

Vysoká škola ekonomická  
**Fakulta informatiky a statistiky**



Semestrální práce

# **Disciplined Agile Delivery (DAD) framework**

Kurz: 4IT421 Zlepšování procesů budování IS

Autor: Bc. Radim Klepetko

Vypracováno: ZS 2011/2012

## Obsah

---

Obsah.....	2
1 Úvod .....	3
2 Agilní metodiky.....	3
3 DAD framework.....	4
4 Principy DAD.....	5
4.1 Lidé na prvním místě .....	5
4.2 Orientace na učení .....	7
4.3 Agilita.....	7
4.4 Hybridní procesní framework .....	8
4.5 Orientace na IT řešení .....	9
4.6 Životní cyklus zaměřený na dosažení cílů .....	10
4.7 Spolupráce v rámci podniku .....	11
4.8 Orientace na rizika a hodnotu .....	12
5 Závěr .....	15
6 Zdroje .....	16

## 1 Úvod

---

Tato práce si klade za cíl seznámit čtenáře s principy frameworku Disciplined Agile Delivery (DAD) tak, jak je nastínili Scott Ambler a Mark Lines ve svém whitepaperu *Disciplined Agile Delivery: An introduction* a dalších zdrojích. Po stručném úvodu do agilních metodik a základní seznámením s DAD bude v práci prezentováno osm základních principů DAD.

## 2 Agilní metodiky

---

Metodiky pro vývoj software používané na konci 20. století v podstatě kladly software do role sériového výrobku – co nejlépe jeho vývoj popsat, naplánovat, přesně se plánu držet a následně rigidně, stabilně a pokud možno bez jakýchkoliv velkých zásahů a změn vyvinout a předat zákazníkovi. Tyto nepružné metodiky, které se už ve své podstatě brání jakýmkoliv změnám a alteracím se v dnešním rychlém a dynamickém obchodním prostředí nehodí pro všechny projekty.

Zákazníci softwarových projektů obvykle vyvíjené produkty nasazují proto, aby získali konkurenční výhodu, snížili náklady a podobně. Všechny tyto případy mají společnou jednu skutečnost: společnosti nechtějí čekat, a tak se vývojové projekty neustále zrychlují a termíny zkracují. To vše, aniž by však docházelo k výraznému navyšování zdrojů projektům alokovaných ani ke snižování požadavků na ně kladených. Zákazníci stále trvají na široké funkcionalitě, ze které nechtějí ustoupit. Na rigidní a tím pádem do jisté míry i pomalé metodiky, jako je vodopádový vývoj nebo RUP, není čas. Z toho důvodu vznikly agilní metodiky, které vycházejí z premisy, že jediný způsob, jak ověřit správnost navrženého systému, je co nejrychleji ho vytvořit a dát k posouzení zákazníkovi. Důraz je kladen na

rychlost, krátké iterace a přizpůsobivost ke změnám. Mezi nejznámější agilní metodiky patří extrémní programování, SCRUM, Lean Development, Test Driven Development, Crystal, Adaptive Software Development a další.

### 3 DAD framework

---

DAD framework je dílem hlavního metodologa IBM pro agilní/lean vývoj, Scotta Amblera. Jedná se v podstatě o hybridní metodiku nebo spíše sadu vzájemně se doplňujících doporučení, která vycházejí z nejčastěji používaných metodik (Scrum, extrémní programování, agilní modelování, lean a unifikovaný proces). Většina stávajících agilních metod se totiž některými oblastmi vývoje vůbec nezabývá. Scrum tak například vůbec nehovoří o tom, jak by měl být veden životní cyklus produktu, a zaměřuje se primárně na management a odhlíží tak například i od architektury. DAD se inspirovuje a vybírá nejlepší postupy z nejrůznějších agilních metodik.

DAD také automaticky nezahazuje všechny postupy známé z klasického vývoje softwaru jen proto, že „teď se vyvíjí agilně“, a stále na ně klade důraz (například nutnost spolupracovat s ostatními projekty, věnovat pozornost podnikové architektuře, databázím a podobně). DAD se také snaží nevyužívat terminologie, která už je zažitá v konkrétních agilních metodikách (například „sprint“). Místo toho využívá terminologii obecnou, která je srozumitelná bez ohledu na to, jakou metodiku (a tím pádem i terminologii) který vývojář používá.

## 4 Principy DAD

### 4.1 Lidé na prvním místě

---

Stejně jako i manifest agilního programování, i DAD chápe lidi jako základní a neoddělitelnou součást vývoje softwaru. Lidé a způsob, jakým spolupracují, jsou primárním faktorem úspěšného vývoje. DAD poskytuje návod pro spolupráci DAD týmů, nicméně nepředepisuje žádné povinné postupy. DAD nedoporučuje v rámci týmů hierarchii a týmy by se podle jeho doporučení měly skládat z jedinců, které mají sadu různých dovedností a jsou schopni zastávat různé funkce. Čím rozsáhlejší jsou schopnosti a dovednosti člena týmu, které nesouvisí s jeho primární odpovědností, tím efektivnější bude využití zdrojů a také se sníží nutnost spoléhat na formální dokumentaci.

DAD definuje pět základních rolí v týmu:

- **Stakeholder** – kdokoliv, kdo je nějakým způsobem materiálně ovlivněn výsledkem vytvářeného řešení. Tedy nejen koncový uživatel, ale také jeho manažer, vývojáři pracující na řešeních, které s ním souvisejí, profesionálové, kteří mají za úkol údržbu systému a další osoby, které jsou vývojem projektu potenciálně ovlivněny.
- **Vlastník produktu** – osoba, která je mluvčím zákazníka. Z pohledu DAD představuje všechny potřeby zákazníka, vyjasňuje detaily a určuje prioritu jednotlivých požadavků. I když vlastník nemusí být schopen zodpovědět všechny otázky, je zodpovědný za to, aby se odpovědi na ně k vývojářům bez zbytečného prodlení dostaly. Pokud vlastník úzce spolupracuje s vývojovým týmem, výrazně

tak klesá potřeba dokumentace. Každý DAD (sub)tým má jediného vlastníka produktu. Sekundárním úkolem vlastníka je prezentace produktu veřejnosti a komunikace s klíčovými stakeholdery.

- **Člen týmu** – někdy také vývojář nebo programátor. Podle DAD však ne každý člen týmu musí nutně vytvářet kód. Jako člena týmu DAD chápe kohokoliv, kdo nějakým způsobem přispívá ke tvorbě řešení pro stakeholdery (testování, analýza, design, programování, plánování,...).
- **Vedoucí týmu** – agilní kouč, který udržuje tým v tempu a pomáhá mu dosahovat výsledků, které byly stakeholderům přislíbeny. Funguje jako vůdce, motivuje, pomáhá optimalizovat procesy, zajišťuje týmu dostatečné zdroje a včas se zbavuje všech překážek, které by mohly činnost týmu narušit.
- **Vlastník architektury** – rozhoduje za tým o celkovém řešení architektury a dohlíží na její vývoj a dodržování. Architektura je totiž klíčovým zdrojem projektového rizika. Pro snížení tohoto rizika je podle DAD nezbytné, aby za architekturu byla zodpovědná jedna konkrétní osoba. Důležitým faktem je, že vlastník architektury nediktuje její podobu, pouze vede její tvorbu. U malých projektů plní roli vlastníka architektury vlastník produktu.

DAD nezná roli testera ani byznys analytika. Místo toho očekává, že jejich funkci zastanou řadoví členové týmu. V DAD se totiž od každého člena týmu očekává široké penzum vědomostí a schopností a stejně tak i ochota se nové věci za běhu v případě potřeby doučit.

## 4.2 Orientace na učení

---

Podle tvůrců DAD patří mezi nejefektivnější organizace právě ty, které aktivně školí své zaměstnance a podporují je v dalším vzdělávání. DAD definuje tři základní oblasti, ve kterých je nutné se učit:

- **Doménová oblast:**
  - *Jak nejlépe rozpoznat potřeby stakeholderů a definovat je?*
- **Procesní oblast:**
  - *Jak vylepšovat procesy na úrovni jednotlivce, týmu i celého podniku?*
- **Technologická oblast:**
  - *Jak efektivně využívat nástroje a technologii, které se pro tvorbu řešení využívají?*

## 4.3 Agilita

---

DAD se drží manifestu agilního vývoje a dále jeho myšlenky a principy rozvíjí. Podle autorů DAD se ukázalo, že týmy využívající iterativní nebo agilní vývoj dosahují kvalitnějších výsledků, vyššího uspokojení stakeholderů a vyšší návratnosti investic než týmy postupující podle tradičního (vodopádového) vývojového schématu nebo týmy pracující ad-hoc (bez jasně definovaného postupu) a zároveň jsou výsledek schopny dodat v kratším čase.

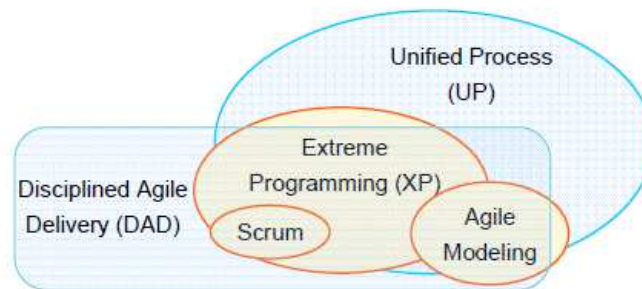
Vysoká kvalita je zajišťována technikami jako je kontinuální integrace (CI – continuous integration), regresní testování, vývoj, kterému testy předcházejí a refaktoring. Vyšší návratnost investic plyne ze soustředění na nejdůležitější aktivity, prioritizace, automatizace všeho, co automatizovat lze, úzké spolupráce a obecného přístupu, který vybízí pracovat chytře, nikoliv nutně tvrdě.

Větší spokojenost stakeholderů je dosažena pomocí aktivní spolupráce stakeholderů na vývoji řešení, inkrementálnímu dodávání funkcí v jednotlivých iteracích a možnost stakeholderů formulovat své požadavky v průběhu celého projektu.

#### 4.4 Hybridní procesní framework

DAD adoptuje strategie a postupy z nejrozšířenějších agilních metod i dalších zdrojů a vytváří tak ucelený framework, který je jakýmsi hybridem (proto „hybridní“). DAD vychází z následujících metodik.

Obrázek 1: DAD komponuje přístupy několika různých metodik



Zdroj: [6]

- **Scrum** – Scrum se zaměřuje hlavně na vedení projektu a některé aspekty řízení požadavků. DAD od Scrumu přijímá celou řadu myšlenek, které si dále upravuje. Jde například o zpracování jednotlivých jednotek práce na základě priority, existence vlastníka produktu, který je odpovědný za reprezentování stakeholderů a vytváření potenciálně použitelného produktu v každé iteraci. S výjimkou *vlastníka produktu* však DAD opouští terminologii Scrumu a zavádí si vlastní.
- **Extrémní programování** je pro DAD důležitým zdrojem vývojovým postupů. Patří sem například kontinuální integrace, refaktorování, vývoj řízený testy a mnohé další.



- **Agilní modelování** pro DAD slouží jako zdroj modelovacích a dokumentačních technik.
- **Unifikovaný proces:** DAD si bere za své celou řadu strategií řízení z OpenUP a AUP (Agile Unified Process). Konkrétně jde hlavně o nepřiliš rozsáhlé milník a jasně vyjádřené fáze. Důraz je kladen také na vytvoření architektury v začátku projektu, aby tak byly snížena rizika v dalších částech životního cyklu.
- **Agilní data (AD)** jsou pro DAD zdrojem databázových postupů, jako je databázový refaktoring, testování databází a agilní datové modelování. DAD si z AD také bere prostupy pro spolupráci agilních týmů s podnikovými architekty a správci podnikových dat.
- **Kanban:** DAD si odtud bere dva důležité koncepty: snahu omezit rozpracovanou práci a vizualizaci práce. Tyto koncepty jsou využity spolu se sedmi koncepty lean developmentu.

#### 4.5 Orientace na IT řešení

---

Hlavním cílem týmu pracujícího podle DAD je vytvářet řešení, ne software. Je to totiž právě řešení, co stakeholder chce a co mu také přinese největší možnou hodnotu. Nejde tedy jen o samotný software. Součástí řešení může být například i nový hardware, optimalizace procesů stakeholdera nebo změna jeho organizační struktury. V tomto ohledu se DAD poněkud rozchází s manifestem agilního programování. Jeho tvůrci totiž byli primárně programátoři, vývojáři nebo konzultanti. Zatímco manifest je tak orientován výhradně na programování softwaru, DAD zachází dále a uvažuje o řešení jako celku, který jde za hranice samotného softwaru.

## 4.6 Životní cyklus zaměřený na dosažení cílů

---

DAD popisuje životní cyklus projektu od samotného spuštění, přes tvorbu až po finální nasazení do produkce. DAD zároveň explicitně konstatuje, že ne každá iterace je stejná. Projekt se v průběhu svého životního cyklu mění a spolu s ním i oblast, na kterou se tým soustředí. Tento životní cyklus má čtyři základní vlastnosti:

- **Jedná se o životní cyklus zaměřený na dodání** - DAD ještě rozšiřuje životní cyklus, který využívá Scrum, a zdůrazňuje to, že se zabývá životním cyklem od samotného začátku projektu až po nasazení řešení do provozu.
- **Existující explicitní fáze** - životní cyklus DAD do tří rozdílných pojmenovaných fází, které odráží agilní 3C rytmus (coordinate-collaborate-conclude).
- **Životní cyklus v kontextu** – DAD bere na vědomí, že životní cyklus probíhá až roky předtím, než započne skutečný vývoj produktu, a navíc, že pokračuje i poté, co je produkt doručen zákazníkovi (provoz a podpora).
- **Explicitní milníky** – v DAD existující jasně dané milníky, které pomáhají v řízení produktu a snižují rizika.

DAD definuje tři fáze projektu:

- **Počáteční fáze** – před samotnou výstavbou projektu je vhodné identifikovat všechny cíle, kterých má projekt dosáhnout. DAD doporučuje fázi plánování o délce několika dnů až několika týdnů (stojí tak na pomezí mezi klasickými metodikami, kde plánování trvá velmi dlouho, a agilními metodikami, kdy se v některých případech neplánuje vůbec).
- **Konstrukční fáze** – konstrukční fáze zahrnuje časový úsek, během kterého se vytváří požadovaná funkcionalita. Tato fáze se dělí na další časově ohraničené

iterace (o délce typicky od jednoho do čtyř týdnů). Na konci každé iterace probíhá testování a vytvořená část projektu se prezentuje stakeholderům.

- **Fáze přechodu** – tato fáze je zaměřena na nasazení systému do provozu. Patří sem také betatestování systému v malých skupinách uživatelů nebo například pořízování nového hardware (je-li potřeba). Fáze přechodu končí ve chvíli, kdy jsou stakeholdeři připraveni a systém je plně nasazen.

#### 4.7 Spolupráce v rámci podniku

---

S výjimkou start-upů nepracují agilní týmy nikdy ve vakuu. Obvykle již existují nějaké běžící systémy, které by systém vyvíjený týmem DAD měl využívat. Obvyčejně zde jsou také další týmy, které pracující paralelně s DAD týmem, a bývá výhodné využívat výsledky jejich práce, přičemž druhý tým jistě také uvítá výsledky DAD týmu. Společnost, pro kterou DAD tým pracuje, může mít nějakou svoji vizi, a práce týmu by měla probíhat v souladu s ní. Stejně tak by měla existovat určitá strategie globálního řízení, která týmu v práci pomůže.

DAD pro spolupráci v rámci podniku definuje tři základní zásady:

- **Využití podnikových zdrojů:** Každý podnik má (nebo by alespoň mít měl) celou řadu zdrojů, které je možné využít. V první řadě to jsou nejrůznější standardy (programovací, databázové, bezpečnostní, pro uživatelské rozhraní,...). Dále může jít o celé knihovny již hotových komponent ve formě služeb, které se dají využít a vylepšit tak, aby se z nich dalo vycházet ve všech dalších řešeních.
- **Obohacení ekosystému organizace:** Řešení vytvářené DAD týmem by mělo do ekosystému cílové společnosti přinejmenším zapadnout (ekosystémem zde myslíme byznys procesy a systémy, které je podporují). Pokud je to aspoň trochu možné, mělo by tento ekosystém obohatit. Aby to bylo možné, je třeba využívat

již existujících podnikových zdrojů, jak bylo popsáno výše, úzce spolupracovat s uživateli a je-li to možné spolupracovat s integračním týmem, který bude průběžně testovat, jak vytvářené řešení v ekosystému společnosti funguje.

- **Otevřený a poctivý monitoring:** I když je celý agilní vývoj založen na důvěře, není podle DAD na škodu uplatňovat přístup „důvěřuj, ale prověřuj a následně ved“<sup>1</sup>. Jedna z možných strategií navrhuje, aby kdokoliv, kdo má informace o postupu DAD týmu zájem, přišel na každodenní setkání týmu a vyslechl si ho. Jiná navrhuje využití specializovaného softwaru, který v reálném čase postup projektu zachycuje na dashboardech. Třetí navrhanou strategií je uchýlit se k vývoji orientovanému na rizika, kterému se věnuje další část práce

#### 4.8 Orientace na rizika a hodnotu

---

DAD uznává filozofii „likvidovat rizika dříve, než ony zlikvidují nás“, která je součástí i dalších metodik, například Scrumu a extrémního programování. Pokud jde o hodnotu, snaží se DAD doručovat v každé iteraci potenciálně použitelné řešení. Doručované funkce mají být ty, které mají pro stakeholdery co možná nejvyšší hodnotu. Životní cyklus zaměřený na hodnotu se potýká se třemi základními problémy:

- neschopnost dodat řešení;
- riziko dodání nesprávné funkcionality;
- riziko nedostatečně viditelných výsledků práce týmu.

DAD doporučuje těmto rizikům předcházet (mimo jiné) třemi základními principy, které jsou v agilním vývoji běžné. Jde o:

- **Potenciálně použitelná řešení** – v každé iteraci by mělo být vytvořeno potenciálně použitelné řešení. Jedná se o rozšíření filozofie softwaru, který je možné potenciálně dodat (a kterou využívá metodika Scrum). Opět zde o jednu ze základních myšlenek DAD – nevytváříme pouze software, ale celá řešení, a tak nám nejde jen dodání, ale i o použitelnost. Tento princip snižuje riziko, neboť stakeholderům umožňuje nasadit řešení do produkčního provozu prakticky kdykoliv (pokud to tedy má vzhledem k funkcionalitě smysl).
- **Ukázky po každé iteraci** – na konci každé iterace by měl DAD tým připravit pro stakeholdery ukázkou toho, co vytvořil. Primárním cílem těchto ukázek je získat zpětnou vazbu, což snižuje riziko špatné funkcionality. Sekundárním cílem je zviditelnění výsledků a práce týmu.
- **Aktivní účast stakeholderů** – ideální stav je takový, že stakeholdeři (respektive jejich zástupce – vlastníci produktu) se nejen podílejí na rozhodování a včasně dodávají potřebné informace, ale zároveň se také aktivně účastní vývoje. Například mohou aktivně modelovat a účastnit se tak celé iterace a ne pouze ukázek.

Tyto základní principy slouží jako solidní začátek, nicméně DAD u nich rozhodně nekončí.

DAD dále identifikuje několik základních milníků v životním cyklu každého projektu. Jedná se o:

- **Konsenzus se stakeholdery** – cílem tohoto milníku, který patří na závěr počáteční fáze, je dojít k rozumnému konsenzu ohledně předpokládaného výsledku projektu. Snižují se tím rizika spojené s funkcionalitou i dodání.
- **Osvědčená architektura** – i když návrhy vysokoúrovňové architektury vytvářené v průběhu počáteční fáze pomohou při promýšlení architektury, fakt, zda bude

definovaná architektura podporovat všechny požadavky stakeholderů, se ukáže až v okamžiku psaní kódu. DAD týmy tedy píší software, aby potvrdily, že architektura bude projektu vyhovovat a odstraní tak tento druh risku již brzy po začátku projektu.

- **Trvalá životaschopnost** – Scrum má pravidlo, podle kterého by stakeholderi měli na konci každého sprintu posuzovat životaschopnosti projektu. Jakkoliv je sice tato myšlenka zajímavá, v praxi to podle autorů DAD málokdy funguje. Proto navrhuji méně časté posuzování životaschopnosti a to po čtvrt roce. Tento milník se tak u delších projektů může několikrát opakovat, zatímco u kratších nemusí nastat vůbec.
- **Dostatečná funkcionalita** – milník produkční fáze, který se považuje za dosažený v okamžiku, kdy má řešení dostatek funkcionality na to, aby se vyplatilo jeho nasazení do provozu. Řešení musí splňovat všechna akceptační kritéria, která byla v průběhu projektu stanovena nebo jim musí být alespoň dostatečně blízko.
- **Připravenost k nasazení do provozu** - na konci přechodové fáze musí klíčoví stakeholderi rozhodnout, zda bude řešení nasazeno do provozu. V této fázi musí být spokojení byznys stakeholderi i osoby zodpovědné za provozování i údržbu systému.

## 5 Závěr

---

Podle zkoumání provedených autory DAD přinášejí agilní metodiky lepší výsledky než metodiky klasické. To je možné považovat za dobrou zprávu. Špatnou zprávou však zůstává fakt, že (dnes již) standardní agilní metodiky vždy pokrývají pouze část celkového procesu dodávky softwaru (nebo, v případě DAD, řešení). Disciplined Agile Delivery framework (DAD) je hybridním frameworkem, který si vybírá ty nejlepší postupy a doporučení ze stávajících metodik a formuluje z nich komplexní doporučení, která pokrývají celý životní cyklus produktu. DAD tak může zajímavou cestou jak zkombinovat to nejlepší z různých agilních metodik a vytvořit komplexní sadu doporučení, která pokrývá (na rozdíl od většiny ostatních agilních metodik) prakticky celý vývoj softwaru.

Otázkou zůstává, na jaké typy projektů a v jakých situacích DAD nasazovat. Autoři DAD uvádějí, že jejich metodika se dá s použitím techniky Agility@Scale (viz příloha k práci) prakticky libovolným způsobem škálovat. I přesto, že toto škálování s sebou nese zvýšenou pracnost projektu, tvrdí autoři, že metodika je vhodná pro projekty od deseti členů až po projekty s tisícovkou členů, projekty na jednom místě, i projekty s týmy po celém světě, projekty s nízkým rizikem i projekty kritické apod. Jediným kritickým faktorem, který je třeba dodržet, je vysoká odbornost, pružnost a ochota učit se u projektového týmu, jehož členové musí být schopni zvládnout zastat široké spektrum (sub)rolí a neustále svoji specializaci dále rozšiřovat.

## 6 Zdroje

---

[1] AMBLER, Scott; LINES, Mark. Disciplined Agile Delivery:An introduction. [online]. 2011, [cit. 2011-11-03]. Dostupný z WWW:

<<http://public.dhe.ibm.com/common/ssi/ecm/en/raw14261usen/RAW14261USEN.PDF>>.

[2] LINES, Mark; AMBLER, Scott. Disciplined Agile Delivery : A Practitioner's Guide to Agile Software Delivery in the Enterprise [online]. [s.l.] : IBM Press, 10. června 2012 (předpoklad) [cit. 2011-11-03]. Dostupné z WWW:

<<http://my.safaribooksonline.com/9780132810098?portal=informat>>. ISBN 0-13-281013-1.

[3] AMBLER, Scott. IBM : Developer works [online]. 3. 11. 2009 [cit. 2011-11-03]. Disciplined Agile Delivery. Dostupné z WWW:

<[https://www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/entry/disciplined\\_agile\\_delivery?lang=en\\_us](https://www.ibm.com/developerworks/mydeveloperworks/blogs/ambler/entry/disciplined_agile_delivery?lang=en_us)>.

[4] AMBLER, Scott. The Agile Scaling Model (ASM) : Adapting Agile Methods for Complex Environments. [online]. 2009, [cit. 2011-11-03]. Dostupný z WWW:

<<ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/raw14204usen/RAW14204USEN.PDF>>.

[5] Introduction to Disciplined Agile Delivery. [online]. 2010, [cit. 2011-11-03]. Dostupný z WWW:

< [http://www.xelaration.com/presentations\\_rug/montreal\\_rug\\_presentation\\_2010\\_03\\_30.pdf](http://www.xelaration.com/presentations_rug/montreal_rug_presentation_2010_03_30.pdf)>.



[6] Disciplined Agile Delivery (DAD): The Foundation for Scaling Agile. [online]. 2011, [cit. 2011-12-15]. Dostupný z WWW:

<<http://www.torontoagilecommunity.org/sites/default/files/Ambler%20Disciplined%20Agile%20Delivery%20and%20Scaling%20Agile.pdf>>.

Škálování DAD

Disciplined Agile Delivery (DAD):  
The Foundation for Agility@Scale



Zdroj: [6]