

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	Zimní; Akademický rok 2016/2017
Autoři	Michal Šimonovský – xsimm82 Jiří Hudec – xhudj19 Marek Zima – xzimm04
Téma	Using SEMAT and Essence at Fujitsu UK
Datum odevzdání	18.12.2016

Abstrakt

Práce obsahuje shrnutí metodiky metodik Essence od iniciativy SEMAT (Software Engineering Method and Theory). Hlavní náplní je popis způsobu implementace Essence ve Fujitsu UK, především z organizačního pohledu. Cílem je ukázat východiska, ze kterých se při implementaci vycházelo, jaké problémy byly řešeny, jaké kroky bylo nutné podniknout, aby byla implementace úspěšná a souhrn rad pro další organizace. Práce dále představuje dvě případové studie, kde jsou ukázány konkrétní výsledky implementace částí Essence do procesu budování informačních systémů.

Klíčová slova

Essence, SEMAT, Fujitsu, kernel, jádro, alfa

OBSAH

1	Úvod.....	4
1.1	Cíl práce.....	4
2	SEMAT	4
3	Essence	5
3.1	Architektura Essence	5
3.1.1	Jádro	6
3.1.2	Jádro a iterace	8
3.1.3	Jádro a škálovatelnost vývoje.....	9
4	Essence ve Fujitsu UK	9
4.1	Před implementací Essence	10
4.2	Měnicí se svět	11
4.3	Užívání úhlů pohledu.....	12
4.4	Benefity úhlů pohledu	13
4.5	Rady pro implementaci Essence.....	13
4.6	Benefity Essence pro Fujitsu	13
5	Essence v ostatních společnostech	14
5.1	Munich Re	14
5.2	Globální výrobce komunikačních technologií.....	15
6	Závěr.....	17
7	Literatura	18

1 ÚVOD

SEMAT je relativně mladá iniciativa. I přes její mládí se jí už podařilo rozšířit jejich metodiku metodik Essence i do velkých firem. V tomto případě jde o firmu Fujitsu, konkrétně její pobočku ve Spojeném království. Díky tomu, že se ve Fujitsu rozhodli podělit o zkušenosti s implementováním Essence v jejich firmě s veřejností, tak mohla vzniknout tato práce. Která tedy na začátku krátce popíše Essence, nicméně její hlavní náplní je výtah z publikovaného článku o Essence ve Fujitsu UK. Aby práce nebyla založena pouze na jednom zdroji, tak byly rešeršně zpracovány další dvě případové studie.

1.1 CÍL PRÁCE

Cílem práce je seznámit čtenáře se stručným popisem metodiky SEMAT & Essence a podrobně popsat, jak tuto metodiku implementovali ve společnosti Fujitsu UK a dalších.

2 SEMAT

Software Engineering Method and Theory je iniciativa pro zlepšování kvality softwarového inženýrství, která vznikla v roce 2009 a na jejímž vývoji se podíleli Ivar Jacobson, Bertrand Meyer a Richard Soley. SEMAT je zde pro to, aby z něj měly prospěch veškeré zainteresované strany, především jde o rozvoj a podporu mezinárodních standardů softwarového inženýrství.

SEMAT vznikl v důsledku nedostatečné kvality výuky softwarového inženýrství. Lidé, kteří studují softwarové inženýrství nejsou po ukončení školního procesu přímo vhodní na to být okamžitě zaměstnáni. Současné univerzity si kladou za cíl poskytnout co nejširší spektrum vzdělání napříč všemi obory, a to dostatečně kvalitně. Nicméně poskytnout kvalitní vzdělání není pro univerzity jednoduchý úkol. Za tímto účelem vznikl SEMAT a také Essence jako nástroj, který nabízí plné pokrytí metod softwarového inženýrství přístupnější cestou pro studenty (SEMAT Inc. and SEMAT Office, 2009-).

SEMAT je rozšířený po celém světě, především v Číně, Jižní Americe, Jižní Africe, Rusku, Japonsku a Koreji. Například japonská pobočka vznikla v roce 2013 a nyní má více než 250 členů. Několik organizací, jako univerzita Waseda University se rozhodly zahrnout SEMAT ve svých vzdělávacích kurzech (Wikipedia, 2016).

3 ESSENCE

Současné metody jsou založené na postupu, jak vytvořit činnost, jež má za cíl dosažení výsledku podle přesné definice. Takové postupy vedou k nepřehlednosti a softwarová inženýři se následně přiklání k metodě pokus omyl.

Vytvořit komplexní software vyžaduje velkou schopnost spolupráce a komunikace mezi jednotlivými členy vývojového týmu. Komunikace tkví nejen ve schopnosti vedoucího týmu delegovat úkoly a mluvit s členy, ale také mezi zadavatelem softwaru a jeho dodavatelem. Valná část problémů vzniká z nedostatku pochopení právě mezi těmito dvěma stranami. Problémy mohou také vzniknout v různých fázích a dimenzích projektu.

Jádro Essence poskytuje návod, jak efektivně pracovat s požadavky zákazníka, vyvstávajícími příležitostmi, mezi jednotlivými zainteresovanými stranami, rozdělování práce, způsoby, jak vykonávat danou činnost a samozřejmě především úspěšnou dodávku software. Poskytuje tak jednoznačný základ pro softwarové projekty různého rozsahu. Doporučení a koncepty v Jádro jsou především intuitivní, aby jim porozuměl každý. Toto Jádro je v současné době stále ve vývoji a v budoucnu lze očekávat další rozšíření (Kadlec, a další, 2014).

3.1 ARCHITEKTURA ESSENCE

Architektura již byla podrobně zpracována v předchozí semestrální práci¹, proto je tato kapitola pouze stručným přehledem a pro lepší pochopení si přečtěte odkazovanou práci.

Architektura Essence se dá rozdělit na čtyři části (Kadlec, a další, 2014):

- **Metody** – Jsou popis postupů a návodů práce, podle kterých může tým dosáhnout svých cílů. Metoda je především popis, jak bude tým postupovat a také poskytuje pomoc při řešení dané úlohy.
- **Praktiky** – „Opakovaná snaha s konkrétním cílem. Praxe poskytuje systematický a ověřený postup, jak řešit jednotlivé aspekty úlohy a může být součástí několika metod“ (Kadlec, a další, 2014).
- **Jazyk Essence** – „Doménový specifický jazyk, pomocí kterého definujeme metody, postupy a Jádro pro konkrétní projekt“ (Kadlec, a další, 2014).

¹ <http://spicenter.vse.cz/wordpress/wp-content/uploads/2015/02/Essence.pdf>

Čtvrtá část Jádru je kvůli své důležitosti podrobněji popsána v následující kapitole.

3.1.1 Jádru

Jádru poskytuje jednoduché a intuitivní postupy, jak překonávat problémy v týmu a efektivně řešit problémy během vývoje softwaru. Jádru se však často dynamicky mění i se slovníkem, který je také velmi často upravován.

Organizace Jádru se dělí na jednotlivé části, kde se každá z těchto částí zabývá rozdílnou dimenzí softwarového inženýrství. Tyto tři části jsou především Zákazník, Úsilí a Řešení. Řešení je výsledek vývoje a vynaloženého úsilí během něj. Vývoj zahrnuje alespoň jednoho zákazníka, kterému má být software doručen (Kadlec, a další, 2014).

Mezi základní elementy Jádru patří místa aktivit, kompetence a alfy.

Místa aktivit

„Vyjadřují určitou aktivitu, která je potřebná a musí být provedena pro úspěšný vývoj softwaru“ (Kadlec, a další, 2014).

Jádru nedefinuje přesně, jaké aktivity jsou potřeba, ale namísto toho definuje ona místa aktivit, jakožto metodologicky nezávislý symbol (Jacobson, 2013).

Kompetence

Pro úspěšné řízení softwarového vývoje je potřeba nejen technických znalostí k vykonávání vlastní práce, ale také mít schopnost porozumět práci ostatních členů týmu. Největším nedostatkem je, když člověk neví o vlastních mezerách ve znalostech. Jádru pomáhá udělat tyto nedostatky více zřejmé, aby se mohly včas vyřešit a nezpůsobily vývoji softwaru dlouhodobější problémy (Kadlec, a další, 2014).

Alfy

Jde o sedm kritických Alf (oblastí), na které je třeba se soustředit pro zaručení úspěchu projektu. Každá z těchto Alf má několik „Alfa stavů“, které nastanou pouze po splnění podmínek pro tento stav.

Například jedna z Alf jsou Požadavky. Ty mají definováno 6 Alfa stavů. Každý z těchto Alfa stavů má svůj jasně definovaný odškrtačací seznam podmínek, které musí být splněny pro jeho dosažení. Takže pro dosažení Alfa stavu Koherence, musí být splněny podmínky jako „důležité scénáře užití musí být vysvětleny“ viz Obrázek 1.

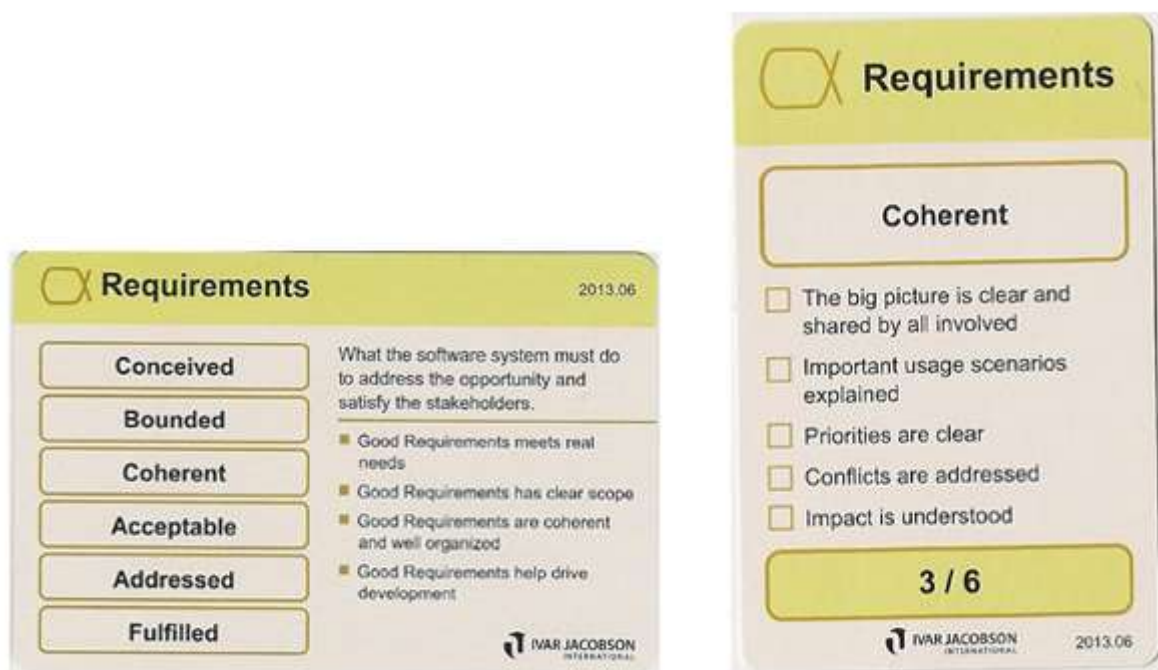
Použitím Alf lze dosáhnout toho, že budou splněny všechny nezbytné procesy a metody. Protože lze v jakémkoli stavu projektu definovat jakých Alfa stavů by se mělo dosáhnout (Ste, 2016).

Dále mezi Alfy patří například příležitosti, Akcionáři (Stakeholder) a další. Tyto alfy existují bez ohledu, zda je definujeme v dokumentaci či ne. Jsou to prvky, které často ovlivňují chod projektu.

Pochopit, jak prakticky využívat jádro je to nejdůležitější, avšak pro lidi, kteří s ním doposud nepřišli ve styk to nemusí být úplně jednoduché. Existuje více způsobů, jak pochopit Jádro, tím nejjednodušším je pomocí karet. Karty vývojářům zjednodušují pochopit jádro a naučit se ho používat. Tyto karty se dělí na karty definice alf a stavové karty.

Velikost karet definice alf je zhruba 5x3 palce a zevrubně popisují o čem daná alfa je. Na druhou stranu velikost stavových karet se pohybuje okolo velikosti vizitek a říkají, v jakém stavu se daná alfa může nacházet (Kadlec, a další, 2014).

Příkladem těchto karet je Obrázek 1.



Obrázek 1: Alfa Požadavky, její stavy a podmínky dosažení stavu Koherence (Zdroj: (Ste, 2016))

3.1.2 Jádru a iterace

Určitý čas pro vývoj otestovaného přírůstku softwaru se nazývá iterace. Rozsah softwarového projektu se zvětšuje s rozsahem jednotlivých iterací a času na nich strávených. Iterace většinou trvá okolo 3 týdnů.

Kus práce, který lze jednoznačně popsat se nazývá úloha. Každá iterace v sobě zahrnuje jednu a více úloh. Cílem každé iterace je další přírůstek stabilního software. Tyto cíle jsou sepsány v backlogu, který slouží jako seznam cílů, jež je potřeba dokončit v dané iteraci (Kadlec, a další, 2014).

Cyklus iterace

Essence popisuje 4 části iterace, konkrétně Plan (plánuj), Do (proved'), Check (zkontroluj), Adapt (přizpůsob). Při plánování se zjistí současný stav iterace, cíle představují nějaký stav v budoucnu, kterého je potřeba dosáhnout. Veškeré aktivity potřebné k dosažení onoho stavu se zařadí v části Do, při kontrole se zjišťuje, zda očekávaný budoucí stav koresponduje s danými požadavky, a nakonec v poslední fázi adaptace se zkusí najít lepší způsoby práce a zlepšení její kvality (Kadlec, a další, 2014).

Plánování iterace

Pilířem plánování iterace jsou tři otázky, které je nutné si zodpovědět.

- Kde jsme?
- Kam směřujeme?
- Jak se tam dostaneme?

První otázka nám zodpoví současný stav doposud vyvinutého úsilí. Co dál nám řeknou cíle pro další iteraci a třetí dotaz vyřeší problém, jak se dostat ke splnění vytyčených cílů a stanovení úloh (Kadlec, a další, 2014).

Provádění iterace

Výše uvedený cyklus se opakuje během každé iterace. Na tabuli se často kreslí sloupce jako Todo, Doing a Done. V každém sloupci si členové týmu jednotlivé úkoly rozdělí a mohou na nich následně pracovat. V rámci iterace se úkol pohybuje ze sloupců Todo, do Doing a nakonec, když je hotový, tak skončí v Done (Jacobson, 2013).

3.1.3 Jádru a škálovatelnost vývoje

Mezi důležité aspekty vývoje softwaru se řadí pružnost a schopnosti přizpůsobit se změnám, které mohou nastat. Díky této škálovatelnosti je možné upravit systém tak, aby vyhovoval specifickým podmínkám.

V Essence se berou v úvahu tři dimenze škálovatelnosti (Kadlec, a další, 2014):

- První dimenze se nazývá přiblížení, jedná se o poskytování rad na rámec jádra. Jedná se o praktiky, které jsou rozšířením Jádra. Popis těchto praktik umožňuje snadnější pochopení pro členy týmu. Essence poskytuje způsob, jak tyto praktiky srozumitelně popsat a sestavit.
- Druhá dimenze je o rozšíření Jádra metodami pro unikátní požadavky, které je potřeba splnit u různých druhů vývoje. Týmy si vytvoří vlastní metody, které jsou nad rámec toho, co popisuje Jádro. Jedná se o souhrn postupů a pokynů, jak by měli členové týmu při vývoji postupovat.
- Poslední dimenzí je škálovatelnost, kde je nutné se zabývat komplexním systémem s velkým množstvím požadavků.

4 ESSENCE VE FUJITSU UK

Fujitsu stejně jako ostatní velké společnosti používalo velké množství metodik a postupů, které nashromáždilo za léta existence. Fujitsu prvotně chtělo pouze revidovat celou oblast metodik a postupů používaných ve vývoji aplikací. Při implementaci těchto „historických“ metodik a postupů nevadilo, že nebyly nijak provázané. Nicméně v průběhu revize současného stavu ve Fujitsu pochopili, že tento stav není možné dále udržet. A pro zvýšení úspěšnosti projektů je třeba metodiky a postupy integrovat. Protože pokud používáte na projektu několik metodik a postupů odděleně, tak je téměř jisté, že se zapomene na nějaký důležitý problém, a to povede k nákladným důsledkům (tj. zpoždění projektu, nesplněné požadavky na aplikaci atd.). Ve Fujitsu najednou potřebovali nějakou metodiku, která by tyto odlišné metodiky a postupy dokázala bezproblémově provázat. Ve stejnou dobu, kdy ve Fujitsu dospěly k tomuto výsledku se začaly výrazněji využívat agilní přístupy k vývoji aplikací. Takže k prvotní potřebě provázat původní metodiky a postupy se přidala potřeba skloubit klasický (tj. vodopádový) a agilní přístup. Bez splnění těchto dvou podmínek by ve Fujitsu nemohli spolehlivě sledovat důležité parametry současných a budoucích projektů (tj. zdraví, jestli se nezpožďují atd.).

Ačkoliv se Essence primárně zaměřuje na softwarové inženýrství, jak bylo popsáno v kapitole 3, tak ve Fujitsu zjistili, že se dá stejně dobře použít i k systémovému inženýrství.

Principy Essence ve Fujitsu dokázaly využít pro integraci postupů pro vývoj aplikací a servisních služeb. A to pouze díky nepatrnému upravení „Alfa softwarového systému“ na „Alfa systém“. Díky Essence ve Fujitsu získaly metodiku, která jim umožnila dívat se na pracovní procesy ve všech oblastech firmy a zároveň jim rozumět. Dále jim umožnila jasně a stručně komunikovat mezi jednotlivými pracovními procesy jejich problémy a existujícími mezerami.

Essence je ve Fujitsu využíván i ke komunikaci a koordinaci mezi národními pobočkami. Protože každá národní pobočka může mít specifické procesy a není možné vytvořit jeden super proces, který je bude uplatňovat všude. Proto je využit Essence, který abstrahuje od konkrétních činností a umožňuje mluvit nad rozdílnými procesy stejným jazykem (Ste, 2016).

4.1 PŘED IMPLEMENTACÍ ESSENCE

První, co ve Fujitsu udělali poté co se rozhodli implementovat Essence bylo uspořádání několika workshopů s klíčovými jedinci.

Tento přístup je kritický pro přesvědčení klíčových jedinců o důležitosti implementace. Ve Fujitsu zvolili zástupce z businessu, projektového managementu, prodeje, architektury a designu, správy a zabezpečení a testingu a servisních služeb. Pokud by nevybrali některé z výše zmíněných oblastí, tak to mohlo vést k nechuti přijmout tuto novou metodiku a k přesvědčení, že jde o další technickou zbytečnost.

Jednou z hlavních náplní workshopů bylo použití Essence na imaginární projekt z reálného světa. Role v projektu byly náhodně přiděleny účastníkům workshopu (tj. zadavatel, týmový manažer, tester a další). V imaginárním projektu byly identifikovány klíčové oblasti a tyto oblasti byly odehrány za použití Essence principů a Alfa k definování opravdového stavu projektu.

Jednou z odehraných oblastí byly Požadavky. Jedinci zastupující zadavatele měli za úkol nadefinovat co by jim měl projekt přinést. A ostatní jedinci zjišťovali, co pro ně znamenají výstupy zadavatele. Poté byly obě skupiny spojeny dohromady, aby všichni pochopily, kolik energie a práce bylo potřeba pro oba výstupy. K uspořádání porozumění Alfa stavů došlo díky dotazování všech účastníků, jestli jim přišly zástupci zadavatele zapojení a do jaké míry.

Výstupem z těchto workshopů bylo identifikování událostí z reálného světa a zjištění, jak by byly řešeny bez použití a za použití Essence, což vedlo k vášnivým diskuzím.

V průběhu workshopů bylo využito karet (popsaných v kapitole o Jádru Essence). Karty jednotlivých Alf a jejich stavů byly připnuty na zeď a ve chvíli, kdy bylo rozhodnuto, že jsou splněny podmínky pro dosažení stavu, tak musel konkrétní jedinec vstát a posunout kartu na zdi do polohy dosaženo. Tento primitivní přístup byl důležitý, protože taktilní a pohybová stránka změny stavu umožnila utvrdit co se skrývá pod jednotlivými stavy v myslích účastníků.

Využití workshopů pro pochopení principů a výhod Essence u klíčových jedinců má mnohem větší účinek než pouhé schůzky s přednesem teorie. Protože účastníci musí obětovat vlastní čas, proto se snaží opravdu pochopit principy a výhody. Jinak by totiž zbytečně ztráceli čas (Ste, 2016).

4.2 MĚNÍCI SE SVĚT

Svět IT projektů se za posledních pár let výrazně proměnil. Z toho důvodu už vždy nelze použít klasických vodopádových přístupů. To je zapříčiněno i nástupem cloudových služeb ve variantách IaaS, PaaS a SaaS.

Díky těmto změnám není možné použít jeden jediný přístup na všechny druhy projektů. Díky obecnějšímu přístupu Essence, který se soustřeďuje na obecné oblasti (Alfy) a jejich stavy (Alfa stavy) je možno posoudit jakýkoliv projekt bez ohledu na konkrétní použitou metodiku. To umožňuje managementu posoudit stavy projektů mnohem rychleji a rozpoznat potenciální problémy mnohem dříve.

Například lze díky Alfám zjistit nesoulad mezi jednotlivými akcionáři (zadavateli). A management na tento problém může zareagovat a snažit se ho řešit. A ať už ho vyřeší úspěšně a podaří se mu pohled zadavatelů sjednotit, nebo neúspěšně a akcionáři (zadavatelé) ztratí zájem o projekt. Každopádně má management možnost řešit problémy mnohem dříve než bez využití Essence.

Díky obecně formulovaným Alfám je použit společný jazyk ke komunikaci a tím pádem odpadá potřeba vše neustále vysvětlovat. A je možné se soustředit rovnou na existující problém (Ste, 2016).

4.3 UŽÍVÁNÍ ÚHLŮ POHLEDU

Ve Fujitsu je zároveň využíván přístup úhlů pohledu. To znamená, že se vezme daný úhel pohledu a tím se díváme na zkoumanou oblast a soustředíme se na klíčové aspekty oblasti. Úhly pohledu se dají použít jednak na celý segment businessu (např. výroba), ale i na projekty a menší celky.

Úhly pohledu se dají použít i v kombinaci s Essence, jejich spojení umožňuje dívat se danou oblast a plně pochopit její jednotlivé stavy.

Ve Fujitsu používají tyto úhly pohledu:

- Podnikový úhel pohledu
- Programový úhel pohledu
- Projektový úhel pohledu

Podnikový úhel pohledu

Je zaměřen z pohledu podnikové architektury zahrnující obecnou vizi a cíle podniku. To je ta nejvyšší vrstva podniku obsahující dlouhodobé business strategie. Tyto strategie jsou od toho, aby ukazovaly odkud společnost jde a kam míří. Takže tento úhel pohledu v kombinaci s Essence umožňuje managementu reálně zhodnotit, jak moc se tyto strategie vyplňují.

Tento úhel pohledu se úzce zaměřuje na Alfa Akcionáři, a to z důvodu, že plná podpora akcionářů a investorů je nejdůležitější pro podporu plnění strategických cílů.

Zároveň jsou zde využity i Alfa Tým a Způsob práce. A to pro úspěšnou akceptaci změny firemní kultury, bez které jsou i reálně úspěšně ukončené projekty odsouzeny k neúspěchu.

Programový úhel pohledu

Tento úhel pohledu se soustředí na střednědobé plány, které se skládají ze souboru nezávislých projektů, které dohromady naplňují jeden ze strategických cílů. Alfa Akcionáři není v tomto úhlu pohledu až tak důležitá, ale stále se tu vyskytuje. Alfa Tým se zaměřuje na toho, kdo dodává jednotlivé projekty a Alfa Způsob práce zajišťuje, koordinuje průběh všechny nezávislých projektů, tak aby naplnil strategický cíl.

Projektový úhel pohledu

Tento úhel pohledu aplikuje Essence na jeden konkrétní projekt. Alfa Způsob práce je zde mnohdy chápána jinak než v programovém úhlu pohledu. A to jako kontakt konkrétních

pracovníků na denní bázi. Díky této úpravě jsou pracovníci pány vlastního osudu a můžou fungovat pod agilními postupy (Ste, 2016).

4.4 BENEFITY ÚHLŮ POHLEDU

Umožňují soustředit každou z Alf tým správným směrem se správnou úrovní zaměření. Současně umožňují pozorovat různé úrovně dodání projektů, aniž by docházelo ke zmatení. Například dodání cílů střednědobých plánů může trvat rok a více, ale jednotlivé projekty z plánu mohou mít mnohem kratší dobu dodání.

Lze si to představit na projektu stavba domu (programový úhel pohledu). Ten sestává z mnoha menších projektů např. postavení zdí, střechy a dalších (projektový úhel pohledu). A ačkoliv by všechny dílčí projekty dopadly dobře a dům stojí, tak na konci zjistíte, že jsou špatné základy a všechno bylo k ničemu.

Kombinace výše zmíněných úhlů zajišťuje úspěšné dodávání projektů i střednědobých plánů. Protože by se na problém se základy přišlo mnohem dříve díky programovému úhlu pohledu (Ste, 2016).

4.5 RADY PRO IMPLEMENTACI ESSENCE

Fujitsu na základě zkušeností s implementací Essence doporučuje začít pomalu a poctivě. Prvně vytrénovat menší skupinu zaměstnanců, tak aby to opravdu pochopila a uměla použít. Tato skupinka poté může předávat tyto znalosti dál (Ste, 2016).

4.6 BENEFITY ESSENCE PRO FUJITSU

Důvody pro implementaci Essence byly zmíněny výše. Stručně to je poskytnout společný jazyk pro debatu nad projekty, které můžou být vedeny jinak. A zároveň umožnit komplexní pohled na projekty a odhalit mezery, které by v konečném důsledku zapříčinily neúspěch projektů.

Tyto cíle byly naplněny a zároveň ve Fujitsu zjistili, že nemusí tvořit jeden velký monolitický super proces. Essence jim umožňuje zachovat původní procesy a metodiky. Akorát se musí jednotlivé postupy a metodiky přizpůsobit Essence a tím pádem dojde k provázání a umožnění úspěšných projektů (Ste, 2016).

5 ESSENCE V OSTATNÍCH SPOLEČNOSTECH

Na internetu je možné najít případové studie ilustrujících využití Essence pro sjednocení a restrukturalizaci procesů v mnohých společnostech po celém světě. Z dostupných prací vybíráme dvě, které velice dobře ukazují flexibilitu a univerzálnost principů Essence a jejich aplikování v různých odvětvích.

5.1 MUNICH RE

První společností, která vylepšila svoje procesy v oblasti softwarových projektů, je německá pojišťovací společnost Munich Re (McDonough, 2014). Hlavní oblastí podnikání je zajišťování (zjednodušeně se jedná o poskytování pojišťovacích služeb pro ostatní pojišťovny, které následně tyto služby poskytují koncovým zákazníkům), společnost má sídlo v Mnichově a celosvětově zaměstnává zhruba 43 tisíc zaměstnanců.

Během posledních deseti let se Německo potýkalo s nedostatkem kvalifikovaných vývojářů a testerů, vedení IT společnosti Munich Re se proto v roce 2008 rozhodlo transformovat společnost, aby bylo možné velké množství práce outsourcovat. Po dvou letech takové spolupráce s dodavateli ovšem jak zaměstnanci, tak obchodní partneři cítili, že původní pracovní model nevede k efektivním výsledkům a bude nutné přesněji definovat způsob vývoje aplikací.

Cílem bylo definovat kostru pro typický projekt, včetně jednoznačných metrik, které měří jeho průběh (a tedy jeho zdraví). Při pokusu o tuto definici se narazilo na problém, kdy neexistoval jednotný jazyk, kterým by bylo možné projekt definovat, protože každá ze stran používala buď jiné termíny, nebo dokonce stejné termíny s jiným významem.

V tento moment se vedení Munich Re obrátilo pro pomoc na externí konzultanty a ti doporučili použít jazyk Essence pro popis všech procesů a metodik tak, aby jim stejně rozuměli zaměstnanci pojišťovny i jejich dodavatelé. Dále za použití alf definovaných v Essence připravit požadované metriky pro hodnocení průběhu projektů. Jak zhodnotil jeden z architektů, dříve on a jeho kolegové byli schopni vycítit, že se projekt začíná bortit, ale až po zavedení praktik Essence byli schopni to slovně popsat a komunikovat s dodavateli.

Výsledkem implementace Essence v Munich Re bylo roztřídění projektů do tří skupin a vytvoření typických životních cyklů pro jednotlivé skupiny:

- Průzkumný projekt: typicky vývoj nového produktu, pro který není dopředu známá konkrétní architektura ani přesný rozsah požadavků. Jedná se o projekty, které mají vysoké riziko výskytu komplikací.
- Standardní projekt: typicky se jedná o větší zásah do stávajícího produktu. Architektura i požadavky jsou známé, hlavním rizikem tedy zůstává jen rozsah projektu a jeho dopady.
- Malé vylepšení: typicky údržba systémů nebo úpravy malého rozsahu, které spolu nenesou velká rizika dopadu na společnost.

Konkrétní praktiky pro vývoj software si společnost Munich Re dále vyvíjí sama, všechny ovšem musí být definovány pomocí jazyka Essence a být použitelné pro každý z výše zmíněných scénářů. Tato jednotnost pomáhá s udržením konzistence mezi jednotlivými procesy, usnadňuje předávání znalostí a zkušeností a v neposlední řadě ulehčuje každodenní práci pracovníkům.

5.2 GLOBÁLNÍ VÝROBCE KOMUNIKAČNÍCH TECHNOLOGIÍ

Druhou společností, která využila Essence pro transformaci svých procesů pro agilní způsob práce, byl blíže nespecifikovaný globální poskytovatel informačních a komunikačních technologií (Ivar Jacobson International, 2016). IT divize této firmy se skládala celkem z 3000 zaměstnanců a 700 externích pracovníků, týmy byly rozděleny podle náročnosti vyvíjených aplikací po 5 až 200 lidech. Některé produkty byly navrženy speciálně pro danou firmu, jindy se jednalo o úpravu stávajícího řešení, či o mobilní nebo webové služby.

Na problémy se naráželo převážně při propojování jednotlivých systémů či týmů, dále se vyskytovaly nesrovnalosti při komunikaci s externími dodavateli a v neposlední řadě docházelo k velké fluktuaci pracovníků v týmech. Všechny tyto faktory nepříznivě ovlivňovaly potřebný čas a náklady a výslednou kvalitu produktů. Jako východisko byla zvolena agilní transformace.

První zavedení agilních přístupů proběhlo v roce 2008 v oddělení výzkumu a vývoje (R&D), další vlna týkající se ostatních oddělení přišla na řadu o 5 let později, v roce 2013. Nicméně zavádění neprobíhalo podle původních představ, jednotlivé týmy díky své diverzitě přijaly různé postupy a nebylo ani možné využít původní zkušenosti z roku 2008, protože týmům chyběl jednotný jazyk, kterým by mohly komunikovat a předávat si získané znalosti.

Aby se tyto problémy odstranily, oddělení prozkoumala mnoho různých agilních frameworků (např. Scrum, XP, SAFe, LeSS a další), zjistilo se však, že žádný z nich nesplňuje všechny dané požadavky. Nicméně mnoho z nich obsahovalo zajímavé dílčí postupy, které by bylo možné zaintegrovat do celkového řešení. To ovšem nebylo možné bez sjednocujícího jazyka, který by sloužil jako základ pro budovaný celek.

Jako hledaný základ byl vybrán Essence a jako jádro vznikajícího řešení posloužilo 7 alf Essence. Týmy zjistily, že alfy jim umožnily dohodnout se na sktruktuře základních pracovních postupů a že díky jazyku Essence je možné tyto postupy jednoznačně a srozumitelně popsat. Dalším přínosem sjednocení terminologie bylo odstranění duplicitních postupů, které předtím existovaly.

Nad jednotnou architekturou bylo dále možné budovat jednotlivé znovu použitelné metody. Pro jednoduchost byly alfy seskupeny do dvou dimenzí:

- Rozsah – Dělení projektů podle počtu vývojových týmů zahrnutých do vývoje a velikosti skupin ovlivněných uživatelů. Zahrnuje především alfy Akcionáři a Tým. Pro malé projekty např. není nutné provádět Mapování Akcionářů (Stakeholder Mapping), zatímco pro velké se jedná o klíčovou aktivitu.
- Integrace – Dělení projektů podle míry integrace s ostatními systémy společnosti, které odpovídá alfám Požadavky a Software Systém. Pro jednoduché projekty není např. nutné sestavovat integrační tým, zatímco složitý projekt se bez něj neobejde.

U každé dimenze jsou nadefinované dva stupně složitosti (nízký a vysoký) a postupy, jak pro projekty rozhodnout, který stupeň jim odpovídá. Kombinací možných hodnot v dimenzích vznikly čtyři typické pracovní postupy (2 dimenze * 2 stupně složitosti). Výsledné scénáře jsou podrobně zadokumentované v Knowledge Management systému, kde kromě jejich teoretického popisu jsou uvedené také příklady týmů a projektů, kde byla metoda úspěšně použita.

Hlavním přínosem této standardizace bylo podpoření adopce agilních postupů napříč celým IT oddělením. Jednotlivé týmy měly k dispozici shodné informace a zkušenosti, rozšiřování těchto postupů mezi jednotlivé pracovníky tak bylo mnohem rychlejší, po dvou letech rozvoje a tréninků agilní metody používalo zhruba 5000 pracovníků. Pozitivním výsledkem pro ostatní divize bylo zkrácení vývojových cyklů a zvýšení kvality poskytovaných služeb.

6 ZÁVĚR

Iniciativa SEMAT a její metodika metodik Essence je sice relativně mladá, ale i přesto a možná právě proto, bychom o ní měli vědět. Jména firem, které si ji zvolily mluví za vše. A pokud vezmeme v úvahu fakt, že se stále nijak výrazně nedaří zlepšovat statistiky úspěšnosti projektů v IT (a tím není myšleno pouze dodání produktu, ale celkové hodnocení projektu), tak je potřeba hledat nástroj, který by tento fakt změnil.

Z popisu Essence a jejího implementování ve Fujitsu je vidět, že Essence může být tím nástrojem. Což potvrzují i další dvě vybrané případové studie.

Odlišnosti zaměření businessu firem z vybraných případových studií a jejich rozdílné implementování Essence dokazuje její univerzalitu. Není důležité, jak jsou organizace velké, nebo jaký druh byznysu je pro ně klíčový. Společné zůstává, že klíčovým krokem v optimalizaci procesu vývoje software je sjednocení jazyka a používané terminologie. Bez tohoto kroku není možné efektivně porovnávat jednotlivé postupy či metodiky a provádět smysluplné úpravy a vylepšení. A to je Essence schopná poskytnout.

7 LITERATURA

Ivar Jacobson International. 2016. Practice Architecture on Top of Essence - A Key Enabler to Manufacturer's Agile Transformation. [Online] 5. 7 2016. [Citace: 10. 12 2016.] https://www.ivarjacobson.com/sites/default/files/field_iji_file/article/global_technology_manufacturer_essentialization.pdf.

Jacobson, Ivar, a další. 2013. *The Essence of software engineering: applying the SEMAT kernel.* místo neznámé : Addison-Wesley Professional, 2013. 978-0-321-88595-1.

Kadlec, P., a další. 2014. Essence. *SPI center.* [Online] 2014. [Citace: 29. 10 2016.] <http://spicenter.vse.cz/wordpress/wp-content/uploads/2015/02/Essence.pdf>.

McDonough, Ann. 2014. Munich Re and ESSENCE – Kernel and Language for Software Engineering Methods: A Case Study. *Object Management Group.* [Online] 01. 12 2014. [Citace: 10. 12 2016.] http://www.omg.org/news/whitepapers/Munich_Re_Essence_Case_Study-2014-12-01_JP.pdf.

SEMAT Inc. and SEMAT Office. 2009-. Why Should I Care as a Teacher and as a Student? - SEMAT. *semat.org.* [Online] SEMAT Inc. and SEMAT Office, 2009-. [Citace: 29. 10 2016.] <http://semat.org/why-should-i-care-as-a-teacher-and-as-a-student->.

Ste, Nadin. 2016. Using SEMAT and Essence at Fujitsu UK. *InfoQ: Software Development News, Videos & Books.* [Online] 2016. [Citace: 30. 10 2016.] <https://www.infoq.com/articles/semat-essence-fujitsu>.

Wikipedia. 2016. SEMAT - Wikipedia. *Wikipedia: the free encyclopedia.* [Online] San Francisco (CA): Wikimedia Foundation, 2016. [Citace: 29. 10 2016.] https://en.wikipedia.org/wiki/SEMAT#Regional_Chapters_of_SEMAT.