

Smarter GQM-Editor mit verringerter Einstiegshürde

Raphael Pham, Kurt Schneider, Leibniz Universität Hannover

{Raphael.Pham, Kurt.Schneider}@inf.uni-hannover.de

Zusammenfassung

Die GQM Methode ist bereits ausreichend definiert. Man muss sie jedoch selbst einmal angewendet haben, um sie durchdringen zu können. Dabei können sogenannte *Abstraction Sheets* unterstützend eingesetzt werden. In der Lehre an der Leibniz Universität Hannover hat sich gezeigt, dass das Ausfüllen dieser Formulare vielen Studenten Probleme bereitet. Sie sind von Abstraction Sheets überfordert oder erkennen den Sinn darin nicht. Als Folge lehnen die Studenten die GQM Methode ab und erlernen sie nicht. Wir stellen einen smarten Editor vor, welcher konkret die Erstellung des Abstraction Sheets und die Entwicklung von zugehörigen Metriken unterstützt. Der Anwender wird mit prozess-gesteuerten Fragestellungen zum korrekten Ausfüllen beider Formulare angeregt. Unerfahrene Anwender können den vollen Unterstützungsumfang des smarten Editors ausschöpfen und so an die GQM Methode herangeführt werden. Erfahrene Benutzer können den Editor verwenden, um Abstraction Sheets, Questions und Metriken zu erstellen und zu verwalten.

Motivation

Ein Fallstrick beim Messen von Software Entwicklungsprozessen oder Produkten ist, dass man das misst, was einfach zu messen ist - und nicht das, was man messen sollte, um aussagekräftige Ergebnisse zu erhalten (Basili, 1992). Man wendet sich an bereits bekannte oder einfach zu erhebende Metriken. Hier verwendet die GQM Methode einen Top-Down Ansatz: Anstatt mittels vorhandener Werkzeuge (z.B. bekannte Metriken) projekt-individuelle Ziele zu messen (Bottom-Up), werden diese Ziele konkretisiert und durch eigens entwickelte Metriken operationalisiert. Dies soll gewährleisten, dass tatsächlich gemessen wird, was auch gemessen werden soll - und auch nicht mehr.

Grundlegende Ideen zu GQM entstanden während einer Untersuchung am Goddard Space Flight Center (Basili u. Weiss, 1984). Später stellt (Basili, 1992) das GQM Paradigma vor und (Van Latum u. a., 1998) erweitern die GQM Methode um sogenannte *Abstraction Sheets*. Seit seiner Einführung konnte die GQM Methode erfolgreich in verschiedenen Industrieprojekten eingesetzt werden. (Van Latum u. a., 1998) und (Birk

u. a., 1998) beschreiben den Einsatz bei Schlumberger RPS. (Gantner u. Schneider, 2003) haben den GQM-Prozess in zwei Fällen aus der Automobilindustrie angewendet und beschreiben eingehend den Einsatz von Abstraction Sheets. (Kilpi, 2001) setzt eine angepasste GQM Variante bei Nokia ein. (Van Solingen u. Berghout, 1999) bietet eine eingehende Betrachtung von GQM samt Fallbeispielen.

Dem Erfolg von GQM in der Praxis folgend, wird die GQM Methode auch an der Leibniz Universität Hannover gelehrt. Dem Thema GQM ist ein Kapitel der Vorlesung zur Software-Qualität gewidmet und es wird auch im dazugehörigen Übungsbetrieb behandelt. Erfahrungsgemäß lässt sich ein besserer Lernerfolg erzielen, wenn GQM praktisch angewendet wird. Das Konzept von Abstraction Sheets und deren korrekte Erstellung bereitet den Studierenden jedoch öfter Schwierigkeiten - obwohl der Prozess zum Erstellen eines Abstraction Sheets klar vorgegeben ist. Studierende zeigten wiederholt Schwierigkeiten Abstraction Sheets korrekt auszufüllen. Das mühsame Aufzeichnen der Formulare per Hand könnte eine zusätzlich Hürde darstellen. In dieser Arbeit stellen wir einen Editor für Abstraction Sheets vor, der Studierende mit Hilfsmechanismen aktiv beim Anwenden der GQM Methode unterstützt. Der unsichere Anwender wird mit gezielten Fragestellungen durch den Ausfüllprozess geführt. Studierenden soll so eine mögliche Ablehnung gegenüber der komplex anmutenden GQM Methode genommen werden. Studierende sollen mit diesem Werkzeug Gelegenheit erhalten, das korrekte Ausfüllen von Abstraction Sheets zu üben. (Werner, 2012) hat die in dieser Arbeit vorgestellten Konzepte in einem lauffähigen Webtool realisiert.

Die GQM Methode

Die GQM Methode wurde bereits in zahlreichen Veröffentlichungen behandelt. Im folgenden wird ein kurzer Überblick über die GQM Methode gegeben (in der Form, in welcher diese an der Leibniz Universität Hannover gelehrt wird (Schneider, 2007)), um den Wirkungsbereich unseres GQM-Editors aufzuzeigen.

Der grundlegende Ablauf der GQM Methode sieht die schrittweise Erstellung eines GQM-Modells (auch GQM-Baum genannt) vor: Abstraktere Ziele werden

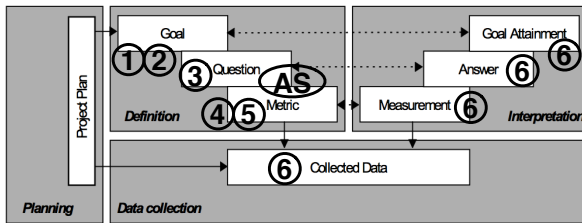


Abbildung 1: GQM Methode an der Leibniz Universität Hannover, basierend auf (Van Solingen u. Berghout, 1999)

in konkrete Subziele (Goal) aufgeteilt. Subziele werden durch beschreibende Fragen (Question) scharf und zweckmäßig umrissen. Zur Beantwortung der charakterisierenden Fragen werden Metriken (Metric) erstellt, die den Erfüllungsgrad der Fragen und somit den Erreichungsgrad des Ziels angeben. Abb. 2 zeigt ein solches GQM-Modell.

Zu Lehrzwecken spalten wir Teile der GQM Methode auf und definieren sechs Schritte:

1. Ziele erheben und verfeinern (Goal)
2. Facettenbeschreibung der Ziele (Goal)
3. Ableitung von charakterisierenden Fragen (Question) zu den Zielen
4. Ableitung von zugehörigen Metriken, (Metric)
5. Messplan für eigentliche Datenerhebung erstellen (Metric)
6. Datenerhebung und Auswertung (Collection, Interpretation)

Die Einordnung dieser Schritte in das umfangreiche Prozessmodell der GQM Methode nach (Van Solingen u. Berghout, 1999) ist in Abb. 1 eingezeichnet.

Schritt 1 produziert einen sogenannten Zielbaum, welcher ein high-level Messziel auf konkretere Subziele verfeinert und in Beziehung miteinander setzt. Aus diesem Zielbaum werden Subziele zur weiteren Bearbeitung ausgewählt. Die Facettendarstellung des Messziels (Schritt 2) forciert die Auseinandersetzung mit verschiedenen Aspekten eines Ziels und regt zur abgewandelten Betrachtung an.

Für die Ableitung von charakterisierenden Fragen zu einem Messziel (nun vorliegend als Zielfacette, Schritt 3) wird ein spezielles Formular eingesetzt, das sogenannte *Abstraction Sheets* (Van Latum u. a., 1998), (Van Solingen u. Berghout, 1999), (Gantner u. Schneider, 2003). Es bietet eine festgelegte Struktur. Mittels vier vordefinierten Fragestellungen baut der Anwender systematisch ein Modell des zu untersuchenden Messziels auf und gelangt zu tiefgreifendem Verständnis. Der Qualitätsaspekt des Betrachtungsgegenstandes (z.B. "Verständlichkeit" von "Quellcode") wird auf konkrete und projektindividuelle Eigenschaften

heruntergebrochen. Diese heißen *Qualitätsfaktoren* des Qualitätsaspektes. Ziel ist es, für jeden konkreten Aspekt des Qualitätsaspektes einen beeinflussenden Faktor zu finden (sogenannte *Einflussfaktoren*). Beide gefundenen Faktoren sind über eine Einflusshypothese zu verbinden. Nun ergibt sich eine das Messziel charakterisierende Frage (Question) als Zustandsabfrage von Qualitätsfaktoren.

Die aufgestellte Einflusshypothese wird in Schritt 4 und 5 überprüft. Dabei wird eine Metrik zur Zustandsabfrage des Qualitätsfaktors basierend auf dem Zustand des Einflussfaktoren entwickelt: Es werden verschiedene (mögliche) Ausprägungen für den Einflussfaktoren ermittelt und jeweils der Einfluss auf den Qualitätsfaktor gemessen. Für diesen Vorgang wird ein eigenes Tabellentemplate verwendet.

Die einzelnen Prozessschritte samt Formularen skizziert Abb. 2. Eingehende Betrachtungen der GQM Methode finden sich in (Schneider, 2007) oder auch (Van Solingen u. Berghout, 1999).

Der GQM-Editor nimmt ein Facettenziel entgegen und unterstützt maßgeblich die Erstellung eines zugehörigen Abstraction Sheets und zugehöriger Messpläne (Schritt 3 und Schritt 4). Die Erstellung und Verfeinerung von Messzielen oder deren Darstellung als Facette werden nicht unterstützt (Schritt 1 und Schritt 2). Beide Schritte lassen sich relativ schnell auf Papier bewältigen und erfordern (gemäß Lehrerfahrungen) weniger Unterstützung. Auch die eigentliche Durchführung und anschließende Interpretation werden nicht von unserem GQM-Editor unterstützt.

Das Abstraction Sheet

Das Abstraction Sheet (AS) bietet dem Anwender eine streng definierte Struktur um methodisch und zielsicher charakterisierende Fragen zu einem gegebenen Messziel abzuleiten. Dabei ist die Ausfüllreihenfolge der Quadranten vorgegeben und bildet ein U. Jeder Quadrant verlangt dem Anwender ab, bestimmte Aspekte des Messziels anzugeben. Auf diese Weise wird der abstrakte Qualitätsaspekt des Messziels in Qualitätsfaktoren konkretisiert, beeinflussende Faktoren (Einflussfaktoren) identifiziert und zugehörige Einflusshypthesen aufgestellt. Begonnen wird ein AS mit dem Eintragen des Messziels in Facettenform (oberer Teil von 3), welche aus dem vorigen Prozessschritt 2 übernommen wurde. Der Ablauf im Detail:

1. Oben, links: Der Qualitätsaspekt wird auf konkrete Eigenschaften des Betrachtungsgegenstandes konkretisiert, sogenannte *Qualitätsfaktoren*. Welche Merkmale machen den Qualitätsaspekt in diesem Projekt ganz konkret aus? In Fall von Abb. 3: Verständlichkeit von Quellcode bedeutet für den Ausfüllenden z.B. die Verzweigungstiefe.
2. Unten, links: Zu jedem Qualitätsfaktor wird eine *Ausgangshypothese* aufgestellt, wie es zu Messbe-

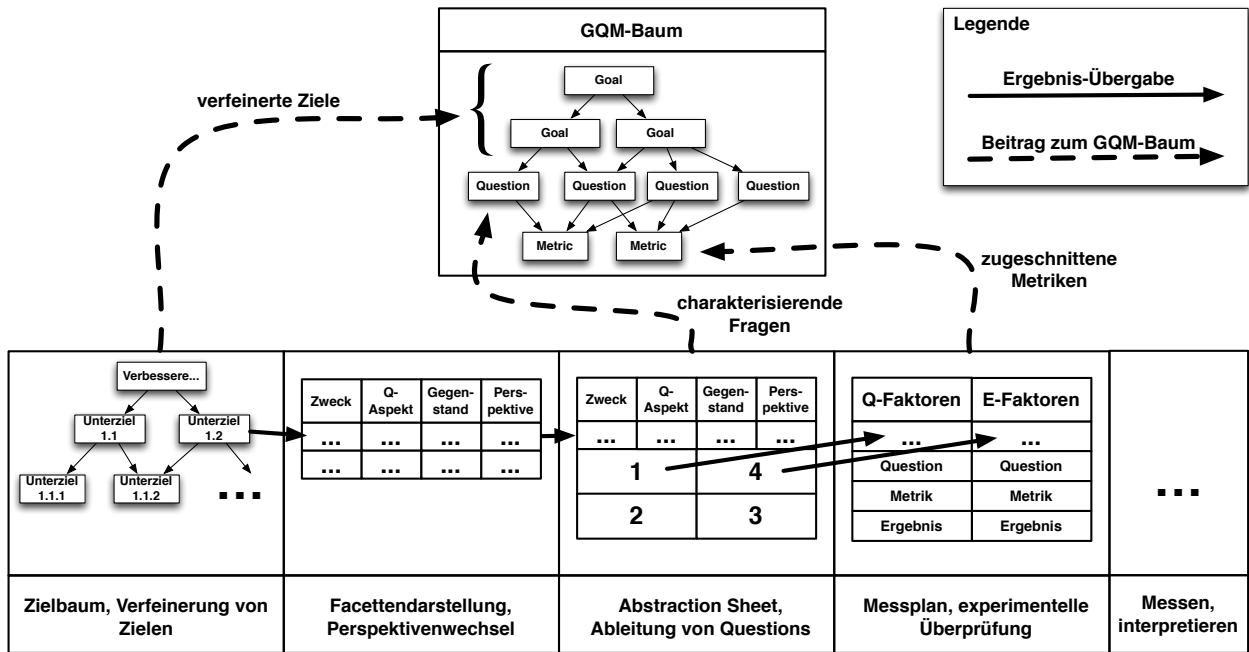


Abbildung 2: Übersicht über die GQM Methode, Einordnung der Prozessschritte und Formulare

ginn darum steht. Alternativ können auch nachgemessene Werte eingetragen werden.

3. Unten, rechts: Der Anwender stellt *Einflussypothesen* für jeden Qualitätsfaktor auf: Wie könnte der Qualitätsfaktor zu beeinflussen sein?
4. Oben, rechts: Aus den Einflussypothesen werden *Einflussfaktoren* extrahiert.

Abb. 3 zeigt ein ausgefülltes AS. Obwohl die Anforderungen an den Inhalt eines Quadranten klar definiert sind, bereitet es Studenten öfter Schwierigkeiten, korrekte und brauchbare AS zu produzieren.

Auf einen Studierenden, der das Konzept des Abstraction Sheets noch nicht vollständig durchdrungen hat, kann dieses Formular abschreckend und komplex wirken. Zählt man das Eintragen des Messziels mit, so werden mehr als vier verschiedene Abfragen an den Studierenden gestellt. Hinzu kommt, dass diese Felder konzeptbedingt jeweils eng gefasste Ziele verfolgen und korrekte Antworten für die erfolgreiche Anwendung voraussetzen: Werden z.B. im ersten Quadranten die Qualitätsfaktoren nicht mit konkretisierenden Attributen belegt — sondern bleiben so abstrakt wie das Messziel — so führt das Abstraction Sheet nicht zu dem gewünschten Ergebnis. In Abb. 3 würde der ungeübte Studierende als ersten Qualitätsaspekt möglicherweise einfach ein Synonym für das Messziel eintragen, z.B. Verständnis. Damit ist in diesem Schritt jedoch nichts gewonnen. Der Studierende füllt die nachfolgenden Felder aus — und kommt dennoch nicht zum erwarteten Ergebnis und ist enttäuscht. Damit dieser Fall nicht eintritt, bietet unser Editor dem Studierenden gezielte Hilfestellung.

Durch kontextabhängige, konkret formulierte Fragestellungen wird der Anwender durch das Abstraction Sheet geführt.

Auch wenn der Hauptgrund der nicht-erfolgreichen Anwendung des Abstraction Sheets auf Unverständnis des zugrundeliegenden Konzeptes zurückzuführen ist, so kann eine derart geführte (wiederholte) Anwendung zur Durchdringung des Konzeptes führen bzw. beitragen.

Der smarte GQM-Editor

Die Inhalte der Quadranten 1 bis 4 sind abhängig von einander: Inhalte werden von Quadrant zu Quadrant weitergegeben und verarbeitet. Z.B.: Hat der Anwender in Quadrant 1 das Qualitätsziel auf zwei konkrete Faktoren heruntergebrochen, so tauchen diese beiden Faktoren mit einer zusätzlichen Statureinschätzung (genauer: Ausgangshypothese) im zweiten Quadranten wieder auf. Dieses Beispiel zeigt zwei Ebenen, auf denen der Anwender agieren muss, um das AS auszufüllen: Zum einen muss er einen kreativen Schritt tätigen und das Q-Ziel konkretisieren - dabei muss er genau wissen, was gerade von ihm verlangt wird. Zum anderen muss er die Einträge manuell in den nächsten Schritt übertragen und dort weiterverarbeiten. An diesen Stellen setzt unser GQM-Editor an: Der Anwender wird bei einem kreativen Vorgang durch eine gezielte Fragestellung als dezenter Hinweistext unterstützt (*Grauer Hilfstext*). Weiterhin übernimmt der GQM-Editor manuelle Schritte, überträgt Zwischenergebnisse in Folge-Quadranten und versucht damit die nächsten Einträge teilweise zu generieren (*Automatischer Hilfstext*). Beide Unterstützungsvarianten

| Zweck: | Qualitätsaspekt: | Betrachtungsgegenstand: | Perspektive und Umgebung |
|------------|------------------|-------------------------|--------------------------|
| Verbessere | Verständlichkeit | Quellcode | Entwickler |

| | |
|--|---|
| <p>Qualitätsfaktoren: Hilfestufe: Grauer Text ▾</p> <p>Welche Faktoren definieren den Qualitätsaspekt?</p> <p>1) Verzweigungstiefe Vorschläge</p> <p>2) Aussagekraft der Variablennamen Vorschläge</p> <p>Weiteres Eingabefeld</p> | <p>Einflussfaktoren: Hilfestufe: Grauer Text ▾</p> <p>Was hat Einfluss auf die Qualitätsfaktoren?</p> <p>1) Auslagerung von Funktionen Vorschläge</p> <p>2) Welcher Faktor hat laut der aufgestellten Einflusshypothese Einfluss auf die/den "Aussagekraft der Variablennamen"? Vorschläge</p> <p>Weiteres Eingabefeld</p> |
| <p>Ausgangshypothesen: Hilfestufe: Grauer Text + Automatischer Text ▾</p> <p>Momentane Erwartung über die Qualitätsfaktoren?</p> <p>1) "Verzweigungstiefe" ist/sind momentan zu hoch. Momentan beträgt "Verzweigungstiefe": über 15 McCabe-Wert pro Klasse Vorschläge</p> <p>2) "Aussagekraft der Variablennamen" ist/sind momentan zu niedrig. Mind. 20% der Variablennamen werden nicht eindeutig abgekürzt. Vorschläge</p> <p>Weiteres Eingabefeld</p> | <p>Einflusshypothese: Hilfestufe: Grauer Text + Automatischer Text ▾</p> <p>Was beeinflusst wie die Qualitätsfaktoren?</p> <p>1) Der Qualitätsfaktor "Verzweigungstiefe" wird beeinflusst durch: Anwendung von Auslagerungsmechanismen Vorschläge</p> <p>2) Der Qualitätsfaktor "Aussagekraft der Variablennamen" wird beeinflusst durch: einheitliche Konventionen, <u>Sensibilisierung</u> der Entwickler Vorschläge</p> <p>Weiteres Eingabefeld</p> |

Abbildung 3: Ein ausgefülltes Abstraction Sheet im GQM-Editor

werden dynamisch anhand von vorhergegangenen Eingaben im AS generiert. Auf diese Weise lassen sie eine kontextbezogene Unterstützung zu. Dies ist eine der Stärken des GQM-Editors von denen besonders unerfahrene Anwender profitieren können. Beide Unterstützungstexte werden als vorgeschlagene Antwortträge in den Quadranten eingeblendet — die der Anwender nach Belieben ändern und anpassen kann.

Für erfahrene Benutzer können die beiden Unterstützungsmechanismen ausgeschaltet werden, sodass der GQM-Editor als reiner Editor fungiert. Alle drei Mechanismen (keine Hilfe, grauer Hilfstext, automatischer Hilfstext) können für jeden Quadranten individuell eingestellt werden (siehe Dropdownen in den Quadranten, Abb. 3).

Die Hilfstexte haben eine weitere Funktion: Sie erscheinen in der Reihenfolge, in der das AS ausgefüllt werden sollte. Jeweils der nächst-auszufüllende Quadrant wird mit grauem oder schwarzem Hilfstext gefüllt. Dies soll den Anwender durch den Ausfüllprozess leiten. Da es sich bei den Hilfstexten nur um Vorschläge handelt, kann der Anwender aber auch gegen diese Reihenfolge verstoßen und z.B. die voreingetragenen Werte entfernen.

Grauer Hilfstext

Das Einblenden von grauem Hilfstext ist die erste Unterstützungsstufe unseres GQM-Editors und für unerfahrene Anwender gedacht. Der Hilfstext soll dem Anwender an geeigneter Stelle ins Gedächtnis rufen, was in dem aktuellen Quadranten einzutragen ist. Er

| |
|---|
| <p>Qualitätsfaktoren: Hilfestufe: Grauer Text ▾</p> <p>Welche Faktoren definieren den Qualitätsaspekt?</p> <p>1) Verzweigungstiefe Vorschläge</p> <p>2) Was ist für einen "Entwickler" ein Merkmal für "Verständlichkeit", wenn es um "Quellcode" geht? Vorschläge</p> <p>Weiteres Eingabefeld</p> |
|---|

Abbildung 4: Hilfestellung im Quadrant 1

dient als Gedankenstütze, um die Aufgabe klar zu definieren: Der Anwender wird explizit — und angepasst an vorherige Eingaben — nach dem einzutragenden Inhalt gefragt. So erscheint der graue Hilfstext im ersten Quadranten, sobald der Anwender das Messziel als Zielfacetten eingetragen hat. Die eingetragenen Werte werden dynamisch im Text verarbeitet und sind kontextualisiert: Anhand der eingegebenen Zielfacetten in Abb. 3 (gelb unterlegt) generiert der Editor den grauen Hilfstext: "Was ist für einen Entwickler ein Merkmal für Verständlichkeit, wenn es um Quellcode geht?" (siehe Abb. 4).

Sobald der Anwender auf ein Feld mit grauem Hilfstext klickt und einen Eintrag vornehmen will, verschwindet dieser und das Textfeld ist für den Anwender freigegeben. Die graue Farbe wurde bewusst dem Anschein eines flüchtigen Tooltips nachempfunden.

Ausgangshypothesen: Hilfestufe: Grauer Text + Automatischer Text

Momentane Erwartung über die Qualitätsfaktoren?

1) "Verzweigungstiefe" ist/sind momentan zu hoch. Momentan beträgt "Verzweigungstiefe": über 15 McCabe-Wert pro Klasse Vorschläge

2) "Aussagekraft der Variablenamen" ist/sind momentan zu niedrig. Mind. 20% der Variablenamen werden nicht eindeutig abgekürzt. Vorschläge

Weiteres Eingabefeld

Einflusshypothese: Hilfestufe: Grauer Text + Automatischer Text

Was beeinflusst wie die Qualitätsfaktoren?

1) Der Qualitätsfaktor "Verzweigungstiefe" wird beeinflusst durch: <bitte Wert eintragen> Vorschläge

2) Welcher Faktor könnte Einfluss auf die/den "Aussagekraft der Variablenamen" haben und wie sieht dieser Einfluss aus? Vorschläge

Weiteres Eingabefeld

Abbildung 5: Hilfestellung im Quadranten 2 und 3

Automatischer Hilfstext

Die zweite Unterstützungsstufe wendet sich an sehr unerfahrene Anwender: Der GQM-Editor generiert Teile des Eintrags als schwarzen Text vor und lässt gezielt Lücken, die der Anwender ausfüllen soll. Dies lenkt die Gedanken der Anwender in gezielte Bahnen und es werden konkrete Werte verlangt.

Das Ausfüllen von Quadrant 1 bedarf eines kreativen Schrittes, der durch den grauen Hilfstext unterstützt wird. Teile von Antworten zu generieren ist hier nicht möglich. Quadrant 2 (Ausgangshypothese) bietet jedoch die Möglichkeit, die Q-Faktoren automatisch in die rudimentäre Form einer Hypothese zu überführen. Dabei kann der Aussagenkern der Hypothese nicht generiert werden, da dies wieder eines kreativen Schrittes bedarf. Jedoch wird das Eintragen dieses Aussagenkerns durch einen gezielten Lückentext unterstützt:

Bevor der Anwender in das Feld zum Eintragen der ersten Ausgangshypothese klickt, hat der Editor dort in grau eingetragen: "Wie steht es momentan um die/den "Verzweigungstiefe"? Welche Erfahrungswerte, Vermutungen und/oder Messwerte gibt es zu diesem Qualitätsfaktor?" Dies erinnert den Anwender an die von ihm geforderte Ausgangshypothese. Sobald er in das Feld klickt, erscheint die folgende vorgenerierte Antwort in schwarzem und editierbarem Text: "Verzweigungstiefe ist/sind momentan zu <bitte Eigenschaft eintragen>. Momentan beträgt Verzweigungstiefe: <bitte Wert eintragen>." Der Benutzer kann diese Vorlage verwenden, um seine Ausgangshypothese zu verfassen oder eine eigene formulieren (s. Abb. 5).

Ähnlich lässt sich mit Quadrant 3 verfahren (siehe Abb. 5), jedoch nicht mit Quadrant 4: Die Einflussfaktoren zuverlässig aus den resultierenden Einflusshypothese zu extrahieren, ist aufgrund von sprachlichen Abweichungen und Eigenarten schwierig.

Letztendlich lässt sich diese zweite Unterstützungsstufe nur für den zweiten Quadranten (Ausgangshypothese) und dritten Quadranten (Einflusshypothese) einschalten. Quadranten 1 und 4 können lediglich die erste Unterstützungsstufe einsetzen.

Da möglicherweise die generierte Antwort nicht zu dem aktuellen Messprojekt passt, lässt sich diese generierte Antwort wie ein normales Textfeld bearbeiten und auch löschen. In Abb. 5 hat der Anwender die zweite Ausgangshypothese angepasst.

Der Messplan

Nachdem der Anwender mithilfe des Abstraction Sheets Einflussfaktoren der Qualitätsfaktoren und zugehörige Einflusshypothese aufgestellt hat, folgt der Schritt der experimentellen Überprüfung eines solchen Zusammenhangs. Der sogenannte *Messplan* fordert den Anwender auf, verschiedene Ausprägungen des Einflussfaktors zu ermitteln und dann zu überprüfen, wie sich der Qualitätsfaktor verhält. Auf diese Weise soll die Einflusshypothese validiert bzw. falsche Annahmen aufgedeckt werden.

Die beiden Faktoren werden automatisch aus dem Abstraction Sheet übernommen und bleiben an der selben Stelle stehen. Dies soll Verwirrungen beim unerfahrenen Anwender vermeiden. Bedingt durch die Idee der experimentellen Überprüfung muss der Anwender mit der rechten Seite des Messplans beginnen und zuerst den Einflussfaktor betrachten. Da dies etwas unintuitiv erscheinen mag, wird standardmäßig die linke Seite mit einem Hinweis ausgeblendet (siehe Abb. 6). Persönliche Workflows können auch realisiert werden; der Hinweis lässt sich ausschalten.

Jede Seite des Messplans gliedert sich in drei Teile, die nacheinander ausgefüllt werden.

1. Frage nach den möglichen Ausprägungen des Faktors.
2. Handlungsanweisungen, diese Ausprägungen zu ermitteln.
3. Ausprägungen ermitteln und eintragen.

Die Formulierung der einleitenden Frage nach den Ausprägungen wird mit grauem Hilfstext bzw. schwarzem Lückentext unterstützt (siehe Abb. 6, beide Texte in den Feldern "Frage" sind generiert). Jedoch sind Schritt 2 und 3 stark projektabhängig und individuell. Dementsprechend kann der GQM-Editor diese nicht

Hilfestufe: Grauer Text + Automatischer Text

| | |
|--|--|
| Qualitätsfaktor Verzweigungstiefe | Einflussfaktor Auslagerung von Funktionen |
| Frage Welche Frage in Bezug zum Qualitätsfaktor "Verzweigungstiefe" möchtest du mit dem Messplan beantworten? | Frage Welche verschiedenen Ausprägungen kann der Einflussfaktor "Auslagerung von Funktionen" annehmen? |
| Messplan/Metrik Welche Schritte sollen durchgeführt werden, um die Frage für den Qualitätsfaktor zu beantworten? Vorschläge | Messplan/Metrik Welche Schritte sollen durchgeführt werden, um verschiedenen Ausprägungen des Einflussfaktor zu ermitteln? Vorschläge |
| 1. Schritt: <input type="text"/> 2. Schritt: <input type="text"/> 3. Schritt: <input type="text"/> Weiteres Eingabefeld: <input type="text"/> | 1. Schritt: <input type="text"/> 2. Schritt: <input type="text"/> 3. Schritt: <input type="text"/> Weiteres Eingabefeld: <input type="text"/> |
| Ergebnisse Linke Seite freigeben! Welche Ergebnisse erhalten Sie beim Messplan, wenn der Einflussfaktor die verschiedenen Ausprägungen annimmt? | Ergebnisse Welche verschiedenen Ausprägungen haben sie für den Einflussfaktor ermittelt? |
| Ergebnis, mit 1. Ausprägung: <input type="text"/> Ergebnis, mit 2. Ausprägung: <input type="text"/> Ergebnis, mit 3. Ausprägung: <input type="text"/> | 1. Ausprägung: <input type="text"/> 2. Ausprägung: <input type="text"/> 3. Ausprägung: <input type="text"/> |

Abbildung 6: Linke Seite des Messplans wird zu Beginn ausgeblendet.

mit automatisch generierten Antworten unterstützen, bietet jedoch hier die erste Hilfestufe an (grauer Text als Gedankenanstoß). Weiterhin gibt der graue Hinweistext dem unerfahrenen Benutzer die Reihenfolge zum Ausfüllen vor. Jede Eingetragene Ausprägung für den Einflussfaktor ist einem Eintragungsfeld für eine Ausprägung gegenübergestellt. Auf diese Weise soll dem Anwender die Gegenüberstellung und der Effekt der verschiedenen Faktor-Ausprägungen vereinfacht werden.

Verwandte Arbeiten

Zur Unterstützung der GQM Methode wurden bereits einige Tools entwickelt. Jedoch ist keines der untersuchten Tools fokussiert auf unerfahrene Benutzer oder bietet Hilfestellung beim eigentlichen Ausfüllen von Abstraction Sheets und Messplänen.

Die Entwicklung von GQMAspect wurde an der Universität Kaiserslautern begonnen und später am Fraunhofer IESE (Institut für experimentelles Software Engineering) weitergeführt (Hoffmann u. a., 1997), (Voigtlaender, 1999). GQMAspect unterstützt die gesamte GQM Methode: Aufnahme und Verwaltung der Goals, Erstellen von Abstraction Sheet mittels Editor bis hin zur Verwaltung von Messplänen. Mit GQMAspect lassen sich Beziehungen zwischen einzelnen Produkten der Arbeitsphasen der GQM Methode herstellen (Ein Goal wird einem AS zugewiesen, etc) und es können Vollständigkeitsabfragen gemacht werden (wurde jedes definierte Goal mit einem AS weiterverfolgt?). Der Ansatz unseres GQM-Editors unterstützt hier lediglich

die Erstellung von Abstraction Sheets und Messplänen — Goals werden nicht nennenswert behandelt. Im Gegensatz zu unserem GQM-Editor bietet GQMAspect dem Anwender jedoch wenig Unterstützung, wenn es um das eigentliche Erstellen von neuen GQM-Inhalten geht. Anwendergruppe von GQMAspect sind erfahrene GQM-User, während GQM-Editor für den Lehrgebrauch bzw. für unerfahrene Anwender konstruiert wurde.

GQM-DIVA (GQM-Definition Interpretation Validation) wurde im Rahmen einer Diplomarbeit an der Universität Kaiserslautern entwickelt (van Maris u. a., 1995). Das Tool überprüft die Konsistenz der Einträge eines GQM-Baumes (hier genannt: GQM-Plan), damit keine fehlerhaften Daten weiterverwendet werden. Dieser GQM-Baum kann mit fiktiven Messwerten in Simulation ausgewertet werden. Damit sollen Lücken (fehlende Daten) im GQM-Baum aufgedeckt werden. GQM-Diva benötigt ein fertig ausgefülltes Abstraction Sheet als Eingabe. Dessen Erstellung wird nicht unterstützt. Auch die Erstellung eines Messplans wird von GQM-DIVA nicht abgedeckt.

(Chen u. a., 2003) stellen mit dem Multi-Agent System ISMS (Intelligent Software Measurement System) eine Vision eines intelligenten Wizzards zur Durchführung der GQM Methode vor. Der Anwender gibt ein Unternehmensziel ein und beantwortet interaktive Fragen. Schrittweise werden verfeinerte Messziele, individuelle Metriken, etc. anhand der Anwenderantworten auf intelligente Weise herausgeschält. Dies soll mithilfe von eigenen Agents und einer angeschlos-

senen Knowledge Base samt Reasoning Mechanism geschehen. Am Ende soll ein fertiger und einsatzbereiter Messplan ausgegeben werden. Der Ansatz ist interessant, scheint jedoch noch nicht praktisch umgesetzt zu sein. Auch werden Abstraction Sheets nicht betrachtet.

MetriFlame ist von VTT Electronics entwickelt worden und dient zur Sammlung, Analyse und Präsentation von Messdaten. Die Grundidee ist, dass dieses Tool automatisch Daten für eine anstehende Messung sammelt, indem es sich mit verschiedenen Datenbanken anderer Tools verbinden kann. (Parviainen u. a., 1997) setzte MetriFlame bei der Andererwendung der GQM Methode ein, um passende Metriken zu finden und Daten zu sammeln. MetriFlame erlaubt die Generierung von GQM Bäumen auf Basis von bereits gespeicherten Metriken. Das Aufstellen von Abstraction Sheets oder Messplänen wird nicht unterstützt.

Das GQM-Tool von (Lavazza, 2000) bietet die Möglichkeit GQM Bäume zu erstellen. Deren Struktur wird auf Konsistenz überprüft, indem z.B. kontrolliert wird, ob jedes Messziel zu mindestens einer Frage verfeinert wurde und es zu jeder Frage mindestens eine Metrik gibt. Messziele, Abstraction Sheets, Questions und Metriken werden als Produkte der GQM Methode unterstützt. Einzelne Komponenten von GQM-Plänen können wiederverwendet werden. Das GQM-Tool ist mit einer Datenbank verbunden. Auch dieses Tool bietet keine Hilfestellungen an, die erklären, wie ein Abstraction-Sheet auszufüllen ist.

Diskussion

Der smarte GQM-Editor verwendet Texteingaben des Anwenders, um nächst-auszufüllende Felder mit kontextualisierten Hilfstexten oder sogar vorgenerierten Antworten zu bestücken. Die Konstruktion der grauen und schwarzen Hilfstexte basiert auf Textmustern, welche wir durch wiederholtes Bearbeiten und Lehren von Abstraction Sheets in der Lehre erkannt haben. Grob gesagt hat das begleitete Erklären von Abstraction Sheets anhand solch gearteter Kommentare den höchsten Lernerfolg bei Studenten erzielt.

Der smarte GQM-Editor setzt eine einfache Textersetzung nach diesen Regeln ein. Dieser Mechanismus ist zuverlässiger, je eindeutiger die Eingabetexte sind. Z.B. lässt sich anhand der starren Form der Messziel-Facette des Abstraction Sheets (vier Eingabefelder, jeweils meist nur ein Wort) ein relativ passender und hilfreicher grauer Hinweistext im ersten Quadranten erzeugen (siehe Abb. 4). Je variabler die Eingabetexte werden, desto unzuverlässiger kann der generierte Text im nächsten Feld ausfallen. Dies kann dazu führen, dass der generierte Text unbrauchbar wird. Graue Hilfstexte verschwinden jedoch auch beim Anklicken und vorgenerierte Antworten (schwarzer Text) können einfach anwenderseitig bearbeitet oder gelöscht werden. Werden diese Textstücke generell zu verwir-

rend, kann der Anwender diese Unterstützung auch insgesamt ausschalten.

Die Feinheiten der deutschen Grammatik (männliche, weibliche Artikel, Plural, Singular) werden nicht gesondert behandelt. Die generierten Texte sind in dieser Hinsicht möglichst allgemein ausgelegt und verwenden mitunter unschöne Doppelbelegungen wie z.B. "die/den". Ein Plugin zur korrekten Lösung solcher Fälle lässt sich jedoch nachträglich einfügen. Generell priorisieren wir den Lerneffekt beim unerfahrenen Anwender gegenüber vollkommen korrekter Sprachkonstrukte. Mögliche sprachliche Fehler oder Ungenauigkeiten werden hingenommen, solange der Anwender versteht, was gemeint ist bzw. was gerade von ihm verlangt wird.

Der smarte GQM-Editor stellt keine Silver-Bullet für die GQM Methode oder das Erstellen von Abstraction Sheets dar. Der Anwender soll jedoch verstehen, was von ihm verlangt wird und wie solche Formulare gehandhabt werden. Der smarte GQM-Editor "hilft dabei auf die Sprünge" — kreative und korrektive Arbeitsschritte seitens des Anwenders werden dabei aber nicht ausgeschlossen.

Die praktische Nützlichkeit der hier vorgestellten Konzepte konnte nicht im universitären Umfeld evaluiert und validiert werden. Eine umfangreiche Evaluation ist bereits in Planung. Da diese Konzepte aber auf praktischer Lehrerfahrung basieren, sind wir zuversichtlich, dass Studierende davon profitieren werden.

Zusammenfassung und Ausblick

Die GQM Methode wurde seit seiner Vorstellung durch (Basili, 1992) erfolgreich in Industrieprojekten eingesetzt und hat die Entwicklung von verschiedenen Werkzeugen angetrieben. Lehrerfahrung an der Leibniz Universität Hannover zeigt jedoch, dass die GQM Methode bei Studierenden nur Akzeptanz findet, wenn sie praktisch ausprobiert und eingesetzt wird. Jedoch schrecken die komplex anmutenden Formulare wie Abstraction Sheet oder Messplan Studierende ab.

In dieser Arbeit wird ein lauffähiger smarterer GQM-Editor vorgestellt. Er bietet eine webbasiert Editor-Oberfläche für die beiden Formulare Abstraction Sheet und Messplan und richtet sich in erster Linie an unerfahrene GQM-Anwender (wie z.B. Studierende).

Der GQM-Editor verarbeitet vorhergegangene Anwendereingaben, um kontextualisierte Hinweistexte an passenden Stellen einzublenden. Z.B. wird der Anwender in jedem Quadrant eines Abstraction Sheets explizit und individualisiert nach den einzutragenden Inhalten gefragt. Diese Hinweistexte sollen als Denkanstoß dienen und dem unsicheren Studierenden helfen, diese Formulare zu verstehen.

Zusätzlich kann der smarte GQM-Editor Antwortteile im Abstraction Sheet auf Wunsch anhand vorheriger Eingaben generieren. Antwortteile, die nicht automatisch generiert werden können, werden durch

Lückentexte repräsentiert. Diese sollen den Anwender anregen, passende Antworten einzutragen.

Die generierten Texte erscheinen in der angedachten Ausfüllreihenfolge. Sie können vom Nutzer angepasst, gelöscht bzw. ausgeschaltet werden. Die erstellten Abstraction Sheets und Messpläne können vom Anwender gespeichert und verwaltet werden.

Eine umfangreiche Evaluierung des vorgestellten smarten GQM-Editor in der universitären Lehre wird vorbereitet. Dabei soll die Akzeptanz der GQM Methode und der zugehörige Wissensgewinn bei Studierenden gemessen werden.

Die Konzepte des smarten GQM-Editors sind als Initiative zu verstehen, komplizierte Techniken besser zu erlernen und vermitteln. Tool-basiert kann die SE-Lehre damit unterstützt werden.

Literatur

- [Basili 1992] BASILI, Victor R.: *Software modeling and measurement: the goal/question/metric paradigm*. College Park, MD : Maryland Univ., 1992 (Computer science technical report Series)
- [Basili u. Weiss 1984] BASILI, Victor R. ; WEISS, David M.: A Methodology for Collecting Valid Software Engineering Data. In: *Software Engineering, IEEE Transactions on SE-10* (1984), nov., Nr. 6, S. 728 –738. <http://dx.doi.org/10.1109/TSE.1984.5010301>. – DOI 10.1109/TSE.1984.5010301. – ISSN 0098–5589
- [Birk u. a. 1998] BIRK, A. ; SOLINGEN, R. van ; JARVINEN, J.: Business impact, benefit, and cost of applying GQM in industry: an in-depth, long-term investigation at Schlumberger RPS. In: *Software Metrics Symposium, 1998. Metrics 1998. Proceedings. Fifth International*, 1998, S. 93 –96
- [Chen u. a. 2003] CHEN, T. ; FAR, B.H. ; WANG, Y.: Development of an intelligent agent-based GQM software measurement system. In: *Proceedings of ATS*, 2003, S. 188
- [Gantner u. Schneider 2003] GANTNER, Thomas ; SCHNEIDER, Kurt: Zwei Anwendungen von GQM: Ähnlich, aber doch nicht gleich. In: *Metrikon 2003* (2003)
- [Hoffmann u. a. 1997] HOFFMANN, M. ; BIRK, A. ; ELS, F. van ; KEMPKENS, R.: *GQMAspect V1. 0: User Manual*. Fraunhofer-IESE, 1997
- [Kilpi 2001] KILPI, T.: Implementing a software metrics program at Nokia. In: *Software, IEEE* 18 (2001), nov/dec, Nr. 6, S. 72 –77. <http://dx.doi.org/10.1109/52.965808>. – DOI 10.1109/52.965808. – ISSN 0740–7459
- [Lavazza 2000] LAVAZZA, L.: Providing automated support for the GQM measurement process. In: *Software, IEEE* 17 (2000), may/jun, Nr. 3, S. 56 –62. <http://dx.doi.org/10.1109/52.896250>. – DOI 10.1109/52.896250. – ISSN 0740–7459
- [van Maris u. a. 1995] MARIS, M. van ; DIFFERDING, D.I.C. ; GIESE, D.I.P.: Ein Werkzeug zur Definition, Interpretation und Validation von GQM-Planen. (1995)
- [Parviainen u. a. 1997] PARVIAINEN, P. ; JARVINEN, J. ; SANDELIN, T.: Practical experiences of tool support in a GQM-based measurement programme. In: *Software Quality Journal* 6 (1997), Nr. 4, S. 283–294
- [Schneider 2007] SCHNEIDER, Kurt: *Abenteuer Softwarequalität: Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement*. Dpunkt, 2007
- [Van Latum u. a. 1998] VAN LATUM, F. ; VAN SOLINGEN, R. ; OIVO, M. ; HOISL, B. ; ROMBACH, D. ; RUHE, G.: Adopting GQM based measurement in an industrial environment. In: *Software, IEEE* 15 (1998), jan/feb, Nr. 1, S. 78 –86. <http://dx.doi.org/10.1109/52.646887>. – DOI 10.1109/52.646887. – ISSN 0740–7459
- [Van Solingen u. Berghout 1999] VAN SOLINGEN, R. ; BERGHOUT, E.: *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. McGraw-Hill, 1999
- [Voigtlaender 1999] VOIGTLAENDER, und Kempkens R. C.: *GQMAspect II. System Documentation*. Fraunhofer-IESE, 1999
- [Werner 2012] WERNER, Florian: *Entwurf eines regelbasierten GQM-Editors mit geringer Einstiegshürde*, Leibniz Universität Hannover, Fachgebiet Software Engineering, Bachelor's Thesis, 2012