

Agenda: Softwarelife-Cycle (1)

- **Software-Erstellungsprozess**
- **Phasenmodell**
- **Entwicklungstätigkeiten**
- **Entwicklungsdaten /Softwaretest /Softwaretest**
- **Softwarequalitätsprüfung, Prüfautomatisierung**
- **Software-Review / Softwarequalitätsbeurteilung**
- **QM-Doku / QM-Berichtserstattung**
- **Software-Assessments und Prozeßverbesserung**
- **Review**
- **Abnahme / Abnahmeverfahren / Endabnahme**
- **Abrechnung / Gewährleistung, Haftung**

Agenda: Softwarelife-Cycle (2)

- **Softwarewartung**
- **Wartungsverträge / Wartungsphase (Life-Cycle)**
- **Maintenance Engineering**
- **System Change Request**
- **Trennung von Wartung und Entwicklung ?**
- **Konfigurationsmanagement**
- **Versionsverwaltung**
- **Releaseplanung**
- **Beispiel eines Standard Verfahrens zur Konfektionierung von Produkt Updates**
- **SLA-Management**

Software-Erstellungsprozess

- Softwareentwicklung ist Umsetzung eines Anforderungskatalogs
 - Umsetzung in Algorithmen, die in Form eines normalisierten Textes (Quelltext) aufgeschrieben werden
 - Text wird mittels eines Übersetzers (Compiler) in Befehle übersetzt oder durch einen Interpreter direkt abgearbeitet
- Programmierung ist abstrakter Prozess
 - oft umfangreich und komplex
 - bedarf viel Erfahrung für die Zerlegung der Anforderungen in Aufgaben und der Umsetzung in Quelltexten.

Software-Erstellungsprozess

- Anforderungskatalog liegt bei ausgeschriebenen Projekten in Form eines Lasten- und Pflichtenheftes vor
- Dokumente enthalten eine verbale Beschreibung der Probleme, die durch das Softwareprodukt gelöst werden sollen
- Produktfertigung erstellt ein Leistungsprofil, das den Leistungsumfang beschreibt und abgrenzt
- Leistungsprofil muss ggf. für die Anpassung an eine geänderte Marktlage modifiziert werden

Software-Erstellungsprozess

- Man unterteilt den Lebenszyklus eines Programms bei dem Wasserfallmodell in folgende Phasen:
 - Lastenheft
 - Pflichtenheft
 - Analyse
 - Grobentwurf, Feinentwurf
 - Kodieren
 - Test
 - Integration
 - Wartung und Pflege

Software-Erstellungsprozess

- Diese Phasen werden nacheinander von oben nach unten durchlaufen
- Treten in einer Phase jedoch Fehler oder Unstimmigkeiten auf, so erfolgt Rücksprung in die fehlerhafte Phase
- Durchführung der Korrektur
- Anschließend Durchlauf alle nachfolgenden Phasen mit den Korrekturmaßnahmen

Software-Erstellungsprozeß

- Analyse
 - Die Entwicklung beginnt mit der Analyse der Aufgabe auf Basis des Lasten-, Pflichtenhefts
- Designphase
 - Auf Analyse folgt Designphase. Dazu wird zunächst ein Grobentwurf erstellt,
- Feindesign
 - Grobentwurf wird verfeinert und spezifiziert, bis detaillierte Aufgabebeschreibung
- Codierungsprozess
 - Codierung der einzelnen Aufgaben

Software-Erstellungsprozeß

- Für das Design können folgende graphische Methoden und Werkzeuge zur besseren Anschaulichkeit und Erfassbarkeit eingesetzt werden:
 - Blockdiagramm
 - Entity-Relationship Diagramm
 - Statusdiagramm
 - Fluß- bzw. Struktogramm
 - Klassenbaum
 - Funktionsbaum
 - Zeigerdiagramm (Zeiger auf Strukturen und deren Lage im Speicher)

Software-Erstellungsprozeß

- Designphase
 - Aufbau der Software spezifiziert
 - Modularisierung durchgeführt
 - Schnittstellendefinition zwischen Programmmodulen
 - Modularisierung erlaubt parallele Arbeit von Entwickler-gruppen (saubere Definition der Modulschnittstellen)
 - Festlegung von Kommunikationsprotokolle und Daten-strukturen
- Codierung
 - startet, wenn prinzipieller Aufbau der Software feststeht
 - Design der einzelne Module abgeschlossen ist

„Fehlerbeseitigung wird immer teurer, je später ein Fehler entdeckt wird“

Das Phasenmodell

- Grundlage für den Erfolg ist Aufspaltung der Komplexität
 - Projekt wird in verschiedene Phasen aufgeteilt, die ein festes Ziel und einen definierten Termin haben
 - Ende einer Phase wird als Meilenstein bezeichnet
 - An Meilensteine richtet sich Kontrolle des Projektverlaufs aus
 - Meilensteinberichte sollten wahrheitsgemäß den Projektstatus wiedergeben; Unstimmigkeiten und Abweichungen sind aufzuzeigen

Das Phasenmodell

- Problemfelder
 - Frühzeitig Unstimmigkeiten / Abweichungen kommunizieren
 - Zunächst scheinbar kleine Probleme frühstmöglich beseitigen
 - Personalveränderungen und –Aufstockungen projektverträglich gestalten
 - Bei komplexen Abhängigkeiten Netzplan einsetzen
 - Klarheit über Abhängigkeiten der einzelnen Phasen, Aufgaben
 - Prioritäten, Reihenfolge
 - Kritischer Pfad
 - Notwendige Ressourcen

Entwicklungstätigkeiten

- Zu Projektbeginn ist Kontaktperson beim Auftraggeber, bzw. Fachbereich zu bestimmen (Produktmanager / Fachverantwortliche)
 - Kompetenzen:
 - Direkter Zugang zu Entscheidungsträgern
 - Fachliche Kompetenz und Weisungsbefugnis
 - Befugnis, Änderungen im Anforderungskatalog aufzunehmen
- Kurze Wege zwischen Entwicklungsteam und Endanwendern herstellen
- Permanente Abstimmung der Projektergebnisse sicherstellen

Entwicklungstätigkeiten

- Schulung der Projektmitarbeiter auf neue Entwicklungswerkzeuge oder Methoden
- Unverzögliche Einleitung geeigneter Maßnahmen zur Machbarkeitsprüfung
 - Simulation oder Prototypenbau
 - Testen der technischen Machbarkeit
 - Testen der Funktionstüchtigkeit
- Im Grobdesigns erste Überprüfung möglicher Problemstellen zur Verifizierung der Randbedingungen
 - Antwortzeiten
 - Durchlaufzeiten

Entwicklungstätigkeiten

- Erkenntnisse des Entwicklungsteam in Knowhow-DB dokumentieren
 - Erfolge
 - Probleme und deren Bewältigung
 - Misserfolge
- Einforderung von ordentlicher Dokumentation in allen Phasen des Projektfortschrittes
 - Neuer Mitarbeiter kann sich einarbeiten
 - 'Knowhow' geht nicht verloren

Übersicht Entwicklungsdokumente

- Bei Eintritt in die Entwicklungsphase sollten spätestens jetzt folgende Planungsdokumente erstellt werden:
 - Produkt- und Leistungsstrukturplan
 - Projektphasenplan (enthält Termine, Meilensteine, Netzplan)
 - Personalplan
 - Betriebsmittelplan
 - QM-Plan
- Die Dokumente werden während der weiteren Projektphasen verfeinert und angepasst

Übersicht Entwicklungsdokumente

- Im Laufe der Realisierung werden in der Entwicklungsphase folgende Dokumentationen erstellt:
 - Grobdesign
 - Feindesign
 - Spezifikation
 - Programmdokumentation
 - Quellcode
 - Testprotokolle
 - Handbücher (z.B. Installationshandbuch, Benutzerhandbuch etc.)

Übersicht Entwicklungsdokumente

- Prinzipiell unterscheidet man zwei Dokumententypen
 - Benutzerdokumentation
 - Systemdokumentation
- **Wichtig:**
 - **Ein Produkt gilt rein rechtlich nur dann als vollständig, wenn die Benutzerdokumentation mitgeliefert wird**
 - **Der Auftraggeber kann also bei fehlender Dokumentation die Abnahme verweigern**
- Systemdokumentation
 - Festlegung des Umfangs
 - Festlegung des Übergabetermins

Übersicht Entwicklungsdokumente

- Nicht geforderten Dokumente sind mit gleicher Gründlichkeit zu erstellen
 - notwendig im Gesamtkontext
 - ermöglicht Wartung
 - Dokumentation für evtl. rechtliche Streitigkeiten mit AG
- Richtlinien für Formulare, Dokumente
 - Vorgabe Layout je Dokument
 - Vorgabe von Struktur der Kapitel
 - Zentrale Bereitstellung der Layouts

Übersicht Entwicklungsdokumente

- Formatierung von Quellcode beinhaltet einheitlich
 - Header,
 - Strukturierung (Klammersetzung etc.)
 - Kommentare
- Kunde prüft selten den Code, führt aber vielfach optischen Check durch, bei dem er formal nach Headern und Kommentaren sucht
- Weitere unterstützende Hilfsmittel sind Namenskonventionen zur Quellcodeerstellung und -pflege
 - für Programme, Module und Funktionen
 - Prefixe vor Variablen und Funktionen, aus denen der Variablentyp oder Deklarationsstelle hervorgehen

Softwaretest

- Softwaretest soll die Einhaltung des spezifizierten und geforderten Umfangs eines Softwareproduktes nachweisen
- Qualität des Softwaretests muss ständig überwacht und verbessert werden
- Bei Testmaßnahmen gewonnene Informationen und Erkenntnisse fließen als Korrekturmaßnahmen in den Testprozess

Softwaretest

- Neue Softwareprodukte erfordern auch neue Methoden der Qualitätssicherung
 - Allein aus diesem Grunde ist regelmäßige die kritische Bewertung der angewandten Testmaßnahmen notwendig
- Effektive Softwarequalitätssicherung bedeutet auch Einbindung der Benutzer:
 - Management und Mitarbeiter der Fachabteilungen zu sensibilisiert und motiviert (Mehrarbeit!)
 - Einweisungen und Schulungen sind im ausreichenden Umfang durchgeführt worden

Testplanung

- Festlegung der Qualitätsmerkmalen durch :
 - Auswählen
 - Klassifizieren
 - Gewichten
- Festlegen der Qualitätsanforderungen unter Berücksichtigung der Realisierungsmöglichkeiten
- Viele Prüfungen werden durch konstruktive Vorarbeiten der Softwareentwicklung erst möglich
 - Prüfbarkeit muss eingeplant und implementiert werden
 - Testschnittstellen müssen definiert werden

Testplanung

- Einsatz automatisierter Prüfung
 - Eingabemasken
 - Ausgabemasken
 - Listoutput
 - Batchläufe
 - Import-, Exportschnittstellen
- Testverfahren sollten wiederholbar sein
- Qualitätssicherungs-Maßnahmen begleiten gesamten Erstellungsprozesses von Software

Testorganisation

- Testorganisation ist im Projekt mit vorzusehen
- Trennung von Entwicklungs- und Testteam
- Tests sind bis zur Freigabe vom jeweiligen Entwicklerteam durchzuführen.
- Testverantwortlichkeiten innerhalb der Gruppe sind eindeutig festzulegen
- Testteam ist nach Freigabe durch Entwicklerteam für Durchführung entsprechender Test nach den Softwarequalitäts-Sicherungsverfahren zuständig
- Die unabhängige Prüfung des Entwicklungsergebnisses durch die Softwarequalitätssicherung entbindet Entwickler nicht von der Verpflichtung der Selbstprüfung.

Qualitätslenkung

- Qualitätslenkung ist in Projektorganisation als möglichst unabhängige Funktion vorzusehen
- Ziel der Qualitätslenkung ist Erfüllung der Qualitätsanforderungen
- Qualitätslenkung umfaßt Überwachung und Korrektur der Softwareprogramme, -Module, Schnittstellen
- Koordiniert und überwacht Vorgehensweisen zur Fehlerverwaltung und -behebung
- Verfahren sind festzuschreiben wie Fehlermeldungen innerhalb des Projektes behandelt und an die betreffende Stelle weitergeleitet werden

Softwarequalitätsprüfung

- Softwarequalitätsprüfung umschreibt alle Tätigkeiten zur Feststellung geforderter Funktionalität und Fehlerfreiheit für:
 - Softwareprogramme
 - Modul
 - Funktion
 - Testdokumentation
 - Systemdokumentation
 - Anwenderdokumentation
- Teststrategien entscheiden nicht zuletzt über die Qualität des auszuliefernden Softwareprodukts

Prüfumfang und Zielsetzung einer Prüfung

- Ziel der Prüfung:
 - Möglichst alle Fehler zu entdecken
 - Fehlerursachen zu finden
 - Fehlerverhütungsmaßnahmen erkennen
 - Qualität des Softwareprodukts festzustellen
- Einsatz geeignete Werkzeuge
- Testdurchläufe durch Verzweigungen müssen kenntlich gemacht werden, um beim Testabschluss eine Aussage über die Testtiefe treffen zu können
- Softwaremodule müssen einzeln testbar sein; Hierdurch sind automatische Tests von Input und Output möglich (Whitebox-Test)

Prüfmethoden

- Einsatz von Checklisten und Prüfwerkzeugen
 - Checklisten bewirkt einen gleichbleibenden, wieder-holbaren Ablauf der Prüfung
 - bedeutet Unabhängigkeit vom Bearbeiter
 - Prüfwerkzeugen ermöglicht eine personenunabhängig Durchführung der Prüfung
 - Aufwand für Softwareprüfung sinkt (Standardisierung)
- Ausarbeitung von Bedienungsabläufen, Testplan
 - Prüfplan für Bedienungsabläufe erleichtert Erstellung von projektspezifischen Testplänen
 - Projektspezifische Tests sind allgemein vollständiger, Testergebnisse sind leichter auswertbar
 - Der Prüfplan beinhaltet Festlegung von Prüfmerkmalen und vorgegebenen Werten/ Grenzwerte mit zu erwartenden Ergebnissen

Prüfmethoden

- Ausarbeitung von Bedienungsabläufen / Testplan
- Bereitstellung eines Prüfplanes für Bedienungsabläufe erleichtert die Erstellung von projekt-spezifischen Prüfplänen
 - Ergebnisse projektspezifischen Tests werden allgemein vollständiger
 - Prüfergebnisse lassen sich leichter auswerten
- Prüfplan beinhaltet Festlegung von Prüfmerkmalen
 - mit vorgegebenen Werten / Grenzwerte
 - zu erwartenden Ergebnissen
 - Festlegung wesentlicher Testfälle und Testdaten

Prüfmethoden

- Erfassung von Fehlermeldungen und Kundenreklamationen
- Erfassung und Dokumentation von Fehlern
- Auswertung und Verdichtung
 - nach Fehlerarten
 - nach Fehlerort
 - nach Fehlerursache
- Fehlerbeseitigung projektspezifisch freigegeben und dokumentieren

Prüfmethoden

- Analyse der Fehlerdaten aus Einzelprojekte
- Ableitung von Korrekturmaßnahmen
- Für Erfassung von Fehlern sind geeignete Hilfsmittel festzulegen, z.B. Formulare mit klassifizierender Fehlerbenennung.
- Allgemeingültige Regelungen zu treffen

* Quelle: AkSeQM

Prüfautomatisierung

Die Testentwicklung von automatisierten Tests erfordert einen deutlich höheren Qualifizierungsgrad der Tester sowie einen höheren Aufwand durch die Programmerstellung.

Automatisierte Tests sind nur dann sinnvoll, wenn das Softwareprodukt über einen längeren Zeitraum gepflegt wird oder eine hohe Testtiefe in kurzer Testzeit erreicht werden muss.

Der wesentliche Vorteil automatisierter Prüfabläufe gegenüber den manuell durchgeführten Softwaretest liegt in der einfachen Wiederholbarkeit und der Möglichkeit zur Einführung von Regressionstest.

* Quelle: AkSeQM

Prüfautomatisierung

Die Schwierigkeit bei der Einführung von automatischen Testabläufen liegt in der Regel in der Gewinnung von eindeutigen Testdaten zur Verifizierung der Testergebnisse.

Meistens erfordert dies die Festlegung von Testschnittstellen schon während der Planungsphase des Softwareprojekts, wobei die Einschaltung und Mitwirkung der Entwickler und Tester hierbei schon zu einem sehr frühen Zeitpunkt der Projektplanung gefordert werden muss.

* Quelle: AkSeQM

Prüfautomatisierung

Die mangelnde Bereitschaft bei den Auftraggebern für den Mehraufwand der zu erstellenden Testsoftware die notwendige Zeit und Kosten vorzusehen, führt oft dazu, dass auf die Testautomatisierung verzichtet wird.

Es wird dabei grundsätzlich ignoriert, dass der Softwaretest in der Regel mehrfach durchgeführt werden muss, da Änderungen der Software bis kurz vor Auslieferung zum Kunden üblich sind.

Manuelle Tests führen gegen Ende des Projekts dazu, dass nur noch die geänderten Teile kurz angetestet und dann freigegeben werden. Automatische Tests sind hingegen ermüdungsfrei und können sogar während der Nacht- oder Feiertagsstunden durchgeführt werden.

* Quelle: AkSeQM

Prüfautomatisierung

Die Testtiefe kann durch geeignete Programmierung deutlich erhöht werden, indem auf die Generierung von Zufallszahlen in abgesicherten Grenzen zurückgegriffen wird. Es muss dabei aber darauf geachtet werden, dass die errechneten Zufallszahlen reproduzierbar sind. Dies ist durch die Initialisierung des Zufallszahlengenerators mit einer Konstanten möglich, da sonst Regressionstest nicht durchführbar sind.

Der Mehraufwand für automatische Tests amortisiert sich oft schon beim 2. oder 3. Regressionstest eines Softwareprodukts. Die höhere Testtiefe und die Möglichkeit zu Regressionstests in kurzer Zeit, auch unter Einbeziehung arbeitsfreier Zeit, sollten die Mehraufwendungen bei der Erstellung von automatisierten Tests rechtfertigen.

* Quelle: AkSeQM

Prüfautomatisierung

Ein Vorteil der Testautomatisierung gegenüber der manuellen Methode liegt zumindest darin:

- Die Testdokumentation liegt in Form von Source-code vollständig und immer auf aktuellem Stand vor.
- Eine Nachführung der Testpläne und der damit entstehende Dokumentationsaufwand entfällt.

Eine Frage stellt sich allerdings auch. Was ist mit Tests, die fehlerhaft programmiert sind ?

Im Gegensatz zum Softwareprodukt werden diese Programme nicht ausgeliefert und unterliegen damit auch nicht einer direkten Bewertung durch den Kunden. Für die Erstellung von Testsoftware gelten aber die gleichen Regeln und Qualitätsanforderungen wie für die Erstellung des zu testenden Produktes

* Quelle: AkSeQM

Prüfautomatisierung

Wenn eine Testsoftware fehlerhaft ist, kann diese unter Umständen Fehler im Programmsystem des Softwareprodukts aufdecken oder schlichtweg Fehler anzeigen wo keine sind.

Zumindest erfolgt hier eine zusätzliche Überprüfung der möglichen Fehlerquelle in der zu testenden Software.

Weitaus problematischer ist es, wenn die Testsoftware vertraglich zugesicherte Funktionen als fehlerfrei erkennt, obwohl diese fehlerbehaftet sind.

Softwaretester sind und müssen sich immer bewusst sein, dass die eigene Testsoftware auch fehlerbehaftet sein kann. Dementsprechend sind nicht nur die Testfälle zu überprüfen, die Fehler oder vermeintliche Fehler aufgedeckt haben, sondern es müssen auch die Testfälle überprüft werden, die keine Fehler erkennen.

* Quelle: AkSeQM

Prüfautomatisierung

Die Testentwicklung unterscheidet sich hierbei in keinsten Weise von der Softwareentwicklung für Programmsysteme.

Auch hier müssen die Regeln des Softwareengineerings eingehalten werden und geeignete Maßnahmen zur Sicherstellung der Testqualität ergriffen werden.

Software-Walk-Thru in Form von Codeinspektionen und Reviews bieten hierzu die geeignete Basis.

* Quelle: AkSeQM

Software-Review

Die Zielsetzung dieser Prüfung ist es, Unstimmigkeiten zwischen den Bedürfnissen des Benutzers und der Anforderungsdefinition für das zu erstellende Produkt frühzeitig zu erkennen.

Zur Prüfung wird ein Reviewteam gebildet das sich aus Teilnehmern von Marketing, Service, Entwicklung, Produktplanung und der Softwarequalitätssicherung zusammensetzt.

* Quelle: AkSeQM

Software-Review

In festgesetzten Zeitabständen erfolgt ein Test-Review. Hierbei sind die eingesetzten Teststrategien und deren Abdeckungsmaß zu beurteilen.

Gerade wegen der Komplexität der Softwareprodukte und der damit verbundenen Unmöglichkeit, eine vollständige Prüfung aller möglichen Eingangsbedingungen und aller Ablaufkombinationen durchzuführen, ist die Auswahl der Testfälle besonders wichtig.

Ebenso wird geprüft, ob die spezifizierten Testanweisungen logisch richtig und verständlich sind und genügend Informationen für die Ausführung der Tests beinhaltet.

.

* Quelle: AkSeQM

Software-Review

Eine Checkliste für die Prüfung von Testanweisungen sollte folgende Fragen enthalten:

1. Werden alle Funktionen getestet ?
2. Werden alle Fehlerausgänge getestet ?
3. Werden alle Schnittstellen (Software-/Hardware-) mindestens einmal benutzt ?
4. Werden alle Pfade mindestens einmal durchlaufen ?

* Quelle: AkSeQM

Softwarequalitätsbeurteilung

Hier erfolgt die Überwachung der

- Qualitätssicherungsmaßnahmen,
- Auswertung von Qualitätsdaten
- Entscheidungen über das weitere Vorgehen, wie
 - Freigaben und korrektive Maßnahmen zur Beseitigung von Fehlerursachen (die Behebung eines Fehlers zählt nicht darunter).

* Quelle: AkSeQM

Dokumentation der Prüfung

Für den Nachweis der Qualitätsprüfungen sollte der Prüfberichte folgende Punkte enthalten:

1. Identifikation des Prüfobjektes.
2. Zeitraum der Prüfung.
3. Tag des Erhalts der zu prüfenden Software und der Abschlusstag.
4. Name des bzw. der Prüfer.
5. Angabe der bei der Prüfung angewandten Methoden, Hilfsmittel und Werkzeuge.
6. Vergleich der Prüfergebnisse mit den betreffenden Qualitätsanforderungen.
7. Angaben zur Weiterbehandlung (Freigabe, eingeschränkte Freigabe, Rückweisung).
8. Erfassung der festgestellten Fehler.
9. Bewertung der Prüfergebnisse.

* Quelle: AkSeQM

Qualitätsberichterstattung

Die Qualitätsberichterstattung erfolgt zu festgesetzten Zeitpunkten. Dabei ist festzulegen, welche Informationen in der Berichterstattung enthalten sind und an welchen Verteiler die Berichte gehen.

Werden gleiche Softwareteile in unterschiedlichen Projekten eingesetzt, so werden die Fehleranalysedaten aus allen Projekten zu einer verdichteten Fehleranalyse zusammengeführt. Damit lassen sich projektübergreifende, systematische Fehlerschwerpunkte in den einzelnen Bereichen erkennen und entsprechende Korrekturmaßnahmen einleiten.

Durch eine Analyse des zeitlichen Verlaufs des Fehlergeschehens lassen sich Trends bezüglich der Qualitätslage der Softwareprodukte feststellen, die wiederum zum Einleiten von Korrekturmaßnahmen zur Vermeidung der Fehler genutzt werden können.

* Quelle: AkSeQM

Qualitätsberichterstattung

Die Qualitätsberichterstattung muß für jeden Berichtszeitraum folgende Punkte enthalten:

1. Relative Häufigkeit von Kundenreklamationen und Kundenfragen zur Benutzung
2. Relative Häufigkeit von Systemausfällen bei der Benutzung
3. Relative Häufigkeit von Änderungen an Programmen
4. Welche Programmteile (Module) wurden am meisten geändert ?
5. Analyse der Fehlerursachen und korrektiven Maßnahmen

* Quelle: AkSeQM

Software-Assessments und Prozeßverbesserung

Software-Metriken werden eingeteilt in drei Gruppen:

1. Produktmetriken: Ermittlung und Analyse von Messzahlen für das Software-Produkt
2. Projektmetriken: Unterstützung von Aufwandsprognosen (Zeit, Kosten)
3. Prozessmetriken: Charakterisierung und Quantifizierung der Software-Entwicklungs-Prozesse mit den Zielen der Prozessverbesserung und –optimierung

Für eine Bestandsaufnahme und Bewertung von Software-Entwicklungsprozessen, die in einer Organisation zur Anwendung kommen, gibt es mehrere Assessment-Methoden wie CMM, Bootstrap, SPICE, Trillium, TickIt und SynQuest.

* Quelle: AkSeQM

Reviewtechnik: Definition

- Manuelle, semiformale Prüfmethode, um Stärken und Schwächen eines schriftlichen Dokuments anhand von Referenzunterlagen zu identifizieren und durch den Autor beheben zu lassen

oder:

- Ein Review ist ein mehr oder weniger formalisierter Prozess zur Überprüfung von schriftlichen Dokumenten durch Gutachter, um Stärken und Schwächen des Dokuments festzustellen

Reviewtechnik: Charakteristika (1)

- Ziel der Prüfung
 - Feststellung von Mängeln, Fehlern, Inkonsistenzen und Unvollständigkeiten
 - Feststellung von Verstößen gegen Vorgaben, Richtlinien, Standards, Pläne
 - Formale Planung und Strukturierung der Bewertungsprozesse und formale Abnahme des Prüfobjektes
- Objekte der Prüfung
 - Jeder in sich abgeschlossene, für Menschen lesbare Teil von Software, z.B. ein einzelnes Dokument, ein Quellcode-Modul, ...
- Referenzunterlagen für die Prüfung (Bezugsobjekte)
 - Vorgaben für die Erstellung des Prüfobjektes
 - Relevante Richtlinien und Standards
 - Fragenkataloge die im Review zu beantworten sind

Reviewtechnik: Charakteristika (2)

- Beschreibungsform der Prüf- und Bezugsobjekte
 - Prüfobjekte: informal, z.B. Pflichtenheft
semiformal, z.B. Pseudocode
formal, z.B. Quellcode
 - Bezugsobjekte: informal, z.B. Richtlinien
semiformal, z.B. Pseudocode
formal, z.B. OOA-Modell
- Ergebnisse
 - Review Protokolle
 - Empfehlung über Freigabe
 - überarbeitetes Prüfobjekt

Reviewtechnik: Charakteristika (3)

- Vorgehensweise
 - Menschliche Begutachtung
- Ablauf der Prüfung
 - Statische Prüfung, d.h. in der Reihenfolge der Aufschreibung des Prüfobjekts
- Vollständigkeit der Prüfung
 - Stichprobenartig
- Teilnehmer
 - Review Team, bestehend aus 4 - 7 Personen
(Moderator, Autor, Protokollführer, 2 - 5 Gutachter)

Reviewtechnik: Charakteristika (4)

- Durchführung
 - Eingangsprüfung
 - Optionale Einführungssitzung (Vorstellung von Prüfobjekt und Umfeld)
 - Individuelle Vorbereitung (jeder Gutachter prüft das Prüfobjekt nach den ihm zugeteilten Aspekten)
 - Review Sitzung
 - Bewertung durch die Gutachter
 - max. zwei Stunden Dauer
 - keine Probleme lösen oder beheben
 - Autor ist passiv
 - Überarbeitung des Prüfobjektes (durch den Autor)
 - bei gravierenden Mängeln erneute Review-Sitzung

Reviewtechnik: Charakteristika (5)

- Aufwand
 - 15 - 20% des Erstellungsaufwands des Prüfobjektes
- Nutzen
 - 60 - 70% der Fehler in einem Dokument werden gefunden
 - Reduktion der Fehlerkosten in der Entwicklung von 75% und mehr
 - Nettoeinsparungen für die
 - Entwicklung von ca. 20%
 - Wartung ca. 30%

Abnahme

Allgemeines zur Abnahme

Die Abnahme ist das formelle Ende eines Werkvertrages. Nach einer Abnahme greift die Gewährleistung. Ebenfalls gelten nach einer Abnahme zusätzliche Vereinbarungen, wie:

- die Regelungen zu Wartung und Pflege
- Weiterentwicklung der Anwendersoftware.

In dem Werkvertrag wird ein Gewährleistungszeitraum von mindestens sechst, üblicherweise zwölf Monaten (gesetzlichen Regelungen im gewerblichen Bereich), ausnahmsweise auch 24 Monate festgelegt, in dem der AN alle vom AG nachgewiesenen Fehler kostenfrei für den AG beseitigt.

Wichtig: Der entscheidende Punkt nach der Abnahme ist, dass eine Umkehr der Beweislast eintritt.

* Quelle: AkSeQM

Abnahme

Vor erfolgter Abnahme kann der AG bei der Überprüfung (der vertragsgegenständlichen Lieferungen und Leistungen) Abweichungen von dem vertraglichen (im Leistungsverzeichnis des AN festgelegten) Lieferumfang feststellen und rügen / bemängeln.

Die Juristensprache bezeichnet dies dann als Mangel mit den entsprechenden Rechtsfolgen:

Wandlung oder Minderung

* Quelle: AkSeQM

Abnahme

Der (weitere) technische Begriff "Fehler" und der (engere) juristische Begriff des "Mangels" unterscheiden sich also nur um die Rechtsfolgen.

In einem (vom AG behaupteten) Fall des Mangels hat der AN die Beweislast, das heißt, dass er nachweisen muss, dass der von AG gemeldete sorgfältig dokumentierte Fehler entweder keiner ist, oder nicht in die vereinbarte Fehlerklasse gehört oder ordnungsgemäß beseitigt wurde.

Im Fall der Beseitigung obliegt es dem AG, diesen Sachverhalt zu überprüfen.

* Quelle: AkSeQM

Abnahme

Nach erfolgter Abnahme kann der AG bei der Überprüfung zwar wieder das Vorliegen eines Fehlers behaupten, aber diesmal muss er (an Hand von aussagekräftigen Dokumenten u. U auch Zeugen!) beweisen, dass solche Abweichungen vom vertraglichen Soll existieren oder nicht sachgerecht beseitigt wurden.

Es liegt auf der Hand, dass dies ein sehr konfliktreiches Gebiet ist, denn in der Praxis ist es recht häufig, dass die Leistungsbeschreibungen des Lastenheftes und die des Pflichtenheftes (falls es diese überhaupt gibt!) sich nicht vollständig entsprechen, keine Absprachen über die Art der Fehler und deren Behandlung / Beseitigung existieren und dass (wegen der o. g. Rechtsfolgen !) ein zähes Ringen um die Einteilung in Fehlerklassen usw. eintritt.

* Quelle: AkSeQM

Abnahme

An den bisherigen Darstellungen ist erkennbar, welche Bedeutung klare Absprachen und die Forderung nach Dokumentation aller wesentlichen Ereignisse haben.

Die Abnahmehandlung selbst soll ebenfalls als förmlicher Akt behandelt und dokumentiert sein.

* Quelle: AkSeQM

Abnahme

Daneben gibt es auch den Fall der "**konkludenten Abnahme**".

Dieser Fall liegt vor, wenn der AG die Abnahme verweigert, aber das System (Software) im produktiven Betrieb bereits ganz oder teilweise nutzt.

Auch hier gibt es ein erhebliches Konfliktpotential.

Sinnvoll ist eine Regelung im Vertrag, dass bei einer Abnahmeverweigerung seitens des AG, wobei die Gründe ausschließlich in seinem Einflussbereich liegen, die Abnahme automatisch z.B. nach 21 Tagen nach Erklärung der Abnahmebereitschaft erfolgt ist.

* Quelle: AkSeQM

Abnahmeverfahren

- Grundsätzlich wird nur eine technisch und handwerklich einwandfreie und fehlerfreie Arbeit abgenommen.
- Über das Prozedere der Abnahme wird zum Zeitpunkt des Vertragsabschlusses (spätestens zur Halbzeit der Projektlaufzeit) eine schriftliche Vereinbarung getroffen.
- Für bestimmte Lieferungen und Leistungen können im Vertrag auch Teilabnahmen vereinbart werden. Bei Teilabnahme und Inbetriebnahme einzelner Gewerke beginnt die Gewährleistung zum Zeitpunkt der Abnahme dieser Teilbereiche.
- In jedem Fall ist eine Regelung sinnvoll, ab welchem (fiktiven) Zeitpunkt die Gewährleistung über einen (Rest-) Zeitraum beginnt.

* Quelle: AkSeQM

Fehlerbegriffe

- Ein **Fehler** (error nach ANSI/IEEE) ist die Abweichung zwischen dem berechneten, beobachteten oder gemessenen Wert oder einem Zustand der Betrachtungseinheit und dem entsprechend spezifizierten oder theoretisch richtigen Wert
- Ein **Defekt** (defect) ist eine Abweichung von der festgelegten (erwarteten) Ausprägung eines Merkmals einer Betrachtungseinheit
- Ein **Ausfall** (failure) ist die Beendigung der Fähigkeit der Betrachtungseinheit, die geforderte Funktion auszuführen
- Die **Störung** (fault) ist die Unfähigkeit der Betrachtungseinheit, ihre geforderte Funktion auszuführen

Termingestaltung

- Der Abnahmetermin wird einvernehmlich zwischen dem Auftraggeber und dem Auftragnehmer schriftlich festgelegt.
Der Termin liegt spätestens 10 Werkzeuge nach der Erklärung der Abnahmebereitschaft durch den Auftragnehmer.
- Für den Zeitraum der Abnahme hat der Auftragnehmer das notwendige Personal, die notwendigen Werkzeuge und Messgeräte zur Verfügung zu stellen.
- Stellt sich während der Abnahme heraus, dass wegen Mängel in der Ausführung eine Abnahme nicht erteilt werden kann, so kann vom Auftraggeber eine Wiederholung der Abnahme innerhalb einer angemessenen Frist verlangt werden.

* Quelle: AkSeQM

Termingestaltung

- Die entstehenden zusätzlichen Kosten für die Wiederholungen der Abnahme trägt der Auftragnehmer
- Eine Nachbesserung ist zwei bis max. dreimal zulässig
- Danach ist der nächste Schritt eine Wandlung oder Minderung des Auftrags möglich
- Sollte es aus zwingend betrieblichen Gründen notwendig sein, Teile des Gewerkes seitens des AG vorzeitig in Betrieb zu nehmen, so gilt dieser Gewerketeil als noch nicht entgültig abgenommen

* Quelle: AkSeQM

Endabnahme

Die Endabnahme eines EDV-Systems erfolgt, wenn die vollständige Dokumentation vorliegt und bei der Prüfung der Funktionalität keine gravierenden Mängel vorlagen. Bestandteil dieses Prozedere sind neben den konkreten Prüfverfahren und Terminen auch Fehlerklasseneinteilung, wie z.B.:

- **Klasse A:** Fehler, die wesentliche Funktionen beeinträchtigen und nicht umgangen werden können und daher einer Abnahme entgegenstehen
- **Klasse B:** Fehler, die Funktionen beeinträchtigen, aber mit technisch organisatorischen Maßnahmen umgangen werden können
- **Klasse C:** Fehler, die keine Funktion behindern (und daher notfalls auch gegen Minderung des Preises hingenommen werden können)

* Quelle: AkSeQM

Endabnahme

- Diese Fehlerklassendefinition muss bei Vertragsabschluss (spätestens aber zur Halbzeit) den jeweils vorliegenden Verhältnissen genau angepasst werden und überprüfbare Kriterien, Messverfahren und Messgrößen vorgeben.
- Seit 1994 gibt es eine DIN Norm (DIN 66271), die eine allgemein verbindliche Fehlerdefinition umschreibt und ein Fehlerwirkungsmodell aufstellt. Dieses Modell ist aber nicht unumstritten und bei Anbietern selten bekannt.
- Die endgültige Abnahme (Gesamtabnahme) erfolgt durch den Auftraggeber, am Ende des Zeitraums, wenn alle festgestellten Fehler der (vorher definierten!) **Klasse A** beseitigt wurden.

* Quelle: AkSeQM

Endabnahme

- Vor einer Abnahme einzelner Teilprojekt / Teilgewerke müssen alle dafür definierten und erforderlichen Unterlagen vom AN vollständig vorgelegt werden.
- Alle Dokumente müssen in einer abgestimmten Form geliefert werden. Kriterien können sein, dass die Dokumente in Word und Tabellen in Excel zu übergeben sind.
- Die komplette Dokumentation muss dem AG neben dem Ausdruck auf Papier auch auf Datenträger mit den Formaten obiger Programme inkl. notwendiger Zusatzdateien wie z.B. verwendeter Makros oder Formatvorlagen übergeben werden.

* Quelle: AkSeQM

Endabnahme

Bei der Erstellung der Dokumentation sind folgende Kriterien zu beachten:

- die Unterlagen müssen so gekennzeichnet sein, dass eine Übereinstimmung zwischen Datenträger und Papierformat gewährleistet ist.
- ausführliche Bedienungs- und Wartungsunterlagen in deutscher oder entsprechend vereinbarter Sprache für alle Komponenten.

Die vorgelegten Unterlagen werden vom Auftraggeber über einen vorher definierten Zeitraum (in aller Regel höchstens 21 Arbeitstage) eingehend geprüft.

- Die Lieferung von Handbüchern (Benutzerhandbücher und Systemhandbücher) ist zwingend vorgeschrieben.
- Falls diese nicht vorliegen, handelt es sich nicht um einen Mangel, sondern es ist eine Nichterfüllung des Vertrages (!) mit drastischen Rechtsfolgen.

* Quelle: AkSeQM

Abrechnung

Beispielsweise könnte folgender Abrechnungsmodus bei Auftragsvergabe vereinbart werden:

1. 30 % bei Auftragserteilung - Abdeckung erfolgt über Bankbürgschaft bis zum Termin gemäß Pkt. 2.
2. 30 % bei Installationsbeginn - Lieferung und Lieferbereitschaft
3. 30 % nach Lieferung und Installation aller im Leistungsverzeichnis ausgewiesenen Bestandteile
4. 10 % nach kompletter Abnahme (inklusive Funktionstest).

Anmerkung: Es wird oft auch ein Rückbehalt von 5% während der Gewährleistungspflicht praktiziert.

* Quelle: AkSeQM

Abrechnung

Bei der Schlussrechnung haben folgende Dokumente vorzuliegen:

- Nachweis der Spezifikation durch Messprotokolle und ergänzende Dokumente entsprechend den Positionen in der Leistungsbeschreibung
- Erklärung des Auftragnehmers, woraus hervorgeht, dass die gesamte Anlage gemäß den Richtlinien, gängigen Vorschriften und behördlichen Bestimmungen erstellt wurde und frei von Rechten Dritter ist,
- das rechtsgültig unterschriebene Abnahmeprotokoll (und nicht etwa nur die Bestätigung der erfolgten Lieferung!)

* Quelle: AkSeQM

Gewährleistung

Der Gewährleistungszeitraum beginnt am Tage der "mängelfreien" Gesamtabnahme.

Für die LWL-, Kupferkabel, Verteilerschränke und passive Komponenten gilt ein Gewährleistungszeitraum von 5 Jahren als vereinbart.

- Bei aktiven Komponenten (wie z. B. EDV-Equipment) gilt eine einjährige Gewährleistungsfrist.
- Während des Zeitraums der Gewährleistung wird mit den erforderlichen Arbeiten innerhalb von 8 Stunden ab Anzeige des Mangels begonnen. Dies gilt auch für Wochenenden und Feiertage.
- Reparaturen müssen innerhalb von 24 Stunden abgeschlossen sein.

* Quelle: AkSeQM

Ersatzteilverhaltung

- Als Bestandteil des Angebots ist der Anbieter aufgefordert, eine Liste aller aktiven Komponenten zu erstellen, die für die schnelle Fehlerbehebung notwendig sind
- Diese Ersatzteilliste sollte die genaue Anzahl aller Ersatzteile bei der angestrebten Gesamtkonfiguration enthalten
- Die Versorgung mit Ersatzteilen muss auf mindestens zehn Jahre garantiert werden
- Der Auftragnehmer hat für eventuelle Gewährleistungsansprüche sowie für die Wartung zuständige Ansprechpartner und Servicestellen schriftlich zu benennen und als Anlage dem Angebot beizufügen

* Quelle: AkSeQM

Haftung

- Der Auftragnehmer haftet über eine nachzuweisende Betriebshaftpflichtversicherung für alle bei seinen Arbeiten entstehenden Schäden situativ in Abhängigkeit vom Auftrag (nach Möglichkeit unbegrenzt), z.B.:
 - Sachschäden mindestens bis zu einem Betrag 2 Mill. €,
 - Personenschäden mindestens bis 1 Mill. €,
 - Vermögens-, Betriebsausfallschäden bis ???
- Der Auftraggeber behält sich vor,
 - bei Gefährdung des Inbetriebnahmetermins,
 - bei nachgewiesenem Unvermögen des Auftragnehmerseine dritte Fachfirma zu Lasten des Auftragnehmers mit der Beseitigung der festgestellten Mängel zu beauftragen.

* Quelle: AkSeQM

Möchten Sie hochkomplexe Software weiterentwickeln?

Problem: Es gibt Grenzen für Größe und Komplexität von Software, danach wird es hoffnungslos.:

- Zu fehleranfällig, um sie zu benutzen
- Zu komplex zum fixieren von Bugs
- Zu groß für ein Redesign

Lösung: Kontrollieren der Komplexität während der Entwicklung und Wartung

* Quelle: AkSeQM

Software-Zuverlässigkeit

Zuverlässigkeit ist ein auf die Zeit bezogener Qualitätsfaktor, der für eine definierte Umgebung die Wahrscheinlichkeit bis zum Eintreten eines Fehlers charakterisiert.

Direkte Qualitätsmetriken der Zuverlässigkeit beinhalten Fehlerraten pro Zeiteinheit und Zeitdauer zwischen Versagensfällen. Erlaubte Grenzwerte sind bei größeren (Auftrags-)Projekten vertraglich festgelegt.

Software Reliability Engineering (SRE) hat sich im Bereich der Prognose, Modellierung und insbesondere der Konstruktion zuverlässiger Komponenten und Systeme als eigenständige Disziplin vor allem in der Telekommunikations- und Raumfahrttechnik, entwickelt.

* Quelle: AkSeQM

Software-Zuverlässigkeit

Software Reliability Engineering (SRE) umfasst folgende Teilbereiche:

- Auswahl von kundenorientierten und indirekten Zuverlässigkeitsmetriken (incl. Process Quality Checks)
- Vorhersage der zu erwartenden Software-Zuverlässigkeit mit Modellen des Entwicklungsprozesses und seines Einflusses auf Fehlerentstehung, -entdeckung und -korrektur
- Unterstützung und Auswahl von Teststrategien, die realisierungsabhängig (white box, Kontrollflussabdeckung) oder nutzungsabhängig (black box, operational profiles) sind
- Einsicht in den Entwicklungsprozess und dessen Einflüsse auf die Software-Zuverlässigkeit
- Verbesserung des Entwicklungsprozesses, um die Zuverlässigkeit zu erhöhen

* Quelle: AkSeQM

Software-Zuverlässigkeit

Die Kundenorientierung in SRE kommt am deutlichsten zum Ausdruck in der Unterscheidung zwischen Fehlverhalten (failure), zu korrigierendem Software-Fehler (fault) und Fehlerursache (error).

Kunden sind üblicherweise an der Reduzierung des Fehlverhaltens interessiert. Spätestens im Systemtest spielt die berücksichtigten Testverfahren mit den realistischen Testfällen (Geschäftsvorfällen) eine große Rolle, in dem oft noch ein großes Sparpotential steckt.

Software-Zuverlässigkeit ist eine Funktion der Restfehler in der Software und der Gebrauchshäufigkeit realisierter Anforderungen.

* Quelle: AkSeQM

Softwarewartung

Im wesentlichen lassen sich zwei unterschiedliche Kategorien der "Softwareproduktion" unterscheiden.

1. Lieferung von Standardsoftware:

- a) Der Softwarehersteller **entwickelt und vertreibt** ein Anwendungsprogramm als **Komplettlösung** für ein spezifiziertes Anwendungsgebiet. Diese Komplettlösungen werden im allgemeinen Sprachgebrauch als Standardsoftware bezeichnet.
- b) Der Softwarehersteller **entwickelt** eine **Basissoftware** zu einem spezifiziertem Anwendungsgebiet (z.B. CAD, CIM, etc.). Die Anbindung von Applikationen (Anwendungsprogrammen) erfolgt über zur Verfügung gestellte APIs (Application Programmers Interface).
- c) Der Softwarehersteller **entwickelt und vertreibt eine Applikation**, die auf der API einer **Basissoftware** aufsetzt.

* Quelle: AkSeQM

Softwarewartung

2. Erstellung von Individualsoftware:

- a) Nach Abnahme geht der Sourcecode in den Besitz des Auftraggebers über. Die Wartung sowie die Weiterentwicklung des Systems erfolgt anschließend durch den Auftraggeber, sofern kein Wartungsvertrag abgeschlossen wird.
- b) Die Abnahme der Software erfolgt nach den Vorgaben des Lastenheftes / Pflichtenheftes sowie den vertraglichen Regelungen. Der Sourcecode verbleibt beim Auftragnehmer, eine kostenfreie Wartung erfolgt im Rahmen der vereinbarten Gewährleistungsfrist und beschränkt sich auf die Fehlerbehebung.

Sowohl bei Individual- als auch bei Standardsoftware kann notwendige (Zusatz-) Hardware zum Lieferumfang gehören. Die jeweiligen Verträge sollten jedoch unbedingt getrennt gehalten werden.

* Quelle: AkSeQM

Wartungsverträge

Im Rahmen der Gewährleistung (Garantie) verpflichtet sich der Hersteller eines Produktes zur kostenlosen Fehlerbehebung.

Soll das Vertragsverhältnis zwischen Softwarehersteller und Anwender über den Gewährleistungszeitraum hinausreichen, so bietet sich der Abschluss eines Wartungsvertrages an.

Die Kosten variieren hierbei zwischen 1.0% (reine Fehlerbeseitigung) und 1.5% (mit Weiterentwicklung) des vereinbarten Lizenzpreises, pro Monat (15 bis 18% / Jahr).

Auf Grund der vertraglichen Gestaltung (Softwarepflege und -wartung werden meist nicht getrennt) sind diese Wartungsverträge von entscheidender Bedeutung für die **Weiterentwicklung** der Systeme.

* Quelle: AkSeQM

Wartungsverträge

Eine Trennung von **Wartung** (im Sinne einer Reparatur) und (funktionaler) **Weiterentwicklung eines Systems** ist, speziell bei Standardsoftware, kaum möglich.

Die Hersteller von Standardsoftware bieten im Rahmen der Wartungsverträge daher regelmäßige **Updates** an. Diese Updates bestehen aus einer Kombination von Weiterentwicklungen und Fehlerbehebungen.

Je nach Gestaltung des Wartungsvertrages sind Updates entweder kostenlos oder gegenüber dem Listenpreis deutlich im Preis reduziert.

* Quelle: AkSeQM

Wartungsphase im (Life-Cycle) System

Da eine Entsorgung von Softwareprodukten nur in Ausnahmen, z.B. unter bestimmten sicherheits- und datenschutztechnische Aspekten, sinnvoll ist und Software nicht verschleißt, höchstens veraltet, stellt die Wartungsphase den längsten Teil im Lebenszyklus einer Software dar.

Wartungsaktivitäten stehen für ca. 70% der life-cycle Kosten eines Softwareproduktes, wobei hiervon etwa 60% in die Weiterentwicklung fließen.

Kernprozess der Softwarewartung ist damit die **Weiterentwicklung** (Pflege) des Systems, d.h. die Einführung neuer oder erweiterter Funktionalitäten, die Verbesserung softwareergonomischer Gesichtspunkte sowie Performancesteigerungen und Anpassungen an neue oder erweiterte Hardwaresysteme.

* Quelle: AkSeQM

Wartungsphase im (Life-Cycle) System

Mit einem Anteil von ca. 28% an den life-cycle-Kosten hat die **Fehlerbehebung** (Wartung) aber nach wie vor entscheidenden Einfluß auf die Systemkosten.

Die Qualität des ursprünglichen Entwurfs und die Realisierung eines "wartungsfähigen Produktes" bestimmen somit wesentlich den zur Verfügung stehenden Freiraum für Neuentwicklungen.

Je geringer die tatsächlich notwendigen Aufwendungen des Softwareherstellers für Fehlerbehebung und Weiterentwicklung ausfallen, desto mehr Ressourcen stehen für neue Projekte zur Verfügung.

* Quelle: AkSeQM

Maintenance Engineering

Dieses Arbeitsfeld befasst sich mit

- der Prozessführung bei Fehlerbehebungen,
- der Anpassung von Programmfunktionalitäten an veränderte Anforderungen (*Funktionalitätsanpassungen*),
- der Weiterentwicklung bestehender Systeme, d.h. der Einführung neuer oder erweiterter Funktionalitäten (*Funktionalitätserweiterungen*).

* Quelle: AkSeQM

Maintenance Engineering

Fehlerbehebung

Unter diesem Begriff werden alle Arbeiten zusammengefasst, die durch die Nichterfüllung dokumentierter Anforderungen notwendig werden.

Dazu gehören neben offensichtlichen Programmfehlern auch Fehlreaktionen der Software, die sich nur indirekt aus den Anforderungen ableiten lassen.

Die Spanne reicht dabei von nicht initialisierten Variablen über fehlerhafte Ausgaben bis zu funktionalen Abweichungen gegenüber dem Pflichten-/Lastenheft.

* Quelle: AkSeQM

Maintenance Engineering

Bei der Definition von Fehlerklassen, die gleichbedeutend mit Priorisierungen bzw. den Dringlichkeitsstufen sind, bieten sich die vier Kategorien des IEEE (Computer Society of the Institute of **E**lectrical and **E**lectronics **E**ngineers) Standard Computer Dictionary an.

Demnach lassen sich Programmfehler in folgende Kategorien einteilen:

- **failure** (fehlerhaftes Ergebnis),
- **fault** (fehlerhafte Anweisung oder Definition),
- **defect** (Unvollständigkeit) oder
- **mistake** (fehlerhafte Eingaben etc.)

* Quelle: AkSeQM

Maintenance Engineering

Funktionalitätsanpassungen

Hierunter werden diejenigen Arbeiten gesammelt, die zur Anpassung der Software an neue Hardware und/oder geänderte Betriebssysteme notwendig sind.

Weiterhin werden geänderte bzw. neue Schnittstellen zu Vor- und Nachlaufprogrammen unter diesem Stichwort eingeordnet.

Ebenfalls als Anpassung wird in diesem Zusammenhang die Korrektur von Spezifikationsfehlern, d.h. deren programmtechnische Umsetzung bezeichnet.

* Quelle: AkSeQM

Maintenance Engineering

Funktionalitätserweiterungen

Neben der Einführung neuer oder erweiterter Funktionalitäten, Performance-Verbesserungen und gleichzeitiger Nachführung von Benutzer Dokumentation, wird hierunter beispielsweise die Anpassung an (vorher nicht unterstützte) Betriebssysteme als Weiterentwicklung geführt.

* Quelle: AkSeQM

System Change Request

Die Erfassungs-, Planungs- und Controllingaspekte des Maintenance Engineering lassen sich durch ein einheitliches Datenbank gestützte Verfahren automatisieren.

Hierbei beschränkt es sich nicht auf die Fehlerverfolgung sondern auf das gesamte Änderungswesen und wird als Begriff "**System Change Request**" beschrieben.

In der Praxis hat sich die Aufteilung des Verfahrens zur Systemänderung (Fehlerbehebung und Weiterentwicklung) in **fünf Bearbeitungsstufen** bewährt.

Für jede dieser Stufen können zuständige Abteilungen festgelegt werden, was im Sinne einer prozessorientierten Arbeitsteilung allerdings eher als Empfehlung zu verstehen ist.

* Quelle: AkSeQM

System Change Request

Im Einzelnen sind dies die Abteilungen

1. **Support** bzw. **Hotline** für die Erfassung,
2. **Test** für Klassifizierung, Nachvollziehbarkeit und Problembeschreibung (Reproduktion),
3. **Projektmanagement** für die Projektzuordnung und Auftragsvergabe,
4. **Entwicklung** für die Durchführung der Änderung
5. **Test/QS** für die Freigabe (Verifikation) der durchgeführten Änderungen.

* Quelle: AkSeQM

System Change Request

1. Auswertung der Hotlineanfragen

Da erfahrungsgemäß neben den **Fehlermeldungen** auch die meisten **Wünsche** zu Funktionalitäten bei der Hotline ankommen, ist der Support die zentrale Abteilung für Anregungen zur Weiterentwicklung eines Softwareproduktes. Um **statistisches Material** über die Benutzerfreundlichkeit u.ä. Qualitätskriterien zu gewinnen, sollten daher auch solche Meldungen in ein Datenbanksystem eingebracht werden, die sich augenscheinlich nicht auf Programmfehler beziehen. Dazu gehören beispielsweise **Fehleingaben durch den Benutzer** u.ä.. Zur **Verbesserung der Softwareergonomie** ist es zwingend notwendig, **Kenntnisse über die Erwartungen des Benutzers** zu gewinnen. Während der Wartungsphase ist die **Auswertung der Hotline-Anfragen** daher die **zuverlässigste Meßplatte für die Qualität einer Software**.

* Quelle: AkSeQM

System Change Request

2. Nachweis der Reproduzierbarkeit

Neben der Übersetzung der Anforderung in die "Sprache der Entwickler", wird hier eine erste Zuordnung zu Projekt, Priorität und Anforderungsart (Menüführung, Darstellungsfehler, Berechnungsfehler, ...) vorgenommen.

Die Hauptaufgabe in dieser Stufe liegt auf der eindeutigen **Reproduzierbarkeit** und der Beschreibung wie das Verhalten (Fehler) zu reproduzieren ist.

Für **Fehleranalysen** interessant ist die Einteilung von Fehlermeldungen in Programm-, Dokumentations- und Anwendungsfehler.

* Quelle: AkSeQM

System Change Request

3. Priorisierung

In dieser Stufe erfolgt die Priorisierung und Projektzuordnung, unter Berücksichtigung vorhandener Projektpläne und Ressourcen. auf der Ebene des Projektmanagements.

Dokumentations- oder Konfigurationsprobleme werden an die entsprechenden Abteilungen adressiert und zu programmierende Lösungen werden als Programmierauftrag der entsprechenden Stelle zugewiesen und terminiert.

Entscheidend ist die richtige Gewichtung der Änderung, um die Effizienz der durchzuführenden Arbeiten zu maximieren.

* Quelle: AkSeQM

System Change Request

4. Durchführung der Änderung

Neben der Auftragsbearbeitung, d.h. der eigentlichen Durchführung der Software-Änderung, ist die Entwicklungs-/Wartungsabteilung für die **Dokumentation** der Änderung sowie für das Erstellen der **Testanforderung** zuständig.

Diese Testanforderung dient der folgenden Stelle als Arbeitsgrundlage für den abschließenden Test.

Ergeben sich aus der Auftragsbearbeitung heraus notwendige Anpassungen von Konfiguration und/oder Dokumentation, so werden die entsprechenden Anforderungen innerhalb dieser Stufe als entsprechender Änderungsauftrag formuliert.

* Quelle: AkSeQM

System Change Request

5. Verifikation

Die abschließenden **Tests** dienen der **Vermeidung von Folgefehlern** und damit als Freigabeverfahren.

Sollte eine Freigabe wieder Erwarten nicht möglich sein, geht der Vorgang, entsprechend kommentiert, zurück zur Stufe 3.

* Quelle: AkSeQM

Trennung von Wartung und Entwicklung ?

Als Ergebnis von Untersuchungen führt die organisatorische Trennung von Wartung und Entwicklung zu einer deutlichen Effektivitätssteigerung der Wartungsaktivitäten.

Insbesondere werden hierdurch (nicht transparente) Prioritätsüberlegungen vermieden, die erfahrungsgemäß fast ausschließlich zugunsten von Neuentwicklungen entschieden werden.

Sich hier gegen kurzfristige (und -sichtige) Kostenargumente zur Wehr zu setzen ist Aufgabe des Projektmanagements und damit des Qualitätsmanagements!

* Quelle: AkSeQM

Trennung von Wartung und Entwicklung ?

Wartungsteams rekrutieren sich aus der gesamten Entwicklungsmannschaft.

Für die Bereiche Wartung und Entwicklung ist jeweils ein Vorgesetzter mit fachlicher Weisungsbefugnis zuständig (beispielsweise Produkt-, Projektmanager), die disziplinarische Weisungsbefugnis wird durch eine übergeordnete Stelle wahrgenommen.

Die Wartbarkeit der Software bleibt bei dieser Organisationsart gewährleistet, wenn Fehlerbehebungen nicht automatisch vom Autor des ursprünglichen Sourcecodes durchgeführt werden. Die Verteilung von Wartungsaufgaben erfüllt damit den gleichen Zweck wie die organisatorische Trennung von Wartung und Entwicklung.

Zusätzlich wird eine breitere Streuung des Know-how erreicht, was zur Folge hat, dass seitens der Projektleitung flexibler auf zusätzliche Anforderungen reagiert werden kann.

* Quelle: AkSeQM

Trennung von Wartung und Entwicklung ?

Erfahrungsgemäß erarbeitet ein hochmotiviertes Team wartungsfreundliche und fehlerarme Software, da es sich mit seinem Produkt identifiziert und an dessen "Marktfähigkeit" interessiert ist.

Das Qualitätsmanagement hat hierbei unterstützende und lenkende Funktionen, indem es für die Schaffung entsprechender Rahmenbedingungen sorgt.

Damit lautet die Antwort auf die am Anfang des Kapitels gestellte Frage =>

„Integration der Wartung in die Entwicklung, ist ganz im Sinne einer Weiterentwicklung in Richtung
Total Quality Management.“

* Quelle: AkSeQM

Konfigurationsmanagement

Aus dem Verständnis der Softwareentwicklung heraus befaßt sich das Konfigurationsmanagement (KM) mit den Entwicklungsstadien von Softwareprojekten.

Darunter ist die Verwaltung und Versionierung von Standard- bzw. Individualsoftware sowie die Konfektionierung von fertiggestellten Komponenten zu Programmversionen (Releases) zu verstehen.

KM ist damit in einen verfahrenstechnisch orientierten (administrativen) und einen unterstützenden (lenkenden) Zweig aufteilbar.

Software-Versionsverwaltung ist in diesem Sinne integraler Bestandteil des Software-Konfigurationsmanagements (SKM).

* Quelle: AkSeQM

Konfigurationsmanagement

Begriffsdefinitionen:

- **Version** (modification level; version)/ Synonym:
Modifikationsstufe
Ein Begriff zur Kennzeichnung der jeweiligen Änderungsstufen eines bestimmten Objekts, etwa eines Programms oder einer Datei. Um verwaltungsmäßig gleichzeitig verschiedene Entwicklungs- oder Änderungsstufen ein- und desselben Objekts behandeln zu können, werden häufig fortlaufende Versionsnummern vergeben. (aus "Lexikon der Informatik und Datenverarbeitung, Hans-Jochen Schneider")
- **version number**
Identification of a release of software. ... Version numbers, such as 3.1a or 3.11, indicate a follow-up release only to fix a bug in the previous version, whereas 3.1 and 3.2 usually indicate routine enhancements in the product. (aus "The Computer Glossary; Alan Freedman")

* Quelle: AkSeQM

Konfigurationsmanagement

- **version control**

Management of source code in a large software project. Version-control software provides a database that keeps track of the revisions made to a program by all the programmers involved in it. (aus "The Computer Glossary; Alan Freedman")

- **Konfiguration**

Zusammensetzung eines Objekts. ... Gegenseitige Positionierung oder Anordnung der Teile oder Elemente eines Objekts. Im Qualitätsmanagement ist das Objekt ein materielles oder immaterielles oder ein kombiniertes Produkt. (aus "Begriffe zum Qualitätsmanagement; DGQ")

- **Konfigurationsmanagement**

Gesamtheit der Tätigkeiten zur Sicherstellung der zweifelsfreien Rückverfolgbarkeit der Konfiguration über alle Stufen der Weiterentwicklung eines materiellen oder immateriellen Produkts von einer bezeichneten Bezugskonfiguration an. (aus " "Begriffe zum Qualitätsmanagement; DGQ")

* Quelle: AkSeQM

Versionsverwaltung

Der administrative Zweig des KM befaßt sich mit der Versionskontrolle (-verwaltung), d.h. der Verwaltung der verschiedenen Versionen aller im Rahmen eines Softwareprojekts produzierten Dateien sowie deren Bereitstellung für die Entwickler.

Damit eng verbunden sind die notwendigen Freigabeverfahren (Check in / check out; Sperrlisten, Zugriffsberechtigungen, etc.) wie sie für größere Softwareprojekte unverzichtbar sind. Die Versionsverwaltung ist in hohem Maße durch Datenbank gestützte Hilfsroutinen automatisierbar.

* Quelle: AkSeQM

Versionsverwaltung

Softwareprojekte bestehen aus einer Vielzahl von Einzelteilen wie Spezifikationen, Datenbanken, Modulen, Rechenkerne, Quellcodedateien, Benutzerhandbücher, etc., wobei jedes Teil als Komponente des Gesamtprojekts bezeichnet werden kann.

Über die Projektlaufzeit werden ständig neue Versionen dieser Komponenten erzeugt (weiterentwickelt oder geändert). Eine ganz bestimmte Versionen einer Komponente wird als Objekt bezeichnet, die Gesamtheit der Objekte bildet die Projektbibliothek.

* Quelle: AkSeQM

Versionsverwaltung

Ersetzt ein Objekt sein Vorgängerobjekt (Weiterentwicklung, Fehlerbehebung), wird das entstandene Objekt als "Revision" des Vorgängerobjekts bezeichnet.

Existiert ein neues oder geändertes Objekt parallel zu einem Objekt derselben Komponente (Anpassung an spezielle Hardware, Betriebssystem, ...), wird es als Variante des Vorgängerobjekts bezeichnet.

Die Zusammensetzung von Objekten zu größeren Einheiten bildet schließlich eine Konfigurationen.

* Quelle: AkSeQM

Versionsverwaltung

Zur Bearbeitung eines Objektes wird es aus der Projektbibliothek in den Arbeitsbereich des zuständigen Bearbeiters kopiert (check-out) und für den Zugriff anderer Bearbeiter gesperrt (locked).

Nach erfolgter Änderung wird das Objekt in die Projektbibliothek zurückgestellt (check-in) und für die weitere Bearbeitung mit neuer Versionskennung freigegeben (unlocked).

* Quelle: AkSeQM

Versionsverwaltung

Wird an mehreren Entwicklungsstandorten an einem Projekt gearbeitet, so ist die Versionsverwaltung unter diesen Standorten zu standardisieren. Für eine globale Kooperation bedeutet dies beispielsweise die Verwendung eines englischsprachigen Tools sowie GMT als Systemzeit.

Aufgabe der Versionsverwaltung ist es somit, Verfahren und Methoden zur Sicherstellung einer funktionellen Entwicklungsumgebung zur Verfügung zu stellen und die Rückverfolgbarkeit von Entwicklungen über die Rekonstruktion vorhergehender Softwarestände zu gewährleisten.

* Quelle: AkSeQM

Releaseplanung

Voraussetzung für eine Versions- oder Releaseplanung ist die **Skalierbarkeit** eines Gesamtprojektes in Teilprojekte, mit definierten Merkmalen bezüglich Benutzerfreundlichkeit, Funktionalität, Performance und Stabilität.

Bei einer durchschnittliche Produktlebensdauer von größeren Softwarepaketen zwischen 5 und 10 Jahren, liegt die Aufgabenstellung des Software-KM in der Sicherstellung der Wartbarkeit und damit der Weiterentwicklungsfähigkeit der Software.

* Quelle: AkSeQM

Releaseplanung

SKM als technische Projektunterstützung

Zu den entwicklungsunterstützenden Aufgaben des KM gehört die Bereitstellung von Informationen über:

- Compilereinstellungen (verwendete Compileroptionen) zur Übersetzung von Modulen oder Programmteilen, um eine bestimmte Version des Programms X für den Kunden Y zu erzeugen.
- Zuordnung von Versionen zu Entwicklungs-, Test- und Installationsumgebungen (HW, Netz, Peripherie, SW-Tools)
- Verwaltung der Abhängigkeiten von Softwarekomponenten (Schnittstellendefinitionen)
- Konsistenzkontrolle der unterschiedlichen Versionen von Spezifikationen, Sourcecode, Objectcode, Dokumentation
- Change Request und Bug Tracking

* Quelle: AkSeQM

Releaseplanung

Anforderungen an ein SKM-Tool (1)

- Auslegung für größere Projekte (>10000 Files)
- Plattformunabhängig (Client-Server)
- Unabhängig von Dateiformaten (Binär, ASCII, Sourcecode, Testanweisungen)
- Integrierbar in eine evtl. vorhandene Entwicklungsumgebung
- Zugriffskontrollen durch lock und unlock Funktionen sowie Zugriffs- und Sperrlisten
- Revisionsmanagement (paralleles Entwickeln an verschiedenen Objekten eines Projektes)
- Versionsverwaltung (Zusammenstellung von Dateirevisionen zu Gesamtsystemen [Konfiguration] unter einem Versionsnamen)
- Verwaltung von Make/Build Komponenten (Rekonstruktion der Compilereinstellungen, ...)

* Quelle: AkSeQM

Releaseplanung

Anforderungen an ein SKM-Tool (2)

- History-Funktionalitäten (Verfolgung der Entwicklungsgeschichte von Objekten)
- Darstellung von Beziehungsbäumen (Erkennen von Abhängigkeiten zwischen Objekten innerhalb einer Konfiguration)
- Parallelentwicklung einer Variante aus dem Hauptentwicklungszweig (Branch and Merge Funktionalität)
- Kennzeichnung der Entwicklungsstufe des Objektes (Entwicklung, QS, Produktion)
- Notify-Funktionalitäten, z.B. bei der Erreichung von Projektmeilensteinen zur Übergabe an eine andere Programmiergruppe
- Release Identifikation (automatische Vergabe von Revisions und Versionskennungen)
- Change Request Management (Erfassung und Verfolgung von Änderungsaufträgen)
- Report-Funktionen

* Quelle: AkSeQM

Releaseplanung

SKM-Plan

Die genannten Tätigkeiten sollten in einem Konfigurationsmanagementplan zusammengefasst werden, um eine Normierung von Projekten zu erreichen.

Teilweise werden Konfigurationsmanagenent-Pläne bereits von den Auftraggebern gefordert.

* Quelle: AkSeQM

Releaseplanung

QM-Aufgaben

Als unterstützende Maßnahmen seitens der QA sind die Verwendung von Namenskennungen (Nomenklatur, Objekt- und Source-Extensions) sowie die Anwendung der Programmierrichtlinien (Vorgaben für Dokumentation im Source) zu kontrollieren. Weiterhin ist die Aufnahme von Testprogrammen, Testanweisungen in das SKM anzustreben.

Die Ergebnisse eines funktionierenden KM zeigen sich in folgenden Punkten:

- Sicherstellung der Rückverfolgbarkeit von Entwicklungs-, Änderungsschritten
- Rückverfolgbarkeit kundenspezifische Anpassungen
- Eindeutige Identifizierbarkeit der Version eines jeden Softwareelements
- Eindeutige Festlegung des Entwicklungsstandes einer Softwareelements
- Eindeutige Zuordnung von Programm- und Userdokumentation zu Releases

* Quelle: AkSeQM

Beispiel eines Standard Verfahrens zur Konfektionierung von Produkt Updates

- Kontrolle auf ISO 9660 (High Sierra Group proposal) konforme Benennung der Files für die Master-CD.
Diese Norm stellt den kleinsten gemeinsamen Nenner aller Betriebssysteme dar und sichert somit die Lesbarkeit der Dateinamen unter den verschiedenen Betriebssystemen.
- Festlegung der dem Anwender zu präsentierenden Versionsnummern. **Voneinander abhängige Dateien werden generell zusammen versioniert und kompiliert und haben identische Versionsnummern.**
- Erstellung einer Datenbank für das Installationsprogramm (z.B. Auszug der Projektdirectory) mit den Feldern:
 - Filename, Filegröße, Filedatum und Versionsnummer
 - Filename Vollständigkeitskontrolle
 - Filegröße Platzberechnung
 - Filedatum Kontrollmechanismen für Userdaten
 - Versionsnummer Rückverfolgbarkeit innerhalb der Entwicklungsumgebung (Fehlerbehebung)

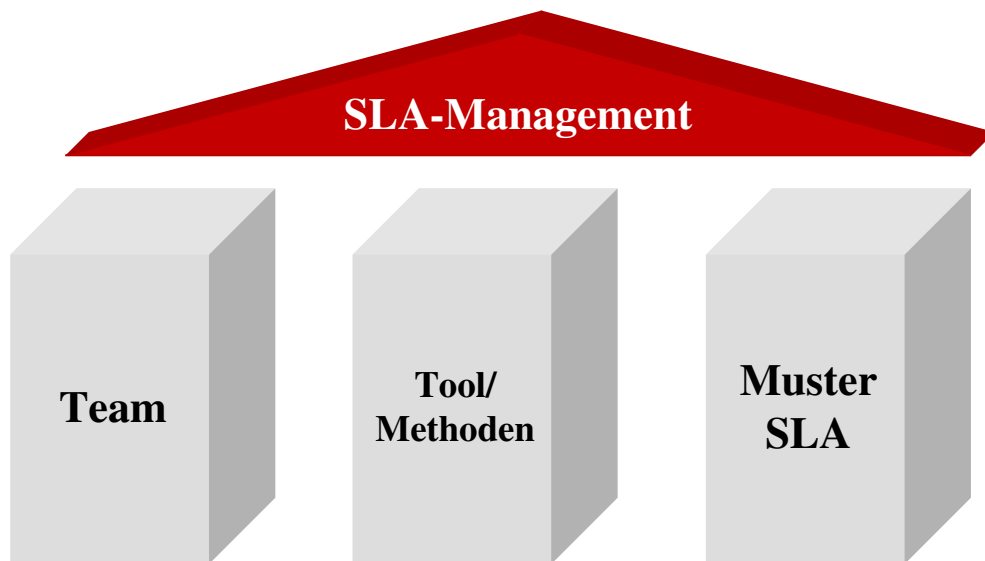
* Quelle: AkSeQM

Beispiel eines Standard Verfahrens zur Konfektionierung von Produkt Updates

- Zuordnung der Benutzerdokumentationen wie Handbuch, Online Help, Tutorial, ReadMe-File, Beispiele und Informationen zu "Known Bugs" zum Auslieferungsprodukt.
- Durchführung des Acceptance Test
- Erstellung der Musterdatenträger als Referenzexemplar und Master-CD. Die Master-CD (Kopierwerkfassung) wird inklusive eines mit Release-Nr. versehenem Coverfilms zur Produktion weitergeleitet.
- Jeweils ein Test auf technische Funktionsfähigkeit (Kopierfehler) und Virenfreiheit pro Charge produzierter CD's
- Bereitstellung zur Auslieferung

* Quelle: AkSeQM

SLA-Management



SLA-Management

Servicezeiten		
Kategorie A	Montag bis Freitag	7 - 18 h
	Samstag	7 - 14 h
Kategorie B	Montag bis Freitag	18 - 7 h
	Samstag	14 - 24 h
	Montag	0 - 7 h
	Sonntag, Feiertag	0 - 24 h

Wartungsfenster

Systemwartungsarbeiten können am Sonntag von 20 – 24 h nach vorheriger Absprache durchgeführt werden.

System Reboot

Am 1. Sonntag im Monat wird um 20 h das System neu gestartet, um das Risiko für ungeplante Ausfälle zu minimieren.

SLA-Management

System Funktion 1	Servicezeit Kategorie A	99 % pro Monat 98,5 % pro Monat 98 % pro Monat	erfüllt minder erfüllt nicht erfüllt
System Funktion 1	Servicezeit Kategorie B	98,5 % pro Quartal 98 % pro Quartal 97,5 % pro Quartal	erfüllt minder erfüllt nicht erfüllt
System Funktion 2	Servicezeit Kategorie A	98 % pro Monat 97 % pro Monat 96 % pro Monat	erfüllt minder erfüllt nicht erfüllt

Als Ausfall gilt jede Situation, in der mit der Schnittstelle des Rechenzentrums nach außen (Router) keine Kommunikation von außen her möglich ist.

SLA-Management

Problem-priorität	Von Ausfall oder Störung betroffene Komponente(n)	Auswirkungen
1	Gesamtsystem Rechner, Betriebssystem, wesentliche Teile der Applikation, zentrale Kommunikations-SW/HW (alle Router, zentraler Netzwerk-/Konfigurationsserver)	Gesamtausfall betroffen: alle Benutzer
2	Funktionen der Applikation (nicht Klasse 1) Datenbankprobleme (nicht Klasse 1), Performanceprobleme im zentralen Bereich, Subsysteme, zentraler Server für Datenverteilung	Teilausfall: betroffen: Großteil Benutzer
3	Datenleitung , zentraler Router, Ausfall zentraler HW- und SW-Komponenten mit Umgehungsmöglichkeit (Backup), Probleme im zentralen Bereich	Gesamtausfall für einzelne Systeme
4	Routerprobleme , Probleme mit Adressbereichen	Teilausfall innerhalb eines Systems
5	Userproblem , Problem mit Adressing, HW-Problem, Anbindungsproblem zu zentralen Server	Einzelner Arbeitsplatz

SLA-Management

Priorität	Maximale Suchzeit	Einleitung Bearbeitung	Bearbeitungsdauer (grob)	Maßnahmen durch Call Desk	Maßnahmen durch Call Desk
1	30 Min.	sofort	< 2,5 Std.	INFOS sofort nach 75 Min. nach 120 Min. nach 180 Min. Dokumentation	Kunden-Betreuer und Service Manager Account Manager und Geschäftsführung Einberufung Task Force Katastrophenvorsorgeplan Service Center und Monatsbericht
2	45 Min.	sofort	< 3 Std.	INFOS sofort nach 120 Min. nach 180 Min. nach 360 Min. Dokumentation	Kunden- Betreuer und Service Manager Account Manager und Geschäftsführung Einberufung Task Force Katastrophenvorsorgeplan Service Center und Monatsbericht
3	4 Std.	sofort	< 6 Std.	INFOS sofort nach 180 Min. nach 360 Min. Dokumentation	Kunden- Betreuer und Service Manager Account Manager Geschäftsführung Service Center und Monatsbericht
4	5 Std.	gleicher Tag	< 12 Std.	nach 180 Min. nach 360 Min. Dokumentation	Kunden- Betreuer und Service Manager Account Manager Service Center und Monatsbericht
5	2 Std.	gleicher Tag	< 24 Std.	Info an	Kunden- Betreuer und Service Manager