Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	ZS 2018/2019
Autoři	Patrik Fedor, fedp01 Martin Matějka, xmatm82 Jan Tlustý, tluj00
Téma	Peer Reviews Either Sandbag or Propel Agile Development
Datum odevzdání	21.12.2018

Abstrakt

Téma této práce je o tom, jak správně provádět vzájemnou kontrolu práce, u čím dál více populárního agilního vývoje. Provádění kontinuálního a často se měnícího zadání projektu si žádá dělat průběžnou kontrolu, zda byli jednotlivé úkoly porozuměny a provedeny správně.

Vývoj probíhá v jednotlivých iteracích, které představují určitý časový úsek například týdenní iterace. Na začátku každé iterace jsou definovány úkoly a hlavní cíle, kterých se má v tomto sprintu dosáhnout. Sprint označuje probíhající aktuální iteraci v projektu. V jednotlivých sprintech by měla proběhnout pro každý splněný úkol nějaká forma vzájemné kontroly práce. Tato vzájemná kontrola práce je důležitou složkou při agilním vývoji a má velký vliv na úspěšnost projektu, jelikož se odráží v mnoha aspektech ve výsledném produktu, jakožto v jeho kvalitě.

Jelikož je vzájemná kontrola tak důležitá je potřeba ji provádět správně. Takové správné provádění není tak jednoduché, jak se může na první pohled zdát, jelikož častým objektem kontroly nejsou, přesně definovatelné správné a špatné řešení, jako je to například u testů ve škole. Je proto potřeba si definovat co by mělo být kontrolováno a jak u jednotlivých úkolů.

Při špatném provádění kontroly může dojít, že taková činnost se stane pouze *přítěží*, která bere při vývoji čas a jejímž výstupem nejsou relevantní, či adekvátní připomínky. Tento problém je v titulku zmíněk jakožto tzv. “sandbag”, neboli v překladu pytel s pískem, který představuje zmíněnou přítěž.

Klíčová slova

- Peer review
- Agilní vývoj
- Kontrola
- Posuzovatel

Obsah

.

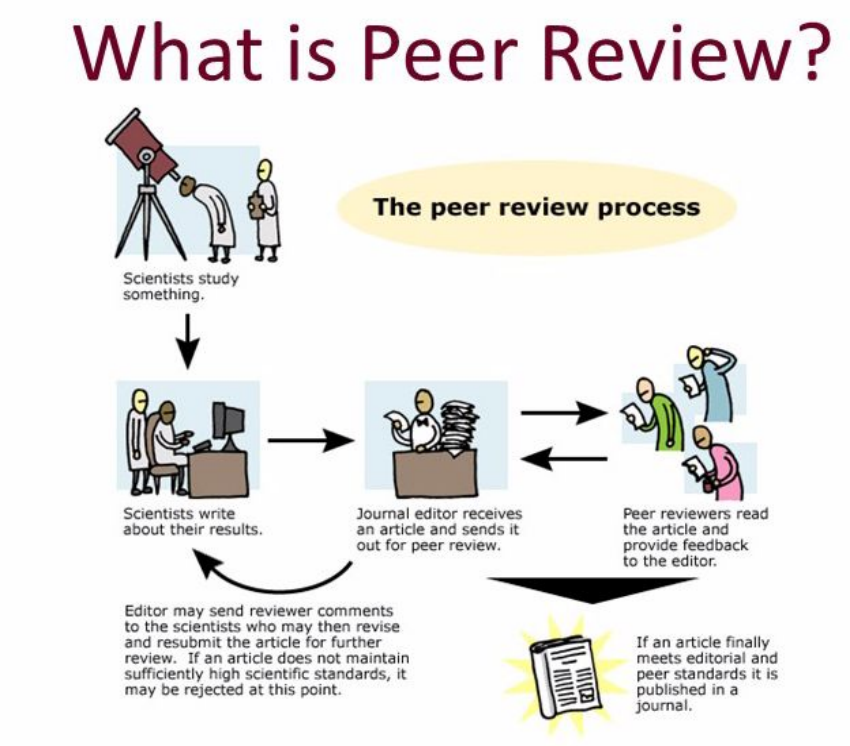
Úvod

Cílem této práce je uvést čtenáře do problematiky peer review při vytváření kvalitního produktu se zaměřením se na peer review při agilním vývoji. Jelikož je nejefektivnější a také o něco lehčí určitou problematiku vysvětlovat na konkrétních případech a také poněvadž s trendem používání agilního vývoje se můžeme setkat převážně u vývoje informačního softwaru, bude se i tato práce se tímto směrem ubírat.
V práci by jsme chtěli čtenáře obeznámit s problematikou správného provádění peer review, kontroly kódu, vysvětlit principy, zásady, výhody a potenciální problémy a celkově uvést čtenáře do dané problematiky.

Co je peer review

Peer review v oblasti vývoje softwaru znamená postup, při kterém je již napsaný zdrojový kód počítačového programu kontrolován dalšími osobami, které jej nevytvářely, za účelem identifikace možných nedostatků, vylepšení kódu a ověření, že kód je napsán tak, aby splňoval business požadavky zadavatele.

Přestože někteří programátoři poukazují na to, že takové kontroly zaberou příliš mnoho času, všeobecně převládá názor, že toto je vyváženo pozitivy, které peer reviewing přináší. To jsou zejména menší množství chyb v kódu, méně situací, kdy je potřeba určitou věc dělat od začátku znovu a lepší komunikace a soudržnost týmu (TechRepublic, 2001).



Obrázek č.1 - Co je to vzájemná kontrola?

Pracovní postupy

Peer review se obvykle skládá z následujících částí (Elsevier, 2018):

Příprava

Tato fáze vyžaduje, aby byl kontrolovaný kód kompletní a podle programátora, který ho vyvíjel, připravený ke kontrole. Je nutné, aby účastníkům peer review byly k dispozici všechny komponenty, které jsou kontrolovaným kódem ovlivněny. To zahrnuje zejména různé druhy dokumentace, testovací případy, funkční požadavky a akceptační kritéria.

Účastníci peer review by se měli seznámit s kódem a ostatními materiály před samotným hodnocením, aby byli připraveni se účastnit diskuze ohledně jednotlivých bodů a diskuze byla co nejpřínosnější.

Kontrola

Kontroly jsou obvykle plánovány podle potřeb konkrétních firem. Jejich četnost závisí na rychlosti, jakou je vytvářen nový kód. Zastoupení jednotlivců v jednotlivých reviews závisí na velikosti týmu. Například v týmu tří programátorů se pravděpodobně každé review budou účastnit všichni členové. Ve větších týmech je možné určovat požadovanou účast jednotlivců na jednotlivých reviews na základě jejich specializace, zkušeností a míry obeznámení s konkrétním tématem. Diagram průběhu kontroly je na obrázku č.2, níže.

Pokud je to možné, každé review by se měl zúčastnit programátor, který je autorem kontrolovaného kódu, dva posuzovatelé, zapisovatel a vedoucí. Posuzovatelé vznášejí připomínky a kladou otázky programátorovi. Programátor musí být připraven na ně reagovat. Příklady otázek, které mohou být programátorovi kladeny, jsou následující:

- Jak přesně tato část kódu vyřeší daný business požadavek?
- Nebyla by v tomto případě vhodnější *case* konstrukce, než použité *if-else*.

Je důležité, aby všichni zúčastnění byli připraveni diskutovat o připomínkách ostatních otevřeně a bez předsudků. Programátoři musí být připraveni přijímat připomínky a návrhy na změny a nesmí je považovat za upozorňování na jejich chyby. Posuzovatelé musí naopak svoje poznámky a připomínky formulovat tak, aby nevyznívaly jako osobní kritika. Vedoucí rozhoduje o tom, jaké připomínky budou přijaty a zapracovány do kódu. Vedoucímu tato sezení zároveň slouží k tomu, aby si udržel přehled o pracích na projektu. Každého sezení by se měl rovněž účastnit zapisovatel zodpovědný za zdokumentování průběhu sezení.

Sezení by měla být zahájena tím, že vedoucí shrne jeho cíle a určí pravidla, podle kterých bude postupováno. Poté programátor stručně představí jím napsaný kód, sdělí, čeho mělo být dosaženo, jaké požadavky jsou tímto kódem splněny, a jaké komponenty systému jsou dotčeny. Jednotliví účastníci vznášejí otázky, komentáře a připomínky. Programátor na ně reaguje a vysvětluje, jaké důvody vedly k tomu, že se rozhodl pro jím zvolenou implementaci.

Follow-up

Z dlouhodobého hlediska je nutné sledovat, zda peer reviews plní svůj účel a zda jsou důsledně dodržována pravidla, která byla v počáteční fázi nastavena. Přínos peer review bude snížen, pokud rozhodnutí dosažená při jednotlivých sezeních nebudou aplikována, a v dlouhodobém horizontu bude celý systém peer review programátory považován za neefektivní plýtvání časem.

Dalším klíčovým faktorem úspěšnosti peer review je důsledné uchovávání záznamů o již proběhlých sezeních. Ty by měly sloužit jako podklady pro prosuzovatele pro přípravu dalších sezení. Je z nich možné zjistit, jaké typy otázek a komentářů byly vzneseny nejčastěji a čemu je v budoucnu vhodné věnovat zvýšenou pozornost. Pro vedoucího mohou posloužit jako inspirace, jakým způsobem je možné zlepšit způsob, jakým jsou sezení vedena.

Peer review při vývoji softwaru

Každý process vzájemné kontroly má stejný základní průběh. Autor napíše nový kód nebo upraví stávající. Potom adekvátní osoba, nebo osoby které provádění kontrolu, projdou změny v kódu a snaží se najít určité nedostatky, či chyby a zaznamenají je. Autor si vezme jejich zpětnou vazbu k srdci a snaží se jim vyhovět. Existuje několik známých typů kontrol, které se trochu liší svým průběhem a načasováním.

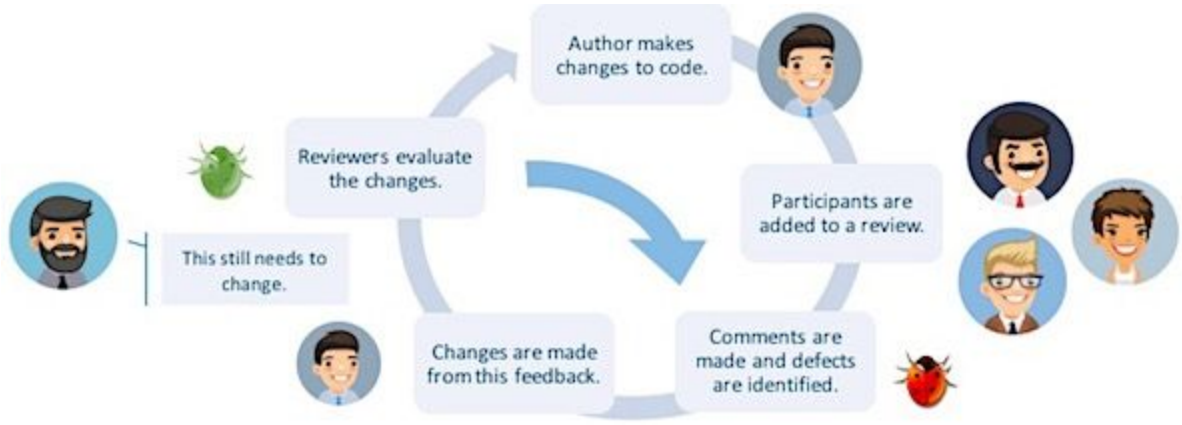
- Pre-commit
- Post-commit
- Ad-hoc
- Meeting-based
- Tool-based

Bez ohledu na to o jaký typ vzájemné kontroly se jedná, měl by každý správně strukturovaný process v agilním vývoji zahrnovat tyto tři kritické prvky:

1. Vyjasnit očekávání a určit výslovně pravidla, která se budou dodržována

Pokud si někdo řekne o kontrolu jeho práce o co přesně žádá? Žádá o názor na jeho implementované řešení a popřípadě, jak ho architektonicky zlepšit, nebo chce zkontrolovat, že se v jeho kódu nenalézají žádné fatální chyby. Kvalitní zpětnou vazbu jsme schopni mnohem rychleji získat pokud jsou předem vyjasněny její představy. Ve strukturovaném procesu vzájemné kontroly je žádáno po posuzovateli, aby posoudil kvalitu x, y, a z. Na předmětech, co jednotlivé znaky znamenají si musí stanovit tým sám (Je kód dostatečně srozumitelný? Dostatečně okomentovaný? Dostačující architektura?).

Takový proces peer review v rámci teamu není na škodu upřesnit pomocí diagramu.



Obrázek č.2: Peer review process

2. Průběh a výsledky kontroly by měly být všem dostupný

Agilnímu vývoji přidalo také na popularitě to, že pro plachého programátora není pohodlné neustále požadovat po někom případnou kontrolu, nebo aktivně sdělovat někomu výsledky jeho práce. Díky zavedenému strukturovanému procesu, nemusí se ke každé činnosti s někým osobně kontaktovat, ale prostě jej automaticky vykonat a spoléhat na to, že je případná osoba také obeznámena následným postupem.

3. Process kontroly by měl být automatizovaný process

K podpoře zautomatizování procesu vzájemné kontroly existuje dnes již spousta softwarů, které jsou schopny podpořit určitou funkcionalitou každou činnost procesu review. Příklady určitých nástrojů jsou zmíněny v kapitole Nástroje pro sledování procesu.

Nástroje pro sledování procesu

Thumbs up / down
Line of Code
Crucible - Atlassian's on-premise code review tool.
Gerrit - Open source git code review tool originating out of Google.
GitHub - Git hosting and pioneer of the "Pull Request".
LGTM - Automated Git code review for GitHub and Bitbucket pull requests for finding security vulnerabilities and code quality issues.
Phabricator - Open source git/mercurial/svn code review tool originating out of Facebook.
PullRequest - Code review as a service for GitHub pull requests.
Pull Reminders - Automated Slack reminders and metrics for GitHub pull requests.
Reviewable - Code review tool built on top of GitHub pull requests.
Review Board - Open source review tool that is SCM/platform neutral.
Rubberduck - Browser extension to adds code-aware navigation to GitHub pull requests.
Sider - Automated code review service for GitHub.
Upsource - JetBrain's on-premise git/mercurial/perforce/svn code review tool.

Výhody a nevýhody peer review

Vzhledem k tomu, že vývoj softwaru směřuje k agilnímu přístupu, dochází k častějším vydání softwarových verzí. Pokud společnosti nejsou schopni urychlit své cykly kontrol softwaru společně s rychlostí vývoje, pravděpodobně obětují jeho kvalitu, aby dosáhli termínů. To se pak projevuje nárůstem technického dluhu. Peer review je nedílnou součástí aktuálně poměrně populárního agilního vývoje (Londa, 2018) a poskytuje následující výhody:

- **Nový pohled na věc -** Hodnotitelé zachytí věci, které vy ne. Poskytnou jiný pohled na věc. Velmi důležité v případě, že programátoři nemají mnoho znalostí v odvětví, do něhož software vyvíjejí.
- **Větší povědomí o projektu napříč všemi členy týmu** - Jednotliví členové týmu si průběžně udržují přehled o důležitých událostech na projektu. To je velmi důležité proto, aby byli v případě nutnosti (nemoc, úmrtí..) schopni co nejsnadněji převzít práci po kolegovi.
- **Větší šance, že bude chyba odhalena včas** - Nový pohled na věc zároveň poskytuje širší spektrum chyb, které je peer review schopné odhalit. Tedy i menší šanci, že se tyto chyby budou vyskytovat v hotovém produktu.
- **Efektivnější komunikace -** Proces peer review vynucuje v týmu určitou míru komunikace a tato komunikace je velmi přínosná pro celý projekt, zejména proto, že důležité věci jsou diskutovány včas.
- **Užší vztahy v týmu díky intenzivnější komunikaci** - Intenzivnější komunikace pomáhá utvářet těsnější vazby mezi členy týmu.
- **Sdílení znalostí v týmu** - Při identifikování potencionálních nedostatků v kódu nutně dochází k přenesení znalostí od hodnotitele k autorovi kódu.
- **Základ kvality** - Pro týmy, které pracují ve vysoce regulovaných průmyslových odvětvích, mohou být peer reviews nezbytnou součásti projektu pro zajištění kvality softwaru.

Nevýhody

Vzájemné kontroly jsou zejména časově náročné. Nutné přípravy na správné zavedení do praxe a s tím související počáteční frustrace zaměstnanců. Zaměstnanci jsou zahlcení prací, která pro ně nemusí představovat velký přínos a vzbuzuje v nich pouze přítěž. Často vyžadovaná a potřebná komunikace odpoutává jejich pozornost od hluboké práce a nemohou se plně soustředit na vývoj samotného produktu.

Pro správnou implementaci a fungování vzájemných kontrol je nutné dodržovat určité základní doporučení:

- Předcházení se osobním sporům - autor kódu si musí uvědomit, že kritika jeho/její kódu neznamená útok na jeho/její osobu.

- Zachování profesionality při přijímání kritiky
- Striktní dodržování stanovených pravidel
- Správně určený rozsah (scope) - přesná definice úkolu, funkčnosti a akceptačních kritérií, které musí být vyřešeny pro splnění úkolu.
- Definice týmového standardu - tým se musí shodnout na společných prvcích, které chtějí dodržovat, jako jsou například používání určitých návrhových vzorů, frameworků, či komentářové syntaxe.
- Zaznamenávání jednotlivých kontrol - samozřejmostí by měla být i dokumentace jednotlivých kontrol.
- Zvolit si variantu kontroly
    - Pre-commit - kontrola se provádí před vydáním produktu a zabezpečí vyšší kvalitu. Hodí se při vývoji důležitého SW, kdy sa klade veliký důraz na kvalitu a bezchybovost.
    - Post-commit - kontrola sa provadí až dodatečně, po releasnutí produktu. Využívá se při projektech s častými změnami.
- Nahrazení manuálního nastavení automatizovanými systémy a šablonami
- Vyvarovat se běžným problémům, jakými jsou:
    - Účastníci nerozumí fungování review procesu
    - Review proces není dodržován
    - Neúčast kompetentních lidí
    - Při reviews se více řeší styl než kvalita kódu
    - Provedena kontrola, ale již už ne, kontrola opravy kontroly

Obrázek č.3: Peer review

Závěr

V práci byl čtenář seznámen s problematikou peer review, jak u klasického business projektu, tak i u projektu agilního vývoje informačního systému. Byly zmíněny obecné a přesto v praxi často nedodržované potřebné náležitosti pro provádění správného strukturovaného peer review. Byly poskytnuty informace o některých z nástrojích pro podporu takového procesu. Výhody a nevýhody procesu byly bodobě definovány.

Zdroje

Literatura

Developers Guide to Peer Reviews. *TechRepublic* [online]. [cit. 2018-12-21]. Dostupné z: https://www.techrepublic.com/article/developers-guide-to-peer-reviews/

What is Peer Review [online]. [cit. 2018-12-21]. Dostupné z: https://www.elsevier.com/reviewers/what-is-peer-review

LONDA, Patrick. Peer Reviews Either Sandbag or Propel Agile Development [online]. 12.7.2018 [cit. 2018-12-218]. Dostupné z: https://www.infoq.com/articles/peer-reviews-sandbag-propel?utm_source=infoqemail&utm_medium=culture-methods&utm_campaign=newsletter&utm_content=07172018

Obrázky

Obrázek č.1: http://s3.amazonaws.com/libapps/accounts/5449/images/PeerReviewSlide.jpg
Obrázek č.2: http://amanek.com/images/posts/code-review-process.png
Obrázek č.3: https://australianclimatemadness.files.wordpress.com/2014/07/peer_reviewr4_scr.jpg