

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky



Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	LS 2019/2020
Autor	Marek Tomašík, tomm17
Téma	Agile Maturity Model

Abstrakt

Semestrální práce se zabývá modelem zralosti Agile Maturity Model, který vytvořili autoři Jez Humble a Rolf Russell. Cílem je tento model popsat, vysvětlit jeho fungování a uvést kroky při jeho použití v organizaci. Čtenáři tak práce nabídne kompletní vysvětlení tohoto modelu zralosti tak, aby jej pochopil a mohl vyzkoušet uplatnit ve své organizaci. Před popisem samotného modelu je vysvětlen agilní vývoj a jeho škálování. Závěrem jsou popsány přínosy modelu, je porovnán s modelem CMMI a také jsou uvedeny související studie.

Klíčová slova

model zralosti, Agile Maturity Model, škálování agilního vývoje

Obsah

1. Úvod	3
2. Agilní vývoj a jeho škálování	4
2.1. Agilní vývoj	4
2.2. Škálování agilního vývoje	4
3. Agile Maturity Model	6
3.1. Základní charakteristika	6
3.2. Kategorizace dle Horlach et al.	7
3.3. Popis modelu	7
3.3.1. Úrovně	7
3.3.2. Oblasti	7
3.4. Použití modelu	8
3.5. Přínosy	9
3.6. Porovnání s modelem CMMI	9
3.7. Související studie	10
4. Závěr	11
5. Literatura	12

1. Úvod

Cílem této semestrální práce je vytvořit strukturovaný popis modelu zralosti zvaný Agile Maturity Model od autorů ze společnosti ThoughtWorks Studios. Zejména vysvětlit fungování modelu včetně úrovní a oblastí, jeho vlastnosti, přínosy a rozdíly oproti podobným modelům zralosti v agilním prostředí.

Dalším cílem práce je vytvoření postupu, který v několika krocích popíše využití zmíněného modelu při vývoji softwaru v různých organizacích.

2. Agilní vývoj a jeho škálování

Tato kapitola definuje agilní vývoj a uvede agilní manifest včetně jeho principů. Dále vysvětlí co jsou škálovatelné agilní metodiky, vyjmenuje ty nejnámější a popíše souvislost s Agile Maturity Modelem.

2.1. Agilní vývoj

K začátkům vývoje softwaru neoddělitelně patří vodopádový přístup. Tedy přístup, kdy vývoj probíhá v jediném cyklu. Na začátku se definují požadavky na produkt, poté se implementuje, otestuje a na závěr nasadí k zákazníkovi. Rigorózní metodiky, jenž používají vodopádový přístup, však postupem času přestávaly být efektivní. Především kvůli obtížné a zdoluhavé reakci na změny. Z toho důvodu došlo v roce 2001 k vytvoření agilního manifestu, kterým jeho tvůrci vyjádřili své hodnoty při vývoji softwaru a principy, kterých se chtějí držet s cílem vývoj zefektivnit. Postupem času se tyto hodnoty a principy začaly používat v dalších společnostech a v dnešní době je tento přístup upřednostňován před vodopádovým.

Agilní manifest (Beck et al., 2001) upřednostňuje:

- jednotlivce a interakce před procesy a nástroji
- fungující software před rozsáhlou dokumentací
- spolupráci se zákazníkem před vyjednáváním o smlouvě
- reakci na změny před dodržováním plánu

Zároveň přidává 12 principů agilního vývoje. Mezi prvních pět patří (Beck et al., 2001):

1. Naší nejvyšší prioritou je vyhovět zákazníkovi častým a průběžným dodáním hodnotného softwaru.
2. Víťame změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.
3. Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.
4. Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.
5. Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.

Právě agilní manifest je základním kamenem agilního vývoje. Agilním vývojem se rozumí vývoj produktu nebo služby za pomoci hodnot a principů z agilního manifestu. Je to vývoj s využitím metod a principů, které se zaměřují na rychlejší reakci na změny a na provádění těch činností, jenž vytvářejí hodnotu pro zákazníka.

2.2. Škálování agilního vývoje

Se stále populárnějším agilním vývojem se agilní přístup dostal přes malé týmy až do velkých nadnárodních firem. Tam jej začaly používat malé týmy, ovšem bylo žádoucí agilní přístup aplikovat na větší týmy či na celou organizaci.

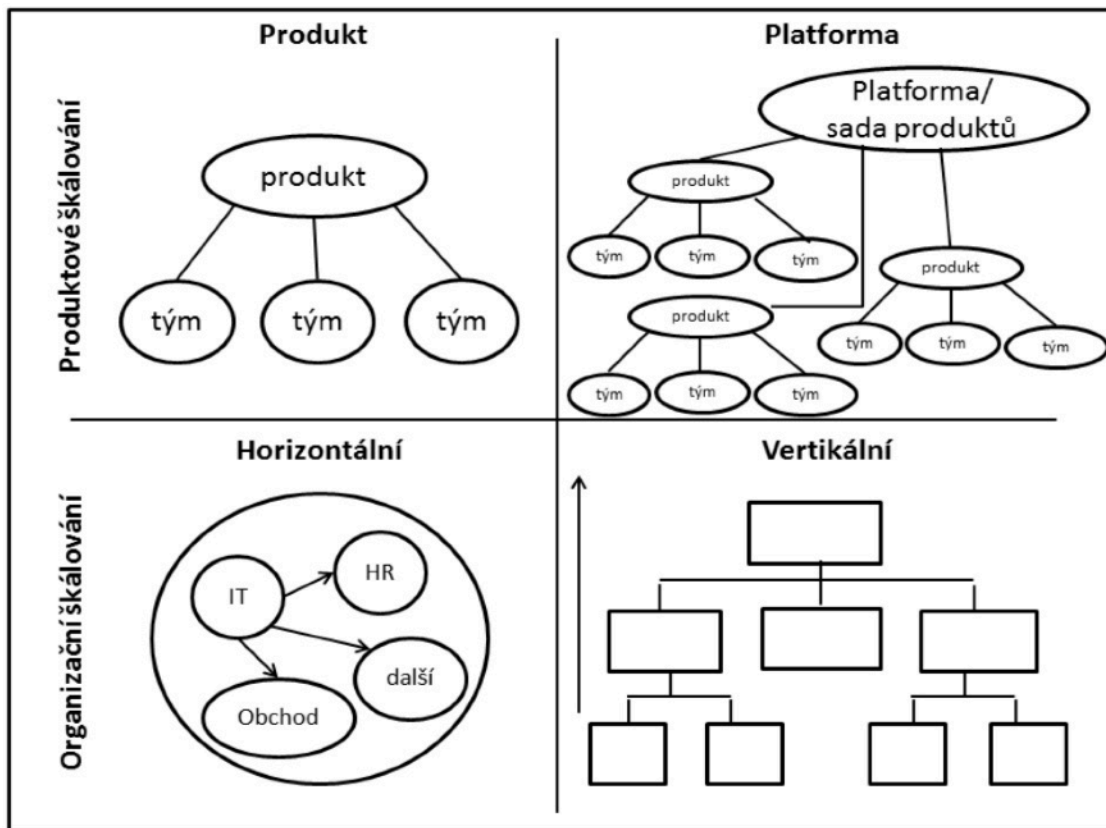
Agilní přístup má určitá omezení. Je to například velikost týmu - tým nemá být příliš velký. To prakticky znemožňuje aplikaci přístupu ve velkých firmách. Další omezení je v tom, že tým má úzce komunikovat se zákazníkem. V mezinárodních společnostech obvykle fungují distribuované týmy rozprostírající se do celého světa. Tento tým nemůže neustále fyzicky komunikovat se zákazníkem (Buchalceová, 2020).

Omezení agilního přístupu řeší právě škálovatelné metodiky. Mezi nejnámější patří:

- SAFe
- Scrum of Scrums
- LeSS
- Disciplined Agile Delivery

Škálovatelné metodiky nám umožňují aplikovat agilní přístup na rozsáhlejší týmy, na sady produktů, které jsou obhospodařovány několika týmy, nebo na další oddělení, mimo IT, v rámci organizace.

Škálování agilních praktik přes více týmů popisuje ve své knize Buchalceová (2018). Rozděluje škálování na produktové a organizační. Produktové škálování ještě rozděluje na produkt jako takový, kde několik týmů vyvíjí jeden produkt, nebo platformu, kde skupiny týmů vytvářejí produkty, jenž dohromady vytvářejí určitou platformu. Organizační škálování rozděluje na horizontální, kde se snažíme agilní přístup zavést v dalších odděleních organizace, nejen do IT, ale i HR, Sales, Finance a dalších, a na vertikální, kde postupujeme hierarchicky podle úrovně v organizační struktuře.



Obrázek 1: Škálování agilních praktik přes více týmů (Buchalceová, 2018)

Agile Maturity Model patří do škálovatelných rámců, jelikož má vlastnosti škálovatelnosti. Můžeme jej aplikovat na tým o malé velikosti a tým v řádu desítek či stovek členů. Týmy o různé velikosti si mohou podle modelu vyhodnotit, na jaké úrovni se pohybují na základě vykonávaných praktik. Tyto praktiky jsou nezávislé na velikosti týmu.

3. Agile Maturity Model

Tato kapitola popisuje model zralosti zvaný Agile Maturity Model. Nejdříve jej stručně charakterizuje a kategorizuje podle Horlach et al. Následně vysvětlí jeho fungování včetně úrovní a oblastí, popíše použití, vyjmenuje přínosy, porovná s modelem CMMI a nakonec nabídne související studie.

3.1. Základní charakteristika

Agile Maturity Model je nástroj, jenž pomáhá společnostem identifikovat aktuální praktiky ve vývoji softwaru a zároveň je zlepšit s cílem zefektivnit agilní přístup. Společnost usilující o zlepšení agilního přístupu či provádějící agilní transformaci je podle modelu schopna zjistit stávající situaci praktik vývoje, respektive jejich úroveň, a soustředit se na praktiky z vyšší úrovně. Tímto společnost dokáže rychleji reagovat na měnící se požadavky zákazníků či trhu. Zároveň tím podpoří inovační cyklus produktů a služeb.

Tento model není typickým zástupcem škálovatelných agilních rámců. Přestože disponuje vlastnostmi, které mu dovolují škálovat od malých týmů na rozsáhlé mezinárodní týmy, jedná se o model zralosti. Modely zralosti slouží k posuzování schopnosti vykonávat určité činnosti a přesně tomu se Agile Maturity Model věnuje.

Model není obecný, zaměřuje se pouze na proces sestavování a nasazování softwaru. Tvůrci jej přizpůsobili na aktivity, které je potřeba vykonat po samotné implementaci řešení. Tyto činnosti zahrnují vytvoření spustitelného souboru, nasazení na zvolené prostředí, otestování a následné nasazení k zákazníkovi. Konkrétně v modelu uvádějí tyto oblasti činností:

- Build management and Continuous integration
- Environments and Deployment
- Release management and Compliance
- Testing
- Data management

“Tvrdíme, že tento proces by měl být celý vykonáván souvisle, nikoliv po částech samostatně. Cílem je vytvoření plně automatizovaného, spolehlivého, předvídatelného a transparentního procesu se srozumitelnými a kvantifikovatelnými riziky” (Humble a Russell, 2009). Tvůrci tedy zastávají názor, že výše uvedené oblasti činností mají tvořit jeden kompaktní celek. Tímto se zajistí efektivní a spolehlivý vývoj. Tvůrci popisují ideální stav agilního vývoje, kde každá změna v softwaru by měla projít tímto automatickým procesem. Takzvaná “deployment pipeline”, jak pojmenovávají tento proces, by měla vytvořit binární soubory, spustit na nich automatizované testy a relevantním osobám oznámit výsledek procesu (Humble a Russell, 2009).

Tabulka 1 ukazuje základní informace o Agile Maturity Modelu. Model byl publikován na webu společnosti ThoughtWorks Studios, jejíž tehdejší zaměstnanci Jez Humble a Rolf Russell

Název	Agile Maturity Model
Podnázev	Applied to Building and Releasing Software
Web modelu	https://info.thoughtworks.com/rs/thoughtworks2/images/agile_maturity_model.pdf
Datum vzniku	Září 2009
Autoři	Jez Humble, Rolf Russell
Kategorizace dle Horlach	Transformation Focus → Transformation Progress

Tabulka 1: Základní charakteristika Agile Maturity Modelu

tento model vytvořili. V současné době ho lze považovat za již starší, jelikož byl vytvořen v roce 2009. Od té doby nebyl aktualizován, ani dále rozšiřován.

3.2. Kategorizace dle Horlach et al.

Podle Horlach et al. (2018) lze model zařadit do kategorie Transformation Focus, konkrétně do Transformation Progress. Důvod je ten, že model posuzuje úroveň agilní transformace ve společnosti. Určuje, jak daleko společnost v agilní transformaci pokročila. Neposkytuje žádnou roadmapu, ani neříká jakým způsobem škálovat.

3.3. Popis modelu

Agile Maturity Model představuje tabulku, která je vertikálně rozdělena na 5 úrovní a horizontálně na 5 oblastí vývoje softwaru. V jednotlivých buňkách tabulky jsou uvedeny praktiky příslušící úrovni a oblasti. Model nedefinuje žádné role.

3.3.1. Úrovně

Vertikálně model uvádí úrovně zralosti od nejnižší -1, až po nejvyšší 3. Konkrétně jsou úrovně následující:

- Úroveň -1: Regressive

Tvůrci pojmenovávají počáteční úroveň jako Regresivní a označují ji číslem -1. Procesy se na této úrovni neopakují a jsou nedostatečně řízeny. Vývoj na této úrovni je chaotický.

- Úroveň 0: Repeatable

Procesy na nulté úrovni se začínají evidovat a dokumentovat. Zároveň dochází v oblastech vývoje k částečné automatizaci.

- Úroveň 1: Consistent

Na úrovni 1 už společnost aplikuje praktiky, jenž umožňují automatizaci procesů napříč všemi oblastmi procesu agilního vývoje softwaru.

- Úroveň 2: Quantitatively managed

Další úroveň začíná procesy měřit a řídit. Nastavují se různé ukazatele a metriky ve všech oblastech procesu.

- Úroveň 3: Optimizing

Poslední úroveň se zaměřuje na zlepšování. Jakmile společnost vyhodnocuje procesy pomocí stanovených metrik, může je lépe optimalizovat.

3.3.2. Oblasti

Horizontálně je tento model zralosti rozdělen do pěti oblastí. Tvůrci tímto chtěli zdůraznit, aby společnosti dávaly každé z nich patřičnou pozornost. Oblasti jsou následující:

Build management and Continuous integration

Model definuje tuto oblast jako "vytváření a udržování automatizovaného procesu sestavení aplikace, vykonání testů nad každou změnou a poskytnutí zpětné vazby nad tímto procesem celému týmu" (Humble a Russell, 2009).

Environments and Deployment

Prostředím se rozumí vybavení nezbytné pro funkci vyvíjecího softwaru. Jedná se například o hardware, infrastrukturu, počítačovou síť nebo externí služby. Na takové prostředí je potřeba software nasadit.

Release management and Compliance

Tato oblast se zabývá řízením nasazování softwarového produktu do zvoleného prostředí. Tvůrci k tomuto řízení přidali i oblast Compliance, jež zahrnuje dodržování pravidel stanovených zákonem, regulátorem nebo samotnou společností.

Testing

Testování je nezbytnou součástí jakéhokoliv vývoje softwaru. Tvůrci se v této oblasti soustředí především na testování spojené se sestavováním a vydáváním softwaru (např. na jednotkové testování).

Data management

Jako poslední oblast uvádějí Data management, tedy zpracování dat a jejich uložení v databázi. Tvůrci považují tuto oblast za nepostradatelnou, jelikož je často zdrojem problémů při nasazování.

Obrázek 2 ukazuje celou tabulku Agile Maturity Modelu podle Humbla a Russella (2009).

Practice	Build management and continuous integration	Environments and deployment	Release management and compliance	Testing	Data management
Level 3 - Optimizing: Focus on process improvement	Teams regularly meet to discuss integration problems and resolve them with automation, faster feedback, and better visibility.	All environments managed effectively. Provisioning fully automated. Virtualization used if applicable.	Operations and delivery teams regularly collaborate to manage risks and reduce cycle time.	Production rollbacks rare. Defects found and fixed immediately.	Release to release feedback loop of database performance and deployment process.
Level 2 - Quantitatively managed: Process measured and controlled	Build metrics gathered, made visible, and acted on. Builds are not left broken.	Orchestrated deployments managed. Release and rollback processes tested.	Environment and application health monitored and proactively managed. Cycle time monitored.	Quality metrics and trends tracked. Non functional requirements defined and measured.	Database upgrades and rollbacks tested with every deployment. Database performance monitored and optimized.
Level 1 - Consistent: Automated processes applied across whole application lifecycle	Automated build and test cycle every time a change is committed. Dependencies managed. Re-use of scripts and tools.	Fully automated, self-service push-button process for deploying software. Same process to deploy to every environment.	Change management and approvals processes defined and enforced. Regulatory and compliance conditions met.	Automated unit and acceptance tests, the latter written with testers. Testing part of development process.	Database changes performed automatically as part of deployment process.
Level 0 – Repeatable: Process documented and partly automated	Regular automated build and testing. Any build can be re-created from source control using automated process.	Automated deployment to some environments. Creation of new environments is cheap. All configuration externalized / versioned	Painful and infrequent, but reliable, releases. Limited traceability from requirements to release.	Automated tests written as part of story development.	Changes to databases done with automated scripts versioned with application.
Level -1 – Regressive: processes unrepeatable, poorly controlled, and reactive	Manual processes for building software. No management of artifacts and reports.	Manual process for deploying software. Environment-specific binaries. Environments provisioned manually.	Infrequent and unreliable releases.	Manual testing after development.	Data migrations unversioned and performed manually.

Obrázek 2: Agile Maturity Model (Humble a Russell, 2009)

3.4. Použití modelu

V popisu modelu zralosti Agile Maturity Model tvůrci zmiňují jeho použití ve společnostech velmi obecně. Uvádějí, že se používá jako interní nástroj ve společnosti ThoughtWorks, což je firma, kde pracovali oba dva autoři modelu. Zároveň tvrdí, že se používá i v jiných organizacích. Bohužel statistiky použití tohoto modelu v jiných organizacích nejsou známé a tak můžeme předpokládat, že tyto organizace jsou primárně klienty ThoughtWorks. Společnost ThoughtWorks

je globální lídr ve vývoji softwarových nástrojů pro agilní vývoj a také konzultační společnost. Proto lze předpokládat, že Agile Maturity Model doporučuje používat svým zákazníkům.

Model je kvůli své schopnosti škálovat použitelný pro malé startupy, kteří vyvíjejí produkt v malém měřítku, i pro mezinárodní společnosti, u kterých jsou činnosti během procesu vývoje velmi podobné. Avšak velikost společnosti při použití modelu hraje roli. Velké organizace nejsou schopny dosáhnout vyšší úrovně rychle. Provést změnu je u nich obtížné a trvá dlouho. Tvůrci uvádějí, že pokud se velká společnost pokusí provést změnu z počáteční úrovně na nejvyšší napříč všemi odděleními v jednom kroku, velmi pravděpodobně selže (Humble a Russell, 2009). Malé startupy mají výhodu, že mohou relativně rychle dosahovat vyšších úrovní, protože jsou hodně flexibilní.

Jez Humble a Rolf Russell popisují kromě samotného modelu i kroky, jak model používat v organizaci. V krátkém návodu vychází z Demingova cyklu Plan-Do-Check-Act. Pět kroků popisují následovně (Humble a Russell, 2009):

1. Určete aktuální praktiky v organizaci a podle modelu zralosti zjistěte, na jaké úrovni se nacházíte. Může se stát, že jedna nebo více oblastí bude mít jinou úroveň než ostatní.
2. Vyberte si, na jaké oblasti se zaměříte. Určete si, jakých praktik chcete dosáhnout, kolik to bude organizaci stát a jaké přínosy to pro ni ve výsledku bude mít. Také je potřeba si stanovit akceptační kritéria, aby jste zjistili, jestli změna byla přínosná.
3. Sestavte si plán provedení změn stanovených v minulém kroku. Následně změny proveďte.
4. Zkontrolujte přínosy provedených změn pomocí akceptačních kritérií.
5. Zopakujte tyto kroky. Aplikujte v nové iteraci nabitě zkušenosti. Inkrementálně vylepšujte skrze celou organizaci.

Společnost například zjistí, že nemá dostatečně zautomatizované sestavování buildu, vydávání nových verzí je stále nepravidelné, ovšem na druhou stranu oddělení ověřování kvality má všechny druhy testů automatizované. V takovém případě je v první a třetí oblasti na úrovni 0 a ve čtvrté na úrovni 1. Když se společnost rozhodne sjednotit tyto oblasti na úroveň 1, musí si vytvořit plán, aby dosáhla praktik první a třetí oblasti na úrovni 1. Tímto postupem může společnost zvýšit svou úroveň agility.

3.5. Přínosy

Přínosy tohoto modelu zralosti souvisí s přínosy agilního přístupu k vývoji softwaru. Samotní tvůrci však popisují několik přínosů zavedení Agile Maturity Modelu v organizaci (Humble a Russell, 2009).

- zkrácení času iterací (čímž je možné reagovat na požadavky zákazníků rychleji)
- snížení počtu chyb
- zvýšení předvídatelnosti celého procesu dodání softwaru
- schopnost reagovat na zákony a jiné požadavky regulátorů
- schopnost efektivně řídit rizika
- snížení nákladů díky lepšímu řízení rizik a nižšímu počtu chyb

3.6. Porovnání s modelem CMMI

Pokud se bavíme o Agile Maturity Modelu jakožto o modelu zralosti, je namístě ho srovnat se známým modelem Capability Maturity Model Integrated (CMMI). Ten předepisuje množiny praktik, jenž nazývá procesní oblasti, a posuzuje schopnost vykonávat tyto praktiky. Agile Maturity Model umožňuje organizacím porozumět svým aktuálním praktikám a zjistit jaké praktiky by měly aplikovat, aby byly schopny včasné reagovat na změnu trhu nebo aby zlepšily tvorbu inovací (Humble a Russell, 2009).

CMMI a Agile Maturity Model se liší i v názvech a číslování úrovní zralosti. Zatímco CMMI začíná od úrovně 1, model od Humbla a Russella začíná na úrovni -1. Jak ukazuje Tabulka 2

Agile Maturity Model	CMMI
Úroveň 3: Optimizing	Úroveň 5: Optimising
Úroveň 2: Quantitatively managed	Úroveň 4: Quantitatively managed
Úroveň 1: Consistent	Úroveň 3: Defined
Úroveň 0: Repeatable	Úroveň 2: Managed
Úroveň -1: Regressive	Úroveň 1: Initial

Tabulka 2: Porovnání Agile Maturity Modelu a CMMI (Schweigert et al., 2013)

vycházející z článku od Schweigerta et al. (2013), tak rovněž pojmenování třech nejnižších úrovní je jiné.

3.7. Související studie

Existuje několik studií zabývajících se agilními modely zralosti. Většinou se jedná o jejich porovnání, zjišťování jejich podobností a nebo o snahu najít jeden obecný model. Téměř každý z nich zmiňuje i Agile Maturity Model popisovaný v této práci, ovšem už jej podrobně nezkoumají jako ty ostatní.

Model zmiňuje případová studie hodnotící agilní modely zralosti (Ozcan-Top a Demirörs, 2013). Ta zhodnotila 9 agilních modelů zralosti, ovšem hned na začátku vyřadila mj. i model od Humbla a Russella. Důvodem je nedostatečný popis modelu pro provedení studie a také jeho nepublikování na jednu ze známých konferencí nebo do známých časopisů.

Další studií zmiňující tento model je Comparison of Agile Maturity Models (Schmitt, Theobald a Diebold, 2019). Studie identifikovala 28 agilních modelů zralosti, z nichž následně vybrala 14 na zhodnocení podle stanovených kritérií. Do úzké skupiny se dostal i model od Humbla a Russella. Výsledky studie ovšem nepopisují konkrétní výsledky pro tento model. Pouze se omezují na pár tvrzení směrem k němu. Například, že model zahrnuje behaviorální praktiky jako je spolupráce členů. Více informací nelze k modelu zjistit, jelikož studie výsledky kritérií nepřirazuje ke konkrétním modelům.

4. Závěr

Tato semestrální práce nabízí strukturovaný popis modelu zralosti Agile Maturity Model od autorů ze společnosti ThoughtWorks Studios. Hlavní cíl tedy naplnila. Kromě popisu vysvětluje fungování modelu včetně úrovní a oblastí. Definuje postup, který v několika krocích popisuje využití modelu při vývoji softwaru v různých organizacích. Dále vyjmenovává přínosy modelu, popisuje rozdíly oproti podobným modelům zralosti v agilním prostředí. Konkrétně je srovnán s modelem CMMI.

Vzhledem k tomu, že Agile Maturity Model se od svého vzniku nerozšířil, byla jediným zdrojem, který ho popisuje počáteční publikace autorů. Tato publikace model stručně charakterizuje, avšak kvůli jeho malému rozsahu nenabízí detailnější popis. Z toho důvodu je model ve studii sice uváděn, ale není podrobně analyzován. Následně se tato skutečnost projevila v této práci, která nemůže poskytnout detailní statistiky a srovnání s ostatními agilními modely zralosti popř. jinými modely.

5. Literatura

BECK, Kent et al., 2001. *Agilní manifest* [online]. [cit. 2020-05-20]. Dostupné z: <https://agilemanifesto.org/>

BUCHALCEVOVÁ, Alena. *Zlepšování procesů při budování informačních systémů*. Praha: Oeconomica, 2018. ISBN 978-80-245-2235-7.

BUCHALCEVOVÁ, Alena. *Škálování agilních metodik* [přednáška]. Praha: VŠE, 2020.

HORLACH, Bettina et al., 2018. *IT Governance in Scaling Agile Frameworks* [online]. [cit. 2020-05-20]. Dostupné z: https://www.researchgate.net/publication/323685823_IT_Governance_in_Scaling_Agile_Frameworks

HUMBLE, Jez a Rolf RUSSELL, 2009. *The Agile Maturity Model: Applied to Building and Releasing Software* [online]. [cit. 2020-05-20]. Dostupné z: https://info.thoughtworks.com/rs/thoughtworks2/images/agile_maturity_model.pdf

OZCAN-TOP, Ozden a Onur DEMIRÖRS. *Assessment of Agile Maturity Models: A Multiple Case Study* [online]. In: *Communications in Computer and Information Science*, 2013 [cit. 2020-05-20]. Dostupné z: https://www.researchgate.net/publication/255483624_Assessment_of_Agile_Maturity_Models_A_Multiple_Case_Study

SCHMITT, Anna, Sven THEOBALD a Philipp DIEBOLD. *Comparison of Agile Maturity Models* [online]. 2019 [cit. 2020-05-20]. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-030-35333-9_52

SCHWEIGERT, Tomas et al., *Agile Maturity Model: A Synopsis as a First Step to Synthesis* [online]. Berlin, Heidelberg, 2013 [cit. 2020-05-20]. Dostupné z: https://link.springer.com/chapter/10.1007/978-3-642-39179-8_19