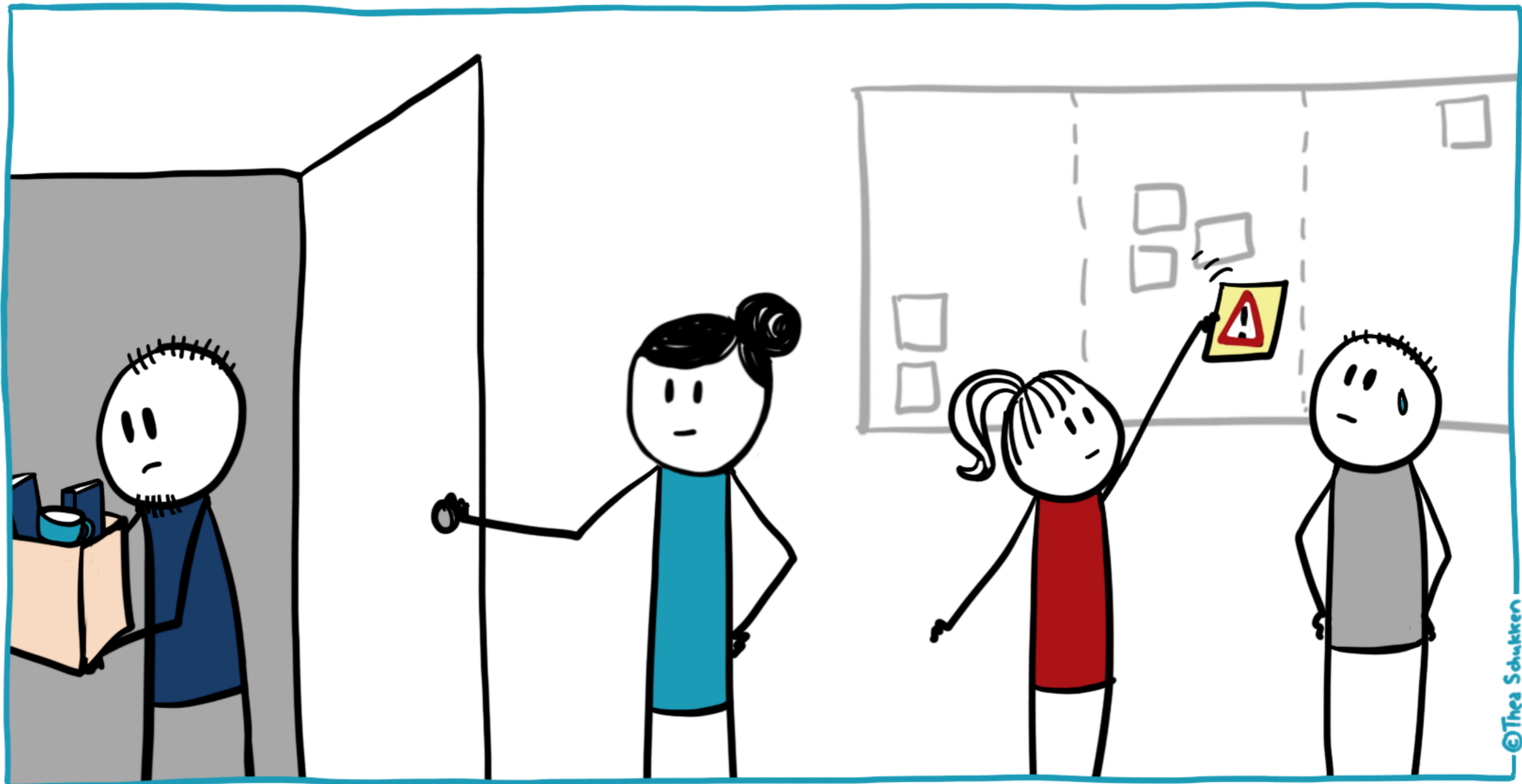
Mýty o Scrumu

Refinement is a required meeting for the entire Scrum Team



Mýty o Scrumu

The Scrum Master can't remove people from the Scrum Team



Mýty o Scrumu

The Product Owner is a proxy for stakeholders



Mýty o Scrumu

Story Points are required in Scrum



Spotify model

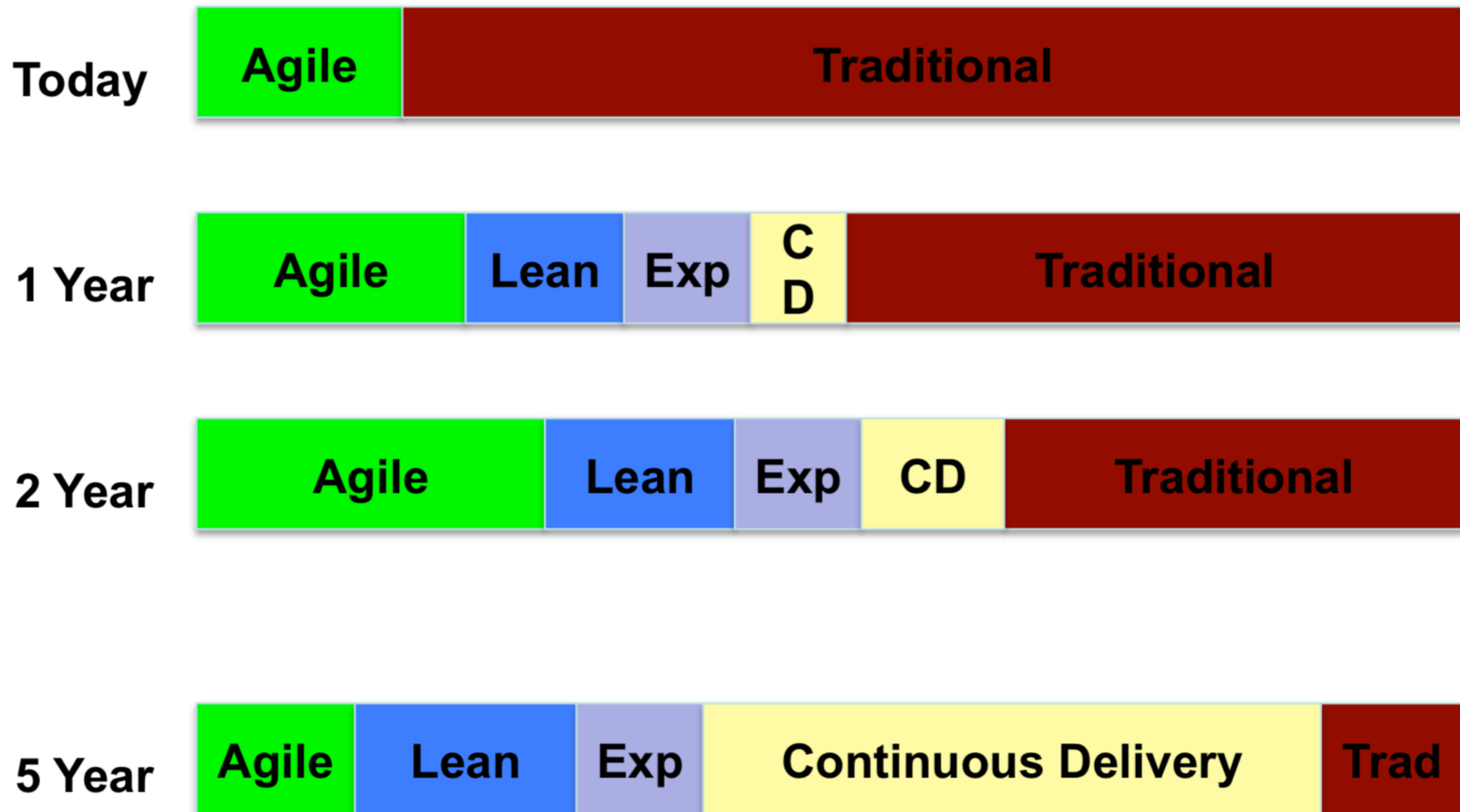


<https://youtu.be/4GK1NDTWbkY>

Hodnoty

- *Agile > Scrum*
- *Principles > Practices*
- *Servant > Master*

Postupné zavádění do společnosti



Extrémní programování

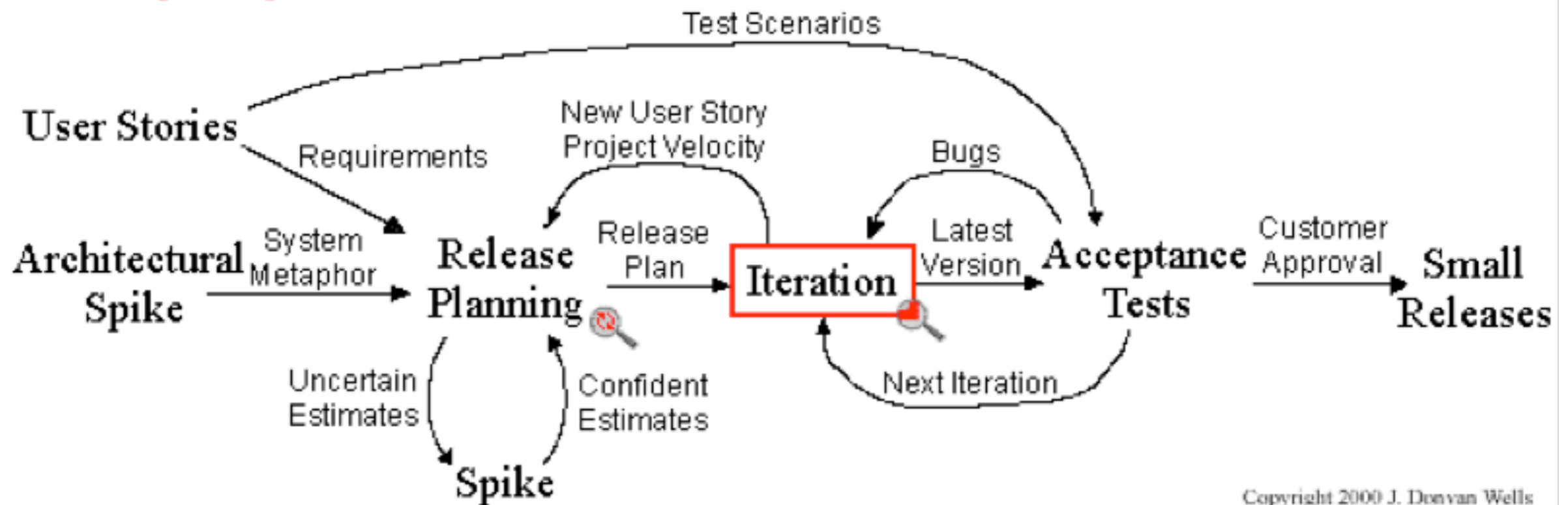
- Vytvořil Kent Beck
- Zaměřen hlavně na techniky
- Definuje psaní kódu jako klíčovou činnost
- Slabé stránky původních programátorských metod se snaží zakrýt silnými



Životní cyklus projektu



Extreme Programming Project



Základní hodnoty XP

- Komunikace - zákazníkovi se nepoloží správné otázky, programátor neřekne kolegovi o kritické změně návrhu...
- Jednoduchost - místo složitých věcí raději implementovat to jednodušší za cenu pozdější pracnosti
- Zpětná vazba -
 - Pro programátory - testy
 - Pro zákazníka - jsou specifikace jednoznačné?
 - Pro PM - je zákazník spokojený s kvalitou produktu, ovládá se dle jeho představ?
- Odvaha - Pokud později odhalíte chybu architektury, musíte ji opravit i za cenu zvýšených nákladů

Postupy vývoje dle XP

- Plánovací hra - určíme rozsah projektu (scope) kombinací priorit businessu a technických odhadů, plány později aktualizujeme
- Malé verze - co nejdříve uvedeme systém do provozu
- Metafora - vývoj vedeme na základě toho, jak bude zákazník se systémem pracovat (obdoba User Stories, User Scenario)

Postupy vývoje dle XP

- Testování - programátoři píší unit testy, zákazníci funkční testy
- Refaktoring - úprava kódu bez změny funkčnosti
- Párové programování - veškerý kód píší společně dva programátoři na jednom počítači

Postupy vývoje dle XP

- Společné vlastnictví - všichni mohou měnit kód kdykoliv a kdekoliv
- Nepřetržitá integrace - sestavujeme buildy několikrát denně, zpravidla po dokončení scénáře
- Pracovní týden má 40 hodin

Postupy vývoje dle XP

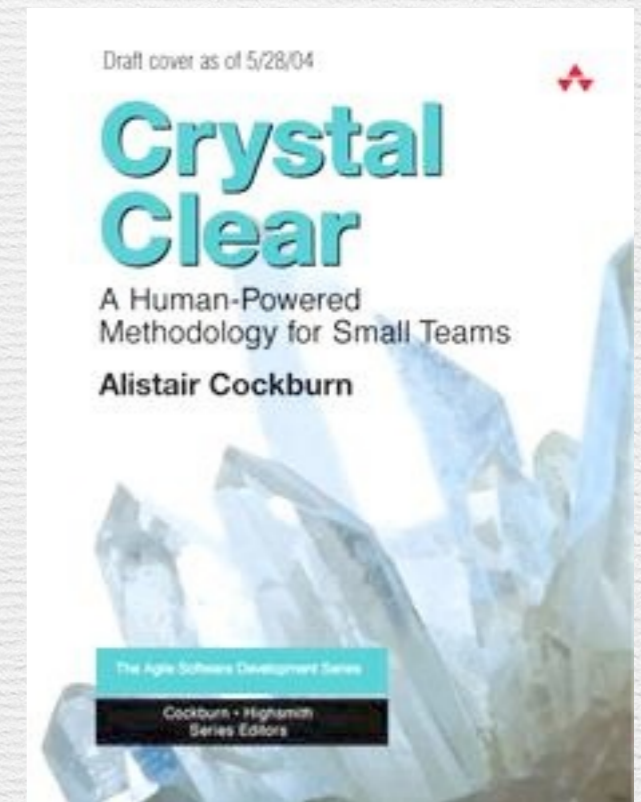
- Zákazník na pracovišti - do týmu patří skutečný uživatel systému, je k dispozici na plný úvazek
- Standardy pro psaní zdrojového kódu - programátoři píší zdrojový kód v souladu s pravidly

Problematická místa XP

- Onsite zákazník
- User Stories místo UC
- Minimální design
- V SW firmách nemůžou existovat jen XP týmy
- Párové programování po celou dobu vývoje
- Zahození části kódu

Crystal Clear

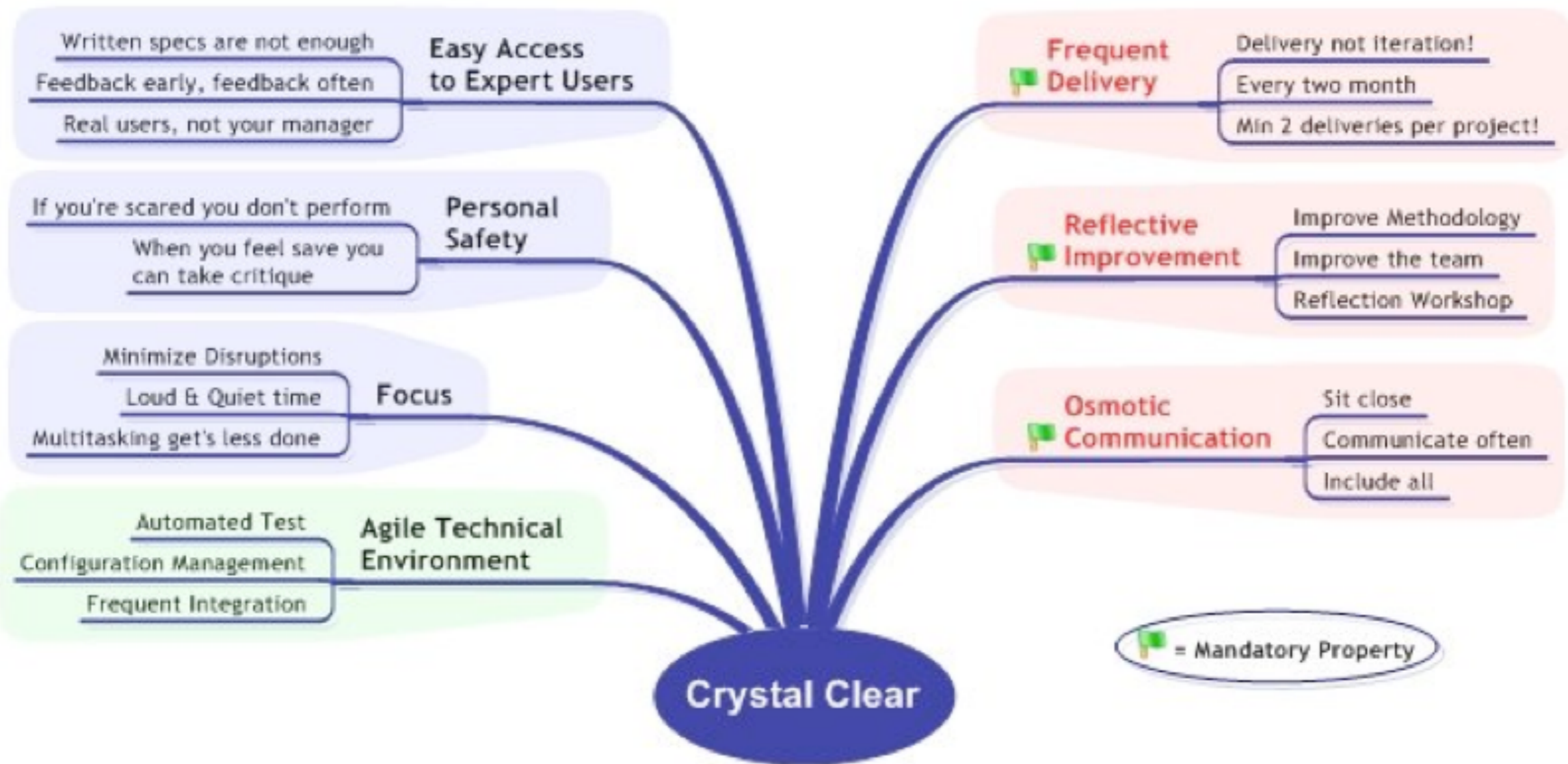
- Hlavním autorem A. Cockburn
- Nejedná se o metodiku, ale o návod, jak si podle projektu metodiku vytvořit.
- Opět se zaměřuje na lidi, ne na procesy či technologie



Pro vlastní metodiku potřebujete...

- Architekta a 2-7 vývojářů,
- jednu větší místnost,
- tabuli a flipchart,
- snadný přístup k uživatelům (expertům),
- kteří nejsou vyrušováni a doručují spustitelný kód,
- každý měsíc, maximálně čtvrtletně,
- a neustále se přizpůsobují měnícím se podmínkám.

7 vlastností Crystal clear



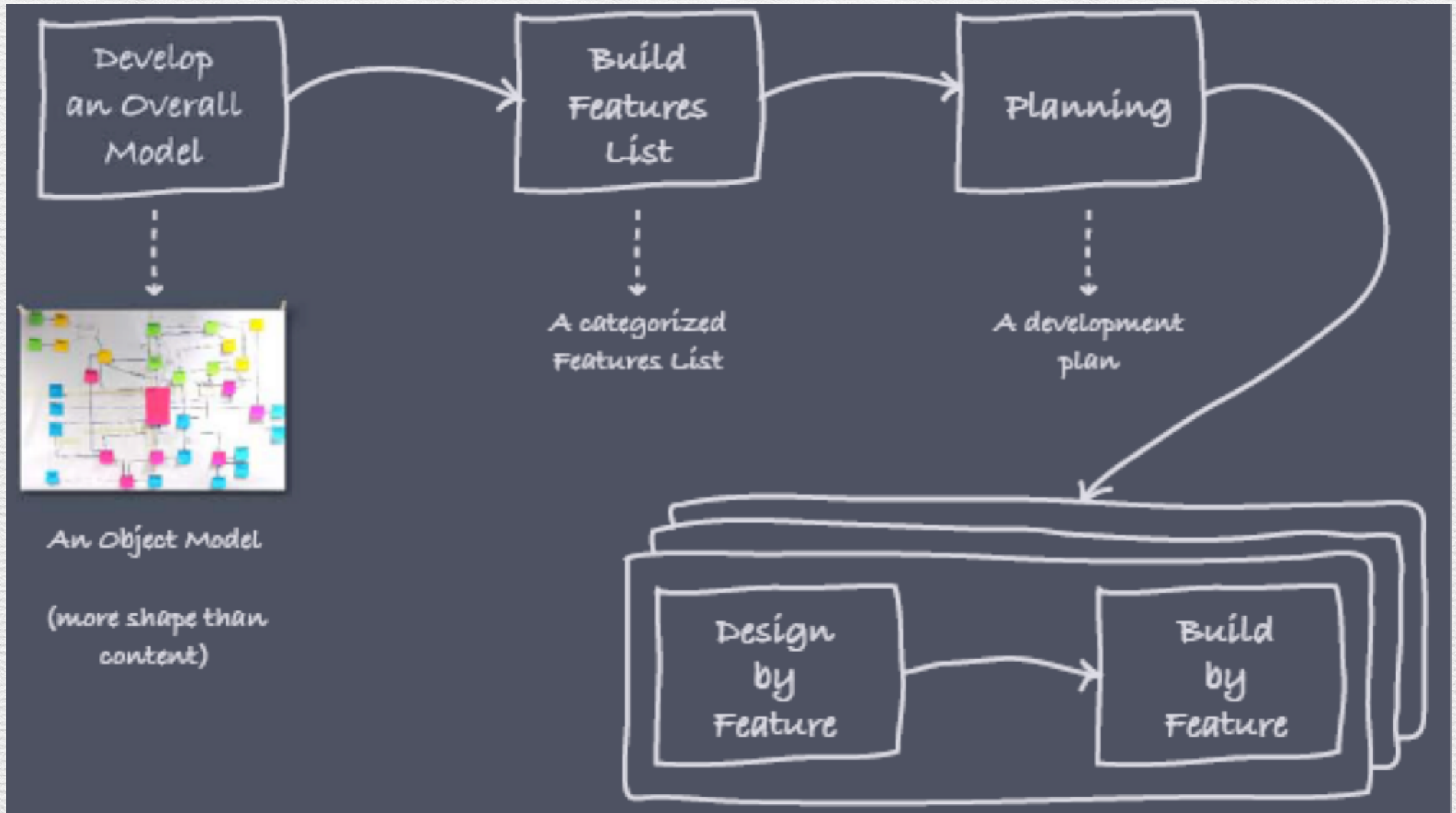
Cvičení - vlastní metodika

- Sesedněte si spolu týmy z ROPR
- Na základě vašich zkušeností si definujte vlastní metodiku vývoje vhodnou pro váš tým.
- Nezapomeňte definovat i doporučené oblasti z Crystal Clear, ne jen povinné.
- Výsledek prezentujte na tabuli (před tabulí).

Feature driven development

- Autoři Jeff De Luca a Peter Cod
- Jedná se o vývoj řízený rysy aplikace (požadavky). Klíčové je pochopení potřeb.
- Patří do skupiny vývoje řízený modely (Model-driven development).
- Předepisuje 5 základních aktivit.

Activity FDD



Aktivita FDD

- Globální model - obvykle vázán na doménu (DSM)
- Sestavení seznamu rysů - posloupnost
 - Subject area->Business activity->Feature
- Plánování - vytvoření vývojového plánu, delegování úkolů vývojářům
- Návrh podle rysu - detailní design vybraných rysů (max. 2 týdny)
- Sestavení podle rysu - implementace, testování, code review, build

Agile Unified Process

- Vytvořil Scott Ambler
- AUP je odlehčená verze RUPu
- Na rozdíl od RUPu má pouze 7 disciplín (Model, Implementation, Test, Deployment, CM, PM, Environment)
- Zvláštnost - dva druhy iterací (development release, production release)

Dynamic systems development method

- Motivace:
 - Nic se nedá udělat napoprvé dobře
 - Obchodní prostředí vždy vygeneruje nějakou změnu
 - Klíčem k úspěchu jsou lidé - zapojte všechny stakeholdery
- Co můžeme ovlivnit:
 - Kvalita
 - Kontrola
 - Přehnané očekávání uživatelů

Dynamic systems development method

- Vytvářeno od roku 1984, první verze uvolněna až v roce 1995.
- Kromě doporučených postupů obsahuje i framework, pod kterým je vyvíjen SW.
- Založen na 8 principech, většina vychází z Agile Manifesto.
- Dokáže vhodně kombinovat práci více týmů na jednom projektu.

<http://www.agilemanifesto.org/>

Manifest Agilního vývoje software

Objevujeme lepší způsoby vývoje software tím,
že jej tvoříme a pomáháme při jeho tvorbě ostatním.

Při této práci jsme dospěli k těmto hodnotám:

Jednotlivci a interakce před procesy a nástroji

Fungující software před vyčerpávající dokumentací

Spolupráce se zákazníkem před vyjednáváním o smlouvě

Reagování na změny před dodržováním plánu

Jakkoliv jsou body napravo hodnotné,
bodů nalevo si ceníme více.

Jednotlivci a interakce nad procesy

- DSDM
 - Týmy vedené dle DSDM musí mít pravomoci pro rozhodování
 - Základ pro úspěch je zainteresovaný uživatel (zákazník)

Fungující software nad dokumentací

- DSDM
 - Zaměřte se na frekvence doručování produktu zákazníkovi
 - Kritérium hodnocení úspěšnosti projektu je jakým způsobem pomáhá businessu (podporuje podnikový proces)
 - Požadavky jsou popsány s vysokou úrovní abstrakce
 - Testování je integrováno v každé fázi životního cyklu

Spolupráce se zákazníkem před vyjednáním

- DSDM
 - Základ pro proces vývoje je kolaborace a kooperace mezi všemi zúčastněnými na projektu

Reakce na změnu před dodržováním plánu

- DSDM
 - Iterativní a inkrementální přístup doručuje přesné řešení
 - Všechny změny uskutečněné během vývoje se dají vrátit

Principy

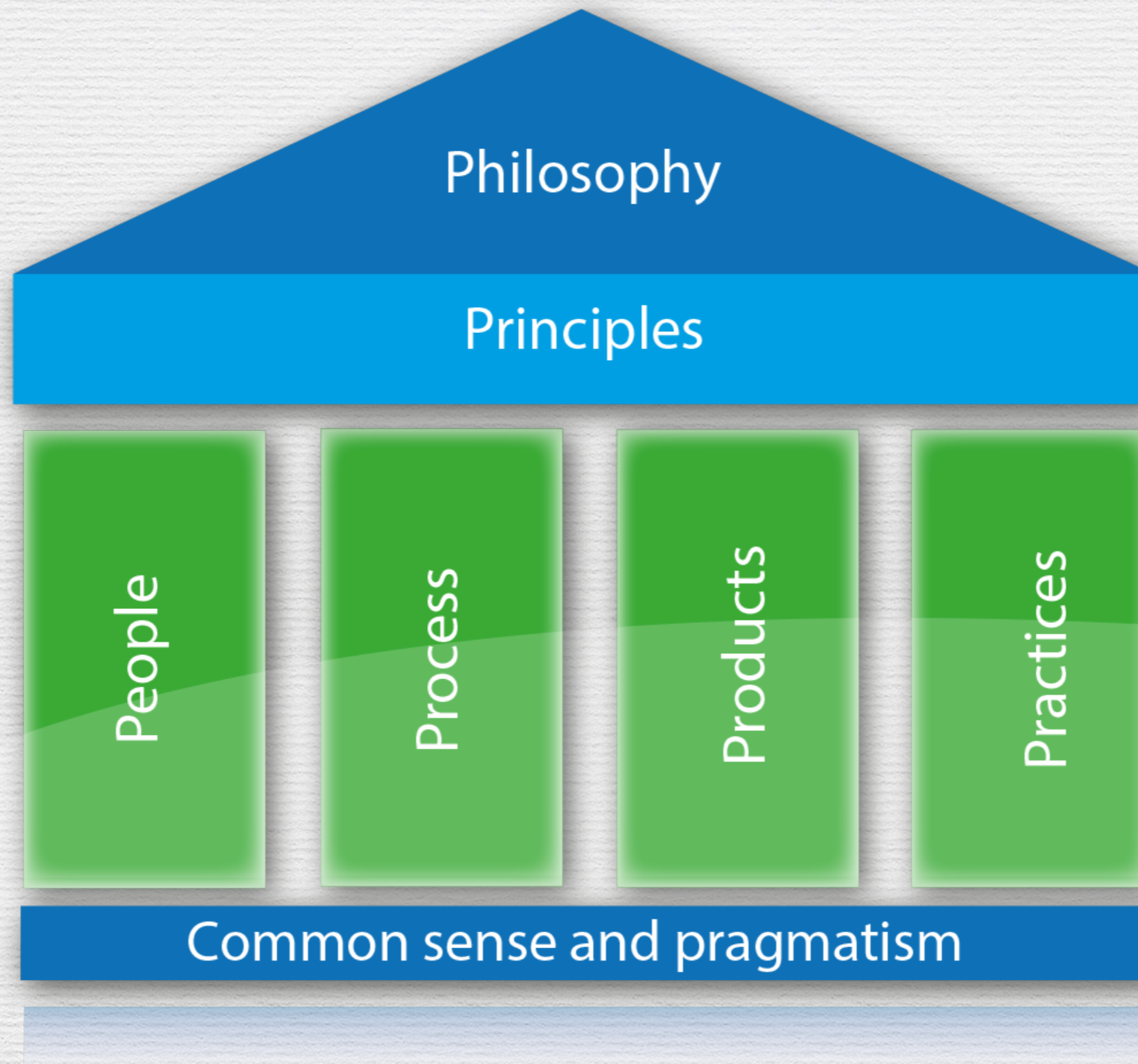
- Focus on business needs
- Deliver on time
- Collaborate
- Never compromise quality



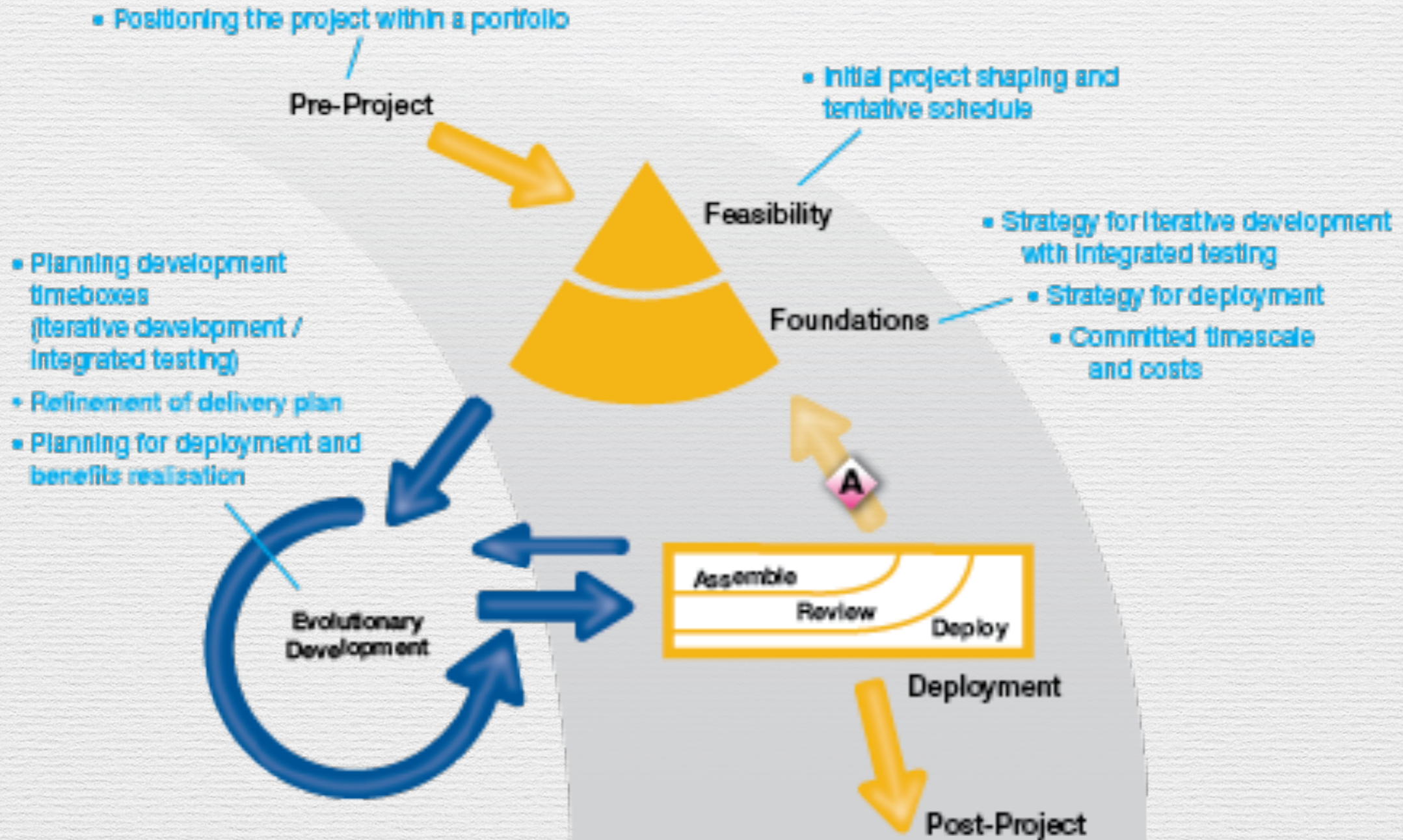
Integrace s ostatními metodikami/ frameworky/technikami?

- Prince 2
- XP
- Project Management Institute
- KANBAN
- SCRUM

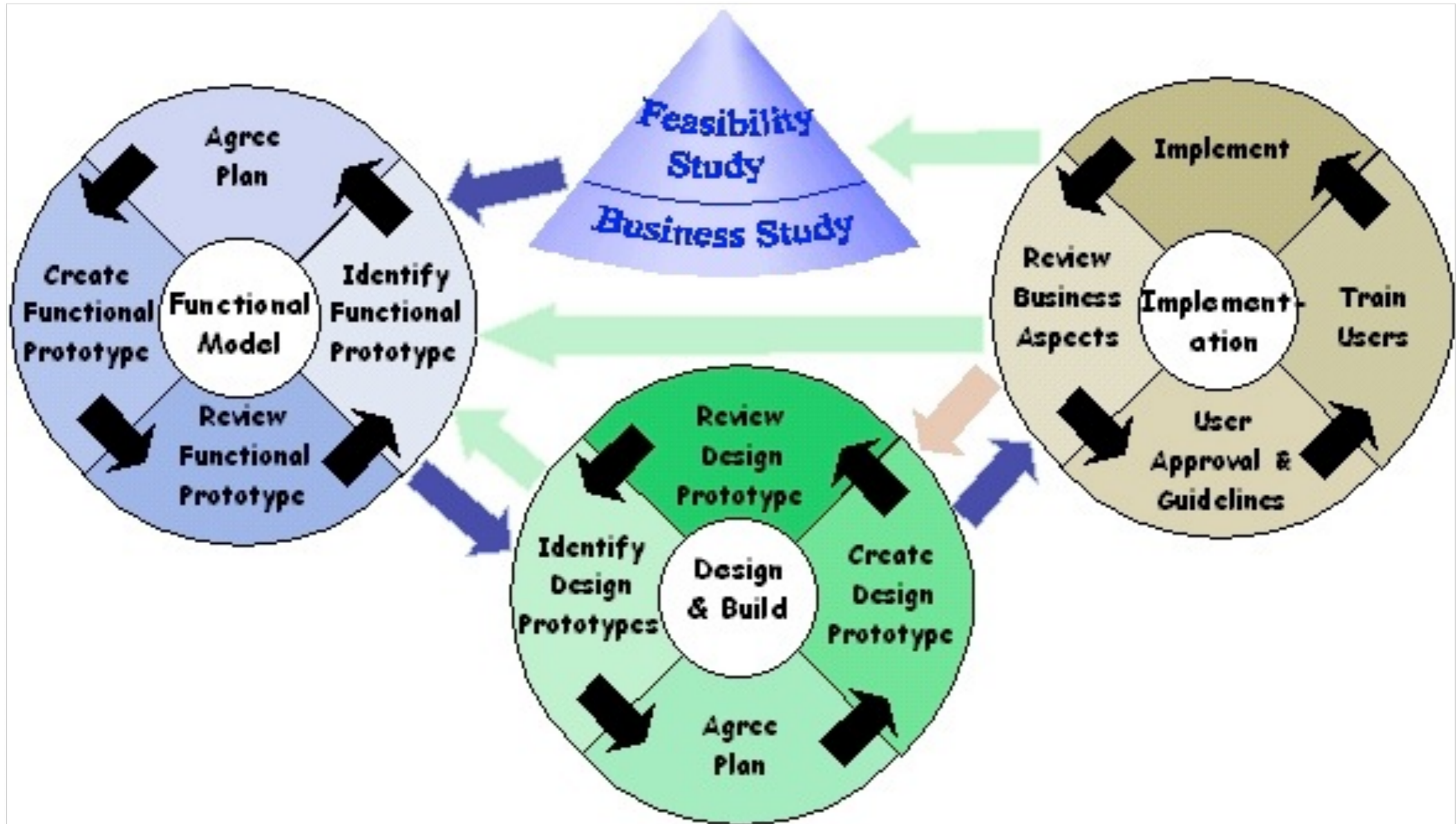
Struktura DSDM

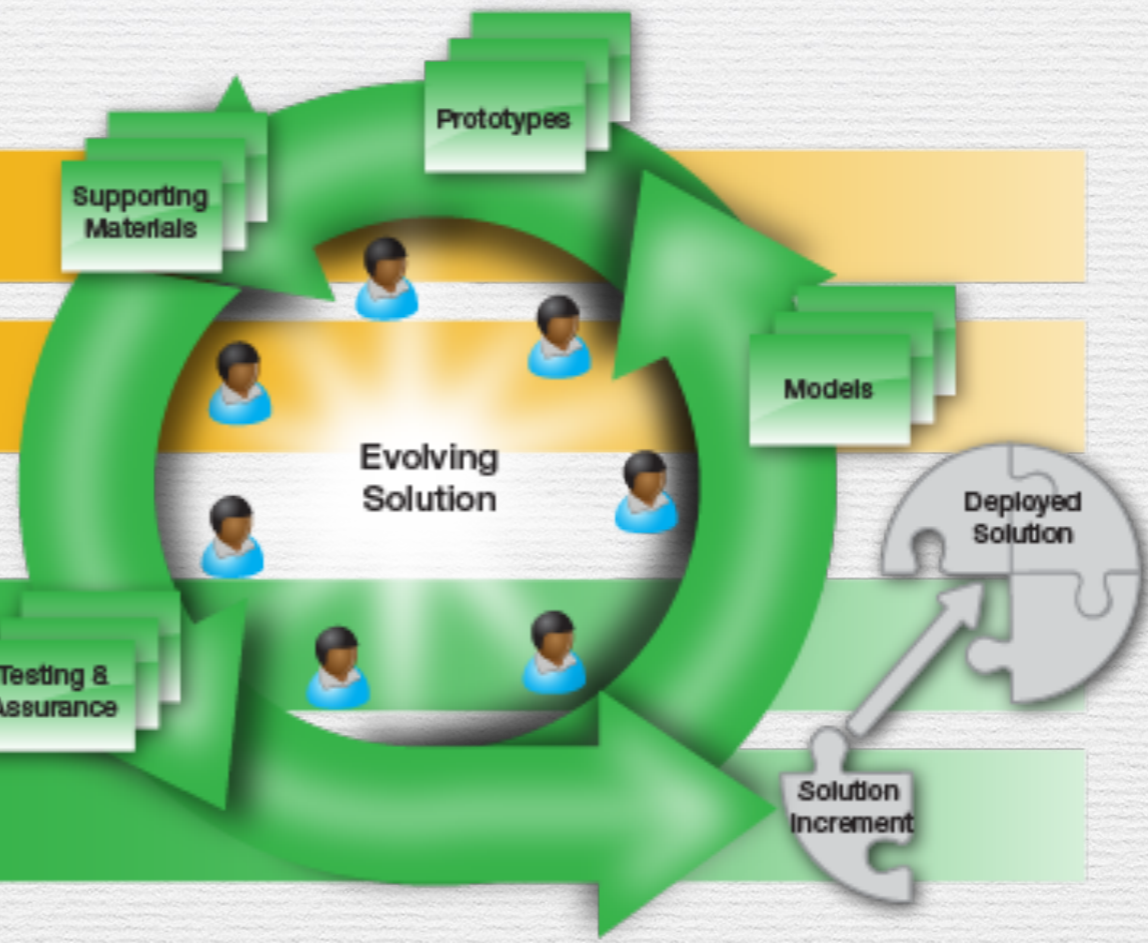
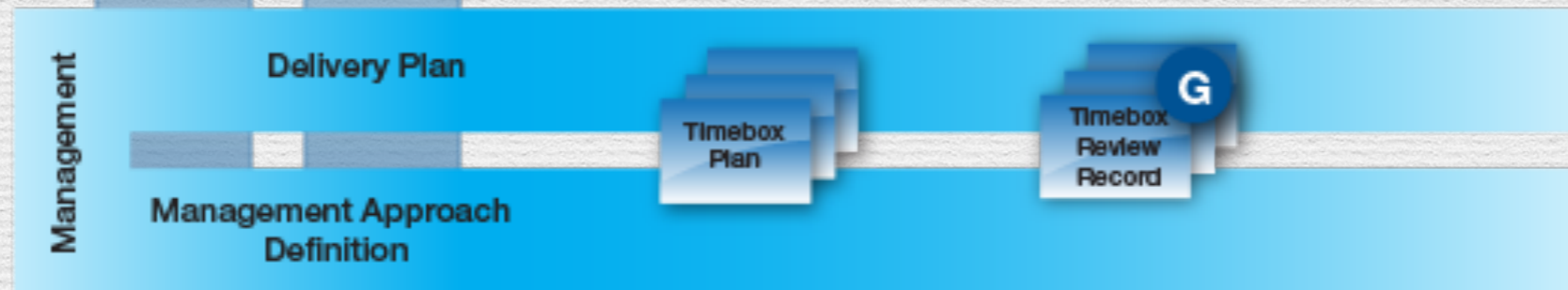
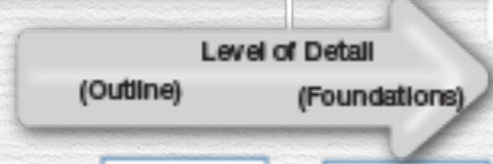
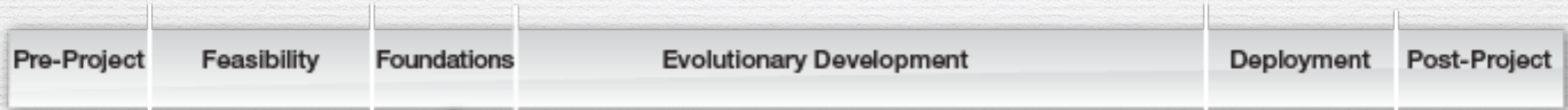


Proces podle DSDM



Životní cyklus DSDM



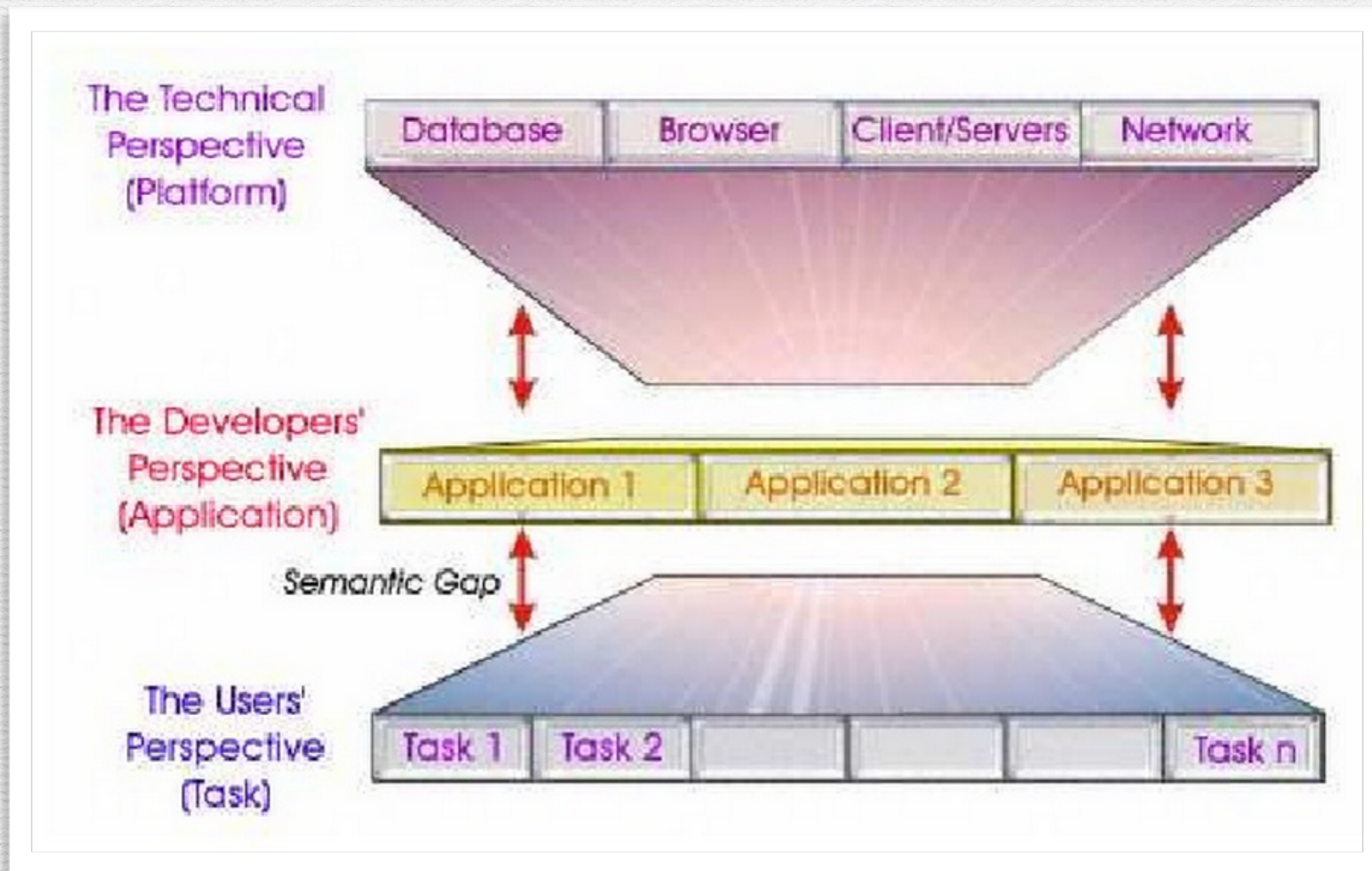


Životní cyklus DSDM

- Studie proveditelnosti
- Obchodní studie
- Stanovení modelu funkcí
- Návrh a implementace
- Nasazení
- Kontrola, zda projekt splnil původní očekávání

Uživatel vám vždy řekne cokoli na
co se ho zeptáte

... a nic víc.



In scope for this timeframe

(Project / Increment / Timebox)

Must Have



Typically
no more
than
60% effort

Should Have



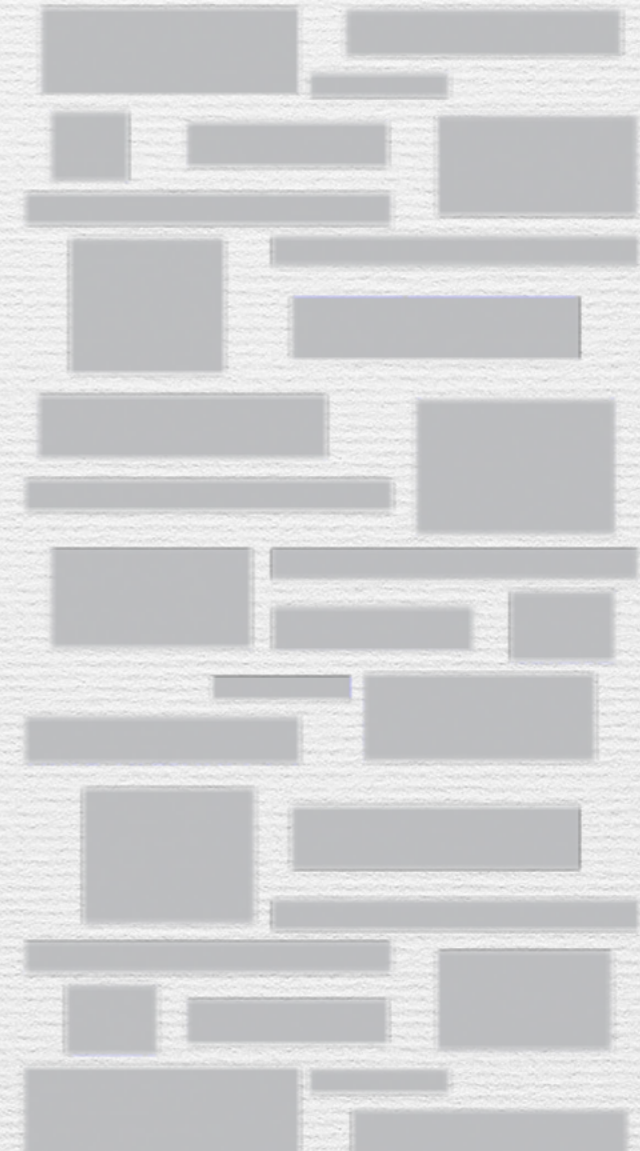
Could Have



Typically
around
20% effort

Out of scope for this timeframe

Won't Have this time



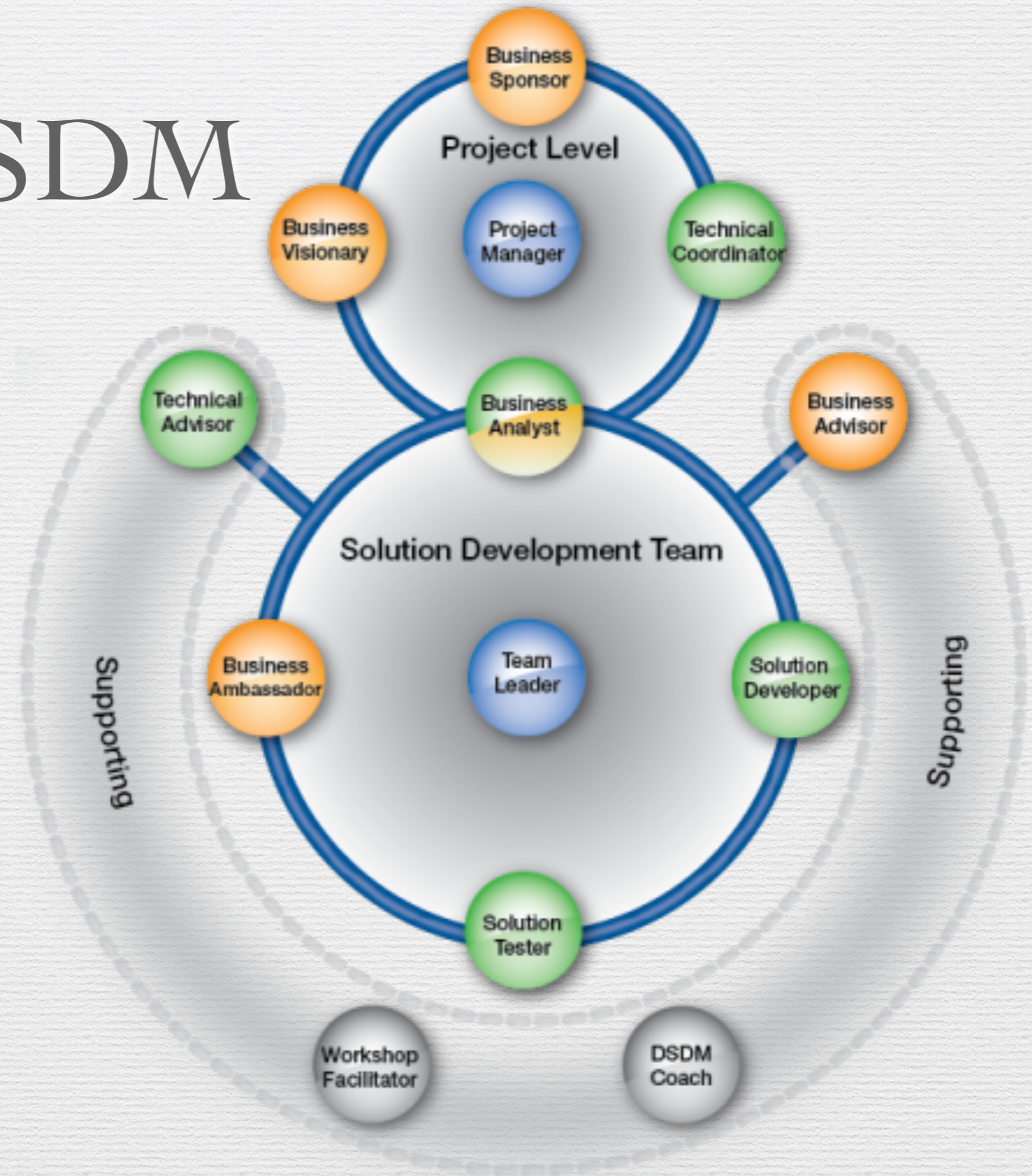
Nejdůležitější postupy

- **Modelování**
- **Prototypování**
- **Časový rámec (timeboxing)**
- **MoSCoW** - způsob prioritizace požadavků
 - *Must* - nezbytné požadavky na budovaný systém
 - *Should* - doporučené požadavky, neohrožují životaschopnost projektu
 - *Could* - absence těchto požadavků na projekt nemá žádný dopad
 - *Won't* - požadavky, které lze realizovat kdykoliv později

Timeboxing

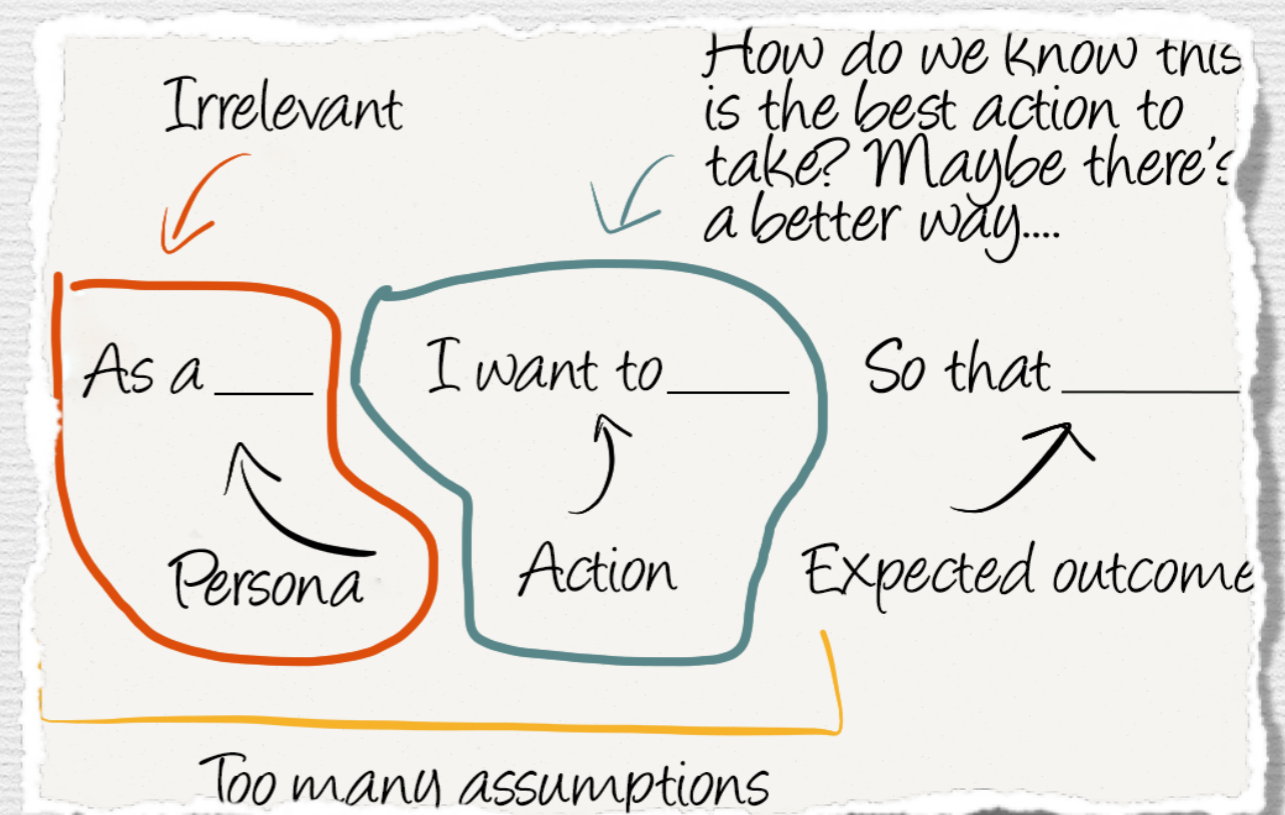


Role v DSDM



Správna User Story dle DSDM

- **As a** <role>
- **I need** <requirement or feature>
- **So that** <goal / value>



Správna User Story dle DSDM

- **As an** Investor,
- **I need** to see a summary of my investment accounts,
- **So that** I can decide where to focus my attention.

Disciplined Agile Delivery (DAD)

Disciplined Agile Delivery

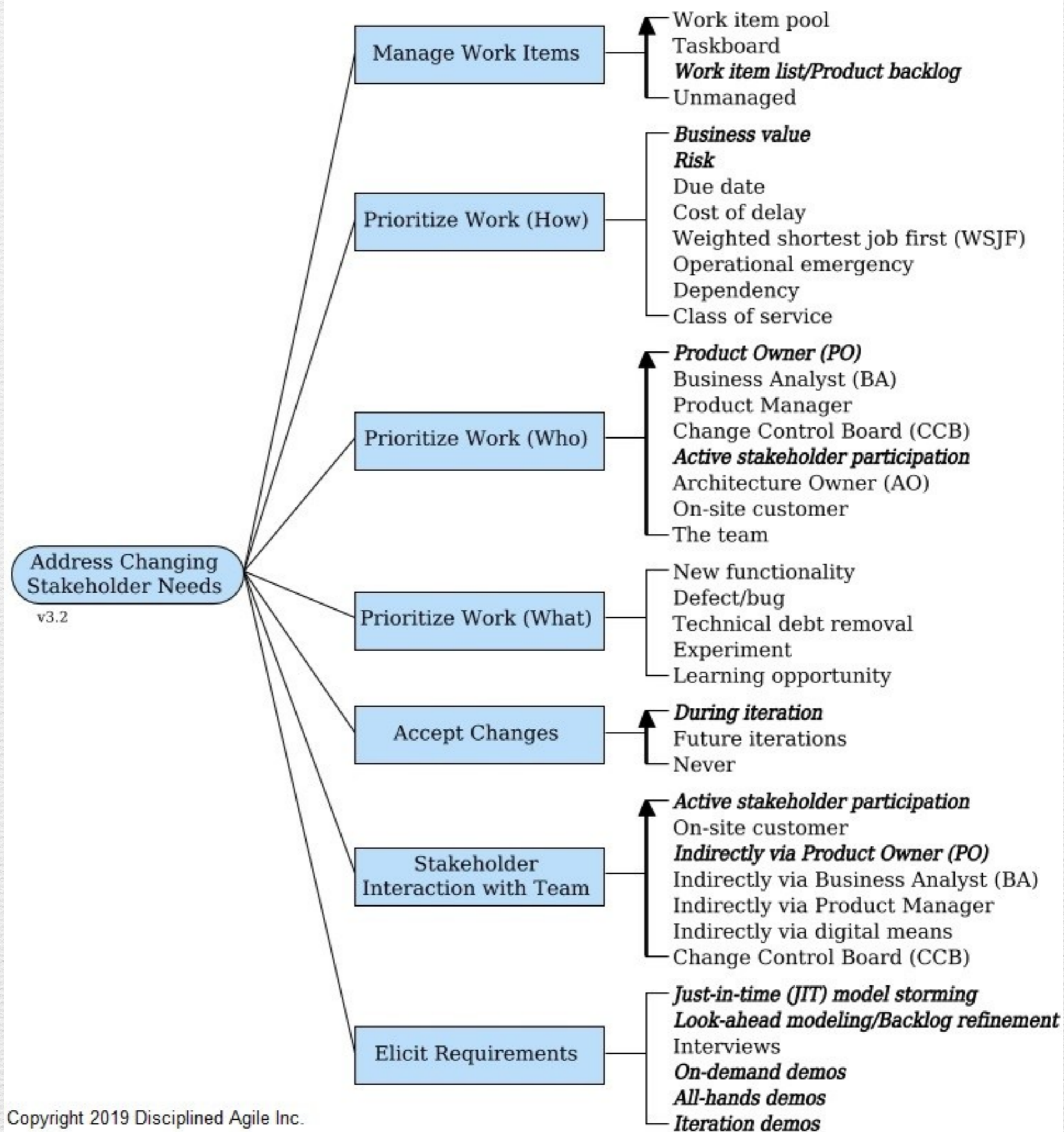
- Pomáhá se rozhodovat, který proces (technika, praktika) je pro daný projekt vhodný.
- Založen na myšlence iterativně inkrementálního vývoje.
- Staví na základech Scrumu a doplňuje ho o možnosti škálování projektu.
- Inspirací pro oblasti frameworku byly různé projekty z IBM Rational (2006-2012)
- V podstatě je kombinací XP, Scrumu, Kanbanu, Agile Modeling, Agile data...



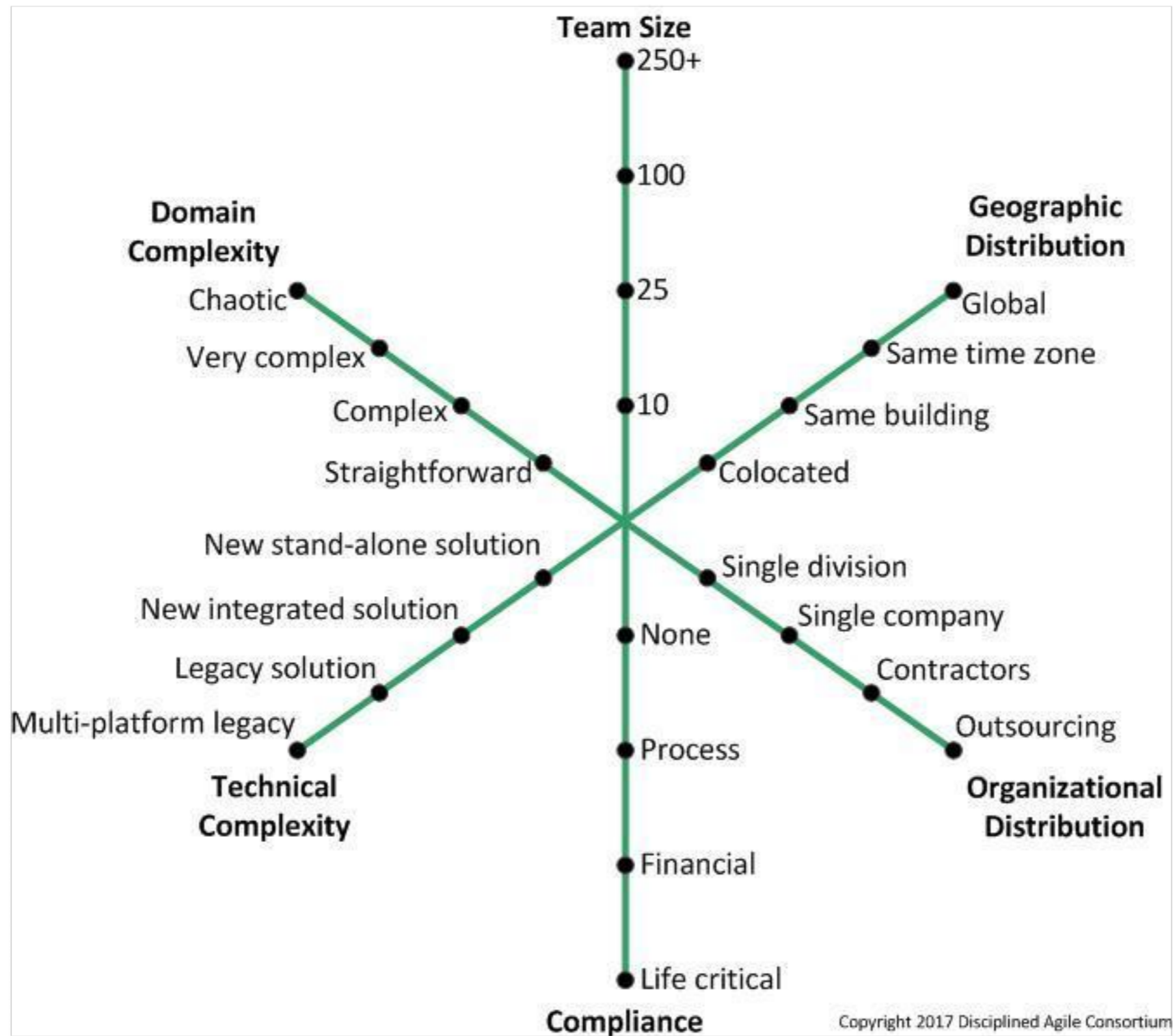
Obecné principy

- **Delight Customers** - Zákazníka potěšíte tím, že produkt nejenom plní jeho zadání, ale v některých oblastech předčí očekávání.
- **Be Awesome** - Motivovaní jedinci vytvářejí lepší práci (stejně jako u předchozích přístupů).
- **Pragmatism** - Kladem extrémní důraz na efektivitu, nestačí nám jen to, že jsme agilní.
- **Context Counts** - Každý člověk, projekt a organizace jsou jedineční. Musíme pro danou situaci najít nejlepší strategii a postupně ji rozvíjet. Nebojte se nových technik, nepoužívejte golden hammer.
- **Choice is Good** - Je dobré mít několik alternativ, abyste se mohli rozhodnout, který přístup je pro danou situaci ten nejlepší.
- **Optimize Flow** - Organizace je živá entita, musíte zajistit, aby byly týmy dobře sladěny a průběžně jste kooperaci zlepšovali.
- **Enterprise Awareness** - Členové týmy znají business, pro který software vyvíjejí. Díky tomu dokážou posoudit dopad jejich rozhodnutí na fungování dané společnosti.

Choice is good

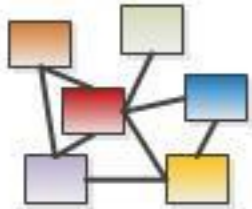


Context counts





Community Awareness
“How can I give back to my community?”



Enterprise Awareness
“How can I help my organization?”



Departmental Awareness
“How can I help my department?”



Team Awareness
“How can I help the team?”



Individual Awareness
“How can I be the best me?”

Copyright 2013-2014 Disciplined Agile Consortium

Enterprise Awareness

Životní cyklus

Inception

- Vytvoř tým
- Stanov vizi projektu
- Přizpůsob firemním potřebám (kultuře)
- Urči rozsah projektu
- Vytvoř technickou strategii
- Vytvoř release plan
- Zformuj pracovní prostředí
- Dohodni financování
- Identifikuj rizika

Construction

- Vytvoř potenciálně použitelný produkt
- Zapracuj změny identifikované zákazníkem při zpracování projektu
- Přibliž se k “doručitelné” verzi
- Zlepšuj kvalitu
- Ověř architekturu co nejdříve

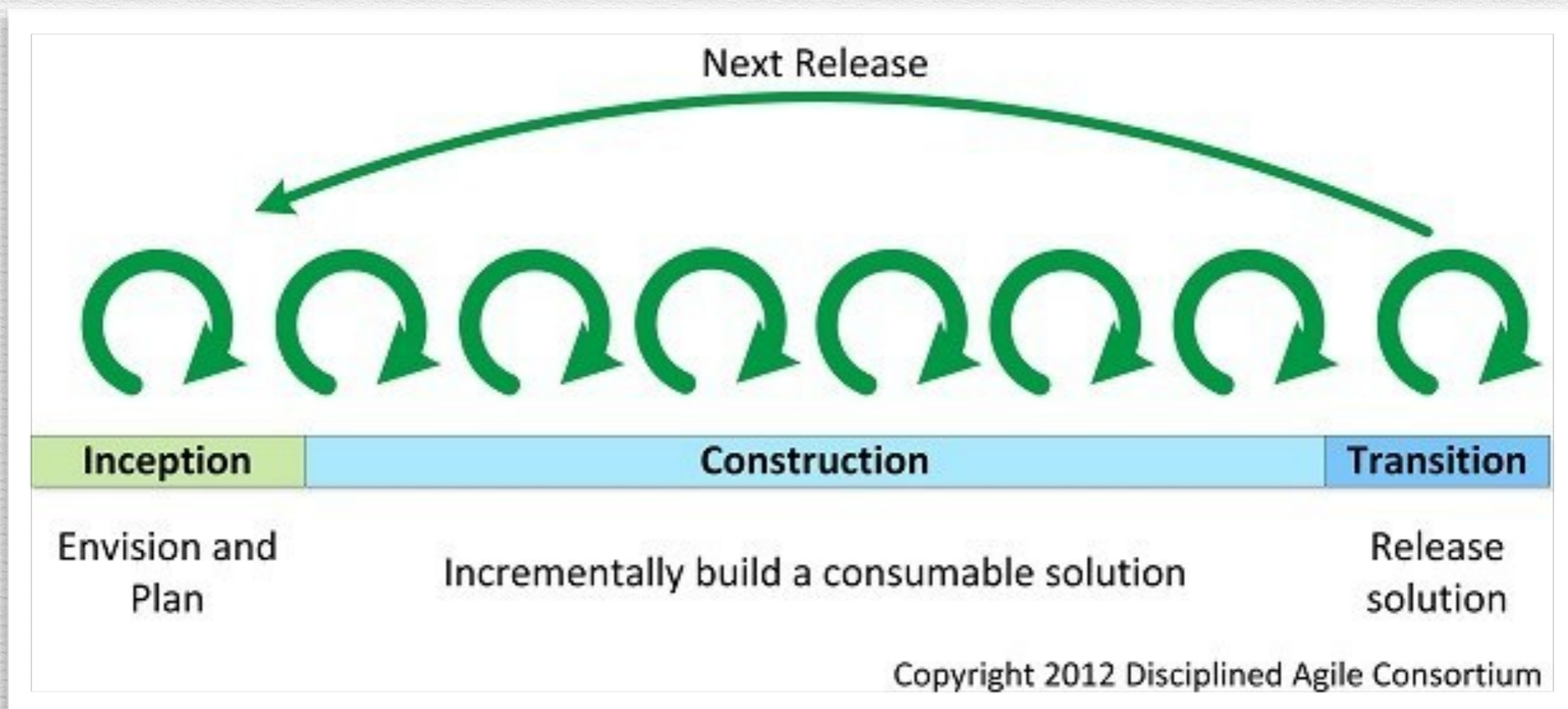
Transition

- Ověř, že řešení je použitelné
- Nasad' řešení

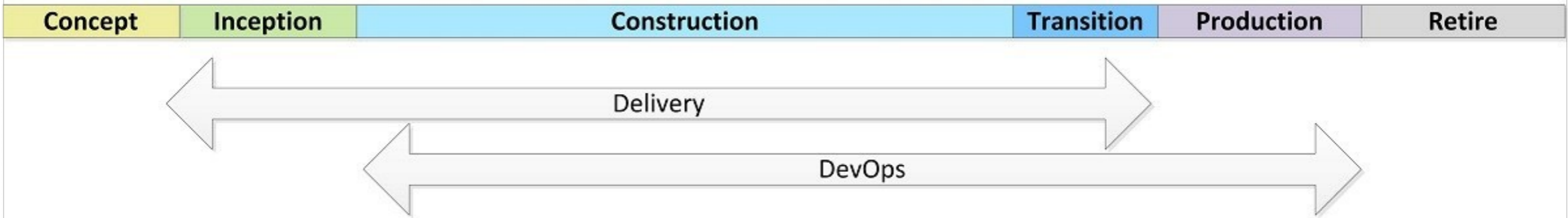
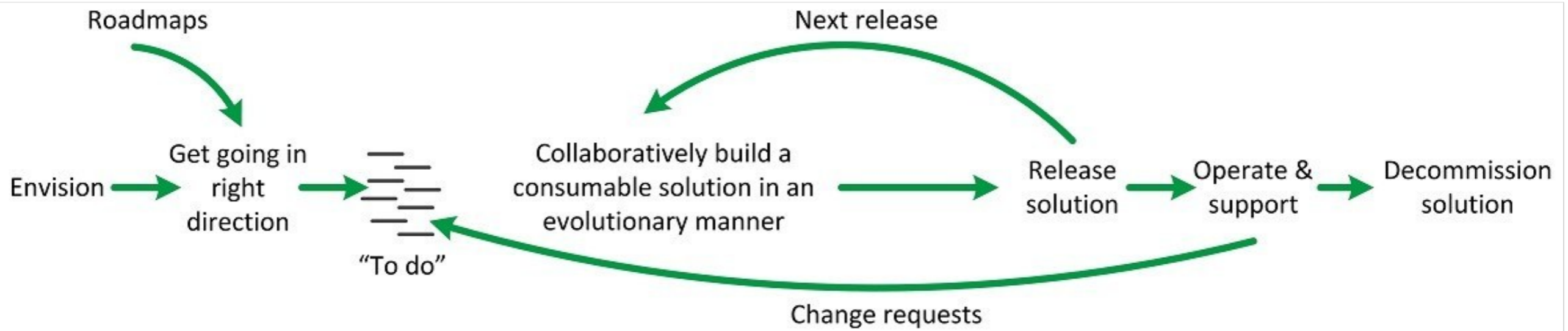
Průběžné cíle

- Kontroluj, že splňuješ misi projektu
- Umožni profesní růst členů týmu
- Pojmenovávej rizika
- Zlepšuj proces a prostředí
- Vylepšuj stávající infrastrukturu

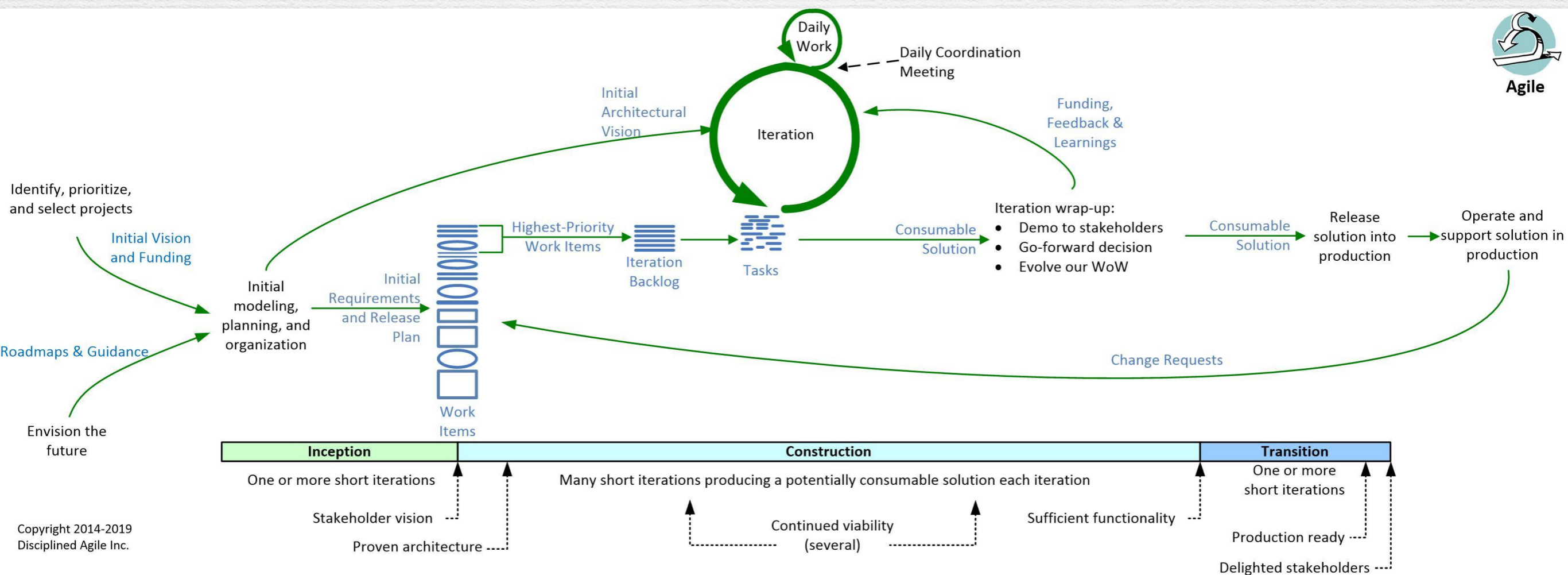
Životní cyklus



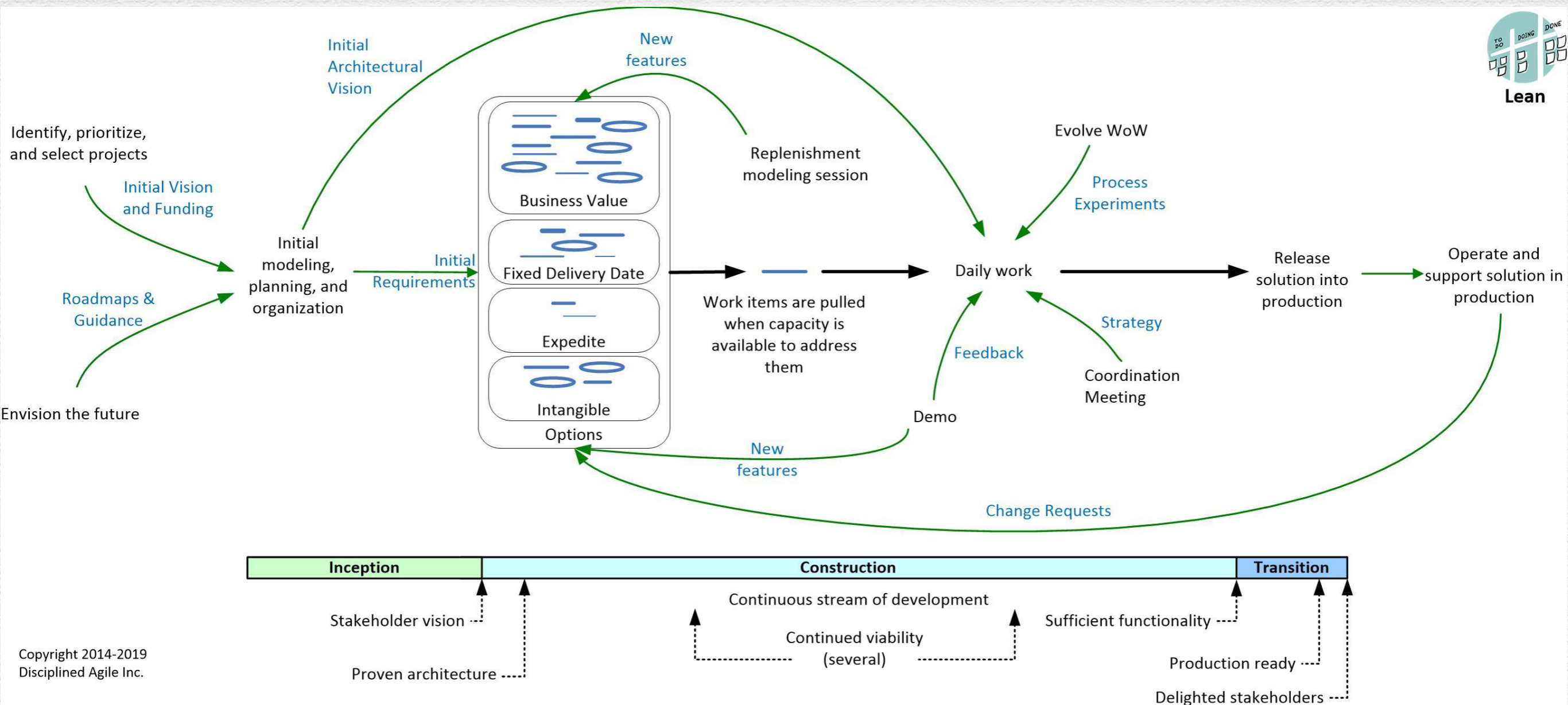
Životní cyklus



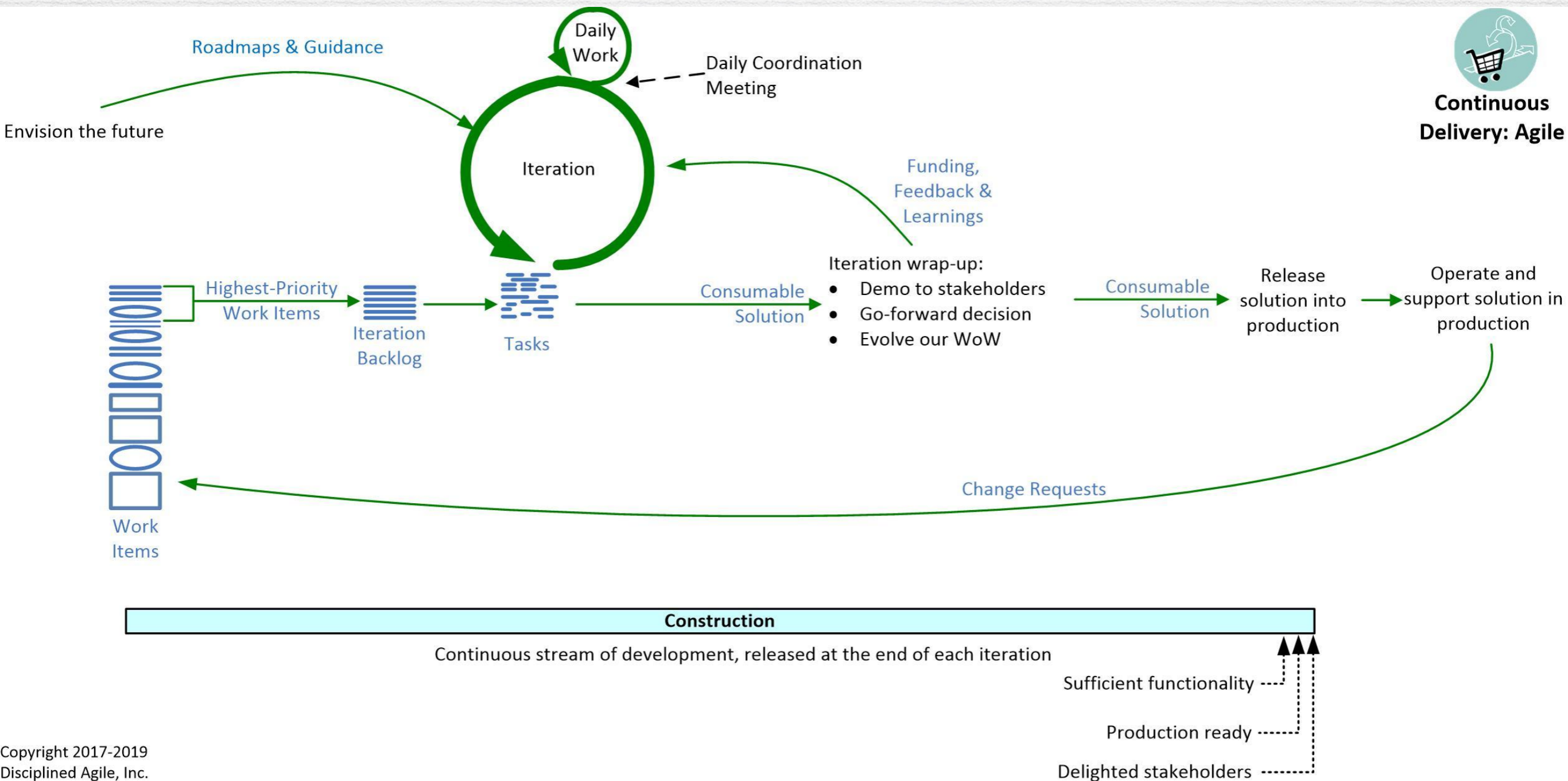
Životní cyklus - Scrum



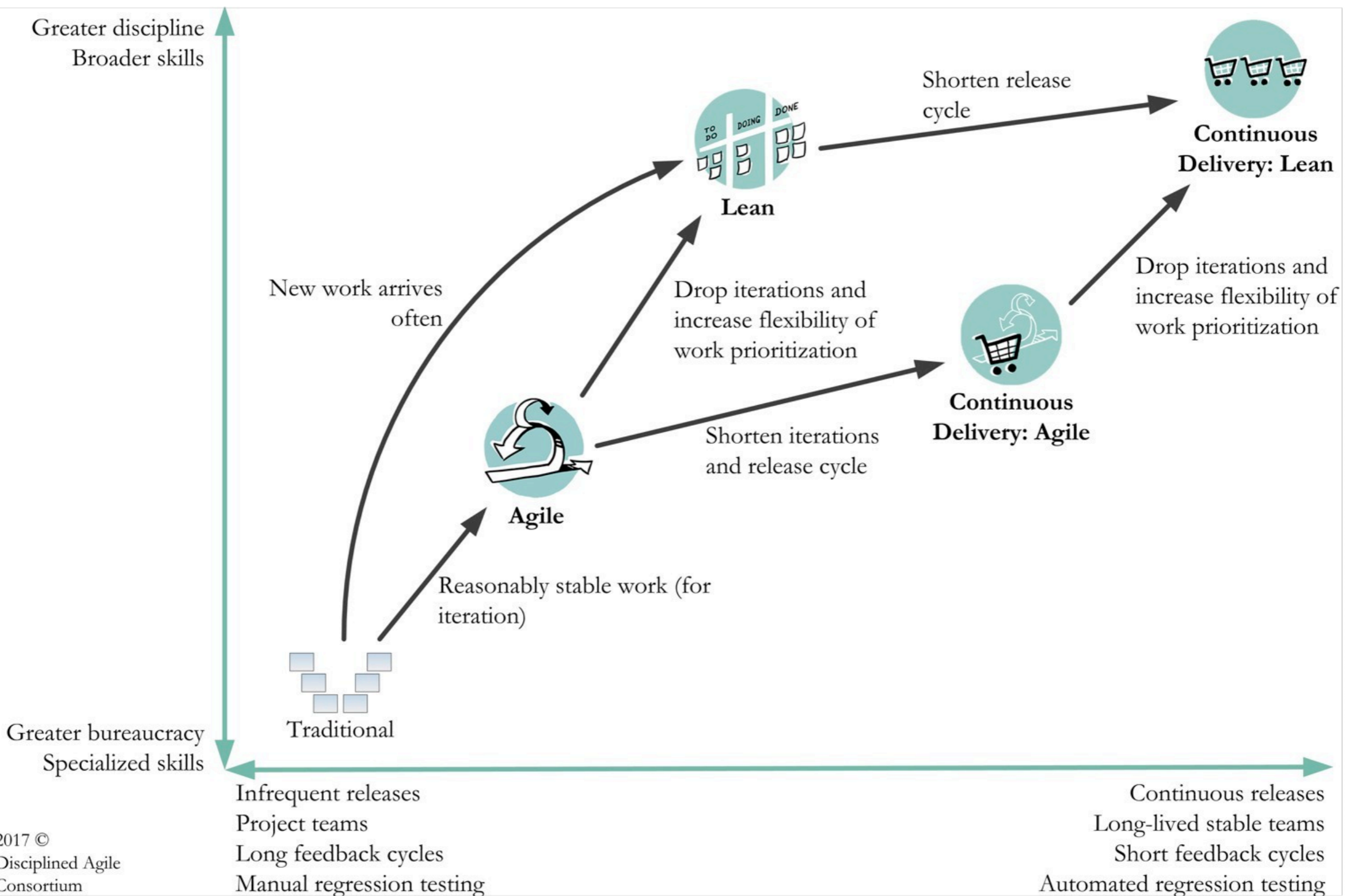
Životní cyklus - Kanban (Lean)



Životní cyklus - Continuous Delivery



Evolve



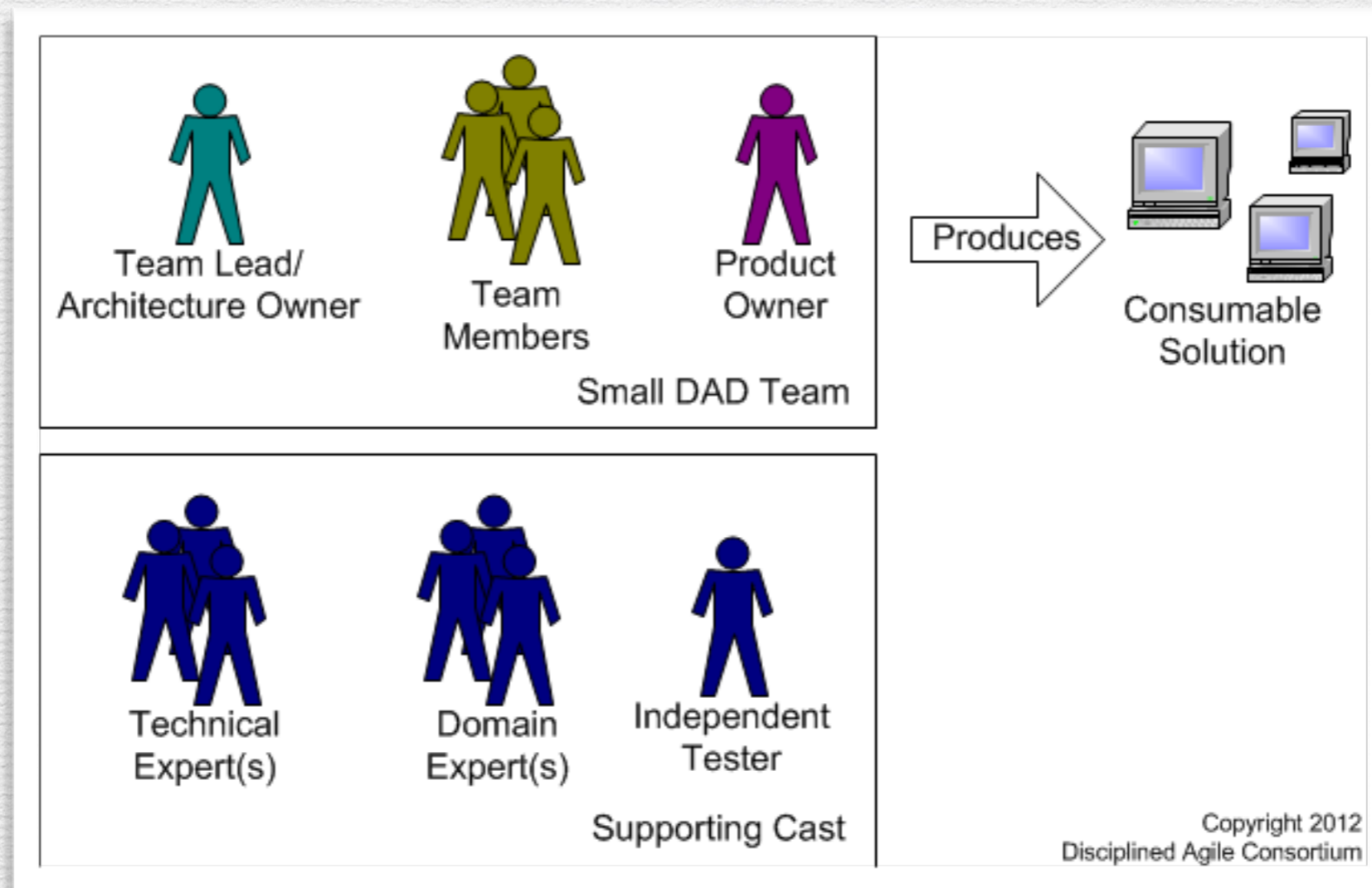
Hlavní Role

- **Stakeholder** - Kdokoliv, kdo je materiálně zainteresovaný na úspěšném dokončení projektu.
- **Product Owner** - Člověk, který je součástí týmu a reprezentuje názor všech stakeholderů na projektu.
- **Team Member** - Vytváří řešení pro stakeholdery (analýza, návrh, architektura, programování, odhady)
- **Team Lead** - Většinou se označuje termínem “agilní kouč”, zodpovědný za proces, odstraňuje překážky, zajišťuje, že tým má nezbytné zdroje
- **Architecture Owner** - Má rozhodovací slovo v oblasti architektury.

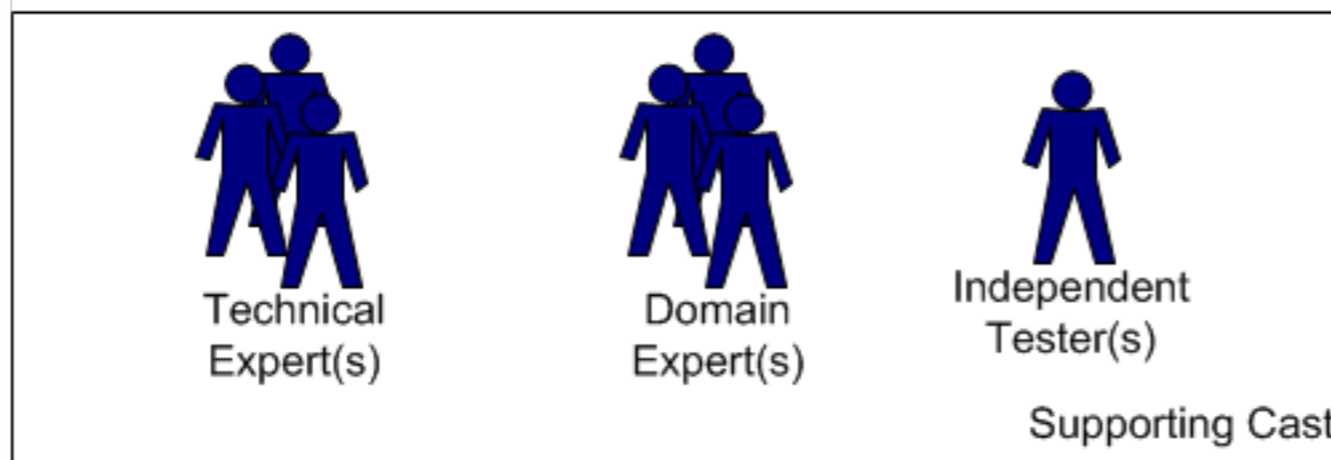
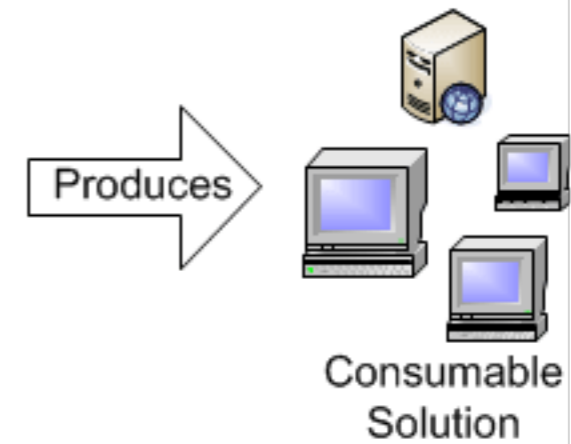
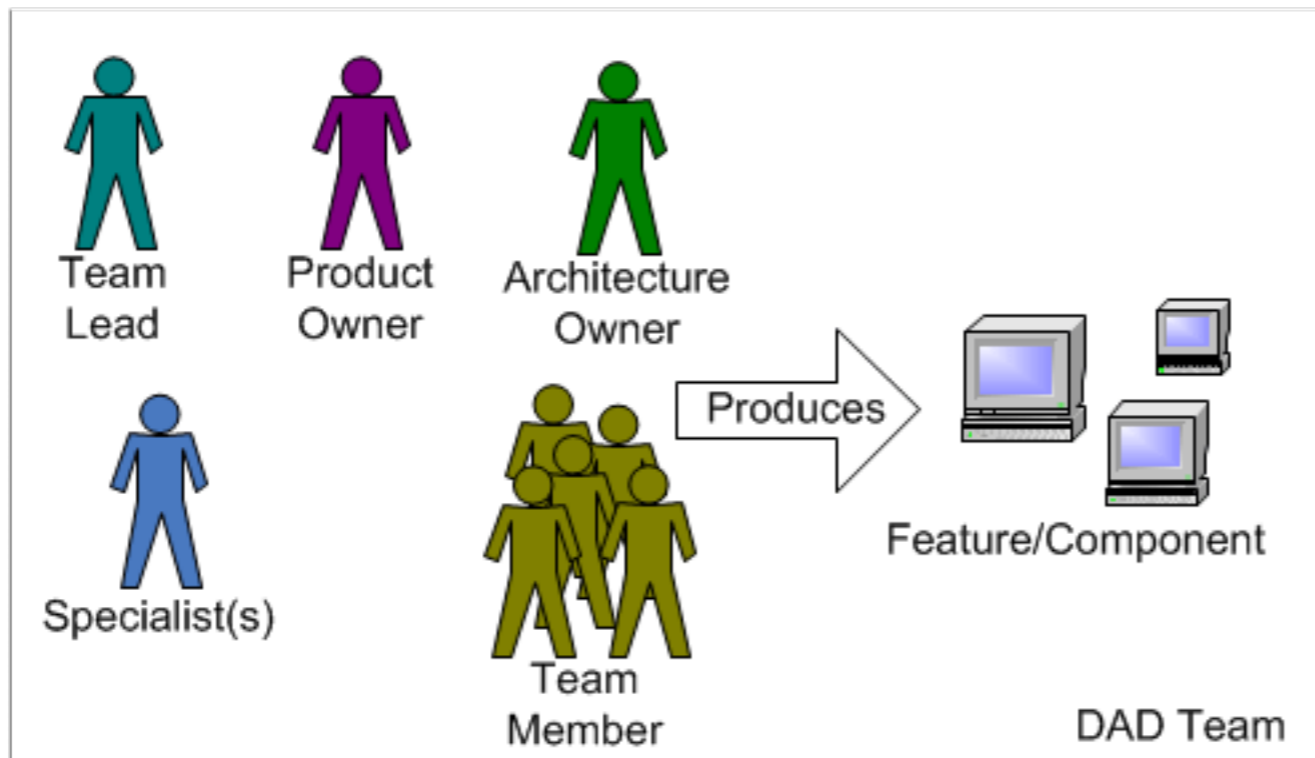
Podružné role

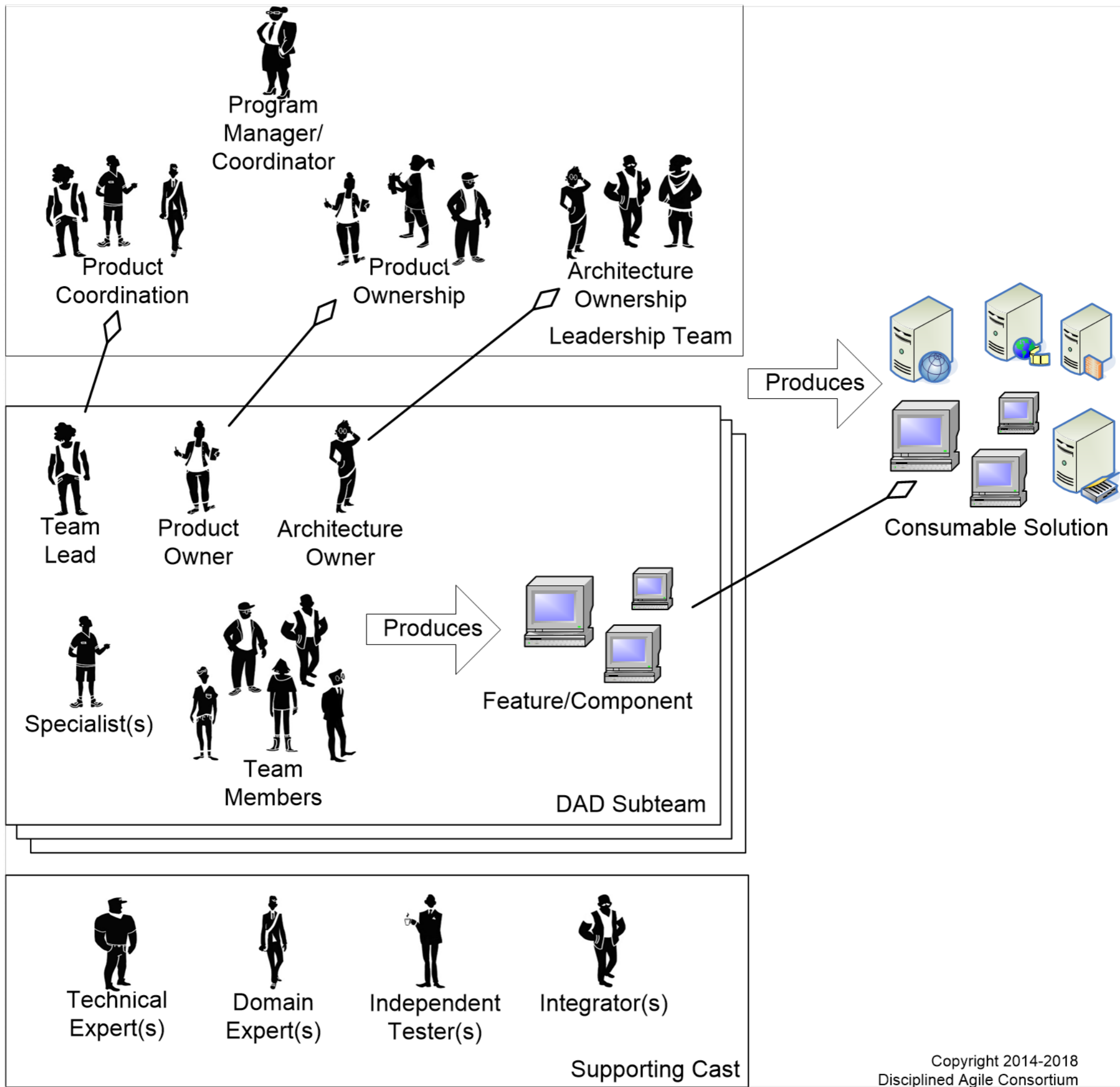
- **Specialist** - Obecně jsou všichni členové týmu specialisty, tato role je většinou spojena s externím poradcem
- **Domain Expert** - Člověk se znalostí konkrétní domény (telco, automotive, bankovníctví)
- **Technical Expert** - Specializovaná odborná role - UX designer, řešení performance issues, odborník na bezpečnost
- **Independent Tester** - U složitějších projektů je přizván externí tým (interní tým stále testuje vlastní práci)
- **Integrator** - U složitých systémů slouží jako architekt celého řešení, navrhuje propojení jednotlivých systémů (integraci)

Malý tým



Středně velký tým





A co Agile a velké
projekty?

Plánování Sprintu

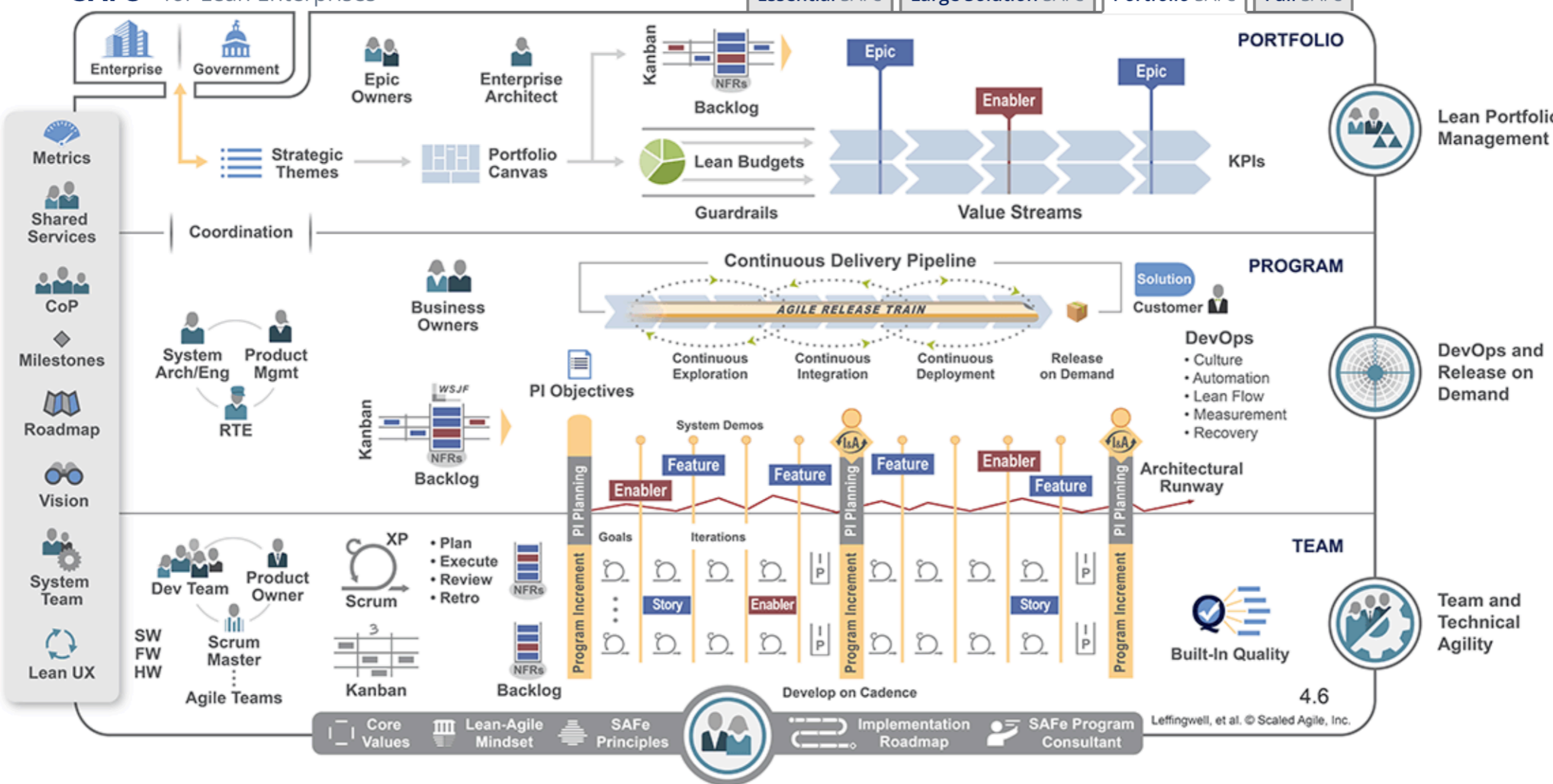


Možnosti

- Scaled Agile Framework (Leffingwell)
- Disciplined Agile Delivery (Ambler)
- Large-Scale Scrum (Larman & Vodde)
- Scrum at Scale (Suttherland)
- Enterprise Scrum (Beedle)

SAFe® for Lean Enterprises





Essential SAFe | Large Solution SAFe | Portfolio SAFe | Full SAFe



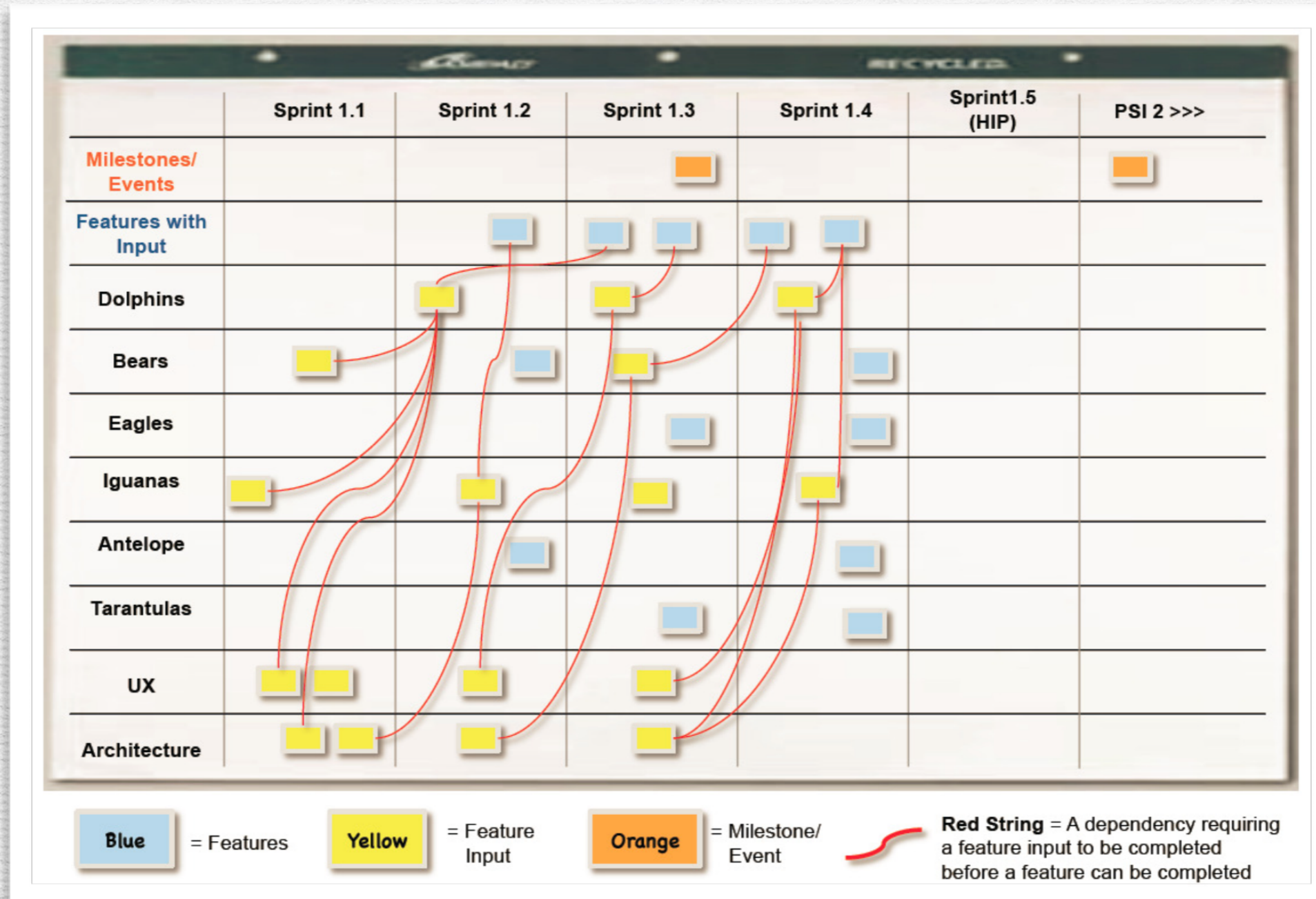
day 1

8:00-9:00	Business Context	
9:00-10:30	Product/ Solution Vision	
10:30-11:30	Architecture Vision & Development Practices	
11:30-1:00	Planning Context & Lunch	
1:00-4:00	Team Breakouts	1 2 3 4
4:00-5:00	Draft Plan Review	
5:00-6:00	Management Review & Problem Solving	

day 2

8:00-9:00	Planning Adjustments	
9:00-11:00	Team Breakouts	1 2 3 4
11:00-1:00	Final Plan Review & Lunch	
1:00-2:00	Program Risks	!
2-2:15	PSI Confidence Vote	
2:15-???	Plan Rework If Necessary	1 2 3 4
After Commitment	Planning Retrospective & Moving Forward	

Dependency Board

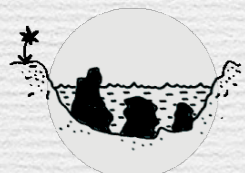


LeSS

<https://youtu.be/e7mzpKHOAHs>

LeSS

LeSS BOOK CHAPTER 2:
INTRODUCTION



LEAN THINKING



SYSTEMS THINKING



PRINCIPLES



COACHING



ADOPTION



CONTINUOUS IMPROVEMENT

FEATURE TEAMS



TEAMS



STRUCTURE



ORGANIZATION



COMMUNITIES



CONTINUOUS INTEGRATION

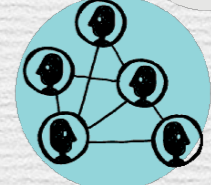
TECHNICAL EXCELLENCE



ARCHITECTURE & DESIGN



ROLE OF MANAGERS

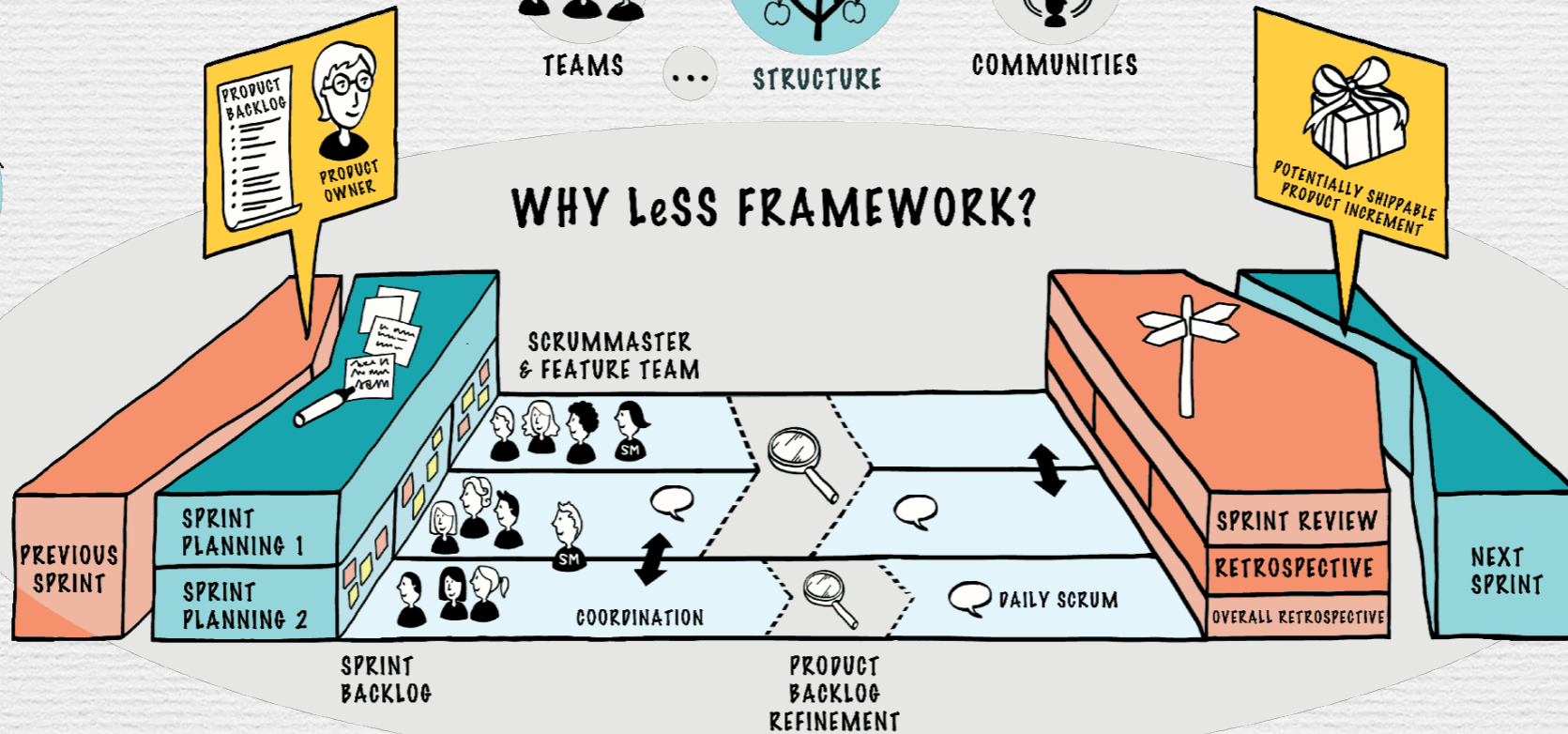


MANAGEMENT



GO SEE

WHY LeSS FRAMEWORK?



PRODUCT OWNER TEAM



LeSS HUGE



REQUIREMENT AREAS

Structure - Organizing by customer value

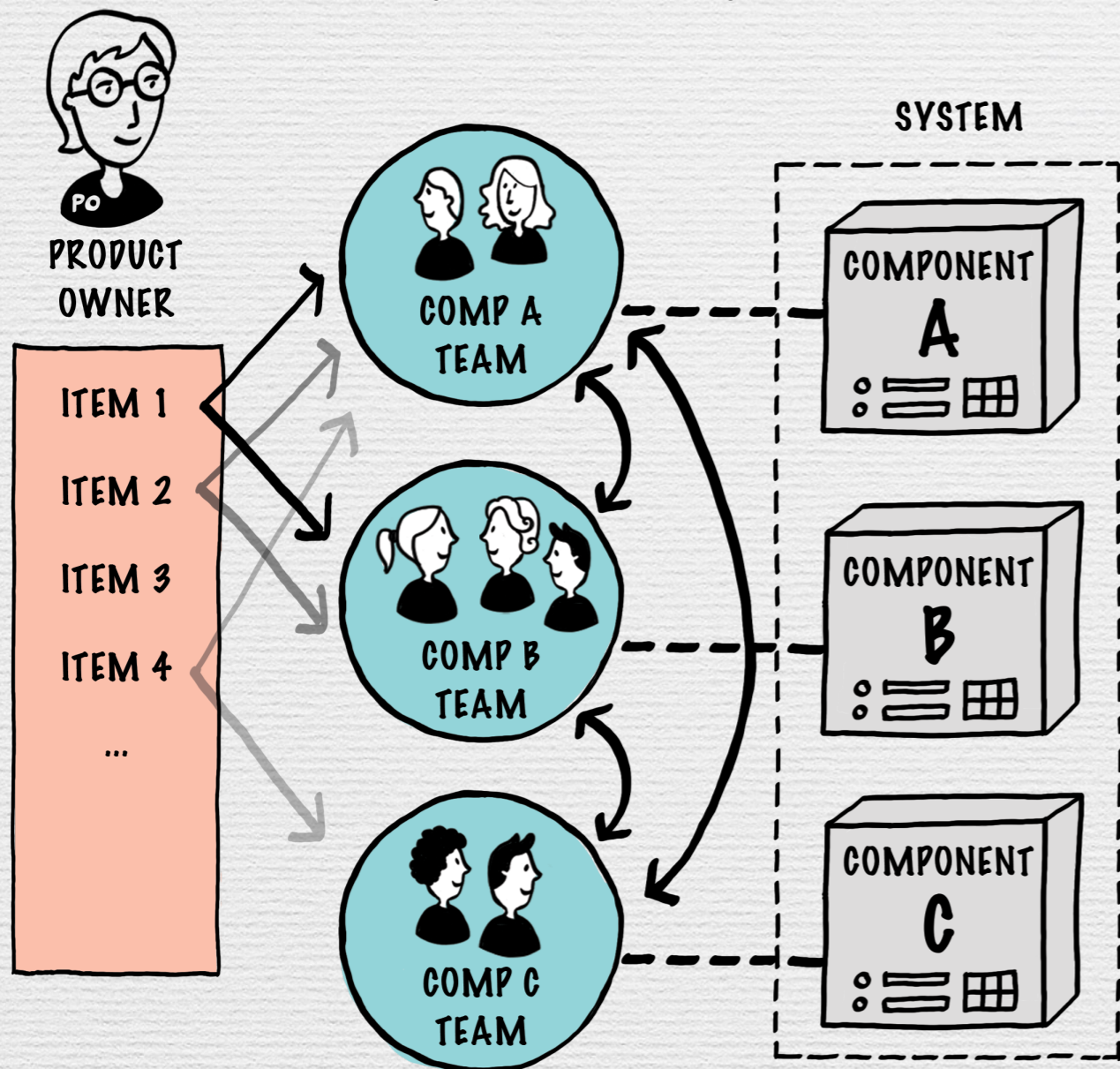
- Cross-functional týmy
 - Dokáže realizovat komplexní hodnotu pro zákazníka.
- Feature týmy
 - Užší zaměření a snaží se co nejvíc minimalizovat předávání informací přes prostředníky mezi zákazníkem a týmem.
- Specializace na zákaznický rozměr
 - Většinou se specializuje podle technologie (iOS development, Android development)
 - Lépe je organizovat podle zákaznické domény (mobilní platby, administrace, reporty)

Structure - týmy v LeSS

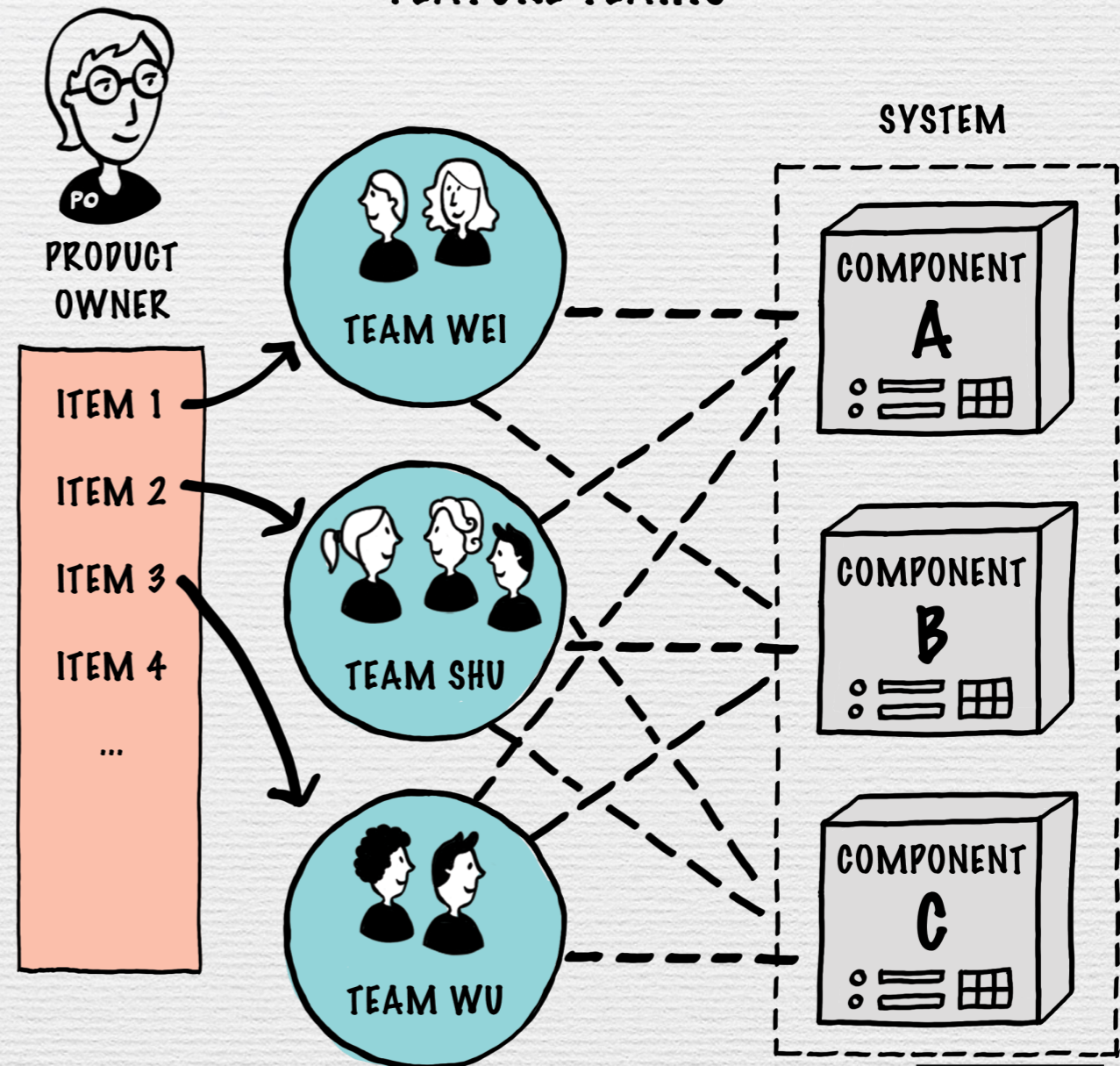
- **Dedikované týmy** - každý člen týmu v něm má 100% utilizaci, lépe se udržuje týmová zodpovědnost, vytváření vlastní procesy
- **Cross-functional týmy** - Každý člen udržuje znalosti o funkcionalitě, aby mohl být produkt doručen (UI část, servisní část, etc.)
- **Co-located týmy** - Všechny týmy jsou umístěny ve stejné místnosti pro lepší sdílení znalostí odborníků.
- **Long-lived týmy** - Čím déle je tým konzistentní, tím více hodnoty vytváří.

Component vs. Feature týmy

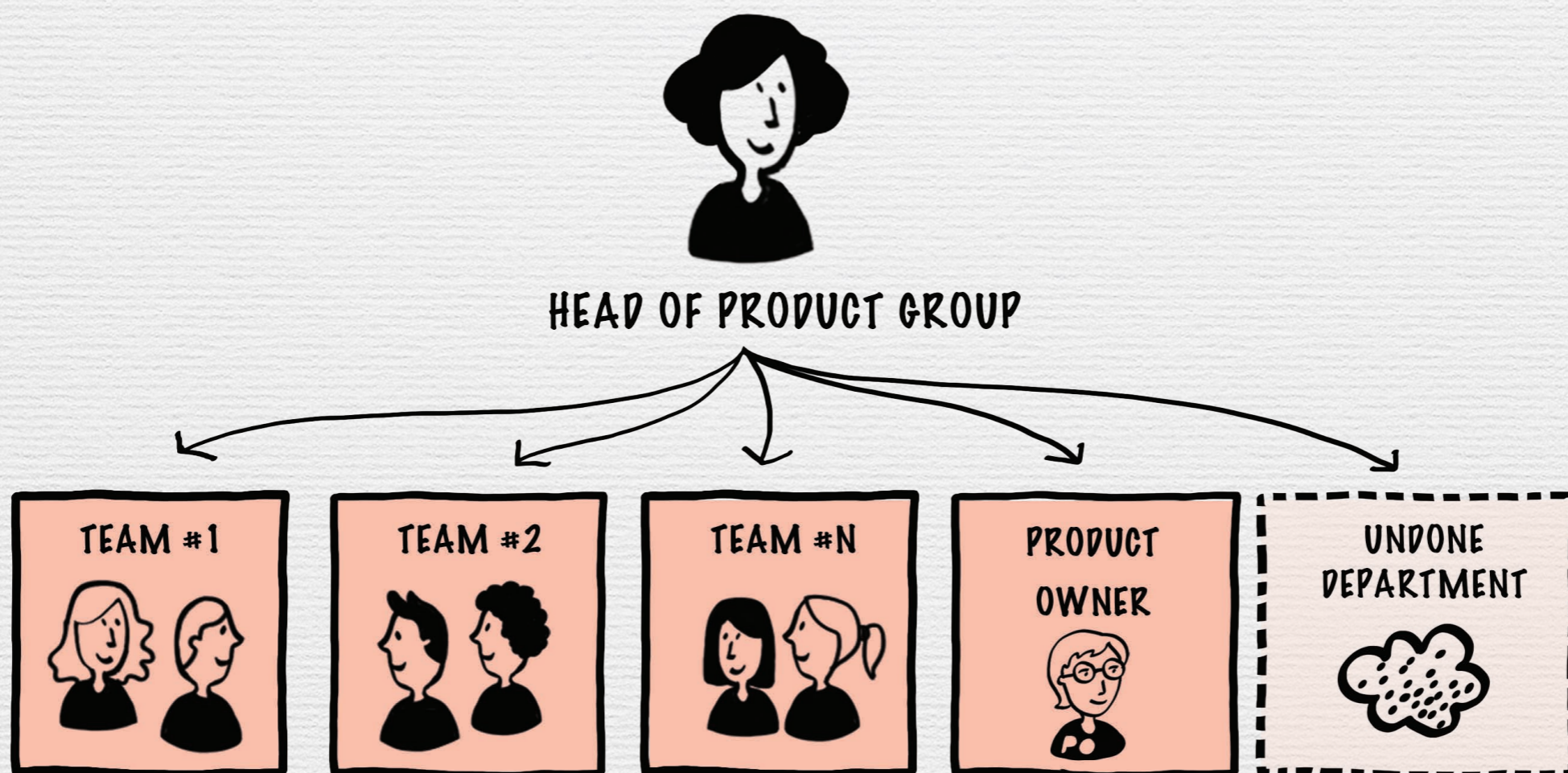
COMPONENT TEAMS



FEATURE TEAMS



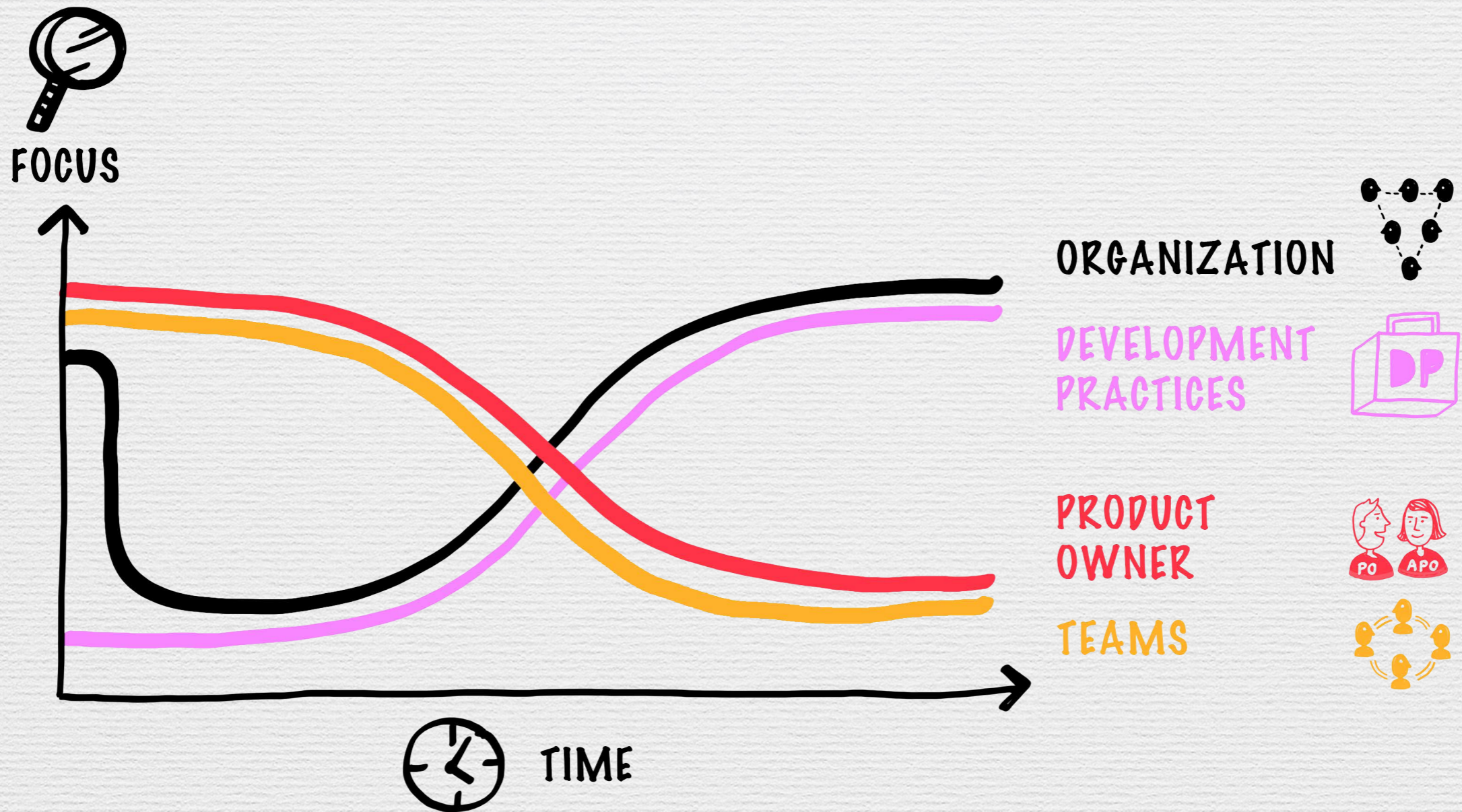
Organizační struktura v LeSS



Organizační struktura v LeSS

- **Head of project group** - Zodpovědný za celý projekt, řídí feature týmy i product ownery. Většinou se jmenuje v různých podnicích různě.
- **Feature týmy** - Vývoj, tvorba hodnoty pro zákazníka. Každý tým má scrum mastera, je cross-functional a self-managing.
- **Product Owner** - Většinou nazýváno Product Management, zodpovědný za podoby produktu, kooperuje s feature týmy.
- **Undone department** - Ideálně by nemělo existovat. Někdy týmy nejsou schopné doručit hodnotu (není splněno DoD), jsou to role, které pomáhají splnit dodatečné požadavky na kvalitu (QA, performance, business analyst group).

Scrum master v LeSS



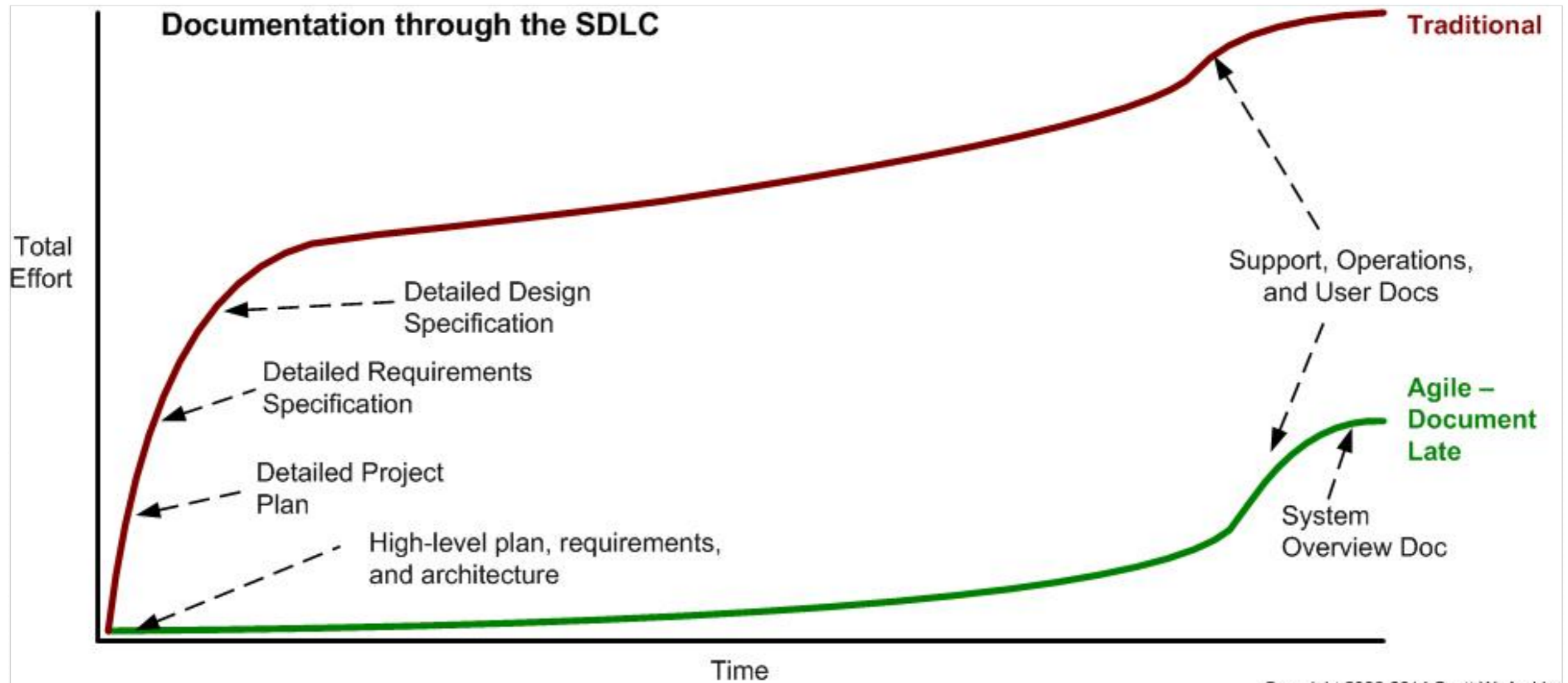
Technical excellence

- Specification by example
- Continuous integration
- Continuous delivery
- Test automation
- Architecture & Design
- Acceptance testing
- Clean code
- Thinking about testing
- TDD
- Unit testing

Mylné představy o Agile

- V agilních přístupech nejsou dokumenty
- Neexistuje návrh architektury
- Agilní jsou jen vývojáři, zbytek týmu pracuje stejně jako dříve
- Není důležitá podpora vedení firmy
- Agilní přístupy jsou konkrétní metodika

V agile nejsou dokumenty



V agile nejsou dokumenty

- S jakými dokumenty jste se setkali v předmětu ROPR?

Neexistuje návrh architektury

- Jaké druhy architektury znáte?
- Jak se vypořádáme s tím, že nemáme všechny požadavky detailně na začátku projektu?

Agilní jsou jen vývojáři

- Stačí ke spokojenosti zákazníka pouze tým agilních vývojářů?
- Co je to Wegnerova lema?

Není důležitá podpora vedení firmy

- Jaký je důsledek toho, že týmy pracují “skrytě” agilně bez podpory managementu?

Agilní přístupy jsou konkrétní metodika

- Jaká je definice metodiky?
- Kam byste zařadili tyto pojmy: Scrum, XP, FDD, RUP, OpenUP, Crystal Clear.

Klíčové agilní techniky

- TDD
- Refaktoring
- Párové programování
- Neustálá integrace (CI)
- Retrospektiva
- Planning poker
- Burn-down chart