

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS		
Semestr:	LS 2019/2020	
Autor:	Nikola Hažmuková	Hazn00
Téma:	Strukturovaný popis škálovatelného rámce Matrix of Services	
Datum odevzdání:	24.5.2020	

- **Název:** Matrix of Services
- **Základní zdroj k rámci:** <http://www.continuousagile.com/>
- **Web rámce:** <http://www.continuousagile.com/>
- **Rok vzniku:** 2016
- **Autor:** Andy Singleton
- **Kategorizace dle Horlach et al.:** Mezi-týmově zaměřený framework

Obsah

Úvod	3
1. Continuous Agile.....	4
1.1. Stavební bloky	4
1.2. Lean principy	4
2. Matrix of Services.....	6
2.1. Role.....	6
2.2. Rozdělení SW aplikace a infrastruktury.....	6
2.3. Organizace týmů.....	7
2.4. Organizace práce	8
2.5. Škálovatelnost	9
3. Srovnání s rámcem SAFe	10
3.1. Týmy	10
3.2. Zákazníci / Uživatelé	10
3.3. Agilní metody a prostředí.....	11
3.4. Produkt / Řešení	11
3.5. Složitost	12
3.6. Hodnotový řetězec	12
4. Použití Matrix of Services	12
Závěr.....	13
Reference	14
Seznam Obrázků.....	14
Seznam Tabulek.....	14

Úvod

Agilní vývoj softwaru je skupina metod vývoje softwaru založených na iterativním a inkrementálním vývoji, kde se požadavky a řešení vyvíjí díky spolupráci mezi samoorganizačními, mezifunkčními týmy. Podporuje adaptivní plánování, vývoj a vývoj, časový iterační přístup a podporuje rychlou a flexibilní reakci na změny. Jedná se o koncepční rámec, který podporuje předpokládané interakce v průběhu vývojového cyklu. Agile Manifesto zavedl termín v roce 2001.

Škálovatelnost dobrou a žádanou vlastností systému nebo procesu. Jedná se o rozšiřitelnost, schopnost pracovat s náhlými změnami potřeby obsluhy a schopnost ustát vysokou zátěž nebo naopak snížit výkon během zátěže nízké. Tato semestrální práce se zabývá škálovatelným rámcem Matrix of Services, zkráceně MAXOS.

Cílem semestrální práce je strukturovaný popis frameworku Matrix of Services a jeho fungování. V první části bude představen pojem Continuous Agile, na jehož principech se Matrix of Services zakládá. V druhé části bude vysvětleno fungování Matrix of Services, rozdělení rolí, softwarových aplikací, infrastruktura, organizace týmů, organizace práce a škálovatelnost frameworku. V třetí části potom budou popsány podobnosti a rozdíly mezi Matrix of Services a velmi populárním škálovatelným frameworkem Scaled Agile Framework, zkráceně SAFe. Vzhledem k velmi omezenému počtu zdrojů práce vychází především z oficiální webové stránky Matrix of Services. Zdrojem pro porovnání MAXOS a SAFe je Collabnet Versionone, kde byly velmi přehledně popsány silné i slabé stránky obou frameworků.

1. Continuous Agile

Matrix of Services stojí na principech kontinuální agility (Continuous Agile). Tyto principy Andy Singleton na stránce Maxos definuje na své webové stránce, popisuje jimi kombinaci Kanban Agile task management a management kontinuálního dodávání kódu (continuous delivery code management). Čerpá hlavně z rychle se rozvíjejícího umění programování a testování, dále bere inspiraci z používání Kanbanu pro testování a nasazení kódu.

1.1. Stavební bloky

Příručka s principy continuous agile popisuje sadu stavebních bloků, ze kterých mohou programátoři i testeři čerpat a vybrat si z ní, co jim vyhovuje. Tyto stavební bloky jsou:

- Správa úkolů
- Správa kódu
- Testování
- Produktový management
- Porozumění vedoucích týmů

Správa úkolů spočívá v taktice, kdy vše začíná vytvářením seznamu (roadmapping) a pokračuje plánováním vydávání kódů v dávkách. Následně je možná pokračovat kontinuálně nebo Kanbanem.

Pro Code management je klíčové mít repositář. Singleton uvádí, že se osvědčuje centralizovaný proces, kdy se všechno nové přispívání a testování provádí na jedné verzi, nebo „distribuovaný“ proces, kde jsou nové části kódu testovány na oddělených verzích.

Agilní přístup vyžaduje schopnost rychlých změn. Často se stává, že je třeba že je zapotřebí otestovat změnu co nejrychleji a nejefektivněji. K tomu je žádoucí mít ve firmě kulturu, která takové testování podporuje. Proto Singleton v principech kontinuální agility uvádí, že investice do testování, především automatizovaného, je to nejdůležitější pro posunutí ke kontinuální agilitě.

Product management cílí hlavně na samotné developery. Umožňuje jim vydávat releasy na základě jejich vlastních potřeb, aby nemuseli s vydáním čekat, nebo naopak být ve stresu z blížícího se deadlineu.

Vedoucí týmů by měli rozumět všem možnostem a nejen těm, se kterými jsou již seznámeni. Měli by být schopní měnit taktiku, pokud se dostanou do nějakých obtíží, zpoždění, nebo naopak čelí novým příležitostem.

1.2. Lean principy

Pro dosažení efektu, kdy je možné vytvořit software rychleji a s menším množstvím stresu, využívá Continuous Agile podle Andyho Singletona některé „lean“ principy.

- Neustálé zlepšování. Mít pravidelný engineering review a průzkum spokojenosti. Měřte výsledky a současně zlepšujte měření.
- Snižujte dávky práce, dokud nedojdete k „toku jednoho kusu“; Provedte release každého nového kusu kódu, kdykoliv je připraven, a nezastavujte se na dvoutýdenních dávkách.
- Použijte princip, kdy každý pracovník nebo tým převezme úkol v momentě, kdy je připraven na něm pracovat.
- Omezte počet aktivních úkolů - tzv. „Work in Progress“. To zlepšuje soustředěnost, předvídatelnost a rychlost.

- Releasey nemusí být plánovány předem a dávány dohromady pod zátěží. Je možné použít přístup „pull what is ready“. Na začátku je možné prioritizovat a na konci naplánovat (sestavit) release.
- Pokud je to možné, zaměřte se především na správu kódu, ne lidí. Lidé (contributoři, developeři) vytváří kód téměř automaticky, pokud je jim dán správný prostor. Také je mnohem snadnější management kódů než management lidí.
- Provádějte automatizace využitím více strojů, abyste ušetřili práci lidí a zvýšili jejich produktivitu.
- Zvyšte produktivitu vynecháváním různých rituálů, jako jsou nepotřebné pravidelné meetingy. Překvapivě lze docílit stejných výsledků i bez těchto rituálů.

2. Matrix of Services

2.1. Role

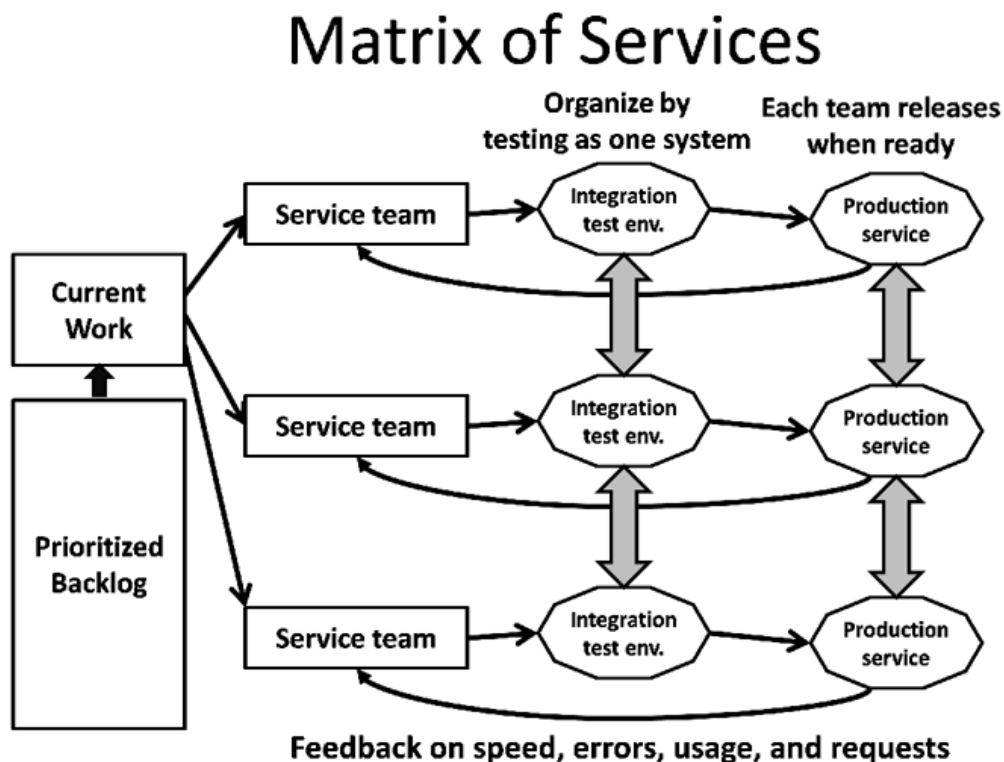
Popis Matrix of Services neposkytuje žádné přesné definice rolí, které mají být zastoupeny. Framework se zaměřuje především na existenci takzvaných servisních týmů. Podle Collabnet Versionone (2014) jsou týmy téměř nezávislé a nejsou multifunkční. Důležitá je orientace na vývojáře (týmy jsou developer-centrické). Developéři bývají v týmu oproti testerům v poměru 10:1, protože vývojáři vytváří automatizované testy.

Týmy mají velké kompetence. Vývojář sami rozhodují o tom, kdy proběhne release jejich nového kódu, na rozdíl od většiny jejich přístupů, kdy o tom rozhodují produktoví manažeři nebo QA testeři. MAXOS uvádí, že je tento přístup v zájmu samotných developerů, neboť je snazší předejít chybám v důsledku špatného načasování vydání kódu. Takové chyby mohou být například nedostačující otestování, nekvalitně napsaný kód nebo nedostačující review kódu.

Framework také podporuje silnou a rychlou komunikaci mezi členy v rámci jednoho týmu a komunikaci mezi jednotlivými týmy.

2.2. Rozdělení SW aplikace a infrastruktury

Webová stránka MAXOS uvádí, že softwarové aplikace a infrastruktura jsou v Matrix of Services rozděleny do malých služeb. Jednou službou může být načítání webové stránky e-shopu. Tato služba následně volá další službu, která poskytuje informaci o dostupnosti konkrétních produktů v e-shopu. Každá služba je přidělena právě jednomu týmu, který má zodpovědnost za design, programování, testování i release. Tento tým také kontinuálně dostává aktuální informace o službě, jejím využívání, výkonu, chybách i problémech nahlášených uživateli a sám je napravuje.

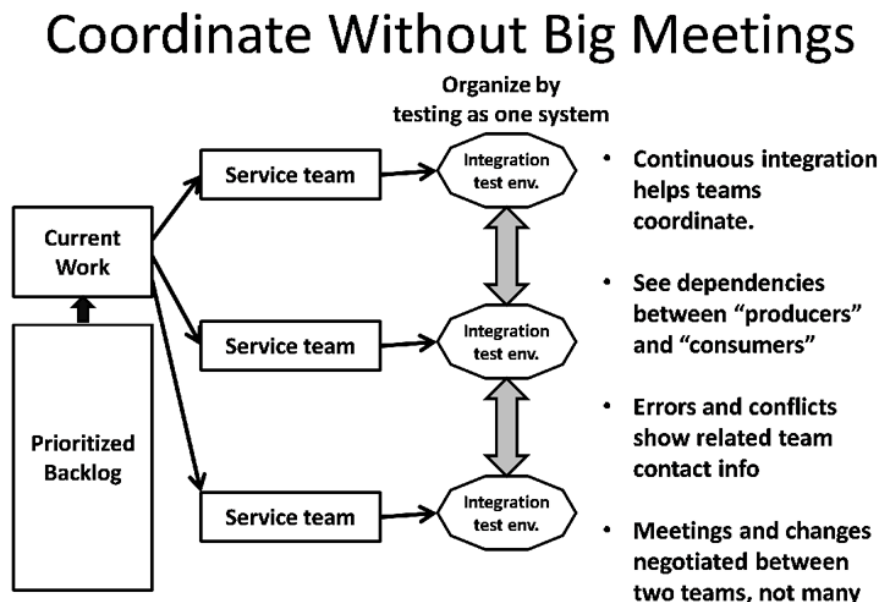


Obrázek 1 – MAXOS Overview
[http://www.continuousagile.com/unblock/ea_matrix.html]

2.3. Organizace týmů

Často se při koordinaci velkých projektů používají různé „ceremonie“ a meetingy, aby se ujasnily závislosti mezi prací jednotlivých týmů. Příkladem takového meetingu je Scrum of Scrums. Matrix of Services se tomu snaží vyhnout pomocí nepřetržité integrace. Ta probíhá díky neustálému ukládání nových částí kódu do společných integrovaných depozitářů. Tyto depozitáře slouží zároveň jako testovací systémy, které spouští automatické testovací skripty. Toto testování odhaluje problémy uvnitř služeb nebo při volání závislejších služeb. Automaticky následně informuje příslušné servisní týmy.

Tuto koordinaci nazývá stránka MAXOS trvalou integrací a uvádí, že tento způsob snadno odhaluje chyby v kódu a efektivně informuje zodpovědné týmy. Vzniká tak samostatná organizace práce bez meetingů a velkých zásahů z pozice manažerů.



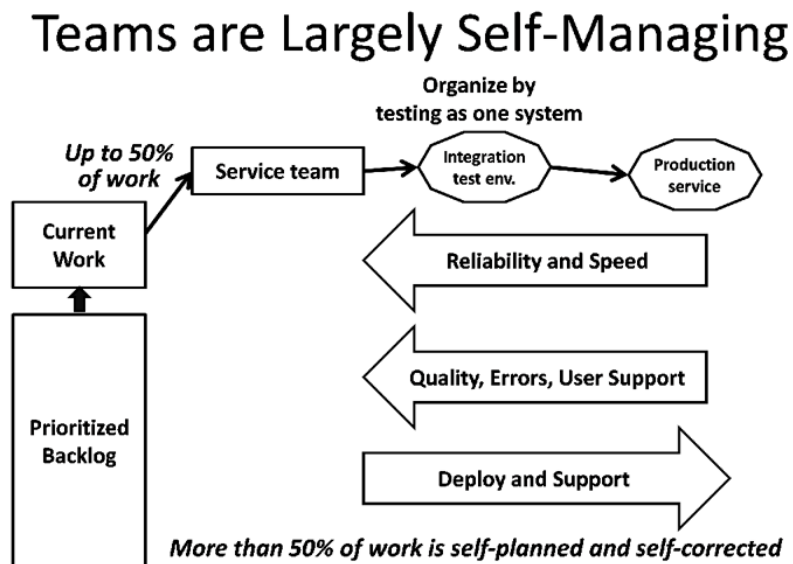
Obrázek 2 – MAXOS Self – Coordination
[http://www.continuousagile.com/unblock/ea_matrix.html]

2.4. Organizace práce

Matrix of Services podle webových stránek MAXOS funguje bez zásahů managementu nebo prioritizačních systémů. To je umožněno díky výše zmíněným automatizovaným testům, které konstantně zasílají informace o službách konkrétním týmům. Týmy si tak přesně definují své úkoly na základě feedbacků z testovacích systémů. Jsou také schopny reagovat na neplánované události rychle a kvalitně.

Touto organizací práce ušetří čas, který by manažeři a programátoři strávili zadáváním úkolů do seznamu, jejich kategorizací, prioritizací a výběrem. Díky tomu také odpadají velké meetingy, kde se úkoly a jejich návaznosti probírají.

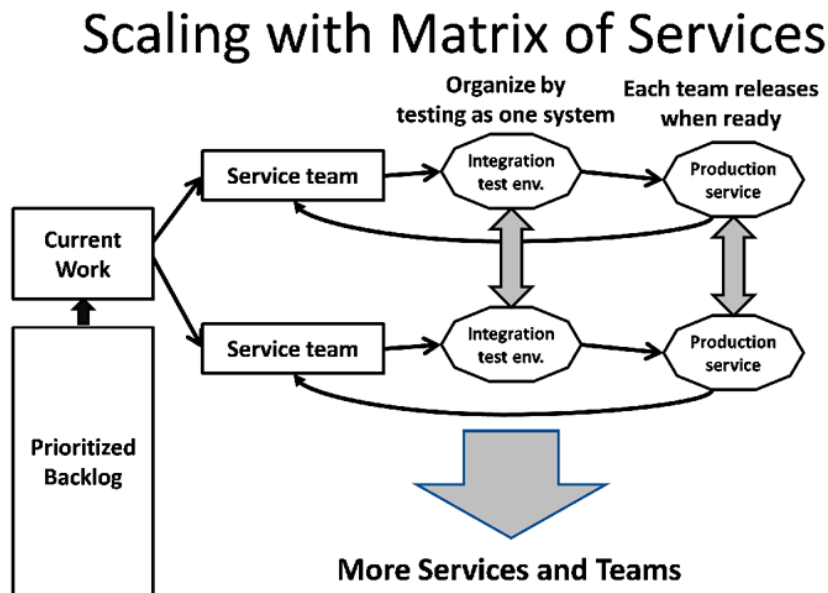
Programátoři mají sami možnost určit si, kolik nových požadavků přijmou. Stránka MAXOS poukazuje na fakt, že při vývoji software je většina času trávena opravami a je málo času na vývoj nových funkcí a architektur. Týmy, které řídí samostatně samy sebe, toto vnímají jako výhodu, jelikož se tak brání přetěžování a zaměřují se na skutečně důležité úkoly.



Obrázek 3 – MAXOS Self – Management
[http://www.continuousagile.com/unblock/ea_matrix.html]

2.5. Škálovatelnost

Organizace Matrix of Services je založená na servisních týmech a oddělených službách. Proto velmi jednoduše umožňuje škálovatelnost přidáním či odebráním servisních týmů nebo služeb podle aktuální potřeby. Samotné servisní týmy mají potom jednoduchou strukturu orientovanou na technické vedení a programátory. Stránka Maxos uvádí, že díky této jednoduché organizaci je rámec Matrix of Services vhodný jak pro malý tým deseti lidí, tak velké desetitisícové organizace.



Obrázek 4 – MAXOS Scaling
[http://www.continuousagile.com/unblock/ea_matrix.html]

3. Srovnání s rámcem SAFe

Srovnání dvou škálovatelných rámců, MAXOS a Scaled Agile Framework, provedl v roce 2014 Collabnet Versionone způsobem, který je aplikovatelný i na srovnání jiných škálovatelných frameworků. V první části bylo zvoleno šest aspektů škálování. Tyto aspekty jsou:

1. Týmy
2. Zákazníci / Uživatelé
3. Agilní metody a prostředí
4. Produkt
5. Složitost
6. Hodnotový řetězec

V rámci těchto šesti aspektů škálování byl sestaven velmi komplexní seznam 25 agilních parametrů. Tyto parametry byly rozděleny do šesti tabulek podle aspektů a posuzovány třemi hodnotami. Hodnota Sweet Spot, indikována zelenou barvou, znamená vhodnost rámce pro dané kritérium. Modře byly hodnoceny případy tzv. Challenging Zone. Toto hodnocení znamená, že rámec je v daném aspektu použitelný, ale jeho použití není pohodlné a znamená určitou výzvu při aplikaci. Třetím hodnocením může být Unfit Zone, které je ve srovnání značeno červenou barvou. Může znamenat, že aspekt není ve frameworku vůbec aplikovatelný, nebo je jeho použití velmi nevhodné.

Například u kritérií zabývajících se složením týmů, konkrétně kritérium Developer-centric, definuje MAXOS velmi dobře, proto získal hodnocení Sweet Spot. Naopak v SAFe by přístup se zaměřením na vývojáře byl velmi nevhodný, proto je hodnocen jako Unfit Zone.

Pro zjednodušení se tato práce zabývá především kritérii, kde jsou MAXOS a SAFe nejodlišnější, tedy kritérii, kde jeden z frameworků byl hodnocen Sweet Spot, a druhý Unfit Zone.

3.1. Týmy

Oba frameworky jsou vhodné pro menší týmy a pro případy, kdy jsou všechny týmy součástí jedné firmy nebo společnosti. Také podporují samostatně organizované týmy s minimální účastí vyššího vedení.

Aspekt	Parametr	MAXOS	SAFe
Geografické umístění týmů	Týmy nejsou na stejném místě, ale jsou ve stejném časovém pásmu.	Unfit Zone	Sweet Spot
Kompozice týmů	Týmy orientované na developery	Sweet Spot	Unfit Zone
	Týmy jsou cross-functional.	Unfit Zone	Sweet Spot
Hierarchie týmů	Týmy jsou autonomní.	Sweet Spot	Unfit Zone

Tabulka 1 - MAXOS vs. SAFe: Týmy

3.2. Zákazníci / Uživatelé

Oba frameworky podporují vývoj pro jednoho či více zákazníky, pokud jsou požadavky hned z počátku velmi jasné, nebo alespoň relativně jasné při větším snažení. Rozdíl mezi frameworky nastává, když si zákazník některé z požadavků rozmyslí.

Aspekt	Parametr	MAXOS	SAFe
Jasnost požadavků a potřeb zákazníka	Objevené a schválené potřeby uživatelů se mohou rychle změnit.	Sweet Spot	Unfit Zone

Tabulka 2 - MAXOS vs. SAFe: Uživatelé

3.3. Agilní metody a prostředí

V oblasti agilních metod a prostředí, ve kterých jsou používány, se MAXOS i SAFe lišily i shodovaly hned v několika bodech. Oba frameworky nefungují dobře zároveň se Scrumem. Zároveň oba frameworky upřednostňují automatizované testování a kontinuální integraci, dodání a deployment. Odlišnosti jsou zaznamenány v tabulce 3.

Aspekt	Parametr	MAXOS	SAFe
Agilní metody a nástroje	Scrumban	Unfit Zone	Sweet Spot
	Scrum + Lean	Unfit Zone	Sweet Spot
	Lean + Continuous Delivery	Sweet Spot	Unfit Zone
	Lean Startup	Sweet Spot	Unfit Zone
Kadence pro týmy	Lean-Flow (nejsou časově omezené iterace)	Sweet Spot	Unfit Zone
	Všechny týmy mají stejnou kadenci kroků	Unfit Zone	Sweet Spot
Frekvence dodání řešení	1 – 4 za rok	Unfit Zone	Sweet Spot
	Jednou denně	Sweet Spot	Unfit Zone
	Několikrát denně (continuous delivery)	Sweet Spot	Unfit Zone
Kontrola	Centralizovaná top-down	Unfit Zone	Sweet Spot
	Distribuovaná na základě integrovaného uložení	Sweet Spot	Unfit Zone
Adaptabilita vs. Schopnost predikce	Schopnost predikce velmi důležitá, adaptabilita méně důležitá	Unfit Zone	Sweet Spot
	Schopnost adaptability velmi důležitá, predikce méně důležitá	Sweet Spot	Unfit Zone

Tabulka 3 - MAXOS vs. SAFe: Agilní metody a prostředí

3.4. Produkt / Řešení

Matrix of Services i Scaled Agile Framework dobře fungují, pokud jsou používány při vývoji vysoce modulárních software nebo nepříliš propojených služeb, ve kterých jsou nízké dopady případných defektů. Naopak nejsou aplikovatelné při vývoji typu produktu, kde je úzce spjat hardwarem se software poskytovaným jako službou, což mohou být například software do aut, zdravotnických zařízení nebo pro armádu.

Aspekt	Parametr	MAXOS	SAFe
Modularita	Střední	Unfit Zone	Sweet Spot
Typ produktu / řešení	Zabalený SW instalovaný a spravovaný Customer IT oddělením	Unfit Zone	Sweet Spot
	On-line dodaný SW instalovaný a spravovaný Customer IT oddělením	Unfit Zone	Sweet Spot
	SaaS hostovaný Customer IT	Unfit Zone	Sweet Spot

Tabulka 4 - MAXOS vs. SAFe: Produkt / Řešení

3.5. Složitost

V kategorii komplexity nemají frameworky SAFe a MAXOS výrazné odlišnosti, naopak se spíše shodují. Oba frameworky dobře fungují, pokud technologie, se kterou je pracováno, je vyzkoušena a již s ní mají vývojáři mnoho zkušeností. Také dobře fungují se střední komplexitou. Matrix of Services už ale nefunguje dobře, pokud je technika velmi komplexní (nová nebo heterogenní). Scaled Agile Framework s takovým řešením pracovat dokáže, ale obtížně. To samé platí o doménové složitosti. Oba frameworky pracují dobře s nízkou a střední doménovou složitostí. Vysoká a velmi vysoká doménová komplexita je výzvou pro SAFe, MAXOS je pro vysoce doménově komplexní řešení nevhodný. Organizačně oba frameworky fungují dobře, pokud se jedná o samostatně organizované týmy.

3.6. Hodnotový řetězec

Frameworky Matrix of Services a Scaled Agile Framework jsou vhodné, pokud je ve společnosti kladen důraz na plánování a konstrukci, naopak nevyhovují při zaměření na marketing, prodeje a zákaznickou podporu. Největší rozdíl nacházíme v tom, jak často a jakým způsobem je sbírán feedback od uživatelů a zákazníků.

Aspekt	Parametr	MAXOS	SAFe
Hloubka hodnotového řetězce ve společnosti	Měření hodnot a analýza	Sweet Spot	Unfit Zone
	Občasný, získávaný manuálně, manuální analýza	Unfit Zone	Sweet Spot
Feedback od uživatelů	Zcela automatizovaný, obsahující i feedback pro developery, automatizované big data analýzy, automatizované reporty	Sweet Spot	Unfit Zone

Tabulka 5 - MAXOS vs. SAFe: Hodnotový řetězec

4. Použití Matrix of Services

Webová stránka MAXOS uvádí, že framework Matrix of Services je díky způsobu fungování na oddělených servisních týmech a službách použitelný stejně dobře pro malé týmy jako pro tisícihlavé. Collabnet Versionone ve svém srovnání udává, že počet týmů nad tisíc už by mohl být pro Matrix of Services výzvou.

Konkrétní případy firem, kde je Matrix of Services používán, jsou poměrně nejasné vzhledem k velmi limitovanému počtu zmínek frameworku mimo jeho oficiální stránku. Andy Singleton, zakladatel Matrix of Services, uvádí, že způsob Matrix of Services vynalezl a používal Amazon už v počátcích třetího tisíciletí, kdy hledal způsob, jak jednodušeji spravovat a dále rozšiřovat jejich vysoce komplexní systém.

Jako další příklad na blogu Andy Singleton uvádí HubSpot. Tam se aplikace Matrix of Services vyznačuje rozdělením marketing services na 200 rozdílných služeb, přičemž každá služba je stavěna, testována a spravována malým týmem. Takových týmů bylo v roce 2016 kolem třiceti, ve kterých pracovalo dohromady 90 programátorů.

Závěr

V semestrální práci se podařilo popsat strukturovaně škálovatelný framework Matrix of Services a způsob jeho fungování na základě rozdělení služeb a samostatných nezávislých servisních týmů, které jsou orientovány na programátory. Také bylo definováno, v čem spočívá škálovatelnost frameworku a jak jí docílit. V druhé části práce byly vymezeny silné a slabé stránky frameworku v porovnání s hojně používaným rámcem Scaled Agile Framework na základě porovnání provedeného týmem Collabnet Versionone. Zjištění Collabnet Versionone o slabinách i výhodách rámce Matrix of Services korespondovala s fakty zjištěnými z ostatních citovaných zdrojů. Cíl práce byl tedy úspěšně naplněn.

Reference

- Collabnet Versionone. 2014.** Scaling Agile Your Way: SAFe vs. MAXOS (Part 2 of 4). *Collabnet Versionone*. [Online] 18. Zář 2014. [Citace: 7. Duben 2020.] <https://resources.collab.net/blogs/scaling-agile-your-way-safe-vs-maxos-part-2-of-4>.
- Highsmith, Jim. 2001.** History: The Agile Manifesto. *Manifesto for Agile Software Development*. [Online] 2001. [Citace: 8. Květen 2020.] <http://agilemanifesto.org/history.html>.
- Linders, Ben. 2015.** Q&A with Andy Singleton on Unblock! A Guide to the New Continuous Agile. *InfoQ*. [Online] 10. Březen 2015. [Citace: 7. Duben 2020.] <https://www.infoq.com/articles/book-unblock-continuous-agile/>.
- MAXOS.** Continuous Agile Principles. *Unblock! A Guide to the New Continuous Agile*. [Online] [Citace: 8. Květen 2020.] http://www.continuousagile.com/unblock/renzoku_principles.html.
- . Future Vision. *Unblock! A Guide to the New Continuous Agile*. [Online] [Citace: 7. Duben 2020.] <https://guides.co/g/unblock-a-guide-to-the-new-continuous-agile/20539>.
- . Matrix of Services. *Unblock! A Guide to the New Continuous Agile*. [Online] [Citace: 7. Duben 2020.] http://www.continuousagile.com/unblock/ea_matrix.html.
- . MAXOS versus Scrum and SAFe. *Unblock! A Guide to the New Continuous Agile*. [Online] [Citace: 7. Duben 2020.] http://www.continuousagile.com/unblock/ea_not_safe.html.
- . Why Continuous Agile? *Unblock! A Guide to the New Continuous Agile*. [Online] [Cited: Duben 7, 2020.] <http://www.continuousagile.com/unblock/>.
- Power, Brad a Singleton, Andy. 2014.** How Tech Leaders Get Business Results With Large-Scale Continuous Delivery. Boston : autor neznámý, Červen 2014.
- Singleton, Andy. 2017.** Lessons from Zuckerberg, Page, and Bezos. *LinkedIn*. [Online] 12. Leden 2017. [Citace: 7. Duben 2020.] <https://www.linkedin.com/pulse/meet-apex-competitors-capitalism-andy-singleton/>.

Seznam Obrázků

Obrázek 1 – MAXOS Overview	6
Obrázek 2 – MAXOS Self – Coordination.....	7
Obrázek 3 – MAXOS Self – Management	8
Obrázek 4 – MAXOS Scaling	9

Seznam Tabulek

Tabulka 1 - MAXOS vs. SAFe: Týmy	10
Tabulka 2 - MAXOS vs. SAFe: Uživatelé	10
Tabulka 3 - MAXOS vs. SAFe: Agilní metody a prostředí	11
Tabulka 4 - MAXOS vs. SAFe: Produkt / Řešení	11
Tabulka 5 - MAXOS vs. SAFe: Hodnotový řetězec	12