

Zielgerichtete Kontrolle von Software-Entwicklungsprojekten

Jens Heidrich

Fraunhofer-Institut für Experimentelles Software Engineering
Fraunhofer Platz 1, 67663 Kaiserslautern
jens.heidrich@iese.fraunhofer.de

Abstract: Vielen Software-Entwicklungsorganisationen fehlt die notwendige Unterstützung, um quantitative Kontrolle über ihre Entwicklungsprozesse ausüben zu können; etwa, um die Performanz zum Einsatz kommender Entwicklungsprozesse oder die Qualität erzeugter Produkte zu vermessen. Eine systematische Unterstützung zur Identifikation kritischer Planabweichungen und Projektrisiken sowie die Einplanung entsprechender Gegenmaßnahmen zur Erreichung explizit definierter Projektziele fehlen üblicherweise. Ein Mittel, um quantitative Projektkontrolle auf der Basis expliziter Modelle zu etablieren, ist die Entwicklung und Einführung so genannter Software-Projektleitstände zur systematischen Qualitätssicherung und Unterstützung des Projektmanagements. Ein Leitstand ist vergleichbar mit einem Kontrollzentrum im Bereich der mechanischen Produktion. Seine Aufgaben umfassen das Sammeln, Interpretieren und Visualisieren von Messdaten. Darüber hinaus werden Informationen kontext-, zweck- und rollengebunden für alle beteiligten Interessengruppen während der Durchführung eines Software-Entwicklungsprojektes angeboten (z.B. für Projektmanager und Qualitätssicherer). Der vorliegende Artikel gibt einen Überblick über bestehende Leitstandlösungen und beschreibt den Specula-Ansatz zur zielgerichteten Generierung von Leitständen. Darüber hinaus werden Evaluierungsergebnisse aus der industriellen Anwendung vorgestellt.

1 Einleitung

Die zunehmende Komplexität von Software-Entwicklungsprojekten in Bezug auf zu realisierende Anforderungen sowie zu koordinierende Entwicklungsprozesse erschwert die effektive und effiziente Steuerung der Projekte. Nach einer Studie der Standish-Gruppe [Gro03] müssen 15% der Entwicklungsprojekte abgebrochen und mehr als 50% mit deutlichen Zeit- und Budget-Überschreitungen zu Ende geführt werden. Dies lässt sich nicht zuletzt darauf zurückführen, dass entsprechende Mittel zum effektiven und effizienten Einsatz geeigneter Kontrollinstrumente fehlen, um Risiken und Planabweichungen im Projekt frühzeitig erkennen und entsprechende Gegenmaßnahmen einleiten zu können. Ein Mittel, um quantitative Projektkontrolle auf der Basis expliziter Modelle zu etablieren, ist die Entwicklung und Einführung so genannter Software-Projektleitstände [MH04] zur systematischen Qualitätssicherung und Unterstützung des Projektmanagements. Ein Software-Projektleitstand ist vergleichbar zu einem Leitstand aus dem Bereich der mecha-

nischen Produktion, in dem zentral Daten zusammenlaufen, entsprechend verarbeitet und zum Beobachten und Kontrollieren des Produktionsprozesses aufbereitet werden. Ein Projektleitstand sammelt Daten aus verschiedenen Datenquellen im Projekt. Die Daten werden verarbeitet und für verschiedene Stakeholder im Projekt angeboten. Ein Projektmanager verwendet beispielsweise Leitstände um einen schnellen Überblick über den Stand des Projekts zu erhalten und ein Qualitätssicherer benutzt Leitstände zur Kontrolle von Qualitätsanforderungen an das zu entwickelnde Produkt. Neben dieser primären Nutzergruppe von Leitständen kann prinzipiell jede im Projekt wahrgenommene Rolle von der angebotenen Funktionalität direkt oder indirekt profitieren. Die Bedürfnisse in Bezug auf Projektkontrolle unterscheiden sich jedoch in Bezug auf verschiedene Organisationen, Projekte und wahrgenommene Rollen. Sie hängen maßgeblich von organisationsweiten Zielen, dem Reifegrad der Entwicklungsprozesse und anderen Kontextfaktoren ab. Bei der Entwicklung eines multidisziplinären, verteilten Entwicklungsprojekts müssen beispielsweise Messdaten von verschiedenen Standorten integriert und in ein einheitliches, konsistentes Bild überführt werden.

Ein wesentlicher Erfolgsfaktor liegt darin, dass Leitstände an die spezifischen Ziele und Charakteristiken einer Organisation und von Projekten angepasst werden. Der Specula-Ansatz (*lat.* Überwachungsturm) [Hei09] ermöglicht die Generierung von Leitstandfunktionalität basierend auf wiederverwendbaren Komponenten zur Projektkontrolle, die entsprechend explizit spezifizierter Kontrollziele selektiert, angepasst und angewendet werden. Dabei werden die folgenden Zielsetzungen verfolgt:

- Z1 (Effektive Leitstandnutzung): Planabweichungen und Projektrisiken sollen möglichst frühzeitig und vollständig im Entwicklungsprozess identifiziert werden, so dass entsprechende Gegenmaßnahmen rechtzeitig eingeleitet werden können.
- Z2 (Effizientes Aufsetzen und Anwenden): Das effiziente Aufsetzen und Anwenden von Projektkontrollmechanismen hängt hochgradig von Expertenwissen ab. Die Identifikation der benötigten Funktionalität und der zum Einsatz kommenden Kennzahlen zur Projektkontrolle sowie deren effizienter Einsatz stellt in der Praxis eine Herausforderung dar. Der Specula-Ansatz soll das Aufsetzen und Anwenden geeigneter Mechanismen zur Projektkontrolle erleichtern.
- Z3 (Erweiterbarkeit und Anpassbarkeit): In der Regel werden die Projektkontrollmechanismen nicht systematisch basierend auf spezifischen Zielsetzungen und Charakteristiken abgeleitet, sondern die zum Einsatz kommenden Werkzeuge (wie Projektplanungs- oder Code-Vermessungswerkzeuge) bestimmen die Kennzahlen zur Projektkontrolle. Der Specula-Ansatz soll sicherstellen, dass die Projektkontrollmechanismen zielgerichtet erweitert und angepasst werden können.

Der Artikel gliedert sich wie folgt: Abschnitt 2 gibt einen kurzen Überblick über den Stand der Praxis und Forschung im Bereich Projektkontrolle, Abschnitt 3 stellt die grundlegenden Konzepte des Specula-Ansatzes vor und Abschnitt 4 fasst den Artikel zusammen, geht auf die industrielle Erprobung des Ansatzes ein und gibt einen Ausblick auf zukünftige Arbeiten.

2 Stand der Praxis und Forschung

Ein Überblick typischer Vertreter von Forschungsansätzen zur Projektkontrolle findet sich in [MH04]. Existierende (generische) Ansätze zur Projektkontrolle stellen jeweils nur Teillösungen zur Verfügung. Insbesondere der zielgerichtete Einsatz von Leitständen basierend auf einer erweiterbaren und anpassbaren Menge an Techniken und Methoden wird nur unzureichend unterstützt. Die SME (Software Management Environment) [GSF92] leitet beispielsweise zwar die angebotenen Kontrollmechanismen zielgerichtet ab, bietet aber lediglich eine festgeschriebene Menge an Mechanismen zur Datenanalyse und -visualisierung an. Der Nachfolger WebME (Web Measurement Environment) [TZ97] erreicht größere Flexibilität durch den Einsatz einer Skriptsprache zur Anpassung, hat jedoch Defizite in Bezug auf die zielgerichtete Auswahl vordefinierter Kontrollmechanismen. Ansätze wie Amadeus [SPSB91] bieten zwar vordefinierte flexible Komponenten zur Projektkontrolle an, jedoch keinen umfassenden Mechanismus zu deren zielgerichteten Auswahl und Anpassung.

In der Praxis kommen häufig organisationspezifische Dashboards (meist basierend auf Spreadsheet-Anwendungen) zum Einsatz. Die zur Anwendung kommenden Kennzahlen hängen von den spezifischen Zielsetzungen und Charakteristiken ab. Die Kontrollbereiche sind typischer Weise breit gestreut [PMI04]. Es gibt keinen Standardsatz an Kennzahlen, der in allen Entwicklungsprojekten in der gleichen Art und Weise zum Einsatz kommen kann. Nach [MCJ⁺02] sollten „gute“ Kennzahlen (a) die Analyse von Informationsbedürfnissen unterstützen, (b) notwendige Analysearten ermöglichen, (c) einen angemessenen Detailgrad anbieten, (d) auf mögliche Gegenmaßnahmen hinweisen und (e) Informationen rechtzeitig verfügbar machen, so dass entsprechende Gegenmaßnahmen eingeleitet werden können. Die Kennzahlen selbst sollten systematisch, basierend auf definierten Projektzielen, abgeleitet werden, beispielsweise unter Zuhilfenahme der Goal/Question/Metric-(GQM)-Methode und des Quality-Improvement-Paradigms (QIP) [BCR94]. Typische Beispiele für derartige Indikatoren sind in [GSF90] zu finden.

3 Specula-Ansatz

Specula beschreibt einen systematischen Ansatz zur zielgerichteten Generierung von Software-Projektleitständen [Hei09]. Frei konfigurierbare Kontrollbausteine werden dabei basierend auf explizit definierten Zielsetzungen in Bezug auf Projektkontrolle ausgewählt, an den jeweiligen Anwendungskontext angepasst und zur Anwendung gebracht, d.h., entsprechende Messdaten werden im Projektverlauf erhoben, entlang der spezifizierten Mechanismen analysiert und schließlich rollenspezifisch aufbereitet. Der Ansatz besteht aus vier Bestandteilen: (1) einer logischen Architektur für zielgerichtete Projektleitstände, (2) einem konzeptionellen Modell zur formalen Spezifikation von Kontrollbausteinen (Kontrollkomponenten) zur Sammlung, Verarbeitung und Visualisierung von Messdaten, (3) einer Methode zur Auswahl und Anpassung der Kontrollbausteine basierend auf Projektzielen und -charakteristiken und (4) der prototypischen Umsetzung eines unterstützenden Werkzeugs (Specula Project Support Environment, PSE) inkl. eines vordefinierten

Satzes an Kontrollbausteinen. Die Methode basiert dabei auf dem QIP zur Umsetzung eines verbesserungsorientierten Lernzyklus sowie dem GQM-Ansatz zur Spezifikation von Messzielen in Bezug auf Projektkontrolle und zur Ableitung der entsprechenden Leitstand-funktionalität. Die folgenden Abschnitte gehen auf die Basiskonzepte des Ansatzes, die logische Architektur sowie die Methode zur zielgerichteten Komposition von Kontrollkomponenten ein.

3.1 Basiskonzepte

Das konzeptionelle Modell des Specula-Ansatzes formalisiert die Prozesse der Sammlung, Verarbeitung und Visualisierung von Messdaten. Die Menge der Kontrollkomponenten zur Erfassung von Messdaten, zur Analyse und Interpretation der erfassten Daten und zur Visualisierung der verarbeiteten Daten wird *Visualisierungskette* genannt. Dabei werden für diese drei Bereiche Komponenten auf Typen-Ebene und Instanzen-Ebene unterschieden:

Datentypen beschreiben die Struktur der zu verarbeiteten Daten. Dies können einfache Strukturen, wie z.B. eine Zeitreihe (eine Sequenz von Paaren aus Zeitstempeln und entsprechenden Werten) sein oder auch komplexere, wie ein Projektplan (eine hierarchische Menge an Aktivitäten mit Start- und Endzeitpunkt sowie geplantem Sollaufwand).

Datenzugriffspakete beschreiben, auf welche Weise auf Datentypen zugegriffen werden kann. Beispielsweise, ob Daten in einer relationalen Datenbank vorrätig gehalten werden oder aus Dateien heraus eingelesen werden müssen. Datenzugriffspakete spezifizieren eine Reihe von individuellen Parametern für jede Zugriffsart eines Datentyps.

Webformulare beschreiben eine bestimmte Art, Daten eines bestimmten Typs in einen Leitstand mit Benutzerinteraktion einzulesen bzw. zu verwalten. Über ein Webformular kann beispielsweise ein in Dateiform vorliegender Projektplan vom Benutzer in den Leitstand eingelesen oder einzelne Bestandteile des Datensatzes gezielt verändert werden. Ein Webformular modifiziert auf diese Weise eine Reihe von Daten eines bestimmten Typs. Im Gegensatz zu Datenzugriffspaketen, die automatisiert über definierte Schnittstellen Daten einlesen, erfordern Webformulare Benutzerinteraktion.

Funktionen beschreiben Techniken und Methoden zur Projektkontrolle (wie z.B. Toleranzgrenzenprüfung, Meilensteintrendanalyse oder Arbeitswertverfahren). Eine Funktion verarbeitet Daten verschiedener Typen und produziert ihrerseits eine Reihe von Daten. Darüber hinaus wird eine Funktion über Parameter an die Projektspezifika angepasst. Eine Funktion zur Überprüfung von Soll/Ist-Planabweichungen verfügt beispielsweise über einen Parameter zur Spezifikation des Toleranzbandes.

Sichten beschreiben bestimmte Arten der Datenvisualisierung (wie z.B. allgemeine Diagrammtypen oder methodenspezifische Visualisierungsformen wie Dreiecksdiagramme zur Darstellung eines Meilensteintrends). Eine Sicht visualisiert Daten verschiedener Typen für bestimmte Rollen im Projekt und ist entsprechend parametrisierbar.

Datencontainer stellen Instanzen bestimmter Datentypen dar und repräsentieren die tatsächlich vom Leitstand verarbeiteten Daten. Datencontainer werden über ein Datenzu-

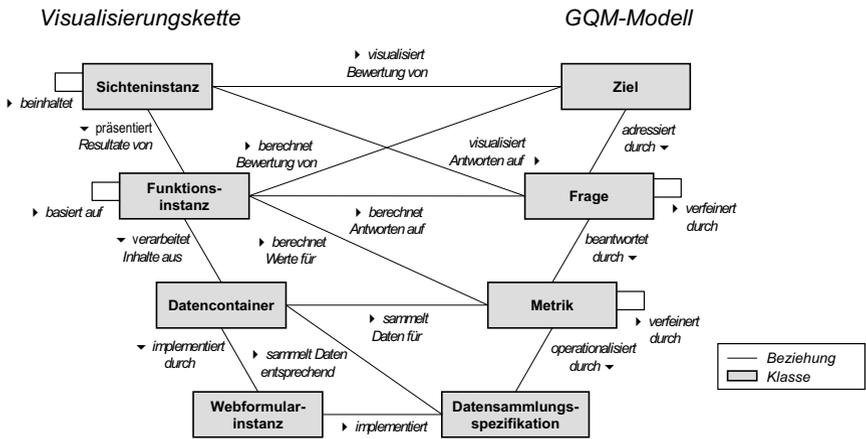


Abbildung 1: Zielgerichtete Ableitung einer Visualisierungskette.

griffspaket befüllt und können über Webformularinstanzen verwaltet werden.

Webformularinstanzen verwalten Datencontainer mit Benutzerinteraktion. Entsprechend der Spezifikation des zugehörigen Webformulars wird die Komponente parametrisiert und mit den benötigten Datencontainern versehen, aus denen Daten gelöscht, verändert oder hinzugefügt werden können.

Funktionsinstanzen wenden entlang der Spezifikation der entsprechenden Funktionen Kontrolltechniken auf einen Satz eingehender Daten an und produzieren die entsprechenden ausgehenden Daten. Funktionsinstanzen können die Ausgaben anderer Funktionsinstanzen als Eingaben weiterverarbeiten und ein Funktionsgeflecht bilden.

Sichteninstanzen wenden die Sichtenspezifikationen auf einen Satz eingehender Daten an. Sichteninstanzen können andere Sichteninstanzen referenzieren, um eine Hierarchie aufzubauen (etwa alle Sichten für eine bestimmte Rolle im Projekt).

Jede Kontrollkomponente enthält entsprechende Prüfmechanismen, um die Konsistenz und Vollständigkeit der Spezifikation sicherzustellen. Eine Visualisierungskette instanziiert Kontrollkomponenten aus der Typen-Ebene und besteht aus einem Geflecht von Datencontainern, Webformularinstanzen, Funktionsinstanzen und Sichteninstanzen und spezifiziert folglich alle zur Kontrolle eines spezifischen Entwicklungsprojektes benötigten Mechanismen. Visualisierungsketten sind projektspezifisch, enthalten aber Muster von Komponenten, die projektübergreifend wiederverwendet werden können.

Bei der zielgerichteten Auswahl der Kontrollkomponenten kommt der GQM-Ansatz zum Tragen, mit dessen Hilfe Messziele formalisiert und über einen strukturierten Ansatz über eine Reihe von Fragen schließlich Metriken abgeleitet werden können, welche quantifizierte Antworten auf die Fragen und darüber schließlich eine Aussage in Bezug auf die Erfüllung des Messziels liefern. Der GQM-Plan wird durch eine Datensammlungsspezifikation (DSS) vervollständigt, die festlegt, wer welche Daten aus welcher Datenquellen zu

welchem Zeitpunkt zu erfassen hat. Abbildung 1 zeigt einen Überblick der Kontrollkomponenten auf Instanzen-Ebene und deren Bezug zu den logischen Bestandteilen des GQM-Ansatzes. Datencontainer sammeln Messdaten entlang der GQM-Metrikspezifikation und der DSS. Funktionsinstanzen berechnen Daten für komplexe Metriken, evaluieren die Antworten auf GQM-Fragen unter Einbeziehung der entsprechenden GQM-Metriken und Anwendung des in der Funktion spezifizierten Interpretationsmodells und evaluieren die Erreichung des GQM-Messziels unter Einbeziehung der Evaluierungsergebnisse der zugehörigen GQM-Fragen. Sichteninstanzen visualisieren die Evaluierungsergebnisse der GQM-Fragen und GQM-Messziele durch eine der adressierten Rolle angemessenen Darstellung.

3.2 Logische Architektur

Abbildung 2 zeigt einen Überblick der logischen Architektur des Specula-Ansatzes. (1) Das *Planungsmodul* selektiert die Kontrollkomponenten basierend auf Projektzielen und -charakteristiken aus einem entsprechenden Repository und passt diese an den jeweiligen Kontext an. Die so entstehende Visualisierungskette spezifiziert alle zur Anwendung kommenden Projektkontrollmechanismen. (2) Das *Ausführungsmodul* sammelt Messdaten während der Projektdurchführung, analysiert dieselben entlang der spezifizierten Datenverarbeitungsmechanismen und erzeugt rollenspezifische Visualisierungen. Die so angebotenen Kontrollmechanismen werden von entsprechenden Stakeholdern zur Identifikation von Planabweichungen und Projektrisiken verwendet, so dass darauf basierend Gegenmaßnahmen eingeleitet werden können. (3) Das *Packaging-Modul* sammelt das Feedback aller Stakeholder in Bezug auf die zur Anwendung kommenden Kontrollmechanismen und analysiert deren Verbesserungspotential (beispielsweise ob alle Planabweichungen und Risiken identifiziert wurden oder die eingeleiteten Gegenmaßnahmen eine Wirkung gezeigt haben). Der Werkzeugprototyp Specula Project Support Environment (PSE) implementiert die logische Architektur und stellt ein Repository aus Kontrollkomponenten zur Verfügung, die bei Bedarf entsprechend selektiert und angepasst werden können. Während die Anwendung einer Visualisierungskette automatisiert erfolgt, ist der Selektionsprozess der Kontrollkomponenten, basierend auf Zielen und Charakteristiken eines Projektes, lediglich teilautomatisiert. Das Packaging-Modul wird in Bezug auf die Verwaltung des Kontrollkomponenten-Repositories unterstützt. Auf Basis der Specula PSE wurde auch die Erprobung des Ansatzes im Rahmen industrieller Fallstudien durchgeführt.

3.3 Zielgerichtete Komposition von Kontrollkomponenten

Die Specula-Methode zur zielgerichteten Generierung von Projektleitständen unterscheidet (analog zum QIP) sechs grundlegende Phasen, die im Folgenden anhand eines vereinfachten Beispiels veranschaulicht werden.

Phase I (Charakterisieren der Kontrollumgebung): In der ersten Phase werden relevante Kontextfaktoren in Bezug auf Projektkontrolle spezifiziert und die organisatorischen

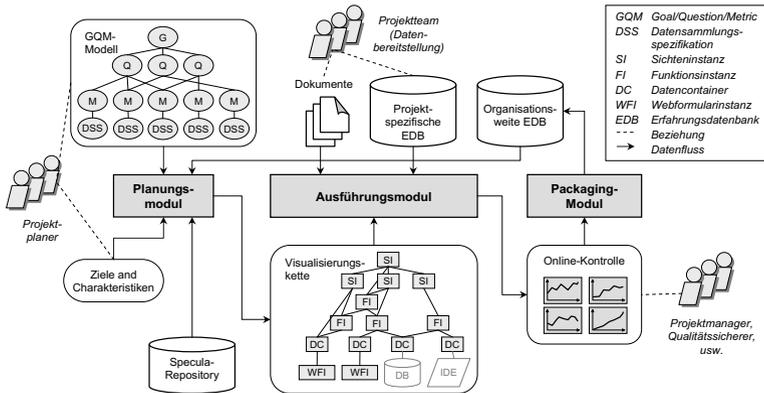


Abbildung 2: Konzeptionelle Architektur des Specula-Ansatzes.

Zuständigkeiten festgelegt. *Beispiel:* Ein Projekt X im Bereich der Entwicklung mobiler Services mit 17 Teammitgliedern soll durch eine Leitstand unterstützt werden. Kontrollaufgaben sollen durch den Projektmanager und Qualitätssicherer wahrgenommen werden.

Phase II (Definition von Kontrollzielen): In der zweiten Phase werden Projektkontrollziele auf Basis der Kontrollumgebung sowie höher gelegener Ziele einer Organisation (etwa Geschäftsziele) definiert. Diese werden mit Hilfe der GQM-Methode spezifiziert und über GQM-Fragen zu entsprechenden Metriken verfeinert bis ein vollständiges GQM-Modell entstanden ist. *Beispiel:* Eine Analyse der Kontrollziele für Projekt X führt zu insgesamt 6 Zielen im Bereich der Kontrolle des Aufwandsplans, des Zeitplans, der Qualität der Aufwandserfassung, der Konsistenz des Projektplans, der Qualität des erstellten Codes und der Effizienz von Fehlerfindungsaktivitäten im Projekt. Ein Ausschnitt des zugehörigen GQM-Plans wird in Abbildung 3 auf der linken Seite dargestellt.

Phase III (Zielgerichtete Komposition): Basierend auf dem GQM-Plan erfolgt die zielgerichtete Auswahl und Anpassung der entsprechenden Kontrollkomponenten aus dem Repository. Dies umfasst die Spezifikation eines Messplans (DSS), entsprechender Interpretationsmodelle, die Ableitung der Kontrollkomponenten basierend auf den in Abbildung 1 dargestellten Beziehungen, das Überprüfen der Konsistenz und Vollständigkeit der Visualisierungskettenspezifikation sowie das Konfigurieren und Etablieren des Leitstandwerkzeugs. *Beispiel:* Ein Ausschnitt der für das Projekt X abgeleiteten Visualisierungskette wird in Abbildung 3 auf der rechten Seite dargestellt. Zur Erfassung der Aufwandsrohdaten (Aufwand pro Aktivität und Mitarbeiter) wurde ein entsprechender Datentyp instanziiert und über ein Datenzugriffspaket an die Aufwandserfassungsdatenbank der Organisation angeschlossen. Zur Erfassung des Projektplans werden strukturelle Informationen und die Sollaufwandsdaten in getrennten Datencontainern verwaltet. Beide Container werden durch eine Webformularinstanz befüllt, welche die entsprechenden Daten aus einer MS-Project-Datei importiert. Eine Funktionsinstanz aggregiert die Aufwandsrohdaten um den Ist-Aufwand pro Aktivität des Projektes zu erheben. Die Funktionsinstanz der Aufwandsanalyse vergleicht Ist- und Sollaufwand, um die Aufwandsplanabweichung zu ermitteln.

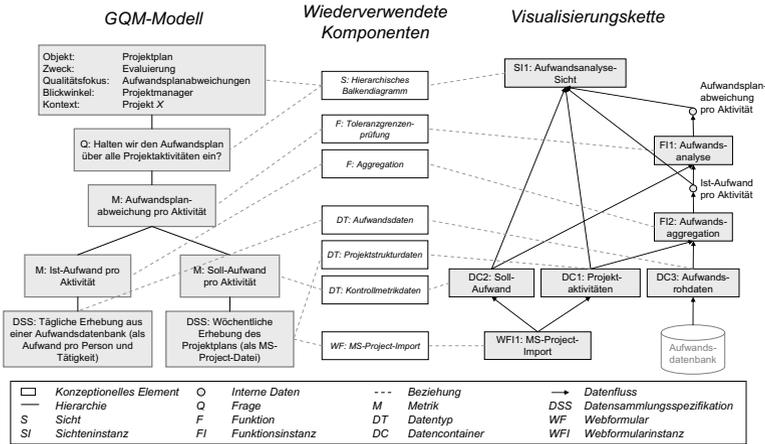


Abbildung 3: Beispielableitung einer Visualisierungskette.

Die Resultate dieses Verarbeitungsprozesses münden schließlich in der Aufwandsanalyse-Sicht zur Beantwortung der entsprechenden GQM-Frage bzw. des GQM-Ziels. Die vollständige Visualisierungskette für das Beispielprojekt X besteht aus einem Geflecht von 13 Webformularinstanzen, 15 Datencontainern, 8 Funktionsinstanzen und 8 Sichteninstanzen.

Phase IV (Anwendung der Kontrollmechanismen): In der fünften Phase werden entsprechend der Visualisierungskettenspezifikation Messdaten über Datenzugriffspakete und Webformularinstanzen gesammelt, durch Funktionsinstanzen verarbeitet und mittels Sichteninstanzen für alle Stakeholder zur Erkennung von Planabweichungen und Projektrisiken aufbereitet. *Beispiel:* Die Team-Mitglieder des Projektes X stellen entsprechende Daten über Webformularinstanzen in den Leitstand ein bzw. füllen die über Datenzugriffspakete angebotenen Datenquellen. Der Projektmanager und Qualitätssicherer verwenden die angebotenen Sichteninstanzen zur Kontrolle des Projektes.

Phase V (Analyse der Resultate): Nach Beendigung des Projektes wird die Visualisierungskette in Bezug auf ihre Stärken und Schwächen analysiert. Dazu werden alle Planabweichung und Projektrisiken auf ihre rechtzeitige und vollständige Erkennung hin untersucht und die Ursachen für zu späte und unvollständige Erkennung analysiert. Dies hat dann eine Anpassung des GQM-Plans, Messplans, der Interpretationsmodelle oder der Visualisierungskette zur Folge. *Beispiel:* Nehmen wir an, dass sich für Beispielprojekt X ein Meilenstein verspätete und dies nicht durch den Leitstand erkannt wurde. Eine Analyse könnte ergeben, dass dies auf die fehlende Aktualität des Projektplans zurückgeführt werden könnte.

Phase VI (Verpacken der Resultate): Auf Basis der Analysephase werden Verbesserungsmaßnahmen für zukünftige Projekte abgeleitet, d.h., GQM-Pläne, Messpläne, Interpretationsmodelle und Kontrollkomponenten entsprechend überarbeitet. *Beispiel:* Um die Aktualität des Projektplans zu verbessern, muss der Messplan entsprechend angepasst werden (etwa von wöchentlichen auf tägliche Aktualisierungszyklen des Projektplans in

kritischen Projektphasen).

4 Zusammenfassung und Ausblick

Der vorliegende Artikel illustriert den Specula-Ansatz zur zielgerichteten Generierung von Software-Projektleitständen. Wiederverwendbare Kontrollkomponenten werden basierend auf Zielen und Charakteristiken eines Software-Entwicklungsprojektes ausgewählt und entsprechend angepasst. Basierend auf den zur Anwendung kommenden Kontrollkomponenten wird die benötigte Leitstandfunktionalität generiert. Dem Ansatz liegt eine Methode zur Selektion, Anpassung, Anwendung und kontinuierlichen Verbesserung der Kontrollkomponenten zu Grunde.

Der Specula-Ansatz wurde im Rahmen industrieller Fallstudien im Kontext des vom Bundesministerium für Bildung und Forschung geförderten Soft-Pit-Projektes (Förderkennzeichen: 01ISE07A) in Bezug auf die Forschungsziele (Z1) effektive Nutzung, (Z2) effizientes Aufsetzen und Anwenden und (Z3) Erweiterbarkeit und Anpassbarkeit des Leitstands untersucht. Drei Projektpartner wendeten die Leitstandstechnologie im Rahmen industrieller Fallstudien über drei Iterationen des Projektes an. Dabei wurden die folgenden Ergebnisse in Bezug auf die Forschungsziele der Arbeit erzielt [Hei09]: **Z1:** Der Ansatz führte im Rahmen der Fallstudienprojekte zu einer frühzeitigeren und vollständigeren Erkennung von Planabweichungen und Projektrisiken. Zwischen 25% und 80% der zu findenden Abweichungen und Risiken wurden früher und zwischen 20% und 63% wurden zusätzlich aufgedeckt. Auf Grund der limitierten Menge an Fallstudien sind diese Ergebnisse allerdings lediglich als Tendenzen zu interpretieren und haben keine statistische Aussagekraft. **Z2:** Im Rahmen der Fallstudien wurde der subjektive Nutzen und die Benutzbarkeit des Ansatzes während des Aufsetzens und Anwendens des Leitstands untersucht. Beides wurde (mit Variationen über die Projektiterationen) mit statistischer Signifikanz als positiv beurteilt. **Z3:** Im Bereich der Erweiterbarkeit und Anpassbarkeit wurde neben den rein technischen Aspekten im Wesentlichen der Aufwand, der mit dem Aufsetzen und Anwenden des Leitstands verbunden ist, charakterisiert. Der anteilmäßige Aufwand für die Verwendung des Leitstands beläuft sich auf 10-12% des Gesamtaufwands. Hierbei ist anzumerken, dass auch Aufwand zur Datensammlung enthalten ist und der Overhead für den Einsatz des Werkzeugprototyps, insbesondere in der Anfangsphase und bei kurzer Einsatzzeit, als sehr hoch anzusehen ist. Im produktiven Einsatz eines Leitstands sollte sich dieser Aufwand entsprechend relativieren. Auch hier sind die Ergebnisse lediglich als Tendenzen zu interpretieren und haben keine statistische Aussagekraft.

Zukünftige Arbeiten konzentrieren sich auf die speziellen Herausforderungen im Bereich der globalen, verteilten Entwicklung von Software sowie die Unterstützung von Systementwicklungsprojekten (mit Hardware- und Softwarebestandteilen). Darüber hinaus ist es von zunehmender Wichtigkeit, den Beitrag von Softwareentwicklungsaktivitäten (bis hin zur quantitativen Steuerung einzelner Projekte) zu den Geschäftszielen einer Organisation klar aufzeigen zu können, um notwendige Budgetmittel und Ressourcen zu rechtfertigen und die Auswirkungen von Zieländerungen auf der Software-Ebene zu verstehen. Die GQM⁺Strategies[®]-Methode [BHL⁺07] adressiert erste Schritte in diese Richtung.

Literatur

- [BCR94] V.R. Basili, G. Caldiera und D.H. Rombach. *Encyclopedia of Software Engineering*, Kapitel The Experience Factory, Seiten 511–519. Wiley-Interscience, 1994.
- [BHL⁺07] V.R. Basili, J. Heidrich, M. Lindvall, J. Münch, M. Regardie und A. Trendowicz. Aligning Business Strategies with Software Measurement. In *1st International Symposium on Empirical Software Engineering*, Seiten 488–490. IEEE Computer Society, 2007.
- [Gro03] The Standish Group. CHAOS Chronicles v3.0. Bericht, The Standish Group, 2003.
- [GSF90] GSFC. *Manager's handbook for software development: revision 1*. National Aeronautics and Space Administration, Goddard Space Flight Center, Greenbelt, Md., 1990.
- [GSF92] GSFC. *Software management environment (SME), concepts and architecture*. National Aeronautics and Space Administration, Goddard Space Flight Center, Greenbelt, Md., rev. 1. Auflage, 1992.
- [Hei09] J. Heidrich. *Goal-oriented Quantitative Software Project Control*, Jgg. 24 of *PhD Theses in Experimental Software Engineering*. Fraunhofer-Institut für Experimentelles Software Engineering, Kaiserslautern; Univ. of Kaiserslautern, 2009.
- [MCJ⁺02] J. McGarry, D. Card, C. Jones, B. Layman, E. Clark, J. Dean und F. Hall. *Practical software measurement: objective information for decision makers*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
- [MH04] J. Münch und J. Heidrich. Software project control centers: concepts and approaches. *Journal of Systems and Software*, 70(1-2):3–19, 2004.
- [PMI04] PMI. *A Guide To The Project Management Body Of Knowledge (PMBOK Guides)*. Project Management Institute, 2004.
- [SPSB91] R.W. Selby, A.A. Porter, D.C. Schmidt und J. Berney. Metric-driven Analysis and Feedback Systems for Enabling Empirically Guided Software Development. In *13th International Conference on Software Engineering*, Seiten 288–298. IEEE Computer Society, 1991.
- [TZ97] R. Tesoriero und M. Zekowitz. *The Web Measurement Environment (WebME): A Tool for Combining and Modeling Distributed Data*, 1997.



Jens Heidrich ist Leiter der Abteilung für Prozesse und Messverfahren am Fraunhofer-Institut für Experimentelles Software Engineering (IESE) in Kaiserslautern. Seine Forschungsinteressen liegen in den Bereichen Software-Projektsteuerung, Kostenschätzung und Software-Qualitätsbewertung sowie der empirischen Untersuchung von Techniken und Methoden des Software-Engineerings in der industriellen Praxis. Herr Heidrich studierte Informatik an der Universität Kaiserslautern und schloss 2001 mit Auszeichnung ab. Von 2001 bis 2007 war er Mitarbeiter der Arbeitsgruppe „Software Engineering“ an der Universität Kaiserslautern. Bis 2003 war er Mitglied des SFB 501 „Entwicklung großer Systeme mit generischen Methoden“. Herr Heidrich promovierte im Bereich der quantitativen Steuerung von Software-Entwicklungsprojekten unter Leitung von Prof. Dr. Dieter Rombach. Diese Arbeit wurde mit Auszeichnung bewertet und mit dem Dissertationspreis des Fraunhofer IESE 2008 ausgezeichnet.