

TEAMSCALE – Die Software-Intelligence-Plattform*

Elmar Juergens, Roman Haas

CQSE GmbH

Lichtenbergstr. 8, 85748 Garching bei München

{juergens, haas}@cqse.eu

Zusammenfassung

In diesen Artikel stellen wir TEAMSCALE vor, unsere Plattform für Software-Intelligence. Dabei skizzieren wir die Architektur, gehen beispielhaft auf typische Fragestellungen ein, die TEAMSCALE beantworten kann und verweisen auf weiterführende Literatur. Teamscale ist für Forschung, Lehre und Open-Source Systeme frei unter www.teamscale.io verfügbar.

1 Software Intelligence

In Anlehnung an *Business Intelligence* werden bei *Software Intelligence* Rohdaten ausgewertet und so analysiert, dass verständliche und nützliche Informationen über Software und deren Entwicklungsprozess gewonnen werden. So können Fragen zur Softwareentwicklung auf Basis der eigenen Daten quantitativ beantwortet werden.

2 Teamscales Architektur

Zentrale Idee hinter TEAMSCALE ist, relevante Daten an einer zentralen Stelle zu sammeln, unmittelbar zu analysieren und zugänglich zu machen. TEAMSCALE verarbeitet Daten aus Versionskontrollsystemen, Issue-Trackern und diversen externen Analysetools (s. Abb. 1). Die Daten werden für die komplette Historie und alle Branches vorgehalten.

3 Beispiele

Im Folgenden gehen wir auf typische Fragestellungen ein. Diese und viele weitere lassen sich mit TEAMSCALE beantworten.

3.1 Statische Code-Analyse

Wie erhalte ich als Entwickler Feedback? TEAMSCALE gibt Entwicklern in Echtzeit Feedback und das Commit-spezifisch für ihren Entwicklungs-Branch [1].

Haben wir Kopfmonopole im Team? Die Auswertung der Versionshistorie deckt Kopfmonopole auf, wodurch rechtzeitig gegengesteuert werden kann.

*Das diesem Artikel zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **Q-Effekt, 01IS15003A** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

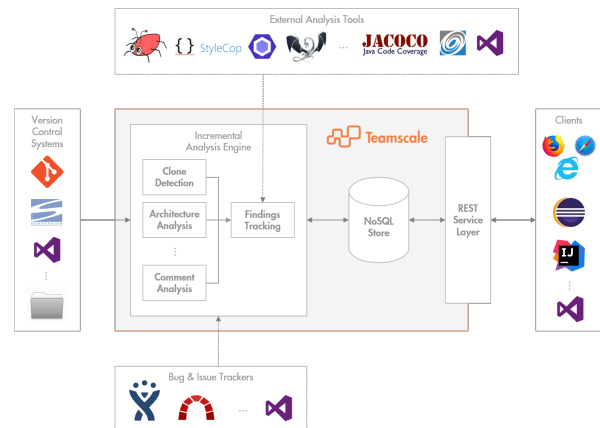


Abbildung 1: Teamscale Architektur

3.2 Test-Gap-Analyse

Rutschen uns wichtige Änderungen ungetestet durch? TEAMSCALE findet Code-Änderungen, die in manuellen oder automatisierten Test nicht getestet wurden [3], um rechtzeitige Nachtests zu erlauben.

Sind die Code-Änderungen für mein Ticket getestet? TEAMSCALE verknüpft Daten aus dem Issue-Tracker mit den Testausführungen, sodass für jedes Ticket erkennbar ist, was noch nicht getestet wurde.

3.3 Nutzungsanalyse

Haben wir nutzlosen Code? TEAMSCALE deckt Code auf, der zur Laufzeit überhaupt nicht ausgeführt wird [2] und daher vermutlich gelöscht werden kann.

Referenzen

- [1] V. Bauer, L. Heinemann, B. Hummel, E. Juergens, and M. Conradt. A framework for incremental quality analysis of large software systems. In *Proc. of ICSM*, 2012.
- [2] A. Göb. How much of your code do you actually use? Blog, June 2016. <http://cqse.eu/blog-code-usage>.
- [3] E. Juergens and D. Pagano. Haben wir das Richtige getestet? Erfahrungen mit Test-Gap-Analyse in der Praxis. In *Software-QS-Tag*, 2016. <http://cqse.eu/test-gap-paper>.