

**Universität Hannover
Fachgebiet Software Engineering
Institut für Angewandte Systeme
Fachbereich Informatik**

Integration von Tailoring in eine webbasierte Prozess- Beschreibung

Bachelorarbeit

im Studiengang Informatik

von

Leif Singer

**Prüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Prof. Dr.-Ing. Christian Grimm**

Hannover, 29. September 2005

Inhaltsverzeichnis

Kurzfassung	4
1 Einleitung	5
Das Unternehmen WABCO.....	5
Zielsetzung des Projekts bei WABCO.....	5
Projektstruktur.....	6
Gliederung dieser Arbeit.....	6
1 Einleitung.....	6
2 Sub-Projekt "Tailoring".....	6
3 Sub-Projekt "SPIN Erweiterung".....	6
4 Zusammenfassung & Ausblick.....	6
2 Sub-Projekt "Tailoring"	7
Tailoring in der Software-Technik.....	7
Definition.....	7
Ablauf und Bestandteile.....	7
Der Common Product Development Process.....	8
Die Roadmap Software.....	11
Reviews.....	11
Konzept.....	11
Prototyp & Industrialisierung.....	12
Validierung.....	14
Stabilisierung.....	14
Wahl und Tailoring des eigenen Prozesses.....	14
Anforderungserhebung mit den Qualitätsmanagern.....	15
Tailoring bei WABCO.....	15
Altlasten.....	16
Varianten und Fehlerbehebungen.....	16
Bürde Roadmap.....	16
Respekt vor dem Prozess.....	17
Breite und Tiefe.....	17
Identifizierte Anforderungen.....	17
Entwurf: Einführung von Projekttypen.....	17
Umzusetzende Anforderungen.....	17
Projekttyp Weiterentwicklung.....	18
Projekttyp Kleine Änderung.....	19
Verfahren beim Tailoring der Projekttypen.....	19
Implementierung.....	20
Projekttyp Weiterentwicklung.....	20
Projekttyp Kleine Änderung.....	22
Tests.....	25
3 Sub-Projekt "SPIN Erweiterung"	26
Vorstellung des Systems.....	26
Webbasierte Informationssysteme.....	26
HTTP-Anfragen und -Antworten.....	26
Das "Front Controller" Design Pattern.....	27
Model-View-Controller in Webanwendungen.....	27
Analyse des vorhandenen Systems.....	28
Einstieg.....	28
Die Klasse Theme.....	29
Die "Modul"- und "Prozess"-Klassen.....	29
Wahl und Tailoring des eigenen Prozesses.....	29

Anforderungserhebung.....	30
Entwurf: Übernahme des Vorhandenen.....	31
Implementierung.....	32
<i>Neue Funktionalitäten.....</i>	<i>32</i>
<i>Anpassungen.....</i>	<i>33</i>
<i>Zusammenführung der beiden Sub-Projekte.....</i>	<i>33</i>
Tests.....	33
<i>Testen von Webanwendungen.....</i>	<i>34</i>
<i>Abhängigkeiten vom Entwurf.....</i>	<i>34</i>
4 Zusammenfassung & Ausblick.....	35
Bedeutung der Arbeit.....	35
Bewertung der Durchführung.....	35
Ausblick.....	36
5 Anhang.....	38
Auszug aus der Roadmap "SPIN Erweiterung"	38
Abkürzungen und Glossar.....	42
Quellenverzeichnis.....	45
Eidesstattliche Erklärung.....	46

Kurzfassung

Die Anwendung eines Software-Entwicklungsprozesses ist in den meisten Organisationen inzwischen eine Selbstverständlichkeit. Doch hat jedes Projekt Eigenheiten, die nicht durch einen allgemein gehaltenen Entwicklungsprozess erfasst werden können. Um trotzdem eine möglichst optimale Prozessleitung zu erreichen ist es oftmals üblich, den Prozess an diese Bedürfnisse anzupassen. Dieser Vorgang nennt sich *Tailoring*. Dessen Durchführung ist auf Basis verschiedener Ansätze, von der intuitiven, auf impliziten Erfahrungen basierenden Beurteilung bis hin zu genau definierten Tailoring-Regeln denkbar.

Diese Arbeit dokumentiert, wie ich bei der Firma WABCO einen auf die dortigen Gegebenheiten ausgerichteten Tailoring-Ansatz entwickelte und in die bei WABCO bereits vorhandene webbasierte Prozess-Beschreibung *SPIN* integrierte. Hierbei wandte ich den bei der Firma WABCO seit etwa 2001 eingeführten Common Product Development Process an, der ebenfalls vorgestellt wird.

Der Leser erhält eine Einführung in das Tailoring in der Software-Technik und bekommt anhand der Beschreibung der praktischen Arbeit einen Einblick in die Anwendung dieses Konzepts sowie die hierbei notwendigen Abwägungen.

1 Einleitung

Das Unternehmen WABCO

Das in dieser Bachelorarbeit beschriebene Projekt wurde in den Monaten Juni bis September 2005 bei der WABCO Development GmbH in Hannover durchgeführt. Diese repräsentiert die Sparte Forschung und Entwicklung der WABCO Vehicle Control Systems, welche eine Tochterfirma der American Standard Companies, Inc. ist. WABCO ist mit etwa 6500 Mitarbeitern weltweit und 2400 am Standort Hannover einer der führenden Hersteller von Fahrzeugregelsystemen, insbesondere elektronischer Bremssysteme wie ABS und EBS für Lastkraftwagen. WABCO ist unterteilt in die Business Units (BUs)

- Drucklufterzeugung und Bremsen,
- Antriebsstrang,
- Anhänger-Systeme,
- Fahrdynamik-Regelung und
- PKW-Systeme.

Bei der Erstellung derlei sicherheitskritischer Software ist es unabdingbar, entsprechend ausgefeilten Entwicklungsprozessen zu folgen. Dies wird organisiert durch das Technologiezentrum T3. Hier fand die in dieser Arbeit dokumentierte Entwicklung statt. Es ist verantwortlich für den Software-Entwicklungsprozess, Qualitätssicherung und die projektübergreifende Standardisierung von Werkzeugen und Methoden.

Betreut wurde ich durch den Leiter desjenigen T3-Teams, das für die Definition des Software-Entwicklungsprozesses zuständig ist. Er nahm auch die Rolle des Kunden ein. In dieser Arbeit werden abwechselnd der Kunde und mein Betreuer erwähnt; dies meint stets die selbe Person, jedoch in der jeweils genannten Rolle.

Zielsetzung des Projekts bei WABCO

Um die verschiedenen bei WABCO laufenden Projekte in der Umsetzung des Software-Entwicklungsprozesses und der Qualitätsanforderungen zu unterstützen gibt es sieben, über die Business Units verteilte Qualitätsmanager. Diese helfen den Projektteams, den Prozess an die Bedürfnisse des Projekts anzupassen – dieser Vorgang nennt sich Tailoring ("Zurechtschneiden") des Prozesses.

Des weiteren existiert im WABCO-Intranet eine webbasierte Prozess-Beschreibung namens *SPIN*, die den Software-Entwicklungsprozess strukturiert und übersichtlich darstellt.

Zur Entlastung der einzelnen Projekte bei Projektstart war es der Wunsch des Kunden, über *SPIN* zusätzlich Hilfen zum Tailoring anzubieten. Es sollte also eine möglichst automatisierte Methode gefunden werden, das Tailoring abzukürzen und diese Möglichkeit dann in *SPIN* zu integrieren.

Projektstruktur

In Absprache mit meinem Betreuer unterteilte ich diese Aufgabe in zwei Sub-Projekte, namentlich "Tailoring" und "SPIN Erweiterung".

Im Sub-Projekt "Tailoring" sollte ein Tailoring-Ansatz gefunden werden, der sich für die Integration in die webbasierte Prozess-Beschreibung eignet und die Anpassung des Software-Entwicklungsprozesses beim Projektstart erleichtert.

Im Sub-Projekt "SPIN Erweiterung" sollte dieser Ansatz in die vorhandene webbasierte Prozess-Beschreibung *SPIN* möglichst geeignet integriert werden, so dass die Qualitätsmanager eine intuitiv bedienbare Hilfestellung beim Tailoring hätten.

Gliederung dieser Arbeit

Diese Arbeit beginnt mit einer Kurzfassung, die die Problemstellung und den Lösungsansatz kurz umreißt. Sie wird gefolgt von fünf Kapiteln:

1 Einleitung

Die Einleitung gibt einen kurzen Überblick über die Firma WABCO, die Aufgabenstellung und die vorliegende Arbeit.

2 Sub-Projekt "Tailoring"

In diesem Kapitel wird der Ablauf des Sub-Projekts "Tailoring" beschrieben. Es beginnt mit einer Einführung zum Tailoring in der Softwaretechnik und beschreibt dann den bei WABCO angewandten Common Product Development Process sowie die darin enthaltene Roadmap Software im Detail. In der Folge wird das Tailoring des eigenen Prozesses begründet sowie die eigentliche Durchführung des Projekts in Phasen unterteilt dargestellt.

3 Sub-Projekt "SPIN Erweiterung"

In der Erläuterung des Sub-Projekts "SPIN Erweiterung" gebe ich zunächst einen Überblick über webbasierte Informationssysteme im Allgemeinen und beschreibe dann in diesem Kontext die Analyse des bereits vorhandenen Systems *SPIN*. Auch für dieses Sub-Projekt lege ich die Gründe für das Tailoring des eigenen Prozesses dar und beschreibe dann die Durchführung.

4 Zusammenfassung & Ausblick

Dieses Kapitel fasst die Arbeitsergebnisse zusammen und wirft einen kritischen Blick auf den Projektverlauf. Es werden mögliche Weiterentwicklungen, Abwandlungen und Alternativen bezüglich der Arbeitsergebnisse im Hinblick auf weitere Wünsche und Pläne der Firma WABCO bzw. des Technologiezentrums T3 diskutiert.

Den Abschluss bildet der Anhang. Hier befinden sich Glossar, Abbildungs-, Tabellen- und Quellenverzeichnis. Ebenso ist hier aus anschaulichen Gründen ein Auszug (das Serienrelease) aus der Roadmap Software zum Sub-Projekt "SPIN Erweiterung" zu finden. Der Name meines Betreuers ist hier unkenntlich gemacht.

2 Sub-Projekt "Tailoring"

Tailoring in der Software-Technik

Definition

Die englischen Begriffe "Tailoring" bzw. "(to) tailor" sind zu übersetzen mit "zurechtschneiden". Im Rahmen dieser Arbeit ist hiermit das in der Software-Technik mögliche Anpassen eines Software-Entwicklungsprozesses gemeint. Diese Ausweitung der Bedeutung weg vom klassischen "Schneiden" hin zu allgemeineren Konzepten ist bereits in der englischen Sprache vorhanden:

"*tai-lored, tai-lor-ing, tai-lors*: 1. [...] 2. [...] 3. To make, alter, or adapt for a particular end or purpose [...]" [DICT]

Folgend findet sich eine mögliche Definition, die das Tailoring in der Software-Technik beschreibt:

"*Tailor*: To modify a process, standard or procedure to better match process or product requirements." [CMM]

Das Ziel des Tailorings ist es also, einen gegebenen Prozess an die Bedürfnisse und Einschränkungen eines konkreten Projektes anzupassen. Hiermit soll unter Anderem erreicht werden, dass Umfang und Tiefe des Prozesses dem Projekt angemessen sind.

Ein sehr aktueller Bezug zum Tailoring findet sich im *V-Modell XT*, das im Februar 2005 veröffentlicht wurde. Es ist der Nachfolger des *V-Modell 97* und bringt an Neuerungen vor allem die Unterstützung für agilere Projekte und ein Tailoring des Prozesses mit. Dies wird unter anderem erreicht durch ein modulares Baukastensystem, mit dem projektspezifische Prozesse aus dem *V-Modell XT* abgeleitet werden können.

Ablauf und Bestandteile

Vor Beginn des Tailorings sind ein *Prozess* und ein *Projekt* gegeben. Des weiteren gibt es eine Menge von *Tailoring-Regeln* aus verschiedenen Quellen, die die Anpassung des Prozesses reglementieren und lenken.

Der *Prozess* ist ein gerichteter Graph aus hierarchisch verschachtelten Aktivitäten. Einige der Aktivitäten produzieren definierte Artefakte. Für den Übergang von einer Aktivität in eine Andere gibt der Prozess Regeln vor – beispielsweise müssen andere Aktivitäten ebenfalls abgeschlossen oder bestimmte Artefakte erzeugt worden sein. Der Prozess wird meist von der Organisation vorgegeben, beispielsweise um effizienter und kostengünstiger zu entwickeln oder Qualitätsanforderungen von Kunden gerecht werden zu können.

Das *Projekt* hat spezifische Anforderungen an den Prozess bezüglich seiner Komplexität, seiner Formalität und der benötigten Kontrolle. Diese müssen beim Tailoring des Prozesses berücksichtigt werden.

Die *Tailoring-Regeln* können stammen aus

- Vorgaben der Organisation,

Leif Singer: Integration von Tailoring in eine webbasierte Prozess-Beschreibung

- expliziten, dokumentierten Erfahrungen aus dem Tailoring vorhergehender Projekte,
- impliziten Erfahrungen der am Tailoring beteiligten Mitarbeiter,
- Gesetzen und Normen (hier als konkretes Beispiel: die SPICE-Norm; aber auch: gesetzliche Anforderungen an die Sicherheit von Kraftfahrzeugen).

Die Tailoring-Regeln treffen Aussagen darüber, wie mit Aktivitäten beim Tailoring zu verfahren ist. Dies kann einem der folgenden Fälle entsprechen:

- Manche Aktivitäten müssen genau so durchgeführt werden, wie sie vom Prozess gegeben sind; sie sind nicht anpassbar.
- Einige Aktivitäten dürfen unter bestimmten Bedingungen
 - aus dem projektspezifischen Prozess entfernt werden,
 - an die Bedürfnisse des Projekts angepasst werden, also in ihrem Umfang, ihrer Tiefe und ihrer Formalität verschärft oder verringert werden,
 - durch andere Aktivitäten ersetzt werden.
- Bestimmte Aktivitäten dürfen hinzugefügt werden.

Nach Anwendung der Tailoring-Regeln erhält man den neuen, getailorten Prozess. Der Vorgang des Tailorings wird in Abbildung 1 verdeutlicht.

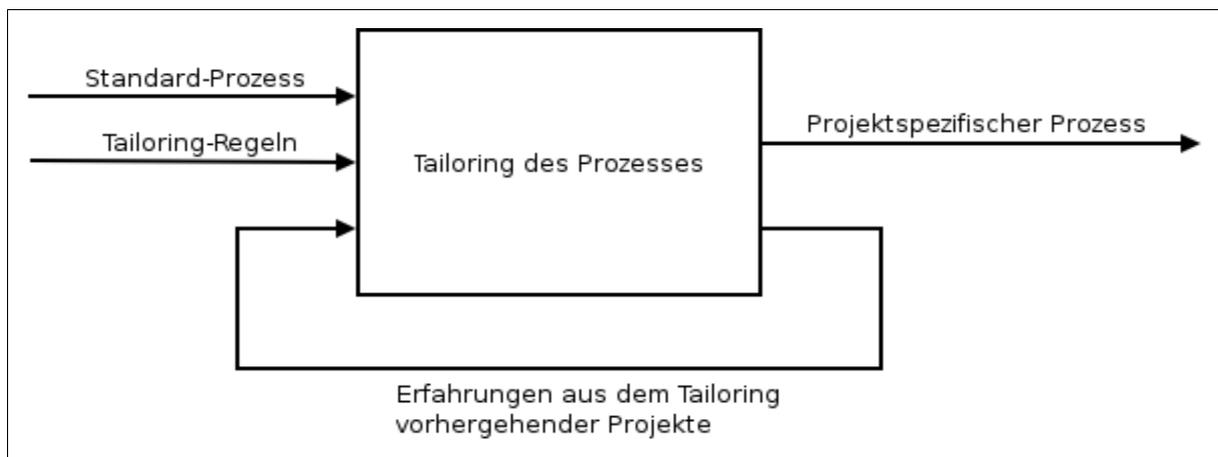


Abbildung 1: Prinzipieller Ablauf des Tailorings in der Software-Technik

Einige der soeben in diesem Abschnitt dargelegten Informationen wurden – nicht wörtlich – [STSC] entnommen.

Der Common Product Development Process

Der bei WABCO angewandte Prozess ist der Common Product Development Process, kurz CPDP. Er unterteilt sich in die sechs Subprozesse

- Projektmanagement,
- System-Entwicklung,
- mechanische Hardware-Entwicklung,

- elektronische Hardware-Entwicklung,
- Software-Entwicklung und
- externe Produkt-Entwicklung.

Hinzu kommt der relativ schlank gehaltene Subprozess "Small Variants", der nur für sehr kleine Projekte mit einem Budget von unter 60.000 EUR verwendet werden darf.

Für jeden dieser Subprozesse stellt der CPDP eine so genannte Roadmap bereit. Dies ist eine strukturierte Zusammenstellung der jeweils anfallenden Tätigkeiten in einer "Microsoft Excel"-Datei und ist jeweils in die sechs Phasen

- Definition,
- Konzept,
- Prototyp,
- Industrialisierung,
- Validierung und
- Stabilisierung

unterteilt. Pro Phase sind die jeweils relevanten Aktivitäten in einer hierarchischen Struktur aufgelistet und mit Vorlagen, Arbeits- und Verfahrensanweisungen versehen.

Tabelle 1 stellt die verschiedenen Sub-Prozesse und Phasen im Zusammenhang der Entwicklung eines kompletten Produkts dar.

<i>Definition</i>	<i>Konzept</i>	<i>Prototyp</i>	<i>Industrialisierung</i>	<i>Validierung</i>	<i>Stabilisierung</i>
Projektmanagement					
	System-Entwicklung				
	Mechanische Hardware-Entwicklung				
	Elektronische Hardware-Entwicklung				
	Software-Entwicklung				
	externe Produkt-Entwicklung				

Tabelle 1: Sub-Prozesse und ihre Phasen im CPDP, nach [STRUEBER]

Eine solche Roadmap wird für jedes (Sub-)Projekt pro betroffenem Subprozess geführt. Im Verlauf des Projekts wird sie mit Anmerkungen und Hyperlinks zu entstandenen Dokumenten gefüllt. Atomare Aktivitäten werden mit einem Status versehen ("not started", "in progress", etc.), welcher in übergeordneten, zusammengesetzten Aktivitäten automatisch geeignet reflektiert wird. Sind beispielsweise alle atomaren Tätigkeiten in der Implementierung fertiggestellt, so erhält auch diese den Status "fertiggestellt". Die Statusangaben sind mit jeweils intuitiv verständlichen Farben hinterlegt (rot für "nicht termingerecht fertiggestellt", grün für "rechtzeitig fertiggestellt", weiß für "nicht anwendbar", gelb für "nicht angefangen" und "in Arbeit"). Für jede Aktivität sind in der Roadmap gegebenenfalls vorhandene

Arbeitsanweisungen und Vorlagen verlinkt. Diese liegen als PDF-Dateien und Microsoft Office Dokumente vor.

Die Roadmaps der beteiligten (Sub-)Projekte, die erzeugten Dokumente und Produktteile werden in einem Konfigurationsmanagementsystem gehalten. Dies entspricht einem Versionskontrollsystem nicht nur für Quellcode, sondern für ganze Konfigurationen. Eine Konfiguration ist dementsprechend der Zustand eines Projektes zu einem bestimmten Zeitpunkt und beinhaltet jegliche vom Prozess erzeugte Dokumente, Quellcode und Testumgebungen wie beispielsweise verwendete Compiler.

Im Rahmen dieser Arbeit ist nur die Roadmap Software von Relevanz. In dieser sind die Phasen Prototyp und Industrialisierung zu einer einzigen Release-Phase zusammengefasst. Sie wird vom ersten über das n-te bis zum Serienrelease im Sinne inkrementeller Software-Entwicklung wiederholt. Als Beispiel ist im Anhang ein Auszug aus einer von mir im Zusammenhang mit dieser Arbeit geführten Roadmaps beigefügt – es ist das Serienrelease des Sub-Projekts "SPIN Erweiterung". Zur weiteren Verdeutlichung trägt Abbildung 2 bei.

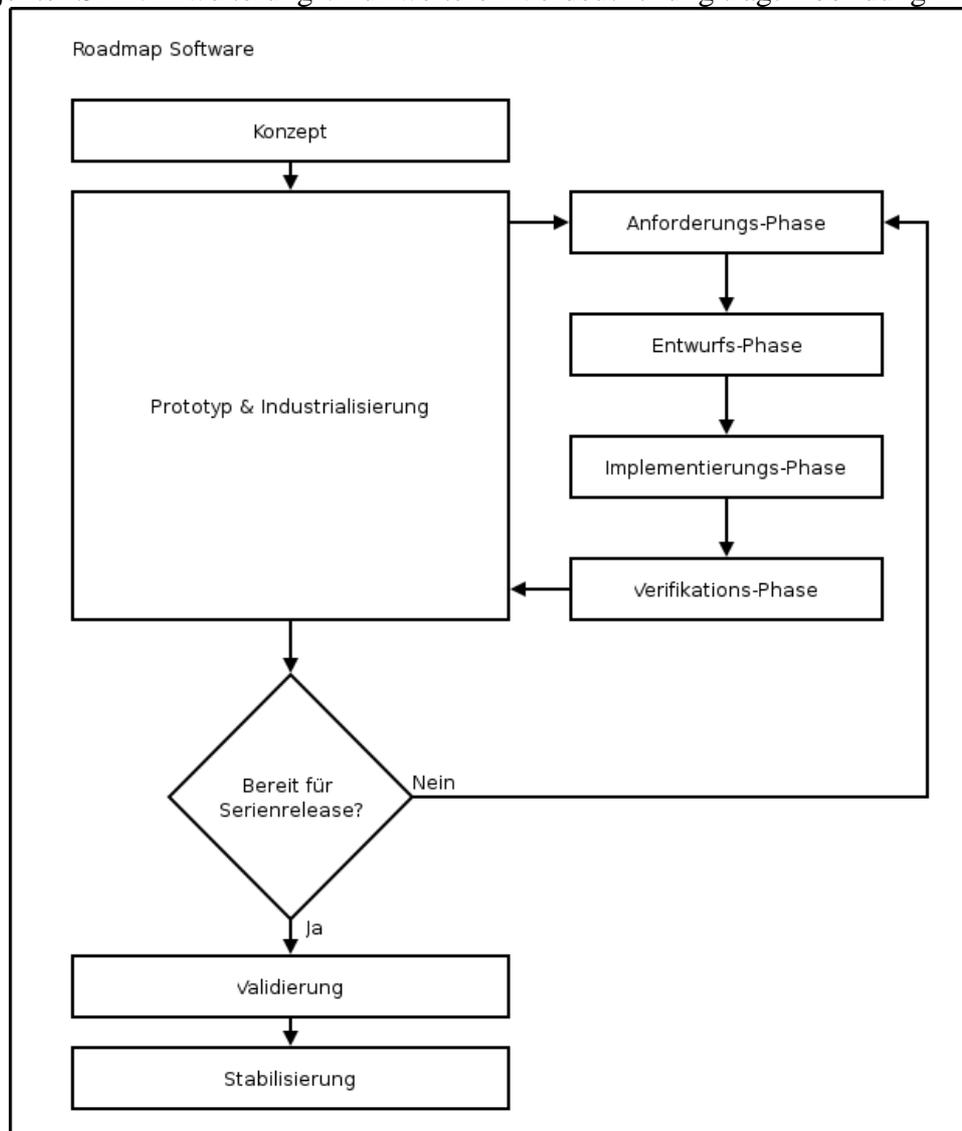


Abbildung 2: Phasenübergänge in der Roadmap Software

Zur Koordination und Unterstützung in der Anwendung des CPDP gibt es über die Business Units verteilt 2 Vollzeit- und 5 Teilzeit-Software-Qualitätsmanager (QMs). Hierin enthalten ist ein zentraler QM, der wiederum das Software-Qualitätsmanagement koordiniert. Alle QMs sind beteiligt an Prozess-Definition und -Anpassung, Beratung der Projektteams, Förderung, Überwachung und Analyse des Software-Entwicklungsprozesses.

Die Tätigkeiten der Software-Qualitätsmanager schließen Unterstützung und Genehmigung projektspezifischer Anpassungen des Prozesses an die jeweiligen Bedürfnisse eines Projekts mit ein: zusammen mit den Projektteams nehmen sie das Tailoring vor.

Die Roadmap Software

Zunächst möchte ich die Roadmap Software und die in ihr verlinkten Dokumente noch näher beschreiben. In der Roadmap Software existieren Aktivitäten für die Phasen "Konzept", "Prototyp & Industrialisierung" (zusammengefasst), "Validierung" und "Stabilisierung". Für die im CPDP erste Phase "Definition" fallen hier keine Tätigkeiten an. Jede Phase ist als eigenes Arbeitsblatt organisiert, wobei ein Weiteres für die Ergebnisse der Phasenreviews vorhanden ist.

Da eine grobe Kenntnis der im Prozess relevanten Dokumente für das Verständnis der später beschriebenen Implementierung erforderlich ist stelle ich im Folgenden jede Phase sowie die in ihr enthaltenen Vorlagen kurz vor.

Reviews

Auf dem ersten Arbeitsblatt der Roadmap Software befindet sich eine Projekt-Übersicht sowie die Ergebnisse der bisher durchgeführten Phasen-Reviews. Diese sind in der Regel nach dem Abschluss einer jeden Phase durchzuführen und prüfen den Fortschritt sowie die Prozesskonformität des Projekts. Anwesend sind gegebenenfalls der Sub-Projektleiter, der Projektleiter, ein Qualitätsmanager sowie die Leiter des entsprechenden Produktsegments und des Produktionszentrums.

Am Ende des Reviews wird das Projekt nach den Risikoaspekten Zeit, Kosten und Technik bewertet. Eine Bewertung ist einer der Werte "grün", "gelb" oder "rot". Eine grüne Bewertung entspricht einem problemlosen Verlauf. Eine gelbe Bewertung steht für Probleme, die einem Fortschreiten in die nächste Phase nicht widersprechen. Bei einer roten Bewertung muss vom Projektleiter und gegebenenfalls vom Leiter des Produktionszentrums beurteilt werden, ob die diese Bewertung verursachenden offenen Punkte ein Verweilen des Projekts in der aktuellen Phasen zur Folge haben sollten.

Konzept

In der Konzept-Phase wird zunächst die vom Projekt benötigte Infrastruktur aufgesetzt. Die komplette Zeit- und Kostenplanung wird initial durchgeführt. Auch das Tailoring des projekteigenen Prozesses geschieht hier. Die erste Anforderungsspezifikation wird begonnen.

Zu erzeugende Dokumente sind

- die Projektplanung in Form eines "Microsoft Project"-Projekts,
- der Software-Releaseplan, in dem Anforderungen aus Lastenheft und Anforderungsspezifikation den geplanten Releases zugeordnet werden,

- der Software-Qualitätsplan, in dem festgehalten wird, welche Qualitätsziele das Projekt erreichen möchte und wie dies zu geschehen hat,
- die Projektstruktur, in der dargestellt wird, welche Mitarbeiter in welchem Rollen und mit welcher Kapazität am Projekt beteiligt sind,
- die erste Anforderungsspezifikation, die erst im ersten Release ihren eigentlichen Umfang erreicht,
- die Anforderungsverfolgungsliste, in der zu jeder funktionalen Anforderung ihr Status, ihre Priorität, der Ort ihrer Umsetzung in Entwurf und Quellcode sowie der Test-Zustand (Test-Prozedur, Testfall, Test-Report) festgehalten wird und
- der Konfigurationsmanagement-Plan. Hier wird dokumentiert, wie, wo, wann und von wem Änderungen an der aktuellen Konfiguration des Produkts im Konfigurationsmanagementsystem vorgenommen werden dürfen und müssen.

Prototyp & Industrialisierung

Diese beiden Phasen wurden in der Roadmap Software zu einer einzigen Phase zusammengefasst. Dies ist begründet durch die fließenden Grenzen, die relativ zur Hardwareentwicklung zwischen diesen Phasen in der Software-Entwicklung bestehen. Ebenso ist hier die verschiedene Gewichtung entscheidend: ein Softwareprodukt für die Massenfertigung vorzubereiten (seine Industrialisierung durchzuführen) ist ungleich simpler als die Industrialisierung eines Hardwareprodukts, bei der gegebenenfalls noch geeignete Produktionsmethoden entwickelt werden müssen.

Der einmalige Durchlauf dieser Phase entspricht einem Release. Sie kann also so oft wiederholt werden, bis ein zufriedenstellendes Endprodukt als Serienrelease herausgegeben werden kann. Da der erste und letzte Durchlauf Sonderfälle darstellen sind hierfür jeweils gesonderte Arbeitsblätter vorhanden, die leicht abweichende Formulierungen und Aufgaben enthalten. Zwischen dem ersten und dem Serienrelease befindet sich das "n. Release" – dieses Arbeitsblatt wird für jedes Release zwischen dem ersten und dem letzten kopiert und stellt die iterierte Phase dar.

Da dies die umfangreichste Phase ist, stelle ich sie unterteilt in die Sub-Phasen Anforderungs-, Entwurfs-, Implementierungs- und Verifikations-Phase vor. Des weiteren stelle ich die Tätigkeiten vor, die unabhängig von den Sub-Phasen durchzuführen sind.

Tätigkeiten unabhängig von den Sub-Phasen

Die Projektplanung ist zu aktualisieren, d.h. es sind Kosten- und Zeitpläne zu kontrollieren und Planungsbesprechungen durchzuführen. Der Zustand des Projekts im Konfigurationsmanagementsystem ist aktuell zu halten. Hierzu zählt das formale Einpflegen von Änderungen und das bereitstellen definierter Zustände der Projekt-Konfiguration. Das einzige betroffene Dokumente, das bisher noch nicht erwähnt worden ist, ist

- der Software-Änderungsantrag, der je nach Projektfortschritt und vom Projekt benötigter Formalität bei Änderungen an bereits beschlossenen Planungen in Anforderungen, Entwurf oder Implementierung formal gestellt werden muss.

Anforderungs-Phase

In der Anforderungs-Phase wird die Anforderungsspezifikation mit allen bekannten Anforderungen gefüllt. Entsprechend der Bedürfnisse der Software-Entwicklung werden diese zunächst identifiziert, dann analysiert und schließlich spezifiziert. Um die Qualität der Anforderungsspezifikation im Speziellen und die des Produktes im Allgemeinen sicherzustellen enthält diese Phase zusätzlich noch qualitätsrelevante Tätigkeiten, wie beispielsweise eine Inspektion der Anforderungsspezifikation. Diese Qualitätssicherung beinhaltet folgende Dokumente:

- die Software-Teststrategie, in der definiert wird, welche Arten von Erzeugnissen auf welche Weise und in welcher Umgebung zu testen sind,
- den Software-Prüfplan, der konkrete Erzeugnisse angemessenen Testverfahren zuordnet,
- das Inspektionsprotokoll, auf Basis dessen prinzipiell alle Inspektionen dokumentiert werden und
- die Software-Testspezifikation, in der Testfälle definiert und Anforderungen zugeordnet werden.

Nach dieser Sub-Phase ist ein Review vorgesehen, allerdings weniger umfangreich als die bereits erwähnten Phasen-Reviews, die sich nur auf die Hauptphasen beziehen.

Entwurfs-Phase

Die Entwurfs-Phase beinhaltet die Erstellung des Entwurfs und der Schnittstellenbeschreibungen sowie deren Inspektion und die Vorbereitung der Tests. Im Kontext dieser Arbeit interessante, neu hinzugekommene Dokumente sind

- der Software-Grobentwurf, der die Strukturierung der Software in Komponenten vornimmt und den einzelnen Komponenten Anforderungen zuordnet,
- der Software-Feinentwurf auf Komponenten-Ebene, der jede Komponente in Module zerlegt und diesen jeweils Anforderungen zuordnet,
- der Software-Feinentwurf auf Modul-Ebene, der je Modul die in ihm enthaltenen Funktionen auflistet und
- die Schnittstellenbeschreibungen, die die Parameter und Rückgabewerte der definierten Funktionen beschreiben. Sie werden meist als Kommentare im Quellcode gehalten.

Auch nach dieser Sub-Phase ist ein Review vorgesehen um den Projektfortschritt zu kontrollieren.

Implementierungs-Phase

In der Implementierung-Phase werden zunächst projektspezifische Regelungen bezüglich der zu verwendenden Tools, Namenskonventionen für Quell-Dateien, Kommentare, Schlüsselwörtern, Schreibweisen und Formatierungen festgehalten. Zunächst wird dann sichergestellt, dass alle für das aktuelle Release notwendigen Anforderungen implementiert sind. In der Folge werden die Software-Module erstellt und integriert. Im Anschluss geschehen Prüfungen, die teilweise automatisiert sind, wie beispielsweise die Analyse des Quellcodes durch ein geeignetes Werkzeug und die Durchführung von Modul-Tests. Zum

Abschluss werden je nach der im Prüfplan festgelegten Regelung Code-Inspektionen durchgeführt.

In dieser Phase erscheint als neues Dokument lediglich

- der Test-Report. Er dokumentiert für einen Test die getesteten Objekte, den Testverlauf und -ausgang sowie die Bewertung von Umfang und Verlauf des Tests.

Verifikations-Phase

In die letzte Sub-Phase eines Releases, die Verifikations-Phase, fallen alle noch nicht durchgeführten Tests und integrierenden Tätigkeiten, d.h. die Integration der Module in ihre Komponenten, den Test der fertigen Komponenten sowie Integration und Test aller Komponenten im Gesamtsystem.

Damit ist ein Release abgeschlossen und es wird ein Phasen-Review durchgeführt. In diesem Rahmen wird bei Eignung ein Freigabe-Dokument ausgestellt.

In den gegebenenfalls folgenden Wiederholungen der Phase "Prototyp & Industrialisierung" sind die selben Tätigkeiten durchzuführen, wobei das Erstellen von Dokumenten durch die Änderung beziehungsweise Aktualisierung eben dieser zu ersetzen ist. Nach Abschluss des Serienreleases ist die Kunden- sowie die interne Freigabe für das Produkt einzuholen. Dies erfordert ebenfalls spezielle, definierte Dokumente, die jedoch in diesem Zusammenhang nicht relevant sind: es sind offizielle Formulare, die im Rahmen des Tailorings nicht zur Diskussion stehen.

Validierung

Die Phase "Validierung" kann nach dem Serienrelease beginnen. Hier werden nach dem Release entdeckte Fehler in Quellcode und Dokumentation beseitigt und gegebenenfalls erforderliche Tests und Inspektionen wiederholt. Auch diese Phase endet mit einem Phasen-Review.

Stabilisierung

Vor Abschluss eines Projektes steht die Stabilisierungs-Phase. Hier sind dieselben Tätigkeiten angesetzt wie auch in der Validierungs-Phase.

Wahl und Tailoring des eigenen Prozesses

Der von mir in meinen Projekten verwendete Prozess ist der CPDP, spezieller die in ihm enthaltene Roadmap Software. Dies entsprach den Kundenvorgaben. Zusammen mit meinem Betreuer tailorte ich den Prozess auf die Eigenheiten dieses Projekts zu.

Charakterisierend waren hier insbesondere die kurze geplante Dauer von nur 3 Monaten für den Durchlauf zweier kompletter Projekte sowie die besonderen Eigenschaften der zu erzeugenden Produkte. Dieses waren in diesem Projekt vorgetailorte Dokument-Vorlagen zur Verwendung mit "Microsoft Office". Die aus diesen Aspekten gezogenen Konsequenzen waren ein Verzicht auf strikte Projekt- und Terminplanung sowie der Verzicht auf automatisierte Tests.

Ein weiterer entscheidender Punkt war die Tatsache, dass es sich hier um kein offizielles und finanziertes Projekt von WABCO handelte. Einige Prozesselemente, wie beispielsweise die Kostenplanung oder eine formale Aufnahme des Projekts in die firmenweite Projektverwaltung waren hierdurch nicht oder nicht sinnvoll anwendbar.

Die beiden genannten Punkte sowie die Tatsache, dass keine anderen (Hardware-)Projekte von diesem Projekt abhingen führten dazu, dass die Anzahl der Releases auf ein einziges, das Serienrelease beschränkt wurde und die Phase Stabilisierung entfernt wurde.

Nachdem das Serienrelease abgeschlossen war wurde in Einvernehmen mit dem Kunden und dem Betreuer auf die Phase "Validierung" verzichtet, da das Projekt zufriedenstellend abgeschlossen war.

Anforderungserhebung mit den Qualitätsmanagern

Bei der Anforderungserhebung ging es darum, einen allgemeinen Tailoring-Ansatz für den Einsatz bei WABCO zu finden. Dieser sollte folgende Rahmenbedingungen erfüllen:

- erleichtert das Tailoring des projektspezifischen Prozesses
- ist in SPIN integrierbar
- der Umfang ist einer Bachelorarbeit angemessen

Der Kunde legte mir Gespräche mit den Qualitätsmanagern nahe, da diese ohnehin bereits maßgeblich am Tailoring beteiligt sind. Bei Projektstart entscheiden sie zusammen mit dem Projektteam über die Anpassung des Prozesses und sind größtenteils auch selbst an Projekten beteiligt. Sie kennen daher die Bedürfnisse der Projekte und gegebenenfalls häufig auftretende Anpassungen und Probleme.

In vier Produktsegmenten gibt es jeweils einen Teilzeit-Qualitätsmanager. In einem Produktsegment gibt es sowohl einen Teilzeit- als auch einen Vollzeit-Qualitätsmanager. Der zentrale Qualitätsmanager im T3 ist zusätzlich zuständig für ein weiteres Produktsegment.

Einer der Qualitätsmanager war während meiner Anwesenheit bei WABCO leider nicht im Haus. Ich führte mit allen verfügbaren Qualitätsmanagern jeweils ein etwa 30- bis 60-minütiges Interview. Ich bereitete mich mit speziellen und allgemeineren Fragen vor, notierte während des Interviews Stichworte und arbeitete das Erfahrene im Anschluss auf.

Zunächst konzentrierte ich mich in den Gesprächen darauf, den Prozess besser zu verstehen und Einsichten in das bisherige Tailoring zu erlangen. Nachdem ich mir hier eine Übersicht verschafft hatte versuchte ich herauszufinden, wie das Tailoring durch Software unterstützt werden könnte. Hierbei spielte ich verschiedene Lösungsansätze mit den Qualitätsmanagern durch, bis sich dann der schließlich gewählte als der Geeignetste herausbildete. Diesen Abwägungsprozess werde ich ihm Kapitel Entwurf näher beschreiben, zunächst möchte ich nun jedoch die Informationen darstellen, die ich durch die Interviews mit den Qualitätsmanagern erhielt.

Tailoring bei WABCO

Bei Beginn eines neuen Projektes wird, wie bereits erwähnt, zunächst in Zusammenarbeit eines zuständigen Qualitätsmanagers mit dem Projektteam der Prozess auf die speziellen Bedürfnisse des Projekts abgestimmt.

Diese Tätigkeit ist bei WABCO bisher nicht formal festgelegt – von der Organisation werden keine direkten Tailoring-Regeln vorgegeben. Lediglich durch die Anforderungen der SPICE-Norm ist hier eine Einschränkung gegeben. Diese besteht meinen Informationen zufolge aus der Forderung, dass der getailorte Prozess eine Untermenge des originalen Prozesses zu sein hat. Es dürfen also nur Aktivitäten entfernt oder in ihrer Tiefe geändert, jedoch nicht hinzugefügt werden.

Die Qualitätsmanager lassen sich beim Tailoring primär von den konkreten Gegebenheiten des Projekts und ihrer Erfahrung leiten. Alle Entscheidungen werden zusammen mit dem Projektteam getroffen.

Altlasten

Als häufiges Problem bei der Anwendung des Prozesses kristallisierte sich schnell die Weiterentwicklung bereits vorhandener Produkte heraus. In der Organisation der Entwicklung hat sich in den letzten Jahren nicht nur der angewendete Prozess verändert, sondern auch die Infrastruktur – beispielsweise geschieht inzwischen der Großteil der Entwicklung in der Hochsprache C, während ein großer Anteil der vorhandenen Software noch in Assembler geschrieben ist. Dies stellt eine erhebliche Hürde für die Dokumentation der Software, insbesondere des Entwurfs und der Schnittstellen, dar. Hinzu kommt, dass viele der bereits vorhandenen Projekte nicht oder nur schlecht dokumentiert sind, da dies zum Zeitpunkt der erstmaligen Produktentwicklung vom Prozess nicht oder nur in einem geringeren Umfang gefordert gewesen war. Dies ist beim zur Zeit bei WABCO gängigen Tailoring ein häufiger Ansatzpunkt: da es nicht möglich ist, diese fehlende Dokumentation vollständig nachzuarbeiten, muss der Prozess diesen Gegebenheiten angepasst werden, so dass, soweit es vertretbar ist, darauf verzichtet wird, beispielsweise den vollständigen Entwurf zu dokumentieren. Stattdessen werden nur die vom aktuellen Projekt betroffenen Bereiche dokumentiert. Dies trifft jedoch nicht auf die Anforderungsspezifikation zu – diese wird stets vollständig fertiggestellt.

Varianten und Fehlerbehebungen

Häufig vorkommende Projekte sind auch jene, in denen ein vorhandenes Produkt nur geringfügig geändert wird. Dies mag an einer kleinen Anpassung für die bei einem anderen Kunden leicht verschiedene Hardware liegen oder eine einfache Fehlerbeseitigung sein. In jedem Fall ist in diesen Projekten die Aufgabenstellung recht klar und relativ schnell umzusetzen. Alle qualitätssichernden Maßnahmen müssen natürlich trotzdem durchgeführt werden. Aber die Anzahl der zu durchlaufenden Release beschränkt sich häufig auf ein einziges Serienrelease.

Bürde Roadmap

Die bereits vorhandenen Projekte passen also teilweise nicht zum Prozess. Dies stellt für viele Mitarbeiter eine psychologische Hürde dar: man "fühlt sich schlecht", weil man die Forderungen des Prozesses nicht korrekt erfüllen kann. Die Mitarbeiter erhalten teilweise das Gefühl, gegen den Prozess arbeiten zu müssen und empfinden so bereits die Pflege der Roadmap – eine Sammlung recht einfacher Arbeitsblätter – als eine unangenehme Bürde.

Respekt vor dem Prozess

Eine weitere Hürde beim Tailoring ist der Respekt vor dem Prozess. Jede Tätigkeit hat natürlich ihren Sinn und trägt – besonders im Falle von Test-Tätigkeiten – dazu bei, die hoch gesteckten Qualitätsansprüche der Firma WABCO zu erfüllen. Daher falle es teilweise schwer, Aktivitäten in ihrer Tiefe zu verringern oder gar ganz auszulassen. Selbst wenn dies im konkreten Fall sinnvoll sein sollte, entsteht doch ein "schlechtes Gefühl" dabei.

Breite und Tiefe

Das Entfernen von ganzen Aktivitäten ist im Allgemeinen nicht häufig möglich. In der Regel beschränkt sich das Tailoring auf die Verringerung der Tiefe, mit der eine Aktivität durchgeführt wird. Dies begründet sich in den zu erfüllenden Qualitätsansprüchen und den einzuhaltenden Vorgaben der SPICE-Norm.

Identifizierte Anforderungen

Durch die Gespräche mit den Qualitätsmanagern erhielt ich die folgenden Anforderungen an einen softwaregestützten Tailoring-Ansatz:

- Unterstützung der Tailoring-Tätigkeiten bei den identifizierten häufiger vorkommenden Projekttypen.
- Umgang mit dem Prozess nicht nur praktisch, sondern auch psychologisch erleichtern – durch den Versuch der Kompensation der Unterschiede zwischen dem vom Prozess geforderten und dem wirklich vorhandenen Projekt.
- Klarere Vorgaben zum Tailoring sind notwendig, um es einerseits praktisch zu erleichtern und andererseits die psychologische Hürde vor sinnvollem Tailoring zu verringern.
- Tailoring primär in der Tiefe, d.h. am Umfang der Aktivitäten – nicht jedoch ihr Vorhandensein in Frage stellen.

Entwurf: Einführung von Projekttypen

Umzusetzende Anforderungen

Natürlich können nicht all diese Anforderungen im Rahmen einer Bachelorarbeit vollständig umgesetzt werden. Daher beschloss ich, diese zu relativieren, um sie einerseits realistisch umsetzbar zu halten, aber dennoch den größtmöglichen Nutzen aus ihnen zu ziehen. Hierzu möchte ich zunächst die von mir betrachteten Lösungsansätze vorstellen.

Als aufwendigste Lösung notierte ich mir ein vollständiges Prozess-Werkzeug, das den gesamten Prozess in einer Datenbank abbildet. Die einzelnen Prozesselemente könnten mit ihren Nutzdaten (Rollenverteilung, Arbeitsanweisungen, Vorlagen, etc.) sowie Tailoring-Regeln attribuiert werden. Bei Beginn eines neuen Projektes müssten lediglich durch die Regeln gegebene Fragen zum Projekt beantwortet werden. Diese Antworten hätten definierte Auswirkungen auf bestimmte Aktivitäten. Nach Abschluss dieses Tailorings würde das Werkzeug bereits getailorte Dokumente inklusive der Roadmap generieren. Ebenso ist in diesem Zusammenhang die Nutzung von Erfahrungen besonders gut vorstellbar,

beispielsweise durch das statistisch gestützte Herausbilden von häufig auftretenden Projekttypen auf Basis von historischen Daten.

Dieser Ansatz sprengt den Rahmen einer Bachelorarbeit. Er erfordert die Entwicklung von für die Firma WABCO allgemeingültigen Tailoring-Regeln und bietet einige nicht unerhebliche technische Hürden.

Der einfachste Ansatz wäre die Identifizierung von häufig auftretenden Projekttypen. Auf Basis dieser könnte dann jeweils ein vorgetailorter Prozess zur Verfügung gestellt werden.

Zwischen diesen beiden Ansätzen sind diverse Variationen denkbar. Ich wählte den Ansatz der Projekttypen kombiniert mit einem Fragenkatalog, der in Form eines Wizards (einer Folge von einfachen Dialogen, die eine komplexere Aufgabe lösen) bei der Findung des geeignetsten Projekttyps assistiert. Die gegebenen Fragen und ihre Auswirkungen stellen sich nicht den Anspruch der unbedingten Korrektheit sondern sollen lediglich in eine grobe Richtung zeigen, an der sich der Benutzer orientieren kann. Diese Entscheidung wurde mit dem Kunden abgesprochen.

Durch die gewählte Lösung wird also das Tailoring häufig vorkommender Projekttypen unterstützt und ein grober Leitfaden bereitgestellt. Innerhalb dieses Sub-Projektes geschah die Ausarbeitung der Projekttypen und die Anpassung der in ihnen enthaltenen Vorlagen. Die Integration des Ansatzes in die bereits vorhandene webbasierte Prozess-Beschreibung wird im Kapitel 3 beschrieben.

Im Folgenden stelle ich den Entwurf der beiden identifizierten häufigen Projekttypen vor.

Projekttyp Weiterentwicklung

Eine "Weiterentwicklung" ist die Fortführung eines bereits vorhandenen Produkts. Hier ist das primäre Problem bei der Anwendung des Prozesses, dass viele alte Projekte vor Einführung des Prozesses begonnen wurden und dementsprechend einige nun vom Prozess verlangte Erzeugnisse fehlen. Dies sind unter Anderem eine vollständige Anforderungsspezifikation sowie vollständige Entwurfsdokumente. Ein Nacharbeiten dieser Erzeugnisse kommt laut der Qualitätsmanager aus Kosten- und Zeitgründen nicht in Frage. Lediglich die Anforderungsspezifikation wird bei Unvollständigkeit nachgearbeitet.

Die Aufgaben eines vorgetailorten Projekttyps "Weiterentwicklung" wären demzufolge

- Unterstützung in der Nutzung ggf. vorhandener Strukturen wie dem Konfigurationsmanagement
- Verzicht auf das Nacharbeiten nicht vorhandener Erzeugnisse soweit verantwortbar, ausgenommen Anforderungsspezifikation
- Bestehen auf die Dokumentation der im aktuellen Projekt am Produkt durchzuführenden Änderungen

Projekttyp Kleine Änderung

Ein Projekt vom Typ "Kleine Änderung" stellt eine nur geringfügige Anpassung eines bereits vorhandenen Produkts dar. Dies können beispielsweise Fehlerkorrekturen oder leichte Abwandlungen für einen anderen Kunden ("Varianten" genannt) sein. Hier wird meist nur ein einziges Release, das Serienrelease, benötigt. Beim Tailoring zu berücksichtigende Aspekte sind

- Unterstützung in der Nutzung ggf. vorhandener Strukturen wie dem Konfigurationsmanagement
- Zusammenfassung aller Tätigkeiten auf ein einziges Release, hieraus resultierend eine Reduktion des Aufwands für Projektmanagement
- eine dem Projekttyp "Weiterentwicklung" ähnliche Strategie bezüglich des Nacharbeitens nicht vorhandener Erzeugnisse

Verfahren beim Tailoring der Projekttypen

Die in der Roadmap Software vorhandenen Aktivitäten lassen sich in die folgenden Klassen unterteilen:

- Support (unterstützende Maßnahmen, wie beispielsweise Konfigurationsmanagement)
- Engineering (die "eigentliche" Projektdurchführung – d.h. Anforderungserhebung, Entwurf, Implementierung, Test und Integration)
- Management (Planung und Verwaltung)

Aus dieser Einteilung und der vorhergehenden Charakterisierung der beiden Projekttypen "Weiterentwicklung" und "Kleine Änderung" lässt sich folgende Tabelle ableiten:

	<i>Support</i>	<i>Engineering</i>	<i>Projektmanagement</i>
Weiterentwicklung	- Unterstützung vorhandener Strukturen	- Verzicht auf das Nacharbeiten fehlender Erzeugnisse, jedoch Erstellen einer für das Produkt vollständigen Anforderungsspezifikation - Vollständige Dokumentation der am Produkt durchzuführenden Änderungen	- Keine Änderung, volle Unterstützung benötigt
Kleine Änderung	- Unterstützung vorhandener Strukturen	- Verzicht auf das Nacharbeiten fehlender Erzeugnisse, jedoch Erstellen einer für das Produkt vollständigen Anforderungsspezifikation - Vollständige Dokumentation der am Produkt durchzuführenden Änderungen	- Reduktion - Zusammenfassung zu nur einem Release

Tabelle 2: Charakterisierung der Projekttypen

Um diese Einteilung auf die konkreten Vorlagen anzuwenden, d.h. das Tailoring durchzuführen, ist es sinnvoll, die Vorlagen ebenso zu klassifizieren. Dies ist in der folgenden Tabelle geschehen.

<i>Support</i>	<i>Engineering</i>	<i>Projektmanagement</i>
Änderungs-Antrag	Anforderungs-Spezifikation	Prototypen-Freigabe
Inspektionsprotokoll	Anforderungsverfolgung	Prüfplan
KM-Plan	Feinentwurf Komponente	Releaseplan
Qualitätsplan	Feinentwurf Modul	Roadmap Software
	Grobentwurf	
	Test-Report	
	Test-Spezifikation	
	Test-Strategie	

Tabelle 3: Klassifizierung der Dokument-Vorlagen

Die Aufgabe der Implementierung ist es nun, die Forderungen aus Tabelle 2 mit Hilfe der Klassifizierung aus Tabelle 3 auf die konkreten Vorlagen anzuwenden.

Implementierung

Die Dokumentation der konkreten Implementierungen gebe ich im Folgenden in Form von Tabellen an. Zu jeder Vorlage sind die durchgeführten Anpassungen und jeweils die Tailoring-Entscheidung begründende Anmerkungen aufgeführt.

Projekttyp Weiterentwicklung

<i>Support</i>		
Vorlage	Anpassungen	Anmerkungen
Änderungs-Antrag	Keine Anpassungen.	Abweichungen von bereits festgelegten Anforderungen, Entwurfs- oder Implementierungsentscheidungen sollten in jedem Falle korrekt dokumentiert werden (Verfolgbarkeit, Verantwortung gegenüber Kunden)
Inspektionsprotokoll	Keine Anpassungen.	Inspektionen sollten stets gleich dokumentiert werden. Die vorhandene Vorlage erfordert wenig Aufwand und hält doch alles Relevante fest.

KM-Plan	In Kommentaren Hinweis auf Nutzung eines möglicherweise bereits vorhandenen KM-Plans bzw. Speicherortes im Konfigurationsmanagementsystem.	Analog zur Forderung "Unterstützung vorhandener Strukturen"
Qualitätsplan	Angabe des Projekttyps mit Referenz auf diese Auflistung.	Integration des Tailorings aus dem jeweiligen Projekttyp in den Qualitätsplan.
<i>Engineering</i>		
Vorlage	Anpassungen	Anmerkungen
Anforderungs-Spezifikation	Kommentar bezüglich gegebenenfalls notwendiger Nacharbeit hinzugefügt.	Notwendigkeit des Erstellens einer für das Produkt vollständigen Anforderungsspezifikation.
Anforderungsverfolgung	Lediglich zwei leere Arbeitsblätter entfernt.	Einfach gehaltene Tabelle, keine Anpassungen nötig. Leere Arbeitsblätter entfernt um eine mögliche Irritierung seitens der Benutzer zu verhindern.
Feinentwurf Komponente	Kommentar bezüglich gegebenenfalls nicht vorhandener alter Entwurfsentscheidungen hinzugefügt, Verzicht auf Nacharbeit.	Verzicht auf das Nacharbeiten fehlender Erzeugnisse, jedoch vollständige Dokumentation der am Produkt durchzuführenden Änderungen
Feinentwurf Modul	Kommentar bezüglich gegebenenfalls nicht vorhandener alter Entwurfsentscheidungen hinzugefügt, Verzicht auf Nacharbeit.	Verzicht auf das Nacharbeiten fehlender Erzeugnisse, jedoch vollständige Dokumentation der am Produkt durchzuführenden Änderungen
Grobentwurf	Kommentar bezüglich gegebenenfalls nicht vorhandener alter Entwurfsentscheidungen hinzugefügt, Verzicht auf Nacharbeit.	Verzicht auf das Nacharbeiten fehlender Erzeugnisse, jedoch vollständige Dokumentation der am Produkt durchzuführenden Änderungen

Test-Report	Keine Anpassungen.	Einfach gehaltene Vorlage, um Testergebnisse festzuhalten. Alle genannten Informationen werden benötigt. Format sollte möglichst konsistent sein (Wiedererkennungswert von Projekt zu Projekt).
Test-Spezifikation	Keine Anpassungen.	Vorlage macht keine Aussage über den Umfang der Testfälle oder der zu testenden Anforderungen, ist daher als Formular zum Ausfüllen zu verstehen – Testfälle sind aus Test-Strategie und Anforderungen zu entwickeln.
Test-Strategie	Keine Anpassungen.	Tests sind stets zur Gänze durchzuführen.
Projektmanagement		
Vorlage	Anpassungen	Anmerkungen
Prototypen-Freigabe	Keine Anpassungen.	Offizielles Freigabe-Dokument. Anpassungen weder notwendig noch angemessen.
Prüfplan	Keine Anpassungen.	Testumfang hat gleich zu bleiben.
Releaseplan	Lediglich ein leeres Arbeitsblatt entfernt.	Einfach gehaltene Tabelle, keine Anpassungen nötig. Leeres Arbeitsblatt entfernt um eine mögliche Irritierung seitens der Benutzer zu verhindern.
Roadmap Software	Diverse Formulierungen geändert. Betrifft Aktivitäten, bei denen Erzeugnisse gegebenenfalls bereits vorhanden sind bzw. Nacharbeit ausgeschlossen oder forciert werden soll.	Analog zu den Forderungen an die beiden Bereiche "Support" und "Engineering".

Tabelle 4: Implementierung des Projekttyps "Weiterentwicklung"

Projekttyp Kleine Änderung

Support

Vorlage	Änderung	Anmerkungen
Änderungs-Antrag	Keine Anpassungen.	Abweichungen von bereits festgelegten Anforderungen, Entwurfs- oder Implementierungsentscheidungen sollten in jedem Falle korrekt dokumentiert werden (Verfolgbarkeit, Verantwortung gegenüber Kunden)
Inspektionsprotokoll	Keine Anpassungen.	Inspektionen sollten stets gleich dokumentiert werden. Die vorhandene Vorlage erfordert wenig Aufwand und hält doch alles Relevante fest.
KM-Plan	In Kommentaren Hinweis auf Nutzung eines möglicherweise bereits vorhandenen KM-Plans bzw. Speicherortes im Konfigurationsmanagementsystem.	Analog zur Forderung "Unterstützung vorhandener Strukturen"
Qualitätsplan	Angabe des Projekttyps mit Referenz auf diese Auflistung. Reduktion der Phasen-Reviews auf ein einziges nach der einzigen Phase "Serienrelease".	Integration des Tailorings aus dem jeweiligen Projekttyp in den Qualitätsplan. Anpassung der Phasen-Reviews laut Forderung "Zusammenfassung zu nur einem Release"
<i>Engineering</i>		
Vorlage	Änderung	Anmerkungen
Anforderungs-Spezifikation	Kommentar bezüglich gegebenenfalls notwendiger Nacharbeit hinzugefügt.	Notwendigkeit des Erstellens einer für das Produkt vollständigen Anforderungsspezifikation.
Anforderungsverfolgung	Lediglich zwei leere Arbeitsblätter entfernt.	Einfach gehaltene Tabelle, keine Anpassungen nötig. Leere Arbeitsblätter entfernt um eine mögliche Irritierung seitens der Benutzer zu verhindern.

Leif Singer: Integration von Tailoring in eine webbasierte Prozess-Beschreibung

Feinentwurf Komponente	Kommentar bezüglich gegebenenfalls nicht vorhandener alter Entwurfsentscheidungen hinzugefügt, Verzicht auf Nacharbeit.	Verzicht auf das Nacharbeiten fehlender Erzeugnisse, jedoch vollständige Dokumentation der am Produkt durchzuführenden Änderungen
Feinentwurf Modul	Kommentar bezüglich gegebenenfalls nicht vorhandener alter Entwurfsentscheidungen hinzugefügt, Verzicht auf Nacharbeit.	Verzicht auf das Nacharbeiten fehlender Erzeugnisse, jedoch vollständige Dokumentation der am Produkt durchzuführenden Änderungen
Grobentwurf	Kommentar bezüglich gegebenenfalls nicht vorhandener alter Entwurfsentscheidungen hinzugefügt, Verzicht auf Nacharbeit.	Verzicht auf das Nacharbeiten fehlender Erzeugnisse, jedoch vollständige Dokumentation der am Produkt durchzuführenden Änderungen
Test-Report	Keine Anpassungen.	Einfach gehaltene Vorlage, um Testergebnisse festzuhalten. Alle genannten Informationen werden benötigt. Format sollte möglichst konsistent sein (Wiedererkennungswert von Projekt zu Projekt).
Test-Spezifikation	Keine Anpassungen.	Vorlage macht keine Aussage über den Umfang der Testfälle oder der zu testenden Anforderungen, ist daher als Formular zum Ausfüllen zu verstehen – Testfälle sind aus Test-Strategie und Anforderungen zu entwickeln.
Test-Strategie	Keine Anpassungen.	Tests sind stets zur Gänze durchzuführen.
<i>Projektmanagement</i>		
Vorlage	Änderung	Anmerkungen
Prototypen-Freigabe	Keine Anpassungen.	Offizielles Freigabe-Dokument. Anpassungen weder notwendig noch angemessen.

Prüfplan	Zusammenfassung zu einem Release, dem Serienrelease.	Analog zur Forderung "Zusammenfassung zu nur einem Release". Testumfang hat gleich zu bleiben.
Releaseplan	Zusammenfassung zu einem Release, dem Serienrelease. Ein leeres Arbeitsblatt entfernt.	Analog zur Forderung "Zusammenfassung zu nur einem Release". Leeres Arbeitsblatt entfernt um eine mögliche Irritierung seitens der Benutzer zu verhindern.
Roadmap Software	Zusammenfassung zu Phase "Serienrelease". Nur noch ein Phasen-Review. Auslassung der Patent-Analyse. Teilweise Umformulierungen.	Analog zu den Forderungen "Reduktion" und "Zusammenfassung zu nur einem Release". Basierend auf der Roadmap "Minor Updates", entwickelt vom zentralen QM.

Tabelle 5: Implementierung des Projekttyps "Kleine Änderung"

Tests

Traditionelle Software-Tests sind für die hier bearbeiteten Dokument-Vorlagen nur wenig sinnvoll. Es bestünde die Möglichkeit, die formalen Eigenschaften der Dokumente wie Strukturierung und verwendete Stile automatisiert zu testen. Diese Variante schien mir jedoch im Rahmen dieses Projekts nicht geeignet, da der Aufwand erheblich größer als der Nutzen gewesen wäre. Aus diesem Grund habe ich mich in Absprache mit dem Kunden auf die zwei folgenden Tests beschränkt.

Nach abgeschlossener Bearbeitung der Vorlagen prüfte ich in meiner Rolle als Entwickler zunächst die Form, Rechtschreibung und Grammatik. In der Folge überprüfte ich die getroffenen Tailoring-Entscheidung nochmals auf Basis des bereits beschriebenen Entwurfs.

Nach abgeschlossener Prüfung meinerseits übergab ich die getailorten Vorlagen dem Kunden, der sie seinerseits einer formlosen Inspektion unterzog.

3 Sub-Projekt "SPIN Erweiterung"

SPIN ist die bei WABCO bereits vorhandene webbasierte Prozess-Beschreibung. Diese sollte im Rahmen dieses Sub-Projektes derart erweitert werden, dass sie die im Sub-Projekt "Tailoring" entwickelten Projekttypen unterstützt. Hierzu führte ich zunächst eine Analyse des vorhandenen Systems durch und leitete dann aus dieser Analyse, den Ergebnissen des Sub-Projekts "Tailoring" und weiteren Wünschen des Kunden die Anforderungen her. Ich entschied mich für einen Entwurf, implementierte die Änderungen und testete sie.

Vorstellung des Systems

SPIN ist als Webanwendung im WABCO Intranet verfügbar und stellt eine strukturelle, textuelle und graphische Aufbereitung der Roadmap Software dar.

Über eine vereinfachte graphische Repräsentation des Subprozesses Software gelangt der Nutzer für verschiedene Tätigkeitsbereiche wie Anforderungsanalyse, Entwurf, Qualitätssicherung, Test usw. auf jeweils detaillierte Beschreibungsseiten. Hier findet sich dann eine Liste von Inputs, Outputs, Tätigkeiten und Dokumenten, die für den jeweiligen Bereich relevant sind.

Des weiteren bietet *SPIN* Informationen zu Definitionen und Festlegungen bezüglich Rollen, Methoden und Werkzeugen sowie eine Übersicht über alle im Software-Entwicklungsprozess vorkommenden Verfahrens- bzw.- Arbeitsanweisungen sowie Vorlagen.

Um Kritik und Anregungen besser annehmen zu können und eine Diskussion des Prozesses zu fördern wurde ein Forum integriert. Sämtliche Inhalte sind über einen Verwaltungsbereich editierbar.

SPIN wurde im Rahmen einer Bachelorarbeit von einer Studentin der Fachhochschule Münster entwickelt. Die Arbeiten am System waren etwa einen Monat vor Ende meiner Präsenz bei WABCO abgeschlossen.

Webbasierte Informationssysteme

Um die Beschreibung der Entwicklung sowie meine Abwägungen dem Leser verständlicher zu gestalten möchte ich zunächst ein paar allgemeine Worte über webbasierte Informationssysteme und in ihnen übliche und bewährte Konzepte in einer vereinfachten Weise vorstellen.

HTTP-Anfragen und -Antworten

Die Entwicklung und Ausführung von Webanwendungen wird bestimmt durch die Eigenheiten des HTTP, dem Hypertext Transfer Protocol. Durch seine Zustandslosigkeit zerstückelt es die Ausführung der Anwendung in Paare aus Anfragen des Benutzers und Antworten des Webservers. HTTP basiert auf einer Client-Server-Architektur – Clients, meistens Webbrowser, senden also Anfragen an Server und erhalten daraufhin Antworten. Alle Anfragen und Antworten sind unabhängig von gegebenenfalls zu einem früheren Zeitpunkt versandten Anfragen oder Antworten.

In der Anfrage fordert der Client meist eine auf dem Webserver liegende Datei an, optional versehen mit Parametern (GET-Methode) oder eigenen Daten (POST-Methode). Der Webserver antwortet mit der Rücksendung der angeforderten Datei oder einem Fehler. Diese Dateien können jegliches Format aufweisen, interessant sind in diesem Zusammenhang jedoch die eigentlichen Webseiten, also HTML-Dateien. HTML (Hypertext Markup Language) ist eine Auszeichnungssprache für Hypertext, also verlinkten Text, die in ihren verschiedenen Ausprägungen die Standardsprache für die Beschreibung von Webseiten ist. Vom w3c (World Wide Web Consortium), einem Zusammenschluss von verschiedenen am Fortschritt des World Wide Web interessierten Organisationen, wurden und werden verschiedene Definitionen (DTDs, Document Type Definitions) der HTML als Empfehlungen veröffentlicht.

Um aber wirkliche Anwendungen über das WWW zu ermöglichen, reichen statische HTML-Dateien nicht aus. HTML ist nur eine Auszeichnungssprache, also nur für die semantische Strukturierung von Text gedacht und konsequenterweise auch nicht Turing-vollständig. Für die Abbildung wirklicher Anwendungen auf das Web ist daher eine wirkliche Programmiersprache notwendig. Das in ihr geschriebene und vom Client angefragte Programm erhält üblicherweise vom Webserver die Daten der Anfrage und generiert dann eine Antwortseite, die sie an den Webserver zurückgibt. Dieser sendet sie seinerseits an den anfragenden Client zurück.

Das "Front Controller" Design Pattern

Da dieses Vorgehen für komplexere Anwendungen schnell zu unübersichtlich wird hat sich der so genannte *Front Controller* als strukturgebendes Design Pattern durchgesetzt. In Anwendungen dieser Architektur wird jede Anfrage vom Webserver an ein zentrales Teil-Programm der Anwendung übergeben. Dieses führt zunächst Arbeiten aus, die für jede Antwortseite zu erledigen sind und übergibt dann die eigentlich auszuführende Aufgabe an das zuständige Teil-Programm. Hierdurch wird sichergestellt, dass Code nicht mehrfach gehalten wird, da alle bei jeder Anfrage auszuführenden Aufgaben im *Front Controller* gehalten werden und die eigentlichen Webseiten nur noch diejenige Aufgabe übernehmen, auf die sie spezialisiert sind. Änderungen, die die gesamte Anwendung betreffen, sind auf diese Art sehr viel einfacher umzusetzen. Des weiteren verbessert die Spezialisierung der Webseiten die Überschaubarkeit des Systems.

In modernen Webanwendungen ist dieses Design Pattern in der einen oder anderen Form sehr häufig anzutreffen. Um eine wirklich wartbare Anwendung zu erhalten, wird in großen Projekten aber der *Front Controller* mit dem *Model-View-Controller* Design Pattern kombiniert.

Model-View-Controller in Webanwendungen

Das klassische *Model-View-Controller* Design Pattern trennt in Anwendungen die Benutzeroberfläche (den View) von der Geschäftslogik und der Datenhaltung (dem Model) und verbindet sie durch einen Controller. Hierbei gibt es für jeden View einen Controller, der die vom View ausgelösten Events (Benutzeraktionen) in für das Model verständliche Befehle übersetzt. Auf diese Weise können Model und View leichter ausgetauscht werden und die Testbarkeit des Models verbessert sich im Rahmen von Modultests erheblich. In der so genannten "*MVC Model 2*"-Architektur wird statt der einzelnen Controller für alle Views ein

einzig, ein *Front Controller* verwendet. Dies wird den besonderen Anforderungen von Webanwendungen gerecht, die für jede Anfrage bestimmte Aktionen immer durchführen müssen, wie beispielsweise Benutzer-Authentifizierung oder das Einbinden einer Template-Engine. Zusätzlich erhält man die verbesserte Austauschbarkeit und Testbarkeit des Modells durch die Kombination mit dem *Model-View-Controller* Design Pattern.

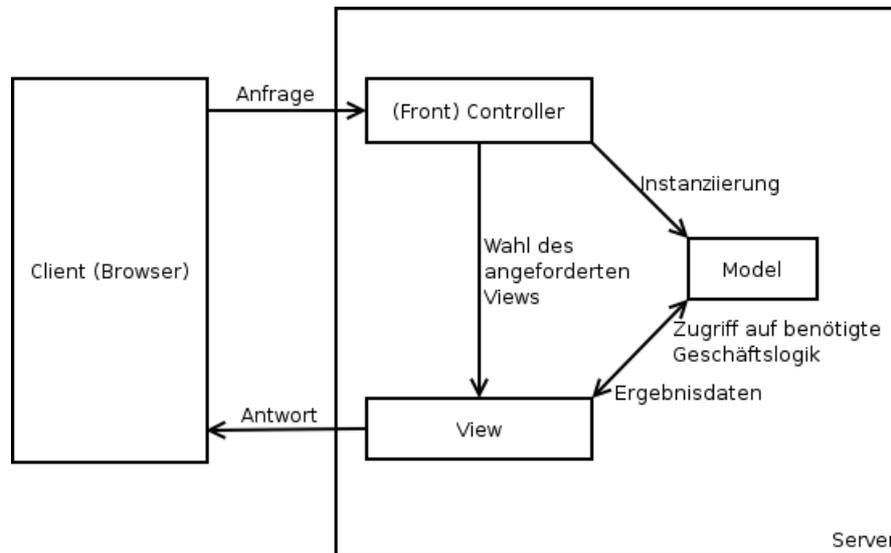


Abbildung 3: Funktionsweise von "MVC Model 2"-Architekturen

Analyse des vorhandenen Systems

Die vorhandene Anwendung ist in der Skriptsprache PHP geschrieben, die als Modul in einem Apache Webserver läuft. Die Datenhaltung ist komplett im Dateisystem gehalten, eine Datenbank wird nicht verwendet.

SPIN bietet eine graphische Übersicht über die Roadmap Software, die mit Erläuterungen zu den einzelnen Phasen verlinkt ist. Darüber hinaus gibt es einige zusätzliche Webseiten, die beispielsweise alle vorhandenen Dokumente in einer Übersicht darstellen, über die bei WABCO gängige Tools informieren oder Ansprechpartner für bestimmte Methoden (wie etwa den Verwalter des Konfigurationsmanagements) nennen. Zusätzlich gibt es die so genannte "Bestandspflege"; dies ist ein passwortgeschützter Verwaltungsbereich, über den die Inhalte des Systems bearbeitet werden können.

Im Folgenden beschreibe ich den Ablauf einer Anfrage im vorhandenen *SPIN*-System und bringe diesen dann mit den soeben vorgestellten Entwurfskonzepten in Verbindung.

Einstieg

Der initiale Aufruf des Systems verläuft bei jeder Anfrage über die Datei `index.php`. Hier wird die Sessionverwaltung initialisiert, Logging-Aufgaben abgehandelt und die Benutzerauthentifizierung durchgeführt. Des Weiteren werden alle im System vorhandenen Programmdateien eingebunden – sie befinden sich in den Verzeichnissen "modules" und "processes". Zum Abschluss wird eine Instanz der Klasse `Theme` erstellt, deren "display"-Methode aufgerufen und die durch diese generierten Inhalte ausgegeben.

Die Klasse Theme

Die Klasse Theme enthält eine Vielzahl von Hilfsfunktionen für die Ausgabe von HTML-Elementen und die Darstellung der Systembestandteile. Diese sind in einen Seitenkopf und -fuss, ein linkes und ein rechts Seitenelement sowie einen mittigen Bereich für die Inhalte aufgeteilt.

Bei Aufruf der Methode "display" werden zunächst der Seitenkopf und das linke Seitenelement ausgegeben. Das linke Seitenelement enthält das Menü.

Nun unterscheidet das System zwischen "Modulen" und "Prozessen". "Module" sind Systemteile, die eine bestimmte Funktionalität enthalten – sie sind mit den einzelnen Webseiten gleichzusetzen. "Prozesse" könnten als ein "Modul" zur Darstellung der einzelnen Phasen des Entwicklungsprozesses angesehen werden.

Wird in den Parametern des Aufrufs ein "Modul" übergeben, so wird eine Klasse mit dem Namen des "Moduls" instanziiert und deren "display"-Methode aufgerufen. Andernfalls wird eine "Prozess"-Klasse mit dem in den Parametern übergebenen Prozessnamen gesucht, instanziiert und wiederum deren "display"-Methode aufgerufen.

Am Ende der "display"-Methode werden das rechte Seitenelement und der Seitenfuss ausgegeben.

Bei Instanziierung der Klasse Theme wird die Datei constants.php eingefügt, in der die für die geforderte Unterstützung von mehreren natürlichen Sprachen benötigten Konstanten definiert sind. Hier wird auch überprüft, welche Sprache aktuell ausgewählt ist, welche Konstanten also aktuell gültig zu sein haben.

Die "Modul"- und "Prozess"-Klassen

In den jeweils aus der Klasse Theme aufgerufenen "display"-Methoden der "Modul"- und "Prozess"-Klassen steckt sowohl Geschäftslogik als auch die Generierung des HTML für die Benutzerschnittstelle. Teilweise sind Bereiche in eigene Funktionen ausgelagert, in denen aber auch Geschäftslogik und Benutzerschnittstelle nicht voneinander getrennt werden. Eine gemeinsame Schnittstelle beispielsweise in Form einer gemeinsamen Eltern-Klasse ist nicht vorhanden. Um Klassen, die einem "Modul" im Sinne des SPIN Systems entsprechen und eigentliche Software-Module auseinanderzuhalten referenziere ich "Modul"-Klassen stets mit dem Wort "Modul" in Anführungsstrichen.

Im vorhandenen System wird also das *Front Controller Design Pattern* in Form der Datei index.php verwendet. Eine Trennung von Geschäftslogik und Benutzerschnittstelle erfolgt jedoch nicht, *MVC Model 2* kommt also nicht zum Einsatz.

Wahl und Tailoring des eigenen Prozesses

Auch in diesem Projekt wurde die Roadmap Software des CPDPs verwendet. Dies entsprach auch hier den Kundenvorgaben. Mein Betreuer war mir wie auch im Projekt "Tailoring" beim Anpassen des Prozesses hilfreich.

Die Anforderungen an den Prozess waren sehr ähnlich – ein relativ kurzes, alleinstehendes Software-Projekt, das kein offizielles WABCO-Projekt ist. Daher wurde auch hier die Phase "Stabilisierung" entfernt sowie für nur ein einziges Release geplant. Der Unterschied bestand lediglich im erzeugten Produkt, so dass die Tests zunächst etwas umfangreicher gehalten wurden. Auch für dieses Projekt wurde nach Abschluss des Serienreleases auf eine Validierungs-Phase verzichtet.

Genauere Informationen und Begründungen hierzu sind in der "Remark"-Spalte des im Anhang beigefügten Auszugs der zu diesem Sub-Projekt gehörigen Roadmap zu entnehmen – hierbei handelt es sich um die Phase "Serienrelease". Getailorte Aktivitäten sind diejenigen, die mit einem "na"-Status ("not applicable", Englisch für "nicht anwendbar") und einem Kommentar versehen sind.

Anforderungserhebung

Die Phase der Anforderungserhebung für dieses Sub-Projekt setzt sich zusammen aus der Analyse des vorhandenen Systems, Gesprächen mit dem Kunden sowie der Anforderungserhebung und dem Entwurf des Sub-Projekts "Tailoring".

Das bereits bei WABCO vorhandene System sowie die im Rahmen seiner Entwicklung entstandene Anforderungsspezifikation können als Funktionsgrundlage für das zu entwickelnde System betrachtet werden. In der Anforderungsspezifikation für dieses Sub-Projekt wurde dies als die funktionale Anforderung "kein Funktionalitätsverlust" mit einer Referenz auf die bereits vorhandene Anforderungsspezifikation vermerkt. Zum besseren Verständnis liste ich an dieser Stelle diese Anforderungen auf:

- das System unterstützt mehrere natürliche Sprachen
- das System wird auf gängigen Browsern möglichst ähnlich dargestellt
- es gibt eine graphische Darstellung des Entwicklungsprozesses für Software
- es gibt einen festen Seitenrahmen und ein stehenbleibendes Menü
- es gibt eine Fussleiste mit Statusinformationen
- der vom Benutzer gewählte Navigationspfad wird angezeigt
- es gibt eine Zusammenfassung der Arbeitsanweisungen
- für die Prozessphasen gibt es eine Darstellung mit Inputs, Tätigkeiten und Outputs
- es gibt eine Übersicht über die an der Entwicklung beteiligten Rollen und deren Definitionen
- es gibt eine Übersicht der letzten Änderungen an den Arbeitsanweisungen und sonstigen im System enthaltenen Dokumenten
- für die Prozessphasen gibt es Links mit weiterführenden Dokumenten
- es gibt eine einfache Möglichkeit, die Inhalte zu warten
- die Entwicklerdokumentation ist in das System eingebunden
- es gibt eine Übersicht über die zu verwendenden Tools

Durch die Anforderungserhebung und den Entwurf im Sub-Projekt "Tailoring" ergaben sich weitere Anforderungen. Ich entschied mich dort für die Implementierung von Projekttypen und einen Fragenkatalog als Hilfe zur Entscheidungsfindung. Hieraus lassen sich die folgenden Anforderungen ableiten.

Im Folgenden wird der Begriff *Verwalter* im Sinne eines für die Verwaltung der Inhalte des SPIN-Systems authentifizierten Benutzers verwendet.

- werden Prozessdokumente verlinkt, so sind diejenigen des jeweils ausgewählten Projekttyps zu verwenden
- der Benutzer hat die Möglichkeit, den Projekttyp frei zu wählen
- der Benutzer hat die Möglichkeit, mit Hilfe eines Wizards durch die Beantwortung von in einem Fragenkatalog hinterlegten Fragen einen Hinweis auf den wahrscheinlich vorzugsweise zu verwendenden Projekttypen zu erhalten
- zu den in der Änderungsliste eingetragenen Änderungen ist der jeweils betroffene Projekttyp vermerkt
- ein Verwalter hat die Möglichkeit, neue Projekttypen anzulegen
- ein Verwalter hat die Möglichkeit, vorhandene Projekttypen umzubenennen
- ein Verwalter hat die Möglichkeit, vorhandene Projekttypen zu entfernen
- bei der Bearbeitung der Prozesse und Prozessbeschreibungen geschehen die Änderungen an den Prozessen und Prozessbeschreibungen eines vorher ausgewählten Projekttyps
- bei der Bearbeitung der Prozessdokumente geschehen die Änderungen an den Prozessdokumenten eines vorher ausgewählten Projekttyps
- ein authentifizierter Benutzer hat die Möglichkeit, den vom Wizard verwendeten Fragenkatalog zu bearbeiten

Dem Kunden war es wichtig, dass die einmal getroffene Auswahl eines Projekttyps möglichst nicht-flüchtig ist. Dies führte mich zu der Anforderung,

- den gewählten Projekttyp clientseitig zu speichern (dies wurde später durch einen Cookie implementiert – dies ist eine von einer Webseite über den Browser im Dateisystem des Benutzers gespeicherte und wieder auslesbare Datei)

Entwurf: Übernahme des Vorhandenen

Ich entschloss mich dazu, den bereits vorhandenen Entwurf des SPIN Systems beizubehalten. Zwar wäre es besonders für einfacher durchführbare Modultests vorteilhafter gewesen, die Geschäftslogik von der Benutzerschnittstelle zu trennen. Ein entsprechender Entwurf hätte jedoch die Entwicklung eines komplett neuen Systems erfordert. Der hierfür erforderliche Aufwand wäre meiner Einschätzung nach jedoch nicht mit dem Zeitrahmen dieser Arbeit vereinbar gewesen, da ich die Hälfte meiner Ressourcen für das Sub-Projekt "Tailoring" benötigte. Auch der Kunde zog eine einfache Erweiterung des vorhandenen Systems vor.

Die neue Funktionalität sollte also in die bestehende Struktur eingepasst werden, dies tat ich am Beispiel bereits vorhandener Systemteile. Die Prozessdokumente waren bisher in dem Verzeichnis "processes" und einem dem jeweiligen Prozess entsprechenden Unterverzeichnis gespeichert. Zur Unterstützung von Projekttypen erstellte ich eine weitere Verzeichnisebene

im Verzeichnis "processes". Die Prozessdokumente des Prozesses "Configuration Management" für den Projekttyp "Kleine Änderung" sind nun beispielsweise im Verzeichnis "processes/minor_change/Configuration_Management" abgelegt.

Implementierung

Die Implementierung teilte sich in solche Funktionalitäten, die komplett neu hinzukamen und solche, die durch die Anpassung vorhandenen Quellcodes zustand kamen.

Neue Funktionalitäten

Tailoring

Um den Projekttyp frei wählen zu können wurde das neue "Modul" *Tailoring* entwickelt. Die im System vorhandenen Projekttypen werden hier als Links aufgelistet. Die Auswahl eines der Links speichert den betroffenen Projekttypen in einem Cookie.

Wizard

Als Entscheidungshilfe bei der Wahl des passenden Projekttyps wurde der *Wizard* entwickelt. Der Benutzer beantwortet sequentiell die im System hinterlegten Fragen. Diese Fragen sind in einem definierten Format im Dateisystem hinterlegt. Nach der Beantwortung einer Frage berechnet der Wizard auf Basis der zur Frage hinterlegten Daten eine neue Gewichtung der Projekttypen in Form einer Punktzahl. Nachdem alle Fragen beantwortet wurden gibt der Wizard diejenigen Projekttypen als Empfehlung aus, die die meisten Punkte erhalten haben.

Der von mir hinterlegte Fragenkatalog enthält drei Fragen, die auf die im Sub-Projekt "Tailoring" entwickelten Projekttypen ausgelegt sind. Sie stellen einen Ansatzpunkt dar und sollten mit der Zeit aufgrund von mit dem Wizard gemachten Erfahrungen erweitert werden.

Bestandspflege: Projekttypen

Um Projekttypen hinzufügen, umbenennen und entfernen zu können wurde das neue "Modul" *Projekttypen* entwickelt. Es verwaltet die im Dateisystem abgelegten Order, die die Prozessdokumente enthalten sowie je Projekttyp eine Textdatei, in der die für die Mehrsprachigkeit benötigten Übersetzungen gehalten werden.

Bestandspflege: Fragenkatalog Wizard

Zur Verwaltung des Fragenkatalogs für den Wizard wurde das "Modul" *Fragenkatalog Wizard* entwickelt. Es erlaubt das Bearbeiten der im Dateisystem hinterlegten Fragen, ihren Übersetzungen und ihren Auswirkungen auf die Gewichtung der Projekttypen.

Anpassungen

Prozess anzeigen

Bei der Anzeige eines Prozesses werden die ihn begleitenden Prozessdokumente dargestellt. Durch die Anpassung werden nun die entsprechenden Prozessdokumente des im Cookie gespeicherten Projekttyps angezeigt.

Dokumente

Das "Modul" *Dokumente* ist eine Übersicht über alle im Software-Entwicklungsprozess vorhandenen Prozessdokumente wie Vorlagen, Arbeitsanweisungen und Checklisten. Durch die Anpassung werden nun die entsprechenden Prozessdokumente des im Cookie gespeicherten Projekttyps angezeigt.

Änderungen

Das "Modul" *Änderungen* stellt eine Liste von an Dokumenten vorgenommenen Änderungen dar. Zu jedem Eintrag wird nun auch der betroffene Projekttyp angezeigt.

Prozesse bearbeiten

Die Bearbeitung von Prozessen arbeitet nun in dem Verzeichnis eines vorher vom Verwalter gewählten Projekttyps.

Dokumente bearbeiten

Die Bearbeitung von Dokumenten arbeitet nun in dem Verzeichnis eines vorher vom Verwalter gewählten Projekttyps.

Änderungen verwalten

Beim Einfügen eines neuen Eintrags in die Änderungsliste durch den Verwalter muss von nun an der Projekttyp festgehalten werden, in dem sich das geänderte Dokument befindet. Vor die Auswahl des geänderten Dokuments wurde daher eine Abfrage nach dem gewünschten Projekttyp gesetzt. Um diesen zu speichern wurde das Format der Änderungsliste angepasst.

Zusammenführung der beiden Sub-Projekte

Nachdem das SPIN System derart erweitert worden war war es ein Leichtes, die Erzeugnisse aus dem Sub-Projekt "Tailoring" als Datenbasis dieses Sub-Projektes zu verwenden. Die Integration der beiden Sub-Projekte war damit abgeschlossen.

Tests

Bevor ich die Durchführung der Tests am erweiterten SPIN-System beschreibe, möchte ich zunächst einige übliche Möglichkeiten beim Testen von Webanwendungen vorstellen.

Testen von Webanwendungen

Beim Testen von Webanwendungen sind zwei Kategorien von Tests zu unterscheiden. Zum Einen gibt es Stress-Tests und Ähnliches um zu überprüfen, ob Server und Anwendung mit großen Benutzerzahlen zurechtkommen. Diese Tests sind aber in diesem Zusammenhang wenig interessant, da die Zielgruppe der Anwendung (Qualitätsmanager, interessierte Software-Entwickler) aus weniger als 200 Benutzern besteht – eine solche Benutzerzahl ist im Regelfall leicht durch den Webserver zu bedienen. Zum Anderen gibt es die funktionalen Tests, welche überprüfen, ob jegliche Funktionalität korrekt implementiert wurde. Tests dieser Kategorie sind für dieses Projekt schon weitaus interessanter. Aber auch hier gilt es, verschiedene Vorgehensweise zu unterscheiden.

Wie beim klassischen Software-Test besteht auch bei Webanwendungen natürlich die Möglichkeit, Modultests durchzuführen. Im Optimalfall sind Geschäftslogik und Benutzeroberfläche voneinander getrennt, so dass wirklich nur die reine Funktionalität getestet werden kann.

Eine andere Möglichkeit sind durch Skripte automatisierte Clients, die wirkliche Anfragen an den Server senden und die erhaltene Antwort interpretieren. Dies ist allerdings eher aufwendig und lohnt sich erst bei umfangreicheren Anwendungen. Schließlich müssen die notwendigen Skripte geschrieben und für einen reibungslosen Ablauf auch die notwendigen Werkzeuge vorhanden und am Besten bereits eingeführt sein.

Natürlich ist es auch möglich, die Funktionalität manuell zu testen, indem man mit einem Browser die Webanwendung benutzt und so verschiedene Testfälle durchspielt. Man vergleicht dann manuell die Antwort des Servers mit den im Testfall festgelegten Soll-Werten und entscheidet dann über Bestehen oder Nicht-Bestehen des Testfalls.

Wie bei jeder Software mit Benutzerschnittstelle ist es auch hier möglich, Usability-Tests in Betracht zu ziehen. Diese sind im Regelfall gesondert und von Usability-Experten durchzuführen. Da diese aber oftmals in einer Organisation nicht verfügbar sind bietet es sich im Zusammenhang mit manuellen Tests an, die Webanwendung auch nach Usability-Kriterien zu bewerten bzw. gleich mitzutesten.

Abhängigkeiten vom Entwurf

Ich entschloss mich beim Testen des erweiterten SPIN Systems für manuell durchgeführte funktionale Tests. Da im Entwurf die Geschäftslogik und die Ausgabe der Benutzeroberfläche vereint sind schienen mir Modultests nur wenig geeignet zu sein, da sie auch das ausgegebene HTML zeichengenau hätten prüfen müssen. Geskriptete Benutzertests schienen mir in Relation zum Nutzen in diesem Falle einen zu hohen Aufwand zu erfordern.

Die von mir erstellten Testfälle lehnten sich an die spezifizierten Anforderungen an das System an. Als Grundlage dienten hier beide Anforderungsspezifikationen, d.h. sowohl die bereits vorhandene als auch die im Rahmen dieses Sub-Projekts erstellte.

Auf dieser Basis definierte ich wie vom Prozess vorgegeben Testfälle und erstellte nach erfolgreichem Test einen Test-Report.

4 Zusammenfassung & Ausblick

In dieser Bachelorarbeit habe ich dargelegt, wie ich dem Software-Entwicklungsprozess der Firma WABCO folgend in Zusammenarbeit mit den Qualitätsmanagern einen Tailoring-Ansatz gewählt, implementiert und schließlich in die vorhandene webbasierte Prozess-Beschreibung integriert habe. In diesem Zusammenhang habe ich die hinter diesen Tätigkeiten stehenden Konzepte verdeutlicht und ihre Verbindung zu meiner Tätigkeit und meinen Abwägungen hergestellt.

Bedeutung der Arbeit

Im Anschluss an das abschliessende Phasen-Review zum Serienrelease meines Projekts bei WABCO hielt ich einen kurzen Vortrag über den Verlauf des Projekts. Mein Betreuer bzw. der Kunde und einer der Qualitätsmanager waren anwesend – die anderen Qualitätsmanager waren leider verhindert. Nach meinem Vortrag besprachen wir die Auswirkungen meiner Arbeit auf das Tailoring bei WABCO sowie die diesbezüglichen Pläne des T3 und der Qualitätsmanager.

Wir kamen zu dem Schluss, dass durch diese Arbeit die Infrastruktur geschaffen wurde, um das Tailoring in der Software-Entwicklung bei WABCO voranzubringen. Die Erweiterung des *SPIN*-Systems vereinfacht die Erfahrungsnutzung und somit die Verbesserung und Weiterentwicklung der von mir geschaffenen Datenbasis, die als Ausgangspunkt dient.

Im Verlauf meiner Tätigkeit bei WABCO erhielt ich einen spannenden Einblick in die Prozesslandschaft eines grossen Unternehmens und den dortigen Arbeits-Alltag. Ich führte erstmals eine systematische, von den Regeln der Software-Technik geleitete Erweiterung eines vorhandenen Systems durch. Der recht reibungslose Verlauf nahm mir einen Teil des Unbehagens, den die Übernahme von vorhandener Software zur Weiterentwicklung für mich stets hatte.

Ich setzte mich intensiv mit dem Thema Tailoring auseinander, dessen korrekte Durchführung unter anderem durch das kürzlich (Februar 2005) veröffentlichte V-Modell XT langsam zum selbstverständlichen Bestandteil der Software-Entwicklungsprozesse in der Industrie wird. Ich bin der Meinung, mein Wissen und meine Fähigkeiten um einen wichtigen und interessanten Bereich erweitert zu haben.

Während die Erweiterung des *SPIN*-Systems für potentielle Leser wahrscheinlich weniger interessant sein wird gehe ich davon aus, in dieser Arbeit eine verständliche Einführung in das Tailoring von Entwicklungsprozessen gegeben zu haben. Die gegebene Einsicht in einen in der Industrie angewandten Software-Entwicklungsprozess dieses Ausmaßes dürfte ebenfalls interessant sein.

Bewertung der Durchführung

Ich muss zugeben, dass ich vor Abschluss des Projektes das Gefühl hatte, "nicht genug getan" zu haben. Denn das von mir vorgenommene Tailoring an den Prozessdokumenten beschränkte sich in der praktischen Ausführung auf Änderungen einiger Formulierungen, das Hinzufügen von Kommentaren und einige Zusammenfassungen.

Die positive Rückmeldung nach Projektende stimmte mich jedoch um. Um einen SPICE-konformen Prozess zu erhalten müssen bestimmte Voraussetzungen gegeben sein – und diese durch Tailoring wieder zu entfernen wäre ein Schritt in die falsche Richtung gewesen.

Trotz dem zufriedenstellenden Ausgang möchte ich den Projektverlauf kritisch betrachten. Meine Anwesenheit bei WABCO war ursprünglich nur für die Monate Juni bis August geplant, fand aber erst am 28. September seinen Abschluss. Diese Entwicklung war nicht überraschend, sondern war bereits nach etwa der Hälfte der ursprünglich geplanten Projektzeit absehbar. Um das Projekt dennoch korrekt abschliessen zu können nahm ich diese Verlängerung in Kauf.

Den Grund für die Verlängerung meiner Tätigkeit bei WABCO vermute ich in zu ausgedehnten Phasen der Anforderungserhebung und des Entwurfs im Sub-Projekt Tailoring. Ich bin mir zwar im Klaren darüber, dass sorgfältig erhobene Anforderungen und ein guter Entwurf integrale Bestandteile einer reibungslosen Implementierung sind. Dennoch verbrachte ich in diesen Phasen meines Erachtens zu viel Zeit damit zu versuchen, die verschiedenen Vorstellungen der Qualitätsmanager und meine Erwartungen an den Umfang des Tailorings zu vereinen. Schlussendlich entschloss ich mich dazu, nicht die scheinbar perfekte, sondern die praktikabelste Lösung zu suchen – was genau die vom Kunden gewünschten Ergebnisse hervorbrachte.

Ich vermute, dass dieses Problem durch das Fehlen von Erfahrungen im Tailoring sowie dem Umgang mit derlei aufwendigen Entwicklungsprozessen meinerseits entstand und dadurch bei mir eine leichte Desorientierung eintrat. Im Rahmen der erfahrungsbasierten Prozessverbesserung bin ich aber durchaus optimistisch, hieraus nützliche Erfahrungen und Lehren für meinen persönlichen Umgang mit Entwicklungsprojekten ziehen zu können.

Ausblick

Die Firma WABCO befindet sich zur Zeit in einer Umbruchsphase. Ein SPICE-konformer Prozess, der CPDP, wurde etwa im Jahr 2001 eingeführt. Jedoch ist es noch nicht allen Projekten möglich, diesem auch vollständig zu folgen. Um ein langsames Hereinwachsen der Projekte in den Prozess zu unterstützen wurde ein individuelles Tailoring des Prozesses als Strategie gewählt.

Der im Rahmen dieser Arbeit vorgestellte Tailoring-Ansatz wird bei WABCO die Basis für die weiteren Vorhaben des T3 bezüglich der projektspezifischen Anpassung des Software-Entwicklungsprozesses sein und kontinuierlich verbessert werden. Solange die Projekte noch nicht alle Forderungen des Prozesses erfüllen können ist jedoch darauf zu achten, dass das Tailoring stets als Anpassung des Prozesses an die *Bedürfnisse* der Projekte verstanden wird – und nicht an deren Unzulänglichkeiten. Diese Problematik ist den Mitarbeitern des T3 bewusst; die Tailoring-Maßnahmen werden diesbezüglich überwacht.

Das nächste zu lösende Problem ist laut Aussage meines Betreuers die Integration der Software-Entwicklung mit den anderen Subprozessen (Systementwicklung, mechanische und elektrische Hardware-Entwicklung, etc.). Diese wenden zwar ebenfalls bereits den CPDP an, jedoch verläuft die Kommunikation von Prozessdokumenten zwischen den Subprozessen noch suboptimal.

Sobald sich die erwähnte Phase des Umbruchs ihrem Ende nähert ist es meiner Einschätzung nach wahrscheinlich sinnvoll, im Rahmen des werkzeuggestützten Tailorings bald einen

anderen, allgemeingültigeren Ansatz einzuführen. Beispielsweise würde die Erarbeitung von wirklichen Tailoring-Regeln wohl deutlich gangbarer. Diese könnten explizit festgehalten werden und nach ihrer Anwendung einen immer noch SPICE-konformen Prozess garantieren – dies ist beim derzeitigen Vertrauen auf die Erfahrungen der Mitarbeiter nicht unbedingt gegeben.

Zur Unterstützung eines solchen neuen Ansatzes wird sich das vorhandene *SPIN*-System wahrscheinlich weniger eignen. Abgesehen von den Inhalten hat *SPIN* nur wenig mit Entwicklungsprozessen zu tun: es könnte auch als System zur Inhalteverwaltung für gänzlich andere Anwendungen dienen. Ich bin der Meinung, dass ein zukünftiges Werkzeug "Domänen-Wissen" in sich tragen sollte, also speziell auf die Bedürfnisse einer Strukturierung, Speicherung, Bearbeitung und Darstellung von Entwicklungsprozessen ausgerichtet sein sollte.

Dies wäre beispielsweise gegeben durch eine Webanwendung, die Zugriff auf den in einer Datenbank gespeicherten Prozess hat. Ein hierfür geeignetes Datenbankmodell ist leicht herzuführen, entspricht ein Prozess doch einem gerichteten Graphen. Dessen Elemente können ebenfalls wieder als Prozesse aufgefasst werden, bis nach einer hinreichenden Aufteilung nur noch atomare Prozesse, d.h. Aktivitäten bleiben. Jeder Prozess könnte mit optionalen Eingangs- und Ausgangserzeugnissen, Verfahrens- und Arbeitsanweisungen, Vorlagen, Checklisten, Rollen, Methoden usw. assoziiert werden.

Gelänge es, Tailoring-Regeln zu entwickeln, die auf dem so hinterlegten Prozess arbeiten können, könnten diese ebenso in die Anwendung integriert werden. Ähnlich wie in dieser Arbeit könnte ein – selbstverständlich weitaus aufwendigerer – Wizard diese Regeln abarbeiten, so dass die Anwendung nach diesem Durchlaufen der Tailoring-Regeln für jedes Projekt einen getailorten Prozess sowie die zugehörigen Dokumente generieren könnte. Für das Ausfüllen dieser Dokumente könnte die Anwendung spezielle Masken bzw. Editoren zur Verfügung stellen, so dass die gesamte Prozessdokumentation noch einheitlicher und dadurch übersichtlicher würde.

Da das WABCO-Intranet zumindest im Werk Hannover durch Terminals für jeden Mitarbeiter verfügbar ist könnte durch eine Integration des Projektmanagements in dieses Werkzeug die Kommunikation zwischen voneinander abhängigen Projekten verbessert werden. Die Teilnehmer eines jeden Sub-Prozesses könnten sich am System anmelden und erhielten Informationen über die Projekte an denen sie beteiligt sind. Hier sind aktuelle Aufgaben, Statusmeldungen aus anderen Sub-Prozessen und -Projekten usw. denkbar. Dies scheint mir insbesondere im Licht der bereits erwähnten mangelnden Kommunikation der Dokumente zwischen den verschiedenen Sub-Prozessen ein interessanter Gedanke zu sein.

Diese Ideen lassen natürlich diverse in der Wirklichkeit vorhandene Einschränkungen und Ausnahmen ausser Acht und sind daher weniger als fertiges Konzept, sondern vielmehr als ein möglicher Denkanstoss für weitere Überlegungen zu betrachten.

Abkürzungen und Glossar

ABS

Anti-Blockier-System

Apache Webserver

<http://httpd.apache.org/>

BU

Business Unit, Produktsegment

CPDP

Common Product Development Process

DTD

Document Type Definition

EBS

Electronic Braking System

HTML

Hypertext Markup Language

HTTP

Hypertext Transfer Protocol, zur Zeit gängige Version 1.1 definiert in RFC 2616

PHP

PHP Hypertext Preprocessor, <http://php.net/>

QM

(Software-)Qualitätsmanager

SPICE

Software Process Improvement and Capability Determination - eine Norm zur Feststellung des Reifegrads eines Prozesses, ähnlich dem Capability Maturity Model

T3

Technologiezentrum 3

tailored, tailoring, tailors

To modify a process, standard or procedure to better match process or product requirements.

W3c

World Wide Web Consortium, <http://w3.org/>

WABCO

Westinghouse Airbrake Company

Wizard

In der Benutzerführung eine Folge von einfachen Dialogen, die den Benutzer eine komplexere Aufgabe leichter lösen lassen.

Abbildungsverzeichnis

Abbildung 1: Prinzipieller Ablauf des Tailorings in der Software-Technik	8
Abbildung 2: Phasenübergänge in der Roadmap Software	10
Abbildung 3: Funktionsweise von "MVC Model 2"-Architekturen	28

Tabellenverzeichnis

Tabelle 1: Sub-Prozesse und ihre Phasen im CPDP, nach [STRUEBER].....	9
Tabelle 2: Charakterisierung der Projekttypen.....	19
Tabelle 3: Klassifizierung der Dokument-Vorlagen.....	20
Tabelle 4: Implementierung des Projekttyps "Weiterentwicklung".....	22
Tabelle 5: Implementierung des Projekttyps "Kleine Änderung".....	25

Quellenverzeichnis

[CMM]

M. Paulk, e.a., Capability Maturity Model for Software, CMU/SEI 93-TR-25, 1993.

[DICT]

The American Heritage® Dictionary of the English Language, Fourth Edition

[STRUEBER]

Beate Strüber, Präsentation "Der Software-Entwicklungsprozess bei WABCO", 2005

[STSC]

Faye C. Budlong, Paul A. Szulewski, and Ralph J. Ganska, Process Tailoring for Software Project Plans, 1996

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst habe und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe.

Hannover, 29. September 2005

Leif Singer