

Das GQM-Paradigma

Sascha-Ulf Habenicht

Universität Oldenburg
Department für Informatik
26111 Oldenburg, Deutschland

Zusammenfassung Softwareprojekte werden von Fragestellungen begleitet, wie z.B. “wie teuer wird das Projekt”, “treten bestimmte Fehler häufiger auf als andere”, “was soll die Software leisten”. Mit Hilfe des Goal-Question-Metric-Paradigmas (GQM-Paradigma) dienen die Ziele des Projektes als Messgrundlage.

Innerhalb eines Softwareprojektes werden oft ziellos Messungen vorgenommen und anschließend ausgewertet. Dabei werden nach der Messung Interpretationen gesucht. Anschließend werden auf Basis dieser Daten Entscheidungen getroffen. Mit dem GQM-Paradigma werden Messungen gezielt erfasst. Als erstes werden die Ziele des Projektes aufgestellt; darauf bezogen werden Fragestellungen gesucht und erst dann wird entschieden, welche Messungen geeignet sind, diese Fragen zu beantworten. Thema dieser Arbeit ist das GQM-Paradigma. Zunächst wird dazu das GQM-Paradigma durch ein Beispiel vorgestellt, mit einem Beispiel. Darauf aufbauend wird der GQM-Plan vorgestellt. Als Abschluss wird GQM+-Paradigma von Frauenhofer Institut vorgestellt.

1 Einleitung

Oft setzen Softwareprojekte Metriken ein, die Bottom-Up erfasst werden. Hierbei werden im nachhinein aufgrund der vorhandenen Daten Interpretationen entwickelt. Anschließend werden auf den gewonnenen Erkenntnissen Optimierungen vorgenommen. Dieses Verfahren ist jedoch oft eine falsche Entscheidungsgrundlage, da diese Erkenntnisse nicht immer die Ziele des Projektes widerspiegeln. Mit dem Goal-Question-Metric-Paradigma (GQM-Paradigma) wird ein Top-Down Ansatz zur Planung der Messmethoden verfolgt. Hierbei stehen zunächst nicht die Messmethoden, sondern die Ziele des Projektes, im Vordergrund. Auf diesen Zielen aufbauend werden Fragen formuliert, die das jeweilige Ziel wiedergeben. Jede Frage wird mit einer Hypothese begründet, damit bleibt der Kontext der Frage auch bei späteren Phasen erkennbar. Anschließend werden geeignete Messmethoden zur Beantwortung der Fragen gesucht. Hierdurch erarbeitet man sich ein Entscheidungsmodell, welches mit Hilfe von Messmethoden verfolgte Ziele widerspiegelt.

Dieses Vorgehen wird durch einige Studien [5], welche Modelle und Metriken auf Wirksamkeit untersuchten, bestätigt. Victor R. Basili und G. Caldiera und H. D. Rombach [5] nennt hierzu drei wesentliche Kriterien:

1. Konzentration auf konkrete Ziele.
2. Einbeziehung aller Produkte, Zwischenprodukte, Prozesse und Ressourcen.
3. Interpretation im Kontext der Organisation, der Umwelt und Ziele.

Das GQM-Paradigma hebt sich besonders durch die Möglichkeit hervor, verwendbar für alle Produkte zu sein. Es ist damit nicht allein für das Qualitätsmanagement anwendbar, sondern auch für das Projektmanagement.

Nach [2] unterscheidet sich hier das GQM-Paradigma gegenüber anderen Top-Down-Paradigmen wie Quality Function Deployment (QFT) oder Software Quality Metrics (SQM). Die Top-Down Paradigmen QFT und SQM sind nur im Qualitätsmanagement einsetzbar.

In dieser Ausarbeitung wird in Kapitel 2 das GQM-Paradigma vorgestellt, dazu werden die Ziele, Fragen und Metriken beleuchtet. In Kapitel 3 wird das GQM-Paradigma mit Hilfe des GQM-Plan umgesetzt. In Kapitel 4 wird das GQM-Paradigma als Informationsfluss und Rollenkonzept innerhalb eines Softwareprojektes vorgestellt. Dieses Kapitel folgt aus den Zusammenhang der Beteiligten Personen des GQM-Plan und der allgemeinen Problematik der Software Engineering, wie es in der begleitenden Vorlesung thematisiert wurde. Das 5 Kapitel stellt GQM+ des Fraunhofer Institut vor und im letzten Kapitel wird eine zusammenfassende Bewertung abgegeben.

2 Das GQM-Paradigma

Das GQM-Paradigma ist ein Vorgehen, um in einem Softwareprojekt Ziele mit geeigneten Metriken verfolgen zu können. Dazu werden zuerst Ziele ohne Fragen oder Metriken erhoben. Wenn ein Ziel definiert ist, werden hierzu geeignete Fragestellungen abgeleitet, die die Ziele wiedergeben. Anschließend werden von den Fragen aus geeignete Metriken abgeleitet. Dies kann als Baumstruktur gesehen werden, wie in Abbildung 1 dargestellt ist.

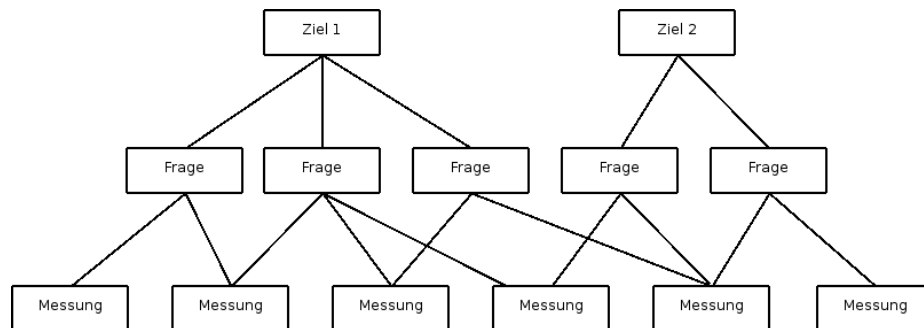


Abbildung 1. GQM hierarchische Struktur [5].

2.1 Ziele

Ziele stehen immer im Bezug zu einem Objekt. Nach [5] sind messbare Objekte:

- Produkte: Artefakte, Leistungen und Dokumente wie z.B. Spezifikationen, Entwürfe, Protokolle.
- Prozesse: Zusammenhängende Arbeitsschritte um etwas innerhalb der Organisation fertig zu stellen, wie Interviews, Spezifikationen, Tests, Planung.
- Ressourcen: Nutzung durch Prozesse, wie z.B. Personal, Hardware, Software, Büros.

Diese Objekte werden innerhalb verschiedener Softwareprojekte anderen Gesichtspunkte und Qualitätsanforderungen unterliegen. Ziele stehen oft zu einander im Konflikt, z.B. kann in einem Softwareprojekt das Ziel existieren Zuverlässig zu sein und ebenso das Ziel minimale Ressourcen um Kosten zu sparen. Um jedoch eine hohe Zuverlässigkeit garantieren zu können, braucht man Ressourcen die Kosten verursachen und Zeit in Anspruch nehmen.

2.2 Fragen

Zu jedem Ziel werden mehrere Fragen gesucht. Diese Fragen sollen die Ziele möglichst gut wieder geben, es hilft dabei die Ziele aus mehreren Sichten zu betrachten. Jede Frage stellt eine Hypothese auf, mit welcher das Ziel weiter verfolgt werden kann. Mit Hilfe der Hypothese werden Entscheidungen im Kontext der Fragestellung ersichtlich.

Angelehnt an [6] sind Teilkriterien zur Erfüllung der notwendigen Fragestellung:

- Sichtweise: Auftraggeber, Entwickler, Nutzer usw.
- Anwendungsbereich: System, Entwicklung, eingesetzte Technologien, Personal usw.
- Zweck: Verbesserung, Analyse, Kontrolle, Verständnis usw.
- Kontext: selbst, außerhalb, vorgegebene Strukturen

Das GQM-Paradigma dient mit seinen Fragen durch die verschiedenen möglichen Sichtweisen auch als Kommunikationsinstrument zwischen den Beteiligten des Softwareprojektes.

2.3 Metriken

Die Daten der Messungen werden mit Hilfe der gestellten Fragen, Hypothesen und Ziele interpretiert. Dabei können die Daten objektiver (z.B. Lines of Codes, Aufwand der Personalstunden, Anzahl Versionen) oder auch subjektiver Art (z.B. Zufriedenheit der Nutzer) sein.

Die quantitative Auswertung der Messungen steht somit immer im Bezug zu Fragen, welche wiederum einen Bezug zu den Zielen des Softwareprojektes haben. Die Messungen liefern damit keine Daten, zu denen man erst eine geeignete

Interpretation suchen muss. Diese Interpretation folgt über die Fragen zu den Zielen.

Eine Stärke des GQM-Paradigma ist es, sowohl bei den Software-Produkten (Anforderungskatalog, Wartungsdokumente usw.) als auch bei Software-Prozessen (Aktivitäten, die zur Produktion von Software-Produkten dienen) eingesetzt werden zu können. Nach [4] können folgende Prozessmetriken unterschieden werden:

1. Aufgewandte Zeit des Prozesses
2. Verbrauchte Ressourcen durch den Prozess
3. Anzahl Fehler im Verlauf des Prozesses

Dabei werden 1. und 2. häufig benutzt, um Prozessänderungen zu analysieren. Produktmetriken hingegen befassen sich mit Eigenschaften der Software, wie z.B. Messungen während der Ausführung des Programms (dynamische Metrik) oder Systemdarstellungen z.B. Entwurf (statische Metriken).

2.4 Ein einfaches Beispiel

Das folgende Beispiel ist angelehnt an [6], wobei es in der Quelle um das Ziel Webprojekt-Erfolgsaussicht geht. Das ausgesuchte Ziel in dieser Ausarbeitung ist die Projekt-Erfolgsaussicht eines Softwareprojektes, in dem eine grafische Oberfläche zu einer Datenbank entwickelt werden soll. Zu diesem Projekt wird ein Framework benutzt. In Tabelle 1 werden zum genannten Ziel Fragen gestellt und dazu geeignete Metriken ausgewählt.

Fragen

Warum scheitern solche Projekte?

Welche Fähigkeiten besitzt das Management?

Wie kommen die Entwickler mit dem Framework zurecht?

Welche Risiken liegen in solchen Projekten?

Wie stabil sind die Anforderungen an die grafische Oberfläche?

Wie häufig treten technische Probleme mit dem verwendeten Framework auf?

Wie kommen die Anwender mit der grafischen Oberfläche zurecht?

Metriken

Projektumfang der umzusetzenden Anforderungen

Fachkenntnisse, Erfahrungen des Managements

Fachkenntnisse, Erfahrungen der Entwickler

Auflistung Risikofaktoren und prozentuale Schätzung

Änderungen an der Datenbankviewstruktur

Verbrauchte Ressourcen um aufgetretene technische Probleme im Umgang mit dem Framework zu lösen

Testpersonen geben subjektive Einschätzungen ab

Tabelle 1. Fragen und Metriken zum Ziel Softwareprojekt-Erfolgsaussicht

Jeweils ein Ziel wird in einem GQM-Plan verarbeitet. Im Projekt-Plan werden diese GQM-Pläne dann umgesetzt. So werden geeignete Messungen vorgenommen, anfallende Daten aller zugehörigen Prozesse dokumentiert und die

Interpretation durchgeführt. Darauf kann man Verbesserungsmaßnahmen gezielt ergreifen und anschließend den Erfolg oder Misserfolg über erneute Messungen verfolgen.

3 GQM-Plan

Ein GQM-Plan wird für jedes Ziel aufgestellt. Im GQM-Plan steht das Ziel, die Fragen und Hypothesen sowie die zu erhebenden Metriken. Bei jeder Entscheidung, wie z.B. welche Metrik im Bezug zu einer Frage bestimmt wird, wird im GQM-Plan dokumentiert. Es wird dabei mit dem GQM-Paradigma top-down gearbeitet. Interpretationen der vorgenommenen Messungen hingegen werden anschließend bottom-up im Kontext des Zieles vorgenommen. An dem GQM-Plan arbeiten alle Projektteilnehmer, die zum Erreichen des Zieles notwendig sind und bringen so ihre jeweilige Sichtweise mit ein. Auch bei der Interpretation der Daten werden über die Teilnahme der Projektteilnehmer alle Sichtweisen in den Schlussfolgerungen mit eingebracht.

In Abbildung 2 werden vier Phasen zur Umsetzung des GQM-Paradigmas gezeigt. Während der Definitionsphase wird der GQM-Plan entwickelt, wobei mit Hilfe des GQM-Paradigma die Ziele, Fragen und geeignete Metriken festgestellt werden. Sind die Metriken ausgesucht, werden Messungen veranlasst. Diese werden als Antworten aufbereitet, welche die Zielerreichung festhalten. Zu jeder Phase werden die anfallenden Daten gesammelt.

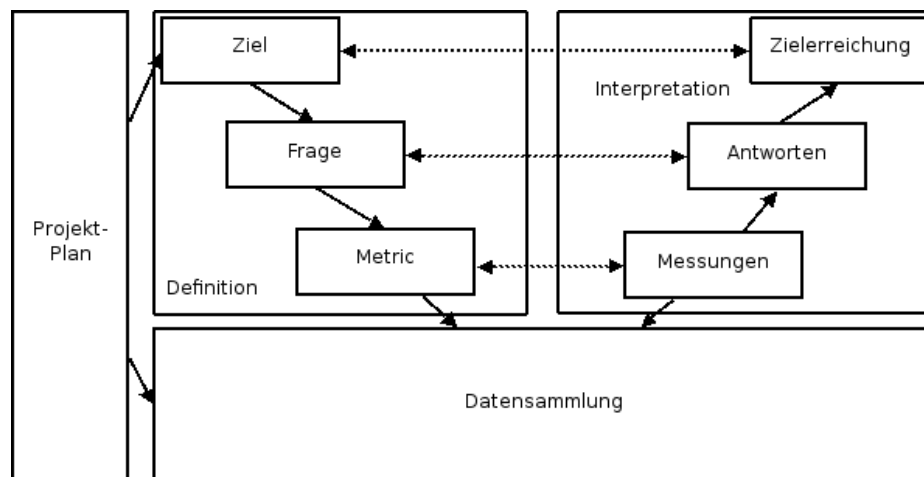


Abbildung 2. Vier Phasen zur Umsetzung des GQM-Paradigma [7].

Projektmitglieder kooperieren bei der Erstellung und Interpretation des GQM-Plans. Der GQM-Plan dient hierdurch als Kommunikationswerkzeug. Durch die

große Anzahl von Sichtweisen können Verbesserungsmaßnahmen vorher beurteilt werden und durch den Vergleich der Messungen vorher und nachher weiter fein justiert werden. Durch die Interpretation der Messergebnisse anhand von Fragen, dazugehörigen Hypothesen und Zielen können z.B. technische Details aus der Entwicklung für andere Projektteilnehmer vereinfacht werden.

3.1 Beispiel GQM-Plan

Nun soll ein Beispiel eines GQM-Plan gezeigt werden. Das Beispiel ist an [3] angelehnt. Der GQM-Plan hat ein Ziel, eine Revisionsnummer und das Datum der letzten Bearbeitung. Zu den Ziel werden zugehörige Fragen aufgelistet, wobei die Hypothese und das Maß mit angegeben wird. Der in diesem Beispiel gezeigte GQM-Plan gibt die Einschätzung der Beteiligten Personen zum Ziel Zuverlässigkeit wieder. Durch eine Auswertung des GQM-Plan können gezielt Maßnahmen zur Verbesserung des Zieles veranlasst werden. Den Erfolg dieser Maßnahmen kann durch den GQM-Plan anschliessend weiter verfolgt werden.

GQM-Plan Zuverlässigkeit

Dieser GQM-Plan beinhaltet das Messprogramm zur Zuverlässigkeit in der Abteilung Testen

Revision: 0.01 vom 27. Juni 2008

<u>Analysiere</u> den Entwicklungsprozess	[Objekt]
<u>zum Zweck</u> des Charakterisierens	[Zweck]
<u>bezüglich</u> der Zuverlässigkeit	[Qualitätsaspekt]
<u>aus dem Blickwinkel</u> des Tester	[Blickwinkel]
<u>bei</u> der Abteilung Testen	[Kontext]

Prozeß Definition

Frage D 1:

Werden die Richtlinien der Abteilung eingehalten?

Hypothese: Arbeiten anhand der Abteilungsrichtlinien haben sich in der Fehlerfindung bewährt.

Maß: Richtlinien eingehalten: [Ja / Nein / Prozess nicht bearbeitet]

Frage D 2:

Wenn der Prozess nicht bearbeitet wurde, geschah das aus Zeitmangel?

Hypothese: Bei Zeitmangel wird die Fehlersuche vernachlässigt.

Maß: Zeitmangel Testen: [Ja / Nein]

Frage D 3:

Hält der Code die Standards des Softwareprojektes ein?

Hypothese: Wenn der Code die Standards des Softwareprojektes, insbesondere bei Modulen einhält, treten weniger Fehler in der Zusammenarbeit mit anderen Code-Abschnitten auf.

Maß: Standard Code: [Ja / Nein]

[...]

Frage D 9:

Anzahl der Methoden in einem Modul

Hypothese: Je mehr Methoden sich in einem Modul befinden, desto anfälliger wird das Modul für Fehler.

Maß: Modulname: [Text]

Anzahl der Methoden im Modul:

Ende des GQM-Plans Zuverlässigkeit

4 Informationsfluss und Rollen

Dieses Kapitel folgt aus den Zusammenhang der Beteiligten Personen des GQM-Plan und der allgemeinen Problematik der Software Engineering, wie es in der begleitenden Vorlesung thematisiert wurde. In einem Softwareprojekt hat jeder Projektteilnehmer eine oder mehrere Rollen mit unterschiedlichen Sichtweisen und Anforderungen hinsichtlich der Software-Produkte und Software-Prozesse. Software-Produkte sind alle notwendigen Dokumente des Lebenszyklus der Software, von der Entwicklung über der Wartung und Pflege. Software-Prozesse sind alle Aktivitäten, die zur Produktion von den Software-Produkten notwendig sind, wie z.B. Implementierung, Wartung. Häufige Rollen in einem Softwareprojekt sind:

- Auftraggeber
- Management
- Entwickler
- Tester
- Wartungspersonal

Der Auftraggeber verfolgt das Ziel einer hohen Qualität mit niedrigen Kosten. Das Management will in kurzer Zeit ein hohes Qualitätsprodukt mit den Einsatz minimaler Ressourcen erreichen. Der Entwickler wird versuchen im Rahmen seiner vorgegebenen Ressourcen ein Qualitätsprodukt abzuliefern. Der Tester wird Ressourcen verbrauchen, um Fehler im Softwareprodukt oder den zugehörigen Anforderungen aufzudecken und zudem sicherstellen wollen, das bekannte Fehler nach der Behebung nicht wieder auftauchen. Das Wartungspersonal braucht die anfallenden Dokumente um die Software einzusetzen. Dazu wird in der Wartung des Software-Produkt stabilisierend sowie korrektiv und optimierend bearbeitet. In der Pflege werden Anpassungen, Änderungen und Erweiterungen durch das Wartungspersonal vorgenommen.

Die Sichtweisen der jeweiligen Rollen stehen oft im Konflikt zueinander. Die Arbeit des Testers verbraucht Ressourcen, die das Ziel des minimalen Ressourceneinsatz beim Management und die niedrigen Kosten gegenüber dem Auftraggeber beeinflussen. Der Entwickler wird mehr Ressourcen verbrauchen, wenn die

Dokumente nicht nur für die Entwicklung gedacht sind, sondern auch Wartungsdokumente sein werden.

Am GQM-Paradigma werden die Ziele unter Beteiligung aller Rollen und deren Sichtweisen formuliert, verarbeitet und Verbesserungsmaßnahmen vereinbart. Beim Ziel Kosten kann z.B. die Frage vom Auftraggeber gestellt werden, ob die Kosten des Ressourcenverbrauchs der Tester wirklich notwendig sind. Als mögliche Metrik hierzu kann man dann die Anzahl der in der Wartung erst auffallenden Fehler stellen. Der Tester kann die Frage stellen, wie oft das Entwicklungsteam bei der Fehlerbehebung das Wartungspersonal unterstützen muss. Durch dieses Zusammenspiel von Zielen, Fragen und Metriken kann jetzt das Management und auch der Auftraggeber Maßnahmen und deren Wirkungen besser einschätzen.

5 GQM+ des Fraunhofer Instituts

GQM+ des Fraunhofer Institut setzt zum klassischen GQM einen Bezug von Messzielen zu unternehmensweiten Geschäftszielen und Softwarezielen. Hier wird das GQM-Paradigma um zwei Vorstufen erweitert. Bevor man die Messziele aufsetzt, durchläuft man die Geschäftsziele, entwickelt hierzu geeignete Strategien. Dann folgen Softwarezielen, welche durch Verbesserungsszenarien umgesetzt werden sollen zu den Messzielen [1]. Es wird hierbei also ein Bezug zwischen Messzielen, Softwarezielen und Geschäftszielen erreicht. Als oberstes Ziel steht immer das Geschäftsziel, gefolgt vom Softwareziel. Anschließend wird dann das GQM-Paradigma umgesetzt, siehe auch Abbildung 3.

Zwischen dem Geschäftsziel und dem Softwareziel besteht die Verbindung Strategien. Eine Strategie hilft dabei, Softwareziele zu identifizieren im Kontext des Geschäftszieles. Als Verbindung zwischen Softwareziel und dem GQM-Paradigma werden Verbesserungsmaßnahmen benutzt. Nun werden mit Hilfe des GQM-Paradigmas Messziele gesucht und umgesetzt.

Das GQM+ Modell sorgt für die Ausrichtung aller Ziele an der strategischen Ausrichtung der Geschäftsziele. Am Fraunhofer Institut werden hierzu Werkzeuge entwickelt, die Erfahrungen und Wissen sammeln und so zur Wiederverwendung bereit stehen.

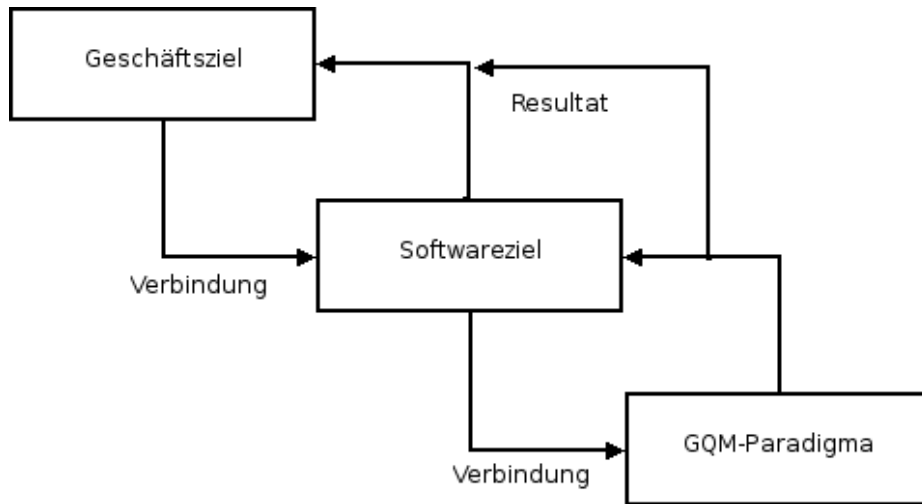


Abbildung 3. GQM+ des Fraunhofer Institut [1].

6 Zusammenfassung

Das GQM-Paradigma hilft die Qualität von Software-Projekten zu sichern, Optimierungen gezielt zu verfolgen und Planzahlen gezielt zu erfassen. Es dient als Entscheidungsmodell, um Ziele des Projektes umzusetzen. Ziele werden hiermit klar erfasst und erst dann werden Metriken über Fragen immer im Bezug zu den Zielen gesucht. Durch dieses Modell können Fragen und ihre Ziele jederzeit die Zielumsetzung im Projekt verfolgt werden.

Das GQM-Paradigma bietet mehrere Vorteile gegenüber Messungen ohne zugehörige Zielsetzung. Auf der anderen Seite können Messungen mehreren Fragen und Zielen dienen. Hierdurch werden keine Ressourcen mit Messungen ohne Fragen verschwendet, dies verhindert zudem einen ungewollten Einfluss auf Entscheidungen durch Messungen, welche keine Zielen zuzuordnen sind.

Mit Hilfe des GQM-Paradigmas werden z.B. Prozesse besser beurteilbar. Mit gezielten Messungen werden Verbesserungsmaßnahmen entschieden, welche durch fortlaufende Messungen auf ihren Erfolg messbar werden. Mit Hilfe der Metriken zum Bezug der Fragen und Ziele kann man nun Optimierungen eines Prozesses anschließend mit dem Zustand vorher vergleichen und so weitere Feinjustierungen vornehmen.

Mit den GQM-Plan steht eine in der Praxis erfolgreiche Vorlage für Messpläne zur Verfügung. Im GQM-Plan steht immer ein Ziel, zu jeder Frage werden die Hypothesen aufgeführt. Hierdurch sind die GQM-Pläne jederzeit nachvollziehbar und interpretierbar. Vergleiche zwischen Messungen zu anderen Zeitpunkten sind übersichtlich und gut organisiert.

Der notwendige Informationsfluss unter den Rollen im Softwareprojekt zur Umsetzung des GQM-Paradigma unterstützt dabei die Ziele bestmöglichst um-

zusetzen. Viele gesetzte Ziele stehen im Konflikt zueinander und müssen feinjustiert werden. Durch die zugehörigen Fragen und Metriken zum jeweiligen Ziel werden allen Rollen, und damit deren Sichtweisen, über Auswirkungen von Maßnahmen besser unterrichtet.

Am Fraunhofer Institut wird mit dem Modell GQM+ eine Erweiterung des GQM-Paradigmas verfolgt, wobei die Ziele des GQM-Paradigma eine vorherige Ausrichtung am Geschäftsziel und erst anschließend am Softwareziel orientieren. Das Modell GQM+ bindet das GQM-Paradigma gezielter in das Management und in die Unternehmensphilosophie ein.

Literatur

- [1] GQM+ strategies. URL http://www.iese.fraunhofer.de/fhg/iese_DE/products/gqm/index.jsp. Besucht am 8. Mai 2008; 21:28 Uhr.
- [2] C. Differding. Ein Objektmodell zur Unterstützung des GQM-Paradigmas. 1993. Diplomarbeit.
- [3] D. Hamann. Integration von GQM-Plänen und Prozeßmodellen durch Meßpläne. 1996. Diplomarbeit.
- [4] I. Sommerville. *Software Engineering 6. Auflage*. 2001.
- [5] V. R. B. und G. Caldiera und H. D. Rombach. Goal Question Metric Paradigm. In *Encyclopedia of Software Engineering*, volume 1, pages 528–532. John Wiley & Sons, 1994.
- [6] R. D. und Mathias Lothar und Cornelius Wille und Fritz Zbrog. *Web Engineering*. Pearson Studium, 2003.
- [7] R. van Solingen und E. Berghout. *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. McGraw-Hill, 1999.