

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Studijní obor: Informační systémy a technologie

Předmět: 4IT421 Zlepšování procesů budování IS



Název seminární práce:

## **Metodika Disciplined Agile Delivery - charakteristika, současný stav**

Autoři:

Tomáš Hoch – xhoct02

Adam Holub - xhola00

## Obsah

|     |                                                 |    |
|-----|-------------------------------------------------|----|
| 1   | Úvod .....                                      | 3  |
| 2   | Základní popis metodiky.....                    | 3  |
| 3   | Popis životního cyklu metodiky DAD.....         | 6  |
| 4   | Detailní životní cyklus řešení .....            | 7  |
| 5   | Životní cyklus řízený cíli .....                | 10 |
| 6   | Řízení životního cyklu riziky a hodnotami ..... | 12 |
| 7   | Spolupráce v rámci podniku .....                | 12 |
| 8   | Škálování.....                                  | 13 |
| 9   | Současný stav .....                             | 14 |
| 9.1 | Adaptace agilních metodik.....                  | 14 |
| 9.2 | Výuka .....                                     | 15 |
| 10  | Závěr .....                                     | 16 |
| 11  | Zdroje.....                                     | 17 |

## 1 Úvod

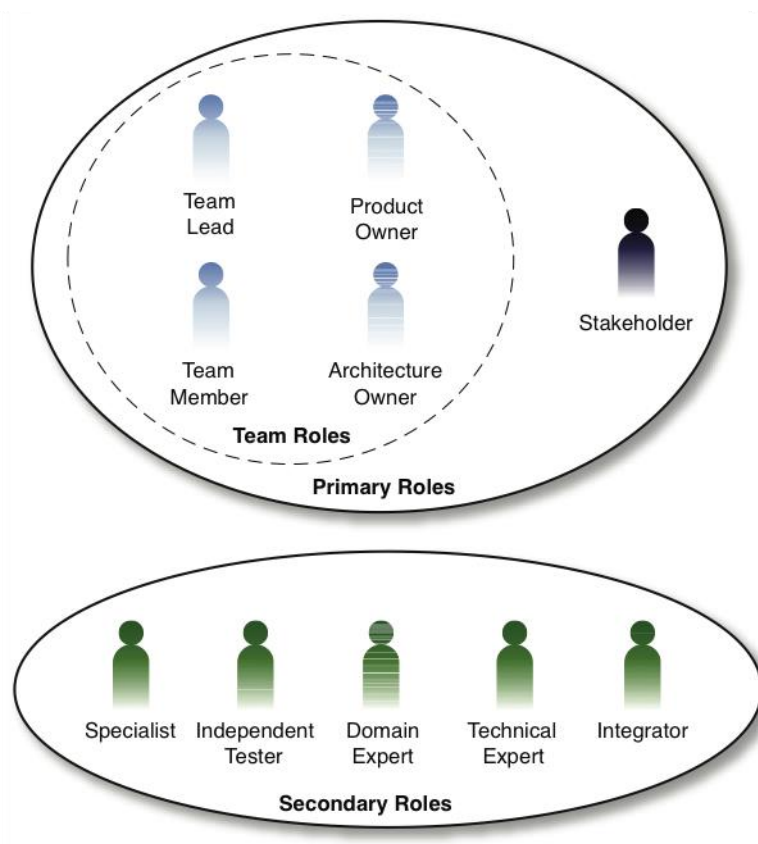
Cílem této práce je charakterizovat metodiku Disciplined Agile Delivery, popsat její životní cyklus, podrobněji se podívat na jednotlivé týmové role, představit její hlavní myšlenky a přístupy a rozebrat její možnosti škálování pro konkrétní týmy nacházející se v různých situacích a prostředích. V závěrečné části je pak popsán současný stav metodiky.

## 2 Základní popis metodiky

Disciplined Agile Delivery (DAD) je metodika poskytující postupy pro dodání IT řešení zákazníkovi (typicky dodání nějakého SW). Tvůrcem metodiky je Scott W. Ambler, který na metodice pracuje od roku 2007, poté co začal pracovat jako vedoucí metodolog pro IT v IBM. Metodika je ale vyvíjena nezávisle na IBM a je volně dostupná.

Důvodem vzniku metodiky byly nedostatky metodik stávajících. Typickými problémy agilních metodik je nedostatečná škálovatelnost, mnoho takových metodik se nedá použít pro velké týmy a projekty, a také to, že nezahrnují celý životní cyklus vývoje softwaru. Dle S. Amblera bylo před vznikem této metodiky nutné, si pro každý projekt nejprve vytvořit metodiku vlastní, spojením jednotlivých metodik. Podobným způsobem vzniká i metodika DAD. Vychází z metodiky Scrum, který rozšiřuje o prvky z dalších metodik (Agile Modeling, Extreme programming, Unified Proces). Výsledkem je tedy hybridní agilní metodika, definující vlastní terminologii, principy, role a životní cyklus dodávaného řešení, kde jednotlivé prvky vznikaly hlavně sjednocením nebo výběrem z nejlepších prvků různých metodik. [1]

Role definované metodikou Disciplined Agile Delivery [2]



The roles of Disciplined Agile Delivery (DAD)

Obrázek 1 - The roles of Disciplined Agile Delivery (DAD)

Na výše uvedeném obrázku je vidět, že metodika DAD definuje deset rolí, které jsou rozděleny na primární a sekundární. Takto vysoký počet je dán komplexitou metodiky. Vzhledem k tomu, že metodika DAD se zabývá celým životním cyklem řešení, včetně například technických aspektů, musí být všechny tyto oblasti pokryty rolími.

Není ale nutné, aby všechny uvedené role byly v projektu obsazeny, primární role budou v typickém projektu obsazeny vždy, pro sekundární role platí, že v projektu často budou jen velmi krátce, nebo vůbec.

Metodika zároveň dovoluje, aby jedna role byla obsazena více osobami, jedna osoba měla více rolí současně a osoby měnily role v průběhu projektu.

Mezi primární role patří:

### **Stakeholder (Investor, Sponzor)**

Stakeholder je osoba materiálně spojená s výsledkem řešení. Překlad Investor, nebo Sponzor je zde trochu zavádějící, protože se může jednat o koncového uživatele, manažera uživatelů, pracovníka podpory nebo provozu, auditora, nebo kohokoli dalšího kdo bude nějak ovlivněn řešením.

Spolupráce se stakeholderem by měla probíhat na denní bázi.

### **Team Member (Člen týmu)**

Team Member je koncový pracovník, který tvoří výsledné řešení. Členové týmu provádějí činnosti jako analýza, vývoj, design, testování, plánování a další. Není nutné, aby každý člen týmu prováděl všechny tyto činnosti, ale je vhodné, aby byl schopen provádět několik z nich.

### **Team Lead (Vedoucí týmu)**

Metodika DAD záměrně nepoužívá pojem manažer. Manažer je vnímán jako někdo, kdo řídí tým, přiděluje úkoly, plánuje, odhaduje, kontroluje. Metodika používá roli leadera jako někoho, kdo vede tým ve smyslu vedení týmu k cíli. Team Lead je někdo, kdo slouží týmu k tomu, aby pro něj vytvářel co nejvhodnější podmínky. Team Lead usnadňuje práci členům týmu, odstraňuje překážky, motivuje, udržuje tým zaměřený na naplnění cíle.

### **Product Owner (Vlastník)**

Product Owner funguje jako spojení mezi Stakeholdery a pracovními týmy. Je zodpovědný za prioritizaci požadavků. Je to osoba, které se jednotlivé týmy ptají na nejasnosti a řeší s ním problémy spojené se zadáním, přičemž má právo o konečné podobě požadavku přímo rozhodovat. Aktivně se účastní většiny fází projektu. Zajišťuje finance. Osvětluje pracovním týmům byznysovou stránku projektu. Prezентuje stakeholderům řešení a to jak částečné, v průběhu projektu, tak i finální.

Product Owner by měl být někdo s výbornými komunikačními a vyjednávacími schopnostmi, měl by vědět, kdo jsou jednotliví stakeholderi, a komunikovat by měl hlavně osobně, nikoliv přes emaily a dokumenty. [3]

### **Architecture Owner (Architekt)**

Architekt dohlíží a je zodpovědný za architektonické aspekty dodávaného řešení a dále za to, že řešení je vhodným způsobem testované a bude snadno podporovatelné. Je zdrojem informací týkajících se architektury pro pracovní týmy. Architekt by měl znát běžně používané návrhové vzory a praktiky a také standardy organizace, ve které pracuje a měl by dohlížet na jejich používání. Typicky se bude jednat o zkušeného (seniorního) developera. V malých projektech může být architektem Team Leader. [4]

Mezi sekundární role patří:

### **Specialist (Specialista)**

V případě, že nějaká součást řešení je příliš komplexní, a tým se s touto součástí není schopen vypořádat sám, je vhodné využít v týmu alespoň dočasně specialistu na danou oblast. Využití specialistů se také doporučuje v případě, že pro tým je agilní vývoj něčím novým.

### **Domain Expert (Expert na určitou oblast)**

V případě že součást řešení vyžaduje nějaké konkrétní znalosti, kterými Product Owner nedisponuje, je vhodné připojit k týmu experta na danou oblast, který týmu s řešením pomůže. Například při vývoji účetního systému je vhodné přizvat do týmu experta na účetnictví.

### **Technical Expert (Technický expert)**

Analogicky k předchozím dvěma rolím je Technical expert využíván v případě kdy jsou potřeba expertní znalosti z oblasti bezpečnosti, databáze, integrace, nasazení a podobně.

### **Independent Tester (nezávislý tester)**

Nezávislý tester je využíván jako pomocná síla v případě, že pracovní tým nemá pro testování dostatečné kapacity a zkušenosti, nebo pokud testování vyžaduje nějakou speciální technologii.

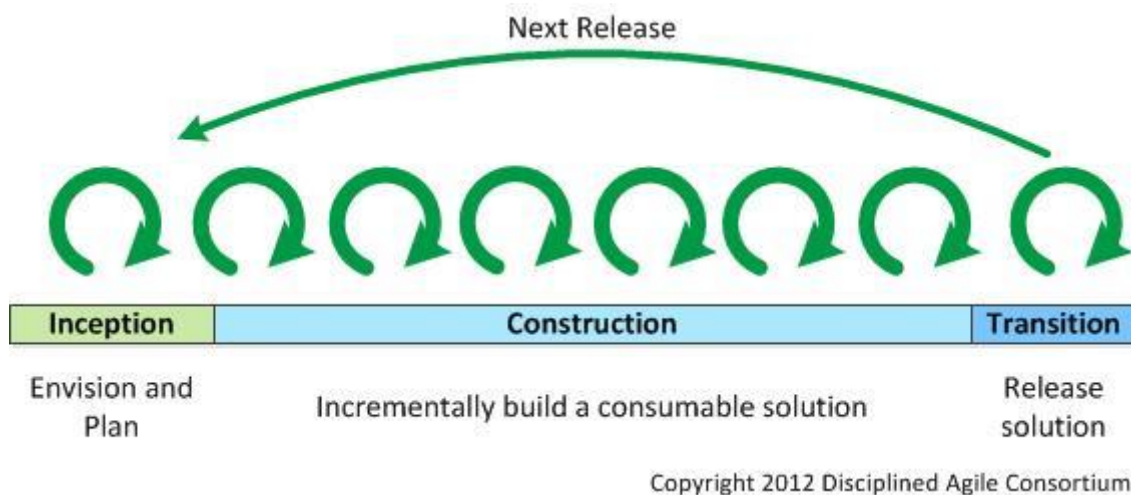
### **Integrator (Integrátor)**

V případě velkých projektů, kdy je pracovní tým rozdělen na jednotlivé subtýmy, které dodávají jednotlivé části řešení, je integrátor zodpovědný za integraci těchto jednotlivých částí do fungujícího systému. Integrátor často spolupracuje s testovacími týmy a zajišťuje integrační testy systému.

### 3 Popis životního cyklu metodiky DAD

Klíčovým aspektem metodiky DAD je životní cyklus řešení. Metodika popisuje nejprve obecný životní cyklus, a následně z něj vycházející detailní životní cykly[5]

Obecný životní cyklus řešení



Obrázek 2 - high-level view of the DAD lifecycle

Jak ne vidět z obrázku, obecný životní cyklus řešení obsahuje tři fáze.

#### Inception (Zahájení)

V zahajovací fázi jsou prováděny činnosti nezbytné pro zahájení projektu. Cílem této fáze je vyjasnění problému který je potřeba řešit, nalezení technicky možného řešení, stanovení přístupu jakým bude problém řešen, sestavení pracovního týmu[6].

Metodika DAD doporučuje zajistit, aby tato úvodní fáze netrvala déle než pár týdnů.

#### Construction (Konstrukce)

Během této fáze pracovní týmy tvoří části řešení. Práce probíhá typicky v iteracích, na jejich konci jsou dodávány jednotlivé přírůstky řešení.

#### Transition (Přechodová fáze)

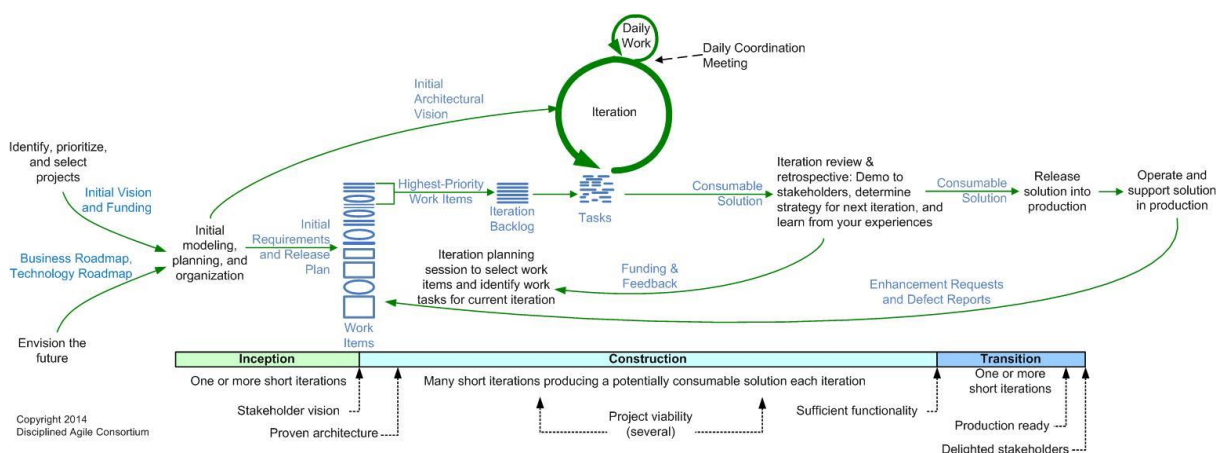
V této fázi se dodává hotové řešení stakeholderům. Řešení musí být připravené k dodání, tedy musí splňovat požadavky stakeholderů a musí být dodáno v požadované kvalitě. Zároveň stakeholdeři musí být připraveni řešení přijmout, tedy musí dopředu vědět, že dojde k nasazení řešení a musí být odpovídajícím způsobem proškoleni. V případě aplikování principu nepřetržitého dodávání lze tuto fázi vynechat. [7]

## 4 Detailní životní cyklus řešení

Existuje několik důvodů proč metodika DAD podporuje více životních cyklů. Jako hlavní důvod je uváděno to, že každý projekt je jedinečný a jeden životní cyklus tedy nemůže pasovat na všechny. Zadruhé, metodika DAD se snaží vyhnout tomu, aby byla příliš preskriptivní, mnoho týmů se dostalo do problémů jen díky tomu, že se snažili až příliš přesně následovat nějakou konkrétní metodiku.

Zatřetí, pracovní týmy se vyvíjejí, proto je vhodné jim nabídnout nejprve jednodušší verzi životního cyklu a zároveň jim umožnit později přejít na složitější.

### The Agile/Basic Lifecycle: Extending Scrum (základní životní cyklus)

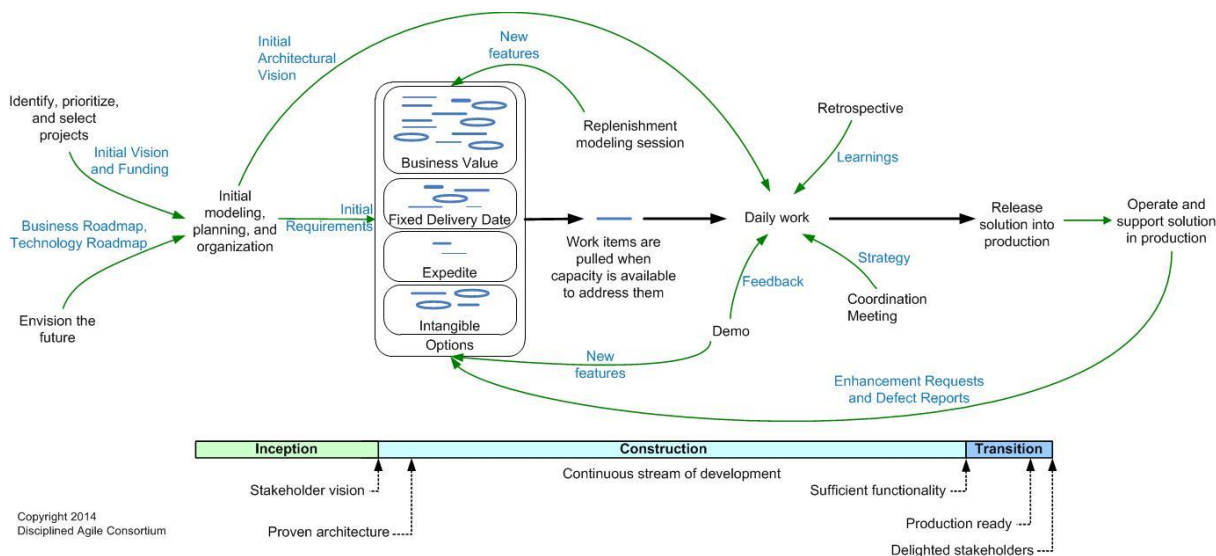


Obrázek 3 - The Agile/Basic Lifecycle: Extending Scrum

Výše uvedený obrázek zobrazuje základní životní cyklus metodiky DAD. Měly by ho použít týmy, které s metodikou začínají. Vychází z životního cyklu Scrumu, který rozšiřuje o aspekty specifické pro metodiku DAD:

- Je založený na iteracích, podobně jako Scrum.
- Používá vlastní terminologii. Terminologie Scrumu byla nahrazena, ale na terminologii nijak výrazně nezáleží a pokud by někdo chtěl používat terminologii Scrumu, je to možné
- Obsahuje vnější vstupy. Jako součást životního cyklu jsou zde uvedeny i činnosti, které se dějí před zahájením projektu, před jeho první fází, a také po jeho skončení. Také jsou zde zobrazeny nové úkoly, které mohou přijít z provozu jako chyby nebo změnové požadavky.
- Obsahuje seznam pracovních úloh (work item list). Na rozdíl od backlogu použitého v metodice Scrum, který obsahuje hlavně požadovanou funkcionalitu, jsou v seznamu pracovních položek obsaženy také defekty, změnové požadavky, školení, ale i volno.
- Obsahuje milníky

## The Advanced/Lean DAD Lifecycle (pokročilý životní cyklus)

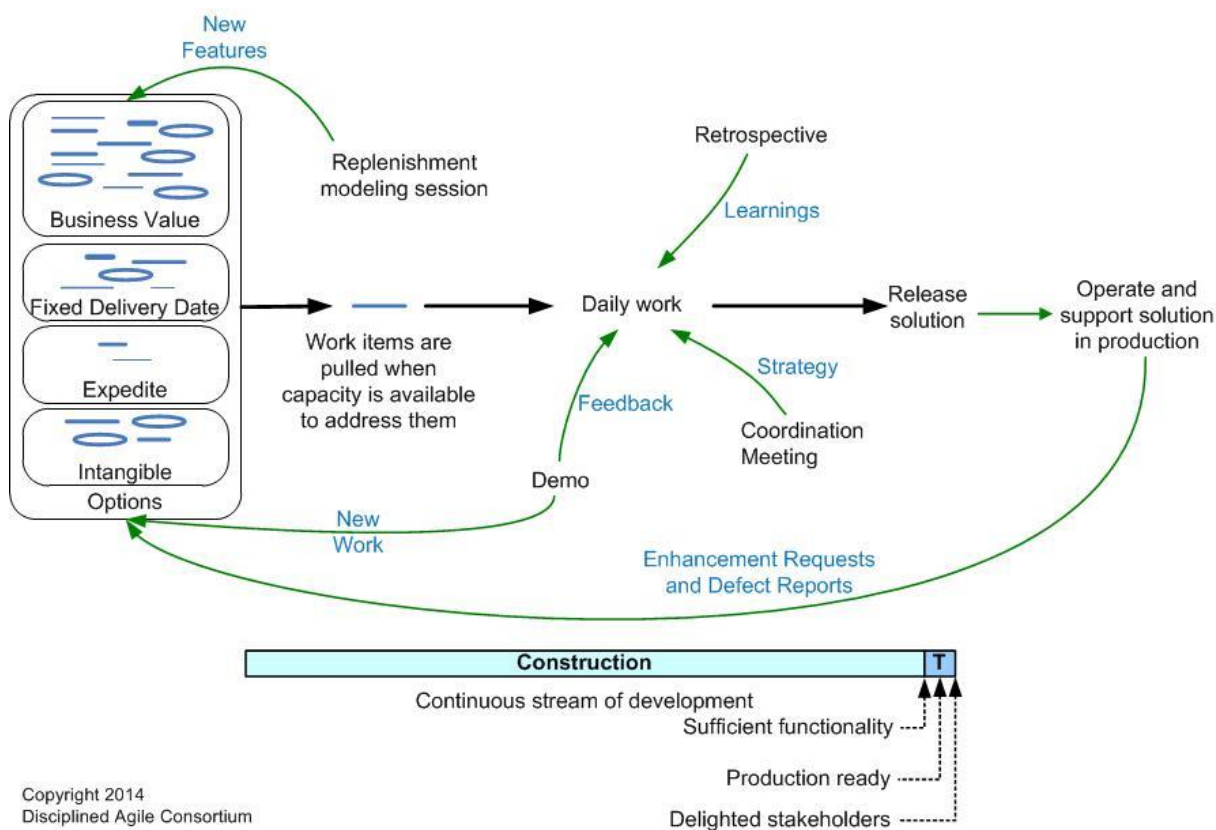


Obrázek 4 - The Advanced/Lean DAD Lifecycle

Výše uvedený obrázek zobrazuje pokročilý životní cyklus metodiky DAD. Pokročilým je nazýván proto, že týmy typicky začnou pracovat dle základního životního cyklu, který postupně optimalizují a transformují do podoby tohoto pokročilého. Rysy tohoto životního cyklu jsou následující:

- Podporuje kontinuální dodávání. Nové řešení je nasazeno kdykoliv to má smysl. Tým začne pracovat na úkolu hned, jakmile má volné kapacity, již ne na začátku iterace.
- Postupy mají vlastní rytmus. Jednotlivé činnosti jsou prováděny, když to má smysl, ne tak jak jsou naplánovány.
- Obsahuje fond pracovních úloh (work item pool). Fond pracovních úloh umožňuje třídit a organizovat úkoly, které je potřeba zpracovat.

## The Continuous Delivery DAD Lifecycle (Životní cyklus při nepřetržitém dodávání)



Obrázek 5 - The Continuous Delivery DAD Lifecycle

Zde je pro doplnění také obrázek s životním cyklem při nepřetržitém dodávání. Jedná se v podstatě o zkrácenou verzi předchozího životního cyklu, s tím, že typickým rysem je pravidelné a časté dodávání nových verzí řešení, například každý týden, nebo i každý den.

## 5 Životní cyklus řízený cíli

Mnoho týmu má problém se zlepšením svých procesů vývoje softwaru. Buď vycházejí z detailně popsaných procesu rigorózních metodik, jako například RUP, nebo z agilních metodiky definující spíše jen postupy, praktiky a procesy popisují jen okrajově. Problém spočívá v tom, že nedokáží tyto výchozí metodiky zeštíhlit, případně tedy posílit, tak, aby to přineslo požadovaný efekt. Navíc úsilí vynaložené na přizpůsobení metodiky je stojí další náklady. Na jedné straně je potřeba zamezit příliš podrobnému popisu procesu a poskytnout tak dostatečnou flexibilitu a možnost škálování. Na straně druhé je pak potřeba poskytnout dostatečná vodítka přesahující rozsah agilních metodik, aby nedošlo k opomenutí klíčových záležitostí v procesu vývoje softwaru. Alternativou mezi těmito dvěma extrémy je právě metodika DAD, která řeší tento problém zavedením termínu „Goal-Driven“ neboli řízení pomocí cílů.

| Goals for the Inception Phase                                                                                                                                                                                                                                                                                                                                          | Goals for Construction Phase Iterations                                                                                                                                                                                                                 | Goals for the Transition Phase                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>- Form initial team</li> <li>- Develop common project vision</li> <li>- Align with enterprise direction</li> <li>- Explore initial scope</li> <li>- Identify initial technical strategy</li> <li>- Develop initial release plan</li> <li>- Form work environment</li> <li>- Secure funding</li> <li>- Identify risks</li> </ul> | <ul style="list-style-type: none"> <li>- Produce a potentially consumable solution</li> <li>- Address changing stakeholder needs</li> <li>- Move closer to deployable release</li> <li>- Improve quality</li> <li>- Prove architecture early</li> </ul> | <ul style="list-style-type: none"> <li>- Ensure the solution is consumable</li> <li>- Deploy the solution</li> </ul> |
| <b>Ongoing Goals</b> <ul style="list-style-type: none"> <li>- Fulfill the project mission</li> <li>- Grow team members</li> <li>- Address risk</li> <li>- Improve team process and environment</li> <li>- Leverage and enhance existing infrastructure</li> <li>- Coordinate activities</li> </ul>                                                                     |                                                                                                                                                                                                                                                         |                                                                                                                      |

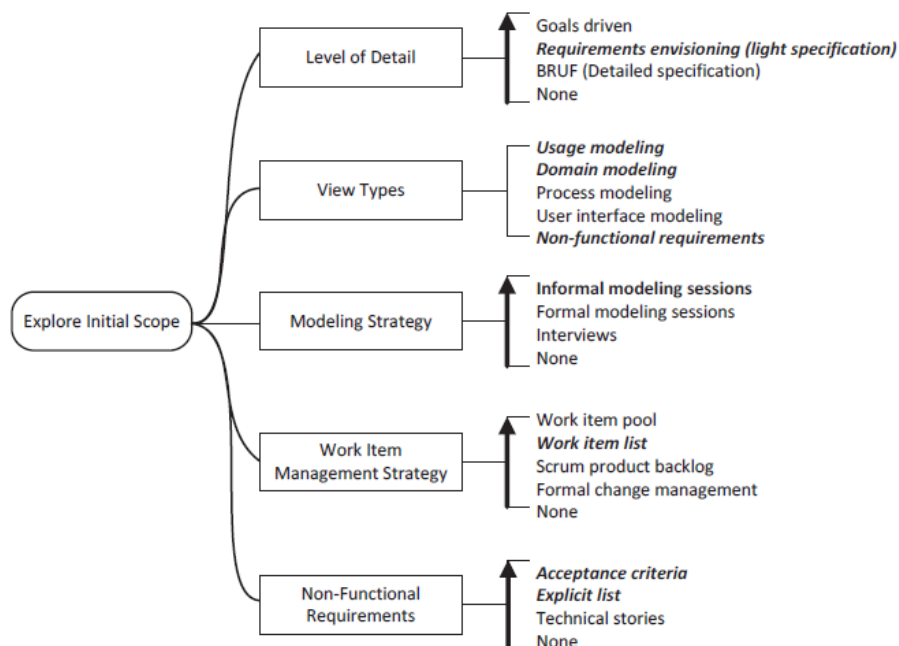
Obrázek 6 - Tabulka s identifikovanými cíli pro jednotlivé fáze[8]

Předchozí obrázek ukazuje, jak může vypadat přehled identifikovaných cílů pro jednotlivé fáze, včetně cílů společných pro celý životní cyklus. Tento přehled obsahující výčet cílů, je navíc podle DAD konzistentní se všemi projekty bez ohledu na jejich typ. Ovšem samotné takto identifikované cíle by pro zajištění úspěšného dokončení projektu měli pramalý význam. DAD proto navrhuje každý cíl ještě rozpadnout na několik dílčích otázek, jejichž vyřešením zvýšíme pravděpodobnost, že cíl bude úspěšně splněn. Jinými slovy, sníží se tím riziko nesplnění cíle splnění cíle.

Rozpad cíle na dílčí otázky demonstruje následující obrázek. Na něm vidíme jeden z cílů z přechodí tabulky „Explore Initial Scope“, tedy cíl provést nějakou úvodní studii. Aby byla úvodní studie úspěšná DAD navrhuje provést i stanoví detailu specifikace, rozhodnout typech použitých modelu, o strategii s jakou budeme modelování provádět atd.

Pro vyřešení dílčích otázek DAD opět navrhuje možnosti, který by danou otázku řešili. Například v otázce míře detailu specifikace navrhuje, že detail může být pouze na úrovni řízení prostřednictvím cílů, na úrovni základního vymezení požadavků, detailní specifikace anebo žádná specifikace. Řešení zapsané kurzívou a tučně zvýrazněné označují tu nejlepší volbu pro typicky „Disciplined Agile Delivery“ projekt. Některé seznamy navíc obsahují i šipku, která indikuje, které řešení jsou podle DAD nejvíce typické a efektivní. Ovšem jedná se pouze o doporučení vycházející z dlouhodobých zkušeností tvůrců DAD nic není závazné. Samotná volba je již zcela na řešitelském týmu. Ten si navíc může celý „goal

diagram“ doplnit o vlastní otázky či přidat další možnosti řešení. Díky tomu si tak tým může metodiku přizpůsobit zcela svým potřebám a konkrétní situaci, ve které se nachází.



Obrázek 7 - Goal diagram [9]

Kromě přístupu doporučování metodika také zavádí strategii „goal tables“, která má pomoci u každého řešení zvážit všechny výhody i nevýhody a vybrat tak to nejlepší pro konkrétní situaci. Příklad „goal table“ ukazuje následující obrázek.

| Strategy         | Potential Advantages                                                                                          | Potential Disadvantages                                                  | Considerations                                                                                                                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| More than annual | Appropriate for low priority systems or for high-risk deployments (e.g. embedded software)                    | Very risky<br>Requires internal releases to obtain some feedback         | This is common for infrastructure systems, such as a database or transaction managers, that have many other systems highly dependent upon them                                                                                                                           |
| Annual           | Appropriate for low-to-medium priority systems or medium-to-high risk deployments                             | Requires internal releases                                               | Don't include the year in the name of the release, for example ProductX 2014, if the ongoing release cadence is going to change.                                                                                                                                         |
| Bi-annual        | Good starting point for agile teams because it motivates adoption of disciplined strategies                   | Can be difficult for stakeholders who are used to less frequent releases |                                                                                                                                                                                                                                                                          |
| Quarterly        | Enables simpler requirements management practices due to lower impact of a feature moving to the next release | Requires disciplined continuous integration (CI) strategies              | CI refers to the practice of regularly, at least daily, building/compiling and regression testing your solution. CI is described in Chapter 15. This is a major milestone for teams moving towards an "advanced" lean-agile strategy as it motivates greater discipline. |

Obrázek 8 - Goal table [8]

## 6 Řízení životního cyklu riziky a hodnotami

Snahou DAD je identifikovat všechny možná rizika již v rané fázi životního cyklu. Zastává totiž filozofii „zaútočte na rizika, než ony zaútočí na vás“ [8]. „Risk-value driven lifecycle“, neboli Řízení životního cyklu riziky a hodnotami má za úkol tuto filozofii naplnit. Jedná se vlastně o rozšíření přístupu „value-driven lifecycle“ z metodiky Scrum a XP. Ovšem zatímco výstupem každé iterace ve Scrumu je potenciálně použitelný software, tak z pohledu DAD by výstupem každé iterace mělo být potenciálně použitelné řešení. Tedy, jak už bylo naznačeno v přecházejících částech, jedná se řešení, které zákazník může ihned zařadit do provozu. Dále ve Scrumu jsou funkce v zásobníku požadavků řazeny pouze podle hodnoty, kterou jejich zhotovení přinese. DAD kromě hodnot požadavků bere v úvahu i jejich rizika a rizikovým požadavkům dává nejvyšší prioritu[12]. Tím nejenže identifikuje všechny potencionálně rizikové požadavky, ale zároveň zajistí jejich přednostní zpracování, díky čemuž lze v brzkých fázích projektu odhalit třeba nemožnost jeho realizace, respektive realizace zadaných požadavků.

Klasické agilní metodiky přechází určitým rizikům pomocí dodávání řešení a předvedení výsledku práce zákazníkovi na konci každé iterace nebo zapojení zákazníka (či jeho zástupce) po celou dobu vývoje. Metodika DAD tyto základní přístupy rozšiřuje a navíc zavádí i další způsob snížení rizik a to prostřednictvím pevně stanovených milníků v průběhu celého životního cyklu. Stanovené milníky jsou:

- **Shoda zájmových skupin.** Zařazen na konec fáze Zahájení. Jeho úkolem je zajistit, že zájmové skupiny došly k rozumné shodě s tím, co bude dodáno jako výsledné řešení. Sníží se tím tak riziko dodání nechtěné funkcionality nebo nechtěného řešení a zároveň se investovalo jen velmi málo prostředků, takže konečné zamítnutí projektu nebude mít silný dopad.
- **Ověření architektury.** Milník pro ranou část Konstrukční fáze. Snahou je zajistit, že namodelovaná architektura bude doopravdy fungovat. Podle DAD, jediný způsob, jak se opravdu ujistit správnosti architektury, je ověření pomocí fungujícího kódu.
- **Další životaschopnost projektu.** Během konstrukční fáze, v malých časových obdobích, je nutné se na základě prozatímních výsledků neustále přesvědčovat, že projekt je stále životaschopný a může pokračovat.
- **Dostatečná funkcionality.** Na konci konstrukční fáze je nutné se přesvědčit, zda bylo dosaženo požadované funkcionality a řešení může být postoupená do Přechodové fáze
- **Produkce připravena.** Ke konci Přechodové fáze je nutné rozhodnout, zdali by dodávané řešení mělo být nasazeno na produkci. Zájmové skupiny se rozhodují, zda můžou dodané řešení akceptovat nebo provozní personál kontroluje dokumentaci atd.
- **Uspokojení zainteresovaných skupin.** Na konci Přechodové fáze a celého životního cyklu je řešení nasazeno na produkci a zájmové skupiny jsou s ním spokojeny.

## 7 Spolupráce v rámci podniku

„Enterprise Aware“, neboli spolupráce napříč podnikem je klíčovým aspektem DAD frameworku. Je důležité se uvědomit, že žádný tým (vyjma mále firmy) nepracuje v naprostém vakuu. Většina informačních systémů již běží v pracovním prostředí a řešení, které je dodáváno, má na ně dopad. Nebo ještě lépe, jak uvádí Scott Abler, „Doufejte, že vaše řešení ovlivní stávající funkcionality a data v produkci“ [8]. Často je tak potřeba úzce spolupracovat s podnikovými speciality, rozumějící podnikové architektuře, technické infrastruktuře, produktovému portfoliu, architektuře byznysu a tak dále.

Spolupráce v podniku je také důležitá proto, že může ušetřit čas a tím i náklady. Například vytváření znovu použitelných nebo vylepšování stávajících komponent či aplikací, které mohou být znovupoužity

na projektech. Další týmy tak nemusejí ztrácet čas vývojem již jednou vytvořených částí. Sdílení informací a své zkušeností napříč celým podnikem má za následek optimalizování vývoje na úrovni celého podniku, nejenom v rámci jednoho projektu, respektive v rámci dodávky jednoho řešení. Proto je důležité, aby se členové jednotlivých týmů chovali jako profesionálové a dělali to, co je zajímavé a důležité pro celý podnik, ne co je zajímavé a důležité pro ně samé.

## 8 Škálování

Jak už bylo řečeno dříve, DAD, díky přístupu řízení pomocí cílů místo přesného popisu procesu, dává možnost přizpůsobit agilní postupy a praktiky podle potřeby. Ovšem je nutné si uvědomit, že přizpůsobením podle potřeby se zde nerozumí pouze velikost týmu. Můžou zde být i další faktory, které je potřeba rozlišit. Jejich určení pak vychází z kontextu, v němž se daný tým nebo projekt nacházejí. Mezi škálovací faktory podle DAD patří:

- **Geografické rozdělení.** Zde je nutné uvažovat a určit, kde se náš tým vlastně nachází. Jsme týmem sedícím v jedné místnosti? Nebo sedíme ve stejném patře, ve stejné budově, ve stejném městě, v několika zemích?
- **Velikost týmu.** Již zmiňovaná velikost. Je potřeba určit, zdali je náš tým je malá skupina o několika lidech nebo jej tvoří tisíc lidí
- **Dodržování předpisů a norem.** Jedná se o určení předpisů a norem, jejichž dodržování značně ovlivní činnosti nebo postupy týmu (například vyšší množství dokumentace nebo větší důraz na testování). Nemusí se jednat pouze o externí regulátory (při vývoji softwaru ovlivňující zdraví lidí), ale mohou to být interní regulátory (CMMI nebo ISO normy).
- **Komplexnost domény.** Rozdílná aplikace agilních postupů a praktik bude také odlišná v případě, kdy řešením je pouze obyčejná webová prezentace a v případě, kde se bude jednat o vybudování komplexního interní aplikace pro velký podnik.
- **Technická složitost.** Jiný bude také vývoj softwaru, kdy začínáme na zelené louce, s použitím jedné technologické platformy a bez návaznosti na další systémy. A jiný bude ve chvíli, kdy budeme vyvíjet komplexní komponentu do již zaběhlého informačního systému obsahující x dalších komponent a bude potřeba řešit záležitosti, jako jsou integrace, datová migrace, stejné uživatelské rozhraní a tak podobně.
- **Organizační rozložení.** Bude třeba také přihlídnout k tomu, jestli jsme tým složený s lidí pracujících pro stejnou firmu ve stejném oddělení nebo jsme tým složený z lidí z jiných oddělení jedné firmy, z více firem nebo mezi sebou máme i konzultanty.

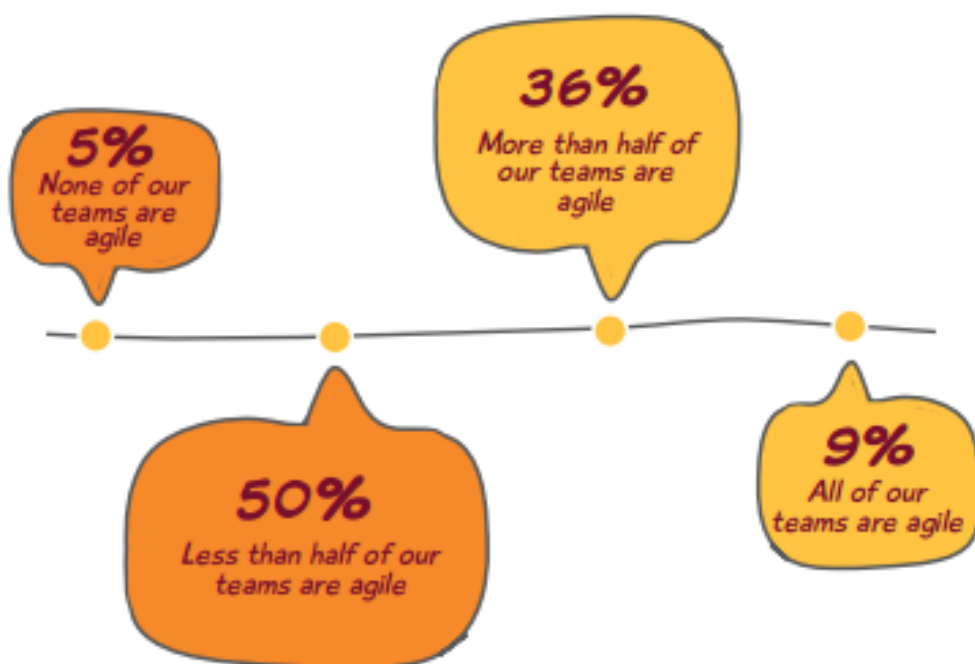
Určením každého z těchto faktorů pomůže týmu lépe škálovat, respektive přizpůsobit, agilní postupy a praktiky jeho situaci, ve které se právě nachází.

## 9 Současný stav

### 9.1 Adaptace agilních metodik

Poslední trend stoupající oblíbenosti agilních metodik stále pokračuje. To dokazuje i průzkum společnosti VersionOne provedený v roce 2014. 94 % z dotázaných organizací uvedlo, že má zkušenost s agilními praktikami. Z toho 35 % dotázaných má více jak 5 000 zaměstnanců a 20 % dotázaných má dokonce více než 20 000 zaměstnanců.

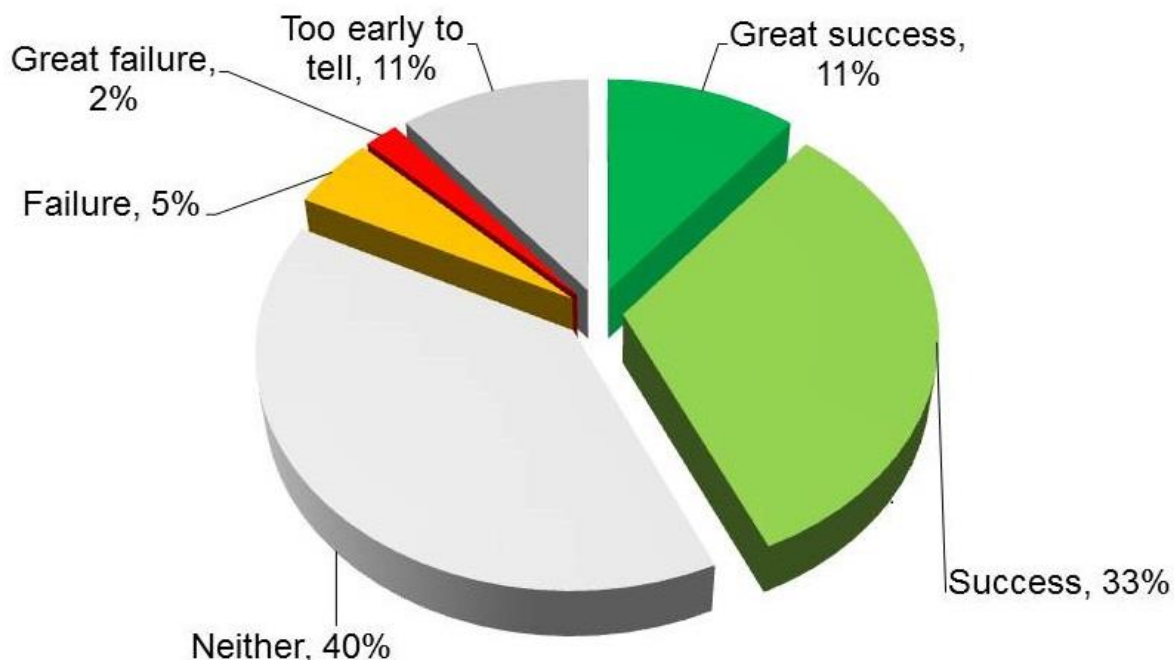
Jak ukazuje následující obrázek, 45 % dotázaných řeklo, že většina jejich týmu pracuje agilní. Pouze 5 % dotázaných stále používá tradiční, neagilní metodiky. Ve srovnání s rokem 2009, kdy agilní metodiky nepoužívalo 31 % respondentů, to je snížení o 26 % [10].



Obrázek 9 - míra adaptace agilních metodik [10]

Ovšem, jak ukazuje průzkum Scotta Ablera na úspěšnost adaptace agilních metodik, úspěšnost adaptace agilních metodik nepřesahuje 50 %. Ovšem je nutné přihlídnout fakt, že 40% respondentů sice neodpovědělo záporně, ale ani kladně.

I would rate my organization's adoption of agile as...



Copyright 2014 Scott Ambler + Associates  
Source: SA+A 2014 Agile Adoption Survey  
[Ambysoft.com/surveys/agileJanuary2014.html](http://Ambysoft.com/surveys/agileJanuary2014.html)

SCOTT AMBLER  
+ Associates

Obrázek 10 - úspěšnost adaptace agilních metodik<sup>1</sup>

## 9.2 Výuka

Školení metodiky v současné době probíhá pod záštitou „Disciplined Agile Consortium“, jehož součástí je 12 certifikovaných mentorů, včetně Scotta Ablera a Marka Lineše. Konsorcium umožňuje získat „Disciplined Agile“ certifikáty, které jsou rozděleny do 4 úrovní. Každá úroveň je označena barevným páskem představujícím stupeň znalostí[11]. Barvy pásků jsou:

- **Bílý** – Certifikovaný má za sebou úvodní školení do praktik metodiky
- **Žlutý** – Označuje vlastníka certifikátu s bílým páskem, který je ochotný se dál učit a zlepšovat své schopnosti softwarového profesionála
- **Zelený** – Certifikovaný má zkušenosti s metodikou a může školit ostatní, ovšem pod dohledem senior mentora (vlastník certifikátu s černým páskem)
- **Černý** – Označuje experta v metodice s možností vést školení a zavádět nebo přizpůsobovat metodiku přímo v organizacích

<sup>1</sup> 2014 Agile Adoption Survey Results. 2014. 2014 Agile Adoption Survey Results [online]. [cit. 2015-05-14]. Dostupné z: <http://www.ambysoft.com/surveys/agileJanuary2014.html>

Pro jednotlivé certifikáty je nutné splnit následující požadavky:

|                             | Bílí pásek                                                                                              | Žlutý pásek | Zelený pásek    | Černý pásek     |
|-----------------------------|---------------------------------------------------------------------------------------------------------|-------------|-----------------|-----------------|
| Certifikační test           | Ne                                                                                                      | Ano         | Ano             | Ne              |
| Zkušenosti s agilem         | Ne                                                                                                      | Ne          | Více jak 2 roky | Více jak 5 let  |
| Zapojení do komunity        | Ne                                                                                                      | Ne          | Ano             | Ano             |
| Účast na jednom z workshopů | -DAD Experience Workshop<br>-Introduction to DAD<br>-Product Owner Workshop<br>-Disciplined Agile DW/BI | Požadováno  | Požadováno      | Ne              |
| Interview členskou radou    | Žádné                                                                                                   | Žádné       | Žádné           | Ano             |
| Vedení školení / týmu       | Žádné                                                                                                   | Žádné       | Žádné           | Více jak 2 roky |

Obrázek 11 - certifikační požadavky<sup>2</sup>

## 10 Závěr

Discipline Agile Delivery poskytuje alternativu mezi dvěma extrémy. Na jedné straně agilními metodikami, které nevyhovují prostředí větších podniků a na straně druhé rigorózními metodikami, které je potřeba ztíhli. Metodiku je totiž možné pomocí „Goal driven“ přístupu škálovat podle konkrétních potřeb, které vyžaduje konkrétní situace a prostředí týmu. Možnost škálovatelnosti je nejsilnější stránkou celé metodiky. Metodika navíc prosazuje i další přístupy pro snížení rizik nebo na podporu spolupráce agilních týmů napříč organizací. Se stále pokračujícím trendem adopce agilních metodik ve větších organizacích, je jasné, že potřeba metodiky, která dokáže přizpůsobit agilní metody a principy konkrétnímu firemnímu prostředí, stále poroste. Toho jsou si dobře vědomi i tvůrci metodiky, Scott Ambler a Mark Lines, kteří si okolo sebe vytváří silnou komunitu „Disciplined Agilistů“ a schopných mentorů, kteří k rozšíření metodiky budou dále přispívat.

<sup>2</sup> Disciplined Agile Consortium. 2014. Disciplined Agile Consortium [online]. [cit. 2015-05-14]. Dostupné z: <http://www.disciplinedagileconsortium.org/apply>

## 11 Zdroje

- [1] Disciplined agile delivery. 2013. AMBLER, Scott W. a Mark LINES. Crosstalkonline.org [online]. [cit. 2015-05-14]. Dostupné z: <http://m.crosstalkonline.org/issues/10/82/>
- [2] Roles in Disciplined Agile Delivery. 2012. AMBLER, Scott W. Wordpress.com [online]. [cit. 2015-05-14]. Dostupné z: <https://disciplinedagiledelivery.wordpress.com/2012/12/18/roles-in-disciplined-agile-delivery/>
- [3] The Product Owner Role: A Stakeholder Proxy for Agile Teams. 2014. AMBLER, Scott W. Agile Modeling [online]. [cit. 2015-05-14]. Dostupné z: <http://www.agilemodeling.com/essays/productOwner.htm>
- [4] The DAD Role of Architecture Owner. 2012. LINES, Mark. Wordpress.com [online]. [cit. 2015-05-14]. Dostupné z: <https://disciplinedagiledelivery.wordpress.com/2012/05/28/the-dad-role-of-architecture-owner/>
- [5] Full Agile Delivery Lifecycles. 2014. AMBLER, Scott W. Disciplined Agile Delivery: An agile process decision framework for the enterprise [online]. [cit. 2015-05-14]. Dostupné z: <http://www.disciplinedagiledelivery.com/lifecycle/>
- [6] It Requires Discipline to Keep Inception Short. 2012. AMBLER, Scott W. Disciplined Agile Delivery: An agile process decision framework for the enterprise [online]. [cit. 2015-05-14]. Dostupné z: <http://www.disciplinedagiledelivery.com/it-requires-discipline-to-keep-inception-short/>
- [7] It Requires Discipline to Evolve Transition From a Phase to an Activity. 2012. AMBLER, Scott W. Disciplined Agile Delivery: An agile process decision framework for the enterprise [online]. [cit. 2015-05-14]. Dostupné z: <http://www.disciplinedagiledelivery.com/it-requires-discipline-to-evolve-transition-from-a-phase-to-an-activity/>
- [8] AMBLER, Scott W a Mark LINES. 2012. Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise. Upper Saddle River, NJ: IBM Press, xxvii, 513 p. ISBN 978-013-2810-135.
- [9] AMBLER, Scott W. 2013. Going Beyond Scrum: Disciplined Agile Delivery. Going Beyond Scrum: Disciplined Agile Delivery [online]. (1) [cit. 2015-05-14]. Dostupné z: <http://disciplinedagileconsortium.org/Resources/Documents/BeyondScrum.pdf>
- [10] VersionOne Releases 9th Annual State of Agile™ Survey Results. 2015. VersionOne [online]. [cit. 2015-05-14]. Dostupné z: <http://www.versionone.com/about-us/press-releases/article/VersionOne-Releases-9th-Annual-State-of-Agile-Survey-Results/>
- [11] Disciplined Agile Consortium [online]. 2014. [cit. 2015-05-14]. Dostupné z: <http://www.disciplinedagileconsortium.org/>
- [12] Agile modeling. 2014. Agile modeling [online]. [cit. 2015-05-14]. Dostupné z: <http://agilemodeling.com/essays/prioritizedRequirements.htm>
- Disciplined Agile Delivery [online]. 2015. [cit. 2015-05-14]. Dostupné z: <https://disciplinedagiledelivery.wordpress.com/>
- Disciplined Agile Delivery [online]. 2015. [cit. 2015-05-14]. Dostupné z: <http://www.disciplinedagiledelivery.com/>