Institut für Softwaretechnologie

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Fachstudie Nr. 190

Werkzeuge für Code-Reviews und Code-Abnahmen

Sebastian Beck, Nikolai Neugebauer, Daniel Pfeiffer

Studiengang:	Softwaretechnik
Prüfer/in:	Prof. Dr. rer. nat. Stefan Wagner
Betreuer/in:	Dipl.-Inf. Ivan Bogicevic
Beginn am:	01.11.2013
Beendet am:	03.05.2014
CR-Nummer:	D.2.9, K.6.3

Kurzfassung

Code-Reviews sind ein anerkanntes Mittel um die Codequalität zu verbessern. Sie sind heutzutage zum Standard in der Industrie geworden. Bei der Firma AEB werden seit längerem agile Codereviews für den implementierten Programmcode durchgeführt. Jedoch ist man mit dem Rahmen, in dem die Reviews durchgeführt werden, nicht zufrieden.

In dieser Arbeit wurden Prüfwerkzeuge für Codereviews getestet und bewertet. Anhand dieser Bewertung wird eine Empfehlung an AEB gegeben um die Unterstützung bei Codereviews zu verbessern.

Abstract

Code reviews are recognized means of improving code quality. They became standard in the industry nowadays. At AEB they use an agile form of code reviews to check their implemented code. Though they are not pleased with the framework around the reviews.

In this study code review tools were tested and rated. As a result of this rating a recommendation were given to AEB to get a better support while doing reviews.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Codereviews sind im Software-Engineering ein anerkanntes Mittel um die Qualität des Codes zu erhöhen und das Wissen über den Code in der Firma weiter zu geben. Sie werden weitgehend in der Industrie verwendet. Die Firma AEB aus Stuttgart verwendet seit Jahren eine auf sie angepasste Form des Reviews. Allerdings sind Sie mit der derzeitigen Integration der Reviews in den Entwicklungsprozess nicht zufrieden. Sie wollen statt den derzeit verwendeten Ansätzen eine Lösung haben, die den Ablauf optimiert und kontrollfähig macht.

1.2 Ziel

Ziel der Arbeit ist es eine Empfehlung zu abzugeben, wie die Review - Situation bei AEB verbessert werden kann. Dazu werden verfügbare Prüfwerkzeuge getestet und nach den Gesichtspunkten überprüft, ob die von AEB geforderten Eigenschaften erfüllt werden. Falls kein Prüfwerkzeug empfehlenswert ist, soll eine andere Lösung gefunden und eine Empfehlung ausgesprochen werden, die die derzeitige Situation verbessern würde.

1.3 Übersicht

Kapitel 2 Reviews
In diesem Kapitel werden Reviews nach dem Prinzip des Software-Engineerings erklärt und erläutert wie sie derzeit bei AEB umgesetzt werden.
Kapitel 3 Review-Werkzeuge
Hier wird die gefundene Auswahl der Tools aufgezählt und die Gründe genannt, warum manche Werkzeuge es nicht in die finale Testphase geschafft haben. Es wird die Bewertungsmatrix erläutert und die 7 Finalisten beschrieben und bewertet.
Kapitel 4 Zusammenfassung und Empfehlung
In diesem Kapitel werden die Pro und Contras erörtert und eine Empfehlung ausgesprochen.

2 Reviews

2.1 Reviews bei AEB

Bei unserem Industriepartner AEB sind Code-Reviews vorgesehen bevor Code ins Produkt übernommen wird. Reviews finden dabei informell statt. Weiterhin sind keine Treffen für Reviews vorgesehen, sondern die Code-Abnahme läuft zumeist verteilt und asynchron ab. Zusätzlich gibt es keinen festgeschriebenen Review Workflow, sondern es werden drei Formen parallel genutzt.

2.1.1 Eclipse Diff-Dateien

Eine der drei Vorgehensweisen ist die Generierung von Diff-Dateien mit Eclipse. Diese wird dann per E-Mail an die Reviewer geschickt. Feedback erhält der Autor dann ebenfalls via E-Mail.
Dieses Vorgehen wirft eine Reihe von Nachteilen auf. Zum einen sind gleiche Eclipse Base-Stände notwendig, sodass die Diff-Datei einfach angewendet werden kann. Außerdem können E-Mails leicht untergehen. Eine Kontrolle durch Dritte ist bei diesem privaten Austausch nicht möglich. Auch das Übermitteln von Feedback ist umständlich, da nur schwer vermittelbar ist, auf welchen Codeblock sich ein Kommentar bezieht.
Insgesamt ist dieses Vorgehen nicht zufriedenstellend, da das Vorgehen sehr umständlich ist und außerdem die Prozesssicherheit und Nachvollziehbarkeit des Reviews nicht gewährleistet sind.

2.1.2 CodeReviewer

Parallel zum Austausch von Diff-Dateien, wird auch das Tool CodeReviewer eingesetzt. Das Programm weist allerdings viele Schwächen auf. Insbesondere ist es nicht möglich den Code aus dem Tool in den Workspace zu laden, sodass er lokal, mit gewohntem Syntaxhighlighting und gewohnten Eclipse Features eingesehen werden kann. Auch das Ausführen des Codes zu Testzwecken, oder das Überprüfen der Lauffähigkeit von Testfällen ist bei diesem Vorgehen nicht möglich. Weitere Probleme des Tools ist die Unübersichtlichkeit der Kommentare, sowie die fehlende Möglichkeit Commitrechte für das Repository einzustellen, sodass nur gereviewter Code ins Produkt übernommen werden kann.

2.1.3 Side-by-side Abnahme

Zusätzlich zu den oben geschilderten Vorgehensweisen kommen auch Side-by-side Abnahmen zum Einsatz.
Probleme treten dabei insbesondere dadurch auf, dass alle Reviewteilnehmer gleichzeitig Zeit für das Review haben müssen. Dafür ist die Kommunikation sehr direkt und es gibt keine komplizierten Umwege. Außerdem kann alles in der gewohnten Entwicklungsumgebung stattfinden.
Durch diese direkte Art der Kommunikation ist keinerlei Nachvollziehbarkeit des Reviews gegeben. Weiterhin ist keine Kontrolle durch Dritte möglich. Die Prozesssicherheit ist also nicht gegeben.

3 Review Tools

3.1 Marktübersicht

3.1.1 ReviewClipse

- Custom EULA, kostenlos, nicht Open Source
- http://www.inso.tuwien.ac.at/projects/reviewclipse/
- keine Weiterentwicklung seit 2009 → keine detaillierte Betrachtung

3.1.2 ReviewMate

- Custom EULA, proprietär
- http://www.elementriver.com/reviewmate

3.1.3 Gerrit mit MyLyn Connector

- Apache 2.0 Lizenz / Eclipse Public License
- http://code.google.com/p/gerrit/

3.1.4 Jupiter

- Apache 2.0 Lizenz
- https://code.google.com/p/jupiter-eclipse-plugin/
- nur mit Eclipse Juno kompatibel → keine detaillierte Betrachtung

3.1.5 MyLyn Review4Eclipse

- Eclipse Public License
- https://www.eclipse.org/r4e/

3.1.6 Reviewboard

- MIT Lizenz, kostenpflichtiger Powerpack verfügbar
- http://www.reviewboard.org/

3.1.7 CodeBrag

- kostenlose Preview, später kostenpflichtig
- http://codebrag.com/

3.1.8 Malevich

- Microsoft Public License, kostenlos
- http://malevich.codeplex.com/
- Betarelease 2009 → keine detaillierte Betrachtung

3.1.9 Barkeep

- * MIT Lizenz
- http://getbarkeep.org/

3.1.10 Phabricator

- Apache 2 Lizenz
- http://phabricator.org

3.1.11 cahoots

- Custom EULA, proprietär
- http://download.klocwork.com/

3.1.12 CodeStriker

- kostenlos, Open Source
- http://codestriker.sourceforge.net/
- Keine Weiterentwicklung seit 2009 → keine detaillierte Betrachtung

3.2 Bewertungsmatrix

Die Bewertungsmatrix ist auf Basis von Gesprächen mit AEB entstanden. Nach einer ersten Auflistung der gewünschten Punkte sind die Punkte in Must-haves und Nice-to-haves unterteilt wurden. Basis für jedes Programm (keine Punkte in der Matrix) ist die Unterstützung von SVN. Bei manchen Tools wurde eine Ausnahme gemacht (Gründe stehen in dem jeweiligen Bericht).

Preislich ist AEB bereit für ein Tool Lizenzgebühren bis zu 100 Euro/User/Jahr zuzahlen. Open-Source wird aber auch gerne gesehen.
Einer der wichtigsten Punkte ist die Usability, denn das Tool wird über eine lange Zeit von vielen Entwicklern benutzt werden. Punkte gab es für Optik, Erreichbarkeiten von Features und Übersichtlichkeit.

Zu festgestellten Must-haves gehört eine Rollenverteilung mit zugehörigen Einschränkungen. Ein weiterer wichtiger Bestandteil des Tools sollte ein Eclipse-Plugin sein. Damit sollen entweder in Eclipse das Review durchgeführt werden (bevorzugt) oder ein Review generiert werden. Sicherheit und Bugs sollen auch berücksichtigt werden, deshalb sind regelmäßige Updates/-Patches/Releases sehr wichtig und notwendig.

Um die Usability zu steigern wäre es gut, wenn das Produkt E-Mails bei neuen Reviews und Änderungen an die betreffenden Personen sendet.

Weitere Punkte, sowohl Must-haves als auch Nice-to-haves wurden erwähnt und in die Bewertungsmatrix aufgenommen.

Um eine vollständige Bewertung abzugeben, gibt es den Extrapunkt Zusatzpunkte bzw. Punktabzüge. Ist das Tool durch besondere Features oder Bugs aufgefallen, bekommt es mit dem Grund versehen gegeben falls einen Bonus oder Malus.
Somit wird auch neben der Usability die eigene Erfahrung mit dem Programm gewertet.

Die Punkteverteilung wurde auf Grund von persönlicher Erfahrung und durch Gespräche mit AEB eingeschätzt und besprochen.

Die komplette Bewertungsmatrix findet sin in Tabelle 3.1.

3.3 Toolbewertungen

3.3.1 Codebrag

Codebrag ist ein Review –Tool, entwickelt von der Firma SoftwareMill. Die Versionen 1.x sind kostenlos, die nachfolgenden Versionen werden kostenpflichtig sein. Das Tool arbeitet mit hinterlegten Repositories. Es wird sowohl SVN als auch Git unterstützt. Dabei kann das Repository lokal oder in der Cloud (z.B Github, Bitbucket) gehostet werden. Im Moment wird nur ein Branch unterstützt, was innerhalb aber nachfolgenden Versionen verbessert werden

Kriterium	Punkte / max. Punkte
Allgemein	
Preis	5
Usability	25
	30
Must-have	
Verteilung von Rollen	3
Rollen Einschränkungen	6
Codevorschläge durch Reviewer	10
Globale Kommentare	5
Eclipse Plugin	10
Code / Testfälle ausführbar	8
Daten zum Review hinzufügen / löschen	5
Unterstützung von binären Dateien	4
Guter Support	6
Aktuell: regelmäßige Releases / Updates / Patches	5
	62
Nice-to-have	
Notifications	3
Review in SVN-Branches	2
Automatische Erzeugung von Branches	2
Schnittstelle für Dokumentation in Issuetracker	1
	8
Zusatzpunkte / Abzüge	
	0
Gesamtpunkte	**100**

Tabelle 3.1: Bewertungsmatrix

soll.
Die einzige Sprache die unterstützt wird ist Englisch.

Installation

Um Codebrag zu benutzen benötigt man Java 1.7. Das Tool unterstützt Windows und Linux. Zuerst muss Codebrag als Zip-Datei von der Homepage heruntergeladen werden. Nachdem man den Inhalt entpackt hat, muss das gewünschte Repository in den Ordner codebrag/repos geklont werden.
Um auf das Repository zugreifen zu können, muss man in der codebrag.conf Benutzername und (optional) Passwort angeben. Ebenso kann dort der E-Mail-Dienst konfiguriert werden, falls man E-Mail-Benachrichtigen bei Änderungen oder neuen Commits haben möchte. Starten tut man das Tool nun mit der Batchdatei codebrag.bat. Mit der Batchdatei wird ein Jetty-Server gestartet, dessen Einstellungen konfigurierbar sind.

Nutzung

Die Benutzung des Tools erfolgt über den Browser (keine Einschränkungen). Als Startbildschirm erscheint ein Anmeldeformular. Dort gibt man die Daten ein, die innerhalb von der Konfigurationsdatei spezifiziert wurde. Nachdem man sich dort angemeldet hat, befindet man sich auf der Hauptoberfläche von Codebrag. Oben rechts gibt es einen Konfigurationsbutton, in der man ein Profil einrichten kann, andere User einladen und sich abmelden kann.
Insgesamt gibt es zwei Menüpunkte: Commits und Follow-Ups.
Bei „Commits“ sieht man jeden Commit der im Branch gemacht wurde. Wählt man einen Commit aus, sieht man alle Änderungen die durch den Commit im Quellcode gemacht wurden. Falls mehrere Dateien bei Commit verändert wurden, kann diese bei „FILES IN THIS COMMIT“ auswählen. Geänderte Zeilen werden rot (alt) und grün (neu) markiert. Jede Zeile kann kommentiert und „geliked“ (die Zeile kann mit ein Herz bekommen) werden. Zu jedem Kommentar kann eine Antwort geschrieben werden. Codevorschläge werden nicht syntax - gehighlightet. Ist das „Review“ beendet, kann man den Commit als „MARK COMMIT AS REVIEWED“ markieren.
Bei „Follow-ups“ werden alle Anmerkungen und Kommentare, die andere Teammitglieder geschrieben haben aufgelistet. So hat man einen Überblick wer was zum eigenen Quellcode geschrieben hat, ohne den speziellen Commit aufzurufen zu müssen.
Bei Fehlermeldungen oder Wünschen gibt es noch den „Feedback“ - Button. In diesem Formular kann man das Entwicklerteam kontaktieren.

Bewertung

Das Tool ist kostenlos (aber nicht Open-Source) und bietet eine aufgeräumte Oberfläche. Die Bedienung ist einfach und schnell zu lernen. Die wenigen Menüs sind mit wenigen Mausklicks schnell erreichbar. Einer der großen Hauptprobleme ist aber die Einschränkung, dass nur ein Branch unterstützt wird. Grundsätzlich merkt man bei der Benutzung des Tools, dass

es nur sehr wenige Features gibt. Es gibt keine Rollen, kein Eclipse – Plugin. Zusätzlichen Punktabzug gab es auf Grund der Eigenschaft, dass nur Commiteter Quellcode gereviewed werden kann. Jegliche Korrekturen müssen dann erst commitet werden, um dann wiederum gereviewt werden.

In Tabelle 3.2 findet sich eine ausgefüllte Bewertungsmatrix für Codebrag.

Zusammenfassung

Codebrag hat wenige Features. Diese laufen zwar tadellos, es wirkt aber trotz allem nur wie eine Demo-Version mit eingeschränktem Umfang. Das Tool ist nur sinnvoll, falls man ein kleines Projekt mit maximal drei Entwicklern hat.

Urteil: 35 Punkte → nicht empfehlenswert.

3.3.2 ReviewMate

ReviewMate ist ein Eclipse Plugin für Reviews und ist somit plattformübergreifend auf allen Systemen verfügbar, die Eclipse unterstützt. Wie jedes Eclipse Plugin ist auch ReviewMate in Java geschrieben. Element River LLC. hat ReviewMate entwickelt und vertreibt es auch.
ReviewMate ist ein reines Eclipse Plugin und kommt ohne Serverbackend aus. Die Daten eines Reviews werden jeweils in einer Reviewdatei (*.rvw) gespeichert.
Im Feld der kommerziellen Tools ist ReviewMate das Günstigste. Eine Lizenz kostet 59,84 Euro, wobei Mengenrabatte verfügbar sind. Hierdurch läge der Preis im von AEB kalkulierten Abnahmerahmen, von 50-100 Lizenzen, bei 44,88 Euro pro Lizenz. Zum Testen des Plugins kann ReviewMate für 30 Tage in vollem Umfang kostenlos genutzt werden.
Das Plugin ist nur auf Englisch verfügbar.

Installation

Die Installation von ReviewMate gestaltet sich einfach, da das gewohnte Eclipse Plugin System genutzt wird. In Eclipse kann direkt die Downloadseite von ReviewMate angeben und das Tool von dort installiert und eingebunden wird.
Nach dem Neustart von Eclipse wird der Nutzer aufgefordert die Lizenzinformationen einzugeben, oder die kostenlose 30-Tage Version zu aktivieren (Abbildung 3.1).

Weiterhin erhält der User anhand einiger Screenshots eine Einführung in die Nutzung von Reviewmate. (Abbildung 3.2)

Im Anschluss erfolgt die Initialisierung des Plugins. Hierbei kann ein Name gewählt werden, unter dem Review Kommentare erstellt werden, sowie eine Farbe für die Reviewkommentare definiert werden. (Abbildung 3.3)

Kriterium	**Punkte / max. Punkte**
Allgemein	
Preis	5 / 5
Usability	13 / 25
	18 / 30
Must-have	
Verteilung von Rollen	0 / 3
Rollen Einschränkungen	0 / 6
Codevorschläge durch Reviewer	0 / 10
Globale Kommentare	5 / 5
Eclipse Plugin	0 / 10
Code / Testfälle ausführbar	0 / 8
Daten zum Review hinzufügen / löschen	0 / 5
Unterstützung von binären Dateien	4 / 4
Guter Support	2 / 6
Aktuell: regelmäßige Releases / Updates / Patches	3 / 5
	14 / 62
Nice-to-have	
Notifications	3 / 3
Review in SVN-Branches	0 / 2
Automatische Erzeugung von Branches	0 / 2
Schnittstelle für Dokumentation in Issuetracker	0 / 1
	3 / 8
Zusatzpunkte / Abzüge	
	0
Gesamtpunkte	**35 / 100**

Tabelle 3.2: Bewertungsmatrix Codebrag

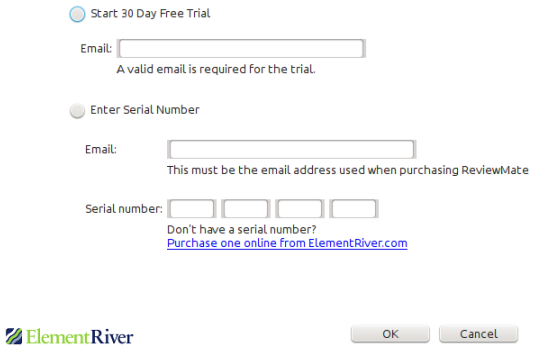

Abbildung 3.1: ReviewMate Registrierung

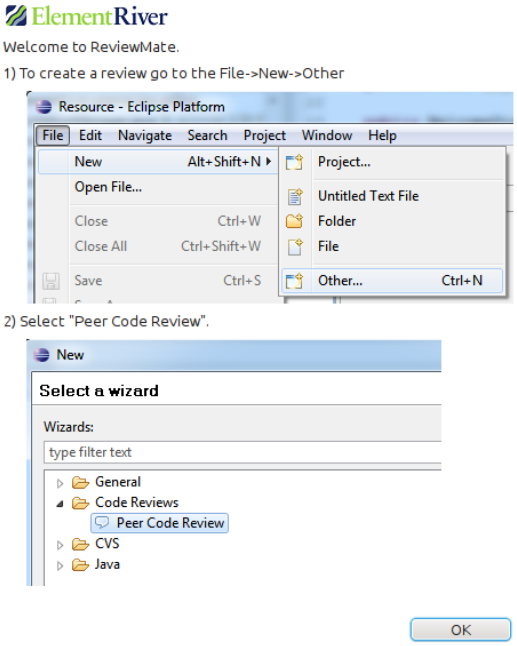

Abbildung 3.2: ReviewMate nach Installation

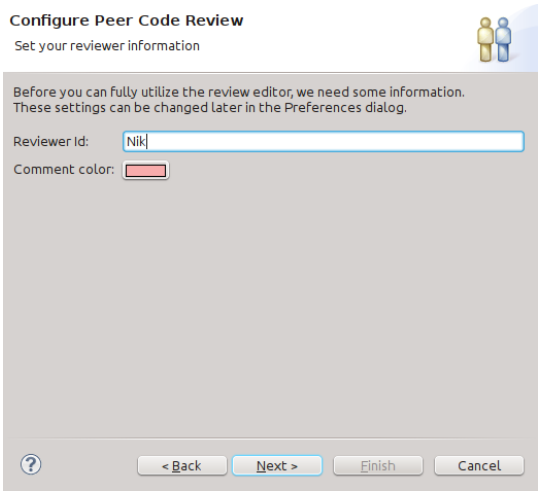

Abbildung 3.3: ReviewMate Initialisierung

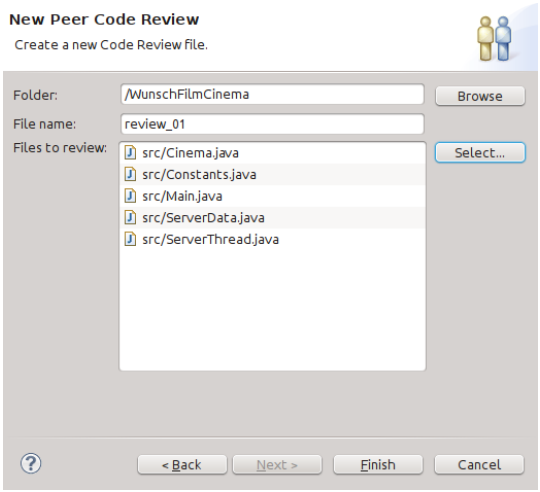

Abbildung 3.4: ReviewMate Neues Review 1

Nutzung

Ein Review kann direkt in Eclipse erstellt werden. Da jedes Review eine eigene Datei ist, muss für ein Review eine neue Review Datei erstellt werden. Dies funktioniert auf die von Eclipse gewohnte Art und Weise entweder über einen Rechtsklick auf entsprechenden Ordner bzw. entsprechendes Paket, oder über das File Menü. Anschließend können einzelne Dateien aus dem Eclipse Projekt zum Review hinzugefügt werden (Abbildungen 3.4, 3.5).

Nach dem Erstellen eines Reviews wird dieses im Review-Editor des ReviewMate Plugins angezeigt (Abbildung 3.6).

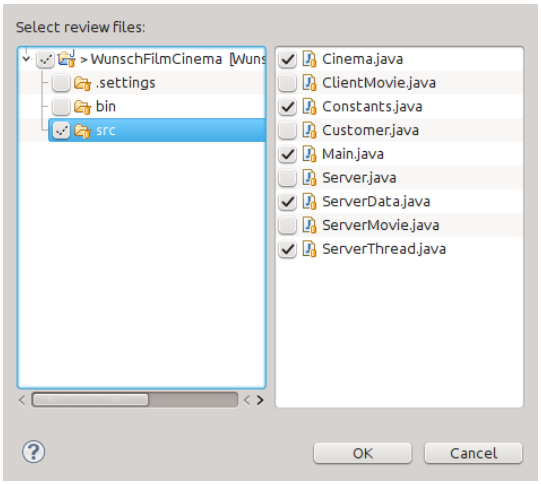

Abbildung 3.5: ReviewMate Neues Review 2

Der vom Hersteller empfohlene Workflow sieht vor, dass die Reviewdatei jetzt ins Repository commitet wird, um das Review den anderen Reviewteilnehmern zugänglich zu machen.
Im Review-Editor können die einzelnen zu reviewenden Dateien angeschaut werden. Hierzu der Review-Editor verwendet, der zwar Syntaxhighlighting für viele Sprachen zur Verfügung stellt, allerdings fehlen alle anderen gewohnte Eclipse Features, wie zum Beispiel das Einblenden von JavaDoc Kommentaren für aufgerufene Methoden, oder das Springen zur Deklaration verwendeter Komponenten.

Das Kommentieren des Codes ist intuitiv: markieren des Blocks und Drücken des Buttons Ädd Comment"(Abbildung 3.7).
Alle Kommentare werden im Revieweditor am Rand markiert und Kommentare werden bei Mouseover Events eingeblendet. Verweise auf die Kommentare in allen anderen Editoren sind nicht vorhanden.
Weiterhin gibt es die Möglichkeit neue Dateien, oder neue Versionen von bisherigen Dateien zum Review hinzuzufügen.

Bewertung

Auffällig bei der Bewertung ist insbesondere das sehr schlechte Urteil für die Usability. Einige Probleme wurden bei der Benutzung bereits aufgeführt, wie die fehlenden Features des Review-Editors. Ein schwerwiegenderes Problem ist jedoch die fehlende Möglichkeit nur einzelne Codepassagen zum Review hinzuzufügen. Auch gibt es keine Möglichkeit nur geänderte Codestellen für ein Review zu betrachten. Die Kommunikation welche Teile einer Datei geändert wurden muss also abseits des Reviews erfolgen, um das Reviewen von nicht geänderten Ausschnitten zu vermeiden.
Eine weitere Schwäche in der Nutzung offenbart sich in der Struktur der Reviewdatei. Diese

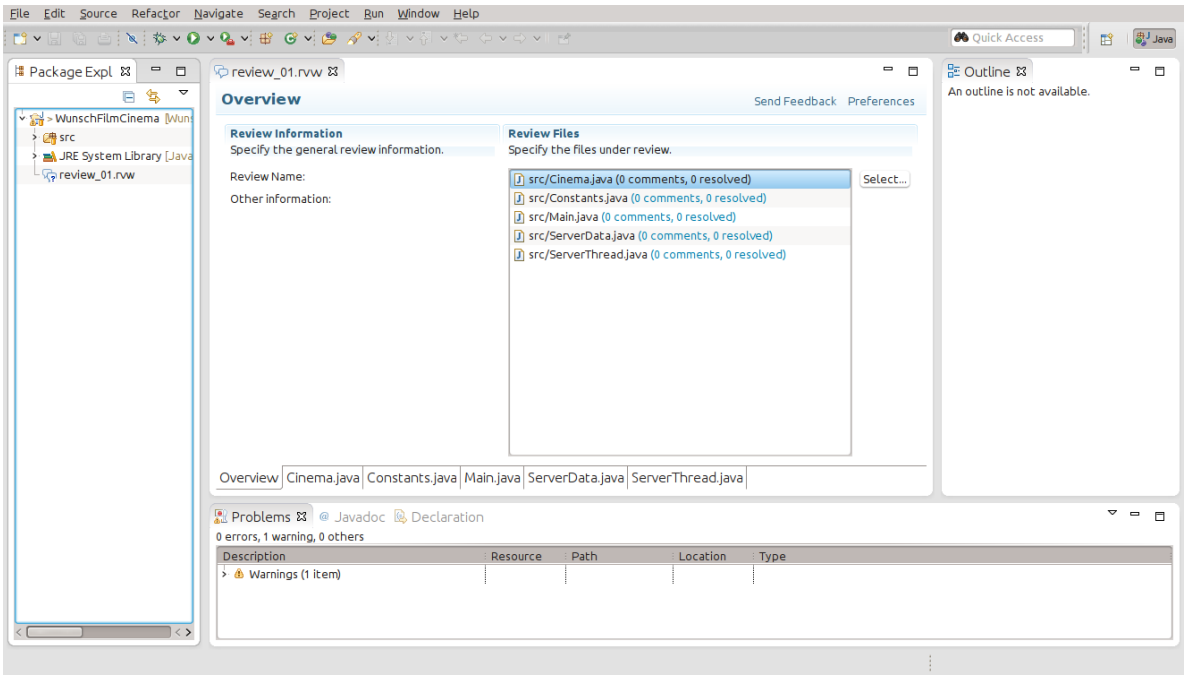

Abbildung 3.6: ReviewMate Review Editor Übersicht

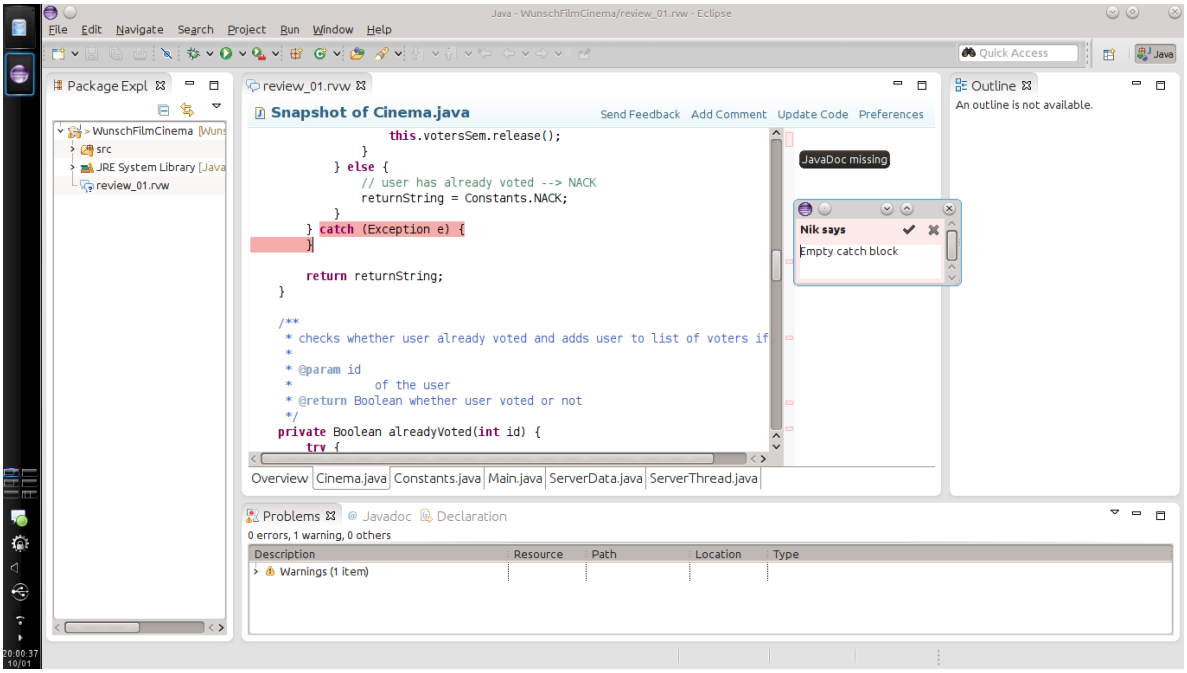

Abbildung 3.7: ReviewMate Review Editor Übersicht

sind überwiegend xml Format, allerdings sind manche Informationen auch binär gespeichert. Beim vorgeschlagenen Workflow, das die Reviewdatei im Repository liegt, kann es also zu Merge Problemen kommen, wenn mehrere Reviewteilnehmer gleichzeitig das Review bearbeitet haben. Binäre Codeteile manuell zu mergen ist nicht möglich und somit wäre das Review in diesem Fall unbrauchbar.
Das Hinzufügen einer neuen Version einer zu reviewenden Datei ist die größte Schwäche von ReviewMate. Hierbei wird die alte Datei komplett aus dem Review entfernt. Dies betrifft insbesondere auch alle Anmerkungen der Reviewer. Es ist dann also nicht mehr möglich zu überprüfen, ob alle Probleme gelöst wurden, oder die einzelnen Abläufe des Reviews nachzuvollziehen.
Einige Features, wie die Rollenverteilung, die Möglichkeit für Codevorschläge oder das Ausführen des Codes fehlen komplett. Globale Kommentare sind nicht vorgesehen, allerdings kann jedes Review eine Beschreibung in Textform haben. Dieses könnte für globale Kommentare genutzt werden.
Die Dokumentation ist kurz gehalten, ein Supportformular ist auf der Homepage des Herstellers vorhanden. Eine Release History ist nicht verfügbar.

In Tabelle 3.3 findet sich eine ausgefüllte Bewertungsmatrix für ReviewMate.

Zusammenfassung

Der größte Vorteil von ReviewMate ist die direkte Integration in Eclipse. Weiterhin ist die Benutzung schnell verständlich und einfach. Allerdings zeigen sich auch viele Nachteile, wobei der Verlust von Reviewdaten beim Hinzufügen einer neuen Version einer Datei wohl den größten Nachteil darstellt. Viele Features sind außerdem gar nicht vorhanden.

Urteil: 35 Punkte → nicht empfehlenswert

3.3.3 barkeep

barkeep ist ein sehr schlankes, webbasiertes Reviewtool. barkeep ist Open-Source und steht unter der MIT Lizenz. Geschrieben wurde das Programm mit Hilfe des Webframeworks Ruby on Rails und ist damit plattformunabhängig. Das Userfrontend ist webbasiert, wird also über den Webbrowser aufgerufen und ist damit ebenfalls plattformunabhängig.
barkeep arbeitet mit einem Post-Commmit Workflow und kann für Reviews aus beliebig vielen git Repositories genutzt werden.

Installation

Um barkeep betreiben zu können wird neben barkeep selbst eine MySQL Datenbank, einen Webserver, Ruby und das Rails Framework benötigt. Zudem müssen einige Ruby Abhängigkeiten (gems) installiert werden. Für Ubuntu basierte Systeme gibt es ein Bash-Script, das alle Abhängigkeiten automatisch herunterläd und installiert. Eine Installationsanleitung, oder eine genaue Erläuterung der benötigten Abhängigkeiten bietet barkeep nicht. Möchte man barkeep

Kriterium	**Punkte / max. Punkte**
Allgemein	
Preis	2 / 5
Usability	9 / 25
	11 / 30
Must-have	
Verteilung von Rollen	0 / 3
Rollen Einschränkungen	0 / 6
Codevorschläge durch Reviewer	0 / 10
Globale Kommentare	2 / 5
Eclipse Plugin	10 / 10
Code / Testfälle ausführbar	0 / 8
Daten zum Review hinzufügen / löschen	5 / 5
Unterstützung von binären Dateien	4 / 4
Guter Support	2 / 6
Aktuell: regelmäßige Releases / Updates / Patches	0 / 5
	23 / 62
Nice-to-have	
Notifications	1 / 3
Review in SVN-Branches	0 / 2
Automatische Erzeugung von Branches	0 / 2
Schnittstelle für Dokumentation in Issuetracker	0 / 1
	1 / 8
Zusatzpunkte / Abzüge	
	0
Gesamtpunkte	**35 / 100**

Tabelle 3.3: Bewertungsmatrix ReviewMate

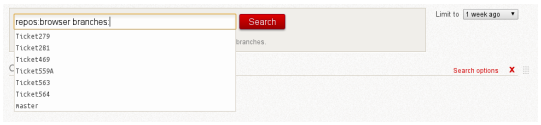

Abbildung 3.8: barkeep Suche 1

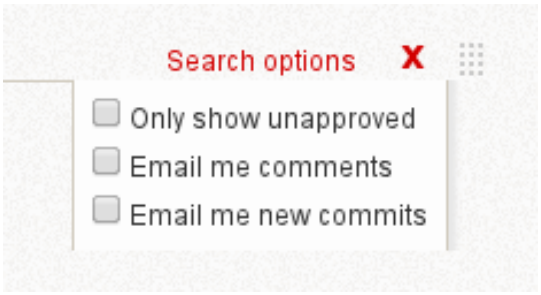

Abbildung 3.9: barkeep Suche 2

also auf einem anderen Betriebssystem als auf Ubuntu installieren, muss man die benötigten Abhängigkeiten einzeln aus dem Bash-Script herauslesen. Selbiges gilt zum Beispiel auch, wenn man barkeep mit dem Apache und nicht dem nginx Webserver betreiben möchte, der sonst automatisiert mitinstalliert wird. Auch das Installationsverzeichnis kann über das vorgefertigte Bash-Script nicht angepasst werden.

Der erste Nutzer, der sich einloggt erhält automatisch Administrator Rechte.

Nutzung

Jeder Commit in eines der Repositories, die barkeep bekannt sind kann als Basis eines Reviews genutzt werden. Über eine Suchfunktion und Filter (Abbildungen 3.8, 3.9) können noch nicht gereviewte Commits gefunden werden. Außerdem kann für jeden Commit manuell ein Review beantragt werden.

Die Änderungen eines Commits können sowohl via Side-by-side Diff, als auch via Inline Diff betrachtet werden. Kommentare sind für ganze Commits und für einzelne geänderte Codezeilen Änderungen möglich. Weiterhin kann ein Commit als gereviewed markiert werden.

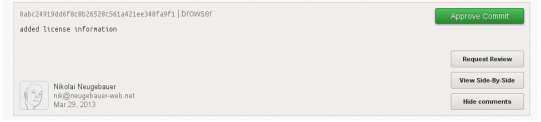

Abbildung 3.10: barkeep Reviewübersicht

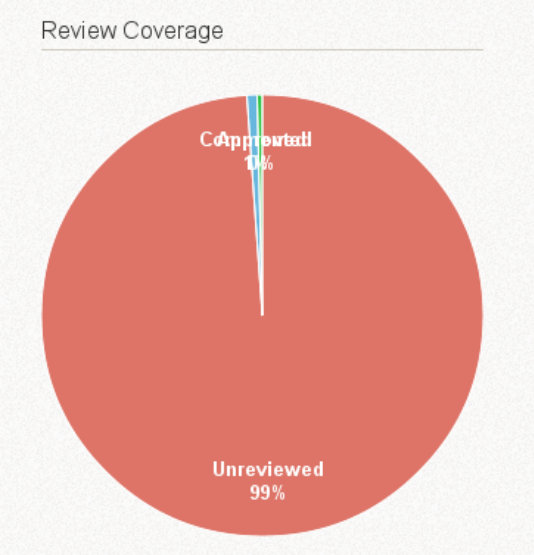

Abbildung 3.11: Reviewboard 2

Um eine Übersicht über den Reviewstand der Repositories zu erhalten steht eine kleine Statistik zur Verfügung, die anzeigt, wie viele Commits kommentiert, gereviewed beziehungsweise noch nicht betrachtet wurden (Abbildung 3.11).

Bewertung

Neben dem sehr geringen Funktionsumfang fehlen Punkte, insbesondere bei der Usability.

Eine Übersicht über Commits gibt es nicht, alles muss manuell über die Suche gefunden werden. Diese bietet zwar dank Auto-complete einen gewissen Komfort, einfache Bedienung ist dennoch etwas anderes.

Bei der Einführung von barkeep für ältere Repositories sind die Statistiken außerdem kaum aussagekräftig, da natürlich der Großteil der Commits nicht gereviewet wurde. Ebenso fehlt die Möglichkeit Commits abzulehnen, sodass diese zwar als gereviewet markiert werden, aber eben nicht im positiven Sinn. Ein sinnvoller Workflow, wie fehlerhafte Commits zu behandeln sind ist außerdem ebenso nicht vorgesehen.

Ein weiterer großer Nachteil ist die erzwungene Nutzung von Google Accounts.

Die komplette Bewertungsmatrix ist in Tabelle 3.4 dargestellt.

Kriterium	Punkte / max. Punkte
Allgemein	
Preis	5 / 5
Usability	13 / 25
	18 / 30
Must-have	
Verteilung von Rollen	0 / 3
Rollen Einschränkungen	0 / 6
Codevorschläge durch Reviewer	0 / 10
Globale Kommentare	5 / 5
Eclipse Plugin	0 / 10
Code / Testfälle ausführbar	0 / 8
Daten zum Review hinzufügen / löschen	0 / 5
Unterstützung von binären Dateien	4 / 4
Guter Support	2 / 6
Aktuell: regelmäßige Releases / Updates / Patches	3 / 5
	14 / 62
Nice-to-have	
Notifications	3 / 3
Review in SVN-Branches	0 / 2
Automatische Erzeugung von Branches	0 / 2
Schnittstelle für Dokumentation in Issuetracker	0 / 1
	3 / 8
Zusatzpunkte / Abzüge	
	0
Gesamtpunkte	**35 / 100**

Tabelle 3.4: Bewertungsmatrix barkeep

Zusammenfassung

barkeep bietet nur sehr wenige Features und diese sind nicht sonderlich ausgereift, beziehungsweise sind umständlich zu bedienen. Da es sich zusätzlich um ein reines git Tool handelt und bei AEB momentan SVN eingesetzt wird ist barkeep keine Option.

Urteil: 35 Punkte –> nicht empfehlenswert

3.3.4 Cahoots

Klocwork Cahoots 3.1 ist ein kostenpflichtiges Review – Tool. Der aktuelle Preis pro User/-Jahr beträgt 99$. Unterstützte Repositories sind SVN, Git und einige weitere. Die einzige unterstützte Sprache ist Englisch.

Installation

Um Cahoots benutzen zu können, muss ein Server (Apache Tomcat) und ein Eclipse-Plugin installiert werden.
Der Server wird auf Windows mittels einer *.exe - Datei installiert. Während der Installation werden drei Dienste installiert, die beim Start des Betriebssystems den Server, die benötigte Datenbank und das Lizensierungssystem starten. Port und Adresse des Servers kann in der Datei /config/servers_config.xml geändert werden. Möchten die User über Reviews Benachrichtigungen erhalten, muss das Programm kwauthconfig.exe aufgerufen werden. Im Menüpunkt „Mail Server Configuration“ kann man die E-Mail Konfigurationseinstellungen anpassen.

Das Eclipse – Plugin kann über den Eclipse Marketplace mit der Suche nach „Cahoots“ gefunden und installiert werden.

Nutzung

Um Reviews erstellen zu können, müssen zuerst User erstellt werden. Bei jeder Anmeldung eines Users, dessen Anmeldenamen noch nicht existent ist, wird der Benutzer neu angelegt. Eine andere Methode ist die Verknüpfung mit einem LDAP.
Als nächstes werden die Reviews erstellt. Das besondere ist, es können nur geänderte, nicht commiteter Quellcode als Review hinzugefügt werden.
Das Tool generiert mit wenigen Schritten ein Review, aber ist sehr eingeschränkt in dessen Funktionen. Es reicht ein Rechts-Klick auf die geänderte Quelltextdatei. Im Kontextmenü wird mit dem Klick auf „Create Cahoots Code Review“ ein Dialog geöffnet in der man Kommentare und Reviewer hinzufügen kann. Mit dem Klick auf „Submit“ erhält man einen Link zu dem eben erstellten Review. Dieser ist nun auf dem Server verfügbar.
Alle Nutzer, die für das Review als Reviewer hinzugefügt wurden, werden nun sowohl auf der eigenen Benutzeroberfläche der Cahootsseite benachrichtigt, als auch, falls konfiguriert, als Mail. Der zu betrachtende Quelltext kann nun betrachtet werden. Es stehen zwei Ansichten zur Auswahl: Nur den geänderten Quelltext begutachten oder im Vergleich zum letzten Commit.

Alle Zeilen können kommentiert werden, ebenso können Aufgaben bezüglich der Codezeile gestellt werden. Diese können mit einem Klick auf „Done“ als erledigt markiert werden. Reviews können ebenfalls mit einem allgemeinen Kommentar beschrieben werden.

Ein weiteres Feature ist ein freikonfigurierbarer Feed, der den User benachrichtigt ob in Reviews, die den User betreffen (als Reviewer oder Commiter) eine Neuigkeit gibt.

Bewertung

Negativ anzusehen ist, dass nur geänderte *.java - Dateien vom Plugin erkannt werden. Wurden Binäre oder andere Dateien verändert oder hinzugefügt, erkennt das Plugin diese nicht an und wird beim Review nicht mitgeschickt. Ebenso ist der Aspekt, dass nur noch nicht commitete Dateien beim Review mitgeschickt werden können, als negativ anzusehen. Es ist auch nicht möglich schon commitete Dateien innerhalb eines Reviews zu vergleichen oder überhaupt einem Review hinzuzufügen.
Die Möglichkeit Rollen zu vergeben ist zwar möglich, aber nicht im gewünschten Sinne. Man kann die Commitrechte von Usern nicht einschränken.

In Tabelle 3.5 findet sich eine vollständige Bewertungsmatrix für Cahoots.

Zusammenfassung

Ein Eclipse - Tool ist zwar vorhanden, aber nicht ausgereift. Das Programm ist zwar ausgereift und hat einige nützliche Features, erfüllt aber nicht die Anforderungen von AEB.

Urteil: 38 Punkte → nicht empfehlenswert

3.3.5 Phabricator

Phabricator ist eine webbasierte Produktfamilie für die Zusammenarbeit in Softwareentwicklungsteams. Entwickelt wurde Phabricator zunächst für den internen Gebrauch bei Facebook. Inzwischen wird Phabricator, von Facebook unabhängig, von der Phacility Inc weiterentwickelt und ist unter der Apache 2 Lizenz zugänglich.
Geschrieben ist Phabricator in der Scriptsprache PHP. Die Serversoftware ist damit theoretisch plattformunabhängig, allerdings liegen keine Installationsskripte für Windows bei, sodass ein Betrieb auf einem Windowsserver nicht möglich ist.

Das Userfrontend wird über den Webbrowser aufgerufen und ist für alle modernen Webbrowser und damit auch für alle Betriebssysteme ausgelegt. Eine handyfreundliche Benutzeroberfläche ist für viele Produkte integriert.

Kriterium	**Punkte / max. Punkte**
Allgemein	
Preis	1 / 5
Usability	8 / 25
	9 / 30
Must-have	
Verteilung von Rollen	2 / 3
Rollen Einschränkungen	3 / 6
Codevorschläge durch Reviewer	1 / 10
Globale Kommentare	5 / 5
Eclipse Plugin	2 / 10
Code / Testfälle ausführbar	0 / 8
Daten zum Review hinzufügen / löschen	5 / 5
Unterstützung von binären Dateien	0 / 4
Guter Support	4 / 6
Aktuell: regelmäßige Releases / Updates / Patches	4 / 5
	26 / 62
Nice-to-have	
Notifications	3 / 3
Review in SVN-Branches	0 / 2
Automatische Erzeugung von Branches	0 / 2
Schnittstelle für Dokumentation in Issuetracker	0 / 1
	3 / 8
Zusatzpunkte / Abzüge	
	0
Gesamtpunkte	**38 / 100**

Tabelle 3.5: Bewertungsmatrix Cahoots

Zu Phabricator gehören auch Tools für Codereviews, Differential und Audit. Auf diese richtet sich das Hauptaugenmerk des Produkttests.

Installation

Einen Phabricator Server aufzusetzen ist mit der Dokumentation auf der Homepage einfach möglich. Alle benötigten Abhängigkeiten sind aufgelistet und typische Namen in den Paketmanagern der Linuxdistributionen sind angegeben.
Installiert werden müssen die folgenden Programme:

- git
- ein Webserver (apache, nginx, ...)
- MySQL
- PHP
- diverse PHP Erweiterungen

Phabricator kann dann direkt aus dem git- Repository geklont werden. Im Anschluss daran muss der Webserver konfiguriert werden. Für apache2, nginx und lighttpd liegen der Installationsanleitung Beispielkonfigurationen bei. Für die Erzeugung des benötigten Datenbankschemas legt Phabricator ein Skript bei. Die Verwendung des Skripts ist in der Installationsanleitung erklärt.

Phabricator ist jetzt einsatzbereit. Sollten noch Konfigurationen vorgenommen werden müssen, so wird der Administrator im Programm selbst auf die Probleme hingewiesen.

Nutzung

Phabricator unterstützt sowohl einen Pre- als auch einen Post-Commit Workflow.
Die Post-Commit Features sind in Phabricator sehr schlank gehalten. Für die Durchführung ist es notwendig, dass Phabricator das Repository des Projekts kennt. Jeder Commit in das Repository kann mit Hilfe des Tools Audit in ein Review verwandelt werden. Zu diesem können in der Weboberfläche User als Reviewer hinzugefügt werden, die darüber benachrichtig werden. Sowohl diese User, als auch alle anderen Benutzer, mit den notwendigen Rechten für das Repository können einen Commit kommentieren, akzeptieren oder Bedenken äußern. Weiterhin können einzelne Codezeilen oder Codeblöcke kommentiert werden.

Ein Hinzufügen von Daten zu einem solchen Review ist nicht möglich, womit keinerlei Verbesserungen nachvollzogen werden können.
Für den Pre-Commit Workflow wird das Tool Differential verwendet. Ein Differential Review kann mit den Kommandozeilen Tool Arcantist erstellt werden. Außerdem ist es möglich eine Diff-Datei in der Weboberfläche hochzuladen und auf Basis dieser ein Review zu erstellen.

Der Ersteller des Reviews wird als Autor festgehalten. Es gibt weiterhin die Möglichkeit Reviewer und Abonnenten zum Review hinzuzufügen, sowie Zugriffsberechtigungen für das Review zu definieren. Wird dem Review ein Repository zugeordnet, so wird der Kontext der in der Diff-Datei protokollierten Änderungen aus dem Repository geladen und er kann im Review eingesehen werden. Außerdem können Reviews in Projekten organisiert werden. Zu späteren Zeitpunkten können weitere Diff-Dateien hochgeladen werden, um Änderungen zum Review hinzuzufügen.

Jeder Reviewer und jeder andere User mit benötigten Rechten, vom Autor abgesehen, kann die Änderungen akzeptieren, das Review kommentieren oder Änderungen fordern.

Bei der Erstellung von Reviews über Arcantist ist es möglich Unit-Tests automatisiert auszuführen und das Ergebnis im Review darzustellen.
Bei beiden Varianten können sowohl globale Kommentare als auch Kommentare auf einzelne oder mehrere Zeilen bezogen hinzugefügt werden. Kommentare können gegebenenfalls wieder kommentiert werden. Alle Kommentare unterstützen weiterhin Syntax - highlighting für Codeblöcke.

Für die Betrachtung der Diffs gibt es verschiedene Ansichten, wie Inline und Side-by-side. Das Review findet komplett in der Weboberfläche statt.
Die Benutzerverwaltung in Phabricator kann über LDAP erfolgen.

Bewertung

Für die Bewertung wird hauptsächlich der Pre-Commit Workflow herangezogen, da dieser mehr Features, wie zum Beispiel das Hinzufügen von Daten unterstützt. Außerdem passt dieser Workflow besser zum Vorgehen bei AEB.

Abzüge für Phabricator gibt es insbesondere bei der Usability. Das erzeugen eines neuen Reviews ist unnötig aufwendig, da Diff-Dateien manuell erzeugt und hochgeladen werden müssen. Sie können nicht einfach aus Commits in einen bestimmten Branch erzeugt werden.

Weiterhin ist die Darstellung der Reviewhistory sehr unübersichtlich, da zum Beispiel keinerlei Abtrennung einzelner Aktionen vorhanden ist.

Zudem muss jedes Review durch einen Kommentar beendet werden. Das heißt bevor eigene Kommentare im Code veröffentlicht werden, muss ein globaler Kommentar hinzugefügt werden. Ein Hinweis darauf, dass es unveröffentlichte Aktionen gibt, erfolgt nicht.

Die Zuordnung von Reviews zu Projekten ist optional und wird dadurch leicht übersehen. Dadurch kann es bei vielen Reviews schnell unübersichtlich werden, da die Filtermöglichkeiten nicht mehr greifen.
Da es kein Eclipse Plugin für Phabricator gibt, Codevorschläge nur über Kommentare möglich sind und der Code nicht ausführbar ist, fehlen einige weitere Punkte.
Ganze Dateien können nicht zu Reviews hinzugefügt werden, sondern nur Diff-Dateien. So ist

Kriterium	Punkte / max. Punkte
Allgemein	
Preis	5 / 5
Usability	15 / 25
	20 / 30
Must-have	
Verteilung von Rollen	3 / 3
Rollen Einschränkungen	5 / 6
Codevorschläge durch Reviewer	2 / 10
Globale Kommentare	4 / 5
Eclipse Plugin	0 / 10
Code / Testfälle ausführbar	2 / 8
Daten zum Review hinzufügen / löschen	4 / 5
Unterstützung von binären Dateien	1 / 4
Guter Support	4 / 6
Aktuell: regelmäßige Releases / Updates / Patches	5 / 5
	30 / 62
Nice-to-have	
Notifications	3 / 3
Review in SVN-Branches	0 / 2
Automatische Erzeugung von Branches	0 / 2
Schnittstelle für Dokumentation in Issuetracker	0 / 1
	3 / 8
Zusatzpunkte / Abzüge	
	0
Gesamtpunkte	**53 / 100**

Tabelle 3.6: Bewertungsmatrix Phabricator

es auch nicht sinnvoll möglich ausführbare Dateien dem Review hinzuzufügen.
Wird das Repository über Phabricator verwaltet, so sind Commitrechte einstellbar.

Tabelle 3.6 ist eine ausgefüllte Bewertungsmatrix.

Zusammenfassung

Phabricator ist ein sehr umfangreiches Tool. Wird es allerdings nur für Reviews verwendet, so erscheint es, zum Beispiel durch sein komplexes Benachrichtigungssystem, schnell überladen. Zentrale Features, wie ein Eclipse Plugin, das für die bessere Integration in die Entwicklung vorgesehen ist, fehlen.

Durch seine API ist Phabricator als Basis für eine Weiterentwicklung denkbar.

textbfUrteil: 53 Punkte → nicht empfehlenswert

3.3.6 Reviewboard

Reviewboard ist ein web-basiertes Reviewtool entwickelt von der Firma Beanbag Incorporation. Das Review - Tool ist kostenlos, bietet jedoch die Möglichkeit eines Power Packs für ca. 100$/Anwender/Jahr. Dieses Power Pack umfasst Github Support, die Möglichkeit auch Dokumente einem Review zu unterziehen, als auch für größere Unternehmen die Verwaltung von SSH-Keys etc.. Zudem werden Support-Verträge nach Anfrage gemacht. Es werden mehrere Versionskontrollen unterstützt, unter anderem git und SVN. Die neuste stabile Version ist 1.7.25, welche am 22.4.2014 erschienen ist. Ein release - Kandidat für die Version 2.0 ist bereit zum Download verfügbar.

Installation

Seit der Version 1.7 wird der offizielle Support für Windows eingestellt, da es zu viele Probleme mit Python gab. Es wird deshalb eine Installation auf Linux empfohlen.
Reviewboard selbst wird über ein Python Script installiert, jedoch müssen davor einige andere Programme installiert werden.
Als erstes braucht man sowohl eine Datenbank als auch einen Server. Hierbei werden als Datenbanken MySQL v5.0.31+, PostgreSQL und sqlite v3+ und als Server Apache mit mod_python, fastcgi oder mod_wsgi beziehungsweise lighttpd mit fastcgi unterstützt.
Bevor Reviewboard installiert wird, müssen noch folgende Programme installiert werden:

- Python Setuptools
- Python Development Headers
- memcached
- patch

Genaue Kommandos und Versionsnummern können der Installationsanweisung entnommen werden.

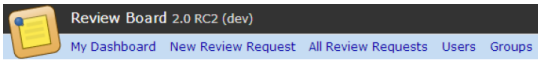

Abbildung 3.12: Reviewboard 1

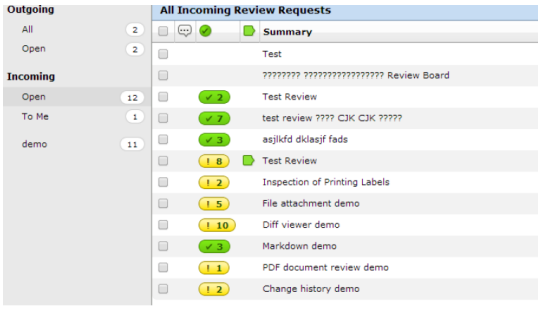

Abbildung 3.13: Reviewboard 2

Nutzung

Reviewboard basiert auf einer Web-UI. Die Benutzer können über einen LDAP-Server verwaltet werden. Diesen Benutzern kann der Administrator verschiedene Rechte geben, die er aus einer sehr überschaubaren, aber sinnvollen Menge auswählen kann.
Als Benutzer hat man nach dem einloggen 5 verschiedene Reiter. (Abbildung 3.12)

My Dashboard ist für jeden Benutzer selber konfigurierbar und zeigt alle Reviews an. In der linken Aktionsleiste lassen sich diese dann eingrenzen, zum Beispiel Reviewanfragen, die auf den Benutzer bezogen sind oder Reviews, die nicht geschlossen sind. In der Übersicht sieht man dann den Namen des Reviews, wie viele Probleme in diesem Review noch offen sind oder schon bearbeitet wurden und einiges andere mehr.

Unter New Review Request kann ein neues Review anlegt werden. Zuerst muss dazu eine Diff-Datei hochgeladen werden. Dies kann entweder manuell erstellt werden oder mit Hilfe von rbtools, einem Kommandozeilenprogramm für Reviewboard, welches kostenlos ist, hochgeladen werden. Im Anschluss kann man Beschreibungen und andere Informationen angeben(Abbildung 3.13) und anschließend die Reviewanfrage publizieren.
Die Reiter All Review Request zeigt alle Reviewanfragen an: Mit einer Beschreibung, dem Autor, dem Erstellungsdatum und wann sie zuletzt bearbeitet wurden.
Die Reiter Users und Groups beinhalten Listen von Nutzern und Gruppen.

Wenn man ein Review öffnet, sieht man eine Beschreibung und die angehängte Diff-Datei. Wenn diese anklickt wird, kann man sie entweder einzeln oder side-by-side mit einem anderen Stand vergleichen. Durch Klicken auf die Zeilenzahl bzw. Ziehen über einen bestimmten Bereich von Zeilenzahlen lässt sich ein Kommentar erstellen. Kommentare können dabei mit einer einfachen Markup - Sprache namens Markdown formatiert werden. Indem man 4 Leerzeichen voran stellt, lässt sich auch syntax-highlighted Code als Kommentar verfassen. In dem Kommentarfeld

Abbildung 3.14: Reviewboard 3

lässt sich auch einstellen, ob der Kommentar ein neues Problem eröffnen soll. Wenn der Autor daraufhin seinen Code verbessert, kann die Diff-Datei geupdatet werde. Erst wenn alle Probleme abgehandelt sind und auf „Ship it!“ gesetzt wurden kann das Review geschlossen werden.

Bewertung

Reviewboard ist in der Benutzung sehr einfach und verständlich mit Ausnahme des Hochladens einer Diff-Datei. Die Kommentare werden leider nur in Form einer Liste an ein Review angehängt, was bei vielen Kommentaren unübersichtlich ist. Die wichtigen Kommentare, welche ein Problem melden, werden jedoch oben separat verlinkt, wodurch diese leicht zu erreichen sind. Das Hochladen von Diff-Dateien ist nicht gut gelöst, jedoch durch rbtools wird es im Vergleich zu anderen Werkzeugen akzeptabel gelöst.
Die Dokumentation ist sehr ausführlich und gut erklärt, wird jedoch bei der Installation auch benötigt, da viele Sachen einzeln gemacht werden müssen. Da die Installation nicht täglich gemacht werden muss, wurde dafür nur 1 Punkt abgezogen. 3 weitere Punkte wurden dafür abgezogen, dass es teils zu Fehlern beim Einloggen kam. Ob dies an der Installation oder dem Programm an sich lag ist nicht auszumachen. Abzüge bei der Usability wurden unter anderem durch die Farbgestaltung gegeben. Dies sollte bei Benutzung des Tools im Code dringend geändert werden, da Code mit vielen Änderungen in leuchtendem grün angezeigt wird, was das längerfristige Arbeiten beziehungsweise Betrachten behindert.
Falls das Tool erweitert werden soll, zum Beispiel in Form eines Eclipse Plugins steht eine Rest-API zur Verfügung. Das Programm selbst ist in Ruby geschrieben. Die API ist in der Dokumentation gut beschrieben.
Das Powerpack lohnt sich hierbei unseres Erachtens nach für unseren Industriepartner nicht.

Tabelle 3.7 ist eine komplette Bewertung anhand der Bewertungsmatrix.

Kriterium	**Punkte / max. Punkte**
Allgemein	
Preis	5 / 5
Usability	18 / 25
	23 / 30
Must-have	
Verteilung von Rollen	3 / 3
Rollen Einschränkungen	3 / 6
Codevorschläge durch Reviewer	2 / 10
Globale Kommentare	2 / 5
Eclipse Plugin	0 / 10
Code / Testfälle ausführbar	0 / 8
Daten zum Review hinzufügen / löschen	5 / 5
Unterstützung von binären Dateien	4 / 4
Guter Support	5 / 6
Aktuell: regelmäßige Releases / Updates / Patches	5 / 5
	29 / 62
Nice-to-have	
Notifications	3 / 3
Review in SVN-Branches	0 / 2
Automatische Erzeugung von Branches	0 / 2
Schnittstelle für Dokumentation in Issuetracker	0 / 1
	3 / 8
Zusatzpunkte / Abzüge	
Stellenweise kleine Bugs	-3
	-3
Gesamtpunkte	**53 / 100**

Tabelle 3.7: Bewertungsmatrix Reviewboard

Zusammenfassung

Aufgrund des fehlenden Eclipse Plugin und anderen Schwächen bekommt es nur eine durchschnittliche Punktzahl, wäre jedoch als Grundlage einer Weiterentwicklung eventuell interessant.

Urteil: 53 Punkte → nicht empfehlenswert

3.3.7 MyLyn Review4Eclipse

Review4Eclipse ist ein Eclipse Plug-in, welches auf Mylyn Tasks aufbaut. Es ist ein kostenloses Tool und Open Source. Es fällt dabei unter die Eclipse Public License 1.0. Die neuste Version ist die 0.20, welche am 31.07.2013 veröffentlicht wurde. Das Plug-in ist auf Englisch.

Installation

Es müssen zwei Mylyn Veröffentlichungsseiten (http://download.eclipse.org/mylyn/releases/latest, http://download.eclipse.org/r4e/updates) unter „Help->Install new Software->Add“ eingetragen werden. Von diesen Seiten werden alle nötigen Bestandteile heruntergeladen. Für die Benutzung von SVN muss zusätzlich noch Subclipse über die Veröffentlichungsseite (http://subclipse.tigris.org/update_1.8.x) auf dieselbe Weise heruntergeladen werden und der passende MyLyn connector installiert werden. Zudem muss ein gemeinsamer Ordner für alle Benutzer dieses Plugins freigegeben werden. Die Benutzerverwaltung kann über einen LDAP-Server verwaltet werden.

Nutzung

Zuerst muss eine Reviewgruppe angelegt werden. Dieser Gruppe wird dann ein gemeinsamer Ordner zugewiesen. Der Gruppe können dann einzelne Personen oder Personenlisten hinzugefügt werden. Über einen Rechtsklick auf die Gruppe kann für diese Gruppe ein neues Review angelegt werden. Dabei können zwischen 3 verschiedenen Stufen von Reviews ausgewählt werden:

- Basic: Eine grundlegende Form des Reviews.
- Informal: Eine agile Form des Reviews.
- Formal: Ein Review nach dem IEEE Std. 1028. Hierbei wird zusätzlich Angeboten, dass die Planung und Einladungen für Treffen über das Tool geregelt werden.

Programmcode, welcher Teil des Reviews sein soll wird durch Rechtsklick auf die entsprechenden Klassen hinzugefügt. Es werden dabei auch Änderungen, die durch neue Commits entstanden sind, vorgeschlagen. Diffs kann man sich dabei side-by-side als auch einzeln anschauen. Für einzelne Bestandteile des Reviews lassen sich separat Leute hinzufügen, die diesen Bestandteil durchschauen sollen.
Um bei dem Review Fehler zu dokumentieren, kann man eine neue Anomalie an dieser Stellen

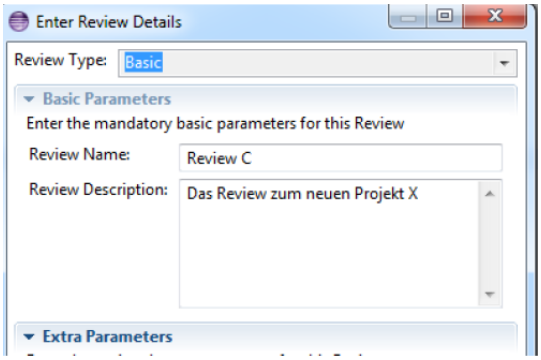

Abbildung 3.15: MyLyn 1

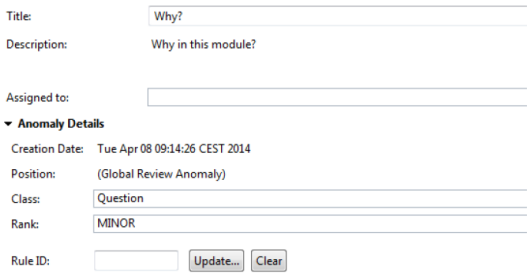

Abbildung 3.16: MyLyn 2

hinzufügen. Das geht sowohl global als auch lokal. Eine Anomalie wird im Tool wie folgt dargestellt.

Dabei kann für jede Anomalie die Klasse gewählt werden, die angibt, was mit dieser Anomalie ausgedrückt werden soll. Mögliche Einordnungen sind zum Beispiel Improvments und Missing. Zudem kann auch die Schwere des Verstoßes eingestellt werden. Auf Anomalien kann man mit einem Kommentar antworten.
Review4Eclipse bietet auch die Möglichkeit Richtlinie anzugeben, wie der Code aufgebaut sein muss. Auch für diesen Punkt lassen sich Anomalien erstellen mit einem Verweis auf den begangenen Verstoß.
Ist man mit dem Review eines Teiles des Reviews fertig, kann man diesen separat als bereits bearbeitet markieren oder aber auch das gesamte Review direkt als bearbeitet seinerseits markieren.
Es lassen sich über einen SMTP Server halb-automatisch Mitteilungen versenden. Dabei wird vom Benutzer die Art der Mitteilung ausgewählt. Man kann zum Beispiel an alle Teilnehmer des Reviews eine Mitteilung schicken, dass das Review fertiggestellt ist oder man kann dem Autor direkt Fragen zu dem Prüfling stellen ohne es direkt als Anomalie für alle sichtbar zu machen.

Während und vor allem am Ende des Reviews lässt sich ein html-Bericht automatisch erstellen, der alle wichtigen Daten des Reviews als auch statistische Daten des Reviews darstellt.

Bewertung

Die Bedienung von Review4Eclipse ist bei den Standardaufgaben sehr einfach zu verstehen und schnell erlernt. Jedoch sind einige Bereiche nicht optimal gelöst, teils sind sie auch dem Unterbau von MyLyn geschuldet. So ist die allgemeine Darstellung eher unübersichtlich, da alles im gleichen, einfachen Stil gehalten ist und nichts hervorgehoben ist(vergleiche Abbildung 2). Ebenso ist es nicht optimal, dass zuerst manuell ein gemeinsamer Ordner für jede Reviewgruppe angelegt werden muss.
Zudem gibt es einige kleinere Bugs, die beim Arbeiten mit Review4Eclipse aufgefallen sind, die zwar nicht kritisch sind, jedoch die Freude am Arbeiten mit dem Tool mindern. So lässt sich zum Beispiel sobald die beiden Hauptfenster im selben View liegen, erst das Review auswählen, nachdem der erste Klick ein auf das andere Fenster gewechselt hat und man manuell wieder in die Übersicht geht und erneut den Klick ausführt. Wenn solche Sachen jedoch beachtet werden, lässt sich das Plugin gut bedienen.

In Tabelle 3.8 findet sich eine ausgefüllte Bewertungsmatrix für Review4Eclipse.

Zusammenfassung

Im Gesamten gesehen bietet es jedoch viele der geforderten Funktionen. Manche können leider nur über Umwege erreicht werden, wie zum Beispiel, dass der Code erst ausführbar wird, wenn man sich den neuen Commit holt. Da man jedoch schon in der IDE ist, geht dies schneller als bei web-basierten Tools. Es ist eine gute Dokumentation vorhanden, welche auch ein bebildertes Tutorial enthält. Der letzte Commit zu dem Projekt liegt schon ca. 9 Monate zurück.

Urteil: Urteil: 64 Punkte → nicht bis bedingt empfehlenswert

3.3.8 Gerrit mit MyLyn Connector

Gerrit ist ein webbasiertes Reviewsystem für git. Gerrit wurde ursprünglich von Google entwickelt, ist inzwischen jedoch ein Open-Source Projekt und ist unter der Apache 2 Lizenz verfügbar.

Die Serverkomponente von Gerrit ist in Java implementiert und kann somit auf allen gängigen Betriebssystemen genutzt werden. Das Userfrontend ist webbasiert und somit auch palttformunabhängig.
Das Eclipse Plugin wird unabhängig von Gerrit entwickelt. Entwickler ist Tasktop Technologies.

Kriterium	Punkte / max. Punkte
Allgemein	
Preis	5 / 5
Usability	18 / 25
	23 / 30
Must-have	
Verteilung von Rollen	3 / 3
Rollen Einschränkungen	2 / 6
Codevorschläge durch Reviewer	3 / 10
Globale Kommentare	5 / 5
Eclipse Plugin	10 / 10
Code / Testfälle ausführbar	2 / 8
Daten zum Review hinzufügen / löschen	5 / 5
Unterstützung von binären Dateien	4 / 4
Guter Support	3 / 6
Aktuell: regelmäßige Releases / Updates / Patches	2 / 5
	39 / 62
Nice-to-have	
Notifications	2 / 3
Review in SVN-Branches	0 / 2
Automatische Erzeugung von Branches	0 / 2
Schnittstelle für Dokumentation in Issuetracker	0 / 1
	2 / 8
Zusatzpunkte / Abzüge	
	0
Gesamtpunkte	**64 / 100**

Tabelle 3.8: Bewertungsmatrix MyLyn Review4Eclipse

Das Eclipse Plugin ist ein MyLyn Connector, verwendet also für die lokale Taskverwaltung das MyLyn Plugin. Außerdem wird das EGit Plugin benötigt. Alle drei Plugins sind unter der Eclipse Public License veröffentlicht.

Die Plugins sind in Java implementiert und an Eclipse gebunden. Somit sind sie in jeder Umgebung, die von Eclipse unterstützt wird, lauffähig.

Installation

Die Installation der Eclipse Plugins erfolgt wie gewohnt über das Plugin System von Eclipse. Als Download URL fungiert für MyLyn und den Gerrit MyLyn Connector die MyLyn Download Adresse 0. Die Download URL für das EGit Plugin ist http://download.eclipse.org/egit/updates.
Auf Serverseite muss das JDK 1.6 vorhanden sein. Weiterhin wird für die Nutzung als Livesystem eine Datenbank (MySql, PostgreSQL, Oracle), sowie ein Webserver (Apache 2, nginx, ...) empfohlen.

Wird eine Datenbank verwendet, so muss eine Datenbank für Gerrit angelegt werden. Weiterhin wird ein User benötigt, der für entsprechende Datenbank alle Rechte besitzt.

Die Installationsroutine ist ein Konsolenprogramm in Java. Diesem wird das Installationsverzeichnis übergeben. Während des Installationsprozesses wird die Datenbank konfiguriert. Im Anschluss daran kann Gerrit verwendet werden.

Für den dauerhaften Betrieb auf einem Server empfiehlt es sich die Gerrit Serversoftware automatisch beim Systemstart zu starten. Außerdem ist es empfehlenswert Gerrit hinter einem Reverse Proxy eines Webservers, zum Beispiel Apache oder nginx, zu betreiben. Entsprechende Anleitungen liegen der Gerrit Installationsanleitung in der Gerrit Dokumentation bei.

Standardmäßig nutzt Gerrit OpenID als Authentifizierungsmethode. Diese kann jedoch per Konfiguration auf LDAP umgestellt werden. Entsprechende Anleitungen finden sich ebenfalls in der offiziellen Gerrit Dokumentation.
Der erste Benutzer, der sich einloggt wird als Administrator angelegt.

Nutzung

Für die Verwendung von Gerrit ist es notwendig, dass das komplette Repository Hosting über Gerrit erfolgt.
Das Erzeugen eines Reviews erfolgt durch das Pushen eines Commits in die Gerrit Staging Area. Hierfür muss in den Branch HEAD:refs/for/< BRANCH> gepushed werden. Gerrit erzeugt dann ein Review mit einem ersten Patch Set. Über die Weboberfläche können die Änderungen via Side-by-side oder unified Diff eingesehen und kommentiert werden. Kommentare sind für die Commit - Nachricht, Dateien, das Review, einzelne Codeblöcke oder für Patch Sets möglich.

Sind Änderungen notwendig so kann über den git commit –amend Befehl ein weiteres Patch Set zum Review hinzugefügt werden. Dieses kann dann wie das erste Changeset kommentiert werden.

Weiterhin kann eine Gesamtbewertung in Form einer Punktzahl angegeben werden. Diese liegt mit Standardeinstellungen zwischen -2 und +2. Ab einer bestimmten Punktzahl wird ein Commit für einen Merge vorgesehen.
Das Eclipse Plugin bietet dieselbe Funktionalität wie die Weboberfläche. Alle offenen Reviews werden dabei in der MyLyn Task Liste angezeigt. Für den Vergleich der Versionen wird der Diff-Editor von Eclipse herangezogen. Außerdem ist es möglich den lokalen Stand auf den eines bestimmten Patch Sets zu bringen.

Der Gerrit Connector erweitert außerdem die Funktionalität des EGit Plugins, was den Umgang mit den Pushs in bestimmte Review-Branches etc. vereinfacht.

Bewertung

Gerrit und auch der Eclipse MyLyn Connector sind aufgrund der vielen Features zunächst sehr komplex. Weiterhin ist der Workflow mit den speziellen Branches und dem hinzufügen weiterer Patch Sets zu einem Review ziemlich umständlich. Außerdem ist die Darstellung sehr verschachtelt, was keinen schnellen Überblick ermöglicht. Aus diesen Gründen fehlen Punkte bei der Usability.
Codevorschläge sind nur über Kommentare möglich. Die Ausführung des Codes ist nur möglich, wenn über das Eclipse Plugin der lokale Stand manuell auf den des zu reviewenden Patch Sets gebracht wird.

Tabelle 3.9 ist eine ausgefüllte Bewertungsmatrix für Gerrit.

Zusammenfassung

Gerrit ist von allen Tools am besten bewertet und bietet wichtige Teile der geforderten Funktionalität. Insbesondere durch das sehr gute Eclipse Plugin ist Gerrit sehr gut dazu geeignet die Code Reviews besser in das gewohnte Entwicklungsumfeld zu integrieren.

Gerrit ist jedoch sehr tief mit git verwachsen, wodurch eine Nutzung von git - SVN oder ähnlichen Brücken für SVN nicht möglich ist. Für die Nutzung ist also eine Umstellung von SVN auf git notwendig.

Urteil: 75 Punkte, unterstützt aber nur git → keine Empfehlung

Kriterium	Punkte / max. Punkte
Allgemein	
Preis	5 / 5
Usability	19 / 25
	24 / 30
Must-have	
Verteilung von Rollen	3 / 3
Rollen Einschränkungen	6 / 6
Codevorschläge durch Reviewer	3 / 10
Globale Kommentare	5 / 5
Eclipse Plugin	10 / 10
Code / Testfälle ausführbar	2 / 8
Daten zum Review hinzufügen / löschen	5 / 5
Unterstützung von binären Dateien	4 / 4
Guter Support	4 / 6
Aktuell: regelmäßige Releases / Updates / Patches	5 / 5
	47 / 62
Nice-to-have	
Notifications	3 / 3
Review in SVN-Branches	0 / 2
Automatische Erzeugung von Branches	0 / 2
Schnittstelle für Dokumentation in Issuetracker	1 / 1
	4 / 8
Zusatzpunkte / Abzüge	
	0
Gesamtpunkte	**75 / 100**

Tabelle 3.9: Bewertungsmatrix Gerrit

4 Zusammenfassung & Empfehlung

4.1 Zusammenfassung

Die Analyse und Bewertung der Tools hat gezeigt, dass kein Produkt Öut of the boxälle Anforderungen erfüllt. Insbesondere bei den Must-have-Features "Codevorschläge durch Reviewer", Ëclipse Pluginünd "Code/Testfälle ausführbarërreichen nur sehr wenige Tools gute Punktzahlen. Außerdem sind viele Tools veraltet und werden nicht mehr weiterentwickelt.

CodeBrag merkt man schnell an, dass sich das Programm derzeit noch in der Entwicklung befindet. Es sind kaum Features vorhanden, wodurch kein Live-Einsatz möglich ist.

ReviewMate liefert gute Ansätze, wie zum Beispiel die komplette Integration des Reviews in die Entwicklungsumgebung. Für die Praxis ist das Tool jedoch nicht durchdacht. Es gibt keine zuverlässige Möglichkeit, dass mehrere User an einem Review teilnehmen. Außerdem ist eine Dokumentation eines Reviews nicht möglich.

Barkeep erfüllt nur sehr wenige Anforderungen und ist sehr umständlich zu bedienen. Außerdem ist es ein Tool, das nur für git - Repositories funktioniert.

Cahoots lässt sich ebenfalls via Plugin in Eclipse integrieren. Weder die Weboberfläche noch das Eclipse Plugin erscheinen wirklich ausgereift. Ein Review in der Entwicklungsumgebung ist ebenfalls nicht möglich.

Phabricator verliert insbesondere durch seine Komplexität und seine daraus resultierende Unübersichtlichkeit an Punkten. Außerdem gibt es kein Eclipse Plugin, das eine einfache Integration des Reviews in die gewohnte Entwicklungsumgebung ermöglichen würde.

Reviewboard lief in unserer Testumgebung nicht immer stabil. Zwar traten dabei keine schwerwiegenden Fehler auf, insbesondere keine, die sich nicht durch Aktualisieren der Seite lösen ließen, aber störend ist dies trotzdem. Wie auch für Phabricator gibt es kein Eclipse Plugin.

MyLyn Review4Eclipse ist ein reines Eclipse Plugin. Die Funktionalität ist prinzipiell gut, allerdings ist die Bedienung oftmals komplex und umständlich. Dies liegt vor allem daran, dass alles lokal verwaltet und eingestellt werden muss, da es keinen Server im Hintergrund gibt.

Gerrits größte Schwäche ist die fehlende SVN Unterstützung. Durch die starke Abhängigkeit des Gerrit-Workflows zu git, müsste die komplette Entwicklung bei AEB auf git umgestellt werden. Insbesondere das gute Eclipse Plugin, sowie der gesamte Funktionsumfang sind bei Gerrit allerdings sehr überzeugend. Schwächen liegen vor allem in der Komplexität der Benutzeroberflächen, die durch die vielen Funktionen schnell unübersichtlich wirkt.

4.2 Empfehlung

Da bei AEB momentan SVN eingesetzt wird und ein Umstieg auf git nicht geplant ist, können wir keine Empfehlung für ein Tool aussprechen. Lediglich Gerrit erreicht eine, aus unserer Sicht, für eine Empfehlung geeignete, Punktzahl.
Aus unserer Sicht bieten sich vier theoretisch mögliche Empfehlungen an:

1. Weitermachen wie bisher
2. Verzicht auf ein Eclipse Plugin
3. Eigenentwicklung eines Tools / Eclipse Plugins
4. Umstieg auf git

Weitermachen wie bisher

Diese Empfehlung ist nicht sinnvoll. Die Fachstudie wurde aus gutem Grund gestartet, da die aktuelle Situation in vielerlei Hinsicht nicht zufriedenstellend ist. Insbesondere die schlechte Nachvollziehbarkeit und die nicht gegebene Prozesssicherheit würden bleiben.

Verzicht auf ein Eclipse Plugin

Theoretisch ist diese Lösung zwar denkbar, da einige Tools deutlich besser abschneiden würden, wenn man auf das Eclipse Plugin verzichten würde. Allerdings ist die Praxistauglichkeit dieser Lösung fragwürdig. Reviews bleiben umständlich, da keine Integration in die gewohnte Entwicklungsumgebung vorhanden ist. Es ist also zu befürchten, dass aus Bequemlichkeit weiterhin Diff-Dateien per E-Mail ausgetauscht würden oder Code-Abnahmen Side-by-side stattfinden würden.

Eigenentwicklung eines Tools / Plugins

Die Eigenentwicklung eines Programms ist immer mit hohen Kosten verbunden. Insbesondere müssen dabei auch die langfristige Weiterentwicklung und Wartung bedacht werden.

Billiger als eine komplette Neuentwicklung ist die Entwicklung eines Eclipse Plugins auf Basis eines vorhandenen Reviewprogramms.

Sowohl Phabricator als auch Reviewboard haben eine umfangreiche JSON REST-API über die Reviews erstellt, Reviewdaten ausgelesen und bearbeitet werden können. In diesem Fall fällt die komplette Backend Entwicklung weg, was eine Neuentwicklung billiger macht. Jedoch entsteht von der angebotenen API eine große Abhängigkeit, deren Entwicklung außerhalb des Einflusses von AEB liegt. Die Wartungs- und Instandhaltungskosten sind daher kaum abzuschätzen.

Die Reviewboard API bietet einen bedeutend größeren Funktionsumfang, als die von Phabricator. Außerdem haben die Phabricator Entwickler eine Überarbeitung der API angekündigt. Die Entwicklung eines Eclipse Plugins ist momentan folglich nur auf Basis von Reviewboard sinnvoll.

Umstieg auf git

Eine Umstellung auf git würde die Nutzung von Gerrit ermöglichen. Allerdings ist die Umstellung auf ein anderes Versionskontrollsystem mit einigem Aufwand und somit auch mit Kosten verbunden, die zusätzlich zum Aufwand der Einführung eines neuen Review-Workflows und Reviewtools auf AEB zukommen.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift

Ort, Datum, Unterschrift

Ort, Datum, Unterschrift