

4 Qualitätsplanung im Projekt

Um Qualität im Softwareprojekt sicherzustellen, müssen Qualität und deren Ziele definiert und das Erreichen von Qualitätszielen gemessen und verifiziert werden. Die dafür notwendige Planung ergänzt die bereits vorhandene Projektmanagementplanung um den Qualitätsmanagementaspekt.

Die Planung erstreckt sich hierbei auf alle Aspekte des Projekts-Produkte, Prozesse und Personen.

Projektmanager können mit den vorgestellten Ansätzen zuverlässig Software-Prozesse planen und Produktziele vorgeben; Entwickler und Tester können Ziele mitbestimmen und Mindestanforderungen an Qualität einschätzen und definieren; Qualitätsmanager können Planung als wirkungsvollen Ansatz zur Qualitätssicherstellung und -kontrolle verwenden.

4.1 Projektmanagement

Projektmanagement (PM) beschäftigt sich mit der Planung, Kontrolle und Steuerung von Projekten. Diese Aktivitäten werden von zahlreichen Faktoren beeinflusst. So ist die Planung von Projekten nicht nur vom anfallenden Aufwand und den zur Verfügung stehenden Personen abhängig, sondern unter anderem auch von Ihren technischen und fachlichen Fähigkeiten, sowie ihrem sozialen Umgang in einem Team.

4.1.1 Schätzung von Aufwand und Terminen

Die genaue Schätzung des Aufwands für ein Projekt ist ein entscheidender Faktor. Wird der Aufwand zu gering eingestuft, ergibt sich dadurch ein hoher Zeitdruck für die Mitarbeiter, daraus resultiert:

- ein mangelhaftes Produkt und / oder
- überarbeitete Mitarbeiter.

In der Praxis sind oft beide Aspekte zusammen und als direkte Folge voneinander anzutreffen. Des Weiteren ergeben sich dadurch auch wirtschaftliche Probleme für den Auftragnehmer, da dieser höhere Kosten als Einnahmen für ein Projekt hat und dadurch in finanzielle Schwierigkeiten geraten kann.

Wird der Aufwand jedoch zu hoch eingestuft, steigen damit auch die veranschlagten Kosten und ein Projekt wird möglicherweise nicht gestartet. Auch bei exakter Schätzung bedarf es oft einer aufwendigen Argumentation gegenüber dem Kunden, um diesen bei einer hohen Aufwandsschätzung vom Nutzen des Projekts zu überzeugen.

4.1.2 Machbarkeitsstudie

Eine Machbarkeitsstudie beleuchtet zunächst das Projektumfeld. Hier ist nicht nur von Interesse was getan werden soll, sondern auch, unter welchen Umständen es getan werden soll.

- *Projekttyp*: Bevor die Machbarkeit eines Projekts überhaupt festgestellt werden kann, muss zuallererst der Typ des Projekts genau bestimmt werden. Es kann sich um eine vollständige Softwareentwicklung unter Einbeziehung aller Arbeitsschritte (Analyse, Entwurf, Implementierung, Test, Inbetriebnahme) oder nur um ein Wartungsprojekt (z.B. ein Programm Jahr-2000-fähig machen) handeln.
- *Qualität wieder verwendbarer Produkte*: Bei zahlreichen Projekten müssen Teile von bestehenden Systemen weiter verwendet oder zumindest deren Daten übernommen werden. Aufgrund von z.B. veralteten Technologien kann die Integration dieser Systeme einen hohen Aufwand bedeuten.
- *Funktionalität*: Hier werden die konkreten Fähigkeiten der zu entwickelnden Software festgehalten. Dabei ist darauf zu achten, dass diese so detailliert beschrieben werden, dass eine erste sichere Aufwandsschätzung möglich ist. Besonders nichtfunktionalen Anforderungen kommt bei der Beurteilung des Aufwands für die Funktionalität eine hohe Bedeutung zu, da diese den Aufwand um große Faktoren erhöhen könnten.
- *Qualität*: Die geforderte Qualität schlägt sich hauptsächlich in der Erstellung von Dokumentation und der systematischen Projektkontrolle seitens des Projektmanagements und des Qualitätsmanagements nieder. So sind die notwendigen Zeiten für die Dokumentation, die Abhaltung von Reviews und der Aufwand für Änderungen in der Summe ein nicht zu unterschätzender Aufwandsposten.
- *Risiken*: Hier werden nochmals basierend auf den in den vorgegebenen Abschnitten zusammengetragenen Informationen all jene Dinge identifiziert, die das Projekt verzögern könnten: Diese Risiken können im fachlichen, technischen oder auch im organisatorischen Bereich liegen.

4.1.3 Methoden zur Aufwandsschätzung

Einen wesentlichen Teil der Überlegungen zur Umsetzung nimmt die Aufwandsschätzung ein, auf deren Basis jegliche Zeitplanung und Kostenschätzung ja überhaupt erst beginnen kann.

Ansätze zur rechnerischen Schätzung

- *Analogiemethode*: In vorangegangenen Projekten werden Parallelen zum aktuellen Vorhaben gesucht
- *Multiplikatormethode*: Das Projekt wird in einheitliche „Bausteine“ zerlegt; einzelne Teile werden genau geschätzt, der Gesamtaufwand ergibt sich durch Multiplikation.
- *Gewichtsmethode*: Identifizieren von „Aufwandstreibern“ (Funktionsmerkmale), Berechnung des Aufwands mittels Formel (z.B. COCOMO)
- *Prozentsatzmethode*: Aus der detaillierten Schätzung einer (Teil-) Phase (oder den Daten einer bereits abgelaufenen Phase) wird auf den Gesamtaufwand hochgerechnet.

Praktisch eingesetzte Schätzverfahren (z.B. Function – Point – Methode) sind stark formalisiert und beinhalten Elemente aller angeführten Basismethoden. Allgemein gilt, dass gutes Schätzen

viel Erfahrung mit der verwendeten Methode und deren Parametern (Projekt, Betrieb, Organisation, Mitarbeiter usw.) erfordert.

Ein häufiger Fehler ist, den für „Nebentätigkeiten“ wie Projektmanagement, Versionsverwaltung und Qualitätssicherung nötigen Aufwand bei der Schätzung des Gesamtaufwands zu vergessen. Dieser kann entweder explizit berücksichtigt oder den „Produktivtätigkeiten“ als „Lastfaktor“ (Prozentsatz) zugeschlagen werden.

4.1.4 Projektvereinbarung

Ergibt die Machbarkeitsstudie nun ein positives Bild und erfolgt eine Entscheidung für das Projekt, ist es wichtig, die getroffenen Annahmen auch in der Projektvereinbarung vertraglich zu fixieren. Zumindest folgende Punkte sollen vorhanden sein, um unangenehme Überraschungen möglichst zu vermeiden:

- *Pflichtenheft*: Ein Projekt führt einen momentanen Ist-Zustand in einen Soll-Zustand über. Dabei kann es sich um das Lösen eines Problems oder um das Ausschöpfen eines Potentials handeln. Beide Zustände sind in einer Weise zu beschreiben, die die bei dem Projekt zur Lösung anstehenden Aufgaben (d.h. Pflichten) gut erkennen lässt.
- *Leistung*: Hier wird salopp gesprochen festgelegt, wofür bezahlt wird. Im Idealfall handelt es sich dabei um die Erfüllung der Aufgaben des Pflichtenhefts. Es ist jedoch im Allgemeinen sinnvoll, den Umfang möglichst exakt zu beschreiben.
- *Dokumentation*: Zusätzlich zur projektinternen Dokumentation wird in der Regel auch Dokumentation speziell für den Kunden angefertigt (Anwenderhandbuch).
- *Abnahme*: Es genügt nicht nur, Pflichten und Leistungen zu definieren – es muss auch exakt entscheidbar sein, ob bzw. wann diese erfüllt sind. Hier sind möglichst quantitative Aussagen nötig, wie zum Beispiel eine Maximalanzahl an reproduzierbaren Fehlern, die vom Kunden im Probebetrieb in einer bestimmten Zeit nach der Installation gefunden werden dürfen.
- *Pflichten des Kunden*: Das Überlassen von spezifikationskonformen Testdaten oder die Bereitstellung der Infrastruktur für den Systemtest sind mögliche Beispiele für die Pflichten des Kunden. Neben dieser nicht ganz selbstverständlichen Unterstützung durch den Kunden müssen folgende Aspekte zu den Pflichten des Kunden gerechnet werden: rechtzeitiges Melden von veränderten Voraussetzungen, Information über entdeckte Fehler, Bereitstellung von Kontaktpersonen aus Anwendungsdomäne, usw.
- *Gewährleistung bei Problemen*: Die Fälle, in denen Gewährleistung zugesprochen wird, werden beschrieben. Auch die Gewährleistungsfrist und der Zeitraum, in dem die Fehler nach Meldung behoben werden, werden festgelegt. Statt der Fehlerbehebung können auch Minderungszahlungen vereinbart werden. Haftungseinschränkungen werden bestimmt.
- *Pflege und Wartung*: Ähnlich wie bei der Gewährleistung werden auch hier Fristen und Leistungen festgesetzt. Zusätzlich wird festgelegt, wie diese Tätigkeiten verrechnet werden. Will der Kunde diese Arbeiten selbst durchführen, benötigt er Quelltext und

technische Dokumentation; auch die Umstände und Bedingungen, unter denen diese überlassen werden, sind zu regeln.

- *Vertragsbeendigung*: Für den Fall, dass Dinge nicht so laufen wie vorgesehen, müssen auch die Umstände festgelegt werden, unter denen der Vertrag vorzeitig gelöst werden kann. Wichtig sind hier exakte Kriterien und Fristen.
- *Ansprechpartner und Kompetenzen, Eskalationsprozeduren*: Wichtig ist festzuhalten, wer welche Dinge kompetent entscheiden kann. Wer entscheidet z.B. über Änderungen an der Spezifikation, die ja zusätzliche Kosten verursachen um im Projektverlauf berücksichtigt werden sollen.
- *Vergütung*: Hier wird festgelegt, wann und in welcher Form Gegenleistungen für durchgeführte Arbeiten erfolgen. Das kann beim Erreichen von Teilzeiten oder erst bei Projektabschluss sein; die Bezahlung kann pauschal oder nach Aufwand (Arbeitszeit) erfolgen. Auch Sachleistungen sind denkbar.

4.2 Qualitätsdefinition im Projektplan

Das Projektmanagement hat hohe Anforderungen an die Qualitätsplanung (QP). Diese Anforderungen umfassen beispielsweise die Schätzung von Aufwand und Terminen, die Durchführung von Machbarkeitsstudien und die Erstellung von Projektvereinbarungen. Eine Grundvoraussetzung für die Überwachung der Ergebnisse, des Projektfortschrittes und der „Qualität“ der Prozesse und Produkte ist die Möglichkeit, geeignete Kenngrößen zu definieren und zu messen. Die Definition von Kenngrößen ist in vielen Fällen, abhängig von den beteiligten Rollen, widersprüchlich und lösen bisweilen Konflikte aus, die in geeigneter Weise ausgeräumt werden müssen. Im Kapitel 12 stellen wir die Methode EasyWinWin (EWW) zur Verhandlung von Anforderungen vor, die auch für Erarbeitung von Messgrößen herangezogen werden kann.

Die DIN EN ISO 8402, August 1995, Ziffer 3.3 definiert Qualitätsplanung als „*Tätigkeiten, welche die Ziele und Qualitätsanforderungen sowie die Forderungen für die Anwendung der Elemente des Qualitätsmanagementsystems festlegen*“.

Qualitätsplanung umfasst dabei:

- die Planung bezüglich des Produkts (Identifizieren, Klassifizieren und Gewichten der Qualitätsmerkmale; Festlegen der Ziele, der Qualitätsforderungen und der einschränkenden Bedingungen).
- die Planung bezüglich der Führungs- und Ausführungstätigkeiten (Vorbereiten der Anwendung des Qualitätsmanagement-Systems mit Ablauf- und Zeitplänen).
- das Erstellen von Qualitätsmanagement-Plänen und das Vorsehen von Qualitätsverbesserungen.

4.2.1 Qualitätsplanung im Qualitätsmanagementsystem

Das Entwicklungsteam muss festlegen, wie es die Qualitätsforderung an das System erfüllen will. Die Qualitätsplanung muss mit allen anderen Forderungen an ein Qualitätsmanagement-System

übereinstimmen. Sie muss in einer Struktur dokumentiert werden, die der Arbeitsmethode des Entwicklungsteams entspricht. Das Entwicklungsteam muss in angemessener Weise die folgenden Aspekte bedenken, um die festgelegten Forderungen zu erfüllen:

- das Ausarbeiten von Qualitätsmanagement-Plänen;
- das Bereitstellen von Lenkungsmaßnahmen, Prozessen, Einrichtungen, Mitteln und Fertigkeiten, die zur Erfüllung der Qualitätsforderung benötigt werden;
- Verträglichkeit des Designs, des Entwicklungsprozesses, von Installation, Wartung, Prüfverfahren und der zugehörigen Dokumentation;
- Forderungen an die Messtechnik einschließlich derjenigen, die dem bekannten Stand der Technik voraus sind, um Spielraum für die Entwicklung der benötigten Fähigkeiten zu haben;
- geeignete Verifizierung an zweckmäßigen Stellen der Produktrealisierung;
- die Festlegung von Annahmekriterien bezüglich aller Merkmale und Forderungen, eingeschlossen diejenigen, die ein subjektives Element enthalten;
- Führen von Qualitätsaufzeichnungen.

Qualitätsmanagement-Pläne können in Form eines Verweises auf die passenden Verfahrensanweisungen bestehen, da diese fester Bestandteil des Qualitätsmanagement-Systems des Entwicklungsteams sind.

4.2.2 Messen von Qualität

Qualität ist meistens nicht als objektive Größe messbar, es müssen Kriterien aufgestellt werden, an denen das Ausmaß an Qualität gemessen werden kann. Diese Qualitätskriterien sollen physisch messbar sein und die gesetzten und festgelegten Ziele widerspiegeln. Es sollen also die interessanten Merkmale des Produkts quantitativ erfassbar gemacht werden. Es wird zwischen direkten und indirekten Maßen unterschieden. Direkte Maße sind objektiv bzw. direkt messbar, während bei indirekten Maßen kein direktes Maß verfügbar ist oder der Aufwand für die Erfassung der Daten zu aufwendig ist.

Um aber auch indirekte Maße bestimmen zu können, müssen geeignete Indikatoren gefunden werden, welche die zu messenden Merkmale umschließen.

Eine Möglichkeit, Qualität messbar zu machen ist das *Goal-Question-Metrik-Modell* (GQM), das von V. Basili und D. Rombach entwickelt wurde und in Kapitel 5 näher beschrieben wird. Ausgehend von Qualitätszielen und geforderten Eigenschaften werden dabei geeignete Fragen formuliert, deren Beantwortung als Nachweis für die Erfüllung dieser Eigenschaften verwendet werden kann. Unter Verwendung geeigneter Metriken wird eine Messgröße für den Erfüllungsgrad ermittelt. Beispielsweise können die folgenden Metriken verwendet werden:

- Mittels der Kenngröße *MTBF* (mean time between failure, d.h. durchschnittliche Zeitspanne zwischen zwei auftretenden Fehlern) kann das Qualitätsziel "*Zuverlässigkeit*" über die *Fehlerhäufigkeit* gemessen werden [Hindell, 1996].

- Eine der *MTBF* ähnliche Metrik ist die *MTTR* (mean time to recover). Darunter versteht man die Zeit, die ein System benötigt, um von einem Fehler wieder in einen fehlerfreien Modus zu gelangen.
- Eine weitere Metrik, um die Fehlerhäufigkeit zu messen ist die *Failure Rate*. Diese gibt an, wie viele Fehler pro Zeiteinheit bei einem System oder einer Komponente auftreten.
- Eine Möglichkeit, das Qualitätsziel „*Usability*“ einer Software messbar zu machen, ist die *Success Rate*. Diese einfach zu messende Metrik gibt den Prozentsatz an Aufgaben an, die ein User mittels der Software fehlerfrei durchführen kann.

Diese Auswahl an Metriken ist natürlich nicht vollständig. Die Menge an vorhandenen Metriken für die verschiedenen Qualitätsmerkmale ist um vieles umfangreicher.

4.2.3 Qualitätsmerkmale von Prozessen

Stellt das Produkt das Ergebnis des Projektes dar, so ist der Prozess das eigentliche Vorgehen im Projekt, um das Produkt herzustellen. Er beschreibt den Planungs- und Realisierungsablauf.

Ein Prozess sollte:

- effektiv sein, d.h. vorgegebene Aufgaben und Ziele sind erfüllt.
- wirtschaftlich (effizient) sein, d.h. die Aufgaben werden durch ein Minimum an Ressourceneinsatz erfüllt.
- kontrollier- und steuerbar sein, d.h. der Prozesszustand ist erkennbar und korrektive Maßnahmen sind möglich.
- anpassbar sein, d.h. Veränderungen in der Prozessumgebung sind einfach durchführbar.

Die Reife eines Software-Prozesses ist ein Maß dafür, inwieweit ein bestimmter Prozess definiert, geführt, gemessen, gesteuert, kontrolliert und verbessert werden kann und dabei auch effektiv ist. Das Prozess-Reifegradmodell nach CMM (*Capability Maturity Model*) unterscheidet fünf Reifegrade von Prozessen (dieses Prozessmodell wird in Kapitel 13 ausführlich behandelt):

- Initial- bzw. chaotische Prozesse.
- wiederholbare Prozesse.
- definierte Prozesse.
- geführte Prozesse.
- optimierende Prozesse.

4.2.4 Qualitätsmerkmale von Managementvorgängen

Um Qualitätsmanagement im Unternehmen einzuführen und erfolgreich umzusetzen bedarf es bestimmter Rahmenbedingungen, die durch Management, Technologie und Mitarbeiter - sog. Schlüsselgrößen - gestaltet werden.

Für die Qualität von Managementvorgängen spielt die Vorbildfunktion (leadership) eine herausragende Rolle. Mit dem Begriff leadership können folgende Aufgaben in Verbindung gebracht werden:

- Betreuung von Kunden, Mitarbeitern und Unternehmenseigentümern
- Ziele und Wege festlegen, wie die Unternehmung sich langfristig entwickeln soll
- Teambildung durch die Motivation und die Befähigung anderer, Verantwortung zu übernehmen
- Corporate Identity vorleben, d.h. Werte der Organisation nach außen und innen vertreten
- Zulassen von Meinungsvielfalt und Schaffen einer Atmosphäre des Vertrauens, der Offenheit und der Aufrichtigkeit
- Erfolge belohnen und Fehler zulassen
- Distanz wahren

In den Bereich der Schlüsselgröße Management gehören neben den eigentlichen Managementvorgängen auch der Einsatz eines Qualitätsmanagementsystems, die Qualitätskostenanalyse und das organisationsbezogene Änderungsmanagement.

4.3 Produkte

Produkte sind greifbare Ergebnisse im Entwicklungsprozess. Gleichwohl stellt deren Vielfalt, Änderungsfrequenz und Komplexität hohe Anforderungen an die Qualitätsdefinition und -messung. Dieser Abschnitt führt praktische Beispiele für typische Produkte an (etwa Abrechnungseinheit Ticketverkauf, Auswertungen für Management, Autokonsole und zugehörige Web-Seite).

4.4 Prozesse

Prozesse und deren überprüfbare Qualität sind maßgeblicher Bestandteil der Software-Entwicklung.

Dieser Abschnitt geht auf ausgearbeitete Beispiele für Qualitätsplanung anhand einiger typischer Prozesse ein (z.B. Configuration Management, Prozessbeurteilung, Prozessverbesserung).

Einfach ausgedrückt ist ein Prozess eine Folge von Arbeitsschritten, die ausgeführt werden müssen, um eine bestimmte Aufgabe (z.B. Bau eines Autos, Entwicklung eines Softwaresystems) zu erledigen. Genauer betrachtet stellt sich aber heraus, dass Prozesse ein komplexes Rahmenwerk für den Einsatz von Methoden, Werkzeugen und Menschen in Organisationseinheiten (z.B. Firmen, Projekten, Arbeitsgruppen) bilden.

Es wird zwischen Steuerungs- und Unterstützungsprozessen unterschieden. Zu den Steuerungsprozessen gehören Angebotsmanagement, Anforderungs- und Änderungsmanagement, Projektplanung, Projektverfolgung und -steuerung, Risikomanagement, Problemmanagement und Qualitätsmanagement.

Zu den Unterstützungsprozessen gehören Configuration Management, Infrastrukturmanagement, Technologiemanagement, Wiederverwendungsmanagement, Wissensmanagement, Kapazitätsmanagement, Subunternehmermanagement.

Auf einige der Steuerungsprozesse sind wir schon genauer eingegangen. Wir wollen uns nun mit den Unterstützungsprozessen genauer beschäftigen:

4.4.1 Configuration Management

Das Configuration Management (Konfigurationsmanagement) umfasst vor allem die Einrichtung einer Versions- und Release-Verwaltung. Nur durch werkzeunterstützte Versionsverwaltung ist ein Arbeiten an größeren Projekten durch eine Vielzahl von Entwicklern überhaupt noch denkbar. [Berczuk, 2002] [Mette et al, 2003]

Versionsverwaltung bedeutet, dass alle relevanten Dokumente und Programmbestandteile mit ihrem jeweiligen Änderungsstand archiviert und dokumentiert werden. Dies bedeutet, dass jederzeit ein Überblick der aktuellen Versionen aller Dateien vorhanden ist (d.h. es wird nicht mit veralteten Dokumenten bzw. Modulen gearbeitet) und dass bei Bedarf mit älteren Versionen verglichen werden kann (für die Analyse von Fehlern). Weiterhin bietet sich die Möglichkeit zwischen den Versionen einer Datei zu unterscheiden, die sich in unterschiedlichen Phasen der Software-Entwicklung befinden (z.B. Version 10 befindet sich in der Entwicklung, Version 9 im Systemtest und Version 8 bereits in Produktion).

Release-Verwaltung heißt, dass die einzelnen Releases (Auslieferungsstände) der Applikation dokumentiert und verwaltet werden. D.h. bei Bedarf kann auf einen älteren Release-Stand zurückgegriffen werden (z.B. wenn sich ein neues Release trotz Test als fehlerhaft erweist). Ein Release dokumentiert die Versionsnummern aller Dateien, die benötigt werden, um das System zu einem bestimmten Zeitpunkt ausliefern zu können. D.h. das Release 5.0.0 des Systems beinhaltet z.B. die Version 3 der Datei "applicat.dll" die Version 7 der Datei "applicat.exe" und die Version 2 der Datei "manual.doc".

4.4.2 Prozessbeurteilung

Die Qualitätsplanung sieht auch vor, dass die eingesetzten Methoden zur Qualitätsplanung ständig auf Schwachstellen untersucht und diese ausgebessert werden. Es werden auch die Prozesse untersucht und unter Umständen Verbesserungen vorgenommen, um die verlangte Qualität sicherzustellen. Standardisierte Beurteilungsschemata ermöglichen eine Einstufung des Ist-Zustand des Unternehmens im Vergleich zu Referenzmodellen und Normen. Sie ermöglichen weiters einen direkten Vergleich von Unternehmen bzw. dienen als Basis für Verbesserungsprogramme.

Bekannte Prozessbeurteilungsverfahren:

- [Capability Maturity Model \(CMM\) \[Paulk et al, 1993\]](#)
- [SPICE / ISO 15504 \(Software Process Improvement and Capability Determination\)](#)
- [Bootstrap – EU-Programm zur Prozessverbesserung](#)

- Trillium – Erweiterung von CMM, speziell für Telekommunikationssoftware

4.4.3 Prozessverbesserung

Die Prozessverbesserung wird durch einen kontinuierlichen (Meta-)Prozess definiert. Dieser ist ein Kreislauf aus Definieren (Dokumentieren, Schulen und Institutionalisieren), Verwenden (Messen, Auswerten und Lenken) und Ändern (Anpassen, Schwächen eliminieren und Automatisieren). Dieser Kreislauf unterzieht die Prozesse einer ständigen Überprüfung und versucht Schwachstellen zu eliminieren.

4.5 Personen

Qualitätsplanung muss sich auch mit den beteiligten Personen und Rollen befassen. Der Abschnitt präsentiert ausgearbeitete Beispiele für einige typische Rollen wie *Benutzer*, *Betreiber*, *Designer*, *Tester* oder *Programmierer*. Insbesondere werden auch Probleme mit der Messung von personenbezogenen Daten und Lösungsansätze aufgezeigt.

Die Ausmaße heutiger Softwareprojekte sind dermaßen groß, dass sie schon lange nicht mehr von einer Person bewältigt werden können. An einem Projekt sind also verschiedenste Personen beteiligt, sowohl an der Entwicklung als auch später im Einsatz. Jede dieser Personengruppen hat verschiedene Anforderungen an das System und deshalb auch verschiedene Anforderungen an die Qualität des Produkts. Diese verschiedenen Qualitätssichten sind ein wichtiger Aspekt für das Qualitätsmanagement.

4.5.1 Benutzer

Die erste Gruppe sind die Benutzer des Systems. Unter Benutzern kann sowohl eine Person als auch ein anderes technisches System verstanden werden. Die Qualitätsanforderungen dieser Gruppe betreffen meist Benutzerschnittstelle bzw. Systemschnittstellen. Die Anforderungen sind, wenn es sich nicht um eine technische Schnittstelle zu einem anderen System handelt, meist subjektiver Natur und beinhalten z.B. einfache Bedienbarkeit und Übersichtlichkeit der Benutzeroberfläche.

4.5.2 Betreiber

Unter Betreibern eines Produkts verstehen wir Institutionen, welche das Produkt in ihrem Einflussbereich den Benutzern zur Verfügung stellen. Die Betreiber organisieren die Verwendung des Software Produkts und legen den Lifecycle fest. Die Anforderungen dieser Gruppe sind einsatzorientiert und betreffen die Zukunft des Produkts. Als Beispiele wären einfache Wartbarkeit, geringe Anschaffungs- und Betriebskosten sowie effizientere Arbeit der Benutzer mit dem Produkt und als Folge eine Produktivitätssteigerung zu nennen.

4.5.3 Designer

Die Designer legen die technische Struktur des Produkts fest. Diese besteht aus verschiedenen Komponenten und deren Aufgaben im Gesamtsystem. Die Qualitätsziele dieser Gruppe betreffen meist die Strukturierung der Architektur und zielen auf die Erfüllung der Spezifikation bzw. des Pflichtenheftes und damit die Befriedigung von Benutzer- und Betreiberanforderungen ab. Es kann aber auch vorkommen, dass interne Anforderungen wie einfache Erweiterbarkeit für geplante Updates berücksichtigt werden müssen.

4.5.4 Programmierer

Die Programmierer realisieren die Komponenten des Systems entsprechend den festgelegten Aufgaben in Form von Programmen und Modulen. Ihre Qualitätsziele liegen in der Regel in der Programmstruktur, dem Programmierstil, den einzelnen Algorithmen und der Dokumentation.

4.5.5 Tester

Der Tester ist verantwortlich für die Kernaktivitäten des Testens, welche die Durchführung der eigentlichen Tests und die Ergebnisprotokollierung einschließen. Seine Aufgabe ist es geeignete Testmethoden und Verfahren zu bestimmen oder zu entwickeln. Besonders wenn eine Testautomatisierung gefordert wird sind die Ansprüche an den Tester besonders hoch. Ein großer Teil der eingeplanten Zeit für ein Softwareprojekt wird für das Testen aufgewendet und ist daher ein wichtiger Bestandteil des Softwareentwicklungsprozesses. Auch der Aufbau einer Testumgebung fällt in seinen Zuständigkeitsbereich. Die Ergebnisse der Tests müssen dokumentiert und verifiziert werden. Der Umfang und die Durchführung der Tests tragen entscheidend zur Qualität einer Software bei, da hier Fehler und Schwachstellen identifiziert und rechtzeitige Änderungen veranlasst werden müssen.

4.6 Zusammenfassung

Um Qualität im Softwareprojekt sicherzustellen, muss Qualität und ihre Ziele definiert und das Erreichen von Qualitätszielen gemessen und verifiziert werden. Die dafür notwendige Planung ergänzt die bereits vorhandene Projektmanagementplanung um den Qualitätsmanagementaspekt.

Die wesentlichen Aufgaben innerhalb des Projektmanagements sind die Schätzung von Aufwand und Terminen, die Durchführung von Machbarkeitsstudien und das Treffen einer vertraglichen Projektvereinbarung. Für den ersten Punkt gibt es Schätzverfahren, wie zum Beispiel Analogiemethode, Multiplikatormethode, Gewichtsmethode oder Prozentsatzmethode, die bei der Abschätzung des Aufwands unterstützen sollen.

Bei der Durchführung von Machbarkeitsstudien geht es neben dem Projekthalt vor allem um die Beleuchtung der Projektumgebung. Dabei werden der Projekttyp, die geforderten Funktionalitäten und Qualitäten, sowie mögliche Risiken genauer unter die Lupe genommen. Im Rahmen der Projektvereinbarung werden wesentliche Punkte des Projekts vertraglich festgehalten. Damit sollen böse Überraschungen während des Projekts vermieden werden.

Bei der Qualitätsplanung spielt neben der Qualitätsdefinition die Qualitätsmessung eine wichtige Rolle. Da Qualität an sich nicht als objektive Größe messbar ist, müssen Kriterien aufgestellt werden, an denen das Ausmaß an Qualität gemessen werden kann. Diese Kriterien betreffen nicht nur das Produkt, sondern auch die verwendeten Prozesse und die agierenden Personen. Eine Möglichkeit, Qualität messbar zu machen, ist die Goal-Question-Metrik, bei der ausgehend von definierten Qualitätszielen Fragen formuliert werden, deren Beantwortung als Nachweis für die Erfüllung der geforderten Eigenschaften verwendet werden kann.

Die Qualität eines Prozesses kann durch folgende Merkmale definiert werden: Effektivität, Effizienz, Kontrollierbarkeit, Steuerbarkeit, Anpassbarkeit. Für die Beurteilung der Prozessqualität gibt es mehrere Verfahren, wie CMM, SPICE, Bootstrap oder Trillium.

Damit Qualitätsmanagement in einem Unternehmen funktioniert, ist aber nicht nur die Qualität von Prozessen wichtig, sondern auch die der Managementvorgänge. Die wesentlichste Rolle dabei spielt die Vorbildfunktion (*leadership*) des Managements. Überhaupt spielt die Einstellung aller an einem Projekt beteiligten Personen zur Qualität bzw. zur qualitativ hochwertigen Arbeitsweise, eine große Rolle.

4.7 Literaturreferenzen

[Berczuk, 2002] Berczuk, Steve; Appelton, Brad: „Software Configuration Management Patterns: Effective Teamwork, Practical Integration“, Addison-Wesley Verlag, ISBN: 0-201-74117-2.

[Kellner, 2001] Kellner, Hedwig: „Die Kunst, IT Projekte zum Erfolg zu führen“, Hanser Verlag, ISBN: 3-446-21673-1.

[McConnel, 1997] McConnel, Steve: „Software Project Survival Guide“, Microsoft Press, ISBN:1-572-31621-7.

[Mette et al, 2003] Mette, Anne; Hass, Jonassen; Hass, Glenn: „Configuration Principles and Practice“, Addison Wesley Verlag, ISBN: 0-321-11766-2.

[Paulk et al, 1993] Paulk, Mark C.; Curtis, Bill; Chrissis, Mary Beth; Weber, Charles V.: „Capability Maturity Model For Software, Version 1.1“, Software Engineering Institute, CMU/SEI-93-TR-24, DTIC Number ADA263403, February 1993.

[Paulk et al, 1993] Paulk, Mark C.; Weber, Charles V.; Garcia, Suzanne M.; Chrissis, Mary Beth ; Bush, Marilyn W.: „Key Practices of the Capability Maturity Model For Software, Version 1.1“, Software Engineering Institute, CMU/SEI-93-TR-25, DTIC Number ADA263432, February 1993.

[PMI] PMI: „A Guide to Project Management Body of Knowledge“, Project Management Institute, <http://www.pmi.org>.

[Thaller, 2000] Thaller, Georg Erwin: „Software Qualität“, VDE Verlag, 2000, ISBN: 3-800-72494-4.

[Wallmüller, 2001] Wallmüller, Ernest: „Software-Qualitätsmanagement in der Praxis“, 2. völlig überarbeitete Auflage, Hanser Fachbuch, 2001, ISBN: 3-446-21367-8.

4.8 Übungen und Fragen

1. Nennen Sie Methoden zur Aufwandsabschätzung und beschreiben Sie deren Funktionsweise!
2. Beschreiben Sie 3 negative Szenarien, die bei einer nicht gut durchdachten Projektvereinbarung, entstehen können!
3. Was ist die Goal-Question-Metrik und wie funktioniert sie?
4. Welche 5 Prozess-Reifegrade gibt es? Zählen Sie diese auf und nennen Sie zusätzlich 3 Prozessbeurteilungsverfahren!