

Qualität in Echtzeit mit Teamscale*

Nils Göde, Lars Heinemann, Benjamin Hummel, Daniela Steidl

CQSE GmbH

Lichtenbergstr. 8, 85748 Garching bei München

{goede, heinemann, hummel, steidl}@cqse.eu

Zusammenfassung

Existierende Werkzeuge für statische Qualitätsanalysen arbeiten im Batch-Modus. Die Analyse benötigt für jede Ausführung eine gewisse Zeit, was dazu führt, dass Entwickler sich oftmals bereits mit anderen Themen beschäftigen wenn die Ergebnisse verfügbar sind. Zudem lässt sich aufgrund der getrennten Ausführungen nicht zwischen alten und neuen Qualitätsdefiziten unterscheiden – eine Grundvoraussetzung für Qualitätsverbesserung in der Praxis. In diesem Artikel stellen wir das Werkzeug *Teamscale* vor, mit dem sich Qualitätsdefizite zuverlässig während der Evolution des analysierten Systems verfolgen lassen. Durch die inkrementelle Funktionsweise stehen Analyseergebnisse wenige Sekunden nach einem Commit zur Verfügung, wodurch sich Qualität in Echtzeit überwachen und steuern lässt.

1 Einleitung

Es existiert bereits eine Vielzahl an statischen Analysewerkzeugen um Qualitätsdefizite in Softwaresystemen aufzudecken. Hierzu zählen unter anderem ConQAT [5], SonarQube [8] und die Bauhaus Suite [4]. Obwohl diese Werkzeuge eine umfangreiche Auswahl an Analysen bieten, haben sie doch zwei Probleme gemein. Zunächst werden die Werkzeuge im Batch-Modus ausgeführt. Bei aufeinanderfolgenden Analysen wird das komplette System neu analysiert – selbst wenn sich nur wenige Teile geändert haben. In der Zeit, die vom Anstoßen der Analyse bis zur Verfügbarkeit der Ergebnisse vergeht, beschäftigen sich Entwickler oftmals schon mit anderen Themen. Das führt dazu, dass die aufgedeckten Qualitätsdefizite in der Praxis oft keine unmittelbare Beachtung finden. Die Wahrscheinlichkeit, dass ein solches Defizit zu einem späteren Zeitpunkt behoben wird, ist sehr gering.

Das zweite große Problem besteht darin, dass die getrennte Ausführung der Analysen für verschiedene Versionsstände kein zuverlässiges Verfolgen von Qualitätsdefiziten über die Zeit zulässt. Zwar kann im

Nachgang versucht werden die Ergebnisse verschiedener Analysen aufeinander abzubilden, dies führt aber in den meisten Fällen zu ungenauen oder unvollständigen Ergebnissen aufgrund fehlender Informationen (z.B. Umbenennung von Dateien). Zudem benötigt die Abbildung zusätzliche Zeit.

Für die kontinuierliche Qualitätsverbesserung ist es notwendig, dass Entwickler ohne nennenswerte zeitliche Verzögerung nach Durchführung ihrer Änderungen über neue Probleme informiert werden. Dabei ist es ebenso wichtig, dass ein Qualitätsdefizit nur dann als neu markiert wird, wenn dies auch der Tatsache entspricht und es sich nicht um ein schon lange bestehendes Legacy-Problem handelt. Sowohl das zeitliche Problem als auch das Verfolgen von Qualitätsdefiziten wird von dem Werkzeug *Teamscale* [6, 7] gelöst.

2 Teamscale

Teamscale ist ein inkrementelles Analysewerkzeug das die kontinuierliche Qualitätsüberwachung und -verbesserung unterstützt. *Teamscale* analysiert einen Commit innerhalb von Sekunden und informiert somit in Echtzeit darüber, wie sich aktuelle Änderungen auf die Qualität – insbesondere die Wartbarkeit des Quelltextes – ausgewirkt haben. *Teamscale* speichert die komplette Qualitätshistorie des Systems und ermöglicht es dadurch mit minimalem Aufwand herauszufinden wann und wodurch Qualitätsdefizite entstanden sind. Durch den Einsatz verschiedener Heuristiken werden Defizite auch verfolgt wenn diese sich zwischen Methoden oder Dateien bewegen [10]. Die zuverlässige Unterscheidung von neuen und alten Qualitätsdefiziten erlaubt es sich auf kürzlich eingeführte Defizite zu konzentrieren. Die Weboberfläche und IDE Plug-Ins stellen die Informationen Entwicklern und anderen Beteiligten zur Verfügung.

Architektur

Teamscale ist eine Client-Server-Anwendung. Die Architektur ist in Abbildung 1 skizziert. *Teamscale* verbindet sich direkt mit dem Versionskontrollsystem (Subversion, GIT oder TFS von Microsoft) oder dem Dateisystem. *Teamscale* unterstützt u.a. Java, C#, C/C++, JavaScript, ABAP oder auch Projekte mit einer Kombination dieser Sprachen. Zudem können

*Das diesem Artikel zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **EvoCon, 01IS12034A** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

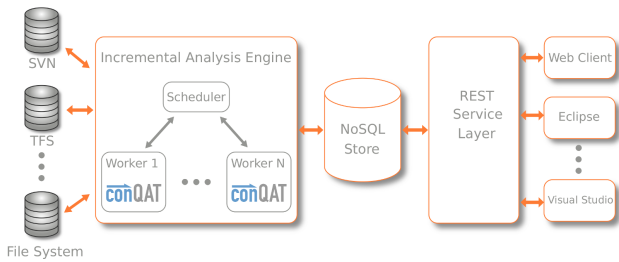


Abbildung 1: *Teamscale* Architektur

auch Informationen über Bugs und Change Requests integriert werden. Unterstützt werden u.a. Jira, Redmine und Bugzilla.

Die inkrementelle Analyse-Engine baut auf dem Werkzeug ConQAT [5] auf und wird für jeden Commit im angebundenen Versionskontrollsystem aktiv. Dabei werden nur die von Änderungen betroffenen Dateien neu analysiert, wodurch die Ergebnisse innerhalb von Sekunden zur Verfügung stehen. Durch die Analyse jedes einzelnen Commits lassen sich Qualitätsdefizite präzise anhand der Änderungen am Code verfolgen. Die Ergebnisse werden zusammen mit der Historie jeder Datei in einer noSQL-Datenbank, wie z.B. Apache Cassandra [11], abgelegt und über eine REST-Schnittstelle Clients zur Verfügung gestellt.

Clients

Teamscale beinhaltet einen JavaScript-basierten Webclient, der verschiedene Sichten auf die Evolution des Systems und dessen Qualität bietet um verschiedenen Rollen (z.B. Entwickler oder Projektleiter) gerecht zu werden. Die Sichten umfassen eine Übersicht der Commits und deren Auswirkungen auf die Qualität, eine Übersicht über den Code (ganz oder in Teilen), eine Delta-Sicht zum Vergleich mit einem früheren Code-Stand und ein frei konfigurierbares Dashboard.

Zusätzlich beinhaltet *Teamscale* Plug-Ins für Eclipse und Visual Studio. Qualitätsdefizite werden in der IDE am Code annotiert. Dadurch können Entwickler neue Qualitätsdefizite unmittelbar in der jeweiligen Entwicklungsumgebung inspizieren und beheben.

3 Analysen

Teamscale implementiert eine Vielzahl bekannter Qualitätsanalysen. Die zentralen Analysen, die unterstützt werden, sind im folgenden aufgezählt.

Strukturmetriken. *Teamscale* erhebt zentrale Strukturmetriken, wie die Länge von Dateien, die Länge von Methoden und die Schachtelung des Codes.

Klonerkennung. *Teamscale* führt eine inkrementelle Klonerkennung durch um durch Copy und Paste erzeugte redundante Codeabschnitte zu erkennen.

Code Anomalien. *Teamscale* untersucht das System hinsichtlich Code Anomalien, wie z.B. Verstöße gegen Coding Guidelines und häufige Fehlermuster. Zudem können externe Werkzeuge wie z.B. Find-

Bugs [1], PMD [2] und StyleCop [3] eingebunden und deren Ergebnisse importiert werden.

Architekturkonformität. Sofern eine geeignete Architekturspezifikation vorliegt, die die Komponenten des Systems und deren Abhängigkeiten beschreibt, kann *Teamscale* diese mit der Implementierung vergleichen um Abweichungen aufzudecken.

Kommentierung. *Teamscale* beinhaltet eine umfangreiche Analyse der Kommentare im Code [9]. Hierbei wird u.a. geprüft ob bestimmte Vorgaben eingehalten werden (ist z.B. jedes Klasse kommentiert) und ob Kommentare trivial oder inkonsistent sind.

4 Evaluation

Eine erste Evaluation von *Teamscale* wurde als Umfrage unter professionellen Entwicklern bei einem unserer Evaluierungspartner aus der Industrie durchgeführt [7]. Laut Aussage der Entwickler bietet *Teamscale* ihnen einen guten Überblick über die Qualität ihres Codes und erlaubt es auf einfache Weise aktuelle Probleme von Legacy-Problemen zu trennen. Eine umfassendere Evaluation steht noch aus.

5 Zusammenfassung

Teamscale löst zwei große Probleme bestehender Analysewerkzeuge. Durch die inkrementelle Vorgehensweise stehen die Analysergebnisse in Echtzeit zur Verfügung. Entwickler sind so stets über die Auswirkungen ihrer aktuellen Änderungen informiert und können neu entstandene Qualitätsdefizite gleich beseitigen. Zudem bietet *Teamscale* eine vollständige Verfolgung von Qualitätsdefiziten durch die Historie. Dadurch kann zuverlässig zwischen alten und neuen Qualitätsdefiziten unterschieden und die Ursache von Problemen mit minimalem Aufwand ermittelt werden.

Referenzen

- [1] FindBugs. <http://findbugs.sourceforge.net>.
- [2] PMD. <http://pmd.sourceforge.net>.
- [3] StyleCop. <https://stylecop.codeplex.com>.
- [4] Axivion GmbH. Bauhaus Suite. www.axivion.com.
- [5] CQSE GmbH. ConQAT. www.conqat.org.
- [6] CQSE GmbH. Teamscale. www.teamscale.com.
- [7] L. Heinemann, B. Hummel, and D. Steidl. Teamscale: Software quality control in real-time. In *Proceedings of the 36th International Conference on Software Engineering*, 2014. Accepted for publication.
- [8] SonarSource S.A. SonarQube. www.sonarqube.org.
- [9] D. Steidl, B. Hummel, and E. Juergens. Quality analysis of source code comments. In *Proceedings of the 21st IEEE International Conference on Program Comprehension*, 2013.
- [10] D. Steidl, B. Hummel, and E. Juergens. Incremental origin analysis of source code files. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, 2014. Accepted for publication.
- [11] The Apache Software Foundation. Apache Cassandra. <http://cassandra.apache.org/>.