

1. Einleitung

Ich fange an, wo der letzte Mann aufgehört hat.

(Thomas Alva Edison)

Inhaltsangabe

1.1. Zielsetzung	2
1.2. Aufbau	2

Der erfolgreiche Einsatz von Metriken im Unternehmen hat nachweislich positiven Einfluss auf den Entwicklungsprozess von Softwaresystemen [31]. Ein gesunder Entwicklungsprozesses, wirkt sich wiederum in vorteilhafter Weise auf die Produktqualität, die Projektkosten und die Entwicklungszeiten aus [50]. Das Software Engineering Institute (SEI) hat mit dem CMMI¹-Referenzmodell eine Sammlung von grundsätzlichen Praktiken entwickelt, um Unternehmen bei der Prozessverbesserung zu unterstützen. Eine wesentliche Rolle spielt die unternehmensweite systematische Anwendung von Metriken für die Tätigkeit des Projektmanagements.

Die Forschungsgruppe Softwarekonstruktion arbeitet aktuell an einem Softwaresystem namens MeDIC, welches die Verwaltung, Zuteilung und Überwachung von Metriken für verschiedene Software-Projekte eines Unternehmens ermöglicht. Im Allgemeinen ist MeDIC (Measure Documentation, Integration and Configuration) eine Werkzeugsammlung zur Unterstützung des Projektmanagements von Softwareprojekten. Den Hauptprogrammteil bildet das MeDIC-Dashboard, welches in systematischer und geordneter Weise projektspezifische Metriken visualisiert, um den Entwicklungsprozess transparenter und verständlicher zu gestalten. Die Metriken sind mit Diagrammen (Dashboard-Items) assoziiert, welche auf Basis von Informationsbedürfnissen für ein Dashboard zusammengestellt werden.

Zur Integration heterogener Systeme in MeDIC, wurde das Messarchitekturmodell Enterprise Measurement Infrastructure (EMI) entworfen. Ein elementarer Bestandteil dieser Architektur sind die Rechenkerne (auch Metrikmodelle genannt), die für die Berechnung der Metriken zuständig sind. Gegenwärtig fehlt der Messarchitektur ein Konzept, um Variabilität für Metriken zu beschreiben. Für den Projektleiter in seiner Rolle als Dashboard-Nutzer bedeutet das keine Möglichkeit zur Anpassung von Dashboard-Items.

¹Capability Maturity Model Integration

Variabilität ist ein fortwährendes Forschungsthema im Bereich des Software Engineering. Insbesondere die Produktlinienentwicklung hat sich als methodischer Ansatz für die Softwareentwicklung bewährt. Im Wesentlichen geht es bei Variabilität um die Wiederverwendung von Wissen und Erfahrung. Auch für die Spezifikation von Metriken, wird die Einbeziehung von Variabilität als erfolgversprechend angesehen [58].

1.1. Zielsetzung

Das MeDIC-Dashboard sollte die Anpassung der dargestellten Metriken erlauben, damit ein Projektleiter während der Entwicklungsphase die Möglichkeit hat, auf sich ändernde Informationsbedürfnisse zu reagieren. Die Zielsetzung dieser Arbeit bestand darin, die Möglichkeiten der Metrik-Variabilität im Umfeld von MeDIC-Dashboard zu untersuchen und das Werkzeug um die Funktionalität zu erweitern die Dashboard-Items entsprechend modifizieren zu können. Dazu sollte eine Literaturrecherche zum Thema Variabilität durchgeführt werden und die Arbeitsweise bestehender Lösungen untersucht werden. Anschließend sollte ein Modell konzipiert werden, welches die Definition von Metrik-Variabilität erlaubt und in die Kommunikation zwischen Rechenkern und Dashboard integriert ist. Des Weiteren sollte aus Sicht der Entwickler und Metrik-Experten die Definition von Variabilität ohne übermäßigen Aufwand zu bewerkstelligen sein. Am Ende der konzeptionellen Arbeit sollten die erarbeiteten Konzepte anhand eines Prototypen realisiert und im Rahmen einer Evaluierung bewertet werden.

1.2. Aufbau

Der Aufbau der Arbeit gliedert sich in insgesamt neun Kapitel.

Nach diesem Abschnitt folgt in Kapitel 2 eine Übersicht über die technischen Grundlagen. Kernpunkte sind die Funktionsweise des Java-EE-Anwendungsserver und verschiedener Webservice-Technologien.

Anschließend folgt in Kapitel 3 eine Übersicht über die aktuelle Systemlandschaft von *MeDIC* und der zugrundeliegenden Messarchitektur in Form der *Enterprise Measurement Infrastructure* (EMI).

In Kapitel 4 werden verschiedene Modellierungskonzepte von Variabilität behandelt. Auf die grundlegenden Prinzipien der Produktlinienvariabilität folgt, unterstützt durch Beispiele, eine Erläuterung des Aufbaus von Feature-Diagrammen und des orthogonalen Variabilitätsmodells.

Kapitel 5 beschäftigt sich mit Vorgänger-Arbeiten zum Thema Variabilität und der Dashboard-Software Geckoboard. Hier wird speziell auf den Widget-Editor eingegangen,

mit dem es möglich ist Dashboard-Widgets mit eigener Variabilität zu definieren.

Darauf folgt mit Kapitel 6 die Konzeption verschiedener Papierprototypen und des Modells für die Webschnittstelle.

In Kapitel 7 folgt ein Überblick über die technische Integration der erarbeiteten Lösungsansätze.

Anschließend erfolgt in Kapitel 8 die Auswertung einer zweigeteilten Evaluation des Architekturmodells und des implementierten Prototyps und des konzipierten Papierprototyps.

Schließlich wird in Kapitel 9 eine Fazit der Ergebnisse und ein Ausblick für mögliche Weiterentwicklungen gegeben.

9. Fazit und Ausblick

Leave the world better than you
found it.

*(Robert Baden-Powell, Gründer
der Pfadfinderbewegung)*

Nach einer Einführung in die verwendeten Technologien und die Systemlandschaft von MeDIC wurden verschiedene Konzepte zur Modellierung von Variabilität in der Softwareentwicklung beleuchtet. Insbesondere die Theorie des orthogonalen Variabilitätsmodells hat eine bedeutende Rolle für die spätere Konzeption gespielt. Durch das Studium verwandter Literatur und der Analyse der Dashboard-Software Geckoboard konnten wichtige Erkenntnisse über den Einsatz von Variabilität im Zusammenhang mit Metrik-Spezifikationen und Dashboard-Software gewonnen werden.

Nachdem eine erste Liste der Interessen der identifizierten Stakeholder aufgestellt wurde, konnte mit der Konzeption eines Papierprototyps begonnen werden. In einer iterativen Vorgehensweise unter Einbeziehung von MeDIC-Projektpartnern aus der Industrie wurden die Interessen stetig verfeinert und der Papierprototyp kundenspezifisch angepasst.

Nach einer ausführlichen Anforderungsanalyse für die Webschnittstelle der Rechenkerne wurde ein Konzept zur Kommunikation zwischen den EMI-Schichten „Berechnung und Persistierung“ und „Visualisierung“ erarbeitet. Den Kern des Konzepts bildet ein Variabilitätsmodell zur Beschreibung von Variationspunkten und Varianten. Das Modell erlaubt zum einen die Definition von vordefinierten Varianten, die mit einem geschlossenen Variationspunkt assoziiert sind. Zum anderen unterstützt ein offener Variationspunkt die Parametrisierung des Datentyps und somit alle Varianten des Wertebereichs dieses Datentyps. Darüber hinaus ist es möglich mithilfe von Beschränkungen (Constraints) Bedingungen für die Auswahl von Varianten festzulegen.

Für die Umsetzung der Webschnittstelle wurde das Modell eines RESTful Webservice implementiert. Als Datenaustauschformat wurde JSON gewählt und nach einer Analyse der Leistungsmerkmale eine geeignete Java-Bibliothek zur Serialisierung und Deserialisierung ausgewählt. Die Realisierung der korrekten Deserialisierung des Variabilitätsmodells erwies sich als schwierig, da abstrakte Klassen und parametrisierte Typen nicht automatisch verarbeitet werden können. Nach ein paar Modellanpassungen und der Implementierung eines benutzerdefinierten Deserialisierers für die abstrakte Klasse `VariationPoint` konnte mit der Entwicklung eines Einstellungsdialogs für einen Dashboard-Container begonnen werden.

Der realisierte Einstellungsdialog basiert technisch auf einer Composite Component, welche für alle Dashboard-Items und die damit assoziierten Varianten, entsprechende Eingabefelder generiert. Zusätzlich wurde eine automatische Diagramm-Vorschau implementiert und dessen Funktionsweise eingehend beschrieben.

Für die Evaluation der Forschungsergebnisse wurden zwei Befragungen durchgeführt. Eine unter MeDIC-Entwicklern zur Bewertung der Architektur und eine weitere unter MeDIC-Nutzern zur Evaluation des Konzepts und des implementierten Prototyps.

Das Architekturmodell wurde grundsätzlich als positiv empfunden. Gelobt wurden die Einfachheit und die schnelle Integration neuer Varianten für einen Rechenkern.

Der Prototyp des Einstellungsdialogs hat ebenfalls einen guten Eindruck hinterlassen. Als besonders angenehm wurde die Eigenschaft wahrgenommen, dass Änderungen an den Eingabefeldern sofort im Diagramm reflektiert werden. Als Kritikpunkte wurden die etwas kryptischen Fehlermeldungen genannt sowie potentiell lange Listen von Variationspunkten durch die man durchscrollen muss.

Der Widget-Ansatz des Konzept konnte im Rahmen dieser Arbeit nicht umgesetzt werden. Weiterhin war das Konzept eines Metrik-Frameworks angedacht, welches kontextuell zwischen Dashboard und Rechenkern sitzt. Hier wäre eine Trennung von darstellungs- und metrik-spezifischer Variabilität möglich. Außerdem soll ein Metrikexperte mehr Möglichkeiten haben benutzerspezifische Variabilität zu definieren. Eine Idee ist die Definition von Variabilitäts-Voreinstellungen der Rechenkerne in Form von (Einstellungs-)Paketen, die als neue Varianten definiert werden können. Außerdem soll es möglich sein Varianten und Variabilität vor dem Benutzer zu verstecken, um die Komplexität für diesen abzuschwächen. Ein offenes Problem ist die Handhabung von Informationsbedürfnissen, die mit einem Dashboard-Item verknüpft sind. Die Auswahl neuer Varianten erfordert unter Umständen eine neue Verknüpfung der Informationsbedürfnissen.

Die Masterarbeit von Thomas Röllig [46] versucht den Ansatz des Metrik-Frameworks weiterzuführen und zu erforschen in welcher Form ein Widget-Konzept in die Architektur von MeDIC integriert werden kann.