

Durch die Unterstützung von Metriken wie z.B. dem *Cost Performance Index* können Probleme viel früher entdeckt und behoben werden. Der CPI eines Projektes sollte möglichst dauerhaft größer oder gleich 1 sein, da ansonsten höhere Kosten als geplant entstehen. Allein durch diese einfache Metrik kann der Projektmanager besser entscheiden, ob er beispielsweise ein Projekt einstellen soll.

Wichtig ist aber nicht nur das Messen der Prozesse im Unternehmen z.B. mittels CMMI, sondern auch die Qualität der eigenen Produkte um seine Kunden nicht zu verlieren. Ohne eine Vermessung der eigenen Produkte kann nicht geprüft werden, ob die eigenen Qualitätsziele erreicht werden. Qualitätsziele sind nach der ISO Norm 9000:2000 (Quelle: DGQ [11]) „Etwas bezüglich Qualität Angestrebtes oder zu Erreichendes“. Das heißt Qualitätsziele eines Unternehmens sind die angestrebten Qualitäten für die eigenen Produkte. Zum Beispiel könnte gefordert sein, dass die Wartbarkeit der Software hoch sein muss. Doch die metriklose Bewertung des Endprodukts hinsichtlich der Wartbarkeit ist sicher nicht objektiv. Als Ansatz zur Messung der Wartbarkeit wäre es z.B. sinnvoll die innere Komplexität des Programmcodes zu messen. In die Bewertung der Komplexität einer objekt-orientierten Software könnte z.B. die Anzahl der Klassen, die Anzahl der Klassenbeziehungen, die Tiefe der Vererbungshierarchien und ähnliches eingehen. Um den Nutzen von Metriken weiter zu verdeutlichen sei auf den Artikel von Seibert [23] verwiesen, wo deutlich gemacht wird, dass man mithilfe von Metriken beispielsweise das Problem zu niedriger Planvorgaben vermindern kann. Im Management ist der Irrglaube, niedrige Planvorgaben steigern die Produktivität, nämlich immer noch weit verbreitet. Mithilfe von Metriken wie COCOMO (siehe Abschnitt A.1.1) kann man bei vernünftiger Kalibrierung dieses Problem leicht beheben, da eine plausible, objektive, empirisch nachgewiesene Metrik für das Management überzeugender ist, als der Hinweis des Entwicklerteams, die Planvorgaben seien zu niedrig.

Somit ergibt sich, dass Metriken bei der Softwareentwicklung sehr nützlich sein können und wahrscheinlich eine immer größere Rolle in der Industrie spielen werden. Allerdings muss darauf hingewiesen werden, dass Metriken nur bei größeren Projekten sinnvoll sind. Bei kleinen Projekten ist der Overhead, der durch die Entwicklung, Berechnung und Interpretation von Metriken entsteht, zu groß. Um das Konzept einer Metrik besser zu verstehen, finden sich im Anhang Beispiele für Metriken.

### 2.2. Metrikprozesse

Nachdem im vorherigen Abschnitt eine kurze Einführung in die Softwaremetriken geliefert wurde, geht dieser Abschnitt nun auf den Metrikprozess ein, d.h. unter anderem auf die Entwicklung von Metriken. Metrikprozesse sind das Hauptthema dieser Arbeit, da die hier erstellte Software den Metrikprozess unterstützen soll, indem sie unterstützend bei der Kommunikation zwischen einem

Metrikkunden (meistens ein Projektleiter) und der Metrikabteilung eingesetzt wird. Die ISO/IEC [16] definiert den Metrikprozess als „process for establishing, planning, performing and evaluating measurement within an overall project, enterprise or organizational measurement structure“. Dementsprechend beinhaltet der Metrikprozess nicht nur die Entwicklung von Metriken, sondern auch deren Etablierung, Ausführung und Auswertung. Die Metriken können dabei unternehmensweit oder nur projektweit definiert und benutzt werden. Im nächsten Kapitel wird der GQM-Metrikprozess vorgestellt, da dieser auch in abgewandelter Form (vgl. Abschnitt 2.2.2) beim Kooperationspartner verwendet wird.

### 2.2.1. GQM

Bei GQM handelt es sich um ein Verfahren zur Erstellung von Metriken zur Befriedigung bestimmter Informationsbedürfnisse. Es wurde von Basili u. a. [3] am NASA Goddard Space Flight Center entwickelt, um die gezielte Erstellung von Metriken zu forcieren. GQM steht für Goal Question Metric, was das Verfahren in seinem Kern beschreibt (siehe Abb. 2.2). Das Verfahren beginnt damit, das

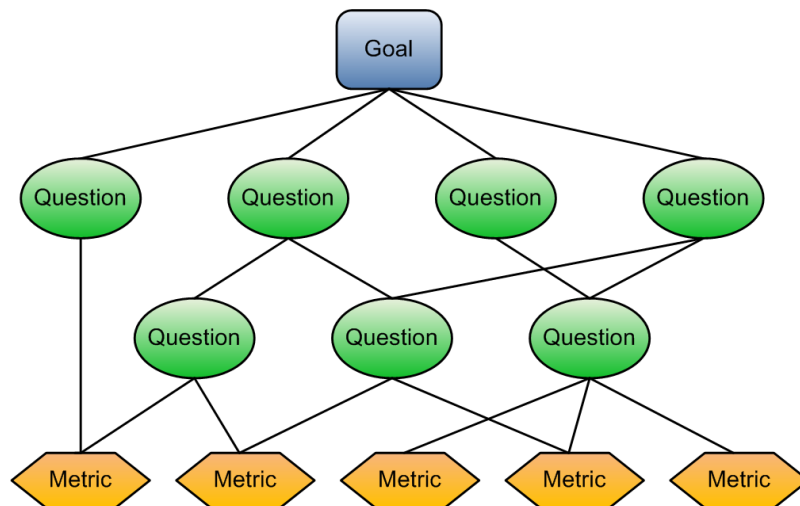


Abbildung 2.2.: Darstellung des hierarchischen Aufbaus des GQM-Ansatzes

man das Ziel definiert, das man erreichen möchte. Aus dem Ziel entwickelt man Fragen, die versuchen das Ziel zu zerlegen und aus verschiedenen Perspektiven zu betrachten. Diese Fragen können weiter durch neue Fragen zerlegt werden. Wenn die Fragen ein genügend geringes Abstraktionsniveau erreicht haben, so leitet man aus diesen Metriken ab, die die Fragen beantworten. Auf diese Weise werden die zur Überprüfung des Ziels nötigen Metriken identifiziert. Dadurch baut sich eine baumartige Struktur auf (per Definition ist es kein echter Baum), dessen Wurzel das Ziel darstellt. Darunter liegen Teilbäume deren Blätter Metriken sind und auf den Pfaden zu den Blättern liegen Fragen.

Wie an diesem Modell gut zu erkennen ist, handelt es sich bei der Entwicklung

von Metriken nach GQM um einen Top-Down-Ansatz. Zentral ist die Bestimmung des Ziels. Dieses muss nach bestimmten Gesichtspunkten definiert werden. Erstens muss man den Zweck definieren, wie z.B. *Verbesserung*. Dann wird das Thema definiert, das z.B. *Termineinhaltung* sein könnte. Es folgt die Definition des Objekts, welches begutachtet wird. Mögliche Messobjekte sind Produkte, Prozesse und Ressourcen. Ressourcen sind Objekte, die zur Erstellung von Produkten in Prozessen benötigt werden. Die Messobjekte seien *Meilensteine*. Die Begutachtung wird dann aus der Perspektive des *Projektmanagers* vorgenommen. Somit ergibt sich als Ziel: *Verbesserung der Termineinhaltung der Meilensteine aus Sicht des Projektmanagers*. Die Perspektive ist wichtig, da sie die Bewertung festlegt. So ist eine Zeitüberschreitung von einem Tag für einen einzelnen Programmierer unter Umständen eher nebensächlich, nicht aber für den Projektleiter, da sich durch diese Verzögerung auch andere Teile des Projekts verzögern.

Zur Entwicklung eines Ziels werden drei Informationsquellen benötigt. Aus der ersten Quelle, der Unternehmenspolitik und -strategie, leitet man den Zweck und das Thema ab. Um das Messobjekt zu benennen, bezieht man sich auf die Beschreibung der Produkte und Prozesse des Unternehmens, die ein Modell für das jeweilige Objekt liefern. Als drittes benötigt man ein Modell des Unternehmens, um den Blickwinkel zu definieren.

Um das Ziel zu erreichen, müssen Fragen gestellt werden, die das Abstraktionsniveau durch Zerlegung des Ziels senken. Diese Fragen können auch mehreren Zielen dienen. Genauso können die Metriken später verschiedenen Fragen dienen. Deshalb ergibt sich keine echte Baumstruktur, sondern ein ungerichteter Graph. Basili u. a. [3] unterteilen die Fragen in drei Kategorien. Die erste Kategorie umfasst Fragen, die versuchen das Messobjekt in Bezug auf das ganze Ziel zu charakterisieren. Ein Beispiel wäre: „Wie sieht momentan die Termineinhaltung der Meilensteine aus?“. Die nächste Kategorie von Fragen beschäftigt sich mit den relevanten Attributen. Die Frage „Wie hat sich die Termineinhaltung über die Zeit entwickelt?“ ist dafür ein Beispiel. Die dritte Kategorie beinhaltet Fragen, die sich auf die Charakteristiken des Objekts in Bezug auf das Thema beziehen. Hier könnte man z.B. fragen: „Ist die momentane Termineinhaltung der Meilensteine für den Projektmanager zufriedenstellend?“.

Wenn die Fragen präzise genug sind, werden Metriken verwendet, um sie zu beantworten. Es empfiehlt sich Metriken zu verwenden, die bereits vorhandene Datenquellen nutzen. Außerdem ist es sinnvoll reifen Messobjekten objektive Metriken zuzuordnen, da diese gut genug verstanden sein sollten, um auch eine objektive Metrik zu finden. Als reife Messobjekte bezeichnen Basili u. a. [3] formale und stabile Objekte. Bei weniger reifen, instabilen Objekten kann man subjektive Metriken wählen.

Die Evaluation des GQM-Modells selbst ist ein weiterer wichtiger Punkt, da bei Veränderungen innerhalb des Unternehmens, der Messobjekte, etc. eine Anpassung des GQM-Modells notwendig ist. Ansonsten könnten die Messergebnisse verfälscht werden, da sie auf falschen Annahmen beruhen.

### 2.2.2. GAM

Bei GAM (Goal Attribute Metric) handelt es sich um eine Abwandlung des oben beschriebenen GQM-Prozesses. Der GQM-Ansatz hat laut Sazama [22] das Problem, „dass die Erarbeitung der Fragen innerhalb des GQM eine gewisse Schwierigkeit darstellt.“ Sazama beschreibt zwei Probleme aus seiner Erfahrung. Einerseits sind die Fragen häufig zu nah an der Zieldefinition oder sind sogar „eine Repräsentation der Ziele in Form einer Frage.“ Andererseits kommt es aber häufig auch vor, dass die Fragen zu nah an den Metriken sind, u.U. sogar Teile der Metrikdefinition beinhalten. Weiterhin ist die Zieldefinition problematisch, da häufig nicht klar ist, welches Ziel man eigentlich verfolgt. Allerdings kann man meistens einen Informationsbedarf bestimmen.

Um den GAM-Prozess einfacher zu gestalten, schlägt Sazama ein „Abstraction Sheet“ vor, das ausgefüllt werden muss. Es stellt somit den Leitfaden für GAM dar. Das Abstraction Sheet beinhaltet das GAM-Messziel, das ähnlich wie das GQM-Ziel mehrere Facetten besitzt. Erst muss das zu beobachtende Objekt bestimmt werden, dann die Eigenschaft, die bewertet werden soll. Außerdem muss auch hier wieder die Perspektive für die Bewertung angegeben werden. Zusätzlich ist ein Kontext gefordert, in welchem die Eigenschaften beobachtet werden. . Dazu gehören beispielsweise Randbedingungen. Dieser Kontext wird bei Basili [2] als *Umwelt* bezeichnet. Ferner ist ein Zweck gefordert. Diese Definition weicht somit nicht von der Definition eines Ziels bei GQM ab.

Im nächsten Schritt wird die zu messende Eigenschaft begutachtet. Hierbei ist die Frage, ob man eine Metrik findet, die dieses Attribut bzw. diese Eigenschaft direkt misst, entscheidend. Findet man keine solche Metrik, so muss man das Attribut in „Unterattribute“ zerlegen. Man sucht also nach Attributen, die das Ursprungsattribut als Gesamtheit beschreiben. So kann nach Ludewig u. Lichter [18] das Attribut „Wartbarkeit“ in die Attribute „Prüfbarkeit“, „Änderbarkeit“ und „Portabilität“ zerlegt werden. Die „Änderbarkeit“ kann wiederum in „Strukturiertheit“, „Simplizität“, „Knappheit“ und „Lesbarkeit“ unterteilt werden. Die Lesbarkeit könnte bestimmt werden, indem man den Dokumentationsanteil bestimmt, sowie den Anteil des Codes, der den Formatierungsrichtlinien entspricht. Die Strukturiertheit könnte dann unter anderem durch die CBO (vgl. Abschnitt A.1.1) bestimmt werden.

Die Ersetzung der Fragen durch Attribute soll zu einer einfacheren Anwendung des GA(Q)M-Prinzips führen, da dieses Vorgehen für unerfahrene Mitarbeiter leichter nachzuvollziehen ist. Wie dieser Abschnitt zeigt, ist der Unterschied zwischen GAM und GQM nicht sehr groß. Der GAM-Ansatz scheint aber dennoch etwas einfacher anwendbar zu sein.