

**VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE**

**FAKULTA INFORMATIKY A STATISTIKY**



Scaling Agile – Slice and Understand Together

<b>Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS</b>	
<b>Semestr</b>	LS 2018
<b>Autoři</b>	Jan Šrubař (xsruj02)
<b>– jméno, příjmení , xname</b>	Jiří Gabauer (xgabj07)
<b>Téma</b>	Scaling Agile - Slice and Understand Together

## Abstrakt

Tato práce řeší způsoby plánování s důrazem na škálované agilní plánování. Jsou zde velice stručně představeny zjednodušené metody plánování podle vodopádu a vrstev a pak detailněji popsána metoda Scalling Agile – Slice and understanding together, která je primárně zaměřena na postup při zpracování požadavků.

## Klíčová Slova

Scaling Agile,Škalované plánování, Use case camp, Slice and Understand Together

## Obsah

Abstrakt .....	2
Klíčová Slova .....	2
Obsah.....	2
1. Úvod .....	3
2. Agilní přístup.....	4
3. Představení Scaling Agile .....	5
4. Přístupy k plánování a k práci.....	6
4.1. Vodopád .....	6
4.2. Vrstvy.....	7
4.3. Plátky .....	8
4.4. Metoda Slice and Understand Together .....	9
4.4.1. Jednotlivé principy metody .....	9
4.4.2. Postup při use case camp – příklad využití metody .....	13
5. Porovnání metody s agilním přístupem .....	15
6. Závěr .....	16
Zdroje.....	16

## 1. Úvod

Tato práce se zabývá způsoby plánování a představení navrhovaného řešení plánování. Toto téma je velice důležité, protože bez správně sestaveného plánu je velice obtížné dosáhnout požadovaných výstupů projektu, a protože se stále ukazuje, že je to často slabé místo v celém procesu.

Horké téma v agilním vývoji softwaru je nyní škálování. Jak vytvoříme celou organizaci na agilní metodice? Neformální průzkumy prováděné Scrum Alliance ukazují, že většina agilních týmů napětí cítí mezi tím, jak fungují týmy, a tím, jak je běží zbytek organizace. Týmy a jejich supervizoři jsou frustrováni a chtějí vědět: je možné vytvořit celou organizaci podle agilních metodik? Můžeme najít způsob, jak tuto zátěž odstranit? Můžeme agilní metodiky škálovat? (Cottmeyer, 2013)

Mnoho organizací se zabývá způsoby, jak škálovat agilní implementace, ale může to být obzvláště náročné pro podniky, které nejsou v agilních praktikách zblhlé. Správné škálování tradičnějších Agile rámců může být obtížné pro organizace s nadměrným personálem nebo s větším množstvím jednotlivých týmů pracujících na jednom nebo více produktech současně. (Denning, 2016)

Škálovatelnost je schopnost systému, či procesu obsáhnout rostoucí počet objektů, zvládnout zvyšující se objem práce, rozšíření o dodatečné komponenty či mít k takovému rozšíření předpoklady. Škálování je metoda rozšiřování daného systému, či procesu o dodatečné komponenty a objekty. Dále pak úprava jeho praktik tak, aby odpovídaly a refletovaly novou velikost systému, či procesu, počet jeho elementů a objem práce.

(Zikmund, 2012)

Autor metodiky explicitně neuvádí, jaký typ škálování má pro svou metodiku na mysli. Nicméně vzhledem k tomu, že jeho metodika utváří z celku menší, samostatně uchopitelné dílky, i pomocí USE-CASE diagramů, dá se předpokládat, že se jeho metodika nachází v

produktovém škálování a můžeme ji využít pro škálování platformy nebo produktu, protože máme malé nezávislé celky, které můžeme přiřazovat navzájem nezávislým týmům.

Cílem práce je představit metodu škálovaného plánování, popsat další možnosti plánování a agilní přístup. Práce bude zaměřena především na popis jedné z metod škálovaného plánování, a to konkrétně na „Rozdělení a pochopení celku“ (Slice and understand together). Následně popíšeme rozdíly mezi klasickým agilním přístupem a touto konkrétní metodou. Abychom cíle dosáhli budeme čerpat z dostupných odborných článků týkajících se této relativně nové problematiky.

## 2. Agilní přístup

Vzhledem k tomu, že tento odstavec slouží především k tomu, aby bylo následně možné alespoň částečně porovnat klasický agilní přístup s metodou Slice and understand together, tak zde uvedu pouze hlavní principy při aplikaci agilního přístupu, které jsou podle mě porovnatelné i s výše zmíněnou metodou.

- Nejvyšší prioritou je vyhovět zákazníkovi průběžným dodáváním hodnotného SW.
- Jsou vítány změny v požadavcích, a to i v pozdější fázi.
- Lidé z byznysu a vývoje spolupracují denně po celou dobu projektu.
- Nejlepší způsob sdílení informací je osobní komunikace.

Tyto čtyři principy mají největší vztah i k výše zmíněné metodě. Pro úplnost uvedu další principy.

- Dodávky fungujícího SW jsou v intervalu týdnů až měsíců, s preferencí kratší periody.
- Projekty jsou budovány kolem motivovaných jedinců. Je jim vytvářeno vhodné prostředí, podporovány jejich potřeby a mají maximální důvěru, že odvedou dobrou práci.
- Hlavním měřítkem pokroku je fungující SW.
- Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.
- Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.
- Jednoduchost-umění maximalizovat množství nevykonané práce je klíčová.
- Nejlepší architektury, požadavky a návrhy vzejdou ze samo-organizujících se týmů.
- Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším,
- a následně koriguje a přizpůsobuje své chování a zvyklosti.

(Agi)

### 3. Představení Scaling Agile

Škálované plánování, včetně řezání a plánování, je pragmatickým přístupem, kterým lze řešit vaše plánovací výzvy. Škálované plánování využívá jako výchozí bod obecné strategické cíle organizace (cíl ve středu obr.1) a odtud zahrnuje čtyři úrovně plánování:

Řezání, Hlavní plánování, Big room planning, Plánování Sprintů

*Obrázek 1: Škálované plánování-4 úrovně <https://res.infoq.com/articles/making-scaling-agile-work-2/en/resources/2figure1-1509136389020.png>*

Zatímco různé škálované rámce poskytují užitečný rámec pro čtvrtletní big room planning, kde se všechny týmy a zúčastněné strany setkávají na pár dní, a přestože většina organizací ví, jak dělat plánování sprintů, mnohé mají potíže s tím, být na 100% připraveny na big room planning. Toto je místo, kde se dá využít škálované plánování.

#### Řezání

Poté, co máme celkovou představu o produktu, který chceme dodat, je čas rozdělit práci na menší části.

Stejně jako mnoho špatných způsobů, jak řešit řezání dortu (horizontálně, příliš velké / malé kusy) - a pouze jedna správná cesta (svisle od středu k okraji a dolů) existuje také mnoho nesprávných způsobů, jak uspořádat požadavky, a jen jedna správná cesta.

Existují (přinejmenším) tři způsoby, jak lidé rozdělují práci:

Vodopád

Vrstvy

Plátky

Stručně řečeno, chceme rozdělit všechny práce na menší části a chceme, aby každý kus byl jeden kus systému.

(Jepsen, 2017)

## 4. Přístupy k plánování a k práci

Existuje několik přístupů. V následujících kapitolách se je pokusím detailněji popsat.

### 4.1. Vodopád

Vodopádový přístup krok za krokem s předávkami mezi částmi projektu. Zde je příklad, vodopádového přístupu při pohledu na vývoj nového bankomatu:

Analyzujte kombinace bankovek, které lidé upřednostňují

Navrhněte celkové principy UX

Napište dokument o zahájení projektu (PID)

Nastavte strategii testu

Rozhovory s cílovými skupinami zákazníků

Šablony pro oznamování stavu

Hodnocení technologií

Přezkoumání a schvalování PID v řídicí komisi

Celková technologie

Příprava a kickoff projektu

Všechny tyto věci chápeme jako relevantní a důležité a často na začátku projektu budeme na začátku projektu potřebovat vyjasnění prvotní představy a parametrů produktu, než začneme vyvíjet. Někteří se na tuto počáteční fázi odkazují na jako sprint nula.

Výše uvedené je však seznam činností, nikoliv seznam věcí, které je třeba vyvíjet, aby se vytvořila hodnota pro zákazníka. Zkušenosti (a Agile) nás naučily, že mít "kusy věcí, které chcete vyvinout, aby zákazník byl spokojený", sloužili jako nejlepší možné stavební kameny pro naše plánování, nejlepší možné ovladače pro naši práci. Zaměřujeme se na to, aby byli naši zákazníci spokojeni. Během našeho vývojového úsilí budeme schopni poté provést potřebné množství analýzy, navrhování, testování, nasazování atd.

(Jepsen, 2017)

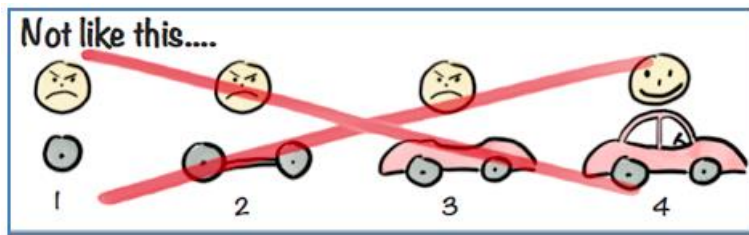
#### 4.2. Vrstvy

Další možností plánování je plánování podle vrstev, která se více zaměřuje na to, co musíme vybudovat. Např. nastavení serverů, vytvoření databází. Toto stále není ideální způsob, jak rozdělit práci, přesto, že tyto komponenty budeme potřebovat. Sami o sobě totiž zákazníka nezajímají.

Například sledováním této strategie bychom mohli vybudovat nejkrásnější uživatelské rozhraní, nechat je uživateli otestovat a možná si ho dokonce zamilují, ale později zjistíme, že bankovní systémy, které jsme připojili, nemají informace, které potřebujeme pro uživatelského rozhraní. Nebo bychom mohli vytvořit nejmodernější databázovou strukturu a připojit ji k systémům backendových bank se skvělými rozhraními a poté budovat uživatelské rozhraní a zjistit, že některé informace, které uživatel skutečně chtěl vidět (např. , který bankomat jsem naposledy použil nebo kterou částku jsem stihl stáhnout?) nebyly nikde uloženy.

Všechny tyto stavební bloky, jsou nezbytné k rozvoji a budeme je samozřejmě během vývoje potřebovat, ale nejsou to ideální stavební bloky pro samotné plánování. Jsou to vrstvy v architektuře a ta samotná nám nezaručí zákaznickou spokojenost. Situaci dokresluje následující obrázek, podle kterého je zřejmé, že zákazníka nezajímají jednotlivé díly bez

funkce, kterou požaduje, jak vidíme na obr.3.



Obrázek 3 Plánování podle vrstev <https://res.infoq.com/articles/making-scaling-agile-work-2/en/resources/1figure2-1509136389020.png>

(Jepsen, 2017)

#### 4.3. Plátky

Posledním způsobem, kterým se budeme zabývat, jakým rozdělit práci, je rozdělení na plátky. Zde je příklad krájení už použitého příkladu bankomatu

Vybrání prostředků z debetní karty

Vybrání prostředků z kreditní karty

Převést peníze

Zkontrolujte zůstatek

Vložit hotovost

Vybírat peníze v eurech

Vybírat hotovost v dolarech

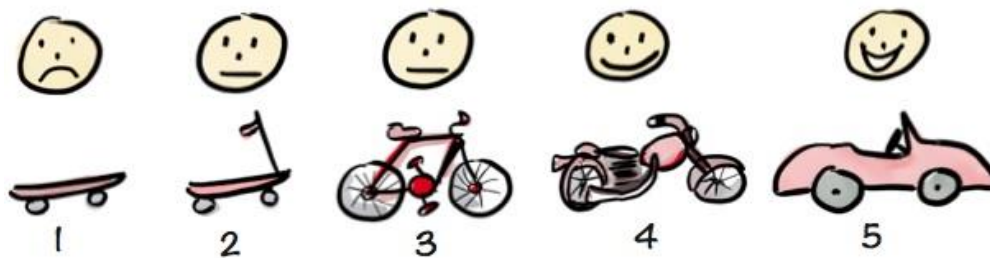
Dobrym způsobem, jak se dostat k tomuto způsobu krájení, je mít, během našeho krájení, jako řídicí otázku "Kdo využije systém a na co ho budou používat?"

To je druh rozpadu, který dělá dobrý plán. Každá z položek na tomto seznamu je něco, co může zákazník využít, viz obr.6. A jsou do značné míry nezávislé, abyste mohli vyvíjet jednu z



nich, nasadit ji a splnit, do nějaké míry, potřeby zákazníků.

## Like this!



Obrázek 6 Slices <https://res.infoq.com/articles/making-scaling-agile-work-2/en/resources/1Kniberg-Skateboard-1509141288752.jpg>

Uspořádání toho, co chcete tímto způsobem, vám umožní mít nejlepší možný základ pro vypracování plánu, který řídí vaše rozvojové úsilí směrem k "spokojenosti zákazníků".

(Jepsen, 2017)

### 4.4. Metoda Slice and Understand Together

Jedná se o metodu, která popisuje, jak správně postupovat při tvorbě a následné práci s požadavky. Metoda je zaměřena především na to, aby dodavatel správně pochopil zákaznickou představu o výsledném produktu a mohl mu tak dodat produkt, který se maximálně shoduje s jeho představou. Je zde tedy popsáno, jaké kroky postupně podniknout k tomu, aby došlo ke vzájemnému porozumění a pochopení požadovaného systému. Až v momentě, kdy se obě strany chápou lze správně nadefinovat požadavky. Jedná se tedy ve většině o činnosti ještě před samotným zahájením práce na samotném systému. Metoda se v jednom z principů okrajově dotýká tématu, jak systém dodávat. Stejně tak, jako metody popisované výše jde o to správně rozdělit práci a moci jí dále plánovat. Zde k tomu slouží správný popis požadavků. Pomocí těchto požadavků jsme schopni práci dělit a plánovat, protože každý požadavek popisuje jinou část systému.

#### 4.4.1. Jednotlivé principy metody

Metoda Slice and Understand Together obsahuje čtyři základní principy, které by měly být dodrženy. Nyní bych tyto principy rád popsal.

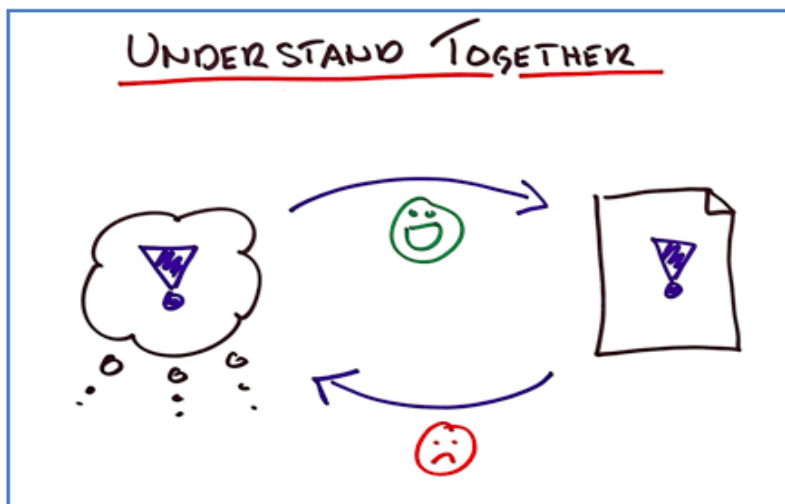
##### 4.4.1.1. Společné pochopení

Základní myšlenkou tohoto principu je, že by se na tvorbě požadavků měl podílet jak zákazník, tak i dodavatel. Tato idea je zásadní pro správnou aplikaci metody. Je tedy důležité, aby obě strany chápaly věci stejným způsobem. Je tedy zřejmé, že nejlepší způsob, jak se

správně pochopit je osobní setkání a diskuse o daném problému/systemu. Největší nedorozumění vzniká, pokud se na tvorbě požadavků podílí například pouze zástupci zákazníka. Dodavatel pak tedy obdrží pouze psaný dokument s požadavky, které si vyloží podle sebe.

Tento první princip metody Slice and Understand Together doporučuje, aby se nejprve sešli zástupci obou stran. Diskutovali spolu o požadavcích na budoucí produkt a případně si navzájem dovysvětlili své představy, pokud se zdá být něco nesrozumitelné. Tato skutečnost zobrazuje také Obrázek 11. Na něm lze vidět, že opačný postup, tedy vytvoření dokumentu s požadavky a až následná diskuse nemusí být vhodný.

(Jepsen, 2017)



Obrázek 11 Vzájemné pochopení představ ([https://www.infoq.com/articles/making-scaling-agile-work-2?utm\\_source=infoqWeeklyNewsletter&utm\\_medium=WeeklyNL\\_EditorialContent\\_culture-methods&utm\\_campaign=10312017news&utm\\_content=other](https://www.infoq.com/articles/making-scaling-agile-work-2?utm_source=infoqWeeklyNewsletter&utm_medium=WeeklyNL_EditorialContent_culture-methods&utm_campaign=10312017news&utm_content=other))

#### 4.4.1.2. Na kolik dílků dělit?

Na tuto otázku lze odpovědět hned na začátku popisu tohoto principu. Je vhodné mít 20-30 dílků. Důvod proč zrovna toto číslo je poměrně jednoduchý. Každý dílek by měl odpovídat určité části systému. Na toto číslo by neměla mít vliv velikost systému. Pokud bychom měli dílků méně, tak je zde velká šance, že nechápeme požadavky dostatečně do hloubky. V případě rozdělení na větší počet dílků se nacházíme na přílišném detailu, který bude zbytečně komplikovat plánování a bude snižovat přehlednost.

Je jasné, že dílky u větších systémů budou větší než u těch menších. Ale u obou je vhodné, pokud se povede dodržet oněch 20-30 dílků. Pokud by nastala situace, kdy máme příliš

mnoho malých dílků, tak je nutné tyto dílky vhodně spojit do větších. V opačném případě, je potřeba velké dílky dále rozdělit, abychom vždy dosáhli počtu 20-30 dílků.

Důležitost rozdělení systému na menší kousky lze demonstrovat na jednoduchém případu systému na rezervaci dovolené:

Představme si situaci, kdy jedním z dílků tohoto systému bude část, která umožňuje rezervaci pouze pobytů, které jsou jednotýdenní. Dále umí nabídnout pouze jeden hotel, jednu třídu v letadle (economy class) a lze platit pouze kreditní kartou Visa.

Je zřejmé, že dodání pouze takto okleštěného systému nemá pro zákazníka příliš velký obchodní přínos. Ale výhoda takového postupu je v tom, že lze zákazníkovi předvést část jeho budoucího systému a získat od něj zpětnou vazbu. Umožní nám to tak ujistit se, zda jsme se vydali správnou cestou a zda jsou naše základní představy shodné se zákazníkem. Zároveň jsou tak budovány základy každého jednotlivého dílku skládačky, která bude na konci prací spojena v systém. Většina nedorozumění tak může být nalezena a vyřešena již při tvorbě základů, a ne až při finální prezentaci konečného systému zákazníkovi.

(Jepsen, 2017)

#### 4.4.1.3. *Use cases místo user stories*

V agilních metodikách se obvykle používá pro psaní požadavků tzv. user story.

Nejprve uvedu příklad, jakým způsobem se user stories píšou. Určitým průvodním znakem user story je použití slova „jako“, „chci“, „abych“.

- a) „Jako bankovní klient chci mít možnost vybírat hotovost ze svého účtu, abych mohl platit u obchodníků, kteří neakceptují karty.“ - Tento příklad ukazuje poměrně jednoduše popsany požadavek, který neobsahuje příliš velký detail.
- b) „Jako bankovní klient chci mít vlastní klíč skládající se 4 čísel, aby se k mému účtu nedostal nikdo jiný.“ - Tento příklad je naopak hodně detailní.

Tento způsob psaní požadavku často vede k tomu, že každý user story je jinak detailní, příběhy se navzájem překrývají a neumožňují dostatečné porozumění a přehlednost. Proto je vhodné se zamyslet nad jiným způsobem psaní a uspořádání požadavků.

Využití use case pro psaní požadavků umožňuje lepší rozdělení na dílky. Dále také napomáhá v lepší přehlednosti a snazšímu pochopení požadavku, a tedy k eliminaci nedorozumění. Use

case umožňuje napsat požadavek v určité ucelené struktuře a dát mu tak nějaký řád. Use case obsahuje zpravidla nějaký název, což je obvykle sloveso a podstatné jméno. Například „Výběr hotovosti“. Dále obsahuje krátký popis, který by měl mít kolem 2-3 vět a je tedy delší, než user story. Poslední částí use case je use case diagram, který je vizuálním znázorněním požadavku.

Výhody psaní požadavků formou use case jsou:

- a) Mají pevně danou strukturu
- b) Umožňují lepší rozdělení systému na jednotlivé dílky
- c) Každý use case má jméno a práce s nimi je tak přehlednější
- d) Use case diagram umožňuje ucelený a vizuální pohled

Při psaní use case je důležité dát si pozor na to, aby se jednalo opravdu o use case a aby byl na správném levelu detailnosti. O zajištění toho, že se jedná o opravdový use case zajistí dodržení určitých struktur a vytvoření všech požadovaných částí takto napsaného požadavku. Pro zajištění toho, aby byl požadavek psaný pomocí use case na potřebné úrovni mohou pomoci tři rady.

- a) Lze jej vytvořit na jedno posezení
- b) Lze jej vytvořit během 2 – 10 minut
- c) Poskytuje uživateli hodnotu

(Jepsen, 2017)

#### 4.4.1.4. *Využití use case camps pro hlubší prozkoumání*

Jedná se o poslední ze čtyř základních principů celé metody Slice and Understand Together, který popisuje, jak využít use case camps pro hlubší prozkoumání situace. Dalo by se říci, že v tomto principu jsou obsaženy i všechny tři předešlé principy. Jedná se tedy o ucelený návod, jak metodu Slice and Understand Together použít a jak správně postupovat při jejím použití. Nejprve je nutné si říci, co to vlastně ten use case camp je. Tento pojem lze definovat dvěma způsoby.

- a) Ve zkratce se jedná o workshop, kterého se účastní jak zástupci byznysu, tak zástupci IT. Na tomto workshopu jsou detailně probrány požadavky. Díky účasti zástupců obou stran může dojít ke konfrontaci názorů a pohledů na řešenou situaci a obě strany by si měli navzájem osvětlit své úhly pohledu tak, aby nedocházelo k nedorozumění.

- b) Jedná se o 10% use case a 90% workshop. To znamená, že účastníci zde diskutují o tom, jak bude budoucí uživatel systém využívat. A právě díky účasti obou stran a využití odlišných perspektiv (perspektiva uživatelů/byznysu a perspektiva tvůrců systému) lze vyvolat hodnotnou diskusi jejímž cílem je správné rozdělení systému na jednotlivé části. To vše díky zapojení všech stakeholderů.

(Jepsen, 2017)

#### 4.4.2. Postup při use case camp – příklad využití metody

Byť si myslím, že vysvětlení pojmu use case camp je poměrně jasné, tak i přes to uvedu příklad, jak vytvoření a průchod takovým use case campem vypadá. Myslím si, že to přinese lepší představu o tomto nástroji. Je vhodné tento nástroj detailněji popsat i z toho důvodu, že se jedná o činnost, která trvá několik dní. Postup v use case camp lze rozdělit na čtyři etapy:

- a) **Nastínění situace** – Je nutné všem účastníkům znovu připomenout o čem vlastně jde, pro koho je systém tvořen a co se od něj očekává. Jedná se o stručné představení toho, co se bude řešit a jaký bude program use case campu.

Výstupem tohoto kroku jsou tedy účastníci, kteří jsou v dostatečné míře informováni, co je v příštích dnech čeká a mají minimálně základní informace o systému, který budou řešit.

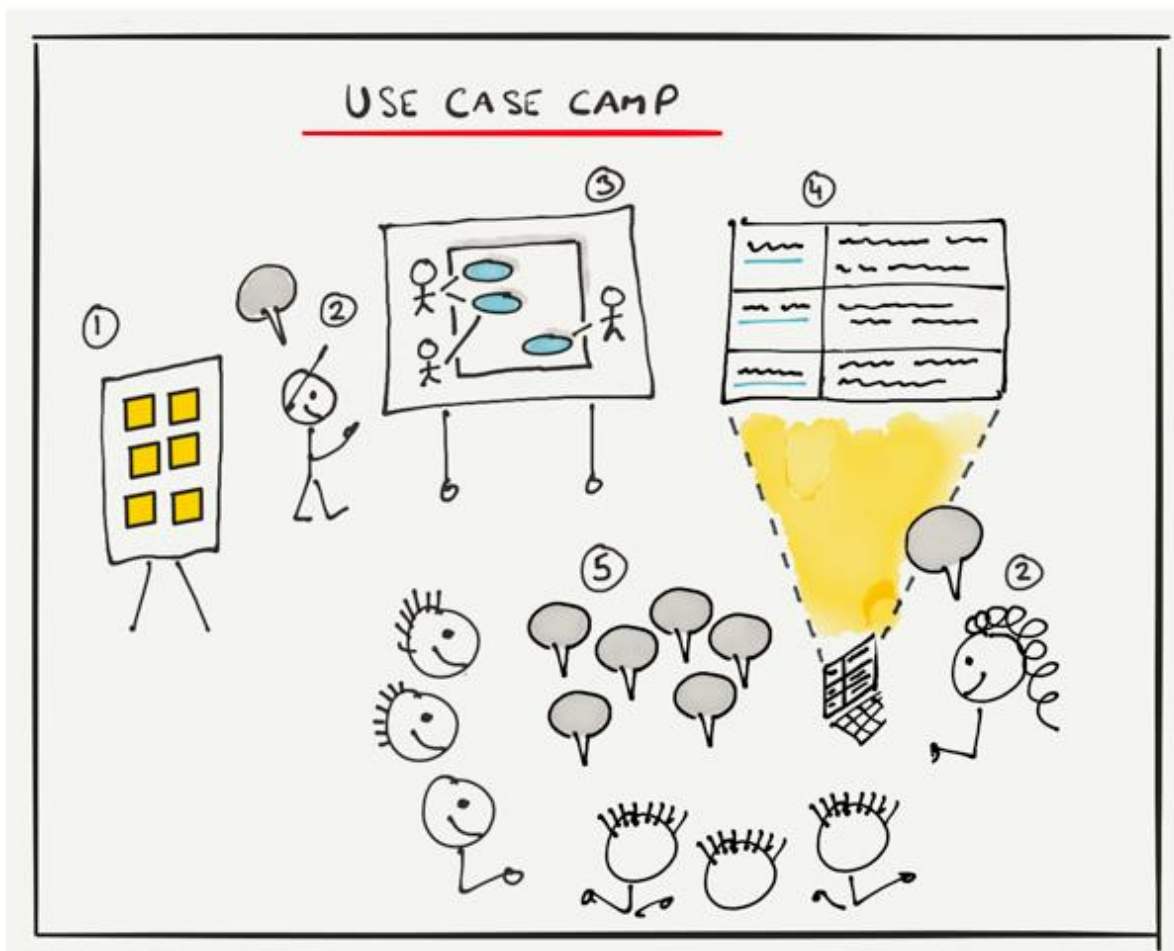
- b) **Brainstorm use cases** – Po představení situace dojde k rozdělení účastníků do skupin po 3-4. Je nutné, aby v každé skupině byl zástupce IT i byznysu. Každou tuto skupinu nyní požádáme o to, aby udělala brainstorming všech use case a poznámky si psali na samolepící papírky. Na tyto papírky týmy píšou také samotné use case. Jako podklady pro brainstorming týmy obdrží scope (jedná se o vše, co má být v projektu vyprodukováno), procesy a definici byznys cílů. Jedná se v podstatě o vše podstatné, co už bylo vytvořeno dříve. Jak vyplývá z pozorování, tak většina účastníků sází na své znalosti a zkušenosti a tyto poskytnuté dokumenty ve většině případů nevyužije.

Po skončení brainstormingu nalepí všechny týmy své samolepící papírky na stěnu, a to bez žádné ucelené struktury nebo pravidel. Je dobré se v tomto kroku vyvarovat diskuzím mezi jednotlivými týmy. Diskuzi je lepší odložit až na další krok.

Výstupem tohoto kroku je polepená zeď s poznámkami a jednotlivými use case od všech týmů. Tyto materiály jsou důležitým podkladem pro další krok.

- c) **Use case diagram a stručný popis** – Jedná se o nejdelší krok celé metody, který obvykle zabere minimálně kolem dvou dnů. Snahou je zde dostat se hlouběji do řešené situace, a to za pomoci diskuze a hledání nových poznatků.

Výstupem jsou hotové use case diagramy spolu s jejich stručným popisem, a to podle společně projednané představy všech stakeholderů.



Obrázek 12 Use case camp ([https://www.infoq.com/articles/making-scaling-agile-work-2?utm\\_source=infoqWeeklyNewsletter&utm\\_medium=WeeklyNL\\_EditorialContent\\_culture-methods&utm\\_campaign=10312017news&utm\\_content=other](https://www.infoq.com/articles/making-scaling-agile-work-2?utm_source=infoqWeeklyNewsletter&utm_medium=WeeklyNL_EditorialContent_culture-methods&utm_campaign=10312017news&utm_content=other))

Jak lze vidět na Obrázek 12 tak tento krok lze rozdělit do pěti etap.

1. Každý účastník si vezme jeden samolepící papírek ze stěny, který obsahuje use case z předešlého kroku. Zkontroluje, zda se jedná o jiný use case, než který již byl v diskusi probrán a poté jej včlení do diskuze.

2. Poté moderátor use case campu zorganizuje diskuzi o názvu pro takto vybraný use case.
3. Po domluvě názvu pro use case je use case namalován na tabuli. Ale nyní jsou k němu již připojeny i uživatelské role a jeho název.
4. Poté moderátor s jeho pomocníkem se připraví k tomu, aby byli schopni zaznamenat vše, na čem se účastníci workshopu domluví v kroku pět.
5. Posledním krokem je diskuze, kde se snaží účastníci domluvit. Moderátor spolu se svým pomocníkem se z této diskuze snaží dát dohromady stručný popis o 2-3 větách a vytvořit use case diagram.

**d) Vytvořená hodnota** – Nyní máme vytvořené všechny use case diagramy spolu s jejich stručným popisem. Je dobré si uvědomit, že přidaná hodnota celého postupu není pouze v tom, že nyní máme rozdělen systém/požadavky do správných dílků, ale v tom, že se nám povedlo dohodnout se se všemi účastníky workshopu na správném chápání všech požadavků a vnímání požadovaného systému je nyní jednotné, protože workshopu se účastnili všichni stakeholderi.

Výstupem jsou tak sladěné pohledy stakeholderů na požadovaný systém.

Ze zkušeností využívání této metody vyplynulo, že je vhodné uspořádání jednotlivých „místností“ workshopu podobně, jako je na obrázku. Tedy úplně vlevo stěna pro lepení papírků, uprostřed mazatelná tabule pro kreslení diagramů a vpravo tabule se stručným popisem. I takováto zdánlivě drobnost může napomoci pro hladký průběh celého use case campu.

(Jepsen, 2017)

## 5. Porovnání metody s agilním přístupem

Vzhledem k tomu, že metoda je zaměřena především na fáze před samotnou tvorbou systému/SW, tak se s klasickými principy agile lze porovnat pouze v těchto čtyřech bodech:

- a) **Nejvyšší prioritou je vyhovět zákazníkovi průběžným dodáváním hodnotného SW.**
  - o Slice and understand také říká, že je vhodné dodávat produkt průběžně, ale má rozdílný pohled na to, jak by tato část měla vypadat. Klasický agile klade důraz, aby byla dodána taková část produktu, která má pro zákazníka hodnotu. V metodě Slice and understand together je kladen důraz spíše na to, aby byla dodána klidně i ta část

produktu, která pro zákazníka nemá byznys hodnotu. Zde jde především o získání zpětné vazby na postup práce.

**b) Jsou vítány změny v požadavcích, a to i v pozdější fázi.**

- o Autor článku to sice nikde přímo nepíše, že by bylo možné měnit požadavky během práce, ale z některých částí článku jsem to tak pochopil. Slice and understand together se ale tuto potřebu změn během práce snaží odstranit již při popisu požadavků, a to díky pořádku use case kempu.

**c) Lidé z byznysu a vývoje spolupracují denně po celou dobu projektu.**

- o Při aplikaci metody Slice and understand together možná není tato spolupráce na denní bázi, ale i tak je poměrně vysoká. Nejintenzivnější je při use case kempu.

**d) Nejlepší způsob sdílení informací je osobní komunikace.**

- o Jak vyplývá z principů metody Slice and understand together, tak osobní komunikace je klíčem pro úspěch a je na ni kladen velký důraz.

Ostatní principy nelze porovnat, protože jsem nikde nenašel příklad z praxe, kde by byla metoda Slice and understand together aplikována na celý životní cyklus projektu.

## 6. Závěr

Cíl práce byl splněn. Omezením práce je, že se nezabývá do hloubky ostatními přístupy k plánování, které pouze hrubě nastiňuje. Jedná se o relativně nový přístup, a proto je k danému tématu limitovaný počet zdrojů, ať už teoretických, nebo praktických od firem, které se tímto přístupem řídí.

Myslím si, že se jedná o poměrně zajímavou metodu. Problém vidím ovšem v tom, že může být složité, aby si všichni stakeholderi dokázali udělat čas a workshopu se zúčastnili v požadovaném počtu a složení. Pokud se to povede, tak si myslím, že bude tato metoda velice prospěšná a může zabránit tomu, aby bylo potřeba dodaný systém zásadně měnit.

Porovnání metody Slice and understand together s klasickými agilními principy také komplikoval fakt, že nejsou dostupné žádné konkrétní případy využití metody v praxi a obávám se, že se jedná pouze o teoretickou metodu, která zatím nebyla aplikována.

## Zdroje

### Bibliografie

*Agile Manifesto*. [Online] <http://agilemanifesto.org/iso/cs/principles.html>.

**Cottmeyer, Mike. 2013.** How to Achieve Shared Understanding When Scaling Agile. *dzone*. [Online] 20. Prosinec 2013. <https://dzone.com/articles/how-achieve-shared>.



**Denning, Steve. 2016.** What does it mean to scale agile. *Forbes*. [Online] 15. Březen 2016.  
<https://www.forbes.com/sites/stevedenning/2016/04/15/what-does-it-mean-to-scale-agile/#322606bb78b9>.

**Jepsen, Ole. 2017.** Scaling Agile - Slice and Understand Together. *InfoQ*. [Online] 29. Říjen 2017.  
[https://www.infoq.com/articles/making-scaling-agile-work-2?utm\\_source=infoqWeeklyNewsletter&utm\\_medium=WeeklyNL\\_EditorialContent\\_culture-methods&utm\\_campaign=10312017news&utm\\_content=other](https://www.infoq.com/articles/making-scaling-agile-work-2?utm_source=infoqWeeklyNewsletter&utm_medium=WeeklyNL_EditorialContent_culture-methods&utm_campaign=10312017news&utm_content=other).

**Powell-Morse, Andrew. 2017.** How organizations scale agile. *airbrake*. [Online] 28. Listopad 2017.  
<https://airbrake.io/blog/sdlc/how-organizations-scale-agile>.

**Zikmund, Jan. 2012.** *Škálování agilních metodik , Bakalářská práce*. Praha : Vysoká škola Ekonomická v Praze, 2012.

#### Seznam obrázků

Obrázek 1: Škálované plánování <a href="https://res.infoq.com/articles/making-scaling-agile-work-2/en/resources/2figure1-1509136389020.png">https://res.infoq.com/articles/making-scaling-agile-work-2/en/resources/2figure1-1509136389020.png</a> .....	4
Obrázek 2: Plánování podle vrstev <a href="https://res.infoq.com/articles/making-scaling-agile-work-2/en/resources/1figure2-1509136389020.png">https://res.infoq.com/articles/making-scaling-agile-work-2/en/resources/1figure2-1509136389020.png</a> .....	6
Obrázek 3: Slices <a href="https://res.infoq.com/articles/making-scaling-agile-work-2/en/resources/1Kniberg-Skateboard-1509141288752.jpg">https://res.infoq.com/articles/making-scaling-agile-work-2/en/resources/1Kniberg-Skateboard-1509141288752.jpg</a> .....	7
Obrázek 4 Vzájemné pochopení představ ( <a href="https://www.infoq.com/articles/making-scaling-agile-work-2?utm_source=infoqWeeklyNewsletter&amp;utm_medium=WeeklyNL_EditorialContent_culture-methods&amp;utm_campaign=10312017news&amp;utm_content=other">https://www.infoq.com/articles/making-scaling-agile-work-2?utm_source=infoqWeeklyNewsletter&amp;utm_medium=WeeklyNL_EditorialContent_culture-methods&amp;utm_campaign=10312017news&amp;utm_content=other</a> )10	
Obrázek 5 Use case camp ( <a href="https://www.infoq.com/articles/making-scaling-agile-work-2?utm_source=infoqWeeklyNewsletter&amp;utm_medium=WeeklyNL_EditorialContent_culture-methods&amp;utm_campaign=10312017news&amp;utm_content=other">https://www.infoq.com/articles/making-scaling-agile-work-2?utm_source=infoqWeeklyNewsletter&amp;utm_medium=WeeklyNL_EditorialContent_culture-methods&amp;utm_campaign=10312017news&amp;utm_content=other</a> )14	