

[Achtung: Verwenden Sie einen Sperrvermerk nur in sehr gut begründeten Fällen!]

[evtl. Sperrvermerk]

Auf Wunsch der Firma [FIRMA] ist die vorliegende Arbeit bis zum [DATUM] für die öffentliche Nutzung zu sperren.

Veröffentlichung, Vervielfältigung und Einsichtnahme sind ohne ausdrückliche Genehmigung der oben genannten Firma und der/dem Verfasser/in nicht gestattet. Der Titel der Arbeit sowie das Kurzreferat/Abstract dürfen jedoch veröffentlicht werden.

Dornbirn,

Unterschrift der Verfasserin/des Verfassers

Firmenstempel

Qualitätsmanagement in Komponententeams

Untertitel

Masterarbeit
zur Erlangung des akademischen Grades

Master of Science (MSc)

Fachhochschule Vorarlberg
Informatik

Betreut von
Prof. Dr. Michael Felderer

Vorgelegt von
Daniel Grießer
Dornbirn, Juli 2018

[evtl. Widmung]

[Text der Widmung]

Kurzreferat

[Deutscher Titel Ihrer Arbeit]

[Text des Kurzreferats]

Abstract

[English Title of your thesis]

[text of the abstract]

[evtl. Vorwort]

[Text des Vorworts]

Inhaltsverzeichnis

Abbildungsverzeichnis	10
Tabellenverzeichnis	11
[evtl. Abkürzungsverzeichnis]	12
1 Einleitung	13
2 Situationsanalyse	15
2.1 Software-Qualität	15
3 [Kapitel]	17
3.1 [Unterkapitel zweite Ebene]	17
3.2 [Unterkapitel zweite Ebene]	17
3.2.1 [Unterkapitel dritte Ebene]	18
3.2.1.1 [Unterkapitel vierte Ebene]	18
3.3 [Unterkapitel zweite Ebene]	18
4 [Kapitel]	20
4.1 [Unterkapitel zweite Ebene]	20
4.1.1 [Unterkapitel dritte Ebene]	20
4.1.1.1 [Unterkapitel vierte Ebene]	20
Literaturverzeichnis	21
[evtl. Anhang]	22
Eidesstattliche Erklärung	23

Abbildungsverzeichnis

1.1	Komponenten Teams in Scrum	14
2.1	Korrelationsmatrix Qualitätskriterien	16
3.1	Aufheizverhalten von PTFE	18

Tabellenverzeichnis

3.1 Stundenplan 18

[evtl. Abkürzungsverzeichnis]

ETW Energietechnik und Energiewirtschaft

SQL Structured Query Language

Bash Bourne-again shell

1 Einleitung

Mit Qualitäts-Analysetools, wie z.B. SonarQube¹, können ganze Softwaresysteme kontinuierlichen Qualitätstests unterzogen werden. Durch die ermittelten Kennzahlen können Aussagen zur Qualität des gesamten Systems, über einzelne Komponenten, bis hin zu einer einzelnen Quellcode-Datei getroffen werden.

Auch Scrum² wird als agiles Vorgehensmodell in der Softwareentwicklung immer beliebter. Dabei wird bei mehreren Teams, die auf vielen Systeme arbeiten, auf 2 Arten von Scrum Teams zurückgegriffen: Feature- oder Komponenten-Teams.

Feature-Teams bieten dabei den Vorteil, näher am Scrum Workflow zu sein, da alle Arbeiten an einem Feature von einem Team erledigt werden. Damit das möglich ist, braucht es aber ein breites Wissen über alle Plattformen innerhalb des Teams. Komponenten-Teams sind im Gegensatz dazu nur für bestimmte Komponenten in einzelnen Plattformen verantwortlich, dafür kann eine Feature-Umsetzung aber mehrere Sprints dauern und es muss bei der Planung auf Abhängigkeiten geachtet werden.

¹ *Continuous Code Quality / SonarQube*. URL: <https://www.sonarqube.org/> (besucht am 05.01.2018).

² *Scrum*. URL: <http://www.scrum.org> (besucht am 05.01.2018).

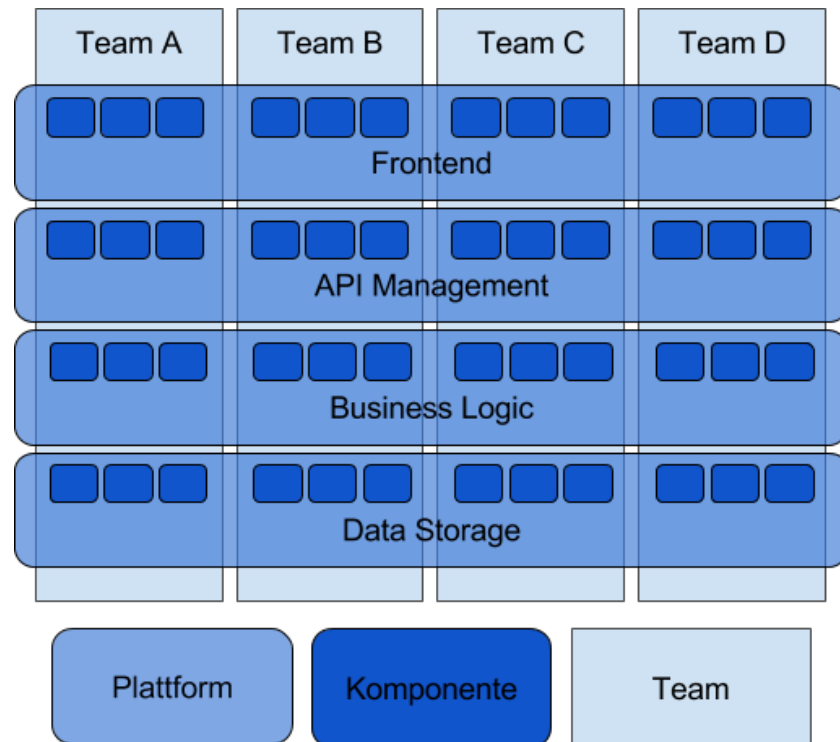


Abbildung 1.1: Komponenten Teams in Scrum

Diese Arbeit beschäftigt sich mit Komponenten-Teams und damit verbunden einem komponentenbasierten Qualitätsmanagement auf teamübergreifenden Plattformen. Das bedeutet, dass Kennzahlen zu Qualitätsmerkmalen nicht auf System-, sondern auf Komponentenebene gesammelt und aggregiert werden, um für jedes Team eine individuelle Sicht auf das Qualitätsmanagement zu ermöglichen. Um das zu ermöglichen, wird erst eine Vorgehensweise zur Ermittlung von relevanten Kennzahlen entwickelt und diese an einer Beispiel-Organisation angewendet. Zur Sammlung, Auswertung und Darstellung dieser Kennzahlen wird eine Software entwickelt, die in eine bestehende Umgebung integriert werden kann.

2 Situationsanalyse

2.1 Software-Qualität³

Eine mögliche Definition von Software-Qualität findet sich in der DIN-ISO-Norm 9126:

Software-Qualität ist die Gesamtheit der Merkmale und Merkmalswerte eines Software-Produkts, die sich auf dessen Eignung beziehen, festgelegte Erfordernisse zu erfüllen.

Wie aus dieser Definition schon erkennbar ist, gibt es viele unterschiedliche Kriterien, um die Qualität von Software zu bewerten. Einige wesentliche Merkmale, um die Qualität von Software bewerten zu können, lassen sich in kunden- und herstellerorientierte Merkmale unterteilen:

Kundenorientierte Merkmale

Nach außen hin sichtbare Merkmale, die sich auf den kurzfristigen Erfolg der Software auswirken, da sie die Kaufentscheidung möglicher Kunden beeinflussen.

Funktionalität (Functionality, Capability)

Beschreibt die Umsetzung der funktionalen Anforderungen. Fehler sind hier häufig Implementierungsfehler (sogenannte Bugs), welche durch Qualitätssicherung bereits in der Entwicklung entdeckt oder vermieden werden können.

Laufzeit (Performance)

Beschreibt die Umsetzung der Laufzeitanforderungen. Besonderes Augenmerk muss in Echtzeitsystemen auf dieses Merkmal gelegt werden.

Zuverlässigkeit (Reliability)

Eine hohe Zuverlässigkeit ist in kritischen Bereichen, wie z.B. Medizintechnik oder Luftfahrt, unabdingbar. Erreicht werden kann diese aber nur durch die Optimierung einer Reihe anderer Kriterien.

Benutzbarkeit (Usability)

Betrifft alle Eigenschaften eines Systems, die mit der Benutzer-Interaktion in Berührung kommen.

Herstellerorientierte Merkmale

Sind die inneren Merkmale, die sich auf den langfristigen Erfolg der Software auswirken und somit als Investition in die Zukunft gesehen werden sollten.

³vgl. Dirk W. Hoffmann. *Software-Qualität*. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN: 978-3-642-35699-5 978-3-642-35700-8, Kapitel 1.2.

Wartbarkeit (Maintainability)

Die Fähigkeit auch nach der Inbetriebnahme noch Änderungen an der Software vorzunehmen. Wird oft vernachlässigt, ist aber essentiell für langlebige Software und ein großer Vorteil gegenüber der Konkurrenz.

Transparenz (Transparency)

Beschreibt, wie die nach außen hin sichtbare Funktionalität intern umgesetzt wurde. Gerade bei alternder Software, kann es zu einer Unordnung kommen, welche auch Software-Entropie (Grad der Unordnung) genannt wird.

Übertragbarkeit

Wird auch Portierbarkeit genannt und beschreibt die Eigenschaft einer Software, in andere Umgebungen übertragen werden zu können (z.B. 32-Bit zu 64-Bit oder Desktop zu Mobile).

Testbarkeit (Testability)

Testen stellt eine große Herausforderung dar, da oft auf interne Zustände zugegriffen werden muss oder die Komplexität die möglichen Eingangskombinationen vervielfacht. Aber gerade durch Tests können Fehler frühzeitig entdeckt und behoben werden.

Je nach Anwendungsgebiet und den Anforderungen der Software haben die Merkmale unterschiedliche Relevanz und einige können sich auch gegenseitig beeinflussen, wie aus der Korrelationsmatrix ersichtlich.

	Laufzeit	Zuverlässigkeit	Benutzbarkeit	Transparenz	Übertragbarkeit	Wartbarkeit	Testbarkeit
Funktionale Korrektheit	-	+		+	+	+	+
Laufzeit		-		-	-	-	-
Zuverlässigkeit			+				+
Benutzbarkeit							
Transparenz				+	+	+	
Übertragbarkeit							
Wartbarkeit							

Abbildung 2.1: Korrelationsmatrix Qualitätskriterien⁴

⁴Dirk W. Hoffmann. *Software-Qualität*. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN: 978-3-642-35699-5 978-3-642-35700-8, S. 11, Abb. 1.3.

3 [Kapitel]

Formatvorlage für den Fließtext. Formatvorlage für den Fließtext. Formatvorlage für den Fließtext. Formatvorlage für den Fließtext. Formatvorlage für den Fließtext. Formatvorlage für den Fließtext. Formatvorlage für den Fließtext. Formatvorlage für den Fließtext. Formatvorlage für den Fließtext.

Formatvorlage für ein längeres direktes Zitat. Formatvorlage für ein längeres direktes Zitat. Formatvorlage für ein längeres direktes Zitat. Formatvorlage für ein längeres direktes Zitat. Formatvorlage für ein längeres direktes Zitat. . . .

Formatvorlage für den Fließtext. Hier eine Liste.

1. Verstehen
2. Üben
3. Können

3.1 [Unterkapitel zweite Ebene]

Formatvorlage für den Fließtext. Die Abbildung 3.1 auf Seite 18 zeigt drei Entladungskurven eines biphasischen Defibrillators.

3.2 [Unterkapitel zweite Ebene]

Formatvorlage für den Fließtext. Jetzt eine Fußnote⁵ Die quadratischen Gleichung (3.1) hat wieviele Nullstellen?

$$x^2 - 2x + 5 = 0. \tag{3.1}$$

Zwei von Einsteins berühmtesten Formeln lauten:

$$\begin{aligned} E &= mc^2 \\ m &= \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \end{aligned}$$

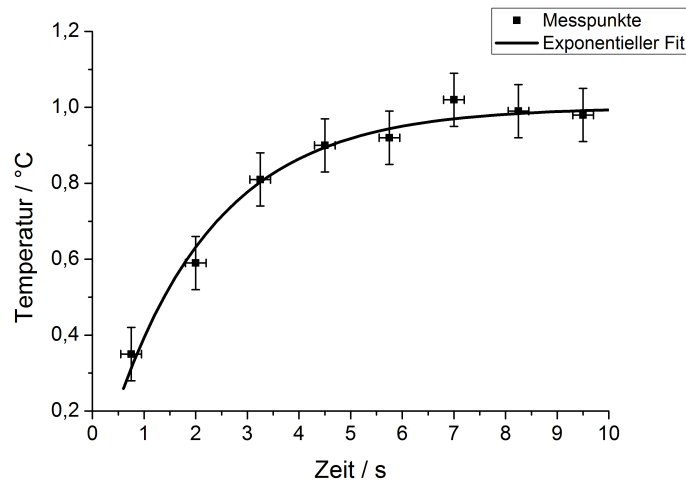


Abbildung 3.1: Das Bild zeigt das Aufheizverhalten von PTFE.
Quelle: eigene Ausarbeitung

Datum	Thema	Raum
Montag	Graphentheorie	U1
Donnerstag	Algebra	MZB23

Tabelle 3.1: Stundenplan des Jahres 2030.
Quelle: eigene Ausarbeitung

3.2.1 [Unterkapitel dritte Ebene]

Formatvorlage für den Fließtext. Hier die einfache Tabelle 3.1

3.2.1.1 [Unterkapitel vierte Ebene]

Formatvorlage für den Fließtext.

3.3 [Unterkapitel zweite Ebene]

Verweise: zu einem Buch mit Details `bathe_finite-elemente-methoden_1990` oder ohne Details `bathe_finite-elemente-methoden_1990` ein Buchteil `areger_problem-based` eine Dissertation `sporn_interaktives_2000` ein Dokument `industriellenvereinigung_best` ein Enzyklopädieartikel `brockhaus_kreativitat_1872` ein Film `de_wilde_through_2008` ein Konferenz-Paper `weber_podcasts._2006` ein Magazin-Artikel `autornachname1_magazi` ein Pordcast `paulus_horen_????` eine Tonaufnahme `horowitz_horowitz_2003` eine Videoaufnahme `fhvlearningsupport_was_2008` ein Vortrag `kohls_literaturverwaltung`

⁵Dies ist eine Fußnote.

eine Website **wedekind_von_2008** ein Zeitschriftenartikel **hofer_wir_2008** und ein Zeitungsartikel **schenkel_tsunami_2012**

4 [Kapitel]

4.1 [Unterkapitel zweite Ebene]

Formatvorlage für den Fließtext.

4.1.1 [Unterkapitel dritte Ebene]

Formatvorlage für den Fließtext.

4.1.1.1 [Unterkapitel vierte Ebene]

Formatvorlage für den Fließtext.

Literatur

Continuous Code Quality / SonarQube. URL: <https://www.sonarqube.org/> (besucht am 05.01.2018).

Hoffmann, Dirk W. *Software-Qualität*. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN: 978-3-642-35699-5 978-3-642-35700-8.

Scrum. URL: <http://www.scrum.org> (besucht am 05.01.2018).

[evtl. Anhang]

Formatvorlage für den Fließtext.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder in gleicher noch in ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Dornbirn, am [Tag. Monat Jahr anführen]

[Vor- und Nachname Verfasser/in]