

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra systémového inženýrství**



**Diplomová práce**

**Návrh tvorby zadání pro vývojový tým v rámci životního  
cyklu produktu ve společnosti Unicorn Solutions a.s.**

**Bc. Michaela Skala**

© 2021 ČZU v Praze



# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Michaela Skala

Systemové inženýrství a informatika  
Projektové řízení

Název práce

**Návrh tvorby zadání pro vývojový tým v rámci životního cyklu produktu ve společnosti Unicorn Solutions a.s.**

Název anglicky

**Proposal for creating assignments for development team within the product life cycle at Unicorn Solutions a.s.**

---

### Cíle práce

Cílem diplomové práce je návrh tvorby zadání pro vývojový tým v rámci životního cyklu softwarového produktu ve společnosti Unicorn Solutions a.s. se zaměřením na předání zadání mezi týmem zadavatele a týmem dodavatele. Součástí diplomové práce je vlastní případová studie z praxe.

### Metodika

Teoretická část zahrnuje popis životních cyklů v organizaci, konkrétně životní cyklus týmu, produktu a softwaru, hodnotící kritéria využívaná k hodnocení úspěšnosti projektů. Dále popisuje předávání informací mezi týmy zákazníka a dodavatele, s důrazem na řízení komunikace a odpovědností.

Praktická část práce bude věnována popisu dělby práce v holdingu Unicorn a.s. včetně stávajícího stavu řízení životního cyklu produktu a vlastnímu návrhu tvorby zadání pro vývojový tým. Také komunikaci mezi týmem zadavatele (produktový tým) a dodavatele (vývojářský tým) v rámci životního cyklu produktu ve společnosti Unicorn Solutions a.s. a řízení spolupráce pomocí aplikací společnosti Unicorn Solution a.s.. Dále bude praktická část obsahovat vlastní návrh, tj. uplatnění navrženého teoretického konceptu, pro vybraný reálný produkt (případová studie). Vlastní návrhy budou průběžně prezentovány a diskutovány v praxi.

**Doporučený rozsah práce**

60 – 80 stran

**Klíčová slova**

Business, user story, business case, plánování, životní cyklus, aplikace, produkt, tým, spolupráce, zadavatel, dodavatel, Unicorn, vývoj aplikace

---

**Doporučené zdroje informací**

- AMBLER, S W. – NALBONE, J. – VIZDOS, M J. *The enterprise unified process : extending the rational unified process*. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference, 2005. ISBN 0131914510.
- GÁLA, L. – POUR, J. – ŠEDIVÁ, Z. – ČESKÁ SPOLEČNOST PRO SYSTÉMOVOU INTEGRACI. *Podniková informatika*. Praha: Grada, 2009. ISBN 978-80-247-2615-1.
- JAKUBÍKOVÁ, D. *Strategický marketing : strategie a trendy*. Praha: Grada, 2013. ISBN 978-80-247-4670-8.
- KOTLER, P. – ARMSTRONG, G. *Marketing*. Praha: Grada, 2004. ISBN 80-247-0513-3.
- KOVÁŘ, Vladimír, et al. Unicorn ES Powered Company – Management. 1. revidované vydání. Praha: Unicorn College, 2009. 120 s. ISBN 978-80-87349-01-4.
- KOVÁŘ, Vladimír, et al. Unicorn ES Powered Company – Strategie. 3. přepracované a rozšířené vydání. Praha: Unicorn College, 2009. 48 s. ISBN 978-80-87349-00-7.
- KOVÁŘ, Vladimír. Unicorn Enterprise System Powered Company : Metodika pro řízení podniku a organizací s přímou podporou informačního systému. [s.l.], 2011. 130 s. Dizertační práce. Univerzita Hradec Králové, Fakulta informatiky a managementu, Katedra informatiky a kvalitativních metod.
- SVOZILOVÁ, A. *Projektový management : systémový přístup k řízení projektů*. Praha: Grada, 2016. ISBN 978-80-271-0075-0.
- SVOZILOVÁ, A. *Zlepšování podnikových procesů*. Praha: Grada, 2011. ISBN 978-80-247-3938-0.
- ŠOCHOVÁ, Z. – KUNCE, E. *Agilní metody řízení projektů*. Brno: Computer Press, 2019. ISBN 978-80-251-4961-4.
- 

**Předběžný termín obhajoby**

2020/21 LS – PEF

**Vedoucí práce**

doc. Ing. Jan Bartoška, Ph.D.

**Garantující pracoviště**

Katedra systémového inženýrství

Elektronicky schváleno dne 19. 1. 2021

**doc. Ing. Tomáš Šubrt, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 22. 1. 2021

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 24. 01. 2021

### **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci "Návrh tvorby zadání pro vývojový tým v rámci životního cyklu produktu ve společnosti Unicorn Solutions a.s." jsem vypracovala samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne 29.3.2021

---

## **Poděkování**

Ráda bych tímto poděkovala doc. Ing. Janu Bartoškovi, Ph.D., za jeho ochotu a vstřícnost, se kterou se mi během psaní mé diplomové práce věnoval, kolegům za konzultace praktické části mé diplomové práce a trpělivost. Ráda bych poděkovala i rodičům za celoživotní motivaci, péči a pozornost, kterou mi věnovali. Manželovi děkuji za podporu a pochopení. Dále bych chtěla poděkovat všem, kteří se mnou v průběhu mého studia a práce spolupracovali.

Stejně tak děkuji své rodině a přátelům, kteří mě v mém studiu podporovali.

# Návrh tvorby zadání pro vývojový tým v rámci životního cyklu produktu ve společnosti Unicorn Solutions a.s.

## Abstrakt

Cílem práce je vytvoření návrhu zadání pro vývojový tým, včetně určení jednotlivých kroků a odpovědných osob, ve společnosti Unicorn Solutions a.s. v rámci metodiky společnosti Unicorn Universe Process. Teoretická část obsahuje popis životních cyklů, jež ovlivňují životní cyklus softwarového produktu. Dále definuje komunikaci, odpovědnost a jejich řízení, metodiky a přístupy k vývoji softwaru a akceptační kritéria. V praktické části je popsána dělba práce v rámci holdingu Unicorn a.s., základní myšlenky přístupu k vývoji softwaru v holdingu, komunikace uvnitř holdingu a životní cyklus softwarového produktu ve společnosti Unicorn Solutions a.s. Proces tvorby zadání byl rozdělen do tří dílčích procesů, tvorba uživatelských příběhů (user story), návrh obchodního model (business model) a návrh změn aplikačního modelu (app model). Práce definuje akceptační kritéria pro výstupy dílčích procesů, kterým zadání musí odpovídat, aby bylo dodavatelskou firmou akceptováno a mohla být zahájena realizace zadání. Výstupy práce byly průběžně konzultovány v praxi.

**Klíčová slova:** Business, user story, business case, plánování, životní cyklus, aplikace, produkt, tým, spolupráce, zadavatel, dodavatel, Unicorn, vývoj aplikace

# Proposal for creating assignments for development team within the product life cycle at Unicorn Solutions a.s.

## **Abstract**

The aim of the thesis is to create a proposal of assignments for the development team, including the determination of individual tasks and the responsible persons for them, within the company Unicorn Solutions a.s. using the company methodology, Unicorn Universe Process. The theoretical part contains descriptions of other life cycles that affect the life cycle of a software product. It also describes communication, responsibility, and their management, plus methodologies and approaches to software development and acceptance criteria. The practical part explains the division of labor within the holding, Unicorn a.s., the basic ideas of the approach to software development in the holding, communication within the holding, and the life cycle of the software product in the company Unicorn Solutions a.s. The creation of the assignment is divided into three sub-processes, creating a user story, designing a business model, and proposing changes to the application model. The thesis defines the acceptance criteria for the outputs of the sub-processes, to which the assignment must correspond, to be accepted by the supplier company, and implementation of the assignment can begin. All steps in the creation of the assignment were continuously consulted in practice.

**Keywords:** Business, user story, business case, planning, life cycle, application, product, team, collaboration, client, supplier, Unicorn, application development



# Obsah

<b>1 Úvod</b> .....	<b>17</b>
<b>2 Cíl práce a metodika</b> .....	<b>19</b>
2.1 Cíl práce.....	19
2.2 Metodika.....	20
<b>3 Teoretická východiska</b> .....	<b>21</b>
3.1 Životní cykly v organizaci.....	21
3.1.1 Životní cyklus týmu.....	21
3.1.2 Životní cyklus produktu.....	22
3.1.3 Životní cyklus softwaru.....	24
3.2 Metodiky a přístupy k vývoji softwaru.....	27
3.2.1 Vodopádový model.....	27
3.2.2 Prototypový model.....	28
3.2.3 Model spirála.....	29
3.2.4 Agilní přístup.....	31
3.3 Komunikace.....	34
3.3.1 Model komunikace.....	34
3.3.2 Řízení komunikace.....	36
3.4 Odpovědnost.....	37
3.4.1 Řízení odpovědnosti.....	37
3.5 Akceptační kritéria a hodnoticí kritéria.....	39
3.5.1 Model KKTR (kvalita, kvantita, termín, rozpočet).....	39
3.5.2 Model “CRaB QueST” (Cost, Risk, Benefits, Quality, Scope and Time).....	40
3.5.3 Metoda OKR (Objectives and Key Results).....	41
<b>4 Vlastní práce</b> .....	<b>42</b>
4.1 Unicorn a.s.....	42
4.1.1 Dělbba práce v holdingu Unicorn a.s.....	45
4.1.2 Produkty Unicorn a.s.....	52
4.1.3 Unicorn Solution a.s.....	53
4.2 Unicorn Approach.....	56
4.2.1 Metodika Unicorn Universe Process – uuP.....	58
4.2.2 Unicorn Application Framework.....	65
4.3 Životní cyklus uuApp.....	67
4.3.1 Model životního cyklu uuApp.....	67
4.3.2 Komunikace mezi dodavatelskými týmy a zadavatelským týmem Unicorn Solutions.....	77

4.4	Životní cyklus vývoje uuApp.....	79
4.5	Proces tvorby zadání v Unicorn Solutions a.s. v období říjen 2020 .....	81
4.6	Návrh procesu tvorby zadání požadavku na implementaci nové funkcionality uuApp ve společnosti Unicorn Solution a.s. ....	82
4.6.1	Cíl procesu.....	84
4.6.2	Akceptační kritéria zadání .....	85
4.6.3	Účastníci procesu.....	89
4.6.4	Vstupy procesu .....	89
4.6.5	Výstupy procesu .....	89
4.6.6	Proces tvorba user story.....	90
4.6.7	Proces úprava business model .....	94
4.6.8	Proces návrh úprav app model.....	96
4.7	Výsledky a diskuse .....	100
4.7.1	Komunikace.....	100
4.7.2	Návrh tvorby zadání .....	101
<b>5</b>	<b>Závěr .....</b>	<b>103</b>
<b>6</b>	<b>Přílohy.....</b>	<b>109</b>

## Seznam obrázků

Obrázek 1 Životní cyklus produktu [7] .....	23
Obrázek 2 Vodopádový model životního cyklu softwaru [11] .....	28
Obrázek 3 Schéma prototypového modelu [10] .....	29
Obrázek 4 Schéma modelu spirála [15] .....	30
Obrázek 5 Myšlenková mapa scrum pojmů [Vlastní zpracování podle [17]] .....	32
Obrázek 6 Model procesu komunikace podle Donelly, Gibson, Konopaske a Ivancevich [19] .....	34
Obrázek 7 Struktura společností holdingu [9] .....	43
Obrázek 8 Dělbá práce v holdingu [32] .....	45
Obrázek 9 Organizační struktura SWF [33] .....	47
Obrázek 10 Schéma komunikace mezi vývojovým týmem SWF a zadavatelem [34] .....	50
Obrázek 11 Společnosti patřící do centrálních jednotek [36] .....	52
Obrázek 12 Rozdělení produktů holdingu Unicorn [32] .....	52
Obrázek 13 Organizační struktura Unicorn Solutions a.s. [32] .....	54
Obrázek 14 Produktový tým (product team) [37] .....	56
Obrázek 15 Pravidlo 31 [41] .....	61
Obrázek 16 Schéma vazeb mezi Unicorn Approach, Unicorn Universe, Plus4U, uuP a uuTerritory [42] .....	62
Obrázek 17 Schéma uuBusinessTerritory [43] .....	63
Obrázek 18 Schéma uuMyTerritory [44] .....	63
Obrázek 19 Základní vlastnosti artefaktu [45] .....	64
Obrázek 20 Schéma vazby mezi artefaktem, aktivitami a elementárními aktivitami [45] .	65
Obrázek 21 Jednotlivé logické vrstvy [46] .....	66
Obrázek 22 Řešení problému z pohledu vlastníka problému [48] .....	67
Obrázek 23 Řešení problému implementací uuApp [9] .....	68
Obrázek 24 Pět plus jedna potřebných znalostí uuApp [49] .....	69
Obrázek 25 Etapy vývoje uuApp a iterace [48] .....	72
Obrázek 26 Pět procesů souvisejících s informačním systémem [9] .....	76
Obrázek 27 Schéma komunikace mezi vývojovými týmy SWF a zadavatelem product team [34] .....	79

Obrázek 28 uuBml schéma procesu tvorby zadání [Vlastní zpracování].....	84
Obrázek 29 uuBml schéma procesu tvorby user story [Vlastní zpracování].....	90
Obrázek 30 BPMN diagram tvorby user story [Vlastní zpracování] .....	91
Obrázek 31 uuBml schéma procesu úpravy business model [Vlastní zpracování] .....	95
Obrázek 32 BPMN diagram úpravy business model [Vlastní zpracování].....	96
Obrázek 33 uuBml schéma procesu návrhu app model [Vlastní zpracování].....	98
Obrázek 34 BPMN diagram návrhu úprav app model [Vlastní zpracování].....	99

## Seznam tabulek

Tabulka 1 Příklad matice RASCI [Vlastní zpracování] .....	38
Tabulka 2 Matice RASCI pro návrh procesu tvorby zadání [Vlastní zpracování].....	101

## Abecední seznam použitých zkratk a slovníček

Acceptance meeting .....	schůzka za cílem akceptace dodaných výstupů
Accountability .....	povinnost zodpovídat
Actor .....	typová role
AM.....	application model
App model .....	application model, aplikační model
Approver.....	schvalovatel
BA .....	business architect
Backlog.....	seznam úkolů, myšlenek, položek určitého typu
BM.....	business model
Board .....	členové představenstva společnosti
BPMN.....	Business Process Modeling Notation, mezinárodní grafický standard pro vizualizaci procesů
BUC.....	business use-case, obchodní případ užití
Business architect .....	obchodní analytik, analyzuje a navrhuje systém řízení podniku a v jeho rámci podnikové procesy, analytické, plánovací a rozhodovací aktivity
Business discipline .....	obchodní disciplína, definuje obchodní odvětví a je .....
.....	vytvořena za konkrétním účelem
business model .....	obchodní model
Business use-case .....	případy užití z obchodního pohledu
Consulted.....	konzultováno
Contributor .....	příspěvatel
Development team.....	vývojářský tým
Driver.....	řidič, osoba odpovědná za koordinaci aktivit vedoucích k rozhodnutí
DTP .....	diagram toku produktů
estimation meeting .....	odhadová schůzka, na této schůzce se připravuje či prezentuje scope (rozsah, kvantita) daného sprintu formou sprint backlog z product backlog
GDPR .....	obecné nařízení o ochraně osobních údajů
IDEF .....	Integrated DEFinition Methods, rodina modelovacích jazyků v oblasti systémů a softwarového inženýrství

Informed .....	informovaný
Integration.....	integrace
IoT .....	Internet of Things, internet věcí
IoT Smart Box .....	chytrý box napojený na internet věcí
ITIL.....	Information Technology Infrastructure Library, soubor praxí prověřených konceptů a postupů
Non-functional requests.....	požadavky, které neobsahují konkrétní funkčnosti
Objectives and Key Results.....	cíle a klíčové výsledky
OKR.....	Objectives and Key Results
onClick uuApp.....	aplikace, která je dostupná na kliknutí a vyznačuje se jednoduchou instalací
PM .....	product manager
First line support.....	primární podpora
PRINCE .....	metodika pro projektové řízení
Product backlog .....	produktový backlog, seznam požadavků na produkt
Product manager .....	produktový manažer
Product supervisor .....	dozor produktového týmu a produktu
Product team .....	produktový tým
Operation .....	provoz
PS.....	product supervisor
Qualified Users .....	kvalifikovaný uživatel
Reengineering.....	revize a obnova
Responsibility .....	odpovědnost
RUP .....	Rational Unified Process je metodika vývoje softwaru vytvořená společností Rational Software Corporation
SCRUM .....	iterativní a inkrementální způsob řízení vývoje softwaru
scrum master.....	vlastník procesu Scrum
Second line support .....	sekundární podpora
SLA.....	Service Level Agreement, je termín využívaný pro smlouvu sjednanou mezi poskytovatelem služby (uuApp aplikace) a uživatelem.
SP.....	struktura produktu

sprint.....	základní jednotka metodiky Scrum, časově ohraničené období, které je dopředu pevně dáno
sprint backlog .....	seznam všech úkolů, na kterých pracuje tým během sprintu
Stakeholders .....	zajímavé osoby
SWF.....	Unicorn Software Factory a.s.
TA.....	technology architect
team leader .....	vedoucí týmu
Technology architect .....	technologický architekt
Test team .....	tým testerů
The Architecture.....	cloudová architektura (také jako Unicorn Mobile-First IoT-Ready Cloud Architecture nebo Unicorn Architecture)
Top Management Office .....	management společnosti
UAF .....	Unicorn Application Framework
UML .....	Unified Modeling Language
Unicorn RL Centre .....	Unicorn Research and Learning center
US.....	user story
user story .....	uživatelský příběh
USO .....	Unicorn Solutions a.s.
uu .....	Unicorn Universe
uuApp .....	uuApp je aplikace vytvořená pomocí uuApp Framework. Spouští se v uuCloud: Používá identitu Plus4U pro autentizaci a uuBusinessTerritory pro autorizaci.
uuApp designer.....	designér uuApps podle standardů metodiky uuP a UAF
uuApp developer .....	vývojář uuApps, který tvoří uuApps podle návrhů uuApp designer
uuApp Patterns .....	soubor rad, doporučení a pravidel pro tvorbu uuApp
uuApp tester .....	tester uuApp
uuBML .....	Unicorn Unified Business Modeling Language je nástroj pro vizuální modelování a komunikaci
uuP.....	metodiky Unicorn Universe Process
uuSprintMan .....	aplikace pro správu sprintu

UX ..... User Experience

Building up ..... vývoj

Wishlist ..... seznam přání, možných budoucích funkcí



# 1 Úvod

Softwarové programy (aplikace) hrají v dnešním světě nezastupitelnou roli. Téměř každý dospělý jedinec má mobilní telefon s připojením k internetu. Stalo se standardem mít mailovou schránku a každý den nás obklopují technologie. Díky světové pandemii COVID-19 v roce 2020 se stalo téměř povinností mít v domácnosti počítač s připojením k internetu. Součástí mobilních i počítačových zařízení jsou aplikace, softwarové produkty. Tato práce popisuje proces zadání pro vývojový tým, který tyto aplikace vytváří.

Na začátku životního cyklu každého softwarového produktu je zadavatel, který je přesvědčen, že díky zavedení softwaru bude vyřešena nějaká jeho potřeba či problém, se kterým se potýká. Pro naplnění tohoto přesvědčení je nutné vhodně popsat problém, definovat požadovaný výsledný stav a očekávání zadavatele. Autorka se s touto problematikou v praxi již vícekrát setkala, poprvé jako členka týmu dodavatele, což bylo motivací pro výběr studijního oboru Projektové řízení. Během studia se zabývala tématem Řízení rizik ve společnosti Unicorn Software Factory a.s. Při rozhovorech se zaměstnanci zjistila, že velmi častou příčinou zpoždění vývoje je nedostatečná specifikace zadání ze strany zadavatele. Autorka v současnosti v praxi vystupuje v roli zadavatele a práci zabývající se návrhem tvorby zadání pro vývojový tým v rámci životního cyklu softwarového produktu ve společnosti Unicorn Solution a.s. si vybrala s cílem rozšířit své know-how v této oblasti s následnou implementací v praxi ve firmě Unicorn Solutions. a.s. a zvýšit efektivitu vývoje díky dostatečné specifikaci zadání.

Vývoji softwarového produktu se většinou věnují týmy specialistů, jedná se tedy o týmovou práci. Pro pochopení vztahů uvnitř týmu je v kapitole Životní cyklus týmu popsán vývoj týmu v čase, komunikace mezi jednotlivými členy a přijetí odpovědnosti. Vývoj softwaru je rychle rostoucí oblast, která se stále posouvá, mění a vyvíjí.

Diplomová práce obsahuje náhled do struktury holdingu Unicorn a.s., již je Unicorn Solutions a.s. součástí. Diplomová práce poskytuje vhled do strategického přístupu vedení k dělbě práce v rámci holdingu a přístup k řízení vnitřních procesů pomocí vlastních metodik, pracovního rámce a architektury. Metodika Unicorn Universe Process je vlastní

metodika Unicornu, jež je inspirována mezinárodními standardy, metodikami, doporučeními, mezi které patří ITIL, RUP, PRINCE, SCRUM, IDEF a dlouhodobé vlastní zkušenosti.

Očekávaným výsledkem práce je návrh procesu tvorby zadání pro implementaci nové funkčnosti do aplikací.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Cílem diplomové práce je návrh tvorby zadání pro vývojový tým v rámci životního cyklu softwarového produktu ve společnosti Unicorn Solutions a.s. se zaměřením na předání zadání mezi týmem zadavatele a týmem dodavatele. Součástí diplomové práce je vlastní případová studie z praxe.

## 2.2 Metodika

Teoretická část zahrnuje popis životních cyklů v organizaci, konkrétně životní cyklus týmu, produktu a softwaru, hodnotící kritéria využívaná k hodnocení úspěšnosti projektů. Dále popisuje předávání informací mezi týmy zákazníka a dodavatele s důrazem na řízení komunikace a odpovědností.

Praktická část práce bude věnována popisu dělby práce v holdingu Unicorn a.s. včetně stávajícího stavu řízení životního cyklu produktu a vlastnímu návrhu tvorby zadání pro vývojový tým, také komunikaci mezi týmem zadavatele (produktový tým) a dodavatele (vývojářský tým) v rámci životního cyklu produktu ve společnosti Unicorn Solutions a.s. a řízení spolupráce pomocí aplikací společnosti Unicorn Solution a.s. Dále bude praktická část obsahovat vlastní návrh, tj. uplatnění navrženého teoretického konceptu pro vybraný reálný produkt (případová studie). Vlastní návrhy budou průběžně prezentovány a diskutovány v praxi.

## 3 Teoretická východiska

### 3.1 Životní cykly v organizaci

Tato kapitola popisuje cykly ovlivňující vývoj softwarového produktu, konkrétně životní cyklus týmu, produktu a softwaru (informačního systémů), jenž souvisí s návrhem zadání pro vývojový tým v rámci životního cyklu softwarového produktu.

Životní cyklus obecně je způsob popisu vývoje entity v čase. Model životního cyklu popisuje vzájemný vývoj veličin během průběhu cyklu. Cyklus je sled po sobě jdoucích fází.

#### 3.1.1 Životní cyklus týmu

Produkt a jeho vývoj je týmová práce. Stejně jako softwarový produkt i tým složený z lidí má své fáze vývoje, můžeme mluvit o životním cyklu. Jedním z psychologů, kteří tuto problematiku popsali, je Bruce Tuckman, jenž ji popsal ve svém článku „Developmental sequence in small groups“, který byl zveřejněn v časopisu Psychological bulletin roku 1965.

Tuckam, Tuckamův model, rozděluje životní cyklus týmů do fází:

- Forming, neboli fáze formování skupiny,
- Storming, krystalizace (bouření),
- Norming, ustálení skupiny (normování),
- Performing, optimální výkon.

V roce 1977 přidal další fázi Adjourning, ukončení. [1]

Ve fázi **formování skupiny** se členové týmu seznamují navzájem anebo s novými úkoly, rolemi. Členové týmu ještě nemusí být v souladu s novými odpovědnostmi a cíli týmu. Je potřeba objasnit smysl vzniklého týmu, stanovení úkolů a cílů. Typicky se jedná o fázi, ve které je potřeba hodně trpělivosti, vysvětlování, tvorby bezpečného prostředí a budování důvěry.

Po fázi formování skupiny, kdy se členové seznamovali navzájem a zároveň s cíli skupiny a jednotlivými úkoly, nastává **fáze krystalizace**, kdy může docházet k napětí ve skupině. Každý člen má názor a právo tento názor vyjádřit. Jsou si blíže a komunikační hranice se posunuly. Mohou nastat výměny názorů, konflikty a vzájemná nespokojenost.

Jedná se o kritickou fázi pro tým a jeho životaschopnost nebo případný rozpad. Úlohou manažera je tým spojit a vést společně ke zdárnému cíli.

Během **fáze normování** dochází k pochopení cílů, norem, procesů a nastavení komunikace, vzniká skupinová soudržnost. Skupina se stabilizuje. [2]

Členové si osvojili své role, odpovědnosti, týmový cíl, normy a procesy. Ve fázi **optimální výkon** je tým nejvýkonnější, všichni vědí, co mají dělat, případně s kým se poradit. Mají týmovou motivaci, jdou za společným cílem.

Po naplnění cíle, pro niž tým vznikl, může tým během **fáze ukončení** zaniknout. Typickým případem je ukončení projektu.

Mezi další teorie patří teorie Cogův žebřík, Wheelenův model, Riegerův model.

### 3.1.2 Životní cyklus produktu

*Produkt/výrobek můžeme definovat jako vše, co můžeme nabídnout ke koupi, k použití, ke spotřebě – co uspokojuje PPO (potřeby, požadavky, očekávání) potenciálních a stávajících zákazníků. [3]*

*Produkt je položka nabízená k prodeji. Produktem může být služba nebo položka. Může to být ve fyzické, virtuální nebo kybernetické podobě. Každý produkt má výrobní cenu a každý produkt je prodán za prodejní cenu. Prodejní cena, kterou lze účtovat, závisí na trhu, kvalitě, marketingu a cíleném segmentu. Každý produkt má životnost, po které je třeba jej vyměnit, a životní cyklus, po kterém musí být znovu vynalezen. Značku lze předělat, znovu spustit nebo rozšířit, aby byla relevantnější pro daný segment a dobu, přičemž produkt bude často téměř totožný. [4]*

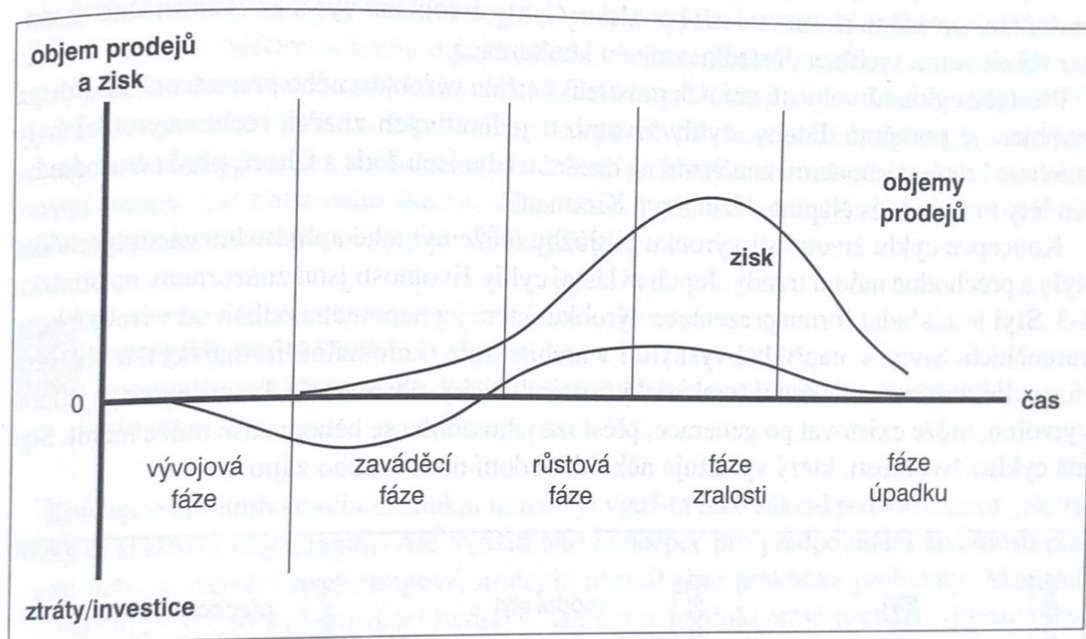
*Souhrn všech znaků/charakteristik produktu/služby, které ovlivňují schopnost uspokojit stanovené a předpokládané potřeby zákazníka projektu. [5]*

**Životní cyklus produktu** je úzce spjat s marketingovým a strategickým řízením. Většinou se graficky znázorňuje jako závislost mezi časem a objemem prodejů (zisku). [6]. Je to sled fází, které provází produkt během jeho „života“. Jedná se pouze o teoretický model, který se vždy přizpůsobuje produktu podle jeho potřeb a délky fází se zpravidla liší. Slouží nám k plánování strategií daného produktu od vývoje, přes prodej až po ukončení podpory.

### Fáze životní cyklu produktu:

- vývojová fáze,
- zaváděcí fáze,
- růstová fáze,
- fáze zralosti produktu,
- fáze úpadku.

Obrázek 1 Životní cyklus produktu [7]



Obrázek 9-2 Vývoj objemu prodeje a ziskovosti během cyklu životnosti produktu

Počátkem životního cyklu produktu je **vývojová fáze**. Je reakcí podniku na průzkum trhu. Během průzkumu trhu bylo zjištěno, že zavedením produktu bude vyřešen problém zákazníka. Podnik tedy tento produkt připraví. Z hlediska povědomí o produktu je vhodné zahájit marketingovou činnost již při vývoji produktu. Dnešní doba jde tomuto faktu naproti. Lidé mají rádi příběhy a u prémiových produktů chtějí být součástí příběhu „svých“ produktů již od počátku. U mnoha start-upových produktů můžeme pozorovat, že firma začíná s marketingem již během vývoje produktu. Příkladem je společnost Tesla, která předprodejem zafinancovala část vývoje. [8] Podobným vývojem prošla většina produktů, které vybírají peníze od koncových uživatelů na portálech skupinového financování (crowdfunding) jako Kickstarter, Indiegogo, Startovač, Crowd Supply, HitHit a další. V **zaváděcí fázi** se výrobek uvádí na trh, postupně se dostává do povědomí zákazníků

a buduje si reputaci. Začíná generovat první příjmy. Plánuje se marketingová strategie. V této fázi je vysoké riziko neúspěchu, zákazníci nemusejí produkt přijmout.

Během **růstové fáze** je již výrobek nějakou dobu na trhu, lidé o něm mluví, doporučují si ho a zisky se zvyšují. Stále je potřeba investovat do marketingu a budovat povědomí o výrobku. Konkurence začíná vytvářet podobný produkt, který řeší zákazníkův problém. Je třeba reagovat inovací a vylepšeními.

V první části **fáze zralosti produktu** zisk stále roste, poptávka začíná být saturovaná. Produkt je dobře zaveden, veřejnost o něm ví. Konkurence se zvyšuje. V druhé části již zisky klesají. Je třeba přemýšlet o inovacích, přijít se způsobem, jak prodloužit životní cyklus produktu, například vylepšením výrobku, novou funkčností, snížením nákladů => snížením ceny, nabídnutím produktu v balíčku s jiným produktem, vstupem na nový trh (jiný geografický trh, jiný segment stávajícího trhu...), nalezením nové přidané hodnoty a další.

Ve **fázi úpadku** je trh zcela saturován, poptávka je uspokojena, a klesá tedy množství prodaných kusů. Dochází ke snížení objemu výroby, což vede ke zvýšení jednotkových nákladů a růstu ceny. Firma se musí rozhodnout, zda prodlouží životní cyklus výrobku (např. inovací) nebo výrobek stáhne z trhu. [7]

### 3.1.3 Životní cyklus softwaru

Životní cyklus softwaru z produktového pohledu je tvořen životními fázemi mapujícími jeho realizaci jako produktu od počátku (analýza) po jeho úpadek, zánik. Podle různých autorů a přístupů má životní cyklus různý počet fází, které mají různou délku a posloupnost.

V literatuře najdeme různé modely a metodiky přístupu k životnímu cyklu vývoje. Je vhodné uvést, že životní cyklus vývoje softwaru se liší od životního cyklu softwaru. Životní cyklus vývoje softwaru končí momentem předání softwaru zákazníkovi. Životní cyklus softwaru po této fázi pokračuje a zabývá se i správou softwaru až do konce jeho životního cyklu jako produktu.

**Životní cyklus softwaru** lze rozdělit na fáze:

- tvorba cílů, vizí a strategií softwaru,
- návrh, vývoj a testování,
- zavádění systému do provozu,
- provoz a servis,
- zánik nebo reengineering.



Během fáze tvorba cílů, vizí a strategií se definují přínosy implementovaného softwaru a jeho cíle. Vstupní analýzy identifikují problémy a z nich vyplývající požadavky na systém. Stanoví se strategie realizace tvorby softwaru. [8]

Požadavky mohou být funkční (jaké funkce má software mít, jaké procesy má automatizovat a jak), nefunkční (vlastnosti softwaru, podmínky fungování, předpokládaný počet uživatelů, bezpečnostní prvky).

Vladimír Kovář definuje **1+10 vlastností dobrého systému**. Systém musí být dostatečně:

1. **výkonný**, aby zajistil požadované úlohy dostatečně rychle pro celou cílovou skupinu, která je předem definována,
2. **spolehlivý**, tedy podávat při stejných situacích stejné výsledky za předpokladu shodných vstupů,
3. **ekonomický**, užitek ze systému musí převyšovat náklady na jeho údržbu a implementaci,
4. **ergonomický**, uživatelsky přívětivý, intuitivní, zejména musí sloužit uživatelům nikoliv jen sám sobě,
5. **stabilní**, v kritických situacích musí být systém spolehlivý, tzn. reagovat očekávaným způsobem a nesmí být ohrožena data,
6. **nezávislý**, vnější vlivy a změna ve vnějším prostředí nesmí ovlivnit či ohrozit fungování systému,
7. **rozšiřitelný**, systém musí být navržen tak, aby bylo možné jej v budoucnu rozšířit bez obtíží,
8. **bezpečný**, bezpečnost musí být zajištěna ve všech ohledech za všech okolností, musí být odolný proti vnitřním i vnějším nebezpečím,
9. **provozovatelný**, systém je udržitelný v chodu bez větších obtíží,
10. **zdokumentovaný**, každý dobrý systém má kvalitní technickou a uživatelskou dokumentaci. [9]

+ Dobrý informační systém zejména **funguje**.

Na fázi tvorba cílů, vizí a strategií naváže **fáze návrh, vývoj a testování**. Během této fáze je zhotoven návrh řešení, implementace návrhu a testování implementovaného řešení. [10]

Na základě výstupů z analýz proběhne návrh řešení problémů, který splňuje všechny vstupní požadavky.

Řešení je následně implementováno podle návrhů. Návrh může být v případě potřeby upraven a předělán. Paralelně s implementací vzniká i technická dokumentace řešení.

V průběhu implementace řešení vzniká testovací strategie, realizované řešení se následně otestuje podle testovací strategie. Jelikož se testování neprovádí v reálném prostředí. Je vhodné, aby testovací data byla co nejvíce podobná reálným datům, aby bylo dosaženo co nejreálnějšího chování systému. Cílem je vyzkoušet co největší množství reakcí systému na zadávaná data a opravit chyby. [8]

Během **fáze zavádění systému do provozu** dochází k nasazení systému do provozu, školení uživatelů a opravě chyb.

Mezi možné strategie zavádění systému do provozu patří:

- souběžná strategie,
- pilotní strategie,
- postupná strategie,
- nárazová strategie.

Souběžná strategie je nejbezpečnější metoda, kdy původní a nový systém běží souběžně po nezbytně nutnou dobu, než je systém spolehlivý a uživatelé jsou s ním seznámeni. Uživatelé mnohdy dělají stejnou práci v obou systémech, nemůže tedy toto období trvat příliš dlouhou dobu, hrozí vznik averze uživatelů vůči novému systému.

Během pilotní strategie se nový systém zavádí postupně, např. ve vybraném oddělení nebo na vybrané skupině uživatelů. Nejdříve se systém otestuje na vybrané části, která je dostatečně komplikovaná, aby byly vyzkoušeny všechny funkčnosti a problematické oblasti. Po ověření funkčnosti se může postupně zavést v celém rozsahu.

Postupná strategie se využívá u systémů se složitými vnitřními vazbami, např. bankovní a energetické systémy, je nutné nasazovat systém postupně a testovat jednotlivé vazby, aby byly kompatibilní a systém byl plně funkční.

Nárazová strategie je nejriskantnější strategie, původní systém je plně a okamžitě nahrazen systémem novým. [10]

Ve **fázi provoz a servis** je systém plně uveden do provozu, používán uživateli, doplněn o uživatelskou příručku, případně kurz pro nové uživatele. Systém musí být udržován, aby stále plnil účel, k jakému byl zřízen a případně inovován. Je nutné, aby měl softwarový produkt podporu pro uživatele, na kterou se mohou obrátit v případě otázek nebo s požadavky na rozšíření, vylepšení systému. Je doporučeno, aby byla podpora rozdělena

na primární a sekundární. Primární podpora slouží pro všechny uživatele, např. formou helpdesku. Sekundární podporu mohou využít kvalifikovaní uživatelé a slouží zejména jako podpora primární podpory pro složitější požadavky, nápady uživatelů na vylepšení a podobně.

Poslední fází je **zánik nebo reengineering**, jelikož oblasti informačních technologií se velmi rychle vyvíjejí, je nutné neustále udržovat krok s konkurencí, sbírat zpětnou vazbu od uživatelů a produkt inovovat. Pokud není produkt inovován, je postupem času jeho zánik nevyhnutelný.

## 3.2 Metodiky a přístupy k vývoji softwaru

Mezi základní patří model vodopád, prototypový model, model spirála, agilní metodiky. [10]

### 3.2.1 Vodopádový model

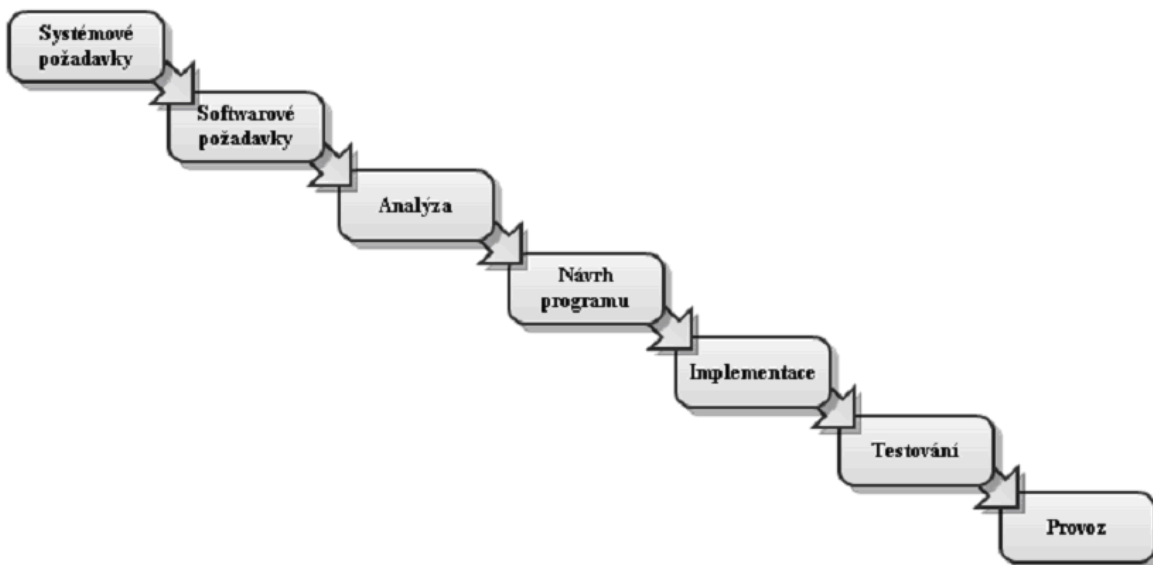
Vodopádový přístup je jedním z nejstarších modelů. Byl popsán již v roce 1970 Winstonem Royce v článku „Managing the Development of Large Software Systems“. [11] Je charakteristický striktní strukturou vyžadující detailní úvodní analýzu a srozumitelně popsané požadavky i očekávání zainteresovaných osob. Jednotlivé fáze na sebe navazují, neprolínají se a zpětně se k nim nevrací. [12]

Životní cyklus softwaru se rozděluje do sedmi fází:

- systémové požadavky,
- softwarové požadavky,
- analýza,
- návrh programu,
- implementace,
- testování,
- provoz.

V průběhu let vzniklo spoustu různých modifikací tohoto modelu, např. V – model životního cyklu. [10][13]

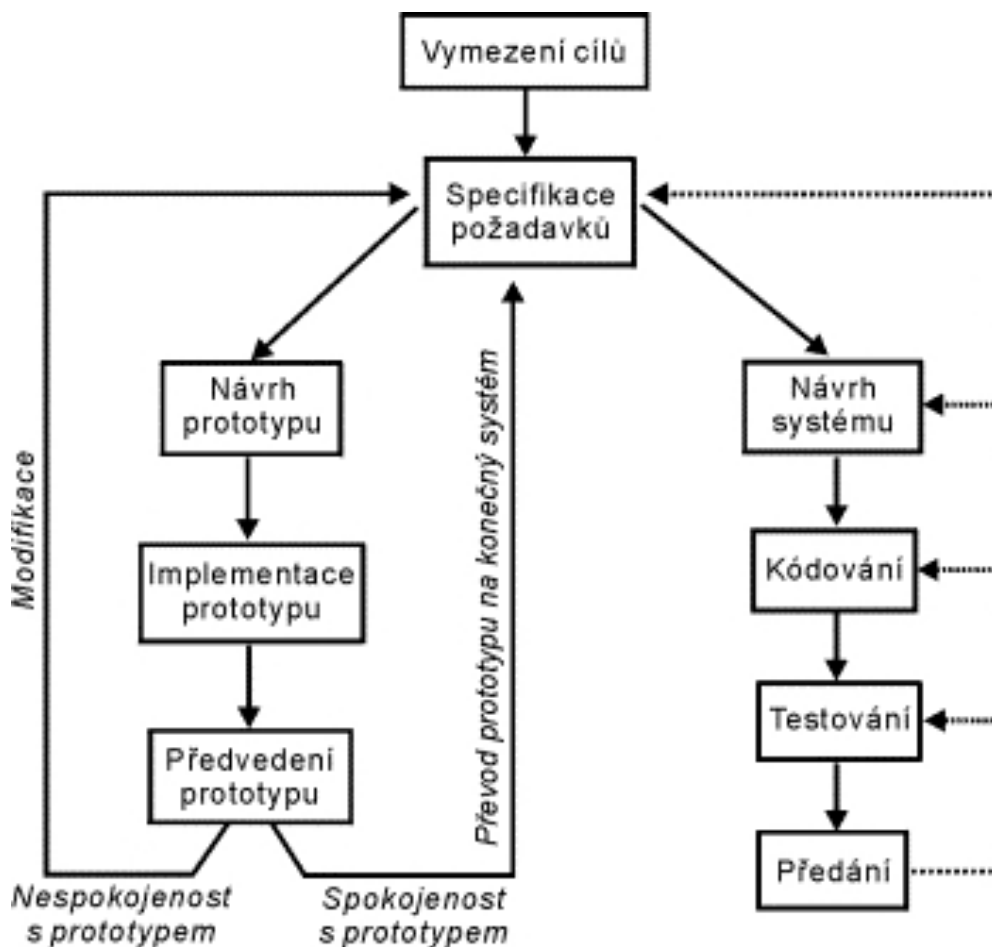
Obrázek 2 Vodopádový model životního cyklu softwaru [11]



### 3.2.2 Prototypový model

O prototypovém modelu lze přemýšlet jako o rozšíření vodopádového modelu, především ve způsobu interakce se zákazníkem a reakcí na změny. Zákazník často na počátku vývoje nedokáže plně formulovat své požadavky, neboť často nemá představu, co vše lze vytvořit a co ve stanoveném rozpočtu a čase nelze dodat. Díky průběžným prototypům softwaru, které jsou zákazníkovi představeny, je získána zpětná vazba. V průběhu vývoje lze zpřesňovat zadání a lépe řídit očekávání, což přináší značnou výhodu v řízení očekávání zákazníka a zvyšuje to šance, že bude zákazník spokojen.

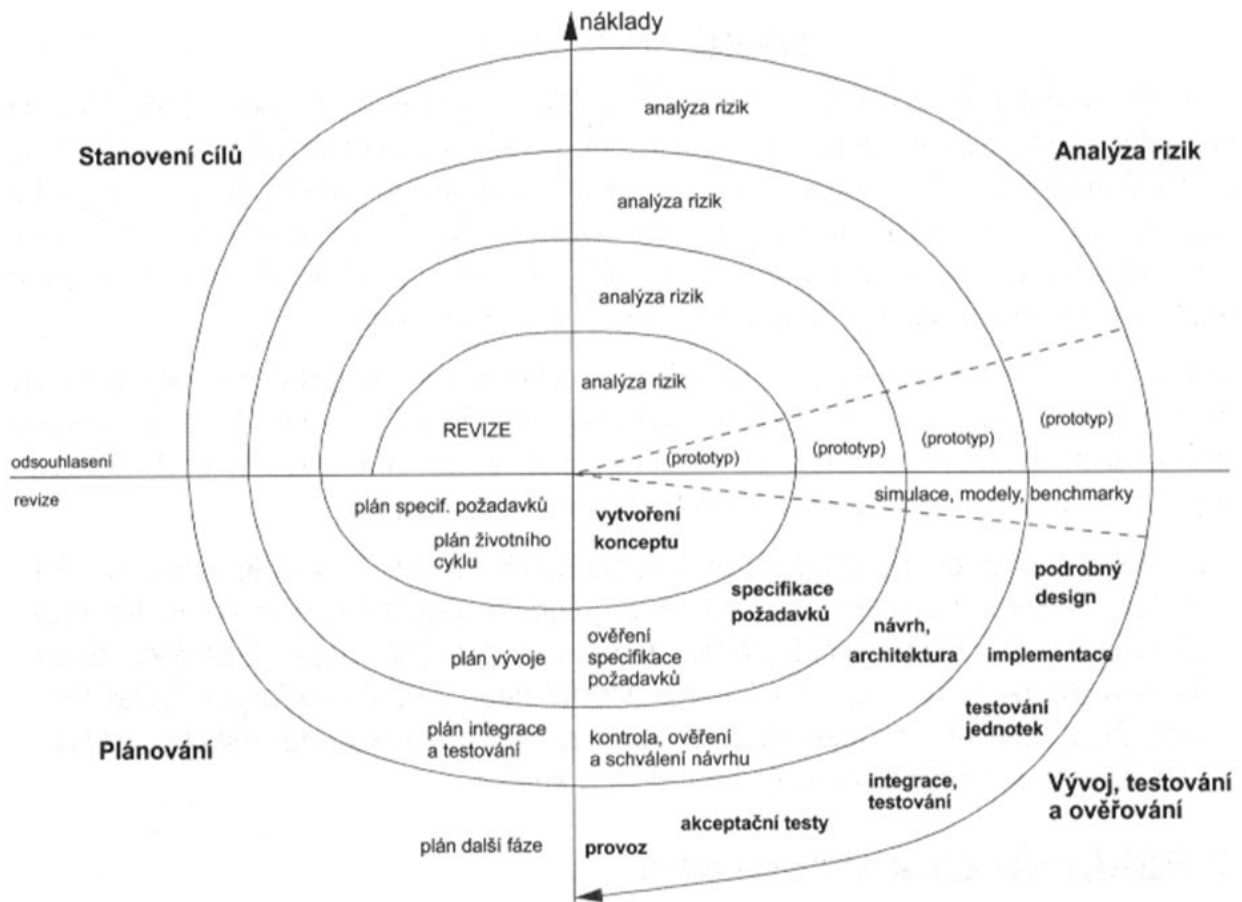
Obrázek 3 Schéma prototypového modelu [10]



### 3.2.3 Model spirála

Spirálový model poprvé popsal Barry Boehm ve svém článku "A Spiral Model of Software Development and Enhancement" již v roce 1988. [14] Kombinuje řízení rizik s prototypováním. Stojí na opakovaných cyklech, iterativním přístupu a analýze rizik. Během analýzy rizik je nutné definovat možná rizika a připravit si scénáře, jak tato rizika eliminovat, případně vyřešit. Rizika řadí do kategorií lidský faktor, projektová rizika, technická rizika, obchodní rizika. [13]

Obrázek 4 Schéma modelu spirála [15]



Na konci každé iterace je prototyp, který se přidá k prototypu k předcházející iterace. V dalších iteracích vždy přidají další části na vyšší úrovni z navazující cyklů. [14]

Každá iterace obsahuje tyto části: [10]

- stanovení cílů, alternativ řešení a omezujících podmínek dané iterace (cyklu),
- analýza rizik a vyhodnocení alternativ,
- realizace prototypu dané iterace (cyklu),
- plánování a vyhodnocení dané iterace (cyklu).

Model spirály obsahuje tyto cykly:

- pojmenování (definování) problému a vytvoření konceptu,
- specifikace požadavků,
- návrh a vytvoření architektury,
- design, implementace, testování, integrace až provoz.

### 3.2.4 Agilní přístup

V roce 2001 skupina amerických vývojářů sestavila Agilní manifest, který definuje hodnoty a principy agilního řízení. [16]

Charakteristikou jsou relativně krátké opakované cykly, během nichž je doručen konkrétní přírůstek se zaměřením na dodání přidané hodnoty pro zákazníka a aktivní zapojení zákazníka. Díky relativně krátkým cyklům lze rychle reagovat a zlepšovat software na základě zpětné vazby a poučení z předchozích cyklů. Klíčovými hodnotami jsou spolupráce, komunikace a dynamická reakce na změny. Oproti klasickému řízení (např. model vodopád), kdy je rozsah práce definován v začátku a za ideálních podmínek je v čase neměnný, se agilní řízení vyznačuje proměnným rozsahem, který by ovšem měl být fixní v rámci jednoho cyklu. Mezi metodiky vycházející z agilního přístupu patří extrémní programování (EX), lean development, vývoj poháněný vlastnostmi daného softwaru (FDD), vývoj řízený testy (TDD), scrum, kanban a další. V této práci bude blíže popsána metodika scrum.

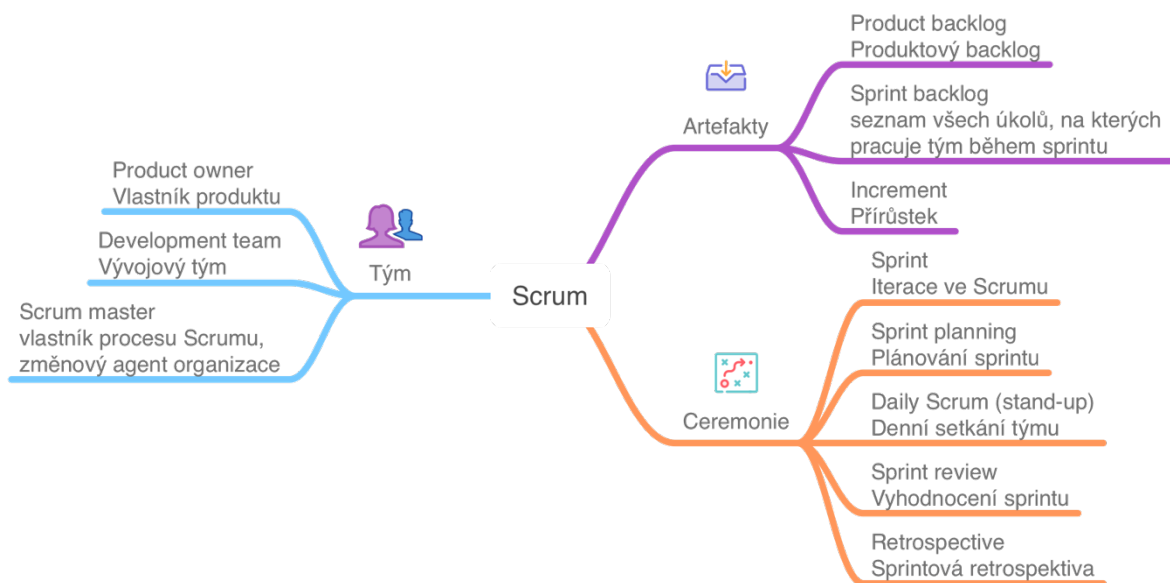
**Scrum** je rámec (framework), díky kterému se nejen vývojové týmy mohou zabývat složitými problémy a zároveň efektivně dodávat produkty s přidanou hodnotou. Podle Kena Schwabera a Jeffa Sutherlanda je scrum *stručný, snadno pochopitelný a obtížný na dokonalé ovládnutí*. [17]

Scrum stojí na třech pilířích:

- transparentnost, všichni zúčastnění by měli mít přehled, jaké kroky se provádí, z jakého důvodu se jednotlivé kroky provádí, jaká je vize produktu, v jakém stavu jednotlivé kroky jsou,
- kontrola, v krátkých časových úsecích probíhá revize stavu jednotlivých kroků,
- adaptace, v případě potřeby je možné jednotlivé kroky či přístup k nim změnit.

Scrum je sestaven z týmu, rolí, ceremonií, artefaktů a pravidel. [17]

Obrázek 5 Myšlenková mapa scrum pojmů [Vlastní zpracování podle [17]]



**Scrum tým** tvoří vlastník produktu, vývojový tým a scrum master.

Vlastník produktu je odpovědný za přínos vytvořeného produktu vývojovým týmem pro zákazníka a kompletní správu produktového backlogu. Měl by být reprezentován právě jednou osobou, může svoji práci delegovat na koordinátora, ovšem osobou zodpovědnou za obsah je vždy vlastník produktu.

Vývojový tým je složen z profesionálů, kteří jsou odpovědní za svoji práci. Mají možnost si zadanou práci v rámci sprintu sami organizovat. Jejich cílem je doručit přírůstek sprintu v rozsahu a kvalitě podle domluvy s vlastníkem produktu, a to do dne vyhodnocení sprintu. Scrum master je průvodcem, který týmu jako celku i jednotlivcům pomáhá pochopit a efektivně uvádět v praxi teorii, principy, pravidla hodnoty scrumu.

Mezi **scrum artefakty** patří produktový backlog, sprint backlog a přírůstek.

Produktový backlog je seznam věcí, které jsou potřeba na produktu vytvořit či upravit. Osobou odpovědnou za jeho správu je vlastník produktu.

Sprint backlog obsahuje vybrané položky z produktového backlogu, na nichž se bude v rámci sprintu pracovat.

Přírůstek je množina dodaných položek, které byly dokončeny v průběhu sprintu a jsou přičteny k stávajícím přírůstkům z uplynulých sprintů.



**Scrum ceremonie** jsou sprint, plánování sprintu, denní setkání týmu, vyhodnocení sprintu, sprintová retrospektiva.

Sprint je časové období, většinou kratší třiceti dnů, na jehož konci je doručen přírůstek. Jednotlivé sprinty na sebe navazují, plánování sprintu nadcházejícího tedy probíhá před skončením probíhajícího sprintu za účelem plynulého navazování sprintů.

Sprint začíná schůzkou k plánování sprintu, které se účastní celý scrum tým a je časově ohraničen za cílem určení rozsahu (scopu) daného sprintu a jeho cíle. Vlastník produktu navrhne, které položky z product backlog by měly být na konci sprintu doručeny v přírůstku a vývojový tým určí, zda svými kapacitami dokáže pokrýt stanovený cíl. Během sprintu mohou být projednány změny v rozsahu sprintu na základě získaných znalostí v průběhu sprintu a aktuálních kapacit vývojového týmu.

Vývojový tým se schází každý den během denního setkání týmu, kdy každý člen zhodnotí, co se událo od posledního denního setkání a jakou práci má naplánovanou do dalšího denního setkání. Schůzka by neměla trvat déle než 15 minut. Je určena k synchronizaci aktivit napříč týmem a zhodnocení stavu sprintu vzhledem k cíli sprintu. Díky denním schůzkám se snižuje množství dalších schůzek, zlepšuje se komunikace v týmu, překážky mohou být však identifikovány a odstraněny, v neposlední řadě se prohlubuje znalost všech členů týmu o produktu.

Na konci sprintu je provedeno vyhodnocení sprintu s cílem představit výsledný přírůstek sprintu a revize produktového backlogu. Vyhodnocení sprintu se účastní celý scrum tým a zainteresované osoby (stakeholders) na pozvání vlastníka produktu.

Sprintová retrospektiva nabízí prostor k zamyšlení týmu nad jejich spoluprací, komunikací a dohodnutí postupu pro zlepšení. Schůzka by měla být vedena pozitivně a produktivně za cílem potencionálního zlepšení.

Podle Kena Schwabera a Jeffa Sutherlanda je smyslem sprintové retrospektivy *prozkoumání průběhu posledního sprintu, co se lidí, vztahů, procesů a použitých nástrojů týče, identifikace a seřazení toho, co šlo dobře, a potenciálních vylepšení, vytvoření plánu na realizaci jednotlivých vylepšení týkajících se způsobu práce scrum týmu.* [17]

### 3.3 Komunikace

Jak bylo zmíněno v kapitole Životní cyklus týmu, vývoj softwarového produktu je týmová práce. Komunikace ovlivňuje jak úspěšnost produktu jako celku, která se odvíjí od úspěšnosti spolupráce mezi týmy zadavatele, dodavatele, tak náladu v týmu, jež je dána úspěšností komunikace mezi jednotlivými členy týmu a odráží se na jejich spolupráci. Komunikace je velmi široký a komplexní pojem, kterým se zabývá mnoho vědních oborů psychologie, lingvistiky a mnoho dalších. V případě vývoje softwaru se jedná zejména o mezilidskou komunikaci mezi dvěma a více lidmi, proto se tato kapitola zabývá komunikací sociální.

Sociální komunikaci lze definovat jako kontinuální dvousměrný proces výměny informací mezi lidmi za účelem dorozumění. [18]

Komunikace mezi zákazníkem a dodavatelem je klíčová pro úspěšné naplnění dohodnuté zakázky a prolíná více kapitol této práce, konkrétně Životní cyklus softwaru, Metodiky a přístupy k vývoji softwaru, Řízení komunikace, Řízení odpovědnosti.

#### 3.3.1 Model komunikace

Donelly, Gibson, Konopaske a Ivancevich ve své knize „Organizations Behavior, Structure, Processes“ uvádí model procesu komunikace, jenž přehledně znázorňuje jednotlivé prvky tohoto procesu. [19]

Obrázek 6 Model procesu komunikace podle Donelly, Gibson, Konopaske a Ivancevich [19]



Pro názornost bude celý proces uveden na konkrétním případu z praxe.

**Zdroj** reprezentuje komunikátora, který zahajuje komunikaci. Komunikátorem může být osoba, která má nápad, informaci, účel (záměr) komunikace, například produktová manažerka, která má nápad na novou funkčnost.

**Kódování** je převedení myšlenek komunikátora do sady symbolů vyjadřující jazyk komunikátora. Produktová manažerka na základě svých předchozích zkušeností zformuje svoji myšlenku do slov.

**Zpráva** je výsledkem kódování. Komunikátorův účel vyjádřen ve zprávě může být verbální nebo neverbální.

**Kanál**, v některé literatuře označován též jako médium, je nositelem zprávy, prostředek, kterým je zpráva odeslána. Například osobní komunikace, telefonický rozhovor, skupinová schůzka, textová zpráva a videokonference. Správně zvolené médium může hrát klíčovou roli v dekódování.

**Dekódování** neboli interpretace obsahuje myšlenkové procesy příjemce na základě jeho předchozích zkušeností, znalostí a okolností.

**Příjemce** zprávy, osoba, se kterou komunikátor komunikuje.

**Zpětná vazba** je žádoucí reakce příjemce na zprávu od komunikátora, která snižuje možnost nepochopení či zkreslení mezi zamýšlenou zprávou a přijatou zprávou. Pokud zpětná vazba chybí, jedná se o jednosměrný komunikační proces, v němž hrozí zkreslení.

**Šum** je jakékoliv narušení procesu komunikace. Může se vyskytovat v jakémkoliv z prvků. Rozlišujeme prvky vnitřní a vnější. Vnější prvky souvisí s prostředím, ve kterém dochází ke komunikaci, např. hluk, přítomnost dalších osob a další. Vnitřní šumy mohou mít mnoho podob od momentálních, jež jsou ovlivněny momentálním stavem (nemoc, stres atd.), po dlouhodobé osobnostní překážky (např. rozdílné kultury, rozdílné znalosti v daném oboru, postoje, společenský status, emoce, různé komunikační dovednosti). Je nutné si šumy uvědomovat a dlouhodobě s nimi pracovat.

Všechny výše popsané prvky spolu souvisí a neměly by být oddělovány, neboť popisují činy, jež je třeba naplnit, aby došlo ke komunikaci a proces komunikace byl celistvý.

Komunikace může být vertikální (nadřízený – podřízený, podřízený – nadřízený) nebo horizontální (kolegové, přátelé, spolužáci a podobně), mezi jednotlivci, skupinami nebo jejich kombinací a měla by vždy obsahovat výše uvedené prvky.

### 3.3.2 Řízení komunikace

Během životního cyklu produktu je vhodné nastavit základní principy komunikace, které budou průběžně revidovány například v agilním vývojovém týmu při retrospektivách. Je nezbytně nutné tyto principy stále připomínat, důsledně trvat na jejich dodržování a seznámit s nimi všechny zúčastněné strany a nově příchozí.

Podle Project Management Institute, publikace „A Guide to the Project Management Body of Knowledge ( PMBOK® Guide )—Sixth Edition“ lze řízení komunikace rozdělit do tří úkonů [20].

**Plánování komunikace** se skládá z analýzy a vytvoření plánu komunikace. Analýza stanovuje interní, externí potřeby a požadavky, např. na komunikaci uvnitř týmů, mezi týmy, k zainteresovaným osobám (stakeholders), k mediím a podobně, dále organizační strukturu, pravidla komunikace, např. počet schůzek, jejich strukturu a formu reportingu. Definuje komunikační kanály, nástroje, použité technologie včetně popisu způsobu a četnosti. [20]

**Zajištění komunikace a její řízení**, jejím základním dokumentem je komunikační plán, který může mít více podob, např. tabulka, matice, seznam, záleží na konkrétním produktu a domluvě. Měl by obsahovat role jednotlivých členů a nástroje (kanály) pro komunikaci, pravidla komunikace, odpovědnosti, požadavky a očekávání. Součástí může být matice nebo model, jehož cílem je definovat, kdo komunikuje s kým, jak často, jakým způsobem, případně kdy. Příkladem může být formát komunikace schůzka, pokud se jedná o pravidelnou schůzku, mělo by být stanoveno, kdo je odpovědný za její konání a průběh, za jakým účelem se koná, co má být jejím výstupem, jakou má frekvenci konání (denní, týdenní, měsíční, případně jinou), definované povinné účastníky a definován postup, pokud chce některý z členů přizvat hosta. Komunikační plán by měl obsahovat plán reportingu. Report by měl obsahovat analýzu dosavadního vývoje včetně stavu provedené práce, budoucí předpokládaný vývoj včetně výhledu plánované práce, aktuální stav, rizika, otevřené otázky, problémy, přehled změn oproti plánu a další důležité informace. Struktura se v jednotlivých společnostech může měnit podle potřeb dané společnosti. [20]

**Sledování a průběžná kontrola** komunikace je základním kritériem pro úspěšnou komunikaci. Je nezbytně nutné ověřovat, zda je komunikační plán dobře nastaven, splňuje očekávání všech zúčastněných a naplňuje svůj účel. Komunikační plán by měl být pravidelně revidován a v případě potřeby aktualizován. Součástí kontroly je podpora komunikace

formou konzultací, workshopů, otevřených diskusí řízených externím moderátorem. Dále neformální setkání s cílem budování týmového ducha a prohlubování vztahů uvnitř týmu, stejně tak jako formální setkání za účelem sdílení informací a ověřování sdílených informací. [20]

### 3.4 Odpovědnost

Odpovědnost je pojem, který je popsán v odborné literatuře vědních oborů jako psychologie, etika, filozofie, teologie i právo. Z latinského respondeo (odpovídat) jej lze vykládat jako odpovídání se někomu nebo dávání odpovědi na otázku. V právní vědě je chápán jako zodpovídat se před soudní instancí, teologie odpovědnost vykládá jako odpovídání se Bohu, nejvyššímu soudci. [21]

V běžném životě je tento pojem obecně chápán jako vykonání určitého úkolu, povinnosti. V profesním životě vyplývá odpovědnost z povinností pracovníka plnit činnosti, úkoly, jež jsou dány popisem jeho pozice (role, pracovního místa) ve stanovené kvalitě. Tedy dosahovat stanovených a očekávaných výsledků ve stanoveném času a kvalitě. [22]

#### 3.4.1 Řízení odpovědnosti

Jak bylo uvedeno v kapitole Životní cyklus týmu, vývoj softwaru je týmová práce. Předpokladem pro úspěšné doručení produktu je odpovědnost jednotlivých členů týmu. Členové mohou odpovědně odvádět svoji práci za předpokladu, že je jim odpovědnost řádně přidělena, plně rozumí jejímu rozsahu a znají kontext. Každý člen scrum týmu je odpovědný za odvedení přidělené práce, proto je stěžejní přehledně určit odpovědnost. K názornosti odpovědností lze využít například kompetenční model či matice odpovědností, která má množství variant.

Kompetenční model definuje všechny kompetence neboli vlastnosti, kvalifikaci, zkušenosti, dovednosti potřebné k plnění přidělených úkolů a může pomoci naleznout vhodnou osobu na danou pozici. [23]

Matice odpovědností slouží k znázornění odpovědností jednotlivých osob v konkrétních pozicích k úkolu v rámci organizace. Existuje množství variant mezi nejznámější patří matice RACI, jejím rozšířením je matice RASCI. Matice využívaná k zobrazení

odpovědnosti při rozhodování je matice DACI (Driver, Approver, Contributor, Informed). [24]

**Matice RACI** je nástrojem používaným pro řízení odpovědnosti. V praxi slouží k určení úrovně odpovědnosti k danému úkolu. [25][20]

Název RACI je akronymem z anglických slov Responsibility, Accountability, Consulted, Informed.

R – responsibility, česky odpovědnost, označuje osobu vykonávající úkol,

A – accountability, někdy též approver, česky povinnost zodpovídat, se označuje osobu odpovědnou za přínos úkolu, jeho účel a vykonání úkolu jako celku,

C – consulted, česky konzultováno, poradit se, radit se označuje osobu, s níž má být úkol konzultován,

I – informed, česky informovaný, označuje osobu, jenž má být informována o úkolu.

Odpovědnost A accountability za daný úkol by měla mít právě jedna osoba, osob s úrovní odpovědnosti R responsibility, zapojených do úkolu, může být více, podle velikosti úkolu. Jedna osoba může mít více odpovědností, ale právě jedna může mít odpovědnost úrovně A.

**Matice RASCI** (Responsibility, Accountability, Support, Consulted, Informed) je rozšířením matice RACI o S – support, česky podpora, označuje osobu, jež reprezentuje osobu zajišťující podporu v průběhu realizace úkolu. [26]

Tabulka 1 Příklad matice RASCI [Vlastní zpracování]

	<b>Produktový manažer</b>	<b>Vývojář</b>	<b>Marketingový specialista</b>	<b>Koordinátor</b>
<b>Zveřejnění příspěvku na sociální síť</b>	A		R	C
<b>Úprava úvodní stránky na webových stránkách</b>	A	R	C	I
<b>získání GDPR souhlasu</b>	A		I	

### 3.5 Akceptační kritéria a hodnoticí kritéria

Vývoj produktu je nutné detailně řídit, stejně jako jakýkoliv jiný projekt. Vývoj produktu můžeme rozdělit na jednotlivé etapy, které lze řídit následně projektově. Níže jsou rozebrány pohledy na hodnocení projektů, tzn. jednotlivých etap vývoje produktu. Následující pasáž pojednává o vybraných metodách a nástrojích používaných pro hodnoticí nebo akceptační kritéria.

#### 3.5.1 Model KKTR (kvalita, kvantita, termín, rozpočet)

Každý projekt je vhodné systematicky řídit. Jednou z možností je sledovat stanovená kritéria podle modelu KKTR. Vždy je nutné sledovat všechny metriky, kterými jsou kvalita, kvantita, termín a rozpočet. Při zaměření pouze na jednu metriku je jisté, že projekt bude problémový, neboť dojde ke zkreslení hodnocení. Je nezbytně nutné najít rovnováhu mezi všemi kritérii.

**Kvalita** reprezentuje úspěšnost vyrobeného produktu, může být definována jako *souhrn všech znaků/charakteristik produktu, které ovlivňují schopnost dodržet stanovené a předpokládané potřeby zákazníka projektu*. [27]

Vyroběný produkt, například software, musí splňovat předem dané požadavky zákazníka, naplnit jeho očekávání a být vyroben dohodnutým postupem.

**Kvantita** představuje předem dohodnutý rozsah funkcí podle požadavků zákazníka. Rozsah požadovaných funkcí se může v průběhu vývoje měnit. Je nutné průběžně výstupy konzultovat se zákazníkem a ujistit se, že funkčnost naplňuje očekávání a předpokládaný způsob užití.

**Termín** udává časový horizont určený pro naplnění požadavků zákazníka. Jedná se o předem vymezený čas pro realizaci zakázky.

**Rozpočet** udává vyhrazené finanční prostředky, které je možno využít na realizaci celé zakázky.

Manažer daného projektu musí udržovat všechny metriky v předem stanovených přípustných mezích, které jsou vymezeny minimálními a maximálními hodnotami, kterých lze u jednotlivých metrik dosáhnout. Tyto hodnoty jsou stanoveny před začátkem projektu.

Cílem manažera je minimalizovat metriky rozpočet a termín, nesmí dojít k překročení horní hranice, jež byla stanovena na začátku zakázky. Naopak metriky kvalita a kvantita se maximalizují. Minimální hranice, jež byla stanovena na začátku zakázky, musí být dosaženo. Je nezbytně nutné dodat funkční systém v minimálním rozsahu a kvality podle dohody.

### 3.5.2 Model “CRaB QueST” (Cost, Risk, Benefits, Quality, Scope and Time)

Dalšími metrikami, podle kterých lze vyhodnotit úspěšnost vývoje produktu, jsou rizika a doručená přidaná hodnota. Je vhodné KKTR rozšířit o tyto metriky. [28]

Rozšířením KKTR o rizika a přidanou hodnotu vznikne komplexnější pohled na projekt a je možné jej detailněji řídit. Díky pohledu na přidanou hodnotu, přínos realizace projektu pro zákazníkův účel, lze lépe pochopit potřeby zákazníka a během projektu tyto potřeby naplnit.

Model “CRaB QueST” lze tedy česky vykládat jako náklady (rozpočet), rizika, přidaná hodnota, kvalita, rozsah (kvantita) a termín.

V kapitole Model KKTR, jsou popsány metrika kvalita, kvantita, rozpočet, termín, jež jsou v modelu “CRaB QueST” reprezentovány slovy **kvalita, rozsah, náklady a termín**. Obsah jednotlivých metrik se překrývá s metrikami kvalita, kvantita, rozpočet, termín, jež jsou popsány v kapitole Model KKTR. Z toho důvodu jsou zde vynechány a je pokračováno popisem metrik nových.

**Rizika** jsou nedílnou součástí každého projektu. Proto, aby mohl být projekt vyhodnocen jako úspěšný, je nutné dodat smluvený výstup v kvalitě, rozsahu, rozpočtu, termínu. Toho lze dosáhnout právě díky detailnímu řízení rizik, jejich předvídání, plánu jejich řešení a práce s nimi během celého projektu.

Každý projekt by měl přinášet **přidanou hodnotou**, kterou se zákazník odliší od konkurence. Přidanou hodnotou může být přínos komunitě či společnosti, vznik inovativního řešení, které zvýší hodnotu zákaznickovy firmy. Přidanou hodnotu je nutné řídit stejně jako rozpočet, termín, kvalitu a kvantitu. Například průběžnou analýzou trhu, jeho použitelnosti pro koncového zákazníka, validací řešení s koncovými zákazníky.



### 3.5.3 Metoda OKR (Objectives and Key Results)

John Doerr ve své knize „Measure what matters: How Google, Bono, and the Gates Foundation rock the world with OKRs“ popisuje systém (metodiku) OKR (Objectives and Key Results), česky cíle a klíčové výsledky, stanovování cílů, klíčových ukazatelů a z nich vyplývající aktivity. John Doerr uvádí, že jsou OKR využívány ve firmách jako Google, Spotify, Twitter, LinkedIn, Airbnb, The Guardian, Intel a mnoha dalších. [29] Za „otce“ metody OKR je považován Andrew Grove, John Doerr se s OKR seznámil v roce 1975 účastí na kurzu o OKR od Andrew Grove.

Výhodou OKR je její jednoduché užívání, cíle jsou snadno nastavitelné, a je tedy možné je přehodnotit v průběhu. Typicky jsou nastavovány kvartálně. Díky jejich jednoduchosti je lze i snadno sledovat.

Doerrův vzorec pro tvorbu OKR nejlépe vystihuje metodu OKR a zní:

***Co udělám (cíl, objectives) měřeno (sadou klíčových výsledků, key results).***

*Cíle jsou nezapomenutelné kvalitativní popisy toho, čeho chcete dosáhnout.*

*Cíle by měly být krátké, inspirativní a poutavé.*

*Cíl by měl tým motivovat a vyzvat ho k akci.*

*Klíčové výsledky jsou sadou metrik, které měří váš pokrok směrem k cíli.*

*Pro každý cíl byste měli mít sadu 2 až 5 klíčových výsledků. Víc než to si nikdo nebude pamatovat. [29]*

Příkladem může být placená mobilní aplikace, kterou uživatel může hradit měsíčně, půlročně nebo ročně. Cílem je lepší uživatelský zážitek (user experience). Klíčovými výsledky budou zvýšení prodejů ročního předplatného, zvýšení prodeje měsíčního předplatného, zvyšující procento vyplnění zpětné vazby zákazníky a množství doporučení v aplikaci App Store, která je platformou pro nákup aplikací uživatelů Apple zařízení.

Po doplnění do vzorce, *Co udělám (cíl) měřeno (sadou klíčových výsledků)* [29] bude formulace příkladu „Zlepším uživatelský zážitek při použití aplikace, výsledky budou měřeny množstvím nákupů měsíčního předplatného, zvýšení prodejů ročního předplatného, zvyšující procento vyplňované zpětné vazby od zákazníků a množství doporučení aplikace v App Store.“.

## 4 Vlastní práce

### 4.1 Unicorn a.s.

*Unicorn je renomovaná evropská společnost poskytující ty největší informační systémy a řešení z oblasti informačních technologií. Dlouhodobě se soustředíme na vysokou přidanou hodnotu a konkurenční výhodu přinášenou našim zákazníkům. Působíme na trhu již od roku 1990 a za tu dobu jsme vytvořili řadu špičkových a rozsáhlých řešení, která jsou rozšířena a užívána mezi těmi nejvýznamnějšími podniky z různých odvětví. Máme ty nejlepší reference z oblasti bankovníctví, pojišťovnictví, energetiky a utilit, komunikace a médií, výroby, obchodu i veřejné správy. Našimi zákazníky jsou přední a největší firmy. Jsme však zároveň provozovateli internetové služby Plus4U, ve které nabízíme široké portfolio služeb založených na robustních softwarových řešeních pro malé a střední podniky, a hlavně pro lidi. V internetové službě Plus4U si každý z nás vytvoří svůj digitální prostor a propojí se s digitálními prostory těch ostatních – jednotlivců i podniků a organizací. Disponujeme detailními znalostmi z celého spektra podnikatelských odvětví. Rozumíme principům jejich fungování, ale i specifickým potřebám svých zákazníků.*

*Naším cílem je vzdělávat lidi a také společnost. Proto jsme v roce 2007 založili Unicorn University, která zároveň provozuje Unicorn Top Gun Academy – komplexní vzdělávací systém pro naše spolupracovníky, tedy pro všechny z nás. Unicorn University není jen tradiční vysoká škola. Jsme plně online. Naší ambicí je podporovat online vzdělávání od základních škol až po univerzity včetně systémů celoživotního vzdělání. Prostě, software je všude.*

*Naše rozsáhlé týmy systematicky vzdělávaných odborníků dokonale ovládají rozsáhlé portfolio v současnosti obvyklých produktových řad, komponent a technologií. Přestože při své práci uplatňujeme mnoho progresivních a moderních myšlenek, vlastní dodávky řešení podléhají osvědčeným kritériím plynoucím z šesti omezení – náklady, **riziko, přínosy, kvalita, rozsah a čas**. Aplikujeme „CRaB QueST“. [30]*

**Společnosti Vladimíra Kováře** (dále holding) zahrnují jak všechny společnosti holdingu Unicorn, tak společnosti spolupracující se holdingem Unicorn. Tyto společnosti zajišťují

podpůrné procesy a zároveň nezávisle pracují na vlastních projektech. Přehled společností holdingu je vyobrazen níže (Obrázek 7 Struktura společností holdingu [9]).

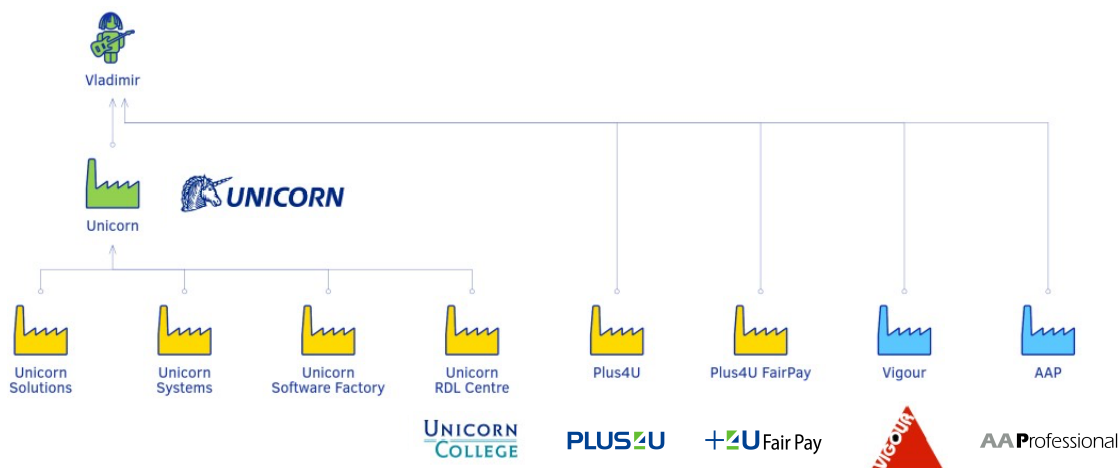
Holding Unicorn a.s. se skládá z firem Unicorn Solutions a.s., Unicorn Systems a.s., Unicorn Software Factory a.s., Unicorn Vysoká škola s.r.o. a množství dalších společností.

Tato práce se podrobněji zabývá společnostmi Unicorn Solutions a.s., jež je jedním ze zadavatelů projektů pro společnost Unicorn Software Factory a.s., která v práci zastupuje stranu dodavatele, a společnosti Plus4U, jež je provozovatelem aplikací uuApps. Společnosti Vigour se starají o spolupracovníky všech společností, poskytují jim zázemí a pečují o jejich rozvoj prostřednictvím Unicorn Research and Learning center. Společnost AAP se stará o podporu společností holdingu v oblastech právních a finančních.

Mohlo by se zdát, že tato struktura společnosti je velmi složitá. Nicméně při hlubším poznání jednoznačně vyplývá, že se jedná o velmi promyšlenou dělbu práce a na ni navazující organizační strukturu, která podporuje efektivní výrobu všech potřebných produktů, meziproductů a řádně kryje související rizika.

V průběhu let se holding rozšířil do několika zemích, mezi které patří Česká republika, Slovensko, Norsko, Ukrajina, Nizozemí, Německo. Primárním jazykem společností holdingu je od roku 2019 angličtina. Metodiky, postupy a myšlenky jsou od roku 2019 psány v anglickém jazyce. Tato práce obsahuje obrázky, které jsou citovány ze zdrojů společnosti a jsou jak anglické, tak české z důvodu různého data vzniku.

Obrázek 7 Struktura společností holdingu [9]



Ve své disertační práci zakladatel společnosti Unicorn a.s. Vladimír Kovář uvádí:

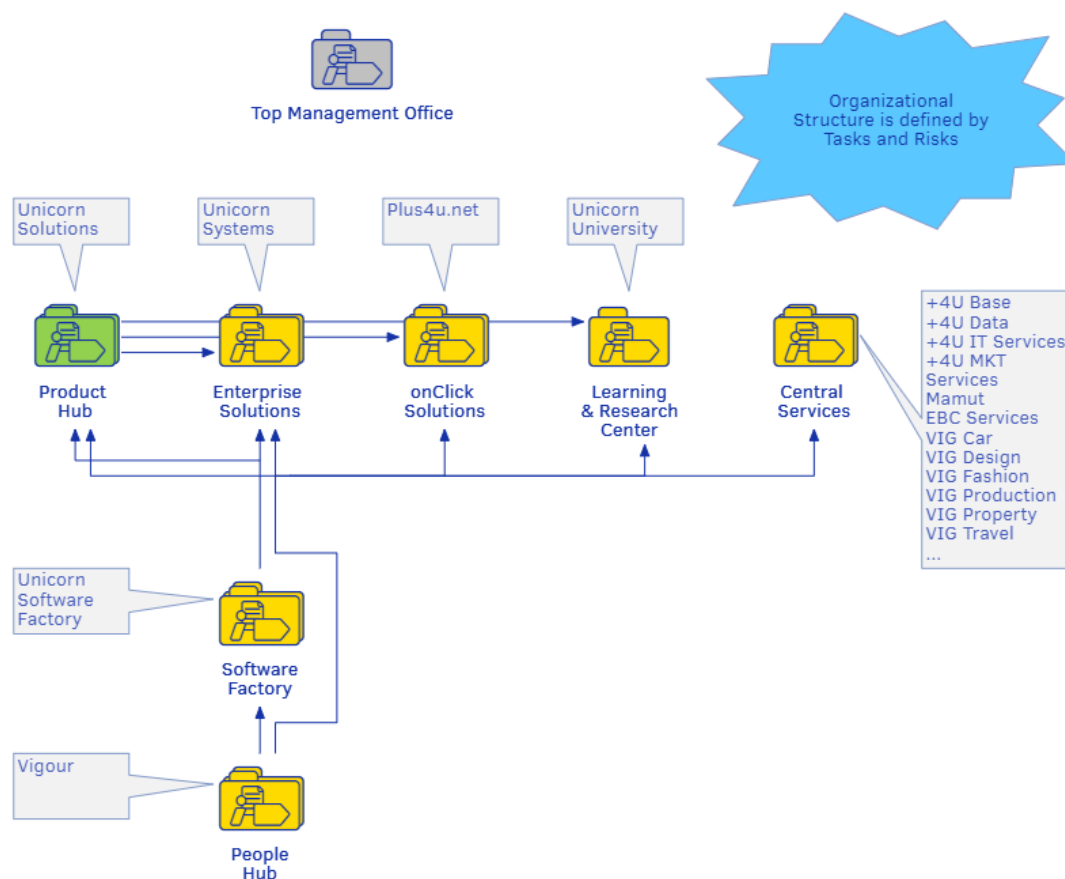
*Domnívám se (vlastně jsem o tom hluboce přesvědčen), že je možné v celé řadě případů podrobnějším řízením, nežli je v současnosti obvyklé, získat podniku konkurenční výhodu. Zároveň jsem však přišel na to, že bez potřebné metodiky a její přímé podpory informačním systémem podrobněji, nežli je obvyklé, řídit nelze. [31]*

V holdingu Unicorn výše uvedená myšlenka není pouze prohlášením, tato myšlenka je aplikovaná na všechny procesy v holdingu. Každý proces v metodice musí být přímo podpořen automatizovaným rozhraním, které podporuje práci. Například v aplikaci pro správu sprintu (uuSprintMan) lze nejen spravovat jednotlivé požadavky v backlog, a k nim vytvářet příslušné úkoly (tickets) v jednotlivých sprintech, ale všichni zúčastnění mají ve svých seznamech úkolů všechny související úkoly. Management má přehled o vykonané práci v rámci sprintu a aktuálním stavu jednotlivých úkolů. Dále má možnost zkontrolovat, že všichni dodržují metodiku. To je však jen jeden příklad z mnoha. Všechny postupy jsou stále aktualizovány podle nabitých zkušeností.

Pro systematické řízení holdingu je nutné dodržovat stanovené procesy, metodiky, více o těchto procesech a metodikách je uvedeno v kapitole Unicorn Approach, a mít jasně definovanou dělbu práce mezi společnostmi holdingu (Obrázek 8 Dělbba práce v holdingu [32]). Pro zpřehlednění dělby práce mezi společnostmi holdingu jsou společnosti řazeny do jednotek podle zaměření.

### 4.1.1 Dělbba práce v holdingu Unicorn a.s.

Obrázek 8 Dělbba práce v holdingu [32]



**Top Management Office** je jednotka, jež je samostatně vymezena v organizační struktuře a je obdobou projektové kanceláře. Pomáhá s organizací práce napříč firmami.

**Product Hub** je jednotka, která se stará o správu produktů a jejich rozvoj a řídí vývoj onClick produktů ve spolupráci se Software Factory, jež zajišťuje vývoj produktů. Je reprezentována společností Unicorn Solutions a.s. a je blíže popsána v kapitole Unicorn Solution a.s.

**Enterprise Solutions** je jednotka věnující se zákazníkům, kteří potřebují řešení na míru, například v sektorech bankovníctví, pojišťovnictví, systémy pro školství, energetiku a v dalších sektorech.

**onClick Solutions** je jednotka zabývající se provozem internetové služby Plus4U a prodejem tzv. onClick Solutions, tedy produktů, které je možné získat jedním kliknutím. Reprezentuje ji společnost Plus4U Net a.s. Tato společnost provozuje a prodává velké

množství produktů vyráběných ve společnostech holdingu (primárně vyráběných pod správou Product Hub, tedy společnosti Unicorn Solution a.s.). Jejimi zákazníky jsou fyzické osoby, malé a střední podniky.

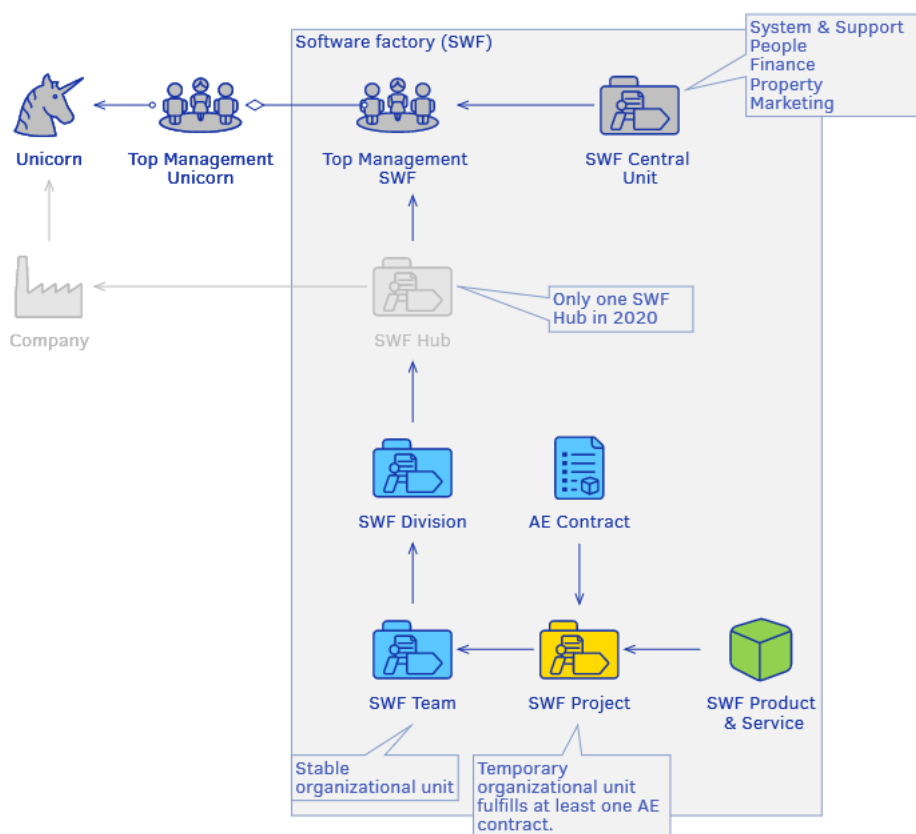
Mezi tyto produkty patří např. systémy pro lékaře (+4U Lékař), internetový obchod Mamut.cz, aplikace pro správu agendy českých živnostníků (+4U OSVČ), nástroj pro tvorbu webů uuWebKit, aplikace uuCourseKit pro tvorbu kurzů a již vytvořené vzdělávací kurzy pro zákazníky od žáků základní školy po vysokoškolské studenty nebo zájemce o konkrétní témata, např. GDPR a další. Ale také virtuální mobilní operátor +4U Mobile, platební kanál +4U Fair Pay, osobní identifikaci idGuard nebo aplikaci pro nabíjecí stanice elektromobilů ChargeUp.

Společnost Plus4U Net provozuje také velké množství produktů, které se využívají k řízení procesů v rámci společností. Například aplikace uuBookKit slouží k tvorbě interaktivních knih a jejich následnému užívání, zejména k vytváření dokumentací, učebnic, manuálů a jiných publikací. Aplikace uuBusinessRequestMan umožňuje uživateli vkládat představy o nové aplikaci nebo aktualizaci již existující aplikace do strukturovaného formátu, vyjádřeného vizí, user stories a non-functional request. Všechny části jsou blíže popsány v kapitole Model životního cyklu uuApp. Aplikace uuAppBusinessModelKit pomáhá automatizovat vytváření business models. Definuje strukturu pomocí actors, products, business use-cases a proceses. Aplikace uuAppModelKit umožňuje definovat technické požadavky aplikace v dobře organizované struktuře a poskytovat automatizaci pro vývoj aplikace, s použitím této aplikace jsou sepsány technické dokumentace každé uuApp. Blíže je představena v kapitole Akceptační kritéria zadání. Aplikace uuSprintMan pro správu sprintů, která poskytuje podporu agilním metodikám od správy backlog přes rozdělení backlog na jednoduché úkoly po předání finálních výstupů zákazníkovi. Uspadňuje koordinaci týmové práce přiřazením jednotlivých úkolů konkrétním členům týmu a plánováním sprintu.

Součástí jednotky **Software Factory** (SWF) jsou expertní týmy Unicornu se zaměřením na vývoj softwaru. Organizační struktura je vyobrazena níže (Obrázek 9 Organizační struktura SWF [33]) řízení Software Factory má na starosti Top Management SWF, jehož členové jsou zároveň součástí Top Managementu Unicornu. Součástí SWF jsou tzv. Central Units – centrální jednotky, které zajišťují chod jednotky. Starají se např. o finance, majetek

nebo marketing. SWF se dělí na tzv. huby (aktuálně pouze jeden) a ty se dále člení na divize. V divizích jsou stabilní organizační jednotky, tedy ustálené týmy. Týmy společnosti Unicorn Software Factory a.s. získávají zakázky primárně prostřednictvím burzy řešení. Na burzu řešení zadávají různí zadavatelé (např. Unicorn Solution) své poptávky a následně o tyto poptávky soutěží různé týmy dodavatelů (např. týmy Software Factory, která dodává softwarová řešení včetně testování nebo VIG design, jenž dodává grafické návrhy, případně další registrovaní dodavatelé).

Obrázek 9 Organizační struktura SWF [33]



Týmy Software Factory se skládají nejčastěji z rolí:

- team leader je vedoucím týmu, který je zodpovědný za rozdělení práce mezi členy týmu, stará se o pochopení zadání od zákazníka a zajistí předání zadání mezi členy týmu, kontroluje, zda je práce vykonaná, interně otestovaná a revidovaná, dohlíží na to, že je komunikace se zákazníkem zajištěna, že je

prováděno sledování rozpočtu, rozložení kapacit týmu v průběhu sprintu, že je zajištěna průběžná péče o členy týmu, organizuje denní schůzky týmu a je jejich aktivním účastníkem,

- senior uuApp designer je odpovědný za podporu uuApp designers, provádí revizi designů ostatních designérů a komunikuje se zákazníkem v případě složitějších návrhů, účastní se odhadů časové náročnosti úkolů, může se podílet na tvorbě business modelu a podobně,
- uuApp designer definuje a tvoří technické zadání pro vývojáře (uuApp developers), podílí se na tvorbě technické dokumentace, zajišťuje úzkou komunikaci s uuApp developer a doplňuje do dokumentace úpravy developer v průběhu,
- senior uuApp developer je odpovědný za podporu developers, provádí revizi kódu ostatních developers a komunikuje se zákazníkem v případě složitějších komponent, účastní se odhadů časové náročnosti úkolů a podobně,
- uuApp developer je odpovědný za implementaci funkčnosti zadané od uuApp designer, za vývoj kódu, přípravu automatizovaných testů, opravy chyb, sloučení a nasazení do vývojového prostředí,
- senior uuApp tester je odpovědný za podporu testerů, zodpovídá za testovací strategii, tvorbu testovacích scénářů, seznam nalezených chyb a komunikuje se zákazníkem i členy vývoje, účastní se odhadů časové náročnosti úkolů a podobně.
- uuApp tester je zodpovědný za testování vlastností produktu.

Software Factory disponuje třemi druhy týmů.

**Smíšený tým** je složen z team leader, developers, designers, tester (testers). Jednou z výhod je účast testers na denním setkání týmu, mají tak přehled o směřování aplikace a dokáží si lépe plánovat práci. Pokud vztah mezi developers a testers správně funguje, strany si důvěřují, rozumí a jsou k sobě otevření. Mohou testers snáze reportovat chyby a ty mohou být rychleji opraveny v důsledku pravidelné každodenní komunikace. Jejich kolegiální vztah může zároveň být nevýhodou a tester může záměrně přehlížet méně závažné chyby, tedy je akceptovat a netrvat na jejich opravě. Tyto týmy se využívají pro menší projekty nebo ve specifických odvětvích, kde je nutné specifické znalosti tester, díky tomu mohou být týmy



zaměřeny na jeden projekt nebo produkt, v rámci společností holdingu se tyto týmy využívají v energetickém sektoru.

**Vývojářský tým** je složen z team leader, developers, designers. Tento tým je přímo zaměřený na vývoj, testování probíhá v oddělených týmech.

**Testerský tým** je složen z team leader, testers. Tento tým je zaměřený na testování a může být sdílen napříč projekty a produkty. Největší výhodou je nezávislost testerského týmu. Nejsou tedy přímo ovlivňováni vztahy ve vývojovém týmu a testy mají větší objektivitu. Další výhodou odděleného testerského týmu je jejich znalost v testování, systematické hlášení chyb, díky němuž se opravuje více chyb, neboť testerský tým je odpovědný za nalezení chyb a je najat za tímto účelem. Tedy cítí větší odpovědnost za nahlášení úplně všech chyb.

**Komunikaci mezi zákazníkem a dodavatelem** lze rozdělit na úvodní, průběžnou, předávací a následnou podporu.

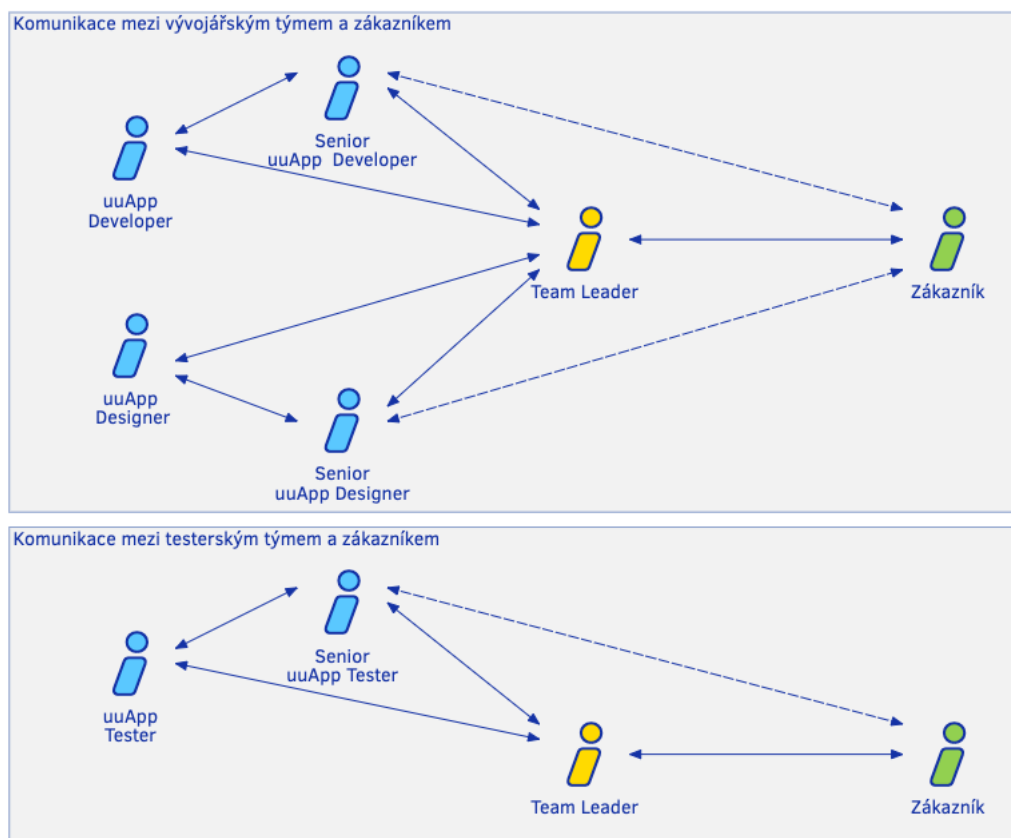
Během úvodní komunikace zákazník popisuje svoji představu o produktu, co od něj očekává, k čemu ho potřebuje používat. Cílem je stanovit formu rozpočtu zakázky, např. pevná cena za zakázku, dynamická cena podle odvedené práce na zakázce s předpokládaným rozpočtem, který je pravidelně reportován, dále stanovit časový harmonogram, termín dokončení zakázky a délku následné podpory. Hlavním cílem průběžné komunikace je reportování stavu zakázky a validace naplnění očekávání zákazníka. Před předáním musí být zakázka otestována, zákazník plně seznámen s výstupem. Podle obsahu smlouvy může být naplánováno zaškolení pracovníků, vypracován manuál, natočena instruktážní videa a případně stanoven následný servis a podpora. Během následné podpory při užívání produktu je získána zpětná vazba o spokojenosti zákazníka s dodaným produktem, jeho funkčností a naplnění očekávání. Tato vazba může být vstupem pro vývoj nových produktů či zlepšení daného produktu v další generaci.

Komunikace mezi dodavatelským týmem a zákazníkem probíhá zejména mezi zástupcem zákazníka (kontaktní osobou, např. product manager) a team leaderem. Součástí kontraktu je definování komunikačních kanálů, kontaktních osob a zástupců kontaktních osob.

Komunikace uvnitř týmu probíhá zejména mezi team leader, senior uuApp designer a senior uuApp developer. Celý tým spolu komunikuje na denní bázi během denních schůzek týmu (daily stand-ups) a poté během dne jsou v kontaktu senior uuApp designer s uuApp designer,

případně senior uuApp developer s uuApp developer, konzultují návrhy a řešení. V případě potřeby se senior uuApp developer či senior uuApp designer kontaktují s team leader nebo se poradí spolu.

Obrázek 10 Schéma komunikace mezi vývojovým týmem SWF a zadavatelem [34]



Jednotka **People Hub** (společnosti Vigour) pečuje o lidské zdroje holdingu Unicorn od nábory přes průběžné pohovory až po správu spokojenosti jednotlivých spolupracovníků, včetně jejich hmotné (IT vybavení) i nehmotné vybavenosti. Průběžné pohovory probíhají v rozmezí od 3 do 12 měsíců podle potřeby, spolupracovníka a podle jeho délky působení ve firmě. Pomáhá individuálně vybírat spolupracovníkům vhodné vzdělávací programy z nabídky Unicorn Top Gun Academy nebo z externí nabídky.

Jednotka **Learning and Research center** reprezentuje výzkum a vzdělávání uvnitř i vně Unicornu prostřednictvím Unicorn University (formálně Unicorn Vysoká škola s.r.o.) a Unicorn Top Gun Academy, která zajišťuje komplexní vzdělání pro všechny spolupracovníky ve společnostech holdingu, a nejen pro ně.

V jednotce **Central Services** je skupina jednotek, jejichž cílem je poskytnout ostatním jednotkám a společností částečnou nebo úplnou správu konkrétních obchodních procesů. Patří sem například zprostředkování služebních cest, správa budov a kanceláří, úklid, účetnictví, právní poradenství, správa data centra, centrální marketing, produkce akcí, správa vozového parku, tvorba grafických a marketingových materiálů. Díky této skutečnosti se mohou ostatní jednotky soustředit na jejich primární obchodní cíl a nevěnují se podpůrným procesům, které poskytují centrální služby.

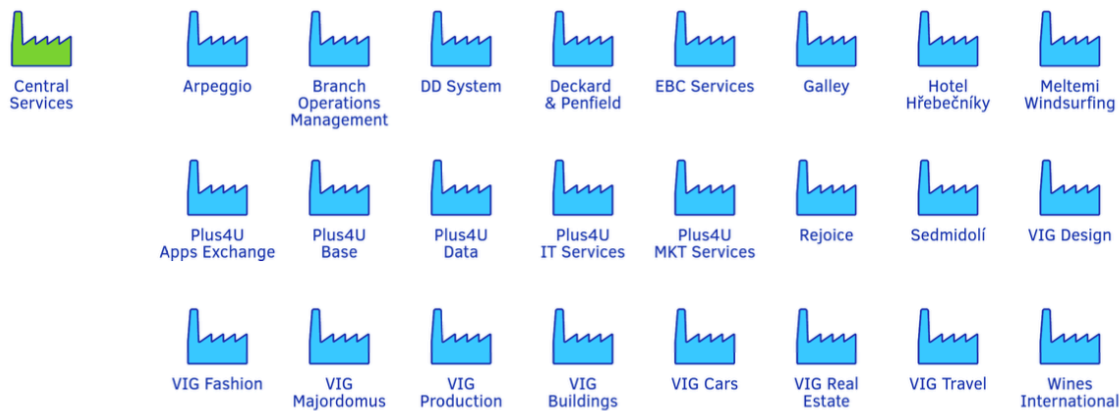
Do Central Services patří i další firmy, které zajišťují zkvalitnění života spolupracovníků a jejich rodinných příslušníků jako sportovně rekreační pobyty, distribuce a výroba oblečení cestovní kanceláře, distribuce harf Camac pro český a slovenský trh a další.

Součástí jednotky Central Services je i **burza řešení**, která podle oficiálního popisu [35] zprostředkovává výběrová řízení na dodavatele uuApps postavených pomocí digitální stavebnice informačních systémů Unicorn Universe a provozovaných v internetové službě Plus4U a další, např. překlady, grafický návrh, testování. Zajišťuje transparentní proces výběru dodavatelů uuApps, kontrolu kvality (certifikaci) jednotlivých výstupů v souladu s pravidly pro vytváření a provoz uuApp. Platí, že vybraný dodavatel realizuje zadanou zakázku ve stanoveném termínu a za dohodnutou cenu.

**Zákazníkem** burzy řešení mohou být firmy nebo živnostníci, kteří chtějí vyřešit problém prostřednictvím uuApp, musí mít připraveno zadání pro uuApp a být schopni zaplatit dohodnutou cenu za vytvoření uuApp. Zadavatelé uuApps mohou na burze řešení poptávat vytvoření designu uuApp dle svého zadání, implementaci uuApp dle designu, otestování uuApp, migraci dat do nové uuApp, vytvoření grafického návrhu, certifikaci uuApp, psaní, úpravy a překlady technické a uživatelské dokumentace v rámci uuApp a další.

**Dodavatelem** se může stát firma nebo živnostník, která disponuje požadovaným know-how. [35]

Obrázek 11 Společnosti patřící do centrálních jednotek [36]

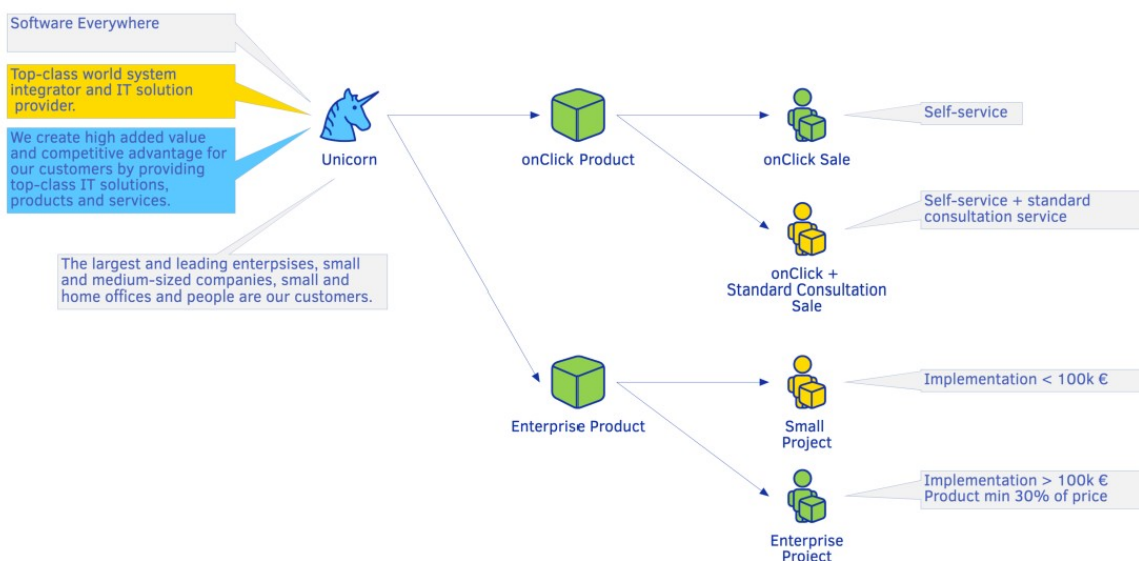


#### 4.1.2 Produkty Unicorn a.s.

Produkt podle Unicorn Approach je entita z reálného světa. Může jí být hmotná věc nebo myšlenka, jednoduše cokoli, jehož rozvoj chceme sledovat. Z pohledu obchodní disciplíny rozlišujeme meziprodukt a finální produkt. Za meziprodukt považujeme produkt, který je využívám v rámci jedné obchodní disciplíny. Finální produkt je produkt, který je používám koncovým uživatelem nebo napříč obchodními disciplínami.

V Unicornu se dělí produkty na onClick Products a Enterprise Product (Obrázek 12 Rozdělení produktů holdingu Unicorn [32]).

Obrázek 12 Rozdělení produktů holdingu Unicorn [32]



**onClick Product** se dělí na onClick Sale, jež jsou připraveny pro použití koncovým zákazníkem z trhu okamžitě a nepotřebují další konzultaci, například kurz v aplikaci uuCourseKit, nástroj pro tvorbu webů uuWebKit a další. Další součástí onClick products jsou onClick + Standart Consultation Sale produkty, jež jsou připraveny pro použití koncovým zákazníkem z trhu okamžitě. Tyto produkty lze nakonfigurovat pro potřeby konkrétního zákazníka, například produkty pro lékaře. Pro lékaře nabízí široké spektrum aplikací, které spolu navzájem komunikují. Aby byly odhaleny potřeby konkrétní ordinace, je vhodná konzultace s obchodníkem a vytvoření balíčku produktů z řady pro lékaře tak, aby odpovídal potřebám konkrétní ordinace a zároveň byl zajištěn přenos dat ze stávajících systémů používaných v ordinaci.

**Enterprise Product** (produkty pro podniky), obsahují projekty, jejichž cílem je vytvoření konkrétního řešení pro daného zákazníka. Pro naplnění cíle projektu jsou využity produkty holdingu Unicorn. Small Project jsou projekty, jejichž rozpočet je menší než 100 000 eur a produkty vyžadují implementační práce menšího rozsahu. Zákazníkem bývají menší až střední společnosti, ale také velké společnosti.

Enterprise Project jsou projekty, jejichž rozpočet je vyšší než 100 000 eur a minimálně 30 % z jejich rozpočtu tvoří cena za enterprise uuApps. Jejich provoz či implementace vyžadují práce většího rozsahu.

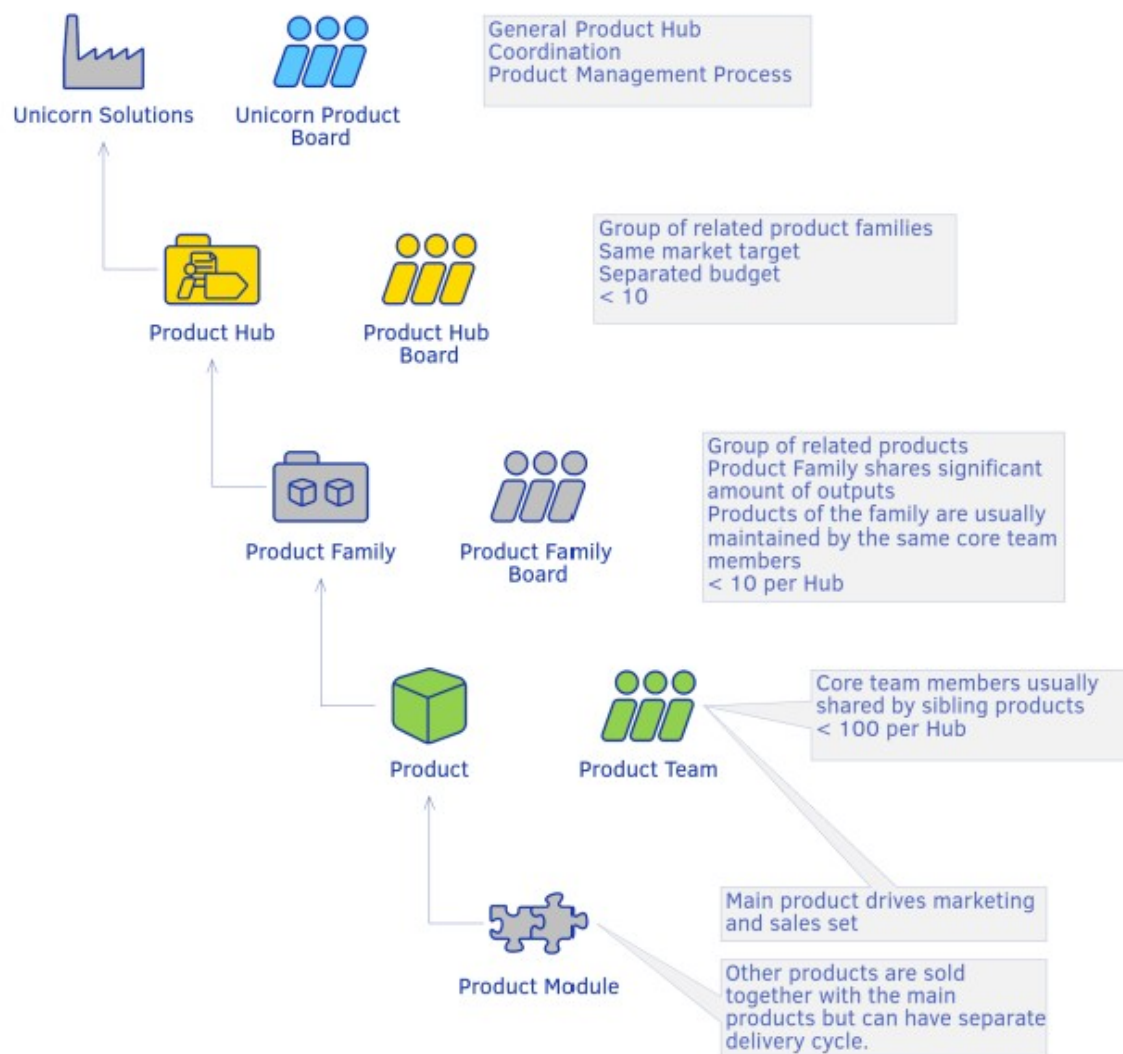
#### **4.1.3 Unicorn Solution a.s.**

Unicorn Solution a.s. (dříve Unicorn Universe a.s.) je společnost vyvíjející onClick produkty pro fyzické i právnické osoby. onClick produkty neboli uuApps využívají digitální stavebnici Unicorn Universe, jež obsahuje jednotlivé komponenty, které byly vytvořeny pro kteroukoliv uuApp s využitím principů uuAppFramework a jsou obecně znovu použitelné ve kterékoliv uuApp díky architektuře The Architecture. Digitální stavebnice Unicorn Universe se využívá pro řízení procesů v rámci holdingu, vně na projektech pro zákazníky holdingu i pro vytváření produktů pro koncové zákazníky a firmy.

Realizace vývoje uuApp probíhá agilně za využití vlastní metodiky uuP, která vychází z obecně známých metodik jako ITIL, RUP, PRINCE nebo SCRUM.

**Řízení společnosti Unicorn Solution a.s.** je rozděleno do čtyř úrovní managementu.

Obrázek 13 Organizační struktura Unicorn Solutions a.s. [32]



**Společnost Unicorn Solutions a.s.** odpovídá za koordinaci a strategické řízení společnosti, které je blíže popsáno v kapitole Metodika Unicorn Universe Process – uuP. Odpovědná skupina za řízení společnosti Unicorn Solutions a.s. je unicorn product board. Unicorn product board ve spolupráci s top management office vytváří a aktualizuje metodiky práce v rámci všech hubs v Unicorn Solutions, zajišťuje transparentnost informací, řídí finance společnosti a spolupráci v rámci společností holdingu.

**Product Hub** je jednotka sdružující families se stejnou cílovou skupinou. Odpovědná skupina za řízení této jednotky je product hub board a odpovědnou osobou je hub manager. Každý hub má samostatný budget a management.

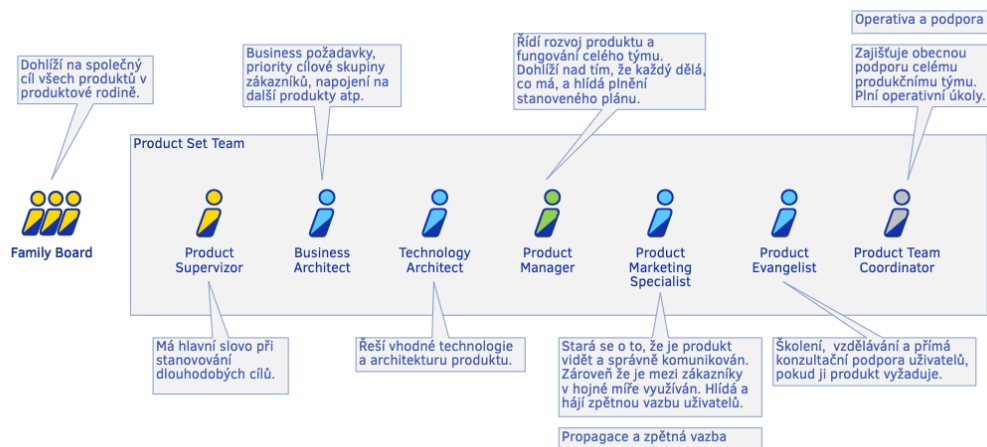
**Product family** je jednotka sdružující související produkty. Odpovědná skupina za řízení této jednotky je product family board a odpovědnou osobou je family manager. V rámci jedné family často dochází ke sdílení členů jednotlivých produktových týmu. (Obrázek 14 Produktový tým (product team) [37])

V rámci Unicorn Solutions si pod pojmem **produkt** můžeme představit uuApp. Odpovědná skupina za řízení produktové jednotky (product unit) je product team a odpovědnou osobou je product manager. Rozsáhlejší produkty se dále dělí do modulů. Každý modul má separátně řízené požadavky na rozvoj. Ty se evidují ve formě tzv. backlogs v aplikaci uuSprintMan a s tím související dodávky výstupů od Software Factory. Díky tomu lze detailně řídit vývoj aplikací, udržovat přehled nad vývojem, efektivně využívat rozpočet a lidské zdroje. Členové produktového týmu jsou sdílení napříč produkty v product family.

Členy **product team** jsou:

- product manager, který je zodpovědný za rozvoj produktu a fungování celého týmu,
- technology architect, dohlíží na dodržení technologických standardů stanovených Unicorn Application Framework a uuApp,
- business architect, který spravuje business požadavky, priority cílové skupiny a spoluvytváří zadání pro vývojový tým,
- product supervisor, většinou člen board hubu nebo family, má hlavní slovo při stanovení dlouhodobých cílů,
- product marketing specialist, pečuje o webové stránky a komunikaci produktu uvnitř společností holdingu i vně, ke každé novince v aplikaci připravuje informační článek pro uživatele,
- product evangelist má know how o produktu, které šíří mezi uživatele a členy product teams jiných uuApps,
- product team coordinator, pomáhá řídit product team, spoluprací mezi jednotlivými členy i spoluprací napříč uuApps, je pravou rukou product manager a plní všechny úkoly mu svěřené.

Obrázek 14 Produktový tým (product team) [37]



## 4.2 Unicorn Approach

Společnosti holdingu Unicorn mají specifický přístup k řízení spolupracovníků a společností jako takových, přístup stojí na klíčových myšlenkách a principech. Je to soubor postupů, principů, myšlenek a zásad, které je vhodné aplikovat při tvorbě softwaru. Aby spolupracovníci dodržovali doporučené zásady, postupy a metodiky, je nezbytně nutné, aby byly všechny oblasti podpořeny systémem, který jejich dodržování podpoří, což bylo jedním z důvodů vzniku metodiky Unicorn Universe Process (uuP), Unicorn Application Framework a také uuApp Framework a Unicorn Mobile-First IoT-Ready (neboli Unicorn Architecture nebo The Architecture).

*Unicorn Approach (Přístup Unicorn) je principiální přístup k podniku a podnikání, ve kterém z Vámi zvoleného portfolia produktů a služeb a s ohledem na Vámi preferovanou dělbu práce, preferované technologie a odhadovaná rizika, odvodíte organizační strukturu, popíšete a předepíšete pracovní postupy jednotlivých podnikových procesů pro všechny zúčastněné role (z řad spolupracovníků, zaměstnanců, dodavatelů atp.), zdůrazníte klíčové produkty a meziprodukty, pro které rovněž předepíšete strukturu informací a uvedete tento mechanismus v život. Váš podnik se pak stane dobrým systémem, který více méně předvídatelně plní účel, pro který byl zřízen. [31]*



Základními pilíři jsou:

- systémový přístup,
- delegace kompetencí,
- zaměření na zákazníky a spolupracovníky,
- orientace na cíl,
- standardizace procesů,
- průběžné hodnocení,
- sdílení informací,
- paměť a poučení,
- soustavné zlepšování,
- stabilita a bezpečnost. [38]

Od roku 1999 byly projekty Unicornu hodnoceny podle KKTR metodiky, každá položka projektu byla hodnocena z hlediska kvality, kvantity, termínu a rozpočtu. Od roku 2019 hodnotí Unicorn projekty v kritériích náklady, kvalita, kvantita (rozsah), termín, přidaná hodnota (benefits) a risk, “CRaB QueST”. Benefit a Risk je řízen podle výstupů konzultace se zákazníkem. Rizika se řídí pomocí kombinace různých metodiky od RUP po PRINCE2, kombinovaných ve vlastní metodice standardizovaných postupů uuP.

Důležitým předpokladem úspěchu je srozumitelně zaznamenat a popsat své myšlenky, které lze efektivně sdílet s okolím. Pro splnění tohoto předpokladu byl vyvinut nástroj pro vizuální modelování a komunikaci **Unicorn Universe Business Modelling Language (uuBml)**, jehož cílem je usnadnit prezentaci myšlenek okolí jak zákazníkům, partnerům, tak i v rámci interních týmů. Principy uuBml spoléhají na obecné znalosti symboliky a lidskou představivost, proto jsou tyto diagramy snadno pochopitelné i neznalému čtenáři. uuBml vychází a je inspirován vizualizačním jazykem UML (Unified Modeling Language), důležitost objektů je reprezentována barevně, a to v tomto pořadí: nejdůležitější zelená, žlutá, modrá, tmavě šedá, světle šedá, dále se využívá červená barva pro problematické objekty. Pro jednotnost interpretace jsou všichni spolupracovníci během vstupních školení detailně seznámeni s principy uuBml. V této práci se čteně vyskytují diagramy a schémata vytvořeny pomocí uuBml.

#### 4.2.1 Metodika Unicorn Universe Process – uuP

Od roku 1990 používá Unicorn různé metodiky. Využívání metodik je na začátku volbou, ale jakmile je metodika jednou zvolena, stává se její aplikace povinností. Všechny metodiky jsou dynamické a neustále se vyvíjejí a zdokonalují. Každá metodika přináší něco prospěšného – zvýšení efektivity, nižší rizika, efektivní komunikaci členů týmu, zrychlení atd. Nestačí metodikám porozumět, je nezbytně nutné mít kvalitní informační systém, který podporuje (umožňuje) jejich aplikaci v praxi. Pokud není metodika podpořena informačním systémem, je velmi náročné až nemožné ji efektivně aplikovat a kontrolovat její dodržování. Metodické postupy je nutné aplikovat napříč holdingem jednotně, aby byla zachována kvalita. Bez automatizace a kontroly je vysoké riziko nedodržení stanovených postupů a tím způsobená rozdílná kvalita aplikace metodiky, což by mohlo vést k různé kvalitě výstupů jednotlivých týmů. Toto bylo v roce 2000 jedním z hlavních důvodů vzniku Unicorn Universe.

Holding Unicorn nejvíce inklinuje k vlastní metodice Unicorn Universe Process, která vychází z metodik, mezinárodních standardů, doporučení, mezi které patří ITIL, RUP, PRINCE, SCRUM, IDEF a dlouhodobé vlastní zkušenosti.

*S metodikou Rational Unified Process (RUP) to býval obrovský problém – když se ho někdo pokusil použít správně, vždy selhal, protože toho bylo příliš mnoho na zvládnutí. I když samotná metodika byla dobrá, bez dobrého informačního systému byla nezvládnutelná. A protože takový informační systém neexistoval, začali jsme budovat uuBusinessTerritory v roce 2000. Přesněji řečeno, byl to jeden z důvodů. Například existuje projekt s deseti případy použití (use cases). V metodice existuje jedna šablona pro vytvoření Vize, jedna šablona pro specifikaci případu použití (use case) a jedna šablona k popisu testovacího scénáře (test scenario). Ale při použití metodiky vytvoříte jednu vizi produktu, deset specifikací případu použití a může to být sto testovacích scénářů. Ve společnosti Unicorn realizujeme 500 takových projektů ročně s 2400 lidmi. Bez systému to nemůže fungovat. [39]*

Výše uvedené metodiky, standardy a doporučení jsou jasným návodem k úspěšnému ukončení projektu, jejich slabým místem je podpořením informačním systémem, který by usnadnil aplikaci metodiky a kontrolu dodržování metodiky.

Každý projekt holdingu pro zákazníka musí být dodělán. Pokud se projekt nedaří, musí být dodělán z rozpočtů projektů, které se povedly. Cílem metodiky uuP je maximálně zvýšit pravděpodobnost úspěchu a dosažení optima. Toho lze dosáhnou edukací a kontrolou dodržování stanovených postupů. Podpořením metodiky informačním systémem a automatizací je zajištěno zjednodušení dodržování postupů daných metodikou, kontrola dodržování metodiky a její revize v čase. Cílem metodiky uuP není dosažení 100 % úspěchu, ale zásadní zvýšení pravděpodobnosti úspěchu. Metodika pomáhá k předvídatelnosti jednotlivých procesů.

Metodika Unicorn Universe Process rozděluje řízení firmy do dvanácti oblastí:

(Příloha 1 Dvanáct procesů metodiky Unicorn Universe Process [31])

Klíčové procesy:	Strategie	Podpůrné procesy:	Management
	Marketing		Růst a vývoj
	Obchod		Systém a podpora
	Produkce		Lidé
	Péče		Finance
			Majetek
			Know-how

V této práci bude popsán pouze jeden proces z metodiky, konkrétně proces strategie, který souvisí s životním cyklem produktů firmy Unicorn Solutions.

**Strategie** v pojetí uuP popisuje produkty a služby, které podnik nabízí s důrazem na jejich přidanou hodnotu.[40]

Proces strategie se pravidelně opakuje a řídíme jej v uzavřených cyklech (strategické období). Každé strategické období má začátek, konec a pevnou dobu trvání. Společnost Unicorn doporučuje volit dobu trvání strategického období jeden rok. Zejména z důvodu připomenutí důležitých bodů dlouhodobé vize spolupracovníkům a konkretizování dílčích cílů.

V dřívější literatuře se můžeme setkat s názvem Unicorn ES Powered Company, metodika byla v roce 2018 přejmenována na Unicorn Universe Process (zkráceně uuP).

Pro vedení podnikové strategie je podle uuP nutné provádět tyto procesy [40].

Nejdříve vzniká **podniková vize**, která obsahuje dlouhodobé poslání a sumarizace myšlenek, které jsou pro daný podnik a jeho dlouhodobé směřování klíčové. Vychází ze strategických dokumentů předchozích období, popř. ze situačních analýz.

Z vize jsou vytvořeny specifikované dílčí **cíle a finanční rozpočet**. Zároveň jsou stanoveny konkrétní milníky a kritéria pro hodnocení plnění cílů.

**Kick-off** seznamuje spolupracovníky napříč podnikem s výstupy z předchozích dvou procesů. Každý spolupracovník musí znát svoji úlohu pro dosahování cílů, které z vize vyplývají. V Unicorn probíhá Kick-off formou osmihodinové konference, které se osobně účastní klíčoví spolupracovníci. Všem spolupracovníkům je konference zprostředkována online přenosem, ke konferenci je umožněn přístup ze záznamu.

V průběhu strategického období je vyhodnoceno plnění cílů z vize na základě předem stanovených metrik a jsou identifikovány případné příležitosti k vylepšení. Výsledky jsou představeny všem spolupracovníkům během **čtvrtletního vyhodnocení**.

Na konci strategického období probíhá **závěrečné vyhodnocení**. Podnik hodnotí naplnění vize, dosažené cíle a plnění rozpočtu. Závěrečné vyhodnocení je následně využito pro stanovení vize podniku v dalším období. Je kladen důraz na poučení ze vzniklých situací a standardizaci osvědčených postupů. [40]

Jelikož je trh nepředvídatelné prostředí, může vzniknout mimořádná událost, která se může stát příležitostí (např. získání velké zakázky) nebo hrozbou (např. ztráta důležitého klienta). Tyto mimořádné situace mohou mít na plnění podnikové strategie velký vliv. Je nutné, aby podnik na tuto situaci strategicky zareagoval, proto do standardního procesu strategie vstupuje proces Strategické překvapení. Příkladem může být reakce společnosti VIG Production na globální pandemii v roce 2020, kdy během pár dnů společnost přesunula všechny plánované konference do online prostředí formou online přenosů ze studia.

Metodika uuP také definuje, jak postupovat, když se projekt vymyká kontrole. Mezi důležitá pravidla v těchto situacích patří:

***Včasná a správná eskalace** – Není chybou mít problém, závažnou chybou je nějaký problém nepřiznat. Manažeři i zaměstnanci musí být vedeni k tomu, aby o problémech informovali své nadřízené co možná nejdříve. Včasné pojmenování a následné sledování*

problému pomáhá při jeho řešení. Čím později o problému víme, tím závažnější mohou být jeho důsledky.

**Pravidlo STOP** – Pokud během činnosti nemůžeme nebo nevíme, jak postupovat dál, je lepší raději svou činnost pozastavit (nebo zpomalit), než bez pevné ruky v řízení pokračovat v nekontrolované činnosti dál a zběsile přitom plýtvat zdroji.

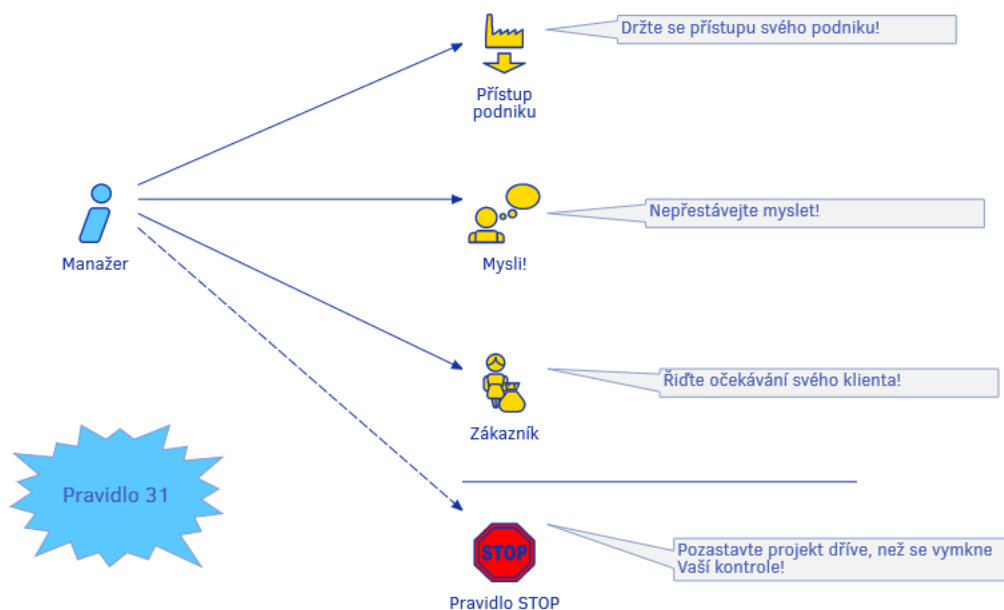
**Pravidlo 31** – V případech, kdy se Vaše práce může dostat z nejrůznějších důvodů do problémů (nebo to alespoň hrozí), nezapomeňte na Pravidlo 31, které říká:

1. Držte se přístupu svého podniku. Tzn., dodržujte postupy ověřené praxí (tj. ověřené znalosti firmy nebo její metodické standardy). Každá firma má svůj přístup, který by měla znát a cíleně ho učit své zaměstnance.
2. Nepřestávejte myslet! A tedy přemýšlejte u všeho, co děláte. „Má to smysl? Jde to dělat lépe nebo efektivněji?“
3. Řiďte očekávání svého klienta. Aby byla daná činnost hodnocena úspěšně, je třeba neustále sledovat očekávání zákazníka s plánovanými výsledky a přínosy ze strany dodavatele. Zákazník musí vědět, co mu naše konání přinese, a musí to chtít.

(+1) A pokud se projekt vymyká Vaší kontrole, tak ho okamžitě zastavte!

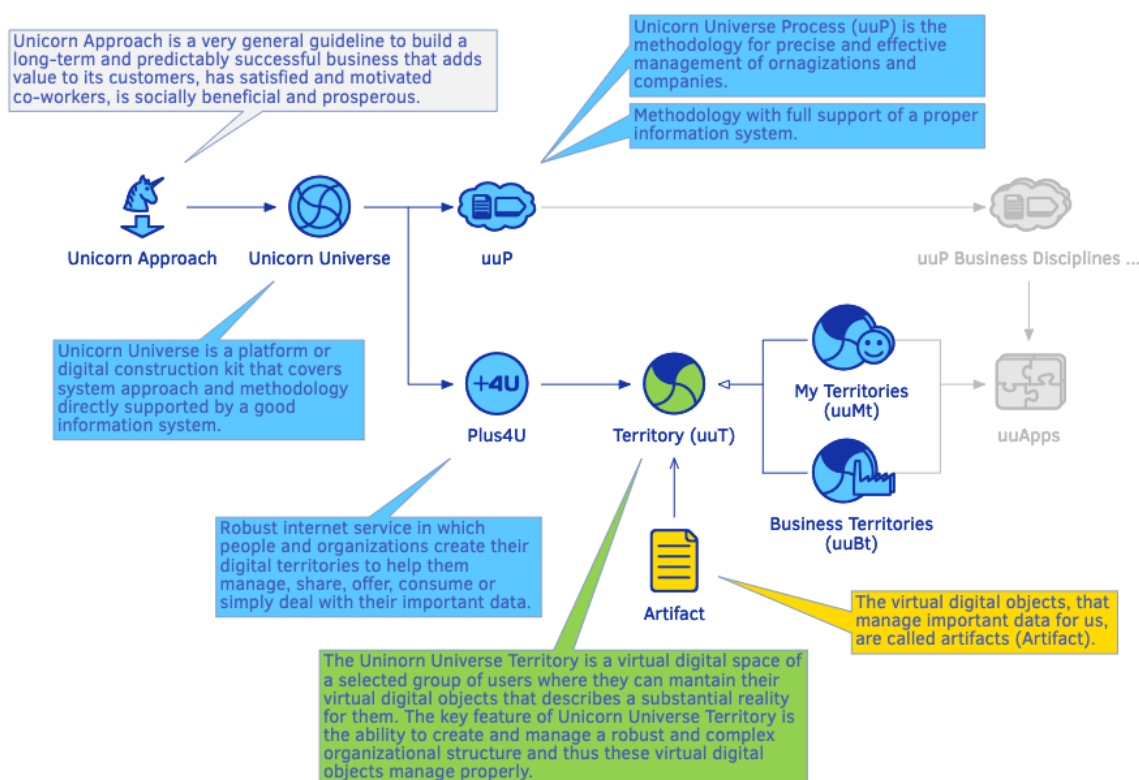
Tzv. Pravidlo STOP. [41]

Obrázek 15 Pravidlo 31 [41]



Mezi klíčové myšlenky a pojmy uuP patří **teritorium**, což je samostatný, nezávislý logický celek (prostor) v rámci internetové služby Plus4U. Tento digitální prostor je určen vybrané skupině uživatelů, ve kterém mohou udržovat své virtuální digitální objekty, jež popisují pro ně podstatné části reality.

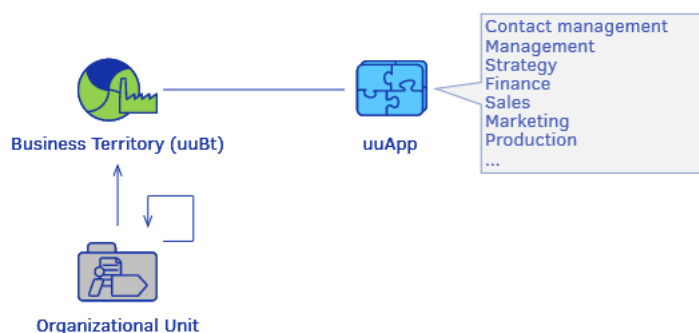
Obrázek 16 Schéma vazeb mezi Unicorn Approach, Unicorn Universe, Plus4U, uuP a uuTerritory [42]



V uuP jsou definovány dva typy teritorií.

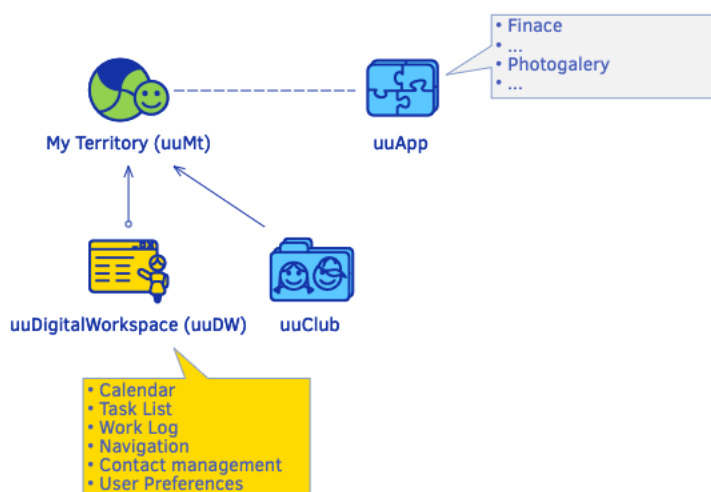
Prvním typem je **uuBusinessTerritory**, jež reprezentuje podnikový informační systém. Vnitřní struktura uuBusinessTerritory je vytvořena pomocí organizačních jednotek (organizational unit), které reprezentují skutečné organizační celky (jednotky) v podniku. Umožňuje digitalizovat a automatizovat procesy, shromažďovat data, vytvářet nové dokumenty, informace a spravovat přístupová práva. Obecně zjednodušuje komunikaci v organizaci a poskytuje možnost spravovat všechny informace do jednoho místa.

Obrázek 17 Schéma uuBusinessTerritory [43]



Druhým typem **uuMyTerritory** reprezentuje osobní prostor uživatele služby Plus4U, ve kterém může spravovat svůj osobní život od financí přes fotografie až po svůj čas.

Obrázek 18 Schéma uuMyTerritory [44]

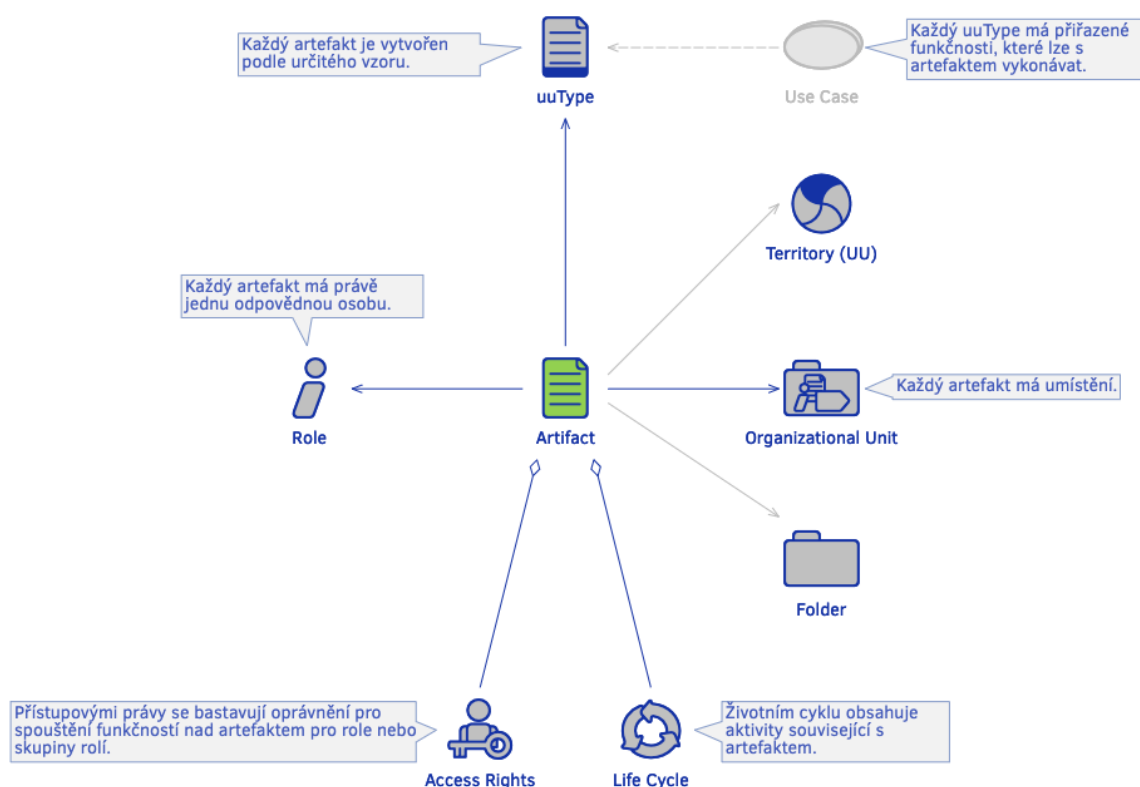


**Artefakt** uchovává záznam stavu produktu v daném okamžiku v minulosti a odhad budoucnosti všech důležitých atributů a produktů tak, aby odhadovaný záznam co nejvíce odpovídal skutečnosti. Nese věcné informace o produktu, co se s ním dělo. Na základě těchto informací lze hodnotit vývoj produktu v minulosti a řídit vývoj produktu v budoucnosti. Je základní nositel informace, který je odvozen od své předlohy (šablona, tzn. uuType). Jedná se o „chytrý“ dokument, pomocí kterého lze ukládat, spravovat a organizovat jakékoliv informace. Artefakt může představovat jak hmotné objekty (automobily, zaměstnance, návrhy, plány, dokumentace a další), tak nehmotné předměty (nápad, úvahy,

know-how, vize, obchodní příležitosti...) nebo události (setkání, konference, večírky, školení...) a je jejich digitálním dvojčetem.

Za každý artefakt je odpovědná právě jedna osoba (role). Artefakt má vždy jasně dané umístění, na jehož základě vyplývají přístupová práva k artefaktu. Artefakt může nabývat různých stavů, jež jsou uloženy v životním cyklu, který pomáhá řídit artefakt. Přístup k artefaktům je řízen pomocí přístupových práv, a to tak, aby se člověk dostal k informacím, které potřebuje, ale zároveň neviděl informace, na které nemá přístupová práva, a tedy je vidět nemá.

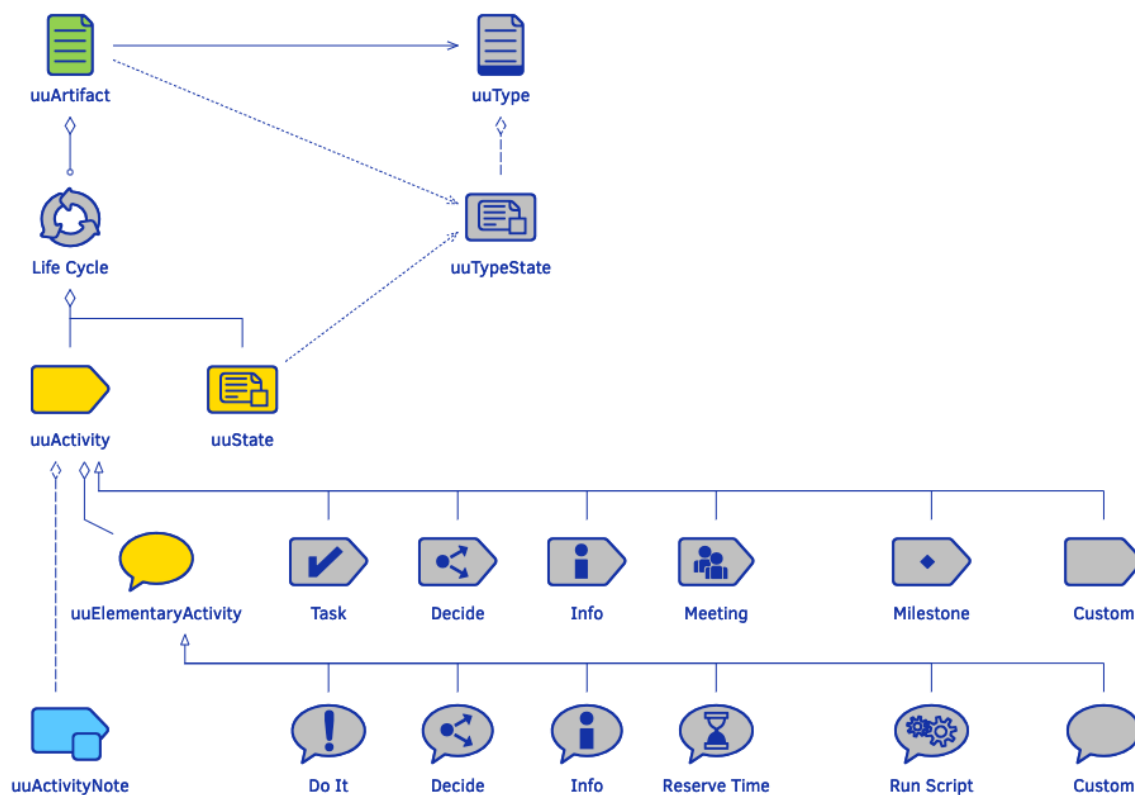
Obrázek 19 Základní vlastnosti artefaktu [45]



Každý složitý úkol nebo proces je rozložen na jednotlivé činnosti (aktivity). Aktivita je akce, která má dobu trvání a cíl. Na naplnění cíle se podílí zadavatel (osoba, která aktivitu řídí) a alespoň jeden řešitel, který aktivitu provádí. Aktivita může být různého druhu: úkol, rozhodnutí, informace, rezervace času, milník, chat a další. Pokud má aktivita více než právě jednoho zadavatele, a právě jednoho řešitele, obsahuje elementární aktivity, díky čemuž může zadavatel snadno sledovat stav jednotlivých aktivit jím zadaných a vidí stav jednotlivých elementárních aktivit, které reprezentují výsledky jednotlivých řešitelů.



Obrázek 20 Schéma vazby mezi artefaktem, aktivitami a elementárními aktivitami [45]



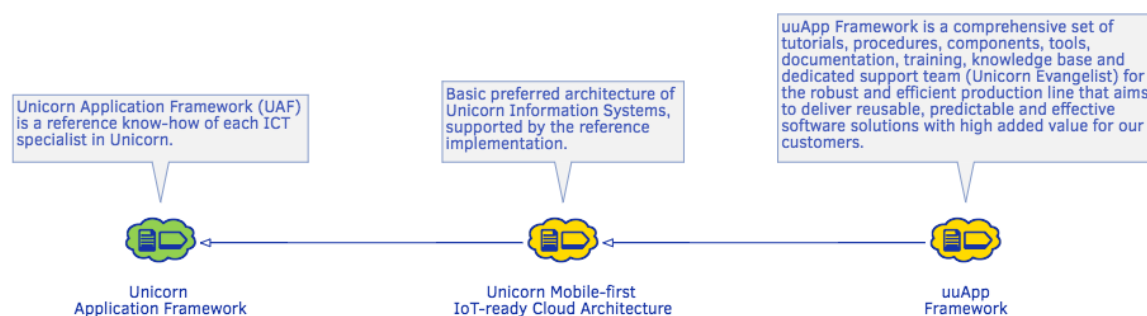
#### 4.2.2 Unicorn Application Framework

Unicorn Application Framework (UAF) obsahuje referenční základ znalostí každého informačního speciality ve společnosti Unicorn, díky němuž mohou vyrábět špičkový software a čerpat z jedinečného firemního know-how. Komplexně řeší potřeby Unicornu a standardizuje každou část výroby software v Unicornu. UAF je kombinací všech výše zmíněných myšlenek a není jednoznačně definováno přesné vnoření těchto přístupů, metodiky a myšlenek, všechny se prolínají. [46]

Unicorn Application Framework (Příloha 2 Struktura Unicorn Application Framework [46] obsahuje kompletní pracovní postupy a návody pro návrh, výrobu, provoz a servis softwarových řešení, dále referenční architekturu, hotové komponenty a nástroje s kompletní dokumentací. Součástí je i konstantní vzdělávání spolupracovníků formou příkladů, školení a rozsáhlou znalostní bází, jež je aktivně podporovaná odbornými fóry a týmy specialistů. [46]

Dále obrázek obsahuje tři logické vrstvy, které se mohou vzájemně překrývat a doplňovat. UAF poskytuje komplexní pohled na procesy a návody, architektura Unicorn Mobile-First IoT-Ready je referenční architektura a uuApp Framework je aplikovaná architektura Unicorn Mobile-First IoT-Ready za dodržení všech pravidel UAF.

Obrázek 21 Jednotlivé logické vrstvy [46]



**Unicorn Mobile-First IoT-Ready Cloud Architecture** je cloudová architektura Unicorn Mobile-First IoT-Ready (neboli Unicorn Architecture nebo The Architecture) je zásadním přístupem k návrhu a implementaci softwaru. Spojuje soubor klíčových principů a návrhových vzorů a vytváří vysoce koherentní plán, jak by měl být moderní aplikační nebo informační systém navržen, postaven a provozován. [46, 47]

**uuApp Framework** je kompletní sada procesů, návodů, komponent, nástrojů, dokumentace, školení, znalostní databáze a týmů, podpory pro výrobu softwarových řešení v rámci Unicorn Application Framework. Podrobně popisuje ideální implementaci referenční architektury UAF a zavádí počáteční postup výroby aplikací v Unicorn. Rámec uuApp představuje efektivní využití UAF při zajištění maximální eliminace rizik a maximálního využití know-how a komponent. [46]

## 4.3 Životní cyklus uuApp

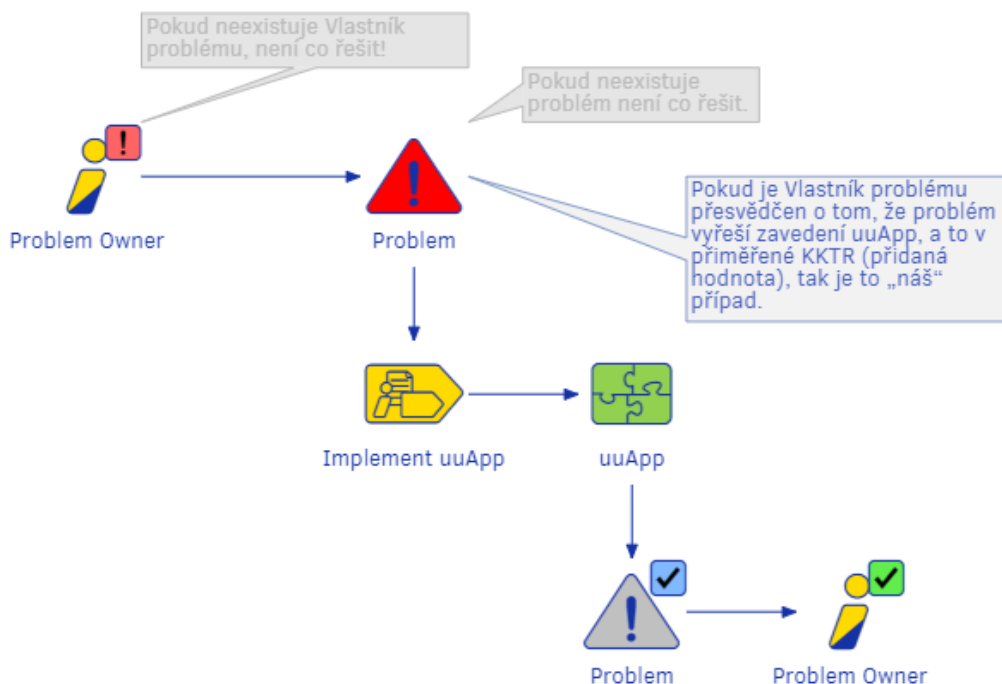
Unicorn Solution a.s. vyrábí onClick produkty, konkrétně uuApp. Jedná se o aplikace postavené s využitím uuAppFramework a The Architecture.

*Definice: uuApp je aplikace vytvořená pomocí uuApp Framework. Spouští se v uuCloud: Používá identitu Plus4U pro autentizaci a uuBusinessTerritory pro autorizaci. [47]*

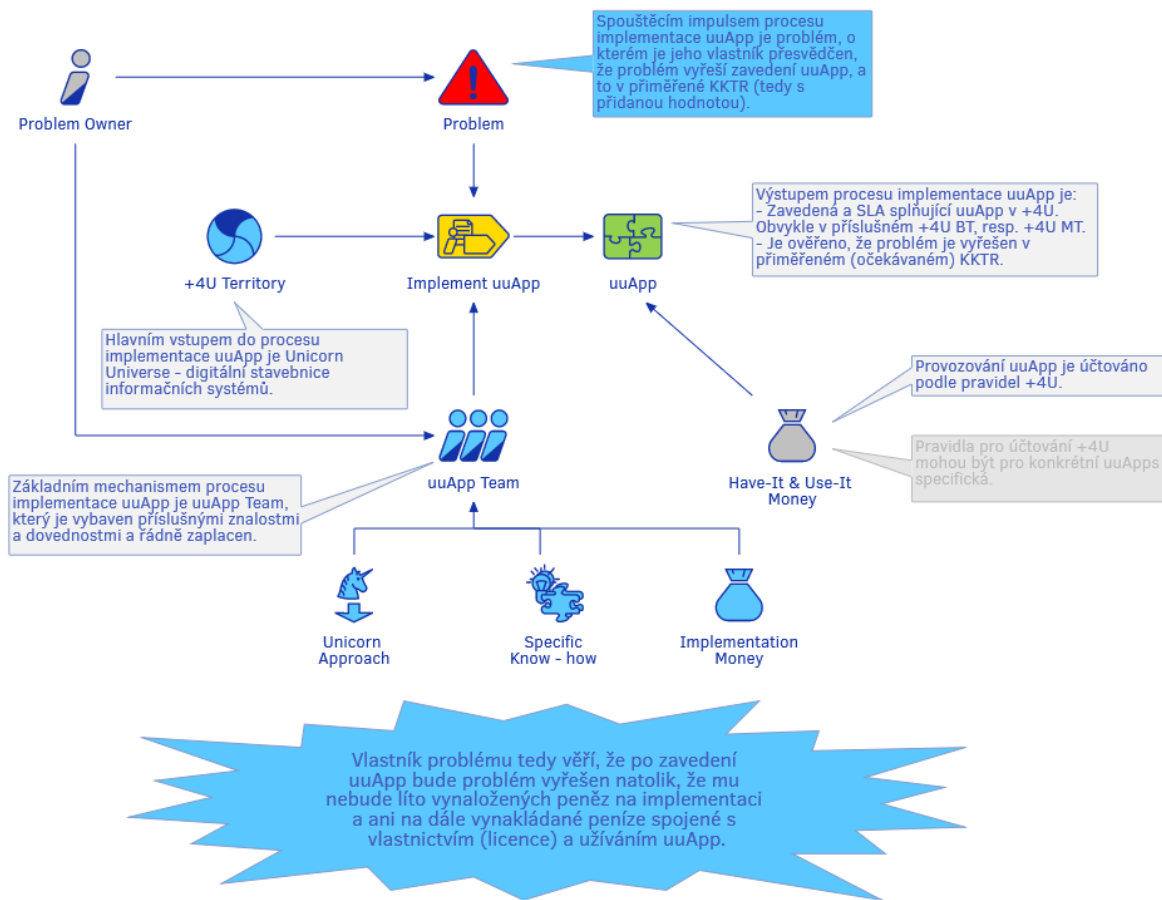
### 4.3.1 Model životního cyklu uuApp

Prvním impulsem k vývoji nové uuApp je problém, potřeba uživatele. Pro zahájení implementace nové uuApp je nezbytně nutné, aby existoval vlastník problému. Vlastníkem problému může být product manager, cílový zákazník nebo výkonný tým pracující pro konkrétního zákazníka.

Obrázek 22 Řešení problému z pohledu vlastníka problému [48]



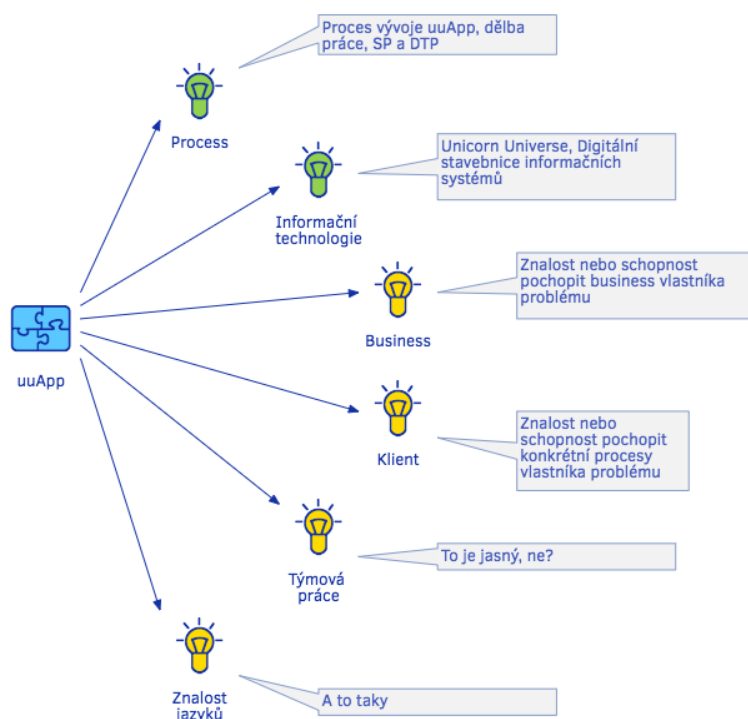
Obrázek 23 Řešení problému implementací uuApp [9]



Podle Vladimíra Kováře je předpokladem pro zahájení implementace uuApp splnění **5+1 potřebných znalostí uuApp**:

- znát procesy vývoje uuApp a dělby práce v týmu,
- ovládat informační technologie uuApp Framework,
- znát business zákazníka (vlastníka problému), tzn. disponovat znalostmi pro doručení hodnoty,
- znát prostředí zákazníka, být schopen pochopit konkrétní procesy vlastníka problému,
- být týmový hráč, protože tvorba uuApp je týmová práce,
- znalost jazyků, protože Unicorn je globální firma a primárním jazykem je angličtina. [9]

Obrázek 24 Pět plus jedna potřebných znalostí uuApp [49]



### Fáze životního cyklu uuApp:

- přípravná fáze produktu,
- příprava vývoje,
- vývoj,
- provoz.

**Přípravná fáze produktu** začíná detailním popisem problému, jež má uuApp řešit definováním scénářů, jak k problému může dojít a jak by jej šlo vyřešit, pokračuje identifikací uživatelů, tvorbou vize produktu a v neposlední řadě i návrhem finanční návratnosti.

**Vize** produktu obsahuje popis aplikace – její poslání, jaký problém (potřebu) uuApp řeší, co je její přidanou hodnotou, textový popis je doplněn o uuBml obrázky. Vize je tvořena na základě problému, požadavků zainteresovaných osob (stakeholders, qualified users). Jedním ze způsobů sběru požadavků jsou uživatelské příběhy (user stories) a jiné požadavky (non-functional requests).

**A4 produktu** podle standardizované předlohy popisuje produkt, zákazníka, konkurenci, prodejní kanály, obchodní disciplínu a je spolu s vizí základním stavebním kamenem.

**User stories**, příběhy uživatelů, popisují funkčnosti, které by měla aplikace poskytovat z pohledu koncového uživatele. User story obsahuje jednoduchý, jasný název, popis z pohledu koncového uživatele ve formátu „jako uživatel aplikace potřebuji mít možnost nahlásit chybu v aplikaci, aby byla odstraněna“. Obsahuje popis, z jakého důvodu by taková funkčnost měla být implementována, kontext, v jakém bude funkčnost používána, popis cílové stavu, v neposlední řadě detailní popis kroků od problému po cílový stav. Jednu user story může definovat více osob, zainteresovaná osoba definuje základní popis a produktový tým user story dokončí, aby mohlo být připraveno zadání pro uuApp designer, který připraví návrh řešení. Návrh řešení následně implementuje uuApp developer. User stories jsou stále aktualizovány a vytvářeny po celý životní cyklus produktu.

**Non-functional requests**, požadavky na aplikaci, neobsahují konkrétní funkčnost. Patří mezi ně například požadavky na aplikaci vyplývající ze smluv (např. servisní smlouva SLA) s provozovatelem aplikace ohledně dostupnosti aplikace, rychlosti oprav aplikace, napojení na jiné aplikace a podobně.

**Produktová rovnice** je nástroj pro předpověď budoucích příjmů a nákladů. Produktová rovnice se průběžně v průběhu životního cyklu produktu vyvíjí a upravuje, v přípravné fázi slouží pro odhad návratnosti a pomáhá v rozhodnutí, zda bude vývoj produktu spuštěn.

Součástí přípravné fáze produktu je i analýza možné **elaborace s aplikacemi třetích stran**. Během ní je zmapováno, jak je řešen problém nebo jeho část řeší jiné produkty na trhu. Během analýzy trhu proběhne seznámení s jejich řešením, licenčními podmínkami a následně je rozhodnuto, zda bude použito jejich řešení v rámci uuApp či nikoliv. V případě, že je potřeba, aby se třetí strana integrovala na uuApp, zjišťují se během analýzy požadavky třetí strany na rozhraní, uuApp se přizpůsobuje jejich požadavkům. Elaborace může být i kombinací výše zmíněných či plnou integrací řešení třetích stran.

Pro eliminaci chyb a zachování konzistence napříč aplikací probíhá příprava vize, user stories a non-functional requests v rámci aplikace uuBusinessRequestMan, která obsahuje návody a šablony pro jednotlivé typy požadavků.

Po vytvoření úvodní vize a základních požadavků je nutné vytvořit model businessu, který má aplikace řešit.

Během fáze **příprava vývoje** definujeme blíže problém, který bude uuApp řešit pomocí business model, který popisuje actors, dílčích produktů aplikace, případů užití (use cases), procesů a vazbami. Jednotlivé případy užití jsou detailně rozpracovány a propojeny s user stories z první fáze. V Unicorn Solution a.s. se pro definování business model používá aplikace uuAppBusinessModelKit, jež obsahuje návody a šablony pro jednotlivé části business model.

**Business discipline**, obchodní disciplína, definuje obchodní odvětví a je vytvořena za konkrétním účelem. Business discipline produkuje finální produkty, zjednodušeně produkty. Popis business discipline se vytváří v rozsahu jedné strany velikosti A4 a nemá definovanou strukturu. Formou popisu business discipline může být text doplněný diagramem v uuBml. Cílem je vystihnout hlavní podstatu businessu. Pokud je potřeba přiblížit jednotlivé části, využívají se pro to tzv. koncepty, které mají volnou (nestrukturovanou) formu popisu.

**Actors**, typová role, reprezentuje konkrétní osobu, umělou inteligenci nebo věc. Má schopnost spustit business use-case. Architektura Unicorn Mobile-First IoT-Ready definuje 4 typy uživatelů – uuPerson, uuAI, uuEE, uuThing. Tedy business use-case může být spuštěn osobou, umělou inteligencí, automatem nebo IoT chytrým zařízením (IoT Smart Box). Podle kvantity dělíme actors na jednotlivce (business role) a skupinu (business group).

**Products** reprezentuje typicky entitu z reálného světa, se kterou se pracuje v rámci business discipline. Typickým znakem produktu je potřeba evidovat si o něm nějaké atributy, případně vazby nebo jeho stavy. Vstupní produkt může být jakákoliv informace z jiné obchodní disciplíny, meziprodukt některého z předchozích procesů v téže obchodní disciplíně nebo finální produkt konkrétní obchodní disciplíny, jednotlivé produkty mohou být od sebe navzájem odvozeny. Tento produkt může být použit jako vstupní produkt jiné obchodní disciplíny. Produkt (product) je více rozepsán v kapitole Produkty Unicorn a.s.

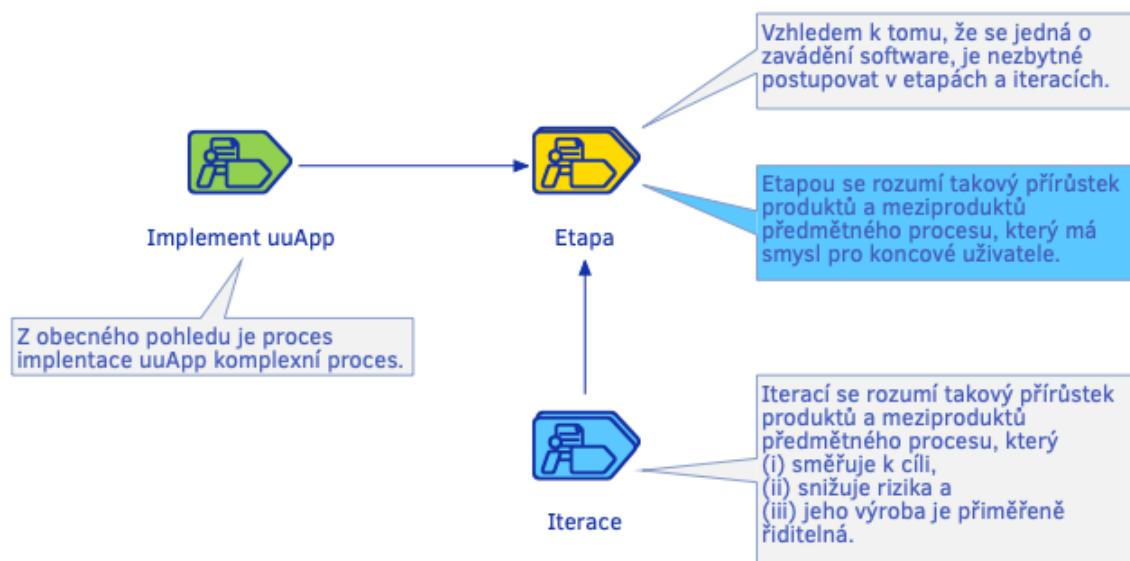
**Business use cases**, množina akcí, probíhají v rámci jedné business discipline. Business use-cases poskytují vysvětlení (kontext), proč a jak actors zacházejí s produkty tak, aby actors dosáhly stanovených cílů a naplnily stanovené scénáře

(business scenarios). Slouží jako zadání pro design (technický návrh řešení) a implementaci technických případů užití (technical use-cases) při automatizaci business discipline.

**Processes** (procesy), série akcí nebo kroků, které vedou k určitému předem stanovenému cíli. Procesy jsou úzce propojeny s business use-cases. Cílem je přetvořit vstupní produkty do výstupních produktů, výstupní produkty mohou být použity jako vstupní do navazujících procesů nebo mohou být již finálními produkty v požadované kvalitě a kvantitě (rozsahu).

V metodice Unicorn Solution a.s. je stanoveno, že každá aplikace má aplikaci pro správu sprintů – uuSprinMan. V aplikaci uuSprintMan má produkt svůj product backlog, kde má backlogy plynoucí z user stories, non-functional request a business model. Backlogy z product backlog jsou využívány pro tvorbu etapy a následně jako zadání pro tým SWF.

Obrázek 25 Etapy vývoje uuApp a iterace [48]



**Vývoj uuApps** probíhá v týmech dodavatelské společnosti SWF. Řízení vývoje uuApps probíhá v dílčích dodávka konkrétních funkčností, které jsou realizovány v rámci etapy za účelem detailního řízení

Product manager připraví zadání etapy, které obsahuje konkrétní nové funkčnosti nebo rozšiřují současné funkčnosti a přináší přidanou hodnotu pro koncového uživatele. Popis jednotlivých funkčností je realizován většinou formou položky z produktového backlogu (Příloha 5 Životní cyklus produktového backlogu [Vlastní zpracování]). Dále etapa obsahuje



předpokládaný rozpočet, např. náklady na jednotlivé sprinty, milníky. Každá etapa je poslána na schválení vedení (board). Po schválení vedením Unicorn Solution a.s. je etapa poslána na burzu řešení buď ve chráněném režimu pro konkrétní tým SWF, nebo bez chráněného režimu, a tedy si ji může vysoutěžit jakýkoliv tým nebo jednotlivec, který dokáže dodat požadované funkčnosti v požadované kvalitě, kvantitě (rozsahu) a za stanovené finanční prostředky.

V této práci je vycházeno z předpokladu, že je etapa realizována pouze týmy SWF, etapy jsou pouze rozvojové nikoliv servisní a každá etapa může mít maximální délku 12 týdnů. (Příloha 3 Průběh etapy [53])

Etapa je rozdělena na jednotlivé zakázky. S vývojem uuApp se pojí typy zakázek, kterými jsou grafický návrh, implementace a testování. Realizace zakázek implementace a testování probíhá v 7denních nebo 14denních sprintech, délka sprintu je pevná. V rámci holdingu Unicorn je dáno, že sprinty mají fixní datum ukončení a jejich délka je pevná, obvykle 14 dnů. Sprint nemůže být delší než 28 dnů.

Tickets se do sprintu plánují podobně jako plánování cíle, kolik tým uplave bazénů. Nemá smysl otázka: „Kolik bazénů uplaveme minulý týden?“, tedy sprint nekončí tehdy, až jsou splněny všechny tickets v něm naplánované, ale v předem stanoveném datu. Nesplněné tickets jsou automaticky nezařazené, ale obvykle jsou přesunuty do sprintu následujícího. Tedy, když je naplánováno, že za 14 dnů tým uplave 120 bazénů, tak se může člen týmu po 14 dnech zeptat, kolik jich tým opravdu uplavat a kolik jich neuplavat. Například tým uplavat 98 bazénů a 22 tedy zbývá. Minulý 14denní sprint tedy skončil uplavením 98 bazénů a 22 bazénu musí tým uplavat v následujících sprintech, např. hned v tom následujícím. Toto rozhodnutí je klíčové a přináší výrazné zjednodušení při plánování zdrojů a organizace práce. Nadcházející části této práce pracují s 14denními sprinty.

Grafické vyobrazení časového rozložení sprintu je uvedeno v příloze. (Příloha 4 Průběh sprintu [53]) Grafické zakázky nejsou rozlišeny na sprinty, ale pouze na milníky.

**Před každým sprintem** probíhá estimation meeting, na této schůzce se připravuje scope (rozsah, kvantita) daného sprintu formou sprint backlog z product backlog. Vývojářský tým následně připraví rozpad jednotlivých položek sprint backlog na tickets (úkoly) (Příloha 6 Životní cyklus ticketu [Vlastní zpracování]).

Během 3 dnů proběhne planning meeting, na kterém SWF představí jednotlivé tickets pro nadcházející sprint a SWF má možnost požádat o upřesnění zadání. Následující nebo též den sprint oficiálně začne a SWF tým zahájí práci na jednotlivých tickets podle priorit stanovených produktovým týmem.

**V průběhu sprintu** po 3-4 pracovních dnech proběhne demo meeting 1 (demonstration meeting) neboli schůzka s ukázkou odvedené práce, během kterého SWF tým představí splněné tickets a proběhne jejich ukáзка. Po demo 1 začne první kolo testování, implementační tým pokračuje v dokončování dalších tickets patřících do sprintu, produktový tým kontroluje dosavadní výstupy a probíhá revize výstupů na SWF od seniorních vývojářů.

Po 7-9 dnech proběhne demo meeting 2, během demo meetingu 2 by měl mít implementační tým téměř hotovo, první kolo testování je dokončeno a zahajuje se druhé kolo testování. Simultánně k vývoji nových funkcí opravuje implementační tým chyby nalezené během prvního a druhého kola testování.

Po 2-3 dnech proběhne acceptance meeting, cílem acceptance meeting je akceptování dodaných výstupů, tzn. implementační tým dodal výstupy zakázky v dohodnuté kvalitě a kvantitě, pokud produktový tým s výstupy souhlasí, proběhne odsouhlasení finální částky za sprint pro testovací a implementační týmy, uzavřou se obě zakázky.

**Plánování nadcházejícího sprintu** probíhá v rozmezí 2-5 dnů od demo meeting 2 probíhá estimation meeting k následujícímu sprintu tak, aby na sebe jednotlivé sprinty plně navazovaly podle plánu fáze.

**Fáze provoz** začíná, jakmile je uuApp připravena pro použití koncovým uživatelem. Dochází k předání aplikace do správy jednotky onClick Solutions (Plus4U Net a.s.), na straně Product Hub (Unicorn Solutions a.s.) stále probíhá rozvoj uuApp a případné opravy. Pro správnou funkci uuApps a její udržitelnou životaschopnost je potřeba zajistit pět procesů (dále 5P). Mezi tyto procesy patří:

- integrace (Integration),
- vývoj (Building up),
- provoz (Operation),
- primární podpora (First line support),
- sekundární podpora (Second line support).

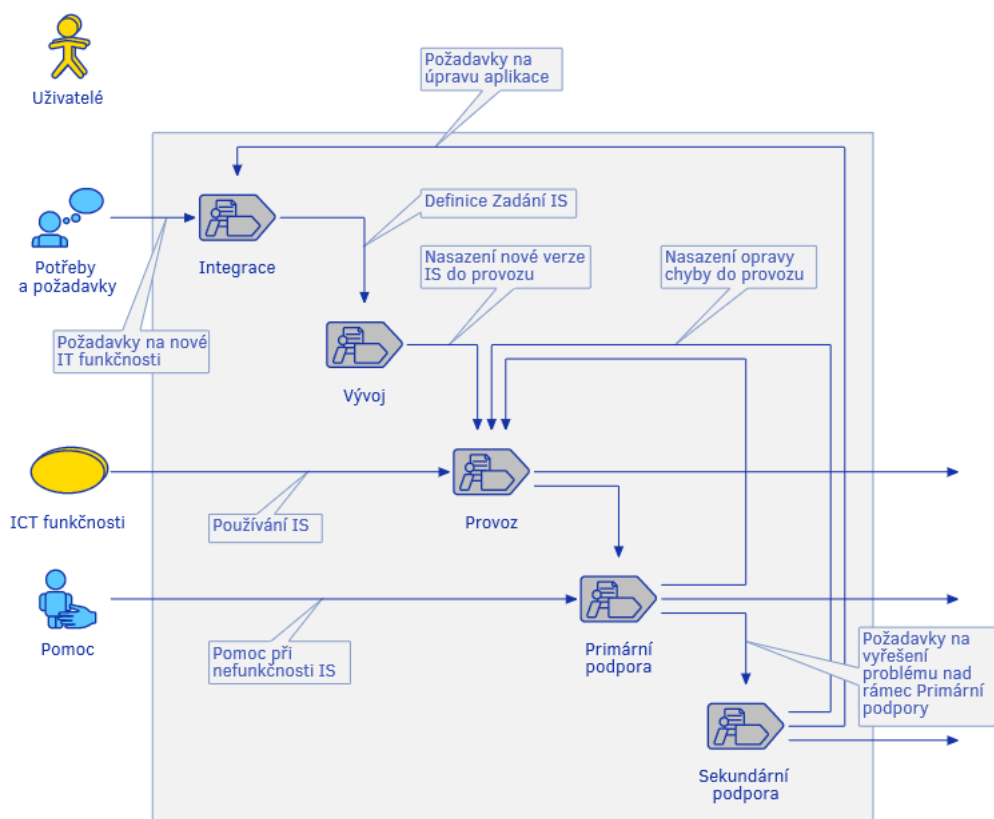
**Prvotní integrací**, vývojem a prvním uvedením do provozu život uuApp teprve započne. Jakmile je uuApp uvedena do provozu, začnou uživatelé generovat otázky a nápady na vylepšení uuApp. Uživatelé a jejich potřeby jsou pro životní cyklus uuApp zcela nezbytní, proto je nutné je vyslechnout a zodpovědět jimi položené otázky, k podpoře péče o uživatele uuApp slouží primární podpora.

**Primární podporu** tvoří tým operátorů z centrálních jednotek (Central Services) využívající aplikace uuFirstLineSupport, která slouží ke sběru požadavků od uživatelů, jejich evidenci a správě. Operátoři primární podpory jsou proškoleni pro zodpovězení běžných uživatelských požadavků. Pokud operátor primární podpory není dostatečně kompetentní k vyřešení požadavku uživatele, přepoše požadavek do sekundární podpory (viz. Obrázek 26 Pět procesů souvisejících s informačním systémem [9]).

**Sekundární podporu** tvoří tým operátorů z centrálních jednotek (Central Services), který je proškolen pro řešení složitějších uživatelských požadavků a většinou má přímé napojení na produktový tým. Pokud je požadavek uživatele spojen s chybou systému, tým sekundární podpory je kompetentní chybu opravit, pokud si není řešením jist, může kontaktovat product manager, který rozhodne, zda ji má opravit tým sekundární podpory nebo bude zařazena na backlog. V případě, že se jedná o požadavek na rozšíření funkcí, založí požadavek jako položku (backlog) v product backlog a tým začíná proces Integrace.

**V procesu integrace** jsou spravovány požadavky od uživatelů a zainteresovaných osob. Probíhá určování jejich priority, detailní rozpracování požadavku a tvorba zadání pro vývojový tým. Při tvorbě zadání může product manager rozhodnout o začlenění požadavku do nadcházející etapy. Po ukončení fáze dojde k implementaci nových funkcí do uuApp. Návaznost procesů je zobrazena na obrázku níže (Obrázek 26 Pět procesů souvisejících s informačním systémem [9]).

Obrázek 26 Pět procesů souvisejících s informačním systémem [9]



Příkladem problému k řešení je nedostatek křečků na trhu, který je způsoben nedostatečnou produkcí a zvýšenou poptávkou po těchto hlodavcích. Následkem toho je pro zákazníka obtížné sehnat křečka. Řešením problému je založit křeččí farmu. Nejprve musí být vytvořena business discipline uuP farma křečků (uuP Hamster Farm), která bude přímo podpořena aplikací uuAppHamsterFarmKit. Znamená to tedy, že existuje kniha obchodních požadavků (Business Request Hamster Farm), která je výstupem aplikace uuBusinessRequestMan. Dále existuje kniha obchodního modelu (Business Model Hamster Farm), tato kniha je výstupem aplikace uuAppBusinessModelKit. Také existuje kniha aplikační model (Application Model Hamster Farm), která je výstupem aplikace uuAppModelKit. Tato business discipline uuP Hamster Farm i aplikace se školí na Unicorn Top Gun Academy a Unicorn University.

Každá z aplikací vyrobených v holdingu má vlastní výše uvedené dokumentace (kniha obchodních požadavků, kniha obchodního modelu, kniha aplikační model).

### 4.3.2 Komunikace mezi dodavatelskými týmy a zadavatelským týmem Unicorn Solutions

V kapitole Dělbá práce v holdingu Unicorn a.s. byl nastíněn princip spolupráce uvnitř holdingu a fungování jednotlivých jednotek.

Na straně zadavatelských týmů je product team, který pošle poptávku obsahující zadání na burzu řešení. Poptávku může vysoutěžit tým SWF. Pro poptávku vývoje či servisu aplikace je na straně dodavatele vývojový tým SWF (development team). Pro poptávku testování aplikace je na straně dodavatele testovací tým (test team).

Je preferované, aby na produktu v rámci jeho životního cyklu pracoval stejný tým, který prohlubuje své znalosti v dané oblasti, kterou daná uuApp řeší. Díky poznání osobností a komunikačních stylů jednotlivých členů v průběhu spolupráce se naučí členové efektivně komunikovat, nastaví si vnitřní pravidla, která se postupně ustálí. Pokud by se dodavatelské týmy stále střídaly a měnily, musel by se komunikační plán revidovat častěji a docházelo by k nedorozumění.

Během října 2020 autorka práce byla přítomna komunikaci mezi vývojovým týmem dodavatele a zadavatelem (product manager) z pozice product coordinator. U jednoho konkrétního produktu společnosti Unicorn Solutions a.s. Je nutné zmínit, že komunikace mezi týmem dodavatele a zadavatele je na každém produktu jiná, neboť jsou jednotlivé týmy tvořeny jinými lidmi. Nejedná se tedy o obecné pozorování a o konkrétní případ.

Komunikace probíhala podle komunikačního schématu (Obrázek 27 Schéma komunikace mezi vývojovými týmy SWF a zadavatelem product team [34]), a to zejména během následujících schůzek:

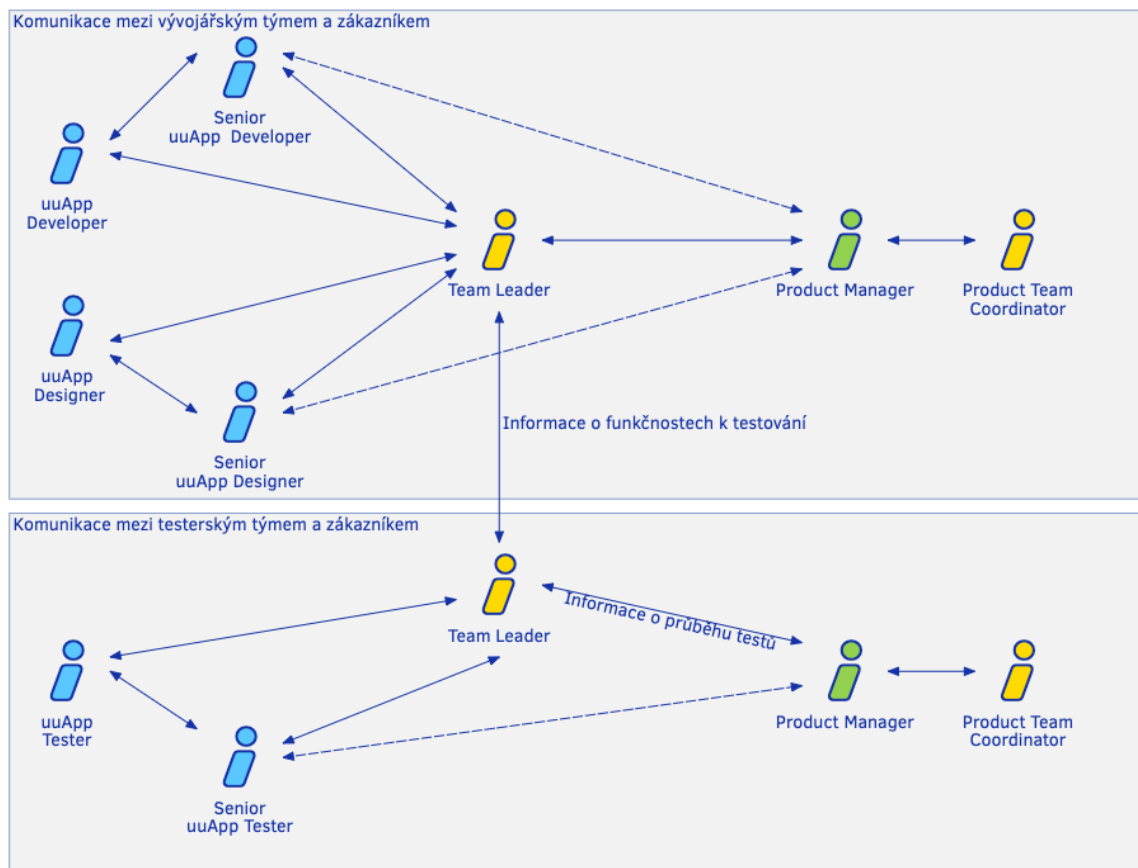
- po vysoutěžení zakázky na burze aplikací byla podepsána smlouva a domluven začátek prvního sprintu zakázky a termín estimation meeting,
- estimation meeting, kterého se za zákazníka účastnili product manager, business architect, technology architect, product team coordinator, za tým dodavatele team leader, senior uuApp designer a senior uuApp developer a bylo předáno zadání a zodpovězeny otázky od týmu dodavatele,
- planning meeting, kterého se za zákazníka účastnili členové product manager, business architect, technology architect, product team coordinator, za tým dodavatele členové team leader, senior uuApp designer a senior uuApp

- developer. Za tým dodavatele testovací zakázky se meetingu účastnili členové team leader a test team. Členové development team leader představili jednotlivé úkoly, které budou v daném sprintu udělány a osobu odpovědnou za daný úkol,
- demonstration meeting 1, kterého se za zákazníka účastnili členové product manager, business architect, technology architect, product team coordinator, za tým dodavatele členové team leader, senior uuApp designer a senior uuApp developer, případně členové, kteří měli otázky. Za tým dodavatele testovací zakázky se meetingu účastnili členové team leader a test team. Členové development team leader představili stav sprintu, proběhla ukázka přírůstku, product team zodpověděl otázky jednotlivých členů development team,
  - demonstration meeting 2, kterého se za zákazníka účastnili členové product manager, business architect, technology architect, product team coordinator, za tým dodavatele členové team leader, senior uuApp designer a senior uuApp developer, případně členové, kteří měli otázky. Za tým dodavatele testovací zakázky se meetingu účastnili členové team leader a test team. Team leader představil stav sprintu, proběhla ukázka přírůstku, v případě otázek od jednotlivých členů development team byly zodpovězeny otázky členy product team,
  - acceptance, kterého se za zákazníka účastnili členové product manager, business architect, technology architect, product team coordinator, za tým dodavatele členové team leader, senior uuApp designer a senior uuApp developer.

Vlivem nedostatečných lidských zdrojů byl do role development team leader jmenován junior uuApp designer bez potřebných zkušeností. V důsledku nedostatečných znalostí a zkušeností si team leader nedokázal vybudovat důvěru týmu. Což nepřímo vedlo k nedodržení postupů definovaných metodikou uuP. Senior uuApp developer komunikoval přímo s product manager bez informování team leadera z důvodu možné časové úspory, kdy problematiku nechtěl vysvětlovat team leader. Product manager a senior uuApp developer komunikovali pouze ústně. O jednáních mezi senior uuApp developer a product manager nebyl proveden zápis v systému a nebyli informováni všichni členové týmů, což v konečném důsledku vedlo k prodloužení demonstration meeting. Na demonstration meeting bylo doručeno jiné řešení, než product supervisor a team leader očekávali a situace musela být vysvětlena a vyřešena. V kapitole Výsledky a diskuse a její podkapitole Komunikace je

návrh kroků ke zlepšení komunikace mezi týmem zadavatele a dodavatele v případě tohoto konkrétního produktu.

Obrázek 27 Schéma komunikace mezi vývojovými týmy SWF a zadavatelem product team [34]



#### 4.4 Životní cyklus vývoje uuApp

Termín uuApp využívá Unicorn pro své aplikace, které jsou založeny na Unicorn Application Framework (UAF) a jsou vyvíjeny pro internetovou službu Plus4U.

Na začátku každé aplikace uuApp je problém, který má daná uuApp vyřešit. Musí být jasně definován vlastník problému a problém. Vlastník problému musí být přesvědčen o tom, že zavedení uuApp vyřeší problém za přiměřených „CRaB QueST“ (náklady, riziko, přínosy, kvalita, rozsah a čas) s přidanou hodnotou. Životní cyklus uuApp vychází z principů metodiky RUP (Rational Unified Process) a dalších. [50, 51]

Fáze životního cyklu uuApp:

- počáteční fáze (Inception),
- elaborační fáze (Elaboration),
- konstrukční fáze (Construction),
- fáze nasazení (Transition),
- provoz (Run Time).

Během **počáteční fáze** probíhá definice požadavků, tvorba zadání (např. specifikace uživatelských příběhů – user stories až 60 % ze všech user stories, případů užití (use cases) a analýza možných způsobů realizace. Výroba prototypů, vznikají první uuSubApps (logická aplikační část, která se dále nedělí) s omezenou funkčností. Pokud jsou známy všechny procesy, dostupné všechny zdroje, může být zahájena konstrukční fáze. V případě že nejsou známy všechny postupy a procesy nebo není dostatek zdrojů, navazuje fáze elaborační.

Cílem **elaborační fáze** je najít většinu komponent a částí systému, které jsou složité a vede k nim jiný postup než k běžným komponentám, naučit se najít věci, které jsou jiné než ostatní a najít k nim postupy, připravit procesy, dále design (návrh) komponent a vytvoření prototypu, který ověří správnost řešení většinou po dílčích částech – v iteracích. Souběžně probíhá revize user stories a tvorba nových user stories, určí se user stories, které jsou výjimečné a slouží k vyzkoušení nového procesu na malém signifikantním vzorku.

Během **konstrukční fáze** probíhá revize, rozšíření user stories, návrh, realizace a testování komplexního řešení. Vývojový tým realizuje jednotlivé obrazovky, implementuje user stories a use cases podle business model. Je nezbytně nutné, aby tým během vývoje vytvářel kompletní, ucelenou dokumentaci do knihy aplikačního modelu (application model) za využití aplikace uuAppModelKit a byla prováděna revize kódu od seniorních členů vývojového týmu. Testovací tým připraví testovací scénáře, které mohou vycházet z user stories. Testovací scénáře validují, že dodané funkčnosti odpovídají původním požadavkům a došlo k naplnění user story. Podle testovacích scénářů je následně testování realizováno.

Ve **fázi nasazení** dochází k nasazení systému na prostředí, kde jej mohou používat koncoví uživatelé. Součástí je ověření splnění očekávání uživatelů během akceptačního testování, doplnění user stories – vývoj nových funkcností podle požadavků uživatelů. Během fáze nasazení běží souběžně starý systém a nově nasazený systém, školení koncových uživatelů.



Po dodání kompletního systému proběhne předání systému zákazníkovi podle předem domluvených podmínek se smluvenými službami během provozu systému, **fáze provoz**.

V případě aplikace je nutné zachovat část kapacity týmu na správu aplikace, inovace aplikace a implementace relevantních požadavků koncových uživatelů. Probíhá sběr požadavků od uživatelů, implantace relevantních požadavků a nasazení nových verzí.

#### **4.5 Proces tvorby zadání v Unicorn Solutions a.s. v období říjen 2020**

Případová studie se zabývá produktem, jehož životní cyklus byl zahájen před vytvořením metodiky a modelu životního cyklu uuApp. Jedná se o obsáhlou aplikaci a díky nedostatku lidských zdrojů není možné aktualizovat změny v jednotlivých dokumentech, které jsou potřebné k dodržení modelu životního cyklu. Ke změnám dochází díky změnám v prioritách při agilním vývoji a stále se měnícímu trhu.

Tvořit detailní zadání pro vývoj je nezbytné a zároveň velmi časově náročné. Pro lepší orientaci v informacích o jednotlivých uuApps by pomohla aplikace na správu produktu. Všechny informace o produktu by byly dostupné z jednoho místa a byl by rozdělen produktový backlog a backlog probíhajícího sprintu. Vývoj této aplikace je jedním z cílů společnosti Unicorn Solution a.s. pro rok 2021.

Pro správu rozvoje produktu je využívána jedna instance uuSprintMana pro řízení produktového backlogu, backlogu aktuálního sprintu i ke správě chyb objevených během testování funkcí vytvořených v probíhajícím sprintu.

V jednom backlogu se tedy nachází přání uživatelů na možné budoucí funkčnosti produktu, požadavky od kvalifikovaných uživatelů, zainteresovaných osob, chyby z produkce, schválené funkcionality probíhající fáze a další.

Zadání probíhá formou business model, stručného popisu v backlogu doplněného o UX návrh, pokud se jedná o vizuální komponenty. Případně je zadání ústně doplněno během estimation meeting, planning meeting a demo meetings. Zadání je ústně na schůzkách upraveno, v ideálním případě je zhotoven zápis ze schůzky a popis backlogu doplněn.

## **4.6 Návrh procesu tvorby zadání požadavku na implementaci nové funkcionality uuApp ve společnosti Unicorn Solution a.s.**

Než bude představen proces tvorby zadání, je vhodné definovat základní typy požadavků. (Příloha 7 Životní cyklus požadavku z primární podpory v uuBml [Vlastní zpracování]) Požadavky jsou sbírány od koncových uživatelů do aplikace primární podpory (uuFirstLineSupport). Kvalifikovaní uživatelé, jimiž mohou být zainteresované osoby, členové produktového týmu, spolupracovníci, mohou požadavky zasílat do aplikace sekundární podpory. Ze sekundární podpory jsou požadavky zaslány do aplikace podporující proces integrace, tato aplikace doposud není vyvinuta. Aktuálně sekundární podporu spravuje produktový tým, který je zodpovědný za řízení rozvoje produktu a řešení uživatelských požadavků. Nyní lze ze sekundární podpory přeposlat požadavek do aplikace na správu sprintů (uuSprintMan). Pokud se jedná o kritickou chybu, je možné požadavek zařadit do aktuálního, běžícího sprintu, aby byla chyba co nejdříve odstraněna. Požadavek může obsahovat nápad na vylepšení aplikace, například formou nové funkčnosti, úpravou stávající funkčnosti a podobně. Pokud je požadavek na vylepšení aplikace detailně definován a je rozhodnuto, že jeho implementace bude přínosem, je možné jej zařadit do nejbližšího sprintu. Za předpokladu, že nápad není detailně zpracován, je nutné před zařazením do backlogu promyslet a detailně zpracovat zadání, tedy zařadit jej na wishlist, promyslet a následně podle priorit zpracovat a zařadit do backlogu. Pro lepší orientaci v informacích o jednotlivých uuApps by pomohla aplikace na správu produktu a procesu integrace. Všechny informace o produktu by byly dostupné z jednoho místa a tato aplikace by poskytla prostředí pro jednoduchou orientaci ve všech aplikacích pro správu produktu, uuApp. Popis primární a sekundární podpory a návaznosti jednotlivých procesů jsou uvedeny v kapitole Model životního cyklu uuApp.

Lze tedy říct, že z požadavku ze sekundární podpory může vzniknout:

- zadání kritické chyby, kterou je nutné zařadit do probíhajícího sprintu, v aplikaci uuSprintMan,
- položka wishlistu, přijetí nápadu, který je nutné blíže specifikovat, v aplikaci uuProductMan,
- nový backlog, přijetí plně definované a schválené funkčnosti, v aplikaci uuSprintMan,

- úkol vytvoření nebo úpravy v application model, v aplikaci uuAppModelKit,
- úkol k vytvoření business model, v aplikaci uuAppBusinessModelKit,
- key task v aplikaci pro správu elementárních činností, v aplikaci uuElementaryManagement,
- požadavek na založení požadavku, v aplikaci uuElementaryManagement,
- a další.

Vzhledem k tomu, že v průběhu času vyvíjí další aplikace, nelze přímo stanovit všechny typy požadavků. Díky jednotné architektuře a frameworku je zajištěno, že všechny aplikace spolu mohou komunikovat, a je tedy možné jakýkoliv požadavek poslat do kterékoliv napojené aplikace.

Požadavek posláný do sekundární podpory se skládá ze:

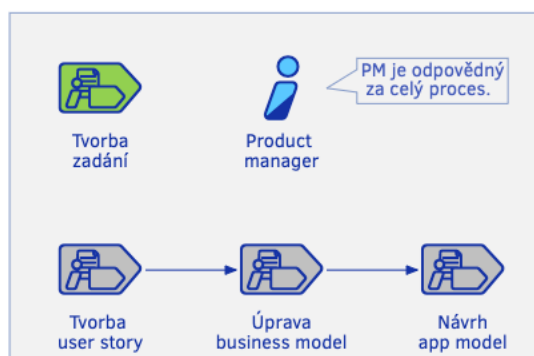
- základních informací (název požadavku, popis požadavku, místo odkud byl požadavek poslán – např. z jaké subaplikace, závažnost požadavku, priorita požadavku, umístění požadavku v systému, čas vytvoření, čas odeslání a čas vyřešení),
- informací o autorovi (jméno, unikátní identifikátor),
- příloh (např. obrázky, videa, související dokumenty a podobně),
- seznamu aktivit, které s požadavkem souvisí,
- vstupy, což je seznam artefaktů, které mohou být použity jako vstupní informace při řešení tohoto problému, například původní požadavek, pokud byl problém přenesen z aplikace primární podpory nebo jiné aplikace,
- referencí, což je seznam artefaktů z jiných aplikací, které se vztahují k tomuto problému. Například jsou vytvořeny v důsledku řešení problému a používají obsah problému jako vstup.

Z důvodů rozsahu a šíře diplomové práce bylo vybráno, že tato **práce popisuje pouze návrh procesu zadání nové funkčnosti**. Všechny diagramy a texty se vztahují pouze na tento jeden konkrétní typ požadavku.

Po přijetí požadavku na novou funkčnost je business architect napsána user story v aplikaci uuBusinessRequestMan, která odpovídá standardům UAF a je podporou metodiky uuP. Jedná se o business discipline sběr požadavků (business request). Následně business architect navrhne business model v aplikaci, která rovněž odpovídá standardům UAF a je

podporou metodiky uuP, business discipline modelování obchodních případů (business modeling). Tato aplikace se nazývá uuAppBusinessModelKit. Díky těmto návrhům se zjistí, zda již máme pro danou funkčnost ověřený standardizovaný postup. Pokud není tento předpoklad splněn a standardizovaný postup není dosud navrhnut, vybere se signifikantní vzorek a nalezne se standardizovaný postup v aplikaci uuAppModelKit, která podporuje business discipline návrh a vývoj aplikace (uuApp design and development).

Obrázek 28 uuBml schéma procesu tvorby zadání [Vlastní zpracování]



Proces zadání lze tedy rozložit na tři dílčí procesy:

- tvorba user story,
- úprava business model,
- návrh úprav app model.

Osobou zodpovědnou za všechny dílčí procesy je produktový manažer.

#### 4.6.1 Cíl procesu

Cílem procesu tvorby zadání požadavků na implementaci v Unicorn Solution a.s. je vytvořit konkrétní zadání práce pro dodavatelský tým ze Software Factory. Toto zadání eliminuje množství variant řešení daného požadavku, zefektivňuje řízení očekávání a vydávání funkcí na produkci. Zadání může mít více forem, tato práce se zabývá pouze zadáním nové funkčnosti. Základní doporučenou formou zadání pro vývoj nové funkčnosti je user story v aplikaci uuBusinessRequestMan, doplněná o business model v aplikaci uuAppBusinessModelKit, v případě potřeby a možnosti technického popisu rozpracování konkrétních části technického modelu uuApp v aplikaci uuAppModelKit.

Díky detailně definovanému zadání požadavků lze lépe řídit očekávání zainteresovaných osob. Detailním zadáním vzroste efektivita schůzek s vývojovým týmem, klesnou náklady na vývoj a zpřesní se plánování vydání jednotlivých funkcí na produkci.

#### 4.6.2 Akceptační kritéria zadání

Zadání musí odpovídat akceptačním kritériím a musí být zřejmé všechny elementární činnosti a hrubý odhad, kolik bude kompletní implementace stát od designu (návrhu) přes implementaci až po testování.

##### **Akceptační kritéria user story:**

- formální správnost:
  - každá user story musí mít vlastní stránku v knize Business Requests, která obsahuje aplikaci uuBusinessRequestMan,
  - user story musí odpovídat struktuře šablon aplikace uuBusinessRequestMan:
    - mít řádně vyplněn název – jednoznačný a srozumitelný,
    - obsahovat srozumitelný popis,
    - mít jasně definovaný kontext a actors dané user story,
    - mít jasně definované kroky, které uživatel v aplikaci provádí ideálně formou algoritmu.
- kontext a shoda s vizí:
  - user story obsahuje kontext prostředí, ve kterém user story probíhá,
  - kontext konkrétní user story odpovídá kontextu user stories stejného produktu,
  - v kontextu user story jsou uvedeny typové osoby, včetně jejich popisu a předpokládaného chování,
  - user story odpovídá aktuálním vizím:
    - produktu,
    - family,
    - Unicorn Solution a.s.
- porozumění a plnění cílů:
  - user story musí mít řádně definovaný cíl a přidanou hodnotu,
  - user story je srozumitelná i pro čtenáře bez znalosti daného prostředí,

- pokud jsou nutné specifické znalosti, musí být definovány v kontextu user story,
- v případě, že user story obsahuje technické informace, např. co systém provede na pozadí, musí být srozumitelná i bez jejich znalostí,
- obrázky a diagramy musí být řádně popsány:
  - pojmenované objekty,
  - pochopitelné vazby, vztahy mezi objekty.

### **Akceptační kritéria business model:**

- formální:
  - je vytvořen do knihy business model s využitím aplikace uuAppBusinessModelKit,
  - odpovídá šabloně a obsahuje:
    - jasný název,
    - popis,
    - vazby na požadavek, např. product backlog, požadavek z primární nebo sekundární podpory, aby byl jasný kontext odkud požadavek přišel a co je jeho cílem,
    - vazbu na user story.
- mapuje všechny dopady na jednotlivé části business model:
  - business discipline,
  - actors,
  - products,
  - business use cases,
  - processes.
- věcná:
  - musí být jasný dopad na všechny produkty v uuAppBusinessModel a jejich atributy:
    - všechny změny musí být řádně označeny, aby bylo možné tyto změny zpracovat do aplikačního modelu a dále zaintegrované do aplikace,
    - pokud požadavek ovlivní produkty, musí být definováno, jak je ovlivní a jaké to bude mít dopady,

- pokud se v rámci požadavku vytvoří nový produkt s vazbou na již existující produkt, musí se zrevidovat dopady na všechny BUCs manipulující s tímto existujícím produktem,
  - pokud přibude nový actor, je potřeba zrevidovat, zda není potřeba jej přidat k již existujícím BUC,
  - pokud vznikne nový proces, je potřeba vyznačit jeho začlenění mezi stávající procesy.
- dále musí být definováno, do jakého z procesů požadavek zapadá, jaké jsou jeho dopady na proces, tzn. jak jej ovlivní,
  - pokud vzniknou nové business use cases, musí být definovány včetně jejich dopadů,
  - případné dopady požadavku do stávajících business use cases musí být definovány a business use cases musí být upraveny,
  - nové pojmy musí být nedefinovány do slovníčku pojmů, pokud byl pozměněn význam stávajícího pojmu, musí být aktualizován ve slovníčku.

#### **Akceptační kritéria návrhu změn v application model (app model) – nepovinná část:**

- Je nutný pouze u rozsáhlých funkcí, například se závislostí na jiných aplikacích, technologicky složitějších řešeních. Aplikace uuAppModelKit zahrnuje objektového schéma jenž mapuje vazby mezi jednotlivými uuSchemas, strukturu dílčí části dané aplikace (uuSubApp), funkce backendu, komponenty a knihovny frontendu, knihovnu skriptů a serverovou knihovnu.
- formální:
  - je vytvořen do knihy Application Model využitím aplikace uuAppModelKit,
  - odpovídá šabloně a obsahuje:
    - jasný název,
    - popis,
    - vazby na požadavek (např. product backlog, požadavek z primární nebo sekundární podpory, backlog v uuSprintMan), aby byl jasný kontext, odkud požadavek přišel a co je jeho cílem,
    - vazbu na business model, případně user story,
    - sekce z odpovídající šablony, které budou dopracovány během sprintu členy týmu SWF (uuApp designéry ve spolupráci s vývojáři).

- Obsahuje hrubý návrh dopadů do všech částí app model:
  - uuSchema – definuje, jaké atributy uuObject má, uuObject představuje specifický elementární záznam,
  - uuSubApp – je logická aplikační část, která se dále nedělí a obsahuje funkce (use cases) a trvalá data (strukturované uuObjects nebo binární uuBinary). Každá uuSubApp je tedy relativně samostatnou jednotkou jak z hlediska funkčnosti, tak z hlediska trvalých dat,
  - uuCommand (uuCmd) – je centralizovaná aplikační logika začleněná do use cases, které běží na aplikačních serverech v cloudu. Pokud tyto use cases patří do rozhraní aplikace (application interface), nazývají se uuCommands (příkazy),
  - uuComponents – jsou prvky na stránce nebo v aplikaci, které mají svou vlastní funkčnost. Komponenta (uuComponent) může být použita různými způsoby s použitím různých nastavení, ale základní funkce musí zůstat stejné. Komponenty mohou být vizuální mít určité funkce od zobrazení textu po ověření formuláře, nebo nevizuální, které zapouzdrují konkrétní chování či funkčnosti, např. práce s daty. Typy uuComponents jsou uuAppServer component a uu5 component,
  - a další.
- věcná:
  - zmapování dopadů do objektového schéma,
  - aktualizace stávajících uuCmd a uuComponent podle předpokládaných dopadů a změn,
  - pokud vzniknou nové uuCmd, musí být definovány dopady do:
    - existujících uuCmd,
    - uuAppServer component,
    - uu5 component,
    - uuScripts.
  - Product team primárně řekne, kam dopady změn a novinek zaznamenat, tým SWF změny zdokumentuje, dopady se případně mohou rozšiřovat. Z návrhu změn od product team je zřejmý hrubý dopad do application model (app model), mírné změny mohou nastat.



### 4.6.3 Účastníci procesu

Vlastníkem procesu je product manager produktu, který zodpovídá za řízení rozvoje produktu a práci produktového týmu, konkrétně spravuje požadavky od uživatelů, zainteresovaných osob a dalších, rozhoduje o prioritách jednotlivých požadavků, deleguje úkoly v rámci produktového týmu, kontroluje výstupy jednotlivých úkolů a je za ně plně odpovědný. Detailní definování procesu návrhu procesu zadání práce standardizuje zadání jednotlivých úkolů pro členy produktového týmu. Definuje kvalitativní i kvantitativní očekávání o výstupech jednotlivých úkolů. Díky akceptačním kritériím bude zjednodušen proces validace zadání pro vývojový tým, kontrola zadání bude efektivnější a díky detailním akceptačním kritériím bude možné tento úkol delegovat.

Dalšími účastníky procesu jsou vybraní členové produktového týmu, konkrétně zejména business architect, technology architect, celý proces je pod dohledem product supervisor.

**Business architect** vytváří user stories, business požadavky na uuApp, řídí priority cílové skupiny zákazníků a společně s TA je zodpovědný za napojení na uuApps.

**Technology architect** řeší složitější technologické otázky, které je třeba zodpovědět před předáním zadání do dodavatelského týmu SWF. Zodpovídá za architekturu produktu a vhodné použití The Architecture (pojem vysvětlen v kapitole Unicorn Approach).

**Product supervisor** je zodpovědný za řízení očekávání členů board a řízení produktu v souladu se strategií Unicorn Solutions a.s.

### 4.6.4 Vstupy procesu

Vstupy procesu jsou plánované funkčnosti aplikace, které mohou pocházet z požadavků od uživatelů, členů board a dalších zainteresovaných osob nebo firem. Většinou jsou přijaty písemně formou požadavku v aplikacích pro primární podporu (uuFirstLineSupport) a sekundární podporu (uuSecondLineSupport), následně je zpracován v procesu integrace (Obrázek 26 Pět procesů souvisejících s informačním systémem [9]).

### 4.6.5 Výstupy procesu

Výstupem by mělo být jednoznačně ucelené popsání problému (požadavku), který by měla funkčnost vyřešit, formou user story a zmapování jejích dopadů do business model. Pokud

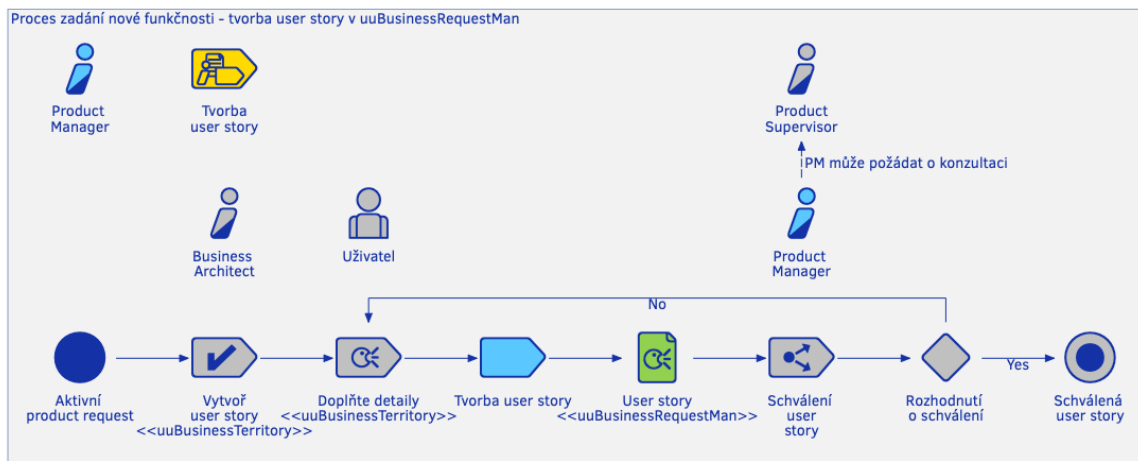
se jedná o složité technologické řešení, mělo by zadání obsahovat i základní návrh technického řešení problematiky.

#### 4.6.6 Proces tvorba user story

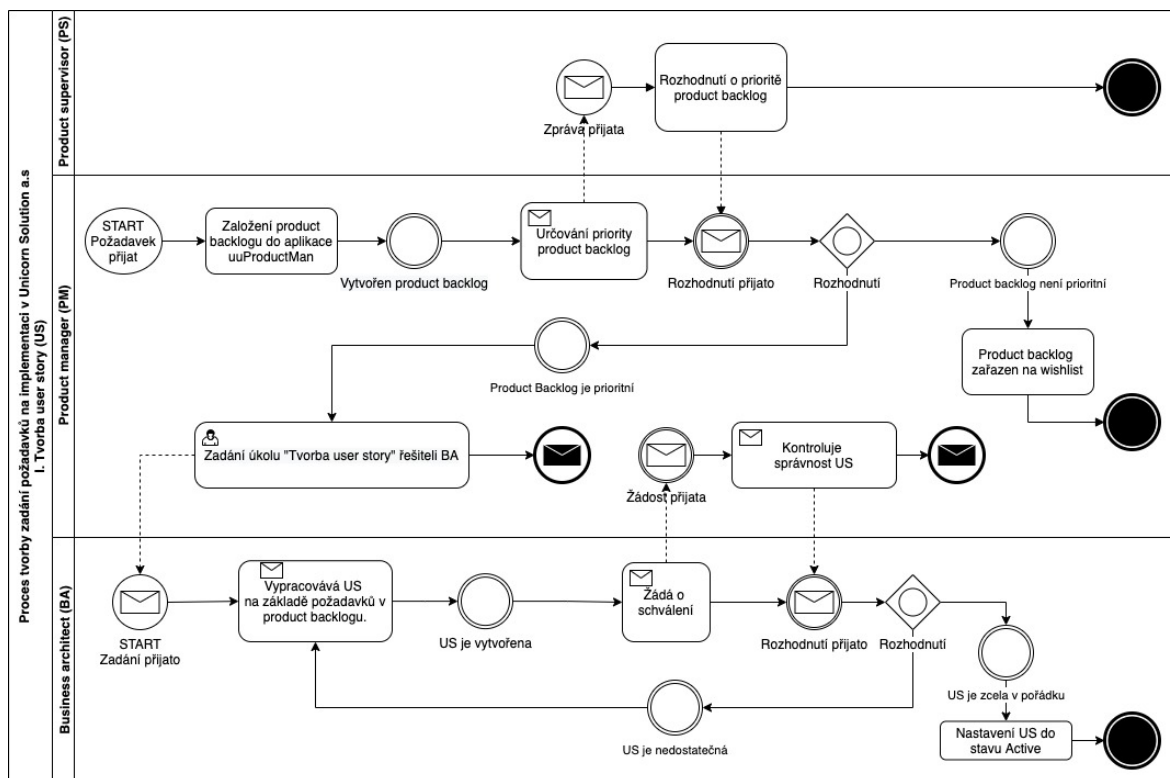
User story popisuje chování konkrétního uživatele v aplikaci z jeho pohledu. Účelem obecně je popis požadavku a očekávání uživatele od produktu, v tomto případě aplikace. Může obsahovat popis uživatele, kontext, z jakého prostředí uživatel daný úkon provádí, z jakého důvodu tuto funkčnost požaduje, jaké kroky mohou vést k požadovanému cíli a jakou přidanou hodnotu mu implementace jím požadované funkčnosti přinese. Případně může user story popisovat entitu z produktu, například může popisovat jednotlivé actory, jejich charakteristiku, vizi, kontext, jaké úkony mohou provádět a jaké informace jsou nezbytné pro práci s nimi. Jedním z cílů user story je popsat požadavek, aby měl vývojářský tým jasné pokyny pro specifické cíle, požadavky a porozuměl kontextu, proč je potřeba právě tuto funkčnost implementovat a jaký dopad to bude na uživatele mít.

Z pohledu metodiky uuP Business Request Management je user story klíčovým druhem pro popis požadavků. User story bývá zpracován zainteresovanými osobami (stakeholders) nebo kvalifikovanými uživateli v souvislosti s konkrétní business discipline. Je to jejich přání nebo nápad, jak vyřešit jejich problémy. User stories jsou napsány neformálním přirozeným jazykem. V souladu s metodikou uuP je součástí této práce vždy návrh procesu za použití anotace uuBml doplněn o BPMN diagram, který doplňuje uuBml diagram o bližší podrobnosti a dokresluje proces jako celek.

Obrázek 29 uuBml schéma procesu tvorby user story [Vlastní zpracování]



Obrázek 30 BPMN diagram tvorby user story [Vlastní zpracování]



Tato práce se zaměřuje na případ, kdy ze sekundární podpory byl přijat požadavek na novou funkčnost. Vychází se z předpokladu, že požadavek byl zadán kvalifikovaným uživatelem. U každého požadavku je znám zadavatel, anonymní požadavky aplikace sekundární podpory (uuSecondlineSupport) nepodporuje. Lze tedy zadávajícího uživatele kontaktovat v případě potřeby a zadání doplnit.

Proces tvorby user story začne momentem, kdy operátor sekundární podpory vyhodnotí, že se jedná o novou funkčnost a pošle požadavek do aplikace pro správu produktu uuProductMan. Aplikace uuProductMan spravuje produktový tým a zodpovědnou osobou je produktový manažer. Produktový manažer rozhodne, zda se jedná o validní požadavek, který bude dále zpracován, duplicitní požadavek a tak podobně.

Pokud produktový manažer rozhodne, že se jedná o validní požadavek, zahájí proces tvorby user story (Obrázek 30 BPMN diagram tvorby user story). Založí požadavek jako product backlog v aplikaci uuProductMan. Požadavek, který vytvořil člen board a v popisu požadavku označil jako prioritní, může být zařazen jako prioritní. Podle data očekávaného

dodání je určeno pořadí backlog pro realizaci. U požadavku, který byl vytvořen běžným uživatelem je vhodné konzultovat jeho prioritu s product supervisor. Product supervisor učiní rozhodnutí o prioritě požadavku, pokud požadavek není prioritní, je vložen do wishlist, kde čeká na tvorbu některé z dalších etap. Tvorba user story je zahájena, když je požadavek (product backlog) od product supervisor označen jako prioritní. Product manager založí nad product backlog aktivitu, na roli příslušného business architect.

Business architect započne vypracovávat user story podle standardů uuApp Patterns, založí stránku v knize business request, založená stránka je ve stavu under construction. V případě otázek je v kontaktu se zadavatelem požadavku, aby zjistil očekávání a kompletní zadání.

User story nemají přesně stanovenou strukturu, aby bylo dosaženo souladu napříč user stories. V rámci uuApp Patterns a aplikace uuBusinessRequestMan jsou definovány šablony, které jsou návodné při vytváření user stories, jejich struktura nemusí být striktně dodržována. Podle metodiky uuP mezi základní typy user stories patří user story popisující chování uživatele v aplikaci, user story popisující nějaký objekt nebo actors. Typy user stories, jejich šablony a návody jsou stále rozšiřovány podle potřeb jednotlivých uuApps.

User story, která se zabývá popisem chování uživatele v aplikaci, by měla odpovědět na otázky, kdo je typický uživatel, jaký je jeho cíl a z jakého důvodu chce tohoto cíle dosáhnout, jaké kroky by měl podniknout, aby tohoto cíle dosáhl. [52]

User story, která popisuje objekt nebo actor, by měla obsahovat popis objektu/actor, jaké může mít stavy, k čemu tyto stavy slouží, jaké interakce může mít s jinými objekty/actor a podobně.

V Unicorn Solution a.s., k vytváření user stories slouží aplikace uuBusinessRequestMan, která disponuje šablonami pro jednotlivé typy požadavků. Základní šablona pro typ požadavku user story je rozdělena do sekcí základní informace, seznam aktivit, z jakého důvodu potřebujeme tuto user story, kontext, cíle, popis vstupní situace, diagram, scénář, vstupy a reference.

- sekce základní informace obsahuje:
  - o název user story,
  - o krátký popis user story, například: „Uživatel Radek provede nějakou akci, aby dosáhl nějakého výsledku, konkrétně Radek potřebuje filtrovat své úkoly podle data, aby našel konkrétní skupinu úkolů, které mají určité jím zvolené datum splnění“,

- stav user story, který definuje, v jaké fázi se user story nachází. Zde je teprve určená k napsání, je rozpracována, čeká na schválení a podobně,
- odhad náročnosti realizace user story,
- prioritu user story,
- zodpovědnou osobu,
- umístění v systému.
- sekce seznam aktivit obsahuje aktivity související s danou user story,
- sekce z jakého důvodu potřebujeme tuto user story obsahuje:
  - popis z jakého důvodu potřebujeme tuto user story,
  - popis problému, který má user story vyřešit.
- sekce kontext obsahuje:
  - názorný popis uživatele, který v dané user story vystupuje,
  - popis prostředí v jakém je daná user story popisována, může obsahovat konkrétní osoby, produkty, a podobně. Cílem je přiblížit prostředí user story,
  - popis jednotlivých uživatelů, kteří mohou danou funkčnost využít a jakou přidanou hodnotu přinese právě tato user story jim. Je vhodné používat uživatele, který byl již v nějaké user story použit, aby vznikl ucelený pohled, jak může jeden konkrétní uživatel v aplikaci postupovat.
- sekce cíle obsahuje:
  - popis cíle, kterého chce daný uživatel dosáhnout a z jakého důvodu jej chce dosáhnout, co mu jeho dosažení přinese.
- sekce popis vstupní situace obsahuje:
  - popis počáteční situace v detailech, pro názornost je vhodné použít příklady ze reálného světa.
- sekce diagram,
  - user story může být pro názornost doplněna o diagram popisující funkčnosti, prostředí a podobně.
- sekce scénář obsahuje:
  - popis kroků, které uživatel musí provést, aby dosáhl stanoveného cíle.
- sekce vstupy obsahuje:
  - seznam artefaktů, které jsou vstupy pro tuto user story
  - například originální požadavek ze sekundární podpory.

- sekce reference obsahuje:
  - o seznam artefaktů z jiných aplikací, které se vztahují k této user story, například odkaz na sprint backlog.

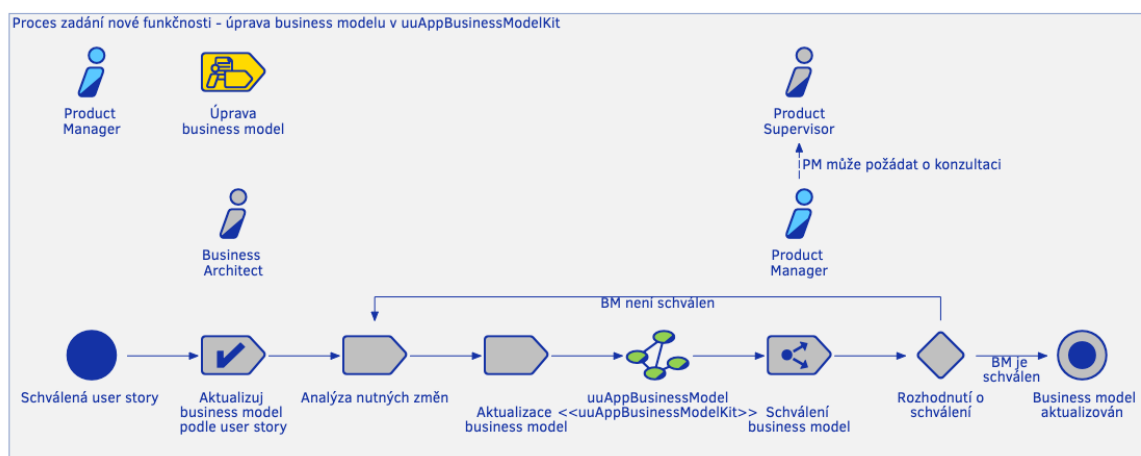
Pokud má business architect při vytváření user story jakékoliv pochybnosti či otázky, může se kdykoliv obrátit s prosbou o radu na technology architect či na product manager. Radu od technology architect může využít například, pokud si není jistý, zda jím navrhovaný scénář je v souladu s uuApp Framework.

Poté, co je user story ze strany business architekt hotová, zkontroluje, zda splnil všechna akceptační kritéria a nastaví stav dokončeno na aktivitě úkolu, jež dostal od product manager, dále pošle aktivitu typu rozhodnutí na product manager. Product manager rozhodne o správnosti user story, zda odpovídá vizím produktu a očekáváním zainteresovaných osob, zda je v souladu se všemi metodikami, akceptačními kritérii. Product manager může požádat o konzultaci product supervisor. Po odsouhlasení user story je stránka nastavena business architect do stavu active. V tomto momentu končí proces tvorby user story a navazuje na něj proces úprava business model. Je důležité zmínit, že user story se může v průběhu životního cyklu modifikovat, nelze říci, že jednou napsaná user story je navždy hotová. Vždy je nutné nepodcenit konkurenci, produkt stále rozvíjet, s čímž souvisí i úpravy v user stories.

#### **4.6.7 Proces úprava business model**

Během přípravné fáze životního cyklu uuApp byly popsány jednotlivé business model (jednotlivé části business model jsou popsány v kapitole Model životního cyklu uuApp). U business model je nezbytné, aby během životního cyklu produktu byl stále aktuální. Z tohoto důvodu je nutné zahrnout do procesu tvorby zadání pro vývojový tým i nutnost úprav v business model. V této kapitole je popsán proces úpravy business model v uuAppBusinessModelKit.

Obrázek 31 uuBml schéma procesu úpravy business model [Vlastní zpracování]

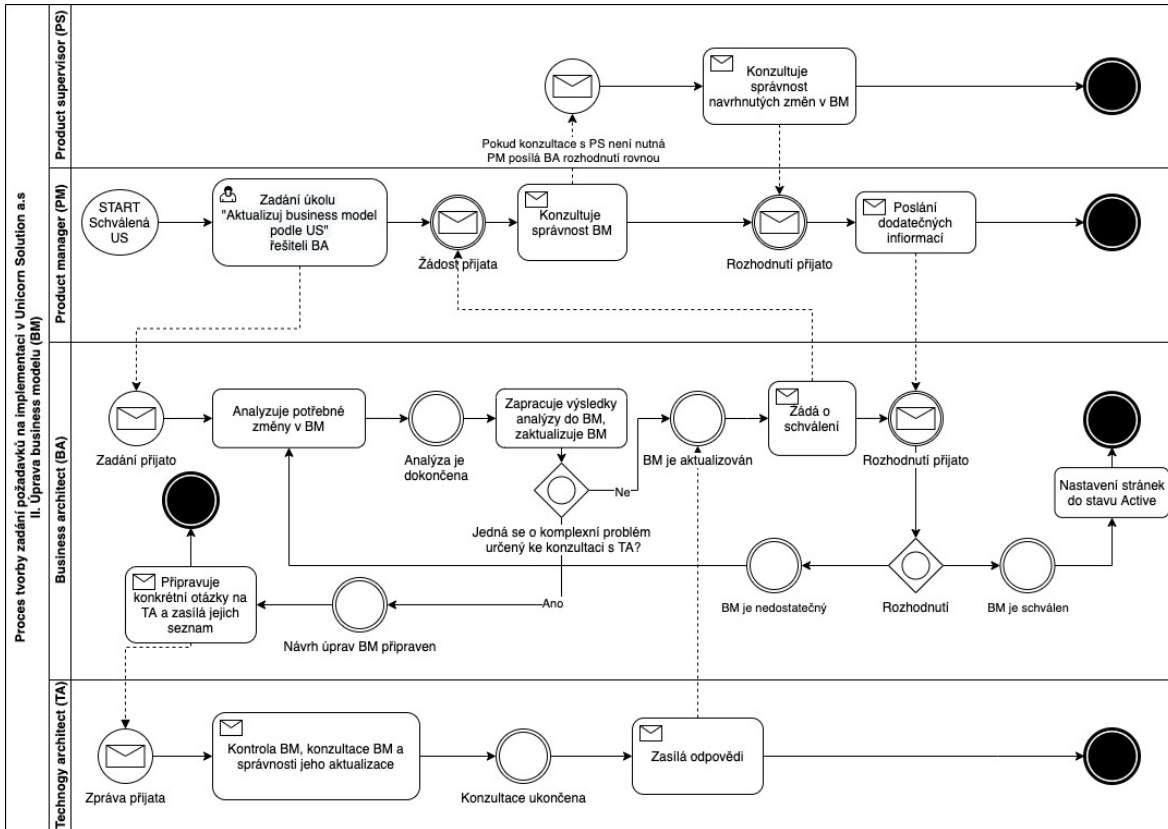


Proces úpravy business model zahrnuje interakci zejména mezi business architect a product manager, v odůvodněných případech může být účastníkem procesu i technology architect, případně product supervisor a další. Na počátku procesu stojí schválená user story a cílem procesu je zmapovat její dopad do business model.

Po nastavení stánky user story do aktivního stavu začne business architect zpracovávat analýzu dopadů na business model na základě úkolu od product manager. Výsledky analýzy zaneše do business model, aktualizuje stávající části, případně business model rozšíří. Kdykoliv během analýzy a aktualizace může konzultovat návrhy a řešení s technology architect, zejména pokud se jedná o rozsáhlé a komplikované změny, které budou mít vliv i na jiné uuApps. V případě, že se jedná o rozsáhlou změnu, je nutné získat schválení od product supervisor a zajistit, že se změna dostane ke všem, které ovlivní například formu open session. Open session jsou setkání, na nichž jsou reprezentovány novinky, změny a zajímavá řešení související s The Architecture. Poté, co je aktualizace business model kompletní, pošle business architect aktivitu typu rozhodnutí na product manager, který rozhodne o správnosti business model, zda je v souladu se všemi metodikami, akceptačními kritérii a The Architecture. Product manager může požádat o konzultaci s product supervisor, tato konzultace je zejména nutná u větších změn, případně v oblastech, kde není jednoznačně určen správný postup. Senior business architect může na základě dohody s product manager rozhodnout o správnosti návrhu osobně a být více samostatný.

V případě, že produktový tým má nedostatečné kapacity, může být vstupním zadáním pro vývojový tým výstup analýzy od business architect se slovním popsáním konkrétních dopadů a požadavků k úpravám, které zmapuje senior uuApp designer.

Obrázek 32 BPMN diagram úpravy business model [Vlastní zpracování]



#### 4.6.8 Proces návrh úprav app model

Jak bylo zmíněno v kapitole Životní cyklus softwaru, každý dobrý systém by měl splňovat 1+10 vlastností dobrého systému. Desátou vlastností je „zdokumentovaný, každý dobrý systém má kvalitní technickou a uživatelskou dokumentaci.“ [9]

Unicorn Application Framework doporučuje mít detailně rozmyšlené řešení požadavku dříve, než bude zahájen samotný vývoj. Výhodou tohoto přístupu je možnost zjištění, zda danou situaci již neřešil jiný team uuApp dříve a mohlo by se využít jejich řešení. Při detailním definování problému a návrhu řešení, lze použít získaného firemního know-how.

Tento přístup podporuje Software Factory množstvím uuApp designérů v jejich týmech. V ideálních podmínkách by platilo, že práce developer následuje až po dokončení designu (návrhu řešení) od uuApp designera, tedy uuApp developer implementuje návrh řešení



od uuApp designer, což v praxi není vždy možné dodržet. Mohou nastat situace, kdy probíhá práce uuApp developera a uuApp designer souběžně. Platí, že developer by měl plnit pouze rozmyšlené zadání (např. jsou určeny základní kroky a cíl uuCmd a v průběhu vývoje společně konzultují řešení).

Návrh řešení musí obsahovat i zmapování, jaké dopady bude mít navrhované řešení do vlastní uuApp, případně dopady do jiných uuApps využívající upravovanou uuApp.

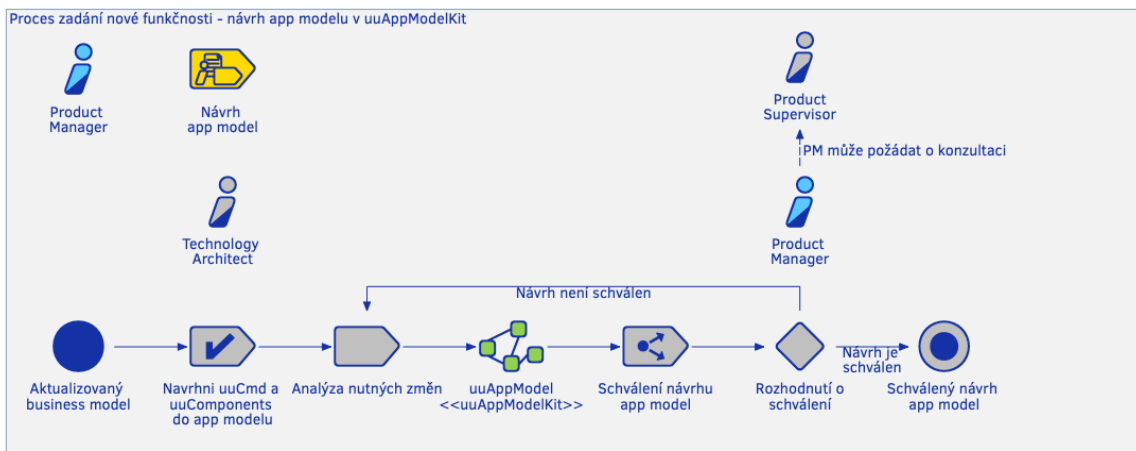
Aplikace uuAppModelKit je určena pro navrhování aplikačních modelů jako součást uuApp Framework a uuP Methodology. Výhodou využití uuAppModelKit pro tvorbu technické dokumentace je fakt, že aplikace udržuje vazby mezi jednotlivými částmi implementace a lze pak snadno zjistit dopad změn. Dále je zajištění stejné kvality napříč dokumentacemi všech uuApps. Tato aplikace je podpořena vzdělávacími kurzy, které musí absolvovat každý uuApp designer a developer před nástupem na danou pozici. Zároveň je zajištěno kontinuální informování o rozšířeních této aplikace a o jejích nových funkcích.

Jak bylo uvedeno v kapitole Model životního cyklu uuApp, tvorba app model probíhá na straně vývojového týmu v průběhu sprintu, kdy uuApp designer připraví návrh řešení daného požadavku, který později uuApp developer implementuje.

Podle kapitoly Akceptační kritéria zadání je výstupem tohoto dílčího procesu návrh možných úprav v app model, které povedou k implementaci požadované funkčnosti. Návrh app model ze strany product team je nutný pouze u technologicky složitějších řešení, například změny, jež ovlivní i aplikace napojené či využívající upravovanou uuApp.

Pokud v product team nejsou dostatečné kapacity technology architect pro detailní zpracování dopadů do app model, může být zadáním pro vývojový tým user story. Řešení následně bude dodáno ve vícero krocích – v prvním sprintu uuApp designer ve spolupráci s senior uuApp designer návrh řešení a zmapuje dopady. Během druhého sprintu jej zkontroluje technology architect a schválí product supervisor. Po schválení může být zařazena implementace schváleného řešení do nejbližšího sprintu. Nevýhodou tohoto postupu je delší doba trvání a je nutné řídit očekávání zainteresovaných osob a seznámit je každý týden s aktuálním stavem.

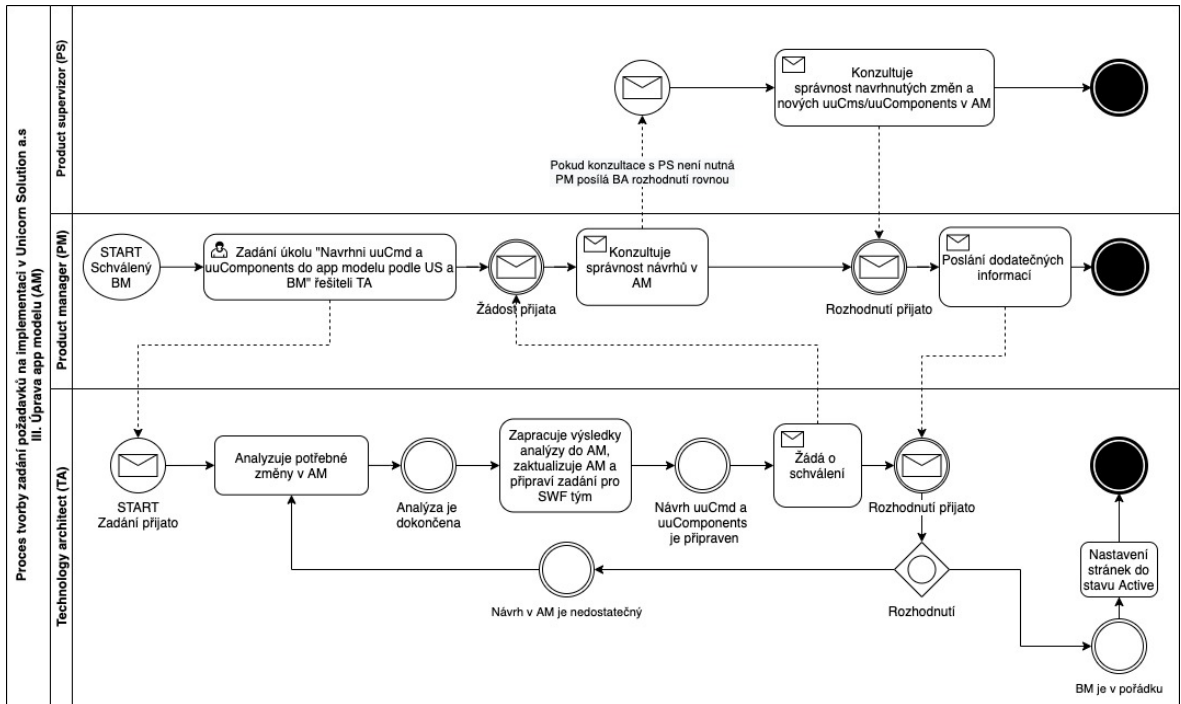
Obrázek 33 uuBml schéma procesu návrhu app model [Vlastní zpracování]



Po schválení výstupu z procesu úprava business model je zahájen proces návrh úprav app model. Výstupem tohoto dílčího procesu je návrh změn v app model, které budou dodány v jednotlivých sprintech.

Osobou odpovědnou za tento proces je product manager, který deleguje analýzu změn a návrhu změn app model na technology architect. Technology architect zahájí práci na úkolu a v případě dotazů k business model se obrátí na business architect pro ujasnění zadání a případné doplnění jím navrhovaných změn do business model. Poté, co technology architect vypracuje analýzu potřebných změn v app model a provede návrh řešení, pak je posláno řešení na schválení product manager. Senior technology architect může na základně dohody s product manager rozhodnout o správnosti návrhu osobně a být více samostatný. Product manager může požádat o konzultaci product supervisor. Pokud se jedná o návrh změn, které budou mít dopady nejen do vlastní uuApp, musí návrh vždy schválit product supervisor.

Obrázek 34 BPMN diagram návrhu úprav app model [Vlastní zpracování]



## 4.7 Výsledky a diskuse

### 4.7.1 Komunikace

Během práce byla zjištěna část příčin neefektivní komunikace mezi týmem dodavatele a zadavatele.

Jednou z příčin byla absence komunikačního schématu a nejednoznačné stanovení odpovědností a kompetencí, což vedlo k prodlužování schůzek a nutnosti hlubšího vysvětlení komunikačních šumů, které vznikly díky netransparentnosti komunikace.

Pro spolupráci mezi týmem zadavatele a dodavatele by byla přínosem účast zástupce týmu zadavatele na denních schůzkách vývojového týmu. Díky tomu by měl product team každý den aktuální informace, jak pokračují práce. Tento návrh vychází z metodiky scrum, která je popsána v kapitole Agilní přístup, kdy vlastník problému, v tomto případě zadavatel, může být účastníkem denních schůzek vývojového týmu. Získané informace by mohl předat na denních schůzkách product team všem členům týmu. Případné otázky a problémy by tedy mohly být řešeny na denní bázi. Detailní návrh komunikace mezi product team, development team a test team během sprintu je v komunikační matici (Příloha 8 Návrh komunikační matice při vývoji produktu v Unicorn Solution a.s. [Vlastní zpracování]).

Další příčinou neefektivní komunikace mezi týmy, jak bylo uvedeno v kapitole Komunikace mezi dodavatelskými týmy a zadavatelským týmem Unicorn Solutions, bylo obsazení junior uuApp designer do pozice team leader. Díky nedostatečným zkušenostem nezískat důvěru týmu a byl členy týmu z přímé komunikace se zákazníkem vyčleňován.

Navrhované řešení je vymezení kompetenčního modelu role team leader a postupné zaučování. Také zajištění maximální podpory v rámci sktruktury SWF, pravidelné schůzky se senior team leader a potřebné kurzy pro rozvoj znalostí a osobnosti. Je nutná komunikace schopností team leader uvnitř development team i směrem k zákazníkovi. Jedná se o nadějného junior uuApp designer, v jehož schopnosti struktura společnosti věří a poskytla mu dostatečnou podporu zkušeného kolegy. Díky těmto opatřením a vhodné komunikaci by byla správně nastavená očekávání všech zúčastněných a osobě v roli team leader by byl poskytnut čas na zaučení.

Příčinou nejasné komunikace ze strany zadavatele bylo nejednoznačné zadání a nejednotný postoj členů product team. Nejednotné vystupování členů product team vedlo ke zmatení týmu dodatele, který následně díky absenci jednoznačného určení odpovědností nevěděl, který názor členů product teamu implementovat, tedy koho mají poslechnout.

**Řešením** by mohlo být zavedení denní schůzky product team s cílem průběžně informovat celý tým o otevřených otázkách a průběžně plánovat jejich řešení, např. naplánování schůzky na dané téma. Nejdříve by se otevřená otázka vyřešila uvnitř product team, což by vedlo k jednotnému vystupování směrem k dodavateli. K jednoznačnému určení odpovědnosti může být využita matice RASCI, díky návrhu matice RASCI pro proces tvorby zadání (Tabulka 2 Matice RASCI pro návrh procesu tvorby zadání [Vlastní zpracování]) je proces transparentní. Dodavatel jednoznačně ví, kdo kterou část zadání má zpracovat a kdo může rozhodnout o dalších krocích.

Tabulka 2 Matice RASCI pro návrh procesu tvorby zadání [Vlastní zpracování]

MATICE RASCI	Product manager	Product team coordinator	Business architect	Technology architect	Product evangelist
Tvorba user story	AC	I	R	C	S
Úpravy business model	AC	I	R	I	S
Návrh úprav app model	A	I	C	R	S

#### 4.7.2 Návrh tvorby zadání

Navrhovaný proces tvorby zadání pro vývojový tým byl konzultován se zaměstnanci společnosti Unicorn Solution a.s., zejména Mgr. Lindou Borovskou, Radkem Dolejšem a Milošem Vodrážkou. Na základě jejich zkušeností z praxe nedisponuje produktový tým dostatečnými kapacitami pro dodržení navrhovaného postupu v plném rozsahu. Mohou nastat situace, kdy je potřeba vypracovat zadání na straně dodavatele od user story až po implementaci.

Zadání podle zpětné vazby z praxe může existovat ve vícero formách, které se liší zejména v míře detailu:

- stručné písemné zadání pro tvorbu user story doplněno o ústní komentář, kdy na straně dodavatele je potřeba vytvořit user story, která bude schválena zadavatelem podle akceptačních kritérií, které jsou uvedeny v kapitole Akceptační kritéria zadání. K této akceptaci může dojít na některém z demonstration meeting nebo při acceptance meeting. Po schválení může dodavatel zahájit tvorbu business model, jenž musí být opět zadavatelem přijat. Po schválení business model bude zahájena práce na app model a implementace,
- vstupem zadání pro dodavatele je detailní user story. Úkolem dodavatele je tedy začít tvorbou návrhu změn v business model. Návrh je následně schválen od product manager, podle akceptačních kritérií zadání, které jsou uvedeny v kapitole Akceptační kritéria zadání. Po schválení může dodavatel návrh změn zapracovat do business model a zahájit práce na app model a implementace,
- zadáním je user story a business model, podle kterého dodavatel implementuje požadovanou funkčnost a vypracuje technickou dokumentaci app model. Tento postup se shoduje s návrhem, jenž byl představen v této diplomové práci,
- zadáním v případě složitých řešení či řešení, jež ovlivní uuApps na upravovanou uuApp napojené, je návrh app model (uuApp Model).

Návrh, jenž je výstupem této diplomové práce, je použitelný ve všech výše zmíněných případech, neboť stanovuje akceptační kritéria a BPMN diagramy názorně vysvětlují vazby mezi jednotlivými kroky a odpovědnosti jednotlivých členů.

Při zadání formou stručného popisu user story s požadavkem na vypracování user story na straně zadavatele je potřeba počítat se zapojením zadavatele. Je vhodné, aby user story vypracovával senior uuApp designer, který již na produktu pracoval a v business discipline daného produktu se orientuje. Lze v tomto případě využít BPMN diagram tvorby user story (Obrázek 30 BPMN diagram tvorby user story [Vlastní zpracování]), kdy roli business architect zastane senior uuApp designer, stejný postup lze aplikovat na proces úpravy business model.

## 5 Závěr

Práce pojednává o vytvoření návrhu zadání pro vývojový tým. V průběhu tvorby diplomové práce bylo zjištěno, že je vhodné rozložit proces tvorby zadání do tří dílčích procesů. Pro každý proces byly stanoveny úkoly a osoby odpovědné za vykonání úkolů i osoba odpovědná za výstup procesů.

Po konfrontování návrhu procesu tvorby zadání v praxi byly zjištěny další možnosti vypracování zadání pro vývojový tým a byly vytvořeny návrhy, které odpovídají požadavkům praxe.

Následně byl navrhovaný proces tvorby zadání ověřen v praxi. Ověření v praxi proběhlo v rámci produktu uuMyTerritory, jenž je klíčovým produktem společnosti Unicorn Solutions a.s.

Aplikace uuMyTerritory poskytuje soukromý prostor pro všechny uživatele služby Plus4U. Na jednom místě mohou najít své úkoly, schůzky a aplikace. Uživatelé mohou do svého uuMyTerritory instalovat aplikace a spravovat přístupová práva k informacím uloženým v těchto aplikacích. Uživatelé mohou vytvářet kluby, kde mohou snadno spravovat přístup pro ostatní uživatele a sdílet s nimi informace, fotografie, události, úkoly atd.

Během praktické části bylo vypracováno zadání pro vývojový tým k implementaci nové funkčnosti, díky které budou uživatelé schopni lépe třídit své úkoly. Zadání se skládá z user story (Příloha 9 User story in uuBusinessRequestMan Aggregated view in uuMT dw by uuIdentity - Submitter/Solver [Vlastní zpracování]) a úprav v business model (Příloha 10 Business use-case in uuAppBusinessModelKit Tasklist aggregated view in uuMT dw by uuIdentity [Vlastní zpracování], Příloha 11 Business use-case in uuAppBusinessModelKit Display tasklist for selected uuIdentity [Vlastní zpracování], Příloha 12 Business use-case in uuAppBusinessModelKit Add selected uuIdentity user to the preferred/unpreferred users (toggle) [Vlastní zpracování]) a backlog z uuSprintMan (Příloha 13 Backlog request in uuSprintMan Aggregated view in uuMT dw by uuIdentity [Vlastní zpracování]). Příslušná část uuAppModel a zdrojový kód budou dodány v průběhu sprintu od dodavatele (tzn. od vývojového týmu).

Přínosem práce je jednoznačné definování úkolů vedoucích k tvorbě zadání pro vývojový tým, stanovení odpovědnosti za jednotlivé úkoly s využitím matice RASCI a stanovení akceptačních kritérií zadání, dále návrh komunikační matice, která díky jednoznačnému definování komunikace zjednoduší komunikaci mezi zadavatelem a dodavatelem.

Během praktické části byl nastíněn kontext prostředí společnosti Unicorn Solutions a.s., dělba práce uvnitř holdingu Unicorn a.s., popis metodik, rámců a přístupu holdingu Unicorn a.s. k vývoji softwaru, včetně základních myšlenek a principů. Byl představen životní cyklus vývoje uuApp i životní cyklus uuApp, komunikace mezi týmy zadavatele a dodavatele. Pro názornost byly hojně užívány interní diagramy a obrázky, jež utváří celistvý pohled a dokreslují kontext holdingu a jeho myšlenek.

V teoretické části byly vysvětleny obecné principy, jež byly následně využity v praktické části. Byly popsány životní cykly, které mají vliv na vývoj softwarového produktu, jelikož výroba softwarového produktu je týmovou prací, byl popsán životní cyklus týmu i životní cyklus softwaru. Dále byl definován produkt a popsán jeho životní cyklus.

K vývoji softwaru existuje množství metodik, modelů a přístupů. V práci byly nastíněny základní přístupy a metodiky, mezi které patří model vodopád, spirála, prototypový model a v neposlední řadě agilní přístup. Agilní přístup k vývoji softwaru je přístupem využívaným i ve společnosti Unicorn Solutions a.s., jež má vlastní metodiku, která je kromě jiných metodik inspirována i metodikou scrum.

Díky tomu, že je vývoj softwaru týmovou záležitostí, kde je nutná spolupráce a jednoznačné definování odpovědnosti, bylo v teoretické části popsáno i řízení komunikace a odpovědnosti včetně obecného definování těchto pojmů.

Zadání by mělo mít akceptační kritéria, aby bylo jednoduché lehce určit, zda byl úkol splněn a zadání může být přijato k dalšímu zpracování vývojovým týmem. Proto v teoretické části byly vysvětleny i modely a metodiky akceptačních kritérií, například metoda OKR, Objectives and Key Results (česky cíle a klíčové výsledky), model "CRaB QueST", (Cost, Risk, Benefits, Quality, Scope and Time), česky náklady (rozpočet), rizika, přidaná hodnota, kvalita, rozsah (kvantita) a termín, a model KKTR (kvalita, kvantita, termín, rozpočet).



## Seznam použitých zdrojů

- [1] TUCKMAN, Bruce W. Developmental sequence in small groups. *Psychological Bulletin* [online]. 1965, **63**(6), 384–399. ISSN 0033-2909. Dostupné z: doi:10.1037/h0022100
- [2] HÁJEK, Martin. *Životní fáze týmu* [online]. 2012 [vid. 2019-10-16]. Dostupné z: <http://www.vedeme.cz/pro-vedeni/kapitoly-vedeni/66-tymy/153-faze-tymu.html>
- [3] KAŠÍK, Milan a Karel HAVLÍČEK. *Marketing při utváření podnikové strategie*. Praha: Vysoká škola finanční a správní, 2009. ISBN 978-807-4080-227.
- [4] BENNETT, Coleman & Co. Ltd. Definition of 'Product'. *The Economic Times* [online]. 2021 [vid. 2021-01-02]. Dostupné z: <https://economictimes.indiatimes.com/definition/product>
- [5] MANDERS, Basak, Henk J. DE VRIES a Knut BLIND. ISO 9001 and product innovation: A literature review and research framework. *Technovation* [online]. 2016. ISSN 01664972. Dostupné z: doi:10.1016/j.technovation.2015.11.004
- [6] JAKUBÍKOVÁ, Dagmar. *Strategický marketing : strategie a trendy*. B.m.: Grada, 2013. ISBN 9788024746708.
- [7] KOTLER, Philip a Gary. ARMSTRONG. *Marketing*. B.m.: Grada, 2004. ISBN 8024705133.
- [8] BAŤKOVÁ, Iva. *Životní cyklus informačního systému* [online]. B.m., 2013. Univerzita Pardubice. Dostupné z: <http://hdl.handle.net/10195/52233>
- [9] KOVAR, Vladimír, David KIMR, Petr CIOCHOŇ a Marek ŠŤASTNÝ. IT v podnicích a organizacích. *Úvod do UAF* [online]. 2020 [vid. 2020-11-07]. Dostupné z: [https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-86b4d5a4a030400784764ebdb972bbda/book/page?code=pohled\\_unicorn\\_na\\_ICT\\_00](https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-86b4d5a4a030400784764ebdb972bbda/book/page?code=pohled_unicorn_na_ICT_00)
- [10] ŠMÍD, V. *Životní cyklus informačního systému* [online]. 2008 [vid. 2021-01-02]. Dostupné z: <https://www.fi.muni.cz/~smid/mis-zivcyk.htm>
- [11] ROYCE, Winston W. MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS Dr. Winston W. Royce INTRODUCTION. *Ieee Wescon*. 1970.
- [12] HLAVA, Tomáš. *Vodopádový model* [online]. 2011 [vid. 2020-11-20]. Dostupné z: <http://testovanisoftware.cz/?s=vodopádový+model>
- [13] VONDRÁŠEK, Jiří. *Metody strukturované analýzy a návrhu informačních systémů* [online]. B.m., 2007. Vysoká škola ekonomická v Praze. Dostupné z: [https://vskp.vse.cz/\\_soubory/07427\\_vondr-ji.pdf](https://vskp.vse.cz/_soubory/07427_vondr-ji.pdf)
- [14] BOEHM, Barry W. A Spiral Model of Software Development and Enhancement. *Computer* [online]. 1988. ISSN 00189162. Dostupné z: doi:10.1109/2.59
- [15] BOEHM, Barry. *Spiral Development : Experience , Principles , and Refinements. Spiral Development Workshop February 9, 2000*. 2000.
- [16] BECK, Kent, Mike BEEDLE, Arie VAN BENNEKUM, Alistair COCKBURN, Ward CUNNINGHAM, Martin FOWLER, James GRENNING, Jim HIGHSMITH, Andrew HUNT, Ron JEFFRIES, Jon KERN, Brian MARICK, Robert C MARTIN, Steve MELLOR, Ken SCHWABER, Jeff SUTHERLAND a Dave THOMAS. *Agile Manifesto. Software Development*. 2001.
- [17] SCHWABER, Ken a Jeff SUTHERLAND. *The Scrum Guide: The Definitive The Rules of the Game. Scrum.Org and ScrumInc*. 2017. ISSN 00195847.

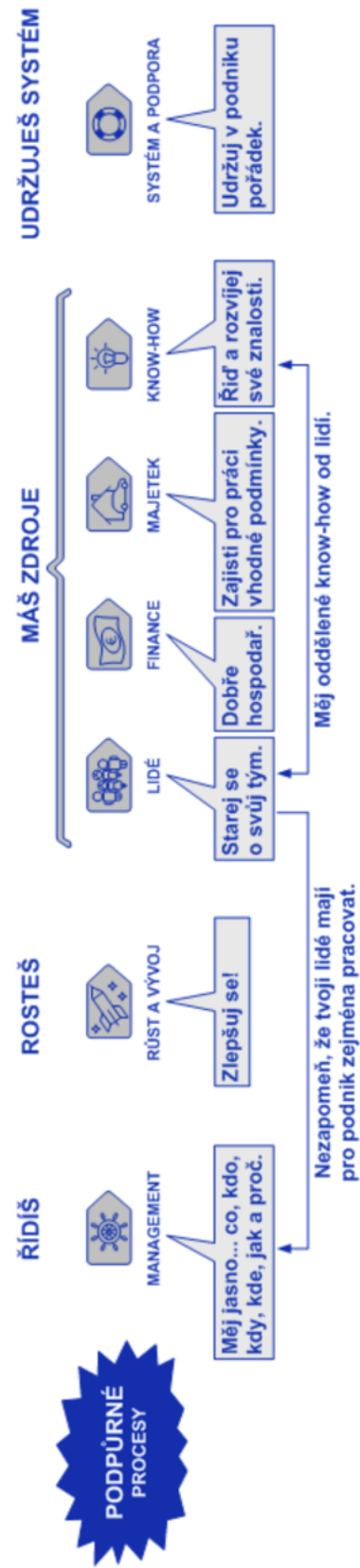
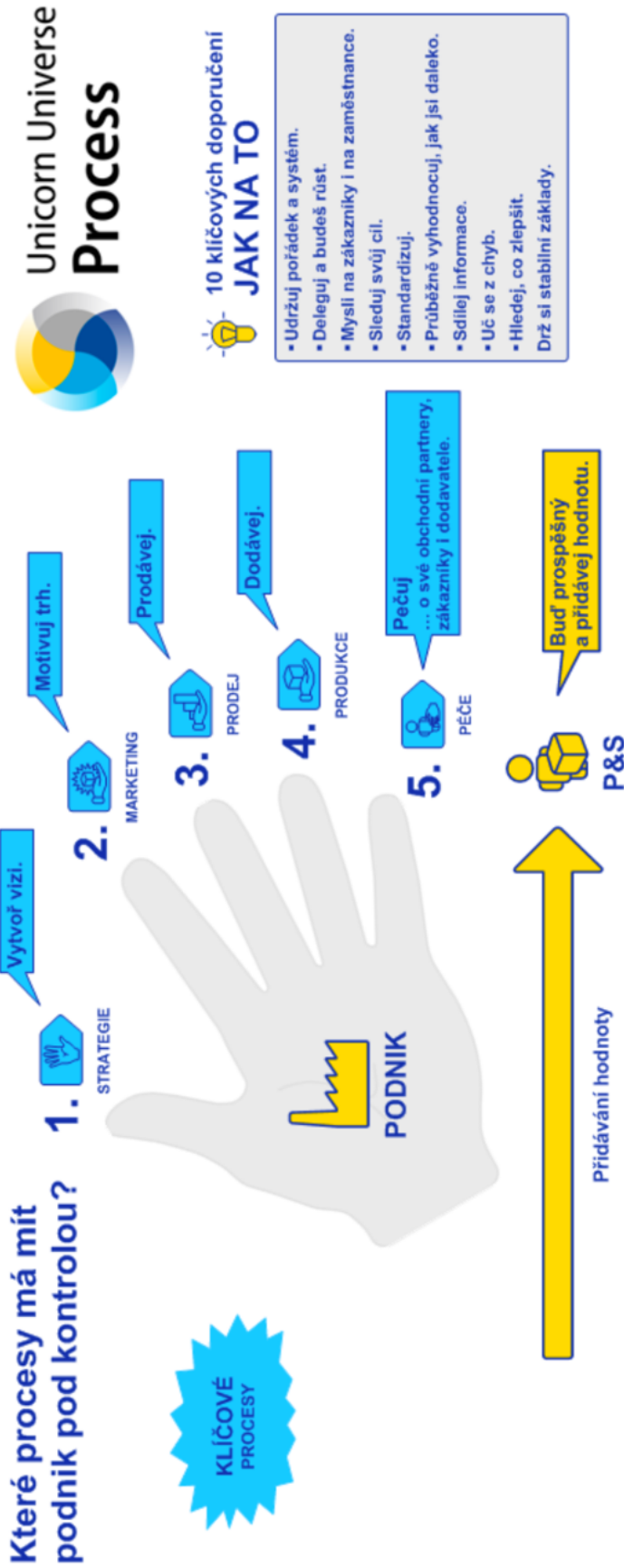
- [18] STŘÍŽOVÁ, Vlasta. *Manažerská komunikace*. 2. Praha: Nakladatelství Oeconomica, 2005. ISBN 80-245-0923-7.
- [19] GIBSON, James L., John M. IVANCEVICH, James H. DONNELLY JR. a Robert KONOPASKE. *Organizations Behavior, Structure, Processes* [online]. Fourteenth. New York: McGraw-Hill, 2017. ISBN 978-0-07-811266-9. Dostupné z: [http://dl.motamem.org/organizations\\_behavior\\_structure.pdf](http://dl.motamem.org/organizations_behavior_structure.pdf)
- [20] PROJECT MANAGEMENT INSTITUTE. *A guide to the project management body of knowledge / Project Management Institute* [online]. sixth. Newtown Square, PA: Project Management Institute, 2017. ISBN 978-1-62825-184-5. Dostupné z: [https://book.akij.net/eBooks/2018/March/5abcc35b666f7/a\\_guide\\_to\\_the\\_project\\_management\\_body\\_of\\_knowledge\\_6e.pdf](https://book.akij.net/eBooks/2018/March/5abcc35b666f7/a_guide_to_the_project_management_body_of_knowledge_6e.pdf)
- [21] NEŠPOR, Zdeněk R., J HOLL, Pavel BARAN, M LUTHER, W WEISSCHEDEL, W KELLER, G FREY, K LORENZ a K JASPERSEM. Odpovědnost. *Encyklopedie sociologického ústavu AV ČR* [online]. 2017 [vid. 2021-01-20]. Dostupné z: <https://encyklopedie.soc.cas.cz/w/Odpovednost>
- [22] WILMINGTON. Odpovědnost (Responsibility). *ManagementMania.com* [online]. 2018 [vid. 2021-01-29]. Dostupné z: <https://managementmania.com/cs/odpovednost>
- [23] WILMINGTON. Kompetenční model (Competence model). *ManagementMania.com* [online]. 2016 [vid. 2021-01-29]. Dostupné z: <https://managementmania.com/cs/kompetencni-model>
- [24] WILMINGTON. DACI Matice (DACI Responsibility Matrix). *ManagementMania.com* [online]. 2020 [vid. 2021-01-29]. Dostupné z: <https://managementmania.com/cs/daci-matice-daci-responsibility-matrix>
- [25] WILMINGTON. Matice odpovědnosti RACI (RACI Responsibility Matrix). *ManagementMania's Series of Management* [online]. 2016 [vid. 2021-01-29]. Dostupné z: <https://managementmania.com/cs/matice-odpovednosti-raci>
- [26] WILMINGTON. Matice odpovědnosti RASCI (RASCI Responsibility Matrix). *ManagementMania.com* [online]. 2020 [vid. 2021-01-29]. Dostupné z: <https://managementmania.com/cs/matice-odpovednosti-rasci>
- [27] WU, Chien Wei, W. L. PEARN a Samuel KOTZ. An overview of theory and practice on process capability indices for quality assurance. *International Journal of Production Economics* [online]. 2009. ISSN 09255273. Dostupné z: [doi:10.1016/j.ijpe.2008.11.008](https://doi.org/10.1016/j.ijpe.2008.11.008)
- [28] SIEGELAUB, Jay M. Six (yes six!) constraints. *PMI Global Congress 2007* [online]. 2007, 1–9. Dostupné z: <https://www.pmi.org/learning/library/six-constraints-enhanced-model-project-control-7294>
- [29] DOERR, John. *Measure what matters : how Google, Bono, and the Gates Foundation rock the world with OKRs*. 2018. ISBN 9780525536222 0525536221 9780525538349 0525538348.
- [30] KOVÁŘ, Vladimír a Jan JAROŠ. Poslání Unicorn O společnosti Unicorn. *Unicorn website* [online]. 2007 [vid. 2020-11-12]. Dostupné z: <https://unicorn.com/cs/company-profile>
- [31] KOVÁŘ, Vladimír. *Metodika Unicorn Enterprise System Powered Company*. B.m., 2011. Univerzita Hradec Králové.
- [32] KOVÁŘ, Vladimír, David KIMR, Marek ŠTASTNÝ, Milan HYNEK, Vladimír ml. KOVÁŘ a Lucie JANDOVÁ. Organizational Scheme of Unicorn Solutions. *Unicorn Solutions* 2020 [online]. 2020 [vid. 2020-11-07]. Dostupné

- z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-88c49812dec941d8b25e24652f5e5c66/book/page?code=93748380>
- [33] KOVÁŘ, Vladimír, Michal ZAJAC, Miloš DVOŘÁK a Lukáš ZRZAVÝ. Vision 2020 Unicorn Software Factory. *Unicorn Software Factory 2020* [online]. 2020 [vid. 2020-11-08]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-e12e6df93f1f43f7a0edfb47aabdfa18/book/page?code=54218139>
- [34] ZAJAC, Michal. Project Roles description Software Factory. *Unicorn Software Factory 2020* [online]. 2020 [vid. 2021-01-25]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-e12e6df93f1f43f7a0edfb47aabdfa18/book/page?code=35013370>
- [35] MICHALCOVÁ, Kristýna a Denisa LISSI. *Burza řešení* [online]. 2014 [vid. 2021-01-25]. Dostupné z: [https://plus4u.net/ues/sesm;jsessionid=94DB585A0842FE5F058728BEE45C7F87.0tcde18?REQID=6lax5FbWfQQ=&WINID=2mlb&action=t\\$a:v@1-0&SessFree=sues%253A29537617%253A50946981946976220](https://plus4u.net/ues/sesm;jsessionid=94DB585A0842FE5F058728BEE45C7F87.0tcde18?REQID=6lax5FbWfQQ=&WINID=2mlb&action=t$a:v@1-0&SessFree=sues%253A29537617%253A50946981946976220)
- [36] LABA, Ondřej, Martin HAVLÍK, Martin KLÍMA, Miloš DVOŘÁK, Volodymyr POHOYDA a Marek ŠŤASTNÝ. *Central Services Kick-Off* [online]. 2020 [vid. 2021-01-25]. Dostupné z: <https://uuos9.plus4u.net/uu-conferenceg01-main/99923616732505139-d5b0031a9bae4524aad94c188062e090/conference/downloads>
- [37] KOVÁŘ, Vladimír, Marek ŠŤASTNÝ, David KIMR, Lucie JANDOVÁ, Milan HYNEK a Jan KONRÁD. Product Team. *Knowledge Base - Unicorn Solutions* [online]. 2020 [vid. 2020-12-08]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/993fff06903f48a3a18489f6009e7ce6/book/page?code=65593999>
- [38] KOVÁŘ, Vladimír, David KIMR, Marek ŠŤASTNÝ, Jan KONRÁD a Jan JAROŠ. 10 klíčových principů Unicorn a.s. *Úvod do UAF* [online]. [vid. 2021-01-01]. Dostupné z: [https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-86b4d5a4a030400784764ebdb972bbda/book/page?code=10\\_klicovych\\_principu\\_unicorn\\_systems](https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-86b4d5a4a030400784764ebdb972bbda/book/page?code=10_klicovych_principu_unicorn_systems)
- [39] KOVÁŘ, Vladimír. Inspiration by the Key Methodologies on the market. *Unicorn Universe Process (uuP)* [online]. 2019 [vid. 2021-02-20]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-c86acb9189cb421892546005a1099ea7/book/page?code=50634193>
- [40] KOVÁŘ, Vladimír, Jindřich KALÍŠEK, Miroslava BAJANOVÁ, Milan TESAŘ a Radko PÖSCHL. *Unicorn ES Powered Company - Strategie*. 3. přeprac. Praha: Unicorn College, 2009. ISBN 978-80-87349-00-7.
- [41] KOVÁŘ, Vladimír, David KIMR, Milan TESAŘ a Radko POSCHL. *Unicorn ES Powered Company – Management*. 1. Prague: Unicorn College, 2009. ISBN 978-80-87349-01-4.
- [42] KOVÁŘ, Vladimír. Unicorn Universe Territory (uuT). *Unicorn Universe Process (uuP)* [online]. [vid. 2021-01-01]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-c86acb9189cb421892546005a1099ea7/book/page?code=06118768>
- [43] KOVÁŘ, Vladimír. Business Territory (uuBt). *Unicorn Universe Process (uuP)* [online]. [vid. 2021-01-01]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-c86acb9189cb421892546005a1099ea7/book/page?code=65669321>
- [44] KOVÁŘ, Vladimír. My Territory (uuMt). *Unicorn Universe Process (uuP)* [online].

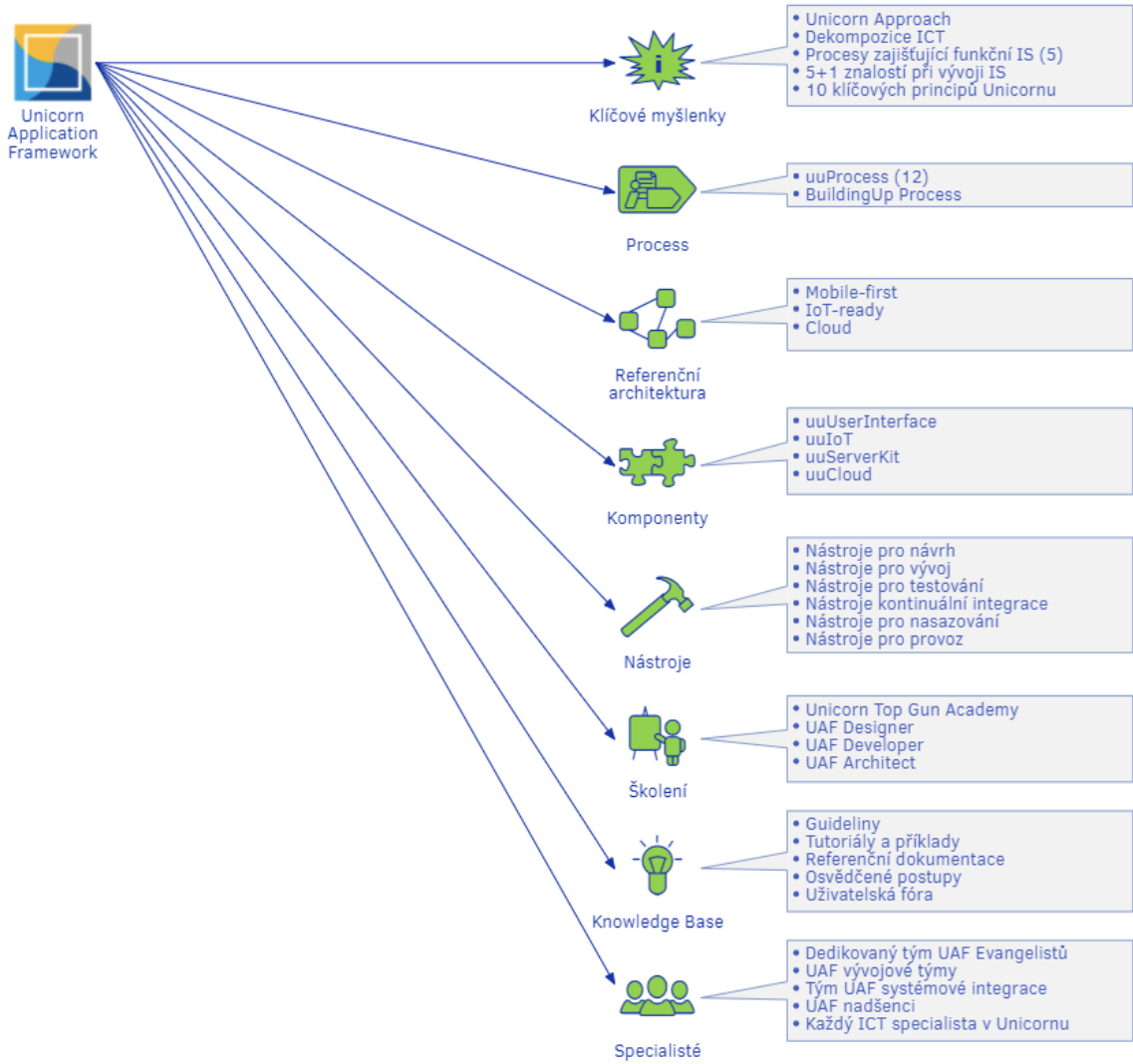
- [vid. 2021-01-01]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-c86acb9189cb421892546005a1099ea7/book/page?code=25318891>
- [45] KOVÁŘ, Vladimír. Artifact (uuArtifact). *Unicorn Universe Process (uuP)* [online]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-c86acb9189cb421892546005a1099ea7/book/page?code=78072795>
- [46] KOVÁŘ, Vladimír, David KIMR, Petr CIOCHOŇ a Marek ŠŤASTNÝ. Unicorn Application Framework - Hlavní myšlenka. *Úvod do UAF* [online]. 2020 [vid. 2020-11-07]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-86b4d5a4a030400784764ebdb972bbda/book/page?code=UAF01>
- [47] KOVÁŘ, Vladimír. Unicorn Mobile-First IoT-Ready Cloud Architecture and the uuApp Framework. *Unicorn Universe Process (uuP)* [online]. [vid. 2021-01-01]. Dostupné z: <https://uuapp.plus4u.net/uu-bookkit-maining01/c86acb9189cb421892546005a1099ea7/book/page?code=74820586>
- [48] KOVÁŘ, Vladimír, David KIMR, Marek ŠŤASTNÝ a Petr CIOCHOŇ. Životní cyklus uuApp. *Úvod do UAF* [online]. 2020 [vid. 2020-11-09]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-86b4d5a4a030400784764ebdb972bbda/book/page?code=e57bc863f57c422d88d6cf6afb183956dba6636c2d949bf99f07d920da11022>
- [49] KOVÁŘ, Vladimír, David KIMR, Marek ŠŤASTNÝ a Petr CIOCHOŇ. 5+1 Potřebných znalostí uuApp. *Úvod do UAF* [online]. 2020 [vid. 2020-11-07]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/78462435-86b4d5a4a030400784764ebdb972bbda/book/page?code=e2135b07a90f46ad9778831d35e0d0de5534eb28b9264383af89fde41633fb38>
- [50] IS, What, T H E RATIONAL a Unified PROCESS. *The Rational Unified Process*. 2003.
- [51] PAPER RATIONAL SOFTWARE WHITE. Rational Unified Process Best Practices for Software. *Development*. 2004. ISSN 07407459.
- [52] KOVÁŘ, Vladimír, David KIMR, Marek ŠŤASTNÝ, Radek DOLEJŠ, Milan MARTÍNEK, Vladimír ml. KOVÁŘ, Lucie MELOUNOVÁ a Linda BOROVSÁ. *How to Create a Business Requests Book What Is a Business Request ?* [online]. 2020 [vid. 2021-01-01]. Dostupné z: <https://uuapp.plus4u.net/uu-bookkit-maining01/0238a88bac124b3ca828835b57144ffa/book/page?code=39741830>
- [53] KOVÁŘ, Vladimír, Milan HYNEK, Marek ŠŤASTNÝ, David KIMR, Jan KONRÁD, Lucie JANDOVÁ a Miloš VODRÁŽKA. Overview - uuPhase , uuSprint Phase , Sprint and Solution Exchange Demand. *Knowledge Base - Unicorn Solutions* [online]. 2020 [vid. 2020-11-09]. Dostupné z: <https://uuos9.plus4u.net/uu-bookkitg01-main/993fff06903f48a3a18489f6009e7ce6/book/page?code=82190287>

## 6 Přílohy

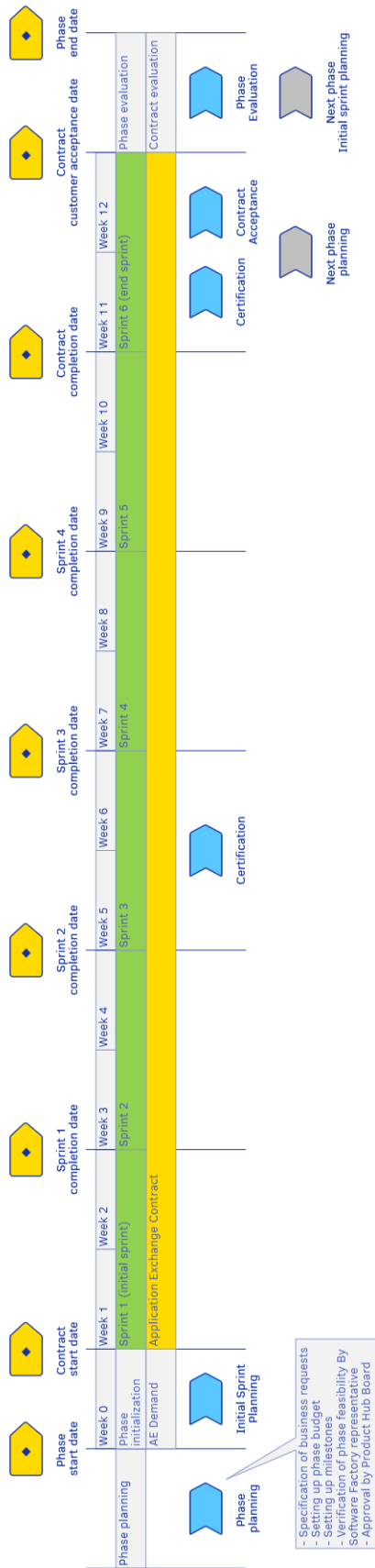
Příloha 1 Dvanáct procesů metodiky Unicorn Universe Process [31] .....	110
Příloha 2 Struktura Unicorn Application Framework [46] .....	111
Příloha 3 Průběh etapy [53].....	112
Příloha 4 Průběh sprintu [53] .....	113
Příloha 5 Životní cyklus produktového backlogu [Vlastní zpracování].....	114
Příloha 6 Životní cyklus ticketu [Vlastní zpracování].....	115
Příloha 7 Životní cyklus požadavku z primární podpory v uuBml [Vlastní zpracování] .	116
Příloha 8 Návrh komunikační matice při vývoji produktu v Unicorn Solution a.s. [Vlastní zpracování] .....	117
Příloha 9 User story in uuBusinessRequestMan Aggregated view in uuMT dw by uuIdentity - Submitter/Solver [Vlastní zpracování].....	118
Příloha 10 Business use-case in uuAppBusinessModelKit Tasklist aggregated view in uuMT dw by uuIdentity [Vlastní zpracování] .....	121
Příloha 11 Business use-case in uuAppBusinessModelKit Display tasklist for selected uuIdentity [Vlastní zpracování].....	126
Příloha 12 Business use-case in uuAppBusinessModelKit Add selected uuIdentity user to the preferred/unpreferred users (toggle) [Vlastní zpracování] .....	129
Příloha 13 Backlog request in uuSprintMan Aggregated view in uuMT dw by uuIdentity [Vlastní zpracování] .....	133



Příloha 2 Struktura Unicorn Application Framework [46]

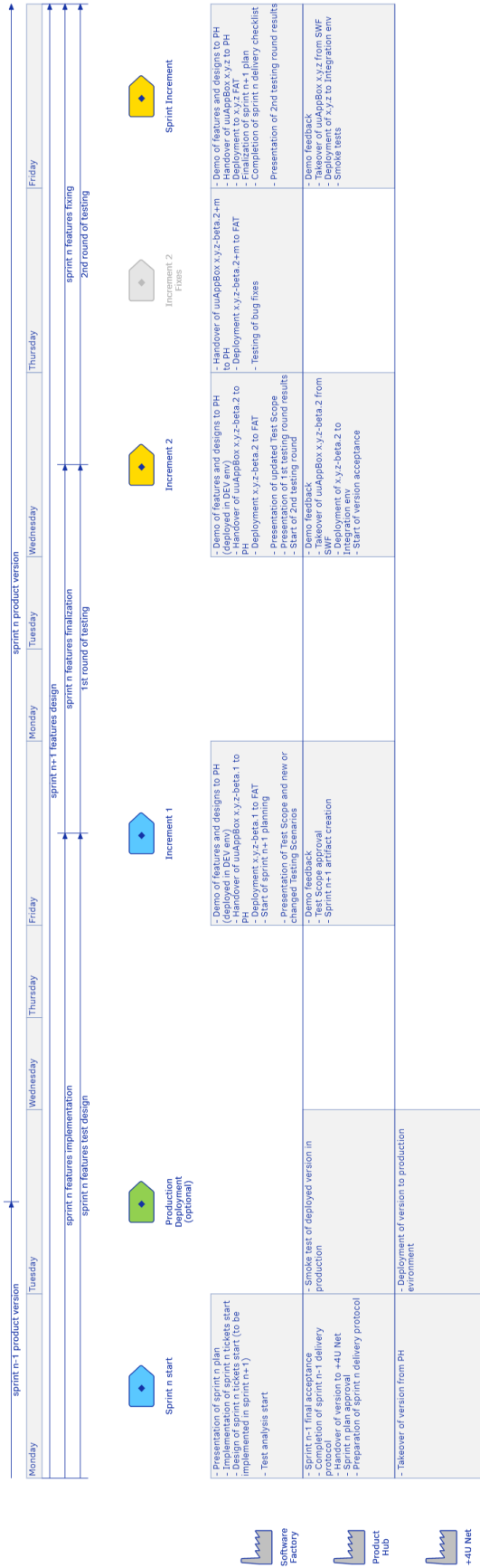


Příloha 3 Průběh etapy [53]

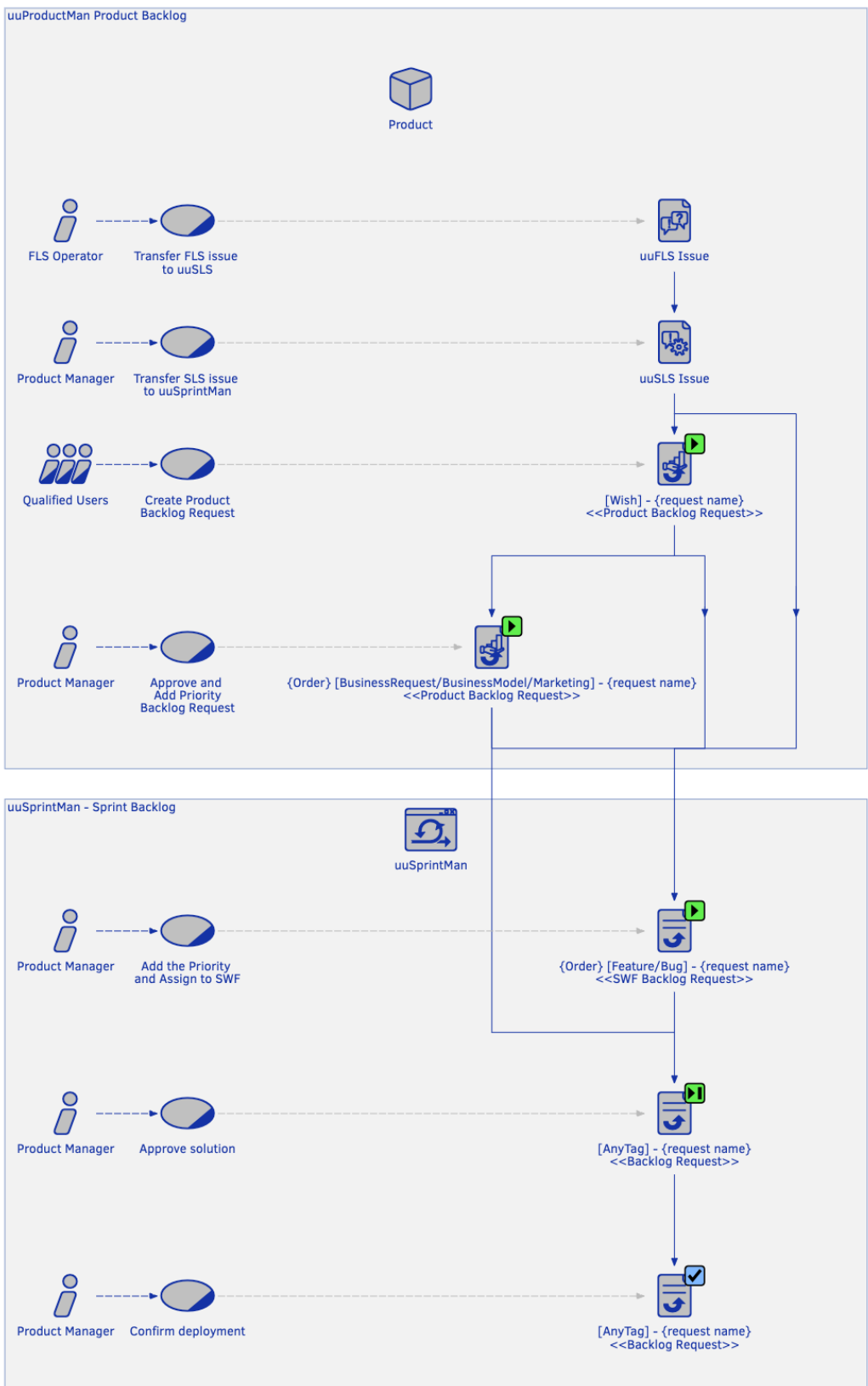




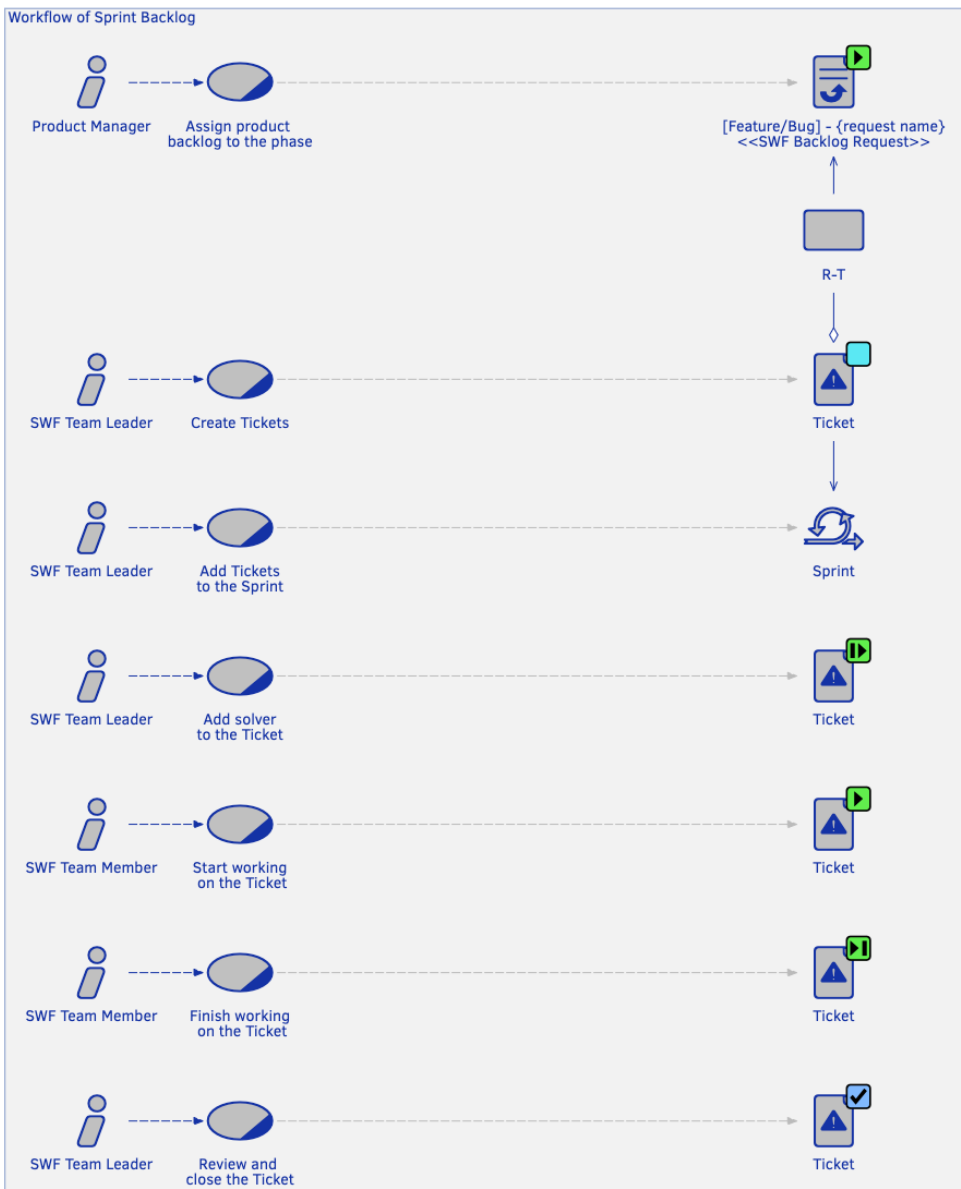
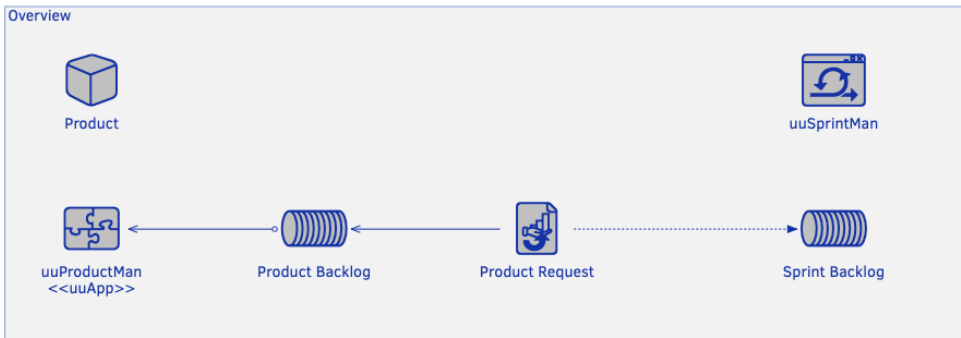
Příloha 4 Průběh sprintu [53]



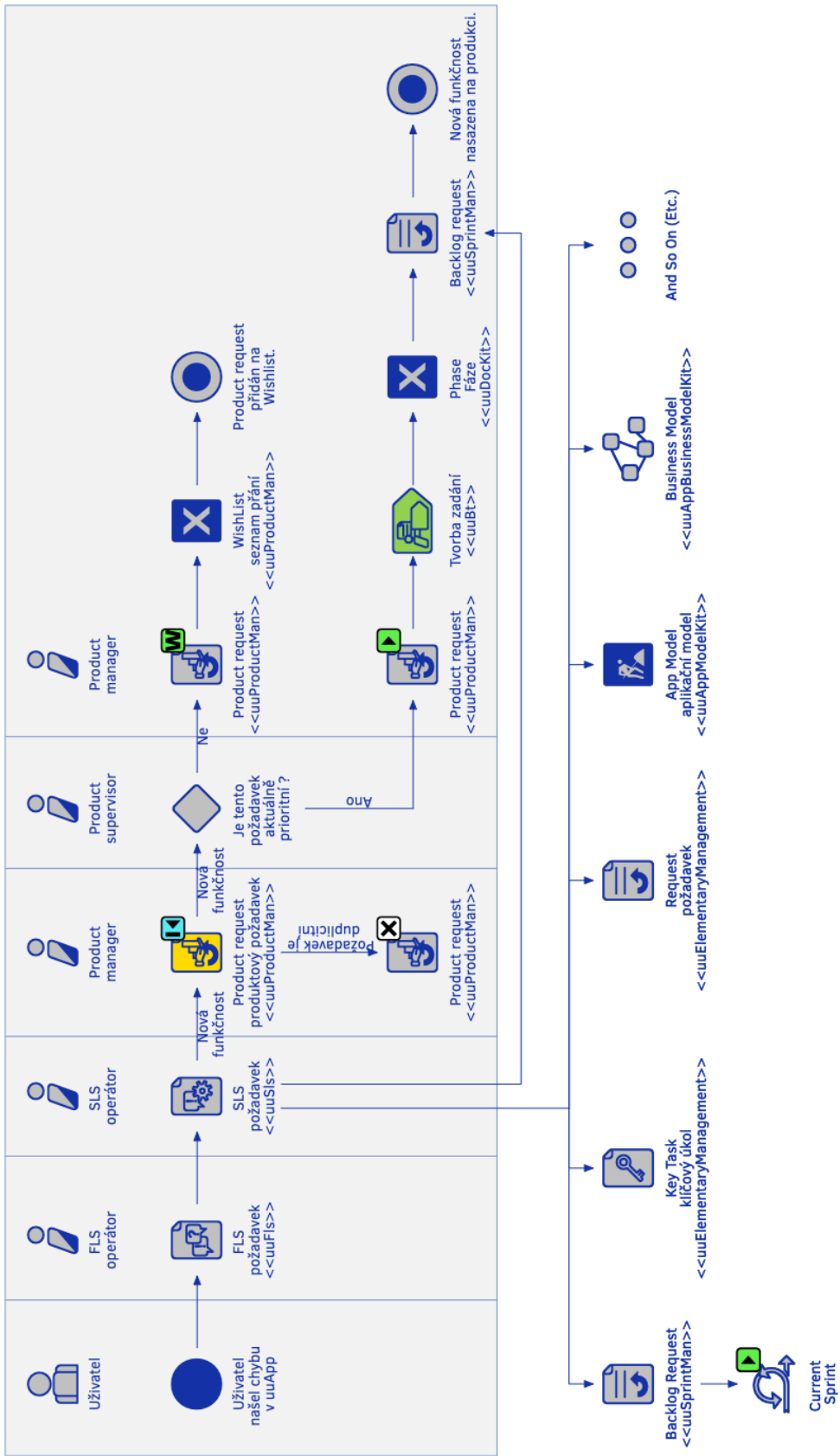
Příloha 5 Životní cyklus produktového backlogu [Vlastní zpracování]



Příloha 6 Životní cyklus ticketu [Vlastní zpracování]



Příloha 7 Životní cyklus požadavku z primární podpory v uuBml [Vlastní zpracování]









Příloha 8 Návrh komunikační matice při vývoji produktu v Unicorn Solution a.s. [Vlastní zpracování]

Typ komunikace	Zodpovědný	Formát	Délka	Výstup / Účel	Frekvence
Product team stand up	product manager	schůzka	15 minut	Zápis, úkoly a určená odpovědná osoba za úkol	denně
Development team stand up	team leader	schůzka	15 minut	Zápis, úkoly a určená odpovědná osoba za úkol	denně
Demonstration meeting #1	team leader	schůzka	60 minut	Zápis, úkoly a určená odpovědná osoba za úkol	1x za 2 týdny
Demonstration meeting #1	team leader	schůzka	60 minut	Ukázka přírůstku, zápis s úkoly a určená odpovědná osoba za úkol	1x za 2 týdny
Estimation meeting	team leader	schůzka	60 minut	Plánování obsahu scriptu	1x za 2 týdny
Planning meeting	team leader	schůzka	60 minut	Prezentace připraveného sprintu	1x za 2 týdny
Acceptance meeting	team leader	schůzka	30 minut	Akceptace odvedené práce během sprintu	1x za 2 týdny
Product status	product manager	schůzka	60 minut	Report aktuálního stavu	týdně
Status assessment	product manager	schůzka	60 minut	Report aktuálního stavu	týdně

Účastníci	product supervisor	product team coordinator	technology architect	business architect	product manager	UX Evangelist	uuApp developer	senior uuApp developer	uuApp designer	senior uuApp designer	team leader - development team	team leader - test team
product supervisor												
product team coordinator		x										
technology architect			x									
business architect				x								
product manager					x							
UX Evangelist						x						
uuApp developer												
senior uuApp developer							x					
uuApp designer												
senior uuApp designer								x				
team leader - development team										x		
team leader - test team												x

## Aggregated view in uuMT dw by uuIdentity - Submitter/Solver

This page contains the description of the User Story **Aggregated view in uuMT dw by uuIdentity - Submitter/Solver**.

Basic Info 	
Name	<b>Aggregated view in uuMT dw by uuIdentity - Submitter/Solver</b>
Brief Description	Radek, an employee of Unicorn holding (a Chief evangelist, Solution Exchange Supplier, and so on), wants to see the filtered task list according to the uuIdentity, who participates in the activity as submitter or solver.
State	 Specified
Complexity	B - Simple
Priority	Must have
Responsible role	 <a href="#">uuMyTerritory Product Manager</a>
Artifact	  <a href="#">Aggregated view in uuMT dw by uuIdentity - Submitter/Solver</a>
Folder	 <a href="#">User Stories</a>
uuType	uu-businessrequestman-maing01/userStory

## Activity List ?

Tue 03/09/2021  
06:00 PM



✓ The first attempt - please review this User story

## Why we need this User Story

Our users have multiple roles in Territories and want to work systematically. They must have an opportunity to access the filtered tasklist easily. For example, tasklists filtered by the uuIdentity, who participates in the activity as submitter or solver. And save some uuIdentity as a favorite for future use.

Those days they must always set the filters because we offer them to save only one view. We will prepare preset touchIcon, which will have already filtered tasks from each submitter.

## User Story context

Radek is a Chief Evangelist from Unicorn Solutions (Product Hub), who has roles in Unicorn Solution uuBT, Unicorn uuBT, uuApp Exchange uuBT, Vigour, Unicorn Grammar school, Unicorn University and so on.

Every day he gets a lot of tasks and can easily be distracted by tasks, which are not important to him. Such as the new demand on uuApp Exchange, he is a contractor on to uuAppExchange and gets a notification about every new demand on uuAppExchange. He doesn't need to read them every day because being a contractor is not his primary job, so he does need to compete for demand and be the first. From time to time, he goes through them.

On the other hand, he works with the Board member on company strategy and updating all employees' education materials, that tasks are a priority. So the tasks from board member are more important, and he wants to go through the tasks from different uuIdentity separately to keep focus. He will also use it for his personal life. He can manage the tasks from the teacher of his son separate from the tasks he received from his wife and so on.

He wants to do it effectively and easily, without always setting filters in the task list.

## User Story goal

We need an aggregation view on the task list. It will be a list of touch icons that aggregate tasks of a certain type and show a badge with the count of tasks. When you click on the icon, it will show the appropriate filter's tasklist (i.e. tasks behind this icon).

Aggregation criteria - tasks by territory, tasks by uuIdentity, tasks by organizational unit, tasks by solver role, tasks by submitter role (more to come).

There can be more types of view than touch icons with badges - line list, bigger tiles, etc., but first, start with touch icons and badges.

## User Story initial situation

Radek has more than 500+ activities in his tasklist. Everyday he gets a lot of activities and he wants to see his tasklist filter by the uuIdentity.

Radek is working on updating education material with the board member and his colleague. He wants to focus on it this week because next week is a University conference where he is supposed to present the recent changes.

## User Story description

Radek chooses the uuIdentity with whom he has common activities and wants to focus. He can also select a favorite uuIdentity who will be displayed on the top of aggregated views list.

## Happy Day Scenario

1. Radek opened his tasklist in uuMT Digital Workspace.
2. He sees a list of icons that represent all uuIdentities with whom Radek has a common tasks.
  - 2.1. The list of touch icons contains a filter bar, where Radek can choose the filters such as new only, term (Past due date, 7+ and 14+), hide meetings, filter From me (I am submitter)/For me (I am solver). Radek chooses filters new only and hide meetings.
  - 2.2. The badge on the icon says how many activities, which match the given filters, Radek has. Because Radek chose the filter "new only" and "hide meetings," the badge number contains only the new activities he has common with the uuIdentity (person) and which are not type meeting.
3. Clicking on the uuIdentity icon (Avatar) opens a task list with filtered common tasks with the uuIdentity that are new and are not meetings.
4. In the new tab, he sees the filtered task list according to the given criteria. He can choose to add this uuIdentity to his favorite in the three dots menu.
5. Once Radek added the uuIdentity to his favorite, the icons' list will have a new row on top with favorite uuIdentity next time.





### Inputs ?



TaskList - aggregated view



## References ?





-  Aggregated view in uuMT dw by uuIdentity
-  Tasklist aggregated view in uuMT dw by uuIdentity
-  Add selected uuIdentity user to the preferred/unpreferred users (toggle)
-  Display tasklist for selected uuIdentity

*Priloha 10 Business use-case in uuAppBusinessModelKit Tasklist aggregated view in uuMT dw by uuIdentity [Vlastni zpracovani]*

## Tasklist aggregated view in uuMT dw by uuIdentity

This page contains the **specification** of the business use-case **Tasklist aggregated view in uuMT dw by uuIdentity**. The specification consists several mandatory business use-case description sections - **Basic Info**, **Task List**, **Business Use-case Specification**, **Process List**, **Business Use-case Launch List** and **Related Artifact List**.

### Basic Info ?

Name	<b>Tasklist aggregated view in uuMT dw by uuIdentity</b>
Description	Aggregated view via persons (by uuIdentity) of the current tasklist.
Priority	Must have
Complexity (mhr)	4
State	▶ Completed
State reason	
Responsible Role	 <a href="#">uuMyTerritory Product Manager</a>
Artifact	 ▶ <a href="#">Tasklist aggregated view in uuMT dw by uuIdentity</a>
Folder	 <a href="#">Business Use-cases</a>
Unit	 <a href="#">uuMyTerritory</a>
uuType	uu-uuappbusinessmodelkit-maing01/businessUseCase

## Business Use-case Specification

Aggregated view via persons (by uuIdentity) of the current tasklist.

Radek, an employee of Unicorn holding (a Chief evangelist, Solution Exchange Supplier, and so on), wants to see the filtered task list according to the uuIdentity, who participates in the activity as submitter or solver.

1. **Step**

Load data from tasklist.

2. **Step**

Create aggregated structure per uuIdentity.

3. **Concurrency**

3.A. **→**

Show aggregated structure with temporary icons of uuIdentities.

3.B. **→**

Load avatars for every uuIdentity (icon).

4. **Sequence**

The component will contain a filter bar, with options Past due date/7+/14+ term, solver/submitter activity, hide/show/only meetings, new only.

4.1. **Sequence**

User opened the selector for term.

4.1.1. **Step**

**If** User chose term Past due date.

System loads to the badge the information how many activities the user has incommon with the chosen uuIdentity to the due day today and older.

It counts all their incommon activities from past to the due day today . Example: Today is 1.3.2021 the badge will contain all activities with due day from past to the 1.3.2021.

4.1.2. **Step**

**If** User chose term 7+

System loads to the badge the information how many activities the user has incommon with the chosen uuIdentity to the due day today + 7.

It counts all their incommon activities from past to the due day today +7. Example: Today is 1.3.2021. The badge will contain all activitieswith due day from past to the 8.3.2021.

4.1.3. **Step**

**If** User chose 14+.

System loads to the badge the information how many activities the user has incommon with the chosen uuIdentity to the due day today + 7.

It counts all their incommon activities from past to the due day today +14 days. Example: Today is 1.3.2021. The badge will contain all activities with due day from past to the 15.3.2021.

4.2. **Sequence**

User opened the selector to filter own activities, activities from him (he is submitter) or activities for him (he is solver).

4.2.1. **Step**

**If** User chose For me.

System loads to the badge the information how many activities the user has incommon with the chosen uuIdentity and the user is a solver. The uuIdentity may be submitter or solver.

4.2.2. **Step**

**If** User chose From me

System loads to the badge the information how many activities the user has incommon with the chosen uuIdentity and the user is a submitter.

4.2.3. **Step**

System will hide the option Own.

4.3. **Sequence**

User opened the selector to filter meetings.

4.3.1. **Step**

**If** User chose Hide meetings

Systems load to the badge the information how many activities the user has incommon with the chosen uuIdentity except activities, which are meetings.

4.3.2. **Step**

**If** User chose Meetings only

Systems load to the badge the information how many meetings activities the user has incommon with the chosen uuIdentity.

4.3.3. **Step**

**If** User chose Including meetings

System loads to the badge the information how many activities the user has incommon with the chosen uuIdentity including activities, which are meetings.

4.4. **Step**

**If** User chose New only

System loads to the badge the information how many new activities the user has incommon with the chosen uuIdentity.

5. **Step**

**If** The User clicks on the chosen uuIdentity.

System opens a task list with filtered common tasks with the uuIdentity that match all the given criteria (filters) in new tab. (BUC Display tasklist for selected uuIdentity)

Process List ?

Search text ...



▶ Manage Tasklist

Product List ?

Search text ...



▶ Tasklist

Action: Read

Classification: Class

Business Use Case Launch List ?

Search text ...



▶ Owner

My territory owner.

## Business Use Case List



### Display tasklist for selected uuIdentity

Show filtered tasklist according to chosen uuIdentity .

## Inputs



Aggregated view in uuMT dw by uuIdentity - Submitter/Solver

## References








Update component for aggregation view by the Territory - add filter bar



Aggregated view in uuMT dw by uuIdentity

## Display tasklist for selected uuIdentity

This page contains the **specification** of the business use-case **Display tasklist for selected uuIdentity**. The specification consists several mandatory business use-case description sections - **Basic Info**, **Task List**, **Business Use-case Specification**, **Process List**, **Business Use-case Launch List** and **Related Artifact List**.

Basic Info 	
Name	<b>Display tasklist for selected uuIdentity</b>
Description	Show filtered tasklist according to chosen uuIdentity .
Priority	Must have
Complexity (mhr)	
State	▶ Completed
State reason	
Responsible Role	 <a href="#">uuMyTerritory Product Manager</a>
Artifact	 ▶ <a href="#">Display tasklist for selected uuIdentity</a>
Folder	 <a href="#">Business Use-cases</a>
Unit	 <a href="#">uuMyTerritory</a>
uuType	uu-uuappbusinessmodelkit-maing01/businessUseCase

## Activity List ?

### Business Use-case Specification

1. **Step**


**If** The user clicked on touchIcon with uuIdentity.

Clicking on the uuIdentity icon (Avatar) system opens a task list with filtered common tasks with the uuIdentity that match all the given criteria in new tab.

2. **Step**

User can save this view and add the uuIdentity to a favorite ones (BUC Add selected uuIdentity user to the preferred/unpreferred users (toggle)).

## Process List ?

Search text ... 

 ▶ [Manage Tasklist](#)

## Product List ?

Search text ... 

 ▶ [Tasklist](#)

Action: **Read**

Classification: **Class**

### Business Use Case Launch List ?



#### Tasklist aggregated view in uuMT dw by uuIdentity

Aggregated view via persons (by uuIdentity) of the current tasklist.

### Business Use Case List ?



#### Add selected uuIdentity user to the preferred/unpreferred users (toggle)

When the User clicks on uuIdentity, the uuIdentity is set as the preferred view for the User in his aggregated list of tasks, and when he removes uuIdentity, uuIdentity will be among the other uuIdentities.

### Inputs ?



Aggregated view in uuMT dw by uuIdentity - Submitter/Solver

### References ?







Aggregated view in uuMT dw by uuIdentity



## Add selected uuIdentity user to the preferred/unpreferred users (toggle)

This page contains the **specification** of the business use-case **Add selected uuIdentity user to the preferred/unpreferred users (toggle)**. The specification consists several mandatory business use-case description sections - **Basic Info**, **Task List**, **Business Use-case Specification**, **Process List**, **Business Use-case Launch List** and **Related Artifact List**.

Basic Info <span>?</span>	
Name	<b>Add selected uuIdentity user to the preferred/unpreferred users (toggle)</b>
Description	When the User clicks on uuIdentity, the uuIdentity is set as the preferred view for the User in his aggregated list of tasks, and when he removes uuIdentity, uuIdentity will be among the other uuIdentities.
Priority	Must have
Complexity (mhr)	
State	▶ Completed
State reason	
Responsible Role	 <a href="#">uuMyTerritory Product Manager</a>
Artifact	 ▶ <a href="#">Add selected uuIdentity user to the preferred/unpreferred users (toggle)</a>
Folder	 <a href="#">Business Use-cases</a>
Unit	 <a href="#">uuMyTerritory</a>
uuType	uu-uuappbusinessmodelkit-maing01/businessUseCase

## Business Use-case Specification

When the User clicks on uuIdentity, the uuIdentity is set as the preferred view for the User in his aggregated list of tasks, and when he removes uuIdentity, uuIdentity will be among the other uuIdentities.

1. **Step**

The user opened the task list from the aggregated view and is looking at the displayed tasklist for the selected uuIdentity.

2. **Step**

**If** The user clicked on the button "Add to favorites".

The system set the uuIdentity as the preferred view for the user in his aggregated list of tasks. The touchIcon list will start with this preferred (favorite) uuIdentities.

3. **Step**

**If** There are no common tasks that meet the given criteria.

The uuIdentity stays in the preferred uuIdentities in the touchIcon list with badge 0.

The system opens a task list with filtered common tasks with the uuIdentity that match all the given criteria (filters) in the new tab and shows the warning "There are no records that match given criteria."

4. **Step**

**If** The user clicked in the touchIcon list of aggregated view by uuIdentity on the preferred (favorite) uuIdentity. The tasklist for the chosen uuIdentity was opened.

5. **Step**

**If** The user clicked on the button "Remove from favorites" .

The system removes the uuIdentity from the list of preferred uuIdentities for the user in his aggregated list of tasks and make the uuIdentity unpreferred.

The touchIcon list will no longer start with this preferred (favorite) uuIdentities. This unpreferred uuIdentity will be part of the rest of uuIdentities, which are not preferred.

Process List 

Search text ... 

 ▶ [Manage Tasklist](#)

Product List 

Search text ... 

 ▶ [uuUserPreference Property](#)

Action: Read, Update

Classification: Class

Business Use Case Launch List 

Search text ... 

 ▶ [Owner](#)

My territory owner.

 ▶ [Display tasklist for selected uuIdentity](#)

Show filtered tasklist according to chosen uuIdentity .

## Business Use Case List

Search text ...



## Inputs

 Aggregated view in uuMT dw by uuIdentity - Submitter/Solver

## References

 Aggregated view in uuMT dw by uuIdentity

# uuMT-BR160-Aggregated view in uuMT dw by uuIdentity ?

## Basic info ?

Name

**Aggregated view in uuMT dw by uuIdentity**

Code

uuMT-BR160

Description

Meeting minutes from [2021.03.10 10:00 - 10:30](#), [Open issues from SWF to sprint 5](#)

**Take the uuIdentity solver and submittery from TileProps up.** Update the guideline, until the 30th March it will be optional, starting from April 2021 it will be mandatory for all uuApps.

It is also an intervention in other applications such as uuFinMan - Misha will have a meeting with them.

**Legacy tile** must send uuIdentity submitter and solver (Legacy record migration)  
The need to migrate records.

---

Prepare component for an aggregated view in uuMt dw by uuIdentity, which will include list of touchIcons with badge, which contain information about number of activities that match set filters.

The component must have filters - new only, for me/from me, term Past due date/7+/14+ look in input there is a business model and user story.

### **It sprint 6 add Favorites**

It must be also possible to add uuIdentity to favourite, which will be preferred in the aggregated view (on the top of the touchIconList).

Priority

**Must have!**

High-level estimate

13

Precise estimate

8

Due date

2021/03/23

Tags

TOP Business Request

State

▶ IN PROGRESS



Artifact

▶ Aggregated view in uuMT dw by uuIdentity

Folder

Backlog

Responsible role

uuMyTerritory Product Manager

Inputs ?

Create relation



Aggregated view in uuMT dw by uuIdentity - Submitter/Solver

View data

Tasklist aggregated view in uuMT dw by uuIdentity

View data

Display tasklist for selected uuIdentity

View data

Add selected uuIdentity user to the preferred/unpreferred u...

View data





Tickets ?

Create





Showing 2 items of 2

Columns

Priority	Name	Responsible Solver
Must have!	 ► Aggregated view in uuMT dw by...	 uuApp Designer UU-B...
Must have!	 ► Aggregated view in uuMT dw by...	 uuApp Developer UU-...

**Activity list**  + Create 

**References**  

List by artifact-artifact relations is empty