

IT-Projektmanagement

Unterstützende Prozesse: QM

Kaiserslautern, WS 2008/2009

Dr. Gerhard Pews

Der Fahrplan durch die Vorlesung

Inhalte

- Einführung
- Das „Was“: Der Gegenstand von Softwareprojekten
- Das „Wie“: Die Tätigkeiten in einem Projekt und wie man sie ausführt
- Vorbereitung eines Projekts
- Projektplanung
- Durchführen eines Projekts
- **Unterstützende Tätigkeiten**
- Soft Factors
- Wirtschaftliche Aspekte

AGENDA

- **Qualitätsmanagement – Motivation**
- Grundlegende Anmerkungen
- QM im Projekt
- QM und Projektmanagement
- QM Normen und Modelle

Qualität

...klingt irgendwie etwas langweilig.

Qualität

- Ist als Begriff oft mit schlechter Qualität verbunden
 - Qualität wird oft synonym genutzt mit
 - Dauerhaftigkeit, Stabilität, Fehlerfreiheit
 - z. B. für Kleidung, Möbel, Autos
 - Qualität ist aber eigentlich ein hoch spannendes Thema:
 - Gute Produktqualität führt zu zufriedenen Kunden
 - Gute Produktqualität führt zu *zufriedenen Mitarbeitern*
- Es macht überhaupt keinen Spaß, schlechte Qualität abzuliefern!**

AGENDA

- Qualitätsmanagement – Motivation
- **Grundlegende Anmerkungen**
- QM im Projekt
- QM und Projektmanagement
- QM Normen und Modelle

Praxisbeispiel: Qualitätsmanager

- In der Softwarefirma XY wurde der Mitarbeiter Müller zum Qualitätsmanager ernannt.
- Die Gründe: er war derjenige, auf den man im Projektgeschäft am einfachsten verzichten konnte.
- Er stellt ein Monster an Formalismen zusammen und nennt es Qualitätsmanagementsystem.
- Man beauftragt eine kleine (von ihren Auftraggebern abhängige) Firma mit der Zertifizierung nach ISO 9000 und erreicht sie tatsächlich auch.

und was ist mit der Software? Die ist immer noch so schlecht wie vorher.

Der Begriff Qualität

- In den folgenden Folien lösen wir uns bei der Motivation vom Begriff Qualität und reden von: „so richtig guter Software“.
- Fragen:
 - Was macht denn Software so richtig gut?
 - Was ist Ihre Lieblingssoftware und was gefällt Ihnen daran?
 - Welche Software gefällt Ihnen nicht? Warum?

Gute Software ist z. T. subjektive Empfindung

- Hängt ab von der Erwartungshaltung des Nutzers
- Hängt ab von sonstigen Eigenschaften und Kontext
 - Spiel, das öfter mal abstürzt
 - Office-Programm, das öfter mal abstürzt
- Aber: es gibt auch klar messbare Kriterien
- Die Zufriedenheit des Kunden hängt aber nicht nur von der Software ab (vgl. Projektinitialisierung – Kundenbefragung)

Wie schreibt man richtig gute Software?

- Praxisbeispiele
 - Beispiel 1 (Wiederholung):
 - PL: Wir liefern am 4. Juli aus.
 - Mitarbeiter: Das schaffen wir nicht.
 - PL: Das ist mir egal. Wenn ich sage, wir liefern aus, dann liefern wir aus. Strengen Sie sich an.
 - Beispiel 2:
 - Software wurde entwickelt
 - Software wird an Kunden übergeben
 - Das Programm, das nachts Daten vom Vortag verarbeiten soll, läuft 46h lang.

Maßnahmen

- Was könnte man in solchen Situationen tun?
- Ein gutes Ergebnis erzielt man, indem man die richtigen Dinge tut.
- Ein gutes Ergebnis erzielt man nicht durch Abnahmeprüfungen. Prüfungen stellen sicher, dass keine Fehler passiert sind. Sie sind eine letzte Sicherheitsmaßnahme.
- Tests, Prüfungen sind ein Beispiel für *analytische* Qualitätssicherung.
- *Konstruktive* Qualitätssicherung verbessert den Software-Erstellungsprozess, der im Projekt gelebt wird.

Wer sorgt für Qualität?

- In der Praxis oft gehörter Satz:
 - „Die Frau Schmidt ist für die Qualität zuständig.“
 - Übersetzt hieße das: „Unser Projekt schreibt so richtig gute Software. Das macht die Frau Schmidt“. Klingt nicht nur unplausibel, ist auch so.
- Damit richtig gute Software rauskommt, muss sich jeder im Projekt anstrengen
- Übersetzt: Qualität geht alle an.

Fazit

- Qualität geht jeden an. Jeder einzelne im Projekt sorgt dafür. Ein Qualitätsbeauftragter unterstützt dabei, mehr aber auch nicht.
- Es gibt konstruktive und analytische Maßnahmen zur Qualitätssicherung.
 - Mehr Qualität erzeugen kann man nur mit konstruktiven Maßnahmen.
 - Analytische Maßnahmen stellen nur sicher, dass ein Qualitätsstandard erreicht ist.

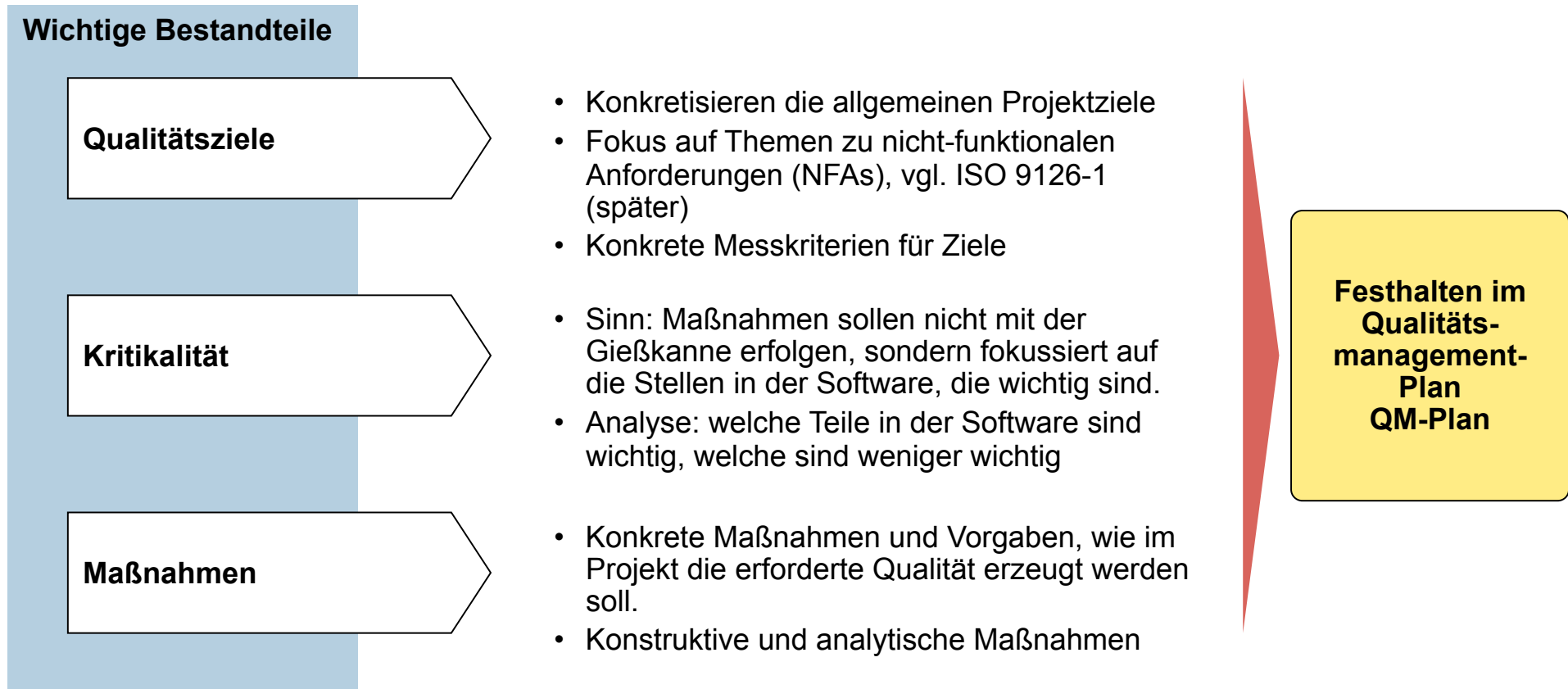
Wichtig: Man kann keine Qualität in Software hineinprüfen. Qualität kann man nur in Software hineinkonstruieren.

AGENDA

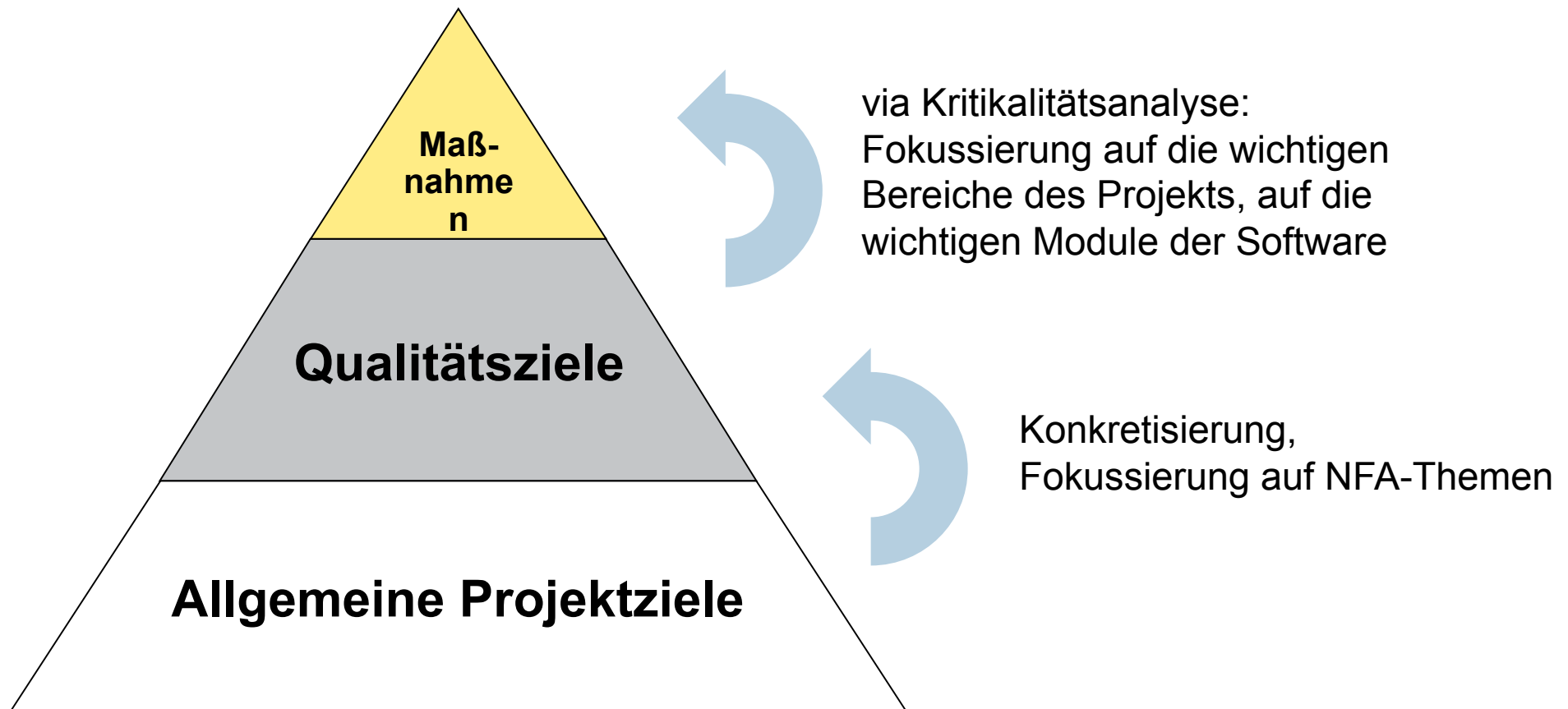
- Qualitätsmanagement – Motivation
- Grundlegende Anmerkungen
- **QM im Projekt**
- QM und Projektmanagement
- QM Normen und Modelle

Die Umsetzung von QM im Projekt hat drei wesentliche Bestandteile: Qualitätsziele, Kritikalitätsanalyse, Festlegen von Maßnahmen.

**Wichtig: Qualitätssicherung im Projekt ist ein Prozess
(wie z. B. auch Projektleitung oder Planung)**



Projektziele, Qualitätsziele und Maßnahmen bauen aufeinander auf.



Qualitätssicherung ist zu Projektbeginn wichtig

- In der Projektinitialisierung werden die wesentliche Weichenstellungen für das Projekt vorgenommen:
 - Das Team wird zusammen gestellt
 - Ziele werden fixiert
 - Die Projektplanung wird festgelegt
- Hier ist der Qualitätssicherer der Sparringspartner des Projektleiters. Falls noch niemand für die Qualitätssicherung benannt ist, sollte der Projektleiter sich hier zumindest ein Gespräch mit einem Qualitätsberater führen.

Konkrete Maßnahmen zur Qualitätssicherung

- Praxisbeispiel

Maßnahmen im Projekt (1/4)

- Generell: als Qualitätssicherer muss man kreativ sein!
- Reviews (analytisch)
 - Code Reviews
 - Dokumentenreviews
 - Workshops, zu denen externe Experten eingeladen werden. Zwischenergebnisse werden vorgestellt und diskutiert
- Tests (analytisch)
 - Funktionale Tests, auch als Regressionstests
 - Lasttests
 - etc.
- Durchstich
 - fachlich
 - technisch

Maßnahmen im Projekt (2/4)

- Aufbau einer insgesamt gut nutzbaren Entwicklungsumgebung (Editoren, Build-Werkzeuge, Code-Analyzer, Test-Frameworks, Anbindung an das Konfigurationsmanagement, Modellierungswerkzeuge, Generatoren, Debugger, etc.)
- Ausbildungsmaßnahmen für Mitarbeiter
 - Schulungen
 - Workshops
 - Vorträge
 - Team-Rotation
- Verbesserung der Kommunikation
 - Spezielle Meetings, Statusrunden
 - Vorträge über Fachthemen

Maßnahmen im Projekt (3/4)

- Guidelines, Richtlinien
 - Coding-Richtlinien
 - Styleguides
 - gemeinsame Glossare
 - Richtlinien zur Erstellung von Dokumenten
- Entwickeln von Checklisten, Templates
 - Operationalisieren die Richtlinien/Guidelines

**→ Vorsicht! Nicht übertreiben! Keine Schrankware produzieren.
Richtlinien müssen noch anwendbar bleiben.**

Maßnahmen im Projekt (4/4)

- Projektablage (Verzeichnis oder Konfig-Mgmt) organisieren
- Dafür sorgen, das man Informationen wieder finden kann:
 - Konventionen für die Ablage von Dokumenten
 - Beschaffen eines Such-Tools
 - Händischer Index (z. B. HTML) über die wichtigsten Dokumente
 - ...

Maßnahmen im Projekt

Auditierung

- Audit:
 - Intensive Untersuchung eines Projekts
 - In der Regel durch externe Auditoren
 - Sichtung von Unterlagen
 - Führen von Interviews
 - Kann sowohl formale Kriterien (einfach) als auch Projektinhalte und Vorgehensweise (schwierig) prüfen.
 - Ergebnisse:
 - Bewertung des Projektstatus, z. B. in Form einer Projektampel
 - Liste vorgeschlagener Maßnahmen

Sinnhaftigkeit von Audits

- Hat ein Auditor überhaupt eine Chance herauszubekommen, was im Projekt los ist?
- Vorab: Ein Audit ist hilfreich
 - Die Vorbereitung auf ein Audit selbst ist schon hilfreich
 - Offenheit hilft, weil man als Auditierter selbst ein offenes Feedback bekommt und man das Projekt verbessern kann.
 - Ein Audit als bloße Kontrollveranstaltung mit anschließender Bestrafung verfehlt seinen Zweck.
- Einschätzung der Projektsituation ist schwierig:
 - gerade bei kurzen Audits (wenige Stunden).
 - wenn die Auditiererten sich selbst und die Projektsituation falsch einschätzen und diese Meinung überzeugend vertreten.
 - Trotzdem zeigt die Praxis, dass der Auditor schon nach kurzer Zeit einen guten Eindruck von der Projektsituation bekommt.
- Bei richtiger Handhabung des Werkzeugs Audit sind „Verschwörungen“ eines Projekts gegen den Auditor nicht nötig und finden nicht statt.

Beispiel für Auditierung

- Vorab werden die wichtigsten Projektunterlagen an die Auditoren verteilt: Projekthandbuch, Projektauftrag, Projektplanung, QS-Plan
- Auditoren bewerten die Unterlagen und führen Interviews. Interviewed werden: Projektleiter, Qualitätsbeauftragter, ggf. Auftraggeber und Projektmitarbeiter
- Auditoren erstellen einen Auditbericht mit einer Bewertung und vorgeschlagenen Maßnahmen.
- Nach einiger Zeit (z. B. 4-6 Monaten) erfolgt ein Nachaudit oder eine Nachbesprechung, um die Wirkung der Maßnahmen zu überprüfen und den Status neu zu bewerten.

Fragestellungen im Audit (1/2)

generell: im Audit kann man die Dinge abprüfen, die "so ganz einfach und selbstverständlich" klingen. Dies sind die bisher in der Vorlesung vorgestellten Inhalte.

- **Projektauftrag:** Ist der Projektauftrag klar definiert? Wo ist der? Ist der unterschrieben? Sind die Ziele klar formuliert? Ist klar, welche Ergebnisse das Projekt liefern soll?
- **Risiken:** Was sind die 5 Top-Risiken im Projekt? Hat dazu jeder im Projekt die gleiche Meinung?
- **Planung:** Gibt es eine aktuelle Planung? Kann der PL diese erklären? Ist die Planung im Team bekannt?
- **Projekthandbuch:** Gibt es ein aktuelles Projekthandbuch?

Fragestellungen im Audit (2/2)

- **Projektvorgehen:** Ist mit dem Auftraggeber das Abnahmeverfahren geklärt? Ist das CR-Verfahren mit dem Auftraggeber abgesprochen? Ist das Verfahren allen bekannt?
- **Organisatorisches:** Ist das passende Know-how im Team vorhanden? Kennt jeder im Projekt das Eskalationsverfahren?
- **Projektstruktur:** Wie sieht die Projektstruktur aus? Ist das Projekt gut gegliedert oder ein unstrukturierter Klumpen?
- **QS-Plan:** Welche QS-Maßnahmen sind geplant?

AGENDA

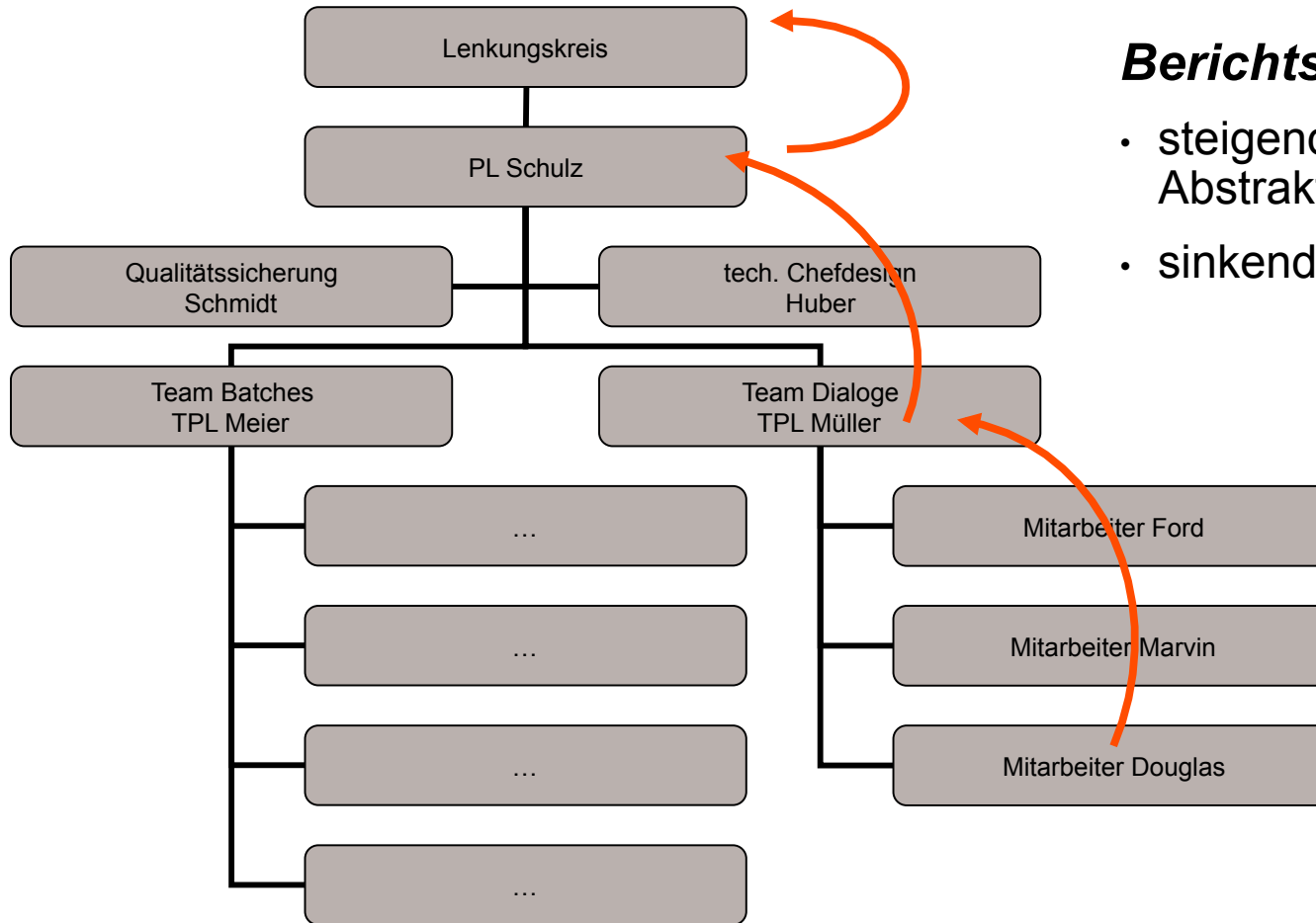
- Qualitätsmanagement – Motivation
- Grundlegende Anmerkungen
- QM im Projekt
- **QM und Projektmanagement**
- QM Normen und Modelle

QM und Projektmanagement haben Überschneidungen

- Qualitätsmanagement und Projektmanagement sind beides keine „theoretischen Wissenschaften“, sie leben von der praktischen Umsetzung.
- Im Idealfall agieren PL und QS Hand in Hand. Der Qualitätssicherer ist Sparringspartner des Projektleiters und bringt sich konstruktiv mit ein.
- Im Problemfall benötigt allerdings der Qualitätssicherer eine Möglichkeit zur Eskalation.
 - Im „Teufelsquadrat“ ist Qualität eine relativ schwache Ecke.
 - Kosten, Zeit, Lieferumfang sind einfacher zu messen als die Qualität

Controlling entlang der Projekthierarchie

Wiederholung



Berichtsweg:

- steigender Abstraktionsgrad
- sinkende Häufigkeit

Wir stehen kurz vor dem Abgabetermin. Wesentliche Teile unserer Software weder fertig noch getestet. Ich sehe keine Chance, dass wir den Termin halten können.



Na gut, einigen wir uns auf
gelb.

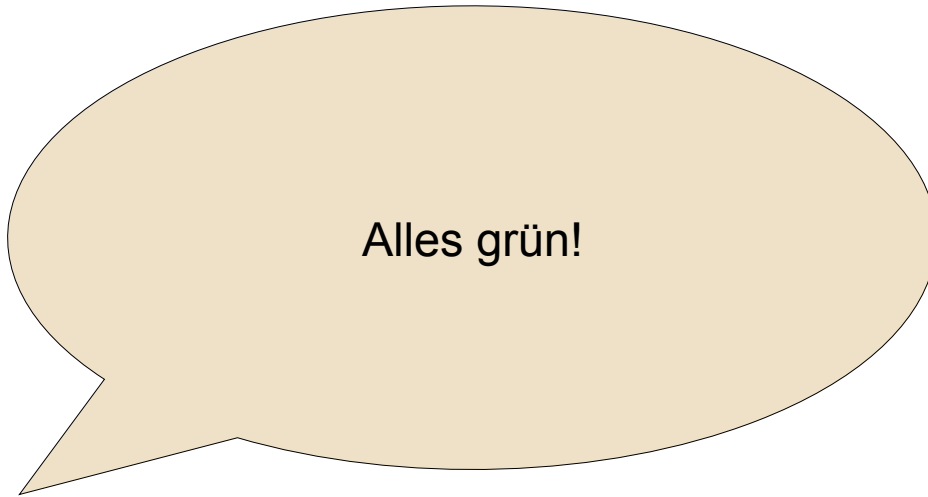


Das Projekt ist gelb. Es wird zwar schwer, den Abgabetermin zu halten, wir könnten es aber noch schaffen.



Also, so lange es noch
eine Chance gibt, ist das
Projekt für mich grün!



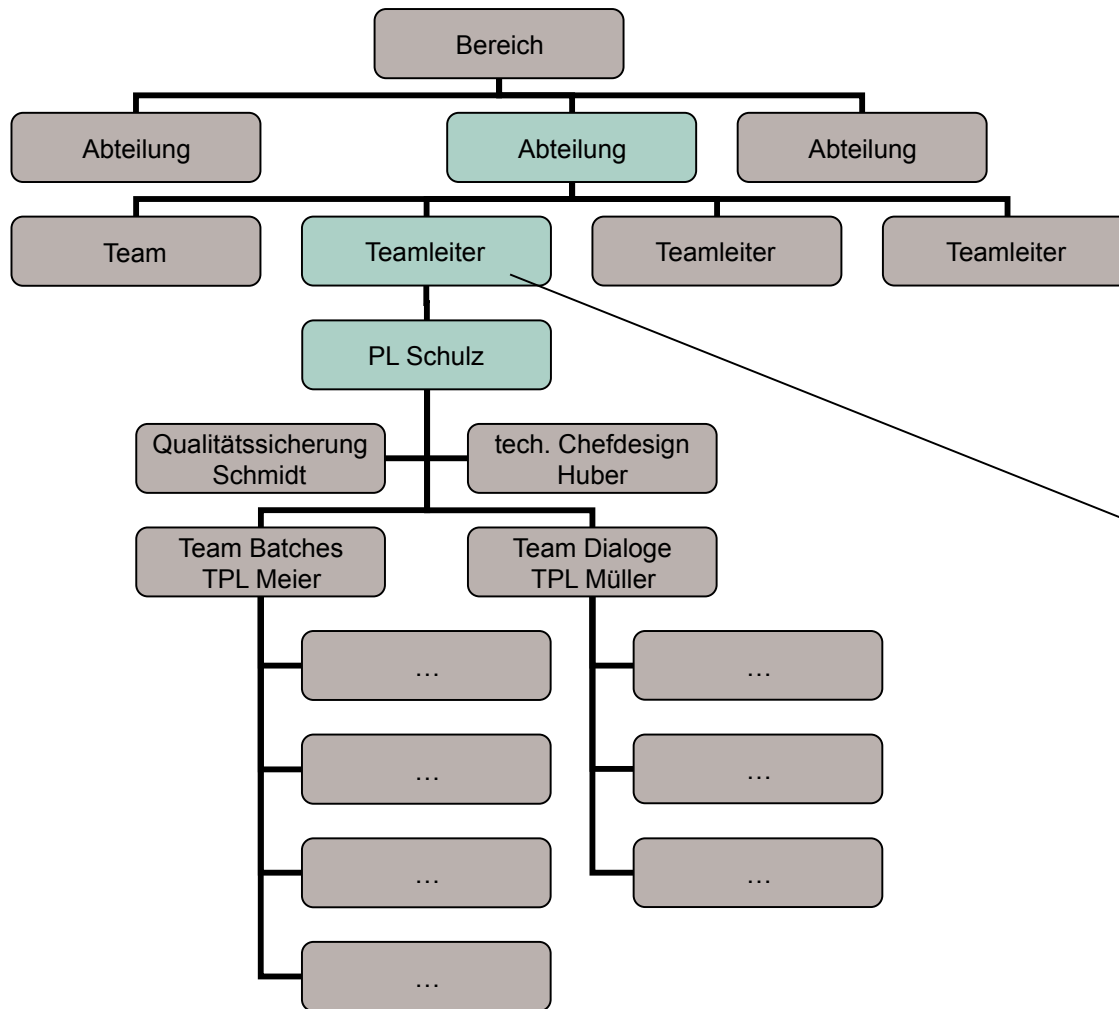


Dank unserer ständigen
Statusberichte haben wir alles unter
Kontrolle und können frühzeitig auf
Probleme reagieren.

Das nenne ich modernes
Management!



Eskalation entlang der Projekthierarchie

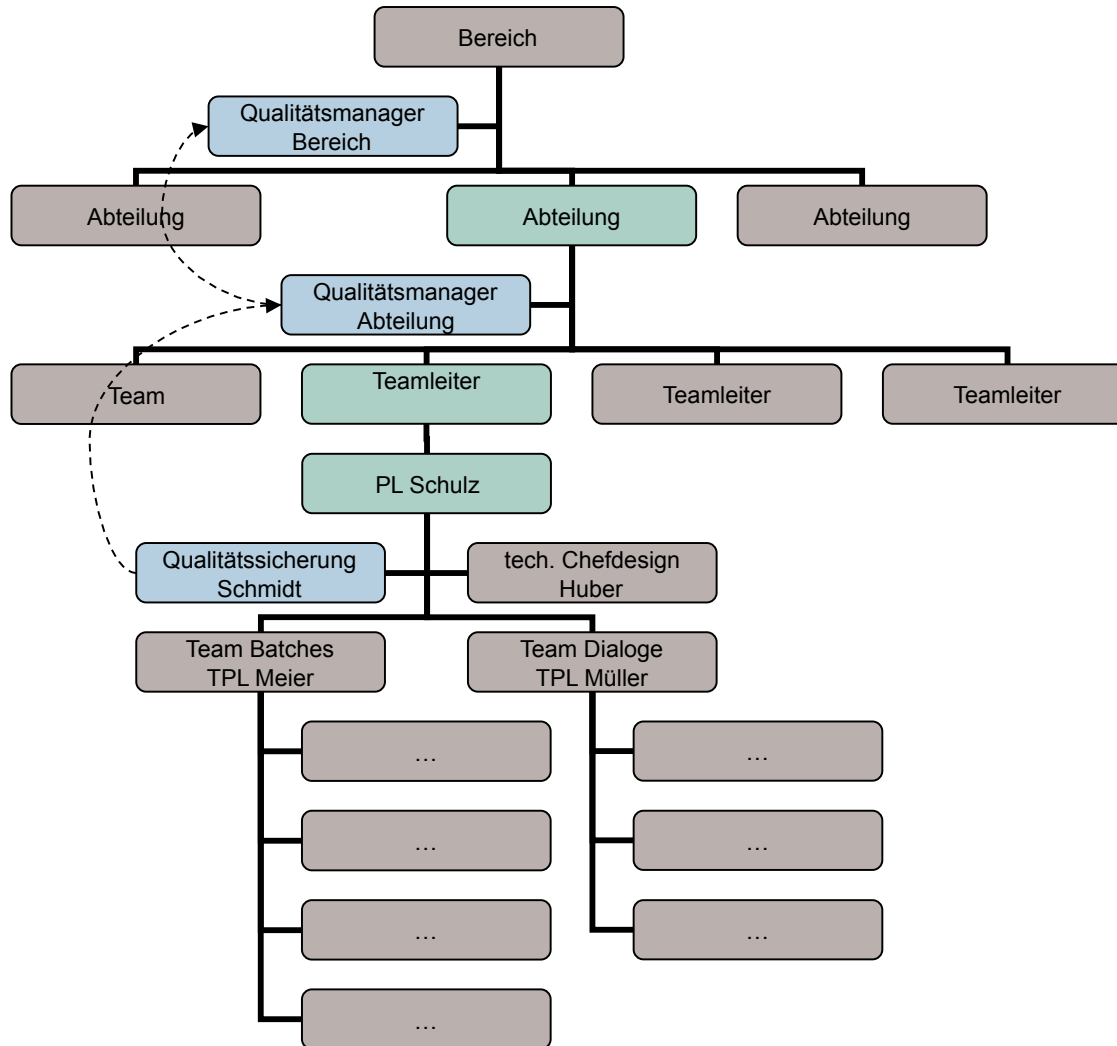


Gefahr:

- Qualitätsprobleme werden entlang des Berichtsweges totgeschwiegen

Mitglied im Lenkungskreis, Chef von Schulz)

Eskalation entlang der Projekthierarchie



- Der Aufbau einer eigenen Hierarchie parallel zur Projekthierarchie sorgt dafür, dass Qualitätsprobleme bei Bedarf eskaliert werden können

AGENDA

- Qualitätsmanagement – Motivation
- Grundlegende Anmerkungen
- QM im Projekt
- QM und Projektmanagement
- **QM Normen und Modelle**

Vorbemerkungen

- Qualitätsmanagement ist eine eigene Vorlesung, im Folgenden geht es um die Ideen und die Einordnung zum Projektmanagement.
- In einem Formalismus stecken auch Gefahren:
 - Es können riesige Regelwerke entstehen, die niemand mehr überblicken kann und deren Inhalte nicht gelebt werden können. Hat ein Formalismus erst einmal diese Größe erreicht, ist er tot. Auch die vorhandenen sinnvollen Inhalte sind dann nutzlos.
 - Eine „hauptberufliche“ QS neigt zur Überreglementierung, indem jeder Handschlag vorgeschrieben ist. Das tötet den Spaß an der Arbeit und letztendlich auch die Qualität.
 - Einen Formalismus betreiben und überwachen kann man auch, wenn man keine Ahnung von den eigentlichen Inhalten der Arbeit hat. Dann hält man sich an Formalia fest, die eigentlich unnütz oder sogar schädlich sind.

Vorbemerkungen

- Jeder leere Formalismus ist nutzlos. Durch Formalismen lässt sich Qualität nicht erzwingen. Wenn Sie im Projekt auf solche Fossilien treffen, beseitigen Sie sie.
- Wenn aber im Projekt Qualität gelebt wird, sind Formalismen hilfreich
 - Sie geben einen Rahmen.
 - Sie stellen sicher, dass nichts übersehen wird oder in Vergessenheit gerät
 - Sie können die Priorität der Ecke „Qualität“ im Teufelsquadrat erhöhen.
- Genau so sollen im Folgenden die Normen und Modelle betrachtet werden: wie helfen sie uns, gute Software zu produzieren?

ISO 9000 ff – Qualitätsmanagementsystem

- ISO 9000 ff behandelt Qualitätsmanagement allgemein – nicht nur für Software. Teil 9000-3 bezieht sich auf Software.
- Idee: Die Gesamtheit der QS-Prozesse, Organisationsstrukturen etc. einer Firma bildet das Qualitätsmanagementsystem (QMS). Man kann Anforderungen an ein solches System definieren und ein QMS zertifizieren. Diese Zertifizierung muss in regelmäßigen Abständen wiederholt werden.
- Fragestellungen:
 - Sind die Prozesse festgelegt und dokumentiert?
 - Werden diese Prozesse auch umgesetzt?
 - Führen diese Prozesse zu guten Resultaten?

ISO 9000 ff – Praxisbeispiele

- Negativbeispiel: „erschlichene Zertifizierung“
- Positivbeispiel:
 - Eine Firma hat ein funktionierendes, schlankes QMS
 - Die Auditierung für die Zertifizierung nach ISO 9000 hält diesem QMS den Spiegel vor: Wo kann man das QMS noch verbessern? Wo sind Punkte, die man aus Betriebsblindheit übersehen hat?
 - Die angestrebte Zertifizierung erzeugt im Projektalltag einen gewollten Druck, damit QS-Aspekte nicht untergehen können.
 - Die Produktqualität steigt, Qualität ist im Bewusstsein aller verankert
 - Die bestandene Zertifizierung ist eine „Belohnung“, ein Lob, über das sich alle freuen.

ISO 9126-1 Qualitätsmerkmale

- Fragestellung: woran misst man Qualität?
- Problemstellung im Projekt:
 - Sicherstellen, dass kein wichtiger Aspekt übersehen ist:
 - Bei der Definition von Qualitätszielen im QM-Plan
 - Beim Erheben von Anforderungen
 - Ein Auftraggeber weiß unter Umständen gar nicht, dass er eine Anforderung zu einem bestimmten Kriterium hat oder er hält sie für so selbstverständlich, dass er sie gar nicht nennt.
- ISO 9126-1 enthält ein Qualitätsmodell, in der Praxis am wichtigsten sind die Qualitätsmerkmale (quality model for external and internal quality)

Qualitätsmerkmale nach ISO 9126-1 und umgangssprachliche Erklärungen (1/5)

- Functionality
 - Suitability: Kann man mit der Anwendung das machen, was man mit ihr tun will? Ist der Funktionsumfang richtig?
 - Accuracy: Liefert die Anwendung die richtigen Ergebnisse in der gewünschten Präzision?
 - Interoperability: Arbeitet die Anwendung korrekt mit anderen Systemen zusammen?
 - Security: Schützt die Anwendung die enthaltenen Daten und Funktionen gegen unautorisierten Missbrauch?

Qualitätsmerkmale nach ISO 9126-1 und umgangssprachliche Erklärungen (2/5)

- Reliability
 - Maturity: Stürzt die Software wegen Fehlern nur selten ab?
 - Fault tolerance: Bleibt die Software funktionsfähig, auch wenn falsche Eingaben gemacht werden oder interne Fehler auftreten?
 - Recoverability: Wenn die Software abgestürzt ist, lassen sich später die Software und die Daten wiederherstellen?

Qualitätsmerkmale nach ISO 9126-1 und umgangssprachliche Erklärungen (3/5)

- Usability
 - Understandability: Versteht man, was die Software macht und ob sie das richtige macht, wenn man mit ihr arbeitet?
 - Learnability: Ist die Nutzung der Software leicht zu erlernen?
 - Operability: Kann man die Software leicht bedienen?
 - Attractiveness: Macht es Spaß, mit der Software zu arbeiten?
- Efficiency
 - Time behaviour: Ist die Software schnell genug, verarbeitet sie schnell genug die Daten?
 - Resource utilisation: Geht die Software sparsam genug mit den Ressourcen, insbesondere der Hardware, um?

Qualitätsmerkmale nach ISO 9126-1 und umgangssprachliche Erklärungen (4/5)

- Maintainability (Wartung und Weiterentwicklung)
 - Analysability: Wie schnell finde ich die Stelle, die ich ändern will?
 - Changeability: Wenn ich die Stelle gefunden habe, wie leicht lässt sich etwas ändern?
 - Stability: Haben Änderungen nur begrenzte Auswirkungen?
 - Testability: Wie leicht kann ich die Software testen?

Qualitätsmerkmale nach ISO 9126-1 und umgangssprachliche Erklärungen (5/5)

- Portability (Umziehen in eine andere Umgebung: HW, SW, Organisation)
 - Adaptability: Kann man die Software durch Änderung an dafür vorgesehenen Stellen an eine andere Umgebung anpassen?
 - Installability: Lässt sich die Software in einer bestimmten Umgebung einfach installieren?
 - Co-existence: Verträgt sich die Software mit anderen Systemen oder blockiert sie z. B. gemeinsam genutzte Ressourcen?
 - Replaceability: Lässt sich die Software gegen eine andere Software austauschen, die für den gleichen Zweck gemacht ist?

Praxisbeispiel ISO 9126-1

- In einem Großprojekt werden die Anforderungen systematisch erfasst.
- Die Anforderungen werden nach den Kriterien der ISO 9126 gegliedert (insbesondere die nicht-funktionalen Anforderungen) Dazu mussten aber noch weitere Gliederungspunkte ergänzt werden:
 - Sicherheit, dass keine relevanten Punkte vergessen wurden
 - Übersichtlichkeit und bessere Kommunikation zum Auftraggeber durch ein allgemein bekanntes Raster

CMMI

- Capability Maturity Model
- von der Carnegie Mellon University (Software Engineering Institute) entwickeltes Qualitätsmanagementmodell (1991)
- 2002 Aktualisierung zu Capability Maturity Model Integration, in der zwischenzeitlich entstandene Spezialformen des CMM integriert wurden: Software-CMM, Software Engineering-CMM, Integrated Product Development-CMM
- Idee:
 - Ausgangsfrage: Wie reif ist eine Organisation? Wie reif sind ihre Software-Engineering Prozesse?
 - Definition von fünf Stufen, um die Reife einer Organisation zu bewerten.

CMMI-Level und umgangssprachliche Beschreibung (1/2)

- Level 1: „Initial“
 - Wir machen es irgendwie.
 - Wir wissen nicht so genau, wie wir es machen.
- Level 2: „Managed“ (alter Name: „Repeatable“)
 - Es gibt einen Software-Engineering Prozess.
 - Was wir einmal hinbekommen haben, kriegen wir auch erneut hin.
 - Das ist aber auch abhängig von den jeweiligen Personen.
- Level 3: „Defined“
 - Der Software-Engineering Prozess ist standardisiert.
 - Wir können den Prozess auch umsetzen, wenn es mal schwierig wird.

CMMI-Level und umgangssprachliche Beschreibung (2/2)

- Level 4: „Quantitatively Managed“ (alter Name: „Managed“)
 - Der Prozess wird organisationsweit standardisiert durchgeführt.
 - Die Ausführung des Prozesses wird dokumentiert und mit Kennzahlen gemessen.
- Level 5: „Optimizing“
 - Aus den Kennzahlen leiten wir ab, wie wir uns verbessern können.

Zusammen. Für nachhaltigen Erfolg.

ZUSAMMEN. FÜR NACHHALTIGEN ERFOLG.

