

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREOVÉHO INŽENÝRSTVÍ



Diplomová práce

**Analýza a návrh iOS aplikace pro  
komunikaci a kooperaci při vývoji software**

*Bc. Pavel Šmejkal*

Vedoucí práce: Ing. David Buchtela, Ph.D.

30. dubna 2015

---

## Poděkování

Děkuji vedoucímu práce Ing. Davidovi Buchtelovi, Ph.D. za odborné vedení mé práce. Poděkování patří také mým nejbližším za jejich podporu po celou dobu studia.

---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. Dále prohlašuji, že jsem s Českým vysokým učení technickým v Praze uzavřel dohodu, na základě níž se ČVUT vzdalo práva na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Tato skutečnost nemá vliv na ust. § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů.

V Praze dne 30. dubna 2015

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2015 Pavel Šmejkal. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Šmejkal, Pavel. *Analýza a návrh iOS aplikace pro komunikaci a kooperaci při vývoji software*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

---

# Abstrakt

Práce se zabývá možnostmi využití nástrojů pro podporu komunikace a kooperace při vývoji software. První část detailně analyzuje produkty na trhu. Následuje rozbor komunikačních požadavků a postupů ve vybrané společnosti. Práce popisuje návrh a implementaci softwarového nástroje pro podporu specifických procesů v dané firmě, založený na aplikaci pro mobilní platformu iOS. Dále bylo provedeno testování a zhodnocení ekonomických přínosů aplikace.

**Klíčová slova** Komunikace, kooperace, vývoj software, efektivita, výkon, mobilní zařízení, iOS

---

## Abstract

This thesis looks at the ways of using software tools for supporting communication and cooperation in the software development process. The first part includes a detailed analysis of the products on the market followed by an investigation of the communication requirements and processes at a company active on the Czech and US market. The thesis also describes the design, implementation and testing of a software tool for supporting specific processes within that company, based on an iOS mobile app. The thesis will also examine the app's merit evaluation process.

**Keywords** Communication, cooperation, software development, efficiency, performance, mobile devices, iOS

---

# Obsah

Úvod	1
<b>1 Cíl práce</b>	<b>2</b>
<b>2 Komunikace a koordinace při vývoji software</b>	<b>4</b>
2.1 Komunikace . . . . .	4
2.2 Komunikace a kooperace při vývoji software . . . . .	5
2.3 Rozdělení komunikace při vývoji software . . . . .	7
2.4 Komunikace mezi vývojáři a projektovými manažery . . . . .	8
2.5 Příležitosti zlepšení komunikace mezi vývojáři . . . . .	9
2.6 Nástroje pro podporu komunikace a koordinace . . . . .	12
<b>3 Analýza nástrojů pro podporu komunikace a kooperace při vývoji software</b>	<b>14</b>
3.1 Obecná kritéria hodnocených nástrojů . . . . .	14
3.2 Nástroje pro výměnu zpráv a dalších typů příloh . . . . .	16
3.3 Software pro videokonference . . . . .	26
3.4 Nástroje pro podporu řízení projektů . . . . .	30
3.5 Nástroje pro ukládání a sdílení elektronického obsahu . . . . .	37
3.6 Celkové shrnutí nástrojů pro podporu komunikace a kooperace	41
<b>4 Analýza procesů a požadavků ve vybrané společnosti</b>	<b>43</b>
4.1 Popis společnosti . . . . .	43
4.2 Vývojový proces a spolupráce na projektech . . . . .	44
4.3 Nedostatky komunikace a kooperace a návrh zlepšení . . . . .	48
<b>5 Analýza softwarového řešení</b>	<b>51</b>
5.1 Cíle projektu . . . . .	51
5.2 Globální architektura systému . . . . .	51
5.3 Požadavky . . . . .	52

<i>OBSAH</i>	xi
5.4 Případy užití . . . . .	55
<b>6 Implementace softwarového řešení</b>	<b>59</b>
6.1 Použité technologie . . . . .	59
6.2 Implementace iOS aplikace . . . . .	62
<b>7 Testování a vyhodnocení přínosů aplikace</b>	<b>65</b>
7.1 Přínosy používání aplikace . . . . .	65
7.2 Testování . . . . .	66
7.3 Možná rozšíření aplikace . . . . .	67
<b>Závěr</b>	<b>68</b>
<b>Literatura</b>	<b>70</b>
<b>A Seznam použitých zkratk</b>	<b>75</b>
<b>B Ukázky aplikace</b>	<b>77</b>
<b>C Uživatelská příručka</b>	<b>85</b>
C.1 Pokyny ke spuštění zkompilevané verze aplikace . . . . .	85
C.2 Pokyny ke zkompileování aplikace . . . . .	85
<b>D Obsah příloženého CD</b>	<b>86</b>



---

## Seznam obrázků

2.1	Rozdílný pohled na zadání softwarového projektu . . . . .	5
2.2	Komunikace v týmu dle metodiky SCRUM . . . . .	10
3.1	Hipchat, komunikační nástroj od společnosti Atlassian . . . . .	19
3.2	Slack, přehledný a jednoduchý IM . . . . .	21
3.3	Flowdock, týmový komunikační nástroj . . . . .	23
3.4	Skype podporuje prakticky všechny zařízení s připojením k internetu	27
3.5	Použití nástroje Trello pro řízení projektu dle metodiky SCRUM .	32
3.6	JIRA představuje komplexní nástroj pro řízení projektů . . . . .	34
4.1	Proces vývoje softwarových produktů ve vybrané společnosti . . .	46
5.1	Globální architektura systému ReportIT . . . . .	52
5.2	Kontextový diagram ReportIT . . . . .	55
5.3	Případy užití přihlášení a výběru společnosti . . . . .	56
5.4	Případy užití po výběru společnosti . . . . .	58
6.1	Ukázka pravidel pro řízení přístupu do datového úložiště Firebase	60
6.2	ReportIT přehled – po přihlášení klienta (vlevo) a vývojáře (uprostřed a vpravo) . . . . .	63
B.1	ReportIT – přihlášení do aplikace . . . . .	77
B.2	ReportIT – výběr aktivní společnosti . . . . .	78
B.3	ReportIT – úprava profilu uživatele . . . . .	79
B.4	ReportIT – správa aktivní společnosti . . . . .	79
B.5	ReportIT – procházení zaměstnanců společnosti . . . . .	80
B.6	ReportIT – správa členů společnosti . . . . .	81
B.7	ReportIT – detail člena společnosti . . . . .	81
B.8	ReportIT – procházení a vytváření projektů . . . . .	82
B.9	ReportIT – nastavení projektu . . . . .	83
B.10	ReportIT – detail projektu . . . . .	83

<i>Seznam obrázků</i>	xiii
B.11 ReportIT – dashboard . . . . .	84

---

# Seznam tabulek

2.1	Bohatost módů komunikace . . . . .	7
-----	------------------------------------	---

---

# Úvod

V dnešní ekonomicky turbulentní době lze rychlost a kvalitu implementace softwarových projektů považovat za jeden z klíčů k jejich úspěchu. Do procesu vývoje se promítají potřeby zadavatele systému, i když on sám často z počátku neví, co vlastně potřebuje. To vše se děje na globálním trhu, kde projektové týmy nemusí mít geografické ani kulturní hranice.

Řízení takových projektů může být komplikované a jejich celková úspěšnost závisí na mnoha faktorech, schopnostech jednotlivců i fungování celého vývojového procesu. Podstatnou roli hraje také efektivita spolupráce a komunikace nejen uvnitř vývojového týmu ale i na rozhraní zadavatelské a vývojářské společnosti.

Výměna informací mezi zainteresovanými osobami musí probíhat hladce, kontinuálně a nezávisle na čase nebo lokaci. Díky mobilním zařízením je však dnes možné komunikovat prakticky kdykoliv a kdekoliv, a tak se chytré telefony a tablety staly nenahraditelnými prostředky v komunikaci a koordinaci při vývoji software.

Nástrojů pro podporu komunikace a kooperace lze na trhu najít mnoho, ale zvolit ideální variantu pro daný typ podniku není jednoduché. V úvahu je nutné vzít nejen poskytované funkcionality a cenu, ale také například integraci s jinými nástroji nebo právě podporu mobilních platforem.

---

## Cíl práce

Efektivita komunikace a kooperace při vývoji softwarových produktů se stala předmětem zkoumání z různých pohledů. Někteří studují její dopady na úspěšnost projektů, jiní vliv na pružnost vývojového procesu nebo náklady.

Tato práce se zaměřuje na problematiku softwarových nástrojů pro podporu spolupráce a dává si za cíl dva dílčí úkoly:

- Analýza nástrojů pro podporu komunikace a koordinace při vývoji software na trhu
- Návrh, implementace, testování a zhodnocení iOS aplikace pro podporu spolupráce dle analýzy vývojového procesu ve vybrané společnosti

### **Analýza nástrojů na současném trhu**

Nejdříve proběhne průzkum dostupné literatury zabývající se komunikací při vývoji software. Na jeho základě budou identifikovány nejpoužívanější typy nástrojů a jejich role ve vývojovém procesu.

U každého typu nástroje pak bude provedena analýza dostupných řešení na trhu a vyhodnocení jejich parametrů s ohledem na potenciálně odlišné priority dle společnosti. U některých lze totiž považovat za nejpodstatnější cenu, u jiných spíše rozsah funkcionalit nebo bezpečnostní prvky.

Na trhu se nachází velké množství různých produktů, které se zaměřují na podporu komunikace při vývoji software. Tato analýza se proto vymezuje zejména na potřeby malých a středních podniků, kde se často preferují jednoduché a uživatelsky přívětivé aplikace před robustními systémy s podporou pokročilých funkcí.

Malé a střední společnosti musí dbát více než kterékoliv jiné na rychlost a kvalitu implementace, které lze společně s flexibilitou obecně považovat za jeden z klíčových faktorů jejich úspěchu. Konkurenceschopnost pak do jisté míry závisí na schopnosti efektivně komunikovat uvnitř společnosti a koordinovat vývojový proces dle požadavků zákazníka.

---

Výběr vhodných nástrojů pro podporu komunikace a koordinace může tedy hrát podstatnou roli a prvním cílem je vyhodnotit vybrané aplikace z hlediska použitelnosti a přínosu pro tento typ firem.

### **Návrh a implementace aplikace**

Ve vybrané firmě proběhne analýza požadavků, procesů a používaných nástrojů pro podporu komunikace a kooperace. Na základě vyhodnocení aktuální situace bude navržen systém pro podporu atypických komunikačních procesů.

Dále bude dle návrhu implementována aplikace pro mobilní platformu iOS, tedy pro zařízení společnosti Apple typu iPhone, iPod a iPad. Pro centralizaci dat mezi zařízeními nebo pro ověřování totožnosti uživatelů bude nezbytná i serverová část systému. Tu zajistí stabilní serverové služby třetích stran a případné detaily budou vyřešeny pomocí jednoduché serverové aplikace založené na javascriptových technologiích.

Nakonec proběhne testování přímo se zaměstnanci a vyhodnocení ekonomických přínosů pro danou firmu. Součástí bude také popis plánů aplikace do budoucna.

---

# Komunikace a koordinace při vývoji software

## 2.1 Komunikace

Komunikace je základním prostředkem pro rozvoj živých bytostí. Lze ji definovat jako akt nebo proces použití slov, zvuků, signálů či chování k vyjádření a výměně informací nebo ke sdílení nápadů, myšlenek a pocitů druhé osobě [1].

Hlavním dorozumívacím prostředkem člověka byla odjakživa verbální komunikace, s příchodem písma se však možnosti komunikovat razantně rozšířily. Tisíce let poté lidstvo čelí novodobé revoluci. Rozmach digitálních technologií změnil způsob komunikace k nepoznání. Nejen, že jsme schopni dorozumívat se textově, hlasově i vizuálně nezávisle na vzdálenostech, nové technologie už pracují i na odstranění jazykových bariér. Společnost Skype totiž začala svým zákazníkům v roce 2014 poskytovat službu překládaných videohovorů v reálném čase [2].

Proto není divu, že se předávání informací stalo předmětem zkoumání v různých vědeckých oborech. Někteří studují psychologickou stránku komunikace, jiní naopak její dopady na fungování a výkon organizací. Americká NCA (National Communication Association) dokonce uznává 9 odlišných disciplín komunikace, ačkoliv se jejich rozsah často překrývá.

Komunikaci lze také dělit podle toho, mezi kým a jakým směrem probíhá. Většina odborníků se shoduje a rozlišuje pod tímto dělením čtyři druhy komunikace [3] [4]:

- intrapersonální (uvnitř jednoho člověka na úrovni myšlenek)
- interpersonální (vzájemná mezi dvěma nebo více lidmi)
- skupinová (zejména jednosměrná komunikace k více lidem)

- masová (jednosměrná k většímu počtu osob prostřednictvím nějakého nástroje nebo technologie)

Tato práce se zabývá především komunikací interpersonální a skupinové uvnitř vývojového týmu a směrem ke klientovi.

### 2.1.1 Komunikace v organizaci

Komunikaci uvnitř nějaké organizace lze definovat jako odesílání a přijímání zpráv mezi souvisejícími jednotlivci uvnitř určitého prostředí nebo za dosažení individuálních nebo společných cílů [5].

Organizace zaměřující se na vývoj zažily v historii mnoho neúspěchů a zklamání. Velká část z nich se dá připsat dvěma klíčovými faktorům – nedostatek zapojení zaměstnanců a neefektivní komunikace [6]. Proto je možné komunikaci považovat za základ efektivity a existence organizace. Spolupracovat nelze bez předávání informací a bez spolupráce nelze dosahovat společných cílů, pro které byla společnost založena.

## 2.2 Komunikace a kooperace při vývoji software

Ve firmách zabývajících se vývojem software je komunikace mezi všemi vrstvami vývoje extrémně důležitá. Ačkoliv celý vývojový tým směřuje ke stejnému cíli, každý typ oddělení má jiný pohled na produkt. Programátoři si mohou interpretovat zadání zcela jinak než produktový manažeréři nebo obchodní zástupci. Tuto problematiku dokonale znázorňuje kreslená vizualizace na obrázku 2.1 nazvaná Project Cartoon [7] [8].



Obrázek 2.1: Rozdílný pohled na zadání softwarového projektu

Zdroj [8]

Dalším důvodem, proč je komunikace při řízení softwarových projektů kritická, jsou časté úpravy a rozšiřování požadavků během vývoje. Příkladem je



projekt, kde v průběhu zadávání nemusí klient vědět, jak by vytvářený produkt měl přesně fungovat, nebo je obtížné funkce specifikovat.

Jedním z řešení je vypracování podrobného zadání, které zahrnuje společné diskuze a analýzu všech detailů. Takové zadání může trvat i několik měsíců, ale po přechodu do vývojové fáze už zásadní změny zadání nastávají jen zřídka.

V menších společnostech, ale často nelze každému klientovi vyhradit tým analytiků, kteří s ním definují zadání tak, aby přesně odpovídalo jeho potřebám. Zároveň může být flexibilita jedna z konkurenčních výhod těchto podniků. Proto se v dnešní době preferuje zavádění agilních metodik, například SCRUM nebo XP (extrémní programování), pro podporu pružnosti celého vývojového procesu [9]. S aplikací těchto metodik souvisí i zlepšování komunikace.

### 2.2.1 Agilní metodiky

Agilní vývoj je alternativou ke klasickému projektovému řízení. Agilní přístupy jsou typicky aplikovány při vývoji software, aby pomáhaly reagovat na nepředvídatelné situace [10].

Některé základní rysy agilních metodik jdou naprosto proti klasickým postupům softwarového inženýrství. Komunikace, schopnost reagovat na změny a rychlost vývoje se totiž upřednostňují před detailní analýzou a dokumentací. Důvodem pro zavádění takových postupů bývají potřeby klienta. Například o dokumentaci se totiž zákazníci často vůbec nezajímají, podstatný je výsledný produkt a zdali přinese očekávané.

### 2.2.2 Komunikace při agilním vývoji software

Agilní metodiky jsou založené na pravidelných iteracích, které se skládají ze tří částí – analýzy, implementace a průběžného předvádění meziprojektu zákazníkovi [11].

V každé z těchto částí je potřeba komunikovat. Během analýzy dostane tým priority od klienta a komunikace se odehrává mezi všemi vrstvami projektového týmu. Ve fázi implementace probíhá komunikace převážně mezi členy vývojového týmu, případně testery a oddělením odpovídající za kvalitu produktu (QA). Při předvádění produktu mezi sebou komunikují zástupce společnosti a zákazník.

M. Hummel, Dr. C. Rosenkranz a Prof. Dr. R. Holten ve své práci [12] rozebírají roli komunikace v agilním vývoji software dle různých studií, které byly na toto téma vypracovány. Shrnutí nasbíraných informací pojednává o komunikaci jako klíčovém prvku agilního vývoje a zároveň o tom, jak agilní metodiky podporují efektivnější předávání informací. Jako jeden z příkladů uvádí vývoj software v distribuovaných týmech, kde členové při kooperaci na projektech čelí mnoha výzvám. Agilní metodiky pomáhají překonat složitosti

tým, že nutí tým komunikovat. Na druhou stranu může být potřeba zavádět specifické komunikační postupy.

## 2.3 Rozdělení komunikace při vývoji software

Ve společnostech pro vývoj software lze tým považovat za vyhraněnou skupinu lidí pro podporu daného procesu, kde členové týmu spolupracují a vyměňují si informace. Komunikace pak můžeme dělit dle různých pohledů.

### 2.3.1 Módy komunikace

Komunikaci lze dělit podle toho, prostřednictvím jakého média probíhá. Příkladem může být email, telefon, videohovor ale také textová a diagramová dokumentace, wiki weby apod.

Dle teorie *Media Richness Theory* [13] [14] lze každý typ ohodnotit stupněm „bohatosti“, který popisuje, jak moc vyměňovaných informací lze reprodukovat. Například videohovor lze považovat za informačně bohatší než email, protože kromě slov obsahuje také informace vyplývající z tónu hlasu a přenášeného obrazu.

Všeobecně se za nejbohatší komunikaci považuje konverzace tváří v tvář. V agilních vývoji se však často vyzdvihuje ještě rozšířená verze – konverzace, při které účastníci mohou používat tabuli pro zápisky a znázornění myšlenek. Ta se pak na stupnici bohatosti řadí samostatně.

Zároveň lze rozlišit, zdali jsou vyměňované informace ukládány a budou k dispozici v budoucnu. Po rozšíření klasických přehledů [15] je možné sestavit následující tabulku. Módy jsou seřazeny od nejbohatšího typu komunikace.

Tabulka 2.1: Bohatost módů komunikace

Uchovatelné	Komunikační mód
Ne	Konverzace tváří v tvář se znázorněním na tabuli Konverzace tváří v tvář Videokonference Konverzace přes telefon
Ano	Elektronická synchronní komunikace (chaty a IM) Konverzace přes email (asynchronní) Psané osobní komunikace (dopisy, vzkazy) Psaná formální komunikace (zprávy, dokumenty) Číselná média (výstupy z počítače)

### 2.3.2 Interní vs. externí komunikace

Na komunikaci při vývoji můžeme nahlížet dle toho, zdali se děje uvnitř firmy, nebo při kontaktu se zástupci klientské společnosti. Na komunikaci s klientem bývá vyčleněna jedna nebo více odpovědných osob pro každý projekt. Ti poté předávají informace ostatním členům týmu a koordinují chod projektu.

Za interní komunikaci lze tedy považovat vše od výměny informací mezi programátory, testery a projektovými manažery až po koordinaci prodeje nebo řízení lidských zdrojů.

Efektivní externí komunikace je důležitá zejména kvůli možnosti reagovat na změny a zpětnou vazbu co nejdříve. Informování klienta o každodenním pokroku na projektu pomáhá předcházet prohlubování problémů a zároveň dává klientovi pocit, že se projekt posunuje správným směrem.

Efektivita různých komunikačních módů se liší podle toho, zdali komunikujeme uvnitř firmy či s klientem. Používání chatu a IM uvnitř vývojového týmu lze považovat za jeden z primárních komunikačních kanálů, proti tomu email preferují zaměstnavatelé zejména při komunikaci s klientem. Stejně tak využívání konceptuálních diagramů a nákresů lze považovat za efektivní spíše uvnitř vývojového týmu [16].

### 2.3.3 Spolupráce v lokálních a distribuovaných týmech

Výzvou pro efektivní komunikaci je také distribuované prostředí. Nejedná se pouze o společnosti operující v různých zemích světa. V dnešní době nemusí být ani zaměstnanci stejné pobočky každý den v jedné kanceláři. Zaměstnavatelé nezdědka umožňují alespoň částečnou práci z domova. Obchodní zástupci nebo projektoví manažeri zase často vyjíždějí za klienty i za hranice státu.

Ať už je distribuce týmu jakákoliv, členové musí být konstantně v kontaktu. Z komunikačních módů je potřeba zvolit ty nejefektivnější/nejbohatší, které lze využít. Proti lokálně založeným týmům, ve kterých je nejúčinnější přímá konverzace tváří v tvář, v distribuovaných týmech se jedná především o videokonference a emaily nebo chat. Ty se musí stát přirozeným a pravidelným způsobem předávání informací [16].

Výzkumníci, zabývající se problémy kooperace v distribuovaných vývojových týmech, však zdůrazňují, že větší vzdálenost i přes znatelné pokroky v komunikačních technologiích negativně ovlivňuje dynamiku týmu [17].

## 2.4 Komunikace mezi vývojáři a projektovými manažery

Projektový management lze v dnešní době považovat za osvojený nástroj pro rozvoj a úspěch podniku. Díky tomu vzniklo během posledních desetiletí mnoho frameworků, metodologií a přístupů.

Agilní metodiky vývoje software umožnily projektovým manažerům soustředit se na zjednodušování fungování týmů a koučování členů namísto přímého úkolování. Role projektového manažera se tedy změnila – místo rozdávání příkazů a kontrolování celého projektu je jeho hlavní náplní udržovat vizi projektu a zjednodušovat podmínky pro práci vývojářů, aby se mohli plně soustředit na vývoj produktu [18]. Tím zároveň zajišťuje, že bude projekt dotažen do cíle.

Komunikační schopnosti projektového manažera jsou jednou z jeho nejkritičtějších dovedností, protože se stává prostředníkem mezi zadavateli a těmi, kteří mají schopnost vytvářet. Musí také dokázat porozumět potřebám klienta a zároveň chápat úskalí vývoje.

### 2.4.1 Komunikace při řízení projektů dle metodiky SCRUM

Dle metodiky SCRUM se odpovědnosti klasického projektového manažera štěpí do dvou koordinačních rolí. Tyto role obvykle zastávají dva různí lidé, není to však pravidlem.

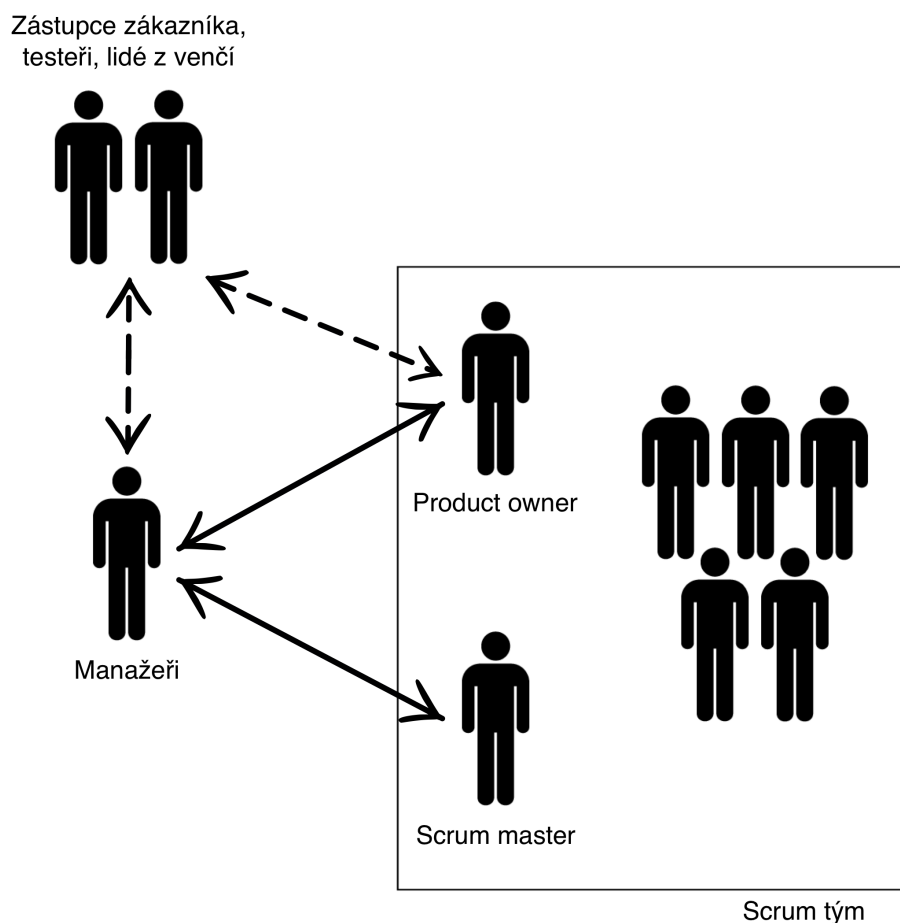
První rolí je product owner, který má za cíl úspěšné dokončení projektu. To sebou obnáší udržování vize a rozhodování, které části produktu mají prioritu v následujícím sprintu [19] [20].

Scrum master je druhým článkem, který se snaží vytvořit ideální podmínky pro práci vývojového týmu. Členy týmu řídí a zároveň je chrání před vnějšími vlivy [20] [21]. Schopnost komunikace a diplomacie by měli rozhodně patřit mezi jeho přední vlastnosti. Pokud vznikne neshoda mezi programátorem a manažerem mimo vývojový tým, scrum master má za úkol spor vyřešit tak, aby se nenarušil vývoj projektu. Obrázek 2.2 znázorňuje vnější komunikaci scrum týmu.

## 2.5 Příležitosti zlepšení komunikace mezi vývojáři

Někteří vývojáři tvrdí, že softwarový projekt může být často úspěšný i bez komunikace uvnitř jejich týmu. Dokonce existuje mnoho příkladů podporujících toto tvrzení, kde skupina programátorů úspěšně pracuje na jednom projektu z různých částí světa. Praxe ale ukazuje, že většina vývojových týmů má možnost zdokonalit své výsledky zlepšením komunikace. Potenciálem jsou zejména [22]:

- prevence nedorozumění
- možnost přiučit se novému od ostatních
- identifikace problémů před tím než nastanou
- porozumění projektu jako celku, omezit neznalost ostatních částí produktu



Obrázek 2.2: Komunikace v týmu dle metodiky SCRUM

Postupů, jak zlepšovat komunikaci uvnitř vývojového týmu, existuje více a zaměstnavatelé i vývojáři se v názorech různí. Za základní lze považovat metody rozebrané v následujících podsekcích.

### 2.5.1 Standardizace používání verzovacích nástrojů

Pokud se jedná o informovanost související s produkovaným kódem a jeho kvalitou, jednou z příležitostí pro zefektivnění je systematické využívání funkcí verzovacích systémů. Lze je standardizovat za pomoci správně nastavené firemní politiky tak, že se zvýší informovanost členů v týmu a případné problémy je možné odhalit včas před vydáním ostré verze. V praxi si pak programátoři kontrolují kód navzájem před tím, než jsou změny propagovány do produkčních větví repozitáře.

### 2.5.2 Zavedení chatů a IM (Instant Messaging)

Chaty a služby IM (posílání zpráv v reálném čase), mohou být považovány za rušivý element při práci zaměstnanců, neboť jsou často spojovány s vyřizováním soukromých záležitostí.

Využívání chatů pro pracovní účely může být ale v některých ohledech jednodušší a efektivnější než domluva tváří v tvář nebo emaily. Například když programátor řeší záludný implementační problém a potřebuje konzultaci nějakeho ze svých zkušenějších kolegů. Protože ještě většinu týmu moc dobře nezná, bude pro něj jednodušší zeptat se hromadně přes veřejný kanál na chatu. Proti konzultaci s jedním z kolegů se zároveň zvyšuje šance, že mu alespoň jeden ze skupiny bude schopný pomoci.

Další výhodou je vyhledávání v historii konverzací. Možnost zobrazení starých zpráv může předcházet opakovanému konzultování již vyřešených problémů. Jeden z veřejných kanálů lze také používat jako sběrnici častých problémů a jejich řešení, ve kterém pak mohou členové týmu vyhledávat.

V neposlední řadě lze za výhodu používání chatů a messengerů považovat možnost komunikovat přes mobilní zařízení, ať už je vývojář v tramvaji cestou do práce nebo na pracovní cestě.

### 2.5.3 Pravidelné schůze vývojářů

Pokud vývojáři pravidelně nekomunikují, může se stát, že budou příliš vyhrazení pro svůj úkol a nemají ponětí, na čem pracují ostatní.

Krátké pravidelné porady zlepšují informovanost a předcházejí nedorozuměním. Mohou také zvyšovat motivaci jedinců. Když totiž někdo slíbí dokončení zadaného úkolu do následujícího týdne, všichni se dozví, že selhal pokud tak neučiní.

Lze aplikovat každodenní schůze neboli „stand-up meeting“, které zavádí metodika SCRUM. Celý vývojový tým se schází na začátku pracovního dne, aby ve stoje probral naplánovanou práci. Podle velikosti týmu trvá schůzka typicky od 5 do 15 minut, díky čemuž je diskuze svižná a k věci.

### 2.5.4 Zavedení extrémního programování

Extrémní programování (XP) je agilní metodika založená na čtyřech základních hodnotách – komunikace, jednoduchost, zpětná vazba a odvaha, které jsou pevně svázané se čtyřmi aktivitami – psaní kódu, testování, naslouchání a vychytávání chyb [23].

Podle zakladatele XP K. Becka z aplikace primárních hodnot a aktivit vyplývá 12 základních praktik, které obsahují rozšíření business procesů, týmových metod i individuálních programovacích návyků. Tyto postupy obecně zjednodušují neformální komunikaci, nejpodstatnější v tomto ohledu však lze považovat párového programování [12].

Zavedení postupů XP může přinést mnoho výhod a podpořit flexibilitu vývojového procesu. Zlepšení komunikace a kooperace jsou nedílnou součástí aplikovaných změn.

## 2.6 Nástroje pro podporu komunikace a koordinace

Komunikaci můžeme vždy podpořit nějakým nástrojem, ať už se jedná o spolupráci na úrovni vývojového týmu, projektového manažerství nebo komunikaci s klientem.

Při osobním setkání to mohou být výše zmíněné tabule pro znázornění myšlenek, nebo také obyčejná tužka a papír. Pokud se ale jedná o softwarové nástroje, pokrok digitalizace v posledních desetiletích přivedl úplně nové možnosti, jak se dorozumívat.

### 2.6.1 Typy komunikačních a kooperačních nástrojů

Email a telefon lze považovat za tradiční nástroje pro komunikaci uvnitř i vně společnosti. Dnes však lze za stejně tradiční brát i mnohé další softwarové aplikace.

V první řadě se jedná o programy a služby pro výměnu zpráv a to nejlépe v reálném čase. Kromě posílání textu a datových souborů jsou však často užitečné také videokonferenční nástroje, které lze považovat za vlastní kategorii produktů.

Společnosti zabývající se vývojem software jsou založeny na produkci softwarových řešení, za kterými stojí nejrůznější elektronický obsah. Ten musí zaměstnanci někde ukládat a sdílet mezi sebou. Nástroje pro sdílení kódu, dokumentů a dalších souborů lze tedy považovat při vývoji software za fundamentální.

Společnost by však v konkurenci nedokázala obstát bez systematické práce. Jedním z klíčových nástrojů jsou proto aplikace pro celkové řízení a organizaci projektů, umožňující efektivní spolupráci.

V malých a středních podnicích pro vývoj software jsou tedy vedle tradičních nástrojů nejpodstatnější následující typy kooperačního software.

- Diskuzní nástroje, podporující výměnu zpráv a dalších typů příloh
- Software pro videokonference a virtuální jednání
- Nástroje pro podporu řízení projektů, správu úkolů a rozdělování práce
- Nástroje pro ukládání a sdílení kódu, dokumentů a jiných souborů souvisejících s vývojem produktu

## 2.6. Nástroje pro podporu komunikace a koordinace

---

Analýza nástrojů v následující kapitole se zaměřuje na různá řešení v těchto čtyřech oblastech.



# Analýza nástrojů pro podporu komunikace a kooperace při vývoji software

Prostředků pro podporu komunikace a kooperace lze na trhu najít velké množství. Proto mají manažeři před sebou těžká rozhodnutí. Musí totiž vyhodnotit současné a případně i budoucí potřeby podniku, na jejichž základě pak vyberou tu nejlepší variantu samozřejmě s ohledem na cenu.

Tato kapitola se zabývá analýzou produktů z každé skupiny komunikačních nástrojů zmíněných v předchozí části práce. Za cíl si klade odhalit kritéria, na základě kterých se může vedení společnosti rozhodnout, jaký software zvolit. Popis vybraných produktů pak nastíní, jaké možnosti při podpoře komunikace a kooperace existují.

## 3.1 Obecná kritéria hodnocených nástrojů

U softwarových nástrojů lze definovat obecná kritéria aplikovatelná na veškerý podpůrný software ve společnosti jako je cena, podpora různých platforem nebo grafické zpracování. Svou roli hrají ale také specifická kritéria. Těmi budou převážně funkcionality daného typu produktu a definována budou zvlášť v příslušných podkapitolách.

### 3.1.1 Cena a platební model

Jedním z rozhodujících kritérií je samozřejmě cena. V dnešní době je většina podniků ochotna platit za nástroje pro podporu svého podnikání. Na druhou stranu by bylo ekonomicky nerozumné vydávat neodpovídající prostředky za příliš jednoduché programy nebo podporu funkcí, které firma ani nepotřebuje.

Platební modely mohou být různé. Často se však jedná o měsíční nebo roční platby. V některých případech se také platí za každého připojeného

uživatele. V tom případě bude hodnocení pracovat s přepočítanou měsíční cenou v malém a středním podniku.

Malý podnik lze podle počtu lidí definovat jako společnost zaměstnávající 10-50 osob, zatímco střední podnik od 50 do 250 zaměstnanců [24]. V této analýze bude pro výpočet ceny softwaru vždy předpokládán průměrný malý nebo střední podnik, tedy 30 respektive 150 zaměstnanců.

#### 3.1.2 Podpora různých platforem

Základem většiny dnešních softwarových nástrojů jsou webové aplikace. Web je možné navrhnout tak, že bude uživatelsky přívětivý na všech typech zařízení (tzv. responsive web design = responzivní webový návrh).

Ačkoliv můžeme web spustit v prohlížeči v mobilním telefonu stejně jako v osobním počítači, implementace speciálně pro danou platformu může zlepšit chování a vzhled nebo přidat offline funkcionality. Aplikace pro přední mobilní platformy iOS a Android jsou tedy podstatnou součástí každého komunikačního nástroje. Proto bude podpora různých platforem jedním z obecných kritérií.

#### 3.1.3 Grafické zpracování a UX

UX (user experience) neboli uživatelská přívětivost, představuje celkovou použitelnost aplikace. UX a grafické zpracování aplikace lze považovat za jedny z kritických charakteristik při jejím prodeji. Se skvěle vypadající a jednoduše ovladatelnou aplikací budou uživatelé pracovat raději a častěji.

Grafický návrh aplikace představuje vyloženě subjektivní kritérium. Pro vyhodnocování UX sice existují systematické a do jisté míry objektivní metody, jejich vypracování však překračuje rozsah této práce. Proto bude hodnocení vzhledu a použitelnosti provedeno společně dle subjektivního názoru autora.

#### 3.1.4 Možnosti zabezpečení a důvěrnost

Existují i komunikační nástroje, které si uživatel může nainstalovat na vlastní servery a spravovat sám. To lze považovat za výhodu nejen kvůli ceně (mohou být i zdarma), ale také bezpečnosti a důvěrnosti – některé podniky nechtějí sdílet data s poskytovatelem nebo preferují běh systému pouze uvnitř firemní sítě.

Pokud ale podnik spravuje aplikaci sám, musí se starat o její běh, škálování, řešit výpadky a další problémy. U bezplatných nástrojů může být také omezená podpora a vydávání nových verzí. Proto mnoho podniků preferuje zakoupení softwaru poskytovaného jako službu, kde nemusí kromě platby řešit nic jiného.

Nelze tedy obecně říct, že jedna varianta je lepší než druhá, záleží na postavení vedení společnosti, oboru ve kterém podniká, finanční situaci a také

jejich vnitřním politikám. Otázka bezpečnosti a důvěrnosti je ale rozhodně při výběru softwaru na místě.

### 3.1.5 Podpora funkcí, na kterých záleží

Na trhu se můžeme setkat i s komplexními nástroji, které podporují vývojový proces od prvního telefonátu klientovi až po proces udržování aplikace a vytváření updatů.

Pro menší společnosti lze v takových softwarech vidět klady i zápory. Vedení si může pochvalovat, že zavedením jednoho produktu podpoří celý vývojový proces. Na druhou stranu může být velké množství funkcí i nevýhodou. Společnost potom platí podporu funkcí, které často nepotřebuje, a uživatelé musí pracovat se složitým nástrojem, což jim ukrájuje čas z pracovní doby.

Zdali systém podporuje pouze funkce, na kterých záleží, lze proto také považovat za jedno z kritérií při výběru produktu. Všeobecně nelze říct, že čím více funkcí systém poskytuje, tím lépe. Uživatelé budou pracovat se třemi přímočarými nástroji raději, než s jedním, ve kterém se nevyznají.

## 3.2 Nástroje pro výměnu zpráv a dalších typů příloh

Prvním typem produktů jsou nástroje pro komunikaci pomocí výměny zpráv a různorodých typů příloh. V zásadě lze rozlišit dva typy tohoto software podle toho, zda dokáží zobrazovat informace druhé straně v reálném čase. Rychlá odezva je podstatným kritériem, proto se tato sekce zaměřuje zejména na nástroje pracující v reálném čase.

### 3.2.1 Nastavení kritérií

Kromě obecně definovaných kritérií lze na tuto kategorii nástrojů vymezit další pro a proti, které by firma měla před zavedením zvážit. Následující podsekce vyzdvihují ty nejpodstatnější.

#### 3.2.1.1 Omezení komunikace pouze na pracovní záležitosti

Posílání zpráv prostřednictvím různých chatů a IM během pracovní doby bývá často spojováno s vyřizováním osobních záležitostí. Jedním z podstatných kritérií proto bude schopnost omezit posílání zpráv pouze na rozsah definovaných uživatelů.

Díky tomu mohou i méně oddaní zaměstnanci získat pocit, že se nejedná o prostředek pro zábavnou komunikaci s přáteli, ale o nástroj pro podporu pracovního procesu.

### 3.2.1.2 Vyhledávání v historii

Členové vývojového týmu často řeší problémy opakovaně. Vyhledávání v předchozích konverzacích jim proto může ulehčit život.

Vyhledávat ale potřebují nejen v odesílaném textu, ale také v přílohách různých typů. Proto je ideální software, který umožňuje ukládání a následné vyhledávání všech odeslaných příloh.

U některých systémů lze pak historii využít jako sběrnici častých problémů a jejich řešení. Pokud programátor přijde na řešení základního problému, může prostým sdílením do vybraného kanálu upozornit ostatní, kteří budou tuto informaci moci v budoucnu využít.

### 3.2.1.3 Integrace s jinými nástroji

Při vývoji software pracují členové vývojového týmu s různými nástroji podporující vývojový proces. Může se jednat o sdílené repozitáře a virtuální úložiště, software pro kontakt se zákazníky, produkty pro řízení projektů a mnoho jiných.

Jejich integrace s interním komunikačním nástrojem představuje automatické hlášení o událostech v nástrojích třetích stran. Taková funkcionality může uživatelům výrazně zjednodušit přístup k datům z mnoha různých zdrojů, navíc údaje nemusí hledat, neboť jdou informace přímo k nim.

### 3.2.1.4 Omezení přístupu

Podporu transparentnosti komunikace ve firmě lze jistě považovat za podstatnou součást firemní politiky. Pokud ale nemluvíme o mikro podnikách, téměř vždy je potřeba vymezit přístup skupinám či jednotlivých osobám. Může to být z důvodu oprávnění vyplývajících z jejich role nebo pouze kvůli tomu, aby je komunikace ostatních nevyrušovala v jejich pracovní náplni.

Proto by měl software podporovat alespoň jednoduché role uživatelů a možnosti nastavení jejich oprávnění. U chatů a IM to lze částečně nahradit vytvářením privátních kanálů a nastavením přístupu pro jednotlivé členy.

Omezení přístupu lze také využít, pokud je kromě interních zaměstnanců potřeba přizvat ke komunikaci i externí uživatele, například klienty nebo prodejce. Těm lze pak vymezit možnosti komunikace pouze na vybrané skupiny zaměstnanců a zakázat přístup do kanálů pro vnitřní záležitosti.

### 3.2.1.5 Editace odeslaných zpráv

Při online komunikaci může jednoduše docházet k překlepům. Editace a případně mazání odeslaných zpráv je proto příjemnou součástí dobře navrženého uživatelského rozhraní.

### 3.2.2 Email

Email byl a ještě pravděpodobně nějakou dobu bude nejrozšířenějším nástrojem pro komunikaci po internetu. Emailovou adresu má totiž každý a schránku lze vybírat na všech možných typech zařízení, prostřednictvím webového prohlížeče, mobilní aplikace, nebo klidně přes příkazovou řádku.

#### 3.2.2.1 Možnosti zabezpečení

Zabezpečení a šifrování emailů bylo poslední dobou často diskutovaným tématem, zejména díky medializaci kauz týkajících se praktik vládních agentur.

Kvůli zvýšené poptávce po šifrovaných emailových službách vznikají projekty, které kódují veškerý digitální obsah svých zákazníků tak, že k němu nemá přístup ani samotný správce serveru. Jedním z takových je například švýcarský ProtonMail [25] [26].

Další možností je zabezpečení dat z vlastních zdrojů na emailovém serveru společnosti.

#### 3.2.2.2 Výhody emailu

Hlavní výhodou představuje jednoduchost a rozšířenost emailu. Právě kvůli tomu je stále používaný a v mnohých případech se jedná o nejlepší způsob komunikace – je běžná a dostupná, zprávy lze archivovat a odesílat všechny typy příloh.

Za další výhodu lze považovat širokou škálu emailových klientů a dalších podpůrných nástrojů, které během let vznikly, například nástroje pro třídění, filtrování, archivování, zabezpečení a mnoho dalších. Důležitou součástí jsou i klienti pro mobilní aplikace, díky kterým máme emailovou schránku dostupnou nepřetržitě.

#### 3.2.2.3 Nevýhody emailu

Hlavní nevýhodou emailů představuje zpoždění, se kterým pracují. Emailové servery si mezi sebou vyměňují zprávy různě rychle v závislosti na jejich umístění, hardwaru nebo vytíženosti. Ale i při komunikaci pod jedním poskytovatelem může výměna zprávy trvat až několik sekund.

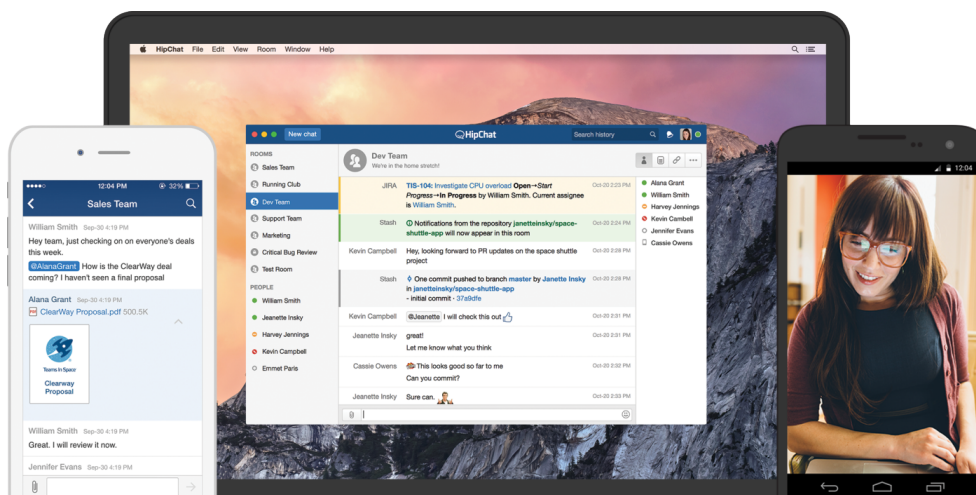
Takové zpoždění není kritické při odesílání denních reportů nebo nabídek klientovi. Při interaktivní komunikaci mezi dvěma nebo více osobami však zpoždění přibývá na významu.

Za další nevýhodu lze považovat špatnou přehlednost. Ta sice do jisté míry záleží na emailovém klientovi, s hromadnými konverzacemi se však i ty nejlepší vypořádat nedokáží.

Jednou z dalších potíží může být nevyžádaná pošta. Uživatel emailu obvykle potřebuje, aby byla jeho emailová adresa alespoň částečně veřejná. S tím přichází riziko nevyžádané pošty, která znepříjemňuje používání emailu.

### 3.2.3 HipChat

HipChat [27] je komunikační nástroj poprvé spuštěn již v roce 2010. Navržen je jako jednoduchá služba pro společnosti a týmy, která byla postavena přímo pro ně. V roce 2012 byl koupen firmou Atlassian [28], která vyvíjí i další produkty pro podporu kolaborace.



Obrázek 3.1: Hipchat, komunikační nástroj od společnosti Atlassian

Zdroj [27]

Software byl založen na myšlence jednoduchosti tehdejších soukromých chatů ve spojení s možností administrace dle firemních politik. Od té doby produkt dospěl ve vypracovaný nástroj. V roce 2014 byl například rozšířen o sdílení pracovní plochy a 1-1 video konverzace (v případě placené verze) [29].

Komunikace pomocí HipChatu nemusí probíhat pouze mezi přihlášenými uživateli. Do kanálů lze integrovat upozornění z téměř 70 služeb třetích stran. Příkladem jsou Git repozitáře GitHub a Bitbucket, CI servery Jenkins a Travis, systémy pro hlášení chyb Crashlytics a Bugsnag nebo systémy pro správu projektů Redmine a Trello.

Procházení historie zahrnuje vyhledávání dle klíčových slov z odeslaného textu, souborů, odkazů a dalších. Kapacita úložiště odeslaných souborů i počet dohledatelných zpráv jsou v bezplatné verzi omezeny.

Jednou z výhod HipChatu je serverová verze softwaru. Od ledna 2015 si mohou zákazníci zvolit, zdali chtějí produkt používat jako službu v cloudu, nebo aplikaci nainstalují na svém vlastním serveru [30]. Tím například mohou komunikaci omezit pouze na svou privátní síť.

### 3.2.3.1 Podpora platforem

V současnosti už HipChat podporuje všechny majoritní desktopové i mobilní platformy, tedy Windows, Mac, Linux, iOS a Android, viz. obrázek 3.1.

Aplikace pro iOS poskytuje odpovídající funkce a po grafické stránce vypadá pěkně. Propojení s jinými aplikacemi v telefonu umožňuje například odesílání příloh přímo z virtuálních úložišť, nebo proklikávání odkazů do webového prohlížeče a naopak. Vytknout lze pouze přehlednost, která by se dala vylepšit.

### 3.2.3.2 Cena

Software je možné používat zdarma s omezením na úložiště odesílaných souborů (5 GB) a maximální počet dohledatelných zpráv (posledních 25 000 odeslaných). To sice pro malé a střední podniky z dlouhodobého hlediska spíše nedostačuje, pro vyzkoušení základních funkcí ale poslouží.

Verze HipChat Plus obsahuje sdílení plochy, 1-1 audio-video hovory a neomezené úložiště pro soubory a celou historii zpráv. Stojí 2 USD za uživatele na měsíc, tedy v malé společnosti 60 USD / měsíc a ve středním podniku 300 USD / měsíc.

Za pronájem licence při instalaci na vlastní server začíná platební model na 10 USD / rok pro mikro týmy (do 10 členů). V malém podniku ale už stojí 3 300 USD / rok a ve střední firmě o 100 až 250 zaměstnancích přijde na 12 000 USD / rok. Cena dále pokračuje až k 72 000 USD / rok při 5 000 připojených uživatelů.

### 3.2.4 Slack

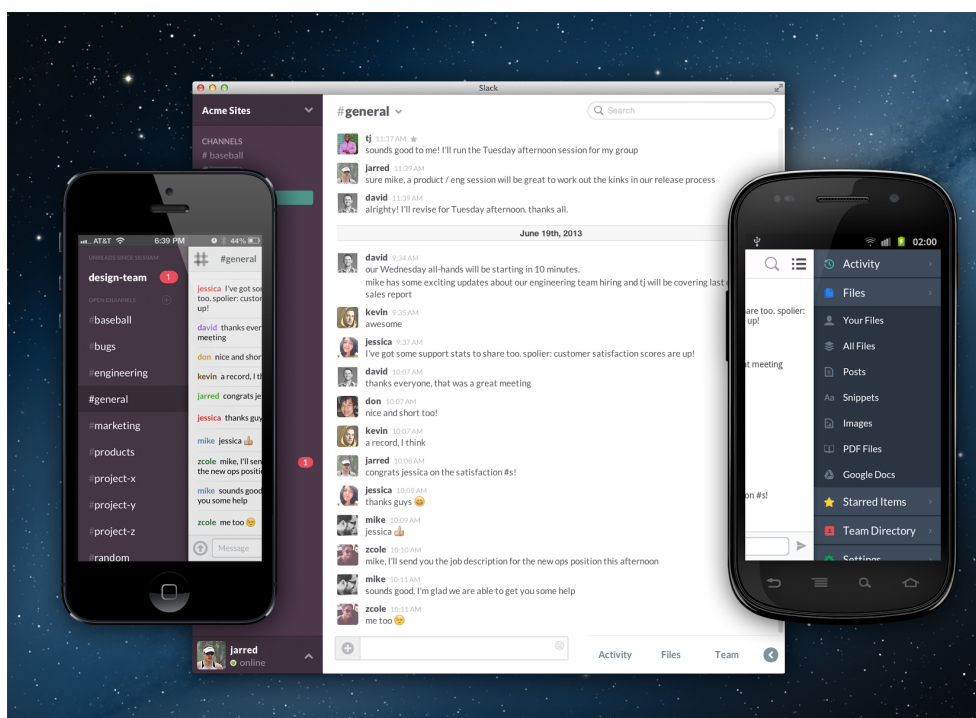
Slack [31] nabízí platformu pro komunikaci v reálném čase vhodnou pro týmy všech druhů. Byl spuštěn teprve v únoru roku 2014 a za první rok existence vyrostl na 500 000 uživatelů denně [32].

Podle některých uživatelů jeho kouzlo spočívá v jednoduchosti komunikovat, archivovat a vyhledávat nezávisle na zařízení. Jinými slovy Slack je ukázkou toho, jak moc záleží na dobře zpracovaném grafickém návrhu a UX.

Uživatelé ho preferují kvůli líbivému uživatelskému rozhraní a přímočarému ovládání. Celá aplikace se zdá být až neskutečně vyladěná na to, že je v provozu teprve rok. Podstatné jsou i drobné funkce jako editace/mazání odeslaných zpráv nebo vyhledávání přímo v poli pro odeslání zprávy pomocí příkazu `/s <hledaný řetězec>`.

Díky tomu mu někteří vedoucí pracovníci dají přednost před nástroji jako je HipChat, které poskytují stejný nebo dokonce větší rozsah funkcí za nižší cenu. Přejít na Slack také zjednodušuje možnost importovat data z původní chatovací služby.

## 3.2. Nástroje pro výměnu zpráv a dalších typů příloh



Obrázek 3.2: Slack, přehledný a jednoduchý IM

Zdroj [31]

Velký podíl na jeho vzestupu lze samozřejmě přičíst také skvělému marketingu a vyladěnému procesu získávání nových uživatelů – od jednoduchosti registrace až po začlenění propagace přes sociální sítě.

Velkou podporu mu dávají i investoři, kteří do něj během prvního roku od spuštění vložili okolo 180 mil. USD, z čehož poslední, největší investice proběhla na podzim roku 2014 – 120 mil. USD, tedy dle tehdejšího kurzu přes 2,4 mld. Kč [33] [34].

Slack, podobně jako HipChat, podporuje integraci s několika desítkami nástrojů. Mezi ně patří například virtuální disková úložiště Dropbox a Google Drive, nástroje pro správu projektů JIRA nebo Pivotal Tracker nebo služby pro analýzu a správu aplikací.

Zároveň existují aplikace pro všechny majoritní platformy. Mobilní verze pro iOS a Android jsou proti HipChatu přece jen o něco líbivější a uživatelsky jednodušší, viz. obrázek 3.2.

### 3.2.4.1 Přístupové oprávnění a správa rolí

I v administraci se Slack snaží být co nejjednodušší. Základní bezplatná verze umožňuje 3 základní role – vlastník, administrátor týmu, člen. Členové mohou vytvářet veřejné nebo privátní skupiny. Administrátor má k dispozici širší



nastavení včetně přizvání nových uživatelů. Výhodou je také možnost povolit automatický přístup všem uživatelům s emailem pod firemní doménou. Administrátor tak nemusí každého člena pozvat zvlášť.

Po zakoupení placené verze přibude možnost rozlišovat ještě členy s omezeným přístupem. To umožní jednoduchý způsob, jak do chatu bez problémů přizvat klienty a další externí osoby.

### 3.2.4.2 Cena

Bezplatná verze má omezenou historii na posledních 10 000 odeslaných zpráv, lze vybrat pouze 5 integrovatelných služeb a nelze přidávat členy s omezeným přístupem.

Slack Standard je nejčastěji používanou verzí, která kromě odstranění omezení integrací a historie přidává možnosti autentizace pomocí Google Authentication, prioritní zákaznickou podporu a další. Cena je uváděna 6,67 USD na uživatele za měsíc, tedy v malé společnosti (30 členů) 200 USD / měsíc a ve středním podniku (150 zaměstnanců) 1000 USD / měsíc. Tato verze zároveň přidává zajištění dostupnosti služby definované pomocí smlouvy – SLA (Service Level Agreement).

Aktuálně Slack nabízí také verzi Plus a do budoucna připravuje ještě Enterprise s cenami 12,5 a 50-99 USD na uživatele za měsíc. Tyto verze budou mít rozšířené statistické a analytické nástroje, možnosti správy vnitřních bezpečnostních pravidel a politik a mnoho dalšího.

### 3.2.5 Google Hangouts

Google Hangouts [35] představuje platformu pro posílání zpráv v reálném čase a přenos videa od společnosti Google. Služba spojuje 3 původní produkty – Google Talk, Google+ Messenger (Huddle) a Hangouts, systém pro video konference.

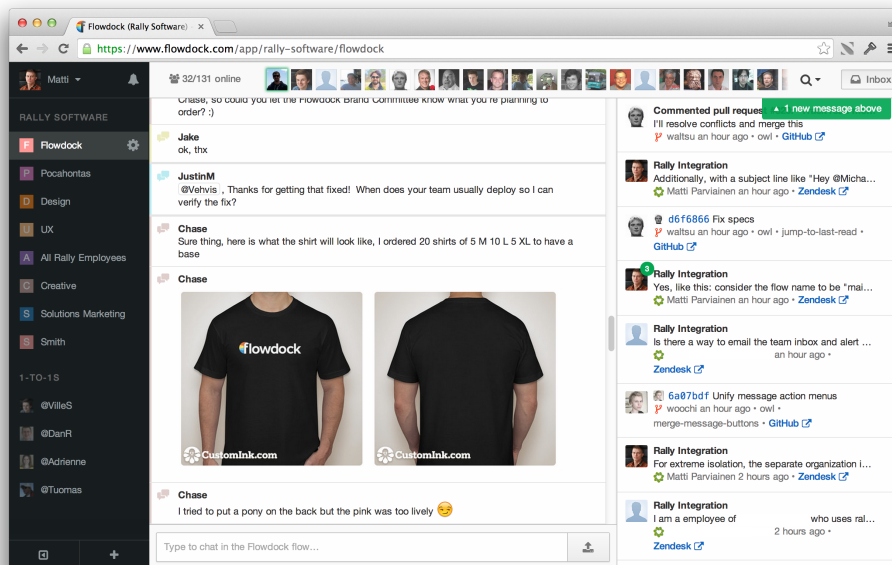
Za jednu z hlavních výhod lze považovat cenu. Google Hangouts je totiž zdarma pro všechny, kdo mají emailový účet vedený pod Google. Pokud tedy firma používá služby Google Apps for Work (zahrnující mimo jiné emaily pod svou vlastní doménou), zaměstnanci se mohou přihlásit i prostřednictvím firemních emailových účtů.

Služba však není mířená pouze na podniky. Každý uživatel tedy může Google Hangouts používat podle svého uvážení, psát zprávy a volat i mimo zájmy společnosti.

Google aktivně vyvíjí aplikace pro desktopové i mobilní platformy včetně vlastního operačního systému Chrome OS. Aplikace pro iOS funguje spolehlivě a ovládání je přímočaré. Grafická podoba se ale až příliš podobá Androidové verzi, proto se tak trochu zdá, jakoby aplikace do systému nepatřila.

### 3.2.6 Flowdock

Aplikace Flowdock [36] představuje nástroj pro týmovou komunikaci. Založen byl v roce 2009 ve finských Helsinkách a od té doby získal několik investic, které vyvrcholili akvizicí společností Rally Software zabývající se produkty pro podporu procesů při vývoji software [37].



Obrázek 3.3: Flowdock, týmový komunikační nástroj

Zdroj [36]

Jednou z předností Flowdocku je, že podporuje integraci s opravdu velkým množstvím nástrojů. Mezi ně patří například přes 20 produktů pro správu projektů (Basecamp, JIRA, Trello a další), 15 různých CI serverů, verzovací systémy, monitoring a nasazování aplikací a mnoho dalších. Zprávy lze do chatu posílat také přes speciální emailové adresy. Široká škála možných integrací spolu s přehledným panelem nedávných událostí (na obrázku 3.3 vpravo) umožňuje jednoduchý náhled na aktuální vývoj produktu.

Software dovoluje editaci a mazání zpráv po jejich odeslání. Užitečnou funkcí je také možnost označovat zprávy pomocí štítků. Vyhledávání ve starých zprávách, informacích z integrovaných služeb nebo štítcích nabízí jednoduché dohledání důležitých milníků a aktivit.

Posílání zpráv je založeno na kanálech neboli „flow“, do kterých lze přiřvat všechny nebo pouze vybrané uživatele. V administraci organizace je poté možné nastavit, zdali budou členové moci komunikovat se všemi ostatními, nebo pouze s těmi, se kterými sdílí alespoň jeden kanál. Správným nastá-

vením lze tedy přizvat také klienty a povolit jim pouze komunikaci uvnitř vybraných kanálů a s vybranými členy organizace.

Flowdock podporuje všechny majoritní platformy, tedy Windows, Mac, Linux, iOS a Android.

### 3.2.6.1 Cena

Pro týmy do 5 členů, neziskové organizace a studenty lze Flowdock používat zdarma. Pro ostatní firma nabízí bezplatnou 30-denní zkušební verzi.

Po uplynutí zkušební verze lze Flowdock používat ve verzi Standard za 3 USD za uživatele na měsíc, tedy v malé společnosti 90 USD / měsíc a ve středním podniku 450 USD / měsíc. V ceně jsou pochopitelně zahrnuty neomezené zprávy, historie i počty integrací.

Pro náročnější lze zakoupit verzi Enterprise doplněnou o přihlašování pomocí SSO (Single sign-on), zaručení podmínek poskytování služby pomocí SLA nebo rozšířených možností práce s daty uvnitř aplikace. Ta stojí 9 USD za uživatele na měsíc.

### 3.2.7 Skype

V některých společnostech používají pro výměnu zpráv službu Skype [38]. Ta je známá především kvůli funkcím audio a video hovorů, výměnu zpráv a příloh však zvládá podobně jako jiné nástroje a to na všech typech zařízení.

Skype ale není vyvíjený přímo pro komunikaci uvnitř společnosti. To představuje problém, neboť jej uživatel může používat jak chce, firma nemůže aplikovat jakékoliv firemní politiky a omezit přístup.

Předností Skype jsou video hovory, proto se o tomto nástroji zmiňuje sekce o softwaru pro videokonference. Pokud ale firma hledá jednoduché a levné řešení pro posílání zpráv a příloh, bezplatný Skype je jednou z možností.

### 3.2.8 Sociální sítě pro podniky

Jednou z dalších možností, jak si vyměňovat zprávy a přílohy ve společnosti, jsou sociální sítě pro podniky. Tyto nástroje jsou založeny na stejných principech jako veřejné sociální sítě Facebook nebo Twitter, rozdíl je pouze v tom, že jsou připojení uživatelé pracovně propojeni.

Funkce se potom liší podle poskytovatele, některé mimo komunikaci zahrnují i mnoho dalších nástrojů například podporu pro řízení projektů, CRM a další.

Jedním příkladem je Yammer (spadající pod Microsoft), který používá přes 200 000 společností po celém světě [39]. Mezi sociální sítě pro podniky se také řadí IBM Connections, Salesforce Chatter nebo Jive.

Hlavním problémem takových systémů může být snižování produktivity. Pozornost zaměstnanců mohou odvádět příspěvky ostatních a z nástroje pro podporu komunikace se může stát rušivý element.

### 3.2.9 Shrnutí

Na trhu lze nalézt různorodé nástroje pro výměnu zpráv a dalších typů příloh. Nejběžnějším nástrojem je email, který ale není příliš efektivní při interaktivní konverzaci více osob.

Pro takové situace je vhodné vybrat nějaký nástroj pro výměnu zpráv v reálném čase. V nejjednodušším případě lze využívat některé bezplatné služby dostupné komukoliv, jako jsou Skype, Google Hangouts nebo Yahoo Messenger.

Pokud si firma platí Google Apps for Work a používá tamní firemní emaily, komunikaci přes Google Hangouts lze standardizovat ve smyslu propojení emailových účtů všech zaměstnanců.

#### 3.2.9.1 Specializované nástroje

Pokud má společnost ale potřebu komunikaci více řídit a zvat klienty nebo externisty do konverzací, je na místě vybrat specializovaný nástroj přímo pro firmy.

Hlavní výhodou je omezení skupiny uživatelů pouze na osoby zainteresované v dané společnosti. Dále jsou to také integrace nejrůznějších nástrojů pro podporu vývojového procesu. Software se pak může stát komplexní informační sběrnici napříč celou firmou.

I v případě specializovaných nástrojů lze nalézt bezplatné verze aplikací, ty jsou ale většinou omezeny buď počtem zpráv, uložištěm souborů, počtem integrací atd.

Mezi nejzajímavější patří zmíněný Slack, HipChat a Flowdock, ale také Hall, Kato nebo Campfire. Při výběru záleží, které parametry jsou pro společnost klíčové. Může to být cena, možnosti integrace, design a jednoduchost, možnosti analýzy dat a další. Cena se u těchto nástrojů pohybuje obvykle v jednotkách USD za uživatele měsíčně.

#### 3.2.9.2 Služba v cloudu nebo na privátním serveru

Většina komunikačních nástrojů je poskytována jako služba. Pravděpodobně je tomu tak proto, že většina společností není tak skeptická a nebojí se krádeže dat.

Pokud ale z nějakých důvodů vyžaduje instalaci nástroje na vlastní server, jednou ze zajímavých, ale vcelku drahých možností je HipChat server.

#### 3.2.9.3 Chat jako součást jiného nástroje

Jednou možností výměny zpráv mezi zaměstnanci je také chat jako součást nějakého většího nástroje. Vedení by si ale nemělo volit komplexní produkt na základě dobrého chatu. Podstatné je, zdali potřebuje funkce, které nabízí.

Příkladem tohoto typu softwaru jsou zmíněné sociální sítě Yammer, IBM Connections, Salesforce Chatter nebo Jive. U těch je ale potřeba standardizovat používání tak, aby nedocházelo k příliš častému užívání softwaru, což může mít za následek snižování produktivity.

## 3.3 Software pro videokonference

Ačkoliv lze osobní konverzaci považovat v mnoha ohledech za nejefektivnější, zaměstnanci a ostatní osoby se nemusí vždy nacházet ve stejné lokaci. Cestování za schůzkami však často nepřipadá v úvahu z finančních nebo organizačních důvodů, setkání totiž mohou být příliš pravidelná nebo mohou být účastníci nadměrně vzdálení.

Řešením jsou videokonferenční nástroje, které umožní osobní nebo skupinové porady přes internet. Zároveň je lze využít i pro online přednášky nebo školení.

### 3.3.1 Kritéria pro výběr nástroje

Při výběru lze kromě ceny, podpory platform, UX a dalších obecných požadavků zohlednit i jiná kritéria.

Hodně však záleží, jakým způsobem chtějí zaměstnanci nástroj používat. V některých firmách jej využijí spíše nepravidelně ke konverzacím mezi 2-3 lidmi. V jiných však může být online video konference pravidelným způsobem dorozumívání.

Jedním z podstatných požadavků bude ale také úroveň kvality přenášeného zvuku a obrazu. Problémy mohou nastávat zejména při konferenci s větším počtem účastníků.

#### 3.3.1.1 Počet připojených uživatelů

Konference mohou být různě velké, od osobní konverzace mezi dvěma lidmi až po hromadné online porady pro celé oddělení.

Pokud bude například vedení organizovat pravidelné celopodnikové porady, v distribuovaném prostředí může mít taková konference i několik desítek účastníků. Při výběru nástroje lze tedy zohlednit i tento faktor.

U některých produktů se také může cena odvíjet od počtu hostitelů, tedy uživatelských účtů, ze kterých je možné videokonferenci založit.

#### 3.3.1.2 Sdílení pracovní plochy a jiné nástroje

Užitečnou funkcí je sdílení pracovní plochy. Pokud jsou účastníci porady ve stejné místnosti, používají různé pomůcky pro znázornění svých myšlenek. Příkladem jsou bílé tabule s fixami, tablety nebo prezentace pomocí projektoru.

Při online konverzaci lze tyto pomůcky alespoň částečně nahradit sdílením pracovní plochy, kdy se namísto obrazu z kamery přenáší jakýkoliv program spuštěný na vybrané ploše. Vybraný nástroj by měl rozhodně podporovat tuto funkcionalitu.

Některé pokročilejší nástroje pak mohou mít přímo zabudované možnosti pro sdílení prezentace, nebo bílé plátno představující tabuli na psaní a nákresy.

#### 3.3.2 Skype

Aplikace Skype [38] představuje jednu z prvních a nejznámějších služeb pro audio a video hovory. Za více než 11 let své existence se dostal do domácností i kanceláří a pravděpodobně má i nadále velkou budoucnost. Toto tvrzení podporuje i akvizice společností Microsoft, která za Skype dala v roce 2011 8,5 mld. USD [40].



Obrázek 3.4: Skype podporuje prakticky všechny zařízení s připojením k internetu

Zdroj [38]

Jednou z velkých výhod Skype je podpora prakticky všech zařízení, které mají připojení k internetu viz. obrázek 3.4. U chytrých telefonů a tabletů existují kromě Android a iOS také aplikace pro Windows Phone a Blackberry. Mimo desktopových a mobilních platforem lze Skype spustit i například na chytrých televizích nebo na herních konzolích Xbox One a PlayStation Vita.

Skype je velice oblíbený pro hovory mezi přáteli a s rodinou. I to je důvod jeho velké rozšířenosti. Svou cestu si ale našel i do společností zabývajících se vývojem software.

### 3.3.2.1 Omezení audio a video přenosů

Při video konferenci je maximální počet připojených osob 10, na oficiálním webu produktu se lze ovšem dočíst, že sám Skype doporučuje maximálně 5 účastníků pro zajištění dobré kvality přenosu. Pro konference bez přenosu obrazu lze připojit až 25 lidí.

Omezení přenosu zvuku a obrazu jsou jednou z hlavních nevýhod Skype, neboť při připojení více účastníků opravdu naráží s kvalitou přenášeného zvuku a hlasu. Navíc ani maximální počet účastníků není proti jiným placeným produktům nějak výrazný.

### 3.3.3 Google hangouts

Jednou z atraktivních možností je také Google Hangouts [35] zmiňovaný v předchozí sekci. Zdarma totiž nabízí videokonference až pro 15 připojených uživatelů.

Službu lze spustit přímo v prohlížeči, což může být často výhodou. Zároveň může hostitel přizvat uživatele pouhým sdílením odkazu na daný přenos. V malých a středních podnicích se Google Hangouts hodí pro menší meetingy například na úrovni vývojových týmů.

Výhodou jsou také integrace s různými službami od společnosti Google. Kromě sdílení pracovní plochy lze při videohovoru také přímo spolupracovat pomocí Google Docs, přehrávat videa z Youtube, nebo sdílet prezentace.

Jednou z drobností, na kterou si někteří zákazníci stěžují, je nepřehledné uživatelské rozhraní webové aplikace. Ta sice na první pohled vypadá jednoduše a příjemně, nalezení požadované funkce může ale trvat delší dobu i po opakovaném používání.

#### 3.3.3.1 Porovnání se službou Skype

Oba nástroje jsou zdarma a podporují hlavní desktopové i mobilní platformy. V oblasti podpory chytrých televizí a dalších zařízení má však náskok Skype.

Ve prospěch Google Hangouts hraje vyšší počet maximálních připojených zařízení a také možnost nastavení konference a sdílení plochy před začátkem hovoru. Zároveň si hostitel přenosu nemusí vyhledávat účty jednotlivých účastníků, aby je mohl přizvat ke konverzaci, stačí jim poslat odkaz na webovou stránku pro daný přenos.

### 3.3.4 AnyMeeting

AnyMeeting [41] je jednou z dalších možností pro online meetingy. Proti předchozím službám umožňuje připojení až 200 uživatelů do jednoho přenosu.

Jednou z výhod jsou různé integrace s dalšími aplikacemi. Příkladem je možnost přehrávání Youtube videa přímo ve videohovoru, nebo přizvání účast-

níků pomocí Google Apps integrace, Facebooku nebo Twitteru. Dále je samozřejmou součástí tohoto nástroje sdílení pracovní plochy.

Užitečnou funkcí je také nahrávání konferencí, proto ve firmách může posloužit při hromadných poradách nebo online školeních.

Podpora mobilních zařízení je ale bohužel slabší. Verze pro iOS i Android nejsou už delší dobu aktualizované a zakládat meetingy lze pouze z aplikací pro stolní počítače.

#### 3.3.4.1 Platební model

Bezplatná verze poskytuje většinu podstatných funkcí kromě nahrávání přenosu a na rozdíl od té placené obsahuje reklamy.

AnyMeeting Pro lze zakoupit ve dvou verzích podle maximálního počtu připojených účastníků – Pro 25 a Pro 200 za 18 respektive 78 USD měsíčně. Při zakoupení více účtů najednou lze získat různé slevy.

#### 3.3.5 Adobe Connect

Adobe Connect [42] představuje jednu z možností pro profesionální online přednášky a videokonference od softwarového gigantu Adobe.

Přenos obrazu probíhá ve vysoké kvalitě a samozřejmostí produktu je také podpora majoritních desktopových platforem v podobě zásuvného modulu prohlížeče. V případě telefonů a tabletů nechybí plnohodnotná aplikace pro iOS, Android a Blackberry.

##### 3.3.5.1 Široká škála funkcí

Hlavní výhodou je vysoká úroveň zabezpečení a podpora nejrůznějších funkcí jako jsou pokročilé nástroje pro záznam a úpravy videa nebo administrační prvky. Proto se tento nástroj hodí zejména pro větší meetingy, přednášky nebo školení.

Mnoho různých funkcí lze zároveň považovat za hlavní nevýhodu tohoto nástroje. Po registraci může být prvotní nastavení a inicializace softwaru trochu složitá až odstrašující. Existuje ovšem řada videí a psaných návodů pro základní orientaci v softwaru.

##### 3.3.5.2 Cena

Levnější verze Adobe Connect Meetings limituje maximálním počet připojených účastníků na 25 a stojí od 55 USD měsíčně za každý účet, který může meetingy vytvářet.

Druhou možností je Adobe Connect Webinars s možností připojení 100, 500 nebo 1000 účastníků. Tyto verze už stojí podstatně více tedy 104, 375 a 458 USD měsíčně za hostitelský účet. Tato verze navíc obsahuje mnoho analytických nástrojů a široké možnosti dalšího nastavení.



### 3.3.6 Shrnutí

Na trhu existuje mnoho nástrojů s různě pokročilými funkcemi. Podnik by si měl vybrat podle toho, jakým způsobem chce službu používat.

Pro příležitostní videohovory lze najít i jednoduchá řešení zdarma. Pokud se však jedná o podnik založený na distribuovaných týmech, funkce pokročilejšího nástroje mohou přijít vhod.

#### 3.3.6.1 Cena vs. možnosti využití

Pro osobní konverzace lze využít jeden z bezplatných nástrojů, například Skype nebo také služby typu Facetime zabudované ve všech Apple produktech. Jednoduché videohovory jsou také součástí některých chatovacích aplikací, například HipChat zmíněný v předchozí sekci.

Při poradách malých a středních týmů už je nutné použít jiný produkt. K takovému účelu je zde zdarma kromě Google Hangouts například appear.in, který umožňuje konverzaci přímo v prohlížeči bez nutnosti registrace (bezplatně až pro 8 lidí).

Pro celopodnikové porady nebo online školení je na místě zvolit nějaký placený nástroj, který podporuje až stovky připojených uživatelů. Cena se často určuje podle počtu hostitelských účtů, které stojí obyčejně v řádu desítek USD měsíčně. Jednou možností je například zmíněný AnyMeeting, ale také GoToMeeting, WebEx, Join.Me a mnoho dalších.

Pro náročnější uživatele lze najít programy jako je Adobe Connect, který poskytuje širokou škálu funkcí, pokročilé možnosti administrace nebo nahrávání přenosu. Připojit se pak může až 1000 účastníků, cena je ovšem v řádu stovek USD měsíčně. Využitelnost takových nástrojů ve středních a menších podnicích pro vývoj software je však diskutabilní.

#### 3.3.6.2 Sdílení pracovní plochy a další funkce

Sdílení pracovní plochy lze považovat za jednu ze základních funkcionalit a podporují ji všechny zkoumané produkty. U některých nástrojů je možné najít i další užitečné funkce, jako jsou integrace s přehrávači videí, přímé zobrazení prezentací nebo nahrávání konferencí.

## 3.4 Nástroje pro podporu řízení projektů

Nástroje pro řízení projektů jsou navrženy k celkovému zefektivnění vývojového procesu a ke zjednodušení komunikace mezi projektovými manažery a členy vývojového týmu. Za hlavní funkce lze považovat předávání informací ohledně termínů, správu dílčích úkolů a obecně zaznamenávání stavu několika projektů najednou.

### 3.4.1 Kritéria při výběru nástroje

Při výběru nástroje pro správu projektů lze hledět na mnoho vlastností. Kromě ceny a dalších obecných kritérií je nejpodstatnější podpora různých funkcí a uživatelská přívětivost aplikace.

#### 3.4.1.1 Podpora koordinace při řízení úkolů

Vývoj softwaru lze dělit na iterace a jednotlivé úkoly. Jednou ze základních funkcí těchto nástrojů je podpora kooperace na úkolech.

V nejjednodušším případě lze úkoly pouze pojmenovat. Za základní funkcionalitu lze také považovat přikládání předloh implementovaných částí softwaru. Další užitečnou funkcí je možnost komentovat průběžný průběh vývoje nebo zaznamenávání odpracovaného času k jednotlivým úlohám.

#### 3.4.1.2 Plánování a sledování průběhu projektu

Při spolupráci na vývoji software je také důležité uspořádání vývojového procesu. Nástroje pro řízení projektů mohou podporovat naplánování průběhu pomocí funkcí pro vytváření kalendářů a přiřazování událostí v čase. Díky tomu lze zaznamenat rozvržení projektu dle různých milníků, případně dle implementace funkcí produktu či systémových modulů.

Během vývoje pak uživatelé vyplňují aktuální stav jejich práce a tím aktualizují procentuální stavy úkolů. Pro grafické znázornění plánovaných událostí v čase může sloužit například Ganttův graf.

#### 3.4.1.3 Další funkce

Aplikace pro řízení softwarových projektů často podporují i mnoho dalších funkcí. Při vývoji softwarového produktu jsou například užitečné integrace s repositáři, systémy pro zpětnou vazbu nebo jinými nástroji pro vývoj produktu. Dalšími mohou být také zabudované funkce pro podporu agilního vývoje, správa iterací či sprintů apod.

#### 3.4.1.4 Složitost používání nástroje

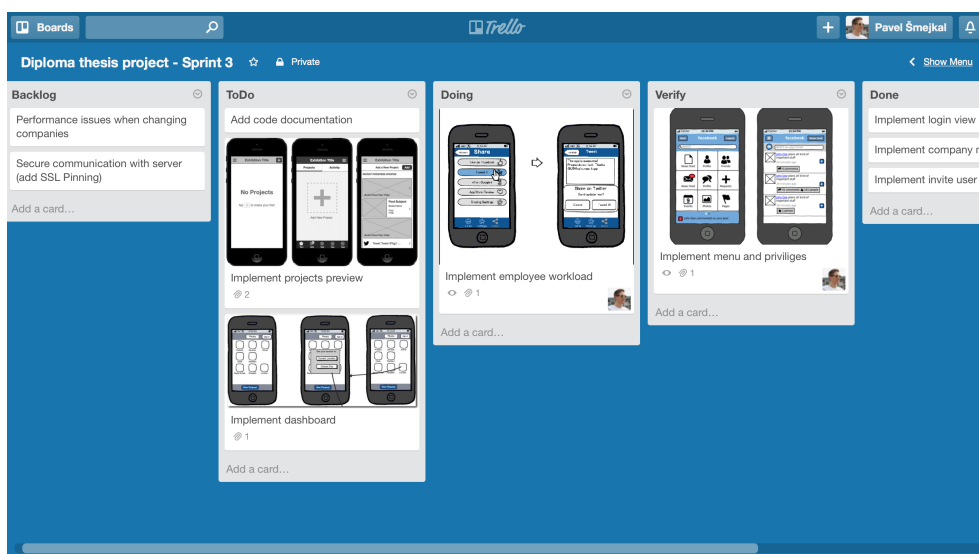
Nástroj musí hlavně sloužit svému původnímu účelu, tedy pozitivně přispívat k vyšší efektivitě při řízení a spolupráci na projektu. To ale nutně neznamená, že musí podporovat široké spektrum funkcí. Pro některé projekty se totiž může hodit spíše přímočarý a přehledný nástroj.

Jak komplexní nebo jednoduchý nástroj je vhodné použít se často odvíjí od počtu lidí spolupracujících na projektu, doby trvání nebo metodiky řízení. Například pro dvouměsíční projekt, na kterém pracují 2-3 lidé, je podstatná jednoduchost při práci se základními funkcemi pro správu úkolů. Proti tomu dlouhodobý vývoj analytického nástroje pro banku nebo pojišťovnu vyžaduje

podporu práv a rolí, zaznamenávání času, plánování milníků nebo zabudované hlášení chyb od zákazníků.

### 3.4.2 Trello

Trello [43] představuje jeden z nejjednodušších nástrojů pro řízení vývoje softwarových produktů. Celá správa projektu je založena na kartách reprezentujících úkoly, kde každá z nich obsahuje přiřazené uživatele, komentáře, přílohy a další. Všechny související změny se zaznamenávají do proudu událostí karty. Ostatní funkce pro řízení projektu pak souvisí zejména s organizováním a přesouváním těchto karet.



Obrázek 3.5: Použití nástroje Trello pro řízení projektu dle metodiky SCRUM

Množiny úkolů lze uspořádat do tzv. nástěnek (z anglického *board*), uvnitř kterých dělíme karty do vertikálních seznamů. Vhodným rozložením nástěnek, seznamů a karet lze Trello použít pro jednoduchou správu projektu dle metodiky SCRUM. Pro každý sprint je možné vytvořit speciální nástěnku s různými seznamy pro kategorizaci úkolů (obrázek 3.5).

Na přehlednost celého systému ještě přidává výsuvné menu s nedávnými aktivitami dané nástěnky. Samozřejmostí produktu jsou také upozornění sledovaných úkolů. Užitečným detailem je i možnost odpovídat na notifikace přímo pomocí emailu, tedy bez nutnosti přihlášení do služby.

Uživatelská přívětivost a přímočarost Trelly jsou jeho přednostmi, a proto se také hodí pro menší projekty nebo týmy, kde se musí soustředit maximum pracovní kapacity na vývoj namísto řízení. Jednoduchost je zároveň i jeho hlavní nevýhodou, protože neobsahuje například možnosti plánování a zaznamenávání času nebo další pokročilé funkce.

Produkt je možné spustit v aplikacích pro iOS, Android, Windows 8 a Kindle Fire. Na stolních počítačích musí postačit webová aplikace, neboť specializované aplikace pro desktopové platformy chybí.

### 3.4.2.1 Cena

Velkou výhodou Trelly je také bezplatná verze. Tu lze používat pro neomezený počet uživatelů i projektů a obsahuje veškeré funkce.

Placená verze Trello Gold přidává za 5 USD měsíčně několik užitečných drobností, například možnost nahrávání příloh o velikosti až 250 MB nebo rozšířené nastavení pracovní plochy a ikonek v komentářích.

Pro podnik lze také zakoupit Trello Business Class, který zajistí všem uživatelům Gold verzi a přidá možnosti administrace, integraci s Google Apps účty nebo deaktivaci uživatelů bez ztráty jejich příspěvků. Rozšířená administrace umožní přiřazení práv pro vytváření/editaci/mazání karet – v bezplatné verzi mohou provádět tyto úkony všichni uživatelé.

### 3.4.3 Basecamp

Nástroj Basecamp [44] reprezentuje produkty pro řízení vývoje software, které poskytují mírně pokročilé funkce při snaze zachování jednoduchosti a uživatelské přívětivosti.

Základ podpory kooperace u jednotlivých projektů představuje vytváření diskuzí a zadávání nebo aktualizace stavů úkolů. Každý projekt má svůj vlastní kalendář termínů a událostí a uživatelé také mohou používat celopodnikový kalendář nebo vytvářet své vlastní.

Doplňky třetích stran pak umožňují různé další funkce, příkladem mohou být zaznamenávání odpracovaného času, odhady nákladů, podpora účetnictví, hlášení chyb, Ganttovi grafy a další.

Součástí produktu je webová aplikace, existuje ale také podpora mobilních platforem iOS a Android v podobě specializovaných aplikací. Ty vypadají velice elegantně a proti jiným aplikacím podobného druhu působí jednoduše a přehledně.

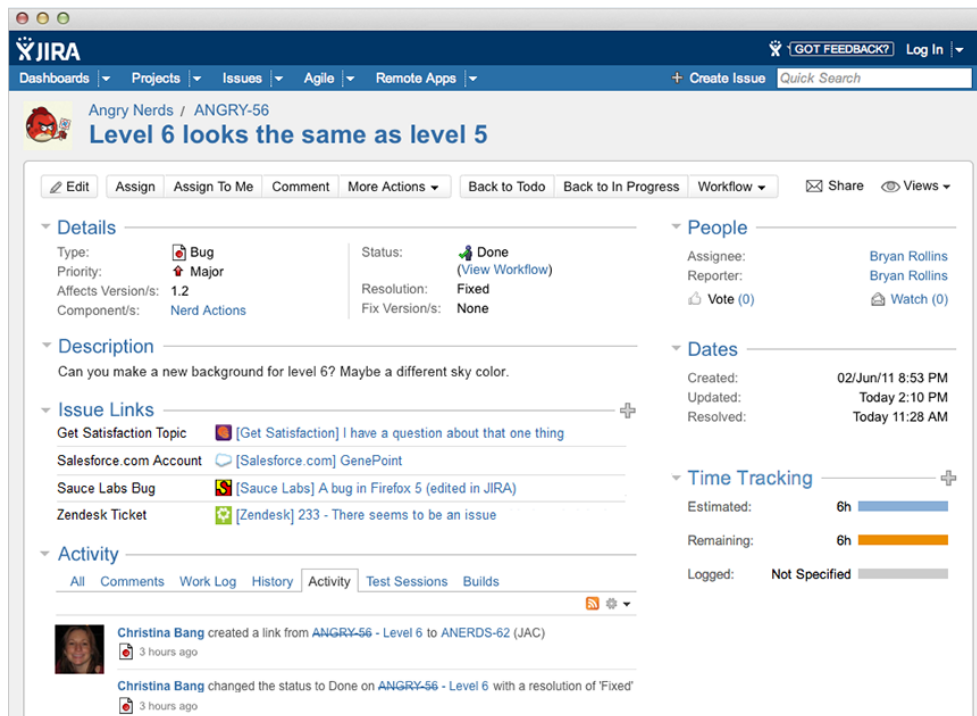
#### 3.4.3.1 Cena

Cena služby Basecamp se pohybuje v rozmezí 20 a 150 USD měsíčně podle omezení na počet projektů (150 USD = neomezený počet).

Produkt je vzhledem k ceně, jednoduchosti a rozsahu funkcí vhodný zejména pro menší týmy. Má uživatelský přívětivý rozhraní snadné pro řízení projektů, domlouvání schůzek nebo termínů a komunikaci ohledně vývoje.

### 3.4.4 JIRA

Společnost Atlassian provozuje kromě HipChatu zmíněného v předchozí sekci také produkt JIRA [45], který se zaměřuje na spolupráci při řízení projektů. Proti nástrojům Trello a Basecamp poskytuje podstatně širší spektrum funkcí nejen pro správu úkolů, ale také plánování, modelování pracovních postupů, přímou podporu agilních metodik nebo mnoho typů integrací.



Obrázek 3.6: JIRA představuje komplexní nástroj pro řízení projektů

Zdroj [45]

Webová aplikace je optimalizována pro zobrazení na mobilních zařízeních. Zároveň existují různé verze specializovaných aplikací pro telefony i tablety.

#### 3.4.4.1 Integrace s repozitáři verzovacích systémů

JIRA podporuje integraci s různými verzovacími systémy na hluboké úrovni a umožňuje přímé přechody do aplikací pro správu repozitářů.

Ve webové aplikaci lze v detailu úlohy jednoduše zakládat nové větve, procházet commity nebo vytvářet a spravovat pull requesty. Zároveň je možné nahlížet do obsahu repozitáře a komentovat provedené změny.

### 3.4.4.2 JIRA Agile

JIRA Agile je rozšíření pro přímou podporu agilních metodik. Při vytváření projektu lze zvolit speciální typy projektu pro agilní vývoj typu Scrum a Kanban.

Projekt typu Scrum nabízí intuitivní prostředí pro přiřazování priorit a sestavování sprintů. V nástěnkách jednotlivých sprintů pak lze interaktivně zaznamenávat stav dílčích úkolů a přiřazovat zodpovědnosti podobně jako v Trello. Výkon v jednotlivých sprintech lze průběžně i zpětně sledovat v grafech.

### 3.4.4.3 Cena

Pro vývoj softwaru se hodí spíše JIRA s integrovaným rozšířením pro agilní vývoj. Její cloudová verze stojí od v řádu stovek USD měsíčně podle počtu uživatelů. Pokud firma vyžaduje instalaci na vlastní servery, za licenci zaplatí v řádu tisíců USD opět podle počtu uživatelů.

Cena ale zdá se odpovídat komplexitě celého systému. Otázkou je, zdali daná menší či střední firma využije všechny tyto funkce.

### 3.4.5 Redmine

Jedním z potenciálních nástrojů pro podporu řízení projektů je bezplatné open source řešení Redmine [46].

Je možné jej spustit na jakémkoliv serveru, pro který existuje implementace Ruby. To lze považovat za výhodu, protože jeho používání lze omezit pouze na privátní síť a tím snížit některá bezpečnostní rizika. Na druhou stranu může být nevýhodné zaměřovat podnikové zdroje z produkce softwaru pro klienty do správy takových nástrojů.

Díky otevřenosti kódu je také možné aplikaci do jisté míry přizpůsobit dle potřeb podniku.

#### 3.4.5.1 Funkce

Redmine kromě správy úkolů podporuje zaznamenávání času stráveného na jednotlivých úlohách, vytváření diskuzních fór nebo nastavení přístupu na základě definovaných rolí. Pro sdílení obecných informací a dalšího textového obsahu poslouží wiki web. Užitečné jsou také roadmapy a přímá integrace s repositáři, díky kterým lze přehledně sledovat vývoj projektu.

Software je v neustálém vývoji už od roku 2006 a během té doby vznikla řada doplňků a rozšíření. Mezi ně patří například integrace s vývojovými prostředími Netbeans, Eclipse a Visual Studio nebo zásuvné moduly webových prohlížečů Firefox a Chrome.

### 3.4.5.2 Podpora platforem

Základem programu je webová aplikace založená na frameworku Ruby on Rails. Práci v prohlížeči na desktopových platformách ale nic nechybí, a pravděpodobně proto nebyl vyvinut v průběhu let žádný klient specializovaný přímo pro Windows, Mac nebo Linux.

V komunitě uživatelů ale vznikly klientské aplikace pro mobilní platformy iOS, Android a Windows Phone. Většinu z nich lze stáhnout zdarma a poskytují všechny základní funkce systému.

### 3.4.6 Shrnutí

Na trhu lze najít různorodé nástroje od těch nejjednodušších po komplexní. U všech je ale řízení projektů založeno na organizaci práce do úkolů. Při výběru nástroje by měl podnik zvážit zejména poměr ceny, funkcí a přehlednosti či uživatelské přívětivosti.

Podporu mobilních platforem lze najít nejen u placených služeb, ale také u menších bezplatných či komunitních produktů.

#### 3.4.6.1 Poskytované funkce a složitost aplikace

Obecně lze říci, že čím větší rozsah funkcionalit produkt podporuje, tím nepřehlednější a složitější se práce s ním stává. U komplexních nástrojů může být při zavádění takového systému dokonce nutné zaškolit zaměstnance nebo upravit stávající vývojový proces.

Proto je nutné zvážit přínosy všech funkcí a rozhodnout, zdali pro daný podnik nevyhovuje spíše jednodušší služba. Pro menší projekty nebo týmy lze použít nástroje jako jsou Trello nebo Insightly. Při kooperaci ve větším týmu nebo na dlouhodobých projektech se využijí produkty poskytující více pokročilejší funkcí pro celkovou organizaci vývojových procesů.

#### 3.4.6.2 Open source řešení

Mezi nástroji pro řízení projektů lze najít i bezplatná open source řešení podporující širokou škálu funkcí. Příkladem je zmiňovaný Redmine, ale dále také Trac, Taiga nebo MyCollab.

Taková řešení mohou být v určitých případech výhodná, ačkoliv instalace a správa na vlastních serverech může odvádět soustředění zaměstnanců od hlavních podnikatelských cílů.

#### 3.4.6.3 Cena a rozsah funkcí

S komplexností aplikace souvisí také její cena. Ty nejjednodušší či omezené služby lze získat i zdarma a často dostačují.

Pokud podnik potřebuje některé pokročilé funkce, za většinu nástrojů zaplatí obvykle v řádu desítek až stovek dolarů měsíčně. Příkladem je diskutovaný Basecamp, ale také Asana nebo Pivotal Tracker, specializující se na agilní vývoj softwaru.

Pro řízení větších týmů a projektů existují komplexní nástroje jako je JIRA, které podporují veškeré procesy související s vývojem projektu. Jejich cena se pak pohybuje ještě o něco výše.

## 3.5 Nástroje pro ukládání a sdílení elektronického obsahu

Mezi nástroje pro podporu komunikace a kooperace při vývoji software lze zařadit také služby a produkty pro ukládání a sdílení kódu, dokumentů a jiných souborů souvisejících s vývojem produktů.

Členové týmu potřebují ukládat různý elektronický obsah, aby byl dostupný z internetu nebo privátní sítě. Soubory mohou být různě velké od malých textových dokumentů až po multimediální data o velikosti několik stovek megabytů (MB). Proto nelze se všemi nakládat stejně, ačkoliv jsou dnes uživatelé zvyklí na rychlost připojení i v řádu desítek MB za sekundu.

### 3.5.1 Typy dokumentů a souborů

Ukládané dokumenty a soubory dat lze rozlišovat dle toho, jaké informace nesou a k čemu slouží. Každá skupina má nějaká specifika a uživatelé s nimi nakládají různě. Mezi nejpodstatnější typy lze zařadit následující.

#### 3.5.1.1 Programový kód a dokumentace

Veškerý vývoj softwaru se točí kolem vytváření kódu, který programátoři potřebují ukládat a sdílet mezi sebou. Při implementaci softwaru vzniká kromě kódu také nejrůznější dokumentace. Může se jednat o popis softwarových součástí, programových rozhraní (API) nebo instalační příručky a manuály.

Udržování úložiště pro programový kód a dokumentaci představuje různé výzvy, protože členové týmu často musejí uchovávat několik na sobě závislých verzí aplikace nebo prohledávat historické změny apod. Za tímto účelem vznikly verzovací systémy popsané v následující podsekci.

#### 3.5.1.2 Podklady pro vývoj

Na začátku projektu je klient s více či méně propracovanými požadavky na projekt. Od této fáze do počátku implementace produktu vzniká mnoho různých podkladů a zadání, které vývojový tým potřebuje sdílet mezi svými členy.

Mezi ně lze zařadit počáteční specifikaci klienta a obchodních zástupců, následující projektovou dokumentaci a podrobným popisem funkcí systému.



Během vývoje se pak jedná převážně o další doplňující specifikace, modely softwarových součástí nebo podklady od grafiků a návrhářů uživatelského rozhraní.

### 3.5.1.3 Dokumenty obsahující celopodnikové specifikace

Mezi elektronický obsah sdílený v podniku zabývajícím se vývojem software lze také zařadit dokumenty popisující firemní pravidla a směrnice, standartní postupy při vývoji nebo komunikaci s klienty a další informace týkající se chodu projektů nebo celé společnosti.

V takovém případě se jedná převážně o textové nebo tabulkové dokumenty, které se často nemění i několik měsíců nebo dokonce let, stále však musí být dostupné všem nebo vybrané skupině zaměstnanců.

### 3.5.1.4 Informace pro zjednodušení vývoje

Během vývojového procesu vznikají také data, která jsou znovupoužitelná v dalším vývoji. Příkladem mohou být různá doporučení, ukázky kódu, nebo záznamy ze schůzek a školení.

Programátoři při vývoji software tráví značnou část času učením se nových postupů a hledáním řešení problémů. Například pokud je zveřejněna nová knihovna nebo změna ve vlastnosti jazyka, jeden z programátorů může funkcionality prozkoumat a prezentovat jí ostatním pro úsporu času. Záznam z prezentace nebo ukázky kódu je potřeba uschovat a sdílet pro budoucí použití.

Takové postupy lze zavádět i mimo programátorský tým, například pro uchování záznamů ze školení testerů a zaměstnanců zákaznické podpory, sdílené šablony pro grafiky nebo návrháře uživatelského rozhraní.

## 3.5.2 Disková úložiště

Většinu dat lze uchovávat na klasických diskových úložištích. Pokud ale potřebují zaměstnanci společnosti tyto data sdílet, existují vedle externích disků, flashových pamětí a jiných nosičů v zásadě dva přístupy. Prvním je instalace datového serveru na svou privátní síť nebo lze využít online služby pro ukládání dat v cloudu. Při zavádění je důležité zvážit zejména následující otázky.

- Jak velký objem dat bude potřeba ukládat?
- Odkud mají být data přístupná?
- Jak moc záleží na rychlosti přístupu a přenosu?
- Jak kritická je otázka bezpečnosti a důvěrnosti?

Odpovědi na tyto otázky pomohou přiřadit priority různým možnostem a usnadní výběr.

### 3.5.2.1 Sdílená úložiště v lokální síti

Jednou z možností uchovávání a sdílení dokumentů a jiných souborů jsou NAS (Network Attached Storage) neboli datová úložiště na síti. Ty představují jednoduchý a spolehlivý způsob zpřístupnění dat uživatelům sítě.

Přístup k datům pak lze nastavit pouze na zařízení připojená do kancelářské sítě, nebo za pomoci dalších technologií (například VPN = Virtual Private Network) i uživatelům mimo lokální firemní síť.

Výhodou lokálních úložišť je rychlost přenosu uvnitř firemní sítě, protože se data proti online službám nemusí stahovat z internetu. Důvodem pro zavedení NAS mohou být i bezpečnostní opatření, kdy firma věří, že dokáže svá data ochránit na vlastních serverech lépe než poskytovatelé služeb pro uchování obsahu.

Pro používání lokálních úložišť však musí společnost vynaložit další zdroje. Nastavit a zabezpečit NAS tak, aby bylo přístupné uvnitř i vně firemní sítě, vyžaduje určitou úroveň odbornosti. Zároveň je potřeba následná údržba, správa uživatelů, záloh, apod. Proto musí podnik disponovat specializovaným zaměstnancem nebo najmout externí firmu, která se o správu postará.

### 3.5.2.2 Online služby

Online úložiště poskytují vzdálený přístup k datům a zálohám přes internet, což přináší několik výhod. Za hlavní lze považovat možnosti sdílení nebo stabilitu a zabezpečení. Firma se díky tomu může naplno věnovat svému podnikání, tedy vývoji software.

Data je možné získat na jakémkoliv počítači například po přihlášení ve webovém prohlížeči. Zároveň lze zpřístupnit obsah jiným osobám na základě uživatelského jména a hesla, případně na základě sdílení odkazu na danou webovou stránku, která po čase expiruje.

### Dropbox a Google Drive

Při vývoji software se často využívají Dropbox nebo Google Drive, představující virtuální cloudové úložiště.

Obě služby poskytují aplikace pro desktopové platformy pro usnadnění ukládání a sdílení souborů. Ty umožňují vyhradit na disku adresář, který se automaticky synchronizuje se serverem, pokud je počítač připojený k internetu. Složkám lze nastavit sdílení s ostatními uživateli, což usnadní například přenos podkladů mezi grafiky a programátory. K dispozici jsou i mobilní aplikace pro iOS a Android.

Mezi klady těchto služeb patří také integrace s jinými nástroji, které společnosti při vývoji software používají. Při nahrání příloh pak uživatel může vybrat soubor přímo z virtuálního disku.

Výhodou Google Drive jsou také integrace s mnoha jinými službami společnosti Google. Mezi ty lze zařadit například Google Docs a Google Sheets

pro vytváření a editaci textových a tabulkových dokumentů přímo v prohlížeči nebo na mobilním zařízení. Zároveň je možné získat základní verzi služby Google Drive zdarma k Google účtu nebo jako součást balíku Google Apps for Work.

### Alternativy

Podobných služeb jako jsou Dropbox a Google Drive lze na trhu nalézt mnoho. Mezi technologickými giganty je možné najít například produkty Amazon Cloud Drive, Apple iCloud nebo Microsoft One. Jsou zde ale i další společnosti zabývající se tímto typem nástrojů a často nabízí různá vylepšení.

Do jedné skupiny spadají například produkty SpiderOak, Tresorit nebo Mega.co.nz, které se zaměřují na zlepšení bezpečnostních prvků a šifrování dat během jejich celého životního cyklu. I tyto služby lze při omezené kapacitě často získat zdarma.

### 3.5.3 Verzovací systémy

Další možností, jak ukládat elektronický obsah jsou verzovací systémy, které poskytují repozitáře obsahující různé verze obsahu (tzv. větve) s možností prohlížet jejich historii.

Taková funkcionalita je nesmírně užitečná zejména při vytváření programového kódu. Například při implementaci různých funkcí softwaru lze vytvářet speciální verze a udržovat na hlavních větvích produkční variantu kódu. Po dokončení implementace může kód procházet revizí před připojením do produkčních větví repozitáře. Pokud i přes kontrolu dojde po nasazení k nějaké zásadní chybě, je možné se vrátit k verzi před spojením větví. Postupy při práci s repozitáři se pak dají standardizovat, což vede k vyšší kvalitě kódu a předcházení problémů.

Verzovací systémy se ale používají nejen pro kód, ale také pro dlouhodobé uchování dokumentů, různých pravidel, manuálů nebo směrnic. Obecně se hodí pro jakékoliv soubory o menší velikosti (maximálně v řádu jednotek až desítek MB).

#### 3.5.3.1 Centralizované a distribuované verzovací systémy

Verzovací systémy lze rozlišit dle struktury připojených uzlů na centralizované a distribuované. V centralizovaných existuje jeden hlavní uzel nebo server, se kterým ostatní uzly musí komunikovat při většině akcí (např. ukládání a procházení změn). Mezi centralizované systémy patří CVS a Subversion.

V distribuovaných systémech má každý uzel svůj repozitář obsahující celou historii změn. Může s ním libovolně nakládat, vracet se ke starým verzím nebo měnit obsah a pokud se rozhodne, může své změny propagovat do jiných uzlů. Nejpoužívanějším distribuovaným verzovacím systémem je Git, existují ale i alternativy Mercurial a Bazaar.

### 3.5.3.2 Správa repozitářů

Nový repozitář lze vytvořit jednoduše pomocí nějakého nástroje, ať už se jedná o grafické nebo terminálové rozhraní. V distribuovaných systémech se obvykle vymezí jeden prvek, který slouží jako centrální uzel pro propagaci změn uživatelů podobně jako v centrálních repozitářích server.

Tento centrální prvek ale musí být všem přístupný (i vzdáleně), proto vývojový tým stojí před podobným rozhodnutím jako u sdílených diskových úložišť – buď mohou repozitář provozovat na vlastních serverech, nebo využijí nějakou službu poskytující online repozitáře.

Služby pro online repozitáře usnadňují členům vývojového týmu jejich správu a přístup. Zároveň často přidávají další funkce jako grafické rozhraní nebo možnosti administrace.

Mezi nejpoužívanější patří například GitHub, BitBucket, GitLab Cloud nebo CloudForge, na trhu jich ale lze nalézt desítky. Většina z nich jsou v omezené verzi zdarma. Placená verze se nejčastěji liší dle počtu uživatelů, privátních repozitářů nebo kapacity uložených dat.

### 3.5.4 Úložiště jako součásti jiných služeb

Další možností, jak ukládat a sdílet soubory související s vývojem software, jsou zabudované funkce jiných nástrojů.

Například některé služby pro výměnu zpráv a jiných příloh zmíněné na začátku kapitoly ukládají odesílané soubory a poté nabízí vyhledávání. Takto lze šířit záznamy z porad a jiné dokumenty.

Podobně je možné využít nástroje pro řízení projektů, které poskytují příkladání souborů k úkolům a diskuzím. Tímto způsobem mohou uživatelé sdílet grafické podklady, modely a jiné specifikace.

## 3.6 Celkové shrnutí nástrojů pro podporu komunikace a kooperace

Komunikace a kooperace při vývoji software je klíčovým faktorem úspěchu a na trhu existuje mnoho nástrojů, které ji dokáží podpořit a zefektivnit. U většiny typů nástrojů lze zvolit mezi bezplatnou a placenou variantou, širokou škálu funkcí a jednoduchostí nebo cloudovou či lokální verzí.

### 3.6.1 Online služby nebo provoz na vlastních zdrojích

U většiny typů nástrojů může firma zvolit, zdali ho využije jako službu, nebo ho bude provozovat na vlastních zdrojích.

Využívání služeb jiných společností umožňuje udržovat plné soustředění podniku na vývoj produktů a nezabývat se nastavováním, zabezpečováním a správou podpůrných nástrojů.

### 3.6. Celkové shrnutí nástrojů pro podporu komunikace a kooperace

---

Naproti tomu provoz na vlastních serverech interními zaměstnanci může mít i jiné výhody jako jsou úspory nebo možnosti nastavení a přizpůsobení, neboť má společnost správu plně pod svou kontrolou. A pokud firma provozuje vlastní servery i z jiného důvodu, nemusí pro administrátory představovat problém zavést systém pro řízení projektů, vytvořit repozitář nebo připojit NAS.

#### 3.6.2 Podpora mobilních zařízení

Obecně lze říct, že většina dnešních aplikací pro podporu komunikace a kooperace nabízí práci na všech majoritních platformách nejen prostřednictvím přizpůsobivé webové aplikace, ale nejčastěji i speciální aplikací pro danou platformu.

U některých placených služeb to lze dokonce považovat za povinnost. Výhodou iOS aplikací proti webu je první řadě možnost práce offline a lepší uživatelské rozhraní, které je přizpůsobeno přímo pro zařízení. Mobilní verze softwaru také může přímo komunikovat s jinými aplikacemi a tím usnadnit práci se soubory, kalendářem, prohlížečem apod.

# Analýza procesů a požadavků ve vybrané společnosti

Tato kapitola se zabývá analýzou komunikačních a kooperačních procesů a podpůrných nástrojů uvnitř vybrané společnosti, která se soustředí na vývoj softwarových produktů na zakázku. Za cíl si klade identifikaci slabých stránek a návrh řešení.

## 4.1 Popis společnosti

Pro tuto analýzu byla zvolena česká společnost STRV s. r. o., dříve podnikající pod názvem uLikeIt. V dnešní době se orientuje především na vývoj celkových softwarových řešení skládajících se z webových a mobilních aplikací.

### 4.1.1 Historie

Historie firmy STRV má kořeny v roce 2004, kdy její zakladatelé začali podnikat v oblasti vývoje webových stránek pro menší české klienty. Po prvních pěti letech začalo zakázek přibývat a bylo potřeba přijímat zaměstnance, a proto byla v roce 2009 založena společnost s ručením omezeným uLikeIt. Ta se zprvu zabývala také zejména tvorbou webových aplikací.

S příchodem prvního chytrého telefonu ale začala vznikat úplně nová oblast ve vývoji software. Po pár letech narostl počet uživatelů chytrých zařízení i v České Republice tak, že firmy začaly mít zájem o aplikace šité na míru mobilním telefonům a tabletům. Od roku 2010 proto začala uLikeIt získávat také zakázky na vývoj aplikací pro mobilní platformy.

Dalším zlomem byl rok 2012, kdy společnost uspěla v projektu CzechAccelerator, který pomáhá českým firmám získat zkušenosti s podnikáním na zahraničních trzích. Firma zůstala v San Franciscu i po uplynutí doby projektu a díky tomu začala tehdejší uLikeIt pomalu navazovat kontakty na budoucí klienty v Silicon Valley.

Během pár let si mezi americkými klienty vydobyla své jméno a do současnosti narostla z hrstky zaměstnanců na střední podnik zaměstnávající přes 70 vývojářů. Podstatnou částí portfolia jsou nově vznikající projekty (startupy) z různých akceleratorů pro začínající společnosti. V létě 2014 byla přejmenována na STRV.

### 4.1.2 Geolokační uspořádání společnosti

Ačkoliv STRV stále spolupracuje na několika českých projektech, během let se přeorientovala především na zákazníky z USA a Kanady.

Postupně proto kromě San Francisca otevřela pobočky také v New Yorku a Los Angeles. Většina vývojového týmu běžně operuje z České Republiky a do USA příležitostně cestují, aby poznali tamní zaměstnance, prostředí a klienty, pro které pracují. Jinak ale americké pobočky představují zejména obchodní oddělení.

Ačkoliv v České Republice sídlí hlavní část vývoje, pražská centrála není často plně vytížená. Zaměstnanci totiž nemají pevnou pracovní dobu ani nemusí chodit do kanceláře. Asi polovina z nich pracuje pravidelně z domova a do Prahy dojíždí pouze příležitostně.

Vývojový tým lze tedy považovat za distribuovaný, i když velká část týmu působí v České Republice. V takovém prostředí může být spolupráce na projektech přece jen o něco složitější, a proto hrají softwarové nástroje pro podporu komunikace a kooperace podstatnou roli.

## 4.2 Vývojový proces a spolupráce na projektech

O vývojovém procesu a koordinaci projektů pojednávají první dvě kapitoly této práce. V této společnosti ale spolupráce na vývoji software probíhá vcelku nezvykle.

### 4.2.1 Nezvyklé řízení softwarových projektů

Hlavní rozdíl představuje absence zaměstnanců v roli projektových manažerů. Do vývojového procesu se ve firmě totiž zapojují pouze zaměstnanci pro podporu prodeje a tým programátorů a grafiků, kteří komunikují přímo s klienty.

Tento přístup nabízí zaměstnancům volnost při organizaci pracovní doby a rozšíření pracovní náplně zároveň ztraktivňuje zaměstnání. Zaměstnavatel tedy vkládá do vývojového týmu velkou důvěru, s čímž přichází také více odpovědnosti.

Na druhou stranu jsou na programátory a grafiky kladeny větší nároky, neboť musí být odborníci ve svém oboru se schopností dotáhnout věci do konce, stejně tak jako musí mít skvělé komunikační schopnosti i v anglickém jazyce.

Podobné však platí o obchodních zástupcích, kteří mimo své zaměření musí rozumět procesu vývoje software, aby byli schopni domlouvat zakázky bez odborného pohledu projektového manažera.

### Zrychlení vývoje

Vývoj bez projektového manažera byl zaveden zejména kvůli klientele, na kterou se firma v USA z počátku zaměřovala. Tou byly začínající společnosti realizující nové nápady, u kterých je podstatná rychlost vývoje.

U těchto projektů se obvykle nejprve implementuje tzv. MVP (minimum viable product), neboli produkt s nejmenší možnou funkcionalitou, který využije značná část cílové skupiny. Implementace MVP trvá řádově několik týdnů až měsíců, poté je produkt spuštěn a podle úspěšnosti vývoj buď pokračuje, nebo je projekt ukončen.

Rychlost implementace hraje proto podstatnou roli. Zároveň zákazník často vůbec neví, jak by měl produkt po technické stránce fungovat. Vývojový tým se tak stává nejen zhotovitelem softwaru, ale také pomáhá řešit technické a funkční detaily celého projektu.

Odstraněním pozice projektového manažera se vývoj značně urychlí, neboť zákazník může komunikovat přímo s programátory nebo grafiky. Tento přístup se postupem času ověřil, a proto přešel v aplikaci i u větších zákazníků.

### 4.2.2 Procesy a komunikace při řízení projektů

Absence projektového manažera mimo urychlení vývoje představuje také rozšíření odpovědností zaměstnanců, protože vývojáři i obchodní zástupci musí být v určitém pohledu schopni zastávat jeho funkce.

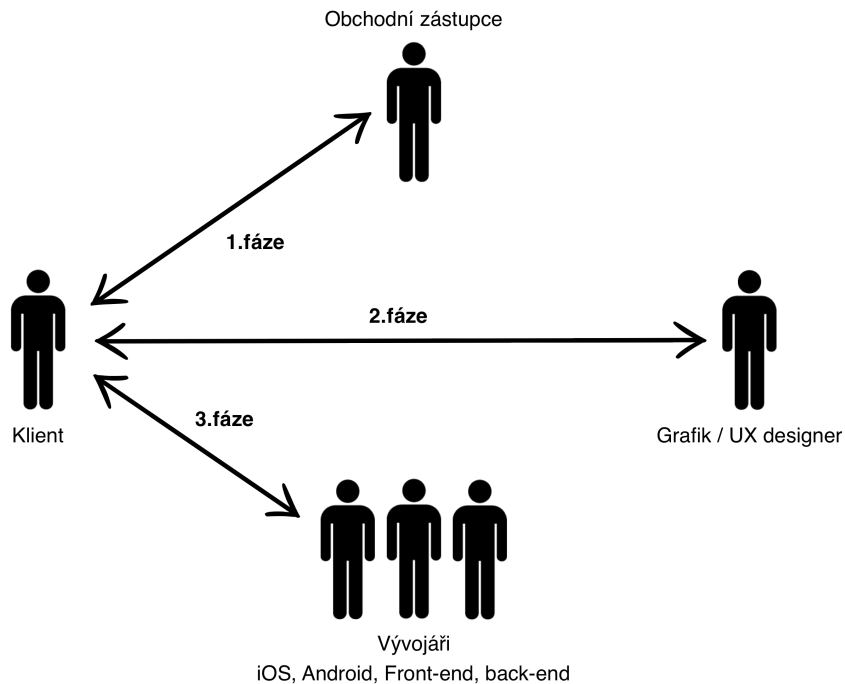
Typický proces řízení softwarových projektů v STRV lze rozdělit v zásadě na 3 fáze. V první je úkolem nalézt zákazníka a domluvit podmínky spolupráce včetně rámcového zadání. Následuje podrobnější specifikace založená na grafickém návrhu a poté je spuštěna samotná implementace softwarového produktu.

Diagram 4.1 znázorňuje hlavní komunikaci mezi klientem, obchodním zástupcem a vývojovým týmem. U některých projektů mohou také různé z fází scházet, například pokud má zákazník svého grafika nebo tým pro podporu serverové části aplikace.

#### Fáze 1 – Rámcové ujednání zakázky

V první fázi figuruje zejména klient a obchodní zástupce. Ten se snaží zákazníkovi poradit s jeho problémem či nápadem a navrhnout řešení, které může firma implementovat. Proto musí všichni obchodní zástupci alespoň částečně rozumět problematice vývojových procesů.





Obrázek 4.1: Proces vývoje softwarových produktů ve vybrané společnosti

Pokud jsou jednání úspěšná, vznikne na základě domluvy rámcová smlouva o vývoji a obchodní zástupce vyčlení tým skládající se z grafika a potřebných vývojářů.

Hlavní kooperace tedy probíhá mezi klientem a obchodním zástupcem. Ten však někdy potřebuje komunikovat i s vývojáři, například pro zjištění technické proveditelnosti klíčových funkcí nebo ohledně předání projektu do následující fáze.

### **Fáze 2 – Dojednání detailů a grafického návrhu**

V tomto období začíná vlastní vývoj softwarového produktu, kterého se účastní především grafik respektive návrhář uživatelského rozhraní – tyto dvě pozice zastává nejčastěji jeden zaměstnanec. Ten komunikuje s klientem často na denní bázi i několik týdnů.

Výsledkem je kompletní grafický návrh aplikace pro všechny definované platformy. Ten slouží jako předloha pro programátory a zároveň upřesňuje všechny funkční detaily.

### Fáze 3 – Implementace

Ve třetí fázi předá grafik všechny podklady týmu vývojářů. Ten se obvykle skládá z jednoho programátora pro každou z implementovaných platforem, například po jednom iOS, Android, frontend a backend vývojáři. Programátoři se pustí do práce dle zadání a při případných nejasnostech kontaktují grafika nebo eventuálně i přímo klienta.

Implementace kódu trvá několik týdnů až měsíců, během kterých programátor vydává průběžné verze softwaru, aby zákazník věděl, jakým směrem se produkt ubírá. Ten pak často upřesňuje další funkční detaily jednotlivých platforem, aby vše odpovídalo přesně jeho představám. Programátoři mu však zároveň odkrývají svůj pohled na změny, aby si uvědomoval související průtahy vývoje.

Na konci této fáze programátoři koordinují s klientem prvotní spuštění aplikace a zajišťují technickou podporu uvedení produktu na trh.

### Další spolupráce

Po spuštění první verze aplikace nebo dosažení definovaného cíle mohou nastat různé scénáře. V pozitivním případě jsou očekávání naplněna a vývoj produktu pokračuje implementací dalších funkcionalit, ve kterých probíhá komunikace podobně jako ve fázích dvě a tři.

Jednu z možností představuje bohužel také neúspěch produktu u uživatelů nebo investorů a následné ukončení projektu. Tento scénář nastává u začínajících projektů poměrně často, zejména u startupů v Silicon Valley, kde některé odhady dokonce udávají úmrtnost až 90 %.

### 4.2.3 Používané softwarové nástroje

Použití softwarových nástrojů pro podporu komunikace a kooperace je nezbytnou součástí vývojového procesu v distribuovaných i lokálně založených týmech a STRV není výjimkou.

#### Komunikační nástroje uvnitř společnosti

Mezi hlavní softwarové nástroje pro podporu komunikace uvnitř společnosti patří služba Slack [31], která téměř úplně nahradila použití emailů. Slack používají všichni zaměstnanci pro výměnu informací ohledně čehokoliv, co souvisí s jejich pracovní náplní.

Zároveň se Slack používá pro získávání přehledných informací ze služeb třetích stran pomocí integrací. Mezi ně patří nástroje podporující projektové řízení, reportování chyb, získávání zpětné vazby apod.

V distribuovaném týmu lze osobní setkání částečně nahradit videokonferenčními nástroji. Zaměstnanci používají Skype [38] i Google Hangouts [35]

pro hovory na úrovni týmů. Jednou týdně se také pořádají celofiremní meetingy, pro které už Google Hangouts nevystačují (lze připojit maximálně 15 účastníků), proto se začala využívat služba GoToMeeting [47].

Kooperace na projektech také vyžaduje své nástroje. Každý tým si může zvolit dle svých potřeb, obvykle se ale vývoj produktu koordinuje pomocí organizace úkolů v aplikaci Basecamp [44]. Pro sdílení kódu a dalších dokumentů se využívá verzovací systém Git s repozitáři na webovém serveru GitHub [48]. Při předávání grafických podkladů se často kromě Basecampu používá také virtuální úložiště Google Drive [49].

Komunikační nástroje se však mohou lišit dle klienta a to zejména v případě, kdy má své vlastní grafiky nebo programátory, kteří používají služby odlišné. Pro projektové řízení proto někteří vývojáři používají také Trello [43] nebo Asanu [50], výměnu zpráv pomocí služby HipChat [27] nebo sdílení souborů ukládáním do Dropboxu [51].

Společnost také využívá Google Apps for Work. Všichni zaměstnanci mají tudíž účet s firemním emailem, který umožňuje přístup ke sdíleným kalendářům, Google Drive nebo Google Hangouts.

#### **Využití softwarových nástrojů při komunikaci s klienty**

V úvodní fázi projektu komunikují s klientem obchodní zástupci, kteří operují přímo v amerických pobočkách. Komunikace proto probíhá především osobně nebo prostřednictvím klasických kanálů, tedy emailem a telefonem.

Ve chvíli, kdy projekt postoupí do další fáze, přichází na řadu vývojový tým založený primárně v České Republice. Osobní schůzky proto nahrazují videokonference, zejména pomocí služby Skype nebo Google Hangouts.

V některých případech je klient také připojen do Slacku, aby mohl vývojářům jednoduše posílat zprávy a oni ho mohli informovat o pokrocích projektu.

### **4.3 Nedostatky komunikace a kooperace a návrh zlepšení**

Softwarové projekty lze řídit odlišně, a proto se postupy různí dle společnosti. Vždy však existují slabší a silnější stránky tohoto procesu ovlivňující rychlost, efektivitu nebo náklady.

V STRV jsou silné i slabé stránky často spjaty s nezvyklým vývojovým procesem, tedy jako následek rozdílné lokace členů vývojového týmu nebo absence projektového manažera. Nezávislost umístění zaměstnanců ztraktivňuje zaměstnání, díky čemuž je možné získat ty nejlepší, kteří jsou schopni řídit sami sebe. Bez projektového manažera pak dokáží pracovat efektivněji a dosáhnout cíle dle potřeby klienta.

Na druhou stranu vznikají neobvyklé problémy vyplývající právě z tohoto přístupu. Zkoumáním vývojového procesu a komunikace uvnitř i vně týmu

byly odhaleny 2 hlavní nedostatky, respektive příležitosti ke zlepšení, které rozebírají následující podsekcce.

#### 4.3.1 Nedostatečná komunikace mezi vývojáři a klientem

Vývojový tým bez projektového manažera musí alespoň částečně zastávat jeho funkce. Jednou z nich je informovat klienta o pokrocích na projektu, aby bylo možné získávat zpětnou vazbu už v průběhu vývoje.

Při grafickém a funkcionálním návrhu aplikace (2. fáze) je přímo nutné předávat průběžné verze zákazníkovi, aby je mohl komentovat a grafik měl možnost zapracovávat připomínky ihned. Tím lze předejít ztrátě velkého množství času na opakovaném přepracovávání návrhu.

V případě programovací fáze vývoje pak zákazník preferuje vidět, že se projekt pohybuje správným směrem a že se neplýtvá jeho drahocenným časem. Proto programátoři vytváří průběžné verze, ve kterých může klient ozkoušet částečně implementovanou funkcionalitu. Jak často předávat rozpracované programové verze zákazníkovi záleží ale do jisté míry na typu projektu. U MVP jsou implementované funkcionality menší, ale i přesto může být složité zkompletovat každý den verzi, kterou lze spustit na mobilním zařízení klienta.

Pokud tedy vývojový tým neodesílá aktuální ukázkovou verzi aplikace nebo grafický návrh každý den, zákazník dostává informace pouze nahodile nebo na vyžádání. Díky tomu ztrácí kontinuální přehled nad projektem a dokonce může mít pocit, že se některé dny vývoj až zbytečně protahuje, ačkoliv vývojový tým dělá vše dle jeho požadavků.

#### 4.3.2 Přehled o vytíženosti programátorů

Při získání nového projektu musí obchodní manažer vyčlenit tým lidí zodpovědných za implementaci. K tomu potřebuje vědět, jak jsou zaměstnanci aktuálně vytíženi, což může v daném prostředí představovat nelehký úkol.

Ačkoliv má k dispozici seznamy projektů a týmů, vytíženost vývojářů nelze vždy tak jednoduše odvodit. Každý projekt totiž vytěžuje tým různě dle fáze, ve které se nachází. Proto musí často komunikovat přímo s vývojáři a získávat od nich průběžný stav vytíženosti. To může projekt zbytečně zdržovat, zejména když operují v jiných časových pásmech.

Přiřazování práce se proto s narůstajícím počtem zaměstnanců stává složitější. Obchodní zástupci by v roli vedoucích pracovníků potřebovali kontinuální přehled o vytíženosti. V tuto chvíli vytíženost odhadují a následně kontaktují vývojáře jednotlivě.

#### 4.3.3 Návrh řešení

Možným řešením zmíněných problémů by bylo přidat jednu vrstvu managementu tedy zaměstnance, kteří by měli na starosti řízení vývojářů a jejich

### 4.3. Nedostatky komunikace a kooperace a návrh zlepšení

---

komunikaci s klienty a obchodními zástupci. Tento scénář však jde proti současné filozofii vývojového procesu – ve firmě jsou pouze zaměstnanci, kteří se dokáží řídit sami.

Pro komunikaci by mohl sloužit nástroj pro správu projektů, kam by měl zákazník nebo obchodní zástupce přístup a mohl by procházet jednotlivé úkoly a komentáře od vývojového týmu. V takovém případě by ale mohl získat přístup i k interním informacím nebo soukromým diskuzím programátorů ohledně projektu. Zároveň by musel informace vyhledávat sám, ačkoliv by preferoval způsob, kdy se k němu informace dostaví sama.

Proto je nutné standardizovat komunikaci mezi vývojovým týmem a klientem a zároveň jim dát možnost hlásit jejich celkové pracovní vytížení. Řešením by mohl být nástroj, který by umožnil programátorům a grafikům vytvářet krátká a výstižná hlášení, na čem zrovna daný den pracovali a jak moc se cítí vytíženi. V praxi by pak takové hlášení mohlo obsahovat například poznámku „Implementace přihlašování do aplikace – hotovo“ s aktualizací stavu vytíženosti na „Stíhatelné“. Na konci každého dne by každý vývojář vložil komentář k jeho aktivním projektům a klient by měl k těmto informacím ihned přístup.

Ke komentářům by také měli přístup ostatní zaměstnanci, díky čemuž by nemuseli procházet systémem na řízení projektů a neustále by věděli, na čem zbytek týmu pracuje.

---

## Analýza softwarového řešení

Na základě předchozích požadavků byl navržen informační systém pro denní reportování aktivit programátorů a grafiků/návrhářů uživatelského rozhraní, pracovně nazvaný ReportIT.

Tato práce se zaměřuje na implementaci funkčního řešení založeného na klientské aplikaci pro mobilní operační systém iOS. Pro dosažení plně funkční verze bude implementována i jednoduchá serverová část aplikace.

### 5.1 Cíle projektu

Projekt si za hlavní cíl klade zjednodušení řízení týmů za pomoci evidence vytíženosti jednotlivých členů, podpoření efektivnější komunikace s klienty a zároveň zlepšení kooperace a informovanosti uvnitř vývojového týmu.

Hlavní myšlenkou je sdílení a shromažďování informací o práci jednotlivých vývojářů. Informace budou dostupné členům týmů navzájem, osobám zodpovědných za projekty a v neposlední řadě klientům, kteří získají přehled o práci developerů na jejich zadání.

Od zavedení aplikace do provozu se očekává zjednodušení kooperace na projektech a zlepšení poskytovaných služeb ve smyslu lepší péče a spolupráce se zákazníkem.

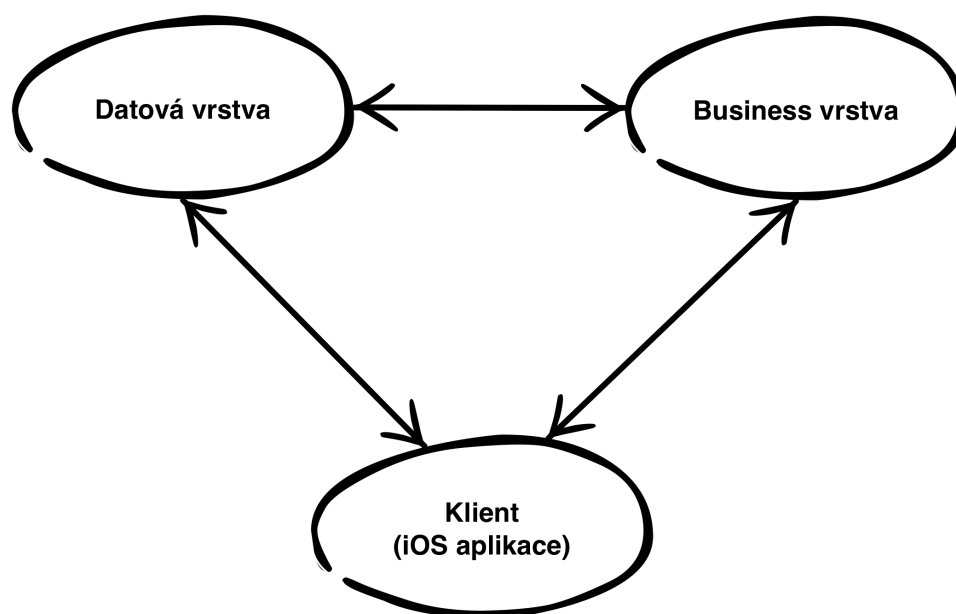
### 5.2 Globální architektura systému

Systém bude implementován jako architektura typu klient/server, viz. obrázek 5.1. Klientskou část systému bude představovat mobilní iOS aplikace a do budoucna bude možné přidat další klienty, zejména webovou a Android verzi.

Serverová část se bude skládat ze dvou vrstev – centrální datové vrstvy v podobě služby třetí strany a programované serverové aplikace.

Datová vrstva musí poskytovat zabezpečení na základě autentizace uživatele a dále musí podporovat různé role pro omezení práv pro čtení a zápis jednotlivých typů dat. Díky tomu bude možné vykonávat jednoduché požadavky pro získání a ukládání dat přímo z klientské aplikace bez nutnosti dotazování vlastního serveru.

Programovaná serverová část aplikace pak bude sloužit pro složitější business operace a podporu dalších aktivit, které nelze vykonat přímo nad datovou vrstvou. Příkladem může být ukládání větších souborů nebo generování denních reportů do emailů.



Obrázek 5.1: Globální architektura systému ReportIT

## 5.3 Požadavky

Na základě konzultace s jedním z vedoucích zaměstnanců společnosti došlo ke vzájemné dohodě na požadavcích aplikace. Následující podsekcce rozebírají funkční a nefunkční požadavky softwarového řešení.

### 5.3.1 Funkční požadavky

#### Správa společnosti

1. Systém umožní vytvářet společnosti

2. U uživatelů přiřazených do společnosti umožní systém rozlišovat role administrátor, vývojář a klient
3. Systém umožní přizvat uživatele do společnosti
4. Systém umožní přijmout pozvánku do společnosti
5. Systém umožní úpravu názvu a loga společnosti

#### **Uživatel**

1. Uživatel se bude schopen přihlásit
2. Uživatel se bude schopen registrovat
3. Uživatel bude schopen upravovat svůj profil (fotografie, jméno, pozice a kontaktní údaje)
4. Uživatel bude schopen zobrazit profily ostatních členů společnosti

#### **Projekt**

1. Systém umožní vytváření projektů uvnitř společnosti
2. Systém umožní změnu názvu projektu
3. Systém umožní přiřazovat členy společnosti k projektům
4. Systém umožní zobrazit přiřazené projekty
5. Systém umožní archivovat a znovuotevřít projekty

#### **Záznamy projektu (hlášení)**

1. Systém umožní přidávat denní hlášení k projektům
2. Systém umožní prohlížení záznamů dle projektů
3. Systém umožní jednoduchou aktualizaci stavu projektů

#### **Vytíženost**

1. Systém umožní aktualizovat vytíženost člena společnosti
2. Systém umožní procházet uživatele dle vytíženosti



### 5.3.2 Nefunkční požadavky

#### Podnikatelské

1. Vývojář potřebuje evidovat své aktivity na základě denních reportů
2. Vývojář potřebuje evidovat svou vytíženost
3. Vedoucí pracovník potřebuje přehled o vytíženosti zaměstnanců
4. Klient potřebuje mít přehled o pokroku na jeho projektu

#### Systemové

1. Systém bude realizován jako architektura klient/server
2. Klientská aplikace bude schopna zobrazit data o hlášeních i po odpojení od internetu
3. Systém bude přístupný pomocí mobilní aplikace pro operační systém iOS
4. Mobilní aplikaci bude možné využít na všech zařízeních používajících iOS 8.1 a vyšší

#### Softwarové

1. Škálovatelnost, dostupnost a výkon

Tyto požadavky se týkají zejména serverové části aplikace. Protože pro serverovou část budou využity služby třetích stran, tyto vlastnosti odpovídají jejich schopnostem. Proto bude podstatné vybrat služby, které bude možné v budoucnu škálovat a zároveň se lze spolehnout na stabilní dostupnost a výkon.

2. Bezpečnost

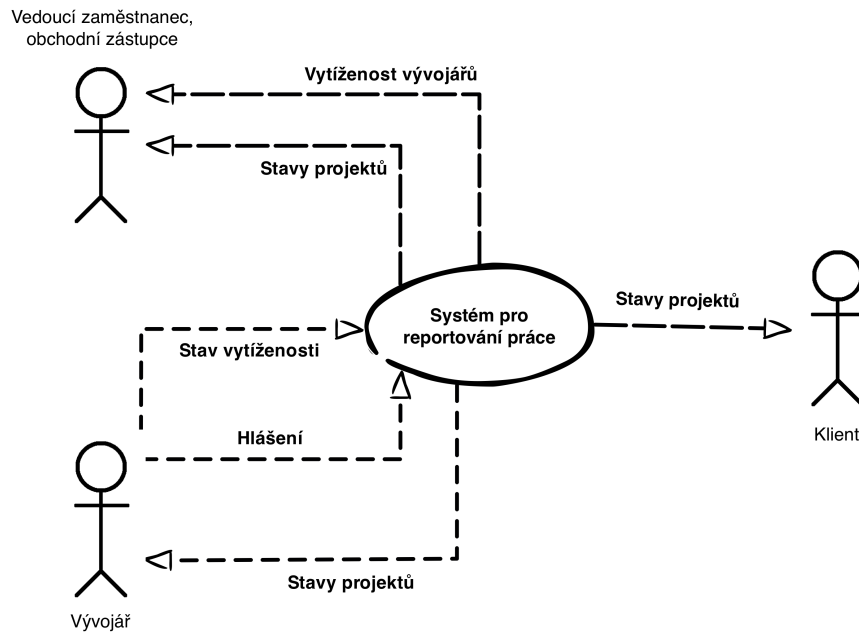
- a) Uživatel získá přístup pouze k údajům společnosti, ke které se připojí, omezených dle přiřazené role
- b) Komunikace klientské aplikace se serverovou částí bude zabezpečena pomocí protokolu HTTPS

3. Další nefunkční požadavky

- a) Pro použití systému bude vyžadováno přihlášení
- b) Přihlásit se bude možné pomocí účtu služeb Google, Facebook, Twitteru a GitHub
- c) Odhlášení ukončí veškerou aktivitu uživatele v systému

### 5.3.3 Kontextový diagram

Kontextový diagram vymezuje hranice mezi systémem a vnějším okolím a je zdrojem pro datovou analýzu a analýzu aktivit.



Obrázek 5.2: Kontextový diagram ReportIT

Schéma na obrázku 5.2 zobrazuje projekci implementovaného systému zaměřeného na externí datové toky, čímž definuje vnější chování systému. Vstupy do systému budou zajišťovat zejména vývojáři, zatímco zobrazení výstupů bude v jisté podobě umožněno všem typům uživatelů.

## 5.4 Případy užití

Na základě požadavků byly navrženy případy užití. Diagramy byly vytvořeny v aplikaci Diagrammix. Případy užití lze rozdělit na dva základní typy dle toho, v jaké fázi se uživatel nachází:

1. Fáze přihlašování a výběru aktivní společnosti
2. Fáze interakcí uvnitř vybrané společnosti

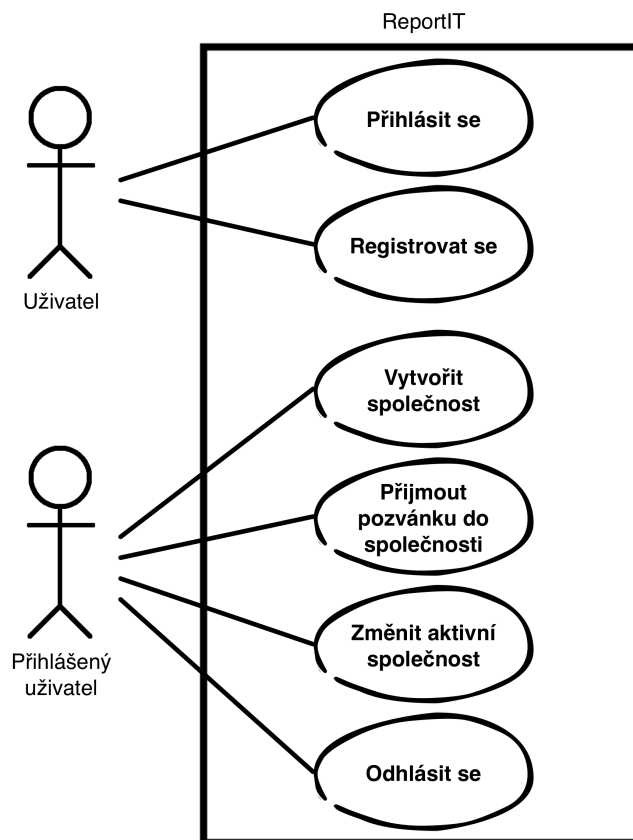
### 5.4.1 Aktéři

Aplikace ReportIT obsahuje několik typů aktérů. Ve fázi přihlašování a výběru aktivní společnosti lze odlišovat různé případy užití pro přihlášeného a nepřihlášeného uživatele.

Po přihlášení a výběru jedné ze společností je možné rozlišit tři různé aktéry dle role, kterou uživatel ve společnosti zastává, tedy administrátor, vývojář nebo klient.

### 5.4.2 Úvodní obrazovka a výběr společnosti

Centrem veškeré funkcionality je entita společnosti, na kterou se váží členové, projekty i hlášení. Nejprve je proto nutné vybrat jednu společnost, uvnitř které bude uživatel interagovat. Diagram 5.3 znázorňuje případy užití pro tento stav aplikace.



Obrázek 5.3: Případy užití přihlášení a výběru společnosti

Po spuštění aplikace bude mít uživatel možnost registrovat se nebo přihlásit pomocí jedné z autentizačních metod. Neregistrovaný uživatel tedy nemůže

aplikaci vůbec používat, jak bylo definováno v nefunkčních požadavcích.

Po přihlášení bude muset vybrat společnost, ve které chce provádět akce. V tu chvíli bude mít zároveň možnost vytvořit novou společnost, nebo se připojit do nějaké existující po ověření pozvánky od administrátora.

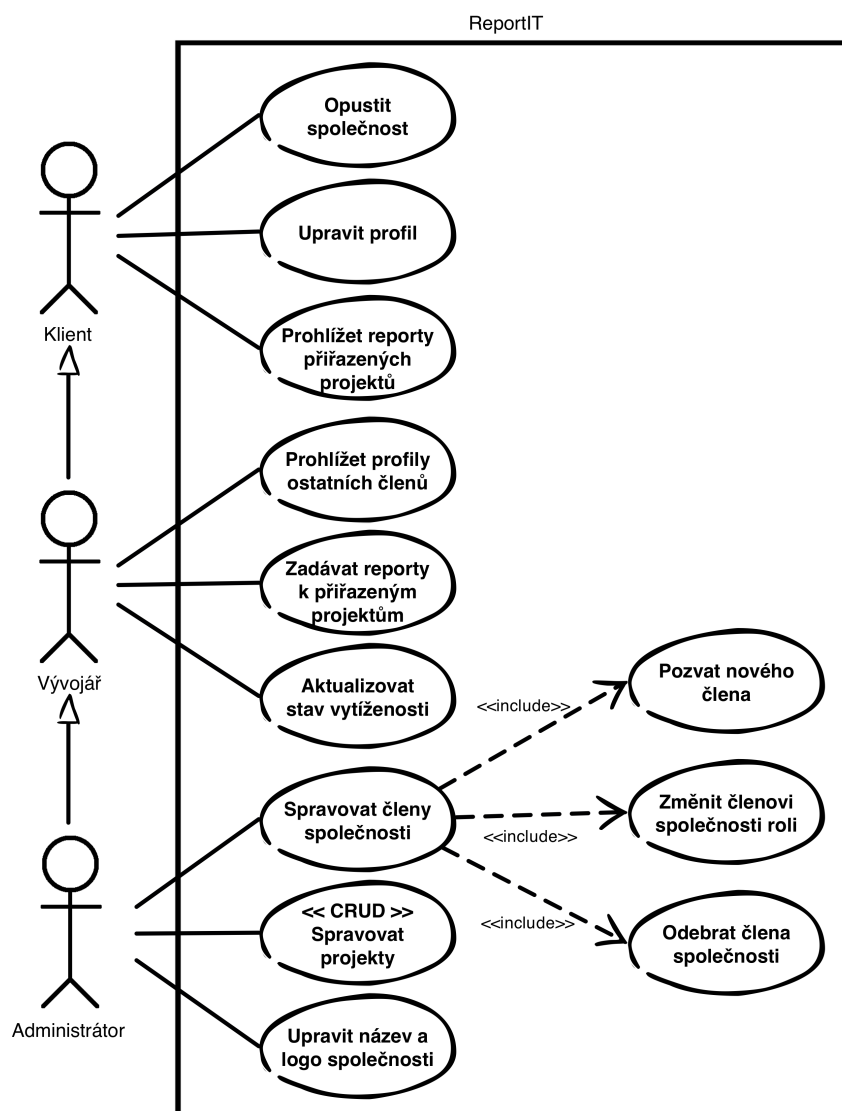
### 5.4.3 Případy užití uvnitř vybrané společnosti

Po výběru společnosti se uživatel ocitne v jedné ze tří rolí – administrátor, vývojář nebo klient. Role představují aktéry, kteří jsou od sebe navzájem odvození. Vývojář má tedy všechny případy užití jako klient a k tomu některé navíc. Administrátor k případům užití vývojáře přidává ještě možnosti administrace projektů a společnosti. Diagram 5.4 znázorňuje případy užití pro každého aktéra.

Případy užití klienta zahrnují pouze editaci profilu, možnosti opuštění společnosti a nejpodstatnější prohlížení reportů u projektů, ke kterým je přiřazen. To je vlastně vše, co klient potřebuje.

Vývojář navíc vidí další spolupracovníky společnosti, díky čemuž může sledovat, na čem ostatní pracují. Kromě toho mu přibývají klíčové případy užití pro zadávání reportů k projektům, ke kterým je přiřazen a možnost aktualizovat svůj stav vytíženosti.

Administrátor už vidí veškeré projekty ve společnosti a může je spravovat, tedy vytvářet, upravovat, archivovat nebo přiřazovat uživatele, čímž jim do projektu povolí přístup. Administrátor může také měnit název a logo společnosti nebo spravovat její členy. Přidávání členů funguje na základě pozvánky, na kterou se role váže. Tuto roli pak administrátor může dodatečně změnit a člena může ze společnosti také odebrat.



Obrázek 5.4: Případy užití po výběru společnosti

---

# Implementace softwarového řešení

Implementace klienta pro mobilní platformu iOS probíhala v současné verzi vývojového prostředí Xcode 6.3 [52], který obsahuje také simulátor operačního systému iOS, na kterém fungují mobilní zařízení společnosti Apple.

Pro vývoj serverové části aplikace v javascriptu byl použitý jednoduchý textový editor Atom [53].

## 6.1 Použité technologie

Následující podsekcce popisují technologie použité pro různé části aplikace. Dle celkové architektury systému popsané v sekci 5.2 lze v systému rozlišit klientskou aplikaci (iOS) a serverovou část skládající se z datové vrstvy a serverové aplikace pro podporu dalších funkcí.

### 6.1.1 Serverová část aplikace – datová vrstva

Pro datovou vrstvu byla vybrána služba Firebase [54], poskytující synchronizaci dat v reálném čase. To v praxi znamená, že lze všechny změny propagovat napříč zařízeními ve stejný okamžik, kdy nastanou.

#### Jednoduchost, síla a bezplatná verze

Za velkou výhodu Firebase lze považovat možnost jednoduché a přímočaré práce s daty ve Firebase. Celá databáze je totiž uložena jako jeden JSON objekt, do kterého lze přistupovat až z desítek různých aplikací v jednu chvíli.

Další výhodou Firebase jsou podpůrné knihovny (Software Development Kit, SDK) pro zjednodušení komunikace se serverem, které lze stáhnout pro

různé webové a mobilní programovací jazyky. Ty zároveň podporují jednoduché transakce nebo vyhledávání v různých uzlech datového JSON objektu. Při programování v jazycích, pro které neexistuje oficiální SDK, je možné provádět stejné operace za pomoci přehledně zdokumentovaného REST API.

Pozitivní je také cena, Firebase lze totiž používat i zdarma. Bezplatná verze má různá omezení, jako maximálně 50 souběžných požadavků nebo úložiště 100 MB. Tyto parametry však pro prvotní verzi produktu bohatě vystačí. Proto se ideálně hodí pro projekty, kde je potřeba aplikaci nejdříve vyzkoušet a v případě velkého zájmu ji lze přesunout do placených plánů.

### Možnosti autentizace a nastavení bezpečnostních pravidel

Velkou výhodou jsou také funkce pro přihlášení, správu autentizačních tokenů a bezpečnostní pravidla. SDK poskytuje metody pro jednoduchou autentizaci klientské aplikace pomocí emailu a hesla nebo prostřednictvím různých služeb třetích stran. Po ověření uživatele si knihovna sama spravuje autentizační token.

Při zpracování požadavku na serveru tedy Firebase zná identitu žadatele, a proto může aplikovat různá bezpečnostní pravidla, která si administrátor sám definuje přímo ve webovém rozhraní.

Za pomoci kombinace hrstky základních příkazů lze tvořit pokročilá pravidla pro omezení přístupu do jednotlivých uzlů datového JSON objektu. Díky tomu je možné do Firebase přistupovat přímo z klientské aplikace, bez zatěžování vlastního serveru.

```
"companies": {
  "$company_id": {
    ".write": "!data.exists() || data.child('owner_uid').val() === auth.uid",
  }
},
```

Obrázek 6.1: Ukázka pravidel pro řízení přístupu do datového úložiště Firebase

Obrázek 6.1 znázorňuje možnou definici pravidel tak, že do uzlu *companies* může uživatel zapisovat, pouze pokud daná společnost prozatím neexistuje, nebo pokud je on sám jejím vlastníkem.

### Nevýhody Firebase

Za jednu z nevýhod lze považovat, že do Firebase není vhodné ukládat větší soubory, například profilové fotografie uživatelů. V případě ReportIT k tomu ale poslouží vlastní serverová aplikace.

Z využití jednoduchého JSON objektu pro ukládání dat vyplývají mimo výhody i nedostatky Firebase. Nelze ji totiž používat jako plnohodnotnou databázi, neboť například nepodporuje vyhledávání dle cizích klíčů. V některých případech je proto nutné vytvářet redundantní záznamy.

## Shrnutí

Služba Firebase má své výhody i nevýhody, pro jednoduchost tohoto projektu ovšem dostačuje a v případě potřeby by neměl být problém některá data nebo operace nad nimi přesunout do vlastní serverové aplikace. Podobný postup už byl v STRV ozkoušen i na jiných projektech.

Vybrána byla také kvůli tomu, že s ní mají vývojáři ve firmě dobré zkušenosti a nedávná akvizice společností Google podporuje tvrzení, že se na stabilitu této služby lze spolehnout i v horizontu několika let.

### 6.1.2 Serverová část – další sdílené funkce

Celé softwarové řešení je založeno na interakci uživatelů s klientskou aplikací a sdílení vložených dat. Proto nebylo potřeba implementovat mnoho funkcí ve vlastní serverové aplikaci.

Do budoucna se ale počítá s potenciálními nastavbami, a proto byla vytvořena jednoduchá aplikace v javascriptu s možností rozšíření pro případné složitější business operace. V rámci této práce jsou implementovány pouze funkce pro ukládání větších souborů (zejména obrázky) a rozesílání emailů s denními reporty klientům.

Serverová část je založena na knihovně Node.js [55], podporující jednoduché neblokující zpracování HTTP požadavků. Její výhodou je jednoduchost, rychlost a velké množství instalovatelných balíčků, obsahující různé funkcionality.

Programování v Node.js je založené právě na využívání těchto balíčků již existujících funkcionalit. Pro jejich instalaci byl využit nástroj npm [56].

Jedním z hlavních použitých balíčků je framework Express [57], který slouží jako nadstavba Node.js a zjednodušuje vytváření robustních API. Samozřejmě je také balík obsahující Firebase SDK, nebo několik menších modulů například pro podporu parsování požadavků nebo asynchronní zpracování.

### 6.1.3 iOS 8 a Swift

Klientská aplikace byla vyvinuta pro současnou verzi iOS 8.1, ve zbrusu novém jazyku Swift [58]. Ten byl oficiálně představen v létě 2014 a od té doby je stále ve vývoji.

Většina odborníků ale Swift považuje za budoucnost programování pro Apple platformy, neboť oproti Objective-C přináší mnoho vlastností moderních jazyků. Mezi ně patří zejména rychlost, výkon a bezpečnostní prvky, které vyžaduje kompilátor, aby dokázal předcházet nedefinovanému chování a pádům aplikace.

Použití tohoto nového jazyka není vynuceno, neboť lze při vývoji iOS aplikací libovolně kombinovat kód Swiftu a Objective-C. Díky tomu lze Swift použít i v rozpracovaném projektu, nebo je možné kombinovat knihovny v obou



jazycích. Některé knihovny v ReportIT jsou také napsány v Objective-C, příkladem je Firebase SDK nebo Facebook SDK.

### **Knihovny a závislosti**

Pro zavádění závislostí do projektů se nejčastěji používá populární (neoficiální, ale nejrozšířenější) balíčkovací systém CocoaPods [59]. Stejně tak je tomu i v případě tohoto projektu, kde jsou za pomoci CocoaPods přidány různé knihovny.

Mezi ně patří mimo Firebase SDK také Alamofire [60], knihovna pro zjednodušení síťové komunikace. Další jsou například knihovny pro podporu autentizace vůči službám Facebook, Google nebo GitHub.

## **6.2 Implementace iOS aplikace**

Implementace aplikace ReportIT proběhla dle požadavků výhradně v anglickém jazyce. Důvodem je v první řadě rozmanitost národností zaměstnanců STRV a zároveň i fakt, že se aplikace zaměřuje na komunitu lidí zabývajících se vývojem softwaru, mezi kterými lze angličtinu alespoň ve psané formě považovat za samozřejmost.

### **6.2.1 Struktura aplikace**

Při programování iOS aplikací je běžné používat mezi programátory notoricky známý architektonický návrhový vzor Model-View-Controller (MVC). Ten je aplikovaný i v případě ReportIT.

#### **Práce s daty a synchronizace s Firebase**

Za zajímavost lze považovat práci s daty, která je kvůli použití služby Firebase poněkud atypická. Model má dvě úrovně – databázový model a Firebase model. S databázovým modelem kontrolery pracují zejména při čtení a zobrazování dat. Firebase model se pak používá k šíření změn a synchronizaci dat v databázi s daty na Firebase serverech. Práci s daty lze tedy rozdělit na tři typy:

- Synchronizace se serverem
- Čtení a zobrazení dat
- Úpravy dat

Synchronizace se serverem probíhá jednoduše. Při zobrazení určité scény si kontroler za pomoci Firebase SDK nastaví sledování serverových událostí a dat souvisejících s danou scénou. Veškeré změny ostatních uživatelů se tedy stahují

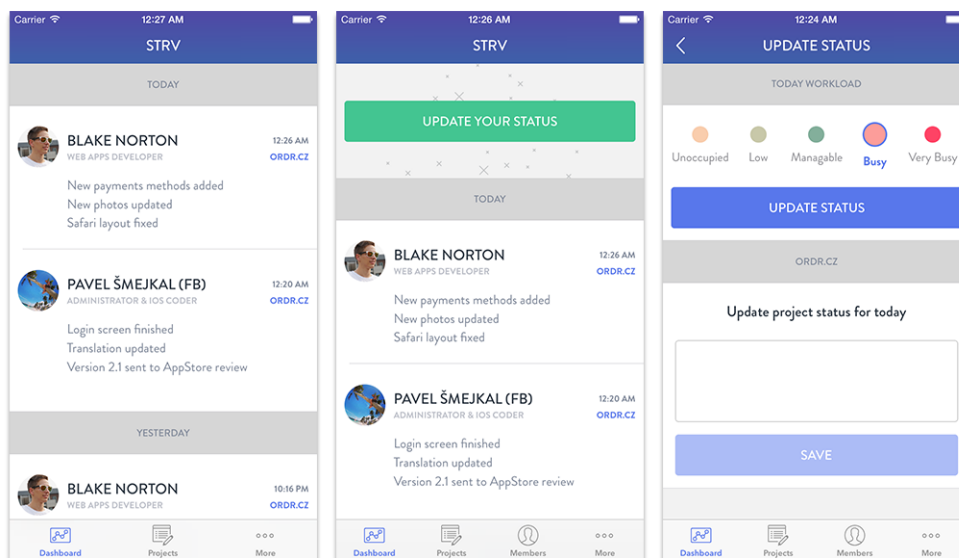
v reálném čase a ukládají do databáze. Při opouštění scény se synchronizace pozastaví.

Při čtení a zobrazování se využívají třídy nativního databázového systému Core Data [61]. Protože se data na pozadí sama aktualizují dle změn na serveru, je potřeba zobrazovat změny databáze, když nastanou. K tomu slouží komponenta NSFetchedResultsController, která umožňuje automaticky reagovat na změny v lokální databázi a tedy neustále zobrazovat aktuální data.

Požadavky na změny dat se naopak posílají přímo do Firebase. Výhodou je zejména jedna z vlastností Firebase SDK – práce offline. Do Firebase lze tedy zapisovat data i bez připojení k internetu a změny se ihned aplikují lokálně (jsou zavolány metody pro automatickou synchronizaci) a na server se propagují až když se telefon připojí k internetu.

### 6.2.2 Implementace hlavní scény aplikace

Po spuštění aplikace je vyžadováno přihlášení a výběr jedné ze společností. Poté se uživatel nachází v hlavní scéně (obrázek 6.2), kde má přístup ke všem funkcím aplikace pomocí navigačního panelu.



Obrázek 6.2: ReportIT přehled – po přihlášení klienta (vlevo) a vývojáře (uprostřed a vpravo)

Kromě podpory funkcí definovaných případy užití byl přidán první panel obsahující celkový přehled (pod anglickým názvem *Dashboard*). Ten se pro různé role nepatrně odlišuje a snaží se zobrazit ty nejpodstatnější informace pro daný typ uživatele.

### 6.2.3 Komplikace při implementaci

Při implementaci aplikace bylo potřeba vyřešit několik problémů. Hlavní překážkou byly například obtíže způsobené použitím nového jazyku Swift nebo implementační problémy spojené s automatickou synchronizací dat.

#### Nový jazyk pod neustálým vývojem

Swift je stále v rané fázi, a proto byly jen během několika měsíců hlavního vývoje vydány dvě nové verze. Každá z nich však přinesla podstatné změny, a proto bylo potřeba kód upravovat dle nových konvencí.

Se Swiftem přicházely také nové verze Xcode, které postupně zlepšovali podporu vývoje, například zlepšením grafického návrháře.

I poslední verze Swift 1.2 a Xcode 6.3 však nepřekonalý některé problémy komplikující vývoj. Mezi ně patří zejména občasné pády vývojového prostředí, pomalé překládání aplikace nebo problémy s našeptáváním.

#### Komplikace při automatické synchronizaci dat

Ze samotného programování lze za jednu z ošemetných částí považovat implementaci automatické synchronizace dat.

Hlavní nebezpečí představuje stav, kdy jsou na serveru některá data změněna nebo smazána. Může se totiž stát, že s nimi uživatel ve stejnou chvíli pracuje na zařízení, a proto je automatická synchronizace nemůže jednoduše upravit nebo smazat z databáze.

Příkladem může být situace, kdy jsou uživateli odebrána práva nebo je vyloučen ze společnosti. Synchronizační metody nemohou prostě smazat všechna data, na která nemá uživatel po změně práva, neboť s nimi může druhá část aplikace zrovna pracovat. Synchronizace dat byla proto implementována tak, že před smazáním dat odesílá upozornění ostatním částem aplikace, které se mohou připravit na smazání nebo úpravy.

---

# Testování a vyhodnocení přínosů aplikace

Tato kapitola uzavírá práci celkovým zhodnocení možností využití aplikace ReportIT. Kromě shrnutí přínosů pojednává také o testování uvnitř společnosti a plánech aplikace do budoucnosti.

## 7.1 Přínosy používání aplikace

Hlavní myšlenkou ReportIT je reportování pokroku na projektech všemi zaměstnanci, kteří se aktivně podílejí na vývoji produktu.

Základem aplikace jsou stručná každodenní hlášení, obsahující informace o tom, na čem vývojář daný den pracoval. Současně lze připojit záznam o stavu vytíženosti.

### 7.1.1 Přínosy aplikace v STRV

Ve společnosti STRV má aplikace jeden hlavní úkol – pomoci více propojit vývojový tým a klienta. Protože zde nejsou projektoví manažeři, přebírají jejich funkci částečně obchodní zástupci a částečně právě sami programátoři a grafici. Ti při vývoji komunikují přímo s klienty a snaží se vyhovět jejich požadavkům.

Používání ReportIT k hlášení pokroku na projektech by mělo standardizovat a vynutit komunikaci s klienty na denní bázi. Tím bude možné předcházet nedorozumění, neboť klient získá kontinuální přehled o vývoji jeho projektu.

Zároveň budou mít stručná výstižná hlášení pozitivní dopad na informovanost uvnitř týmu, který je v STRV distribuovaný. Každý z programátorů bude mít při zadávání reportu možnost sledovat, jak si vedou ostatní platformy.

### Ekonomické přínosy

Ekonomické přínosy aplikace bohužel nelze jednoduše vyčíslit. Vedení společnosti však očekává hlubší propojení vývoje s klienty, kteří budou moci sledovat, jak se produkt pohybuje každý den a možná i dokáží ocenit tvrdou práci programátorů a grafiků.

Pokud si zákazníci odnesou kromě skvělého produktu také dobrý pocit ze spolupráce, zvyšují se šance jejich návratu nebo doporučení pro jiné projekty, což ve výsledku znamená i reálné finanční přínosy.

#### 7.1.2 Využití mimo STRV

Aplikaci lze ale využít i ve společnostech s klasickými postupy při vývoji software. Programátoři, grafici nebo návrháři uživatelského rozhraní ji mohou využít pro reportování mezi sebou nebo svým nadřízeným.

Do nástrojů pro řízení projektů totiž zaznamenávají mnoho informací a pro jejich vedoucí pracovníky může být časově náročné sledovat, kdo na čem pracuje, nebo hledat údaje o pokroku projektu zpětně. V takových případech může stručné a výstižné denní hlášení z aplikace ReportIT najít své uplatnění.

Jinde lze aplikaci použít pro podávání hlášení prakticky jakékoliv zainteresované osobě. Tou může být mimo projektového manažera například zaměstnanec oddělení pro řízení kvality, tester nebo jiný typ vedoucího pracovníka.

## 7.2 Testování

Cílem testování aplikace je odhalení chyb a tedy zvyšování kvality produktu. V případě ReportIT probíhalo testování ve dvou fázích. Nejdříve byl ozkoušen celkový nápad aplikace a po samotné implementaci přišlo na řadu alfa testování uvnitř společnosti.

### 7.2.1 Testování myšlenky zlepšení komunikace

Při analýze komunikačních postupů a procesů byl nalezen prostor pro zlepšení komunikace mezi vývojovým týmem a klientem. Před spuštěním implementace aplikace proto začalo testování samotné myšlenky zavedení denních reportů.

Programátoři a grafici tedy dostali za úkol psát ke každému aktivnímu projektu denní hlášení popisující, na čem daný den pracovali. Hlášení pak hromadně odesílali klientům ve společném emailu pro každý projekt.

Po necelých dvou měsících se dostalo první zpětné vazby, když klient sám od sebe ocenil přístup programátorů a jejich denní reporty. Proto se tento typ komunikace stal standardem ve všech aktivních projektech.

### 7.2.2 Alfa testování aplikace ReportIT

Po implementaci hlavních funkcí probíhalo alfa testování uvnitř společnosti. Vybraná skupina vývojářů používala aplikaci v ostrém provozu, když každý den aktualizovali stav svých projektů.

Během zkoušení bylo odhaleno několik problémů, spojených zejména s právy a zobrazování dat. Ty byly následně opraveny.

Jedním z diskutovaných témat byl také čas odesílání emailů. Většina vývojového týmu pracuje na území Evropy a klienti obvykle pocházejí ze západního pobřeží USA. Časový rozdíl mezi vývojáři a zákazníkem je tedy 9 hodin. Odesílání emailů bylo proto nastaveno na 22:00 středoevropského času, kdy mají všichni vývojáři hlášení pro daný den podané a u klienta je teprve 13 hodin odpoledne.

## 7.3 Možná rozšíření aplikace

Do budoucnosti se počítá s dalším rozvojem aplikace. V první řadě bude implementována podpora ostatních platforem, zejména webová aplikace.

Na základě výsledků alfa testování budou dále přidány některé funkcionality, například možnosti zadání časového pásma vývojového týmu a klienta, nebo přímo nastavení času, kdy se mají odesílat reporty.

Po globálním testování uvnitř společnosti by pak mohl být produkt uveden na trh jako služba. Poté by bylo možné přidávat další rozšíření za předpokladu, že si získá oblibu u uživatelů. Mezi těmi by mohlo být například posílání zpráv mezi klientem a vývojářem, nebo komentáře k jednotlivým reportům. Pro spolupráci uvnitř společnosti by mohly být implementovány funkce pro další využití dat jako je plánování práce apod.

---

# Závěr

Cílem této práce bylo analyzovat nástroje pro podporu komunikace a kooperace při vývoji softwarových produktů se zaměřením na malé a střední podniky a dále na základě analýzy procesů ve vybrané společnosti navrhnout, implementovat, otestovat a zhodnotit iOS aplikaci dle jejich potřeb.

Nejprve bylo potřeba definovat komunikační procesy při vývoji software a jejich význam dle dostupné literatury. Díky tomu bylo možné identifikovat typy nástrojů pro podporu těchto procesů.

Třetí kapitola provádí analýzu trhu pro každý typ nástroje. Zaměřuje se na obecné vlastnosti produktů a porovnává jejich odlišné parametry. Nástrojů existuje velké množství a pro podporu typických komunikačních procesů lze nalézt vhodné řešení prakticky pro jakýkoliv podnik, ať už je jejich prioritou cena, funkce softwaru, možnosti vlastní správy nebo jiné.

Na základě analýzy ve vybrané firmě ve čtvrté kapitole lze však konstatovat, že společnosti mohou mít specifické potřeby na funkce nástrojů pro podporu spolupráce. V daném podniku se například vývoj odehrává v atypickém prostředí založeném na distribuovaných týmech, kde proti klasickému vývojovému procesu chybí zaměstnanci v roli projektových manažerů.

Za nejvýznamnější příležitost ke zlepšení byl identifikován proces komunikace s klientem ve fázi samotné implementace produktu, což částečně vyplývá právě z absence projektových manažerů. Na základě analýzy požadavků popsaných v páté kapitole byl proto navržen nástroj ReportIT pro podporu tohoto procesu.

Šestá a sedmá kapitola pojednává o samotné implementaci, testování a hodnocení řešení, které se skládá z jednoduché serverové části a klientské aplikace pro mobilní platformu iOS.

## Nástroj ReportIT

Hlavní myšlenkou ReportIT je zaznamenávání pokroku na projektech všemi členy vývojového týmu. To vychází z podávání stručných každodenních hlášení

---

s informacemi o tom, na čem vývojář daný den pracoval.

Tyto informace jsou pak přístupné nejen uvnitř společnosti, ale také klientovi prostřednictvím aplikace nebo emailů. Ten díky nim získá kontinuální přehled nad projektem a může poskytovat zpětnou vazbu ihned.

Pro možnosti sdílení dat a centralizaci další logiky byl systém implementován architekturou typu klient/server. Klientskou částí je iOS aplikace programovaná v moderním jazyce Swift. Základ serverové části představuje služba pro datovou synchronizaci Firebase [54] ve spolupráci s vlastní serverovou aplikací implementovanou v javascriptovém frameworku Node.js [55].

### **Testování a vyhodnocení přínosů**

Testování probíhalo uvnitř daného podniku ve dvou úrovních. Nejdříve byla ozkoušena samotná idea podávání reportů a poté také iOS aplikace.

Myšlenka každodenních hlášení prokázala svou přidanou hodnotu, když získala pozitivní ohlas i přímo od klientů, kteří si pravidelné reporty pochvalují. Testování aplikace odhalilo drobné nedostatky, které byly záhy opraveny. Zároveň byly identifikovány potenciální vylepšení pro další rozvoj.

Hlavním přínosem používání aplikace jsou spokojenější zákazníci a další možnosti práce s daty uvnitř společnosti. Ačkoliv tyto dopady nelze jednoduše vyčíslit, vedení společnosti předpokládá, že hlubší propojení vývoje s klienty a lepší komunikace bude mít i reálné ekonomické přínosy.

### **Možná rozšíření ReportIT**

Do budoucnosti se počítá s dalším rozvojem tohoto nástroje. Hlavní prioritou bude podpora ostatních platforem a po následném testování uvedení nástroje na trh. Následujícím krokem bude implementace dalších rozšíření aplikace identifikovaných při interním testování. Příkladem může být plnohodnotný chat mezi zaměstnanci a klientem nebo komentování příspěvků.



---

## Literatura

- [1] Communication – definition and more from the free Merriam-Webster dictionary. 2015, [Přístup 12.3.2015]. Dostupné z: <http://www.merriam-webster.com/dictionary/communication>
- [2] Lowensohn, J.: Skype shown automatically translating multilingual voice calls. 2014, [Přístup 12.3.2015]. Dostupné z: <http://www.theverge.com/2014/5/27/5756166/microsofts-skype-translator-will-translate-voice-calls-on-the-fly>
- [3] Forms of Communication. [Přístup 13.3.2015]. Dostupné z: <http://communicationtheory.org/forms-of-communication/>
- [4] *Essentials of Mass Communication Theory*. SAGE Publications, 1995.
- [5] Survey of Communication Study/Chapter 11 - Organizational Communication. Leden 2015, [Přístup 12.3.2015]. Dostupné z: [http://en.wikibooks.org/w/index.php?title=Survey\\_of\\_Communication\\_Study/Chapter\\_11\\_-\\_Organizational\\_Communication&oldid=2756595](http://en.wikibooks.org/w/index.php?title=Survey_of_Communication_Study/Chapter_11_-_Organizational_Communication&oldid=2756595)
- [6] Mefalopulos, P.: *Development Communication Sourcebook : Broadening the Boundaries of Communication*. World Bank Publications, Červen 2008.
- [7] Project Cartoon"Work Projects Really Work. [Přístup 14.3.2015]. Dostupné z: <http://projectcartoon.com/cartoon/3>
- [8] Tree Swing Pictures. [Přístup 14.3.2015]. Dostupné z: <http://www.businessballs.com/treeswing.htm>
- [9] Švendová, V.: Metodika agilního vývoje softwaru na OVSS ÚVT. *Zpráva ÚVT MU*, 2011. Dostupné z: <http://ics.muni.cz/bulletin/articles/668.html>

- 
- [10] What is the Agile Software Development Methodology. 2015, [Přístup 14.3.2015]. Dostupné z: <http://agilemethodology.org>
- [11] Knesl, J.: Agilní vývoj: Úvod. [Přístup 17.3.2015]. Dostupné z: <http://www.zdrojak.cz/clanky/agilni-vyvoj-uvod/>
- [12] Hummel, M.; Rosenkranz, C.; Holten, R.: *The Role of Communication in Agile Systems Development: An Analysis of the State of the Art*. Springer Science Business Media, Říjen 2013.
- [13] Media Richness Theory. [Přístup 17.3.2015]. Dostupné z: [http://www.utwente.nl/cw/theorieenoverzicht/TheoryClusters/MassMedia/Media\\_Richness\\_Theory/](http://www.utwente.nl/cw/theorieenoverzicht/TheoryClusters/MassMedia/Media_Richness_Theory/)
- [14] Media Richness Theory. [Přístup 18.3.2015]. Dostupné z: [http://www.campus-adr.net/ODRmodule/media\\_richness\\_theory.html](http://www.campus-adr.net/ODRmodule/media_richness_theory.html)
- [15] Cejthamr, V.: *Management a organizační chování - 2., aktualizované a rozšířené vydání*. Grada Publishing a.s., 2010.
- [16] Ambler, S. W.: Communication on Agile Software Teams. [Přístup 14.4.2015]. Dostupné z: <http://www.agilemodeling.com/essays/communication.htm>
- [17] Ramasubbu, N.: *Essays on managing distributed software development*. Dizertační práce, University of Michigan, 2006.
- [18] Mayfield, K. M.: *Project managers' experience and description of decision uncertainty associated with the agile software development methodology: A phenomenological study*. Dizertační práce, Capella University, 2010.
- [19] Scrum Methodology - Scrum Product Owner. [Přístup 19.3.2015]. Dostupné z: <http://scrummethodology.com/scrum-product-owner/>
- [20] Knesl, J.: Agilní vývoj: Scrum. [Přístup 20.3.2015]. Dostupné z: <http://www.zdrojak.cz/clanky/agilni-vyvoj-scrum/>
- [21] Johnson, B.; Watts, G.: Scrum Role Playing. [Přístup 21.3.2015]. Dostupné z: <https://www.scrumalliance.org/community/articles/2008/september/scrum-role-playing>
- [22] Orlova, A.: 10 Ways to Improve Communication Between Software Developers. Červen 2013, [Přístup 12.3.2015]. Dostupné z: <http://blog.azoft.com/communication-within-software-development-company/>
- [23] Beck, K.: *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 2004.

- 
- [24] Definice malého a středního podnikatele. 2015, [Přístup 23.3.2015]. Dostupné z: <http://www.czechinvest.org/definice-msp>
- [25] ProtonMail – Encrypted Email Made Simple. 2015, [Přístup 23.3.2015]. Dostupné z: <https://protonmail.ch>
- [26] Biggs, J.: ProtonMail Is A Swiss Secure Mail Provider That Won't Give You Up To The NSA. Červen 2014, [Přístup 23.3.2015]. Dostupné z: <http://techcrunch.com/2014/06/23/protonmail-is-a-swiss-secure-mail-provider-that-wont-give-you-up-to-the-nsa/>
- [27] Private group chat, video chat and instant messaging for teams – HipChat. 2015, [Přístup 29.3.2015]. Dostupné z: <https://www.hipchat.com>
- [28] Reynolds, A.: Atlassian Acquires HipChat, Group Chat and Collaboration Service. 2012, [Přístup 28.3.2015]. Dostupné z: <http://blogs.atlassian.com/2012/03/press-release-atlassian-acquires-hipcha/>
- [29] Etherington, D.: HipChat Launches One-To-One Video, Audio And Screensharing For All Users. Květen 2014, [Přístup 21.3.2015]. Dostupné z: <http://techcrunch.com/2014/05/06/hipchat-launches-one-to-one-video-audio-and-screensharing-for-all-users/>
- [30] Lardinois, F.: Atlassian Launches HipChat Server, Doubles Down On Team Collaboration. Leden 2015, [Přístup 21.3.2015]. Dostupné z: <http://techcrunch.com/2015/01/27/atlassian-launches-hipchat-server-doubles-down-on-team-collaboration/>
- [31] Slack: Be less busy. 2015, [Přístup 12.3.2015]. Dostupné z: <https://slack.com>
- [32] Bedřich, V.: Slack oslavil první rok existence a vyrostl na 500 000 uživatelů denně – čtěte jeho celý příběh. Únor 2015, [Přístup 11.3.2015]. Dostupné z: <http://www.czechcrunch.cz/2015/02/slack-oslavil-prvni-rok-existence-a-vyrostl-na-500-000-uzivatelu-denne-ctete-jeho-cely-pribeh/>
- [33] CrunchBase - world's most comprehensive dataset about startup activity. [Přístup 11.3.2015]. Dostupné z: <https://www.crunchbase.com/organization/tiny-speck>
- [34] Ptáček, M.: 8 měsíců starý startup Slack raisuje přes 2,4 miliardy Kč. Říjen 2014, [Přístup 12.3.2015]. Dostupné z: <http://www.czechcrunch.cz/2014/10/8-mesicu-stary-startup-slack-raisuje-pres-2-miliardy-kc/>

- 
- [35] Google+ Hangouts. 2015, [Přístup 24.3.2015]. Dostupné z: <http://www.google.com/+/learnmore/hangouts/>
- [36] Flowdock: Group chat for teams. 2015, [Přístup 22.3.2015]. Dostupné z: <https://www.flowdock.com>
- [37] Williams, A.: Rally Software Buys Flowdock, A Real-Time Social Collaboration Hub For Developers. Únor 2013, [Přístup 22.3.2015]. Dostupné z: <http://techcrunch.com/2013/02/13/rally-software-buys-flowdock-a-real-time-social-collaboration-hub-for-developers/>
- [38] Skype | Free calls to friends and family. 2015, [Přístup 24.3.2015]. Dostupné z: <http://www.skype.com/en/>
- [39] Yammer: Enterprise Social network. 2015, [Přístup 24.3.2015]. Dostupné z: <https://www.yammer.com>
- [40] BRIGHT, P.: Microsoft buys Skype for \$8.5 Billion. Why, exactly? 2011, [Přístup 25.3.2015]. Dostupné z: <http://www.wired.com/2011/05/microsoft-buys-skype-2/>
- [41] AnyMeeting – Free web conferencing software. 2015, [Přístup 26.3.2015]. Dostupné z: <http://www.anymeeting.com>
- [42] Adobe web conferencing software | Adobe Connect. 2015, [Přístup 25.3.2015]. Dostupné z: <http://www.adobe.com/products/adobeconnect.html>
- [43] Trello. 2015, [Přístup 29.3.2015]. Dostupné z: <https://trello.com>
- [44] Basecamp is everyone's favorite project management app. 2015, [Přístup 28.4.2015].
- [45] JIRA – Issue and project tracking software | Atlassian. 2015, [Přístup 8.4.2015]. Dostupné z: <https://www.atlassian.com/software/jira>
- [46] Overview – Redmine. 2015, [Přístup 29.3.2015]. Dostupné z: <http://www.redmine.org>
- [47] GoToMeetings – Easy Online Meetings with HD video. 2015, [Přístup 23.4.2015]. Dostupné z: <http://www.gotomeeting.com/online/>
- [48] GitHub. 2015, [Přístup 21.4.2015]. Dostupné z: <https://github.com>
- [49] Google Drive – Cloud Storage File Backups for Photos, Docs More. 2015, [Přístup 25.4.2015]. Dostupné z: <https://www.google.com/drive/>

- 
- [50] Asana – teamwork without email. 2015, [Přístup 28.4.2015]. Dostupné z: <https://asana.com>
- [51] Dropbox. 2015. Dostupné z: [dropbox](https://dropbox.com)
- [52] Xcode – The complete toolset for building great apps. 2015, [Přístup 8.4.2015]. Dostupné z: <https://developer.apple.com/xcode/downloads/>
- [53] Atom – A hackable text editor for the 21st Century. 2015, [Přístup 8.4.2015]. Dostupné z: <https://atom.io>
- [54] Firebase – Build Realtime Apps. 2015, [Přístup 8.4.2015]. Dostupné z: <https://www.firebaseio.com>
- [55] Node.js. 2015, [Přístup 8.4.2015]. Dostupné z: <https://nodejs.org>
- [56] npm. 2015, [Přístup 9.4.2015]. Dostupné z: <https://www.npmjs.com>
- [57] Express – Node.js web application framework. 2015, [Přístup 9.4.2015]. Dostupné z: <http://expressjs.com>
- [58] Swift – Overview – Apple Developer. 2015, [Přístup 9.4.2015]. Dostupné z: <https://developer.apple.com/swift/>
- [59] CocoaPods.org. 2015, [Přístup 9.4.2015]. Dostupné z: <http://cocoapods.org>
- [60] Thompson, M.: Alamofire – Elegant networking in Swift. 2015, [Přístup 9.4.2015]. Dostupné z: <https://github.com/Alamofire/Alamofire>
- [61] Introduction to Core Data Programming Guide. 2014, [Přístup 11.4.2015]. Dostupné z: <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CoreData/cdProgrammingGuide.html>

## Seznam použitých zkratek

<b>USD</b>	United States Dollar
<b>NCA</b>	National Communication Association
<b>IM</b>	Instant Messaging
<b>SCRUM</b>	Software Methology and Project Management
<b>XP</b>	Extreme Programming
<b>QA</b>	Quality Assurance
<b>UX</b>	User Experience
<b>CI</b>	Continuous Integration
<b>SLA</b>	Service Level Agreement
<b>SSO</b>	Single sign-on
<b>CRM</b>	Customer Relationship Management
<b>API</b>	Application Programming Interface
<b>IRC</b>	Internet Relay Chat
<b>MB</b>	Megabyte
<b>NAS</b>	Network Attached Storage
<b>VPN</b>	Virtual Private Network
<b>MVP</b>	Minimal Viable Product
<b>HTTP</b>	Hypertext Transfer Protocol
<b>HTTPS</b>	HTTP Secure, HTTP over SSL

---

**SDK** Software Development Kit

**JSON** JavaScript Object Notation

**REST** Representational State Transfer

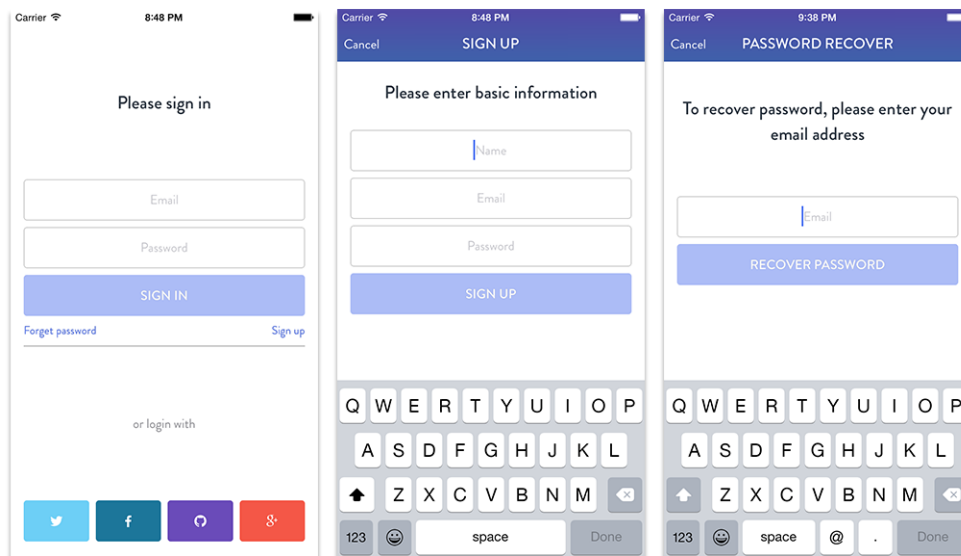
**MVC** Model View Controller

## Ukázky aplikace

Obrázek B.1 ukazuje možnosti po spuštění aplikace. Uživatel se může přihlásit pomocí služeb třetích stran nebo po registraci prostřednictvím emailu a hesla. V případě zapomenutí hesla mu systém nabídne odeslání nového na email.

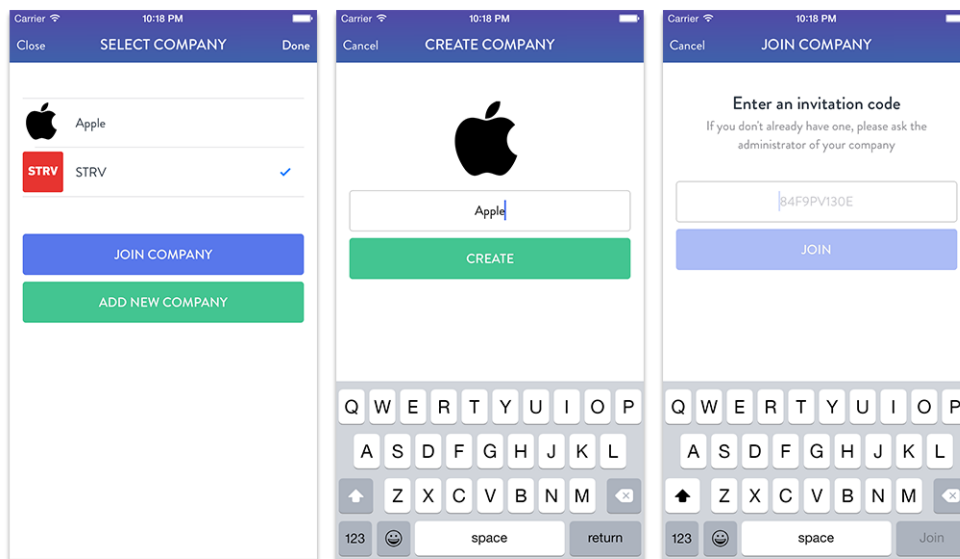
Po přihlášení má uživatel při výběru společnosti možnost buď vytvořit novou, připojit se k existující pomocí pozvánky v podobě kódu nebo vybrat jednu z již připojených (obrázek B.2). Do této nabídky se také může dostat z menu aktivní společnosti, viz. obrázek B.3.

Po výběru aktivní společnosti se uživatel ocitne na hlavní scéně. Obrázek B.3 ukazuje možnosti úpravy uživatelského profilu v záložce *More*. Kromě



Obrázek B.1: ReportIT – přihlášení do aplikace

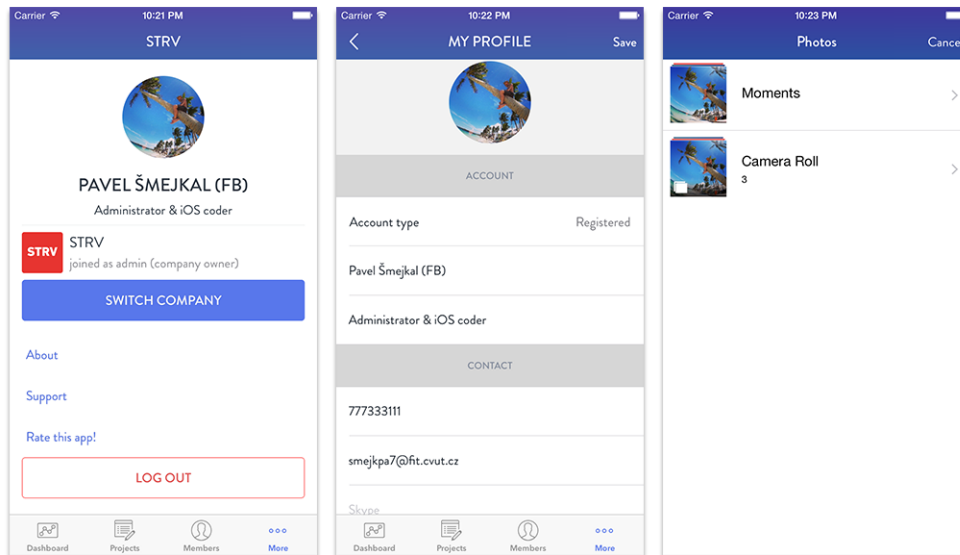




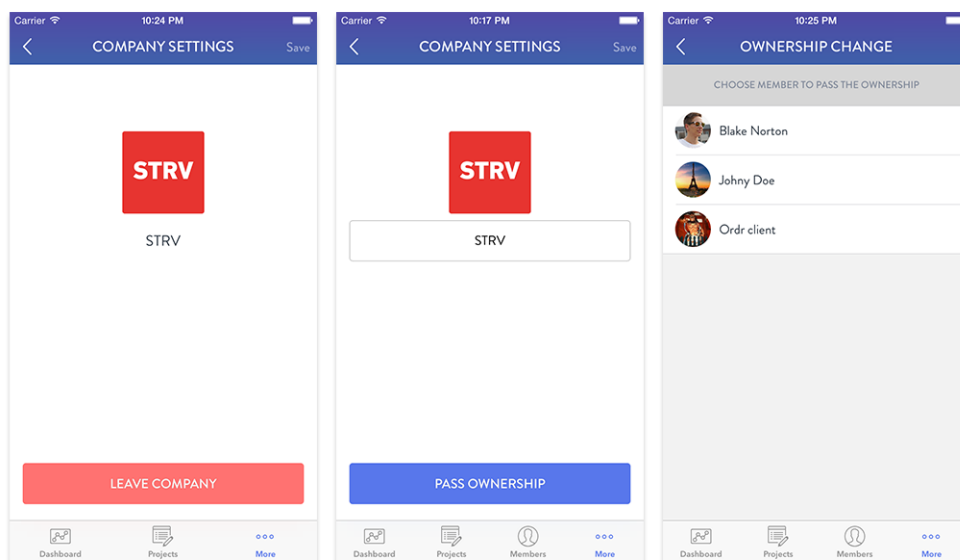
Obrázek B.2: ReportIT – výběr aktivní společnosti

jména a pozice může také ukládat kontaktní informace nebo profilovou fotografií.

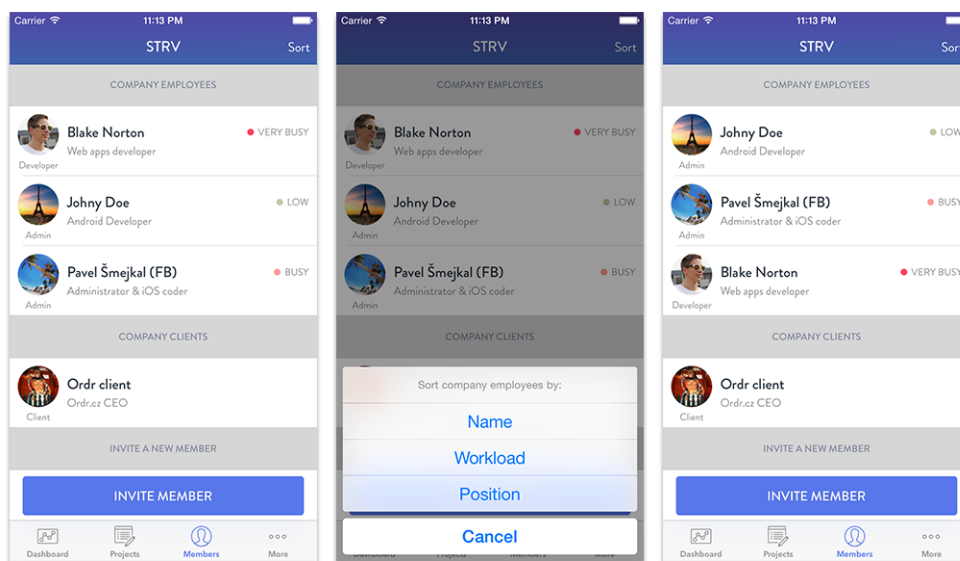
Ze záložky *More* je také možné přejít do nastavení společnosti. Obrázek B.4 ukazuje možnosti klasického uživatele (vlevo), který může společnost pouze opustit. Administrátor pak může měnit logo a název společnosti (uprostřed). Pokud je navíc vlastníkem, může vlastnictví předat jinému členovi (vpravo).



Obrázek B.3: ReportIT – úprava profilu uživatele



Obrázek B.4: ReportIT – správa aktivní společnosti

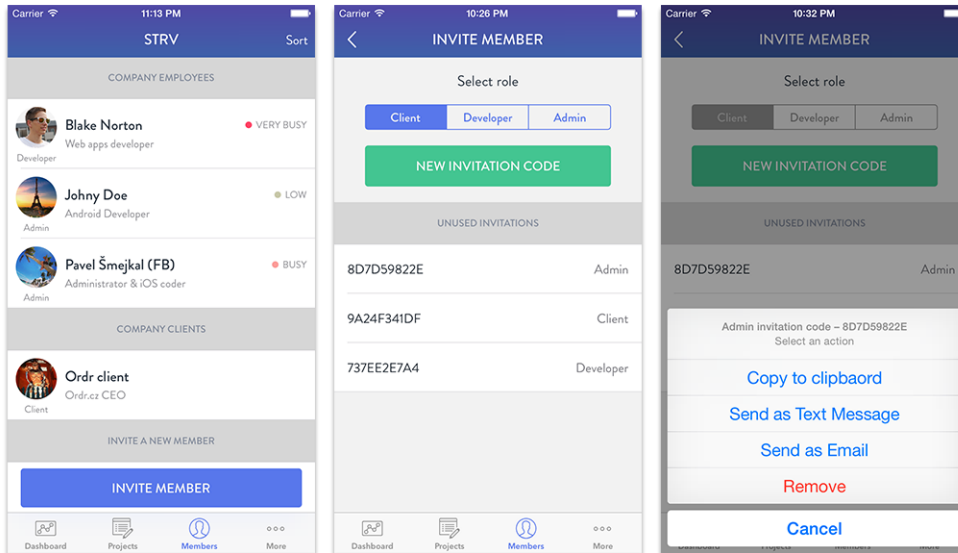


Obrázek B.5: ReportIT – procházení zaměstnanců společnosti

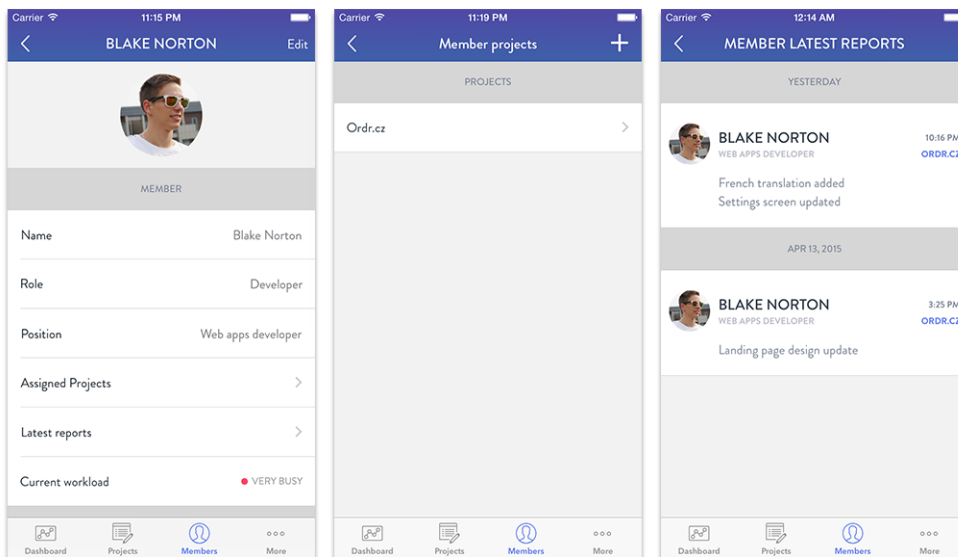
Záložka *Members* obsahuje seznam členů společnosti, tedy zaměstnanců a klientů. Ten je přístupný pouze interním uživatelům a lze jej řadit dle různých atributů, viz. obrázek B.5.

Administrátor má možnost pozvat nové uživatele pomocí generovaných pozvánkových kódů (obrázek B.6).

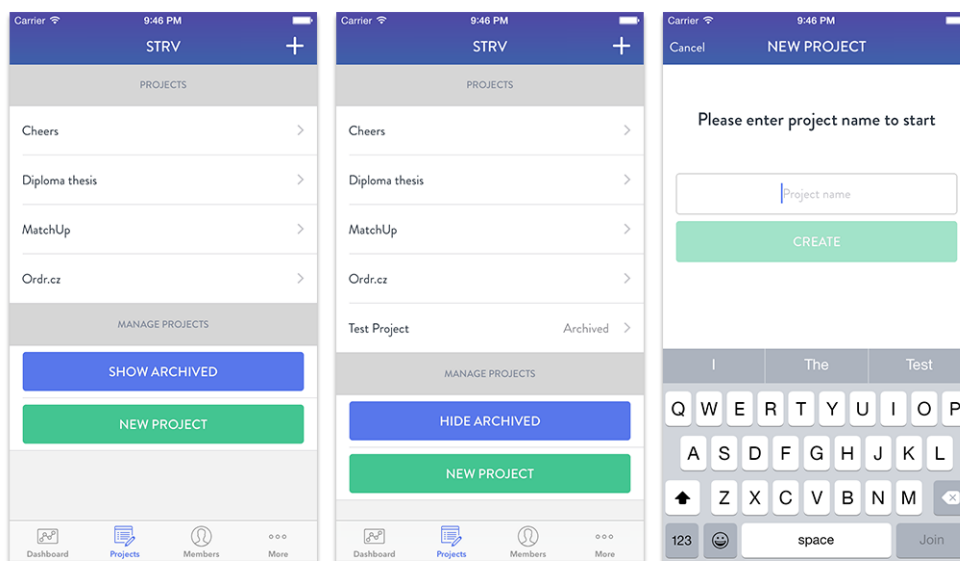
Obrázek B.7 zobrazuje detail člena společnosti, který v případě administrátora obsahuje kromě profilových informací také seznam přiřazených projektů nebo nedávná hlášení.



Obrázek B.6: ReportIT – správa členů společnosti



Obrázek B.7: ReportIT – detail člena společnosti

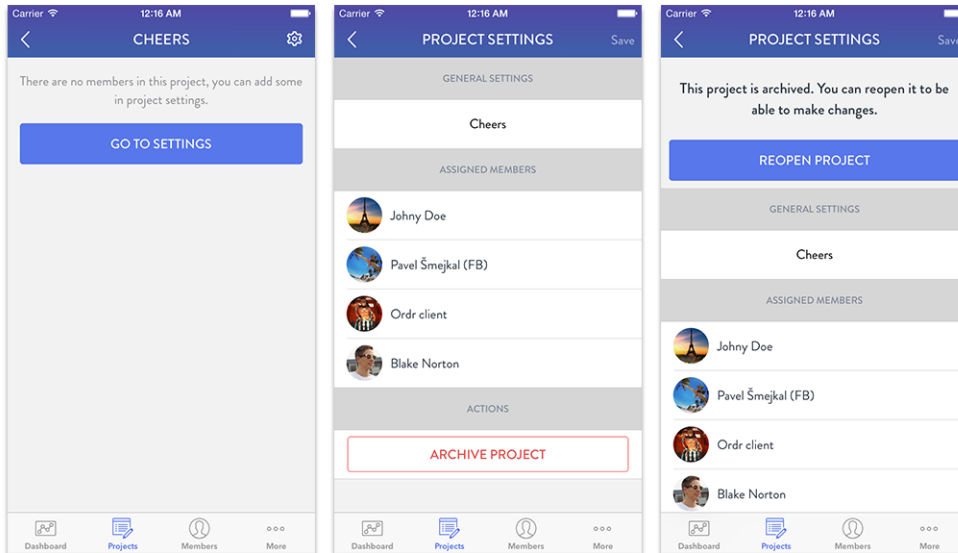


Obrázek B.8: ReportIT – procházení a vytváření projektů

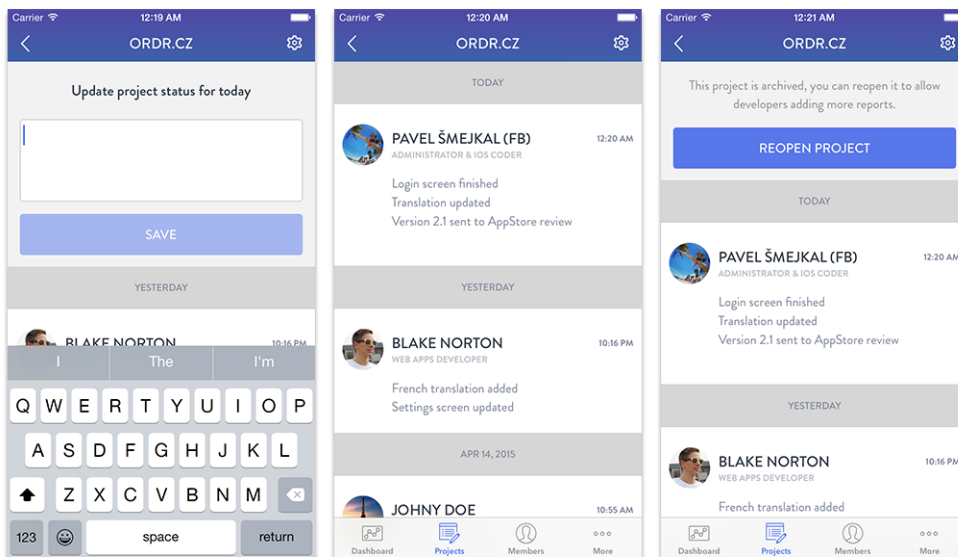
Do záložky *Projects* mají přístup všichni uživatelé. Klienti tam mohou procházet hlášení dle jejich projektů. Administrátor má možnost zakládat nové projekty (obrázek B.8).

U vytvořených projektů může administrátor měnit jejich nastavení. To spočívá v přiřazování členů – ostatní role totiž vidí pouze projekty sobě přiřazené. Dále může měnit název nebo archivovat a případně znovuotevřít projekt, viz. obrázek B.9.

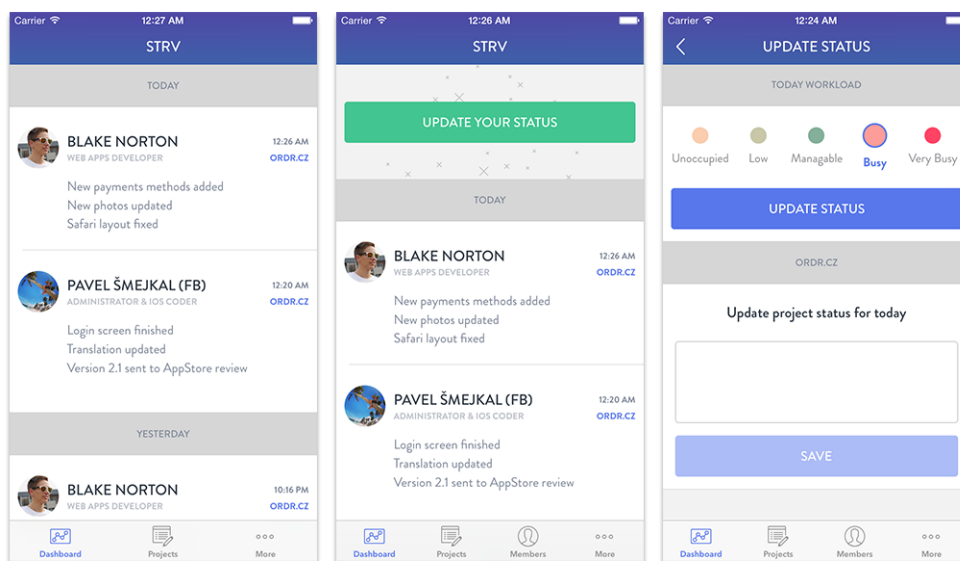
Obrázek B.10 ukazuje detail projektu s jednotlivými hlášeními. V případě přiřazených uživatelů v zaměstnaneckých rolích (administrátor/vývojář) umožňuje přidávat reporty pro daný den za předpokladu, že je projekt otevřen.



Obrázek B.9: ReportIT – nastavení projektu



Obrázek B.10: ReportIT – detail projektu



Obrázek B.11: ReportIT – dashboard

Záložka *Dashboard* se pro různé role nepatrně odlišuje a zobrazuje ty nejpodstatnější informace pro daný typ uživatele. V případě vývojáře tak umožňuje jednoduché zadání statusu na aktivních projektech, viz. obrázek B.11.

---

# Uživatelská příručka

## C.1 Pokyny ke spuštění zkompilevané verze aplikace

Zkompilevaná verze aplikace ReportIT se nachází na přiloženém CD ve složce *bin*. Tu lze kvůli omezení Apple iOS spustit pouze na simulátoru. Aplikace pro mobilní platformu iOS lze totiž na zařízení spustit, pouze pokud byla zkompilevána s vývojářským certifikátem, který je přímo spojený s daným zařízením.

Proto je ke spuštění potřeba počítač s operačním systémem Mac OS a nainstalovaným vývojovým prostředím. Aplikaci lze poté spustit otevřením *ipa* archivu v simulátoru.

## C.2 Pokyny ke zkompilevání aplikace

Zdrojové kódy aplikace ReportIT se nachází na přiloženém CD ve složce *src/impl*. Jedná se o projekt v Xcode 6.3 [52], který podporuje současnou verzi Swift 1.2.

Projekt už obsahuje všechny závislosti zavedené pomocí balíčkovacího systému CocoaPods [59]. Stačí proto otevřít soubor *ReportIT.xcworkspace*, zkompilevat kód a spustit aplikaci na simulátoru nebo zařízení.



---

## Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	bin.....	adresář se spustitelnou formou aplikace
	screenshots .....	adresář s obrázky z aplikace
	src	
	impl .....	zdrojové kódy aplikace
	thesis .....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
	text .....	text práce
	thesis.pdf .....	text práce ve formátu PDF pro el. čtení
	thesis_print.pdf .....	text práce ve formátu PDF (pro tisk)