

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	ZS 2018/2019
Autoři	Zdenka Šunková sunz00, Miroslav Mareš marm04, Martin Kulíček kulm00
Téma	Using Checklists to Organize Software Development Processes
Datum odevzdání	21. 12. 2018

Abstrakt

Tato práce se věnuje využíváním kontrolních seznamů při vývoji softwaru. V první části je obecně vysvětlen význam kontrolního seznamu, jeho výhody a možnosti použití. Dále už se práce zaměřuje na konkrétní části vývoje softwaru a zapojení kontrolních seznamů do nich. Podle známých „best practises“ pro tuto práci je vytvořen jednoduchý kontrolní seznam. V poslední části jsou popsány 3 nejpoužívanější digitální nástroje pro správu kontrolních seznamů, kde jsou popsány jejich funkce, vychytávky a možnosti.

Klíčová slova

kontrolní seznam, vývoj softwaru, checklist

Obsah

1. Úvod	2
1.1. Cíl práce	2
1.2. Struktura práce	2
2. Kontrolní seznam	3
2.1. Výhody.....	3
2.2. Jak kontrolní seznam použít	4
3. Využití kontrolních seznamů při vývoji SW	4
3.1. Kdy kontrolní seznam použít.....	5
3.1.1. Kontrolní seznam pro přípravu projektu	5
3.1.2. Kontrolní seznam pro vývoj.....	6

3.1.3. Kontrolní seznam pro provoz a údržbu.....	6
3.2. Příklad návrhu kontrolního seznamu pro vývoj SW dle „best practises“	7
4. Nástroje pro kontrolní seznamy.....	7
4.1. GoogleSpread Sheet	8
4.2. Todoist.....	8
4.3. Jira.....	10
5. Závěr	11

1. Úvod

V dnešní uspěchané době si ani profesionálové a odborníci nemohou dovolit dělat chyby a své schopnosti a znalosti musejí stále rozvíjet. Aktuální toto je například při vývoji softwaru, kde každá chybička může znamenat prodloužení času a také zvýšení nákladů. Zde přicházejí právě kontrolní seznamy, které mohou být novodobými hrdiny a chránit projekty před selháním. Kontrolní seznamy nám jsou věrným pomocníkem, abychom nic nezapomněli, nepřehlédli a vždy věděli co nás ještě čeká a nemine. Práce se zabývá využívání těchto seznamu při vývoji softwaru. A proč je to tak důležité? Vývoj softwaru ve většině případech je rozsáhlý projekt, který má hned několik částí a pracuje na něm několik týmů. Je velmi náročné dodržet časový harmonogram, splnit všechny potřebné kroky, udržet kontrolu nad průběhem vývoje a také komunikovat mezi sebou. Kontrolní seznamy s tímto vším mohou pomoci a tím zjednodušit a zefektivnit práci celému týmu.

1.1. Cíl práce

Hlavním cílem je uvést čtenáře do tématu využití kontrolních seznamů při vývoji softwaru, za pomoci využití “best practises” a příkladů z praxe.

Dalším cílem pak je i zdůraznit důležitost využívání kontrolních seznamů při práci na rozsáhlých projektech a taky poukázat na výhody jeho užívání.

1.2. Struktura práce

Semestrální práce je rozdělena na tři části. V první části je jednoduše a stručně představen kontrolní seznam, jeho využití a výhody. V části druhé, která je rozsáhlejší, je popsáno a vysvětleno využívání kontrolního seznamu při vývoji softwaru. Tato část

také obsahuje možný návrh kontrolního seznamu vytvořeného podle “best practises”. Poslední část se věnuje konkrétním nástrojům na správu kontrolních seznamů, které se v praxi při vývoji softwaru využívají.

2. Kontrolní seznam

Co je kontrolní seznam, neboli anglicky checklist, není potřeba detailněji rozebírat. Stačí stručně říci, že se jedná o seznam otázek nebo bodů, které charakterizují nějaký stav nebo průběh v určité činnosti nebo procesu. Takový seznam obsahuje podrobně vše, co se plánuje, na co by se nemělo zapomenout a čeho je potřeba dosáhnout. Kontrolní seznam tak může být využit v jakémkoliv odvětví a pro projekty s různým zaměřením.

Kontrolní seznamy jsou hojně využívány hlavně v odborných oblastech jako je lékařství, strojírenství a také softwarový vývoj. Jejich funkce není jen dodržení správných a přesných kroků, ale také eliminace chyb, možnost sdílení důležitých bodů s členy týmu a je to skvělý nástroj na udržování kontroly.

I když se může zdát, že kontrolní seznam je triviální věc, jeho význam spočívá také v udržení na paměti jednoduchých, avšak důležitých kroků, které by jinak váš mozek vytěsnil. (Yung 2018)

2.1. Výhody

Jednou velkou a bezpochybnou výhodou je **úspora času a peněz**. Čas, který věnujete vytvoření kontrolního seznamu může být jen zlomek času, který můžete strávit na opravě chyb, které se staly důsledkem absence seznamu.

Někdo by si mohl říci, že není potřeba vytvářet žádný kontrolní seznam pomocí papíru či určené aplikace, že vše má ve své hlavě. Při základních činnostech to tak může být, ale při velkých rozsáhlých projektech je potřeba svůj mozek využívat na složitější úkoly. Uvolnění mozku od základních informací nazýváme **Brainovo uvolnění paměti**.

Důležitým aspektem je také funkce **udržování disciplíny**. Pro obory jako je například letectví nebo lékařství to je klíčový atribut, protože se jedná o lidský život. Pro

projektového manažera pracujícím na vývoji softwaru je kontrolní seznam skvělým nástrojem na vštěpení disciplíny celému týmu. (Yung 2018)

2.2. Jak kontrolní seznam použít

Je samozřejmostí, že kontrolní seznam jde použít hned na několik způsobů a při různých příležitostech. Stavět dům? Učit se na státnice? To vše jsou životní události, kde člověk může využít jednoduchý kontrolní seznam s hlavním cílem, aby se záměr podařilo dokončit úspěšně. Pokud nebudeme zabrušovat do jednotlivých odvětví a budeme brát kontrolní seznamy jako obecnou věc, můžeme nalézt 3 základní významy.

Prvním z nich je **upomínkový systém**. Jedná se úplně o základní seznam bodů a informací, které nesmějí být zapomenuty, které se musí splnit a díky kterým můžeme dojít k našemu cíli. Tyto body mohou být shlukovány do jednotlivých částí anebo jeden bod může mít více podbodů.

Druhou možností pohledu je **hodnotící systém**. Když už máme hotov seznam bodů a milníků v průběhu času jednotlivé části můžeme hodnotit. Nejjednodušší hodnocení je stylem splněno/nesplněno nebo při používání aplikací to může být zaškrtnuto políčko nebo ne. Avšak nemusíme se spokojit jen s dvěma odpověďmi, je také možnost využít hodnocení jako “v procesu” ne “částečně dokončeno”.

Poslední je pak **vyhodnocující systém**. Díky tomu, že jsme v předchozí části hodnotili jednotlivé body, můžeme pak vyhodnotit celý projekt a zjistit tak jeho úspěšnost. Kontrolní seznam může mít hned několik částí, můžeme vyhodnotit jen některé z nich. Například vždy když se jedna část zhodnotí, může se vyhodnotit její úspěšnost a díky tomu zjistíme informace, které nám nebyly před tím známy a další části tomu můžeme přizpůsobit. (Lepil kina 2018)

3. Využití kontrolních seznamů při vývoji SW

V této části se práce zaměříme konkrétně jakými způsoby lze kontrolní seznamy využít při vývoji softwaru. Pokoušíme se soustředit především, jak mohou kontrolní seznamy napomoci ke zkvalitnění průběhu jednotlivých fází a snížení rizik při vývoji SW.

Ovšem je nutné mít na paměti, že okolnosti použití jsou velmi různorodé. Každá společnost a každé vývojové prostředí je unikátní, a proto nelze striktně nastavit, že právě ten či onen kontrolní seznam bude tím, kterým se má vaše společnost nebo váš projekt řídit. Dále budeme vycházet s tzv. „*best practises*“, které jsou čerpány z několika zdrojů včetně osobních zkušeností.

Kontrolní seznamy jsou využívány jak při vývoji SW za použití rigorózních metodik, tak při použití agilních metodik. Lze tedy říci, že kontrolní seznamy nejsou nikterak závislé na druhu zvolené metodiky.

Existují doporučené konvence, kterými by se tvorba kontrolního seznamu měla řídit. Definovaný seznam by měl být co nejjednodušší a zároveň by měl pokrývat všechny potřebné aspekty pro jednotlivé fáze vývoje. (McLennan, 2016)

3.1. Kdy kontrolní seznam použít

Kontrolní seznamy lze použít pro jakýkoliv úkol, malý nebo velký. Můžeme je využít pro něco jednoduchého jako je například: kontrola barvy tlačítka (tlačítko je červené). Nebo pro komplexnější úlohu například: požadavek návrhu uživatelského rozhraní. Kontrolní seznamy lze tedy využít v jakékoliv fázi životního cyklu vývoje softwaru.

Pokusíme se jednotlivě projít jednotlivé etapy vývoje softwaru a u každé se obecně zaměříme, co by bylo dobré, aby se v jednotlivé fázi kontrolovalo a následně další kapitole uvádíme příklad, jak by mohl takový kontrolní seznam vypadat.

3.1.1. Kontrolní seznam pro přípravu projektu

Specifikace projektu: pro tuto fázi je dobré kontrolovat specifika projektu. Jedná se především o přehled náležitostí daného projektu, zda je naplněna podstata projektu. Zda projekt rozdělen do fází, a každá jeho fáze má jasně definovaná pravidla, jakým způsobem bude řízena a spravována. Dále je možné kontrolovat dokumentaci projektu, zda dodržuje nastavená pravidla a mnoho dalších aspektů vázané k projektu. Jedná se tedy o celkový pohled na projekt formou kontrolního seznamu.

Cenová nabídka: v této části je vhodné kontrolovat, zda jsou naplněny všechny aspekty spojené s financováním projektu.

Start projektu: v této fázi dochází k samotné kontrole těsně před spuštěním projektu. Měly by docházet k finální kontrole, zda jsou splněny všechny podstatné podmínky pro započítání s projektem (podepsaná smlouva, existence vývojových týmů, kontaktní osoby zákazníka, termíny ...)

3.1.2. Kontrolní seznam pro vývoj

Základní principy: v této části je vhodné kontrolovat konvenci a design. Jedná se o čitelnost kódu a jeho částí a dodržování dohodnutého notace v daném kódování znaků (například: UTF – 8).

Sledování úkolů: slouží ke kontrole, zda tým využívá některých z nástrojů pro sledování úkolů (tracking systém) a zda je jeho využití smysluplné a přináší přehlednost v přiřazování úkolů a zobrazování stavu rozpracovaných úloh.

Kontrola zdrojového kódu: slouží především pro softwarové vývojáře, kdy jsou schopni pomocí kontrolních seznamů sjednotit svou práci v týmu tak, aby napsaný kód byl čitelný i pro ostatní členy týmu. Například zde můžeme zařadit kontrolu odřádkování, psaní komentářů do struktury kódu, nebo kontrolu využívání verzovacích nástrojů. (KOVÁCS 2018)

Testování: v této fázi lze kontrolovat, zda navržený systém je pokryt testy, u komponent, které jsou pro systém kritické. Zda existují automatické testy nebo jestli je systém testován na všech úrovních (jednotkové testy, funkční testy, integrační testy, systémové testování a akceptační testy). (Hlava 2011)

3.1.3. Kontrolní seznam pro provoz a údržbu

Pro tento kontrolní seznam je dobré specifikovat kontrolní body pro následné udržování systému. Můžeme zde zmínit například, zda je určena odpovědná osoba pro provoz či údržbu aplikace, zdali bude aplikace monitorována případně jestli je nadefinován scénář při poruše aplikace (KOVÁCS 2018)

3.2. Příklad návrhu kontrolního seznamu pro vývoj SW dle „best practises“

Naštěstí existuje velké množství úspěšných SW projektů, u kterých se můžeme inspirovat z použitých kontrolních seznamů. Můžeme se vždy poučit z přístupu, který byl používán, a to může pro nás být velice užitečné, abychom zajistili, že náš software bude vyvíjen dle poznatků z ostatních projektů (*best practises*). (ARORA 2017)

V tabulce 1 je vyobrazen krátký kontrolní seznam, který má za účel modelově ukázat, jak by mohl vypadat kontrolní seznam pro části vývoje SW, kde dochází k psaní zdrojového kódu a následnému testování.

Tabulka číslo 1: Příklad kontrolního seznamu. Zdroj: autor

ANO NE N/A	Psaní zdrojového kódu	Komentáře
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1. Vygenerovaný podrobný návrh modulů (komponent).	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2. Kódování dodržuje dohodnutou notaci.	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	3. Každý z požadavků uvedených v dokumentu <i>Definice požadavků</i> je navázán na kód.	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	4. Obsahuje každý blok kódu komentář?	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	5. Je každá nová část kódu před nahráním na úložiště otestována (automatizovanými testy)?	
ANO NE N/A	Provedení testů	Komentáře
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1. Všechny moduly byly otestovány jednotkovými testy.	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	2. Konečné verze plánu integrace a testování systému byly dodány sponzorovi projektu a uživatelům ke schválení.	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	3. Veškeré testované materiály, včetně jednotkových testovacích vstupů, výstupů, výsledků a chybových protokolů, jsou uchovávány ve složce XYZ.	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	4. Došlo v průběhu testování k použití automatizovaných testů?	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	5. Došlo k opravení všech kritických chyb během testování?	

4. Nástroje pro kontrolní seznamy

Nejjednodušším způsobem, jak vytvořit kontrolní seznam je obyčejná tužka a papír. Za pár minut tak lze vytvořit kontrolní seznam na cokoliv. Takovéto řešení je však

vhodné pouze pro malé osobní checklisty, protože vytvořený seznam není digitální a nelze ho nijak s nikým sdílet, aniž bychom ho někomu fyzicky nepředali nebo nepřepsali. To znamená že verzování a sdílení seznamu je se zvyšujícím se počtem lidí složitější a benefit kontrolního pomocníka se ztrácí.

Proto existují programy, které buďto podporují tvorbu checklistů, nebo se jimi speciálně zabývají.

4.1. GoogleSpread Sheet

Druhou nejjednodušší volbou po tužce a papíru je online software od společnosti Google. Jedná se o jejich variaci office balíčku od Microsoftu, ve kterém je mimo jiné právě aplikace GoogleSpread Sheet.

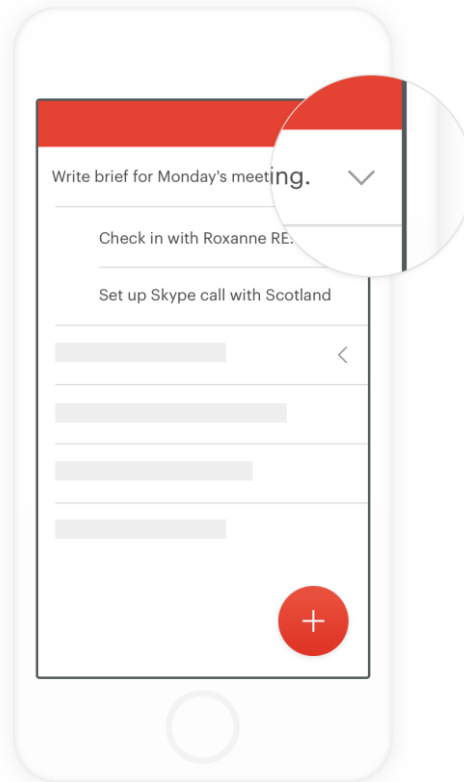
Tato aplikace vystupuje jako osekáná online verze excelu, ve které lze vytvářet kontrolní seznamy pomocí vytvoření seznamu činností v jednom sloupci a modifikace políček na zaškrťovací – checky, ve sloupci druhém.

Takto vytvořený seznam lze libovolně sdílet v rámci celého internetu pomocí sdílení linku, který na soubor odkazuje. Soubor je uložený na Google disku člověka, který seznam vytvořil a může upravovat jeho viditelnost a práva vůči veřejnosti a jednotlivým členům týmu.

GoogleSpread Sheet je služba poskytovaná zcela zdarma, k jeho využívání je třeba pouze Google účtu. Jeho jedinou nevýhodou je fakt, že se stále jedná o primitivní checklist, bez žádného vyššího smyslu a užitku.

4.2. Todoist

Todoist představuje sofistikovanější způsob pro tvorbu checklistů a práci s nimi. Na rozdíl od GoogleSpread Sheetu se speciálně zaměřuje na jejich využívání v rámci činností a řízení projektů.

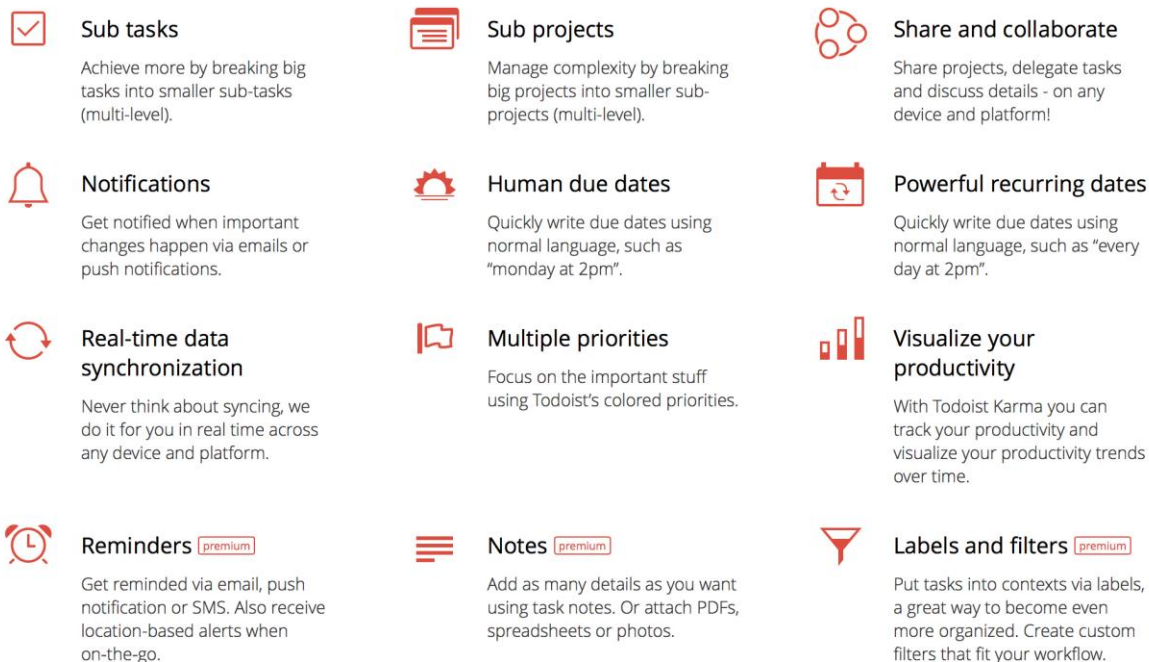


Obrázek číslo 1: Ukázka aplikace Todoist (ANON. 2018b)

Lze ho využívat buďto v rámci jednotlivých činností anebo vytvářením celých projektů plných činností. U každé činnosti lze nastavit, jak je či není důležitá, její návaznosti na další a také způsoby a frekvenci notifikací o progresu. Vše je online synchronizováno skrze cloudové služby.

Nejzajímavější částí Todoistu je však umělá inteligence. Ta na základě chování uživatele dokáže predikovat, jak dlouho mu podobná činnost bude trvat v budoucnosti. Toto plánování není omezeno pouze na data samotného uživatele, ale lze použít také data jiných uživatelů, kteří někdy prováděli podobnou činnost a získat tak ještě přesnější výsledek.

Díky tomu lze v týmu koordinovat činnosti dle návyků, zvyků a pracovní morálky jednotlivých členů týmu a odhad doby trvání činností a celého projektu má mnohem reálnější hodnoty než u konkurenčních programů. Umělou inteligenci lze také využít k tvorbě grafů, která ukáže efektivnost projektu a jeho pracovníků.



Obrázek číslo 2: Ukázka aplikace Todoist (ANON. 2018b)

Dále nabízí Todoist i integraci dat z celkem momentálně 73 různých aplikací jako je například Google kalendář, Dropbox nebo Amazon Alexa. Lze tak používat Todoist jako centrální systém pro plánování skrze různé aplikace, které uživatel běžně využívá.

Program vyvíjí firma Doist a je momentálně dostupný na více jak 10 platformách včetně těch nejzákladnějších jako je Windows, macOS, iOS, Android a další.

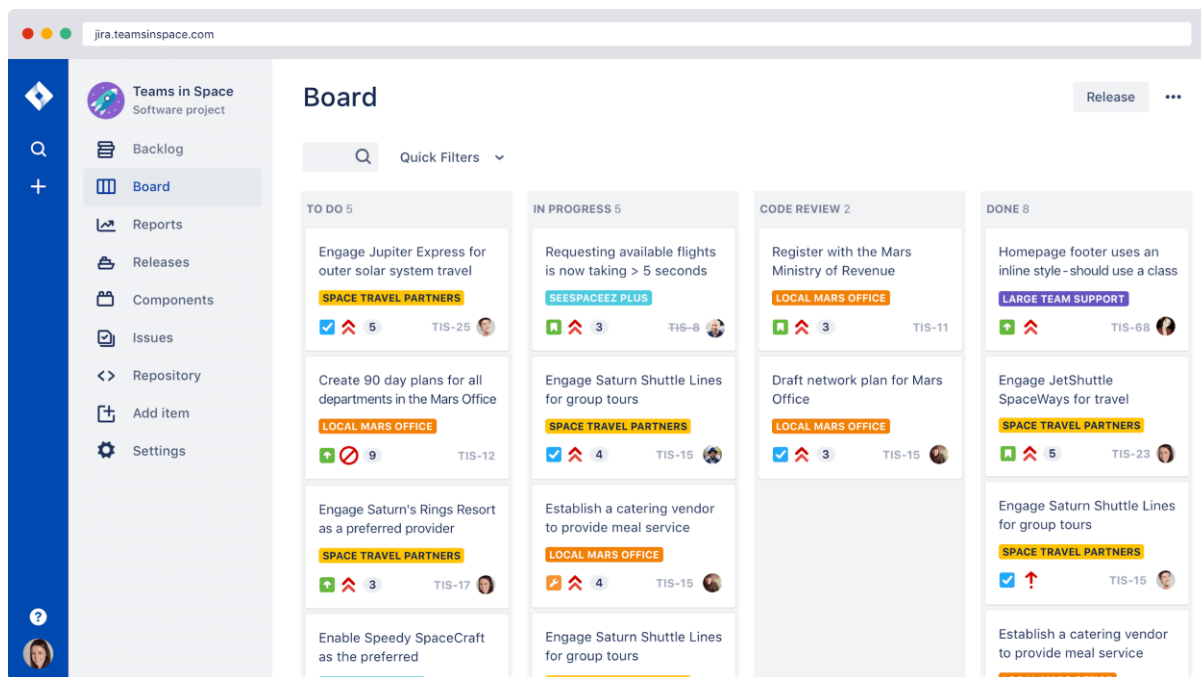
Cenově je nabízen ve třech verzích, které se mezi sebou liší funkcemi. Nejzákladnější osobní nekomerční verze je nabízena zcela zdarma avšak počet projektů a činností je velmi omezen. Stejně tak tato verze neobsahuje chytré plánování a synchronizaci a spoustu dalších funkcí. Střední varianta je nabízena za 36 euro ročně a nejvyšší variantu lze pořídit za 60 euro ročně.

Mezi známé společnosti, které Todoist používají patří například HP, Cisco nebo Gillete. (Kilián 2016)

4.3. Jira

Jira je řešení od firmy Atlassian, které obsahuje mnoho balíčků, mezi kterými je i řízení agilních projektů vývoje softwaru. Tento balíček se jmenuje Jira Software a funguje

skrze cloud. Dalšími balíčky jsou například Jira Ops pro incident management, nebo Jira Core pro základní správu firmy.



Obrázek číslo 3: Ukázka aplikace Jira (ANON. 2018a)

Jira funguje podobně jako Todoist, ale bez umělé inteligence a podporuje méně platform. Oproti tomu má ale vlastní obchod plný nejrůznějších doplňků, které si lze k Jiře doinstalovat.

Jedním z těchto doplňků je právě i funkce kontrolních seznamů, které Jira sama o sobě nepodporuje. Tento konkrétní doplněk se jmenuje *Checklist for Jira*.

Jira je nabízena ve dvou základních verzích, které mají zcela stejné funkce, ale liší se v umístění cloudu, který Jira pro svůj chod používá. Tyto dvě základní verze se dále rozdělují podle počtu uživatelů, kteří Jiru aktivně využívají a tím pádem je cena dynamická. (ANON. 2018a)

5. Závěr

V závěru práce přichází čas na shrnutí dosažených cílů práce a naskytnutých problémů. Jediným omezením bylo nalezení kvalitních zdrojů. Kontrolní seznamy jsou vnímány jako triviální věc během vývoje softwaru, a proto se tomuto tématu literatura ve velkém měřítku nevěnuje. I přesto se podařilo nalézt dostatek odborných článků, díky kterým práce seznamuje čtenáře s danou problematikou. V první části se

práce věnuje vysvětlení významu kontrolních seznamů a jejich výhodám, díky tomu čtenář pochopí důležitost využívání tohoto nástroje. V další části už se naplňuje hlavní cíl práce, a to hlubší pochopení problematiky využívání kontrolních seznamů při vývoji softwaru, kde jsou popsány jednotlivé části vývoje softwaru a zapojení kontrolních seznamů do nich. Návrh možného kontrolního seznamu vychází z „best practises“ a může tak být pro čtenáře inspirací. V poslední části jsou představeny nástroje na správu kontrolních seznamů, které se využívají v praxi a jsou nejpoužívanější v dnešní době.

Zdroje:

ANON., 2018a. *Jira | Software pro sledování požadavků a projektů | atlassian* [online] [vid. 2018-12-20]. Dostupné z: <https://cs.atlassian.com/software/jira>

ANON., 2018b. *Todoist – The Best To Do List App & Task Manager* [online] [vid. 2018-12-20]. Dostupné z: <https://todoist.com/>

ARORA, 2017. *15 best practices for software development projects - GeeksProgramming* [online] [vid. 2018-12-20]. Dostupné z: <https://www.geeksprogramming.com/best-practices-for-software-development-projects/>

HLAVA, Tomáš, 2011. *úrovně Testování | Testování softwaru. testovanisoftwareu* [online] [vid. 2018-12-20]. Dostupné z: <http://testovanisoftwareu.cz/tag/urovne-testovani/>

KILIÁN, Karel, 2016. *Aplikace Todoist přináší umělou inteligenci, která vám naplánuje úkoly* [online] [vid. 2018-12-20]. Dostupné z: <https://www.svetandroida.cz/aplikace-todoist-umela-inteligence/>

KOVÁCS, Kristóf, 2018. *Software project best practices checklists :: Software architect Kristof Kovacs* [online] [vid. 2018-12-20]. Dostupné z: <https://kkovacs.eu/software-project-best-practices-checklist>

LEPILKINA, Diana, 2018. *Types of QA Checklists in Software Development - DZone Agile* [online] [vid. 2018-12-20]. Dostupné z: <https://dzone.com/articles/types-of-qa-checklists-in-software-development>

MCLENNAN, Liam, 2016. *Agile Development Checklist - CodeProject. codeproject* [online] [vid. 2018-12-20]. Dostupné z: <https://www.codeproject.com/Articles/13114/Agile-Development-Checklist>

YUNG, Zakhar, 2018. *Using Checklists to Organize Software Development Processes. infoq* [online] [vid. 2018-12-20]. Dostupné z: <https://www.infoq.com/articles/checklists-software-processes>