

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS

Semestr LS 2014/2015
Autoři Petr Fišer, fisp00
– jméno, příjmení , xname
Téma CHAOS Manifesto 2013
Datum odevzdání

CHAOS Manifesto 2013

Petr Fišer, fisp00@vse.cz

Abstrakt

Práce se zabývá rozborem klíčových částí studie CHAOS Manifesto 2013, která se věnuje podstatným faktorům ovlivňujícím úspěšnost IT projektů. Studie taktéž vydává sadu doporučení pro projektový management malých projektů (do 1 mil. dolarů v nákladech na práci).

V práci je provedena analýza Manifesta a na jejím základě jsou zmíněna klíčová doporučení, výstupy a případně techniky pro řízení malých IT projektů.

Klíčová slova

projektový management, IT projekt, The Standish Group, CKC, CHAOS

Obsah

1 Úvod.....	3
2 CHAOS Manifesto - charakterizace a analýza	4
2.1 Stručný náhled.....	4
2.2 Úspěšnost projektů	5
2.3 Malé projekty	6
2.4 Klíčové faktory pro úspěšnost IT projektu.....	6
2.5 Podpora vedení projektu.....	6
2.6 Zapojení uživatelů	7
2.7 Rozsah projektu.....	7
2.8 Schopnosti řešitelského týmu.....	8
2.9 Projektové řízení	8
2.10 Agilní techniky	9
2.11 Jasně definované požadavky	9
2.12 Týmová chemie	9
2.13 Provedení projektu	10
2.14 Podpůrné nástroje.....	10
3 Výstupy a cenné zkušenosti	10
3.1 Podpora vedení projektu.....	10
3.2 Zapojení uživatelů	10
3.3 Rozsah projektu a vliv použité metodiky	12
4 Závěr	14

1 Úvod

Informační technologie jsou dnes již nedílnou součástí našich životů - počítače, tablety, chytré telefony, Internet jako celek nebo i jen jeho nejčastěji viděná složka World Wide Web. Uživatelé používají mnoho služeb, které však musel nejdříve někdo vytvořit. Bez ohledu na to, zda je služba dostupná široké veřejnosti nebo jen omezenému počtu uživatelů (např. interně firemním zaměstnancům), musela být nějakým způsobem implementována a zprovozněna - a téměř jistě se tak stalo formou IT projektu.

Za každou službou poskytovanou koncovým uživatelům tedy najdeme více či méně úspěšný a více či méně dokončený IT projekt. Množina projektů nám však narůstá i o ty, výsledky jejichž implementace uživatelé nevidí přímo, maximálně třeba pociťují jejich dopady (např. zlepšení dostupnosti a odezvy webových stránek díky implementaci load-balancingu). Jde o různé implementace zálohování, vysoké dostupnosti, integrace, konsolidaci infrastruktury, atd.

Nicméně vždy je taková akce realizována formou projektu určitého rozsahu. Projekt nějakým způsobem probíhá a ve výsledku může skončit úspěšně, neúspěšně (byl zrušen) nebo s částečným úspěchem (došlo k nedodržení termínů, rozpočtu nebo pouze částečné implementaci funkcionality).

Je logické, že se snažíme, aby projekty byly vždy dodány včas, v nákladech a ve slíbeném rozsahu a aby výsledky naší práce uživatelé opravdu používali - zkráceně, aby byl projekt úspěšný. Proto je dobré se zabývat metodami řízení IT projektů, které nám ve snaze naplnit tento cíl pomáhají, a snažit se je neustále zlepšovat.

Za účelem lepšího řízení projektů byly vytvořeny metodiky, z nichž některé byly standardizovány. Mnoho organizací však stále používá své vlastní interně vyvinuté metodiky.

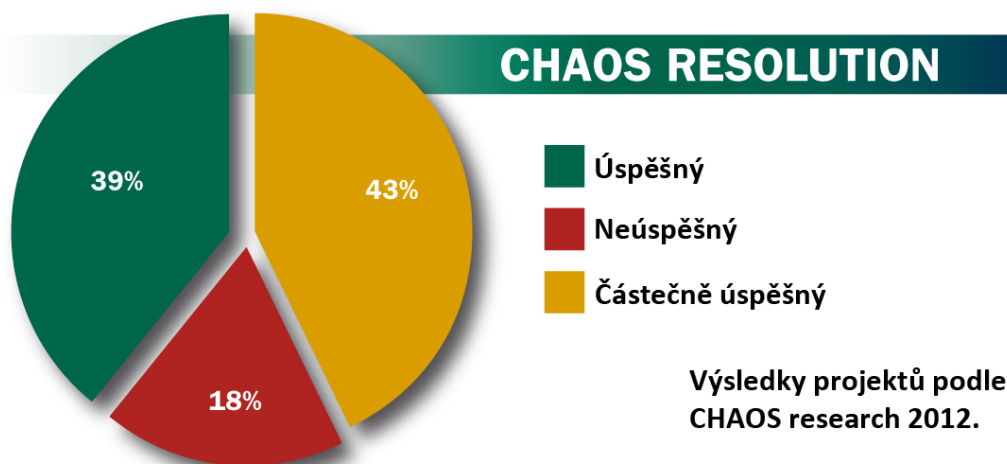
Ovšem úspěch projektu neleží pouze na použité metodice, protože je zde také nekonečná spousta faktorů a vnějších i vnitřních vlivů, které projekt ovlivňují. Většina z nich je nedůležitá, ale najdou se mezi nimi některé, které kriticky ovlivňují šanci na úspěch projektu jako celku.

Právě těmito faktory se zabývá CHAOS Manifesto - analýza každoročně prováděná The Standish Group. Jde o identifikaci klíčových faktorů ovlivňujících úspěšnost IT projektů a následná doporučení. Manifesto je přitom zaměřeno na malé projekty, nikoli na megalomanské akce se stamilonovými rozpočty, a jeho hlavní myšlenkou je, že vše se dá řešit formou malých projektů.

Cíl práce

Cílem této práce je analyzovat zprávu CHAOS Manifesto 2013 a tento poměrně dlouhý text převést do úspornější podoby a to při zachování hlavních informací uvedených v Manifestu. Dále pak z dostupných údajů a kontextu stručně formulovat základní poselství Manifesta.

Tohoto cíle se budeme snažit dosáhnout jednak analýzou textu zprávy CHAOS Manifesto 2013 a jejím následným skloubením s informacemi o projektových a vývojových metodikách. Cílem tohoto snažení bude dostat ucelený obraz o statistikách a průzkumech uvedených v Manifestu a zasadit je do širšího kontextu.



Obrázek 1: Podíl úspěšných, částečně úspěšných a neúspěšných projektů analýzy CHAOS research za rok 2012.

2 CHAOS Manifesto - charakterizace a analýza

V této kapitole uvedeme CHAOS Manifesto a analyzujeme jeho jednotlivé kapitoly včetně doporučení, která dávají.

2.1 Stručný náhled

V roce 1985 začala společnost Standish Group shromažďovat a analyzovat data o IT projektech. Z těchto dat, která prošla analýzou, postupně vznikla tzv. CHAOS Database. Data byla, a jsou, sbírána na základě dotazníků. Databáze je neustále doplňována.

Z těchto sebraných informací postupně vzniklo CHAOS Knowledge Center (CKC), které jednak obsahuje sebrané informace a jejich analýzy, ale také závěry a doporučení pro projektové řízení IT projektů. Celé CKC obsahuje 100 tzv. Best Practice Points (důležitá doporučení).

CHAOS Manifesto je zúžením CKC na oblast malých IT projektů. Data pro něj byla sebrána jak z CKC, tak dalšími cestami - extra dotazníkové průzkumy, sledování vývoje projektu, interview, atd.

Tento průzkum zahrnuje data z osmnáctiletého období (k 2013), nicméně roky 1994 - 2002 byly z dalších analýz vyřazeny (důvodem byla transformace celé CKC). Průzkum i tak pracuje s daty o

zhruba 50 000 projektech.

Protože CHAOS je průzkum globální, je nutné zmínit odkud byla data sebrána. Vesměs se jedná o projekty z USA (60%) a z Evropy (25%). Zbýlých 15% je sebráno z ostatních částí světa. CHAOS Manifesto těchto 15% blíže nespecifikuje.

Data jsou sbírána z různých firem. Cca 50% dat v průzkumech pochází od tzv. Fortune 1000 společností, zhruba 30% dat je ze středně velkých a 20% z malých firem.

2.2 Úspěšnost projektů

Projekty mohou během své realizace spadnout do některé ze tří základních kategorií - úspěšné (successful), neúspěšné (failed) nebo částečně úspěšné (challenged).

Naším cílem jsou projekty úspěšné, tedy takové, které jsou dodány včas, mají všechny slíbené funkce a nebyl překročen rozpočet projektu. Pokud alespoň jeden z těchto požadavků není splněn, ale projekt se i přesto podaří dokončit, mluvíme o projektech částečně úspěšných. Projekty nedokončené, zastavené, jsou pak celkem logicky označovány jako neúspěšné.

V roce 2004 bylo úspěšných jen 29% IT projektů analyzovaných v CKC. S tím, jak se zlepšují projektové metodiky, nástroje, rozhodovací a kontrolní procesy, apod., došlo k výraznému zvýšení úspěšnosti IT projektů. V roce 2012 to bylo již 39%.

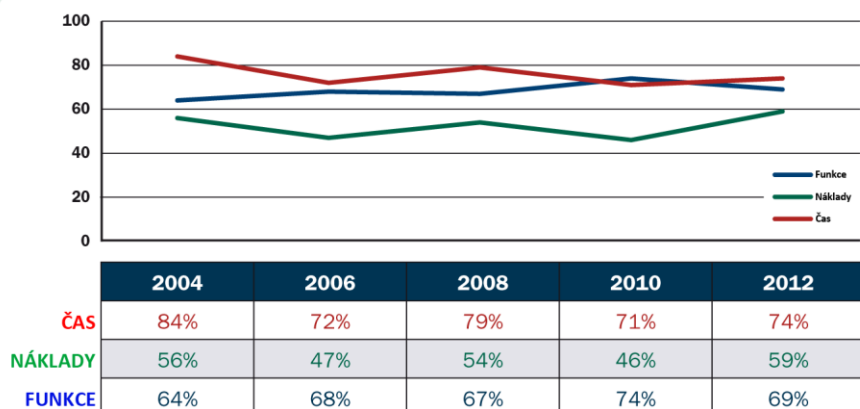
Toto číslo se ale zvýšilo pouze na úkor částečně úspěšných projektů (53% v roce 2004, 43% v roce 2012), zatímco počet projektů neúspěšných zůstal zhruba stejný, okolo 20%.

Z částečně úspěšných projektů se významně snížil počet projektů překračujících termíny. Pokud se podíváme na projekty překračující rozpočet nebo ty, kde nebyly implementovány všechny funkce, zjistíme, že se v těchto trendech nedějí významnější změny. Obrázek 2 ukazuje zmíněné trendy.

OVERRUNS AND FEATURES

Překročení projektu o čas nebo náklady. Modrou barvou je znázorněn počet dodaných funkcí vzhledem k funkcím slíbeným.

Data pochází z CHAOS Research mezi lety 2004 a 2012.



Obrázek 2: Statistika problémů v částečně dokončených projektech za období 2004 - 2012.

2.3 Malé projekty

Základní myšlenkou úspěšného řízení IT projektu je, podle CHAOS Manifesta, udržet projekt malý. Částečně to vyplývá i z Obrázku 2, kde skutečně implementovaná funkcionální je zhruba 70% funkcionality slíbené. Cílem tedy může být nedělat všechny funkce, které má výsledný produkt mít.

Dále je nutné omezit celkovou velikost projektu. Za malý projekt je považován takový, jehož náklady na odvedenou práci nepřekračují 1 mil. amerických dolarů, případně 250 tis. euro.

2.4 Klíčové faktory pro úspěšnost IT projektu

Hlavní částí - a poselstvím - CHAOS Manifesta je identifikace klíčových faktorů pro úspěšnost IT projektu. Těchto faktorů je ve zprávě uvedeno deset nejdůležitějších a je mezi ně rozděleno celkem 100 „důležitostních“ bodů.

CHAOS Manifesto 2013 uvádí tyto klíčové faktory:

1. Podpora vedení projektu (Executive Management Support) - 20b.
2. Zapojení uživatelů (User Involvement) - 15b.
3. Rozsah projektu (Optimization) - 15b.
4. Schopnosti řešitelského týmu (Skilled Resources) - 13b.
5. Projektové řízení (Project Management Expertise) - 12b.
6. Agilní techniky (Agile Process) - 10b
7. Jasně definované požadavky (Clear Business Objectives) - 6b.
8. Týmová chemie (Emotional Maturity) - 5b.
9. Provedení projektu a kontrolní mechanismy (Execution) - 3b.
10. Podpůrné nástroje (Tools and Infrastructure) - 1b.

Faktory 1 - 5 jsou považovány za stěžejní, ostatní - ač důležité - jsou pak spíše doplňkové. Je zajímavé si povšimnout, že oněch prvních pět faktorů dává bodově celých 75 bodů ze sta. S nadsázkou lze tedy říci, že dávají dohromady 75% úspěšnosti IT projektu. V následujících oddílech tyto faktory blíže rozebereme.

2.5 Podpora vedení projektu

Každý projekt potřebuje tzv. sponzora. Tato osoba má hlavní slovo v rozhodování a také má na starosti úspěch projektu jako celku. Pokud sponzor projektu nevěří, je projekt v podstatě odsouzen k nezdaru.

Výhodou v malých projektech navíc bývá, že sponzorem nemusí být vysoce postavený manažer - ať už z důvodu rozpočtu nebo potřebných interních kompetencí. Tedy lze projekt přidělit sponzorovi, který je pro něj „nadšený“ a který nejspíše přijme projekt kompletně za svůj. Tím se velmi vylepšují šance na úspěch.

2.6 Zapojení uživatelů

Projekty jsou dělány protože chceme něco vytvořit. Tento výstup má své uživatele. Například jako firma můžeme chtít pro naše zákazníky vytvořit e-shop. Objednatelem e-shopu bude sice firma, ale uživatelé budou potenciálně všichni lidé - ať už stávající nebo potenciální zákazníci.

Protože výstupy projektů používají uživatelé, je velmi dobrý nápad zapojit je do tvůrčího procesu. Hlavní výhodou je zde porozumění koncovým uživatelům. Při sběru požadavků totiž dostávají realizátoři seznam, **co** má např. naše aplikace umět. Bohužel takový seznam požadavků na funkcionalitu je ze své podstaty nepřesný (a to i kdyby ho formulovali sami budoucí uživatelé). Chybí v něm totiž informace **proč** je daná funkcionalita zapotřebí.

Zapojení uživatelů přímo do procesu vývoje nám umožní velmi rychle tyto informace doplnit, a tím velmi zpřesnit hranice projektu jako takového. Navíc, v pozdějších fázích vývoje, získáváme velmi cennou skupinu uživatelů, kteří jsou schopni pomoci při testování.

2.7 Rozsah projektu

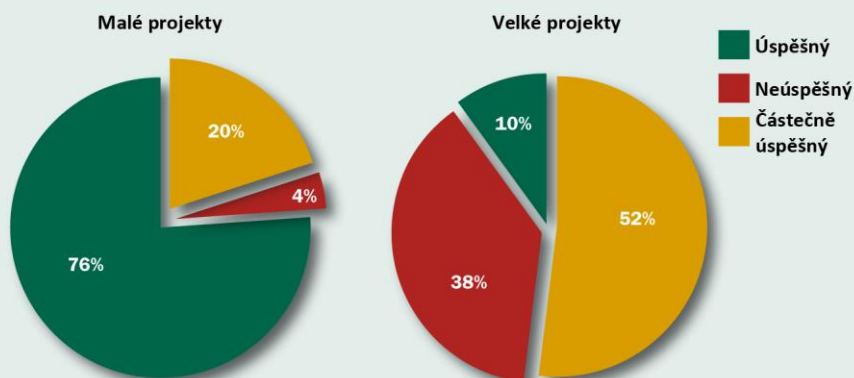
Je nutné velmi pozorně hlídat rozsah projektu jako celku. Ze statistiky CKC vyplývá, že i zde platí pravidlo 80-20. V kontextu vývoje software tak, že 20% implementované funkčnosti pokrývá 80% přínosu pro uživatele. Zbytek pak tvoří funkcionalita, která je zřídka používaná nebo ji uživatelé neocení.

Pokud zmenšíme rozsah projektu o těch 80% času, který bychom věnovali implementaci 20% nedůležitých funkcí, má to velmi příznivý dopad na cenu a složitost celého projektu - jednodušší komunikace myšlenky k/od zákazníka, jednodušší projektový management, kratší fáze analýzy, vývoje a testování. Navíc neimplementování některých funkcí nás neomezuje je podle požadavků dodatečně realizovat.

Ze statistik dále vyplývá, že menší projekty mají velmi vysokou míru úspěšnosti - mnohem vyšší než projekty s deseti nebo stamilionovými rozpočty. Tuto statistiku ilustruje Obrázek 3.

CHAOS RESOLUTION BY LARGE AND SMALL PROJECTS

Úspěšnost IT projektů v roce 2012.
Malé projekty splňují požadavek na náklady na práci maximálně 1 mil. USD.
Jako velké projekty byly vybrány projekty překračující 10 mil. USD v nákladech na práci.



Obrázek 3: Porovnání úspěšnosti malých a velkých IT projektů v roce 2012.

2.8 Schopnosti řešitelského týmu

Protože bychom měli cílit na malé projekty, není v principu zapotřebí velkých projektových týmů. V malém projektovém týmu je ovšem vidět každá nedokončená nebo chybová práce. Proto bychom měli hledat schopné realizátory a ty do svého týmu začlenit.

Podle průzkumů Standish Group se používáním velmi znalých členů týmu dostáváme k optimálnímu poměru náklady/výkon. Nicméně je dobré tým nestavět z těch dobrých, ale především z těch správných. Pokud postavíme tým pouze z velmi znalých lidí, můžeme se dočkat něčeho, co Manifesto nazývá „superstar syndrome“. V takové situaci má přítomnost hvězdy na svědomí špatnou kooperaci nebo v extrémních případech i rozpad projektového týmu. Proto je potřeba tyto případné tendence v projektovém týmu hlídat a adekvátně je řešit.

2.9 Projektové řízení

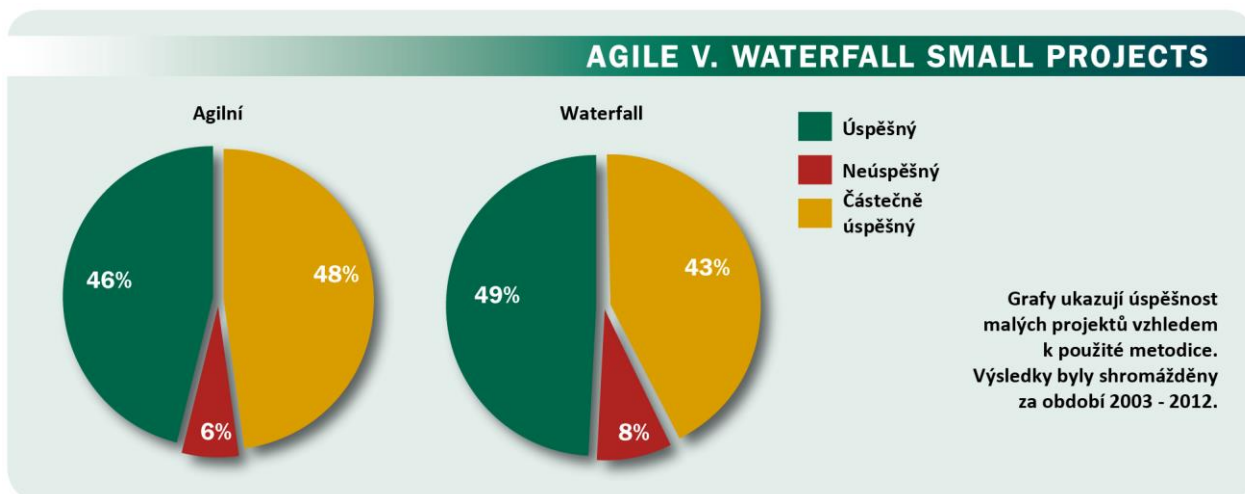
Doporučení Standish Group se zde týká hlavně osobnosti projektového manažera a případně process mastera pokud se pohybujeme v agilních metodikách.

Hlavní pointou projektového manažera je zjednodušení procesu projektového řízení pro ostatní členy týmu tak, aby jim přišel jednoduchý a v podstatě přirozený. V průběhu projektu je také nutné dělat velmi mnoho rozhodnutí, která mohou projekt velmi příznivě, ale i velmi nepříznivě, ovlivnit. Dělat správná rozhodnutí ve správný čas je taktéž doménou dobrého projektového manažera.

Zajímavostí je, že certifikace projektového managementu nezaručují kvalitního projektového manažera. Analytici zde poukazují na státní správu USA a Velké Británie, které zaměstnávají velmi mnoho certifikovaných projektových manažerů, ale jejich projekty mají velmi špatnou úspěšnost.

2.10 Agilní techniky

Z pohledu běžného pozorovatele je tento bod možná trochu překvapivý. CHAOS Manifesto dokládá tvrdými daty jednoduchý fakt - pokud jsou projekty malé, tak metodika projektového řízení není příliš důležitá. Agilní metodiky sice vývoji pomáhají, ale nejsou pro něj nezbytné. Celá analýza je shrnuta do grafu na Obrázku 4.



Obrázek 4: Úspěšnost malých projektů vzhledem k použitým metodikám.

2.11 Jasně definované požadavky

Pro úspěch projektu je nezbytné mít jasně definované business požadavky. U velkých projektů je toto nezbytné hlídat průběžně, protože velký projekt bývá typicky zadán velmi vágně. Na spoustu podrobností se přichází až při realizaci, a je proto nutné business požadavky opakovaně revidovat, doplňovat a konzultovat.

Naproti jasná definice business požadavků je něco, co je pro malé projekty téměř implicitní. Malé projekty se proto mohou nést ve smyslu „máme tento problém, pojďme ho vyřešit“. Zde jsou pak business zadání a požadavky přesně dané již od začátku.

2.12 Týmová chemie

Týmová chemie hraje velmi podstatnou roli při zpracování projektu. Nelze mít tým schopných lidí kteří nekomunikují s vedením projektu, nebo nedejbože ani mezi sebou. Práce v takovém projektovém týmu je potom frustrující pro všechny zúčastněné a na celkové kvalitě projektu se to odrazí.

Dalším problémem bývá často komunikace směrem k uživatelům a sponzorům. Projekt totiž má, vedle své faktické hodnoty, ještě hodnotu pocitovou. Ta je mnohem důležitější než hodnota faktická, protože na ní se zakládá rozhodování o projektu jako takovém. Tedy úspěšnost projektu závisí z „větší poloviny“ na pocitové hodnotě.

2.13 Provedení projektu

Hlavní předností malých projektů je možnost pouštět se do většího rizika. Malé projekty mají ze své podstaty zabudovaný deadline, kterým je vyčerpání přidělených finančních prostředků. Protože jsou projekty malé, jsou malé i případné ztráty vzniklé neúspěchem.

Nicméně protože se pouštíme do většího rizika, nemáme z principu dost informací pro analýzu rizik - náš projekt může například vyšlapávat novou cestiku k používání nám zatím zcela neznámé technologie.

Proto je velmi moudré hned od začátku projektu nastavit metriky a pravidelně podrobovat projekt kontrolám. I přesto, že v malém projektu jde o přidání formalit, do určité míry tím nahrazujeme risk analýzu. Pokud se dostaneme v měření k závěrům, které neodpovídají našim předpokladům a očekáváním, jde nejspíše o znamení nějakého problému.

2.14 Podpůrné nástroje

Podle doporučení CHAOS Manifesto je nejlepší mít co nejméně podpůrných nástrojů pro řízení projektu a vývoje. Než aby se členové týmu učili s mnoha novými aplikacemi, mohli by se v ušetřeném čase věnovat realizaci projektu.

Pointou zde je vytvořit standardizované procesy a zvyky v používání malého množství podpůrných aplikací. A tyto aplikace pak efektivně využívat.

3 Výstupy a cenné zkušenosti

Základním stavebním kamenem CHAOS Manifesto je velké množství sebraných dat za dlouhé časové období. Z následné analýzy pak vyplývají doporučení autorů ohledně toho, jaké faktory velmi ovlivňují úspěšnost projektu a také na co si dát pozor. V této kapitole rozebereme ty nejpodstatnější z nich.

3.1 Podpora vedení projektu

Jak již bylo zmíněno v Oddíle 2.5, podpora vedení projektu je pro jeho úspěch klíčová. Z pohledu realizátora není nic horšího, než když vám nevěří člověk, který vás platí za práci. K tomuto jednoduchému faktu není vlastně co dodat.

3.2 Zapojení uživatelů

V Oddíle 2.6 byla zmíněna důležitost zapojení uživatelů do projektu. Byl zde také uveden krátký příklad s e-shopem a tvrzení, že katalog sebraných požadavků, který říká **co** je třeba implementovat, je nepřesný, protože mu ve většině případů chybí informace **proč** je na něj ten který požadavek zařazen. Abychom to ilustrovali na příkladu, vraťme se k e-shopu.

Příklad

Představme si, že implementujeme nový e-shop. Na katalogu požadavků je uveden požadavek „Obsah košíku lze exportovat do PDF“. Tento požadavek byl sebrán přímo od uživatelů, máme tedy **co** uživatel chce. Dále se tím nezabýváme a funkcionalitu implementujeme.

Co jsme ale nevěděli, bylo, že uživatel chce export do PDF, aby si mohl obsah košíku vytisknout. Protože na jiném e-shopu viděl možnost exportu košíku jako PDF dokumentu, pokládal to za to, co chce od našeho řešení. Učinil tak proto, že si možnost exportu buď nevědomě asocioval s funkcionalitou „Vytisknout obsah košíku“, nebo ho nenapadlo, že vytisknutí lze realizovat i jinak než exportem do PDF.

Protože nám v katalogu chyběl uživatelův důvod **proč**: „protože si to chci vytisknout“ implementovali jsme v e-shopu něco uživatelsky nepříliš přívětivého. Když víme, že hlavním požadavkem je tisk obsahu košíku, můžeme ho implementovat například pomocí JavaScriptu a stránku košíku speciálně pro tiskárnu nastýlovat.

Tedy nám vznikly dvě varianty, pojďme se podívat na jejich workflow.

Export do PDF

1. Uživatel klikne na „Exportovat do PDF“ a systém vygeneruje PDF dokument.
2. Uživatel si PDF uloží.
3. Uživatel najde na disku uložený PDF soubor a otevře ho.
4. Uživatel klikne na „Tisk“ v PDF editoru a potvrdí okno tisku, tiskárna dokument vytiskne.
5. Uživatel zavře PDF a smaže ho z disku.

Tisk z webu

1. Uživatel klikne na „Tisknout obsah košíku“.
2. Uživatel potvrdí okno tisku, tiskárna dokument vytiskne.

Jak je očividné, s ohledem na uživatelskou přívětivost jsme s druhou variantou někde zcela jinde. Nehledě na to, že pokud toto víme od rané fáze vývoje, jsme schopni podle toho upravit rozsah projektu. Ve výše zmíněném případě bychom, dejme tomu, nemuseli používat programové knihovny umožňující export do PDF. Protože knihovny jsou licencované, podařilo se nám snížit náklady.

Zůstaneme-li ještě chvíli u našeho e-shopu, uvědomme si, že uživatelé nám mohou pomoci omezit rozsah projektu velmi výrazně. Díky nim zjistíme, co budou používat nejvíce a co třeba nebudou

používat vůbec. Z principu, pokud si rozebereme jakýkoliv úspěšný internetový obchod, dojdeme k závěru, že jeho základní funkcionalita sestává z několika málo prvků: přihlášení, registrace, uživatelský profil s platebními informacemi, vyhledávání podle kategorií, fulltextové vyhledávání, možnost třídění nalezených výsledků, košík a platba, případně ještě hlídací pes (notifikace při slevách nebo dostupnosti zboží).

Všechny velké e-shopy mají tyto funkce velmi uživatelsky přívětivě zpracované a efektivní na používání. Důvod je jednoduchý - protože uživatelé e-shopu nic jiného prakticky nepoužívají.

Dejme tomu, že budeme chtít implementovat novou funkcionalitu - hlídacího psa notifikujícího uživatele, pokud máme na skladě více než n kusů zboží. Uživatel nám ale nejspíše řekne, že to ho nezajímá, že musí něco hlídat. Uživatel si chce objednat a chce se dozvědět, kdy mu zboží dorazí. Sehnat dostatečný počet kusů má být starost prodejce, nikoli kupujícího.

Tedy najednou se dostáváme od funkce hlídacího psa v e-shopu k řešením Available/Capable to Promise.

Pointou těchto dvou krátkých příkladů je ovšem, že my bychom nic nevěděli, pokud bychom se uživatelů nezeptali. Implementovali bychom pak funkcionalitu, kterou by nakonec možná nikdo nepoužíval.

Pokud odhlédneme od e-shopu, má zapojení uživatelů další pozitivní dopady. Vyvíjíme-li software, získáváme potenciální testery, které můžeme využít v pozdějších fázích vývoje, případně jako participanty v UX testech.

Speciálně v případě tvorby interních řešení v organizaci pak jsou uživatelé, zapojení nejdříve jako zdroje informací, pak jako první uživatelé vpuštění do aplikace, velmi dobrým způsobem pro evangelizaci ostatních. Tvrzení „ale my to už používáme a funguje to“ je v rámci interní komunikace vizitkou slibného projektu, na který se vyplatí dát peníze. Protože management získává podklady od svých podřízených (našich uživatelů), projekt má velmi dobré šance na úspěch.

3.3 Rozsah projektu a vliv použité metodiky

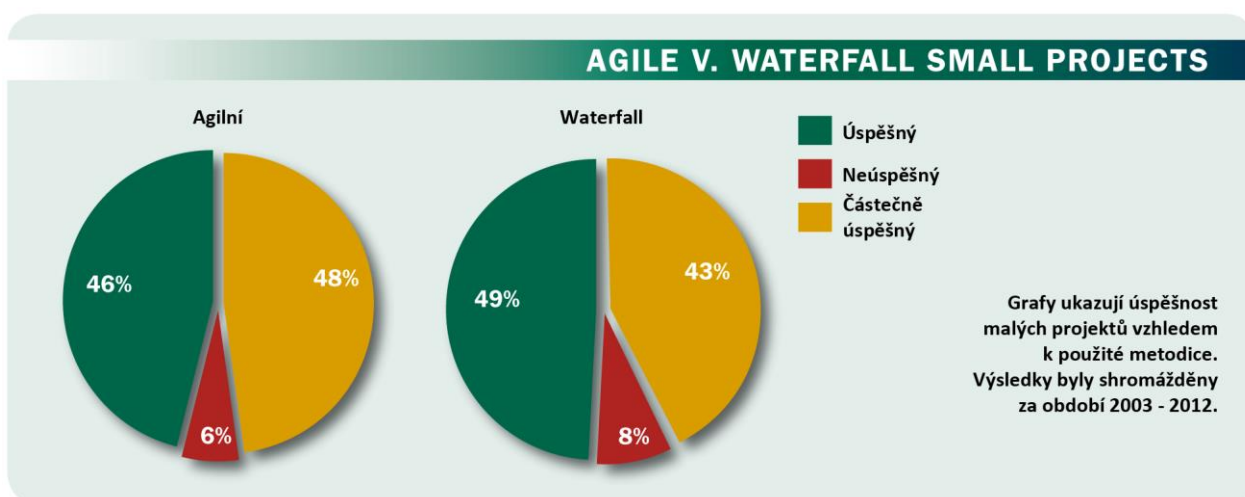
Jak můžeme účinně určit rozsah projektu jsme ukázali v předchozím oddíle, kde jsme do tvůrčího procesu zapojili samotné uživatele. Nyní tuto myšlenku dovedme ještě dále - nedávejme uživatelům to, co chtějí, dejme jim to, co potřebují.

Z analýzy CKC vyplynulo, že cca 20% implementované funkcionality dává 80% uživatelské hodnoty. Ostatní funkcionalita je doplňková - buď se nepoužívá vůbec, nebo jen velmi zřídka. Ať tak či tak, je diskutabilní, zda takovou funkcionalitu vůbec implementovat. Tím můžeme zmenšit rozsah projektu.

Celkově se dá říci, že v rozsahu projektu, respektive v jeho „malosti“ je skrytý úspěch. Podle definice Standish Group je malý projekt takový, jehož náklady v odvedené práci nepřesahují 1 mil. amerických dolarů (dle definice EU malý projekt nepřesahuje 250 tis. euro v odvedené práci). Takový projekt je ze svého principu krátký a velmi dobře ohraničený. Základní kontrolní mechanismy projektu jsou najednou zbudovány implicitně - například zastavení projektu kvůli vyčerpání rozpočtu. Projekt nemá tendenci být nafukován donekonečna.

Velmi zajímavou vlastností malých projektů je jejich krátkost, která má vliv i na použitou metodiku řízení. Velmi často se dnes setkáváme s poměrně propagovaným názorem, že agilní metodiky jsou všelékem na problémy s úspěšností IT projektů. Ve skutečnosti tomu tak není.

Podle výsledků Standish Group, které velmi hezky reprezentuje Obrázek 5, není metodika řízení projektu stěžejní.



Obrázek 5: Agilní vs. klasické metodiky při řízení malých projektů.

Je zde vidět, že úspěšnost se liší jen o několik málo procent. Vzhledem k tomu, že jsme u průzkumů a statistik z cca 50 000 projektů, tak počítejme 1% = 500 projektů. Z tohoto pohledu se dá říci, že oba grafy jsou prakticky stejné. Tento výsledek může být pro někoho poměrně překvapivý. Uveďme krátký příklad, že výsledky analýzy dávají velmi hezký smysl.

Příklad

Předpokládejme projekt, jehož realizace nám zabere dva měsíce. Použijeme-li vodopád, realizujeme od začátku do konce a za dva měsíce je projekt dokončen. Použijeme-li např. SCRUM se čtrnáctidenním sprintem, dostaneme se k tomu, že náš projekt bude mít čtyři sprinty. Je dokonce možné, že sprinty budou pouze tři. V takovém případě ale nestihneme iterativní přístup, díky

kterému je SCRUM velmi silný, skoro použit. A tedy je vlastně jedno, jakou metodiku zvolíme.

Předchozí příklad je pro účel ukázání myšlenky mírně přehnaný. Kdyby projekt trval například půl roku, což je s daným milionem dolarů bez problémů možné, iterací SCRUMu realizujeme více a metodika se „chytí“ a bude fungovat.

Zde nám ale začne opět hrát do karet krátkost projektu. Největší problémy totiž v IT projektech přináší neustálá proměnnost prostředí, trendů a infrastruktury. Bez ohledu na to, jakou metodiku použijeme, nebudeme-li schopni při realizaci na tyto změny reagovat, budeme mít nejspíše smůlu. Pokud jde například o firemní infrastrukturu, tak se dá počítat se stálým stavem (bez změny aplikací, topologie sítě, ...) jako s něčím, co trvá maximálně několik měsíců.

Pokud je projekt krátký, stihneme ho realizovat, aniž by se prostředí nějak citelně změnilo. A pak už je jedno, jakou metodiku k jeho realizaci použijeme - hlavní problémy nemusíme řešit.

Poučení je zde tedy jasné - všechny IT projekty držet „malé“. Podle CHAOS Manifesta nemají mamutí IT projekty vůbec šanci na úspěch. Navíc je u nich problém s lavinovým efektem, kdy je problém dlouho běžící, a adekvátně drahý, projekt prostě zastavit.

4 Závěr

Hlavním cílem této práce bylo podat přehled reportu CHAOS Manifesto 2013 a zasadit některá jeho zjištění a doporučení do širšího kontextu. Tohoto cíle bylo dosaženo v kapitolách 2 a 3. Celý text čerpal z informací uvedených v Manifestu, z obecných informací o projektových metodikách a také z osobních zkušeností autora z praxe v IT - implementací Identity a Access Managementu a databázových a infrastrukturních řešení. Tyto zkušenosti byly použity v kapitole 3 k dokreslení příkladů a výkladu některých částí CHAOS Manifesto.

Pokud bychom měli celou analýzu shrnout, klíčové pro úspěšný projekt jsou tyto faktory:

1. Podpora vedení projektu
2. Zapojení uživatelů
3. Rozsah projektu

Spolu dohromady tvoří, vyjádříme-li se poněkud abstraktněji, polovinu úspěchu. Alespoň pokud bychom se řídili „důležitostními“ body, které jim analytici Standish Group přiřadili.

A pokud bychom měli obsáhnout celé CHAOS Manifesto jednou větou, bylo by to pravděpodobně:
„Řiďte si projekty jakou metodikou uznáte za vhodné, hlavně je dělejte malé a zapojte do nich vedení a uživatele.“

Reference

Níže je uveden seznam zdrojů, které nějakým způsobem přispěly k informacím - či názorům - prezentovaným v tomto textu. Hlavním informačním zdrojem, a jediným zdrojem obrazového materiálu, je text CHAOS Manifesto 2013. Ostatní texty byly použity spíše jako doplňkové, pokud si autor potřeboval vyjasnit některé pojmy.

THE STANDISH GROUP. CHAOS Manifesto 2013: Think Big, Act Small. [online]. 2013, s. 1-52 [cit. 2015-03-15]. Dostupné z: <http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>

SCHWABER, Ken. *Agile project management with Scrum*. Redmond, Wash.: Microsoft Press, c2004, xix, 163 p. ISBN 07-356-1993-X.

CUSUMANO, Michael A. a Stanley A. SMITH. Beyond the waterfall: software development at Microsoft. [online]. 1995 [cit. 2015-03-15]. Dostupné z: <http://hdl.handle.net/1721.1/2593>

POPPENDIECK, Mary a Tom POPPENDIECK. *Lean software development: an agile toolkit*. 10. print. Boston: Addison-Wesley, 2006. ISBN 03-211-5078-3.

DYBÅ, Tore a Torgeir DINGSØYR. Empirical studies of agile software development: A systematic review. *Information and Software Technology*. 2008, vol. 50, 9-10, s. 833-859. DOI: 10.1016/j.infsof.2008.01.006. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0950584908000256>