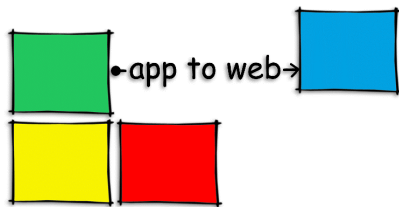


Handbuch

Messen und Bewerten von Webapplikationen mit der Goal/Question/Metric Methode



Autoren:

Danilo Assmann
Ralf Kalmar
Dr. Teade Punter

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministerium für Wirtschaft und Technologie unter den Förderkennzeichen 16IN0047 und 16IN0048 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

IESE-Report Nr. 087.02/D
Version 1.2
23. Dezember, 2002

Eine Publikation des Fraunhofer IESE

Das Fraunhofer IESE ist ein Institut der Fraunhofer-Gesellschaft.

Das Institut überträgt innovative Software-Entwicklungstechniken, -Methoden und -Werkzeuge in die industrielle Praxis. Es hilft Unternehmen, bedarfsgerechte Software-Kompetenzen aufzubauen und eine wettbewerbsfähige Marktposition zu erlangen.

Das Fraunhofer IESE steht unter der Leitung von
Prof. Dr. Dieter Rombach
Sauerwiesen 6
67661 Kaiserslautern

Inhaltsverzeichnis

Inhaltsverzeichnis		v
1	Einleitung	1
1.1	App2Web	1
1.2	Motivation für das „Messen und Bewerten von Web-Applikationen“	2
1.3	Was ist messen?	3
1.4	Worauf es beim Messen ankommt	4
1.5	Zweck dieses Handbuchs	5
1.6	Struktur dieses Handbuchs	5
2	GQM als Methode zum Messen und Bewerten	6
2.1	Einleitung	6
2.2	Der Rahmen eines Messprogramms	8
3	Messplanung	11
3.1	Definition von Geschäfts- und Verbesserungszielen	13
3.2	Definition von Messzielen	14
3.3	Definition von Fragen	25
3.4	Definition von Metriken	29
3.5	Ein Beispiel für eine Frage	32
4	Messinstrumentierung	33
4.1	Entwicklung des Messplans	34
4.2	Vorbereitung der Datenerfassung	34
5	Datenerfassung	37
5.1	Datensammlungsinstrumente	38
5.2	Durchführung der Datensammlung	41
5.3	Validierung	41
6	Analyse	46
6.1	Analyse der Daten	48
6.2	Feedback-Sitzung	48
6.3	Bündeln/Paketieren von Ergebnissen	51
7	Anwenden der Ergebnisse	54
7.1	Definieren von Maßnahmen	54
7.2	Erfolgskontrolle	54

7.3	Kontinuierliches Feedback	56
	Referenzen	58
Anhang A	GQM-Plan-Vorlage	59
Anhang B	Vorlage Messplan	60
Anhang C	Software-Metriken für Webapplikationen	62
Anhang D	Richtlinien für Feedback-Sitzung (FBS)	65
Anhang E	Sammlung von Lessons Learned	70

1 Einleitung

1.1 App2Web

Dieses Handbuch entstand im Rahmen Arbeitspakets 6 (Qualitäts- und Prozess-Management) des Projekts „Application to Web“¹.

Das Projekt zielte darauf, insbesondere kleinen und mittleren, aber auch größeren deutschen Unternehmen zu ermöglichen, ihre Anwendungen und Dienstleistungen effektiv und kostengünstig über das Web verfügbar zu machen. Ein wesentlicher Aspekt davon ist die Bereitstellung und Anwendung von entsprechenden Software-Engineering-Methoden. Andere Arbeitspakete dieses Projekts waren: Verbreitung Stand der Technik und Vorbereitung (AP1), Domänen- und Kontext-Analyse (AP2), Standardumgebungen für web-basierte Anwendungen (AP3), CBSE für die Konstruktion web-basierter Anwendungen (AP4), Strategien zur Migration von Altanwendungen (AP5), Fallstudienbearbeitung Analyse und Produktentwicklung (AP7 und 8), Verbreitung der Ergebnisse (AP9) und Projektmanagement (AP10).

Bei der Software-Entwicklung hat das Messen und Bewerten von Prozessen und Produkten die wichtige Funktion, Prozess- und Produkteigenschaften der „Black-Box“ Software transparent zu machen und somit eine Optimierung der Parameter Zeit, Kosten und Qualität zu ermöglichen. Der Entwicklungsprozess ist bei der Software deshalb so wichtig, da er die Produkteigenschaften maßgeblich beeinflusst.

Die hier dokumentierte Methode zum Messen und Bewerten von Webapplikationen adressiert den Bereich „Qualitäts- und Prozessmanagement“.

Unter dem Begriff „Web-Applikationen“ sind die folgenden drei Ebenen angesprochen:

- HTML-Seiten mit Einbettung von Java/JavaScript/CGI-Skripten oder ähnlichem. Diese Seiten sind dynamisch (daher Applikationen) und können bereits umfangreiche Funktionalität bereitstellen.
- Datenbanken mit einer dynamischen Anbindung zur Bereitstellung eines Web-Frontends. Dies kann z.B. über PHP erfolgen. Bei diesen Seiten liegt ein Großteil der Funktionalität in der DB.

¹ Projektplan Application2Web Making Applications fit for the Internet!, Version 1.1 von 16.6.2000

- Komplexe Software-Systeme, die sich aus mehreren Schichten wie Datenbanken, Middleware und der Web-Schnittstelle zum Kunden-Browser zusammensetzen. Dabei können für die DB/Komponenten-Anbindung der Middleware Object Request Broker (CORBA, DCOM), Web Services (J2EE, .NET) die evtl. weitere Web Services nutzen, oder gekapselte Legacy-Applikationen oder -Komponenten benutzt werden. Dies kann über verschiedene Wrapper oder Schnittstellen durch CORBA J2EE (Java und alle Sprachen die durch CORBA unterstützt werden) oder .NET (Windows Komponenten) geschehen.

1.2 Motivation für das „Messen und Bewerten von Web-Applikationen“

Die Erstellung von Software ist ein komplexer, dynamischer Prozess, der aufgrund von vielen Einflussfaktoren und der Involvierung von Menschen in seiner Auswirkung auf die Produkteigenschaften oft wenig verstanden wird.

Neben den wichtigen Prozesseigenschaften „Kosten“ und „Zeit“ für die Erstellung von Software ist die Qualität besonders bei Web-Applikationen wichtig, da sie meist zentral für eine große Anzahl von Benutzern angeboten wird. Dabei spielen insbesondere folgende Qualitätsfacetten eine Rolle. Zuverlässigkeit, Sicherheit und Schutz vor unberechtigtem Zugriff sind insbesondere in den Bereichen von Internet-Banking und Online-Einkaufen wichtig. Zuverlässigkeit, aber auch Antwortgeschwindigkeit ist bei der Nutzung von Online-Services gefragt. Zusätzlich ist die Benutzbarkeit und Benutzerfreundlichkeit der oftmals komplexen Benutzerschnittstellen von großer Wichtigkeit.

All diese Qualitätsanforderungen sollen gezielt bei der Entwicklung von Web-Applikationen berücksichtigt werden. Dabei ist es wichtig, klare Prioritäten für die Qualitätsanforderungen der entsprechenden Applikation zu setzen und zu verfolgen. ‚Priorität‘ bedeutet auch sich bewusst zu entscheiden, nicht alles zu verfolgen, was vielfach auch nicht notwendig ist.

Um sowohl die tatsächliche Qualität als auch die Priorisierung zu verfolgen, ist ein gezieltes Projekt-Management und -Controlling notwendig. Ein wesentliches Hilfsmittel ist das **Messen und Bewerten von Software-Entwicklungsprozessen und -Produkten**. Das gezielte Messen und die systematische Aufbereitung der Resultate kann die folgenden Vorteile für den Prozess und die Produkte bringen:

- *Verständnis* von momentanem Status und Leistung des Prozesses oder der Produkte durch **objektive, quantitative** Aussagen. Messungen schaffen hier eine Transparenz, die anders nicht zu erreichen wäre. Man versteht beispielsweise folgende Größen besser:
 - Anzahl der entdeckten Fehler pro Modul

- Aufwand einzelner Projektphasen
- *Kontrolle* durch Einblick in den Prozessfortschritt oder die aktuellen Produkteigenschaften. Einflussfaktoren können quantifiziert werden. Man kann nicht kontrollieren, was man nicht messen kann!
 - implementierte Funktionalität/Codegröße
 - gefundene Fehler pro Woche
- *Verbesserung* von Prozessen und Produkten
 - Aufzeigen von Ansatzpunkten für Verbesserungen/Erkennen von Schwachstellen bzw. potentiellen Fehlerquellen
 - Überprüfen des Erfolgs einer (Prozess-)Änderung
 - Wirtschaftlichkeitsüberprüfung mit Kosten/Nutzen-Analysen
 - Optimieren der Ausführung (Workflow)

Neben dem Ansatz der Kontrolle und Verfolgung einer vorgegebenen Qualität bzw. gegebenen Qualitätsanforderungen, kann Messen auch Teil eines Softwareprozessverbesserungs-Programms (SPI, Software Process Improvement) sein. Sinn der Softwareprozessverbesserungsinitiative ist es, Verbesserungspotentiale zu erkennen und mit gezielten Maßnahmen umzusetzen. Messungen sind notwendig um diese Verbesserungen konkret aufzuzeigen.

Nur mit der Unterstützung von Messen und Bewerten ist es bei der Entwicklung von Web-Applikationen möglich die gewünschte Qualität in der geplanten Zeit zu erreichen.

1.3 Was ist messen?

Messen ist ein Prozess, bei dem Zahlen oder Symbole (Messwerte) den Attributen (Eigenschaften) von Objekten der realen Welt zugeordnet werden. Diese Zuordnung erfolgt nach klar definierten Regeln, den Metriken.

Die verschiedenen zu messenden Objekte haben bestimmte Eigenschaften, denen beim Messen ein Wert zugeordnet wird. Nicht alle Werte sind notwendigerweise Zahlen. Sie sind jedoch immer einer definierten Skala zugeordnet (wie z.B. auch „hoch“, „mittel“, „gering“ eine Ordinalskala repräsentiert). Die Werte stehen über die Messung im Bezug zum Messobjekt, was eine Interpretation der Werte im Hinblick auf das Objekt und seinen Kontext ermöglicht.

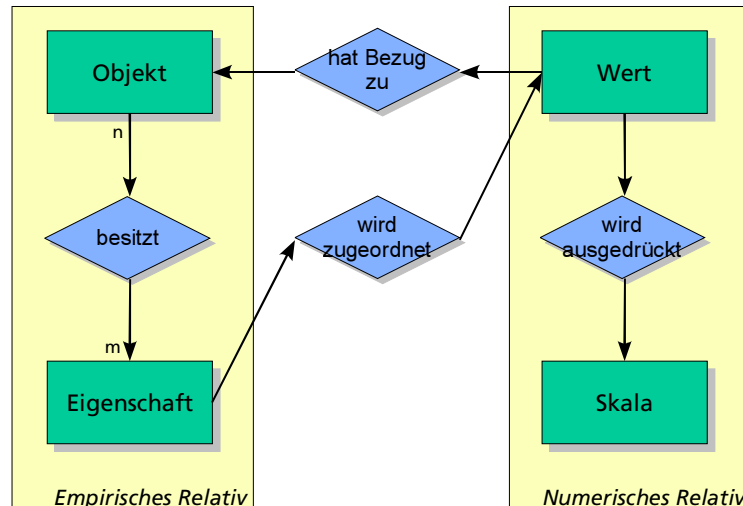


Abbildung 1

Beziehungen beim Messen

Dieser Zusammenhang wird in Abbildung 1 anschaulich dargestellt. Messen bildet Erfahrung und Beobachtung (Empirisches Relativ) auf Zahlenwerte mit einer geeigneten Skala (Numerisches Relativ) ab. Messen erfolgt also „objekt-orientiert“. Die Zahlenwerte bilden dann das Maß oder die Metrik.²

Ein Beispiel: Der Aufwand einer bestimmten Phase des Entwicklungsprozesses kann wie folgt gemessen werden:

Objekt: Phase; *Eigenschaft:* Aufwand; *Wert:* 57; *Skala:* Personentage

1.4 Worauf es beim Messen ankommt

Die folgenden vier Punkte charakterisieren wesentliche Faktoren, die den Erfolg von Software-Messungen maßgeblich bestimmen:

- 1.) Messen erfordert Unterstützung durch das Management/die Projektleitung. Messen benötigt eine Infrastruktur!
- 2.) Allen Beteiligten muss klar sein, weshalb gemessen wird bzw. wofür die Ergebnisse verwendet werden sollen.
- 3.) Die erfassten Daten müssen korrekt sein und möglichst vollständig, damit sie für eine gültige Interpretation/Auswertung herangezogen werden können.

² Maße oder Maßzahlen beziehen sich meist auf Basisdaten, die direkt erhoben werden. Metriken beinhalten auch berechnete und zusammengesetzte „Maße“.

- 4.) Die erfassten Daten werden im Hinblick auf die Messziele tatsächlich ausgewertet und interpretiert. Die Ergebnisse werden zur Erreichung der Messziele/Verbesserungsziele genutzt. So wird auch der Beitrag der Messung zur Zielerreichung klar.

1.5 Zweck dieses Handbuchs

Dieses Handbuch soll Leser dabei anleiten, das Messen und Bewerten von Software-Prozessen und -Produkten im eigenen Projekt einzuführen.

Als methodische Grundlage wird ein zielorientiertes Vorgehen gemäß dem GQM-Paradigma verwendet (Goal-Question-Metrik, Ziel-Frage-Metrik). Konkrete Beispiele für Ziele, Messfragen und Metriken werden für den Bereich der Web-Applikationen angegeben.

1.6 Struktur dieses Handbuchs

Im Kapitel 2 wird ein Überblick für eine allgemeine Vorgehensweise nach der „GQM-Methode“ vorgestellt. Die nachfolgenden Kapitel beschreiben wesentliche Schritte daraus:

- Kapitel 3: Planung/Vorbereitung
- Kapitel 4: Instrumentierung
- Kapitel 5: Datenerfassung
- Kapitel 6: Analyse und Interpretation, Erfahrungssicherung

Die Durchführung von Maßnahmen aufgrund der Messergebnisse wird in Kapitel 7 adressiert. Eine Liste zu weiterführender Literatur wird im Abschnitt Referenzen gegeben. Der Anhang (Anhang A bis Anhang E) enthält Vorlagen und ausführliche Beispiele.

2 GQM als Methode zum Messen und Bewerten

2.1 Einleitung

Wie misst man? Auf diese Frage gibt es viele Antworten, wir wollen uns hier jedoch nur auf ‚zielorientiertes‘ Messen konzentrieren. Zielorientiert bedeutet, dass das Messen nicht Selbstzweck ist, sondern zur Erreichung eines (übergeordneten) Zieles dient. Es werden systematisch und top-down – ausgehend von Geschäfts- oder Projektzielen – Verbesserungsziele definiert, von denen dann Messziele und damit das gesamte Messprogramm abgeleitet wird. Dahinter steht der Gedanke (und die Erfahrung), dass es in der Regel keine Menge von Metriken gibt, die für alle Organisationen und Projekte passen. Aus dieser Vorgehensweise leitet sich auch der Name der Methode ab: GQM – von den Zielen (**G**oals) über Fragen (**Q**uestions) zu den Metriken (**M**etrics).

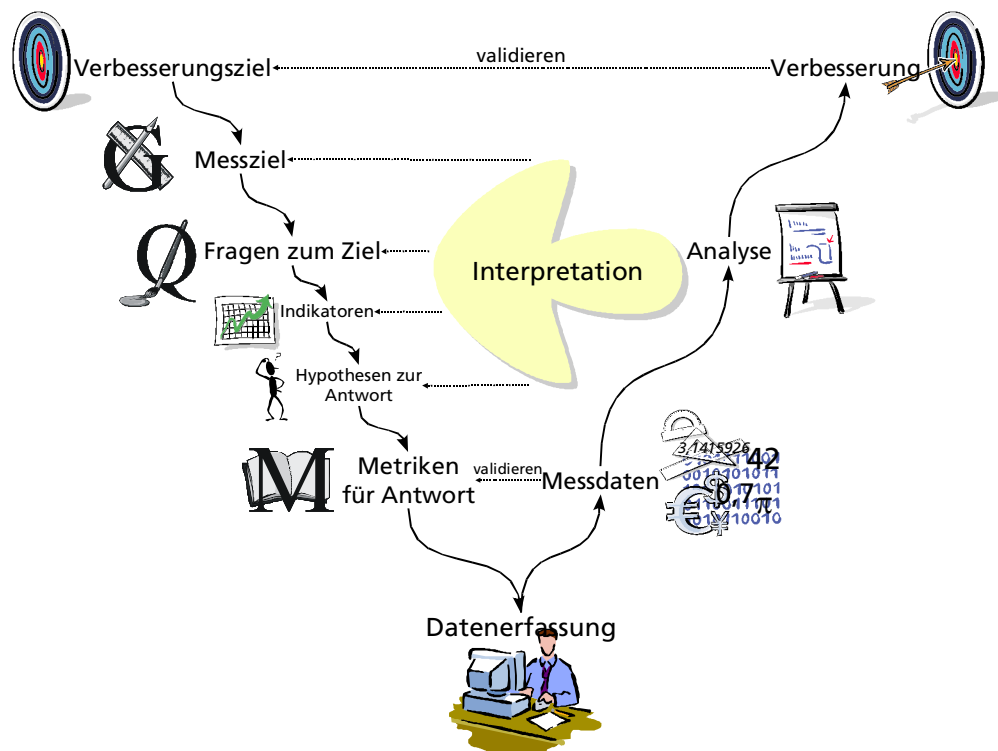


Abbildung 2

GQM zum zielorientierten Messen bei der Software-Erstellung

Das Messen ist auf die Analyse ausgerichtet und aus der Analyse der Messung können dann Verbesserungen abgeleitet werden. Wie in Abbildung 2 dargestellt handelt es sich dabei um ein mehrstufiges Verfahren.

Für jedes Verbesserungsziel wird ein sogenanntes „Messziel“ definiert. Um das Verständnis des Messziels zu erhöhen und es operationalisierbar zu machen, werden Fragen zu dem Messziel formuliert. Diese Fragen helfen ein Verständnis dafür zu bekommen, wann das Messziel erreicht ist. Die Beantwortung der Fragen dient der Zielüberprüfung.

Sind die Fragen definiert, können sie mit Hilfe von ‚Indikatoren‘ (Diagrammen) visualisiert werden. Dies erlaubt schon zu dem Zeitpunkt der Definition ein Gefühl dafür zu bekommen, wie letztendlich die Beantwortung (aufgrund der Messdaten meist ein Diagramm) der Frage aussehen wird. Zum Festhalten der Erwartungen und Einschätzungen bezüglich der Antwort auf die Frage, werden an dieser Stelle schon Hypothesen der Antworten (Messergebnisse) aufgestellt. Die Hypothesen können gut in die Indikatoren einfließen. Auf der Basis der Fragen werden letztendlich die Metriken – die tatsächlichen Messinhalte – bestimmt. Nun beginnt die eigentliche Datenerfassung: das Messen. Durch die sorgfältige Auswahl der Metriken wird sichergestellt, dass keine unnötigen Daten erhoben, nur wirklich relevante Antworten gegeben werden und somit auch keine unnötigen Kosten entstehen. Die Daten werden in Feedback-Sitzungen analysiert, um sie zu validieren, Verbesserungsmaßnahmen einzuleiten und eventuell die Wirkung von Verbesserungsmaßnahmen zu kontrollieren.

Die nachfolgenden Kapitel beschreiben, wie dieser Ansatz genutzt wird, um ein zielorientiertes Messprogramm durchzuführen. Dabei wird sowohl der einfache Fall, in dem existierende Elemente (vgl. Abbildung 3) direkt eingebunden werden, behandelt, als auch das Aufsetzen einer neuen Mess-Infrastruktur.

Der Erfahrung nach kann der Prozess in vier Hauptphasen gegliedert werden:

- Planung des Messens
- Instrumentierung (Prozesse, Produkte, Ressourcen und Projekte werden für das Messens instrumentiert)
- Datenerfassung
- Datenanalyse und Erstellung des Ergebnisberichts

Diese vier Phasen werden in den folgenden vier Kapiteln ausführlich behandelt. Der folgende Abschnitt gibt eine kurze Einführung in die Zusammenhänge und klärt Begrifflichkeiten.

2.2 Der Rahmen eines Messprogramms

Der im Folgenden dargestellte Rahmen für ein Messprogramm beschreibt die wesentlichen Schritte, die von einer Mess-Infrastruktur unterstützt werden müssen. Der Rahmen gibt einen Überblick über den Prozess und seine Schnittstellen, d.h. wie die Messtätigkeiten in die Software-Entwicklung eingebettet sind.

Abbildung 3 zeigt eine Übersicht über die vier Hauptschritte beim Messen: Planung, Instrumentierung, Datensammlung und Analyse/Ergebnisbericht, sowie die dazugehörigen Produkte. Die externen Schnittstellen werden als Piktogramme dargestellt. Sie sind zwar wesentliche Anforderungen/Bestandteile des Messens, werden aber nicht zentral für ein Messprogramm erstellt und bilden daher keinen Schwerpunkt dieses Handbuchs.

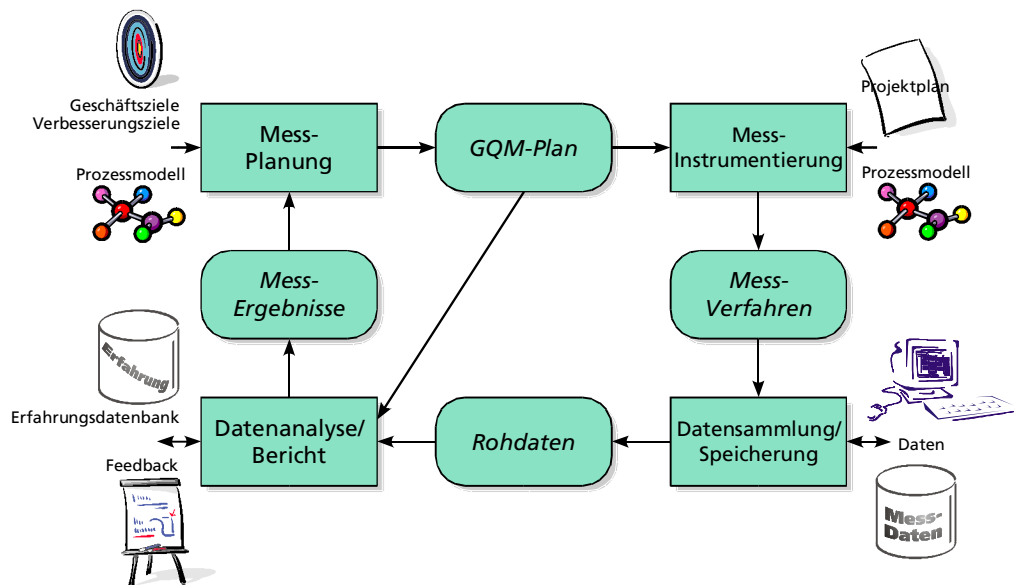


Abbildung 3

Die Hauptelemente eines Messprogramms

Der Messprozess muss auch immer im Zusammenhang mit dem zugehörigen Entwicklungsprozess gesehen werden. Abbildung 4 skizziert diesen Zusammenhang. Drei Punkte sind zu betonen:

- 1.) Planung und Instrumentierung wird bereits vor dem Entwicklungsprozess durchgeführt
- 2.) Datenanalyse und das Erstellen von Berichten erfolgt parallel zur Entwicklung (und nicht erst am Ende)
- 3.) Datenanalyse und Berichte reichen über den Entwicklungsprozess hinaus

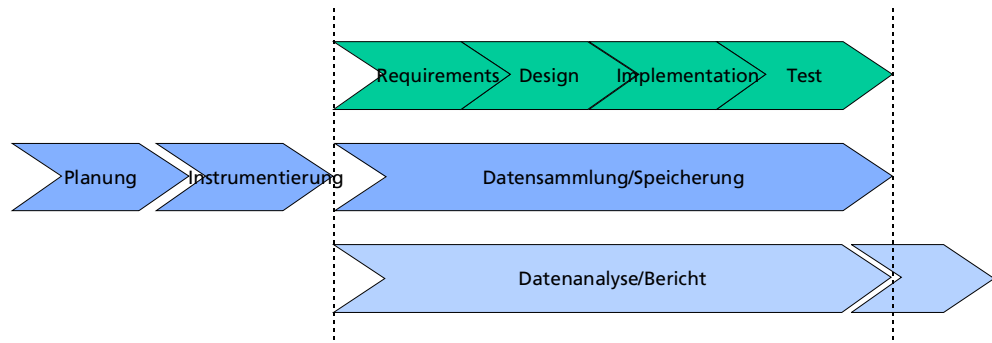


Abbildung 4

Zeitlicher Zusammenhang zwischen dem Messprozess und dem Entwicklungsprozess

Die Schnittstellen zwischen diesen Arbeitsschritten bestehen aus Produkten, die für die Messplanung und Ausführung gebraucht werden. Das folgende Referenzmodell zeigt in Abbildung 5 diesen Produktfluss.

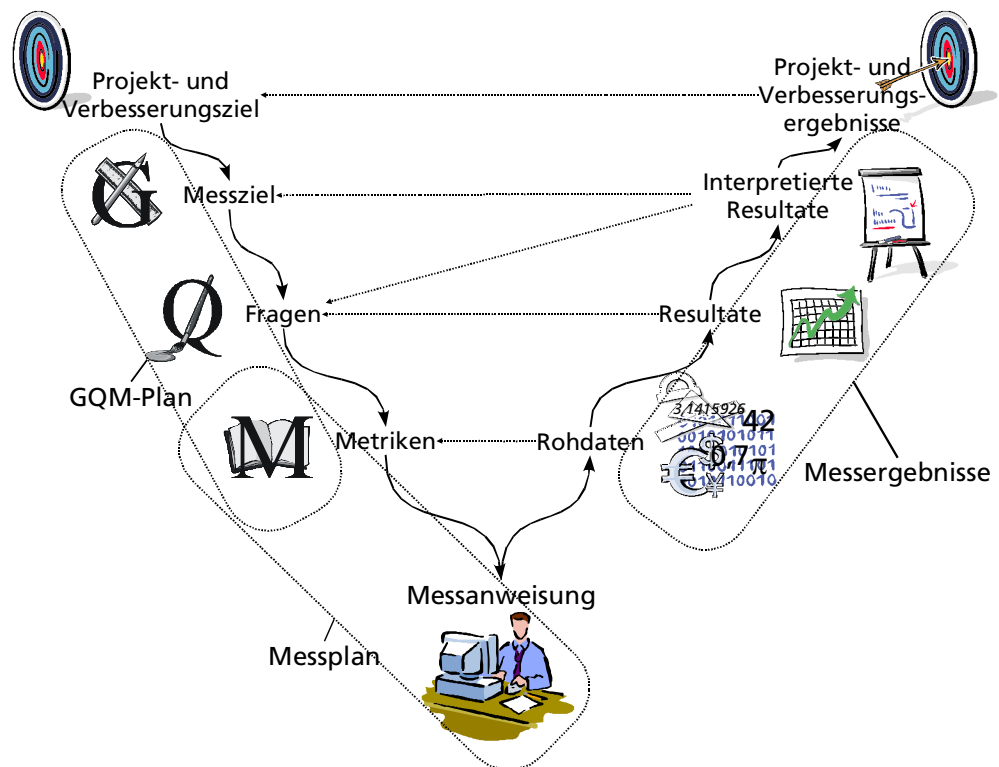


Abbildung 5

Produkte in einem Messprogramm

Messen basiert auf Projekt- und Verbesserungszielen und soll unmittelbare Ergebnisse für Projekte und ihre Verbesserung bringen, als auch mittelbar die Abläufe und Prozesse der Organisation verbessern. Der Ablauf innerhalb dieses

Referenzmodells folgt der Ableitung der Geschäfts- und Prozessziele zu Messzielen. Aus den Messzielen werden die Fragen gebildet und aus den Fragen lassen sich wiederum Metriken ableiten. An dieser Stelle kommen noch die Produkte (z.B. Dokumente, Daten), wie sie in den folgenden Kapiteln gebraucht werden, hinzu. Die *Messziele*, die *Fragen* (mit ihren Erweiterungen wie Hypothesen und Visualisierungen) und die Maße (*Metriken*) werden unter dem Begriff des *GQM-Plans* zusammengefasst. Die Erweiterung der Metriken um die *Messanweisungen* Messen nennt man den *Messplan*.

Die erforderlichen Daten werden gemäß definierter *Messanweisungen* (Datenerfassung wie z.B. Ausfüllen eines wöchentlichen Fragebogens) erfasst. Unter dem Begriff *Messergebnisse* werden alle entstehenden Ergebnisse zusammengefasst. Dies schließt vor allem auch die *Rohdaten* ein. In einem ersten Schritt, werden die Fragen durch die *Resultate*³ beantwortet. Die eigentlichen Ergebnisse bilden dann die *interpretierten Resultate*.

³ Die Resultate entsprechen den Indikatoren, jedoch jetzt auf Basis echter Daten.

3 Messplanung

Die Planung ist die Grundlage des Messprogramms. An dieser Stelle wird die Messung mit ihren Zielen, Methoden und Werkzeugen festgelegt. Diese Planung kann (und muss) dynamisch den sich ändernden Projektbedingungen angepasst werden. Da eine gute Planung von großer Bedeutung für das Gelingen der Messung ist, wird ihr in diesem Handbuch viel Raum gewidmet.

Ziel der Planung ist es, den Umfang des Messprogramms abzustecken. Sie umfasst die Definition

- der Messziele
- der relevanten Fragen, um die Ziele zu erreichen
- der Hypothesen um die Vorstellung/Erwartungen von den Messergebnissen festzuhalten
- der Maße (Metriken), die erlauben die Fragen zu beantworten.
- Veranschaulichungen für die Daten (Indikatoren), z.B. Diagramme oder Tabellen

Diese bilden zusammen den GQM-Plan. Ziele der Organisation und des Projektes, aber auch Vorgehensbeschreibungen oder Prozesshandbücher dienen als Ausgangspunkt für die Planung. Abbildung 6 zeigt eine vereinfachte Darstellung einer UML-Spezifikation eines GQM-Plans.⁴

⁴ Die Verfeinerung von Fragen durch Fragen wurde der Übersichtlichkeit wegen nicht modelliert.

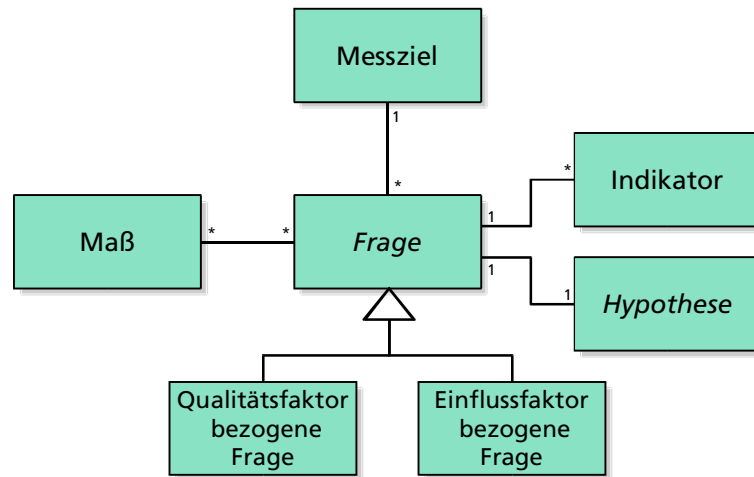


Abbildung 6

UML Definition einer Mess-Spezifikation

Einem Messziel sind eine oder mehrere Fragen zugeordnet. Man kann zwei Klassen von Fragen unterscheiden: Fragen können entweder einen direkten Bezug auf den Qualitätsaspekt haben (Qualitätsfaktor, z.B. Frage nach Anzahl der Fehler für den Qualitätsaspekt der Zuverlässigkeit) oder aber wichtige Einflüsse auf den Qualitätsfaktor adressieren (Einflussfaktor, z.B. Erfahrung der Entwickler).

Es wird auch definiert, wie die Daten, die für das Messen gesammelt werden, aggregiert und verarbeitet werden müssen, damit eine Visualisierung der Fragen (d.h. Darstellung der Ergebnisse in einem Diagramm, z.B. Tortengrafik) möglich ist.

Abbildung 7 gibt einen Überblick über die Schnittstellen der Messplanung.

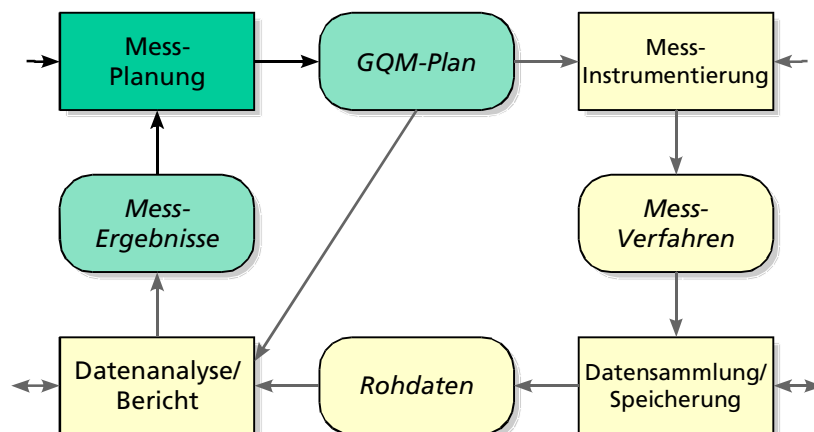


Abbildung 7

Schnittstellen der Planung

Die Struktur in diesem Kapitel folgt der Verfeinerungsstruktur für Ziele (siehe auch Abbildung 8).

- Definition von Geschäfts- und Verbesserungszielen
- Definition von Messzielen
- Definition von Fragen (einschließlich der Indikatoren und Hypothesen)
- Definition von Metriken

Diese einzelnen Punkte werden in den folgenden Abschnitten behandelt und anhand von Beispielen veranschaulicht. Die dargestellten Tipps weisen auf Erfahrungen hin, die beim praktischen Einsatz gemacht wurden und sich als zeit- und kostensparend erwiesen haben.

3.1 Definition von Geschäfts- und Verbesserungszielen



Messungen sollen Ergebnisse liefern, die es erlauben einen Prozess zu verstehen, zu steuern oder zu optimieren, und damit ein Produkt zu verbessern (siehe auch Abschnitt 1.2). Dazu müssen die Ziele der Messung mit den Zielen der Organisation, in der die Messung vorgenommen wird, verknüpft sein. Die Ziele der Organisation kann man als ein *Mission Statement* verstehen, sie können aber auch mit einem spezifischen Problem oder Wunsch verbunden sein, z.B. die Umstrukturierung der Firma, das Erreichen eines kürzeren *Time-to-Market* der Produkte.

Ein Organisationsziel kann sich auch auf ein Entwicklungsprojekt beziehen. Wenn beispielsweise die Firma dem Kunden Dienstleistungen direkt anbieten will und diese Angebote über das Web verfügbar macht. Das Ziel des Entwicklungsprojektes könnte somit sein: entwickeln von effektiven und kostengünstigen Web-Applikationen.

Ein entsprechenden Organisationsziel würde in diesem Fall als Projekt- oder Abteilungsziel adressiert. Für die Messung ist die Ebene (Organisation/Abteilung) nicht von so großer Bedeutung. Ein Organisationsziel wird als vage Beschreibung benötigt, die den einzuschlagenden Weg vorgibt.

In Abbildung 8 wird skizziert, wie aus dem Organisationsziel das Messziel (es kann auch mehrere geben), die Fragen und die Metriken abgeleitet werden. Das Ziel der Messung wird dabei in messbaren Begriffen beschrieben (siehe dazu auch den folgenden Abschnitt 3.2).



Verbesserungsziele können aus bekannten konkreten Problemen abgeleitet werden. Ein Problemfindungs-Workshop kann helfen, bekannte Probleme zu sammeln und hinsichtlich Ihrer Schwere/Dringlichkeit zu priorisieren.

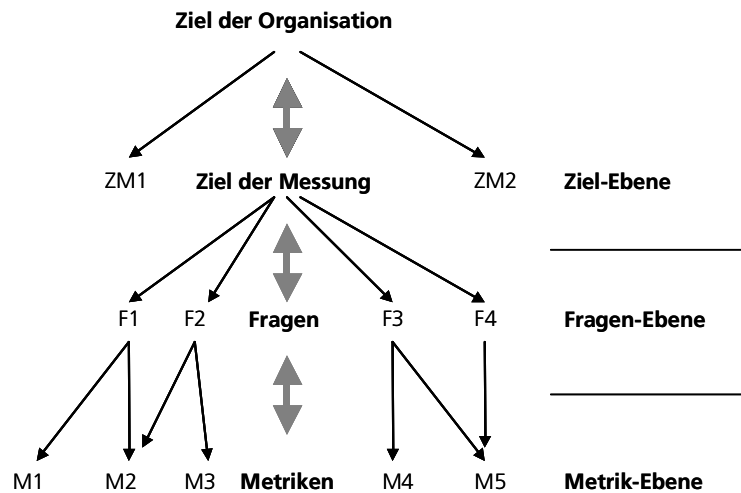


Abbildung 8

Die unterschiedlichen Ebenen bei einer Messung

Ein Hilfsmittel zur Unterstützung bei der Definition und Auswahl der Messziele, kann die Nutzung der „sieben Fragen“ sein:

- Was sind die strategischen Ziele der Organisation?
- Was hat einen Einfluss auf die strategischen Ziele der Organisation?
- Wie kann die Performance verbessert werden?
- Was sind die größten Herausforderungen/Probleme?
- Was sind die Verbesserungsziele?
- Wie können die Verbesserungsziele erreicht werden?
- Was sind mögliche Messziele und ihre Prioritäten?

3.2 Definition von Messzielen



Messziele müssen auf einer Abstraktionsebene bestimmt und beschrieben werden, so dass die Zielerreichung durch eine Analyse der gesammelten Daten überprüft werden kann. Daher sollten Messziele bestimmte Informationen enthalten. Dazu wird das Ziel am besten in mehreren kurzen Sätzen beschrieben. Dabei sollten die folgenden Aspekte berücksichtigt werden:

- Objekt: Was ist das Objekt der Analyse?
 - Produkt, Prozess: z.B. Dokument, Kode, Anforderungsprozess
- Zweck: Warum und zu welchem Zweck wird dieses Objekt analysiert?
 - charakterisieren, evaluieren, verbessern, kontrollieren, vorhersagen

- Qualitätsfokus: Welche Attribute des Objektes werden analysiert?
 - z.B., Kosten, Effektivität, Zuverlässigkeit, Qualität, Wartbarkeit, Dauer/Laufzeit, ...



Sämtliche verwendeten Begriffe, wie z.B. „Qualität“ sollten eindeutig definiert sein, damit alle Beteiligten dasselbe Verständnis von Ihnen haben. Dies ist insbesondere wichtig für die Datenerfassung und die spätere Interpretation.

- Blickwinkel: Wer ist der Interessensnehmer (stakeholder) für das Ziel, wessen Sicht spiegelt sich wider?
 - Entwickler, IT-Manager, Kunde, Testgruppe
- Kontext: In welchem Kontext wird gemessen?
 - Projekt X, Abteilung Y, Produkt Z

Im folgenden Abschnitt wird zunächst eine Notation – *Abstraction Sheet* – zur Ermittlung der Ziele und wesentlicher Faktoren und Zusammenhänge vorgestellt. Anschließend werden die fünf Aspekte im Detail beschrieben.

3.2.1 Festlegen von Messzielen

Mögliche Messziele müssen sehr sorgfältig und unter Beteiligung aller vom Messprogramm betroffenen Parteien aufgestellt werden. Die Einbindung aller Beteiligten dient zum einen dazu, alle wichtigen Aspekte im Messziel zu berücksichtigen, zum anderen hilft es, die Personen bereits bei der Definition des Messprogramms zu beteiligen, es zu „ihrem“ Messprogramm zu machen. Dies ist wichtig, da eine Messung nur dann akzeptiert wird, wenn die Interessen aller beteiligter Personen (z.B. Vertreter des Managements, Entwickler) im Messprogramm berücksichtigt wurden.



Am Anfang nicht mehr als ZWEI oder DREI Messziele gleichzeitig verfolgen!

Starten Sie klein – Sie werden ohnehin nur eine begrenzte Anzahl von Verbesserungsmaßnahmen umsetzen können.

Bevor Messziele formal definiert werden, sollten zunächst alle möglichen Messziele in einem Brainstorming gesammelt werden. Dieses Brainstorming kann entweder in einem Workshop oder durch vier bis fünf Interviews geschehen. In jedem Fall ist es wichtig, dass die Ziele von verschiedenen Prozessbeteiligten erhoben werden, um einen Gesamtblick zu erhalten. Die Dokumentation der Ziele erfolgt im *Abstraction Sheet*. Typischerweise entstehen mehrere (unter Umständen widersprüchliche) *Abstraction Sheets*, die im GQM-Plan konsolidiert werden.



Das Erfassen der Informationen auf einem Blatt Papier hat sich insbesondere dann bewährt, wenn die Informationen mit Interviews gesammelt werden. Alternativ kann man im Rahmen eines Workshops auch ein Flip-Chart verwenden.

Nach einer Priorisierung werden dann die zwei oder drei Ziele ausgewählt, die weiter verfeinert und verfolgt werden sollen. Um die informell beschriebenen Ziele zu formalisieren können die Vorlagen aus dem nächsten Abschnitt verwendet werden.

Ausfüllen eines *Abstraction Sheet*

Für das Sammeln von relevanten Aspekten zu einem Messziels kann ein sogenanntes *Abstraction Sheet* genutzt werden, um unterschiedliche Arten von Informationen zu gruppieren.

Man unterscheidet Aspekte⁵, die direkt dem Qualitätsfokus zuzuordnen sind von Aspekten, die einen Einflussfaktor beschreiben, welcher einen Qualitätsfaktor beeinflusst (engl. *variation factor*). Einflussfaktoren sind als sekundär zu betrachten. Einflussfaktoren sollten immer genannt werden. Es besteht immer die Möglichkeit sie später bewusst nicht zu messen, dennoch sollten sie nicht einfach weggelassen werden. Ist der erwartete Einfluss auf einen Qualitätsfaktor jedoch groß, so sollte der entsprechende Einflussfaktor mit erfasst werden. Daneben sollten immer Hypothesen über die Ausprägung eines Qualitätsfaktors oder die Art und Weise eines Einflussfaktors genannt werden. Diese sind auch zu dokumentieren. Beim Einflussfaktor ist zuerst die Hypothese des Einflussfaktors anzugeben und dann die Art des Einwirkens auf den Qualitätsfaktor.



Es gibt immer Einflussfaktoren! Auch wenn diese nicht gemessen werden, sollte man sich ihrer bewusst sein.

Das *Abstraction Sheet* dient der gemeinsamen Darstellung dieser unterschiedlichen Kategorien, wobei die Bezüge untereinander mit erfasst werden (z.B. durch entsprechende Nummerierung oder Pfeile). Eine präzise Darstellung der Beziehungen und Abhängigkeiten der GQM-Fragen, Hypothesen und Faktoren untereinander ist in Abbildung 9 gegeben.

⁵ Fragen, die genutzt werden um diese Aspekte zu benennen werden detailliert im nächsten Abschnitt behandelt. Allerdings lässt sich die Erfassung mit dem *Abstraction Sheet* nicht strikt von Fragen trennen. Man kann auch konkrete Fragen eintragen.

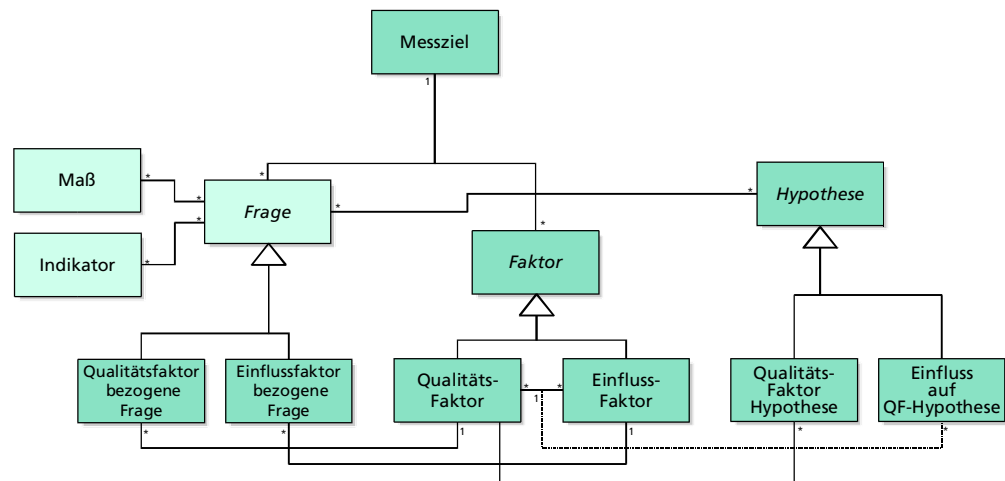


Abbildung 9

Beziehungen zwischen Hypothesen und Qualitäts- und Einflussfaktoren

Qualitätsfaktor: Was sind die gemessenen Eigenschaften von dem Objekt nach Meinung der Projektteilnehmer?

Hypothese: Was ist das momentane Wissen der Projektteilnehmer bezüglich dieser gemessenen Eigenschaften?

Einflussfaktoren: Von welchen (Umgebungs-)Faktoren erwarten die Projektteilnehmer, dass sie einen Einfluss auf die gemessenen Eigenschaften haben?

Einfluss auf Hypothese: Wie beeinflussen die Einflussfaktoren die gemessenen Eigenschaften?

Objekt	Zweck	Qualitätsfokus	Perspektive	Kontext
QUALITÄTSFAKTOREN: Welche Faktoren/Metriken müssen betrachtet werden?		EINFLUSSFAKTOREN: Welche Variablen haben einen Einfluss auf die Qualitätsfaktoren?		
HYPOTHESE: Was sind die momentanen Erwartungen bezüglich der Qualitätsfaktoren?		EINFLUSS AUF HYPOTHESE: Wie beeinflussen die Einflussfaktoren die Qualitätsfaktoren?		

Tabelle 1

Abstraction Sheet zur Unterstützung der Kommunikation zwischen den Beteiligten bei der Definition von den Faktoren und Hypothesen

Ein Beispiel für ein *Abstraction Sheet*

Das hier gezeigte Beispiel für ein *Abstraction Sheet* bezieht sich auf das Messen und Bewerten von Fehlern. Ausgangspunkt waren die folgenden Fragen zu den Qualitätsaspekten:

- Wie viele Fehler wurden in den folgenden Prozessen gefunden (absolut und relativ für ein Release): Komponententest, Code-Review, Regressionstests, GUI-Test, GUI-Regressionstest, Bug-Review, Produktion? (Fehlerattribute)
- Was war die **Fehlerursache** für die in den Prozessen Komponententest, Regressionstest, GUI-Test, GUI-Regressionstest, Produktion entdeckten Fehler? (Verteilung und Gegenüberstellung Fehler je Prozess)

Anhand dieser Fragen wurde das *Abstraction Sheet* in Tabelle 2 definiert.

Objekt	Zweck	Qualitätsfokus	Perspektive	Kontext
QS-Aktivitäten	Charakterisierung	Effektivität	SW Entwickler	Firma x
QUALITÄTSFAKTOREN		EINFLUSSFAKTOREN		
QF1: Gesamtzahl von Fehlern QF2: Verteilung der Fehler auf Problemklassen QF3: % der Fehler, die bei Inspektionen gefunden wurden QF4: Gesamtaufwand für Korrektur (in Stunden) ...		EF1: Typ der Code-Inspektion EF2: Erfahrung des Entwicklungsteams ...		
HYPOTHESE		EINFLUSS AUF HYPOTHESE		
QF1: Gesamtzahl: 100 QF2: 75% unkritisch, 25% kritisch QF3: 20% QF4: Gesamtaufwand: 1000 h ...		EF1 zu QF1: Inspektionstechnik hat Einfluss auf die Anzahl der gefundenen Fehler EF1 zu QF2: Inspektionstechnik hat Einfluss auf die Art der gefundenen Fehler EF1 zu QF3: <i>ad hoc</i> Inspektionen sind weniger effektiv als andere Typen EF1 zu QF4: Anzahl und Art der bei Inspektionen gefundenen Fehler hängen vom Inspektionstyp ab (Anzahl und Art der Fehler beeinflussen direkt den Korrekturaufwand) EF2 zu QF1: Erfahrenere Entwickler fügen weniger Fehler ein ...		

Tabelle 2

Beispiel für ein *Abstraction Sheet*

3.2.2 Festlegen des Messobjekts

Das Objekt der Messung festzulegen und zu beschreiben bedeutet festzuhalten, über was man etwas erfahren möchte. Im Folgenden werden zwei Möglichkeiten zur Beschreibung von Messobjekte angegeben. Zuerst ein generelles Modell mit Beispielen für Metriken des Objektes (vgl. auch Abbildung 10):

- Prozess: Aufwand (gemessen in ‚Personentagen‘, angewendet vom Anfang bis zum Ende des Prozesses), Zeitdauer, Konformität zum Standardvorgehen
- Produkt: Umfang (z.B. Seitenzahl bei Dokumenten, Kode-Zeilen bei Software), Zuverlässigkeit (Ausfallhäufigkeit, Downtime)
- Ressource: Erfahrung von Projektmitarbeitern (‚Zahl der bisher durchgeführten Projekte‘)
- Projekt: ‚Earned Value‘ (Wertschöpfung)

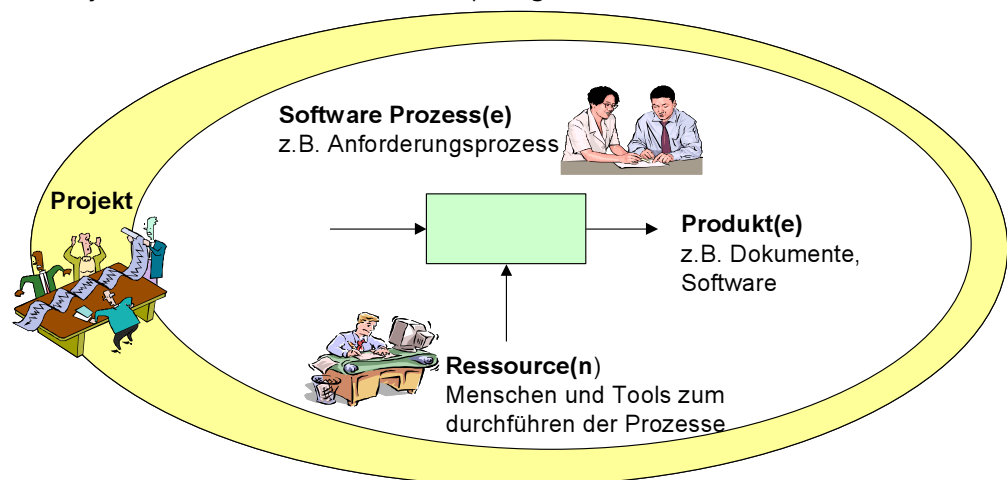


Abbildung 10

Zusammenhang zwischen Organisation, Projekt und dem Prozess mit seinen Elementen

Das zweite Modell zielt auf Webapplikationen ab und nutzt die spezifischere Aufteilung von Produkten an Hand der Tier-Struktur, wie sie in Abbildung 11 gezeigt wird. Dies ist ein reines Produktmodell, somit kann es nur genutzt werden, falls das Objekt ein Produkt sein soll.

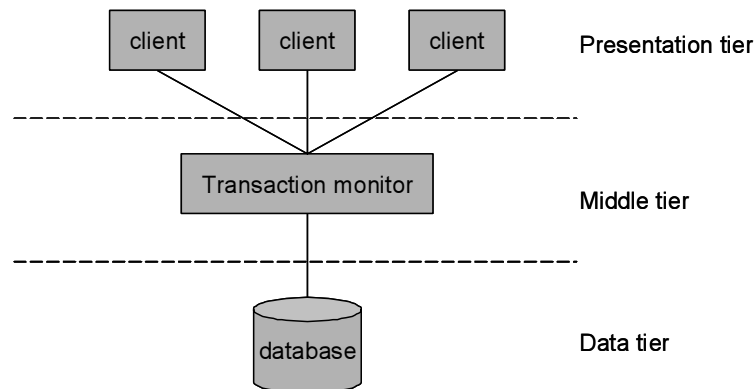


Abbildung 11 Typischer Aufbau einer Tier-Struktur

3.2.3 Festlegen des Zwecks

Mögliche Zwecke für ein Messziel sind charakterisieren (verstehen), verbessern (erklären), überwachen/kontrollieren, auswerten und vorhersagen. Die verschiedenen Zwecke bauen aufeinander auf. Bei einem Messprogramm wird man mit charakterisieren beginnen, wenn nur wenig Wissen vorhanden ist. Für das Projektmanagement ist überwachen und kontrollieren ein wichtiger Zweck. Mit verbessern werden wird eine langfristige Optimierung der Prozesse angestrebt. Dies setzt jedoch ein gutes Verständnis und Management der Prozesse voraus.

3.2.4 Festlegen des Qualitätsfokus

Der Vorteil des GQM-Ansatzes ist, dass nicht auf vorgefertigte Ziele/Objekte/Qualitätsfoki etc. zurückgegriffen werden muss. Der GQM-Plan mit all seinen Elementen wird individuell definiert. Letztendlich ist der Qualitätsfokus ein Element des Messziels (siehe Abschnitt 3.2). Jede Vorgabe kann daher nur unterstützend sein und sollte nicht als Ausgangspunkt zur Zielfindung dienen.

Als Anregung folgt eine Übersicht von Qualitätsattributen für Software, die auf Standards basiert. Der Schwerpunkt wird dabei auf spezifische Charakteristika für Web-Applikationen gelegt. Die meisten der genannten Qualitätsfoki beziehen sich auf das Produkt. In GQM kann der Qualitätsfokus sich auf das Produkt und auf den Prozess beziehen.

- Zuverlässigkeit/Verfügbarkeit⁶ – wichtig, weil Webseiten die von unzuverlässiger Software abhängen, von Kunden nicht wiederholt besucht werden.

⁶ Verfügbarkeit ist gemäß ISO 9126 synonym für Zuverlässigkeit

Dabei geht es um die 24/7/365⁷ Zugreifbarkeit der Webseite und das auch über verschiedene Browser (keine Produktspezifität)

- Benutzbarkeit – beschreibt die Eigenschaft der Applikation vom Benutzer verstanden, gelernt, benutzt und als angenehm empfunden zu werden, wenn sie wie spezifiziert benutzt wird. Die Applikation soll dem Benutzer erlauben, zu verstehen, ob und wie die Software für eine bestimmte Aufgabe genutzt werden kann. Sie unterstützt den Benutzer auch im Erlernen des Umgangs mit der Software. Weiterhin wird dem Benutzer ermöglicht die Applikation zu benutzen und zu kontrollieren, als auch den Umgang mit ihr als angenehm zu empfinden. Benutzbarkeit ist ein wichtiges Qualitätsattribut für Webapplikationen, weil die Benutzer wenig Bindung zu einer Webseite zeigen. Wenn eine Seite nicht nutzbar ist, werden die Benutzer schnell Alternativen suchen.
- Sicherheit (security) – ist eine Eigenschaft der Applikation, Informationen und Daten so zu schützen, dass unautorisierte Personen sie nicht lesen oder ändern können, aber autorisierten Personen oder Systemen der Zugriff nicht verwehrt wird. Weiterhin gehört Sicherheit auch in den Bereich von Vertrauen und Privatsphäre von Daten. Sicherheit ist ein wichtiges Thema für Webapplikationen, denn Webseiten (oder Anwendungen) müssen mit Kundendaten und allen anderen elektronischen Informationen so sicher und geschützt wie möglich umgehen.
- Funktionale Sicherheit (safety) – die Einhaltung dieses Kriteriums verhindert die physische Verletzung oder Schädigung der Gesundheit von Menschen, entweder direkt oder indirekt als ein Ergebnis von Schäden von Gütern oder der Umwelt. Hier gibt z.B. die Norm ISO 61508 Maßgaben für die Ausfallsicherheit sicherheitsrelevanter Komponenten vor. Dieser Aspekt von Sicherheit ist für Webapplikationen eher weniger von Belang.
- Skalierbarkeit – die Anwendung muss so entworfen sein, dass sie schnell mitwachsen kann, was die Anzahl der Nutzer, aber auch die Anzahl der angebotenen Dienste angeht. Zum Beispiel sollte eine Webanwendung in der Lage sein 10 bis 1.000.000 Zugriffe pro Tag zu bewältigen.
- Wartbarkeit – beschreibt die Eigenschaft einer Anwendung verändert werden zu können. Änderungen können Korrekturen, Verbesserungen oder Anpassungen der Software an Veränderungen im Umfeld oder den Anforderungen sein. Wartbarkeit ist ein wichtiges Qualitätsattribut für eCommerce-Anwendungen, weil Webapplikationen flexibel sein müssen, um sie ändernden Geschäftszielen und Umgebungen anpassen zu können. Auch kundenspezifische Anforderungen für Performance, Interaktivität und Qualität des Dienstes (QoS – Quality of Service) müssen abgedeckt werden.

⁷ 24 Stunden am Tag/7Tage die Woche/365 Tage im Jahr (außer in Schaltjahren)

- Anpassbarkeit – dieses Kriterium beschreibt die Eigenschaft einer Anwendung, sich auf den Benutzer einzustellen und von ihm anpassen zu lassen. Dies kann die Gestaltung der Benutzeroberfläche beinhalten als auch die Unterstützung von Abläufen in einer benutzerspezifischen Reihenfolge oder das Bereithalten von häufig benutzten Daten und Funktionen.
- ‚Time to market‘ – dies ist keine Eigenschaft des Produktes, sondern eher des Prozesses. Dennoch ist ein wesentlicher Faktor bei jeder Entscheidung für ein Produkt, ob und wann es auf dem Markt verfügbar ist.
- Performance – bei Webanwendungen ist in der Regel der Einfluss des Netzwerks auf die Performance größer als durch die Software. Dennoch kann bei stark frequentierten Sites der Server der Engpass sein. sollte bei der Wahl der Algorithmen, bei der Planung der zu übertragenden Datenmenge und bei der Netzanbindung (Wahl des Providers/Netzarchitektur) an diesen Punkt gedacht werden (siehe auch *Effizienz*).
- Effizienz – die Anwendung zeigt eine angemessene Performance, in bezug auf die genutzten/zur Verfügung stehenden Ressourcen. Gerade weil bei Webanwendungen die gemeinsam genutzten Ressourcen wie Speicher, Netzwerkbandbreite und Rechenzeit einen Engpass darstellen, müssen Webanwendungen (unter bestimmten Umständen) effizient sein.
- Portabilität – die Möglichkeit, eine Applikation von einer Umgebung in eine andere zu transferieren. Die Umgebung kann dabei organisatorisch verändere Hardware/Software sein. Die Anwendung ist dann portabel, wenn sie sich verschiedenen, spezifizierten Umgebungen durch ausschließlich für diesen Zweck vorgesehene Aktionen anpasst. Die Anwendung kann in jedem spezifizierten Kontext installiert werden. Portable Software kann mit anderen Anwendungen in einer gemeinsamen Umgebung koexistieren und gemeinsame Ressourcen nutzen. Teile der Anwendung können auch statt einem anderen Produkt für den gleichen Zweck genutzt werden (dies setzt eine modulare Struktur, klare Schnittstellen und die Nutzung von offenen Standards voraus, siehe auch den Punkt Schnittstellenkompatibilität). Für eCommerce Applikationen ist Portabilität wichtig, weil es möglich sein muss, verschiedene Plattformen, Hardware und Betriebssysteme zu unterstützen.
- Schnittstellenkompatibilität – die verwendeten und bereitgestellten Schnittstellen der Anwendung müssen klar und stabil sein. Bei der Nutzung/Implementierung von standardisierten Schnittstellen müssen die Spezifikationen erfüllt und die beschriebenen Standards genutzt werden. Bei Webanwendungen, die einen hohen Abhängigkeitsgrad von anderen Applikationen haben (z.B. Webservices, CORBA-Komponenten), muss bei Entwurf und Testen ein besonderes Augenmerk auf die Schnittstellen gelegt werden.

Abhängig von einer spezifischen Webapplikation kann man die relevanten Charakteristiken festlegen. Ein sogenanntes Qualitätsprofil kann dabei eine Hilfe sein. In dem Profil sind die drei für Webapplikationen wichtigen, generellen Bereiche definiert: Geschäft (Business), Applikation (Application) und Kunde (Customer). Anhand des Messobjekts, welches zu einem dieser drei generellen Bereiche gehört, kann man dann den Qualitätsfokus konkretisieren.

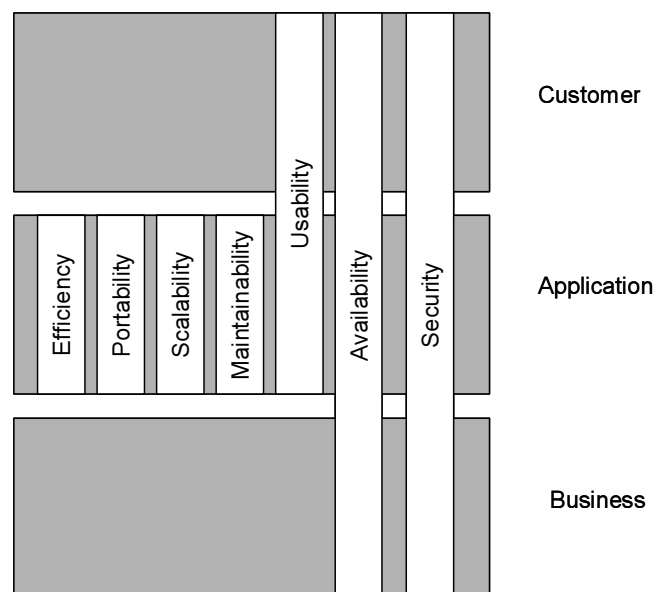


Abbildung 12 Qualitätsprofil für Webapplikationen

3.2.5 Festlegen des Blickwinkels

Der Blickwinkel legt die Perspektive fest, aus der ein Qualitätsaspekt des Messobjekts gesehen wird. Hier sind beispielsweise denkbar:

Kunde – z.B. Sicht des Endkunden (z.B. Betreibers eines Webservices) auf das Produkt mit Qualitätsaspekt Benutzbarkeit oder *total cost of ownership*

Entwickler – z.B. Sicht auf den Implementierungsprozess mit Qualitätsaspekt Aufwand

Testabteilung – z.B. Sicht auf das Anforderungsdokument mit Qualitätsaspekt
Vollständigkeit für das Ableiten von Testfällen

Projektleiter – z.B. Sicht auf den Entwicklungsprozess mit Qualitätsaspekt Kosten

(Unter-)Vertragsnehmer – z.B. Sicht auf das API mit Qualitätsaspekt Dokumentation

3.2.6 Festlegen des Kontexts

Der Kontext legt fest, für welche Umgebung die gemachten Angaben gelten sollen und wo die Messungen durchgeführt werden. Dies ist wichtig, da Messdaten in der Regeln nur in dem Kontext Gültigkeit haben, in dem sie erfasst sind und eventuelle Annahmen bei der Erfassung (z.B. es ist wird von einer regulären Wochenarbeitszeit von 40 Stunden ausgegangen) nur in einem bestimmten Kontext Gültigkeit haben müssen. Meist macht es Sinn, den Kontext so genau wie möglich festzulegen (Firma, Abteilung, Projekt, Zeitraum).

3.2.7 Ein Beispiel für ein Messziel

Innerhalb einer Firma soll zum besseren Verständnis des Testens von Webapplikationen ein Messziel aufgestellt werden.

Objekt

Zentrales Objekt der Untersuchung ist der Regressionstest. Regressionstests werden im *super-Web* Projekt nach dem JUnit-Framework mit dem gleichnamigen frei erhältlichen Tool JUnit-TestRunner durchgeführt. Auch ein Teil der grafischen Oberfläche wird mittels HttpUnit einem GUI-Regressionstest unterzogen. Um die Qualität der Regressionstests im Integrationstest besser beurteilen zu können, werden entdeckte Fehler von den folgenden Testprozessen (im Sinne des Entdeckungszeitpunkts) betrachtet (das Produkt in Produktion ist mit aufgeführt, damit die Gesamtfehlerzahl bestimmt werden kann).

Zweck

Das Verständnis über den Test soll erhöht werden. Von daher ist der Zweck „Charakterisierung“.

Qualitätsfokus

Die Qualität des Regressionstest wird über drei Elemente definiert:

- Anzahl und Art der entdeckten Fehler (daraus lässt sich der prozentuale Anteil der mit Regressionstests gefundenen Fehler von der Gesamtfehlerzahl ermitteln).
- Testabdeckung (eine ausreichende Anzahl Testfälle existiert für jede funktionale Anforderung)

- Verhältnismäßigkeit der Tests (Aufwand des Testens/ Anzahl entdeckter kritischer Fehlverhalten), Stichwort: „design for testability“

Blickwinkel

Das Messziel ist aus Sicht des Projektleiters zu sehen, der gleichzeitig die Rolle des Prozessverantwortlichen innehat.

Kontext

Firma: WebWarrior, Abteilung: Web-Develop, Projekt: *super-Web-Suite*, Zeitraum: 12.03.2002 bis 30.11.2002

Damit ergibt sich als Ergebnis der Zielfindung:

Messziel

Charakterisiere die Qualität des Regressionstest für das Produkt super-Web aus Sicht des Projektleiters im Kontext von super-Web-Suite.

3.3 Definition von Fragen



Basierend auf Qualitäts- und Einflussfaktoren, die in *Abstraction Sheets* dokumentiert sind, werden Fragen definiert, die diese Faktoren genauer erfassen. Die Antwort der Fragen unterstützt damit das Messziel. Zusätzlich sollten Hypothesen bezüglich möglicher Antworten dokumentiert werden (z.B. Abschätzungen von wahrscheinlichen Antworten/Ergebnissen).

Für jeden Qualitäts- und Einflussfaktor, der im *Abstraction Sheet* definiert ist, wird in der Regel eine Frage abgeleitet. Fragen die sich auf Einflussfaktoren beziehen, sollen explizit die hypothetische Beziehung zwischen Einfluss- und Qualitäts-Faktor, als Motivation für die Frage, ausdrücken.

Um eine optimale Interpretation der gesammelten Daten für die Beantwortung der Fragen, und letztlich zur Bewertung des Ziels zu gewährleisten, müssen die Fragen sich an der Sichtweise der Beteiligten orientieren. Die Fragen können weiterhin verfeinert und in eine Fragen-Hierarchie strukturiert werden. Falls nötig, sollte jede Frage bezüglich spezifischer Information wie Terminologie, besondere Umstände, etc. kommentiert werden. Zusätzlich wird für jede Frage eine Antwort als Hypothese formuliert.



Nutzen Sie Hypothesen, um Selbstbetrug zu vermeiden! Durch die Dokumentation der tatsächlichen Einschätzung der Situation wird ein späteres „das war uns ja sowieso klar“ beim Vorliegen der Messergebnisse vermieden. Erst dadurch kann man der Ursache für eine eventuelle Fehleinschätzung auf den Grund gehen.

Vorgehen

Um Fragen zu definieren, sollte man sich die Frage stellen: „Was möchte ich über das Objekt/die Eigenschaften im Qualitätsfokus des Ziels lernen?“

Bei der Beantwortung dieser Frage, und ganz besonders wenn es in einer Gruppe⁸ geschieht, sollten die folgenden Punkte bedacht werden:

- Übereinstimmung mit dem Ziel.
Beim Thema bleiben! Gerade den Zweck und den Blickwinkel sollen die Fragen widerspiegeln.
- Quantifizierbar.
Die Fragen müssen so formuliert werden, dass sie über Messwerte beantwortet werden können. Also nicht „Welche Testfälle wären in diesem Kontext gut?“, sondern „Welche Testfälle haben die meisten Fehler gefunden?“. Kompliziertere Fragen sollten in mehrere einfache Fragen unterteilt werden.
- Bedeutung der Antwort.
Man sollte sich die Fragen stellen: „Was machen wir mit der Antwort auf diese Frage?“, „Was sind mögliche Maßnahmen, um dieses Ziel zu erreichen?“. Wenn die definierte Frage zu keinen Verbesserungen/Handlungen führt, sollte die Notwendigkeit der Frage noch mal überdacht werden.
- Verständliche Diagramme.
Es muss klar sein, was auf den Achsen des Diagramms, welche Werte/Beziehungen, dargestellt wird, welche Analyse der Darstellung vorausgeht und welche Daten ausgeschlossen wurden.
- Benutzung von bekannten Diagrammtypen
Man sollte bei der Auswahl der Diagrammart nicht zu kreativ sein, sondern, wann immer möglich, auf bekannte (zum Fragentyp passende; siehe nächsten Abschnitt) Diagramme zurückgreifen.
- Kosten/Nutzen
Wiegt der Nutzen, den man sich aus der Beantwortung der Frage verspricht, die Kosten der Datensammlung auf?

⁸ Wenn Qualitäts- und Einflussfaktoren im Rahmen eines Workshops gesammelt werden, werden oft auch schon entsprechende Fragen formuliert.

- Klarheit der Begriffe
Es sollte sichergestellt sein, dass Begriffe wie *Qualität*, *Effektivität*, *Effizienz* oder *Zuverlässigkeit* präzise definiert sind.
- Vernünftige Anzahl von Fragen
Erfahrungen aus erfolgreich durchgeführten Messprogrammen zeigen, dass zwischen drei bis sieben Fragen pro Ziel definiert werden sollten.
- Gutes Verhältnis zwischen Qualitäts- und Einflussfaktoren
Es sollte durch die Frage „Was beeinflusst diesen Qualitätsaspekt“ überprüft werden, ob es wichtige Einflüsse gibt. Jedoch sollten nicht mehr als etwa dreimal so viele Einflussfaktoren wie Qualitätsfaktoren definiert sein, da man letztendlich an letzteren interessiert ist

Typen von quantifizierbaren Fragen

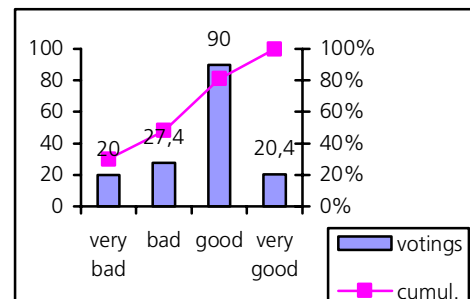
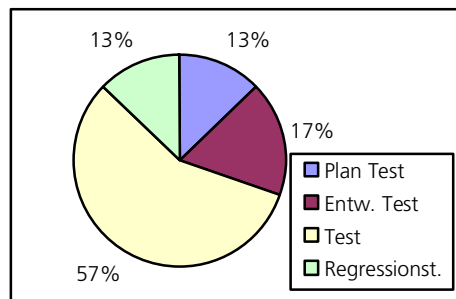
Im Folgenden werden drei Typen von quantifizierbaren Fragen vorgestellt. Für jeden Typ von Frage wird zuerst der Aufbau der Frage, dann ein Beispiel und zum Schluss die empfohlenen Darstellungen (Indikatoren) abgebildet. Da die Typen aufeinander aufbauen, sind die neuen Elemente zusätzlich markiert.

Typ 1:

Aufbau: Was ist **<Qualitätseigenschaften>** von **<Objekt>**?

Beispiel: Was ist **der verbrauchte Aufwand** von **der Testphase**?

Typische Diagramme: Tortendiagramm, Histogramm, BoxPlot

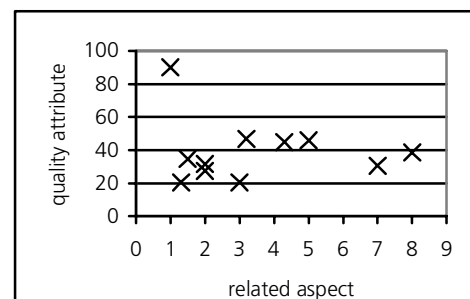
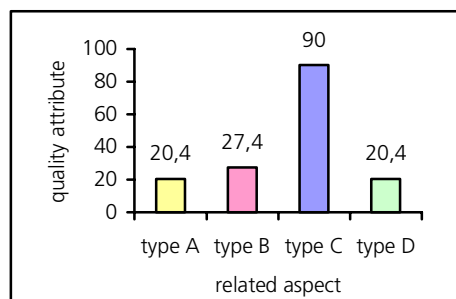


Typ 2:

Aufbau: Was ist **<Qualitätsattribut>** von **<Objekt>** **<Beziehungsoperation>** **<Beziehungsaspekt>**?

Beispiel: Was ist **der verbrauchte Aufwand** von **der Testphase** **verteilt** auf **ihre Unteraktivitäten**?

Typische Diagramme: Balkendiagramm, ScatterPlot, Liniendiagramm

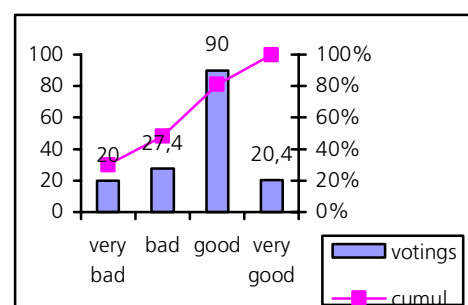
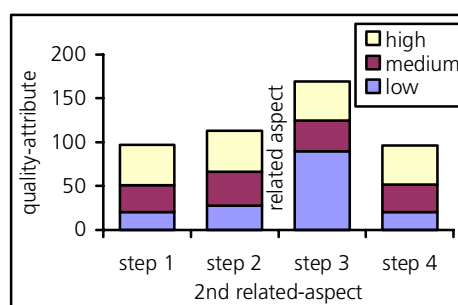


Typ 3:

Aufbau: Was ist <Qualitätsattribut> von <Objekt> <Beziehungsoperation> <Beziehungsaspekt> <2^{te} Beziehungsoperation> <2^{ter} Beziehungsaspekt>?

Beispiel: Was ist der verbrauchte Aufwand von der Testphase verteilt auf ihre Unteraktivitäten **verteilt** über **die Zeit**?

Typische Diagramme: Stapel-Balkendiagramm, Mehr-Liniendiagramm, Bubble Chart, Symplex Chart

**Identifiziere Indikatoren**

Ein Indikator ist eine Visualisierung einer Metrik oder eines Modells, in dem Rohdaten aggregiert werden. Indikatoren werden verwendet, um Messdaten so darzustellen, dass nützliche Information abgeleitet werden kann. Bei problemgetriebenen Messprozessen müssen die Indikatoren nützliche Informationen bezüglich des Problems im Projekt darstellen. Beim Entwurf von Indikatoren muss sichergestellt werden, dass die dargestellte Information eindeutig und korrekt ist. Grafiken können Trends, Abweichungen und Beziehungen zwischen Daten deutlicher darstellen als Tabellen.



Wegen des hohen Nutzens eines Indikators zur Visualisierung der Ergebnisse für die Beantwortung einer Frage, spricht man in der Literatur auch manchmal von „GQIM“.

3.4 Definition von Metriken

Wenn die Ziele in eine Liste von Fragen konkretisiert wurden, müssen die Metriken abgeleitet werden, um die quantitative Information zur Beantwortung in zufriedenstellender Weise zu liefern. Deshalb stellen die Metriken eine Konkretisierung der Fragen auf einer quantitativen Ebene dar, die Produkt-, Ressourcen-, Projekt- oder Prozessmaße beschreiben. Sobald die zu den Metriken gehörenden Messwerte gesammelt sind, steht ausreichend Information zur Verfügung, um die Fragen vollständig zu beantworten.

Im Weiteren müssen auch die Qualitätsfaktoren, welche die Resultate der Metriken beeinflussen, berücksichtigt werden: Faktoren die direkten Einfluss auf die Messwerte haben, haben auch direkten Einfluss auf die Antwort der zugehörigen Frage. Wurden diese Einflussfaktoren bei der Definition des Messprojektes noch nicht berücksichtigt, können aufgrund der Interpretation der gesammelten Daten falsche Schlussfolgerungen gezogen werden. Sie sollten daher in die Liste der Einflussfaktoren aufgenommen werden.

Die definierten Ziele, Fragen und Metriken müssen konsistent und vollständig in Bezug auf das Prozess- und Produkt-Modell des Projektes sein. Um das sicherzustellen, müssen Überprüfungen der Konsistenz und Vollständigkeit während der gesamten Definitionsphase durchgeführt werden. Falls Definitionen fehlen, unvollständig oder inkonsistent sind, müssen die Definitionen an das Prozess-/Produkt-Modell oder das Prozess-/Produkt-Modell und die Ziel/Fragen/Metriken Definitionen angepasst werden. Zusätzlich sollten auch bestehende Datenquellen berücksichtigt werden.



Meist ist die Ableitung von Metriken aus Qualitätsfaktoren und Fragen trivial, dies sollte Sie nicht beunruhigen. Es gibt aber auch kniffligere Fälle, wo dieses schrittweise Vorgehen durchaus Sinn macht und Sie davor bewahrt das Falsche zu messen.

Skalentypen einer Metrik

Die Skala einer Metrik bestimmt, welche Art von mathematischen Operationen man vornehmen darf. Beispielsweise ist die Berechnung eines Mittelwertes bei Ordinalwerten nicht zulässig.

Skalen	Operationen	Beispiel
Nominalskala	Hier werden Nummern oder Symbole zugeordnet um die Mitgliedschaft in einer Klasse zu zeigen. Die Skala besteht aus verschiedenen Klassen und es besteht keine Ordnung auf den Klassen. Erlaubte Operationen: $=$, \neq	Fehler-Klassifikation = {Zuordnung, Überprüfung, Algorithmus, Funktion, Schnittstelle, Zeitverhalten}
Ordinalskala	Dies wird für einfache Ordnungen benutzt. Die Größe der Intervalle zwischen den Klassen kann nicht bestimmt werden. Erlaubte Operationen: $=$, \neq , $<$, $>$	Erfahrung = {hoch, mittel, niedrig} Komplexität des Codes = {trivial, einfach, mäßig, komplex}
Intervallskala	Intervalle zwischen zwei aufeinanderfolgenden Werten, entsprechen gleichen Einheiten der gemessenen Eigenschaft. Damit können neben einer Ordnung auch Distanzen abgebildet werden, so dass die Größe des Sprung von einer zur nächsten Klasse besser verstanden werden kann. Erlaubte Operationen: $=$, \neq , $<$, $>$, $+$, $-$	Datum, Zeit, Temperatur (Celsius/Fahrenheit)

Skalen	Operationen	Beispiel
Verhältnis- oder Ratio- nalskala	Intervallskala mit einem absoluten Nullpunkt, der die völlige Abwesenheit der gemessenen Eigenschaft repräsentiert. Die Skala muss mit Null beginnen und sich in gleichmäßigen Einheiten erhöhen. Alle arithmetischen Operationen können angewendet werden: =, ≠, <, >, +, -, *, /	Zeitintervalle, Länge (LoC), Zyklomatische Komplexität, Aufwand, Temperatur (Kelvin)

Tabelle 3

Übersicht der gängigsten Skalen

Wenn ein Unternehmen keine Erfahrung mit Messaktivitäten hat, ist es sehr wichtig, mit einer kleinen Anzahl von Metriken zu beginnen, anstatt zu versuchen, alles zu messen. Diese kleine Anzahl von Metriken wird durch eine kleine Anzahl von Zielen erreicht (siehe Abschnitt 3.2.1). Zuerst werden Daten für diese Metriken erhoben, dann wird die Messung analysiert, und anschließend können weitere Metriken für dieses Messziel spezifiziert werden.

Für den praktischen Einsatz lohnt sich die Unterscheidung von Metriken in ‚harte‘ und ‚weiche‘ Metriken. Beide sind in bestimmten Kontexten sinnvoll, unterscheiden sich aber in ihrer Mächtigkeit. ‚Harte‘ Metriken sind objektiv und damit mächtiger, lassen sich aber nicht für alle Fragestellungen finden.

	‚harte‘ Metriken	‚weiche‘ Metriken
Typ der Skala	quantitativ	qualitativ
Art der Wertzuweisung	direkt, indirekt	indirekt
Typ der Attribute	intern	extern, ‚Quality-in-use‘
Messverfahren	objektiv	subjektiv

Tabelle 4

Vergleich von ‚harten‘ und ‚weichen‘ Metriken

Eine Übersicht von Web-spezifischen Metriken findet sich im Anhang.

Komplexe Metriken in Modellen beschreiben

Modelle für Metriken dienen der Beschreibung von indirekten Metriken. Sie modellieren, wie sich die indirekte Metrik aus direkten Metriken zusammensetzt und berechnet. Beispiele können Komplexität, Kopplung, Kohäsion oder auch Kostenmodelle sein.

Ein Modell kann die folgenden Dimensionen haben:

- Modellbeschreibung: Beschreibt das Ziel und die Nutzung des Modells
- Annahmen: Spezifiziert Annahmen, auf denen das Modell basiert
- Terminologie: Definiert die im Modell benutzte Terminologie

- Kontextabhängige Eigenschaften: Beschreibt das intuitive Verständnis der Zusammenhänge, in bezug auf die Qualitätsfaktoren in der empirischen (echten) Welt
- Analyse: beschreibt, wie Daten zu den Messattributen gesammelt und analysiert werden, um das Modell mit Werten zu füttern.
- Einheiten/Attribute: Gibt eine Übersicht der Einheiten/Attribute, die gemessen werden sollen, und definiert die Eigenschaften der Attribute basierend auf intuitivem Verständnis.

3.5 Ein Beispiel für eine Frage⁹

Frage: Wie viele Reviews wurden durchgeführt, verglichen mit der Anzahl der geplanten Reviews?

Objekt	Review process												
Indikator	<p>How many reviews have been performed compared to the number of planned document reviews?</p> <p>A pie chart illustrating the completion status of document reviews. The chart is divided into two segments: a larger maroon segment representing 'Not Performed Reviews' at 56% (183 reviews), and a smaller light blue segment representing 'Performed Reviews' at 44% (144 reviews). A separate box on the right indicates the total 'Planned Reviews' at 100% (327 reviews).</p> <table border="1"><thead><tr><th>Category</th><th>Count</th><th>Percentage</th></tr></thead><tbody><tr><td>Not Performed Reviews</td><td>183</td><td>56%</td></tr><tr><td>Performed Reviews</td><td>144</td><td>44%</td></tr><tr><td>Planned Reviews</td><td>327</td><td>100%</td></tr></tbody></table>	Category	Count	Percentage	Not Performed Reviews	183	56%	Performed Reviews	144	44%	Planned Reviews	327	100%
Category	Count	Percentage											
Not Performed Reviews	183	56%											
Performed Reviews	144	44%											
Planned Reviews	327	100%											
Hypothese	ca. 80% durchgeführte Reviews												
Kommentar	Die Anzahl der durchgeführten Reviews entspricht der Anzahl der Protokolle.												

⁹ Eine Erweiterung dieses Beispiels findet sich als Ausschnitt eines GQM-Plans im Anhang A.

4 Messinstrumentierung



Dieser Arbeitsschritt umfasst die operative Planung des Messprogramms für das ausgewählte Projekt und die beteiligten Personen. Dieser Schritt ist von der Messplanung getrennt, weil in vielen Fällen der GQM-Plan in mehreren Umgebungen (z.B. Projekten) genutzt wird. Aktivitäten innerhalb dieses Arbeitsabschnittes sind die Messplanung (Erstellung des Messplans auf der Grundlage des GQM-Plans), die projektspezifische Instrumentierung der Prozesse, Produkte, Ressourcen oder Projekte für die Messung. Beispiele dafür sind der anzuwendende Entwicklungsprozess, Subsysteme oder beteiligte Mitarbeiter. Zusätzliche Informationen, wie der Projektplan, können nützlich sein. Abbildung 13 zeigt die Nutzung und Erstellung von Produkten während der Instrumentierung.

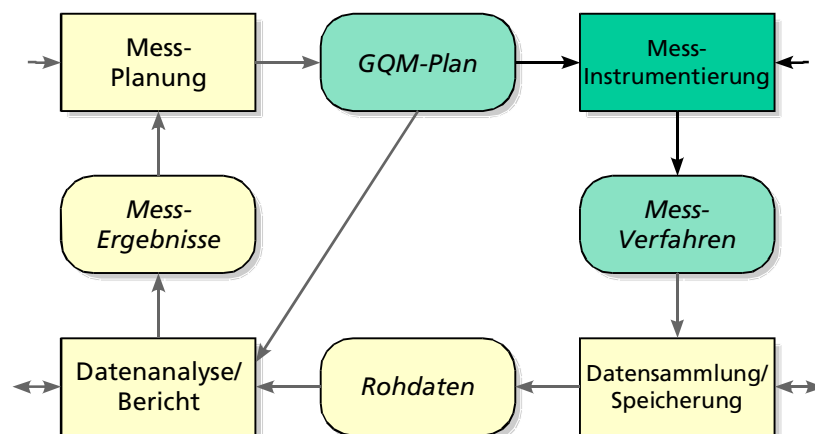


Abbildung 13

Schnittstellen zur Instrumentierung

Der GQM-Plan enthält nur Informationen über das, was gemessen werden muss. Die Messplanung entscheidet, wer, wann und wie die Daten gesammelt werden. Dies schließt die Auswahl der Datenquellen für die Metriken (z.B. Datenbank, Werkzeuge, Fragebögen), die Gruppierung der Metriken unter Berücksichtigung der Rolle und des Zeitpunktes der Erhebung und die Auswahl und Schulung der Personen mit ein. Im Falle der Sammlung von subjektiven Daten (also Daten, die von Personen stammen und nicht von Werkzeugen), muss der Entwurf der Fragebögen geplant werden. Auf Basis des Messplans werden die eigentlichen Messinstrumente angebracht. Ergebnisse der Instru-

mentierung sind Fragebögen, Datenbank-Schemata¹⁰ und die Werkzeuge zur Datensammlung. In Abbildung 13 sind sie alle unter Messverfahren zusammengefasst.

4.1 Entwicklung des Messplans

Aus dem Rahmen, den der GQM-Plan vorgibt, wird der Messplan entwickelt. Im GQM-Plan steht, wie aus den Metriken Fragen über Indikatoren beantwortet werden können. In dem GQM-Plan sind die Messziele beschrieben und auch, wie durch die Anwendung von Maßen diese Ziele erreicht werden können.

Im Messplan ist genau beschrieben, wer aus welcher Quelle zu welchem Zeitpunkt welche Art von Daten erheben muss. Dabei gibt es „kritische“ Werte, d.h. Datenquellen, die zu einem bestimmten Zeitpunkt abgefragt werden müssen. Wird z.B. der Aufwand für ein Projekt nur kumulativ erfasst (d.h. zu einem Zeitpunkt t ist bekannt, wie viel Aufwand insgesamt erbracht worden ist), so kann eine wöchentliche Abfrage nötig sein, um einen Zeitverlauf erstellen zu können.



Im einfachsten Fall kann der Messplan ein weiteres Kapitel des GQM-Plans sein.

siehe Anhang: Vorlage Messplan

4.2 Vorbereitung der Datenerfassung



Der erste Schritt ist oft die Implementierung der Datenbankstruktur für eine Messdatenbank, um in der Lage zu sein, die Messdaten abzulegen (z.B. anlegen von Tabellen und Feldern in einer relationalen Datenbank). In einfachen Fällen kann auch mit einer Tabellenkalkulation gearbeitet werden, z.B. wenn nur wenige Daten erfasst werden.

Existierende oder zugekaufte Datenerfassungs-Werkzeuge werden so angepasst, dass mit ihnen die Daten so erfasst werden können, wie es in der Planung beschrieben wurde. Dabei muss auf die Metriken gemäß dem GQM-Plan, die Rahmenbedingungen des Projektes und die Speicherung gemäß der beschriebenen Datenbankstruktur eingegangen werden.

¹⁰¹⁰ Bei einigen Elementen ist die Trennung zwischen Messplanung und Instrumentierung nicht leicht. Das DB-Schema kann z.B. auch schon in der Messplanung festgelegt werden, besonders wenn generische Strukturen wiederverwendet werden.

Inhouse entwickelte Erfassungs-Tools oder Fragebögen müssen entwickelt werden und mit den zukünftigen Nutzern getestet werden. Vor dem Messstart ist zu überprüfen, dass die Datenerfassungsinstrumentarien allen Beteiligten zur Verfügung stehen (z.B. Zugriff auf Tools möglich, ausreichende Anzahl von Papierfragebögen verfügbar und verteilt).



Auf einer eigenen Intranet-Seite können Sie alle relevanten Informationen für die Beteiligten am Messprogramm bequem zur Verfügung stellen (z.B. Fragebögen).

Die Datenlieferanten sollten für das beschriebene Messverfahren geschult werden, um korrekte, vollständige und konsistente Daten zu erhalten. Jeder Datenlieferant muss wissen, wann und wie Daten von ihm/ihr geliefert werden müssen. Ein gemeinsames Verständnis der Definitionen ist von daher für valide und konsistente Daten unerlässlich. Gegebenenfalls kann in einem kurzen Handbuch zur Datensammlung das Wichtigste für Datenlieferanten zusammengefasst werden: Wer erfasst Daten, wann werden sie erfasst und wie müssen die Daten geliefert werden?



Informieren Sie den Betriebsrat frühzeitig über die geplante Datenerfassung. Eventuell ist das Formulieren einer Betriebsvereinbarung notwendig.

Die Datenschutzrichtlinien sind unbedingt einzuhalten.



Messdaten dürfen nicht dazu verwendet werden, Personen zu kontrollieren oder zu bewerten. Insbesondere bei manuell erfassten Werten müssen Sie das Vertrauen derer gewinnen, die die Daten bereitstellen, um korrekte Daten zu erhalten. Dies gelingt einfacher, wenn das Ziel der Messung allen klar ist.

4.2.1 Werkzeuge für die Datenerfassung

Es gibt nur wenige universelle Werkzeuge für die Datenerfassung. Viele der komplexeren setzen ihr eigenes Vorgehensmodell voraus. Mit Standardsoftware (z.B. Office-Paket) kommt man bereits sehr weit.

Beispiele für Werkzeuge rund um Datenerfassung:

- PSM Insight: <http://www.psm-sc.com/insight.htm>
- VTT Metriflame: <http://www.vtt.fi/ele/research/soh/products/metriflame/>
- TychoMetrics: <http://www.tychometrics.com/>

- Datenbanken mit Eingabe-Formularen (alle großen Hersteller, z.B. Oracle, IBM)
- Online-Fragebogendienste, viele Anbieter, z.B. Globalpark, <http://www.globalpark.com> .

Die Datenerfassung, mit ihren Instrumenten, wird in Abschnitt 5.1 ausführlicher diskutiert.

5 Datenerfassung



Das Ergebnis des Prozessschritts Datenerfassung besteht aus den Rohdaten (s. Abbildung 14), die aufgrund der definierten Messverfahren gesammelt und gespeichert werden.

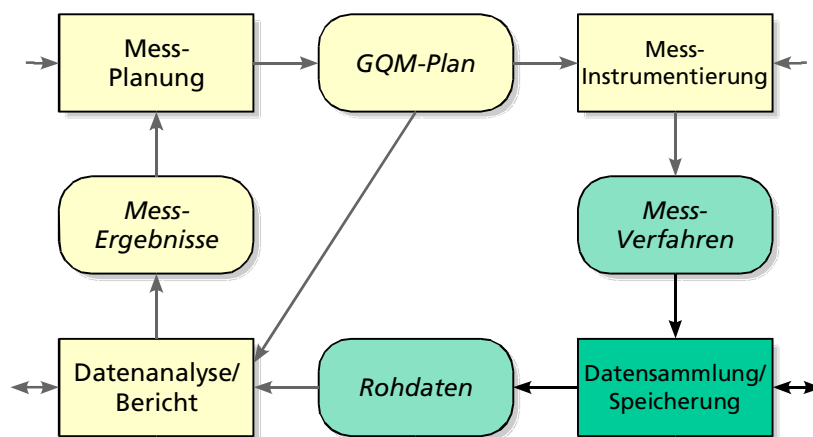


Abbildung 14

Schnittstellen von Datensammlung und Speicherung

Während die Daten gesammelt werden, sollte deren Gültigkeit kontinuierlich überprüft werden. Die Validität der Daten ist eine essentielle Grundlage um eine vernünftige Basis für die Analyse und Interpretation der Daten zu haben. Typspezifische Fehler- oder Plausibilitäts-Überprüfungen können oft direkt während der Datenerfassung vorgenommen werden, falls Werkzeuge genutzt werden. Trotzdem wird empfohlen, einen „Validierer“ für die Daten einzusetzen, der die Datensätze auf Vollständigkeit und Korrektheit der Werte hin anschaut. Es gibt eine große Menge von Produkten die zur Datenspeicherung eingesetzt werden können (z.B., alle relationalen Datenbanken). Die Mehrzahl dieser Produkte bieten JDBC oder ODBC Schnittstellen, so dass Daten über SQL abgefragt werden können. Dennoch ist es wichtig, das richtige Datenmodell einzusetzen, um Speicherung und Zugriff effizient zu gestalten. Metamodelle (Modelle von Modellen) können von Änderungen an den Messverfahren unabhängig machen, aber auch den Zugriff auf die Daten erschweren. Unserer Erfahrung nach ist eine direkte „Fragebogen-Sicht“¹¹ eine gute Lösung zur Speicherung und für den Zugriff.

¹¹ Das Datenbank-Schema bildet einfach die Fragebögen ab.



Unvollständige und invalide Daten können eine Interpretation sinnlos werden lassen. Achten Sie deshalb auf möglichst vollständige und korrekte Messwerte!



Insbesondere bei neu aufgesetzten Messprogrammen sollte die Datenvollständigkeit und die Gültigkeit der Daten bereits früh überprüft werden und ggf. entsprechende Maßnahmen zur Erhöhung der Vollständigkeit und Validität ergriffen werden.

5.1 Datensammlungsinstrumente

Man kann zwischen mehreren grundlegenden Arten der Datenerfassung unterscheiden:

Automatische Datenerhebung über Werkzeuge. Dabei werden über die Werkzeuge Daten über die Produkte (speziell den Code mit sämtlichen Kodemetriken), die Prozesse, das Projekt und die Ressourcen erhoben. Das geschieht in diesen Fällen transparent und ohne Mehraufwand.



Eine automatische Datenerfassung wird üblicherweise nur bei Produktmetriken angewendet, wie z.B. Größenmaße von Dokumenten, Leistungsmessungen von Programmen usw.

Datenblätter und Fragebögen. Dies kann halbautomatisch (z.B. über ein Webfrontend) oder manuell (Papier) geschehen. Die zu beantwortenden Fragen werden von Vertretern der Rollen beantwortet und dann als Rohdaten erfasst. Fragebögen eignen sich gut für statistische Auswertungen und persönliche Informationen (in einem festen Rahmen). Dies erfordert zusätzlichen Aufwand, und wird für Metriken verwendet, die sich nicht automatisch erfassen lassen.

Interviews. Ein Treffen mit einer Person oder Gruppe von Personen, um Daten zu sammeln. Durch den offenen Stil dieses Instruments, kann auch zusätzlicher Input gesammelt werden (wichtig für Verbesserungen, auch des Messprogramms). Dies erfordert zusätzlichen Aufwand, und wird für Metriken verwendet, die sich nicht automatisch erfassen lassen.

Beobachtung. Bei diesem Instrument wird ausgewähltes Verhalten beobachtet, aufgezeichnet und ausgewertet. Das können Meetings, Werkzeugeinsatz, Anwendung einer Methode/Technik sein. Für die ausführenden Rollen bedeutet dieses Instrument keinen zusätzlichen Aufwand, allerdings ist für die Messung selbst eine (oder mehrere) Personen notwendig.

Fragebögen

Fragebögen geben die Möglichkeit nach Erfahrungen aus erster Hand zu fragen: Was wurde getan, was war die Situation, was waren die Gefühle und Wahrnehmungen.

Um Klarheit zu schaffen, sollte immer eine Frage nach der anderen gestellt werden, d.h. zwei oder mehr Fragen in einem Satz/Abschnitt sind zu vermeiden. Die Formulierung der Fragen muss so eindeutig sein, dass jeder Beantwortende wirklich dieselbe Frage beantwortet (unabhängig von Kontext und Lesart). Dazu sind im Zweifelsfall zusätzliche Erklärungen oder Begriffsdefinitionen mit anzugeben. Für die Ausfüllenden muss auch klar sein, wie eine angemessene Antwort aussehen soll. Das Layout des Fragebogens muss ansprechend sein und soll das Lesen, richtige Ausfüllen und Sammeln der Antworten leicht machen.

Datenblätter

Als Alternative zum Fragebogen kann auch ein Datenblatt verwendet werden. Das Datenblatt hat viele eindeutige Metriken, die tabellenartig ausgefüllt werden. Zur Sammlung von Daten sind Datenblätter anwendbar, wenn es nicht zu viel Fragen und Metriken abgefragt werden. Alle Daten die für die abgefragten Metriken sollten auf einer DIN-A 4 Seite dargestellt werden können. Von der ausfüllenden Person werden die Werte direkt in das Datenblatt eingetragen.

[illegible]

Abbildung 15

Beispiel für ein Datenblatt

Interviews

Interviews sind Treffen bei denen Informationen von einer Person oder einer Gruppen von Personen gesammelt werden. Interviews werden normalerweise angewendet, um qualitative Daten zu sammeln.

Ein Interview kann strukturiert geführt werden, wobei die Fragen, die Reihenfolge der Fragen und die Antwortkategorien vorher bekannt sind. Dies ist im Fragebogen (engl. *Questionnaires*) festgelegt. Wenn ein Interview unstrukturiert geführt wird, sind die Fragen vorher nicht festgelegt und nur das Thema ist bestimmt, z.B. Inspektionen oder Fehlerbehebung. Semi-strukturierte Interviews sind eine Zwischenform. Das Thema und die Fragen sind vorher bekannt, aber nicht die genauen Antwortkategorien. Während eines semi-strukturierten Interviews ist es wichtig, die interviewten Personen bezüglich des Themas zu instruieren, welche Informationen benötigt werden.

Beobachtung

Während der Beobachtung nimmt eine Person (der Beobachter) an dem Entwicklungsprozess mit teil. Er/Sie beobachtet was in dem Prozess/der Organisation geschieht. Dafür ist es notwendig (vor oder während des Prozesses) zu definieren was beobachtet werden soll, z.B. die Zusammenarbeit zwischen den Entwicklern, Die Inspektionen von Dokumenten.

Beobachtungen sind bei kleinen Gruppen, einer kurzen Prozesslaufzeit (Wochen statt Jahren) und wenn Zeit kein beschränkender Faktor ist, anwendbar. Die beobachteten Ereignisse (z.B. dass Entwickler nicht berichten, wenn ein Entwurfsdokument fertig ist) sollten so schnell wie möglich systematisch festgelegt werden in ein sogenanntes Logbuch. Die Zuverlässigkeit der beobachteten Ereignisse sollte geprüft werden, z.B. durch Nachfragen innerhalb der Organisation oder durch weitere Beobachtungen (durch andere Beobachter).

Automatisierte Tools

Die Auswahl von automatisierten Tools richtet sich in der individuellen Situation nach folgenden Kriterien aus:

- Was steht in der Organisation bereits zur Verfügung?
- Welche Daten können überhaupt automatisch erhoben werden? (Eine Anzahl von Metriken kann bereits über gängige Entwicklungswerkzeuge , z.B. Konfigurationsmanagement/Versionierungs-Werkzeuge, erhoben werden.)
- Welche Werkzeuge werden bei der Entwicklung eingesetzt (Konfigurationsmanagement, Versionierung)?

- Welche Entwicklungsumgebung/Programmiersprache wird benutzt? (Kodemetriken/Werkzeuge für Kodemetriken sind abhängig von diesen Faktoren.)
- Was sind die verwendeten Datenbank/Report-Werkzeuge?
- Welche Daten werden bereits in einer Datenbank (o.ä.) abgelegt (z.B. Projektfortschrittsberichte)?
- Welche Werkzeuge werden in der Verwaltung eingesetzt (z.B. SAP)?

5.2 Durchführung der Datensammlung

Die Bereitstellung von Daten kann das Ausfüllen eines Fragebogens, die Eingabe von Daten in ein Erfassungswerkzeug oder die Nutzung eines automatischen Werkzeugs sein. In jedem Fall besteht das Resultat aus Rohdaten, die persistent gespeichert sind.

Bei Fragebögen auf Papier müssen die Daten in eine elektronische Form umgewandelt werden, damit sie ausgewertet werden können. Dies kann über Datenerfassungswerkzeuge (automatisch/halbautomatisch, wie z.B. OCR-Systeme oder spezielle Fragebogen-Scanner) oder direkte Eingabe in die Datenbank gemacht werden. Dabei müssen die Eingaben unter Berücksichtigung von Typ, Wert, Vollständigkeit und Plausibilität validiert werden.

5.3 Validierung

Validierung heißt, dass die gefunden Ereignisse und Zusammenhänge real bestehen. Die gesammelten Daten müssen unter Berücksichtigung von Typ, Wert, Vollständigkeit und Plausibilität (z.B. die Beziehung zwischen verschiedenen Maßen oder Ergebnisse die berichten das 80 Stunden am Tag pro Entwickler können nicht richtig sein) validiert werden. Informationen aus dem Kontext (z.B., Urlaubszeiten) sollten im Fall einer manuellen Validierung mitbenutzt werden.

Spezielle Tests können auf Daten des selben Typs durchgeführt werden, z.B., Analyse von Ausreißern (engl. *Outliers*). Ausreißer sind Daten die sehr viel größer oder viel kleiner sind als die Mehrheit der Datenpunkte. Meistens werden sogenannte Boxplots gezeichnet um solche Datenpunkten zu bestimmen.

Ebenen der Validierung

Es können verschiedene Ebenen der Validierung unterschieden werden: Die unterste Ebene ist die der *Attribute* (bezieht sich auf den Qualitätsfokus. Hier können alle Typüberprüfungen (Datentyp, Bereich, Granularität) vorgenommen

werden. Die nächste Ebene betrachtet einzelne *Objekte*¹². Hierbei können Überprüfungen zwischen verschiedenen Attributen des Objekts durchgeführt werden.

Als nächste Ebene wird der *lokale Kontext* eines Objektes betrachtet. Dazu gehören alle Objekte, die mit dem betrachteten Objekt in Beziehung stehen. Auf dieser Ebene sind Tests der Vollständigkeit und Plausibilität möglich. Die Erweiterung dieser Ebene ist die *Multi-Objekt* Ebene. Hierbei können verschiedene Objekte eines Typs (einer Klasse) miteinander verglichen werden. Dies erlaubt statistische Operationen, sowie das Auffinden von Ausreißern.

Die höchste Ebene, die Ebene des globalen Kontexts, betrachtet nicht mehr nur die Objekte, sondern bezieht die ‚echte‘ Welt mit ein. Auf dieser Ebene können bestimmte Auffälligkeiten in den Daten, mit Ereignissen in der echten Welt in Beziehung gebracht und erklärt werden. Eine Übersicht über alle Ebenen ist in Abbildung 16 gegeben.

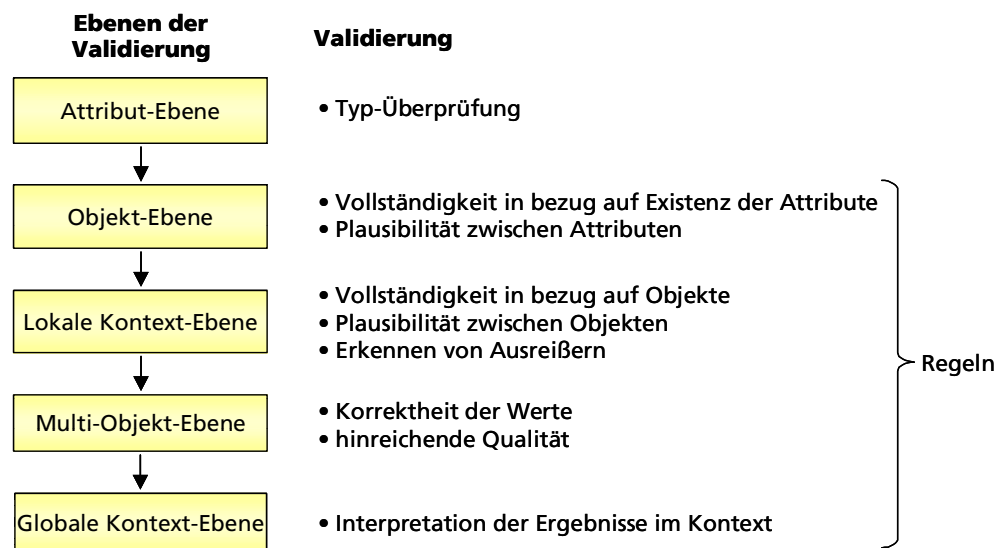


Abbildung 16

Übersicht der fünf Ebenen der Validierung

Phasen der Validierung

Datenvalidierung kann als die Verbindung zwischen den verschiedenen Aktivitäten eines Messprogramms gesehen werden. Diese Aktivitäten, wie in Abbildung 17 dargestellt, sollen unter dem Aspekt der Validierung kurz beleuchtet werden.

¹² Hier sind die Objekte der Messung gemeint, also die Bezugspunkte der Metrik. Diese Objekte sind nicht mit den Objekten des Messziels zu verwechseln.

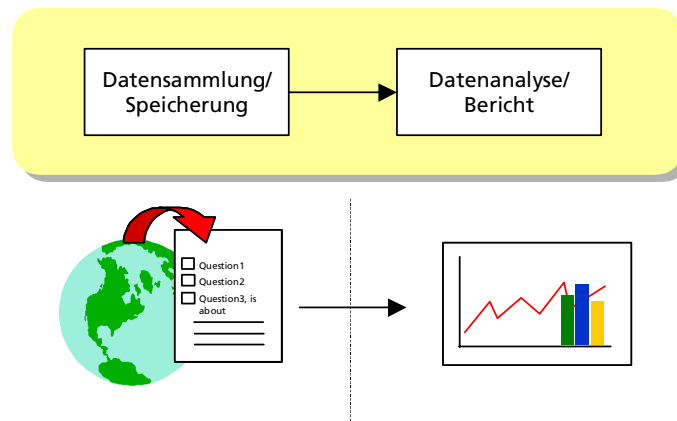


Abbildung 17

Validierung bildet die Verbindung zwischen den Aktivitäten des Messprojektes

Während der *Datensammlung/Speicherung* kann eine Validierung der Daten auf den Ebenen der Attribute, Objekte und im lokalen Kontext durchgeführt werden. Üblicherweise werden diese Prüfungen über das Werkzeug zur Datensammlung und/oder die Datenbank durchgeführt.

Vor der *Datenanalyse/Bericht* wird auf der Multi-Objekt-Ebene validiert. Dies soll die Validität der Messdaten belegen. Die Validierung auf dieser Ebene findet automatisch oder halbautomatisch statt.

Während *Datenanalyse/Bericht* stellen die Methoden zur Analyse und Aggregation der Daten oft einige Voraussetzungen an die Daten (Homogenität, Anzahl der Daten). Diese Voraussetzungen stellen Validitätsregeln dar.

Bei der Interpretation der Daten wird der globale Kontext berücksichtigt. Hier wird überprüft, ob die Ergebnisse mit der ‚echten‘ Welt übereinstimmen. Dies kann in Diskussion und Feedback Sitzungen geschehen.

Ursachen für nicht-valide Daten

Die Ursachen für das Auftreten von invaliden Daten sind vielfältig. Hier sollen einerseits Ursachen angeführt werden, die durch Menschen herrühren als auch Ursachen, die durch Werkzeugen verursacht wurden.

Mensch

Gängige Gründe für durch Menschen verursachte invalide Daten:

- Unzureichendes Wissen der datenbereitstellenden Personen über die Verfahrensweise der Datenerhebung („Ich hatte ja keine Ahnung, dass ich auch das eingeben muss ...“)

- Schlechte Verständlichkeit der Fragen zur Datenerhebung („Was ist der Unterschied zwischen Arbeitsstunden und Personenstunden?“)
- Tippfehler, Flüchtigkeitsfehler (z.B. wurde ,1000‘ statt ,100‘ getippt oder es wurden Felder übersehen und daher nicht ausgefüllt)
- Absichtliche Falscheintragungen („Denen zeigen wir, dass Messen nur extra Aufwand ist und nichts bringt.“)
- Unterlassene Datensammlung (die Leute füllen die Formulare nicht aus, weil sie z.B. unter Zeitdruck stehen)
- Fehlende Motivation (die Leute meinen, dass Datensammlung nicht nützlich ist)
- Fehlende Information (die Leute können die korrekten Daten nicht bereitstellen, weil sie es nicht wissen oder unsicher sind, z.B. weiß ein Mitarbeiter in einem Projekt nicht den Namen der Phase in welcher das Projekt ist)
- Unterschiedliche Interpretation/Wahrnehmung von Begriffen (z.B. Fehlerklassifikation):
 - derselbe Begriff kann für unterschiedliche Dinge benutzt werden (*homonym*)
 - unterschiedliche Begriffe bezeichnen dasselbe Ding (*synonym*)

Zur Vermeidung dieser Ursachen sollten die folgenden Punkte beachtet werden:

- Das Management muss zusätzlichen Aufwand für die Datensammlung und Analyse bereitstellen.
- Alle Personen, die Daten bereitstellen, müssen geschult werden.
- Die Fähigkeit der datenbereitstellenden Personen, korrekte Daten zu liefern, muss regelmäßig überprüft werden.
- Fragen und die vorgegebenen Antworten auf Fragebögen müssen eindeutig formuliert sein.
- Motivation/Information der Leute durch regelmäßige Präsentation der Ergebnisse (Feedbacksitzungen)

Maschine

Gängige Gründe für durch Werkzeuge verursachte invalide Daten:

- Falsche *Scripts*/Regeln zur Überprüfung (z.B. akzeptieren die Regeln korrekte Daten nicht)

- Falsche/unklare Abfragen von Entwicklungswerkzeugen (z.B. im Konfigurationswerkzeug: Abfrage des Status auf „okay“ statt „o.k.“)
- Verlust oder Veränderung von Daten bei der Verarbeitung oder Speicherung
- Schlecht eingerichtete Bereichs- oder Plausibilitätsüberprüfung (z.B. wurde ein automatischer Aufwandprüfer mit wöchentlichen Zahlen eingerichtet, nun hatte man sich auf monatliche Zahlen geeinigt, diese wurden vom Werkzeug nicht akzeptiert)

Zur Vermeidung dieser Ursachen sollen die folgenden Punkte beachtet werden:

- Die Werkzeuge dürfen niemals Werte ändern. Die datenbereitstellende Person muss allen Änderungen zustimmen.
- Mit wenigen *Scripts*/Regeln beginnen. Neue Regeln sollen nur dann hinzugefügt werden, wenn der zu prüfende Fehler regelmäßig/häufig auftritt.



Messdaten sind wertvolle und sensible Daten! So müssen sie auch behandelt werden (auf angemessene Sicherheit und Zugriffsrechte achten; regelmäßige Backups machen).

6 Analyse



Der Arbeitsabschnitt der Datenanalyse und Berichterstellung läuft als kontinuierliche Tätigkeit meist parallel zur Datensammlung und Speicherung ab. Sobald die ersten Daten verfügbar sind, sollte man den Datenlieferanten Rückmeldung geben. Die Analyse wird auf den Rohdaten gemäß den Angaben in des GQM-Plans ausgeführt. Analyse und Interpretation liefern schließlich die Messergebnisse, die dazu benutzt werden die Messziele zu validieren und die Verbesserung voranzutreiben. Abbildung 18 zeigt eine Übersicht der Schnittstellen.

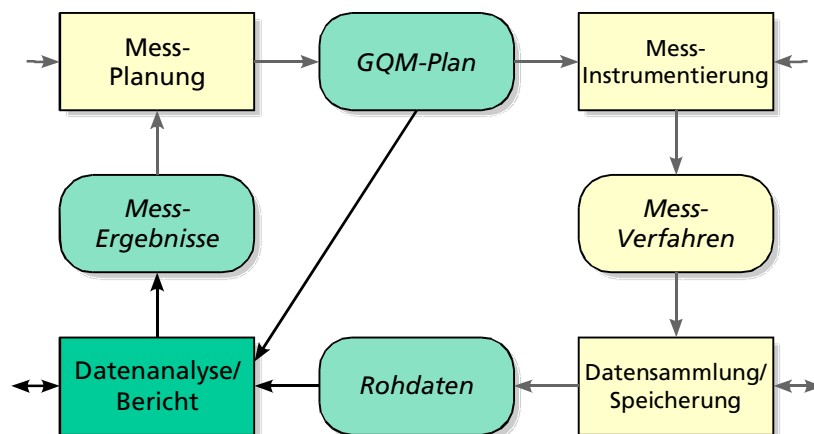


Abbildung 18

Schnittstellen von Analyse und Berichterstellung

Die Analyse erfolgt auf zwei Ebenen: auf der Ebene des konkreten Projektes und auf der Ebene der Prozesse.

Bei der Projektebene liegt der Schwerpunkt auf der Validierung der Daten, der Erklärung von unerwarteten Verläufen, der Sammlung von Verbesserungsvorschlägen und der Umsetzung von projektinternen Maßnahmen. Die Prozessebene geht darüber hinaus auf die Sichtung und Bewertung der Verbesserungsvorschläge im Projekt, das Finden weiterer Verbesserungen und das Umsetzen der prozess- oder organisationsorientierten Verbesserungsvorschläge ein. In diesem Abschnitt wird die Projektebene der Analyse behandelt. Auf die weiterführende Analyse auf der Prozessebene wird im Rahmen von Feedback-Sitzungen in Abschnitt 7.3 eingegangen.

Der erste Schritt der Datenanalyse und Berichterstellung ist die Bearbeitung der Rohdaten gemäß des GQM-Plans. Der GQM-Plan sollte alle Information enthalten, wie die Rohdaten zu einer leicht interpretierbaren Darstellung (meist grafi-

sche Darstellung durch Diagramme) verarbeitet werden können. Übliche Operationen sind Filtern und Aggregieren. Danach können auch statistische Verfahren angewendet werden (z.B. zur Bestimmung von Beziehungen). Wenn die Visualisierung erstellt wurde, wird sie normalerweise zu einer Präsentation (z.B. Folien) oder einem Bericht (z.B. HTML) aufbereitet. Im idealen Fall kann das alles automatisch auf Grundlage des GQM-Plans und der Rohdaten geschehen.

Die Präsentationen und Berichte können als komprimierte Version der Messergebnisse gesehen werden. Sie werden durch Interpretationen und geplante Aktionen, die aus Feedback Sitzungen und Diskussionen stammen, vervollständigt. Ein einfaches Messergebnis ist eine Menge von Fragen mit den zugehörigen Visualisierungen, welche die Fragen beantworten, einer Interpretation der Visualisierung, die der Antwort entsprechen und Schlussfolgerungen (z.B. nächste Schritte für die Prozessverbesserung).

Oft werden die Messergebnisse wieder in den GQM-Plan rückgeführt, um weitere Ziele/Fragen zu motivieren oder diese zu überarbeiten. Diese Änderungen liegen in der Natur von Verbesserungsprogrammen.

Durch die Hinzunahme von Kontextinformation zu den Messergebnissen, kann ein Erfahrungspaket gebildet werden, dass für eine Erfahrungs-Datenbank genutzt werden kann.

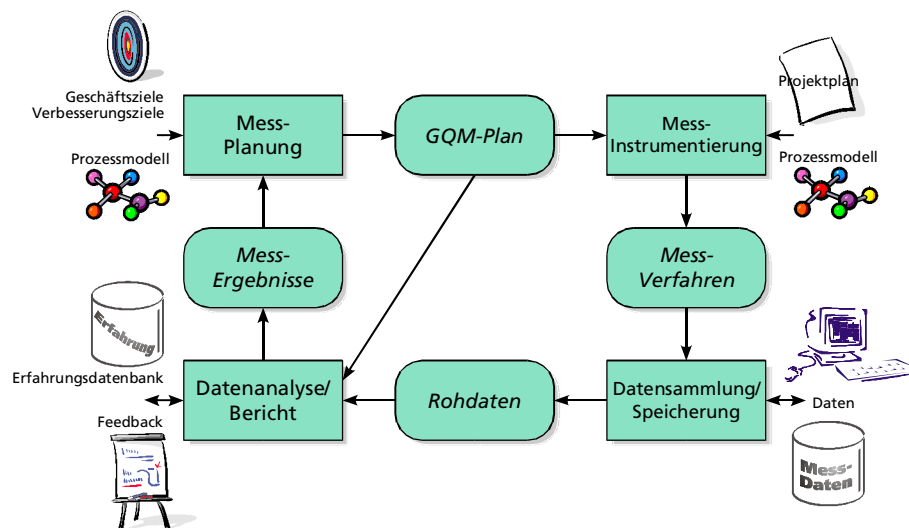


Abbildung 19

Das gesamte Framework für Messen und Werkzeugunterstützung

6.1 Analyse der Daten

In dieser Aktivität wird das Verarbeiten und die Analyse der Datensätze gemäß dem GQM-Plan zusammengefasst. Das bedeutet, dass Modelle oder Metrikdefinitionen angewendet werden um die gewünschten Indikatoren zu erzeugen.

Vielfach können diese automatisch generiert werden, beispielsweise über Reporting-Werkzeuge oder auch Tabellenkalkulationsprogramme.

Alle präsentierten Diagramme sollten die folgenden Elemente enthalten:

- Überschrift mit Bezug zur entsprechenden Frage aus dem GQM-Plan
- Legende/Beschriftung der Achsen
- Angabe des Kontextes (Projekt/Abteilung, Datum der Auswertung)
- ggf. Angabe der angewandten Filter (nach welchem Datum/Woche, nach welchem Projekt etc.)
- ggf. Anzahl der insgesamt in die Auswertung eingeflossenen Daten bzw. absoluten Werte (besonders bei Prozentangaben)

Bei der Verwendung absoluter Zahlen in einer Auswertung (z.B. Aufwand pro Phase, Anzahl der Fehler) ist zu beachten, dass sie nur bei vollständiger Datenerfassung zu interpretieren ist. Eine geringe Dunkelziffer kann hier in Kauf genommen werden, allerdings sollte sie als Faustregel nicht über 5-10% liegen.

Unter der Annahme dass die erfassten Daten repräsentativ sind, sind relative Aussagen (z.B. Verteilung des Gesamtaufwands über die Phasen, Verteilung der entdeckten Fehler über Fehlerklassen) weniger fehleranfällig. Jedoch sollte auch hier die Vollständigkeit nicht unter 70-80% liegen.

Generell gilt: Je höher die Datenqualität in Bezug auf Vollständigkeit und Korrektheit, desto aussagekräftiger die Auswertung.



Neben dem Auswerten gemäß GQM-Plan kann man auch explorativ vorhandene Daten anders als geplant verknüpfen, beispielsweise um neuen Hypothesen über Zusammenhänge nachzugehen. Die explorative Analyse oder das „DataMining“ sollten jedoch auf bestimmte Zeitpunkte (z.B. Ende des Projekts, Ende einer Phase) beschränkt bleiben.

6.2 Feedback-Sitzung

Das Ziel einer Feedback-Sitzung ist das Interpretieren der gesammelten Daten und damit die Überprüfung, ob die Hypothesen (wie sie im GQM-Plan definiert

ist) bestätigt werden können. Feedback-Sitzungen sind wichtig, weil die Datensammlung dadurch zusätzliche Informationen zu den Daten bekommt, was oft in einer unerwarteten aber wichtigen Interpretation der Daten resultiert. Weiterhin bekommen die Datensammler in einer Feedback-Sitzungen Rückmeldung auf die durch sie erhobenen Daten, womit die Einbettung des Messungs- und Bewertungsprozesses in die Organisation größere Chancen bekommt.

Vorbereitung

Die Feedback-Sitzungen bedürfen Vorbereitung, um die Messergebnisse zu interpretieren. Ähnlich wie bei jeder Sitzung müssen ein Raum reserviert und die beteiligten Personen eingeladen werden. Zudem sollte die Präsentationsmethode geklärt, die zu diskutierenden Resultate ausgewählt und das zu präsentierende Material ausgewählt werden. Nicht in jeder Feedback-Sitzung werden alle Fragen/Indikatoren besprochen.

Als Hilfestellung können die Dokumente und Checklisten im Anhang für die Feedback-Sitzungen genutzt werden.

Durchführung

In der sogenannten Feedback-Sitzung werden die aufbereiteten Messergebnisse den Datenlieferanten und den Personen aus dem „Blickwinkel“ des Messziels zur Interpretation vorgelegt. Die Teilnahme der Datenlieferanten ist wichtig, da sie viele der Datenausprägungen erklären können. Die Messergebnisse, werden normalerweise in Form von Diagrammen interpretiert. Dies schließt eine Validierung unter Berücksichtigung der Umgebung (Globale Kontext-Ebene der Validierung, vgl. Abschnitt 5.3), die Identifikation von Problemen sowie die Ursachenanalyse und gegebenenfalls die Nennung von Verbesserungsvorschlägen mit ein. Die Ergebnisse der Interpretation, wie auch die vorgeschlagenen oder geplanten Aktionen, sollten in einem Protokoll dokumentiert werden.

Feedback aufgrund von Messdaten ist wichtig für:

- die Motivation und Einbindung der beteiligten Rollen
- die Korrektheit und Nutzbarkeit der Messergebnisse
- die Fortführung und Verbesserung des Messprogramms

Da die Messergebnisse eine neue Qualität von Transparenz im Projekt liefern, werden Feedback-Sitzungen auch oft genutzt um den Projektstatus zu besprechen. Man kann also die Feedback-Sitzung in reguläre Projekt-Meetings integrieren. Eine Feedback-Sitzung sollte alle 4 bis 6 Wochen stattfinden. In manchen Fällen oder bei kurzen Datenerfassungsperioden (z.B. in der Testphase) sind auch kürzere Intervalle sinnvoll.



Geben Sie den Personen, die Daten erfasst haben, regelmäßig Feedback in Form von Auswertungen der Daten.

Wenn Ihre Datenlieferanten das Gefühl haben, dass die Daten nicht benötigt werden, werden Datenvollständigkeit und -qualität schnell darunter leiden – und damit die Qualität des Gesamtergebnisses!



Rechnen Sie damit, dass aufgrund von Ergebnissen/Feedback auch die momentan laufende Datenerfassung geändert wird, z.B. durch Erweiterung einer Skala oder durch Hinzunahme oder Weglassen von Maßen. Dies ist ein normaler Prozess, besonders bei neu aufgesetzten Messprogrammen.



Trennen Sie projektspezifisches Feedback von Prozess-Feedback. Oftmals ist es schwierig aus dem Projektkontext heraus Feedback zum Prozess zu geben. Hier hilft eine spezielle Feedback-Sitzung z.B. mit Teilnahme der Prozessverantwortlichen. Diese könnte seltener stattfinden, z.B. alle 3 Monate.

Um Schwierigkeiten bei der Vermittlung und Diskussion von Messergebnissen vorzubeugen, sind hier vier wichtige Punkte angegeben:

- 1.) Zu Beginn erklären/klären welche (GQM-)Fragen man beantworten/diskutieren möchte und auch noch mal den Zweck der Frage klar machen (z.B. „Wieviel kostet die Fehlerbehebung pro Fehler (in Stunden)?“; dies ist interessant, da jede Stunde, die man spart, für andere Tätigkeiten verwendet werden kann; 1000 Fehler * z.B. 1 Stunde pro Fehler, würde 1000 Stunden sparen)
- 2.) Es besteht das Risiko der Selbsttäuschung, wenn man die Diagramme (Indikatoren) der Messergebnisse das erste Mal sieht („... das hab' ich schon immer gewusst ...“). Daher ist es sehr wichtig, dass man vorher noch mal die aufgestellten Hypothesen betrachtet oder gegebenenfalls Hypothesen aufstellen lässt. Ein weiterer Vorteil ist, dass so jede(r) einzelne noch mal die Zusammenhänge durchdenken muss. (Konkret fragen welches Ergebnis erwartet wird und wie es qualitativ zu bewerten ist; z.B. „... ich denke 500 Fehler wären o.k., 700 wären fatal ...“; bei komplexeren Fragen/Diagrammen sollte das anhand einer Skizze geschehen, die gleichzeitig auch die Darstellung erklärt.)
- 3.) Nun kann der Indikator mit den Messergebnissen gezeigt werden. Abweichungen zur Hypothese müssen diskutiert werden. Wichtig ist die Ursachensuche (Wie kommt es zu der Abweichung? Was sind (mögliche) Gründe?) und das Festlegen von Folgeaktionen.
- 4.) Die Diskussionen (wichtige Beiträge der Teilnehmer) werden in einem Protokoll dokumentiert.

Nachbereitung

Die Nachbereitung behandelt primär zwei Elemente: Einerseits wird das Protokoll fertiggestellt und verteilt (inklusive des Präsentationsmaterials, Auswertungen und Interpretationen der Daten). Dazu gehört auch, dass offene Fragen dokumentiert werden.

Damit das Messen wirklich zu Verbesserungen auf der Projekt oder Organisationsebene führen kann, müssen die identifizierten Probleme und Verbesserungsvorschläge weitergeleitet, priorisiert und umgesetzt werden (vgl. Abbildung 20).

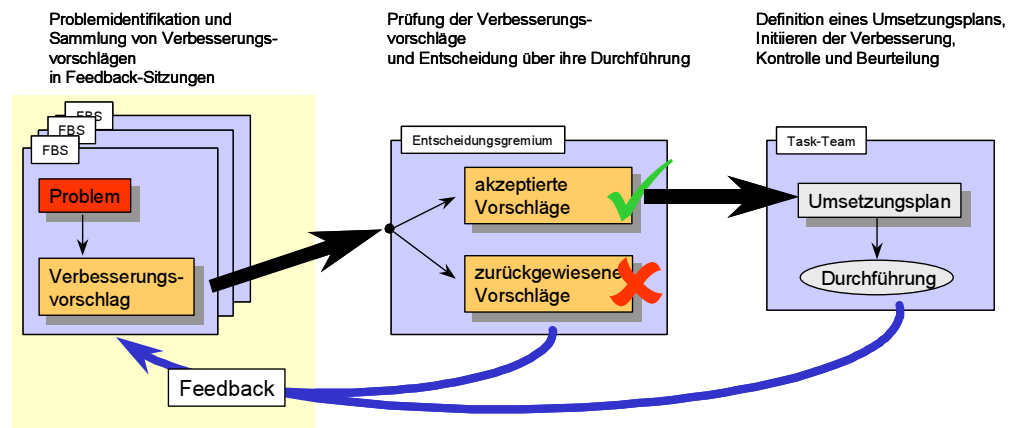


Abbildung 20

Feedback-Sitzungen (FBS) und ihre Auswirkungen



Das primäre Ziel eines GQM-Programms ist Verbesserung. Das Einrichten des Entscheidungsgremiums und des *Taskteams*, einschließlich der Unterstützung durch das Management sind erfolgskritische Aktionen.

6.3 Bündeln/Paketieren von Ergebnissen



Die Integration der Analyseergebnisse (Indikatoren und Diagramme) mit den Interpretationen und der Beschreibung des Kontextes nennt man die Bündelung oder Paketierung der Messergebnisse. Dazu gehört auch die Aktualisierung von existierenden Resultaten (z.B. Ergänzung um neue Interpretationen oder Daten). Dieses Ergebnispaket entspricht einem Erfahrungspaket, dass in eine Erfahrungsdatenbank integriert werden kann.

Die folgenden Abschnitte beschreiben verschiedene Arten eines solchen Erfahrungspakets, das tabellarisch z.B. in einem Ergebnisbericht ein Resultat zusammengefasst werden kann.



Wenn Sie die Erfahrungstabellen bereits zur Dokumentation der Ergebnisse der Feedback-Sitzungen verwenden, entstehen die Erfahrungspakete nach und nach von selbst.

6.3.1 Qualitativ: *Lesson Learned*

Die erste Kategorie von Ergebnissen die gebündelt und für die Zukunft zur Verfügung gestellt werden soll sind die sogenannten *Lessons Learned*. Hierbei geht es um konkrete Erfahrungen die in einem ähnlichen Kontext wieder angewendet werden können. Es werden zwei Kategorien unterschieden: LLs, die das Objekt aus dem GQM-Ziel („Wie kann der Entwicklungsprozess besser werden?“) betreffen und LLs, die das Messprogramm („Wie kann das Messprogramm besser und effektiver werden?“) betreffen.

Die folgende Vorlage kann eine Anregung geben, wie *Lessons Learned* beschrieben werden können. (Sie gilt analog für Entwicklung und Messprogramm; im Falle eines Messprogramms ist die Aktivität im Kontext ‚Messprogramm‘).

Beschreibung	Kurze Beschreibung: Was wurde gelernt? <ul style="list-style-type: none"> • Was war das Problem? • Was hat sich geändert? • Welche Schritte haben zu Änderung geführt? • Wieweit sind die Ergebnisse übertragbar?
Kontext	Ort: ... Prozess: ... Aktivität:
Sachverhalt	Beschreibung der genaue Situation des Problems
Aktionen	Was wurde zur Verbesserung der Situation gemacht? Welche Alternativen wurden gefunden?
Weitere Analyse/ Nachforschung	Wie hat es funktioniert? Welche anderen Effekte hatten die Aktionen noch? Wo kann es noch angewendet werden?
Ausblick	Wie kann es in Zukunft direkt angewendet werden? Was sind Voraussetzungen/Vorgehensweisen?
Referenzen	Was sollte/muss noch gelesen werden? Welche zusätzliche Information steht zur Verfügung?

6.3.2 Quantitativ: Messergebnisse

Auch die konkreten Messergebnisse sollten gebündelt und bereitgestellt werden. Hierbei geht es um die Beschreibung der Frage mit ihren Attributen und der aus den Messergebnissen gebildeten Antwort.

Beschreibung	Was war die Frage? Was wurde wie dargestellt? Was ist noch zum Verständnis notwendig?												
Kontext	Ort: ... Prozess: ... Aktivität:												
Darstellung	Woher kommen die Daten? Aus welchem Zeitraum? Wie ist die Qualität der Daten (Signifikanz, Validität)?												
Diagramm Beschreibung	<p>How many reviews have been performed compared to the number of planned document reviews?</p> <table><thead><tr><th>Category</th><th>Count</th><th>Percentage</th></tr></thead><tbody><tr><td>Performed Reviews</td><td>144</td><td>44%</td></tr><tr><td>Not Performed Reviews</td><td>183</td><td>56%</td></tr><tr><td>Planned Reviews</td><td>327</td><td>100%</td></tr></tbody></table>	Category	Count	Percentage	Performed Reviews	144	44%	Not Performed Reviews	183	56%	Planned Reviews	327	100%
	Category	Count	Percentage										
Performed Reviews	144	44%											
Not Performed Reviews	183	56%											
Planned Reviews	327	100%											
	Was stellt das Diagramm wie dar? <i>Es kann mehrere Darstellungen geben!</i>												
Analyse/ Interpretation	Was sind die Ergebnisse der Analyse und Interpretation?												
Aktionen	Welche Aktionen ergeben sich aus den Ergebnissen?												
Weitere Analyse/ Nachforschung	Welche Seiteneffekte wirkten noch auf die Ergebnisse? Welche Veränderungen gab es während des Messprogramms?												
Ausblick	Was kann in Zukunft besser gemacht werden?												
Referenzen	Welche anderen Messergebnisse sind in diesem Kontext interessant? Was sollte sonst noch gelesen werden?												

7 Anwenden der Ergebnisse

7.1 Definieren von Maßnahmen

Neben dem reinen Informationscharakter von Messergebnissen, geben die objektiven Daten zu Produkten oder Prozessen Sicherheit für die Entscheidung von Maßnahmen für Prozess- oder Produktänderungen.

Das Durchführen eines Improvement-Workshops, bei dem auf Basis der Vorschläge aus den Feedback-Sitzungen konkrete Maßnahmen beschlossen werden, hat sich bewährt. Zur Vorbereitung sollten die Vorschläge für Maßnahmen bereits hinsichtlich Ausgestaltung, Kosten und Umfang vorbereitet werden, damit eine Entscheidung hergeführt werden kann. Die Messdaten können dabei zur Untermauerung des Bedarfs bzw. zur Abschätzung eines Return-on-Investment herangezogen werden.



Der Erfolg eines Messprogramms misst sich am Erfolg der daraus abgeleiteten Maßnahmen, will man das Messen nicht nur als Controlling-Instrument sehen.

7.2 Erfolgskontrolle

Jede Prozessänderung bedeutet zunächst einmal Umgewöhnungs- und Schulungsaufwand und wird erst nach einer Einschwingphase richtig funktionieren. Andererseits kann man nur durch neue Vorgehensweisen neue Stufen der Leistungsfähigkeit oder der Qualität erreichen (vgl. Abbildung 21).

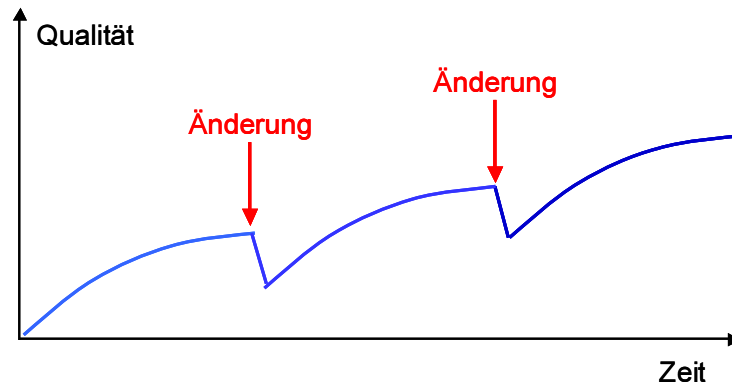


Abbildung 21

Einfluss von Prozessverbesserungen auf ein Qualitätsziel über die Zeit. Die als „Änderung“ markierten Stellen zeigen die Punkte im Prozesseinsatz an, ab denen mit bisher genutzter Technik eine weitere Verbesserung kaum mehr möglich ist. Auf die Änderung folgt zuerst ein Einbruch durch Umgewöhnung und Schulung, bis die eigentliche Prozessverbesserung zum tragen kommt (die nächste Änderung liegt auf einem höheren Qualitätsniveau).

Umso mehr ist die Kontrolle des Erfolgs einer Maßnahme wichtig (i.S. eines Return-on-Investment). Die hierfür nötige Erfolgskontrolle kann wiederum mit einem Messziel beschrieben werden und entsprechend quantitativ belegt werden. Insbesondere der Vergleich von Messergebnissen über die Zeit kann den Erfolg von entsprechenden Prozessverbesserungsmaßnahmen dokumentieren (siehe Abbildung 22).

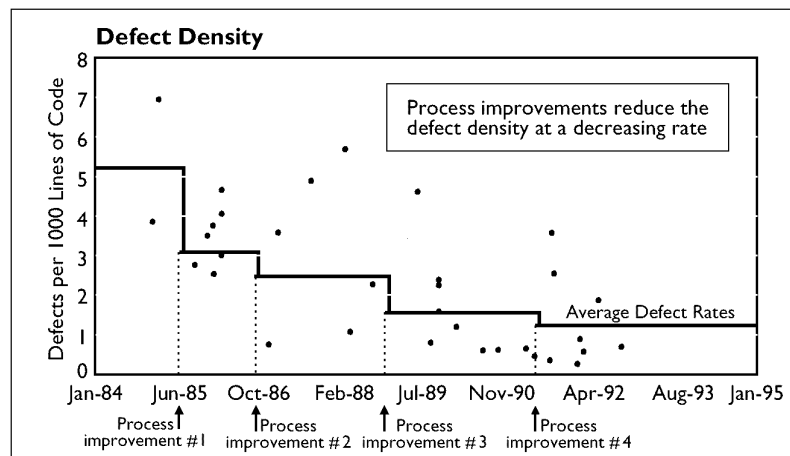


Abbildung 22

Erfolg von Prozessverbesserungsmaßnahmen bei BDM International, Quelle: Communications of the ACM, Vol.41, No.8, S. 70.

7.3 Kontinuierliches Feedback

Neben dem Lernen im Projekt soll durch Messprogramme auch die Organisation als Ganzes profitieren. Dazu ist es nötig, die gemachten Erfahrungen aus dem Projekt zu ‚externalisieren‘. Dies geschieht über die Durchführung von Prozess-Feedback-Sitzungen. Diese nutzen die Resultate der Projekt-FBS (diese wurden in Abschnitt 6.2 diskutiert).

Die nachfolgende Abbildung 23 verdeutlicht das Zusammenspiel zwischen den beiden Typen von Feedback-Sitzungen. Während in den Feedback-Sitzungen auf Projekt/Teilprojekt-Ebene lokale Probleme selbst angegangen werden, bildet die Prozess-Feedback-Sitzung das Glied zwischen Entwicklung und Management. Die dort versammelten Entscheidungsträger sollen neben eigenen Interpretationen insbesondere aus den Projekt-FBS kommende, teilprojektübergreifende Verbesserungsvorschläge und Probleme aufnehmen und entsprechende Maßnahmen festlegen. Der Teilprojekt-Leiter hat die Rolle eines Bindeglieds. Er soll bei Fragen den Standpunkt der Entwickler darlegen können. Dadurch entsteht zwar eine Doppelbelastung dieser Rolle, die Aufnahme der Verbesserungsvorschläge der Entwickler wird aber besser unterstützt. Geht man von einem Verhältnis von Teamleitern zu Teilprojekten von 1:10 aus, so bearbeitet eine Prozess-FBS Ergebnisse und Vorschläge aus zehn Projekt-FBS. Der von den Prozessverantwortlichen beschlossenen Maßnahmenkatalog wird zusammen mit Statusmeldungen über bereits verabschiedete Maßnahmen in einem von den Teamleitern verfassten Dokument den Entwicklern bekannt gegeben.

Es ist wichtig festzuhalten, dass Diagramme mit Daten aus den Projekt-Feedback-Sitzungen vor der Verwendung für die Prozessverantwortlichen anonymisiert werden (z.B. in Form einer Aggregation). Eine Rolle der Prozess-Feedback-Sitzungen ist auch, gruppen- und organisationsübergreifende Verbesserungsmaßnahmen anzustoßen. Bei etablierter *Experience-Factory*-Organisation nimmt diese eine Prüfung auf Eignung in anderen Bereichen der Gruppe/Abteilung/Organisation vor.

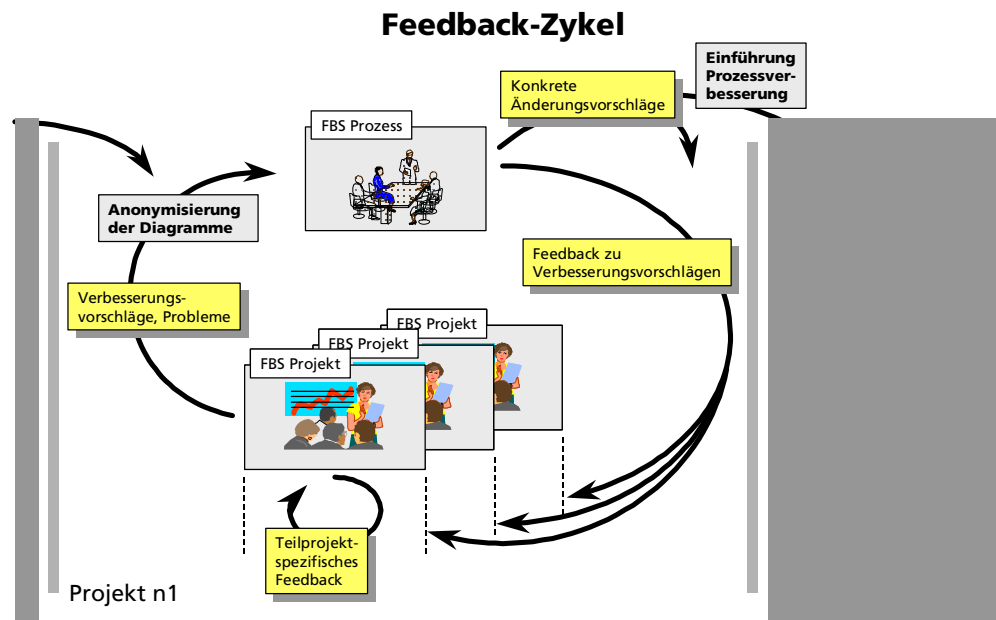


Abbildung 23

Feedback-Zyklus zwischen Projekt- und Prozess-FBS

Referenzen

Literaturhinweise

Robert Park et. al., *Goal-Driven Software Measurement – A Guidebook*, Software Engineering Institute, SEI-HB-96-002,
<http://www.sei.cmu.edu/pub/documents/96.reports/pdf/hb002.96.pdf>

Rini van Solingen, Egon Berghout, *The GQM-Method*, Mc Graw Hill

John Mc Garry et al., *Practical Software Measurement: Objective Information for Decision Makers*, Addison Wesley, 2001.

Practical Software and Systems Measurement V4.0b, Department of Defence, US Army, <http://www.psmc.com/>

Anhang A GQM-Plan-Vorlage

Der GQM-Plan ist das zentrale Dokument, in dem die Ziele, Fragen, Metriken und Beziehungen dokumentiert und begründet sind. Aufgrund des Umfangs des GQM-Plans findet sich hier nur eine Skizze des Inhalts und ein Beispielziel.

Der Inhalt eines GQM-Plans setzt sich aus folgenden Elementen zusammen:

- Einleitung (Kurze Übersicht über GQM; GQM in der Organisation; Ziele und Zwecke; Übersicht über die Messziele; Struktur des Dokuments)
- Darstellung der Ziele und Fragen (jedes Ziel wird in seiner formalisierten Fassung mit allen Fragen angegeben)

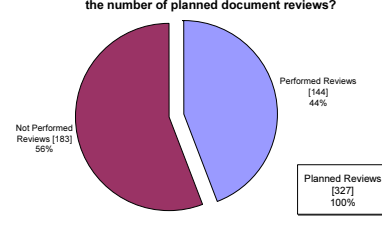
Goal 1	Review Process
	Assess the process conformance of the code and document review process. Investigate the suitability of the invited participants and the completeness of the participation from the point of view of the development organization.
	Object: Review-process (code and documents)
	Purpose: Characterization
	Quality Aspect: Process conformance
	Perspective: Development organization
	Environment: <COMPANY> software development organization.
Quality Aspect related Questions	
g1-Q1	How many reviews have been performed compared to the number of planned reviews?
Object	Review process
Topic	Number of performed reviews
Indicator	<p>How many reviews have been performed compared to the number of planned document reviews?</p>  <p>An analogue indicator will be provided for code reviews.</p>
Hypothesis	around 80% performed reviews expected
Comment	The number of performed reviews equals the number of reviewed protocols.

Abbildung 24

Ausschnitt aus einem GQM-Plan

- Liste der Metriken (mit Einheit und Skala)
- Glossar mit Erklärung der verwendeten Begriffe

Anhang B Vorlage Messplan

Der Messplan beschreibt die Vorgehensweise bei der Messung. Er basiert auf den Metriken und legt für jede Metrik die zur Erhebung benötigten Elemente fest. Aufgrund des Umfangs des Messplans findet sich hier nur eine Skizze des Inhalts und ein Beispiel.

Der Inhalt eines Messplans setzt sich aus folgenden Elementen zusammen:

- Einleitung (Zweck, Notation im Dokument, Struktur des Dokuments)
- Datenquellen (Übersicht der verwendeten Datensammlungswerkzeuge: Werkzeuge, Dokumente, Fragebögen, ...)
- Übersicht über Objekte und Metriken
- Definition der Metriken

Die Metriken werden in einer Tabelle mit der folgenden Struktur beschrieben:

Id	Name	Objekt	Zeitpunkt (k = kritisch)	Datenquelle	Erfassung	Validierer	Bemerkung

Dabei kommt den Feldern folgende Bedeutung zu:

Id	Bezeichnung der Metrik (aus dem GQM-Plan)
Name	Name der Metrik (aus dem GQM-Plan)
Objekt	Objekt, auf das sich die Metrik bezieht (anders ausgedrückt: die Metrik erfasst ein Attribut dieses Objekts)
Zeitpunkt	Zeitpunkt der Erfassung; k die Angabe „kritisch“ bedeutet, dass der Wert zu dem angegebenen Zeitpunkt erfasst werden muss, da er ansonsten nicht mehr verfügbar ist (wird z.B. ein Fehler ohne Angabe eines Datums erfasst, so kann die Metrik „Fehler/Woche“ nur realisiert werden, indem jeweils am Wochenende die Gesamtzahl der Fehler protokolliert wird (die oben genannte Metrik kann dann durch einfache Subtraktion ermittelt werden)
Datenquelle	Angabe, woher die Daten kommen:
Mitarbeiter:	Befragen der Mitarbeiter (mit Tool oder Fragebogen)

Tool:	Extrahieren der Daten aus einem eingesetzten Werkzeug. Anmerkung: speziell für das Messprogramm eingesetzte Tools, werden nicht in dieser Kategorie aufgeführt, sondern unter „Mitarbeiter“
Dokument:	Extrahieren der Daten aus einem Dokument, z.B. X-Auftrag, Subprojekt-Status-Report
Metrik:	Berechnen aus anderen Metriken
Erfassung	abhängig von „Datenquelle“:
Falls Datenquelle = „Mitarbeiter“:	
Werkzeug:	Angabe des Erfassungswerkzeugs (Tool oder Fragebogen);
Erfasser:	Angabe der Rollen, die die Daten liefern müssen;
Berechnung:	Ableitung der Metriken aus erfassten Daten, z.B. die Anzahl der Sätze mit Status „o.k.“, Summe der Aufwands-Einträge mit Phase „Test“.
Falls Datenquelle = „Tool“:	
Werkzeug:	Name des Tools;
Berechnung:	Siehe oben
Falls Datenquelle = „Dokument“:	
Name:	Name des Dokuments;
Berechnung:	Siehe oben
Falls Datenquelle = „Metrik“:	
Metriken:	Angabe der verwendeten Metriken
Berechnung:	Berechnungsvorschrift unter Verwendung der Metriken
Validierer	Rolle/Person, die für die Validierung der Daten zuständig ist;
Bemerkung	weitere Anmerkungen, insbesondere Abhängigkeiten zwischen Metriken, Angabe zum Status (aktiv/inaktiv), mögliche Störvariablen

- Anhang mit zusätzlichen Informationen, z.B. Schema-Definition von Fehler- oder Konfigurations-DB, Liste der Phasen, Liste der Tätigkeiten, Liste von Fehlertypen, ...

Anhang C Software-Metriken für Webapplikationen

Dieser Abschnitt stellt eine allgemeine Übersicht von für Webapplikationen anzuwendende Metriken dar.

Object (related to development phase)	Specification Design / XML	Coding / implementation		Web application		
Quality focus			Link	Browser	Asset/ Content	Web server
Reliability / Availability		Con, PageCom				Con, PageCom, Latency
Usability			Latency, BandW	Latency		Latency, BandW
Security						
Scalability		BandW	ConC, BandW	MTU	MTU	ProC, Con, BandW, MTU,
Maintainability	NoR	Con, PageCom, DbSize, FCW	TTR	FCW, TTR		PageC, ProC, Con, DbSize, NoC, FCW, TTR
Efficiency			Util			ADEE, TDEE, AIW, Util
Portability		StCompl		StCompl		StCompl
Performance		DbSize, BandW	CLB, BandW, MTU	MBC, TTR	MTU	SNE, DLS, WOS, MTU, DbSize, TTR
Concurrency						CACE
Manageable Project	Effort, NoR	Effort, FCW	Effort, FCW	Effort, FCW	Effort, FCW	Effort, FCW
Success of web-site				Hits/sec, ClickT	Hits/sec, ClickT	Hits/sec, ClickT

- ADEE – Application Data Encapsulation Efficiency – the ratio of application data bytes to the sum of one or more overhead bytes (from protocol or otherwise) and application
- AIW – Application Image Weight – the ratio of image-based (e.g., GIFs, JPEGs) application data bytes to the sum of image-based and non-image (e.g., HTML text) application data bytes
- BandW – Bandwidth – the amount of data that can be transmitted over a communications link or network per unit of time, usually measured in bits per second.
- CACE – Component Activity Concurrency Efficiency – the relationship between the total serial execution time (SET) for a set of activities, the total

concurrent execution time (CET) for the same activities, and the non-concurrent execution time (NCET) for the same activities

- CLB – Client Link Bandwidth – average bandwidth in terms of kilobits per second
- ClickT – Click throughs measures the percentage of users that not only view an online add but also click on it to get to the web page behind it
- Con – Connectivity – number of internal links, which are links that are not linked to other web sites only to pages within the application
- ConC – number of concurrent task that can be
- Connectivity density – connectivity divided by page count
- DbSize – database size – the number of words, records or tables in each database. The measure indicates how much data must be handled by the application
- DLS – DNS Lookup Servicetime – DNS lookup request service time (in seconds); target value: 0,005
- Effort – the (estimated or real) elapsed time (number of hours) it took a subject to a) structure the application, b) to interlink the pages in order to build the application's structure, c) to implement the application's interface, d) to test all the link on an application and e) to test all the media on an application.
- EAIW – Encapsulated Application Image Weight
- FCW – functional change workload – counts the number of change requests that affect the functionality of an application. It indicates the amount of work that will be required to implements functional changes, and the stability of the system that is supported or enhanced
- Hits/sec – measures the number of requests for objects served in each second by a website. It is about all embedded objects in a page as separate requests
- MBC – Maximum Brower Connections – Maximum number of web browser TCP connections, target value: 4
- MTU – Maximum Transmission Unit – the largest amount of data that can be sent across a given network using a single packet
- Latency – the time required to complete a request
- LoC – number of lines of code for all programs used by the application
- NoC – number of components – the number of elementary components in the application and the number that are added, modified or deleted.

Changes in the number of estimated and actual components indicate risk due to product size volatility and additional work that may be required

- NoR – number of requirements – counts the number of requirements in the application or its specification. Also counts the number of requirements that are added, modified or deleted
- PageC – Page Count – number of html files used in the web application
- PageCom – Page Complexity – estimation of the complexity of the html pages. Connectivity and Page count can be helpful indicators to determine Page complexity.
- ProC – Program Count – number of cgi scripts, javascript files, java applets used in the application
- SNE – Security Negotiation Efficiency – the ratio of the encrypted application data bytes (E) alone, to the sum of bytes used to negotiate security (N) and encrypted application data bytes (E)
- StCompl – Standards compliance – number of interfaces (or interoperability features) compliant with a specific standard
- TTR – time to restore – time that must be expended to recover from a system failure. This can be indicated by elapsed time, effort and resources needed for system restoration and for problem resolution.
- TDEE – Transport Data Encapsulation Efficiency
- Util – Utilization – the portion of a component resource used or allocated during system operation. Component resource might include CPU, I/O, memory or storage
- WOS – Webserver Object Servicetime – the service time to process an object request (in seconds)

Anhang D Richtlinien für Feedback-Sitzung (FBS)

Checklisten

Vorbereitung

- ☒ Raumreservierung überprüfen
- ☒ Teilnehmerzahl ermitteln
- ☒ Einladungen verschicken
- ☒ Präsentationsmaterial vorbereiten
- ☒ Auswertungen an Teilnehmer verschicken
- ☒ Vorgehensweise und Methodik für Durchführung wählen

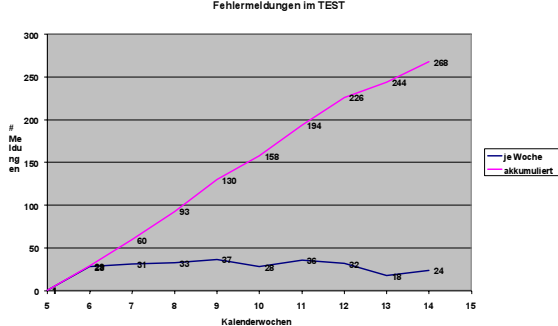
Durchführung

- ☒ Protokollant bestimmen
- ☒ Kopien austeilen
- ☒ Präsentationsfolien zeigen
- ☒ Probleme identifizieren
- ☒ Ursachen erforschen
- ☒ Lösungsvorschläge sammeln
- ☒ Nach Schwierigkeiten bei Datensammlung fragen
- ☒ Termin der nächsten Sitzung festlegen/bekanntgeben
- ☒ Protokoll einsammeln

Nachbereitung

- ☒ Protokoll aufbereiten
- ☒ Protokoll verteilen
- ☒ Lösungsvorschläge weiterleiten

Ablauf

Schritt	Beispiel
1. Vorstellen des Themas <i>Übergeordnetes Thema, zu dem auch mehrere Diagramme gezeigt werden können. (Das Thema kann vorgeschlagen werden oder selbst ausgewählt sein)</i>	Software-Fehler im Integrationstest
2. Warming-Up <i>Einführung in das Thema Abfragen von Hypothesen</i>	Teilnehmer berichten von Ihren Erfahrungen zur Fehler-situation im Integrationstest, Vermutungen zum aktuellen Stand werden geäußert.
3. Visualisierung <i>Darstellung eines Aspekts zum Thema, vorzugsweise in Form eines Diagramms.</i>	 <p>The graph shows two data series over 15 calendar weeks. The upper pink line represents cumulative error reports, starting at 14 and reaching 268 by week 15. The lower blue line represents weekly error reports, with values: 28, 31, 33, 37, 28, 36, 32, 18, 24. The cumulative values are labeled on the pink line: 14, 42, 73, 106, 143, 180, 216, 252, 288.</p>
4. Darstellung verstehen <i>Was ist dargestellt? Was bedeuten die einzelnen Elemente der Darstellung?</i>	Die obere Kurve gibt die aufsummierte Anzahl der Fehlermeldungen für den Integrationstest wieder. Für jede Woche ist ein Messpunkt dargestellt. Die untere Kurve visualisiert die Zahl der neuen Fehlermeldungen pro Woche. Ihre Höhe drückt damit den Steigungsgrad der oberen Kurve aus.
5. Darstellung bewerten/interpretieren <i>Stimmt das dargestellte mit der Realität überein? Was sagt das Diagramm über den Entwicklungsprozeß aus? Treffen frühere Vermutungen zu?</i>	<ul style="list-style-type: none"> - Die Fehlerzahl ist viel zu gering; es werden nicht von allen Leuten alle Fehler gemeldet. - Die Steigung der Kurve ist fast konstant. Zum Testende würde man eine sinkende Steigung erwarten, wenn kaum noch Fehler in der Software sind. Es sieht momentan nicht so aus, als ob dieser Qualitätsstand erreicht wäre. Dies steht im Widerspruch zum unmittelbar bevorstehenden Beta-Test. Der Start des Beta-Test ist an kein Qualitätskriterium gebunden. ¹³ - Zur Beurteilung, ob die Software bereit ist in den Beta-Test gegeben zu werden, sollte die Fehlerpriorität mit in die Darstellung einfließen, z.B. als Gewichtung.

¹³ * = Punkte, die den Entwicklungsprozeß betreffen

<p>6. Wunschvorstellung <i>Wie sähe die Darstellung idealerweise aus?</i> <i>Welche Implikationen hätte dies für den Entwicklungsprozeß?</i></p>	<ul style="list-style-type: none"> - Die Gesamtzahl der Fehler, die im Integrations-test gefunden werden, sollte geringer sein. - Alle Fehler, die man gut bereits im Integrations-test finden kann, sollten auch dort gefunden werden. => Mehr Aufwand für den Integrations-test planen. ✱ - Die Übergabe an den Beta-Test sollte nur erfolgen, wenn ein bestimmtes Qualitätskriterium erreicht ist. Dies könnte z.B. „Nur 10 Prio1-Fehler pro Woche.“ lauten. ✱ - Der Integrationstest sollte schneller von allen beendet werden, die Kurve also steiler und kürzer sein. ✱
<p>7. Analyse <i>Wo im Ablauf/Organisation/Prozess können Änderungen helfen zum Ideal zu kommen?</i> <i>Hypothesen möglicher Zusammenhänge.</i></p>	<p>Es müssen mehr Fehler in früheren Phasen gefunden werden!</p> <ul style="list-style-type: none"> - Eine bessere Werkzeugunterstützung des Host-Tests würde diesen effizienter gestalten, er würde mehr von den Entwicklern akzeptiert und es würden dort mehr Fehler gefunden. ✱ - ... <p>Durch frühe Fehlplanungen wird an Konzeptphase und Integrationstestphase gespart!</p> <ul style="list-style-type: none"> - Da nicht alle Konzepte der einzelnen Teilprojekte bekannt sind, kommt es immer wieder zu späten Anforderungen an andere WPs, die viel Zeit kosten. ✱ - ...
<p>8. Lösungsvorschläge <i>Welche konkreten Maßnahmen müssen ergriffen werden, um zur Wunschvorstellung zu gelangen?</i> <i>Verschiedene Dimensionen:</i> <i>- langfristig/kurzfristig</i> <i>- Projekt-Ebene/Teilprojekt-Ebene</i></p>	<p>Projektebene:</p> <ul style="list-style-type: none"> - Verkürzung der Compile-Test-Recompile-Zyklen im Host-Test durch ... ✱ - Änderung des Entwicklungsprozesses auf Projektebene dahingehend, dass die Konzepte aller Teilprojekte besser untereinander abgestimmt sind. Dies könnte durch Synchronisationspunkte bei Phasenübergängen gewährleistet werden. Idealerweise müssten die Konzepte aller Teilprojekte abgeschlossen sein, bevor mit dem Implementieren begonnen wird. ✱
<p>9. Maßnahmen umsetzen <i>Gibt es Fortschritte bei der Umsetzung?</i> <i>Erste Erfahrungen bei der Umsetzung</i></p>	<p>Der geplante Einsatz des neuen Host-Test-Werkzeugs wird um eine Woche auf dem <Date> vorgezogen. Eine Schulung findet für alle Entwickler am <DATE> statt.¹⁴</p>
<p>10. Erfolgskontrolle <i>Bewertung des Erfolgs einer Maßnahme</i></p>	<p>Planung auf WP-Ebene:</p> <p>4/5 der beteiligten Teilprojekt-Leiter hat die neue Planungsvorschrift bei der Kontrolle des Endtermins geholfen; das Feedback soll dem Prozessgremium zukommen.¹</p>

¹⁴ Fiktives Beispiel.

Protokoll

Im Protokoll sollten inhaltlich die Punkte 5-8 (von der Interpretation bis zum Lösungsvorschlag) dokumentiert werden. Dabei ist es besonders wichtig, daß Verbesserungsvorschläge so detailliert und konkret wie möglich formuliert werden. Auch Vorschläge zur möglichen Umsetzung sind willkommen.

Beispiel:

Verbesserungsvorschlag zur Koordination der Teilprojekt-Konzepte

<i>Symptom:</i>	Durch späte Erstellung von Konzepten <ul style="list-style-type: none">• kommt es zu neuen Anforderungen an Teilprojekte, die sich bereits weit in der Realisierungsphase befinden; dies kann sehr viel Zeit kosten die dann beim Integrationstest fehlt.• sind (spätere) mögliche Schnittstellen-Probleme nur schwer erkennbar, was zu fehlenden Testfällen in Teil <...> führt und damit zu Lücken im Integrationstest.
<i>Lösungsvorschlag:</i>	Vor Phasenübergängen müssen die Schnittstellen innerhalb einer Woche zwischen betroffenen Teilprojekten bekannt gegeben und abgesprochen werde. Idealerweise sollten vor jeglicher Implementierung erst alle Konzepte erstellt werden und die Anforderungen an die Schnittstellen fest stehen.
<i>Erwarteter Effekt:</i>	Durch die Umsetzung des Verbesserungsvorschlags in der Entwicklung wird erwartet, dass <ul style="list-style-type: none">• die Anzahl der späten Anforderungen zwischen den Teilprojekten deutlich verringert wird.• die Effektivität des Integrationstests gesteigert wird.
<i>Risiken:</i>	Es werden keine negativen Auswirkungen auf Kapazitäts- oder Zeitplan erwartet.
<i>Umsetzung:</i>	durch Gruppen- und/oder Projektleiter.

Zusammenfassung

- Ein konkret ausgearbeiteter Verbesserungsvorschlag ist mehr wert als fünf Ideen, die unvollständig formuliert sind.
- Der Gegenstand einer möglichen Verbesserung muss nicht unmittelbar mit dem vorgestellten Diagramm zusammenhängen. Die Dokumentation der Gedankenkette vom beobachteten Phänomen zur vorgeschlagenen Änderung sollte jedoch dokumentiert werden.

- Alle Ideen und Vorschläge müssen detailliert im Protokoll festgehalten werden, damit sie von anderen beachtet werden und umgesetzt werden können.
- Nicht das Diagramm oder die Datensammlung, sondern die Abläufe, die Organisation oder der Entwicklungsprozess sollen in erster Linie verbessert werden.

Anhang E Sammlung von Lessons Learned

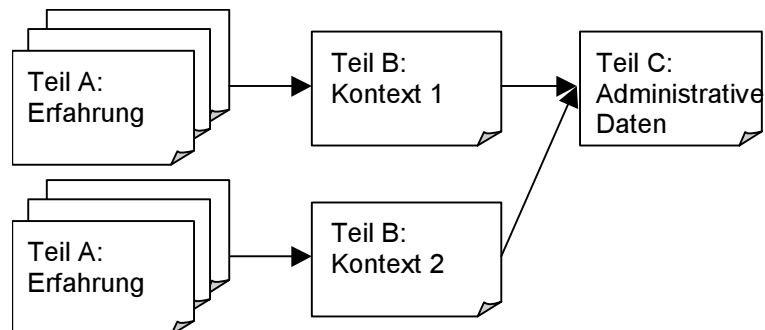
Lessons Learned nennen wir alle positiven wie negativen Erfahrungen, die für den Erfolg anderer Projekte wichtig sein können. Mit diesem Fragebogen werden *Lessons Learned* erfragt, die beim Projekt der Entwicklung gemacht worden sind.

Der wohl wesentlichste Bestandteil der *Lesson Learned* ist „die Erfahrung“. Sie kann eine (positive oder negative) Beobachtung oder ein Problem-Lösungspaar sein.

- eine **Beobachtung** ist eine Beschreibung eines Sachverhalts;
Beispiel 1: „wir haben intensiv Inspektionen durchgeführt, und daher nur wenige Fehler im Integrationstest gefunden“;
Beispiel 2: „die Verteilung eines Workpackages auf zwei Standorte hat zu Missverständnissen bei Schnittstellen geführt, die viel Aufwand für Nachbesserung erforderlich machen“;
„negative“ Beobachtungen sollten nach Möglichkeit als Problem-Lösungspaar beschrieben werden (da ja in der Regel eine - möglicherweise erfolglose - Maßnahme ergriffen wurde).
Falls Sie eine Idee haben, wie zukünftig mit solch einer Situation umgegangen werden kann (d.h. negative vermeiden, positive wiederholen), so können Sie einen **Vorschlag** machen; im Gegensatz zur Maßnahme müssen Sie den Vorschlag noch nicht selber durchgeführt haben.
- ein **Problem-Lösungspaar** ist die Beschreibung von Schwierigkeiten innerhalb eines (Sub-)Projekts, und Maßnahmen, wie sie behoben werden konnten (erfolgreiche Maßnahme) bzw. sollten (erfolglose Maßnahme);
Beispiel: „(Problem:) Die Verteilung der Aufwände auf die Phasen entspricht nicht der verwendeten Regel 40 - 40 - 20%. (Lösung:) es wurde eine Sub-Project-Charakterisierung entwickelt, mit der eine bessere Aufwandsverteilung abhängig vom Typ des Sub-Project gefunden werden kann“.
Zusätzlich kann auch einen **Vorschlag** machen.

Für die Lessons Learned werden die folgenden Dinge erfasst:

- Die Erfahrung (Teil A des Fragebogens)
- Der Kontext (Teil B des Fragebogens) und
- Administrative Daten (Teil C des Fragebogens),



Der Fragebogen stellt eine Unterlage zur Beschreibung einer *Lesson Learned* nach einer entsprechenden Einweisung dar.

Kontext für Fragebogen/alle *Lessons Learned*

Dieser Kontext gilt für alle mit diesem Fragebogen erfassten *Lesson Learned* und wird daher im Fragebogen nicht erfragt. Er ist aber notwendig, um die *Lessons Learned* wiederverwendbar zu speichern:

- Produkt: <...>
- Version: <...>
- Erfassungszeitpunkt: <...>
- Prozess: <...>

Verwendete Abkürzungen

Abkürzungen werden vermieden bis auf einige sehr häufig vorkommende Begriffe:

LL Lesson Learned

PL Projektleiter

Dokumenten Information

Titel:	Handbuch Messen und Bewerten von Webapplikationen mit der Goal/Question/Metric Methode
Datum:	23. Dezember, 2002
Report:	IESE-087.02/D
Status:	Final
Klassifikation:	Öffentlich

Copyright 2002, Fraunhofer IESE.

Alle Rechte vorbehalten. Diese Veröffentlichung darf für kommerzielle Zwecke ohne vorherige schriftliche Erlaubnis des Herausgebers in keiner Weise, auch nicht auszugsweise, insbesondere elektronisch oder mechanisch, als Fotokopie oder als Aufnahme oder sonstwie vervielfältigt, gespeichert oder übertragen werden. Eine schriftliche Genehmigung ist nicht erforderlich für die Vervielfältigung oder Verteilung der Veröffentlichung von bzw. an Personen zu privaten Zwecken.