

Software und Qualitäts- management

TL3

Gruppe SQM-e-learning:

Ralf Gemmel

Michael Ruhnau

Stefan Bregenzer

Kim Michael Jansen

Matthias Köhler

Stefan Kleih

Anbieter:

Dr. PD habil. Christina Klüver (ehem. Stoica)

Tutor:

Prof. Dr. Jürgen Klüver

Abgabedatum:

27.01.2013

Inhaltsverzeichnis

1.	Fortschrittsüberwachung	3
1.1	Fortschrittsüberwachung auf der Aktivitätenebene	3
1.2	Fortschrittsüberwachung auf der Projektebene	4
2.	Konfigurations- und Änderungsmanagement	4
2.1	Versionsmanagement der Dokumente	5
2.2	Versionsmanagement beim Programmcode	6
2.2.1	Versionierung kleiner Änderungen	6
2.2.2	Versionierung größerer Änderungen	6
3.	Berichtserstattung.....	6
4.	Planung der Qualitätssicherung.....	8
4.1	Fachliche Qualitätssicherung	8
4.2	Technische Qualitätssicherung	9
4.2.1	Komponententests.....	10
4.2.2	Integrationstests	10
4.2.3	Systemtests	10
4.2.4	Abnahmetest.....	11
4.2.5	Regressionstests	11
4.2.6	Testcases	11
5.	Gruppenfazit	13
6.	Anhang	16
6.1	Testfall T_0005: Neue Karteikarten können erfasst werden	16
6.1.1	Ausführung	16
6.1.2	Ablauf des Tests	16
6.1.3	Erwartetes Ergebnis.....	16
6.1.4	Eingetretenes Ergebnis.....	17
6.2	Testfall T_0006: Workflow zur Freigabe von Karteikarten.....	18
6.2.1	Ausführung	18
6.2.2	Ablauf des Tests	18
6.2.3	Erwartetes Ergebnis.....	18
6.2.4	Eingetretenes Ergebnis.....	19

Abbildungsverzeichnis

Abbildung 1 - Reportingkanäle im Projekt SQM-e-learning.....	7
Abbildung 2 - V-Modell.....	9
Abbildung 3 - Auszug definierter Testcases	12
Abbildung 4 - Auszug des aktuellen Issue Log.....	13

1. Fortschrittsüberwachung

Allgemein besteht die Fortschrittsüberwachung einer Aktivität, eines Teilprojekts oder des gesamten Projekts darin, den jeweiligen Iststand mit einem geplanten Sollstand zu vergleichen. Der Sollstand wird der Projektplanung entnommen. Für die Ermittlung des Iststandes stehen verschiedene Analysemethoden zur Verfügung, die im Folgenden näher erläutert werden.

1.1 Fortschrittsüberwachung auf der Aktivitätenebene

Zur Überwachung von Aktivitäten wird der tatsächlich investierte Aufwand (z.B. Arbeitstage) für alle Aktivitäten vom jeweiligen Teammitglied festgehalten. Dies muss sowohl für abgeschlossene, als auch für noch nicht begonnene und teilweise fertig gestellte Aktivitäten durchgeführt werden. Hieraus kann nun der Fertigstellungsgrad einer Aktivität, oder des (Teil-)Projekts, ermittelt werden. Er gibt an, zu wie viel Prozent die Aktivität fertig gestellt ist. Zur Ermittlung des Fertigstellungsgrads wird der erbrachte Aufwand durch den geplanten Aufwand geteilt.

Fertigstellungsgrad = erbrachter Aufwand / geplanter Aufwand

Es fällt auf, dass hiernach eine Aktivität dann zu 100% fertig gestellt ist, wenn genau so viel Aufwand erbracht wurde, wie (ursprünglich) geplant wurde. Die Planung wird kontinuierlich angepasst, sodass gegen Ende einer Aktivität der "geplante Aufwand" dem realen Aufwand entsprechen wird.

Aufgrund der vielen kleinen Aktivitäten in unserem Projekt, erschien uns dieses Vorgehen jedoch nicht sinnvoll zu sein. Die Aktivitäten können in wenigen Stunden abgeschlossen werden, sodass sie bei Abfragen entweder noch nicht begonnen, oder fertig gestellt waren. Um die Planung weiterentwickeln zu können, wäre es interessant zu vergleichen, ob der tatsächlich benötigte Aufwand richtig geschätzt wurde, also dem geplanten Aufwand entspricht. Auf Grundlage dieser Erfahrungswerte könnten künftige Schätzungen mit einem Aufschlag/ Abschlag versehen werden und die Teammitglieder lernen generell ihren Aufwand besser einzuschätzen. Für unsere kurze Projektdauer würde der Erfassungsaufwand jedoch nicht in einem angemessenen Verhältnis zur besseren Planung stehen. Daher haben wir hierauf verzichtet. Stattdessen wurden regelmäßig Termine für die kommende Woche, z.B. für die Fertigstellung des Pflichtenhefts, vereinbart. In der Folgewoche konnten die Aktivitäten abgefragt werden, um schnell einen Überblick zu bekommen, welche außerhalb des Zeitplans liegen.

Im Verhältnis zur Auftraggeberin (Frau PD. Dr. Klüver) wurde außerdem ein Ampelbericht erstellt (Mail vom 12.01.2013 - 11:32). Darin wurden der Funktionsumfang, die Kostenentwicklung und die Risikoplanung positiv (also grün) eingeschätzt (vergleiche "Berichterstattung").

1.2 Fortschrittsüberwachung auf der Projektebene

Auf der Projektebene sollen Informationen (z.B. Ampelberichte, Statusberichte, Risikoberichte) von den unteren Ebenen jeweils zur nächst höheren Projektebene geleitet werden, sodass der Projektleiter über den Gesamtfortschritt des Projekts informiert ist. Aufgrund der flachen Hierarchie (zwei Ebenen) in unserem Projekt, erhält die Projektleitung diese Berichte unmittelbar von den Teammitgliedern. Zum Beispiel erfolgt dies per Email oder mündlich in der wöchentlichen Telefonkonferenz (siehe hierzu auch Kap. 3 Berichtserstattung).

Aufgrund der kurzen Projektdauer habe wir uns dazu entschieden keine aufwändige Analysemethode wie die "Meilensteinanalyse", oder die "Earned Value Analysis" anzuwenden. Durch die Abgabetermine der Teilleistungen konnte circa alle vier Wochen ein Meilenstein definiert werden. Mit den hierzu zu erledigenden Aufgaben im Blick und den Planungsdokumenten aus Teilleistung 2 (Projektstrukturplan, Aktivitätenplan, Größen- und Aufwandsschätzung) und der wöchentlichen Telefonkonferenzen konnte ein ausreichender Überblick über den Projektfortschritt gewonnen werden.

2. Konfigurations- und Änderungsmanagement

Für das Projekt wurde im Rahmen der Rollendefinition das Thema Änderungsmanagement besprochen. Aufgrund der kurzen Projektdauer ist es wichtig, dass der Softwareentwicklungsprozess von der Anforderungserhebung im Fachkonzept über die technische Konzeption bis zur Implementierung nur einmal erfolgen muss. Iterationen sollen nur im Notfall erfolgen, daher existiert das Änderungsmanagement bzgl. der Software nur theoretisch in unserem Projekt. Softwareänderungen würden dann in den regelmäßigen Jour Fixen besprochen und im dazugehörigen Protokoll dokumentiert.

Das Releasemanagement ist auch nicht ausgeprägt, da es nur ein finales Release geben wird, welches an die Auftraggeberin ausgeliefert wird.

Die Softwareentwicklung erfolgt über ein Web-CMS, welches mit zusätzlichen JavaScript-Funktionalitäten angereichert wird. Die entsprechenden Dateien liegen im Filesystem des Webservers. Ein automatisiertes Buildmanagement findet nicht statt.

Beim Versionsmanagement ist zwischen dem Versionsmanagement der Dokumente und des Programmcodes zu unterscheiden. Zentrales Repository für das Versionsmanagement im Projekt bzgl. Dokumenten und Programmcode ist Google-Drive, ein Cloud-Storage-System der Firma Google auf das alle Projektmitglieder per Webbrowser zugreifen können.

2.1 Versionsmanagement der Dokumente

Beim Versionsmanagement von Dokumenten wird ein initiales Dokument in GoogleDocs erstellt und im Dokumenten-Repository aufgenommen. Das Dokumenten-Repository ist ein Dokument, was durch die Qualitätsmanager gepflegt wird und in dem alle Projektdokumente zentral erfasst werden. Im erstellten initialen Dokument arbeiten dann eine oder mehrere Personen. Im Falle, dass eine Person am Dokument arbeitet, ist es organisatorisch so im Projekt geregelt, dass alle anderen nur Anmerkungen in Form von Kommentaren hinterlassen dürfen, damit Arbeiten des Erstellers nicht aus Versehen gelöscht werden. Wenn mehrere Personen ein Dokument erstellen, wird dieses in Abschnitte aufgeteilt, für die Verantwortliche im Jour Fixe festgelegt werden. Alle Änderungen am Dokument werden im Hintergrund von GoogleDocs zusätzlich protokolliert. Bei der Verwendung von GoogleDocs ist zu jeder Zeit klar, an welchem Dokument gearbeitet wird, da hier verschiedene Personen gleichzeitig arbeiten können und alle anderen die Veränderungen in Echtzeit sehen. Die Projektmitglieder legen fest, an welchem Termin das Dokument aus GoogleDocs exportiert und als Microsoft Word-Dokument weiterbearbeitet wird, um das finale Layouten und letzte Abstimmungen durchzuführen. In diesem Stadium, wird das Microsoft Word-Dokument per E-Mail verschickt und im Dokumentname versioniert. Da im Rahmen der Bearbeitung die neue aktuelle Version immer an alle Teilnehmer geschickt wird, haben alle immer die gleiche Version. Im Falle einer Änderung wird die Version hochgezählt inkrementiert und wiederum an alle verteilt. So wird das Problem von Merging vermieden. Die finale Version wird dann zurück in Google-Drive importiert und im Falle einer Studienleistung per Email an die betreuende Dozentin geschickt.

2.2 Versionsmanagement beim Programmcode

Das Versionsmanagement des Programmcodes läuft wie folgt ab. Die aktuellste Version der Anwendung befindet sich in Form von PHP-Modulen und anderen Dateien auf dem über das Internet verfügbaren Webserver.

2.2.1 Versionierung kleiner Änderungen

Alle konfigurativen Änderungen in der Webanwendung werden fast vollständig in der dem WebCMS zugrunde liegenden MySQL-Datenbank auf dem Webserver gespeichert. Daher werden für kleine Änderungen regelmäßige Datenbanksicherungen erstellt und im Filesystem des Webserver abgespeichert. Da der Programmcode nicht verändert wird, muss hier das Dateisystem nicht zu häufig gesichert werden. Die Datenbanksicherungen werden regelmäßig über den FTP-Client Filezilla auf dem lokalen PC des Backend-Entwicklers gesichert. Sie haben im Dateinamen einen Zeitstempel, der das Datum und die Uhrzeit der Erzeugung enthält.

2.2.2 Versionierung größerer Änderungen

Um im Falle von unerwarteten Problemen jederzeit auf eine lauffähige Version der Webanwendung zurückgreifen zu können, werden regelmäßig bei größeren Änderungen am Programmcode (z.B. Installation eines neuen PHP-Moduls, Hinzufügung eines PHP-Templates) im Vorfeld lauffähige Komplettsicherungen erzeugt. Diese bestehen aus einer aktuellen Datenbanksicherung und dem gesamten Programmcode. Eine komplette Sicherung wird mit dem FTP-Client auf den PC des Backend-Entwicklers heruntergeladen und daraus ein ZIP-Paket erstellt, welches im Dateinamen das Datum der Erzeugung der Komplettsicherung enthält. Die Zip-Datei wird anschließend in das zentrale Repository Google-Drive hochgeladen und allen Projektmitgliedern zur Verfügung gestellt.

Auf ein Versionsmanagement mit einem Versionsverwaltungssystem wie Subversion wurde verzichtet, da die Programmcodeerstellung im Projekt nur vereinzelt und überschaubar erfolgt. Der meiste Entwicklungsaufwand im Backend besteht in der Konfiguration des WebCMS.

3. Berichtserstattung

Das Projektreporting dient unter anderem der Information der Projektleitung, der Auftraggeberin und dem Lenkungsausschuss und soll einen Überblick über den aktuellen Stand des Projektes vermitteln. Für das Projekt SQM-e-learning sind folgende Informationskanäle berücksichtigt worden:

- Berichterstattung an die Auftraggeberin
- Berichterstattung von Projektgruppe an Projektleitung und umgekehrt
- Berichterstattung bzw. Austausch innerhalb der Sub-Teams „Qualitätsmanagement“ und „IT-Entwicklung“

Eine Übersicht über die Reportingkanäle im Projekt SQM-e-learning bietet folgende Darstellung:

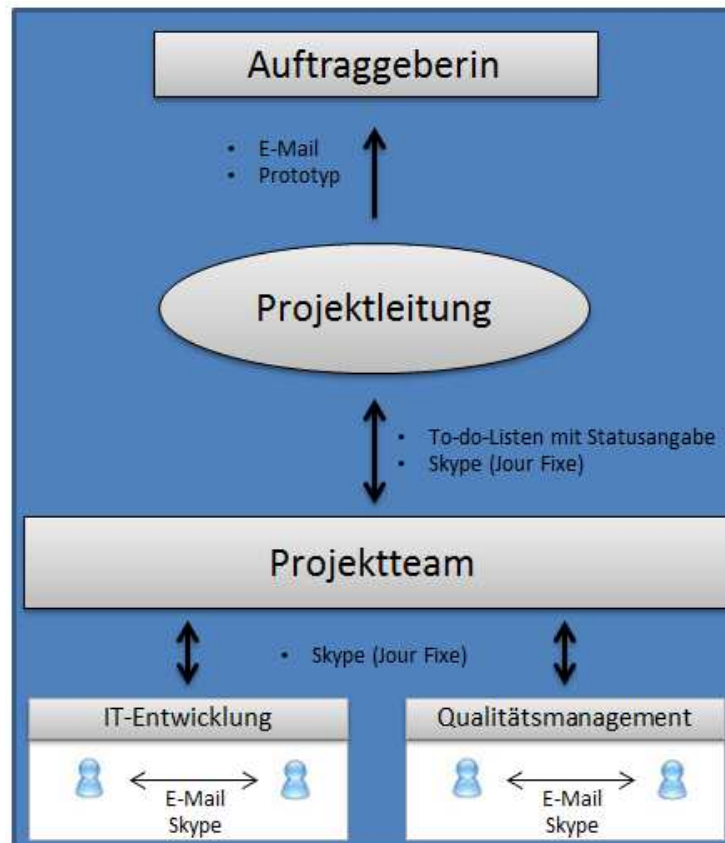


Abbildung 1 - Reportingkanäle im Projekt SQM-e-learning

Quelle: Eigene Darstellung

Wählt man eine Betrachtungsweise von unten nach oben, so findet innerhalb der Subteams „IT-Entwicklung“ und „Qualitätsmanagement“ ein regelmäßiger Austausch in Form von Statusmeldungen statt. Diese erfolgen entweder per Skype oder schriftlich per E-Mail. Ein kontinuierlicher Austausch innerhalb der Untergruppen ist erforderlich, um Doppelarbeiten zu vermeiden. Die Statusmeldung an das Projektteam erfolgt im Rahmen der wöchentlichen Jour-Fixe per Skype.

Das Jour-Fixe wird genutzt, um offene Punkte zu diskutieren und das weitere Vorgehen abzustimmen. Hierdurch bekommt die Projektleitung einen guten Überblick über den aktuellen Projektstatus. Darüber hinaus wird im Jour-Fixe-Protokoll eine To-do-Liste gepflegt, die eine Spalte „Fortschritt“ enthält. Diese Spalte wird laufend durch die

verantwortlichen Projektmitglieder aktualisiert, sodass die Projektleitung auch über dieses Instrument jederzeit in der Lage ist, sich einen guten Überblick über den Stand der Aufgabenerledigung zu verschaffen.

Die Berichterstattung an die Auftraggeberin erfolgt durch die Projektleitung per E-Mail. Auf ein standardisiertes Formular mit Ampelfunktion wurde aufgrund der Projektgröße und der kurzen Projektlaufzeit verzichtet. Das Reporting war zeitlich nicht fixiert, sondern vielmehr an der Erledigung von Arbeitspaketen (z.B. Fertigstellung des Prototyps) geknüpft.

Darüber hinaus hatte die Auftraggeberin bereits zu einem frühen Projektstand die Möglichkeit das „reale Produkt“ (Prototyp) zu testen und so einen Eindruck über die Funktionalitäten und der Benutzeroberfläche zu gewinnen.

4. Planung der Qualitätssicherung

Die einzelnen Schritte, die im Rahmen der Qualitätssicherung zu erfolgen haben, wurden frühzeitig im Projektverlauf geplant und im Fachkonzept durch die Qualitätsmanager dokumentiert. Die Aktivitäten wurden daraufhin abgeleitet und in den Projektstrukturplan bzw. Aktivitätenplan übertragen. Wie bei der Rollenverteilung zu Beginn des Projektes festgelegt, erfolgt die Qualitätssicherung durch Stefan Kleih und Michael Ruhnau.

Der im Anhang beigefügte Softwarequalitätssicherungsplan soll die Rahmenbedingungen der Softwarequalitätssicherung abgrenzen.

Grundsätzlich ist die Qualitätssicherung in eine fachliche und technische Qualitätssicherung zu unterscheiden.

4.1 Fachliche Qualitätssicherung

Die fachlichen Qualitätssicherungsmaßnahmen beziehen sich ausschließlich auf die elektronisch erstellten Karteikarten. So wurde bereits im Fachkonzept festgelegt, dass nach dem Erstellen der Karteikarte das Qualitätsmanagement für die Prüfung der Richtigkeit der Karteikarte verantwortlich ist. Hierfür ist es wichtig die Frage sowie die Antwort mit dem in diesem Kurs zur Verfügung gestellten Skript zu vergleichen. Bei Erstellung der Karteikarte durch einen Qualitätsmanager, ist diese natürlich von einem anderen Qualitätsmanager zu kontrollieren und freizugeben (4-Augen-Prinzip).

Ob alle Themenbereiche der jeweils zugeteilten Kapitel bearbeitet wurden, obliegt dem Verantwortlichen des Kapitels.

Die fachliche Qualitätssicherung der abzugebenden Teilleistungen innerhalb dieses Projektes erfolgt auch erstmalig durch die Qualitätsmanager in Form von Schreibtischtests (Dokument wird überprüft -> kritisch begutachtet -> kommentiert -> zurückgegeben) und wird folgend von allen Beteiligten durchgeführt.

Während der wöchentlichen Jour Fixe stellen die jeweiligen Autoren der Teilbereiche ihr Dokument den anderen Beteiligten zur Überprüfung zur Verfügung, welche daraufhin Teile dieses Abschnitts qualitätssichern (Walkthrough-Review-Methode). Die Ergebnisse dieses Testverfahrens werden in den Jour-Fixe-Dokumenten protokolliert.

4.2 Technische Qualitätssicherung

Im Projekt wird nach dem V-Modell vorgegangen.

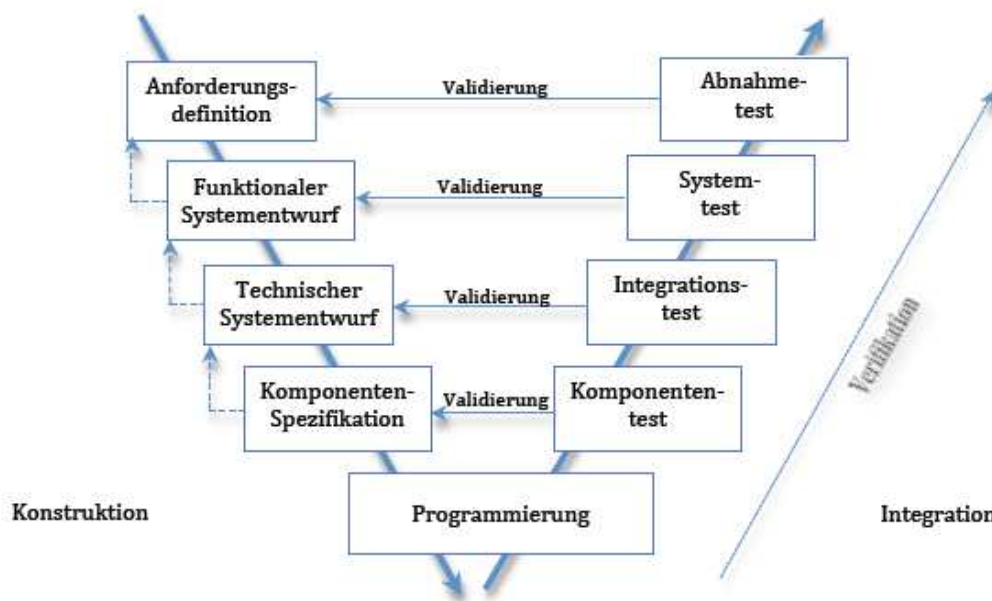


Abbildung 2 - V-Modell

Quelle: Klüver, Christina (2012), S. 79: Modul 2: Softwaremanagement, Skript zum VAWI-Kurs SQM im WS 2012/2013 an der Universität Duisburg-Essen.

Für die Konzeption und Entwicklung der Anwendung liegen folgende Dokumente zur Spezifikation vor:

- Anforderungsdokumentation
- Gesamtsystemspezifikation (Pflichtenheft)

Im Rahmen der technischen und fachlichen Qualitätssicherung werden die einzelnen Schritte des V-Modells (Konstruktionsseite) durch Tests verifiziert. Diese werden im

Folgenden erläutert. Bei den einzelnen Tests wird beispielhaft verdeutlicht, wie diese im Projekt real angewendet wurden.

4.2.1 Komponententests

Im Rahmen der Komponententests werden die einzelnen Module bzw. Komponenten der Software getestet. Es werden einzelne Komponenten isoliert voneinander getestet, aus diesem Grund wird auch oft von Modultests gesprochen. Diese Tests werden in unserem Team von den Entwicklern selbst durchgeführt. Erst nach einem erfolgreichen Komponententest wird das Modul zur Qualitätssicherung gegeben.

Bei dieser Teststufe werden überwiegend Whitebox-Tests angewendet. Bei diesen Tests werden bestimmte Anweisungen und Bedingungen erzeugt, um den Quellcode unter verschiedenen Bedingungen zu testen.

Bei der Entwicklung der Funktionalität zum Umblättern einer Karteikarte wurde ein solcher Komponententest durch den Entwickler durchgeführt. Erst nach dem erfolgreichen Komponententest wurde die Funktionalität zum Testen durch die QS freigegeben.

4.2.2 Integrationstests

Bei den Integrationstests wird das korrekte Zusammenwirken der unterschiedlichen Module der Anwendung getestet. Somit soll gewährleistet werden, dass die in der Software realisierten Abläufe (z. B. Freigabeprozess, Erfassen von Karteikarten, ...) fehlerfrei möglich ist. Bei dieser Teststufe wird noch nicht das gesamte System, sondern die Wechselwirkung und das Zusammenspiel von Komponenten untereinander beleuchtet. Es stehen vor Allem die funktionalen Anforderungen im Vordergrund.

Im Rahmen dieser Integrationstests wurde das Zusammenspiel einzelner PHP-Module der Anwendung getestet. So kommen im Erfassungs- und Freigabeprozess der Karteikarten unterschiedliche Module zum Einsatz (z. B. Modul zum Versenden von Mails, Workflow-Modul, Formular-Modul). Diese Tests wurden vom Entwicklerteam durchgeführt.

4.2.3 Systemtests

Bei den Systemtests wird das vollständige System getestet. Normalerweise würde man hierzu ein Testsystem verwenden. In unserem konkreten Falle arbeiten wir mit einem einzigen produktiven System, gegen welches die Tests durchgeführt werden. Der Fokus

liegt hierbei auf dem Gesamtsystem mit seinen funktionalen und nicht funktionalen Anforderungen.

Die Systemtests wurden im Projekt umgesetzt, da auf dem System frühzeitig gearbeitet wurde. Das System stand somit während der gesamten Projektphase auf dem Prüfstand. Fehler wurden dabei aufgenommen, behoben und wieder geprüft. Somit wurde das System vom Projektteam frühzeitig unter realen Bedingungen angewendet bzw. getestet.

Darüber hinaus wurden von der technischen Qualitätssicherung Testcases definiert, welche dann auch real durchgeführt wurden, um die fehlerfreie Umsetzung der wesentlichen funktionalen Anforderungen zu gewährleisten. Diese Tests wurden nicht von den Entwicklern selbst, sondern von den Qualitätsmanager durchgeführt, um sicherzustellen, dass dies durch jemanden erfolgt, der das System nicht selbst umgesetzt hat. Eine exemplarische Darstellung hierzu befindet sich weiter unten im Dokument.

4.2.4 Abnahmetest

Die Abnahmetests werden gemeinsam im Projektteam als Dienstleistung für die Auftraggeberin erbracht. Jedes Teammitglied hat in einem definierten Zeitraum (15.01. - 23.01.2013) die Gelegenheit, das System zu testen. Nach erfolgreichem Abschluss dieser Testphase gilt das System als fertig gestellt und abgenommen. Ab diesem Zeitpunkt werden keine Änderungen mehr vorgenommen. In dieser Phase der Abnahmetests kommen Blackbox-Tests zur Anwendung, da es hier letztlich um die Prüfung des Systems nach Außen hin zum Anwender geht. Die Funktionalitäten werden aufgerufen und es wird geprüft, ob das gewünschte Ergebnis eintritt.

4.2.5 Regressionstests

Sollte ein Test aufgrund eines Softwarefehlers nicht erfolgreich verlaufen, wird dies an den Entwickler kommuniziert. Der nachgebesserte Code bzw. die von der Änderung betroffenen Bereiche werden durch sogenannte Regressionstests erneut getestet. Somit wurde sichergestellt, dass nach einer Änderung die Anwendung bzw. die geänderten und abhängigen Bereiche fehlerfrei funktionieren.

4.2.6 Testcases

Die folgende Grafik stellt einen exemplarischen Auszug der funktionalen Testcases dar. Diese wurden durch das QS Team definiert und entsprechend durchgeführt. Bei der Erstellung der Testcases wurde darauf geachtet, dass die wesentlichen funktionalen Anforderungen gemäß Anforderungs-Dokumentation umgesetzt wurden. Die folgende Grafik stellt einen exemplarischen Auszug der definierten Testcases dar.

ID	Bereich	Anforderungs-Kennzeichen	Titel
T_0001	Registrierung und Login	e-learning.FA_0001(.1 - .3)	Neuer User registrieren
T_0002	Registrierung und Login	e-learning.FA_0002	Administrator kann Benutzern Rollen zuweisen
T_0003	Registrierung und Login	e-learning.FA_0003	Zuweisung unterschiedlicher Rollen gemäss Rollenkonzept
T_0004	Registrierung und Login	e-learning.FA_0004	Nicht angemeldete User haben Rolle "Anonymer User"
T_0005	Erstellung von Karteikarten	e-learning.FA_0005	Neue Karteikarten können erfasst werden
T_0006	Erstellung von Karteikarten	e-learning.FA_0006	Workflow zur Freigabe von Karteikarten
T_0006.1	Erstellung von Karteikarten	e-learning.FA_0006.1	Übersicht "In Arbeit" zur Anzeige eigener Karteikarten
T_0006.2	Erstellung von Karteikarten	e-learning.FA_0006.2	Übersicht "Zur QS vorgelegt" zur Anzeige eigener Karteikarten
T_0006.3	Erstellung von Karteikarten	e-learning.FA_0006.3	Wechsel des Status für mehrere gewählte Karteikarten
T_0006.4	Erstellung von Karteikarten	e-learning.FA_0006.4	Übersicht "zur QS vorgelegt" für QS Prüfung zeigt alle Karteikarten in diesem Status an
T_0006.5	Erstellung von Karteikarten	e-learning.FA_0006.5	Übersicht "Fertig gestellt" für QS Prüfung zeigt alle Karteikarten in diesem Status an
T_0006.6	Erstellung von Karteikarten	e-learning.FA_0006.6	Freigabe erteilen für mehrere gewählte Karteikarten
T_0006.7	Erstellung von Karteikarten	e-learning.FA_0006.7	Mailbenachrichtigung an QS für neue Karteikarte im Status "zur QS vorgelegt"
T_0007	Unterstützung des Lernprozesses	e-learning.FA_0007	Generierung von Trainings für den User
T_0007.1	Unterstützung des Lernprozesses	e-learning.FA_0007.1	Auswahl von Lernabschnitt und gewusst/nicht gewusst bei Generierung von Trainings für den User
T_0007.2	Unterstützung des Lernprozesses	e-learning.FA_0007.2	Generierung der Trainings
T_0007.3	Unterstützung des Lernprozesses	e-learning.FA_0007.3	System zeigt nur Rückseite an

Abbildung 3 - Auszug definierter Testcases

Quelle: Eigene Darstellung

Testfall Ausführung

In der Anlage befinden sich zwei Protokolle zur Ausführung der Testfälle:

- Testfall T_0005: Neue Karteikarten können erfasst werden
- Testfall T_0006: Workflow zur Freigabe von Karteikarten

Diese zeigen exemplarisch auf, mit welchem Vorgehen das System bei den Systemtests getestet wurde. Diese beiden Testfälle decken das Erstellen einer Karteikarte bis hin zur Freigabe ab. Somit ist gewährleistet, dass einer der wesentlichsten Bereiche der Anwendung ordnungsgemäß funktioniert.

Issue Log

Erkannte Fehler werden in einem Issue Log erfasst und zeitgleich an die zuständigen Personen kommuniziert. Dieses Issue Log wird von allen Mitgliedern gepflegt und von der QS gepflegt. Zu jedem Issue werden folgende Informationen erfasst:

- laufende Nummer
- Datum
- Erfasst von
- Titel
- Beschreibung
- Status
- Anmerkung

Im wöchentlichen Team Meeting wird die Issue Liste durchgesprochen. Somit ist das Team stets über die angefallenen Probleme informiert und es kann auch gut nachvollzogen werden, welche Issues noch zu bearbeiten sind.

Nachfolgend ist ein exemplarischer Auszug aus dem aktuellen Issue Log dargestellt.

ID	Datum	Erfasst von	Titel	Beschreibung	Status	Anmerkungen
1	06.01.2013	Michael Ruhnau	Registrierung	E-Mail-Benachrichtigung zur Aktivierung des Accounts wurde nicht an mich gesendet. Der Account kann nicht genutzt werden.	behoben	Mail wurde nicht versendet, da schon zu viele Nachrichten rausgingen (Begrenzung Aufgrund des Free Hosting Angebots)
2	07.01.2013	Michael Ruhnau	Erfassung Karteikarte: Keine Vorschau implementiert	Gemäss Anforderung e-learning FA_0005.2 soll vor dem Speichern eine Vorschau erscheinen. Diese ist noch nicht implementiert.	behoben	Vorschau wurde eingerichtet (Stefan Bregenzner)
3	09.01.2013	Michael Ruhnau	Umblättern	Das Umblättern der Karteikarten funktioniert nicht ordnungsgemäss. Es erscheint bei mir ein JavaScript-Fehler im Browser (Firefox 17).	behoben	JavaScript-Datei wurde beim Update des Drupal Core überschrieben. Nach einer Wiederherstellung vom Backup funktioniert es nun wieder. (Stefan Bregenzner)
4	09.01.2013	Michael Ruhnau	Workflow zur Freigabe	Es wird keine E-Mail-Benachrichtigung gesendet	behoben	siehe Issue 1
5	12.01.2013	Michael Ruhnau	Datenbank	Beim Öffnen einer Kategorie (Lernabschnitt) unter Datenbank wird eine Liste angezeigt, die immer nur die Überschrift "Karteikarte" enthält. Hier sollte eine Liste der Karteikarten mit der Frage erscheinen. Beim Öffnen sollte sich dann die vollständige Karteikarte öffnen.	behoben	
6	14.01.2013	Kim Jansen	Freigabe	Ich kann die Karteikarten nicht in den Status "fertig gestellt" setzen.	kein Issue	Du hast nicht die Rolle QS-Prüfer. Dieser Status ist nur für diese Rolle sichtbar.

Abbildung 4 - Auszug des aktuellen Issue Log
Quelle: Eigene Darstellung

5. Gruppenfazit

Das Projekt wurde von allen Teammitgliedern positiv erlebt und das war an ihrer Motivation zu spüren, wenngleich sich das Team auch darüber einig ist, dass es eine sehr zeitintensive Studienaufgabe war.

Wir konnten in unserer kleinen Gruppe viel von dem wiedererkennen, was wir auch von realen und zum Teil größeren Projekten auf der Arbeit kannten. Das Projekt hat uns die Gelegenheit gegeben, den Softwareentwicklungsprozess einmal von Anfang bis zum Schluss miterleben (Projektidee, Planung, Durchführung, Steuerung, Koordination, Qualitätssicherung, Abschluss), von dem wir bei der Arbeit nur einen kleinen Ausschnitt kannten. Durch unsere häufige Kommunikation zu allen Fragestellungen und Aufgaben können wir uns nun besser in andere Teammitglieder hineinversetzen, um diese besser zu verstehen und Schwierigkeiten rechtzeitig zu erkennen. Die Zusammenarbeit in der Gruppe war proaktiv und zielorientiert. Die Planung wurde gemeinsam durchgeführt, was gerade dadurch ermöglicht wurde, dass jedes Teammitglied andere Kompetenzen in die Gruppe einbrachte. Das Team hat sich gegenseitig ergänzt, und stets zusammen gearbeitet, was aufgrund der zufälligen Gruppenbildung nicht selbstverständlich gewesen ist. Es war zu beobachten, dass das Team bis zum Schluss motiviert an einem guten Projektergebnis gearbeitet hat und nun stolz ist, dieses erreicht zu haben. Wir sehen außerdem den hohen Lerneffekt, den die Erstellung einer praktischen Arbeit im

Vergleich zu einem gespielten Projekt gebracht hat. Darüber hinaus, kann das erstellte E-Learning-Systems zur Klausurvorbereitung genutzt werden.

Das Projekt musste nicht nur mit den Arbeitgebern der Teammitglieder, sondern auch mit anderen VAWI-Kursen um ihre Zeit konkurrieren. Dies war nicht immer leicht. Beruflich bedingt musste Ralf Gemmel verreisen. Während Urlauben über und nach Weihnachten waren Projektmitglieder nicht zu erreichen. Außerdem waren vor Weihnachten und im Januar Studienleistungen für andere Kurse zu erstellen, wodurch der Stresspegel sein Maximum erreichte. Diese zeitlichen Probleme konnten gemeinschaftlich bewältigt werden. Hier hat sich auch der gute Zusammenhalt in der Gruppe gezeigt.

Auffällig war, dass der Kommunikationsaufwand in einer virtuellen Projektgruppe wesentlich höher war, als in einem realen Projekt. Dies wurde schnell dadurch sichtbar, dass unsere wöchentliche Telefonkonferenz für eine Stunde geplant wurde und die tatsächliche Dauer zwischen 1:30 und 2:00 Stunden lag. Dem Team wurde der "Luxus" klar, wenn Themen zwischen den Bürotüren besprochen werden können. Als Hilfsmittel haben wir die elektronischen Systeme "GoogleDrive" als Dokumentenablage und Skype" für Audiokonferenzen genutzt. Nach einer kurzen Lernphase haben diese Systeme unsere Zusammenarbeit erleichtert und vielleicht dieses gute Ergebnis erst möglich gemacht.

Wie schon erwähnt waren die Studienleistungen sehr zeitintensiv. Die individuelle Studienleistung muss dreimal angefertigt werden, wiederholt sich jedoch stark. Dennoch fordert die Erstellung von den Studenten trotz Wiederholung denselben Zeitaufwand. Unserer Meinung nach wäre es ausreichend zum Schluss einen subjektiven Erfahrungsbericht zu erstellen. Gerade die erste Einzelleistung ist viel zu früh, um überhaupt konkrete Aussagen über das Team treffen zu können.

Ein zweiter Punkt sind die Termine der Studienleistungen. Zwischen der zweiten und dritten Studienleistung ist kaum genug Zeit, um das Skript durchzuarbeiten und die Leistung zu erstellen. Hierbei muss bedacht werden, dass die Studenten nicht von Oktober bis Februar ohne Urlaub durcharbeiten können. Hingegen ist die erste Studienleistung recht spät abzugeben. Hier sehen wir Potential die erste und zweite Studienleistung nach vorne zu ziehen, sodass der Zeitraum zur dritten Studienleistung verlängert wird.

Als dritten Punkt haben wir den Eindruck, dass eine praktische Aufgabe im Vergleich zu einem gespielten Projekt wesentlich mehr Zeit in Anspruch nimmt. Hier sollte die Aufgabenstellung mehr differenziert werden, sodass der Arbeitsaufwand ungefähr gleich ist. In einem Telefonat haben Sie uns Hinweise gegeben, wie wir Zeit bei der Dokumentation einsparen können. Diese Aussagen hätten jedoch konkreter ausfallen können. Zum Beispiel könnte die Aufgabenstellung so gefasst werden, dass Gruppen mit einem realen Projekt die Kapitel Kostenplanung und Risikomanagement nicht in den Studienleistungen bearbeiten müssen. Hierbei hat es uns nicht geholfen, dass wir diese weniger ausgeprägt bearbeiten konnten. Diese Aussage war zu unkonkret. Mit anderen Worten, sollten Ihre Erwartungen bei den unterschiedlichen Optionen klar formuliert sein. Abschließend ist uns aufgefallen, dass der Koordinationsaufwand mit steigender Gruppenzahl unverhältnismäßig steigt. Wenn die Gruppengröße flexibler wäre 3-4 Personen, würde die Gruppe Koordinationsaufwand vermeiden und Arbeitszeit gewinnen.

6. Anhang

6.1 Testfall T_0005: Neue Karteikarten können erfasst werden

6.1.1 Ausführung

ID Testfall	T_0005
Datum der Durchführung	14.01.2013
Durchgeführt von	Michael Ruhnau
Ergebnis	Test bestanden
Beschreibung	Erfassen einer neuen Karteikarte durch Autor, Vorlage zur Freigabe
Anforderungen	Testsystem, Benutzer mit Rolle Autor
Vorbedingungen	keine
Abhängigkeiten	Testcase T_0006

6.1.2 Ablauf des Tests

Nr.	Beschreibung
1	Login mit Benutzer mit Rolle Autor
2	Klick auf Link "Karteikarte anlegen"
3	Klick auf Button "Preview"
4	<ul style="list-style-type: none"> • Status auf „zur QS vorgelegt“ setzen • Optional Eingabe eines Kommentars • Klick auf Button "Save"
	Include: Durchführung Testcase T_0006
5	Karteikarte wird nach Freigabe als freigegeben angezeigt

6.1.3 Erwartetes Ergebnis

Nr.	Beschreibung
1	Erfolgreiches Login in das System, Link "Karteikarte" anlegen ist sichtbar
2	Formular zur Erfassung der Karteikarten öffnet sich, Felder <ul style="list-style-type: none"> • Lernabschnitt

	<ul style="list-style-type: none"> • Vorderseite • Rückseite • Seitenzahl sind bearbeitbar
3	Vorschau der eingegebenen Karteiarte wird angezeigt
4	Karteikarte wird gespeichert, E-Mail wird an QS Prüfer gesendet
5	Nach Durchführung von Testcase T_0006: Karteikarte wird als freigegeben angezeigt

6.1.4 Eingetretenes Ergebnis

Nr.	Beschreibung	Ergebnis
1	<ul style="list-style-type: none"> • Erfolgreiches Login in das System • Link "Karteikarte" anlegen ist sichtbar 	ok
2	Formular zur Erfassung der Karteikarten öffnet sich, Felder <ul style="list-style-type: none"> • Lernabschnitt • Vorderseite • Rückseite • Seitenzahl sind bearbeitbar	ok
3	Vorschau der eingegebenen Karteiarte wird angezeigt	ok
4	Karteikarte wird gespeichert, E-Mail wird an QS Prüfer gesendet	ok
5	Nach Durchführung von Testcase T_0006: Karteikarte wird als freigegeben angezeigt	ok

6.2 Testfall T_0006: Workflow zur Freigabe von Karteikarten

6.2.1 Ausführung

ID Testfall	T_0006
Datum der Durchführung	14.01.2013
Durchgeführt von	Michael Ruhnau
Ergebnis	Test bestanden
Beschreibung	Freigabe einer zur QS vorgelegten Karteikarte
Anforderungen	Testsystem, Benutzer mit Rolle QS
Vorbedingungen	erfolgreiche Ausführung Testfall T_0005
Abhängigkeiten	Testcase T_0006

6.2.2 Ablauf des Tests

Nr.	Beschreibung
1	E-Mail "VAWI-Prüfungstraining - ausstehende QS-Prüfung" öffnen und Link zur Karteikarte anklicken
2	Login mit Benutzer mit Rolle QS
3	Klick auf Register "Workflow"
4	<ul style="list-style-type: none"> • Status auf „fertig gestellt“ setzen • Optional Eingabe eines Kommentars • Klick auf Button "Update Workflow"
	Include: Fortführung Testcase T_0005

6.2.3 Erwartetes Ergebnis

Nr.	Beschreibung
1	<ul style="list-style-type: none"> • E-Mail enthält Link zur Karteikarte • Klicken des Links lädt Anwendung
2	Erfolgreiches Login im System, Karteikarte wird geladen

3	Register „Workflow“ öffnet sich im Bearbeitungsmodus
4	Karteikarte wird freigegeben und gespeichert, E-Mail wird an Autor gesendet

6.2.4 Eintretenes Ergebnis

Nr.	Beschreibung	Ergebnis
1	<ul style="list-style-type: none"> • E-Mail enthält Link zur Karteikarte • Klicken des Links lädt Anwendung 	ok
2	Erfolgreiches Login im System, Karteikarte wird geladen	ok
3	Workflow register öffnet sich im Bearbeitungsmodus	ok
4	Karteikarte wird freigegeben und gespeichert, E-Mail wird an Autor gesendet	ok