

Semestrální práce ke kurzu 4IT421

Semestr	LS 2015/2016
Autoři	Tomáš Kumsta (xkumt00) Ivan Matosyan (xmati06) Maroš Machaj (xmacm82)
Téma	Škálování agilních metodik
Datum odevzdání	13.5.2016

Obsah

Obsah	
Škálování agilních metodik	1
Abstrakt	3
Úvod	4
Cíle práce	4
Agilní metodika	4
Škálování	4
Důvody pro škálování	4
Vlastnosti agilních metodik	4
Výzvy ve velkých komplexních projektech	5
Rozšířenost agilních metodik	5
Obecné principy softwarového inženýrství	5
Škálování v praxi	6
Používané metody a přístupy ke škálování	6
Tipy pro úspěšné škálování	6
Možnosti škálování standartních metodik	6
Scrum	6
Činnosti a odpovědnosti jednotlivých členů v SCRUM týmech	6
Scrum of Scrums	6
Extrémní programování	7
Distribuovaný agilní vývoj	8
Škálování Extrémního programování	8
Pokročilé škálovací frameworky	10
Scaled Agile Framework 4.0 (SAFe)	10
Large-Scale Scrum	12
Disciplined Agile Delivery	13
Závěr	16
Literatura a zdroje	17

Abstrakt

Předmětem této semestrální práce je popsání škálování, důvodu pro škálování a popsání škálovaných agilních metodik, jak standartních tak pokročilých. Práce zahrnuje základní principy, účel a důvody pro škálování. V této práci jsou popsány pokročilé škálovací frameworky, s detailním popisem. V závěru práce je popsán současný stav na trhu s vývojem a shrnutí subjektivních výsledků práce.

Klíčová slova

Škálování, Scrum, Extrémní programování, Scaled Agile Framework, Large-Scale Scrum, Disciplined Agile Delivery

Úvod

Cíle práce

Agilní metodika

Agilní metodiky jsou skupiny metod původně určených pro vyvíjení softwaru založené na iterativním a inkrementálním vývoji. Umožňují rychlý vývoj softwaru a zároveň dokáží reagovat na změnu požadavků v průběhu vývojového cyklu. Podle těchto metodik se správnost systému ověří jedině pomocí rychlého vývoje, předložení zákazníkovi a následných úprav dle zpětné vazby. [Wiki]

Škálování

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 reflektovaly novou velikost systému, či procesu, počet jeho elementů a objem práce. [Zikmund]

Důvody pro škálování

Vlastnosti agilních metodik

Pokud chceme používat agilní metodiky na velkých projektech, tak charakteristické vlastnosti agilních metodik sami o sobě nám říkají, že se musíme zabývat přístupem pro škálování. Tyto vlastnosti následně vyvinutých metodik vycházejí zejména z Agilního Manifesta, uveřejněného v roce 2001. [BECK] Mezi hlavní omezení pro velké projekty vidíme velikost týmu, která je stanovena na 5-9 lidí a potřeba práce na jednom místě z důvodu časté komunikace mezi členy týmu a také směrem k zákazníkovi.

Mezi agilní předpoklady dále dle Ing. Aleny Buchalcekové Ph.D. podle [Zikmund] patří:

- dokumentace a modely nehrají při vývoji klíčovou roli
- požadavky a prostředí se v průběhu vývoje mění
- přizpůsobování procesu vývoje vede k vyšší kvalitě produktu
- vývojáři mají zkušenosti potřebné k přizpůsobování procesů
- viditelnost do procesu bude jen při ukončení přírůstku
- hodnocení produktu je neformální
- cílem není znovupoužitelnost řešení
- software může být vyvíjen inkrementálně

A právě z důvodů těchto předpokladů plynou dle Ing. Aleny Buchalceové Ph.D. podle [Zikmund] tyto omezení pro vývoj agilními metodikami:

- omezená podpora pro distribuované prostředí vývoje
- omezená podpora subdodavatelů
- omezená podpora pro vytváření znovupoužitelných artefaktů
- omezená podpora pro vývoj ve velkém týmu
- omezená podpora pro vývoj kritických aplikací
- omezená podpora pro vývoj velkého komplexního softwaru

Dle [Zikmund] mají výše uvedená omezení svá opodstatnění, například velikost týmu vychází z předpokladu že týmy budou sebe organizující. Velký důraz je také kladen na ad hoc komunikaci.

Výzvy ve velkých komplexních projektech

Dle [Zikmund] komplexní systém na rozdíl od agilních metodik vyžadují určité plánování, jako příklad je uváděna architektura, dále že není možné spoléhat na refaktoring kódu jako u malých projektů.

Pokud se jedná o kritický software, nese to podle [Zikmund] s sebou velké množství požadavků zejména na kvalitu a bezpečnost, na což agilní metodiky ve své původní podstatě nejsou připravené.

Rozšířenost agilních metodik

Jako důvod pro škálování agilních metodik bychom rádi zmínil i rozšířenost agilních metodik jako takových. Což zákonitě u rostoucích organizací vede k otázkám jak růst a zůstat agilní.

Dle průzkumu [VersionOne] až 95 % respondentů praktikuje v nějaké míře agilní přístupy s tím, že agilní metodiky používá více jak pět let až 33 % respondentů.

Obecné principy softwarového inženýrství

Dalším důvodem pro řešení škálování agilních metodik jsou samotné základní principy softwarového inženýrství, zmíníme zejména Brooksův zákon.

Dle Fredericka Brookse, který ve své knize The Mythical Man-Month [Zikmund] uvedl hypotézu, tzv. Brooksův zákon, který říká, že přidávání další lidské kapacity do zpožděných projektů, projekt jen dále zpozdí. Pokud tedy při agilním vývoji dojdeme k omezení zdrojů pro splnění termínů a budeme se snažit přidat další zdroje narušíme jak samotné principy agilního vývoje co do počtu členů týmu tak i můžeme očekávat platnost Brooksová zákona.

Škálování v praxi

Používané metody a přístupy ke škálování

Dle průzkumu [VersionOne] většina respondentů používá pro škálování agilního vývoje Scrum/Scrum of Scrums, popularitu každoročně nabírá SAFe, který si meziročně v roce 2015 oproti roku 2014 polepšil o 6 % a dostal se na 27 %. Mezi nejméně používané metody a přístupy patří u respondentů Recipes for Agile Governance in the Enterprise (RAGE) (1 %), Disciplined Agile Delivery (DAD) (4 %) a Large-Scale Scrum (LeSS) (6 %).

Tipy pro úspěšné škálování

Dle průzkumu [VersionOne] pro úspěšné škálování je klíčové konzistentní proces (43 %), implementace společného nástroje v týmech (40 %), přítomnost agilních konzultantů a trenerů (40 %), dále pak podpora managementů a interní tým pro podporu agilního přístupu.

Možnosti škálování standartních metodik

Scrum

Činnosti a odpovědnosti jednotlivých členů v SCRUM týmech

SCRUM tým se skládá z vlastníka produktu (Product Owner), vývojového týmu a SCRUM Mastro (SCRUM Master). SCRUM týmy se organizují sami z hlediska rozdělení úkolů v produktovém balíčku (Product Backlog) do jednotlivých sprintovat a tím představují multi-znalostní týmy. Samo-organizující se týmy si samy určují, jak nejlépe vykonají svoji práci. Multi-znalostní týmy charakterizuje zejména to, že takové týmy mají potřebné dovednosti a zkušenosti k tomu, aby mohli samostatně vykonávat práci. své úkoly mohou provádět bez závislosti na dalších lidech mimo týmu. V SCRUM je týmový model určen tak, aby optimalizoval produktivitu, hospodárnost, kreativitu a flexibilitu jednotlivých činností. SCRUM týmy vytvářejí produkt na základě iterační a inkrementální přístupu. Tento systém je vybudován tak, aby se maximalizovala možnost příležitostí pro zpětnou vazbu. Využití inkrementální dodávaného "hotového" produktu zajišťuje, že je stále k dispozici použitelná verze funkčního produktu.

Scrum of Scrums

Scrum of Scrums je poměrně jednoduchý způsob a funguje jako důležitá technika škálování pro větší projektové týmy. V praxi tato technika znamená rozdělení

projektového týmu na menší agilní týmy s menším počtem členů 5-10. Každý daily balon v rámci těchto menších týmů končí určením jednoho člena jako ambasadora, který se zúčastní daily setkání v rámci ostatních týmů. Ambasador nemusí být Master Scrum. Scrum of Scrums tak probíhá jako standardní daily meetingy kdy ambasadoři jednotlivých týmů reportují dokončené úkoly, další kroky překážky, zastoupená za daný tým. Důležitou součástí je koordinace mezi ambasadory a jejich týmy, kde se používá tzv backlog, kde každá položka přispívá ke zlepšení.

Zorganizovat efektivní Sprint review meeting s několika týmy, zákazníky a dalšími zainteresovanými stranami je dost náročné. Sprint review meeting probíhá podobně jako vědecké veletrhy. Každý tým si zřídí stanoviště, kde předvede, na čem členové týmu během Sprintu pracovali. Zákazníci, management a další zainteresované strany vytvoří malé týmy. Každý z těchto posuzovatelských týmů pak začíná na jiném stanovišti, kde jim každý tým během 15 minut předvede vlastní výstup Sprintu. Takové prostředí je nabito energií, vzrušením a zábavou [Schatz, 2009]. Dostat všechny do jedné místnosti je skvělý způsob jak komunikovat a sdílet. Pokud není možné uspořádat takový Sprint Review meeting v budově společnosti, je potřeba zvážit alternativní umístění jako například konferenční sál [Pichler, 2010].

Extrémní programování

Co je základem Extrémního programování? Samotný autor metodologie Extrémního programování Kent Back stanovil čtyři základní pojmy Extrémního programování [4]:

1. jednoduchost
2. komunikace
3. zpětná vazba
4. odvaha

Jednoduchost spočívá v pravidle dělat všechno tak jednoduše jak se jen dá. Udělat jen to, co zákazník opravdu chce a nedělat to co by zákazník mohl eventuálně chtít v další iteraci, i když víme, že to bude chtít. Například, pokud zákazník chce již zmíněnou třídu číslo a operaci sčítání, tak toto pravidlo nám káže implementovat pouze operaci sčítání a ne operace násobení nebo dělení, i když jsme si jistý, že v další iteraci je bude zákazník chtít implementovat.

Komunikace patří při této metodice k silným podmínkám úspěšnosti, proto i samotná metodika předpokládá jisté podmínky, jako je umístění týmu v jedné místnosti a podobně.

Zpětná vazba je v této metodice zajištěna silnou participací zákazníka na vývoji, doporučuje se dokonce, aby se reprezentant zákazníka stal členem týmu. Dále je zajištěna tím, že se v podstatě jedná o vývoj řízený testy, jak uvidíme v dále. A v neposlední řadě proto, že jsou stanoveny krátké vývojové iterace, přičemž na konci každé vzniká nějaká doručitelných a otestovatelná funkcionality.

A poslední z pojmů, **odvaha**, naráží na odvahu jednotlivých členů týmu. Odvahu pro změnu, protože samotný vývoj se začíná od jednoduchého návrhu, který se postupně zlepšuje refaktoriací.

Distribuovaný agilní vývoj

Charakteristiky distribuovaného agilního vývoje. Snížení objemu a kvality komunikace - průzkumy v této oblasti ukázaly, že se vzdáleností se snižují oba tyto ukazatele. Je to způsobeno i tím, že je složitější vést neformální diskusi, která je neoddelitelnou částí každého softwarového projektu. 2. Zpoždění v rozhodnutích a distribuci informací - distribuované projekty mohou mít členy, kteří jsou z různých časových pásem, tím pádem pracují v různých časech, je složitější uspořádat on-line setkání a podobně. 3. Většina distribuovaných projektů je i rozsahem a počtem členů týmu velkých. 4. Kulturní rozdíly - protože software je vyvíjen lidmi, je vývoj softwaru silně vázán na lidskou povahu, která je silně ovlivněna i kulturou. Proto se kulturní rozdíly staly jedním z faktorů ovlivňujících vývoj softwaru. Jak bylo zmíněno v první části této eseje, Manifest agilních metod obsahuje 12 základních principů. Z těchto 12 jsou zejména následující tři ovlivněny distribuovaným vývojem: 1. Individuality a interakce - nedostatek komunikace opsaný výše dělá těžším zejména výběr správných individualit, které by byly schopny udržovat efektivní komunikaci mezi sebou. 2. Spolupráce se zákazníkem - blízké vztahy se zákazníkem jsou také ovlivněny distribuovaností projektu, což vyúsťuje opět do menšího objemu a efektivity komunikace. 3. Odpověď na změnu - protože samotné distribuované prostředí přináší extra zdržení, prodlužuje se čas potřebný na změnu.

Extrémní programování a distribuovaný vývoj V této části se pokusím o modifikaci metody extrémního programování tak, aby odpovídal podmínkám distribuovaného vývoje. Tímto vzniká takzvaný

Distribuované extrémní programování. Charakteristikami extrémního programování, které jsou ovlivněny distribuovaností projektu jsou: - plánování hrou - s pomocí videokonferencí mohou být všichni členové týmu a zákazníci zapojeni do procesu vývoje - neustálá integrace - jednotliví členové týmu sdílejí části kódu nad nimiž vytvářejí a spouštějí testy - programování v párech - Vzdálené programování v párech, kdy jsou jednotliví členové vývojového páru umístěni v rozdílných lokalitách, je řešeno rozdělenými pracovními plochami vývojového prostředí - zapojení zákazníků - z důvodu co nejužší spolupráce se zákazníkem se pořádají videokonference se zákazníkem a členy vývojového týmu. K tomu, abychom mohli vyvíjet metodou distribuovaného extrémního programování potřebujeme minimálně tyto prostředky [3]: - členové týmu musí být schopni mluvit jedním jazykem - potřebujeme prostředek na výměnu informací, stanovování rozvrhů, to znamená nějaký komunikační nástroj - prostředek pro management kódu a kolektivní vlastnictví kódu - nástroj umožňující programování v párech, například nástroj pro on-line sdílení pracovní plochy - až neposlední míře nástroje pro on-line komunikaci, například audio a video konference. Kromě problémů s distribuovaností projektu, je třeba se zaměřit i na věrnost projektu, to znamená na rozsah projektu a věrnost týmu. Nejdůležitějšími charakteristikami ovlivněnými rozsahem projektu jsou: - ústní dokumentace

Škálování Extrémního programování

Škálování Extrémního programování naráží hlavně na problémy v ústní dokumentaci, kdy větší část dokumentace je řečena pouze ústně a uložena pouze v hlavách a v paměti jednotlivých vývojářů. Přesně z tohoto důvodu není nutné vytvářet složitou dokumentaci, která by nebyla dodána zákazníkovi a tak by sloužila pouze pro vnitřní potřeby týmu. Problémem se však stává,

pokud pro velikost týmu není možné umístit tým v jedné místnosti, podobně jako v případě distribuovaného vývoje.

tento problém je možné tříšit opačným přístupem, kdy se upustí od zkratky Yang a vytvoří se role správce dokumentace, proč tato úloha se bude považovat Prostě za další z požadavků, i když d není od zákazníka. Tento správce dokumentace je zodpovědný za vytvoření dokumentace, proč stále platí zásada udělatka jen toho, co je potřebné. Tato dokumentace je přesně v takovém rozsahu a na takové úrovni abstrakce, aby pomohla všem členům týmu pochopit cíle a požadavky stanovené v projektu a umožňovala vytvoření ústní dokumentace v rámci týmu.

Distribuovaná komunikace a Extrémní programování Jak bylo již několikrát zmíněno, komunikace má klíčovou roli ve všech agilních metod. Stejně bylo zmíněno, že Extrémní programování a konkrétně jeho metodu programování v párech představuje jednu z komunikačně nejnáročnějších aktivit v rámci agilních metod. Důvodem je úzká spolupráce dvou socializací, které pracují na stejných částech projektu, dokonce lo5du. Nicméně výsledkem jejich snažení je jen jedna implementace této části, na které se musí dohodnout a společně ji naprogramovat. Proto se vztahuje obecně lze říci, že pokud funguje Programování v párech i distribuovaně, tak fungují i ostatní praktiky používané v agilních metod. Z tohoto důvodu byl proveden zajímavý experiment. Tohoto experimentu se zúčastnilo 134 studentů, kteří byli rozděleni do čtyř týmů:

1. 9 skupin, které nebyly distribuovány a nepoužívali programování v párech
 2. 16 skupin, které nebyly distribuovány a používali párové programování
 3. 8 skupin, které byly distribuovány a nepoužívali párové programování
 4. 5 skupin, které byly distribuovány a používali párové programování
- Jednotliví členové týmů se vybírali sami z studentů a černé bylo naprogramovat jednoduchá hra v Javě. Na měření produktivity byl použit parametr Počet řádků loadu za hodinu a pro měření kvality softwaru byl použit parametr získané známky za tento projekt.

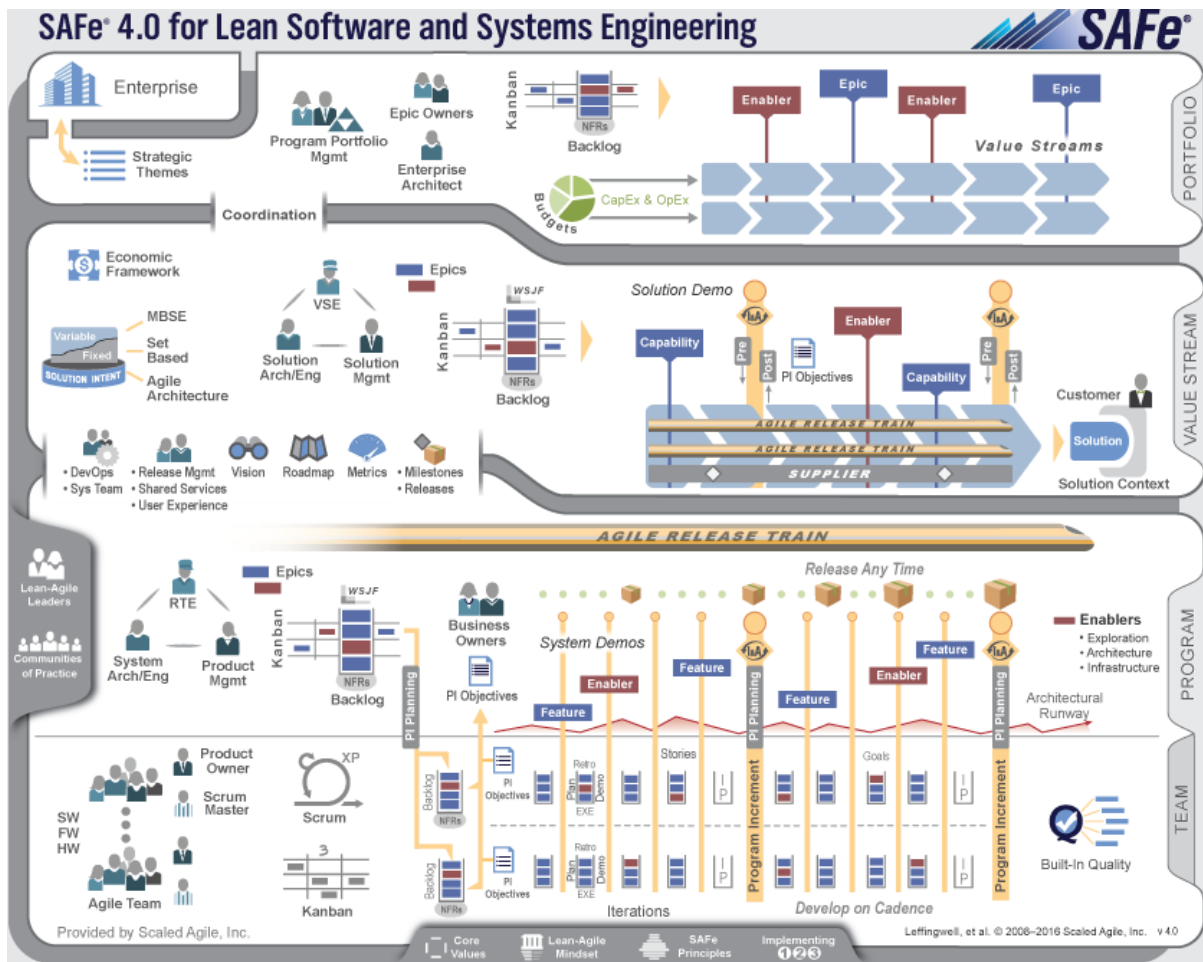
Pokročilé škálovací frameworky

Scaled Agile Framework 4.0 (SAFe)

Jedná se o framework vyvinutý Deanem Levingvelem a společností Scale Agile Inc. Jeho první verze vznikla v roce 2012. Celý framework je postavený na Agilních a Lean principech a jeho hlavním cílem je použití v podnicích, kde IT oddělení mají více, jak tisíc zaměstnanců a stovkách lidí zabývajících se přímo vývojem softwaru, kde už dříve nebylo možné používat standardní agilní metodiky, ale dá se efektivně využít i v týmech o 50-100 lidech. Metodika je dostupná zdarma na stránkách společnosti, kde nabízí také školení a certifikaci. Prošla si čtyřmi hlavními verzemi, které přidávají další úrovně nad řízení vývojových týmů a díky tomu je velmi detailní.

SAFe je založen na 9 základních principech které vycházejí z Agilních a Lean principů:

- Take an economic view
 - Řešení a jeho změny musí brát v úvahu ekonomické cíle.
- Apply systems thinking
 - Použití systémového myšlení a pohledu
- Assume variability; preserve options
 - Předpokládejte změny a zachovejte jejich možnosti
- Build incrementally with fast, integrated learning cycles
 - Vyvíjejte inkrementálně s rychlými integrovanými učitelskými cykly
- Base milestones on objective evaluation of working systems
 - Milníky založená na objektivním hodnocení funkčních systémů
- Visualize and limit WIP, reduce batch sizes, and manage queue lengths
 - Vizualizace a omezení práce v procesu, zmenšení velikosti dávek a správa fronty požadavků
- Apply cadence, synchronize with cross-domain planning
 - Rytmus v procesu, synchronizace plánu napříč úrovněmi
- Unlock the intrinsic motivation of knowledge workers
 - Využijte vnitřní motivace znalostních pracovníků
- Decentralize decision-making
 - Decentralizace rozhodování (s výjimkou strategického rozhodování)



Obrázek X:SAFe 4.0 Big Picture

Na obrázku jsou vidět 4 hlavní úrovně: Team, Program, Value Stream, Portfolio. Portfolio popisuje nejvyšší nejvyšší úroveň SAFe, kde se řeší zařazení portfolia produktů do obchodní strategie podniku. Ve větších společnostech může být i více programových portfolií s vlastními instancemi SAFe. Value stream jsou dlouhodobé kroky k vývoji a rozvoji, které vedou k nepřetržitému přísunu hodnot a řešení k zákazníkovi. Programová úroveň se zabývá řízením týmů v tzv. Agile Release Train, jehož velikost se pohybuje mezi 5 -12 týmy a slouží k doručení hodnot do Value Stream. Cílem je v inkrementálních iteracích rozvíjet software a dodat ho jako plně otestovaný, funkční software. Team je nejnižší úrovní a stará se o jednotlivé týmy v release trainu. Je úzce provázaný s úrovní Program. Týmy využívají ScrumXP (kombinace praktik projektového řízení Scrum a praktik extrémního programování pro samostatný vývoj). Jejich zodpovědností je dodat plánované změny v jednotlivých sprintech, kterými se řídí všechny úrovně. Na všech úrovních je také využíván Kanban, který zlepšuje orientaci a řízení v rámci celého procesu.

Scaled Agile Framework byl implementován například ve společnostech Seamless (rozvoz a objednání jídla z různých lokálních restaurací), Lego (výrobce stavebnic), Tomtom (výrobce navigačních zařízení a softwaru), Intel (jeden z největších výrobců čipů). Po implementaci došlo k následujícím zlepšením byznysu:

- Zvýšení produktivity o 20% - 50%
- Nárůst kvality o 50%
- Zkrácení času pro uvedení na trh o 30% - 75%

- Zvýšení spokojenosti zaměstnanců

Large-Scale Scrum

Byl vypracován Basem Voddem a Craigem Larmanem na základě jejich dlouholetých zkušeností s implementací Scrum. Metoda LeSS pak byla sepsána a zveřejněna v roce 2013.

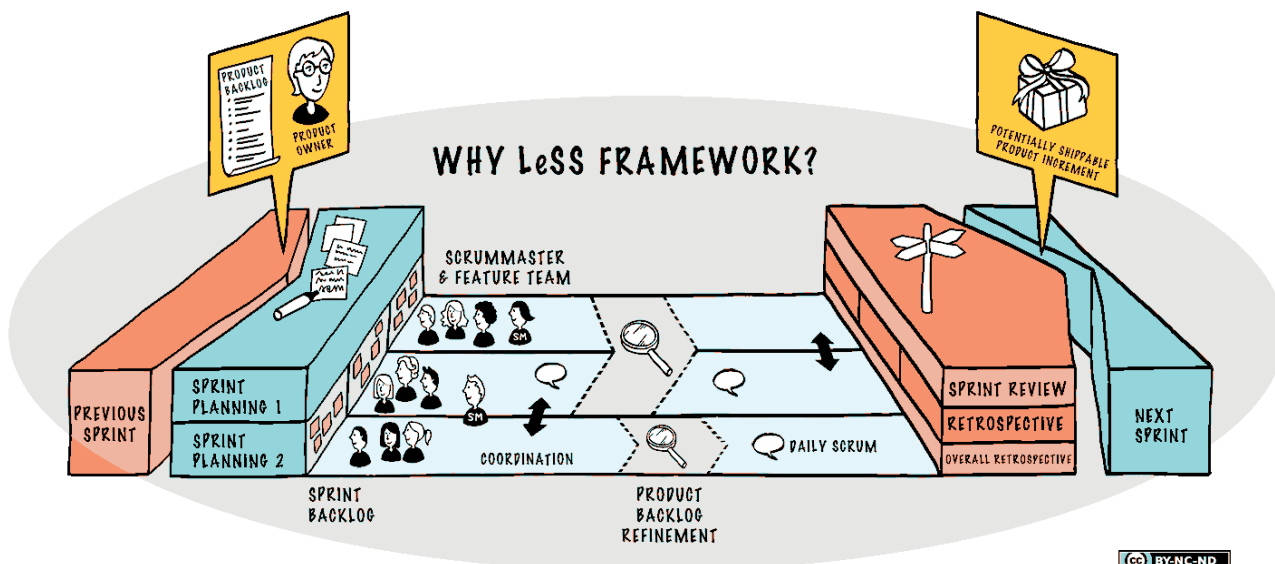
Large-Scale Scrum (LeSS) nabízí 2 různé frameworky pro škálování Scrum při zachování jeho výhod. Jeden je označován jako LeSS a je omezený na maximum osmi týmů po osmi lidech.

Druhý se nazývá LeSS Huge, který může být použita až pro tisíce lidí starající se o jeden produkt. Hlavními rozdíly LeSS Huge je rozdělení do oblastí, což vede k tvorbě nových rolí, změnám v požadavcích a struktuře porad. Klíčovou myšlenkou je, že LeSS je postaven na pochopení cíle jedno-týmového Scrumu a dosáhnout stejného cíle ve větším měřítku při zachování standardních pravidel Scrumu. Metodika není natolik detailní a umožňuje přizpůsobení podmínkám přímo ve firmě.

LeSS je postaven na následujících principech:

- Large-Scale Scrum is Scrum
 - Nejde o o nový a vylepšený Scrum, ale jeho použití ve větším měřítku.
- Empirical process control
 - Zkoumání a adaptace produktu, procesů, návrhu a praktik pro lepší přizpůsobivost místo následování detailního vzoru.
- Transparency
 - Nutná pro posuzování a zlepšování.
- More with less
 - Více vlastnictví, účelu a radosti a méně rolí, plýtvání, speciálních skupin
- Whole-product focus
 - Zaměření na jeden produkt nezávisle na počtu týmů.
- Customer-centric
 - Identifikujte hodnoty a plýtvání podle pohledu zákazníka. Každý má chápat, jak jeho práce přímo ovlivní platícího zákazníka.
- Continuous improvement towards perfection
 - Tvorba a doručení produktu na čas a bez defektů spokojenosti zákazníka. Zlepšujte v každém sprintu.
- Systems thinking
 - Sledování, porozumění a optimalizace systému jako celku.
- Lean thinking
 - Organizační systém na manažerech-učitelích používajících systémové a Lean myšlení za cílem zlepšování k dokonalosti.
- Queuing theory
 - Porozumění frontám a správa jejich velikostí, multitasking, variabilitu

<https://less.works/less/principles/index.html>



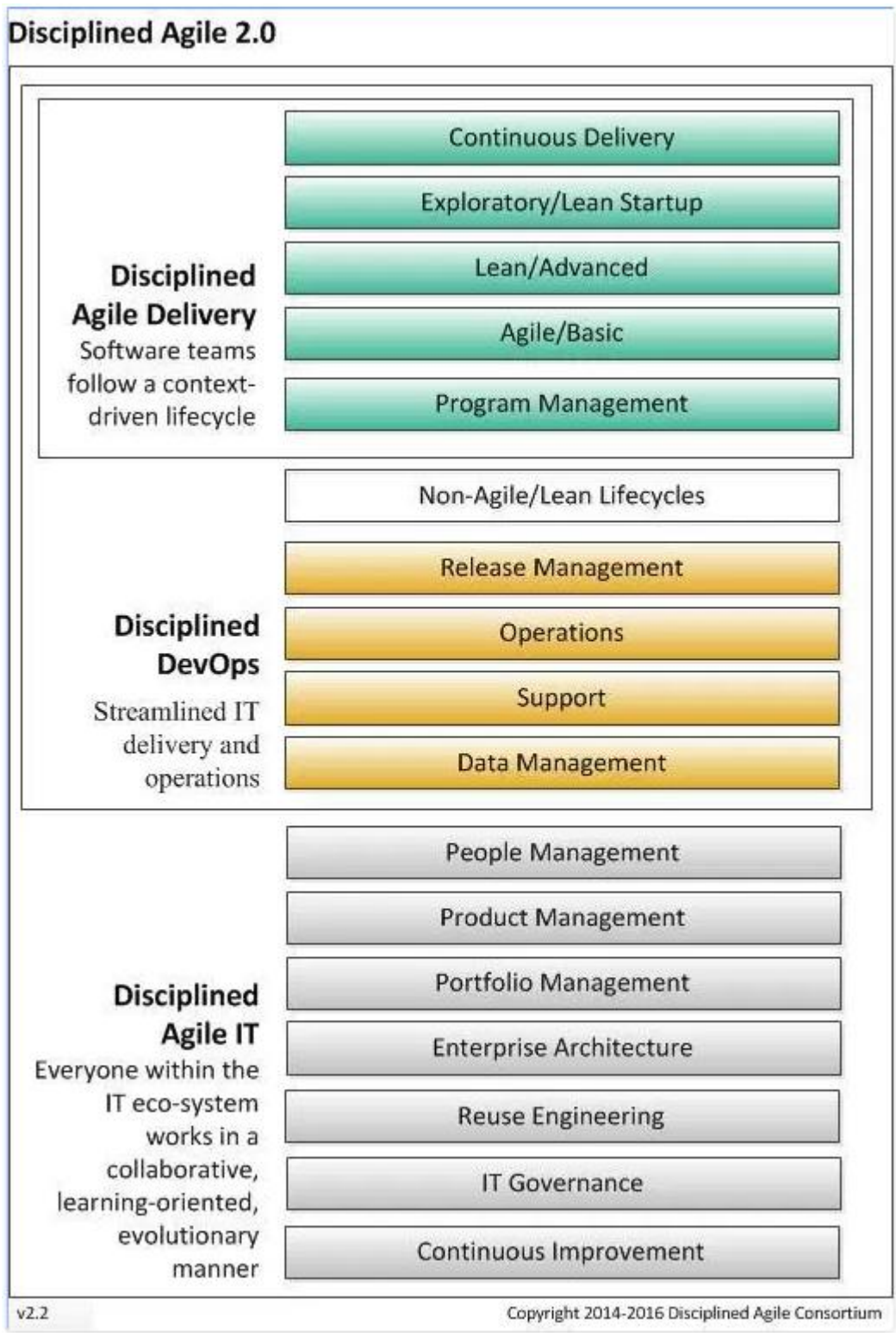
Obrázek X+1: Less Framework

Na obrázku můžeme vidět jak probíhá sprint v rámci LeSS frameworku. Plánování sprintu se skládá ze dvou částí. V Sprint Planning Part 1 se sejdou zástupci (maximálně 2 za tým) jednotlivých týmů a Product Owner rozdělí si úkoly do týmů a dohodnou se na koordinaci podle závislostí úkolů. Sprint Planning Part 2 už probíhá po jednotlivých týmech, i když v rámci koordinace může být přítomen člen jiného týmu, a naplňuje dané úkoly. Dále pak probíhá Daily Scrum, který je znovu držen samostatně každým týmem, ale může se ho znovu zúčastnit zástupce jiného týmu. Product Backlog Refinement (PBR) je schůzka obvykle v rámci týmů a obvykle v půlce sprintů se zabývá rozdělením větších úkolů, odhadem a plánováním úkolů a identifikací úkolů vyžadujících spolupráci. Užitečnou variantou pro LeSS může být PBR pro více týmů zároveň pro lepší učení a koordinaci. Na konci sprintu dojde ke Sprint Review, kde se zhodnotí a prozkoumají jednotlivých inkrementy. Je doporučeno použít styl veletrhu, kde jednotliví členové týmu představují nově vyvinuté části. Retrospective probíhá nejprve v rámci týmu stejně jako v normálním Scrumu a navíc proběhne také Overall Retrospective, která se místo na zlepšení běhu týmu zabývá prozkoumáním možností zlepšením celého systému.

Large-Scale Scrum byl implementován například ve společnostech Alcatel-Lucent (telekomunikační společnost), John Deere (výroba zemědělských strojů), Bwin.party (online hazardní hry).

Disciplined Agile Delivery

Disciplined Agile Delivery (DAD) je framework zabývající se dodáním IT v rámci velkých podniků. Byla vytvořena Scott Amblemem a Markem Linesem a popsána v knize Disciplined agile delivery z roku 2012. V srpnu 2015 vyšla nová verze s upraveným názvem jako Disciplined Agile 2.0. Pro změnu názvu se rozhodli z důvodu změny záběru frameworku, který se v nové verzi, kterou si dále představíme, zabývá nejen na efektivitu dodání IT řešení ale efektivitu celého IT.



Obrazek X+2: Disciplined Agile 2.0

Disciplined Agile 2.0 rozděluje činnosti a procesy IT oddělení do tří kategorií. Disciplined Agile Delivery vychází z předchozí verze frameworku. Na ní navazuje Disciplined DevOps, který si klade za cíl urychlit aktivity kolem vývoje IT řešení a IT operací. Nad oběma pak stojí Disciplined Agile IT, které se zabývá celým ekosystémem IT.

Klíčové aspekty původního Disciplined Agile Delivery jsou:

- Lidé na prvním místě
 - Jsou základním klíčem úspěchu projektu. Kromě jejich schopností je věnována pozornost také jejich spolupráci
- Orientace na učení
 - Podpora blízké spolupráce a vzájemného vzdělávání.
- Hybridní
 - DAD si vybírá a adoptuje prvky prověřených metod jako Scrum, Extrémní programování, Agilní modelování, Kanban aj.
- Celý životní cyklus
 - Zaměření na celý projekt od jeho úplného počátku přes produkci až po aktivity po dodání.
- Řízený procesními cíly
 - Řízení na základě nejdůležitějších faktorů stakeholderů, ať jsou jakékoliv.
- Zaměřený na řešení
 - Zaměření pozornosti na poskytování použitelných řešení než pouhé produkce softwaru.
- Stále porovnávání rizik a přínosů
 - Celý životní cyklus závisí na rizicích a přidaných hodnotách, čímž rozšiřuje Scrum zaměřený na přidanou hodnotu.

https://cs.wikipedia.org/wiki/ Disciplined_Agile_Delivery

Neexistuje mnoho studií zabývajících se implementací Disciplined Agile Delivery nebo Disciplined Agile 2.0 v reálných společnostech. Oficiální stránky Disciplined Agile Consortium uvádějí jedinou studii na severoamerické řetězci pekáren a kaváren Panera Bread, ale její plné znění už není dostupné.

Závěr

Stále více firem uvažuje o přechod na agilní metodiky i ve větších podnicích kvůli vyšší produktivitě a lepším produktům, které se stále mohou v průběhu procesu změnit. Všechny tři pokročilé škálovací frameworky jsou v praxi využívány. Podle průzkumu [VERISONONE] je nejpopulárnější Scaled Agile Framework (SAFe) s 27 % respondentům, kteří ho využívají. Zbylé 2 pokročilé frameworky jsou výrazně méně používané. Large-Scale Scrum (LeSS) využilo 6 % respondentům a Disciplined Agile Delivery (DAD) pouze 4 % respondentů. Stojí za zmínku, že v daném průzkumu byl suverénně nejpoužívanější Scrum/Scrum of scrums se 72 % respondentům. V průzkumu mohli respondenti vybrat více metodik a průzkum byl zaměřen, jak na malé tak větší společnosti, což může být jeden z důvodů popularity Scrumu.

Z 3 pokročilých frameworků je nejpodrobněji zpracován SAFe. Pokrývá všechny úrovně řízení a snaží se upravit celou strukturu společnosti a začlenit tam všechny úrovně řízení. LeSS je méně detailnější a tudíž volnější, což umožňuje větší přizpůsobivost frameworku organizaci. To může být pro zavedení velmi důležité především u společností, kde změna přichází od zaměstnanců a nikoliv od vedení. SAFe bez začlenění vedení přichází o velkou část své účinnosti. DAD respektive Disciplined Agile 2.0 je přes tvrzení autorů o vydání v srpnu 2015 stále nekompletní, což může implementátory větších firem odradit.

Z toho vyplývá, že momentálně je pro větší podniky nejlepší SAFe. Jeho popularita zároveň znamená větší množství expertů pro trénink a implementaci, díky čemuž může nadále rychle růst. Jedinou nevýhodou je nutnost změna organizační struktury a procesů v rámci celé organizace od vedení až po vývojové týmy. Takto rozsáhlé změně může být blokována především pokud je jí zavedení nepodporuje nejvyšší vedení a není ochoten udělat změny. LeSS je zajímavý framework především v momentě, kdy je jasné, že změna a začlenění vedení do agilního frameworku je nereálné. LeSS je možné zavést od vývojových týmů a postupně jej rozšířit výše do podnikové struktury bez ztrát výhod jako při podobném postupu při zavádění SAFe.

Literatura a zdroje

SCRUM ALLIANCE, I. SCRUM ALLIANCE. Advice on Conducting the Scrum of Scrums Meeting [online]. 2007 [cit. 2016-05-2]. Dostupné z: <https://www.scrumalliance.org/community/articles/2007/may/advice-on-conducting-thescrum-of-scrums-meeting>

MINTÁLOVÁ Katarína, Agilná metodika SCRUM a distribuované tímy [diplomová práca], 2013 [online] dostupné z: http://www.crzp.sk/crzpopacxe?fn=*recview&uid=1266394&pageId=resultform&full=0&focusName=bsktchRZ1#

ALLIANCE, A. Agile Alliance. Scrum Of Scrums [online]. 2013 [cit. 2016-05-2]. Dostupné z: <http://guide.agilealliance.org/guide/scrumofscrums.html>

BERTA, I. 2006. Agilné metódy vývoja softvéru: Škálovanie agilných metód. Január 2006. Dostupné na internete: <http://www2.fiit.stuba.sk/~bielik/courses/msislov/kniha/2006/groupd/essay/berta.pdf> [2016-05-2].

HOA, Thai Hai a Tomáš ZRCEK. Hodnocení DaD dle METES. 2015 [online] Praha. Semestrální práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních technologií. Dostupné z: <http://spicenter.vse.cz/wp-content/uploads/2016/01/DADvMETES.pdf>

SVĚTLÍK, Martin a Ondřej KAUSHIK. Škálování SCRUM. 2015 [online] Praha. Semestrální práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních technologií. Dostupné z: http://spicenter.vse.cz/wordpress/wp-content/uploads/2015/02/xkauo00_xsvem00_skalovani_scrum.pdf

STŘÍŽ, Oliver, HAUBERT Marek a Stanislav Nováček. LeSS Framework. 2015 [online] Praha. Semestrální práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních technologií. Dostupné z: http://spicenter.vse.cz/wp-content/uploads/2016/01/LESS_semestrální_práce.pdf

HOCH, Tomáš a Adam HOLUB. Metodika Disciplined Agile Delivery. 2015 [online] Praha. Semestrální práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních technologií. Dostupné z: http://spicenter.vse.cz/wordpress/wp-content/uploads/2015/02/4IT421_DisciplinedAgileDelivery_xhocht02_final.pdf

LUČAN, Martin a Petr VELÍŠEK. Scaled Agile Framework® (SAFe®) 4.0. 2016 [online] Praha. Semestrální práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních technologií. Dostupné z: <http://spicenter.vse.cz/wp-content/uploads/2016/01/SAFe.pdf>

SKŮRA, Tomáš a Martin HAVER. Downscaling SAFe, 2015 [online] Praha. Semestrální práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních technologií. Dostupné z: http://spicenter.vse.cz/wp-content/uploads/2016/01/Downscaling_SAFe.pdf

ZIKMUND, Jan. Škálování agilních metodik. 2012 [online] Praha. Bakalářská práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních technologií. Dostupné z: https://www.vse.cz/vskp/33214_skalovani_agilnich_metodik

BUCHALCEVOVÁ, Alena. Metodiky Budování informačních systémů. Praha: Oeconomica. 2009. [Citace: 2016-05-2] ISBN: 978-80-245-1540-3

Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [2016-05-2]. Dostupné z: [https://en.wikipedia.org/wiki/LargeScale_Scrum_\(LeSS\)](https://en.wikipedia.org/wiki/LargeScale_Scrum_(LeSS))

BASS, J.M., Allison, I.K. & Banerjee, U. 2013, "Agile Method Tailoring in a CMMI Level 5 Organization: Addressing the Paradox", Journal of International Technology and Information Management, vol. 22, no. 4, pp. 77.

Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation: XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers, 2014, Springer International Publishing, Cham.

VERSIONONE, The 10th annual state of afile report: 2016- [2016-05-2]. [online] dostupné z: VERSIONONE.COM

BROOKS, Frederick. The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley, 1975. ISBN 0-201-00650-2.

The above authors this declaration may be freely copied in any form, but only in its entirety through this notice. © [2001]- [2016-05-2]. Dostupné z: <http://agilemanifesto.org>

Agile Software Development. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-2]. Dostupné z: https://en.wikipedia.org/wiki/Agile_software_development 2. Large-Scale Scrum (LeSS)

FARIA, Leonardo, Scrums of Scrums:Runnning Agile on Large Projects, 2013- [2016-05-2]. Dostuúné z: <https://www.scrumalliance.org/community/articles/2013/june/scrum-of-scrums-running-agile-on-large-projects>

Disciplined Agile Consortium, 2014-2016 - [2016-05-12]. Dostupné z: <https://www.disciplinedagileconsortium.org/http/scottambler.com/assets/panera-case-study.pdf>

Disciplined Agile 2.0, 2016- [2016-05-12]. Dostupné z: <http://www.disciplinedagiledelivery.com/agility-at-scale/disciplined-agile-2>

LEFFINGWELL, Dean, Welcome to Scaled Agile Framework C4.0! 2016- [2016-05-12].
Dostupné z: <http://www.scaledagileframework.com/about/>

PAŇKOVÁ, Jana, Mansurova, Dinara Hodnocení Less dle METES ZS 2015 [2016-05-12]
Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních
technologií. Dostupné z: <http://spicenter.vse.cz/wp-content/uploads/2016/01/LessvMETES.pdf>

LEFFINGWELL, Dean, Welcome to Scaled Agile Framework C4.0! 2016- [2016-05-12].
Dostupné z: <http://www.scaledagileframework.com/safe-lean-agile-principles/>

Obrázky

FARIA, Leonardo, Scrums of Scrums:Running Agile on Large Projects, 2013- [2016-05-2].
Dostupné z: <https://less.works/less/framework/index.html>

Disciplined Agile 2.0, 2016- [2016-05-12]. Dostupné z:
<http://www.disciplinedagiledelivery.com/agility-at-scale/strategic-agility-at-scale/>