

Metriken, Statistiken, Dashboards

(Link zum Artikel: <http://www.it-republik.de/dotnet/artikel/3072>)

Glaube keiner Statistik, die du nicht selbst gefälscht hast

Text: von Uta Kapp und Jean Pierre Berchez

Statistiken und Metriken sagen viel über einen Projektstand aus. Jede Gruppe in einem Software-Entwicklungs-Team hat Informationsbedürfnisse auf einer anderen Abstraktionsebene. So braucht ein Teammanager Informationen bis auf Source Code Ebene. Während ein Benutzer vor allem daran interessiert ist, wie schnell eine Anforderung im Durchschnitt realisiert oder ein Bug repariert ist. Metriken bringen Klarheit über den Projektstand.

Der Vorteil einer integrierten Collaboration-Plattform ist, dass alle Daten zu einem Projekt in digitaler Form vorliegen und deshalb ausgewertet werden können. So werden Berichte und Statistiken aus Trackern erzeugt. Mit Metriken lässt sich ermitteln, wie hoch die Fehlerraten sind, ob Meilensteine regelmäßig nicht eingehalten werden und ob der Projektfortschritt gleich bleibend ist. Jede erdenkliche Auswertung ist möglich und zu einer Metrik komprimierbar. Qualitätsmerkmale, wie z.B. die Abdeckung des Codes mit Tests oder die Anzahl offener Fehler werden gemessen. Diese Erkenntnisse sind dann zur Teamreflexion nutzbar und werden für die Neuausrichtung des Projektes genutzt. Das schafft Transparenz. Allgemeine Aussagen zu einem Projekt im Zeitverlauf sind viel wertvoller und ergeben einen besseren Überblick über die Fließgeschwindigkeit eines Projektes, als personenbezogene Informationen. Um qualitativ hochwertige Aussagen zu bekommen, müssen die Daten von allen Beteiligten im Verlauf des Projektes gleichmäßig gut gepflegt und somit vergleichbar sein. Da der Mensch ein Bedeutung erzeugendes Wesen ist, werden Handlungen nur dann ausgeführt, wenn ihnen ein Sinn beigemessen wird. Damit die Informationen in einer Collaboration-Plattform ausreichend detailliert gepflegt werden, muss für die Betroffenen der eigene Vorteil erkennbar sein.

Dashboards Für einen schnellen Überblick zu dem augenblicklichen Projektstand dienen Dashboards. Sie sind eine Zusammenstellung der wichtigsten Statistiken und Metriken des Projektes, meist grafisch dargestellt. Dashboards werden für ein Projekt maßgeschneidert. Beispiele hier sind:

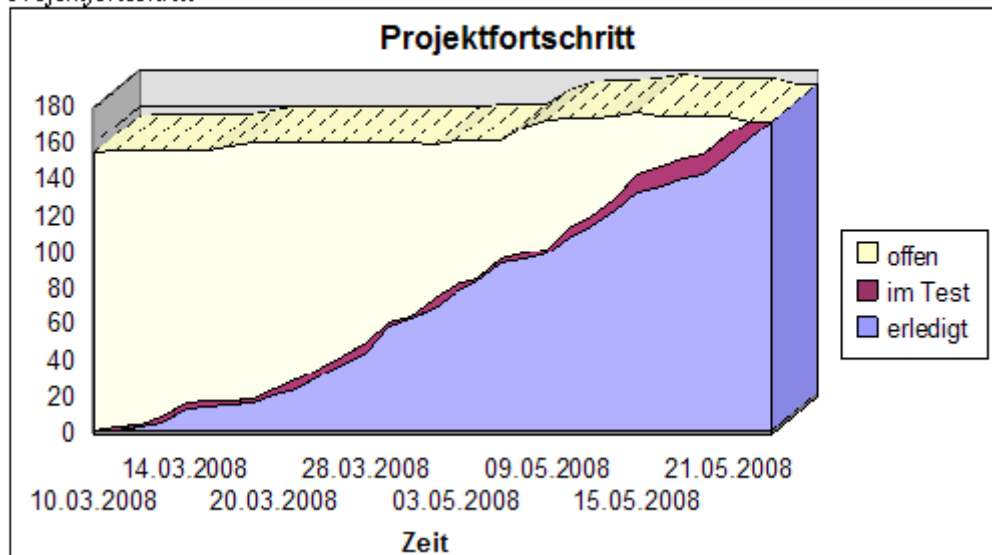
- unerledigte Aufgaben
- Anzahl Zeilen Code täglich produziert
- Dateien täglich geändert
- bereits erledigter Prozentanteil eines Projektabschnitts oder einer Iteration.

Die Bug-Fix Statistik Die Fehlerbehebungsrate wird von einer Bug-Fix Statistik sinnvoll dargestellt. Wie viele reparierte Bugs wurden in der aktuellen Time-Box bereits getestet und sind damit wirklich erledigt? Es ist ziemlich leicht, Metriken positiv zu beeinflussen. Um qualitativ hochwertige Aussagen zu einem Projekt zu bekommen, ist es sinnlos Statistiken zu den Leistungen eines einzelnen Projektmitarbeiters zu generieren. Es würde nur dazu führen, dass unnötig viel Zeit damit verbracht wird, die Metriken gut aussehen zu lassen, statt sich auf das Projektziel zu konzentrieren. Allgemeine Aussagen zu einem Projekt über

einen längeren Zeitraum sind viel wertvoller und ergeben bessere Aussagen z.B. über die Fließgeschwindigkeit eines Projektes, als persönliche Statistiken. Jeder Mitarbeiter hat seine eigene Arbeitsgeschwindigkeit und sowohl der gründliche, als auch der schnelle Entwickler bekommt seine speziellen Aufgaben.

Projektfortschritt als Flussdiagramm Damit der Projektfortschritt in einem Projekt messbar wird, werten wir die Daten der Anforderungs- und Bug-Tracker aus. Wir benötigen dazu Daten über den geschätzten und den getätigten Aufwand, als auch den Status einer Anforderung. Wir können so ein kumuliertes Flussdiagramm erzeugen in dem verbleibender und schon erledigter Aufwand bezüglich eines zu erreichenden Meilensteins angezeigt wird. Der offene Anteil stellt die Summe aller zu bearbeitenden Aufwandspunkte der Features dar. Im Test heißt, kodiert und zu Test freigegeben. Erledigt meint, den implementierten Anteil, der erfolgreich getestet wurde. Die Summe aus beiden ist der für das Projekt insgesamt notwendige Aufwand. In der Darstellung ist das Flussdiagramm mit einem MS-Excel-Diagramm visualisiert (siehe Abb. 1 Projektfortschritt). Die Fließgeschwindigkeit eines Projektes ist ablesbar. Es ist damit auch ein Indikator dafür, wann das Projekt ins Stocken gerät. Dann bleibt die Anzahl der offenen Aufwandspunkte, über mehrere Arbeitstage hinweg, gleich. Außerdem zeigt die Metrik an, wenn es einen zu großen Feature-Creep im Projekt gibt. Das ist eine Situation, wenn Arbeitsaufwand zu einem laufenden Plan für einen Meilenstein hinzu gefügt wird. Dann erhöht sich die Gesamtsumme stark. Diese Metrik kann nur sinnvoll erzeugt werden, wenn in einem Projekt eine hohe Disziplin, im Pflegen der Arbeitseinheiten, besteht. Es müssen genaue Schätzungen von dem Aufwand zur Erstellung eines Features vorliegen und eine korrekte Buchhaltung über den getätigten Aufwand eines implementierten Features geführt werden.

Abbildung 1:
Projektfortschritt



Projektfortschritt

Codecoverage Eine beliebte Metrik ist der Codecoverage. Sie zeigt, welcher Anteil des Codes mit automatischen Tests abgedeckt ist. Für diese Statistik steht Software von verschiedenen Anbietern zur Verfügung. Testverfahren die hier eingesetzt werden sind vielfältig.

Kontrollflussorientierte Testmethoden betrachten die Struktur des Quellcodes. Unit Tests

führen einzelne Funktionstests aus. Fragestellungen, die für die Codeabdeckungsmetrik beantwortet werden, sind:

- Wurde jede Funktion des Programmes getestet?
- Wurde jede Zeile Code von den Tests erfasst?
- Wurden alle Verzweigungen der Bedingungen getestet?
- Wurde jede mögliche Route durch den Code berücksichtigt?
- Wurde jeder mögliche Aufruf und Ausgang einer Funktion für den Test ausgeführt?

Uta Kapp arbeitet als freiberufliche IT-Beraterin und systemischer Coach. Mit einer Kombination aus Fachberatung und Prozessberatung für Softwareprojekte hilft sie Entwicklungsteams bei der Bewältigung der ständig steigenden Komplexität. Hier kommen die agile Softwareentwicklungsmethode Scrum und Organisationsaufstellungen zum Einsatz. Der Einsatz von Collaboration-Plattformen ist ein weiterer Schwerpunkt.

Jean Pierre Berchez ist Geschäftsführer der [HLSC UG](#) und beschäftigt sich seit mehr als 15 Jahren mit den Themen Projektmanagement, Software Engineering und objektorientierte Softwareentwicklung. In den letzten Jahren liegt sein Interesse insbesondere auf den Themengebieten agile Entwicklung mit Schwerpunkt Scrum sowie "Collaborative" Software Development. Neben seiner Tätigkeit als Geschäftsführer ist Jean Pierre Berchez auch als Lehrbeauftragter an den BAs Stuttgart und Heidenheim sowie an der Hochschule Liechtenstein für die Themen "Anforderungsmanagement", "Scrum" und "Collaborative Software Development" tätig. Er organisiert unter anderem den Community-Event "[Scrum-Day](#)" in Deutschland.

Original Artikel auf:

<http://it-republik.de/dotnet/artikel/Metriken-Statistiken-Dashboards-3072.html>

andere Artikel dieser Serie

- [Kollaborieren mit Plattform](#)
- [Integrationen und Schnittstellen – "Wie es Euch gefällt"](#)
- [Bughunting](#)
- [Was macht ein Open Source-Projekt erfolgreich?](#)
- [Kommunikation zwischen Entwicklern, Benutzern und Management](#)
- [Wikinomics](#)
- [Kollabieren oder Kollaborieren](#)
- [Integration statt Brückenbau](#)
- ["Projekt Gold" oder: Der Weg führt zum Ziel](#)