



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**MASARYKŮV ÚSTAV VYŠŠÍCH STUDIÍ**



**DIPLOMOVÁ PRÁCE**

**Implementace rámce SAFe® do organizace**

**Implementation of the SAFe® Framework  
in the Organization**

**2023**

**Aneta Polenová**

**Studijní program:** Projektové řízení inovací

**Vedoucí práce:** Ing. Libor Cupal

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Polenová** Jméno: **Aneta** Osobní číslo: **484149**  
Fakulta/ústav: **Masarykův ústav vyšších studií**  
Zadávající katedra/ústav: **Institut ekonomických studií**  
Studijní program: **Projektové řízení inovací**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Implementace rámce SAFe do organizace**

Název diplomové práce anglicky:

**Implementation of the SAFe Framework in an Organization**

Pokyny pro vypracování:

Cíl: Cílem práce je stanovit vhodné postupy a způsob implementace rámce SAFe do řízení v organizace. Přínos: Přínosem práce pro organizaci je návrh jednotlivých kroků a způsobu implementace agilního přístupu řízení do organizace pomocí frameworku SAFe, tzv. Scaled Agile Framework. Osnova: Teoretická část – úvodní rešerše k agilnímu přístupu v projektovém řízení, jeho charakteristikám a jednotlivým metodologickým rámcům, kam můžeme mimo jiné zařadit Scrum, Kanban, a v neposlední řadě také rámec SAFe. Práce se poté bude podrobněji zabývat rámcem SAFe a popisovat jednotlivé kroky a způsob implementace tohoto rámce do řízení organizace. Praktická část – analýza výchozí situace v organizaci, využití zdrojů dobré praxe k návrhu nejlepšího možného řešení a způsobu implementace rámce SAFe do řízení v organizaci.

Seznam doporučené literatury:

ŠOCHOVÁ, Zuzana a Eduard KUNCE, 2019. Agilní metody řízení projektů. 2. vydání. Brno: Computer Press. ISBN 978-80-251-4961-4.  
RUBIN, Kenneth S., 2013. Essential Scrum. Upper Saddle River: Pearson Education. ISBN 978-0-13-704329-3.  
CAMPBELL-PRETTY, Em a Adrienne L. WILSON, 2019. The ART of Avoiding a Train Wreck: Practical Tips and Tricks for Launching and Operating SAFe Agile Release Trains. 1. Spojené státy americké: Pretty Agile. ISBN 978-1-094-75228-0.

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Libor Cupal Masarykův ústav vyšších studií ČVUT v Praze**

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **05.01.2023**

Termín odevzdání diplomové práce: **27.04.2023**

Platnost zadání diplomové práce: \_\_\_\_\_

Ing. Libor Cupal  
podpis vedoucí(ho) práce

Mgr. František Hřebík, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

prof. PhDr. Vladimíra Dvořáková, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomantka bere na vědomí, že je povinna vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_ Datum převzetí zadání

\_\_\_\_\_ Podpis studentky

POLENOVÁ, ANETA. *Implementace rámce SAFe® do organizace*. Praha: ČVUT 2023.  
Diplomová práce. České vysoké učení technické v Praze, Masarykův ústav vyšších studií.



**MASARYKŮV ÚSTAV  
VYŠŠÍCH STUDIÍ  
ČVUT V PRAZE**

## **Prohlášení**

Prohlašuji, že jsem svou diplomovou práci vypracovala samostatně. Dále prohlašuji, že jsem všechny použité zdroje správně a úplně citovala a uvádím je v příloženém seznamu použité literatury.

Nemám závažný důvod proti zpřístupnění této závěrečné práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Praze dne: 27.04.2023

Podpis:

## **Poděkování**

Tímto bych ráda poděkovala vedoucímu diplomové práce panu inženýrovi Liborovi Cupalovi za ochotu a trpělivost, kterou se mnou během vypracovávání práce měl. Chtěla bych mu také poděkovat za veškerá doporučení, ať už se týkala literatury, struktury práce či praktické části. Také bych chtěla poděkovat společnosti, která mi ochotně poskytla prostředky, postupy a metody, na základě kterých jsem zpracovávala praktickou část této práce. V neposlední řadě bych chtěla poděkovat své rodině za nikdy nekončící podporu během pěti let mého studia.

## **Abstrakt**

Diplomová práce se zabývá tématem implementace znalostního rámce Scaled Agile Framework® do organizace. Cílem této práce je zkoumání teorie agilního přístupu a jak je používán ve vybrané organizaci. Teoretická část je zaměřena na uvedení do agilního přístupu jako takového, popisuje Agilní Manifesto, které je dodnes základním kamenem celé agility. Dále popisuje vybrané metodické rámce, jako například eXtreme Programming, Scrum a Kanban. V neposlední řadě se práce dopodrobna zabývá právě znalostním rámcem SAFe®. Praktická část práce se pak zaměřuje na assessment (vyhodnocení), jak vybraná organizace tento rámec ve svém fungování používá a na základě zjištěných poznatků navrhuje vhodná doporučení na zlepšení. Sběr potřebných dat probíhal třemi způsoby: návštěvy a náslechy SAFe® ceremonií na úrovni Agile Release Trainu (ARTu), studiem dokumentace a osobními rozhovory s lidmi zastupující vybrané role. Assessment nám pomohl odhalit nedostatky při konání ceremonie PI Planning. Důležití členové ARTu se této ceremonie neúčastní, ceremonie se koná ve většině případech on-line, příprava na tuto ceremonii není dostatečná a mnoho dalších.

## **Klíčová slova**

Agilní řízení, SAFe®, Scrum, Kanban, agilita, škálovaná agilita, Spotify Model, assessment

## **Abstract**

This thesis deals with the topic of implementing the Scaled Agile Framework® into an organisation. The aim of this thesis is to explore the theory of the Agile approach, its history and how it is used in a selected organisation. The theoretical part focuses on an introduction to the agile approach itself, describing the Agile Manifesto, which is still the foundation of all agile. It also describes selected methodological frameworks such as eXtreme Programming, Scrum and Kanban. Last but not least, this thesis looks in detail at the SAFe® framework. The practical part focuses on the assessment of how the selected organisation uses this framework in its activities and on the basis of assessment findings propose appropriate recommendations for improvement. The collection of the necessary data was done by using three methods: by visiting and listening to SAFe® ceremonies at Agile Release Train (ART) level, by studying documentation and by personal interviews with people representing the selected roles. The assessment helped us to identify gaps in the PI Planning ceremony. Important ART members do not attend this ceremony, the ceremony is held online in most cases, there is not enough preparation for this ceremony, and many others.

## **Keywords**

Agile, SAFe®, Scrum, Kanban, agility, scaled agile, Spotify Model, assessment

# Obsah

Úvod .....	10
<b>1 Agilní řízení.....</b>	<b>13</b>
<b>1.1 Vzestup agilního přístupu</b>	<b>15</b>
1.1.1 Agilní manifest	16
<b>1.2 Agilní principy a praktiky</b>	<b>18</b>
1.2.1 Univerzální principy agilního řízení	18
<b>1.3 Agilní vs. tradiční přístup projektového řízení</b>	<b>20</b>
1.3.1 Tradiční přístup – Vodopád (Waterfall)	21
1.3.2 Rozdíly jednotlivých přístupů	22
<b>2 Některé agilní metodiky, rámce a postupy .....</b>	<b>23</b>
<b>2.1 Extrémní programování – Extreme Programming (XP)</b>	<b>23</b>
2.1.1 Hodnoty Extrémního Programování	24
2.1.2 Základní postupy extrémního programování	25
2.1.3 Základní činnosti	26
2.1.4 Role uvnitř týmu	26
2.1.5 Výhody a nevýhody extrémního programování	27
2.1.6 Vhodnost	28
<b>2.2 Scrum</b>	<b>28</b>
2.2.1 Tři základní pilíře	29
2.2.2 Pět hodnot Scrumu	30
2.2.3 Scrum role	30
2.2.4 Artefakty Scrumu	32
2.2.5 Scrum ceremonie	34
<b>2.3 Kanban</b>	<b>35</b>
2.3.1 Základní principy Kanbanu	36
2.3.2 Základní hodnoty Kanbanu	37
2.3.3 Metriky Kanbanu	38
<b>3 Škálovaná agilita – Scaled Agile .....</b>	<b>39</b>
<b>3.1 Spotify Model</b>	<b>40</b>
3.1.1 Squads	41
3.1.2 Tribes	42
3.1.3 Chapters	43
3.1.4 Guilds	43
3.1.5 Proč Spotify Model nefunguje?	44



<b>3.2 SAFe® - Scaled Agile Framework</b>	<b>45</b>
3.2.1 Proč implementovat SAFe?	46
3.2.2 Big Picture	46
3.2.3 Role v prostředí SAFe®	51
3.2.4 Agile Release Train – ART	54
3.2.5 Product Management – Produktový management	59
3.2.6 SAFe® Ceremonie	62
<b>4 Assessment používání rámce SAFe® ve vybrané organizaci .....</b>	<b>68</b>
<b>4.1 Cíl praktické části</b>	<b>68</b>
<b>4.2 Výzkumné metody</b>	<b>68</b>
4.2.1 Návštěva ceremonií na úrovni ARTu	69
4.2.2 Osobní rozhovory s členy ARTu	69
4.2.3 Studium potřebné dokumentace	69
<b>4.3 Jak vypadá Assessment?</b>	<b>70</b>
4.3.1 Assessor Sheet – List hodnotitele	73
4.3.2 Závěrečná zpráva	75
<b>4.4 ART Assessment</b>	<b>75</b>
4.4.1 Popis organizace	75
4.4.2 Harmonogram assessmentu	75
4.4.3 Výstupy assessmentu a navrhovaná doporučení	77
4.4.4 Poznatky a navrhovaná doporučení zapsané v listu hodnotitele	98
4.4.5 Závěrečná zpráva	99
<b>Závěr.....</b>	<b>101</b>
<b>Přílohy .....</b>	<b>102</b>
<b>Seznam použité literatury .....</b>	<b>107</b>
<b>Seznam obrázků.....</b>	<b>111</b>

# Úvod

Co si představíte, když se řekne, že je něco „agilní“? Interaktivní, rychlý, dynamický, rychle reagující na změnu, a kdybychom se snažili, vymysleli bychom spoustu dalších synonym. Být agilní neznamena řídit se pouze nastavenými metodami a postupy, ale také žít danou filozofií (Šochová, 2019).

Agilita je v současnosti stále více skloňovaným tématem a není divu. Podnikové prostředí prochází neustálými změnami a organizace se musí rychle přizpůsobovat tak, aby přežily. Organizační agilita, neboli schopnost průběžně měnit, upravovat a přizpůsobovat strategické směřování je stále častěji považována za zásadní faktor úspěchu podnikání (Holbeche, 2018). Podle průzkumu z roku 2018 provedeného webovou stránkou *Stack Overflow* používá agilní techniky při své práci až 85 % softwarových vývojářů (Rigby, 2020).

Současný, neustále se měnící vývoj software a digitální transformace ohrožuje každým dnem samotnou existenci mnoha organizací po celém světě. Tradiční organizační struktury a metody řízení, které v minulosti přinášely organizacím velký úspěch už dnes nedokáží držet krok se stále rozvíjející se digitalizací. Organizace jsou tak nuceny aplikovat a implementovat nové styly řízení, nové metody a postupy, které jim v dosahování dobrých výsledků pomohou (Knaster, 2020).

Cílem této diplomové práce je zjistit, jakým způsobem vybraná organizace používá agilní rámec SAF® (Scaled Agile Framework®) a následně na základě zjištěných skutečností navrhnout taková doporučení, která organizaci pomohou ke správnému používání nastavených metod a postupů tohoto rámce.

Teoretická část práce se zaměřuje na představení agilního přístupu k projektovému řízení. Popisuje Agilní Manifesto, která se stalo základním kamenem dalších agilních metodik, postupů a rámců. V první kapitole také naleznete porovnání tradičního a agilního přístupu a také podrobně popisuje tradiční vodopádový přístup. Druhá kapitole se úzce zaměřuje na některé agilní metodiky. Pro účely práce jsou popsány metodiky a rámce Kanban, Scrum a eXtreme Programming (XP). Poslední kapitola se zabývá škálovanou agilitou. Popisuje stále hodně skloňovaný Spotify Model, jeho fungování a proč tento model nemusí pro ostatní organizace fungovat. V neposlední řadě představuje

teoretická část práce metodický rámec Scaled Agile Framework® (SAFe®), jeho principy a hodnoty. Tato část také podrobně popisuje Agile Release Train zkr. ART, role které se v ARTu nacházejí a ceremonie, které se v průběhu fungování uskutečňují.

Praktická část diplomové práce se zabývá vyhodnocením (assessmentem) fungování Agile Release Trainu (vlaků) jako celku ve vybrané organizaci. Zaměřuje se na fungování a správnost nastavení SAFe® ceremonií, mezi které patří: PO Sync, Scrum of Scrums, Program Increment (PI) Planning, System Demo a Inspect & Adapt. Dalším zaměřením assessmentu je nastavení a popis SAFe® rolí Scrum Master, Product Owner, Business Owner a System Architect. V neposlední řadě se assessment v praktické části zabývá vyhodnocením fungování Product Managementu – PdM. Sběr dat potřebných k provedení vyhodnocení probíhal hned třemi způsoby: návštěvou a naslechem na výše zmíněných ceremoniích, studiem dokumentace, která popisuje ART role, nastavení ceremonií a zápisy z nich, materiály používané při PI Planningu a poté následovaly rozhovory (interviews) s vybranými členy zastupujícími vybrané role uvnitř ARTu.

Praktická část diplomové práce je založena na zkušenostech a výpomoci na uskutečněném Assessmentu používání rámce SAFe® v rámci zaměstnání v poradenské společnosti zaměřující se na projektové řízení a agilitu.

**V příloze** této práce jsou popsány některé agilní a SAFe® pojmy, které se v práci objevují.

# **TEORETICKÁ ČÁST**

# 1 Agilní řízení

Agilita a agilní metody se v posledních letech staly ve světě velmi populárními a čím dál tím rychleji se rozšiřují do dalších odvětví. Můžeme je najít v telekomunikacích, bankách, aerolinkách, univerzitách, automobilovém průmyslu nebo také v armádě (Šochová, 2019). Když v současné době navštívíte téměř jakoukoliv velkou organizaci, narazíte zde na desítky agilních týmů, které pracují na zlepšení zákaznických zkušeností a obchodních procesů. Právě velké společnosti jako John Deere, Bosh a digitální giganti jako Spotify, Amazon a Netflix začlenily agilní metody mezi celou řadu svých inovačních aktivit (Rigby, 2020). Do povědomí společností se tyto metody v České republice dostávají až v posledních letech, ale využívání agilních metodik, používání Scrumu či Kanbanu dnes již není výjimkou a tak se tyto metody a postupy stávají na českém trhu běžnou součástí podnikového řízení (Šochová, 2019).

Důvodem rychlého rozšiřování agilních metod do řízení společností je skutečnost, že agilní přístup osvobozuje inovační myšlení, které je utlumováno byrokratickými strukturami a postupy. Pomáhá tak firmám přetvořit to, co nabízejí svým zákazníkům, ale i to, jak fungují jejich procesy uvnitř. Tyto metody pomáhají proměňovat pracovní prostředí a činí práci lidí užitečnější (Rigby, 2020). Agilní přístup jako takový se stává čím dál tím populárnějším, a to díky výsledkům týmů, které tento přístup adoptovaly: vytvářejí lepší software, lépe mezi sebou spolupracují, uspokojují své uživatele a zákazníky a to vše při mnohem uvolněnějším a příjemnějším pracovním prostředí (Stellman, 2015).

Zde jsou jen některá z řešení, která agilní přístup slibuje (Stellman, 2015).

- Projekty řízené agilním přístupem se realizují včas, což je dobré pro týmy, které se potýkaly s velkým zpožděním projektu či překročením stanoveného rozpočtu.
- Agilní projekty přinášejí software vysoké kvality a odstraňuje tak chyby či vývoj neefektivního softwaru.
- Kód vytvořený agilními týmy je dobře konstruovaný.
- Agilní týmy dělají své uživatele či zákazníky šťastnými a přinášejí jim větší hodnotu.

Využití agilního přístupu zvyšuje efektivitu týmu jako celku a zároveň zvyšuje spokojenost zaměstnanců jako jednotlivců. Minimalizuje plýtvání, které je spojeno se zbytečnými schůzkami, opakujícím se plánováním a nadměrnou a zbytečnou dokumentací. Pomocí agility zvyšujeme zapojení zákazníků do celého procesu a tím zajišťujeme jejich spokojenost. Díky zapojení lidí z různých oborů rozšiřujeme zkušenosti a budujeme vzájemnou důvěru a respekt (Rigby, 2020).

Agilní přístup se vyvinul v soubor principů a postupů, které se rozvíjely v rámci IT odvětví a postupně pak nacházeli uplatnění i mimo něj. Jeho adaptivní, na hodnotu orientovaná a kolaborativní podstata pohání inovace iterativním a inkrementálním způsobem, který je založen na učení se ze zkušeností. Postupem času se z agility vyvinulo mnoho metodik, které odrážejí různé aspekty agilního přístupu od inženýrství, product development a projektového managementu, které jsou neustále ovlivňovány nikdy nekončícím vývojem. Hlavním aspektem agilního přístupu je zpětná vazba a otevřená komunikace mezi všemi zúčastněnými stranami podílejících se na projektu (Moran, 2015). Vztahuje se na specifické metody, jako je iterativní způsob řízení projektů, nástroje Lean managementu (štíhlé výroby) a také přístupy neustálého zlepšování. Tyto metody se uplatňují při vývoji mnoha druhů systémů, včetně webových a mobilních aplikací, ale také při vývoji systémů uvnitř podniku (Holbeche, 2018).

Jak už tedy víme, agilita jako taková je soubor metod a metodik, které pomáhají týmům v organizaci přemýšlet více efektivněji a tím přijímat lepší rozhodnutí. Tyto metodiky a metody se týkají všech oblastí softwarového vývoje, včetně zlepšování jednotlivých procesů, řízení projektů či návrhu a architektury softwaru. Každá z těchto metod se skládá z jednotlivých postupů, které jsou upraveny a optimalizovány tak, aby jejich osvojení a následné používání bylo pro agilní týmy co nejjednodušší.

Agilita ale nejsou pouze metody a rámce, ale také způsob myšlení neboli „**mindset**“. Správný mindset a nastavení myšlení týmu má důležitý vliv na to, jak efektivně tým jednotlivé metody a postupy používá. Toto myšlení pomáhá lidem v týmu sdílet mezi sebou informace, na základě kterých může tým provádět důležitá projektová rozhodnutí místo toho, aby o všech krocích rozhodoval projektový manažer sám. Realita implementace agilního přístupu se u mnoha týmů, které nedosáhly takových úspěchů značně liší od toho, jaké přínosy tento přístup slibuje. Důvodem je často právě zmiňovaný mindset, který tým do každého projektu přináší. Ale co ve skutečnosti znamená „mindset

shift“, neboli změna myšlení? Co má způsob myšlení společného s každodenním plánováním, navrhováním a dodáváním softwaru? Jak se ukázalo, praktiky, které tým používá ke každodenní práci, hodně závisí na přístupu, které k nim každý z členů týmu ale i tým jako celek má (Stellman, 2015).

Agilní týmy jsou malé, multi-funkční, plně specializované a dedikované pracovní skupiny, které jsou zaměřené na vytváření inovativních vylepšení zákaznických produktů a služeb, podnikových procesů a technologií, které tyto procesy umožňují. Každý individuální tým má svého majitele (ownera), který je zodpovědný za dodávání hodnoty zákazníkům či koncovým uživatelům a kouče, který týmu pomáhá se neustále zlepšovat a zajišťuje jeho spokojenost. Členové týmu pak rozdělují složité problémy a požadavky do menších komponentů a poté začínají vyvíjet pracovní verze potenciálních řešení v krátkých cyklech, známých jako sprinty (obvykle je tento cyklus dlouhý 2-3 týdny). Týmem vytvořené řešení je poté testováno společně se zákazníkem či koncovým uživatelem, který na začátku procesu zadal týmu potřebné požadavky, podle kterých se tým při vývoji řídí. Pokud je zákazník spokojený, může být vytvořené řešení okamžitě dodáno, i přesto, že má „owner“ týmu určité výhrady. Celý proces je založený hlavně na transparentnosti vůči všem zúčastněným stranám (Rigby, 2020)

Díky nikdy nekončícímu vývoji a rozvoji se kolem agilního přístupu vyvinulo mnoho specifických metodik (jako například SAFe®, SCRUM, Extreme Programming – XP nebo Kanban) a každá z těchto metodik má své vášnivé podporovatele. Ale stejně jako neexistuje jednotný plán transformace organizace, tak ani neexistuje jedna metodika, která by byla nejlepší pro všechny. Místo toho je pro organizaci mnohem důležitější najít si svou vlastní cestu implementace agilního přístupu a rozvíjet způsoby práce, které jsou vhodné pro její jedinečnou strukturu a kontext (Perkin, 2020)

## **1.1 Vzestup agilního přístupu**

Když lidé slyší pojem agilita, přemýšlí o něm jako o přístupu, který se začal používat teprve nedávno. Ve skutečnosti se však tento pojem začal používat již před 30 lety, kdy už tehdy začalo být nezbytné držet krok s rychlým tempem vývoje podnikového softwaru a s tím spojeného hledání lepšího způsobu, jak organizaci řídit a těmto změnám se co nejlépe přizpůsobit. Organizace postupem času zjišťovaly, že tradiční metodiky projektového řízení, které se spíše zaměřovaly na dlouhodobé plánování, nebyly tak

efektivní pro typy projektů, na kterých pracovaly. Bylo tak pro ně nutností najít lepší způsoby řízení (Hunt, 2018). Rámce pro adaptivní vývoj softwaru, kam spadá i agilní přístup tedy existují již dlouho a projevují se v mnoha podobách. Základem těchto rámců jsou však dvě věci: tvorba hypotéz a spolupráce napříč expertními obory na vývoji jednotlivých systémů (McGregor, 2020). Koncept agilních principů, postupů a metod vznikl jako reakce na potřeby moderních podniků fungovat předvídatelným způsobem i v situacích, kdy je vývoj softwaru extrémně složitý (Holbeche, 2018). Za vznikem agilního hnutí také stojí problémy, které s sebou přinášelo používání tradičních a rigidních procesů (Perkin, 2020).

### **1.1.1 Agilní manifest**

Na základě toho proběhlo v roce 2001 ve státě Utah setkání tehdejších iniciátorů a propagátorů agilních metod a přístupů, kde se diskutovalo o změně přístupu k softwarovému vývoji. Výsledkem těchto diskusí byl tzv. Agilní Manifest (Doležal, 2022).

Agilní manifest pak vychází ze dvou základních tezí (Kadlec, 2004):

1. Je mnohem efektivnější změny přijímat a umožňovat, než se pokoušet jim zabránit.
2. Je důležité být připraven reagovat na jakoukoliv změnu či nepředvídatelnou událost, která s vysokou pravděpodobností přijde.

**Agilní Manifest říká** (Manifesto for Agile Software Development, 2001):

„Objevujeme lepší způsoby vývoje softwaru 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*

Je známo, že týmy s dobrou spoluprací jsou více efektivní a jejich výsledky jsou na jiné úrovni než výsledky týmu, který spolu nekomunikuje a nespolupracuje. Vzájemná komunikace tak převažuje nad správně definovanými procesy a dobře používanými nástroji. Neznamená to však, že by týmy neměly procesy a nástroje pro svou práci používat (Šochová, 2019).



### **Fungující software** před *vyčerpávající* dokumentací

Dokumentace je a vždy bude důležitá, neměla by ale převážet nad vlastnostmi samotného produktu. Snažte se dokumentaci primárně používat jako referenci pro oblasti, které jsou pro zákazníka těžko pochopitelné (Šochová, 2019).

### **Spolupráce se zákazníkem** před *vyjednáváním o smlouvě*

U zákazníka, který je spokojený s výsledným produktem přesně podle jeho požadavků je větší pravděpodobnost toho, že vaši organizaci doporučí ostatním firmám. Důležitost smluv zde však není zpochybňován, neměly by ale nahrazovat spolupráci a komunikaci se zákazníkem (Šochová, 2019).

### **Reagování na změny** před *dodržováním plánu*

Technologie se mění každým dnem a s nimi i požadavky a problémy zákazníka. Jako všichni ostatní, i zákazníci se musí přizpůsobovat měnícím se trendům na trhu. Jako jejich dodavatel nemůžeme zákazníka ve změnách brzdit. Proto je důležité se naučit reagovat na změny požadavků a odstoupit od původního domluveného plánu (Šochová, 2019).

Společně se vytvořenými hodnotami, vytvořila tato skupina také *Dvanáct principů agilního vývoje softwaru*, které zní takto (Manifesto for Agile Software Development, 2001):

1. Naší nejvyšší prioritou je vyhovět zákazníkovi časným i průběžným dodáváním hodnotného softwaru.
2. Víťáme změny v požadavcích, a to i v pozdějších fázích vývoje.
3. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.
4. Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.
5. Lidé z businessu a vývoje musí spolupracovat denně po celou dobu projektu.
6. Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.
7. Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.
8. Hlavním měřítkem pokroku je fungující software.
9. Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.

10. Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.
11. Jednoduchost – umění maximalizovat množství nevykonané práce – je klíčová.
12. Nejlepší architektury, požadavky a návrhy vzejdou ze samo-organizujících se týmů.

## **1.2 Agilní principy a praktiky**

Pod pojmem agilita si představíme mnoho věcí, časté iterování, uvolnění vývoje softwaru, aby mohl dělat to, co umí nejlépe, sladění businessu se softwarem, snaha vytvářet věci, která naši zákazníci opravdu chtějí, rozvoj kultury neustálého zlepšování, ale primárně jde především o přizpůsobování se změnám a reakce na ně (McKenna, 2016). Jak již víme, agilní přístup k vývoji softwaru způsobil revoluci ve způsobu, jakým technologické týmy pracují a vytvářejí software. Chytřejší organizace si díky tomu uvědomily, že má agilita významnou hodnotu i mimo zmíněné technologické týmy, protože podporuje pružnější fungování napříč celou organizací (Perkin, 2020).

### **1.2.1 Univerzální principy agilního řízení**

Principů, které v rámci agilního přístupu k softwarovému vývoji nalezneme je nespočetně mnoho, mezi ty základní si však můžeme zařadit tyto:

#### **Tvorba hodnoty pro zákazníka a zákazník samotný**

Hodnota je při vývoji to nejdůležitější. Pro zákazníka či koncového uživatele znamená hodnota určitý užitek a naplnění jeho potřeb. Pokud výstup nepřináší koncovému uživateli užitek, tak pro něj nemá žádnou hodnotu. Vše, co žádným způsobem nepřispívá k dodávání hodnoty zákazníkovi, je plýtvání, které je potřeba co nejvíce minimalizovat. Hodnotu může přinášet i prototyp nebo část dodaného produktu, služby či funkcionality (Doležal, 2022).

#### **Transparence a zpětná vazba**

Tím, že jsou hodnoty koncovým uživatelům dodávány průběžně, získává tým jedinečnou zpětnou vazbu, zda postupuje správným směrem dle stanovených požadavků. Získáváním zpětné vazby se snižuje riziko zbytečné práce, dalších oprav a předělávek,

nenaplnění zákaznicka očekávání a požadavků či nesplnění termínu dodání. Transparentnost pak zvyšuje porozumění mezi vývojářským týmem a jejich zákazníkem. Bez dostatečného porozumění není tým schopný dodávat koncovým uživatelům očekávané hodnoty (Doležal, 2022).

### **Proměnlivost a změny**

Změna sama o sobě je velice vítaným jevem a při vývoji či tvorbě produktu by měla být vždy akceptována. Změna s sebou nese příležitost dodávat zákazníkům větší hodnotu. Pokud se naskytne možnost cokoliv změnit, nečeká tým na povolení, ale usiluje o ni sám (Doležal, 2022). Vzhledem k těmto častým změnám je důležité průběžně kontrolovat a ověřovat správnost dodaného řešení. Testy jsou řešením jak toho dosáhnout. Testy by měly být automatizované a provedené ještě před samotnou implementací dodaných řešení (Kadlec, 2004).

### **Spolupráce a osobní komunikace**

Díky spolupráci (nejen se zákazníky, ale se všemi zainteresovanými stranami), nejlépe na každodenní bázi, dosahuje tým nejlepších výsledků. Osobní komunikace se zákazníkem by měla být vždy na prvním místě (Doležal, 2022). V ideálním případě by měl být zákazník součástí vývojového týmu a měl by se spolupodílet na návrhu řešení (Kadlec, 2004)

### **Malé multidisciplinární týmy dosahují větší efektivity**

Tvoří se malé agilní týmy, aby se mohly rychle přizpůsobovat změnám. Zahrnují všechny potřebné kompetence, které jsou nutné k dosažení požadovaných výstupů. Týmy jsou často samo-organizované a vyznačují se vysokou úrovní autonomie (Perkin, 2020).

### **Práce je rozdělena do krátkých časových úseků, sprintů**

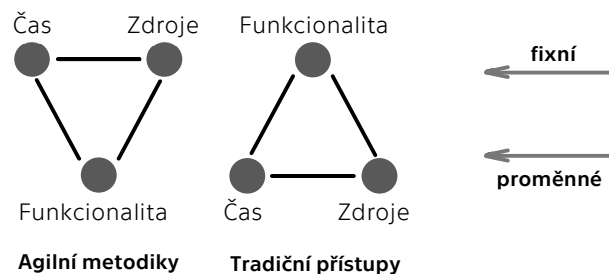
Tým obvykle pracuje iterativně a v krátkých časových intervalech neboli sprintech (obvykle je sprint dlouhý 2-3 týdny). Tyto sprinty probíhají podle stejného schématu. Začínají stanovením priorit jednotlivých položek a plánováním sprintu, jehož cílem je určit, které položky budou v daném sprintu zpracovány. Na konci sprintu probíhá hodnocení, které zachycuje, čeho bylo v průběhu sprintu dosaženo (Perkin, 2020).

## Retrospektiva

Správná a dobře provedená retrospektiva je příležitostí se zaměřit na zlepšení způsobu, jak celý proces funguje. Typicky se retrospektiva zaměřuje na to, co přestat dělat, co naopak začít dělat a v čem pokračovat. Retrospektiva je užitečným nástrojem, na jehož základě se do procesu práce týmů zakotvuje neustálé učení a zlepšování (Perkin, 2020).

### 1.3 Agilní vs. tradiční přístup projektového řízení

Rozdíl mezi agilními metodikami a tradičními přístupy projektového řízení lze nejlépe pozorovat na obrázku, který tyto dva přístupy mezi sebou srovnává (Kadlec, 2004):



OBR. 1: ROZDÍL MEZI AGILITOU A TRADIČNÍMI PŘÍSTUPY (ZDROJ: KADLEC, 2004)

Tradiční přístupy projektového řízení primárně vycházejí z důležitosti a nutnosti naplnění dokumentace požadavků (funkcionalit). Funkcionalita je fixní, zatímco čas a zdroje jsou proměnné. Požadavky koncového uživatele jsou stanoveny a zaznamenány na začátku vývojového procesu. Cílem je pak tyto požadavky splnit. Zdroje a čas jsou proměnné z toho důvodu, že se na počátku procesu těžce odhaduje jak dlouho bude projekt trvat a kolik bude stát. U agilních metodik je toto naopak, funkcionalita je proměnná a časem se mění a přizpůsobuje potřebám zákazníka, kdežto čas a zdroje jsou fixní. Na počátku procesu je stanoven nejdelší možný čas vývoje a nejvyšší možné náklady (Kadlec, 2004). Bez ohledu na to, jakou metodiku si organizace zvolí, je v agilním i vodopádovém přístupu hlavním cílem vytvořit výsledek, který splňuje požadavky a specifikace zákazníka (Hunt, 2018).

### **1.3.1 Tradiční přístup – Vodopád (Waterfall)**

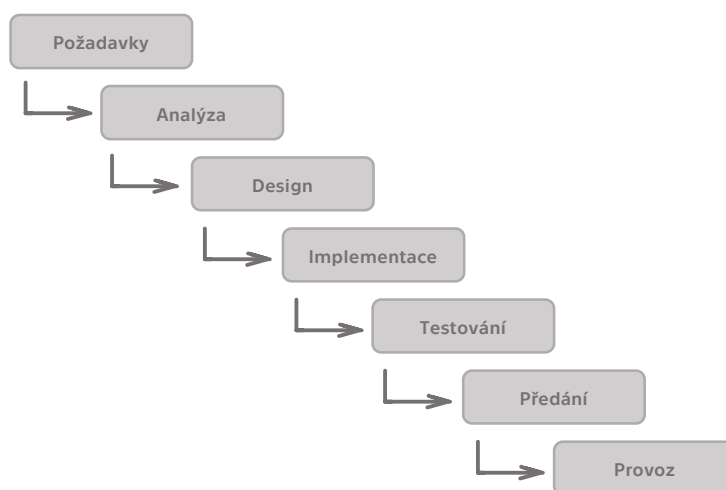
Waterfall model vznikl ve stavebním a výrobním průmyslu, ve vysoce strukturovaném prostředí, kde každá změna vedla k neúnosnému navýšení nákladů na celkový projekt. Jedná se tedy o metodiku, která je navržena tak, aby eliminovala změny a potřebu těmto změnám se přizpůsobovat (Perkin, 2020). Tradiční řízení projektů je založena na detailním a důkladném plánování. Od projektových týmů se pak očekává, že budou stanovený plán dodržovat. Čas, náklady a rozsah práce jsou pečlivě sledovány a kontrolovány. Pokud se tým od plánu odchýlí, musí vedoucí projektu, v tomto případě projektový manažer, jednat a změnu řešit (Fisher, 2021).

Obecně platí, že tradiční přístup projektového řízení klade důraz na to, aby bylo vše důkladně a správně zdokumentováno, od komunikace se zákazníkem až po předávání výsledného produktu či systému. Na projektu se začíná pracovat až ve chvíli, kdy je k dispozici kompletní design toho, jak by mělo výsledné řešení vypadat. Implementace jakékoliv změny je pak často velmi časově náročná, často může trvat až několik týdnů či měsíců, než je tato změna schválena (Myslín, 2016). Waterfall model nejlépe funguje ve stabilním a pomalu pohybujícím se prostředí, kde je potřeba mít pevně stanovené plány, dobře strukturované procesy a do značné míry neměnné podmínky (Perkin, 2020).

Tradičně společnosti využívají při řízení softwarových projektů vodopádový model (waterfall), při kterém se nejprve definují požadavky, naplánuje se celý projekt, vytvoří se design softwaru a poté se výsledný produkt otestuje (Stellman, 2015). Tento model patří mezi nejstarší metodiky řízení projektu a vývoje softwaru, ale pro většinu dnešních projektů je tento model již nedostatečný. I přes svou nedostatečnost je tento model stále velice využíván. Waterfall je založen na jednoduchém postupu – vývoj softwaru se skládá z několika fází, které musí jít postupně za sebou. Jedna fáze projektu skončí a další fáze začíná. Tento proces se vyznačuje jednosměrností, nikdy se nevrací do předchozích fází (Myslín, 2016).

Tuto metodiku tak můžeme charakterizovat v několika bodech (Myslín, 2016):

- **Jednoznačnost** – vždy víme, v jaké části projektu se nacházíme a jaká část následuje.
- **Úplnost zadání** – do další fáze projektu vstupuje jen tehdy, když jsou všechny výstupy předchozí fáze dokončeny.
- **Lineární průběh** – jednotlivé fáze na sebe navazují a nikdy se nevracíme zpět.



**OBR. 2:** WATERFALL MODEL (ZDROJ: MYSLÍN, 2016)

### 1.3.2 Rozdíly jednotlivých přístupů

*International Journal of Managing Projects in Business* rozlišuje mezi těmito přístupy následujícím způsobem (Azanha, 2017):

Přístup	Tradiční	Agilní
Cíle projektu	Zaměření na dokončení projektu v souladu s časem, náklady a požadavky na kvalitu	Zaměření na výsledky a dosahování více kritérií úspěchu
Projektový plán	Soubor činností, které jsou prováděny na podle plánu a splňují trojí omezení (čas, náklady a kvalita)	Organizace a proces k dosažení očekávaných cílů a výsledků pro business.
Plánování	Provádí se vždy na začátku projektu	Provádí se na začátku a kdykoliv během projektu
Manažerský přístup	Rigidní se zaměřením na původní plán	Flexibilní, adaptivní, variabilní
Práce / Realizace	Předvídatelná, měřitelná, lineární a jednoduchá	Nepředvídatelná, neměřitelná, nelineární a komplexní
Organizační vliv	Minimální, nestranný, od zahájení projektu	Projekt je ovlivňován po celou dobu realizace
Kontrola projektu	Identifikace odchylek od původního plánu, uzpůsobení práce tak, aby odpovídala plánu	Identifikace změn v prostředí a úprava plánu podle nich
Aplikace metodiky	Obecné a stejné používá ve všech projektech	Přizpůsobení procesu v závislosti na typu projektu
Styl řízení	Jeden model slouží všem typům projektů	Adaptivní přístup, jeden model neobsáhne všechny typy projektů

**TAB. 1:** ROZDÍLY MEZI TRADIČNÍM A AGILNÍM PŘÍSTUPEM (ZDROJ: AZANHA, 2017)

## 2 Některé agilní metodiky, rámce a postupy

V současné době existuje mnoho dobře zavedených agilních metodik a rámců. Od přístupů, které jsou zaměřené na vývoj produktů jako například eXtreme Programming (XP), Scrum a Kanban, až po složitější metodiky jako je například na architekturu orientovaný metodický rámec Scaled Agile Framework® (SAFe®). Každá z metodik má svou vlastní kulturu, jazyk a postupy, které zpracovává v různé míře podrobnosti (Moran, 2015).

### 2.1 Extrémní programování – Extreme Programming (XP)

*Extreme Programming*, neboli extrémní programování, je původní agilní metodikou, a to v tom smyslu, že její vytvoření a následně zavedení bylo událostí, která uvedla agilní myšlenky a přístupy do popředí zájmu v oblasti softwarového inženýrství. Dnes už je tato metodika méně používání, protože se většina pozornosti přesunula k metodice Scrum. Avšak její principy a postupy byly integrovány do dalších přístupů (Meyer, 2014). Této metodice položil základy známý expert na vývoj softwaru Kent Beck v roce 1999 (Kadlec, 2004). Jméno této metodiky je dovozeno od skutečnosti, že mnohé principy z jiných agilních metodik dovádí do úplného extrému. Správné nasazení a používání této metodiky ve správném projektu může výrazně pomoci k dosažení požadovaného výsledku (Myslín, 2016). Do extrému zavádí například psaní jednotlivých testů před implementací dílčích řešení za účelem zjištění okamžité zpětné vazby o pokroku v plnění požadavků. Výsledkem je pak provázaný soubor technik, které se vzájemně posilují. Extreme Programming je metodika založená na disciplíně, která vyžaduje dodržování stanovených standardů a silnou angažovanost v testování jednotek a integraci. Úplně se vyhýbá většině administrativním náležitostem (včetně reportingu) a zaměřuje se výhradně na postupy softwarového inženýrství (Moran, 2015).

Do extrému tato metodika zavádí také například délku iterace. V jiných agilních metodikách trvají jednotlivé iterace několik týdnů, ale v XP mají tyto iterace trvání i v řádu dní. V podstatě to znamená, že každý den můžeme iteraci ukončit a do softwarového prostředí implementovat novou funkcionalitu. V některých případech můžeme tyto nové funkcionality implementovat i několikrát denně. S testováním je to stejně. Testujeme na konci každé iterace, pokud tedy ukončujeme iterace každý den, musíme i každý den testovat. A o tom XP je, o neustálém testování (Myslín, 2016).

## **2.1.1 Hodnoty Extrémního Programování**

Extreme Programming staví na čtyřech základních hodnotách, kterými jsou: jednoduchost, zpětná vazba, komunikace a odvaha (Kadlec, 2004).

### **Jednoduchost**

Platí zde, že co není teď nutné, to se prostě neudělá / nevyvine. Eliminuje se tak zbytečná a nepotřebná dokumentace, formuláře, ale i potřeba vyvíjet kód, který by mohl být někdy v budoucnosti užitečný. Tým vyvíjí jen to, co je v danou chvíli potřeba (Myslín, 2016). Vývojový tým se tak musí naučit uvažovat o tom, co je pro koncového uživatele nejdůležitější zrovna teď a ne to, co by mohl využít až v budoucnosti. Do té doby se požadavky zákazníka ještě několikrát změní (Kadlec, 2004).

### **Zpětná vazba**

Zpětná vazba je velmi důležitým aspektem extrémního programování. Díky ní dostává tým informace o aktuálním stavu vyvíjeného softwaru, o průběhu vývoje, o požadavcích a potřebách uživatele a o všech ostatních záležitostech týkajících se vývoje. Nejdůležitější zpětnou vazbu pak poskytuje testování (Kadlec, 2004). Na základě zpětné vazby z testování se pak upravuje další postup. To samé platí také pro koncového uživatele. I on sám musí otestovat, zda nová funkcionality odpovídá jeho požadavkům a zda funguje (Myslín, 2016).

### **Komunikace**

Extrémní programování se velmi úzce zaměřuje na komunikační kanály a jejich fungování a případné spravování. V rámci této metodiky je definováno mnoho postupů, které bez komunikace nefungují. Na základě toho definuje XP speciálního člena týmu, který detekuje jakékoliv výpadky v komunikaci a pomáhá navazovat vztahy mezi lidmi (Kadlec, 2004).

### **Odvaha**

Odvaha je další důležitou součástí XP, jelikož bez dostatečné odvahy je použití této metodiky prakticky vyloučeno (Kadlec, 2004). Vývojový tým musí mít dostatečnou odvahu pustit se do řešení jakéhokoliv problému, a to co nejdříve. Odvahu tým potřebuje hlavně v případech, kdy je nucen zahodit týdenní práci na určité funkcionality (Myslín, 2016).



## 2.1.2 Základní postupy extrémního programování

Existuje 12 základních postupů EX, které pomáhají týmům projít nástrahami vývoje softwaru a zároveň pomáhají vyvíjet takový kód, který se dá v průběhu vývoje měnit. Mnohé postupy jsou specifické pro programování a jsou úzce zaměřeny na řešení problémů, které způsobují vytváření nekvalitního softwaru (Stellman, 2015).

Vyjmenujeme si šest nejdůležitějších postupů:

1. **Párové programování (Pair Programming)** – na vývoji kódu spolupracují dva vývojáři u jednoho pracovního stolu. Jeden vývojář píše, zatímco druhý ho sleduje a neustále diskutují o tom, co se děje. Tyto role si mezi sebou v průběhu vývoje střídají. Vytváří se tak méně chyb, jelikož se na jeden problém dívají dva vývojáři zároveň (Stellman, 2015).
2. **Plánovací hra** – cílem této hry je vytvořit plán a spočívá ve vytváření karet se zadáním. Na plánování se podílí každý člen týmu. Skládá se ze tří částí (průzkum, závazek a řízení) (Kadlec, 2004).
3. **Testování** – testy jsou základním stavebním kamenem extrémního programování. Platí zde, že pokud má tým cokoli naprogramovat či vytvořit, vymyslí si nejdříve vhodné testy (Myslín, 2016).
4. **Společné vlastnictví a zodpovědnost** – za všechna řešení a celý systém je odpovědný celý vývojový tým. Každý z členů může cokoli upravit či pozměnit (Myslín, 2016).
5. **Zákazník jako součást týmu** – po celou dobu vývoje se členem týmu stává také zákazník nebo koncový uživatel (Kadlec, 2004). Kdo jiný dokáže posoudit kvalitu výsledného řešení lépe než ten, kdo ho bude poté využívat (Myslín, 2016).
6. **Týmové standardy** – pokud chce tým eliminovat co nejvíce nepotřebné dokumentace, musí jiným způsobem zajistit, aby se vyznal ve zdrojových kódech. A z toho důvodu se tvoří určité standardy, které popisují, jak se pojmenovat proměnné a jednotlivé soubory, jak používat odsazení a mnoho dalších důležitých specifikací (Myslín, 2016).

Jednotlivé postupy mohou fungovat jen tehdy, když jsou dodržovány všechny zároveň. Tyto postupy vyžadují provázanost a vzájemně se podporují (Kadlec, 2004).

### 2.1.3 Základní činnosti

Při vývoji za použití metodiky Extreme Programming existují čtyři základní činnosti, podle kterých by se měly vývojové týmy řídit. Tyto činnosti jsou (Kadlec, 2004):

1. **Design** – design představuje jediný způsob, jak se při vývoji softwaru nedostat do slepé uličky, ze které není cesta zpět. Díky designu se vytváří struktura, která uchopuje a organizuje logiku v systému. Navrhování je každodenní náplní práce každého programátora v extrémním programování. Je to nutnost, ne možnost.
2. **Psaní zdrojového kódu** – tato činnosti začíná v okamžiku, kdy jsou napsány veškeré příslušné a potřebné testy.
3. **Poslouchání** – je velice nutné a důležité, aby vývojáři naslouchali jak svým zákazníkům, tak i svým kolegům. A to z důvodu, aby věděli, jaké řešení mají vůbec implementovat. Poslouchání tak jako komunikace, musí být dobře strukturované, ve výjimečných případech dokonce někdy až řízené.
4. **Testování** – není možné, aby proběhla integrace funkcionality bez toho, aniž by prošla automatizovaným testem. Automatizované testy se programují ještě před tím, než se jednotlivé funkcionality začnou vyvíjet. Tyto testy jsou automatické, izolované a probíhají průběžně.

### 2.1.4 Role uvnitř týmu

Metodika extrémního programování se velmi úzce zaměřuje na složení vývojového týmu, na práci s lidmi, na motivaci a mnoho dalších faktorů, které práci v týmu ovlivňují. Jednotlivé role uvnitř týmu poté definuje takto (Kadlec, 2004):

#### Programátor

Role programátora je základem a srdcem metodiky Extreme Programming. Kromě toho, že programátor navrhuje a vyvíjí softwaru a píše testy, je jeho úkolem komunikace s ostatními členy týmu (Kadlec, 2004). Každý programátor má jedinečné dovednosti, které mu mohou pomoci při řešení problémů (Chappell, 2020).

#### Zákazník

Zákazník je druhou klíčovou rolí této metodik. Programátor sice umí programovat a navrhovat software, ale neví jak by měl takový software vypadat. Zákazník na druhou stranu programovat nemusí, ale ví, co je potřeba programovat. Musí se však sžít

s některými zásady extrémního programování, jako například naučit se psát zadání a testy funkcionalit (Kadlec, 2004). Pokud však zákazník není k dispozici, může se této role ujmou zkušený programátor (Chappell, 2020).

### **Kouč**

Hlavní zodpovědnost této role je proces jako celek. Jeho úkolem je upozorňovat vývojový tým na každou odchylku od stanoveného plánu. Nejdůležitější činnost, kterou kouč provádí je kontrola, zda členové týmu mezi sebou otevřeně komunikují (Kadlec, 2004).

### **Tester**

V rámci této metodiky se tester zabývá především pomáháním zákazníkovi při psaní jednotlivých testů pro každou implementovanou funkcionalitu (Kadlec, 2004).

### **Stopař**

Primárně provádí časové odhady a sbírá veškerou zpětnou vazbu. Díky tomu má povědomí o průběhu projektu, a zda projekt postupuje dle stanoveného plánu. Jeho úkolem je analyzovat důvody, proč se projekt zpožďuje (Kadlec, 2004).

## **2.1.5 Výhody a nevýhody extrémního programování**

Primární výhodou metodiky extrémního programování (XP) je její jednoduchost. Programátoři, kteří tuto metodiku při vývoji funkcionalit využívají, vytvářejí jednoduchý kód, který lze kdykoliv vylepšit (Kukhnavets, 2018). Jednoduchost však může být i nevýhodou. Pro vývojový tým může být těžké naučit se dělat vše jednodušším způsobem. Obtížné může být také jít za zákazníkem a nechat si vysvětlit i ty nejzákladnější koncepty nebo přiznat neznalost programátorských základů (Kadlec, 2004). Mezi hlavní výhody také patří také skutečnost, že týmy využívající XP ušetří spoustu peněžních prostředků, a to z důvodu eliminace přílišné dokumentace. Je to díky tomu, že se tým snaží problémy řešit prostřednictvím diskusí uvnitř týmu. Další silnou stránku je neustálá zpětná vazba, které je třeba naslouchat a včas provést potřebné změny. Extreme Programming pomáhá vytvářet jednotlivé funkcionality softwaru rychleji a díky pravidelnému testování v průběhu vývoje je poté rychlejší i samotná implementace. Metodika extrémního programování není nejlepší volbou, pokud je tým od sebe geograficky oddělen, jelikož jedním z principů XP je právě párové programování, kde jeden kód vytvářejí dva vývojáři vedle sebe (Kukhnavets, 2018).

### 2.1.6 Vhodnost

Metodika Extreme Programming je vhodná spíše pro malé až střední týmy (se dvěma až deseti členy), které se v průběhu vývoje musí přizpůsobit rychle se měnícímu nebo nejasnému zadání. Rozhodně se tato metodika nehodí pro velké týmy, které mají více jak dvacet členů (Kadlec, 2004).

## 2.2 Scrum

Rámec Scrum se používá k řízení komplexního vývoje produktů již od počátku 90. let 20. století (scrum guide). Poprvé byl tento rámec představen Kenem Schwaberm a Mikem Beedlem v roce 1995 na *OOPSLA (Object-oriented, Programming, Systems, Languages & Applications)* konferenci. Oba autoři již před tím prošli několika procesy vývoje softwaru a bylo pro ně důležité vyvinout takovou metodiku či rámec, který by se dokázal rychle a mnohem lépe přizpůsobit měnícím se požadavkům a který by zvýšil efektivitu vývojového procesu (Kadlec, 2004).

*SCRUM Development Process* (používá se pouze název SCRUM) vystřídal Extrémní programování jako nejpoužívanější agilní metodiku, i když nemůžeme tyto metodiky mezi sebou na tolik porovnávat. Scrum se zaměřuje na fungování organizace, kdežto XP je zaměřené technický proces vývoje softwaru (Meyer, 2014). Scrum však nebyl navržen jako metodika či proces, ale jako rámec. Tento rámec je lehký a jednoduchý na pochopení, ale ve své podstatě velmi obtížný na zavedení. Zvládnutí tohoto rámce vyžaduje roky praxe a neustálého zlepšování (Hunt, 2018). Je to tedy rámec, v němž může organizace řešit složité problémy a zároveň produktivně a kreativně vytvářet produkty s nejvyšší možnou hodnotou (Schwaber, 2020). Podobně jako ostatní agilní metodiky a postupy, využívá tento rámec výhod inkrementálního a iterativního přístupu (Kadlec, 2004). Iterativní znamená, že vývojový tým dodává práci často, nikoliv najednou. Inkrementální pak, že ji dodává v malých balíčcích po jednotlivých funkcích (Iqbal, 2021). Tento rámec se skládá z jednotlivých Scrum týmů s nimi spojených rolí, událostí, artefaktů a pravidel. Každá složka rámce pak slouží konkrétnímu účelu a je nezbytná pro jeho úspěch (Schwaber, 2020).

## **2.2.1 Tři základní pilíře**

Scrum je založen na empirické teorii neboli na empirismu. Empirismus tvrdí, že znalosti vycházejí ze zkušeností a rozhodování na základě toho, co už člověk zná. Každou implementaci empirického řízení procesů podporují tři pilíře: transparentnost, kontrola a přizpůsobování (Schwaber, 2020).

### **Transparentnost**

Všechny procesy a vznikající práce musí být vždy viditelné pro všechny osoby, kterých se týkají. Mezi tyto osoby patří členové vývojového týmu, který jednotlivé funkcionality vytváří. Dále pak zákazník, který dané funkcionality přijímá a v neposlední řadě všechny další zúčastněné osoby. Je důležité, aby bylo jasné na čem se pracuje, v jakém stavu jsou jednotlivé funkcionality nebo na čem se bude pracovat v dalším sprintu (Doležal, 2022). Je tedy důležité, aby byly tyto aspekty dobře a jasně definovány společným standardem a aby pozorovatelé sdíleli společnou představu o tom, co vidí (Schwaber, 2020).

### **Kontrola**

Je důležité kontrolovat veškeré artefakty Scrumu a pokrok směrem k cíli sprintu. Provádějí se z toho důvodu, aby se odhalily nežádoucí odchylky. Kontroly by neměly být tak časté, aby nezasahovaly do každodenní práce vývojových týmů (Schwaber, 2020). Veškeré odchylky jsou poté identifikovány a následně jsou předmětem pro další přizpůsobování (Doležal, 2022).

### **Přizpůsobování**

Pokud se při kontrole najde jedna nebo více odchylek je důležité jim věnovat pozornost a provést úpravu co nejdříve, aby se minimalizovali případné další odchylky (Schwaber, 2020). Proto je potřeba, aby měli všechny zúčastněné strany dostatek kompetencí a pravomocí k tomu, aby mohly jednotlivé úpravy či změny provádět (Doležal, 2022). Přizpůsobení také poukazuje na to, jestli byla zákazníkovi dodána taková hodnota, jakou očekával. Čím déle se bude s přizpůsobením čekat, tím se zvyšuje riziko nedostatečné dodávané hodnoty a dalších problémů (Hunt, 2018).

## **2.2.2 Pět hodnot Scrumu**

Mezi pět hodnot rámce Scrum patří: oddanost, odvaha, soustředění, otevřenost a respekt. Díky tomu je tento rámec snadno pochopitelný, ale na druhou stranu se velmi obtížně uplatňuje v praxi. Celý tým musí být oddaný cílům a procesům. Musí mít odvahu zkoušet nové věci, ale také dělat chyby. Soustředit se na dodávání hodnoty a být otevřený novým myšlenkám, názorům, ale i kritice (Hunt, 2018). Tyto všechny hodnoty definují to, jak se tým chová, jak pracuje a toto vše by mělo hodnoty posilovat a nepodkopávat je (Doležal, 2022).

## **2.2.3 Scrum role**

Rámec Scrum je založen na principu samo-organizovaného týmu, co nejvíce transparentní komunikace a kultuře, která podporuje vzájemnou spolupráci a sdílení všech informací. Avšak aby toto vše fungovalo, zavádí Scrum specifické role, které tradiční přístupy k projektovému řízení neměly – a to Product Ownera (vlastníka produktu), Scrum Mastera a Development (vývojový) tým (Šochová, 2019). Product Owner je zodpovědný za to, co bude vyvinuto a v jakém pořadí. Scrum Master je zodpovědný za vedení týmu při vytváření a dodržování Scrum procesu. Development tým je pak zodpovědný za dodání funkcionalit, které zákazník požaduje (Rubin, 2013).

### **2.2.3.1 Product Owner (PO)**

Product Owner neboli vlastník produktu je zodpovědný za maximalizaci dodané hodnoty produktu a za odvedenou práci vývojového týmu (Schwaber, 2020). Má na starosti definici produktové vize a její následnou komunikaci vývojovému týmu, zákazníkům a v neposlední řadě také firmě. Důležitou roli hraje při definici priorit, a rozhoduje na které funkcionalitě se bude pracovat první, na které později a také na kterých se nebude pracovat vůbec (Šochová, 2019). Jeho hlavní odpovědností je spravování produktového backlogu. Produktový backlog je uspořádaný seznam všeho, co by mohlo být pro produkt potřeba a je jediným zdrojem požadavků, které mají být v produktu zahrnuty (Schwaber, 2020).

Vzhledem k tomu, že je Product Owner jedinou odpovědnou osobou za správu backlogu, je jeho úkolem jasně vyjádřit, jaké položky se v backlogu nacházejí a zároveň tyto položky

přeorganizovat a uspořádat tak, aby bylo co nejlépe dosaženo stanovených cílů a aby byly splněny všechny zákaznickovy požadavky (Hunt, 2018).

Je týmu vždy potřeby k dispozici, ale více času tráví se zákazníky, aby je mohl v dostatečné míře poznat a na základě toho, pak rozhodnout, jaké požadavky jsou pro zákazníka ty nejdůležitější, a které mu přinesou největší hodnotu. Tato role by pro úspěšné dodání jednotlivých funkcionalit neměla být kombinována s rolí Scrum Mastera (Šochová, 2019).

### **2.2.3.2 Scrum Master (SM)**

Scrum Master pomáhá všem zúčastněným pochopit a přijmout hodnoty, principy a postupy rámce Scrum. Působí jako kouč, poskytuje vedení procesů a pomáhá vývojovému týmu a zbytku organizace vyvinou vlastní přístup k využívání tohoto rámce. Funguje jako facilitátor, který pomáhá týmu řešit problémy a zlepšovat se v používání Scrumu. Je také zodpovědný za ochranu vývojového týmu před vnějšími hrozbami a zásahy a pomáhá odstraňovat překážky, které snižují týmovou produktivitu (Rubin, 2013).

Scrum Master dělá vše pro to, aby se týmu co nejlépe pracovalo. Je součástí týmu a měl by mu být kdykoliv k dispozici. Měl by proto být ve společné místnosti s vývojovým týmem (Šochová, 2019). Tato role nemá žádnou pravomoc vykonávat kontrolu nad týmem. Scrum Master funguje jako vedoucí, nikoliv jako manažer. Často je tato role popisována jako „servant leader“ vývojového týmu. Servant leader zajišťuje plnění týmových potřeb s vysokou prioritou. Nezeptá se: „Co pro mě dnes uděláte?“, ale namísto toho se servant leader zeptá: „Co mohu udělat já, aby byl tým co nejvíce efektivní?“ (Rubin, 2013). Jeho cílem je vést tým k přijímání zodpovědnosti, být samostatný a samo-organizovaný (Doležal, 2022).

### **2.2.3.3 Development tým (vývojový tým)**

Development týmy jsou „*cross-functional*“ neboli multifunkční, což znamená, že každý člen týmu má veškeré znalosti a dovednosti, které jsou potřeba pro dodání hodnoty na konci každého sprintu. Neznamená to však, že jsou všichni členové týmu experti na jednotlivé problematiky, ale že mají dostatečné povědomí o tom, jak je řešit (Doležal, 2022). Jakmile Product Owner produktu rozhodne, které položky backlogu jsou nejcennější, vytvoří vývojový tým takový inkrement (přírůstek), který odpovídá stanoveným požadavkům koncového uživatele (Hunt, 2018). Tým je také „*self-organized*“ neboli samo-organizovaný. Členové se sami organizují, a to za účelem určení nejlepšího

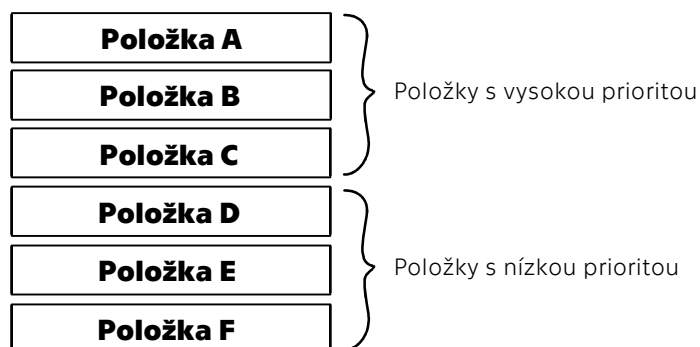
způsobu, jak dodat na konci jednotlivých sprintů co nejvyšší hodnotu. Žádný projektový ani jiný manažer by týmu neměl říkat, jak má pracovat a co by měl v danou chvíli dělat. Ani Scrum Master by toto neměl dělat (Rubin, 2013).

## 2.2.4 Artefakty Scrumu

Artefakty definované rámcem Scrum jsou speciálně navrženy tak, aby maximalizovaly transparentnost klíčových informací, aby měli všechny zúčastněné strany o všem stejné porozumění (Schwaber, 2020).

### 2.2.4.1 Product Backlog

Product (produktový) backlog je neustále se rozšiřující a vyvíjející se seznam toho, co vše je potřeba udělat pro dodání jednotlivých funkcionalit produktu. Představuje jediný zdroj práce prováděné development týmem (Doležal, 2022). Product Owner spolupracuje s interními i externími stakeholdery na shromažďování a definování jednotlivých položek backlogu. Poté zajistí, aby byly položky umístěny ve správném pořadí tak, že položky s vysokou prioritou se objevují v horní části backlogu a položky s prioritou nižší se objevují na nižších pozicích. Na základě toho můžeme říct, že je produktový backlog neustále se vyvíjejícím artefaktem (Rubin, 2013).



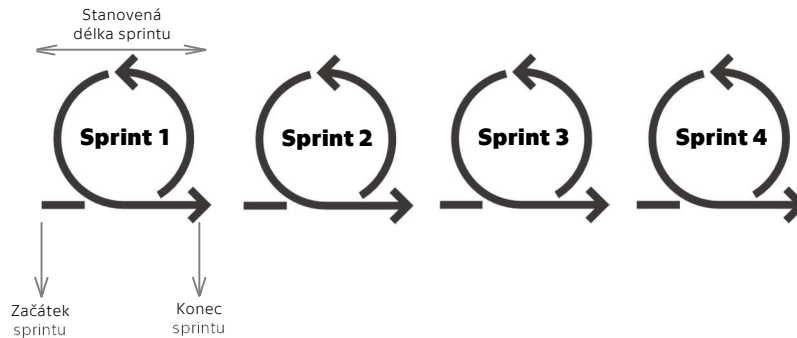
OBR. 3: PRODUCT BACKLOG (ZDROJ: RUBIN, 2013)

### 2.2.4.2 Sprint

Ve Scrumu se pracuje ve stanovených časových cyklech či iteracích, kterým se říká sprint. Sprint obvykle trvá maximálně jeden kalendářní měsíc. Na konci každého sprintu by měly být dodány takové funkcionality, které byly na daný sprint naplánované a měly by zákazníkovi či koncovému uživateli přinést určitou hodnotu (Rubin, 2013). Každou iterací, každým sprintem se vývojový tým učí, jak správně a efektivně pracovat, jak dodávat co



nejlepší hodnotu a získává pravidelnou zpětnou vazbu. Obecně platí, že čím je sprint kratší, tím lepší. Kratší sprinty znamenají rychlejší a pravidelnější zpětnou vazbu a s tím spojenou následnou adaptaci (Šochová, 2019). Nový sprint následuje bezprostředně po tom, co probíhající sprint skončí (Rubin, 2013).



OBR. 4: SPRINT (ZDROJ: RUBIN, 2013)

### 2.2.4.3 Sprint backlog

Sprint backlog je v podstatě jen užší výběr položek z celkového produktového backlogu. Výběr jednotlivých položek ale vychází z priorit, které jim byly přiděleny. Vybíráme ty, které mají nejvyšší prioritu, a které musí být vyvinuty v následujícím sprintu (Hunt, 2018). Sprint backlog zviditelňuje veškerou práci, kterou vývojový tým a Product Owner označí za nezbytnou pro naplnění sprint cíle (Schwaber, 2020).

### 2.2.4.4 Sprint Goal (Sprint cíl)

Sprint cíl patří mezi nejdůležitější části metodického rámce Scrum. Dalo by se říci, že je to v podstatě taková vize pro daný sprint. Tento cíl adresuje požadavky a potřeby zákazníků a zaměřuje se na hodnotu, nikoliv na funkcionalitu jako takovou (Šochová, 2019). Na stanovení tohoto cíle spolupracují Product Owner společně s vývojovým týmem (Doležal, 2022).

### 2.2.4.5 Planning Poker (plánovací poker)

Planning poker je zřejmě nejvíce přesnou metodou, jak relativně odhadovat velikost jednotlivých položek backlogu. Jednotlivé položky se hodnotí podle číselné škály. Jedná se o upravenou Fibonacciho posloupnost (následující číslo je součtem dvou předchozích čísel), například: 0, 1, 2, 3, 5, 8, 13, 20, 40 a 100. Tyto čísla mohou být ještě doplněna o karty nekonečna, kdy je daná položka příliš komplexní a tým nedokáže odhadnout její velikost a o kartu otazníku, kdy se tým potřebuje doptat Product Ownera na další informace ohledně dané položky. Po představení položky Product Ownerem každý z členů týmu

provede odhad velikosti položky a vybere kartu s příslušnou hodnotou. Je samozřejmé, že se všichni členové týmu jednotně neshodnou na velikosti položky. O položce se diskutuje do té doby, než se všichni členové vývojového týmu shodnou na její velikosti (Doležal, 2022).

## **2.2.5 Scrum ceremonie**

Všechny Scrum ceremonie jsou časově ohraničené, což znamená, že každá má stanovenou maximální dobu trvání. V okamžiku, kdy začne sprint, je jeho trvání pevně stanovené a nelze ho prodlužovat ani zkracovat (Schwaber, 2020).

### **2.2.5.1 Daily Scrum (Denní scrum)**

Daily Scrum patří mezi nejznámější praktiky (ceremonie), které se v agilních metodikách a rámcích používají. Tato ceremonie se vyznačuje jednoduchostí na implementaci, takže s ní začíná většina týmů. Jak již vychází z názvu, koná se tato ceremonie každý den (Šochová, 2019). Její trvání je časově omezeno na 15 minut denně. Denní scrum má za úkol dvě věci. Za prvé je jeho úkolem zajistit informovanost všech členů týmu o stavu sprintu a všech vytvářených položkách a za druhé je cílem zajistit, aby v případě jakýchkoliv problémů, došlo k jejich identifikaci a následnému řešení (Myslín, 2016). Scrum Master zde vystupuje pouze v roli facilitátora a moderátora diskuse (Šochová, 2019).

### **2.2.5.2 Sprint Planning**

Během Sprint Planningu se vývojový tým, Scrum Master a Product Owner dohodnou na tom, co má být v následujícím sprintu dosaženo (Rubin, 2013). Trvání plánování sprintu je omezeno maximálně na osm hodin. Omezení osmi hodin platí pro sprint, který má trvání třiceti dnů. U sprintů, které trvají kratší dobu je i toto plánování kratší. Role Scrum Mastera zodpovídá za to, aby se plánování konalo, a aby všichni její účastníci rozuměli, proč a za jakým účelem se plánování provádí. Scrum Master také učí tým, aby neporušoval časové omezení Sprint Planningu (Schwaber, 2020). Planning začíná tím, že Product Owner nejdříve představí cíl sprintu. Na základě stanoveného cíle, pak tým vybírá, jaké položky produktového backlogu mu nejlépe pomohou tohoto cíle dosáhnout. Při výběru nesmí tým zapomenout kapacitu, kterou má tým k dispozici a na hodnotu, která musí být zákazníkovi dodána (Šochová, 2019).

### **2.2.5.3 Sprint Review**

Cílem této ceremonie je přezkoumat výsledek sprintu a určit, zda je potřeba cokoliv změnit nebo přizpůsobit (Doležal, 2022). Pro tuto ceremonii je zásadní konverzace, která probíhá mezi development týmem a dalšími zainteresovanými stranami, kam také patří sponzoři a zákazníci. Celá konverzace je zaměřena na přezkoumávání dokončených funkcionalit v kontextu celkového vývojového úsilí. Všichni účastníci tak získají přehled o tom, co se děje a co je potřeba případně vylepšit (Rubin, 2013). Výsledek je prezentován Product Ownerem a Scrum Masterem a ostatní členové vývojového týmu zde slouží především k odpovídání na případné dotazy týkající se jednotlivých prezentovaných funkcionalit (Myslín, 2016).

### **2.2.5.4 Sprint Retrospective (Retrospektiva)**

Sprint Retrospective neboli retrospektiva je ceremonie konající se obvykle na konci každého sprintu. Na této schůzce tým diskutuje o třech otázkách (Šmite, 2010):

1. Co se během sprintu povedlo?
2. Co se na druhou stranu zase nepovedlo?
3. Co bychom mohli v příštím sprintu vylepšit?

Trvání retrospektivy je časově omezeno na jednu až dvě hodiny a koná se v zasedací místnosti. V rámci této ceremonie se tým může zabývat vším, co se v průběhu celého sprintu událo (jak se plnily úkoly, přístup jednotlivých členů vývojového týmu atd.). Cílem tohoto meetingu však není hodnocení jednotlivců. Této ceremonie se také účastní Scrum Master jako moderátor, který ale také hodnotí průběh celého sprintu a případně navrhuje některá opatření (Myslín, 2016). Na konci každé retrospektivy by měl vývojový tým identifikovat zlepšení, která bude následně implementovat v dalším sprintu (Schwaber, 2020).

## **2.3 Kanban**

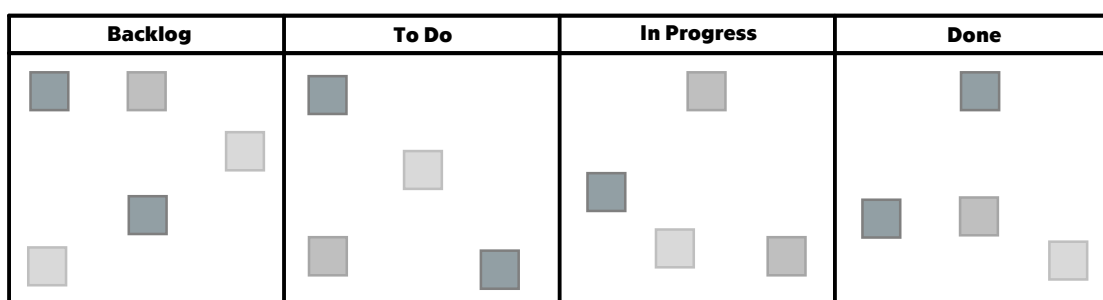
Metoda Kanban slouží jako prostředek k navrhování, řízení a zlepšování systémů toku práce. Kanban je velmi flexibilní rámec, a díky tomu se používá v organizacích tam, kde se ostatní rámce či metodiky (Scrum či eXtreme Programming) úplně nehodí. Kanban proces se více hodí například do administrativy (operations) či do týmů zabývajících se údržbou (maintenance). Na rozdíl od ostatních agilních metodik či rámců nám Kanban nic nenařizuje (Šochová, 2019).

### 2.3.1 Základní principy Kanbanu

Metoda Kanban je založena na třech základních principech: vizualizace, minimalizace rozpracované práce (work-in-progress) a řízení toku práce (managing workflow) (Hammarberg, 2014).

#### Vizualizace

Vizualizovat práci můžeme jednoduchým způsobem, a to tak, že si vytvoříme a následně nalepíme samolepící lístečky (post-it notes), které představují jednotlivé položky práce na předem vytvořenou Kanban tabuli (Kanban board). Toto týmu pomůže sledovat aktuální stav každé položky ve vývoji. Jednotlivé položky pak dle aktuálního stavu dělíme do vytvořených sloupců (kategorií): backlog, to-do, work-in-progress a done. Je to skvělý způsob, jak se seznámit s jednotlivými položkami a jak práce s nimi funguje. Díky zobrazení pracovního postupu na tabuli v jednotlivých sloupcích se proces vývoje stává pro členy týmu viditelným a jasným (Hammarberg, 2014).



OBR. 5: KANBAN BOARD (ZDROJ: ŠOCHOVÁ, 2019)

#### Minimalizace work-in-progress

Tento princip říká, že si tým záměrně stanoví limit množství položek, na kterých se bude pracovat najednou. Při menším počtu rozpracovaných položek, na kterých se pracuje současně, je každá položka hotová a dodána rychleji, než kdyby tým pracoval na velkém množství položek najednou. Tím, že tým minimalizuje a omezuje work-in-progress, vytváří tak určité napětí v pracovním procesu a díky tomu může odhalit problémy, které se v pracovním procesu vyskytují. Omezení počtu rozpracovaných položek také pomáhá odhalovat příležitosti ke zlepšení (Hammarberg, 2014).

## Řízení toku práce (Managing workflow)

Řízení workflow je nikdy nekončící práce. Je to činnost, na které se pracuje stále, jelikož pracovní postup lze vylepšovat vždy. Vždy se najde místo či proces, který tým ve své práci zpomalí. Dobrou zprávou však je, že se tyto problémy odhalí sami, díky vizualizaci procesu prostřednictvím Kanban boardu. Často se tak odhalí ty největší problémy, které může tým hned vyřešit (Hammarberg, 2014). Workflow je řízené přímo na Kanban tabuli, kde se jednotlivé položky přesouvají v reálném čase (Doležal, 2022).

Tyto tři zmíněné principy jsou základem celého rámce Kanban. Později se však ještě rozšířili o další tři a vzniklo tak šest základních postupů Kanbanu. Dalšími třemi principy a postupy jsou (Hammarberg, 2014):

- Jasná formulace zásad procesů
- Zavádění smyček zpětné vazby
- Společné zlepšování a experimentální vývoj

### 2.3.2 Základní hodnoty Kanbanu

Týmy, které využívají Kanban ke zlepšování poskytovaných služeb a dodávání požadovaných funkcionalit, se řídí několika hodnotami (What is Kanban?, 2018):

- **Spolupráce** – tento rámec byl vytvořen ke zlepšení způsobu, jakým lidé mezi sebou navzájem spolupracují.
- **Transparentnost** – díky transparentnosti informací a jejich sdílení se zlepšuje tok podnikových hodnot.
- **Zaměření na zákazníka** – cílem je co nejvíce optimalizovat tok hodnoty pro zákazníka. Zákazník může být jak externí, tak i interní.
- **Vyváženost** – pro dosažení co nejvyšší efektivity, je potřeba vyvážit a brát v úvahu různé aspekty, úhly pohledu, schopnosti a zkušenosti.
- **Dohoda** – všechny osoby podílející se na systému musí být odhodláni tento systém zlepšovat a souhlasí, že se budou podílet na dosažení stanovených cílů. Zároveň však respektují a přijímají rozdílné názory v přístupech a názorech.
- **Respekt** – je důležité si lidí vážit, snažit se je pochopit a projevat k nim ohleduplnost.
- **Pochopení** – individuální poznávání a pochopení aktuální situace je nezbytné pro další zlepšování.

### **2.3.3 Metriky Kanbanu**

Rámec Kanban používá pro měření odhadu doby realizace jednotlivých položek a pro měření týmové kapacity specifické metriky (Doležal, 2022).

#### **Rychlost zpracování položek Kanbanu**

Tato metodika udává, kolik položek / úkolů stejné náročnosti může tým zpracovat v daném časovém období (například v jedné iteraci). Tato rychlost se počítá průběžně a pravidelně. Pokud chceme, aby tato měření dobře fungovala a poskytovala nám co nejméně zkreslené informace, je potřeba začít již na začátku vytvářet položky, které jsou zhruba stejné velikosti. Díky tomu, že víme, jak rychle tým pracuje, dokážeme lépe předvídat, kdy budou všechny potřebné položky dokončeny (Doležal, 2022).

#### **Dodací lhůta, doba trvání (lead time) a čas cyklu (cycle time)**

Všechny tyto metriky definují průměrnou dobu, která je potřeba k dokončení jedné položky (úkolů). Dobu trvání měříme od chvíle, kdy tým obdrží od zákazníka určitou specifikaci či požadavek. Čas cyklu začínáme měřit od okamžiku, kdy tým začne pracovat na úkolu a dodací lhůta nám pak udává, jak dlouho zákazník čeká, než je mu celkový produkt dodán (Doležal, 2022).

## 3 Škálovaná agilita – Scaled Agile

Týmy, které se zabývají vývojem softwaru, čím dál tím více dokazují, že zavedení agilních rámců, jakými jsou například Scrum či Kanban, jim umožňuje dodávat hodnotu zákazníkům rychleji, s větší předvídatelností a umožňují jim rychle reagovat na nové informace či změny. Zavedení těchto postupů na úrovni týmů je poměrně snadné. Skutečnou výzvou je však rozšíření těchto postupů do více týmů ve velké organizaci. Jinými slovy je škálování agility implementací agilních přístupů ve velkém měřítku. A co to znamená, že organizace implementuje agilní přístupy ve velkém měřítku? To znamená schopnost řídit agilitu jak na úrovni týmů, tak zároveň uplatňovat stejné principy, postupy a výsledky na dalších úrovních celé organizace (Berman).

Je přirozené, že vedoucí pracovníci, kteří mají s agilitou ať už v jakékoliv formě zkušenost nebo o ní jen slyšeli, si kladou následující otázky: Co kdyby organizace spustila několik desítek agilních týmů v celé organizaci? Mohly by se takto naučit fungovat různé segmenty organizace? Zlepšilo by škálování agility výkonnost podniku stejně jako agilní metody zvýšily výkonnost jednotlivých týmů? (McGregor, 2020).

Organizace se až příliš často pokoušejí o škálování agility do celé její struktury, aniž by si nejdříve zjistily, zda jsou vůbec schopné být agilní. Investují tak spoustu peněz a času do implementace agilních postupů a metod ve velkém měřítku, místo toho, aby se soustředili na plynulost fungování jednotlivých týmů. Pokud chce organizace agilitu škálovat, musí nejdříve škálovat svou schopnost být agilní. To zahrnuje tři části: organizační schopnost, schopnost koučování a schopnost týmů (Shore, 2021).

### **Organizační schopnost**

Než společnost utratí velké množství peněz za samotné škálování, měla by nejdříve identifikovat překážky, které by tomu mohly bránit. Organizace buď spolupracuje s expertem, který má konkrétní návrhy, nebo se do toho pouští sama. V tomto případě by měla začít s jedním pilotním týmem nebo malou skupinou týmů. Jak se bude pilotní tým rozvíjet, bude postupně identifikovat překážky a problémy, které mu brání v dosažení plynulosti fungování. Tyto problémy je potřeba zaznamenat a co nejdříve vyřešit. Jakmile je organizace schopna podporovat fungování malého množství týmů, může se dále pokoušet o škálování agility do celé organizace (Shore, 2021).

## Schopnost koučování

Při škálování agility do celé organizace je zapotřebí pomoc kvalifikovaného kouče na úrovni týmů, který těmto týmům pomáhá ve zlepšování plynulosti a efektivnosti fungování. Buď si organizace najme kouče externího nebo si ho sama vyškolí. Rychleji může organizace škálovat tak, že přesune kouče do dalšího týmu v okamžiku, kdy stávající tým dosáhl požadované efektivnosti a plynulosti (Shore, 2021).

## Schopnost týmu

Schopnost týmu správně využívat agilního přístupu silně závisí na schopnost koučů tyto týmy vést. Čím zkušenější agilní kouč je, tím rychleji bude tým v agilních přístupech plynulý, ale i přes to všechno to bude nějakou dobu trvat. Tento problém lze vyřešit tím způsobem, že si společnost přizve kvalifikovanou konzultingovou firmu, která jí se škálováním agility pomůže (Shore, 2021).

Škálování nad rámec několika agilních týmů vyžaduje lépe definovaný přístup ke struktuře a sladění vzájemných závislostí. Mezi oblíbené přístupy ke škálování agility patří: SAFe® (Scaled Agile Framework), Nexus, LeSS (Large-Scale Scrum) a stále hodně skloňovaný Spotify Model (Perkin, 2020). Pro účely této práce si popíšeme škálovací rámce SAFe® a Spotify Model.

## 3.1 Spotify Model

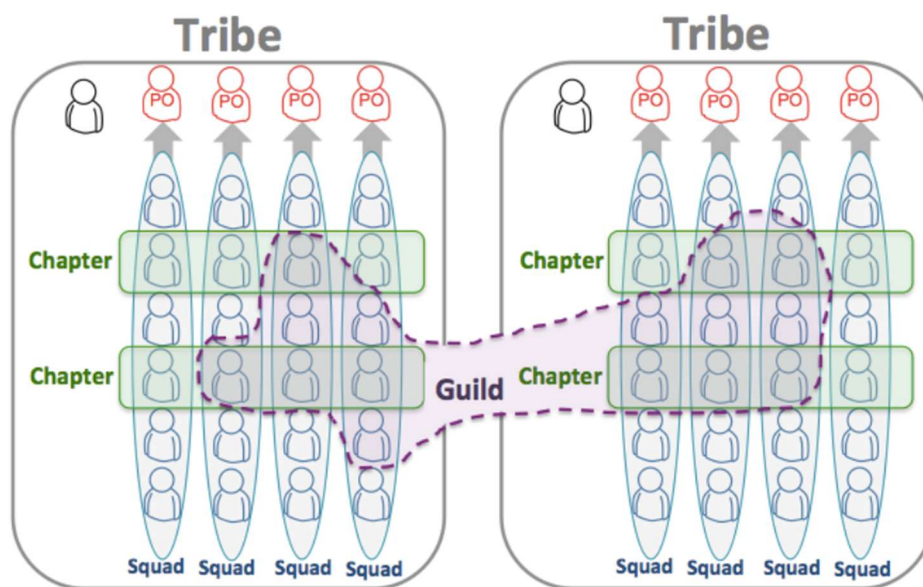
Spotify je největší hudební streamovací službou, která měla v říjnu 2019 celkem 248 milionů aktivních uživatelů měsíčně po celém světě. Společnost Spotify vyvinula vlastní agilní kulturu a přizpůsobila agilní postupy a metody tak, aby vyhovovaly velmi rozsáhlému programu (na kterém pracuje více jak 300 lidí) rozmístěnému ve čtyřech městech (Salameh, 2021).

Model Spotify byl poprvé představen v roce 2012, kdy Henrik Kniberg a Anders Ivarsson publikovali *Scaling Agile @ Spotify*, který představil způsob, jakým společnost Spotify přistupuje k agilitě. Spotify Model je autonomní přístup ke škálování agility založený na lidech, který zdůrazňuje význam kultury a sítě (network). Tento model pomohl Spotify a



dalším společenstvem, které ho implementovaly, zvýšit inovativnost a produktivitu díky zaměření na autonomii, otevřenou komunikaci, odpovědnost a kvalitu (Cruth).

Tento model je tedy založen na vytváření autonomních, ale sladěných týmů. K vytváření těchto týmů využívá organizace adaptivní strukturu. Tato struktura je založena na matici dvou dimenzí. První dimenze je vertikální a představuje komunitu squadů (squads) a tribů (tribes) a druhá dimenze, horizontální, pak představuje komunitu chapters a guilds (Salameh, 2021).



OBR. 6: ŠKÁLOVÁNÍ ZA POMOCI TRIBES, SQUADS, CHAPTERS A GUILDS (ZDROJ: KNIBERG, 2012)

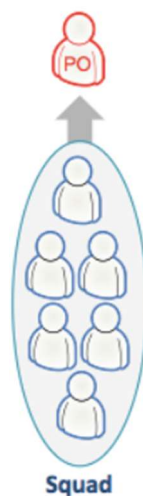
### 3.1.1 Squads

Základní vývojovou jednotkou Spotify modelu jsou Squads. Squad je podobný týmu Scrum a je navržený tak, aby připomínal mini start-up. Členové squadu mají všechny dovednosti a nástroje potřebné k návrhu, vývoji, testování a dodání konečného produktu. Jsou samo-organizovaným týmem a sami si určují, jakým způsobem fungují. Některé týmy používají Scrum, některé Kanban a některé používají kombinaci těchto dvou rámců. Každý squad má stanovený svůj dlouhodobý cíl, na který se při práci zaměřuje a dělá vše pro to, aby ho dosáhl. Protože se každý squad dlouhodobě věnuje jednomu cíli a dodání jedné části, může se časem stát v dané oblasti skutečným odborníkem (Kniberg, 2012).

Většina týmů má k dispozici vlastní pracovní prostory, které zahrnují pracovní plochy, odpočinkovou zónu a téměř na každé stěně visí bílé tabule (whiteboard). V rámci podpory

učení a inovací je každý squad podporován a vybízen k tomu, aby aspoň 10 % svého času věnoval tzv. „hackerským dnům“. Během těchto dní lidé obvykle zkoušejí nové nápady a sdílejí je se svými kolegy (Kniberg, 2012).

Jednotlivé squady nemají formálně stanoveného vedoucího pracovníka, ale mají Product Ownery. Product Owner je zodpovědný za stanovení priorit jednotlivých položek práce v backlogu daného squadu, ale nijak nezasahuje do způsobu, jakým je práce vykonávána. Product Owneři jednotlivých squadů mezi sebou spolupracují na udržování plánu, který ukazuje, kam Spotify jako celek směřuje. Každý squad má také přístup k agilnímu kouči, který mu pomáhá se rozvíjet a zlepšovat způsob práce (Kniberg, 2012).



OBR. 7: SQUAD, ZÁKLADNÍ JEDNOTKA SPOTIFY MODELU (ZDROJ: KNIBERG, 2012)

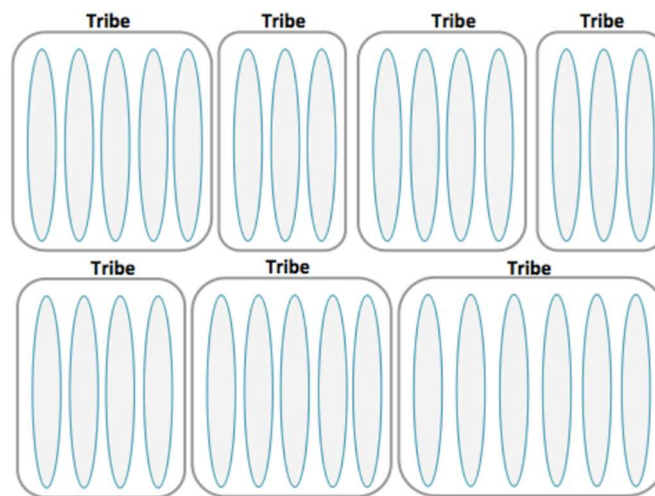
### 3.1.1.1 Závislosti mezi squady

V organizaci s větším počtem squadů se vždy nějaké závislosti objeví. Závislosti mezi squady však nemusí být nutně špatné. Někdy je potřeba vzájemné spolupráce, aby squady vytvořily lepší produkt. Cílem však je minimalizovat závislosti, které zpomalují či blokují práci jednotlivých squadů. Minimalizovat některé závislosti můžeme pomocí pravidelných schůzek, kde se tyto závislosti identifikují a následně se hledá řešení, jak tyto závislosti mezi squady odstranit (Kniberg, 2012).

### 3.1.2 Tribes

Tribe se skládá ze souboru squadů, které pracují v podobných oblastech vývoje. Jsou navrženy tak, aby každý tribe čítal méně než 100 členů. Cílem je podpořit spolupráci jednotlivých squadů a odstraňovat mezi nimi vzájemné závislosti (Salameh, 2021). Každý

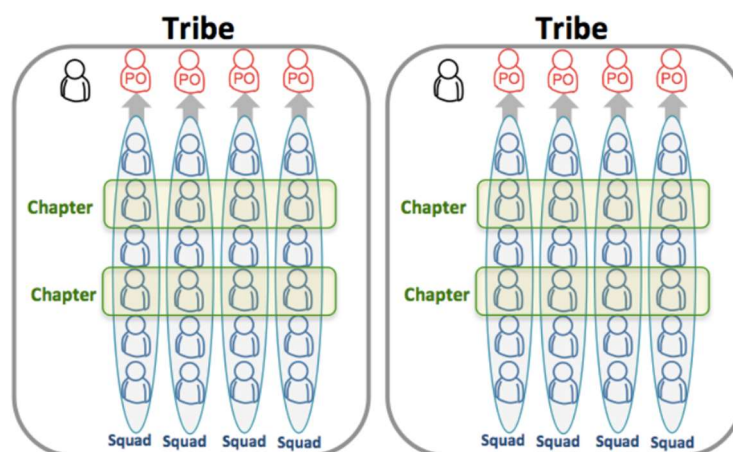
tribe má svého Tribe Leadera, který je zodpovědný za zajištění co nejlepšího pracovního prostředí pro squady uvnitř tribu (Kniberg, 2012).



OBR. 8: TRIBE (ZDROJ: KNIBERG, 2012)

### 3.1.3 Chapters

V rámci každého jednoho tribu existují malé skupiny lidí, kteří sdílejí stejné či podobné dovednosti a pracují ve stejných oblastech kompetencí. Těmto skupinám se říká chapters (Salameh, 2021). Členové chapters se pravidelně setkávají za účelem projednávání specifických problémů pro danou oblast odbornosti, jako například chapter testerů, chapter vývojářů atd (Kniberg, 2012).

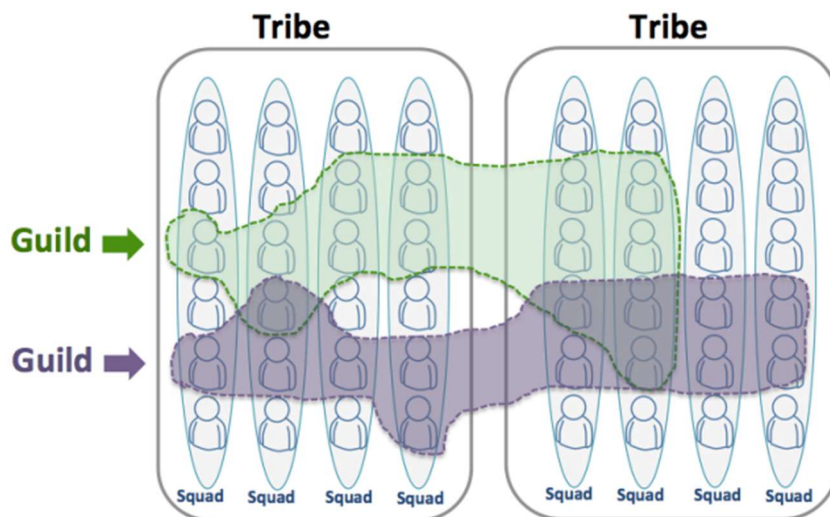


OBR. 9: CHAPTER (ZDROJ: KNIBERG, 2012)

### 3.1.4 Guilds

Guilds jsou rozsáhlé, zájmové skupiny lidí napříč jednotlivými triby, které chtějí sdílet znalosti a postupy v rámci celé organizace (Salameh, 2021). Členem těchto guilds se

může stát kdokoliv, účast je zcela dobrovolná. Zatímco chapters patří do jednoho tribu, guilds se mohou tvořit napříč různými triby (viz Obr. 10). Guild nemá stanoveného formálního vedoucího. Funguje to tím způsobem, že kdokoliv může být vedoucím této skupiny. Jeho cílem je pak danou skupinu pouze koordinovat a zajišťovat, že se pravidelně schází (Cruth).



OBR. 10: GUILDS (ZDROJ: KNIBERG, 2012)

### 3.1.5 Proč Spotify Model nefunguje?

Od roku 2012, kdy společnost tento model uvedla, se o jeho aplikaci snaží mnoho společností různých velikostí. Hlavním důvodem neúspěchu aplikace tohoto modelu je skutečnost, že většina organizací z tohoto modelu přebírá pouze názvy jednotlivých týmů, které již v podniku existují. Spousta společností tento model slepě přejímá, aniž by rozuměly tomu, za jakým účelem tak provádějí, a jaké problémy se tak snaží vyřešit (Ay, 2020).

Velké množství lidí považuje Spotify Model za jakýsi metodický rámec a také se podle něho řídí při škálování agilních přístupů ve své organizaci. Spotify Model však představuje pouze pohled společnosti Spotify na škálování agility jak z technického, tak kulturního hlediska. Je to pouze jeden z příkladů organizace více týmů skrze celou organizaci, která se zabývá vývojem produktů a zdůrazňuje potřebu kultury (Cruth). Sama společnost Spotify přestala tento model v roce 2017 používat a apelovala na organizace, aby jej nadále nekopírovaly (Ay, 2020).

Avšak existuje společnost, která tento model do své organizace úspěšně implementovala a dostala se tak do povědomí celé agilní komunity. Touto společností je Komerční Banka, která s nástupem Jana Juchelky v roce 2017 odstartovala rozsáhlou agilní transformaci s použitím právě Spotify Modelu. Tato velice komplexní transformace do agility se týkala primárně změn v systému fungování, kultury a organizačních struktur společnosti. Přejít na agilní řízení také pomohl rozhodnout o tom, odkud a jak fungují její zaměstnanci. V reakci na koronavirovou krizi se Komerční Banka rozhodla zavést hybridní režim, kdy mají zaměstnanci možnost pracovat jak z pohodlí domova, tak i z kanceláře. Přestože tato agilní transformace doposud neskončila, zaujaly její výsledky Harvard Business School, kde se na příkladu Komerční Banky vyučují principy agilní transformace. Případová studie, která tuto transformaci popisuje se jmenuje: Enterprise Agility at Komerční Banka (Holzman, 2021).

### **3.2 SAFe® - Scaled Agile Framework**

SAFe® neboli Scaled Agile Framework® je volně dostupná znalostní báze pro organizace, které vyvíjejí různé softwary a systémy. Tento škálovatelný a konfigurovatelný rámec pomáhá organizacím dodávat nové produkty a řešení v co nejvyšší kvalitě, s co nejvyšší přidanou hodnotou a v co nejkratším možném čase. Synchronizuje spolupráci a dodávky pro více agilních týmů najednou. Kombinuje agilní principy s principy štíhlé výroby (Lean Development) a systémovým myšlením (System Thinking). Poskytuje návod pro role, odpovědnosti, artefakty a činnosti, které jsou nezbytné k dosažení lepších výsledků (Leffingwell, 2018). Cílem tohoto rámce je pomoci podnikům stát se prosperujícími firmami v dnešním digitálním světě, které svým zákazníkům dodávají konkurenceschopný software v co nejkratší době (Knaster, 2020).

Rámec Scaled Agile Framework® byl vytvořen Deanem Leffingwellem a byl poprvé vydán v roce 2011. Od svého prvního vydání prošel tento rámec několika významnými aktualizacemi, které se vyvíjely společně s trhem a jeho neustále se měnícími způsoby vývoje softwaru a systémů. Přestože nové verze celého rámce jsou vydávány příležitostně, je znalostní báze mezi jednotlivými vydáními průběžně aktualizována (SAFe®, 2021). Dnes tento škálovací rámec používá stovka největších světových firem, aby si udržely svoji konkurenceschopnost na trhu, který se v podstatě mění každým dnem (Leffingwell, 2018). Celý rámec SAFe® a všechny jeho náležitosti lze nalézt také online na internetové stránce [SAFe 5 for Lean Enterprises \(scaledagileframework.com\)](https://scaledagileframework.com).

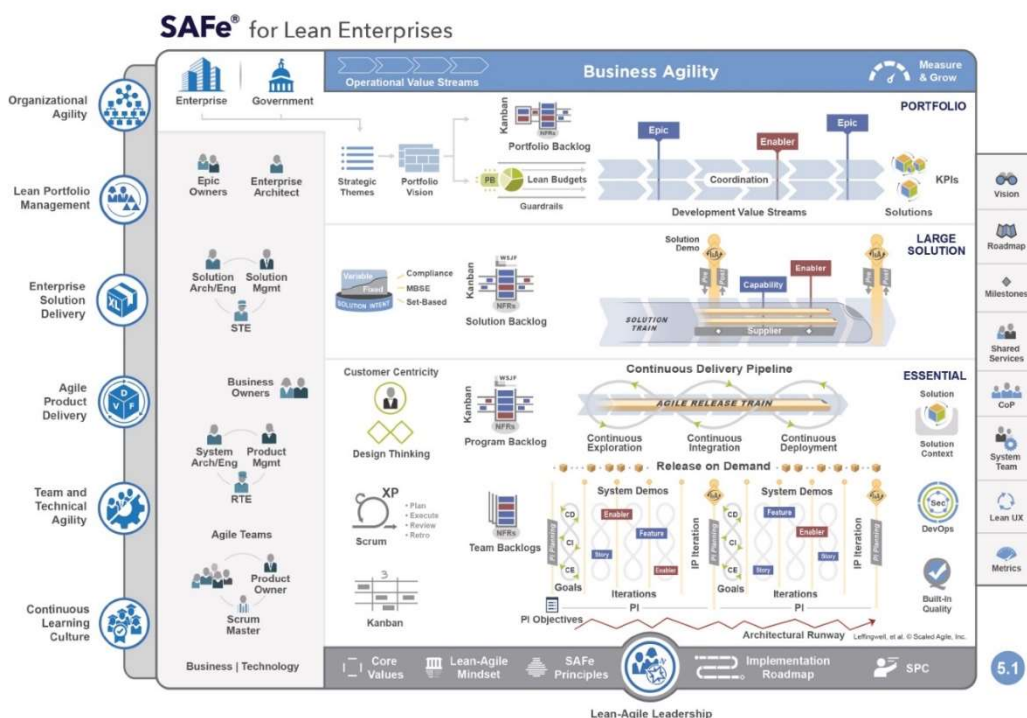
### 3.2.1 Proč implementovat SAFe?

Díky rámci SAFe® mohou společnosti lépe propojit svoji strategii s realizací, rychleji inovovat a hlavně rychleji dodávat vysoce kvalitní řešení na měnící trh. To vše se ukazuje být silnou výhodou, která pomáhá a umožňuje podnikům využít změn ve svuj prospěch. Přijetím SAFe® a uplatněním jeho hodnot, principů a postupů, může společnost dosáhnout větších organizačních i individuálních výhod (Knaster, 2020).

Zlepšuje organizační agilitu a zlepšuje obchodní výsledky organizaci všech velikostí po celém světě. Přináší zlepšení v oblasti času, který je potřeba k uvedení produktu na trh, zapojení zaměstnanců, vyšší dodávané kvality a hodnoty, s čím je úzce spojena vyšší spokojenost zákazníků. V neposlední řadě tento rámec pomáhá organizacím vytvářet kulturu, která je produktivnější, přínosnější, ale i zábavnější (Knaster, 2020).

### 3.2.2 Big Picture

SAFe® obsahuje model „Big Picture“, který poskytuje vizualizaci celého rámce a je hlavním uživatelským rozhraním této znalostní báze. Model „Big Picture“ poskytuje vhléd do postupů a principů, mezi které patří: základní kompetence Lean Enterprise, čtyři konfigurace (Essential, Large, Portfolio a Full), které podporují celou řadu vývojových prostředí, základní principy, hodnoty, způsob myšlení, role a další prvky, které samotný rámec tvoří (Knaster, 2020). Na SAFe® Big Picture si můžeme všimnout, že využívá metodik a rámců Scrum, Kanban a eXtreme Programming.



OBR. 11: SAFe® BIG PICTURE SE 4 KONFIGURACEMI (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

### 3.2.2.1 Sedm základních kompetencí

Sedm základních kompetencí představuje základní kámen pro pochopení a implementaci rámce SAFe®. Každá z nich je souborem znalostí, dovedností a chování, které společně umožňují společností dosáhnout organizační agility. Tyto kompetence jsou (Leffingwell, 2018):

- **Lean-Agile Leadership** – popisuje, jak Lean-Agile (štíhlá agilita) leadeři řídí a udržují organizační změny tím, že umožňují jak jednotlivcům, tak i týmům dosáhnout jejich nejvyššího potenciálu.
- **Týmová a technická agilita** – popisuje kritické dovednosti, principy a postupy Lean-Agile, které týmy používají k vytváření inovativních business řešení pro své zákazníky.
- **Agilní dodávání produktů** – zákaznicky orientovaný přístup k definování, vytváření a dodávání nepřetržitého toku produktů a služeb s vysokou hodnotou pro zákazníky a uživatele.
- **Dodávky podnikových řešení** – popisuje způsob aplikace principů a postupů Lean-Agile pro vývoj větších a sofistikovanějších aplikací, sítí a dalších systémů.
- **Lean Portfolio Management** – sladuje strategii a realizaci implementace přístupů Lean a systémového myšlení.
- **Organizační agilita** – popisuje jak jednotlivci a agilní týmy optimalizují své procesy, vyvíjejí strategii s jasnými závazky a rychle se přizpůsobují organizaci dle potřeby.
- **Kultura neustálého učení** – soubor hodnot a postupů, které podporují jednotlivce a organizaci jako celek k neustálému zvyšování znalostí, kompetencí a výkonnosti.



### 3.2.2.2 Konfigurace

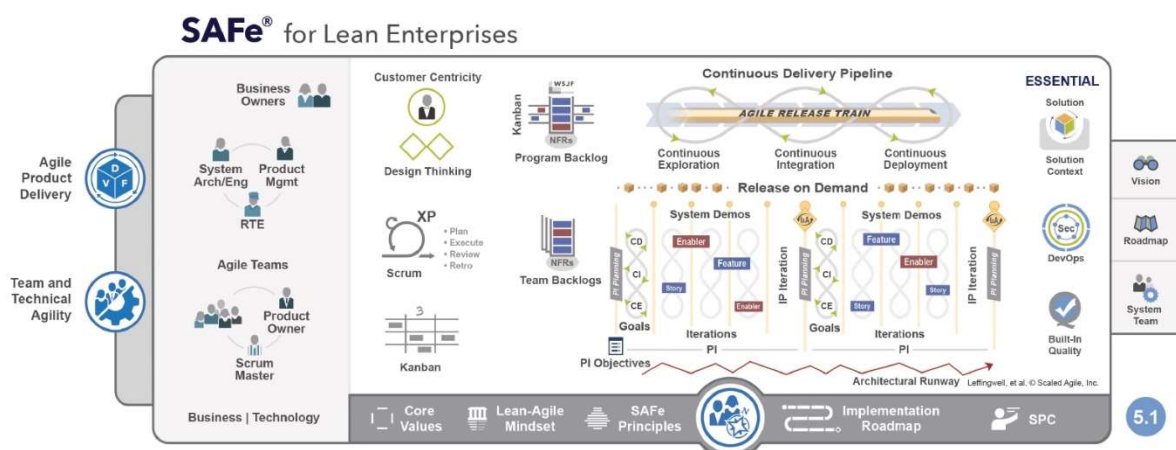
Díky čtyřem vytvořeným konfiguracím podporuje rámec SAFe® celé spektrum řešení, od těch, která jsou určena pro malý počet týmů (Essential konfigurace), až po komplexní systémy (Full konfigurace), které jsou určeny pro složité týmy se stovkami až tisíci členy (Knaster, 2020).



OBR. 12: SAFe® KONFIGURACE (ZDROJ: KNASTER, 2020)

#### Essential SAFe®

Essential SAFe® obsahuje minimální set rolí, událostí a artefaktů, které jsou potřeba k nepřetržitému dodávání business řešení. Je postaven na principech a postupech Lean-Agile Leadership, týmové agility a na agilním dodáváním produktů. Je základním kamenem pro ostatní konfigurace a představuje nejjednodušší bod pro implementaci rámce SAFe®. Základní strukturou Essential SAFe® je Agile Release Train (ART nebo také vlak). Zde se organizace, agilní a technické týmy a klíčové zainteresované strany věnují průběžnému dodávání jednotlivých řešení. Agile Release Train je to, co pohání SAFe® a umožňuje implementaci agility v organizaci (Knaster, 2020).

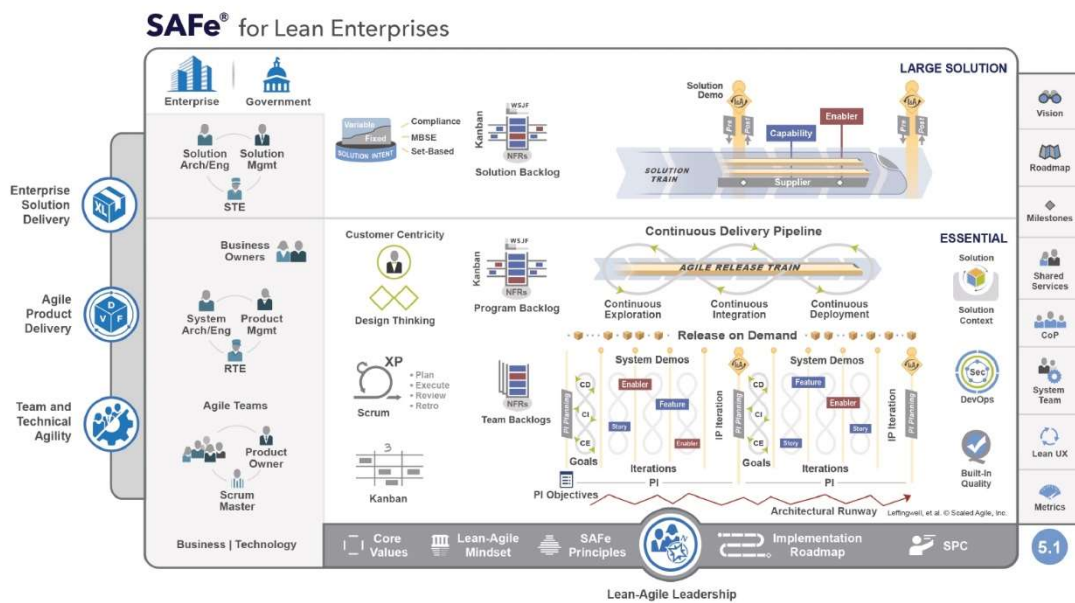


OBR. 13: ESSENTIAL SAFe® (ZDROJ: SCALED AGILE FRAMEWORK, 2020)



## Large Solution SAFe®

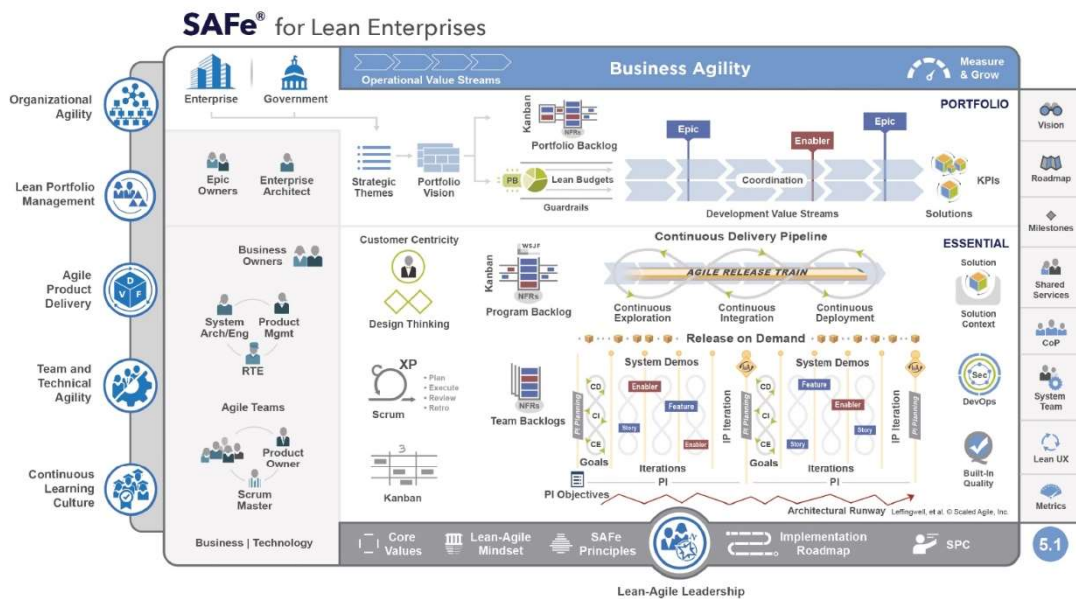
Large Solution SAFe® popisuje dodatečné role a postupy pro vývoj a dodávání velkých a komplexních systémů. Zahrnuje Essential SAFe® a zároveň zavádí kompetenci dodávání business řešení pro podporu vyvíjení složitějších řešení, která vyžadují fungování více ARTů. V této konfiguraci se nad úroveň Agile Release Train nachází Solution Train, organizační prostředek, který koordinuje fungování více ARTů a dodavatelů při dodávání komplexních systémů (Knaster, 2020).



OBR. 14: LARGE SOLUTION SAFe® (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

## Portfolio SAFe®

Portfolio SAFe® obsahuje minimální set kompetencí a postupů, které organizacím umožňují implementovat business agilitu. Tato konfigurace zahrnuje základní kompetence SAFe® a přidává kompetenci Lean Portfolio Managementu, organizační agilitu a kulturu neustálého učení. Pomáhá zajistit, aby se Value Stream (hodnotový tok) a jeho vlaky (ARTs) zaměřovaly na budování takových věcí, které jsou potřeba k dosažení strategických cílů (Knaster, 2020).



OBR. 15: PORTFOLIO SAFe® (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

### 3.2.2.3 Základní hodnoty SAFe®

Základní hodnoty rámce SAFe® definují ideály, které jsou nezbytné pro aplikaci tohoto rámce. Fungují jako vodítko, které organizacím ukazuje, na co všechno se musejí zaměřit, a zároveň jim pomáhají určit, zda jsou na správné cestě k naplnění svých stanovených business cílů (Leffingwell, 2018). Základními hodnotami tohoto rámce jsou: transparentnost, zabudovaná kvalita (built-in quality), sladění (alignment) a plnění programu. Při sdělování a dodržování těchto hodnot hrají důležitou roli vedoucí pracovníci (Knaster, 2020).

#### Transparentnost

Transparentnost buduje důvěru. Vizualizace veškeré relevantní práce je základ. Je důležité přijímat odpovědnost za chyby a omyly. Přiznejte chybné kroky a podpořte ostatní k tomu, aby se z nich poučili. Vytvářejte prostředí, kde jsou veškerá fakta a skutečnosti transparentní (Knaster, 2020). Bez otevřenosti vznikají rozhodnutí založená na nedostatku předpokladů a údajů (Leffingwell, 2018).

#### Built-in Quality – zabudovaná kvalita

Dodávání produktů a služeb nízké kvality má mnohem větší dopad ve škálované agilitě. Zabudované postupy v oblasti kvality zvyšují celkovou spokojenost zákazníků a zajišťují rychlejší dodávku hodnot. Bez zabudované kvality nelze dosáhnout cíle Lean-Agile,

kterým je získání co nejvyšší hodnoty v co nejkratším čase (Knaster, 2020). Zabudovaná kvalita zajišťuje, že každý inkrement dodaného řešení odráží určité standardy kvality. Kvalita se nepřidává až dodatečně (Leffingwell, 2018).

### **Alignment – sladění**

Sladění je důležité k tomu, aby bylo možné držet krok s neustálými změnami, konkurenčními silami či s týmy, které jsou geograficky rozptýlené. Alignment však neznamená přebírání vedení či zavedení kontrol. Je tomu právě naopak. Sladění umožňuje autonomii a decentralizované rozhodování. Díky tomu mohou ti, kteří dodávají hodnoty, přijímat lepší rozhodnutí (Leffingwell, 2018). Pokud jsou vývojové týmy a vedení sladěny na společném cíli a poslání, může tak veškerá energie směřovat k pomoci zákazníkovi či koncovému uživateli. Všichni jsou na stejné lodi a pracují na stejných cílech (Knaster, 2020).

### **Plnění programu**

Nic z toho, co je součástí SAFe® nemá význam, pokud týmy nedokážou realizovat a průběžně dodávat hodnoty. Proto se SAFe® intenzivně zaměřuje na fungující systémy a business výsledky (Leffingwell, 2018).

## **3.2.3 Role v prostředí SAFe®**

### **3.2.3.1 Business Owner – BO**

Business Owners. Jedná se o malou skupinu stakeholderů (obvykle tři až pět) ARTu, kteří mají zodpovědnost za správu, dodržování postupů a pravidel, za návratnost investic a za dodávané business výsledky celého vlaku (Knaster, 2020). Business Owner hraje klíčovou roli při pomoci ARTu dodávat produkty s co nejvyšší hodnotou. Odpovídají za pochopení strategických témat, která ART ovlivňují. Mají znalosti o aktuální situaci organizace a o toku hodnot (value streamu). Business Owner také ručí za dodržování korporátních pravidel a nastavených standardů (Leffingwell, 2018).

**Účastní se ceremonií:** PI Planning (Program Increment Planning), Inspect & Adapt a System Dema.

### **3.2.3.2 Product Manager / Product Management – PdM**

Product Manager je zodpovědný definici produktů, které mají být dodány zákazníkovi. Je zodpovědný za identifikaci potřeb zákazníka a fungování vývoje pomocí Kanbanu. Z programového backlogu vybírá nejdůležitější Features, které se budou vyvíjet v následujícím PI (Program Increment) (Knaster, 2020). Produktový manažer zastupuje zákazníka při komunikaci s týmem a naopak, zastupuje tým při komunikaci se zákazníkem. Určuje priority, plánuje features, ověřuje reakce zákazníků na dodané produkty, poskytuje týmu rychlou zpětnou vazbu a je zodpovědný za přípravu produktové roadmapy. Komunikuje s Business Ownery a dalšími klíčovými stakeholdery a spolupracuje s RTE (Release Train Engineer) a SA (System Architect) (Knaster, 2020).

**Účastní se ceremonií:** PI Planning, PO Sync, System Demo, PI System Demo a v neposlední řadě Inspect & Adapt

### **3.2.3.3 System Architect – SA**

System Architect působí v kontextu Agile Release Train, kde spolupracuje s agilními týmy a poskytuje technickou podporu týkající se subsystémů uvnitř ARTu. Úzce spolupracuje s business stakeholdery, týmy, zákazníky, dodavateli a dalšími zainteresovanými stranami při definování technologické infrastruktury a při rozpadu systémů na jednotlivé komponenty. System Architect poskytuje obecný pohled na architekturu řešení (Scaled Agile Framework, 2020).

**Účastní se ceremonií:** PI Planning, PI System Demo, System Demo a Inspect & Adapt

### **3.2.3.4 Release Train Engineer – RTE**

Release Train Engineer je servant leader, který facilituje realizaci programu, řídí rizika a vzájemné závislosti, pomáhá odstraňovat překážky podporuje neustálé zlepšování a učení. Je také koučem ARTu. Jeho hlavní odpovědností je facilitace některých ceremonií a procesů a pomoc při dodávání hodnoty (Knaster, 2020). Podporuje osobní rozvoj a angažovanost jednotlivců i celých týmů a je odpovědný za jejich správné fungování a efektivitu. Usiluje o neustálý rozvoj ARTu jako celku (např. přes ceremonii Inspect & Adapt). Podporuje spolupráci mezi jednotlivými agilními týmy, System týmem a System Architektem (SA) (Scaled Agile Framework, 2020).

**Účastní se ceremonií:** PI Planning, PO Sync, Inspect & Adapt, System Demo, PI System Demo a Scrum of Scrums

### **3.2.3.5 Product Owner – PO**

Product Owner je člen agilního týmu, který je zodpovědný za definici a prioritizaci User Stories a týmového backlogu s cílem zefektivnit plnění jednotlivých program aktivit. Je tedy zodpovědný za správu týmového backlogu a zajišťuje, aby jeho položky odrážely potřeby zákazníka či koncového klienta (Knaster, 2020). Má důležitou roli v kontrole kvality a jako jediný člen týmu je oprávněn určit, zda jsou jednotlivé User Stories hotové (Leffingwell, 2018). Product Owner rozvíjí konkrétní produkt nebo jeho část a snaží se zákazníkovi dodat co nejvyšší hodnotu. Spolupracuje s Product Managerem na prioritách Program Backlogu a na přípravě PI Planningu. Jeho úkolem je zastupovat zákazníka a jeho zájmy. Týmu je k dispozici po celý Sprint / Iteraci (Scaled Agile Framework, 2020).

**Účastní se ceremonií:** Iteration Planning, Iteration Review, Iteration Retrospective, Daily Stand-up a Refinement (Scaled Agile Framework, 2020).

### **3.2.3.6 Scrum Master – SM**

Scrum Master je servant leader a kouč agilního týmu. Jeho hlavní odpovědností je pomáhat samo-organizovanému a samosprávnému týmu dosahovat jeho stanovených cílů. Také pomáhá týmu vzdělávat se v metodách Kanban, Scrum, eXtreme Programming (XP) a má na starost dodržování dohodnutého agilního procesu. Jeho zodpovědností je odstraňování vnějších i vnitřních překážek, které by mohly tým při jeho práci ovlivňovat a podpora neustálého zlepšování a toku hodnoty. Hraje důležitou roli v plánování Program Incrementu (PI) (Knaster, 2020). Scrum Master je členem agilního týmu a sdílí odpovědnost za výkon týmu jako celku. Pomáhá Product Ownerovi a týmu pochopit a používat nástroje a techniky, tak aby přinášely so nejvyšší hodnotu (Leffingwell, 2018). Jeho úkolem je také chránit tým před tlakem z okolí, které by ho mohlo odvádět od jeho soustředěné práce (Scaled Agile Framework, 2020).

**Účastní se ceremonií:** PI Planning, System Demo, PI System Demo, Scrum of Scrums, Inspect & Adapt (Scaled Agile Framework, 2020).

**Facilituje ceremonie:** Backlog Refinement, Daily Stand-up, Iteration Planning, Iteration Review, Iteration Retrospective (Scaled Agile Framework, 2020).

### 3.2.3.7 Agile Team – Agilní tým

Agilní tým je multifunkční skupina s 5 až 11 členy, kteří definují, vytvářejí, testují a dodávají inkrement (přírůstek) hodnoty v krátkém časovém úseku (sprint či iterace). Agilní tým je základním stavebním kamenem celého agilního vývoje a ARTu. Tyto týmy jsou vysoce výkonné a upevňují kompetence používáním agilních principů a postupů (Knaster, 2020). Agilní týmy se samy organizují, sami řídí a je jejich úkolem dosahovat takových výsledků, které splňují potřeby a očekávání zákazníků a dalších stakeholderů. Součástí každého agilního týmu je jeden Scrum Master a jeden Product Owner (Scaled Agile Framework, 2020).

**Účastní se ceremonií:** Daily Stand-up, Iteration Review, Iteration Planning, Iteration Retrospective, PI Planning (Scaled Agile Framework, 2020).

### 3.2.4 Agile Release Train – ART

Agile Release Train, zkráceně ART či vlak, představuje dlouhodobé uskupení agilních týmů, které spolu s dalšími stakeholdery postupně vyvíjí, dodává a zajišťuje provoz jednoho nebo více řešení, a to vše pomocí stanovených Iterací / Sprintů v rámci Program Incrementu (PI). ART sjednocuje týmy, které mají společný technologický a business cíl (Leffingwell, 2018). ART je multifunkční (cross-functional) a disponuje všemi dovednostmi a schopnostmi (softwarovými, hardwarovými a dalšími), které jsou potřeba k definování, implementaci, testování a dodání nových systémových funkcionalit. Úkolem je ARTu je spravovat a neustále vylepšovat jeden či více produktů, které jsou pro každý ART stanoveny. Celý ART čítá přibližně 50 až 150 členů, kteří jsou celým svým úvazkem dedikováni na práci v daném ARTu. Ukazuje se, že seskupení ARTu nejlépe funguje v dlouhém období, tudíž by nemělo být jeho účelem dodání pouze jednoho produktu či služby. Při řízení ARTu se využívá metod Kanbanu či Scrumu (Scaled Agile Framework, 2020).



OBR. 16: AGILE RELEASE TRAIN - ART (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

### 3.2.4.1 Principy ARTu

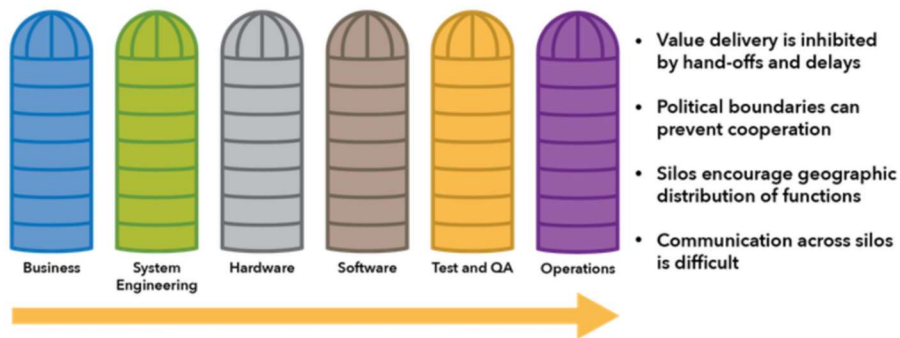
ART funguje v souladu se společnou misí a pomáhá řídit rizika a variabilitu, která jsou spojena s vývojem veškerých řešení (Knaster, 2020). Celý ART funguje na základě několika principů:

1. **Pevný jízdní řád** – vlak začíná podle daného, známého a spolehlivého jízdního řádu, který je dopředu určen. Pokud se Feature do stávajícího vlaku nevejde, přijde na řadu ve vlaku nastávajícím (Leffingwell, 2018).
2. **Nový inkrement každé dva týdny** – každý vlak dodává nové inkrementy každé dva týdny, každou iteraci či sprint. Ceremonie System Demo poté poskytuje mechanismus, který hodnotí funkčnost dodaného systému. Systém je integrace inkrementů ze všech týmů (Scaled Agile Framework, 2020).
3. **Pevně stanovený časový rámec Program Incrementu (PI)** – všechny týmy ARTu mají stejnou délku PI (obvykle je PI stanoveno na 8 až 12 týdnů) a stejné datum začátku a konce iterace (Scaled Agile Framework, 2020).
4. **Známa velocita** – ART může odhadnout, kolik nových features může být v následujícím PI dodáno (Leffingwell, 2018).
5. **Agilní týmy** – agilní týmy akceptují Agile Manifesto a principy a hodnotu rámce SAFe® (Leffingwell, 2018).
6. **Dedikování členové týmu** – lidé jsou na práci v ARTu dedikováni celým svým úvazkem (Leffingwell, 2018).
7. **Face-to-face PI Planning** – ART plánuje svou práci během pravidelných ceremonií Program Increment Planning – PI Planning (Scaled Agile Framework, 2020).
8. **Innovation and Planning (IP)** – IP iterace poskytují „buffer“ (časovou rezervu) pro odhadování (estimaci) položek ARTu, pro PI Planning, průběžné vzdělávání a pro dodatečnou práci na infrastruktuře ARTu (Scaled Agile Framework, 2020).
9. **Inspect & Adapt** – tato ceremonie se koná na konci každého PI. Předvádí a hodnotí se zde aktuální stav vyvinutých řešení (Leffingwell, 2018).
10. **Release on Demand** – vlaky mohou řešení nebo jeho prvky vydat kdykoliv, kdy je potřeba. Musí se však řídit stejnými kritérii jako při obvyklém dodávání (Leffingwell, 2018).

Na rozsáhlejších řešeních může spolupracovat i několik ARTů zároveň. Tomuto uskupení se pak říká Solution Train. Solution train má své vlastní ceremonie, jako například Solution Demo (Knaster, 2020).

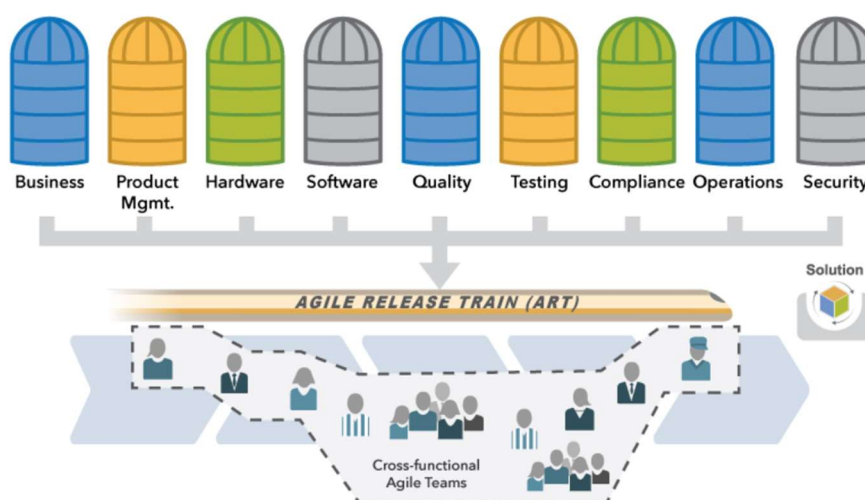
### 3.2.4.2 Organizační struktura ARTu

Vlak je v podstatě organizace uvnitř organizace. Má všechny lidi, kteří jsou potřeba v definici, dodání a provozu jednotlivých řešení. Tato struktura tak rozbíjí tradiční funkční složky, které mohou v organizaci existovat (Scaled Agile Framework, 2020).



OBR. 17: TRADIČNÍ FUNKČNÍ ORGANIZACE (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

V takové funkční organizaci (tradiční) pracují testeři s testery, vývojáři s vývojáři, architekti s architektky, tedy jednotlivé funkční celky pracují samostatně. Tato struktura však dodává hodnotu velmi pomalu, jelikož musí procházet jednotlivými celky popořadě. Namísto toho ART uplatňuje systémové myšlení a úzce se zaměřuje na dodanou hodnotu tak, aby vytvořil cross-functional organizaci, která je nastavena takovým způsobem, aby byl tok hodnoty od nápadu přes dodání až po provoz co nejrychlejší (Leffingwell, 2018).

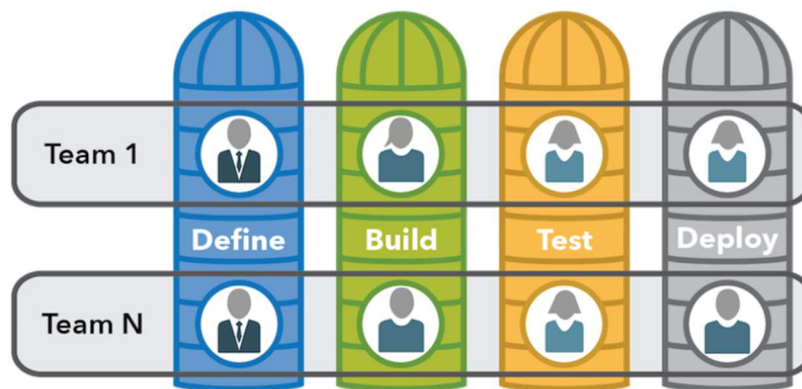


OBR. 18: MULTIFUNKČNÍ ART (ZDROJ: SCALED AGILE FRAMEWORK, 2020)



### 3.2.4.3 Agilní Tým jako základní jednotka ARTu

Jak již bylo dříve uvedeno, Agilní Tým je multifunkční uskupení 5 až 11 lidí, kteří mají na starosti definici, vývoj a dodávání inkrementů ve stanoveném sprintu či iteraci. Týmy SAFe® mají na výběr z několika agilních přístupů, které jsou primárně založeny na Kanbanu, Scrumu či XP. Jednotlivé agilní týmy mohou být buď technologicky zaměřené, tedy dodávají software, hardware či jejich kombinace anebo zaměřené na business (Scaled Agile Framework, 2020).



OBR. 19: MULTIFUNKNČÍ TÝM (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

#### Důležité týmové role

Každý agilní tým má vyhrazené členy, kteří pokrývají všechny potřebné role k vytvoření inkrementu s vysokou hodnotou a kvalitou v dané iteraci. Těmito rolemi jsou (Scaled Agile Framework, 2020):

- **Scrum Master** – servant leader pro tým, facilituje ceremonie, odstraňuje překážky, dohlíží na dodržování nastavených agilních principů.
- **Product Owner** – vlastník týmového backlogu, zastává roli zákazníka při komunikaci s týmem, prioritizuje práci a spolupracuje s Product Managementem na plánování a dodávání řešení.
- **Development tým** – dodává jednotlivá řešení

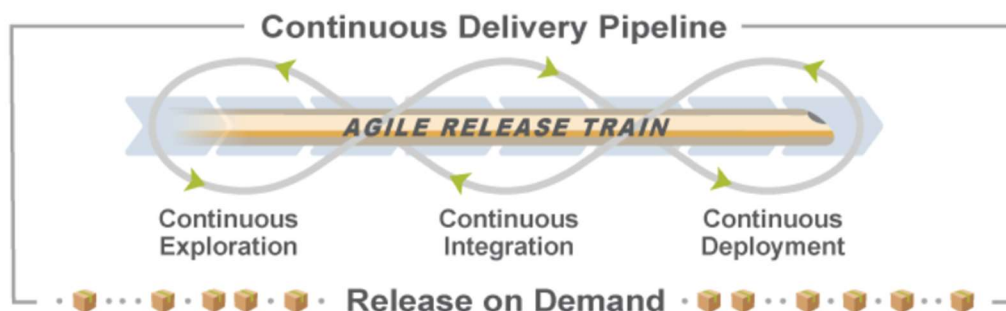
Úspěšné řízení vlaku pomáhají zajistit tyto program-level role:

- **Release Train Engineer** – RTE
- **Product Management** – PdM
- **System Architect** – SA
- **Business Owner** – BO
- **Zákazník / koncový uživatel**

### 3.2.4.4 Continuous Delivery Pipeline – koncept neustálého dodávání

Každý ART vytváří, udržuje a v některých případech také sdílí s ostatními ARTy *pipeline* (postup), která obsahuje prostředky a technologie potřebné k dodávání hodnoty, co nejvíce samostatným způsobem. Continuous Delivery Pipeline se skládá ze čtyř prvků: Continuous Exploration (nepřetržitý průzkum), Continuous Integration (nepřetržitá integrace), Continuous Deployment (kontinuální dodávání) a Release on Demand. První tři prvky se vzájemně doplňují tak, aby podpořily dodávání malých částí nových funkcionalit, které jsou následně dodávány trhu, podle jeho aktuální poptávky (Knaster, 2020).

*Continuous Delivery Pipeline* (CDP) představuje postupy, principy, činnosti a určitou automatizaci, které jsou potřeba k tomu, aby funkcionalita prošla od nápadu až po dodání hodnoty na základě potřeby koncového uživatele (Scaled Agile Framework, 2020).



OBR. 20: CONTINUOUS DELIVERY PIPELINE (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

#### Continuous Exploration – CE

CE podporuje neustálý výzkum a koordinaci toho, co by mělo být vytvořeno. Během toho jsou vzneseny nové myšlenky, které jsou následně definovány, upřesňovány a připravovány jako seznam prioritizovaných features v program backlogu. Během plánování CE jsou features přemísťovány do implementace, čím začíná proces Continuous Integration (Knaster, 2020).

#### Continuous Integration – CI

CI zavádí kvalitu do vývojového procesu tak, že průběžně integruje veškerou probíhající práci mnoha agilních týmů do jednoho řešení. Dokončená práce je poté předána do testování, které probíhá před validací v testovacím prostředí (Knaster, 2020).

### **Continuous Deployment – CD**

CD přebírá funkcionality z testovacího prostředí a dále je předává do produkce. V tomto okamžiku jsou jednotlivé funkcionality ověřovány a monitorovány za účelem zajištění jejich správného fungování. V produkčním prostředí pak funkcionality čekají, dokud organizace neurčí vhodný čas pro jejich dodání zákazníkům. Díky tomuto postupu může organizace v případě potřeby funkcionality opravit či změnit (Scaled Agile Framework, 2020).

### **Release on Demand – RoD**

RoD je schopnost poskytovat nové funkcionality zákazníkům či koncovým uživatelům buď postupně či na základě potřeb trhu a organizace. Díky tomu organizace vydávají funkcionality tehdy, kdy je pro to trh připraven (Scaled Agile Framework, 2020).

## **3.2.5 Product Management – Produktový management**

Již z popisu jednotlivých rolí rámce SAFe® víme, že Product Management je zodpovědný za to, co se vytvoří, jak je to definováno v produktové vizi, v roadmapě a novými features v backlogu. Dále produktový management odpovídá za definování a podporování vytváření žádoucích, fungujících a proveditelných produktů, které splňují požadavky a potřeby zákazníků (Knaster, 2020).

Role produktového management se škáluje se složitostí jednotlivých řešení. U některých řešení může produkt spravovat pouze jeden Product Manager (PdM). U jiných, komplexnějších řešení je potřeba tým produktových manažerů. Každé řešení prochází čtyřmi fázemi vývoje: zavedením, růstem, zralostí a úpadkem. Úkolem PdM je každé řešení těmito fázemi provést takovým způsobem, které přinese nejvyšší hodnotu (Scaled Agile Framework, 2020).

### **3.2.5.1 Odpovědnosti produktového managementu**

Odpovědnosti produktového managementu se rozdělují do pěti základních oblastí:

- Zkoumání trhů a zákazníků
- Spojení se zákazníkem
- Definice produktové strategie, vize a roadmapy
- Řízení a stanovení priorit v rámci ART backlogu
- Dodávání hodnoty

## **Zkoumání trhů a zákazníků**

Produktový management neustále prozkoumává kontext jednotlivých řešení a shromažďuje kvantitativní a kvalitativní poznatky o trhu a preferencích zákazníků. Tyto poznatky slouží jako podklad pro stanovení technické a business strategie a vytvářejí hypotézy, které podporují Continuous Delivery Pipeline .

Tato oblast zahrnuje činnosti (Scaled Agile Framework, 2020):

- Provedení primárního a sekundárního výzkumu
- Segmentace trhu
- Rozpoznání tržních trendů a událostí
- Pochopení potřeb koncového uživatele

## **Spojení se zákazníkem**

Product Management komunikuje přímo se zákazníky v průběhu celého životního cyklu jednotlivých řešení. Touto komunikací se zajišťuje, že jsou potřeby zákazníků již od začátku zahrnuty do strategie produktů, a že se následně promítají do jednotlivých dodaných řešení. Při vývoji těchto řešení se musí počítat s tím, že s neustále se měnícím trhem, se i zákazníkovi potřeby budou v průběhu vývoje měnit.

Tato oblast zahrnuje činnosti (Scaled Agile Framework, 2020):

- Aplikace zákaznický orientovaného myšlení
- Empatie se zákazníkem
- Používání Design Thinking
- Neustálé zapojování zákazníka do procesu

## **Definice produktové strategie, vize a roadmapy**

Produktový management hraje důležitou roli v propojování strategie a realizace při dodávání jednotlivých produktů. Zjištěné potřeby musí být přeneseny do konceptů řešení, které mohou být následně dodány agilním týmem.

Tato oblast zahrnuje činnosti (Scaled Agile Framework, 2020):

- Soulad strategie s business cíli
- Tvorba modelů, které dodávají hodnotu jak zákazníkovi, tak i organizaci
- Tvorba a sdílení vize
- Správa flexibilních roadmap

## **Řízení a stanovení priorit v rámci ART backlogu**

PdM zajišťuje tok práce v rámci stanoveného backlogu a dbá na to, aby pokaždé odrážel co nejvíce aktuální potřeby zákazníků.

Tato oblast zahrnuje činnosti (Scaled Agile Framework, 2020):

- Provádění a pomoc při definici a tvorbě features
- Stanovení priorit features pomocí WSJF (Weighted Shortest Job First)
- Akceptace features
- Účast na ART ceremoniích

## **Dodávání hodnoty**

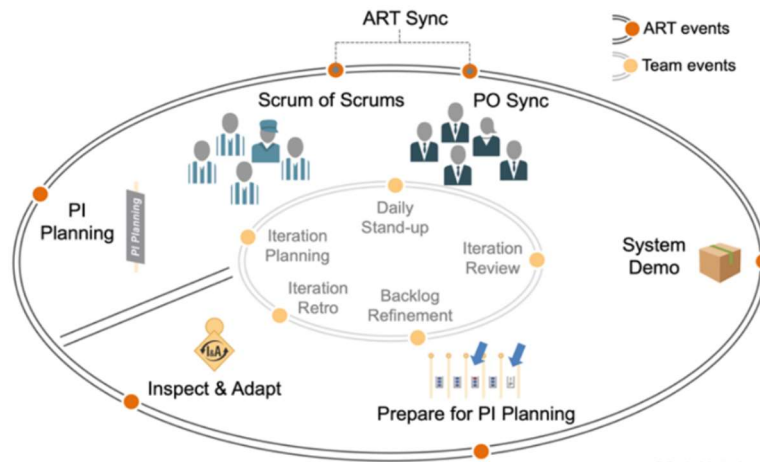
PdM používá Continuous Delivery Pipeline k dodávání hodnot podle aktuální situace na trhu. To znamená dodávání hodnoty několikrát denně, týdně nebo měsíčně. Může to však také znamenat, že se hodnoty dodávají kdykoliv, kdy jsou splněny potřeby zákazníků a jsou naplněny cíle podniku.

Tato oblast zahrnuje činnosti (Scaled Agile Framework, 2020):

- Spolupráce v rámci value streamu
- Zajištění úplnosti produktu
- Zajištění provozu
- Uvolňování hodnot na základě poptávky či vyžádání – Release on Demand
- Plnění business cílů

## 3.2.6 SAFe® Ceremonie

V případě realizace PI pro jeden ART, vytváří sekvence jednotlivých ceremonií uzavřený systém, který pomáhá udržovat celý vlak na správné cestě k požadovanému výsledku (Scaled Agile Framework, 2020).



OBR. 21: PŘEHLED SAFe® CEREMONIÍ (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

Na obrázku si můžeme všimnout, že se ceremonie dělí na dvě úrovně. První úrovní jsou ceremonie, kterých se účastní celý vlak. Patří sem ceremonie: Scrum of Scrums (SoS), PO Sync, System Demo, PI Planning, Inspect & Adapt a v neposlední řadě příprava na PI Planning. Další úrovní jsou ceremonie, kterých se účastní jednotlivé týmy. Jsou to ceremonie: Daily Stand-Up (DSU), Iteration Planning, Iteration Review, Iteration Retrospective a Backlog Refinement.

### 3.2.6.1 Ceremonie na úrovni ARTu

#### PI Planning – Program Increment Planning (PIP)

PI Planning je ceremonie založená na pravidelnosti, která slouží jako základní stavební kámen celého ARTu a synchronizuje všechny agilní týmy ke společné vizi a cílům. PIP je facilitován Release Train Engineerem (RTE) a účastní se ho všichni členové ARTu. Obvykle trvá dva dny a probíhá v rámci IP (Innovation and Planning) iterace, která se zavádí právě pro účely plánování, úprav funkcionalit a případných rezerv (Knaster, 2020). PI Planning je pro fungování rámce SAFe® zcela zásadní, samotný SAFe® říká (Scaled Agile Framework, 2020): „If you are not doing it, you are not doing SAFe®.“

PI Planning má nastavenou agendu, která zahrnuje prezentaci business souvislostí a vize, po které poté následují týmové plánovací bloky, kde jednotlivé týmy vytvářejí plány a cíle pro další PI. Standardně trvá dva dny, ale v případě geograficky distribuovaných týmů může být trvání této ceremonie delší (Scaled Agile Framework, 2020).



OBR. 22: AGENDA PI PLANNINGU (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

**Účastníci:** RTE, SM, PO, PdM, SA, BO, Agilní tým

### Scrum of Scrum – SoS

Ceremonii Scrum of Scrums facilituje Release Train Engineer na týdenní (nebo podle potřeby častější) bázi. SoS pomáhá koordinovat závislosti vně ARTu a poskytuje přehled o pokroku a o překážkách, které se mohly vyskytnout. Účelem této ceremonie je zhodnocení pokroku při dosahování milníků a cílů (PI Objectives). I tato ceremonie je časově omezena na trvání 30–60 minut. Navazuje na ní schůzka, na které mohou zůstat účastníci, kteří potřebují vyřešit konkrétní problémy či odpovědět na některé dotazy (Knaster, 2020).

**Účastníci:** RTE, SM (z každého agilního týmu), další stakeholdeři (pokud je třeba)

### PO Sync

Stejně jako ceremonie SoS i tato událost se obvykle koná jednou týdně (nebo častěji podle potřeby). Její trvání je také nastavené na 30-60 minut a navazující schůzkou týkající se řešení problémů. Někdy jsou ceremonie SoS a PO Sync spojeny do jedné, která se

nazývá **ART Sync** (Knaster, 2020). Účelem této ceremonie je získat přehled o aktuální stavu a progresu stanovených cílů PI či diskutovat o vyskytujících se problémech či příležitostech (Scaled Agile Framework, 2020).

**Účastníci:** PO, PdM, RTE, další stakeholderři (pokud je třeba)

### **System Demo**

Na ceremonii System Demo prezentují Product Owneři a agilní týmy vyvinuté features. Tato ceremonie poskytuje klíčovou zpětnou vazbu od stakeholderů (Business Owners, sponzoři, vedení vývoje a zákazník) ohledně efektivity a použitelnosti vyvíjeného systému. Tito stakeholderři poskytují informace o vhodnosti vyvíjeného řešení. Ceremonie System Demo se koná pravidelně jednou za dva týdny po skončení iterace (Knaster, 2020).

**Účastníci:** PO, PdM, RTE, SA, BO, agilní tým, další relevantní stakeholderři

### **Inspect & Adapt**

Během této ceremonie probíhající na konci každého Program Incrementu (PI), vyhodnocují stakeholderři aktuální stav dodaného řešení. Cílem je také identifikace případných problémů, které mohou negativně ovlivnit dodávky systémů a následné navržení řešení. Tato řešení jsou následně aplikována v následujícím PI a jejich cílem je zlepšení celého vývojového procesu. Trvání této ceremonie je stanoveno na 4 hodiny (Scaled Agile Framework, 2020).

**Účastníci:** Agilní tým, SM, PO, RTE, SA, PdM, BO, další relevantní stakeholderři

## **3.2.6.2 Ceremonie na úrovni týmu**

### **Daily Stand-Up – DSU**

Již z názvu lze pochopit, že se tato ceremonie koná každý den. Účelem této ceremonie je pochopení, jak to tom týmu je, upozorňování na problémy a případné získání pomoci od ostatních členů. Každý člen má za úkol popsat co dělat včera, na čem bude pracovat dnes a na jaké problémy narazil. Tato ceremonie by neměla trvat déle než 15 minut (Knaster, 2020).

**Účastníci:** členové agilního týmu, SM (facilituje ceremonii) a PO (dobrovolně)



## Backlog Refinement

Hlavním cílem této ceremonie je průběžná aktualizace týmového backlogu a zároveň příprava správně popsaných User Stories pro nadcházející iteraci. User Stories musí být popsané tak, aby jim jednotlivý členové týmu rozuměli. Tato ceremonie je opakována do té doby, dokud nejsou položky backlogu dostatečně připraveny pro Iteration Planning ceremonii. Tato ceremonie nemá jasně danou kadenci ani časové ohraničení. Backlog Refinement pomáhá odhalit problémy s aktuálním plánem, které se následně diskutují na ceremonii ART Sync (Scaled Agile Framework, 2020).

**Účastníci:** agilní tým, SM (facilituje ceremonii) a PO

## Iteration Planning

Hlavním cílem této ceremonie je příprava a plánování nadcházející iterace na základě vstupů z ceremonie Backlog Refinement. Tato ceremonie by neměla trvat déle než 4 hodiny pro iterace, které mají trvání dva týdny (Scaled Agile Framework, 2020). Během tohoto plánování tým vykovává tyto činnosti (Knaster, 2020):

- Přezkoumává, upřesňuje a odhaduje velikost User Stories, které poté předkládá Product Ownerovi
- Definuje akceptační kritéria
- Pokud je potřeba, dělí větší User Stories na menší celky
- Určuje, co je tým schopen dodat v nadcházející iteraci
- Zavazuje se ke krátkodobým cílům

**Účastníci:** agilní týmy, SM, PO, další relevantní stakeholdeři

## Iteration Review

Iteration Review představuje ukončení iterace či Sprintu. Agilní tým na této ceremonii předává Product Ownerovi a stakeholderům výstupy z proběhlé iterace. Primární cíl této ceremonie je získat feedback. Product Owner pak jednotlivé výstupy hodnotí a následně akceptuje či vrací zpět k provedení úprav či dalších změn. Koná se na konci každé iterace a trvá mezi 1 až 2 hodinami (Scaled Agile Framework, 2020).

**Účastníci:** agilní tým, SM, PO, stakeholdeři, kteří mají na dodaných výstupech určitý zájem

## **Iteration Retrospective**

Na Iteration Retrospective tým vyhodnocuje svůj dosavadní postup a identifikuje nové problémy a jejich příčiny. Tak jako problémy identifikuje tým i silné stránky a věci, které se mu během iterace povedly. Pravidelná reflexe je způsob jak zajistit, aby se fungování týmu neustále zlepšovalo. Před zahájením dalšího plánovacího cyklu je backlog upraven tak, aby zahrnoval všechna rozhodnutí z Iteration Review a Retrospective.

**Účastníci:** agilní tým, SM (facilituje ceremonii), PO

# **PRAKTICKÁ ČÁST**

## 4 Assessment používání rámce SAFe® ve vybrané organizaci

Rámec SAFe® ať už v jakékoliv formě se v dnešní době snaží implementovat stále více organizací z mnoha různých odvětví. Taková implementace je však složitá a nezaručuje okamžité výsledky. Při implementaci je důležité dodržování nastavených standardů, praktik a postupů.

Praktická část diplomové práce bude proto zaměřena provedení vyhodnocení používání rámce SAFe® ve vybrané organizaci, která bude pro účely práce anonymizována. Assessment<sup>1</sup> bude prováděn na základě postupů a metod, které mi byly poskytnuty poradenskou společností, která se zabývá pomocí při implementaci agilních přístupů do organizace jejích klientů. I tato organizace bude pro účely této práce nejmenována.

Veškeré metodologické náležitosti (struktura assessmentu, list hodnotitele a závěrečná zpráva) praktické části byly poskytnuty poradenskou (konzultingovou) společností, která tyto assessmenty prodává jako svůj produkt ostatním společnostem.

### 4.1 Cíl praktické části

Cílem je vytvoření assessmentu používání rámce SAFe®. Účelem tohoto assessmentu je zjistit, jakým způsobem organizace používá znalostní rámec Scaled Agile Framework® a následně na základě zjištěných skutečností navrhnout vhodná doporučení. Z teoretické části vyplývá důležitost správného nastavení jednotlivých rolí a ceremonií Agile Release Trainu (ART), proto se bude assessment zaměřovat na oblasti ceremonií, rolí a ARTu.

### 4.2 Výzkumné metody

Ke sběru informací o aktuální stavu vybrané organizace bylo použito několik výzkumných metod:

- Návštěva ceremonií na úrovni ARTu
- Osobní rozhovory s členy ARTu
- Studium potřebné dokumentace

---

<sup>1</sup> Assessment = vyhodnocení, hodnocení určité skutečnosti

### **4.2.1 Návštěva ceremonií na úrovni ARTu**

Z teorie víme, že mezi ceremonie na úrovni ARTu patří Program Increment (PI) Planning, Scrum of Scrums (SoS), PO Sync, System Demo a Inspect & Adapt (I&A). Všechny tyto ceremonie byly v průběhu assessmentu navštívené. Návštěva probíhala dvěma formami:

- Osobní návštěva, kdy byl hodnotitel (assessor) přítomen na dané ceremonii
- Hodnotitel se na ceremonii připojil on-line

Hodnotitel se všech ceremonií účastnil pouze pasivně a žádným způsobem do jejich průběhu nezasahoval. Sledoval, jak jednotlivé události probíhají, jak jsou účastníci aktivní, jak je ceremonie vedena, zda odpovídají tomu, jak jsou ceremonie popsány ve znalostním rámci SAFe® a zda slouží stanovenému účelu. V průběhu ceremonií si hodnotitel zapisoval své poznatky (jak negativní, tak pozitivní), které se následně přepisují do listu hodnotitele.

### **4.2.2 Osobní rozhovory s členy ARTu**

Během assessmentu proběhlo několik osobních rozhovorů s různými členy ARTu, kteří zastávají některé role popsané v rámci SAFe®. Děly se čtyři rozhovory se Scrum Mastery, Product Ownery, Business Ownery a System Architektem. Z důvodu různých oblastí odpovědnosti a odlišných činností vybraných rolí, nebyla pevně stanovena struktura jednotlivých rozhovorů. Struktura se odvíjela od toho, s kým byl daný rozhovor prováděn. Každý rozhovor měl trvání 2 hodiny, během kterých se probíralo fungování dané role v rámci ARTu. Například v případě rozhovoru s členem zastávajícím roli Product Ownera se rozhovor zaměřoval na dodávání co nejvyšších hodnot zákazníkovi a správné fungování backlogu. U Scrum Mastera se rozhovor zaměřoval spíše na fungování týmu a nastavení jednotlivých ceremonií.

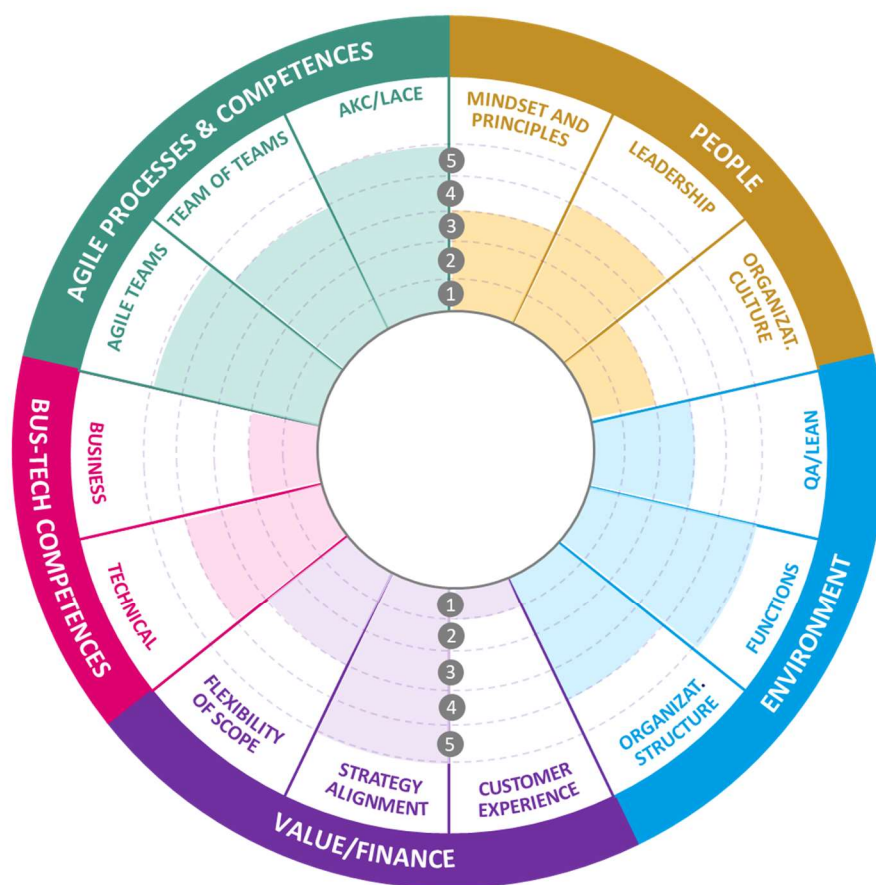
### **4.2.3 Studium potřebné dokumentace**

Studium dokumentace se týkalo veškerých dokumentů klíčových pro fungování daného ARTu. Podstatou studia dokumentace bylo zjištění aktuálního stavu nastavení vlaku jako takového. Samotné studium se týkalo dokumentací, které obsahovaly popis a nastavení vlaku, jednotlivých rolí, zápisy z ceremonií ART Sync (Scrum of Scrums a PO Sync), PI Planning a jeho vstupy, výstupy a dokumenty, které se v průběhu plánování PI využívaly, jak vypadá Kanban tabule (board), backlogy týmů vně vlaku a mnoho dalších skutečností, které ukazují jak hodnocený ART pracuje.

## 4.3 Jak vypadá Assessment?

Prostřednictvím assessmentu se zkoumá správné používání praktik a postupů rámce SAFe® a posuzuje se nastavení fungování těchto praktik v rámci Agile Release Trainu jako celku. Assessment také vyhodnocuje organizační kulturu, jak jednotlivé týmy dodávají hodnotu, jaký mindset má společnost nastavený a mnoho dalších skutečností. Díky assessmentu můžeme identifikovat praktiky a postupy, které vybraná organizace provádí špatně, či je má špatně nastavené. Assessment však také ukazuje všechno, co dělá organizace dobře a jaké jsou její silné stránky, není jen o chybách a slabých stránkách. Assessment sleduje nastavení 5ti oblastí, přičemž se každá oblast dále ještě dělí na několik podoblastí:

- Agile Processes & Competencies
- People
- Environment
- Value / Finance
- Business and Technical Competencies



OBR. 23: SAFe® ASSESSMENT (ZDROJ: PORADENSKÁ FIRMA)

## Agile Processes & Competencies

Tato oblast se úzce zaměřuje na způsob, jakým organizace aplikuje jednotlivé praktiky, postupy a metodiky znalostního rámce Scaled Agile Framework®. Oblast je rozdělena do dalších tří podoblastí:

- **Agile Teams** – Tato podoblast vyhodnocuje zralost agilního týmu ve fungování agilních procesů. Zaměřuje se na fungování událostí (plánování, retrospective, atd.), na nastavení klíčových agilních rolí v týmu (PO, SM, Vývojový tým) a na způsob práce s jednotlivými artefakty (backlog, user stories, features...).
- **Team of Teams** – Vyhodnocuje zralost Agile Release Train jako celku. Zaměřuje se na fungování ceremonií (PIP, SoS, PO Sync, atd.), na nastavení klíčových agilních rolí v týmu (RTE, PdM, SA) a na způsob práce s jednotlivými artefakty (epics, features...).
- **AKC (Agilní kompetenční centrum)** – Vyhodnocuje způsob fungování oddělení/týmu, který má v organizaci na starost zavádění a rozvoj agility. Zaměřuje se na nastavení procesů, jako podpora agilních implementací a streamů v organizaci, monitorování rozvoje agility, držení a poskytování best practices či zajištění klíčových rozvojových kompetencí.

## People

Tato oblast se zaměřuje na způsob, jakým je organizace vedena, jak má nastavený agilní mindset a principy, a jaká je její organizační kultura.

- **Mindset a principles** – Vyhodnocuje, zda je mindset lidí v organizaci/týmu kompatibilní s agilním způsobem práce. Zaměřuje se především na přístup k těmto hodnotám: důvěra, ownership (zodpovědnost za vlastní výsledky) a proaktivita, sebe-organizace, způsob spolupráce, způsob přístupu k řešení problémů, učení se nových věcí.
- **Leadership** – Vyhodnocuje způsob práce manažerů v agilním prostředí a ověřuje, zda je kompatibilní s agilním způsobem práce (servant leadership). Zaměřuje se na to, zda manažeři jdou svým podřízeným příkladem v agilnímu způsobu práce, zda jsou schopni opravdu delegovat a zda jednají v souladu s agilní vizí společnosti a tuto vizi dále šíří.
- **Organizational Culture** – Vyhodnocuje kulturu v organizaci nebo pouze některé její části. Zaměřuje se především na přístup ke změnám a na způsob, jak spolu lidé komunikují a sdílí informace.

## Environment

Tato oblast se soustředí na organizační strukturu společnosti, její jednotlivé funkce a sleduje, jak organizace řídí kontrolu kvality.

- **Organizational Structure** – Vyhodnocuje vztah klasické liniové struktury vůči projektům a agilnímu prostředí. Tj. především efektivitu maticové struktury z pohledu procesů. Posuzuje štíhlost organizace a především se zaměřuje na schopnost přenášet rozhodovací pravomoci na úroveň týmů agilních projektů, zatímco management rozhoduje především o prioritách.
- **Functions** – Vyhodnocuje fungování jednotlivých podpůrných oddělení ve vztahu k agilním týmům. Zkoumá především to, jakou službu poskytuje agilním týmům ve vztahu k dodavatelům oddělení Legal, jak HR rozvíjí agilní kulturu a pomáhá shánět týmům správné lidi a jak IT poskytuje potřebnou techniku a technologie.
- **QA (Quality Assurance) / Lean** – Vyhodnocuje fungování organizace z pohledu kontroly a neustálého zvyšování kvality. Zaměřuje se na schopnost organizace v průběžném zvyšování kvality práce agilních týmů díky práci s lessons (zkušenosti).

## Value / Finance

Tato oblast se zaměřuje na to, jak je organizace schopna se přizpůsobit neustále se měnícím podmínkám na trhu a jak je schopna získávat od svých zákazníků zpětnou vazbu.

- **Scope<sup>2</sup> Flexibility** – Vyhodnocuje, jak rychle je organizace, ART, či tým schopen reagovat na změny priorit případně jaká je schopnost přizpůsobit strategii novým podmínkám na trhu.
- **Strategy Alignment** – Vyhodnocuje pružnost při změnách v portfoliu<sup>3</sup> organizace. Hodnotí, jak je portfolio navázáno na jednotlivé value streamy společnosti a jak pracuje s value a business cases<sup>4</sup> na úrovni jednotlivých projektů.
- **Customer Experience** – Vyhodnocuje schopnost organizace získávat rychlý a pravidelný feedback od koncového zákazníka a systémově a rychle tento feedback implementovat do služeb a produktů organizace.

---

<sup>2</sup> Scope = rozsah či rámec. Takto se označuje vše, co je v určitém projektu zahrnuto. (zdroj: pmconsulting.cz)

<sup>3</sup> Portfolio = seznam všech projektů, které zrovna probíhají (zdroj: easyproject.cz)

<sup>4</sup> Business case = používá se kobjektivě veškerých přínosů projektu a zdůvodňuje jeho realizaci (zdroj: projectman.cz)



## Business and Technical Competencies

Tato oblast se zaměřuje na PdM a jeho schopnost obstát v konkurenčním prostředí a vyhodnocuje schopnost technických týmů dodávat hodnotu.

- **Business Competencies** – Vyhodnocuje schopnost PdM obstát v konkurenčním prostředí trhu. Zaměřuje se na porozumění trhu, schopnost inovace a obecně na způsob produktového myšlení.
- **Technical Competencies** – Vyhodnocuje schopnost technických dodávat efektivně hodnotu.

### 4.3.1 Assessor Sheet – List hodnotitele

Veškeré poznatky (findings), které byly během návštěv jednotlivých ceremonií, osobních rozhovorů a také během studia potřebné dokumentace identifikovány, se následně zapisují do listu hodnotitele (assessor sheet).

#### Team of Agile Teams

Competence		Findings	Recommendations
<b>Ability to processes ART Ceremonies</b>			
PI Planning, Inspect and Adapt	Critical		
	Major		
	Minor		
System Demo, Scrum of Scrum, PO sync, IP Iteration, and others	Critical		
	Major		
	Minor		
<b>ART Roles execution</b>			
ART role (PO, SM, SA, PdM, RTE, tým)	Critical		
	Major		
	Minor		
Business Owner	Critical		
	Major		
	Minor		
<b>Product Management Execution</b>			
Product Management	Critical		
	Major		
	Minor		
<b>Others</b>			
Vize a Roadmapa	Critical		
	Major		
	Minor		

OBR. 24: ASSESSOR SHEET (ZDROJ: VLASTNÍ ZPRACOVÁNÍ)

Z důvodu komplexnosti agilního assessmentu se bude pro účely této práce provádět vyhodnocení pouze oblasti **Agile Processes & Competencies** s užším zaměřením na podoblast **Team of Teams** tedy ART jako celek. Cílem bude vyhodnotit správnost fungování Agile Release Trainu a ověřit, zda využívá základní principy škálované agility. Bude vyhodnocováno, jakým způsobem má vlak nastaveny ART ceremonie a jakým způsobem fungují jednotlivé role. V neposlední řadě bude také analyzováno postavení role Product Managementu.

List hodnotitele je tak rozdělen na jednotlivé kategorie, které se v průběhu celého assessmentu sledují a následně vyhodnocují. Každá kategorie je obsažena sloupec, kam hodnotitel zaznamenává identifikované poznatky (findings), ke kterým se následně navrhuje vhodná doporučení (recommendations) ke zlepšení. Každá kategorie pak dělí jednotlivé poznatky dle závažnosti na: kritické poznatky (critical), závažné poznatky (major) a drobné poznatky (minor).

Dále se každá sledovaná oblast hodnotí na číselné škále od 1 do 5:

1. **Novice – začátečník (trying agile)** – ART a agilní tým jsou na začátku využívání agilních metod. Nezná je, případně se s nimi potkal jen okrajově. ART ani tým neorientuje v základních ceremoniích, neumí pracovat s backlogem a chybí mu základní školení (Scrum, SAFe®...). Nejsou nastavena týmová pravidla a chybí úvodní setup ARTu.
2. **Known – pokročilý začátečník (using agile)** – ART se snaží aplikovat SAFe® ceremonie a principy, často je tu silný vliv manažera, který rozděluje konkrétním lidem práci. Tým si sám neurčuje, na čem a kdo bude pracovat, a tím nepřebírá odpovědnost za výsledky týmu.
3. **Proficient – pokročilý (doing agile)** – ARTu a agilnímu týmu se podařilo zavést agilní metody (např. Scrum, Kanban...) a začíná využívat přístupy tohoto přístupu. Tým si rozhoduje o své práci, je proaktivní a pracuje na svém rozvoji. I přesto je tu prostor pro další zlepšování.
4. **Expert (being agile)** – ARTu není co vytknout. Ve všech oblastech funguje dobře a stal se příkladem pro další ARTy a týmy. Neustále však pracují na svém zlepšování. Zákazník je spokojený s dodávanými produkty.
5. **Master (master agile)** – ARTu/týmu není co vytknout. Sdílí své znalosti s okolím.

### 4.3.2 Závěrečná zpráva

Po provedení assessmentu, identifikaci problémů a následném navržení vhodných doporučení je potřeba všechny tyto skutečnosti a informace zaznamenat do závěrečné zprávy, která se po skončení assessmentu zasílá k zákazníkovi, který si tento assessment objednal. Závěrečná zpráva obsahuje několik částí, které dávají zákazníkům celkový obraz o fungování jejich metod.

První část je **manažerské shrnutí** (management summary), které shrnuje konkrétní poznatky identifikovány a zaznamenány do listu hodnotitele (assessor sheet). Assessment však není jen o problémech, může nám také ukázat, co dělá daná organizace dobře. Tudíž by tyto skutečnosti měly být také v manažerském shrnutí obsaženy. Další část obsahuje nejdůležitější **klíčová doporučení** (key recommendations), která vzešla z jednotlivých poznatků. Každé doporučení by mělo mít svůj krátký popis. Závěrečná zpráva také poukazuje na **silné stránky** (building blocks), a co dělá organizace v oblasti agility a implementace znalostního rámce SAFe® správně.

## 4.4 ART Assessment

### 4.4.1 Popis organizace

Organizace, kde se assessment provádí, je nadnárodní korporací s pobočkami v některých zemích Evropy, Afriky a blízkého východu. Assessment je prováděn na jedné z několika stovek poboček v dané zemi. Z důvodů anonymizace nebude jméno společnosti zmiňováno.

### 4.4.2 Harmonogram assessmentu

Jak již bylo v úvodu praktické části zmíněno, sběr dat potřebných k provedení assessmentu probíhal třemi způsoby: návštěvou ceremonií, studiem dokumentace a rozhovory s členy ARTu. Celý assessment se sběrem veškerých dat, následným vyhodnocením poznatků, navržením vhodných doporučení a sepsáním závěrečné zprávy měl trvání tři měsíce.

Harmonogram sběru potřebných dat pak vypadal následovně:

1. Potvrzení assessmentu s klíčovými lidmi a stakeholdery ARTu
2. Zajištění přístupu na jednotlivé ceremonie – zařizuje Release Train Engineer (RTE)
3. Zpřístupnění ARTových a týmových artefaktů – Program Backlog, Program Board, Týmový Backlog...
4. Návštěva a náslech ceremonií před PI Planningem – PO Sync, Scrum of Scrums, System Demo – zajišťuje RTE
5. Návštěva a náslech PI Planningu a účast na retrospektivě – zajišťuje RTE
6. Návštěva a náslech ceremonií po PI Planningu – PO Sync, Scrum of Scrums, Inspect & Adapt – zajišťuje RTE
7. Rozhovory (interviews) a sběr zpětné vazby s vybranými členy Agile Release Trainu

## 4.4.3 Výstupy assessmentu a navrhovaná doporučení

### 4.4.3.1 Ceremonie PI Planning a Inspect & Adapt

**Poznatek 1:** Ceremonie PI Planning probíhá ve většině případech on-line formou

**Kategorie:** negativní, kritický poznatek

Používání on-line formy při provádění ceremonie PI Planningu je pozůstatkem covidové pandemie, kdy to jinou formou nešlo. Pandemie přešla, ale zvyk komunikace prostřednictvím on-line platforem přetrvává. Lidi si na tento způsob zvykli a do jisté míry jim i vyhovuje.

Avšak používání on-line formy při pořádání této ceremonie se nedoporučuje, jelikož se celá podstata PI Planning zakládá na osobní (face-to-face) účasti všech členů ARTu. Používání on-line formy se používá v případě, kdy je ART či Solution Train rozprostřený do více měst, států či v některých případech dokonce světadílů. Koordinace PI Planningu v ARTu, který je geograficky distribuovaný, je však náročnější, složitější a i samotné trvání této ceremonie může být delší (Knaster, 2020).

Samotný rámec SAFe® přebírá princip face-to-face komunikace z Agilního Manifesta, které říká: „*The most efficient and effective method of conveying information to and within development team is a face-to-face conversation.*“ V překladu tedy: „Tou nejúčinnější a nejefektivnější metodou sdílení informací do a vně vývojového týmu, je face-to-face konverzace (Manifesto for Agile Software Development, 2001).“

Hlavní nevýhodou on-line PI Planningu jsou technologie. Všichni víme, jak probíhaly on-line hovory přes různé komunikační platformy. Pokaždé se stalo, že někomu něco nefungovalo (slabé internetové připojení, komunikační platforma nefungovala, tak jak měla, počítač se zasekl a mnoho dalších). Je za potřebí dalších nástrojů, které pomáhají týmům ve vzájemné spolupráci.

#### **Navrhované doporučení:**

Doporučuje se provádět ceremonii PI Planningu co nejvíce on-site (všichni členové ARTu se účastní osobně). On-line formu používat co nejméně to jde, jen v případech, kdy je to nezbytně nutné a klíčové osoby se nemohou této ceremonie osobně zúčastnit. I když se díky covidové krizi začaly používat softwary pro on-line spolupráci a ukazují se být do jisté míry efektivní a účinné, nic nemůže nahradit osobní spolupráci na vytváření řešení.

**Poznatek 2: Není jednoznačný přehled o tom, co bylo během Program Incrementu (PI) dodáno. Zároveň není jasné, zda byly plánované features dodány ve stanoveném čase**

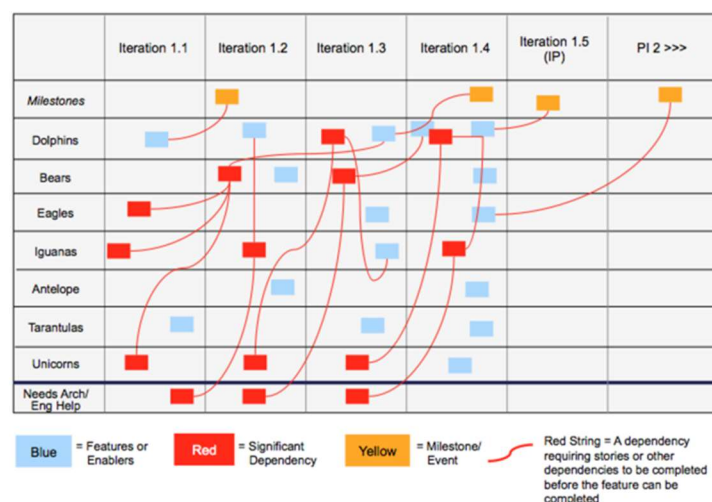
**Kategorie: negativní, závažný poznatek**

Každý Program Increment (PI) začíná ceremonií PI Planning, na které týmy odhadují, co bude během následujícího PI dodáno a na základě toho odhalují a upozorňují na závislosti, které by mohly mezi vývojovými týmy vzniknout. Jedním z výstupů PI Planningu jsou stanovené PI Objectives, kterých chce tým a s tím i celý ART v nadcházejícím PI dosáhnout. Jednotlivé týmy během PI Planningu plánují, jaké všechny features se zavazují dodat (Knaster, 2020).

### Navrhované doporučení:

Aby se zamezilo nejasné a nepřehledné situaci, kdy není jasně viditelné, co bylo během PI dodáno a zda byly jednotlivé features dodány v plánovaném čase, doporučuje se mnohem více využívat Program Board, který je jedním z výstupů PI Planningu.

Program Board (PB) pomáhá vizualizovat jednotlivé features, odhalovat jejich dependence (závislosti, např. na dodání řešení jiného týmu) a na základě toho hledat způsoby, jak tyto dependence co nejvíce eliminovat. PB také upozorňuje na dodací lhůty plánovaných features a na důležité termíny (milníky), na které nesmí agilní tým v průběhu iterace zapomenout (Scaled Agile Framework, 2020). Je důležité, aby byl Program Board co nejvíce komunikován mezi všechny agilní týmy, které se nachází v Agile Release Trainu. Díky viditelnému plánu dodávaných features, jejich dependencí a důležitých termínů roste transparence a jasná představa o tom, co dodáno bylo a co naopak dodáno nebylo. Agilní tým tak může lépe a s větší přesností plánovat další Program Incrementy (PI).



**OBR. 25:** PROGRAM BOARD (ZDROJ: SCALED AGILE FRAMEWORK, 2020)

### **Poznatek 3: Někteří členové agilních týmů nejsou přítomni na PI Planningu**

#### **Kategorie: negativní, závažný poznatek**

Při účasti na ceremonii PI Planning jsme si všimli, že zde chybí někteří členové agilních týmů, což má za následek, že se při plánování těžko odhadují kapacity na další PI, dle kterých se tým rozhoduje, ke kolika dodávaným features se může zavázat.

PI Planning je základním stavebním kamenem celého škálovatelného rámce SAFe® a sladí veškeré agilní týmy k jednotné vizi a cílům. Bez účasti některých agilních týmů (development tým, Scrum Master a Product Owner) se těžko stanovují cíle (objectives) nadcházejících PI, kterých se dosahuje pomocí plánování features. Jedním z předmětů agendy je Team Breakouts, které staví na aktivní účasti všech členů ARTu, tudíž i jednotlivých agilních týmů. V průběhu tohoto plánovacího bloku týmy odhadují svou kapacitu pro nadcházející iterace celého PI a určují položky backlogu, které budou potřeba k vývoji a dodání jednotlivých features. Po tomto následuje další plánovací blok, kde individuální agilní týmy představují návrh PI plánů, které jsou hodnoceny Business Ownerem. Ten může mít k plánům výhrady či připomínky, které tým musí opravit (Knaster, 2020).

Jak si můžeme všimnout, agilní tým je klíčovým účastníkem všech aktivit, které se během PI Planningu dějí, tudíž je jeho absence na něm velkým problémem. Všechny aktivity, činnosti a různá hlasování o schvalování finálního PI plánu je na agilním týmu závislé. Již v předchozím poznatku se zmiňují dependence týmů na sobě, absencí jednoho z agilních týmů vlaku nedocílíme zmapování všech možných závislostí. Důsledkem toho nelze vytvořit přesný plán, který by odpovídal kapacitám, cílům a potřebám všech týmů.

#### **Navrhované doporučení:**

Doporučení k tomuto poznatku je na první pohled zřejmé. Release Train Engineer (RTE) jako facilitátor ceremonie PI Planning by měl dbát na to, aby se všichni členové agilního týmu aktivně podíleli na veškerých aktivitách v průběhu PI Planningu. Účast všech je povinná a tato povinnost by se měla dodržovat. Lze toho docílit tím způsobem, že se vytvoří seznam všech lidí, kteří jsou součástí daného ARTu a bude se kontrolovat, zda jsou všichni na PI Planningu přítomni, ale i zde platí výjimka neúčasti v případech, kdy se člen týmu ceremonie účastnit nemůže.

## **Poznatek 4: Na PI Planningu chybí klíčoví stakeholderi, kteří mají důležité požadavky na vývoj řešení**

**Kategorie: negativní, kritický poznatek**

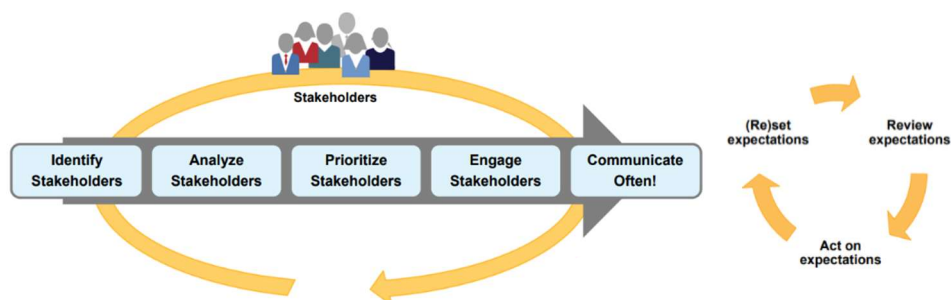
Stakeholderi, tak jako agilní tým, jsou klíčovým účastníkem ceremonie PI Planning. Na začátku PI Planningu vyjadřují stakeholderi své potřeby a požadavky, podle kterých se agilní týmy řídí při vytváření plánu na nadcházející PI.

Bez účasti důležitých zainteresovaných stran týmy těžko plánují své aktivity, kterými dosahují uspokojení požadavků stakeholderů. Zainteresované strany hrají také roli při schvalování návrhů plánů individuálních týmů. Hodnotí vytvořené plány a ke každému poskytují zpětnou vazbu, podle které tým následně plán pozměňuje. Je tedy důležité aby se klíčoví stakeholderi PI Planningu účastnili. Jak jinak tým zjistí, jakých cílů je potřeba v následujícím PI potřeba (Knaster, 2020).

### **Navrhované doporučení:**

Doporučením, jak tento problém vyřešit, je začít aktivně provádět stakeholder management v rámci celého Agile Release Trainu (vlaku).

Stakeholder management spočívá v zapojování všech zainteresovaných stran do procesu od plánování, vývoj až po dodání veškerých požadavků. Tímto procesem se identifikují jednotlivci či skupiny, které mají zájem na dodaných řešení nebo je tato řešení ovlivňují. Zainteresované strany si dělíme na interní (uvnitř organizace – jiná oddělení, manažeři, majitelé, klíčoví lidé, atd.) a externí (zákazníci, koncoví uživatelé). Efektivní komunikace s klíčovými stakeholdery je důležitá pro úspěšné dodávání požadovaných řešení. I přes to, se komunikace a zapojení zainteresovaných stran podceňuje. Nedostatečná komunikace se stakeholdery negativním způsobem ovlivňuje průběh celého projektu. Efektivní zapojování zainteresovaných stran usnadní vlaku každodenní řízení a dodávání požadovaných features (Sedlmajer, 2019).



**OBR. 26:** STAKEHOLDER MANAGEMENT (ZDROJ: SCALED AGILE FRAMEWORK, 2020)



## **Poznatek 5: Chybí jasný plán přípravy PI Planningu**

### **Kategorie: negativní, drobný poznatek**

Bez dostatečné a efektivní přípravy nemůžou členové ARTu očekávat ani efektivní Program Increment Planning. Jelikož se PI Planning koná každé tři měsíce, je důležité, aby příprava na něj byla systematická.

PI Planning je nejdůležitější ceremonií celého rámce SAFe® a vyžaduje přípravu, koordinaci a komunikace mezi všemi jeho účastníky. Všechny účastníky a zainteresované strany je potřeba informovat o jeho konání, aby se mohly dobře a včas připravit (Leffingwell, 2018). Pokud chce ART při PI Planningu uspět, je důležité, aby se příprava nezanedbávala. Připravit se však musí i samotní účastníci, a to přidělením všech důležitých rolí a ujištěním se, že všichni proces PI Planningu správně pochopili. Je také důležité zajistit místo, kde se bude PI Planning konat, to mimo jiné zahrnuje přípravu všech potřebných technologií a materiálů (Blake, 2022).

### **Navrhované doporučení:**

Aby ART věděl, co vše je potřeba před konáním PI Planningu připravit, kdy poslat pozvánky, kdy zajistit místnost a mnoho dalších náležitostí, doporučuje se vytvořit checklist (seznam) všeho co je potřeba. Takový checklist může mít mnoho podob a způsobů zpracování, důležité však je, aby obsahoval všechny potřebné kroky, které jsou klíčové při plánování. Jedním ze způsobů zpracování je vytvoření excelové tabulky, které je rozdělena na jednotlivé role, které se PIPu účastní a ke každé roli jsou vypsány jednotlivé úkoly. Pro větší přehlednost se doporučuje aktivity vypisovat tak, jak na sebe navazují. Checklist také může zobrazovat, jakou dobu před konáním PI Planningu je potřeba jednotlivé aktivity provést (*příklad podoby checklistu naleznete v příloze na str. 104.*)

## **Poznatek 6: V ceremonii Inspect & Adapt chybí backlog úkolů, které z ní vyplývají, a kdo je za tyto úkoly zodpovědný**

### **Kategorie: negativní, závažný poznatek**

Již z teoretické části práce víme, že Inspect & Adapt je ceremonie, která se koná na konci každého PI a při které se hodnotí fungování ARTu a kvalita dodaných řešení.

Z Agilního Manifesta víme, jakou má důležitost neustálé zlepšování. SAFe® tento princip zahrnuje mezi své klíčové kompetence. Ačkoliv se příležitosti k neustálému zlepšování objevují v průběhu celého PI, je důležité zavést a správně si nastavit ceremonii I&A, která pomáhá Agile Release Trainu s identifikací problémů. Pokud si však myslíte, že další ceremonie jen zkrátí čas, který by tým jinak využil k dodávání features, není to pravda. Ceremonie I&A probíhá v iteraci Innovation and Planning, která slouží jako určitá rezerva na konci PI (Scaled Agile Framework, 2020).

Ceremonie Inspect & Adapt se skládá ze tří částí (Leffingwell, 2018):

- **PI System Demo** – Liší se od běžného System Dema. Obvykle se jí účastní více lidí, jako například zákazníci, zástupci portfolia a další klíčoví stakeholdeři. Proto je formálnější a je za potřebí důkladnější přípravy.
- **Kvantitativní a kvalitativní měření** – Týmy přezkoumávají metriky, na nichž se dohodly, diskutují a datech a o vyvíjecích se trendech. Cílem je identifikace potenciálních problémů.
- **Retrospektiva** – Cílem je identifikace několika významných problémů, kterých si tým všiml, a které se rozhodne dále řešit.

### **Navrhované doporučení:**

Je důležité, aby se identifikované problémy dále řešily a jen tak nezahazovaly. Měl by se tedy vytvořit backlog, který obsahuje kroky, jak eliminovat identifikované problémy (improvement features). S tímto backlogem by se mělo pracovat stejně jako s ostatními backlogy. Jednotlivé položky backlogu by se měly prioritizovat a podle důležitosti plánovat do dalších iterací, aby se na problémech mohlo pracovat a průběžně je odstraňovat. Mělo by být jasně stanoveno, kdo má za tento backlog odpovědnost a kdo bude dbát na to, aby se jeho položky plánovaly do iterací a PI. I zde hraje důležitou roli transparence.

## **Poznatek 7: Chybí jasná a srozumitelná roadmapa směřování celého IT ARTu**

### **Kategorie: negativní, závažný poznatek**

Bez srozumitelného plánu kam ART směřuje, se obtížně plánují jednotlivé iterace a PI. Tým tak neví, jaké jsou cíle a milníky, kterých musí během dodávání řešení dosahovat. V tomto případě ART vytvořil roadmapu pro několik následujících PI, ale už nebyl schopen jasně vyznačit důležité milníky a cíle, kterých plánuje dosáhnout. Mohli bychom říct, že pracuje jen na tom, co mu zrovna někdo přidělí.

Roadmapa je v podstatě takový kalendář všech událostí, ceremonií a milníků, které tým informují o plánovaném řešení realizovaném ve stanoveném časovém horizontu. Je však velice důležité si tento časový horizont správně určit. Pokud je moc krátký, může organizace ohrozit synchronizaci týmů a schopnost komunikovat důležité nové features. Pokud je na druhou stranu zase příliš dlouhý, bude ART vycházet z nejisté budoucnosti. Také zahrnuje přehled závazků pro nadcházející PI a předpovídaných výsledků pro několik následujících PI dopředu. Díky roadmapě ART ví, k jakým výsledkům se zavázal (Knaster, 2020).

### **Navrhované doporučení:**

Je potřeba, aby měl ART určenou a dedikovanou roli System Architekta, který má na starost definici a komunikování sdílené technické a architektonické vize pro ART.

Komunikováním těchto vizí pomáhá zajistit, že jsou vyvíjené systémy či individuální features vhodné pro stanovený účel. Systémoví Architekti popisují kontext a záměr dodávaných řešení, určují primární komponenty a subsystémy, identifikují jednotlivá rozhraní a spolupráci mezi nimi. Hrají klíčovou roli při sladování vývojových týmů uvnitř Agile Release Trainu a s těmito týmy spolupracují při vypracovávání požadovaných řešení, ověřování technologických předpokladů a vytváření Continuous Delivery Pipeline (viz str. 57). Ve většině případech je přidělen jeden System Architect na jeden ART, může se však stát, že při realizaci komplexnějších řešení je ARTu přiděleno více Systémových Architektů (Scaled Agile Framework, 2020).

## **Poznatek 8: Důležité zainteresované osoby z oblasti marketingu se účastní PI Planningu**

**Kategorie:** pozitivní

Již v **poznatku 4** bylo zdůrazněna důležitost zapojování všech zainteresovaných stran pro celého procesu od plánování, vývoj až po dodávání jednotlivých features. Skutečnost toho, že se klíčoví stakeholderi z oblasti marketingu účastní ceremonie PI Planning hodnotíme jako pozitivní. Jsou schopni vyjádřit své požadavky, či problémy, na základě kterých pak agilní týmy vytvářejí plány nadcházejících PI, které jsou těmito zainteresovanými stranami schvalovány. Nemůže se tudíž stát, že by tým nakonec dodal řešení, feature či systém, který zainteresovaná strana neschválila (Leffingwell, 2018).

## **Poznatek 9: Kanban i Program Board jsou srozumitelné a přehledné**

**Kategorie:** pozitivní

I přesto, že bylo dříve zmiňované, že by tým měl více používat Program Board, je jeho struktura a vypracování i přesto přehledné a srozumitelné. Díky přehlednosti Program Boardu mohou týmy jasně vidět, jaké features se budou dodávat, jakým týmem jsou tyto features vytvářené a zda mezi nimi existují dependence (Knaster, 2020). Již v úvodu kapitoly škálované agility se zmiňuje, že SAFe při svém fungování čerpá z již používaných metodik (kanban, Scrum či eXtreme Programming). Skutečnost, že tým Kanban využívá při svém fungování a dbá na jeho srozumitelnost a přehlednost, napomáhá týmu dodávat plánované features co nejefektivněji a nejrychleji (Leffingwell, 2018).

#### **4.4.3.2 Ceremonie Scrum of Scrums, System Demo, PO Sync a další**

**Poznatek 10:** Release Train Engineer (RTE) nevede a nespravuje ceremonii PO Sync

**Kategorie:** negativní, kritický poznatek

Ceremonie PO Sync je facilitována buď Release Train Engineerem (RTE) nebo Product Managementem (PdM). Je však důležité jasně stanovit, kdo tuto ceremonii povede, jelikož facilitátor dbá na to, aby se účastníci neodchylovali od stanovené agendy. Tato ceremonie se koná za účelem získat přehled o tom, jak si ART vede při plnění stanovených PI Objectives. V rámci této ceremonie se také diskutují problémy, které si při dodávání features vyskytly a představuje příležitost k prodiskutování příležitostí týkajících se vývoje a dodávání řešení. V některých případech může ceremonie PO Sync sloužit k přípravě na další PI Planning a může zahrnovat Backlog Refinement a stanovení priorit jeho položek pomocí metody Weighted Shortest Job First, zkr. WSJF (Scaled Agile Framework, 2020).

#### **Navrhované doporučení:**

Nejdříve je důležité, aby si vedení a správu ceremonie PO Sync vzal na starost RTE. Následně je nutné si agendu této ceremonie nastavit tak, aby odpovídala potřebám ARTu. V tomto případě je však lepší, aby si facilitaci PO Syncu vzal na starost Release Train Engineer a dbal na to, že je stanovená agenda dodržována, a že se nikdo neodchyluje od diskutovaného tématu. Pokud je role RTE natolik vytížená, že by v budoucnu nestíhala facilitaci této ceremonie, doporučuje se na začátku tuto skutečnost komunikovat a nastavit pravidla tak, aby ceremonie fungovala i bez účasti RTE. Dosáhnout se toho dá tak, že se účastníci postupně naučí PO Sync facilitovat a vést samy takovým způsobem, že účast RTE nebude v konečném důsledku potřeba. Tento způsob je však řešením, které nepřináší výsledky hned. Je nutné dobře vyzorovat, kdy jsou účastníci natolik samostatní, že role RTE již není potřeba.

## **Poznatek 11: Ceremonie Scrum of Scrums někdy přináší velmi omezenou hodnotu a diskutované dependence brzdící pokrok ARTu se nikam dál neposouvají**

**Kategorie: negativní, závažný poznatek**

Z náslechu ceremonie Scrum of Scrums a následného studia dokumentace jsme identifikovali, že SoS nepřináší takovou hodnotu, kterou by měla dodávat. Je to z toho důvodu, že se klade malý důraz na následné řešení závislostí diskutovaných na této ceremonii.

Ceremonie Scrum of Scrums nabízí způsob, jak propojit více agilních týmů, které potřebují spolupracovat na dodávání komplexních řešení prostřednictvím transparentnosti, kontroly přizpůsobování se neustálým změnám. Scrum of Scrums je ve své podstatě virtuální tým složený ze zástupců agilních týmů uvnitř vlaku (Scrum of Scrums, 2020).

Tato ceremonie je facilitována a spravována Release Train Engineerem (RTE) na týdenní bázi (frekvence konání SoS se může měnit) a její trvání je časově omezeno na 15-30 minut. Scrum of Scrum pomáhá odhalovat a koordinovat vzájemné dependence (závislosti) agilních týmů a poskytuje přehled o vyskytujících se překážkách, které brzdí pokrok ARTu. Je velice důležité, aby bylo přehledné a viditelné, jaké závislosti se mezi jednotlivými týmy vyskytují. Tyto týmy tak mají přehled o tom, kde s v případě problému může zpozdit celý proces vývoje a na základě toho pak přijímat opatření, aby zpoždění co nejvíce eliminovaly (Scaled Agile Framework, 2020).

### **Navrhovaná doporučení:**

I v tomto případě se ARTu navrhuje více využívat a komunikovat Program Board, který jasně, přesně a viditelně zobrazuje plánované feature, které se tým zavázal dodat, jejich vzájemné a milníky, které se v průběhu naplánovaného PI objevují a na které se tým musí připravit a dělat vše pro to, aby tyto milníky nepropásl.

Dalším způsobem, jak zajistit, aby bylo s dependencemi dále pracováno, je vytvoření si roadmapy, která bude zaznamenávat individuální kroky směřující k eliminaci co nejvíce závislostí. Je však důležité, aby se s identifikovanými závislostmi dále pracovalo a nenechávaly se jen tak ležet, jelikož mohou tyto vzájemné závislosti významně zpozdit vývoj features a tím ohrozit dodání řešení, která by na PI naplánovaná.

**Poznatek 12: Na všech ceremoniích od PI Planningu, Scrum of Scrums až po řešení klíčových věcí, chybí klíčoví rozhodvatelé (decision makers)**

**Kategorie: negativní, závažný poznatek**

Při návštěvách a následech jednotlivých ART ceremonií jsme narazili na problém absence klíčových *decision makerů*, tedy rozhodvatelů. Absence těchto osob má za následek nesprávnost prováděných rozhodnutí při stanovení PI Objectives, při plánování features do dalšího PI a schvalování plánů.

Pokud nejsou na individuálních ceremoniích a na dalších schůzkách, kde se řeší klíčové věci rozhodvatelé, přišla veškerá příprava a práce na nich vniveč. Rozhodvatelé (decision makers) jsou klíčoví účastníci všech ceremonií, jelikož rozhodují o schválení plánů na další PI, o estimaci a prioritizaci features, o tom jak se budou řešit vyskytující se problémy a mnoho dalších skutečností, při kterých jsou rozhodvatelé potřeba. Mezi rozhodovatele řadíme důležité stakeholdery, tedy zainteresované strany, které mají na dodávaných řešení určitý zájem (Leffingwell, 2018).

**Navrhované doporučení:**

I v této situaci se doporučuje, aby se začal aktivně provádět stakeholder management v rámci celého ARTu.

Stakeholder management spočívá v zapojování všech zainteresovaných stran do procesu celého procesu. identifikuj jednotlivci či skupiny, které mají zájem na dodaných řešení nebo je tato řešení ovlivňují. Zainteresované strany si dělíme na interní a externí. Efektivní komunikace s klíčovými stakeholdery je důležitá pro úspěšné dodávání požadovaných řešení. I přes to, se komunikace a zapojení zainteresovaných stran podceňuje. Nedostatečná komunikace se stakeholdery negativním způsobem ovlivňuje průběh celého projektu. Efektivní zapojování zainteresovaných stran usnadní vlaku každodenní řízení a dodávání požadovaných features (Sedlmajer, 2019).

### **Poznatek 13: ART Sync má dobře nastavenou a dobře strukturovanou agendu**

**Kategorie:** pozitivní

ART Sync patří mezi ceremonie na úrovni ART. Stejně jako PO Sync a Scrum of Scrums se ART Sync soustředí na pokrok PI, upravuje priority features, řídí rizika, závislosti a identifikované překážky. Jelikož jsou agendy PO Sync a Scrum of Scrum do velké míry podobné a jejich účastníci se zajímají o podobná témata, slučují se obě ceremonie do jedné a vzniká z toho právě ceremonie ART Sync. Ve své podstatě nahrazuje tyto dvě ceremonie (Scaled Agile Framework, 2020).

Pozitivně hodnotíme strukturu a nastavení agendy této ceremonie. Díky ní jsou viditelné již zmíněné závislosti, features, milníky a mnoho dalších skutečností, podle kterých se dále mohou agilní týmy orientovat a vždy ví, jak vypadá jejich pokrok a zda dosahují stanovených plánů.

### **Poznatek 14: ART Sync pomáhá synchronizovat očekávání mezi businesssem a agilními týmy**

**Kategorie:** pozitivní

Sladění očekávání všech zainteresovaných stran, v tomto případě businessu a agilních týmů zajišťuje, že všichni v rámci ARTu ví, kam se směřuje a jaké jsou cíle, podle kterých se plánují dodávky požadovaných řešení. Již z předchozího poznatku víme, že ART Sync slučuje ceremonie Scrum of Scrums a PO Sync. (Leffingwell, 2018).



### 4.4.3.3 Nastavení rolí Agile Release Trainu

**Poznatek 15:** Role Release Train Engineer je zaměřena výhradně na dodávky features a řešení a systematicky nepracuje s „mindsetem“ členů týmu, jejich rozvojem a dalším vzděláváním

**Kategorie:** negativní, závažný poznatek

Jako negativní hodnotíme skutečnost, že se Release Train Engineer zaměřuje pouze na dodávky plánovaných features a už neprojevuje zájem o to, zda se agilní tým rozvíjí a neustále vzdělávají v agilních praktikách, bez kterých se vývoj řešení velice zpomaluje.

Release Train Engineer zastává roli servant leadera a do určité míry také roli agilního kouče. Jeho hlavní odpovědností je facilitovat ART ceremonie a pomáhat týmům při dodávání hodnot zákazníkům či koncovým uživatelům. RTE také komunikuje se stakeholdery, eskaluje problémy a překážky, pomáhá řídit rizika a podporuje kulturu neustálého zlepšování. I přes to, že se ART skládá ze samo-organizovaných týmů, musí tento vlak někdo řídit. Tato odpovědnost spadá právě na RTE. Release Train Engineer se také podílí na koučování vedoucích pracovníků, týmů a Scrum Masterů v nových procesech, praktikách, postupech a v nových způsobech myšlení (Scaled Agile Framework, 2020).

Nastává zde však problém, když se RTE věnuje pouze oblasti řízení ARTu a na rozvoj členů agilního týmu zapomíná. V první kapitole teoretické části se zmiňuje důležitost způsobu myšlení při implementaci a následném používání agilního přístupu v organizaci. Bez podpory a přijetí nových agilních praktik agilním týmem, nejsou dodávána řešení takové kvality, kterou zákazníci požadují. Je proto důležité dbát na rozvoj jak jednotlivců tak i celých agilních týmů. Rozvojem se myslí neustálé vzdělávání, které v důsledku vede ke změně „mindsetu“.

#### **Navrhované doporučení:**

Pokud RTE nezvládá řídit individuální procesy ARTu a k tomu rozvíjet agilní týmy, může tuto odpovědnost do nějaké míry předat Scrum Masterovi. Proto se doporučuje, aby měl každý agilní tým svého Scrum Mastera, který je mu na plně dedikovaný, nebo aby existoval pro celý Agile Release Train agilní kouč, který by podporoval roli RTE v činnostech rozvíjejících znalosti a dovednosti týmů.

**Poznatek 16: Osoba, která nastoupila jako Scrum Master později začala pomáhat jako System Architect a Product Owner, což znamená, že tým reálně nemá dedikovaného SM**

**Kategorie: negativní, závažný poznatek**

Provádění rozhovorů s vybranými členy Agile Release Trainu upozornilo na problém, kdy osoba zastávající roli Scrum Mastera, později začala vykonávat také role System Architekta a Product Ownera. Osoba je tak velice vytížená a nedokáže se věnovat jedné oblasti s potřebou efektivnosti.

Scrum Masteři tak jako RTE jsou servant leadeři a koučové agilního týmu. Jejich hlavním úkolem je pomáhat týmu se neustále zlepšovat a vzdělávat ve všech nastavených praktikách a postupů. Také pomáhají při odstraňování překážek a trénují tým v sebeorganizaci a sebeřízení. I Product Owner je člen agilního týmu, který je zodpovědný za maximalizaci dodávané hodnoty zákazníkovi a za soulad týmového backlogu s potřebami koncových uživatelů. V týmu PO zastává roli zákazníka a komunikuje agilnímu týmu jeho požadavky. Tato role je týmu dedikována na plný úvazek. System Architect je pak zodpovědný za definování a následné sdílení architektonické vize. SA popisují kontext dodávaných řešení, určují komponenty, identifikují spolupráci mezi rozhraními (Scaled Agile Framework, 2020).

Na základě popisů rolí SM, PO a SA je hned zřejmé, že není možné, aby všechny tři role zastával jeden a ten samý člověk. Ani jedna role tak není vykonávána efektivně, což má největší vliv na samotné fungování jak agilních týmů, tak i celého ARTu.

**Navrhované doporučení:**

Zde je nutné, aby byla ke každému agilnímu týmu dedikována role Scrum Mastera a to na plný úvazek. Doporučuje se tedy, aby se osoba zastupující všechny tři role zaměřila na vykonávání pouze jedné a odpovědnosti zbylých dvou předala jiným osobám. Bez plně dedikovaného Scrum Mastera, se nemá tým jak posouvat ve svém myšlení a znalostech dál a není zde nikdo, kdo by vývojový tým chránil před překážkami a vnějšími vlivy, které ovlivňují celkové fungování týmu při dodávání features. Je nutné upozornit také na to, že každý tým musí mít svého vlastního Scrum Mastera.

## **Poznatek 17: V Agile Release Trainu (ART) jsou prohozené role Product Managementu (PdM) a Business Ownera (BO)**

**Kategorie: negativní, závažný poznatek**

Při studiu dokumentace, která popisuje jednotlivé role (Job Descriptions), se identifikoval problém, kdy má Agile Release Train prohozené role Product Management a Business Ownera. Prohození rolí PdM a BO v tomto případě znamená, že odpovědnosti, které patří roli Product Managementu provádí Business Owner a naopak, činnosti a odpovědnosti BO provádí PdM.

Hlavní zodpovědností Product Managementu je definice požadovaných řešení, která uspokojují požadavky zákazníků. Jeho úkolem je podpora vývoje během celého životního cyklu produktu. Produktový Management provádí každé dodávané řešení všemi fázemi životního cyklu a zajišťuje, aby byla dodávaná hodnota pro zákazníka co nejvyšší. Business Owner zastává roli stakeholdera (zainteresované strany) a účastní se některých ceremonií Agile Release Trainu. Kromě toho na něho připadají některá povinnosti během PI Planningu. Tyto povinnosti jsou: plánování úkolů, přezkoumávání plánů a jejich návrhů, řešení problémů a také přiřazují business hodnotu stanoveným PI Objectives (cíle PI) (Leffingwell, 2018).

Když si přečteme popis obou rolí, je hned jasné, že každá role má na starosti něco jiného a i jejich odpovědnosti se od sebe liší. Záměna těchto rolí tedy představuje problém, který ovlivňuje fungování ARTu, dodávání požadovaných řešení a plánování cílů a features na další PI.

### **Navrhované doporučení:**

Řešení tohoto problému je vlastně hodně jednoduché. Stačí tyto role prohodit mezi sebou tak, že odpovědnosti, které náleží PdM bude mít na starosti PdM a odpovědnosti BO bude mít na starost BO. Při tomto „prohazování“ odpovědností rolí se však musí dbát na to, aby odpovídaly tomu, jak jsou nastaveny v rámci SAFe® a aby se tyto role i tak komunikovaly jak uvnitř ARTu tak i ven. Je zásadní aby všichni věděli, kdo je Product Management a Business Owner, a co mají tyto role na starost.

## **Poznatek 18: Role Business Ownera opravdu představuje hlas zákazníka**

**Kategorie:** pozitivní

Již dříve popisovala teoretická i praktická část roli Business Ownera a jeho odpovědností, takže víme, že tato role představuje hlas zákazníka a komunikuje všechny jeho požadavky a potřeby jak agilnímu týmu, tak hlavně celému Agile Release Trainu. Pozitivně hodnotíme tedy tu skutečnost, že v rámci ARTu tato role takto funguje a její nastavení je správné, jelikož bez hlasu zákazníka nelze plánovat features a dodávat řešení, které daným zákazníkům přinášejí hodnoty.

#### **4.4.3.4 Vize, roadmapa a backlog**

**Poznatek 19:** Nejsou popsány Key Performance Indicators (KPI), tudíž nevíme, jak měřit naplnění vize

**Kategorie:** negativní, kritický poznatek

V rámci studia dokumentace, která se zaměřovala na popisy plánovaných features jsme zaznamenali, že jim chybí ukazatele výkonnosti, které by ukazovaly, jak vypadá pokrok naplánovaných features a řešení.

Key Performance Indicators (KPI), v překladu klíčové ukazatele výkonnosti, měří a vyhodnocují pokrok směrem k požadovanému výsledku. Tyto ukazatele poskytují orientaci na strategické a provozní zlepšování, vytvářejí základ pro rozhodování a pomáhají zaměřit pozornost na to, co je nejdůležitější. Každý z ukazatelů je specifickým měřítkem business výsledků pro Value Streamy v rámci daného portfolia. Každý ART stanoví minimální metriky, které jsou potřeba k zajištění realizace strategie, vize a k neustálému zlepšování business výsledků (Knaster, 2020).

Bez jasně nastavených metrik a klíčových ukazatelů výkonnosti nelze sledovat pokrok k požadovanému cíli, který si tým či ART na začátku určily. Nemůžeme ani říct, jestli způsob, kterým agilní tým vytváří a následně dodává požadované výsledky zákazníkovi, je efektivní.

#### **Navrhované doporučení:**

Doporučení je zřejmé, každý produkt, řešení či feature musí mít jasně nastavené metriky a Key Performance Indicators, podle kterých lze zjišťovat, jak agilní tým funguje a jak vypadá jeho pokrok ke stanovenému cíli. U každé feature musí být jasně viditelné, jakým způsobem naplňuje existující KPIs.

## **Poznatek 20: Vize by měla být dostatečně konkrétní, specifická a měřitelná**

### **Kategorie: negativní, kritický poznatek**

V současné době se nepracuje s vizí tak, jak je ve znalostním rámci SAFe® stanoveno. Vize sice ukazuje budoucí stav řešení, ale už se nejedná o konkrétní a specifické řešení, které budou uspokojovat koncového uživatele, zákazníka a další důležité stakeholdery.

Vize popisuje budoucí stav vyvíjeného řešení, odráží potřeby zákazníků a klíčových stakeholderů a také navrhuje features, pomocí kterých se identifikované potřeby budou uspokojovat. Vize poskytuje širší kontext a přehled o účelu vyvíjených řešení. Popisuje trhy, segmenty a potřeby zákazníků či koncových uživatelů. V rámci Scaled Agile Framework® rozlišujeme několik druhů vizí dle nastavení konfigurace (viz str.48):

- **Portfolio vize** – Vize portfolia určuje dlouhodobější kontext pro krátkodobá rozhodnutí. Pochopení dlouhodobějšího výhledu pomáhá agilním týmům činit informovanější rozhodnutí o vývoji features jak v krátkodobém, tak i dlouhodobém horizontu.
- **Solution vize** – V konfiguraci Large Solution SAFe® má Product Management a Solution Management odpovědnost za předání Portfolio vize do Solution vize, která indikuje důvod a směr výsledného řešení.
- **Program vize** – Každý ART má svou vlastní vizi, která podrobně popisuje, kam směřují konkrétní features nebo subsystémy, které vytváří. Program vize podporuje vizi Solution, takže by s ní měla být úzce propojena.

Pro vytvoření společného porozumění cílům a záměrům je zásadní neustálý rozvoj, udržování a komunikování vize (Leffingwell, 2018).

### **Navrhované doporučení:**

Při vytváření vize, ať už individuálních Agile Release Trainů, Portfolia či Solution je důležité ji vytvořit tak, aby ukazovala specifické požadavky, na základě kterých vývojové týmy ví, jaké konkrétní výsledky se od nich očekávají. Mít tuto vizi popsanou na nadcházející 2-3 . Nesmíme však zapomenout na to, že z vizí se pojí nutnost sledování určitých metrik, měli bychom se tedy určit konkrétní KPIs (Key Performance Indicators. S vizí se také pojí tvorba dlouhodobé roadmapy, která zviditelňuje události a milníky během vývoje řešení. Na každém PI Planningu by se měla pak tato roadmapa ukazovat, ale pouze v časovém horizontu, který ukazuje nadcházející Program Increment (PI).

## **Poznatek 21: Features dnes nemají popsané jasné benefity, které navazují na vizi**

### **Kategorie: negativní, závažný poznatek**

Feature představuje řešení, které uspokojuje potřeby zákazníků a dalších zainteresovaných stran. Každá feature obsahuje benefit hypotézu a akceptační kritéria a je podle potřeby rozdělena tak, aby mohla být dodána jedním Agile Release Trainem v průběhu jednoho Program Incrementu (PI). Features jsou zásadní pro definici, plánování a implementaci řešení. Každá feature představuje službu, která uspokojuje klíčové potřeby všech zainteresovaných stran. Jsou udržovány v seznamu úkolů, tedy Program Backlogu a jsou rozvrženy tak, aby se vešly do jednoho PI, a aby každý přinášela novou maximální hodnotu. Features jsou vlastněny Product Managementem nebo System Architektem.

### **Navrhované doporučení:**

Doporučuje se, aby se při definování jednotlivých features popsali benefity, které každá individuální feature přináší.

Benefity jednotlivých features se píší tak, aby vyjadřovaly, jakou hodnotu jejich dodání přináší zákazníkovi či koncovému uživateli. Při psaní benefit hypotéz se doporučuje používat standardní formu zápisu, která zní:

Pokud **[tvrzení]**, pak **[přínos]**

Tvrzení pak popisuje, co se plánuje dodat zákazníkovi a přínos na druhou stranu popisuje hodnotu, kterou daná feature zákazníkovi přinese (Steffen, 2019).

I zde se doporučuje si stanovit a určit ke každé plánované feature konkrétní měřitelné klíčové ukazatele výkonnosti (KPI), aby byly vývojový tým i ART schopny sledovat jejich pokrok. Musí z nich být jasně viditelné, na kolik každá individuální feature naplňuje stanovené KPIs.

## **Poznatek 22: Nepracuje se s PI Objectives (PI cíli)**

**Kategorie: negativní, drobný poznatek**

Při provádění assessmentu jsme identifikovali problém, kdy se v žádné formě nepracuje s PI Objectives, tedy s cíli Program Incrementu. Bez stanovených cílů tým neví, co má v nadcházejícím PI dodávat. Tým se spíše zaměřuje jen na dodávání potřebných výstupů než na stanovení cílů, podle kterých následně dodává požadovaná řešení.

PI Objectives představují souhrn obchodních a technických cílů, kterých chce agilní tým nebo celý ART v nadcházejícím PI dosáhnout. Proces stanovení realistických cílů pomáhá zabraňovat přílišnému množství Work in Progress (rozpracované práce). PI Objectives se stanovují během ceremonie PI Planning. Stanovení těchto cílů vyžaduje odhad, plánování, znalost týmové kapacity a definování User Stories pro týmový Backlog. Není pevně dáno, kolik cílů by si měly agilní týmy do dalších PI plánovat, nejvíce efektivním se však zdá být, když si tým stanoví 7-10 cílů. Pokud si tým těchto cílů zvolí příliš mnoho, mohlo by se stát, že nebude schopen tyto cíle splnit (Knaster, 2020).

### **Navrhované doporučení:**

Doporučením je, že by agilní tým měl začít více pracovat s PI Objectives, které by si měl v rámci ceremonie PI Planning stanovit, aby věděl, co má v dalším Program Incrementu očekávat a jakou jednotlivé cíle přinášejí zákazníkovi hodnotu.

## **Poznatek 23: ART i individuální týmy mají vytvořenou roadmapu směřování**

**Kategorie: pozitivní**

Pozitivně hodnotíme zjištěnou skutečnost, že Agile Release Train i agilní týmy vytvářejí svou roadmapu směřování, která jim přehledně zobrazuje důležité události a milníky, na které nesmí při vývoji zapomenout, a na které se musí připravovat. Díky roadmapě dokáží lépe pochopit, co se zavázali v probíhajícím PI dodat. Když týmy vidí, že se jim daří stanovené události a milníky plnit, zvyšuje se tak jejich pocit osobního uspokojení a morálka. Není to tedy jen o tom plnit všechny stanovené cíle, plány a milníky, ale i tom, že jak tým, tak i jednotlivci mají pocit, že něčeho opravdu dosahují, a že jejich práce přináší hodnoty.



## **Poznatek 24: Roadmapa není fixní plán, ale dá se s ní neustále pracovat a měnit ji**

### **Kategorie: pozitivní**

Další pozitivní poznatek, který jsme během assessmentu identifikovali, je práce s roadmapou a její nastavení. ART si správně nastavil a vytvořil roadmapu tak, že se s ní dá neustále pracovat a měnit ji. Flexibilní roadmapa tak pomáhá týmům se přizpůsobit změnám jak na trhu, tak i změnám zákaznickových potřeb. Fixní roadmapa neumožňuje přidávat či odebírat features, které se ART zavázal v probíhajícím PI dodat. V případě, že se dodávání některých feature protáhne, je možné některé milníky posunout, aby měl tým dostatek času k jejich dosažení.

## 4.4.4 Poznatky a navrhovaná doporučení zapsané v listu hodnotitele

### Team of Agile Teams

Competence	Findings	Recommendations
<b>Ability to processes ART Ceremonies</b>		
<b>PI Planning, Inspect and Adapt</b>	<b>Critical</b>	
	<b>C1</b> Ceremonie PI Planning probíhá ve většině případech on-line formou	<b>R1</b> Provádět ceremonii PI Planning co nejvíce on-site (osobně)
	<b>C2</b> Na PI Planningu chybí klíčoví stakeholderi, kteří mají důležité požadavky na vývoj řešení	<b>R4</b> Začít aktivně dělat stakeholder management
	<b>Major</b>	
	<b>Mj1</b> Není jednoznačný přehled o tom, co bylo během PI dodáno. Zároveň není jasné, zda byly plánované features dodány ve stanoveném čase	<b>R2</b> Mnohem více používat a komunikovat Program Board
	<b>Mj2</b> Někteří členové agilních týmů nejsou přítomni na PI Planningu	<b>R3</b> Účast všech členů ARTu je povinná. Zajišťovat, aby se všichni účastnili, vytvořit seznam jmen a kontrolovat zda jsou tam všichni
	<b>Mj3</b> V ceremonii Inspect & Adapt chybí backlog úkolů, které z ní vyplývají, a kdo je za tyto úkoly zodpovědný	<b>R6</b> Vytvořit backlog, který obsahuje improvement features
	<b>Mj4</b> Chybí jasná a srozumitelná roadmapa směřování celého IT ARTu	<b>R7</b> Je zde potřeba role dedikovaného System Architekta
	<b>Minor</b>	
	<b>Mn1</b> Chybí jasný plán přípravy PI planningu	<b>R5</b> Vytvořit checklist, který zobrazuje, co vše je potřeba naplánovat a připravit před konáním PIPu
<b>System Demo, Scrum of Scrum, PO sync, IP Iteration, and others</b>	<b>Critical</b>	
	<b>C3</b> Release Train Engineer (RTE) nevede a nespravuje ceremonii PO Sync	<b>R9</b> RTE by měl facilitlovat ceremonii PO Sync. Postupně se snažit, aby tým k vedení ceremonie RTE nepotřeboval
	<b>Major</b>	
	<b>Mj5</b> Ceremonie Scrum of Scrum někdy přináší velmi omezenou hodnotu a diskutované dependence brzdící pokrok ARTu se nikam dál neposouvají	<b>R10</b> Mnohem více používat a komunikovat Program Board + vytvořit si roadmapu, která bude zaznamenávat kroky k eliminaci dependencí
	<b>Mj6</b> Na všech ceremoniích od PI Planningu, Scrum of Scrums až po řešení klíčových věcí, chybí klíčová decision makeři	<b>R11</b> Začít aktivně dělat stakeholder management
	<b>P3</b> ART sync má dobře nastavenou a dobře strukturovanou agendu	
	<b>P4</b> ART Sync pomáhá synchronizovat očekávání mezi businessem a agilními týmy	
<b>ART Roles execution</b>		
<b>ART Role</b>	<b>Major</b>	
	<b>Mj5</b> Role RTE je zaměřena výhradně na dodávky features a řešení a systematicky nepracuje s "mindsetem" členů týmu, jejich rozvojem a dalším vzděláváním	<b>R12</b> Zajistit, aby měl každý tým svého dedikovaného Scrum Mastera / existence agilního kouče pro celý ART
	<b>Mj6</b> Osoba, která nastoupila jako Scrum Master později začala pomáhat jako System Architect a Product Owner, což znamená, že tým reálně nemá dedikovaného SM	<b>R13</b> Zajistit, aby měl každý tým svého dedikovaného Scrum Mastera + aby se všechny tři role rozdělili mezi tři osoby
	<b>Mj7</b> V Agile Release Trainu (ART) jsou prohozené role Product Managementu (PdM) a Business Ownera (BO)	<b>R14</b> Přehodit role tak, aby se shodovaly s rámcem SAFe®
	<b>P5</b> Role Business Ownera opravdu představuje hlas zákazníka	
<b>Other</b>		
<b>Vize, Roadmap a Backlog</b>	<b>Critical</b>	
	<b>C3</b> Nejsou popsány Key Performance Indicators (KPI), tudíž nevíme, jak měřit naplnění vize	<b>R15</b> Mít jasně nastavené metriky a KPIs pro features, řešení + musí být jasně viditelné, jakým způsobem jsou KPIs naplňovány
	<b>C4</b> Vize by měla být dostatečně konkrétní, specifická a měřitelná	<b>R16</b> Mít jasně nastavené metriky a KPIs pro features, řešení + musí být jasně viditelné, jakým způsobem jsou KPIs naplňovány + vytvořit roadmapu
	<b>Major</b>	
	<b>Mj27</b> Features dnes nemají popsané jasné benefity, které navazují na vizi, viz bod vyše	<b>R17</b> Každá plánovaná feature by měla mít jasně a konkrétně popsanou benefit hypotézu + stanovení KPIs
	<b>Minor</b>	
	<b>Mn2</b> Nepracuje se s PI Objectives	<b>R18</b> Začít pracovat s PI Objectives
<b>P6</b> ART i individuální týmy mají vytvořenou roadmapu směřování		
<b>P7</b> Roadmapa není fixní plán, ale dá se s ní neustále pracovat a měnit ji		

Tab. 2: VYPLNĚNÝ ASSESSOR SHEET (ZDROJ: VLASTNÍ ZPRACOVÁNÍ)

## 4.4.5 Závěrečná zpráva

### Management Summary

Organizace dnes nepoužívá znalostní rámec Scaled Agile Framework® tak, aby využívala celého jeho potenciálu. Základním kamenem celého rámce je ceremonie PI Planning, bez které je fungování SAFe® zcela neefektivní a přináší minimální hodnoty. Z doby covidové zde zůstal zvyk provádět tuto ceremonii ve velké míře on-line, samotný rámec však toto nedoporučuje. Úplně zde chybí osoby (stakeholderi), které jsou zodpovědné za určování požadavků, na základě kterých se následně vyvíjejí požadovaná řešení. Na ceremonii PI Planning také chybí rozhodovatelé (decision makers), kteří mají na starost rozhodovat o plánovaných features, PI Objectives a schvalovat plány na nadcházející PI. Bez těchto klíčových osob se těžko stanovuje a určuje, co a jakým způsobem, má být v dalším PI dodáno. Příprava na tuto ceremonii je také nedostatečná. Další identifikovanou problémovou oblastí je role Release Train Engineera (RTE). Jeho odpovědností je facilitace ceremonie PO Sync, kterou v tomto případě nevede. Dále se RTE zaměřuje pouze na dodávky jednotlivých řešení a nesoustředí se na rozvoj agilního týmu v oblastech mindsetu, dovedností a používání nastavených agilních praktik. Dále byl zaznamenán problém, kdy jeden a ten samý člověk zastupuje tři role ARTu (Scrum Master, Product Owner a System Architect), což má za následek to, že agilní tým nemá svého dedikovaného Scrum Mastera a jeho fungování uvnitř ARTu je tak velice neefektivní. V neposlední řadě upozornil assessment na problém, kdy ART nemá nastavené klíčové ukazatele výkonnosti (KPI), tudíž nevidí vlastní pokrok při vývoji potřebných řešení.

### Key Recommendations – Klíčová doporučení

#### Zaměřit se na správné nastavení a fungování ceremonie PI Planning

- Vytvořit seznam věcí (checklist), který zobrazuje, co vše je potřeba před konáním PI Planningu zařídit, naplánovat a obstarat (rozeslání pozvánek všem účastníkům, zařídit místo konání, vytisknout potřebné materiály, sladit se nad očekáváním – viz str. 81)
- Nastavit PI Planning tak, aby se konal vždy on-site (účastníci se ceremonie účastní osobně) a pouze ve výjimečných případech se mohou někteří účastníci připojit online
- Pozvat na PI Planning klíčové zainteresované strany, které poskytují důležité požadavky na vývoj řešení
- Dbát na to, aby se PI Planningu účastnili všechny agilní týmy

### **Nastavit jednotlivé role tak, aby odpovídaly rámci SAFe®**

- RTE musí nést odpovědnost za facilitaci ceremonie PO Sync. Pokud je RTE vytížený, existuje zde možnost, že by ze začátku tuto ceremonii vedl a spravoval tak, jak je stanoveno a postupem času učil účastníky tuto ceremonii spravovat samostatně
- Více se zaměřovat na rozvoj lidí jak ARTu, tak i individuálních týmů. Doporučuje se, aby měl každý agilní tým svého dedikovaného Scrum Mastera anebo mít agilního kouče k dispozici pro celý vlak (ART), který bude mít rozvoj lidí na starost
- Zpět prohodit role Product Management (PdM) a Business Ownera. Dbát však na to, aby se jejich odpovědnosti ztotožňovali s jejich nastavením v SAFe®

### **Nastavit si jasné metriky, které ARTu a týmu ukazují jejich pokrok**

- Nastavit si jasné klíčové ukazatele výkonnosti (Key Performance Indicators – KPI) pro každou jednotlivou feature, tak aby bylo jasně viditelné, jak jsou klíčové ukazatele v průběhu vývoje feature a řešení naplňovány
- Jasně popsat benefit hypotézy každé feature, tak aby zobrazovaly, jakou hodnotu zákazníkům a koncovým uživatelům přinášejí

### **Building Blocks – Silné stránky v používání SAFe®**

- Ceremonie PI Planning se účastní klíčoví stakeholdeři z oblasti marketingu
- ART vytváří srozumitelný a jasný Kanban a Program Board
- Ceremonie ART Sync má dobře nastavenou agendu a pomáhá synchronizovat očekávání mezi businesssem a agilními týmy
- Role Business Ownera opravdu představuje hlas zákazníka a týmu předává jeho požadavky
- Správně vytvořená a nastavená roadmapa směřování ARTu

# Závěr

Cílem diplomové práce bylo provedení vyhodnocení způsobu používání znalostního rámce Scaled Agile Framework®. Práce popisovala principy Agilního Manifesta a srovnávala mezi sebou tradiční vodopádový přístup a agilní přístup. Podrobněji se práce zabývala samotným rámcem SAFe®, který popisuje způsoby a možnosti, jak škálovat agilní metodiky a praktiky v různých velikostech firem od Essential SAFe® až po Full SAFe®.

Assessment neboli vyhodnocení, které se provádělo v rámci praktické části této práce se zaměřuje na několik oblastí fungování firmy s používáním agilních metodik. Pro účely této práce byla vybrána oblast Agile Processes & Competencies s užším zaměřením na podoblast agilních týmů. Assessment tak vyhodnocoval, jak jednotlivé agilní týmy a ART používají stanovené metody a praktiky. Při provedení assessmentu jsme se zaměřovali primárně na fungování a nastavení ceremonie PI Planning jakožto nejdůležitější ceremonie celého rámce, dále pak na nastavení dalších ceremonií na úrovni ARTu, na nastavení SAFe® rolí a na práci s vizí, roadmapou a backlogem.

Identifikované poznatky byly jak negativního, tak i pozitivního rázu. Nejvíce negativních poznatků pak bylo identifikováno právě v oblasti fungování ceremonie PI Planning, která je stěžejní pro fungování tohoto rámce. Assessment nás upozornil na to, že je příprava na tuto ceremonii podceňována a nemá žádnou systematickou strukturu, která by jasně udávala, co všechno je ke konání této ceremonie potřeba. Také upozornil na absenci některých klíčových rolí ARTu na této ceremonii. Pozitivní však bylo, že se této ceremonie účastní klíčoví stakeholdeři marketingového oddělení a že se správně používají Kanban Board a Program Board sloužící ke znázornění plánovaných features a jejich závislostí.

Při assessmentu rolí jsme narazili také na problematický jev, kdy jedna osoba zastávala tři role uvnitř ARTu, a to Scrum Master, Product Owner a System Architect, což má za následek neefektivní fungování všech těchto rolí. Organizace se ve velmi malé míře soustředí na měření pokroku vývoje features, tudíž nemá dostatek informací o tom, jak si vedou a zda vývoj něco nebrzdí.

Klíčovým doporučením pro organizaci na základě poznatků je se soustředit na zlepšení organizace ceremonie PI Planning a komunikování důležitosti, kterou tato ceremonie hraje. Také doporučujeme soustředit se na nastavení a fungování role Release Train Engineer, který má na starosti facilitaci většinu ART ceremonií.

# Přílohy

## Klíčové agilní pojmy

### User story – uživatelský příběh (US)

User story neboli uživatelský příběh je funkce dodávaného produktu, která zákazníkům přináší určitou hodnotu. User stories se píše z pohledu zákazníka a píše se podle tohoto vzoru (Agile Glossary):

Jako [typ uživatele], chci [provést nějaký úkol], abych mohl [dosáhnout nějakého cíle].

### Feature – funkcionalita

Feature je jednotka funkčnosti softwaru, která před stavuje požadavek zákazníka a plní určitou funkci (Apel, 2009).

### Epic

Epic představuje rozsáhlý soubor prací, který popisuje hlavní oblasti funkcionalit. Obvykle se epic realizuje v několika sprintech (Agile Glossary). Jednotlivé epics se následně rozpadají na menší User Stories (US) (Šochová, 2019).

### Definition of Done (DoD)

Definition of done je konsistentní a dohodnutý soubor kritérií, který musí jednotlivé výstupy splňovat, aby byly považovány za „hotové“. Díky tomu se kvalita dodaných výstupů zvyšuje (Perkin, 2020).

### Definition of Ready (DoR)

Definition of Ready je formální způsob, jakým si organizace definuje kvalitu. Jakmile daná položka splní kritéria Definition of Ready, stává se inkrementem (Agile Glossary).

### Akceptační kritéria

Akceptační kritéria jsou požadavky, které musí každá dodávaná feature splnit, aby byla považována za dokončenou (Agile Glossary).

### Estimace – odhady

Při vývoji softwaru je estimace, neboli odhad vyhodnocením úsilí, které je potřeba k provedení jednoho daného úkolu. Nejčastěji se odhad vyjadřuje v délce trvání úkolu. (Agile Glossary).

## **Velocita – rychlost**

Velocita slouží k lepšímu odhadu práce pro následující sprinty a ukazuje nám, jak moc je tým efektivní. Velocitu, neboli rychlost, měříme pouze v jednom daném týmu a nelze vzájemně porovnávat týmy mezi sebou (Šochová, 2019).

## **Iterace**

Iterace je stanovený časový úsek, během kterého probíhá vývoj jednotlivých stanovených funkcionalit. Doba trvání iterace se může u jednotlivých projektů lišit, ale obvykle je stanovena na období dvou týdnů (Agile Glossary).

## **Inkrement – přírůstek řešení**

Inkrement je artefakt, který udává hodnotu, která bude zákazníkům / konečným uživatelům dodána na konci sprintu. Inkrementy (přírůstky) na sebe obvykle navazují a jsou základem pro dodání kompletního produktu (Agile Glossary).

## **Sprint**

Sprint je vymezený časový úsek, během kterého plní týmy konkrétní požadavky zákazníka a postupují směrem ke stanovenému cíli. Sprinty pomáhají týmům přeměňovat nápady na konečnou hodnotu, která se následně zákazníkovi dodává (A Glossary of Agile Terminology).

## **Backlog**

Backlog je priritizovaný seznam pracovních položek (backlog items), funkcí či požadavků, které musí vývojový tým během sprintu dokončit. Často se tento seznam sestavuje ještě před zahájením sprintu a tým se rozhoduje, na kterých položkách bude pracovat. Většinou to však jsou položky s nejvyšší prioritou (Simpson, 2022).

## **Demo**

Demo představuje ceremonii, při které vývojový tým předvádí zákazníkovi / koncovému uživateli produkt nebo jeho část, kterou vytvořil během sprintu (Šochová, 2019).

## **Dependence – závislost**

Zahrnuje událost, spolupráci či kus nějaké práce, která je závislá na práci či zapojení jiného týmu či jednotlivce. Aby tým mohl danou položku dokončit, je nutné aby splnil všechny potřebné závislosti (Agile Glossary).

## **DevOps (Development and Operations)**

DevOps je soubor předem definovaných postupů a pravidel, které kombinují vývoj (development) a provoz (operations) softwaru. Cílem je zkrátit cyklus dodávek jednotlivých položek softwaru a zároveň zlepšit jeho kvalitu. Obvykle se používá při vývoji, testování a implementování (Simpson, 2022).

## **Minimum Viable Product – MVP**

Minimum viable product (MVP) je koncept, který zdůrazňuje vliv učení na vývoj nových produktů. MVP je taková verze nového produktu, která umožňuje týmu získat maximální množství poznatků o koncových uživateli s co nejmenším úsilím. Účelem MVP je vytvoření takového produktu, který se nabídne zákazníkovi a sleduje se jejich chování při používání produktu nebo služby (Agile Glossary).

## **Klíčové ukazatele výkonnosti – Key Performance Indicator – KPI**

Key Performance Indicators jsou měřitelné ukazatele pokroku směrem ke stanovenému cíli. Poskytují zaměření pro strategické zlepšování, vytvářejí základ pro rozhodování a pomáhají zaměřovat pozornost na to, co je nejdůležitější (Agile Glossary).

## **Některé pojmy rámce SAFe®**

### **Value Stream – Hodnotový tok**

Value Stream představuje postup, který organizace používá k vytváření jednotlivých řešení, která poskytují nepřetržitý tok hodnoty pro zákazníka či koncového uživatele. V rámci SAFe® se Value Streamy využívají k definici a realizaci business cílů a k organizaci ARTu jako celku, tak aby dodával hodnotu co nejrychleji (Leffingwell, 2018).

### **Solution – Řešení**

Každý Value Stream vytváří jedno či více řešení. Pod pojmem řešení si můžeme představit jednotlivé produkty, služby či systémy, které jsou následně dodávány zákazníkovi (Knaster, 2020).

### **Roadmap**

Roadmap je rozpis událostí a milníků, které informují o plánovaných výsledcích v nějakém časovém horizontu (měsíc, čtvrtletí, půl rok, rok...) (Knaster, 2020). Roadmap zahrnuje plánované závazky na nadcházející PI a přináší o plánovaných výsledcích pro několik následujících PI (Leffingwell, 2018).



## **Program Increment – PI**

Program Increment je časový úsek, během kterého dodávají jednotlivé ARTy hodnotu. Délka PI je obvykle mezi 8 až 12 týdny. Ve většině případech obsahuje PI čtyři vývojové iterace a jednu IP (Innovation and Planning) iteraci (Scaled Agile Framework, 2020).

## **PI Objectives**

PI Objectives jsou ve své podstatě shrnutí technických a business cílů, kterých chce ART nebo agilní tým v příštím PI dosáhnout (Knaster, 2020).

## **Lean Enterprise – Štíhlý podnik**

Lean Enterprise představuje dobře fungující organizaci současné digitální doby, které vykazuje business agilitu – rychle reaguje na neustále se měnící trh a na nové příležitosti tak, že svým zákazníkům či koncovým uživatelům dodává inovativní systémy, za co nejkratší možný čas (Knaster, 2020).

## **Innovation and Planning iterace**

IP iterace poskytují „buffer“ (časovou rezervu) pro odhadování (estimaci) položek ARTu, pro PI Planning, průběžné vzdělávání a pro dodatečnou práci na infrastruktuře ARTu (Scaled Agile Framework, 2020).

## **Feature**

Feature je produkt, služba či systém, který naplňuje potřeby stakeholderů. Každá feature obsahuje benefit hypotézu a akceptační kritéria. Podle potřeby jsou features rozdělovány tak, aby je mohl v průběhu jednoho PI dodat jeden ART (Knaster, 2020).

## **Benefit hypotéza**

Benefit hypotéza popisuje veškeré možné přínosy pro zákazníka (Scaled Agile Framework, 2020).

## **Solution Train**

Solution Train je organizační struktura, která je používána k vytváření rozsáhlých a komplexních řešení, které vyžadují spolupráci několik ARTů (Scaled Agile Framework, 2020).

## **Vision – Vize**

Vize je popis budoucího žádoucího stavu vyvíjeného řešení. Tato vize odráží potřeby zákazníků a dalších stakeholderů. Také obsahuje seznam features a způsoby, jak potřeby těchto zákazníků uspokojit (Knaster, 2020).

## Weighted Shortest Job First – WSJF

WSJF je model ke stanovení prioritizace, který se používá pro vytváření sekvence jednotlivých produktů (features, epics...) s cílem dosáhnout co nejvyššího ekonomického přínosu (Scaled Agile Framework, 2020).

## Cíle a klíčové výsledky – Objectives and Key Results – OKRs

OKRs je rámec používaný ke stanovení cílů, zaměřený na objektivní dokazování pokroku (Key Results) při dosahování obchodních cílů (Objectives) (Scaled Agile Framework, 2020).

## Program Board

Program Board je jedním z výstupů PI Planningu a upozorňuje na termíny dodání nových features, na závislosti mezi týmy a na významné milníky (Scaled Agile Framework, 2020).

## Checklist přípravy na PI Planning

ODPOVĚDNOST	2 MĚSÍCE PŘED	1 MĚSÍČ PŘED	2 TÝDNY PŘED	TÝDEN PŘED	1 DEN PŘED PIP
RTE	Rozeslání pozvánky do kalendáře na účastníky (1 rok dopředu) - 1. vlna (SM, PO)	Aktualizace PI planning agendy pokud je nutné	Soupis fyzického materiálu na PI planning	Kontrola a transport fyzického materiálu na místo pro PI planning	Rozeslání účastníkům organizační informace k PI planningu
	Potvrzení si místa a formátu PI planningu (1 rok dopředu)	Poslat agendu v pozvánce na PIP	Stanovit komunikační tool na PI planning - skype chat, Teams aj.	Ověřit a zajistit techniku používanou během PI planningu - fyzicky i online	Kontrola fyzického místa/otestování všech online pozvánek
		Rozeslání pozvánky do kalendáře na účastníky (1 rok dopředu) - 2. vlna	Ověření si příchodu stakeholderů na PI planning	Udělat mapku rozmístění stakeholderů ke stolků v místnosti	
		Poslat týmu plán přípravy na PI planning (V rámci ART syncu - Znalost Backlogu a priorit, Dependence - jsou pozváni lidé bez kterých to nejde)	Aktualizace Business Contextu do prezentace	Mít připravený checklist, komu se kdy předává slovo a na přípravu na místě - rozeslání žádost o místnosti	
		Naplánovat schůzku na projití prezentace s BO	Doplnit do prezentace PI roadmapu	Zjistit funkčnost pozvánek (online verze)	
		Sdílet prezentaci z 1. PIP	Příprava boardů, tabulí a šablon pro PI planning	Potvrzení si obsahu systém dema	
			Přípravit na management review prezentací s plány, riziky a body k rozhodnutí	Sladění řečníků nad prezentací pro PI planning	
PdM		Zrealizovat schůzku BOs a produktového managementu pro potvrzení priorit na následující PIs	Sjednocení si priorit na další PI na úrovni managementu firmy		
		Revize plánovaných milníků, velké release, marketingové kampaně			
SM					
SA			Připravený slide za architekturu a přidán do	Zapojení systém týmu při definování US	

OBR. 27: PŘÍKLAD CHECKLISTU K PŘÍPRAVĚ PI PLANNINGU (ZDROJ: VLASTNÍ ZPRACOVÁNÍ)

# Seznam použité literatury

1. *A Glossary of Agile Terminology* [online]. San Jose, Kalifornie: Wrike [cit. 2023-03-30]. Dostupné z: <https://www.wrike.com/agile-guide/glossary/>
2. *Agile Glossary* [online]. Kalifornie: Aha! Labs [cit. 2023-03-30]. Dostupné z: <https://www.aha.io/roadmapping/guide/agile/agile-glossary>
3. *Agile Glossary* [online]. Knoxville, Tennessee, USA: Agile Alliance [cit. 2023-03-30]. Dostupné z: <https://www.agilealliance.org/agile101/agile-glossary/>
4. *Scaled Agile Framework* [online], 2020. Boulder, Colorado: Scaled Agile [cit. 2023-04-07]. Dostupné z: <https://v5.scaledagileframework.com/>
5. *Scrum of Scrums* [online], 2020. Sydney: Atlassian.com [cit. 2023-04-20]. Dostupné z: <https://www.atlassian.com/agile/scrum/scrum-of-scrums>
6. *What is Kanban?* [online], 2018. Knoxville, Tennessee, USA: Agile Alliance [cit. 2023-04-04]. Dostupné z: <https://www.agilealliance.org/glossary/kanban>
7. APEL, Sven a Christian KÄSTNER, 2009. An Overview of Feature-Oriented Software Development. *Journal of Object Technology* [online]. **8**(4), 1-36 [cit. 2023-03-30]. Dostupné z: [https://www.cs.cmu.edu/~ckaestne/pdf/JOT09\\_\\_OverviewFOSD.pdf](https://www.cs.cmu.edu/~ckaestne/pdf/JOT09__OverviewFOSD.pdf)
8. AY, Selman, 2020. Squads, tribes, guild: To be like Spotify or not?. *Uxdesing.cc* [online]. Spojené státy americké: UX Collective, 29.12.2020 [cit. 2023-04-04]. Dostupné z: <https://uxdesign.cc/squads-tribes-guild-to-be-like-spotify-or-not-13ecf690fd36>
9. AZANHA, Adrialdo a Ana Rita TERANDES TERRA ARGOUUD, 2017. Agile Project Management with Scrum: A case study of a Brazilian Pharmaceutical company IT project. *International Journal of Managing Projects in Business*. **10**(1), 121-142. Dostupné z: doi:10.1108
10. BERMAN, Josh. Agile at Scale. *Atlassian.com* [online]. Sydney: Atlassian [cit. 2023-04-03]. Dostupné z: <https://www.atlassian.com/agile/agile-at-scale>
11. BLAKE, Sean, 2022. The Ultimate Guide to PI Planning [2023 SAFe Edition]. *Easy Agile.com* [online]. Nový Jižní Wales: Easy Agile, 25.01.2022 [cit. 2023-04-19]. Dostupné z: <https://www.easyagile.com/blog/the-ultimate-guide-to-pi-planning-2023-safe-edition/#what-should-we-do-to-prepare-for-pi-planning>

12. CRUTH, Mark. Discover the Spotify model. *Atlassian.com* [online]. Sydney: Atlassian [cit. 2023-04-04]. Dostupné z: <https://www.atlassian.com/agile/agile-at-scale/spotify>
13. DOLEŽAL, Jan, 2022. *Agilní přístupy vývoje produktu a řízení projektu: Komplexně, prakticky a dle světové praxe*. 1. Praha: Grada Publishing. ISBN 978-80-271-3705-3.
14. FISHER, Bruce D. a Loubna CHAREF, 2021. Leadership in an Agile Project Management Environment. *Journal of Leadership, Accountability, and Ethics*. **18**(4), 18-22.
15. HAMMARBERG, Marcus a Joakim SUNDÉN, 2014. *Kanban in Action*. 1. Shelter Island, New York: Manning Publications Co. ISBN 9781617291050.
16. HOLBECH, Linda, 2018. *The Agile Organization: How to build an engaged, innovative and resilient business*. 2. Londýn: Kogan Page, Ltd. ISBN 978 0 7494 8265 7.
17. HUNT, J. Ashley, 2018. *PMI-ACP Project Management Institute Agile Certified Practitioner Exam Study Guide*. 1. New York: John Wiley & Sons, Incorporated. ISBN 9781119434634.
18. CHAPPELL, Erica, 2020. Benefits of XP Agile Methodology in Software Development. *ClickUp.com* [online]. San Diego: ClickUp.com, 20.08.2020 [cit. 2023-03-21]. Dostupné z: <https://clickup.com/blog/what-is-xp-in-agile/#1-what-is-extreme-programming->
19. IQBAL, Mary, 2021. What is Iterative, Incremental Delivery? The Hunt for the Perfect Example. *Scrum.org* [online]. Cambridge, Massachusetts: Scrum.org, 03.11.2021 [cit. 2023-03-25]. Dostupné z: <https://www.scrum.org/resources/blog/what-iterative-incremental-delivery-hunt-perfect-example>
20. KADLEC, Václav, 2004. *Agilní programování: Metodiky efektivního vývoje softwaru*. 1. Brno: Computer Press. ISBN 80-251-0342-0.
21. KNASTER, Richard a Dean LEFFINGWELL, 2020. *SAFe® Distilled: Achieving Business Agility with the Scaled Agile Framework®*. 5.0. Londýn: Pearson Education. ISBN 978-0-13-682340-7.
22. KNIBERG, Henrik a Anders IVARSSON, 2012. *Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds*. 1. Švédsko, 14 s. Dostupné také z: <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf>

23. KUKHNAVETS, Pavel, 2018. Advantages and disadvantages of Extreme Programming (XP). *Hygger.com* [online]. Newark, Delaware: Hygger, 18.03.2018 [cit. 2023-03-23]. Dostupné z: <https://hygger.io/blog/disadvantages-and-advantages-of-extreme-programming/#advantages-of-extreme-programming>
24. LEFFINGWELL, Dean, Richard KNASTER, Inbar OREN a Drew JEMILO, 2018. *SAFe® Reference Guide: Scaled Agile Framework® for Lean Enterprises*. 4.5. Londýn: Pearson Education. ISBN 978-0-13-489286-3.
25. Manifesto for Agile Software Development, 2001. *Manifesto for Agile Software Development* [online]. Utah: Ward Cunningham [cit. 2023-03-08]. Dostupné z: <http://agilemanifesto.org/iso/en/manifesto.html>
26. MCGREGOR a DOSHI, 2020. *Agile: The Insights You Need from Harvard Business Review: Why Agile Goes Awry*. 1. Brighton, Massachusetts: Harvard Business Review Press. ISBN 9781633698963.
27. MCKENNA, Dave, 2016. *The Art of Scrum: How Scrum Masters Bind Dev Teams and Unleash Agility*. 1. New York: Apress L. P. ISBN 9781484222775.
28. MEYER, Bertrand, 2014. *Agile!: The Good, the Hype and the Ugly*. 1. Švýcarsko: Springer International Publishing. ISBN 978-3-319-05155-0.
29. MORAN, Alan, 2015. *Managing Agile: Strategy, Implementation, Organisation and People*. 1. Švýcarsko: Springer International Publishing. ISBN 978-3-319-16261-4.
30. MYSLÍN, Josef, 2016. *Scrum: Průvodce agilním vývojem softwaru*. 1. Brno: Computer Press. ISBN 978-80-251-4650-7.
31. PERKIN, Neil, 2020. *Agile Transformation: Structures, Processes and Mindsets for the Digital Age*. 1. Londýn: Kogan Page, Ltd. ISBN 978 0 7494 9747 7.
32. RIGBY, Darrell, 2020. *Agile: The Insights You Need from Harvard Business Review: Agile: How to Get in the Game (and not Get in the Way)*. 1. Brighton, Massachusetts: Harvard Business Review Press. ISBN 9781633698963.
33. RIGBY, Darrell, Sarah ELK a Steve BEREZ, 2020. *Doing Agile Right: Transformation Without Chaos*. 1. Brighton, Massachusetts: Harvard Business Review Press. ISBN 9781633698710.
34. RUBIN, Kenneth S., 2013. *Essential Scrum*. Upper Saddle River: Pearson Education. ISBN 978-0-13-704329-3.
35. SALAMEH, Abdallah a Julian M. BASS, 2021. An architecture governance approach for Agile development by tailoring the Spotify model. *AI & SOCIETY* [online].

- 2022(37), 761–780 [cit. 2023-04-04]. Dostupné z:  
file:///C:/Users/AnetaPolenova/Downloads/s00146-021-01240-x.pdf
36. SEDLMAJER, Michal, 2019. Stakeholder Management. *Projectman.cz* [online].  
Praha: Projectman.cz [cit. 2023-04-19]. Dostupné z:  
<https://www.projectman.cz/blog/stakeholder-management>
37. SHORE, James, Diana LARSEN, Gitte KLITGAARD a Shane WARDEN, 2021. *The Art of Agile Development*. 2. Sebastopol: O'Reilly Media. ISBN 978-1-492-08069-5.
38. SCHWABER, Ken a Jeff SUTHERLAND, 2020. *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. 3. Spojené státy americké: Scrumguides.org.  
Dostupné také z: <https://scrumguides.org/scrum-guide.html>
39. SIMPSON, Julie, 2022. 26 Agile Terms To Upgrade Your Project Management Vocabulary. *Hive.com* [online]. USA: Hive, 26.08.2022 [cit. 2023-03-30]. Dostupné z: <https://hive.com/blog/agile-terms/>
40. STELLMAN, Andrew a Jennifer GREENE, 2015. *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. 2. Sebastopol: O'Reilly Media. ISBN 978-1-44933192-4.
41. ŠMITE, Darja a Nils BREDE MOE, 2010. *Agility Across Time and Space: Implementing Agile Methods in Global Software Projects*. 1. Berlín: Springer Nature Group. ISBN 978-3-642-12441-9.
42. ŠOCHOVÁ, Zuzana a Eduard KUNCE, 2019. *Agilní metody řízení projektů*. 2. Brno: Computer Press. ISBN 978-80-251-4961-4.

# Seznam obrázků

<b>Obr. 1:</b> rozdíl mezi agilitou a tradičními přístupy (Zdroj: Kadlec, 2004) .....	20
<b>Obr. 2:</b> Waterfall model (zdroj: Myslín, 2016) .....	22
<b>Obr. 3:</b> Product Backlog (zdroj: Rubin, 2013) .....	32
<b>Obr. 4:</b> Sprint (Zdroj: Rubin, 2013) .....	33
<b>Obr. 5:</b> Kanban Board (zdroj: Šochová, 2019) .....	36
<b>Obr. 6:</b> Škálování za pomoci tribes, squads, chapters a guilds (zdroj: Kniberg, 2012) .....	41
<b>Obr. 7:</b> Squad, základní jednotka Spotify Modelu (zdroj: Kniberg, 2012) .....	42
<b>Obr. 8:</b> Tribe (Zdroj: Kniberg, 2012) .....	43
<b>Obr. 9:</b> Chapter (Zdroj: Kniberg, 2012) .....	43
<b>Obr. 10:</b> Guilds (zdroj: Kniberg, 2012) .....	43
<b>Obr. 12:</b> SAFe® Big Picture se 4 konfiguracemi (Zdroj: Scaled Agile Framework, 2020) ..	46
<b>Obr. 13:</b> SAFe® konfigurace (Zdroj: Knaster, 2020) .....	48
<b>Obr. 14:</b> Essential SAFe® (Zdroj: Scaled Agile Framework, 2020) .....	48
<b>Obr. 15:</b> Large Solution SAFe® (Zdroj: Scaled Agile Framework, 2020) .....	49
<b>Obr. 16:</b> Portfolio SAFe® (Zdroj: Scaled Agile Framework, 2020) .....	50
<b>Obr. 17:</b> Agile Release Train - ART (Zdroj: Scaled Agile Framework, 2020) .....	54
<b>Obr. 18:</b> Tradiční funkční organizace (zdroj: Scaled Agile Framework, 2020) .....	56
<b>Obr. 19:</b> Multifunkční ART (Zdroj: Scaled Agile Framework, 2020) .....	56
<b>Obr. 20:</b> Multifunkční tým (zdroj: Scaled Agile Framework, 2020) .....	57
<b>Obr. 21:</b> Continuous Delivery Pipeline (Zdroj: Scaled Agile Framework, 2020) .....	58
<b>Obr. 22:</b> Přehled SAFe® ceremonií (zdroj: Scaled Agile Framework, 2020) .....	62
<b>Obr. 23:</b> Agenda PI Planningu (Zdroj: Scaled Agile Framework, 2020) .....	63
<b>Obr. 24:</b> SAFe® Assessment (Zdroj: poradenská firma) .....	70
<b>Obr. 25:</b> Assessor Sheet (zdroj: vlastní zpracování) .....	73
<b>Obr. 26:</b> Program Board (Zdroj: Scaled Agile Framework, 2020) .....	78
<b>Obr. 27:</b> Stakeholder Management (zdroj: Scaled Agile Framework, 2020) .....	80
<b>Obr. 28:</b> Příklad checklistu k přípravě PI Planningu (Zdroj: vlastní zpracování) .....	106

# Seznam tabulek

<b>Tab. 1:</b> Rozdíly mezi tradičním a agilním přístupem (zdroj: Azanha, 2017).....	22
<b>Tab. 2:</b> Vyplněný Assessor Sheet (zdroj: vlastní zpracování).....	98



