

# PQI – Ein Ansatz zur prozess- und projektorientierten Qualitätsintegration

Timea Illes-Seifert, Frank Wiebelt

EnBW Systeme Infrastruktur Support GmbH  
Business Solutions - Business Consulting & Partner Management  
Durlacher Allee 93  
76131 Karlsruhe

t.illes-seifert@enbw.com  
f.wiebelt@enbw.com

**Abstract:** In vielen großen Unternehmen ist eine heterogene Projektlandschaft vorzufinden, die unterschiedliche Vorgehensmodelle für die Softwareentwicklung erfordert. Gleichzeitig stellt sich die Notwendigkeit einer vergleichbaren Erhebung von Prozess- und Produktkennzahlen. In diesem Beitrag wird ein qualitätsmodellorientierter Ansatz vorgestellt, welcher einerseits ein einheitliches Qualitätsmodell für alle Projekte zugrunde legt und andererseits dessen Verankerung in das Unternehmen durch eine mehrstufige, abstrakte Beschreibung von Prozessen vorsieht. Vorteil des Ansatzes ist, dass einerseits jedes Projekt entsprechend seines Kontextes ein geeignetes Vorgehen aus dem abstrakten Prozessmodell nach vordefinierten Regeln instanziiieren kann. Durch die Verankerung des Qualitätsmodells in die abstrakten Prozessbeschreibungen andererseits, wird die Möglichkeit einer vergleichbaren Erhebung von Kennzahlen geschaffen.

## 1 Einleitung und Ausgangssituation

Der Qualitätsanspruch an die zu entwickelnde oder bereits entwickelte Software im Unternehmen EnBW ist sehr hoch. Jedoch müssen heterogene Geschäftsfelder mit ebenso heterogenen Qualitätsansprüchen bedient werden. Um es an 2 Beispielen kurz aufzuzeigen, gibt es auf der einen Seite hoch komplexe Schnittstellen-Entwicklungen, beispielsweise zur Kraftwerkssteuerung, die ohne menschliche Interaktion, hoch performant, aber auch sehr hohen Sicherheitskriterien gerecht werden müssen. Auf der anderen Seite gibt es Entwicklungen für den Endkundenbereich, um Zählerstände über das Internet zurück zu melden. Hier spielen die Kriterien wie Performanz und Sicherheit im Vergleich zur Usability eine geringere Rolle. Der Endkunde muss sich im letzteren Fall schnell auf der grafischen Benutzeroberfläche zurecht finden, vom System gut geführt werden und passendes Feedback bekommen.

Bedingt durch diese heterogene Projektlandschaft, ergibt sich die Herausforderung, ein Qualitätsmodell zu operationalisieren, das zum einen den unterschiedlichsten Projektspezifika gerecht wird, es aber zum andern auch schafft, vergleichbare qualitätsbezogene Kennzahlen, sogenannte Key Performance Indicators (KPIs) zu erheben. Ähnlich zu den heterogenen Qualitätsansprüchen, gilt es auch die unterschiedlichen Vorgehensmodelle zur Softwareentwicklung bei der Definition und Verankerung des Qualitätsmodells zu berücksichtigen. Auch hier reichen die Bandbreiten von hoch-inkrementellen Vorgehensweisen bei Projekten mit unklaren Anforderungen, bis hin zum klassischen Wasserfallmodell, bei risikoarmen, kleinen Projekten.

Um den beiden vorgestellten Anforderungen gerecht zu werden, wird im Unternehmen EnBW Systeme Infrastruktur Support GmbH ein Ansatz zur „Prozess und projektorientierten Qualitätsintegration“ (PQI) verfolgt. Der Grundgedanke dieses Ansatzes besteht in der Definition eines auf der ISO/IEC 9126 [ISO/IEC01] basierenden Qualitätsmodells. Dieses Modell wird in der Projektlandschaft derart verankert, indem jedes Projekt die definierten Qualitätskriterien in Form von projektspezifischen Qualitätszielen instanziiert und an deren Erreichungsgrad gemessen wird.

Was bedeutet aber „projektorientiert“ in einem Entwicklungsumfeld, in dem von agilen bis hin zu strikten Wasserfall-Methoden, die unterschiedlichsten Vorgehensmodelle vorkommen?

Adenin, Cytosin, Guanin, Thymin bilden die Grundbausteine einer jeder menschlichen DNA. Wenn die Metapher auf die Softwareentwicklung übertragen wird, stellt sich die Frage nach den Grundbausteinen von Vorgehensmodellen zur Software- und Systementwicklung. In vielen großen Unternehmen kann aufgrund der hohen Heterogenität der durchgeführten Projekte nicht nur ein einziges Vorgehensmodell definiert bzw. eingeführt werden, vielmehr sind Koexistenzen mehrerer Problemlösungsstrategien vorhanden. Die Problematik dieser Konstellation besteht in der fehlenden Möglichkeit zur vergleichbaren Erhebung von KPI zur Steuerung der Prozess- und Produktqualität. Das ist auch die Situation bei der EnBW Systeme Infrastruktur Support GmbH. Letztlich muss eine Integration des gewählten Qualitätsmodells in eine firmenweit gültige Prozessbeschreibung geschaffen werden. Um dieser Herausforderung zu begegnen, integriert der PQI-Ansatz die Definition eines Prozessframeworks, welches unterschiedliche Abstraktionsebenen vorsieht. Hierdurch wird es möglich, auf einer abstrakten Ebene, eine allgemeine Repräsentation für alle Prozesse zu ermöglichen (diese Repräsentanten bilden die Grundbausteine aller Prozesse) und das Qualitätsmodell zu integrieren. Auf unteren Abstraktionsebenen werden konkrete Ausprägungen in Form von Vorgehensmodellen modelliert, die aber alle Eigenschaften des abstrakten Modells aufweisen.

Ziel des PQI-Ansatzes ist es, allen Mitarbeitern, die an IT-Projekten beteiligt sind, genügend Unterstützung zur Erstellung qualitativ hochwertiger Softwareprodukte zu bieten, deren Qualität sich über heterogene Projekte hinweg erheben und steuern lässt.

Zusammenfassend lassen sich folgende Anforderungen an den PQI-Ansatz definieren:

**A1: Berücksichtigung der Heterogenität der zu entwickelnden Softwareprodukte**  
Das heterogene Produktportfolio setzt unterschiedliche Qualitätsanforderungen für die jeweiligen Produkte voraus. Demnach muss ein für den jeweiligen Projektkontext anpassbares Qualitätsmodell definiert werden. Dennoch muss ebenfalls eine unternehmenseinheitliche Mindestqualität definiert sein.

**A2: Berücksichtigung der Heterogenität der Entwicklungsprozesse**  
Unterschiedliche Produkte setzen unterschiedliche Vorgehensweisen bei der Softwareerstellung voraus, die ebenfalls berücksichtigt werden müssen. Die Definition eines vereinheitlichten Vorgehensmodells ist somit nicht zielführend, da viel zu unterschiedliche Vorgehensweisen vorherrschen und sinnvoll sind.

**A3: Vergleichbarkeit der KPI**

Trotz unterschiedlicher Softwareprodukte und Vorgehensmodelle stellt sich dennoch die Anforderung der Definition vergleichbarer Kennzahlen für Softwareprozesse und -produkte.

## 2 Begriffe

Zum besseren Verständnis werden nachfolgend in dieser Ausarbeitung verwendete Begrifflichkeiten definiert.

Das **Qualitätsmodell** dient der Definition der Softwarequalität, als Qualitätsvorgabe und zur Qualitätsbewertung. Das in der EnBW entwickelte Qualitätsmodell basiert auf der ISO/IEC 9126 [ISO/IEC01] und definiert Unterbegriffe (Qualitätsmerkmale), die die unterschiedlichen Aspekte der Softwarequalität angeben. Dabei sind die Qualitätsmerkmale hierarchisch angeordnet. Die Qualitätsmerkmale der obersten Stufe werden als Faktoren bezeichnet und werden immer weiter verfeinert. Die untergeordneten Qualitätsmerkmale heißen Qualitätskriterien bzw. Qualitätsindikatoren. Qualitätskriterien werden auf der „untersten Stufe“ über Metriken operationalisiert.

**Ein Prozessmodell** bezeichnet im Kontext der vorliegenden Arbeit eine Verkettung von Phasen, die in allen gängigen Vorgehensmodellen in unterschiedlichen Ausprägungen auffindbar sind. Eine jede dieser Phasen wird mit einem definierten Quality-Gate abgeschlossen.

**Phasen** dienen in der vorliegenden Arbeit als ein Sammelbegriff zur Gruppierung von Aktivitäten, die in allen Vorgehensmodellen in unterschiedlicher Ausprägung aufzufinden sind.

**Ein Vorgehensmodell** bezeichnet die nach „Lehrbuch“ beschriebenen Softwareentwicklungsprozesse, wie beispielsweise das Wasserfallmodell, Rational Unified Process oder auch ein V-Modell bzw. Scrum.

**Engineering Pattern** dient als Oberbegriff für unterschiedliche Patterns, die in der Software Entwicklung zum Einsatz kommen. Sie beinhalten Patterns, die den Bereichen Design Pattern, Test Pattern, Usability Pattern, Interaction Design Pattern, etc. zugeordnet sind. Dabei behält die Definition, dass ein Pattern ein Vorgehen für bekannte Probleme ist, weiterhin ihre Gültigkeit [AI77].

## 3 PQI – Prozess- und projektorientierte Qualitätsintegration

### 3.1 Grundlegender Ansatz

Die Basis für die Integration eines Qualitätsmodells in einem heterogenen Projektumfeld mit jeweils unterschiedlichen Qualitätsansprüchen bildet die Einbettung des Qualitätsmodells in die Entwicklungsprozesse. Dies geschieht dadurch, dass Entwicklungsprozesse auf unterschiedlichen Abstraktionsebenen, dem Prozessframework, beschrieben werden. Die Verankerung des Qualitätsmodells erfolgt auf der obersten Abstraktionsebene, wodurch jedes abgeleitete Vorgehensmodell das Qualitätsmodell erbt.

Wie bereits in der Einleitung angesprochen, besteht die DNA eines jeden Menschen aus gleichen Grundbausteinen. Alle gängigen Vorgehensmodelle bestehen aus einer Ideen-, Anforderungs-, Design-, Test- und Implementierungsphase. Weiterhin müssen Projektmanagementtätigkeiten durchgeführt werden. Diese Prämisse wurde verwendet, um ein allgemeingültiges Prozessframework aufzubauen, wodurch vergleichbare Prozesskennzahlen erhoben werden können. Die Vergleichbarkeit produktorientierter Kennzahlen wird durch die Integration des Qualitätsmodells in die abstrakten Beschreibungen von Prozessmodellen gewährleistet.

Die PQI-Prozessarchitektur besteht aus dem Prozessframework, dem Qualitätsmodell, sowie einem Trigger, welcher die ablauforientierte Prozessbeschreibung und die statische Qualitätssicht operativ verbindet. In den nachfolgenden Kapiteln wird die Prozessarchitektur vorgestellt. Im Abschnitt 3.2 wird das Prozessframework detailliert erklärt, während im Abschnitt 3.3 das Qualitätsmodell erläutert wird. Im Abschnitt 3.4 wird kurz auf den Trigger eingegangen.

### 3.2 Das PQI Prozessframework

Grundsätzlich besteht das Prozessframework aus unterschiedlichen Ebenen, die sich jeweils an unterschiedliche Stakeholder richten.

Die **Prozessmetamodell-Ebene** enthält das Prozess-Metamodell (M2) und bildet somit eine abstrakte Beschreibung für alle abgeleiteten Vorgehensmodelle. Auf Prozessmetamodell-Ebene werden das Qualitätsmodell sowie Quality Gates integriert. Somit ermöglicht diese Ebene die Erhebung von KPIs, die für das Management als Stakeholder des PQI relevant sind.

Die **Vorgehensmodell-Ebene** beinhaltet Vorgehensmodelle für die Softwareentwicklung (M1) wie beispielsweise Scrum oder V-Modell und richtet sich vor allem an Projektbeteiligte. Diese Ebene beinhaltet eine Entscheidungshilfe für Projektleiter zur Auswahl des geeigneten Vorgehensmodells bezogen auf ihren Projektkontext. Der Kontext ergibt sich aus der Projektgröße, dessen Kritikalität, sowie der Teamzusammensetzung. Je nach Ausprägung dieser Faktoren, wird ein Vorgehensmodell auf Basis einer Entscheidungslogik vorgeschlagen. Abweichungen vom vorgeschlagenen Vorgehensmodell müssen vom Projektleiter begründet und dokumentiert werden. Weiterhin nutzt sie eine Sammlung von Engineering Patterns, um die Vorgehensmodelle in den unterschiedlichen Entwicklungsphasen methodisch anzureichern. Die Engineering Patterns bieten Hilfestellung zur Erreichung angestrebter Qualitätsziele und werden unter anderem durch die Attribute Name, Kontext, Vorbedingung, Problembeschreibung, Problemlösung, Nachbedingung beschrieben.

Die **Mapping-Ebene** dient vor allem der Abbildung unterschiedlicher Vorgehensmodelle aufeinander. Insbesondere dient sie der „Explizit-Machung“ der Vererbungshierarchie („instance of“ Beziehungen) und damit äquivalenter Entitäten in den unterschiedlichen Vorgehensmodellen. Diese Ebene enthält redundante Informationen die in dieser Form für den Prozessarchitekten aufbereitet wird, um eine schnelle Vergleichbarkeit der Vorgehensmodelle zu gewährleisten.

Ein **Projekt** konkretisiert ein spezifisches Vorgehensmodell und instanziiert das Qualitätsmodell. Abbildung 1a) veranschaulicht die grundsätzlichen Abstraktionsebenen des Prozessframeworks, Abbildung 1b) zeigt die jeweiligen Modelle auf den jeweiligen Abstraktionsebenen.

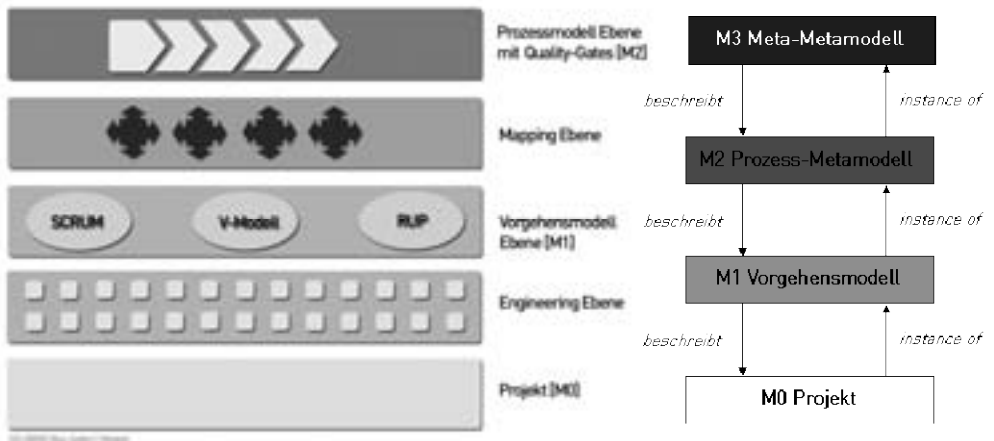


Abbildung 1a - vereinfachte Sicht auf Prozessframework    Abbildung 1b - Prozessframework Ebenen Ansicht

**M3 - Meta-Metamodell** Das Meta-Metamodell bildet die Grundlage zur Beschreibung von Prozessen beliebiger Art in einer Organisation, d.h. es stellt die Basis für die Beschreibung beliebiger Workflows dar. Generell betrachtet ordnen Workflows Phasen, in denen Methoden angewendet werden. Grundprämisse einer jeden Aktivität ist, dass

sie zu einem Ergebnis führt. Abbildung 2 zeigt das Meta-Metamodell M3 des Prozessframeworks.

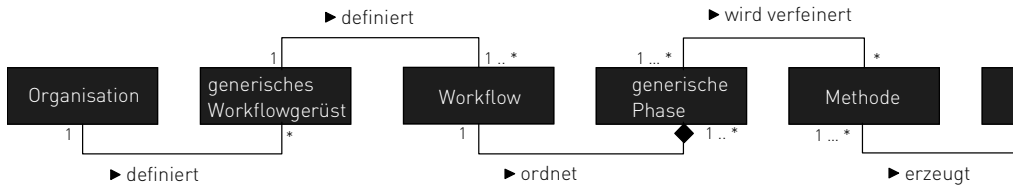
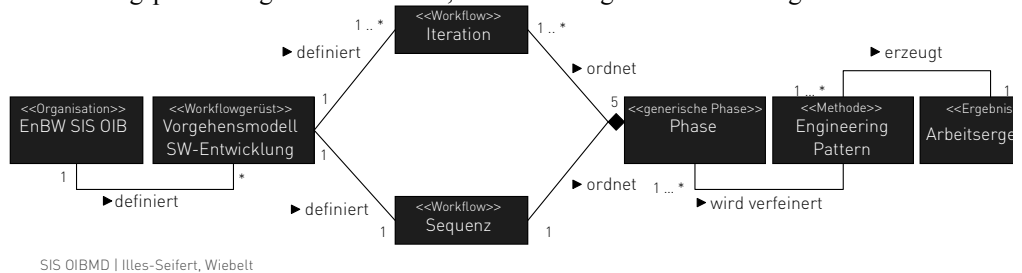


Abbildung 2 - M3 Meta-Metamodell von Workflows

**M2 - Prozess-Metamodell** Das Prozess-Metamodell definiert die „Grammatik“ zur Beschreibung von Vorgehensmodellen (Abbildung 3). Generell können Vorgehensmodelle iterativ oder sequenziell sein. In beiden Fällen setzen sich Iterationen und Sequenzen aus folgenden Phasen zusammen: Idee, Analyse, Design, Implementierung, Test, Projektmanagement, die je nach Vorgehensmodell spezifisch ausgeprägt sind. In den unterschiedlichen Phasen der Softwareentwicklung werden Methoden angewendet, um Arbeitsergebnisse zu erzeugen. Die Idee dieses Ansatzes ist es, nicht die Phasen über mehrere Aktivitäten zu verfeinern, sondern einen Baukasten an Engineering-Patterns zur Verfügung zu stellen, die in unterschiedlichen Entwicklungsphasen angewendet werden, um Arbeitsergebnisse zu erzeugen.



SIS OIBMD | Illes-Seifert, Wiebelt

Abbildung 3 - M2 Prozess-Metamodell

**M1 - Vorgehensmodelle (Wasserfall)** Auf dieser Ebene werden unterschiedliche Vorgehensmodelle definiert. In einem ersten Schritt wurden Wasserfall und Scrum instanziiert. Das Wasserfall-Modell (Abbildung 4) sieht die sequenzielle Durchführung der Entwicklungsphasen vor. Am Beispiel der Phase „Test“ soll verdeutlicht werden, wie Phasen, Engineering-Patterns und Arbeitsergebnisse zusammenhängen. Beispielsweise wird als Arbeitsergebnis das Testkonzept erstellt. Hierfür muss eine Risikoanalyse sowie eine Analyse bisheriger Fehler durchgeführt werden. Zur Erstellung der Testspezifikation können unterschiedliche Testpatterns angewendet werden.

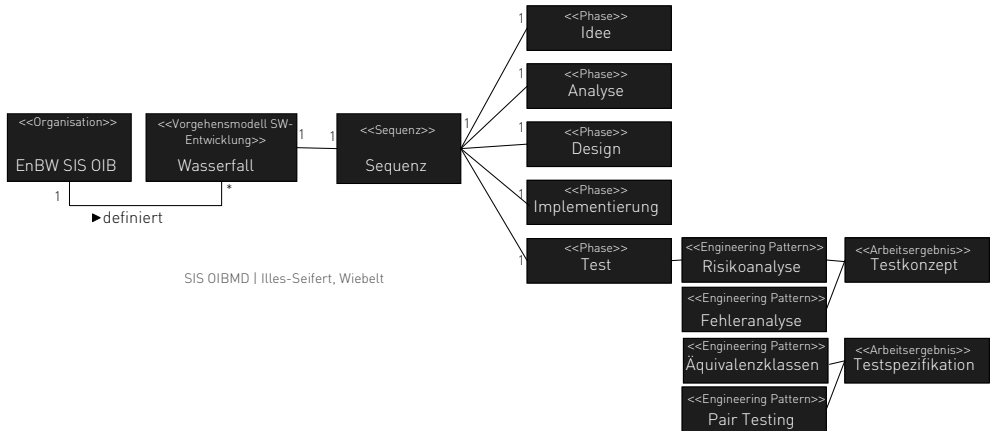


Abbildung 4 - M1 Wasserfall-Modell

**M1 - Vorgehensmodelle (Scrum)** Abbildung 5 stellt die Instanziierung von Scrum schematisch dar und beschränkt sich auf die in dem Vorgehensmodell enthaltenen Konzepte. Die Instanziierung zeigt die in der ersten Iteration stattfindende Festlegung der Vision sowie die Initialdefinition des „Product Backlog“. Die nächsten Iterationen, die in Scrum als „Sprint“ bezeichnet werden, enthalten jeweils ein „Sprint Planning Meeting“ in dem die Ziele, sowie die umzusetzenden Elemente des Product Backlogs definiert werden. Arbeitsergebnisse sind die „Sprint Goals“ sowie die im aktuellen Sprint umzusetzenden Features, die im „Sprint Backlog“ aufgelistet sind. In einem Sprint werden die Entwicklungsphasen überlappend durchgeführt. Für die Ausgestaltung innerhalb dieser Phasen macht Scrum keine Vorgaben. Der PQI-Ansatz sieht ein Baukasten-System zur methodischen Ausgestaltung dieser Phasen vor, indem je nach Qualitätszielen, entsprechende Engineering-Patterns vorgeschlagen werden. Wie die Instanziierung eines Vorgehensmodells in einem Projektkontext aussieht, wird im nachfolgenden Abschnitt beschrieben

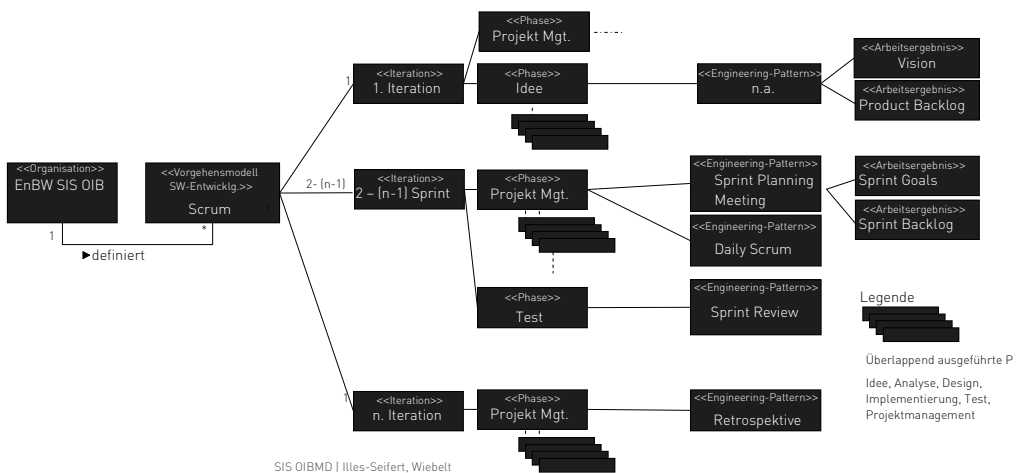


Abbildung 5 - M1 Scrum

**M0 - Projektebene** Softwareentwicklungsprojekte wählen, je nach Kontext, das für sie geeignete Vorgehensmodell aus und instanzieren dieses. Neben der Projektgröße, dessen Kritikalität, sowie der Teamzusammensetzung spielen Qualitätskriterien eine entscheidende Rolle bei der Instanziierung eines konkreten Vorgehensmodells. Grundsätzlich wird die Entscheidung getroffen, ob eine iterative oder eine sequenzielle Vorgehensweise gewählt wird. Weiterhin werden die Engineering-Patterns ausgewählt, die auf der Grundlage der Projektbedingungen, sowie der konkret ausgeprägten Qualitätskriterien vorgeschlagen werden. Steht beispielsweise das Qualitätskriterium „Usability“ im Fokus des Projektes, so werden für die unterschiedlichen Phasen Patterns (z.B. Paper Prototyping, Focus Groups, Fieldstudy, etc.) vorgeschlagen, die eine benutzerzentrierte Projektausrichtung sicherstellen. Die Auswahl der anzuwendenden Engineering-Patterns sowie der zu erzeugenden Arbeitsergebnisse wird durch unterschiedliche Faktoren beeinflusst.

- Faktor firmenweiter Richtlinien: Diese definieren Arbeitsergebnisse, die verpflichtend zu erstellen sind, unabhängig vom gewählten Vorgehensmodell.
- Faktor Vorgehensmodell: Das ausgewählte Vorgehensmodell kann bestimmte Engineering-Patterns oder Arbeitsergebnisse vorschreiben.
- Faktor Qualitätsziele: Die Ausprägung des instanziierten Qualitätsmodells kann dazu führen, dass bestimmte Engineering-Patterns anzuwenden und zugehörige Arbeitsergebnisse zu erstellen sind.

### 3.3 Das PQI Qualitätsmodell

Im Qualitätsmodell werden die Dimensionen interne und externe Softwarequalität, sowie Prozessqualität betrachtet. Zur Strukturierung der internen und externen Softwarequalität wird der ISO-Standard [ISO/IEC01] zu Grunde gelegt und um weitere Aspekte der Prozessqualität erweitert. Diese Aspekte umfassen die Gebiete: Projektplanung, Projektüberwachung sowie das Konfigurations- und Architekturmanagement. Das Qualitätsmodell ist hierarchisch gegliedert und setzt sich aus Faktoren zusammen, die über Qualitätskriterien verfeinert werden. Die Qualitätskriterien werden über Metriken operationalisiert.

Das Qualitätsmodell wird im Prozess-Metamodell verankert und somit allen abgeleiteten spezifischen Vorgehensmodellen zur Verfügung gestellt. Jedes Softwareentwicklungsprojekt wählt ein geeignetes Vorgehensmodell aus und definiert Qualitätsziele basierend auf dem Qualitätsmodell. Um diese Qualitätsziele zu erreichen werden unterschiedliche Engineering-Patterns angewendet. Bei der Definition der Qualitätsziele müssen Verantwortliche die relative Wichtigkeit der unterschiedlichen Qualitätsfaktoren auf einer dreistufigen Skala bewerten, wobei die erste Stufe der Skala den Standard definiert. Durch die getroffene Auswahl (was das Vorgehensmodell sowie die Qualitätsziele betreffen) werden Engineering-Patterns vorgeschlagen, die sich für den angegebenen Kontext am besten eignen. Die während der Softwareentwicklung erzeugte Ist-Qualität wird gemessen und mit der Soll-Qualität abgeglichen.

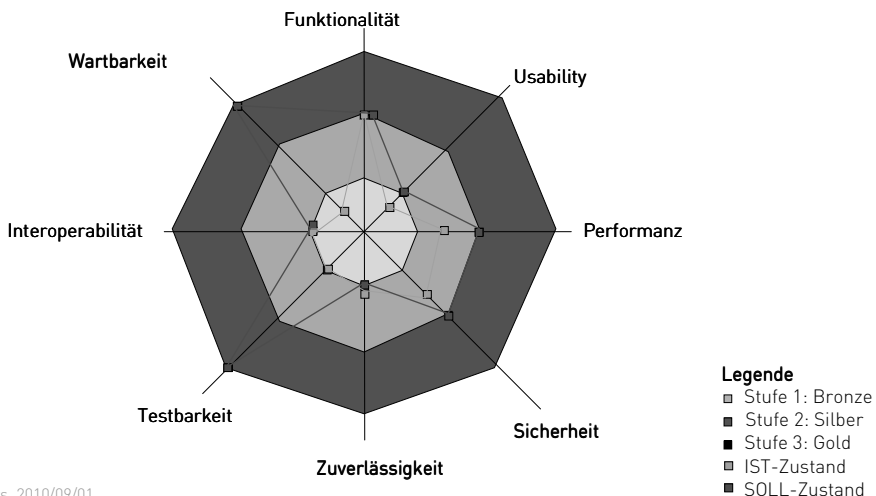


Die im PQI-Ansatz definierten Qualitätsstufen sind:

- Stufe 1 - Bronze: Das ist die „Standardqualität“, die für Softwareentwicklungen bezüglich eines angegebenen Qualitätsfaktors angeboten wird. (Der Kunde ist zufrieden, es fällt so wenig wie möglich negativ auf, das Produkt entspricht in weiten Teilen seinen Anforderungen).
- Stufe 2 - Silber: Das ist eine über den Standard hinaus gehende Qualität. (Der Kunde ist positiv überrascht, das Produkt entspricht vollkommen seinen Vorstellungen).
- Stufe 3 - Gold: Wenn Stufe Gold bezüglich eines Qualitätsfaktors gewählt wird, muss die Qualität die Erwartungen des Kunden weit übertreffen. (Kunde ist begeistert).

Wie wird nun das Qualitätsmodell projektspezifisch instanziiert? Auf Basis von Kundenanforderungen werden Prioritäten für die unterschiedlichen Qualitätsfaktoren festgelegt. Hierbei ist darauf zu achten, dass sich manche Qualitätskriterien widersprechen und das die Erreichung einer höheren Qualitäts-Stufe meist mit einem zusätzlichen Aufwand einhergeht.

Abbildung 6 zeigt exemplarisch, wie in einem Projekt die Qualitätsziele definiert wurden. Im gezeigten Beispiel sind vor allem die Qualitätsfaktoren Wartbarkeit und Testbarkeit besonders wichtig. Aufgrund der Ist-Analyse können der aktuelle Status bezüglich der einzelnen Faktoren gemessen und Engineering-Patterns integriert werden, um die Soll-Qualität zu erreichen. (Anmerkung: Die Engineering-Patterns bieten bewährte Praktiken an, die zur Herstellung und Überprüfung der geforderten Qualität eingesetzt werden können).



Vers. 2010/09/01  
Timea Illies-Seifert, Frank Wiebelt

Abbildung 6 - Instanziiertes Qualitätsmodell

### 3.4 Der PQI Trigger

Die Verknüpfung des ablauforientierten Prozess-Frameworks und des statischen Qualitätsmodells erfolgt über den Trigger [DCF07]. Dieser überwacht den Prozessablauf und meldet die Fertigstellung von Arbeitsergebnissen an definierten Quality Gates. Hierbei werden die Prüfdaten ausgewertet und die Soll- mit der Ist-Qualität verglichen. Die Prüfung kann manuell (Reviews, Test) oder rechnergestützt (statische Analyse) durchgeführt werden.

## 4 Verwandte Arbeiten

Bisherige Arbeiten konzentrieren sich auf die Vorstellung und Verbesserung einzelner Vorgehensmodelle für die Softwareentwicklung. Die ersten Arbeiten dieser Art wurden von Royce [Ro70] oder Boehm [Bo84], [Bo88] vorgestellt. Weitere Arbeiten, die auf einzelne Vorgehensmodelle eingehen, sind beispielsweise RUP [JBR99], Scrum [Sc01], V-Modell XT [Vm08]. Die Integration unterschiedlicher Vorgehensmodelle, insbesondere die Integration „traditioneller“ und agiler Methoden, wurde bisher nur wenig beleuchtet. Aktuelle Forschungsberichte thematisieren den Übergang von wasserfallorientierten Vorgehensmodellen hin zu agiler Entwicklung wie beispielsweise in [PMS09], [Su07], [Be07], [FG07], [Se07]. Der in diesem Bericht vorgestellte PQI-Ansatz zielt auf die *Integration* unterschiedlicher Vorgehensmodelle ab.

Eine weitere Gruppe verwandter Arbeiten fokussiert sich auf die Definition von Qualitätsmodellen. Hierzu gehören generische Frameworks, wie die ISO [ISO/IEC01], die die Softwarequalität gesamtheitlich betrachten. Andere Arbeiten untersuchen spezielle Qualitätsmerkmale, wie die Design-Qualität beispielsweise in [Bo94], [Co98], [Ka98], [BD02]. Die meisten Arbeiten dieser Gruppe konzentrieren sich auf das Qualitätsmodell selbst und weniger auf dessen Integration in Vorgehensmodelle. Der PQI-Ansatz behandelt die Qualitätsmodellierung und die Prozessmodellierung integrativ, womit erreicht wird, dass Softwarequalität als integraler Bestandteil der Softwareentwicklung verstanden und etabliert wird.

## 5 Diskussion und Zusammenfassung

In dem vorliegenden Dokument wird der bei der EnBW Systeme Infrastruktur Support GmbH erarbeitete Ansatz zur prozess- und projektorientierten Qualitätsintegration (PQI) vorgestellt. Dieser sieht eine Beschreibung der Softwareentwicklungsprozesse auf unterschiedlichen Abstraktionsebenen vor. Die Verankerung des Qualitätsmodells erfolgt auf der höchsten Abstraktionsebene, wodurch dessen Verwendung in allen abgeleiteten Vorgehensmodellen und Projekten verbindlich wird. Jedes Projekt kann das Qualitätsmodell instanzieren und Qualitätsziele definieren. Hierdurch wird gewährleistet, dass Projekte an der Erreichung der individuell festgelegten Qualitätszielen gemessen werden (Delta-Messung), wobei gleichzeitig eine firmenweit gültige Standardqualität zu Grunde gelegt wird. Hierdurch kann die Heterogenität der zu entwickelnden Produkte, sowie deren individuellen Qualitätsziele Rechnung getragen werden (A1).

Der hier vorgestellte PQI-Ansatz erfüllt weiterhin die Anforderung der Integration unterschiedlicher Vorgehensmodelle (A2) für die Softwareentwicklung. Das Prozessframework erlaubt die Einführung unterschiedlicher Vorgehensmodelle in das Unternehmen ohne das Prozessframework zu verändern. Die Grundprämisse ist, dass jedes neue Vorgehensmodell vom generischen Workflow abgeleitet ist.

Durch das Zugrundelegen des generischen Workflows, auf dem alle Vorgehensmodelle basieren, ist es möglich, vergleichbare Prozesskennzahlen über Projekte hinweg zu erheben (A3). Weiterhin wird durch die Instanziierung des Qualitätsmodells möglich, vergleichbare Kennzahlen über die Software-Qualität zu generieren. Dabei wird für die Berichterstattung über die KPIs nicht der absolut erreichte Wert verwendet, sondern die Soll-Ist-Differenz. Der Soll-Wert wird zum Projektstart in Zusammenarbeit mit dem Auftraggeber definiert.

Ein Projekt ist durch seine Einzigartigkeit definiert. Durch den hier vorgestellten Ansatz wird es möglich, zum einen ein Grundgerüst vorzugeben, um vergleichbare Kennzahlen zu erheben. Andererseits können Projekte individuell ihr Vorgehensmodell, sowie die anzuwendenden Engineering-Patterns definieren.

Der hier vorgestellte Ansatz beruht nicht auf dem Prinzip des Tailorings, der einen Standard Prozess voraussetzt und Anpasst. Vielmehr werden Regeln definiert, die es erlauben, Vorgehensmodelle je nach Kontext auszuprägen. Der Vorteil der hier gewählten Lösung besteht in der *Einfachheit* der Prozessbeschreibung auf abstrakter Ebene und in der *Integration von bewährten Engineering-Patterns*, als Wissensbasis für die Organisation, die das gewählte Vorgehensmodell durch beschriebene Engineering Patterns anreichern.

Die nächsten Schritte sehen den Ausbau der Design-Patterns, die Erweiterung von etablierten Vorgehensmodellen, sowie die Verankerung des Modells in der gesamten Organisation vor.

## Literaturverzeichnis

- [AI77] Alexander, C., Ishikawa, S.: Murray Silverstein Subject(s) Architecture Publisher Oxford University Press Publication date 1977
- [BD02] Bansiya, J.; Davis, C.: A Hierarchical Model for Object-Oriented Design Quality Assessment, IEEE Transactions on Software Engineering, 28(1), 2002, 4-17.
- [Be07] Beavers, P. A.: Managing a large “agile” software engineering organization, Proceedings of the AGILE Conference, pp. 296-303, 2007.
- [Bo84] Boehm, B.W.: Verifying and Validating Software Requirements and Design Specifications, IEEE Software, vol. 1, no. 1, 75-88, 1984.
- [Bo88] Boehm, B.W.: A Spiral Model of Software Development and Enhancement, Computer, v.21 n.5, p.61-72, May 1988.
- [Bo94] Booch, G.: Object-Oriented Analysis and Design with Applications (2. Auflage). Benjamin/Cummings, Redwood City, CA, 1994.
- [Co98] Cockburn, A.: Object-Oriented Analysis and Design: Part 2. C/C++ Users Journal, 16(6), 1998.
- [DCF07] Damiani, E.; Colombo, A.; Frati, F. und Bellettini, C.: A metamodel for modeling and measuring Scrum development process, In (Concas, G.; Damiani, E.; Scotto, M. und Succi, G. Hrsg.); Proceedings of the 8th international Conference on Agile Processes in Software Engineering and Extreme Programming; Lecture Notes In Computer Science. Springer-Verlag, Berlin, Heidelberg, 74-83.
- [FG07] Fry, C. und Greene, S.: Large scale agile transformation in an on-demand world. In Proceedings of the AGILE Conference, Seiten 136-142, 2007.
- [ISO/IEC01] Standard, 9126. 2001. ISO/IEC Standard 9126: Software Engineering -- Product Quality. Part 1: International Organization for Standardization.
- [JBR99] Jacobson, I.; Booch, G.; Rumbaugh, J.: The unified software development process, Addison-Wesley, 1999
- [Ka98] Kafura, D.: Object-Oriented Software Design and Construction with C++. Prentice Hall, Upper Saddle River, NJ, 1998.
- [PMS09] Pinheiro, C.; Maurer, F.; Sillito, J.; MCIT Solutions, AB, Improving Quality, One Process Change at a Time, 31st International Conference on Software Engineering - Companion Volume, 2009. ICSE-Companion 2009.
- [Ro70] Royce, W. W., Managing the Development of Large Software Systems: Concepts and Techniques, Proceedings of WESCON, August 1970.
- [Sc01] Schwaber, K.; Beedle, M.: Agile Software Development with Scrum. Prentice Hall, Upper Saddle River 21. Oktober 2001.
- [Se07] T. R. Seffernick. Enabling agile in a large organization our journey down the yellow brick road. In Proceedings of the AGILE Conference, pp. 200-206, 2007.
- [Su07] Sumrell, M.; From Waterfall to Agile - How does a QA Team Transition? Proceedings of the Agile2007.
- [Vm08] [www.v-modell-xt.de](http://www.v-modell-xt.de), V-Modell XT.