

Disciplined Agile Delivery

Tomáš Koudelka, Ondřej Sklenář
12.12.2012

OBSAH

Úvod.....	1
Co je Disciplined Agile Delivery	1
Klíčové principy.....	1
Lidé	2
Důraz na učení	4
Cílem je řešení, nikoli software	4
Cíle v rámci životního cyklu.....	4
Tým v kontextu podniku.....	6
Zaměření na hodnotu a rizika.....	7
Škálovatelnost.....	9
Životní cyklus projektu [1].....	11
Zahájení projektu	11
Cíle fáze	12
Průběh fáze	14
Tvorba řešení.....	14
Cíle fáze	15
Průběžné cíle.....	16
Průběh konstrukční fáze	17
Přechodová fáze	17
Cíle fáze	17
Průběh přechodné fáze	18
Školení	18
Závěr	19
Zdroje.....	20

ÚVOD

V této práci se věnujeme základnímu představení metodiky Disciplined Agile Delivery (DAD), jejímž hlavním autorem a prosazovatelem je dlouholetý zaměstnanec IBM Scott W. Ambler.

V první části popisujeme klíčové principy, jimiž se řídí, ve druhé potom jednotlivé fáze, kterými podle ní dodávka IT řešení prochází.

CO JE DISCIPLINED AGILE DELIVERY

Disciplined Agile Delivery je poměrně nová metodika pro postupy a způsoby dodání určitého IT řešení zákazníkovi. Scott W. Ambler na jejím konceptu začal pracovat v roce 2007. [1] Z velké části vychází z již známých agilních metodik (Scrum, Extrémní programování, Agilní modelování a dalších) a staví tedy na základních principech Manifestu agilního programování z roku 2001. Zároveň však tyto rozšiřuje a upřesňuje a přidává například popis celého životního cyklu dodávky IT řešení.

Důvodem vzniku DAD byla především absence metodiky, která by umožnila používat agilní přístup zodpovědně i ve větších organizacích a týmech. Následky v případě neúspěchu by tam totiž byly mnohem závažnější než u týmu malého. Dosavadní převahu malých týmů v agilních postupech dokazuje i průzkum "Agile Practices Survey" z roku 2009, podle kterého je v 68% agilních týmů pouze 1-10 IT profesionálů. [2]

Pokud jde o charakteristiku DAD, dala by se popsat jako hybridní agilní metodika, která klade důraz na jedinečnou úlohu lidí v agilním vývojářském týmu, na význam jejich ochoty a schopnosti se učit, na skutečnou hodnotu vyvíjeného řešení a na životní cyklus dodávky řízený cíli. Dalšími charakteristickými vlastnostmi metodiky je její škálovatelnost a zohledňování okolních podmínek v podniku u každé dodávky IT řešení.

KLÍČOVÉ PRINCIPY

V této části jsou představeny a vysvětleny klíčové principy DAD, které vycházejí ze samotné charakteristiky.

LIDÉ

DAD považuje lidské zdroje a jejich povahu za hlavní faktor úspěchu (či neúspěchu) každého IT řešení. Věnuje se jednak složení týmu a rolím jeho členů a dále jejich vlastnostem a rozložení jejich schopností. Každý člen týmu, stejně jako celý tým, by podle DAD měl být:

- disciplinovaný - to znamená zapojovat se jen do takové práce, kterou může zvládnout a tu pak vykonat s maximální možnou efektivitou,
- organizovaný - tedy plánovat a odhadovat svou práci a snažit se o spolupráci v týmu,
- uvědomělý - tedy vědět, jaké pracovní postupy jim vyhovují a jaké nedostatky je naopak třeba odstranit učním.

Podle DAD by dále všichni členové týmu měli mít komplexní znalosti. Tedy nebýt pouze specialisty ve svém jednom oboru, ale mít i celkový přehled o oborech ostatních, takzvané "cross-skills". Díky tomu mohou totiž pracovat efektivněji a s nižšími náklady - není například nutné dělat podrobnou interní dokumentaci. Zároveň by měli mít maximální možnou snahu nejen o spolupráci a sdílení znalostí s ostatními členy, ale i o získávání a osvojování nových dovedností. Důraz na učení v rámci týmu je blíže popsán dále.

Tímto maximálním propojením týmu DAD poukazuje na to, že hranice mezi jednotlivými částmi vývoje (požadavky, analýza, návrh...) by v agilním prostředí měly být bourány. Jedním z důvodů je snaha o minimalizaci počtu předávek mezi těmito částmi. Při každé takové předávce totiž například vzniká množství dokumentace, která při dostatečných znalostech zúčastněných členů týmu není vůbec nutná a znamená pouze plýtvání časem i penězi (dokumentace musí být za normálních okolností nejen tvořena, ale i pravidelně aktualizována).

Na předávky se lze dívat také jako na potenciální riziko vzniku nedorozumění, a to právě v případě existence dokumentace. Do ní totiž prakticky nelze promítnout tacitní znalosti, které jsou pro porozumění často klíčové, a jež je možné daleko lépe sdílet osobně uvnitř týmu.

Co se týče výčtu týmových rolí, uvádí DAD následující:

- Stakeholder. Je popsán jako ten, jehož ovlivňuje vyvíjené řešení. Nemusí se tedy jednat vždy jen o koncového zákazníka, může jít například i o partnera, s jehož systémem má ten vyvíjený spolupracovat.
- Product owner. Jedná se o člověka, který zastupuje zákazníky. Jeho hlavním úkolem je vysvětlovat a upřesňovat nejasnosti během vývoje, druhotným pak představovat práci týmu stakeholderovi.
- Team members. Lidé, kteří přímo tvoří řešení. DAD je dále nerozlišuje, na rozdíl od jiných metodik.
- Team lead. Je zodpovědný za efektivní fungování týmu, vytváří mu odpovídající podmínky.
- Architecture owner. Označován také jako “technical architect” či “software architect”. Dohlíží na architekturu vyvíjeného řešení, používání existujících frameworků, vzorů apod. V malých týmech může být role spojena s Team lead.

DAD dále každou roli podrobně popisuje, uvádí její práva a povinnosti, avšak na tomto rozdělení jsou nejdůležitější patrně dvě věci:

Jednak je z něho vidět, že DAD nijak dále nedělí řadové členy týmu (“team members”), to souvisí s jejich výše popsanými komplexními schopnostmi. Přesto DAD zcela nezavrhuje skutečné specialisty pouze na určitý obor. Jejich role je však vedlejší (DAD je označuje jako sekundární role a řadí sem například integrátora či nezávislého testera) a mohou tak být součástí týmu například jen na určitou dobu na jakousi výpomoc, aby se ostatní členové mohli věnovat jiné práci (i oni by specialistovu danou činnost zvládli, ale s menší efektivitou).

Druhým důležitým faktem je snaha o aktivní zapojení zástupce zákazníka do vývoje řešení. To umožňuje například pružně reagovat na požadavky během vývoje nebo minimalizovat množství potřebné dokumentace a celkově tak zkrátit čas dodání řešení.

DŮRAZ NA UČENÍ

Autoři DAD konstatují, že za dobu, která uplynula od Manifestu agilního vývoje, se ukázalo, že nejúspěšnější jsou ty organizace, jež pro své zaměstnance vytvářejí dobré podmínky k učení. [1]

V této souvislosti DAD rozlišuje tři oblasti, a to doménovou (identifikování zákaznických potřeb), procesní (zlepšování procesů na individuální i týmové úrovni) a technologickou (efektivní využívání existujících nástrojů). Pro každou z nich dává DAD doporučení a postupy - například pro doménovou oblast inkrementální dodávky již relativně použitelného řešení či již zmíněné aktivní zapojení stakeholdera do vývoje. V duchu hlavních principů DAD dále zdůrazňuje přínosy učit členy týmu novým znalostem z vedlejších oborů, namísto prohlubování znalostí z jejich hlavního oboru (vyzdvihuje “cross-skills” nad specializaci). Základem vhodného prostředí pro učení je vždy dobrá spolupráce v týmu. Avšak DAD poukazuje (na rozdíl od klasických agilních metodik) na důležitost spolupráce nejen mezi členy jednoho týmu, ale i mezi více týmy. Organizace by se tedy měla snažit poskytnout podmínky pro sdílení znalostí, a to například zřizováním interních diskuzí, znalostních bází a podobně.

CÍLEM JE ŘEŠENÍ, NIKOLI SOFTWARE

DAD přichází s lehce odlišným pohledem na celý projekt dodávky IT řešení. Cílem je podle něj dodat takové řešení, které uspokojí potřeby zákazníka (resp. stakeholdera) tak, jak je specifikoval, a které pro něj bude skutečně hodnotné. Většinou tedy nestačí pouze vyvinout software, který zákazník patrně potřebuje, ale často je třeba například upgradovat hardware či pozměnit procesy v organizaci. DAD tímto upozorňuje na zcela zásadní poznatek, že IT profesionálové se věnují mnohem více činnostem než jen vývoji softwaru, a zdůrazňuje komplexnost jejich práce.

CÍLE V RÁMCI ŽIVOTNÍHO CYKLU

Touto oblastí se DAD asi nejvíce vyčleňuje z existujících agilních metodik. Na rozdíl od většiny ostatních se totiž nevěnuje pouze konstrukční fázi projektu, ale popisuje podrobně i jeho počáteční a koncovou etapu. Autoři DAD upozorňují na fakt, že sice existují na jedné straně “těžké” metodiky, jako je Rational Unified Process (RUP) s

detailně popsanými postupy, a na druhé agilní metodiky jako je Scrum, kde jsou postupy popsány povrchně, ale chybí jakási střední cesta. Právě tou by mělo být DAD.

Definovány jsou tři základní fáze životního cyklu:

- Inception Phase (počáteční fáze)
- Construction Phase (konstrukční fáze)
- Transition Phase (přechodová fáze)

Jednotlivým etapám životního cyklu projektu se podrobně věnujeme v druhé části práce.

Důležitým aspektem celého životního cyklu projektu v pojetí DAD je orientace na cíle. V rámci každé fáze DAD vyjmenovává několik hlavních cílů, které je třeba splnit, a navrhuje postupy, jimiž je k nim možné dojít. Seznam cílů samozřejmě není konečný a je pochopitelné, že vždy záleží na předmětu projektu, velikosti týmu a dalších faktorech.

Za hlavní cíle počáteční fáze lze považovat: utvoření týmu, formulování vize projektu (společně se stakeholdery), definování požadavků a rizik, zajištění zdrojů a vytvoření potřebného prostředí.

Cíle v konstrukční fázi jsou opakované v rámci jednotlivých iterací. Vždy je nutné: dodat potenciálně použitelné řešení, reflektovat měnící se potřeby stakeholderů a posunovat se stále blíže produkčnímu (finálnímu) řešení.

Ve finální (přechodové) etapě je například potřeba nejen ujistit se, že řešení je připravené k předání stakeholderovi, ale také že stakeholder je připraven řešení přijmout.

Kromě těchto dílčích cílů dělených dle fází přidává DAD několik soustavných cílů, na které je třeba se zaměřovat po celou dobu projektu. Jde například o snahu zvyšovat dovednosti a schopnosti členů týmu, udržovat a zlepšovat týmové a pracovní prostředí, či věnovat pozornost rizikům.

Další klíčovou charakteristikou je fakt, že DAD tým je vnímán v kontextu celého podniku. Podstatou je, že nestačí sledovat pouze tým samotný, naopak je třeba vědět i o jeho okolí. Autoři DAD si uvědomují, že tým většinou není samostatnou jednotkou, nýbrž působí v rámci nějaké organizace. V té již existuje určité vnitřní uspořádání, stanovená pravidla a postupy, nepsané zvyklosti a podobně.[3]

Často je nutné, aby tým tyto věci respektoval, a to jej pochopitelně ovlivňuje, ať už v pozitivním, či negativním smyslu. Stejně tak tým může zpětně ovlivňovat prostředí kolem sebe, opět pozitivně i negativně.

DAD formuluje ohledně vztahu týmu k okolí několik stěžejních myšlenek:

1 Využívání podnikových aktiv

V podniku pravděpodobně již existují struktury, které je možné a vhodné využít a případně dále rozvíjet. Může se jednat například o bezpečnostní standardy, datové konvence či standardy uživatelského rozhraní. Kromě těchto “nehmotných” zdrojů je třeba využívat také například existující hardware, síťovou infrastrukturu a podobně. V neposlední řadě je vhodná spolupráce s ostatními zaměstnanci podniku, kteří mohou týmu nějak prospět. Využíváním všech těchto zdrojů dochází k celkovému zkvalitňování vyvíjeného řešení, snižování nákladů a zkracování doby realizace.

2 Zlepšování prostředí v organizaci

Řešení, které DAD tým vyvíjí, musí minimálně bezproblémově fungovat v rámci společnosti zadavatele. V ideálním případě by však mělo mít navíc ještě pozitivní vliv na celkové prostředí v této společnosti. Toho je možné dosáhnout zapojením nezávislých testovacích týmů, které prověřují spolupráci dodávaného řešení s ostatními systémy v zadavatelské společnosti.

3 Sdílení poznatků

Jak už bylo zmíněno, jedním z hlavních principů DAD je důraz na učení. To platí nejen uvnitř týmu, ale i ve vztahu týmu k okolí. DAD tým jako celek musí sdílet své vědomosti a znalosti s ostatními týmy v organizaci. Jako podpůrné nástroje mohou sloužit například diskuze, neformální prezentace postupů či přednášky zkušených zaměstnanců.

4 Otevřené sledování týmu

DAD doporučuje několik možností, jak sledovat postup práce týmu. Tou základní je umožnit účast na každodenním setkání DAD týmu jakékoli zainteresované osobě. Tedy udělat tyto mítinky veřejné, a dát tak komukoli možnost, aby se s prací týmu seznámil. Autoři DAD však konstatují, že pracovníci na vyšších pozicích na toto nemají časové možnosti. Proto doporučují (a pokládají za vysoce efektivní) využití specializovaných softwarových nástrojů pro podporu práce agilního týmu. Jedná se například o aplikace pro spolupráci v týmu, plánování, sdílení a sledování úkolů, verzování atd.

ZAMĚŘENÍ NA HODNOTU A RIZIKA

V tomto ohledu DAD rozšiřuje dosavadní agilní metodiky, které se zaměřují na hodnotu (snaha dodat potenciálně použitelný software v každé iteraci, upřednostňování nejdůležitějších požadavků zákazníka apod.), o větší pozornost věnovanou rizikům. Zatímco klasické agilní metodiky definují tři základní rizika (riziko nedodání žádného řešení, riziko dodání špatné funkcionality, riziko nerozpoznatelnosti práce řešitelů), DAD jde v tomto směru dále. Předně formuluje tři základní postupy k minimalizaci těchto rizik:

1 Dodávání potenciálně použitelných řešení.

Tuto již zmíněno strategii využívá například Scrum, který se však snaží především o maximální množství zapracovaných požadavků. Naproti tomu DAD bere v potaz hlavně použitelnost řešení a tedy možnost nasadit jej v jakékoli iteraci do provozu, pakliže o to stakeholder bude stát.

2 Ukázky v rámci iterací

Na konci každé iterace konstrukční fáze je žádoucí předvést řešení stakeholderovi. Účelem je získání zpětné vazby, kterou je možno promítnout do dalších iterací a snížit tak riziko dodání špatné funkcionality.

3 Aktivní spolupráce stakeholderů

Kromě výše zmíněného předvádění by stakeholderi měli být zapojeni i do dalších procesů, jako je například prvotní modelování procesů. Může se jednat třeba jen o práci s tužkou a papírem, ale každopádně je přínosné, pokud stakeholder nejen sděluje své požadavky, ale i aktivně usměrňuje vývoj a podílí se na něm.

Dále pak autoři DAD předkládají několik dalších mezníků v průběhu životního cyklu:

- Odsouhlasení vize se stakeholderem

Na konci počáteční fáze je třeba mít se stakeholderem ujasněnou vizi projektu a rozumět ji. Je třeba si uvědomit, že pokud tomu tak není, nemá projekt význam. Podle zkušeností autorů DAD může tento případ nastat až v 10% projektů.

- Prověřená architektura

I přes tvorbu modelů v počáteční fázi se schopnost vyhovět požadavkům stakeholdera naplno prokáže až ve fázi konstrukční. Psaní kódu je jediný způsob, jak se skutečně přesvědčit, že architektura podporuje všechny stakeholderovy požadavky.

- Stálá životaschopnost

DAD v tomto bodě vychází z postupů Scrumu, který doporučuje posuzovat po každém sprintu (iteraci) životaschopnost řešení stakeholderem. Autoři DAD takto časté prověřování považují za kontraproduktivní a navrhují jej provádět jednou za dva měsíce u kratších projektů a jednou za čtvrtletí u delších.

- Uspokojivá funkčnost

Tento milník na konci konstrukční fáze zajišťuje splnění všech akceptačních kritérií, nebo přiblížení k nim do takové míry, že zbývající problémy mohou být zodpovědně vyřešeny během přechodové fáze.

- Přípravenost ke spuštění

V tomto okamžiku na konci přechodové fáze dochází k rozhodnutí stakeholderů o nasazení řešení do provozu.

- Spokojenost stakeholderů

Finální milník, kdy řešení funguje v provozním prostředí a stakeholdeři jsou s ním spokojeni.

ŠKÁLOVATELNOST

Pojmem škálování se běžně rozumí “individuální úprava jednotlivých praktik agilních metodik a přidávání nových praktik s ohledem na zvětšený objem objektů v projektu.” [4] Přesto, že by se mohlo zdát, že hlavním kritériem škálovatelnosti je velikost týmu, DAD představuje celý soubor faktorů, který je třeba brát úvahu. Jsou to:

- Geografické rozložení

Tým se může nacházet v jedné místnosti, na jednom patře (ale v různých kancelářích), ve více budovách v jednom městě, nebo třeba ve více městech i státech po celém světě.

- Velikost týmu

Velikost agilního týmu se může pohybovat od dvou osob až po stovky, teoreticky až tisíce lidí.

- Dodržování předpisů

Některé týmy musí respektovat dodržování speciálních předpisů, například kvůli typu vyvíjeného řešení (vztahuje se na něj například nějaká zvláštní legislativa).

- Složitost projektu

Agilní přístup může být aplikován jak na jednoduché projekty, jako je třeba tvorba malých webových stránek, tak na složité, například vývoj informačního systému.

- Technologická složitost

Některé týmy vytváří zcela nová řešení “na zelené louce”, od nuly, založené na jediné vybrané technologii nebo platformě, kterou třeba není nutné ani nijak integrovat, jelikož se jedná o samostatný systém. Jiné naopak pracují s několika technologiemi, napříč platformami, či navzájem komunikujícími aplikacemi.

- Organizační rozložení

Agilní týmy pravděpodobně budou většinou pracovat v rámci jedné organizace, jednoho či více oddělení. Není to ale podmínkou - týmy mohou být tvořeny i z členů z různých společností, které se například zabývají podobným oborem. Lze také uvažovat o outsourcingu některých týmových prací.

- Složitost organizace

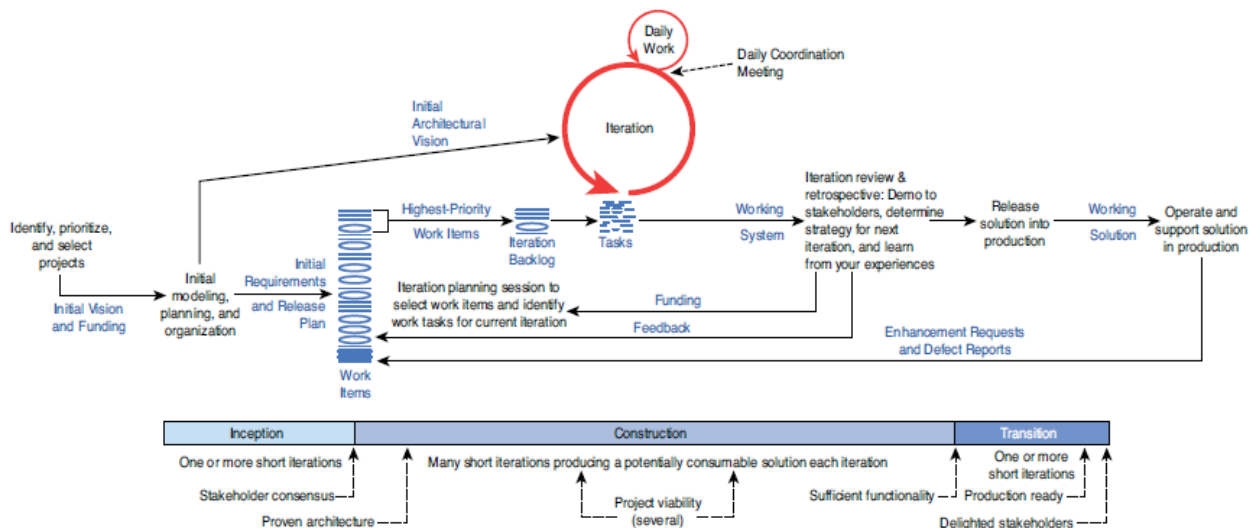
V některých firmách pracují všichni na jednom společném projektu, spolupracují, mají stejný cíl. Jiné však mohou být mnohem více ovlivněny byrokracií, která dělí lidi a týmy do skupin, jejichž cíle a vize vzájemně kolidují.

DAD dále zdůrazňuje, že ačkoli výzvy a cíle jsou v různých pólech škály stejné nebo velmi podobné, cesta k nim může být značně rozdílná.

Vzhledem k tomu, že autoři sami konstatují, že škálování v rámci DAD se budou blíže věnovat v některé z budoucích publikací, lze usoudit, že cítí potřebu tuto problematiku popsat ještě podrobněji. Z toho důvodu je možné výše uvedený přehled považovat jen za základní nástin. Dá se předpokládat, že tvůrci DAD hodlají udělat z této vlastnosti opravdu stěžejní přednost DAD a dokázat tak, že jde o metodiku vhodnou pro široké spektrum agilních vývojových týmů.

ŽIVOTNÍ CYKLUS PROJEKTU [1]

Jelikož je DAD Framework metodikou zabývající se dodáváním IT řešení zákazníkovi, je logické, že má specifikované postupy a jednotlivé fáze projektu. Projekt se skládá z několika fází. Z tohoto pohledu se dá říci, že DAD Framework rozděluje proces tvorby řešení na tři základní části, a to: Zahájení projektu (počáteční fáze), Tvorba řešení (konstrukční fáze) a Uvolňování řešení (přechodová fáze), přičemž životní cyklus je inkrementálního typu.



Obrázek 1: Životní cyklus projektu [1]

ZAHÁJENÍ PROJEKTU

Předtím než začneme řešit samotný průběh zahajovací fáze, je důležité si uvědomit, že DAD je framework neboli rámec nikoliv pak normativní metoda, která by ukládala, jak má jít projekt krok za krokem. To má velkou výhodu v tom, že se dá upravovat a aplikovat na celou řadu projektů. Projektový tým si vždy může vybrat to, co se hodí na daný případ a nemusí nic doslovně následovat.

Typické pro fázi zahájení projektu jsou cíle, kterými je projekt řízen (viz kapitola goal driven). Kromě cílu, které se vážou pro tuto fázi, je nutné respektovat cíle, které provázejí celý projekt, vše je znázorněno na obrázku číslo 2. Cíly pro počáteční fázi jsou: Zformování počátečního týmu, Identifikace vize projektu, Souhlas stakeholders s vizí projektu, Vyrovnání se se zbytkem podniku, Identifikace počáteční technické strategie, požadavků a plánu uvolnění, Vytvoření pracovního prostředí, Stabilizovat financování a Identifikovat rizika.

Zároveň je nutné respektovat a dodržovat cíle, které nás provází celý projekt (viz následující kapitola). Těmi jsou: Naplňování poslání projektu, Zvyšování dovedností a

znalostí členů týmu, Obohatit stávající infrastrukturu, Zlepšovat týmové procesy a prostředí, Využívání stávající infrastruktury a Adresování rizik.

Goals for the Inception Phase	Goals for Construction Phase Iterations	Goals for the Transition Phase
<ul style="list-style-type: none"> - Form initial team - Identify the vision for the project - Bring stakeholders to agreement around the vision - Align with enterprise direction - Identify initial technical strategy, initial requirements, and release plan - Set up the work environment - Secure funding - Identify risks 	<ul style="list-style-type: none"> - Produce a potentially consumable solution - Address changing stakeholder needs - Move closer to deployable release - Maintain or improve upon existing levels of quality - Prove architecture early 	<ul style="list-style-type: none"> - Ensure the solution is production ready - Ensure the stakeholders are prepared to receive the solution - Deploy the solution into production
Ongoing Goals		
<ul style="list-style-type: none"> - Fulfill the project mission - Grow team members' skills - Enhance existing infrastructure - Improve team process and environment - Leverage existing infrastructure - Address risk 		

Obrázek 2: Cíle jednotlivých fází[1]

CÍLE FÁZE

• Zformování týmu

Týmy jsou při používání DAD obvykle malá až střední, tudíž od méně než deseti až do čtyřiceti členů. Zároveň se dá předpokládat, že čím větší tým bude, tím menší pravděpodobnost úspěšnosti projektu. Zároveň se doporučuje, aby pokud možno tým byl v jedné lokaci (ideálně v jedné místnosti). Za ucházející se dá považovat, když jsou všichni schopni se dopravit na koordinační schůzku a to včetně stakeholderů (zúčastněných stran). Co se týče doporučení pro tvorbu týmů, nabízí jich DAD celou řadu, jen ty nejdůležitější jsou: Zaměření týmu na výsledky a eliminace škodlivých prvků, Přizpůsobení týmu prostředí, Důvěra a respekt uvnitř i vně týmu, V týmu musí být bezpečnost (týká se především kritiky a nových poznatků), Poskytování možnosti se zdokonalovat a učit se, Jsou pokud možno co nejmenší, Mají sdílený workspace a mnoho dalších.

Důležitým poznatkem je, že tým se dotváří během projektu. Na začátku projektu je obvykle pouze vlastník produktu, team leader, architekt a možná pár členů týmu. Většina členů přijde během první iterace tvorby produktu a naopak v poslední fázi začíná odcházet.

• Identifikace projektové vize

Projektová vize musí odpovídat na pár otázek: Co se bude produkovat? Jak dlouho to bude trvat? Kolik to bude stát? Jakým způsobem se to bude tvořit? A

zda se rozumí rizikům? Na základě odpovědí na tyto otázky a informací k nim potřebným se sestaví projektová vize. Důležité je zmínit, že k projektové vizi se přistupuje evolučním přístupem, takže se může v průběhu projektu měnit a doplňovat.

- **Souhlas stakeholders s vizí projektu**

Základním účelem Počáteční fáze by mělo být dospění k dohodě se všemi zúčastněnými stranami (stakeholders) ohledně strategie pro zbývající část projektu. Tato část může být celkem snadnou záležitostí, ale i docela velkým oříškem.

- **Vyrovnaní se se zbytkem podniku**

Při tvorbě jakéhokoliv projektu je důležité vycházet dobře a nenarušovat zbytek podniku. Zároveň je třeba kooperovat s ostatními projektovými týmy. To je zvláště důležité, pokud je projekt závislý na jiných projektech či týmech. Zároveň není nutné, aby ostatní projekty a týmy pracovali v agilních metodikách.

- **Identifikace počátečních požadavků, technické strategie a plánu uvolnění**

Aby mohl být projekt úspěšný je nutné týmu sdělit, jaké jsou požadavky na řešení, a to a adekvátním stupněm podrobnosti. Proto je nutné nejdříve si ujasnit tyto požadavky. Zároveň je nutné týmu sdělit technickou strategii, tudíž je potřeba navrhnout základní architekturu a modely. Taktéž je potřeba vytvořit první plán uvolnění, to z toho důvodu, aby se dobře naplánovaly finanční požadavky a kapacity zdrojů. Na základě toho se bude určovat taktéž spolupráce na dosažení projektových cílů.

- **Vytvoření pracovního prostředí**

V případě, že je projekt odsouhlasen okolím podniku, je potřeba vytvořit pracovní prostředí pro tým. Tím může být samotné místo (kanceláře atd.), ale i HW a programové vybavení jako např.: CASE nástroje, IDE, stanice, testovací prostředí. Zároveň se nesmí zapomínat ani na vybrání metodiky, v tomto případě se samozřejmě bude jednat o DAD. Zároveň se dá předpokládat, že některé věci jsou k dispozici a jiné se musí zakoupit (či pronajmout). S tím může vznikat problém, kdy se dají už dokupovat potřebné věci. U velkých firem obvykle není problém a team leadři mají volnou ruku, hned jak dojde k odsouhlasení projektu, jinde se musí vše schvalovat a může docházet tak k prodlevám.

- **Zajištění financování**

Zde se jedná asi o klíčový cíl počáteční fáze. Úkolem je především zajistit dostatečné finance pro Konstrukční fázi projektu, která je nejnákladnější. Důležité je si stanovit způsob financování, což je ideální učinit, ještě před samotným spuštěním projektu. Obvyklé je, že firmy mají přesně nastaveny

pravidla financování, kterých se drží u všech projektů. Agilní metodiky (stejně tak i DAD) znají tyto druhy: Fixní cena, Etapová brána (Stage gate), Čas a materiály.

Fixní cena je odvozena od počátečního odhadu, který je určen obvykle na základě prvních modelů a určení architektury. Obvykle dochází k +/- 30% chybovosti odhadů.

Etapová brána (Stage gate) je založena na financování po částech projektu, přičemž každá část projektu dostane dohodnuté finanční prostředky na danou časovou periodu. Poté je potřeba přesvědčit o životaschopnosti a přidané hodnotě projektu a získat peníze na pokračování projektu. V DAD jsou přitom využívány milníky, po kterých je potřeba přesvědčit a získat další finance.

Čas a materiály je způsob, jenž klade velké nároky na vedení projektu. Tento způsob je často nazýván pay as you go (platím jak jdeš) a je považován za velmi riskantní. Proto je velmi často upravován, tak že materiální náklady jsou propláceny ihned, zatímco mzdové až na základě funkcionality SW. To rozkládá riziko mezi zákazníka a dodavatele, protože materiál zaplatí zákazník vždy (nevzniká škoda dodavateli) a pracovní čas pouze když vše pracuje ve stanoveném termínu. Tím je naopak chráněn zákazník.

V případě, že jsou naplněny všechny cíle počáteční fáze, může se přejít na další fázi, kterou je samotná Tvorba řešení.

PRŮBĚH FÁZE

Vše začíná milníkem Výběr projektu. To je způsobeno ideou autorů, kteří předpokládají, že firma má spravuje portfolio projektů skrze speciální procesy, které sami vybírají a identifikují potencionální projekty. Poté se je projekt předán do Počáteční fáze a musí být rozhodnuto, zdali se bude řešit nebo ne. V případě, že dojde ke schválení, může se začít s projektem.

Nejdříve ze všeho je potřeba si stanovit počáteční vizi a financování projektu (viz cíle výše). Poté se může přejít k stanovení počátečního plánu, modelování a organizaci celého projektu. Tento krok je nesmírně důležitý a má přímý vztah na iterativní vývoj v další fázi, kdy by mělo dojít k potvrzení toho, že byla vybrána vhodná architektura. Následuje stanovení počátečních požadavků a plánu uvolnění. Tato fáze končí souhlasem všech zúčastněných stran. Samotná fáze může proběhnout v jedné nebo i více malých iteracích.

TVORBA ŘEŠENÍ

Tvorba řešení či konstrukční fáze je postavena na principu ostatních agilních metodik, tudíž vše je vyvíjeno v iteracích (někdy označovány jako sprinty). Důležitý je

zde kontakt se zainteresovanými stranami (stakeholdery), ideálně aby s nimi tým pracoval při každé iteraci a konzultoval definované požadavky. Tím by měla být zajištěna co nejvyšší přidaná hodnota produktu.

Hlavním cílem této fáze je vytvořit řešení, které minimálně splňuje, ideálně přesahuje funkční nároky na něj kladené a přidává obchodní hodnotu podniku.

Jak už bylo několikrát řečeno, DAD je řízen cíly, takže i tato fáze má svoje cíle. Těmi jsou: Produkovat spotřební řešení, Reagovat na měnící požadavky zúčastněných stran, Přibližovat se k nasazení schopnému release, Udržovat, ideálně zlepšovat stávající úroveň kvality, Brzy ověřit vhodnost architekturu. Zároveň je nutné stejně jako u první fáze respektovat průběžné cíle.

CÍLE FÁZE

- **Produkovat potenciálně spotřební řešení**

Tento cíl vychází z toho, že je potřeba co nejdříve, již během první iterace vytvořit fungující řešení a to předložit zákazníkům. Ti řeknou svůj názor na něj a na základě toho vznikne zpětná vazba, jestli se tým ubírá správným směrem.

- **Reagovat na měnící požadavky zúčastněných stran**

Tento cíl je velmi ovlivněn Scrumem a vychází z jeho backlogu. Jde o to, že v každé iteraci může zákazník změnit své požadavky na tu část, která ještě není vytvořena. Zároveň však projektový tým si musí pohlídat, jestli je schopen tu změnu zvládnout udělat a zároveň nepřekročit stanovený čas.

- **Přibližovat se k nasazení schopnému release**

Ideálně by po každé iteraci mělo být vytvořeno takové řešení, které se dá nasadit a používat. To obvykle není možné, tudíž by měla být snaha, pokusit se alespoň vždy když je to možné, nasazovat řešení u zákazníka.

- **Udržovat, ideálně zlepšovat stávající úroveň kvality**

Tento cíl vychází z toho, že se jedná o agilní přístup, který je řízen testy (Test driven development). Díky tomu se dá i v případě neustále se zvětšujícího a rostoucího řešení udržet kvalitu na patřičné úrovni.

- **Brzy ověřit vhodnost architekturu**

Tento cíl vychází z faktu, že vadná architektura má za následek poruchovost řešení a celého projektu. Proto DAD razí přístup, aby požadavky, které se nejvíce dotýkají architektury, byly řešeny jako první, a tím bylo případně prokázán vhodný popřípadě špatný výběr architektury.

- **Naplňování poslání projektu**

V průběhu tvorby je potřeba neustále kontrolovat, jestli se jde správným směrem a dochází k naplňování projektové vize, která byla stanovena v první fázi. Tuto kontrolu by měly provádět především zúčastněné strany. Zároveň je důležité seznamovat s vizí i nové členy týmu, aby bylo jednotné vidění projektu.

- **Zvyšování dovedností a znalostí členů týmu**

Je potřeba si uvědomovat, že veškerou přidanou hodnotu SW tvoří pouze lidé. Proto je potřeba mít kvalifikované a špičkové pracovníky, kteří vědí o dané problematice co nejvíce. Doporučuje se taktéž párování členů týmu s různou úrovní dovedností a přibrání do týmu lidí, kteří mají již nějakou zkušenost s agilním vývojem.

- **Obohatit stávající infrastrukturu**

V případě možnosti použití již vytvořených vzorů či čehokoliv, jež mohou být znovu používány je doporučováno jejich využití.

- **Zlepšovat týmové procesy a prostředí**

Chce-li být agilní tým opravdu efektivní, musí neustále hledat možnosti a příležitosti pro svoje zlepšení, tím i svých procesů a prostředí. Zároveň by měl být tým zmocněn, aby mohl procesy a prostředí sám změnit, samozřejmě pouze do určité míry.

- **Využívání stávající infrastruktury**

Vždy je si potřeba zjistit co tým už vlastní a může být použito i pro další projekty, případně je použít. Jedná se především o normy, vzory, mechanismy ale i kódy atd.

- **Adresování rizik**

Jelikož se jedná o životní cyklus řízený riziky (risk-value lifecycle), je vždy nutno odhalovat všechny potenciální riziky a přijmout opatření k jejich eliminaci. Vždy by k tomu mělo docházet co nejdříve je to možné.

PRŮBĚH KONSTRUKČNÍ FÁZE

Tato fáze začíná s koncem fáze předchozí (Počáteční) a souhlasem zúčastněných stran. Nejdříve je nutné rozdělit projekt na tzv. pracovní položky (work items) a ty seřadit dle priorit. Pro to se doporučuje metoda poker plánování, kdy pomocí karetního stanovení se určí ty nejdůležitější. Ty se budou tvořit nejdříve. Poté se připraví backlog a probíhá samotná iterace, která je velmi podobná Scrumu. Každý den probíhá meeting, kdy se určí práce pro daný den. Autoři zmiňují, že se může použít i jiný způsob, např.: koordinační meetingy (Kanban), týdenní status meeting či žádný plánovaný meeting, ale doporučují využít způsob na základě Scrumu.

Na konci iterace je předložena vytvořená práce zúčastněným stranám, které poskytnou zpětnou vazbu. Na základě toho se naplánuje strategie pro další iterace. Obvykle při používání DAD zde také dochází k získávání financí na další vývoj, to z toho důvodu, že je používán jako způsob financování tzv. etapová brána (viz výše). Iterací probíhá obvykle větší množství. V této fázi jsou následující milníky: Osvědčená architektura a Životaschopnost projektu (tento milník se v rámci této fáze objevuje několikrát).

PŘECHODOVÁ FÁZE

V této fázi dochází k nasazení řešení zákazníkovi. I tato fáze se může skládat z několika iterací, přičemž by měly být pokud možno co nejkratší, přičemž rozhodně kratší než iterace konstrukční fáze.

I tato fáze má své cíle, které je nutno naplnit, těmi jsou: Ujistění se o připravenosti řešení k produkci, Ujistění se o připravenosti zúčastněných stran k přijetí řešení, Nasazení řešení do produkce. Opět je nutno respektovat a naplňovat průběžné cíle.

CÍLE FÁZE

- **Ujistění se o připravenosti řešení k produkci**

Tento cíl se skládá z několika dílčích částí. Nejprve je nutné naplánovat přechod, to se děje skrze detailní implementační plán. Následně musí být provedeno testování a případně opraveny všechny chyby, přičemž chyby by se vyskytovat už neměly, ale to neznamená, že být nemohou. Dále je nutné připravit data pro migrace. To se týká samozřejmě pouze v případě, že se nahrazuje starý systém novým. Následně se může spustit beta provoz (pilotní). Díky tomu by se mělo podařit odladit drobné nedostatky popřípadě nové funkční požadavky. Jako poslední krok je potřeba upravit dokumentaci do finální podoby.

- **Ujistění se o připravenosti zúčastněných stran k přijetí řešení**

Před nasazením řešení je nutno připravit uživatele na jeho spuštění. Tento proces začíná na začátku projektu, kdy je neustále komunikováno se všemi zúčastněnými stranami. S postupujícím projektem by se mělo více komunikovat o samotných detailech nasazení produktu, přičemž by se měly probrat především tyto oblasti: Provoz, Marketing, Změně organizační struktury, Regulacích, Sponzorech projektu, Změně vedení, Správních orgánech, Školení uživatelů, Podpory a help desku. Samozřejmostí je, že vždy záleží na povaze projektu a podle toho je potřeba zvolit ty oblasti, kterých se změna bude týkat.

- **Nasazení řešení do produkce**

V průběhu této fáze musí dojít k tomuto cíli. Samotné nasazení může probíhat různými způsoby, vše je určeno podle povahy a velikosti podniku. Většinou se samotné nasazení řídí implementačním plánem. Pro naplnění tohoto cíle je potřeba zahrnout tyto aktivity: Spuštění dávkových úloh oproti starému systému, Zazálohování stávajícího systému, Migrace zdrojových dat, Nasazení všech artefaktů na nové řešení, Provést zaváděcí testování, Zazálohovat nové řešení, Zpřístupnit nové řešení zákazníkům (uživatelům), Spustit podporu, Informovat o nošeném/neúspěšném nasazení.

PRŮBĚH PŘECHODNÉ FÁZE

Tato fáze začíná milníkem, kterým končí fáze předchozí, kterým je splnění požadované funkcionality. Když je vyvinuto takové řešení, jež splňuje všechny požadavky, vstupuje životní cyklus projektu do závěrečné fáze.

Nejprve dojde k uvolnění řešení do produkce a poté k jeho podpoře a vývoji v provozu. Zde je vazba na předchozí fázi, kdy v případě chyb nebo žádostí o vylepšení se vrací životní cyklus zpět. I tato fáze může probíhat ve více iteracích a jsou v ní následující milníky: Připravená produkce a Spokojené zúčastněné strany.

ŠKOLENÍ

Přestože se jedná o relativně novou metodiku, již jsou o ní pořádány kurzy a školení. Nejvíce jich pořádá firma Scotta W. Amblera, Scott Ambler + Associates, což je způsobeno tím, že je autorem této metodiky. Poskytuje celkem 10 druhů školení od úrovně pro začátečníky (3 školení), přes středně pokročilé uživatele (5 školení) až po pokročilé (2 školení). Většinou se jedná o jednodenní školení (7 hodin) až po třídenní (21 hodin). [5]

Další firmou, která poskytuje školení je IBM. Ta v rámci lokalizace České republiky nabízí dva kurzy, a to Introduction to Disciplined Agile Delivery V1.0, kdy se jedná o základní kurz. Školení probíhá ve virtuální třídě a trvá 16 hodin. Jeho cena 1 320\$. [6] Druhým poskytovaným kurzem je Advanced Disciplined Agile Delivery. Zde se jedná o pokročilý kurz trvající tři dny. Školení probíhá v učebně formou cvičení. [7]

ZÁVĚR

Jedná se o novou metodiku, která působí dojmem, že autoři při její tvorbě využili všechny dosavadní agilní metodiky, ty zkombinovali a vytvořili novou. To se odráží i na její používanosti, jelikož se nám nepodařilo najít žádné statistiky, ale ani případové studie jejího použití (vyjma té uvedené v knize).

Největší devízou DAD by mělo být právě to, že obsahuje best practices ostatních metodik. Autoři zároveň velmi sázejí na možnost jejího škálování, kterou prozatím pouze naznačili a budou se jí teprve věnovat. Co se týče jejího praktického použití a výsledků z něho, na to si budeme muset ještě počkat.

ZDROJE

- 1 AMBLER, Scott W a Mark LINES. Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise. 1. vydání. Upper Saddle River, NJ: IBM Press, 2012, xxvii, 513 p. ISBN 978-013-2810-135.
- 2 Agile Practices Survey Results: July 2009. AMBYSOFT INC. Ambysoft: Exxective Practices for Software Solution Delivery [online]. 2009 [cit. 2012-12-12]. Dostupné z: <http://www.ambysoft.com/surveys/practices2009.html>
- 3 Techniques for Disciplined Agile Delivery: Interview with Scott Ambler. INFOQ. *Facilitating the spread of knowledge and innovation in enterprise software development* [online]. 2012 [cit. 2012-10-25]. Dostupné z: <http://www.infoq.com/interviews/scott-ambler-DAD>
- 4 ZIKMUND, Jan. Škálování agilních metodik. 2012. Dostupné z: <https://isis.vse.cz/auth/lide/clovek.pl?id=79489;zalozka=7>
- 5 Training: Scott Ambler + Associates. SCOTT AMBLER + ASSOCIATES. Scott Ambler + Associates [online]. [cit. 2012-12-12]. Dostupné z: <http://scottwambler.wordpress.com/courses/>
- 6 IBM popis kurzu: Introduction to Disciplined Agile Delivery V1.0. IBM. IBM: Česká republika [online]. [cit. 2012-12-12]. Dostupné z: http://www-304.ibm.com/jct03001c/services/learning/ites.wss/cz/cs?pageType=course_description&courseCode=RPV51
- 7 IBM popis kurzu: Advanced Disciplined Agile Delivery. IBM. IBM: Česká republika [online]. [cit.2012-12-12]. Dostupné z: http://www-304.ibm.com/jct03001c/services/learning/ites.wss/cz/cs?pageType=course_description&courseCode=RP252CZ