

# Software- Qualitätsmanagement

**Vorlesung im Modul 10-202-2319  
Software-Management**

Sommersemester 2015

Prof. Dr. Hans-Gert Gräbe

<http://bis.informatik.uni-leipzig.de/HansGertGraebe>

### outside-in

- Kombination aus top-down und bottom-up, um die Vorteile beider zu vereinen und die Nachteile zu minimieren
- man beginnt gleichzeitig von oben sowie von unten und arbeitet zur Mitte hin

### inside-out

- dieselbe Überlegung, aber man beginnt mit den System-Komponenten in der Mitte der Hierarchie und arbeitet nach oben und nach unten
- vereint eher die Nachteile von *top-down* und *bottom-up*, daher nur u. U. mit hardest-first einsetzen

### hardest-first

- zuerst werden die kritischen, d.h. potenziell fehlerhaften und am schwierigsten zu implementierenden Komponenten implementiert und getestet
- damit wird diese Komponente besonders oft getestet

## Integrationstests

Die Integrationstests dienen der Überprüfung der **Schnittstellen** zwischen den System-Komponenten.

- Aufruf von Operationen, Funktionen und Prozeduren mit und ohne Parameterübergabe
- Verwendung von globalen Variablen oder Dateien
- Benutzung von global vereinbarten Konstanten und Typen
- Analyse der Kopplung zwischen den Komponenten

Integrationstests werden unter verschiedenen Aspekten ausgeführt

### Dynamischer Integrationstest

- **Ziel:** Plausibilität der funktionalen Korrektheit durch Testfälle, die ausschließlich in der Integrationsphase nachweisbare Fehler abdecken.
- **Vorgehen:** Stichprobentest, wie beim funktionalen Test von Komponenten

### Strukturorientierter Integrationstest

- wie Strukturtests für Komponenten
- *Kontrollflussorientierter Integrationstest* betrachtet die unterschiedlichen Aufrufbeziehungen (Exporte und Importe) zwischen Komponenten. Mögliche Überdeckungskriterien:
  - jeder Aufruf jeder exportierten Operation muss in jeder importierenden Komponente wenigstens einmal überdeckt sein.
  - alle Aufrufstellen sind in allen möglichen Reihenfolgen (mit Schleifenbeschränkung) zu überdecken.
- *Datenflussorientierter Integrationstest* betrachtet die Programmstellen genauer, an denen importierte Operationen aufgerufen werden.
  - analog datenflussorientierten Tests von Komponenten

### Funktionaler Integrationstest

- Prüft die spezifizierte Funktionalität der einzelnen Komponenten und deren Zusammenwirken.
- Abweichungen liegen vor, wenn die Operation:
  - zu wenig Funktionalität liefert (z. B. fehlende Teilfunktion),
  - zu viel Funktionalität liefert (z. B. Aufruf einer unerwarteten Teilfunktion) oder
  - falsche Funktionalität liefert.
- Diese Fehler resultieren meist aus ungenauen Spezifikationen und werden beim Komponententest nicht erkannt, da dort die Spezifikation als „gesetzt“ gilt.

### Wertbezogener Integrationstest

- Schnittstellen werden mit möglichst extremen Werten getestet
- Entspricht der Grenzwertanalyse

### Statischer Integrationstest

- Analysierende Verfahren des Quellcodes der beteiligten System-Komponenten
- Untersucht die Kopplung zwischen den Komponenten und erfasst Parameter systemweiter Metriken
  - **Ziel:** zusätzliche und unnötige Kopplungen identifizieren und eliminieren
- Die *syntaktische* Kompatibilität der Schnittstellen über Komponentengrenze hinweg wird auf konstruktivem Weg erreicht
  - automatische Überprüfung der Schnittstellendeklarationen durch den Compiler (Header-Dateien, Import-Deklarationen)
- Analyse von Datenflussanomalien wie innerhalb von Komponenten
- **Verifizierende Methoden** können auch komponentenübergreifend eingesetzt werden.

# Der Systemtest

Der **Systemtest** ist der abschließende Test der Software-Entwickler und Qualitätssicherer in der realen Umgebung ohne den Auftraggeber.

- Umfasst Systemsoftware, Hardware, Bedienungsumfeld, technische Anlage
- System muss ggf. vor Beginn des Systemtests von der Entwicklung auf die Einsatz- oder Zielplattform portiert werden.
- **Basis:** Produktdefinition (Pflichtenheft, Produktmodell, Konzept der Benutzerschnittstelle, Benutzerhandbuch)
  - Pflichtenheft sollte sowohl die Qualitätsziele als auch die Testszenarien und Testfälle fixieren.
- Auf der Grundlage werden **Testfälle** aus den bisherigen Testzyklen übernommen und ergänzt.
- Zerlegung des Systemtests in verschiedene **Teiltests** an Hand zu bestimmender **Prüfziele**.
  - Prüfung aller geforderten Qualitätsziele in ihrer jeweiligen Ausprägung

### Prüfziele

- Vollständigkeit
  - Sind alle funktionalen und nicht funktionalen Anforderungen aus dem Pflichtenheft erfüllt? (**Funktionstest**)
- Volumen
  - Systemtest mit umfangreichen Datenmengen (**Massentest**)
- Zeit
  - Systemtest auf Antwortzeiten unter starker Belastung (**Zeittest**)
- Zuverlässigkeit
  - Systemtest unter längerer Spitzenlast im geforderten „grünen“ Bereich (**Lasttest**)
  - auch unter Ausfall einzelner externer Hardware- oder Software-Komponenten
  - Mehrbenutzerbetrieb im Grenzbereich
  - Reaktion auf ungewöhnliche oder widersprüchliche Daten
- Robustheit und Fehlertoleranz
  - Systemtest unter Überlast, im „roten“ Bereich (**Stresstest**)

- Benutzbarkeit
  - Test der Verständlichkeit, Erlernbarkeit, Bedienbarkeit aus der Sicht des Endnutzers (**Benutzbarkeitstest**)
  - Zielgruppenbezogen (Fachtermini, Metaphern etc.)
- Sicherheit
  - Datenschutzmechanismen, Zusammenspiel mit dem umgebenden System (**Sicherheitstest**)
- Interoperabilität
  - Relevant, wenn das System in einen größeren Verbund eingebettet ist (**Kompatibilitätstest**)
  - Kompatibilität der Schnittstellen und der Daten
- Konfiguration
  - wenn vorgesehen, Test der Systemausprägungen für verschiedene Hard- und Softwareplattformen (**Konfigurationstest**)
- Dokumentation
  - Vorhandensein, Angemessenheit und Güte der Benutzer- und Wartungsdokumentation (**Dokumentationstest**)

## Teiltests

### Funktionstest

- Test, ob alle in der Produktdefinition geforderten Funktionen vorhanden und wie vorgesehen realisiert sind.
- Testsequenzen sind aus dem Pflichtenheft zu übernehmen und/oder mit funktionalen Testverfahren systematisch und vollständig herzuleiten.

### Leistungstest

- dient der Überprüfung des in der Produktdefinition festgelegten Leistungsverhaltens
  - Massentest, Zeittest, Lasttest, Stresstest
  - Einsatz eines Testdatengenerators oder realer Daten vom Auftraggeber oder von Pilotkunden
  - Frage der Systemstabilisierung nach Überlastphasen, etwa durch den Entzug von Ressourcen

#### **Benutzbarkeitstest**

- Oft entscheidend für die Akzeptanz eines Softwareprodukts
- Kann sehr aufwändig sein, wenn darauf in der Phase der Produktdefinition zu wenig Wert gelegt wurde

#### **Interoperabilitätstest**

- heutige Systeme sind in der Regel keine alleinstehenden Systeme, sondern in eine Standardumgebung integriert
- umfasst meist eine komplexe GUI-Schnittstelle zum Betriebssystem
- Frage der Interaktion mit diesen Oberflächen (etwa mit der Zwischenablage in Windows)

#### Installations- und Wiederinbetriebnahmetest

- **Installationstest:** Prüft, ob das System mit den erstellten Installationsbeschreibungen installiert und in Betrieb genommen werden kann.
- **Wiederinbetriebnahmetest:** Prüft, ob das System nach einer Unterbrechung oder einem Zusammenbruch des Basissystems mit den vorliegenden Beschreibungen wieder in Betrieb genommen werden kann und ob noch alle Daten aktuell und verfügbar sind.

**Besonderheiten für OO-Systeme** gibt es nicht, da der Systemtest ein Black-Box-Test ist, der gar nicht bemerken kann, ob das System ein OO-System ist.

**Systemtest als Regressionstest:** Aufzeichnen der Testfälle erlaubt es, diese bei späteren Fehlerkorrekturen oder inkrementeller Software-Entwicklung relativ problemlos zu wiederholen.

# Abnahmetest

Der **Abnahmetest** ist eine besondere Ausprägung des Systemtests, bei dem das System getestet wird

- unter Mitwirkung und Federführung des Auftraggebers
- in der realen Einsatzumgebung beim Auftraggeber
- (unter Umständen) mit echten Daten des Auftraggebers

Auftraggeber kann die Testfälle aus dem Systemtest übernehmen, modifizieren und eigene Testszenarien durchführen.

- Konzentration in der Regel auf den Test unter normalen Betriebsbedingungen
- Sollte bereits im Auftrag vereinbart sein, wird aber in der Regel ein „freies Testen“ sein.
- Verfahren des Abnahmetests sollte bereits beim Systemtest zum Einsatz kommen

## Methodik aus Auftraggebersicht

- Erzeugen des zu testenden Systems aus den Quellen
  - hilfsweise Löschen aller Objektdaten
  - Bilden und Speichern einer Prüfsumme über das gesamte System, um dessen Unversehrtheit am Schluss zu prüfen
- Durchführung der Abnahme nach der vereinbarten Testvorschrift
  - Einbeziehung des Benutzerhandbuchs (mindestens alle dort angegebene Beispiele müssen funktionieren)
- regelmäßige einvernehmliche schriftliche Fixierung der Testergebnisse
- regelmäßiges freies Testen und Dokumentation dieser Testfälle
- Abnahme endet mit einer Schluss-Sitzung
  - Wichtung der protokollierten Fehler
  - Entscheidung über Annahmen, Auftrag zur Nachbesserung, Ablehnung

**Abnahme stellt immer einen Kompromiss zwischen optimalem (also fehlerfreiem) und akzeptablem Ergebnis dar.**

# Abnahme größerer Systeme

Mehrstufiges Abnahmeverfahren:

## Werkabnahme

- Abnahme in einer speziellen werksseitig erstellten Testumgebung
- sinnvoll nur, wenn Installation weit entfernt erfolgen soll oder wenn die Installation den Betriebsablauf nachhaltig stört

## Abnahme in der realen Umgebung

- unverzichtbar, evtl. sind Maßnahmen zur Sicherung des Betriebsablaufs zu treffen
- Durchführung auch der Tests, auf die in der Werksabnahme verzichtet werden musste, weil deren Implementierung in der Testumgebung zu aufwändig gewesen wäre

## Betriebsabnahme

- Versuchsbetrieb in der Garantiephase mit aufwändigerer Protokollierung des Betriebs
  - Aufzeichnung aller Fehler, Ergänzung der Testreihe
  - Wiederholung der modifizierten Tests mit dem verbesserten System vor der endgültigen Inbetriebnahme

# Abnahme von Produkten für den anonymen Markt

Auftraggeber und Nutzer sind verschieden.

- Interner Auftraggeber (Marketingabteilung, Produktmanager) nimmt das Produkt ab
- Systeme werden in der Regel einem Alpha- und Beta-Test unterzogen
  - Prüfziele Fehlertoleranz, Benutzbarkeit, Konfiguration und Interoperabilität lassen sich nur schwer durch den internen Auftraggeber testen
  - aufgetretene Fehler werden protokolliert und beseitigt
- **Alpha-Test:** System wird in der Zielumgebung des Herstellers durch Anwender erprobt.
- **Beta-Test:** System wird ausgewählten Pilot-Kunden in deren eigener Umgebung zur Probenutzung zur Verfügung gestellt.
  - nach umfangreichen Fehlerkorrekturen auch Beta2-Phase möglich
  - Pilotkunden erhalten beim späteren Kauf meist einen Rabatt

# Produktzertifikate

Die Qualität eines Software-Systems als *Produkt* ist für den Endkunden von eigenständigem Interesse. Hersteller sind damit an Zertifikaten für ihre Softwareprodukte durch neutrale Instanzen interessiert, die gewöhnlich durch Hersteller übergreifende domänen-spezifische *Gütegemeinschaften* als branchenspezifische Form der Selbstorganisation autorisiert sind.

- Erstmals 1985: Richtlinie der *Gütegemeinschaft Software* (als übergreifende Instanz von Software-Anbietern) zur einheitlichen Prüfung von Software-Produkten
  - Reine Produktnorm, keine Aussage über den Entwicklungsprozess
  - Qualitätsanforderungen beziehen sich auf Produktbeschreibung zu Informationen des Kunden vor dem Kauf, Dokumentation, Programme und Daten
  - Nicht berücksichtigt werden unterstützende Dienstleistungen

# Produktzertifikate

- Überarbeitung als DIN 66285 sowie ISO 12119:1994 „Information technology – Software packages – Quality requirements and testing“
  - Applicable to software packages. Establishes requirements for software packages and instructions on how to test a software package against these requirements. Deals only with software packages as offered and delivered. Does not deal with their production process. The quality system of a supplier is outside the scope of this standard. (<http://www.iso.org>)
- ISO 25051:2006 „Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing“
  - Defines quality requirements for COTS software products and includes the product description, the user documentation, and the software contained on a computer sensible media.
  - The quality requirements, functionality, reliability, usability, efficiency, maintainability, portability, and quality in use are consistent with the definitions of SquaRE. (<http://www.iso.org>)

# Produktzertifikate

- Übergang zu einer integrierten Sicht von Software und Systemen. Neuer Begriff *Ready to Use Software Product* (RUSP)
  - ISO 25051:2014: Software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing“
- ... deals only with providing the user with confidence that the RUSP will perform as offered and delivered. It does not deal with the production realization (including activities and intermediate products, e.g. specifications). The quality system of a supplier is outside the scope (<http://www.iso.org>)

# Produktzertifikate

Aufgegangen in eine ganze Serie von Normen, die durch die ISO 25000:2014 „Systems and Software Engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE“ genauer beschrieben wird

- ISO/IEC 25000:2014 provides guidance for the use of the new series of International Standards named Systems and Software Quality Requirements and Evaluation (SQuaRE).
- The purpose of ISO/IEC 25000:2014 is to provide a general overview of SQuaRE contents, common reference models and definitions, as well as the relationship among the documents, allowing users of the Guide a good understanding of those series of standards, according to their purpose of use.
- It also contains an explanation of the transition process between the old ISO/IEC 9126 and the ISO/IEC 14598 series and SQuaRE. (<http://www.iso.org>)
- IEC = International Electrotechnical Commission, ISO/IEC JTC 001 - einziges gemeinsames Normungsgremium innerhalb der ISO (JTC = Joint Technical Committee)

## Beispiel

(Quelle: <http://www.tuev-sued.de> „Prüfungen Softwarequalität“)

Wir prüfen Ihr Softwareprodukt (Internet, Client, Server, SaaS, ...) auf erfolgreiche Umsetzung der Anforderungen zur Funktionalität, Usability und Datensicherheit.

Die Grundlagen unserer Prüfungen stellen die gesetzlichen Vorschriften und die Vorgaben aus den relevanten anerkannten Standards dar.

- **Funktionalitäts-Prüfung:** Es werden die Anforderungen der ISO 25051: "Software-Engineering - Softwareproduktbewertung - Qualitätsanforderungen an kommerzielle serienmäßig produzierte Softwareprodukte (COTS) und Prüfanweisungen" angewendet.
- **Usability-Prüfung:** Für die Prüfung in ergonomischer Hinsicht wird die DIN EN ISO 9241 Ergonomie der Mensch-System-Interaktion - Teil 110: "Grundsätze der Dialoggestaltung" zu Grunde gelegt.
- **Datensicherheits-Prüfung:** Es werden die gesetzlichen Anforderungen (Bundesdatenschutzgesetz, Telemediengesetz), die BSI-Empfehlung von Standardsicherheitsmaßnahmen (IT-Grundschutz) sowie in der Praxis gängige Standards zu Grunde gelegt.

## Qualitätssicherung in der Phase Betrieb und Wartung

### Bug Tracking

- Einsatz automatisierter webgestützter Systeme wie Bugzilla

### Arbeit mit Power Usern

- Rolle von Alpha- und Beta-Test-Phasen

### Rolle der Qualität der Daten

### ISO 9000 sieht Pflicht zur Nachweisführung vor.

- Wartungsplan, Wartungsaufzeichnungen und -berichte
- Konfigurationsmanagementplan

1. Einführung
2. Qualitätssicherung nach ISO 9000
3. CMM und CMMI
4. ISO 15504
5. Business Engineering und Zusammenfassung

# Qualität auf Projektebene und auf Unternehmensebene

früher: Konzentration auf Qualität des Softwareprodukts

- konstruktive und analytische QS-Maßnahmen als Teil der Prozessplanung

heute: Zusammenhang zwischen Qualität des Produkts und Qualität der Entwicklungsprozesse wird stärker berücksichtigt

- Betonung eines eigenständigen Qualitätsaspekts der Entwicklungsprozesse sowohl auf Projektebene als auch auf der Ebene des gesamten Unternehmens
- evolutionäre Ansätze (schrittweise Verbesserung der Prozessqualität)
  - Darstellung der Qualitätsfähigkeit nach ISO 9000
  - Entwicklung der Reife der Prozesse im Unternehmen (CMM, ISO 15504)
  - Entwicklung einer qualitätsorientierten Unternehmenskultur: totales Qualitätsmanagement (TQM)
- Business Engineering (Qualität durch Prozess-Konstruktion)

# Das ISO 9000-Normenwerk

Allgemeiner QS-Standard (nicht speziell für SW-Entwicklung)

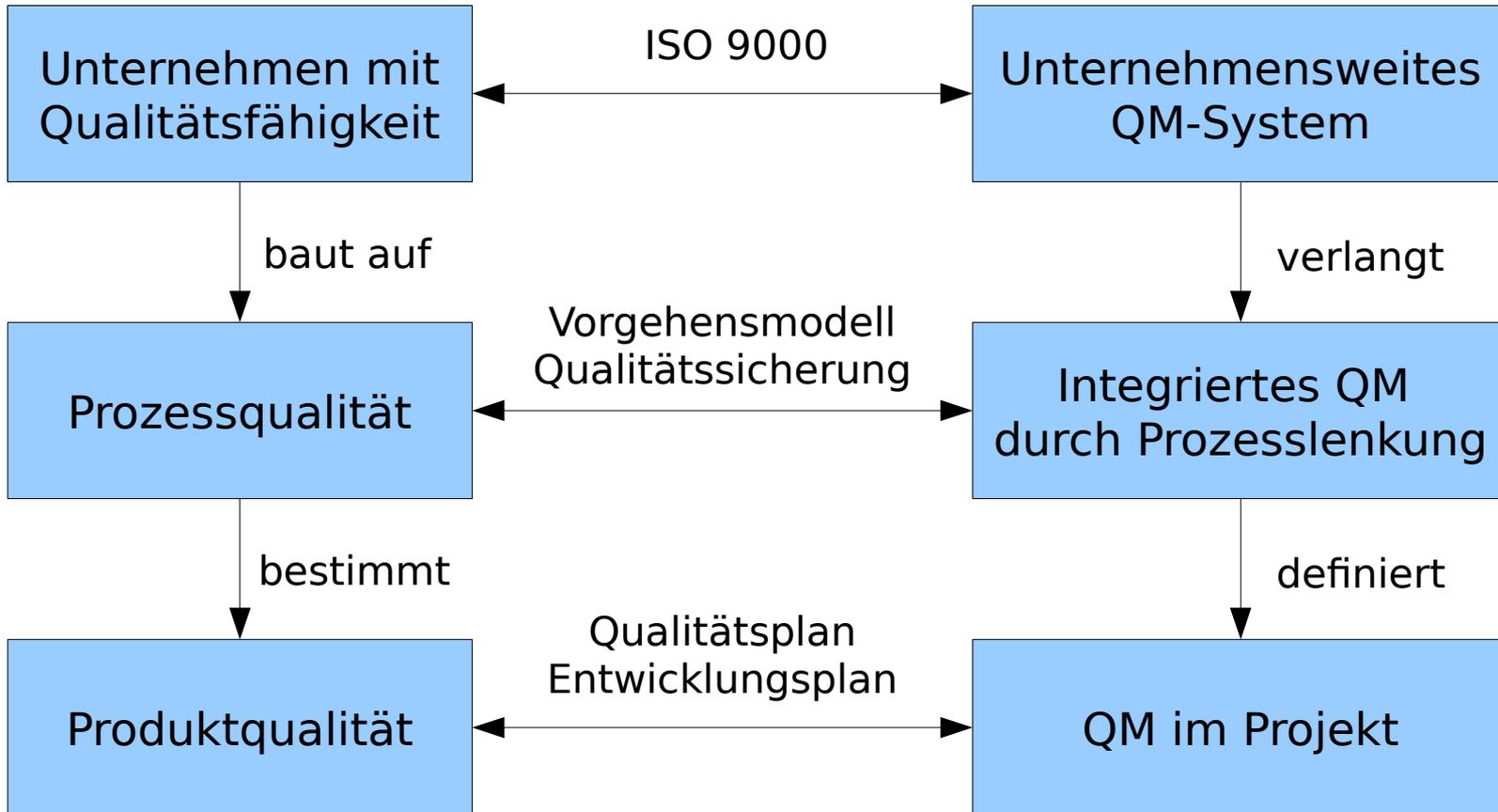
- Europäisches Normenwerk
- Qualität der Zulieferteile wird wesentlich durch die Qualität des Herstellungsprozesses bestimmt
- Normenwerk zum Nachweis für Prozessqualität zur Erstellung materieller und immaterieller Produkte
- Rahmenwerk mit viel Spielraum für Unternehmensstrategie und konkrete Ziele der Geschäftsführung
  - Minimalanforderung an ein QM-System nach ISO-9000: vollständig, dokumentiert, bekannt, überprüfbar, evolutionär

Ziel: Zertifizierung des Qualitätsmanagementsystems des Unternehmens

DIN EN ISO 9000 ff. enthält Mindestanforderungen an den Aufbau und die Ablauforganisation, damit Qualität kein Zufall, sondern das Ergebnis eines beherrschten Prozesses ist.

# 9. QS auf Unternehmensebene

## 2. Qualitätssicherung nach ISO 9000



# Das ISO 9000-Normenwerk

Wesentliche Zielsetzungen und Schwerpunkte des Normenwerks:

- Schaffung einer qualitätsfähigen Aufbau- und Ablauforganisation
- Qualifikation der Mitarbeiter
- Regelung von Zuständigkeiten, Verantwortung und Befugnissen
- Dokumentationspflicht auf allen Ebenen des Unternehmens
- Beherrschung von Risiken und der Wirtschaftlichkeit
- Vorbeugende Maßnahmen zur Vermeidung von Qualitätsproblemen

Für SW-Unternehmen ist das Festschreiben solcher „Spielregeln“ interessant, weil Informatiker dazu neigen, sich als technische Spezialisten zu verhalten und nicht als Problemlöser in gemischten Arbeitsgruppen mit anderen Unternehmensbereichen.

# Das ISO 9000-Normenwerk

ISO 9000: Definiert Grundlagen und Begriffe von QM-Systemen

- ISO 9000:2000 ging weg von stärkerer Fixierung auf Standardprozesse hin zu einer flexibleren prozessorientierten Struktur
  - Reagiert damit auf die Durchsetzung prozessorientierter Ansätze der Betriebsführung
  - Vorher: starke Anlehnung an das Raster der produzierenden Industrie von Entwicklung über Montage bis zum Kundendienst, was die Übertragung in Dienstleistungsbereiche erschwerte
- ISO 9000:2005 nimmt weitere Vereinheitlichung von Begrifflichkeiten vor, ersetzt ISO 8402:1994 "Quality management and quality assurance - Vocabulary"

# Das ISO 9000-Normenwerk

ISO 9001: “Quality management systems – Requirements”. Anforderungen an ein QM-System, mit dem eine Organisation ihre Qualitätsfähigkeit nach außen gegenüber Kunden und allgemeinen behördlichen Anforderungen darstellen kann

- Modellhafte Beschreibung eines gesamten QM-Systems, Basis für spezielle QM-Systeme, letzte Version vom Dez. 2008
- Bildet die Grundlage für Audits nach ISO 19011:2011 “Guidelines for auditing management systems“

ISO 9004: Leitfaden, mit dem die Wirksamkeit und Effizienz eines QM-Systems betrachtet werden kann

- stärker ausgerichtet auf TQM-Ansatz und Qualität als Unternehmenskultur
- Aktuelle Version ISO 9004:2009 „Managing for the sustained success of an organization – A quality management approach“

# Schwerpunkte der ISO 9001 als Rahmenbedingungen des Qualitätsmanagements

## Kundenorientierung

- Kundenbedürfnisse und -erwartungen
  - Stand der Technik
  - gesetzliche und behördliche Rahmenbedingungen
  - Kundenforderungen, Kundenzufriedenheit (Kommunikation, Kundenbefragung, Indikatoren)
- Customer Relationship Management (CRM)

## Konsequente Leitung und Führung

- Topmanagement: Vision, Ziele, Projektunterstützung
- Führung der Projekte
  - strategische Ziele, kritische Erfolgsfaktoren, Risikomanagement
- Führung der Mitarbeiter
  - Rollenverständnis, Motivation, Kommunikation, Konfliktbewältigung

### Einbeziehung der Mitarbeiter

- Verantwortungen, Zuständigkeiten, Befugnisse festlegen
- Fertigkeiten, Erfahrungen, Schulung
- Mitarbeiterzufriedenheit und andere „weiche Faktoren“

### Prozess-, insbesondere Projekt-Management

- Definition, Beschreibung, Ablauf (Vorgehensmodell, Unterstützung)
- Prozesskennzahlen: Jeder Prozess wird in eine Folge von Aktivitäten mit messbarem Input und messbarem Output zerlegt
  - Prozessleistung (Releases/Zeiteinheit)
  - Prozessqualität (Entwurfsänderungen/Zeiteinheit)
  - Prozesskosten (Entwicklungskosten)

### Kontinuierlicher Verbesserungsprozess

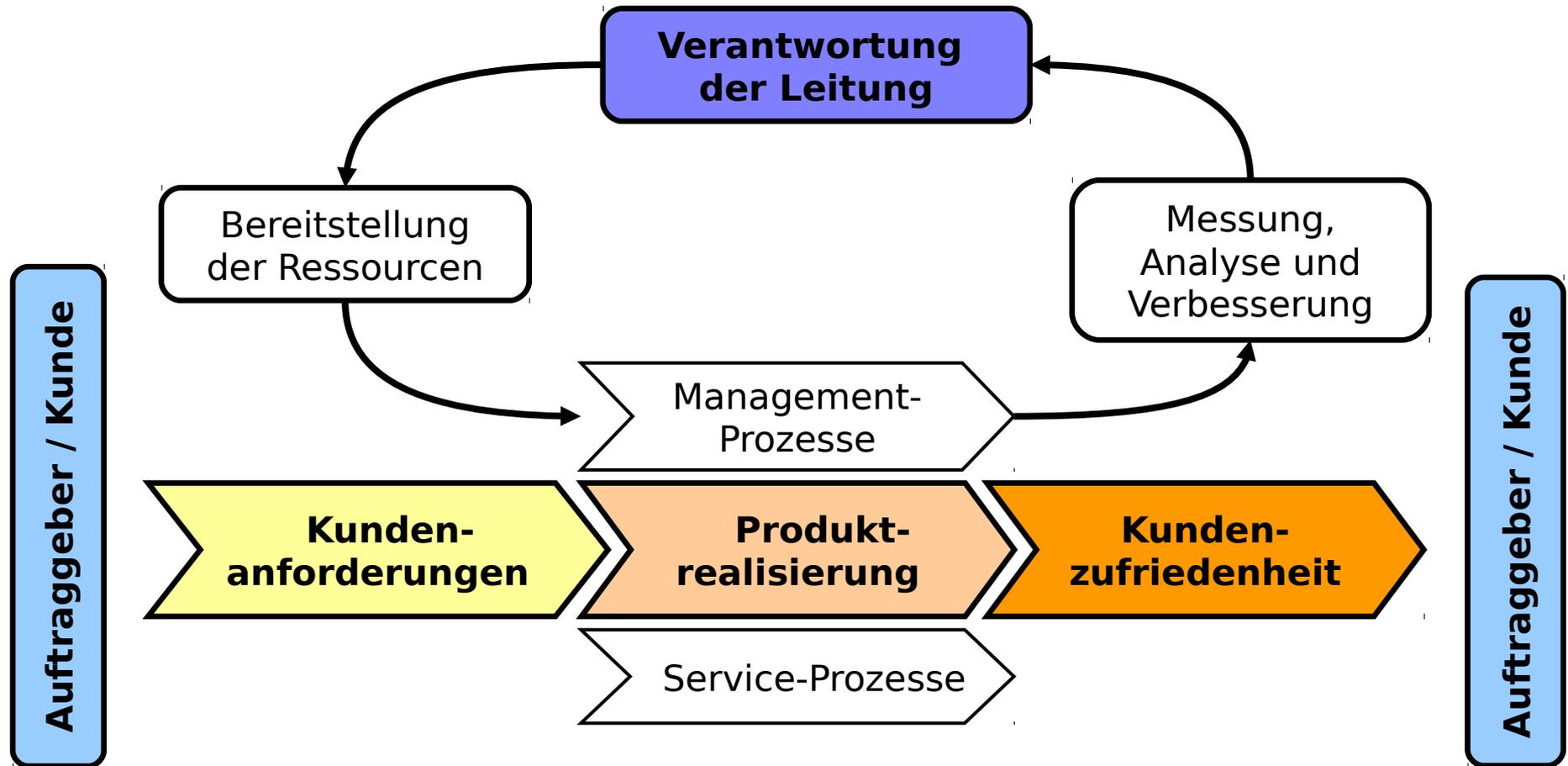
- systemorientiertes Management der Prozesse in ihrer Wechselwirkung
- Entscheidung auf Basis von Fakten
  - Vier-Augen-Prinzip
  - Metriken und Analyse der Messergebnisse
  - Entscheidungsstrategien (bei Sicherheit, bei Risiko, bei Unsicherheit)
- Audits (Begutachtungen)
- Einarbeitung externer (Kunden) und interner (Mitarbeiter) Verbesserungsvorschläge

Die ISO 9001 fordert, dass alle für die Produktion notwendigen Prozesse einschließlich ihrer Wechselwirkungen ermittelt, geplant und angewendet werden.  
Die Qualitätsplanung muss dabei berücksichtigt werden.

# 9. QS auf Unternehmensebene

## 2.1. QM-Systeme nach ISO 9001

### Regelkreis des modifiziertes Prozessmodells der ISO 9001:2008



### ISO 9000 im Softwarebereich

Anwendung der ISO 9001 auf die Spezifik von Unternehmen, die Auftragssoftware herstellen, allerdings ist hier noch viel in Bewegung. Relevante neue Standards der letzten Jahre:

- ISO 90003:2014 „Software engineering – Guidelines for the application of ISO 9001:2000 to computer software“
- ISO 90005:2008 (TR) „Systems engineering – Guidelines for the application of ISO 9001 to system life cycle processes“
- ISO 12207:2008 „Systems and software engineering – Software life cycle processes“
- ISO 15288:2015 „Systems and software engineering – System life cycle processes“

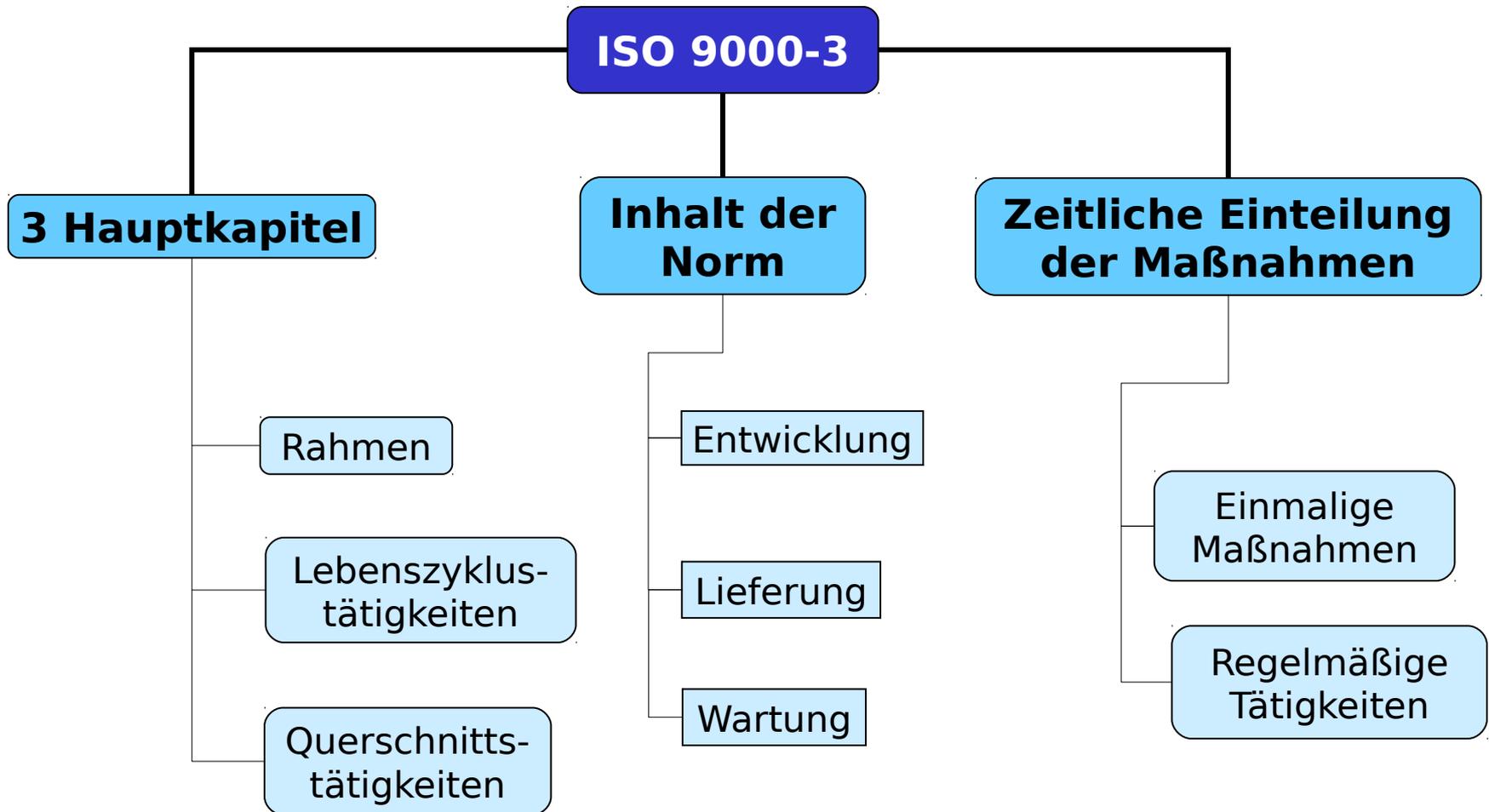
Zusammenwachsen der bisher getrennten Normierungsprozesse für Software-Entwicklung und (komplexeres) System-Design.

### ISO 9000-3 im Softwarebereich

Vorher: ISO 9000-3:1997 “Quality management and quality assurance standards – Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software”. Die Konzepte dieser ISO-Norm sind in der einen oder anderen Form in diesen neueren Standards aufgenommen, deshalb nach wie vor von Interesse.

- **Zielstellung:** Demonstration der Lieferantenfähigkeiten zur Entwicklung, Lieferung und Wartung von Softwareprodukten.
- Unterscheidung von übergreifenden Aktivitäten und Aktivitäten, die einzelnen Phasen des Entwicklungsprozesses zuzuordnen sind:
  - QS-Rahmenwerk, QS-Lebenszyklus-Aktivitäten, QS-Querschnittsaktivitäten
- Qualitätsmanagementsystem nach diesem Normenwerk ist ISO 9001-kompatibel und kann entsprechend **zertifiziert** werden.
  - **Systemzertifikat**, welches die Qualitätsfähigkeit des Unternehmens insgesamt bescheinigt

### Inhalte der ISO 9000-3



### Drei Teile

#### QS Rahmenwerk

Geprägt durch Managementverantwortlichkeiten auf beiden Seiten (Lieferant und Kunde), etablierte Qualitätspolitik, Kommunikation, mit folgenden Elementen:

- Verantwortung der obersten Leitung
- QM-System mit Dokumentation inklusive Qualitätsplan
- Interne Audits des QM-Systems
- Organisation von Korrektivmaßnahmen (Ursachenforschung)

#### QS Lebenszyklus-Aktivitäten

Geprägt durch Aspekte, die sich einzelnen Etappen des SW-Lebenszyklus zuordnen lassen

- Vertragsmanagement (Abnahmekriterien, Problembehandlung)
- Anforderungsspezifikation
- Entwicklungsplanung usw.

### QS Querschnittsaufgaben

Entwicklungsübergreifende Aufgaben, die sich nicht einzelnen Phasen des SW-Lebenszyklus zuordnen lassen

- Konfigurationsmanagement
- Änderungsmanagement
- Lenkung der Dokumente
- Systematisierung und Auswertung der Qualitätsaufzeichnungen (Testprotokolle, Reviewberichte, Datenerfassung über Metriken)
- Werkzeuge und Techniken, Standards, Regelwerke
- Beschaffung, einschließlich der Lieferanten-Beurteilung
- Schulung und Ermittlung von Schulungsbedürfnissen

Die Norm geht von Maßnahmenmix aus, mit dem (1) auf konstruktivem Weg präventiv Qualität geschaffen wird und (2) Qualitätsprobleme erkannt und damit zufriedenstellend umgegangen wird.

### QS Lebenszyklus-Aktivitäten

kein spezielles Vorgehensmodell, aber folgende Annahmen:

- phasenweise Software-Entwicklung,
- die Vorgaben für jede Phase sind festgelegt,
- die geforderten Ergebnisse jeder Phase sind festgelegt und
- die in jeder Phase durchzuführenden Verifizierungsverfahren sind festgelegt

Dokumente, die dazu aufgeführt und spezifiziert werden:

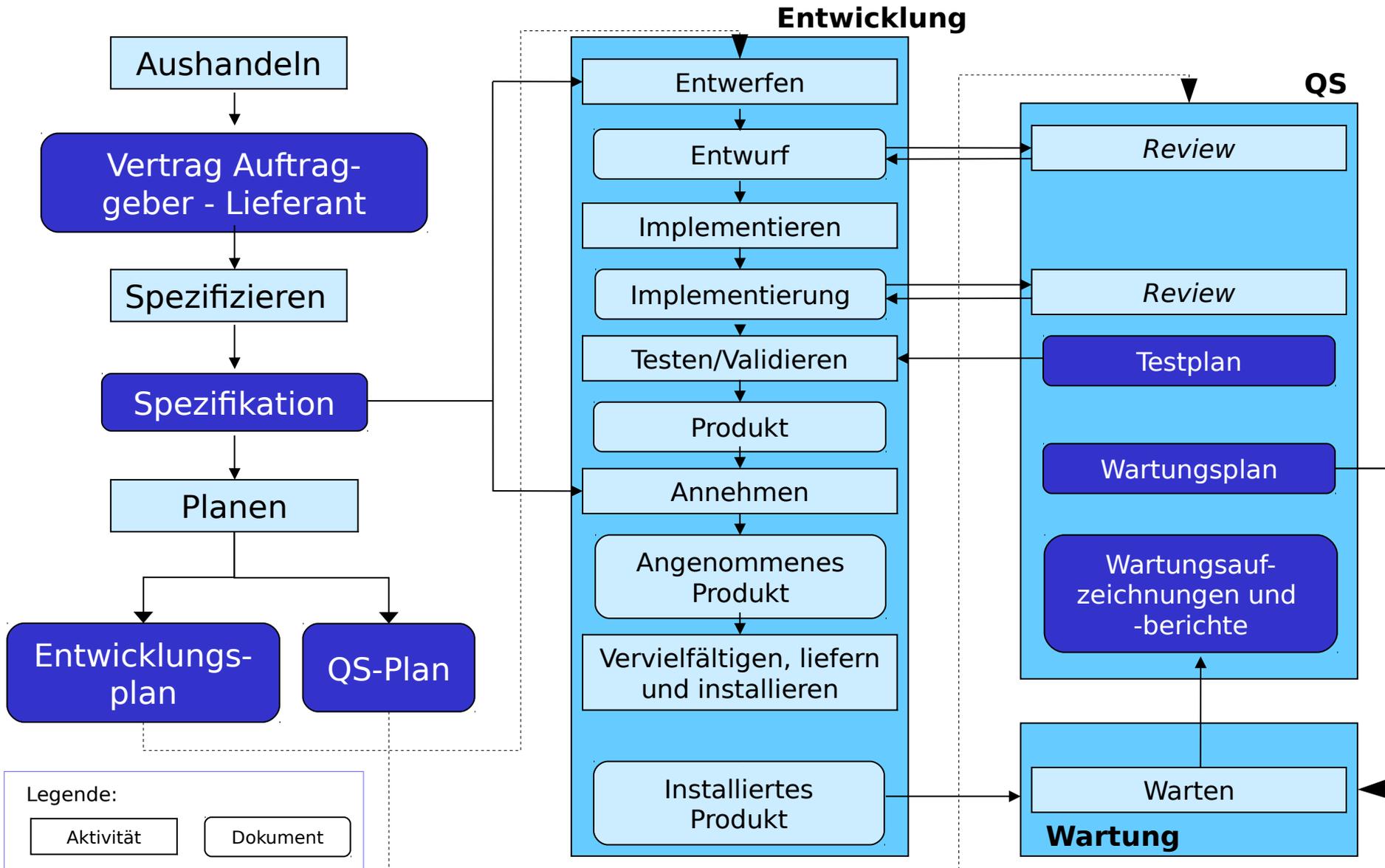
- Vertrag Auftraggeber – Lieferant (qualitätsrelevante Punkte)
  - Annahmekriterien und Auftraggebermitwirkung
  - Änderungen Auftraggeberforderungen während der Entwicklung
  - Behandlung von Problemen nach der Abnahme
  - vom Auftraggeber bereitzustellende Ressourcen
  - anzuwendende Normen und Verfahren

- Spezifikation
  - Vollständiger und eindeutiger Satz von funktionalen Forderungen
  - Leistung, Ausfallsicherheit, Zuverlässigkeit, Datensicherheit, Persönlichkeitsschutz
  - Schnittstellenspezifikation zu anderen SW- und HW-Produkten
- Entwicklungsplan
  - Einbettung des Projekts in andere auf Lieferanten- sowie Auftraggeberseite
  - Planung der Projektmittel einschließlich Teamstruktur, Verantwortlichkeiten, Unterlieferanten ...
  - Entwicklungsphasen (Vorgaben, Ergebnisse, Verifizierung, Problemanalyse)
  - Management (Terminplanung, Fortschrittsüberwachung, Verantwortung, Mittelzuweisung, Schnittstellen zu anderen Gruppen)
  - Entwicklungsmethoden und -werkzeuge
  - Projektplan (Aufgaben, Aufschlüsselung, Zeit- und Mittelplanung, Wechselbeziehungen)
  - Bezug auf die anderen Pläne (QS-Plan, Testplan, Integrationsplan)

- Qualitätssicherungs- und Reviewplan
  - Qualitätsziele und -maße
  - Kriterien für die Vorgaben und Ergebnisse jeder Entwicklungsphase
  - Festlegungen zu Test-, Verifizierungs- und Validierungsmaßnahmen
  - Planung dieser Maßnahmen (Termin, Mittel, Genehmigungsinstanzen)
  - Festlegung von Verantwortlichkeiten
- Testplan
  - Pläne für Modul-, Integrations-, System- und Abnahmetest
  - Aufbereitung der Testfälle, Testdaten und Testszenarien
  - Testumgebung, Werkzeuge und Test-Software
  - Kriterien für die Vollständigkeit der Tests
  - Überprüfung der Produkt-Dokumentation
  - Personal und Schulungserfordernisse
- Wartungsplan
  - Identifizierung der Fremd- und unterstützenden Leistungen
  - Fixierung der auszuführenden Tätigkeiten und der zu verwendenden Technologien, Methoden und Werkzeuge

# 9. QS auf Unternehmensebene

## 2.2. ISO 9000 im Softwarebereich



### Vorgabe- und Nachweisdokumente

Die Qualitätsfähigkeit wird u.a. durch Vorgabe- und Nachweisdokumente geregelt.

#### **Vorgabedokumente:**

- QS-Handbuch
- Plan (einschl. Zeitplan) für die internen Qualitätsaudits
- QS-Plan auf der Basis des QS-Systems
- Reviewplan pro Projekt bzw. Projektphase
- Testplan mit Testspezifikation und Testverfahren
- Software-Wartungsplan
- Konfigurationsmanagement-Plan

### **Nachweisdokumente:**

- Beschreibung der Qualitätspolitik der Organisationseinheit
- Organigramme und Stellen- bzw. Rollenbeschreibungen
- Festlegung des Qualitätsbeauftragten der obersten Leitungsebene
- Dokumentation der Ergebnisse der internen Qualitätsaudits
- Aufzeichnungen über durchgeführte Vertragsüberprüfungen
- Aufzeichnungen über Korrekturmaßnahmen
- Review- und Testprotokolle
- Wartungsaufzeichnungen und -berichte
- Dokumentation der Aktivitäten laut Konfigurationsmanagementplan
- Aufzeichnungen über Schulungen und Erfahrungsmanagement

### Freiheiten der Norm

- Kein spezielles Vorgehensmodell, aber folgende Annahmen:
  - phasenweise Software-Entwicklung,
  - die Vorgaben für jede Phase sind festgelegt,
  - die geforderten Ergebnisse jeder Phase sind festgelegt und
  - die in jeder Phase durchzuführenden Verifizierungsverfahren sind festgelegt
- Prüfmethoden, -verfahren und -werkzeuge können unternehmensspezifisch gewählt werden
- Der Umfang der Qualitätsmaßnahmen für konkrete Projekte kann unternehmensspezifisch variieren
- Ausprägung von Konfigurationsmanagement und Schulungen kann unternehmensspezifisch festgelegt werden
- Projektorganisation und Rollenverteilung kann unternehmensspezifisch festgelegt werden.