

Qualitätssicherung	3
Qualität.....	3
Qualitätssicherung.....	3
Qualitätsmanagement.....	3
QM-Prozess.....	3
Organisatorische Massnahmen.....	3
Konstruktive Massnahmen.....	3
Analytische Massnahmen.....	3
Standards	4
ISO.....	4
CMM.....	4
TMM.....	4
ANSI / IEEE.....	4
DIN.....	4
EFQM.....	4
Qualitätsdimensionen	5
ISO/IEC 9126.....	5
Funktionalität (Functionality).....	5
Zuverlässigkeit (Reliability).....	5
Benutzbarkeit (Usability).....	5
Effizienz (Efficiency).....	5
Wartbarkeit (Maintainability).....	5
Übertragbarkeit (Portability).....	5
Statische Prüfverfahren	6
Definition.....	6
Vorteile / Nachteile.....	6
Informeller Review.....	6
Round Robin Review.....	6
Technischer Review.....	6
Inspektion.....	6
Audit.....	6
Dynamische Prüfverfahren	7
Definition.....	7
Einteilung in Qualitätsdimensionen.....	7
Zuverlässigkeit / Effizienz.....	7
Funktionalität.....	7
Benutzbarkeit.....	7
Wartbarkeit / Übertragbarkeit.....	7
Teststufen	8
Grafik Teststufen.....	8
Unittest.....	8
Integrationstest.....	8
Systemtest.....	8
Abnahmetest.....	8
Regressionstest.....	8
Live-Test.....	8
Testmethoden	9
Definition Testmethode.....	9
Whitebox Test.....	9
Greybox Test.....	9
Blackbox Test.....	9
Äquivalenzklassenbildung.....	9
Grenzwertanalyse.....	9
Diversifizierende Methoden.....	9
Testtools	10
Definition.....	10
Tools zur Testprozessunterstützung.....	10
Tools für statische Prüfungen.....	10
Tools zur Testvorbereitung.....	10
Tools zur Testdurchführung.....	10
Teststrategie	11
Teststrategie im Unternehmen.....	11
Inhalt Teststrategie.....	11
Testverfahren.....	11

Prüfobjekte.....	11
Testprozess.....	12
Testprozess im Unternehmen.....	12
ANSI/IEEE 829.....	12
Testvorbereitung.....	12
Preparation of Tests.....	12
Testdurchführung.....	12
Running the tests.....	12
Testauswertung.....	12
Completion of testing.....	12
Testrollen.....	13
Organigramm Testteam.....	13
Testmanager.....	13
Testdesigner.....	13
Testengineer.....	13
Testautomatisierer.....	13
Testadministrator.....	13
Tester.....	13
Testdokumente.....	14
Verwaltung Testdokumente.....	14
Vorteile Testdokumentation.....	14
Testplan (Prüfplan).....	14
Testkonzept.....	14
Testfallspezifikation.....	14
Testdrehbuch.....	14
Berichtsdokumente.....	15
Testprotokoll.....	15
Testvorfallbericht / Problemmeldungen.....	15
Testbericht.....	15
Testabschlussbericht.....	15
Testmetriken.....	15
Fehlertrendstatistik.....	15
Diagramm Testfortschritt.....	15
Problembearbeitung.....	16
Ablauf Problembearbeitung.....	16
Problemart.....	16
Schweregrad.....	16
Priorität.....	16
Problemstatus.....	16

Qualitätssicherung		QUAL
<p style="text-align: center;">Qualität</p> <ul style="list-style-type: none"> • Erfüllung der Anforderungen • "wenn der Kunde zurückkommt, nicht das Produkt" <p>Voraussetzungen:</p> <ul style="list-style-type: none"> • Vertraulichkeit • Integrität • Verfügbarkeit 	<p style="text-align: center;">Qualitätssicherung</p> <p>Bestandteil des Qualitätsmanagements</p> <p>Zur Qualitätssicherung gehören alle operativen Tätigkeiten, die vorbereitend, begleitend und prüfend die definierte Qualität eines Produktes oder einer Dienstleistung gewährleisten sollen.</p> <p>Zweck:</p> <ul style="list-style-type: none"> • Abwenden von Risiken • Vorhersehbare Projektplanung • Erreichen eines allgemein erwarteten Qualitätsniveaus 	
<p style="text-align: center;">Qualitätsmanagement</p> <p>Oberbegriff für alle Tätigkeiten, Führungsaufgaben und Methoden, die zur Planung, Sicherung, Prüfung und Verbesserung der Qualität eines Produktes oder einer Dienstleistung gehören.</p> <p>Bereiche:</p> <ul style="list-style-type: none"> • Qualitätsplanung (Quality Planning, PLAN) • Qualitätslenkung (Quality Control, DO) • Qualitätssicherung (Quality Assurance, CHECK) • Qualitätsverbesserung (Quality Improvement, ACT) 	<p style="text-align: center;">QM-Prozess</p> <p>(kontinuierlicher Verbesserungsprozess KVP) (Deming Cycle)</p> <pre> graph TD PLAN[PLAN Planungsphase: Organisatorische Massnahmen] --> DO[DO Umsetzungsphase: Konstruktive Massnahmen] DO --> CHECK[CHECK Überprüfungsphase: Analytische Massnahmen] CHECK --> ACT[ACT Korrekturphase: Optimierung und Prozessanpassung] ACT --> PLAN </pre>	
<p style="text-align: center;">Organisatorische Massnahmen</p> <p>Administrative Massnahmen zur Einführung und Pflege eines QM-Systems.</p> <ul style="list-style-type: none"> • Teststrategie festlegen • Definition der Qualitätsziele • Messkriterien festlegen • Aufbau- und Ablauforganisation festlegen • Kompetenzen und Verantwortung festlegen • Qualitätsberichtswesen regeln 		
<p style="text-align: center;">Konstruktive Massnahmen</p> <p>Präventive Massnahmen zur Förderung von Qualität bei der Produktherstellung.</p> <ul style="list-style-type: none"> • technische Methoden und Werkzeuge • Richtlinien und Standards • Vorgehensmodell • Ausbildung • Dokumentation 	<p style="text-align: center;">Analytische Massnahmen</p> <p>Diagnostische Massnahmen zur Bestimmung der Qualität eines Produkts.</p> <ul style="list-style-type: none"> • Verifikation: Prüfung anhand der Ergebnisse aus der vorangehenden Phase, Übereinstimmung der Realisierung mit der Spezifikation, formal <ul style="list-style-type: none"> - statisch: Review (Produkt), Audit (Prozess) - dynamisch: Tests • Validierung/Validation: Fortlaufende Prüfung anhand der Anforderungen an das Produkt, Übereinstimmung der Lösung mit dem zu lösenden Problem, nicht formal (Abnahmetest) 	

Standards

QUAL

ISO

International Organization for Standardization

ISO 9001:
Standardisierung und Zertifizierung von Unternehmensprozessen

ISO 12207:
Erweiterung von ISO 9000
Rahmen für Beschaffung, Lieferung, Entwicklung, Betrieb und Wartung von IT-Systemen

ISO 15504 (Spice):
Software, Process, Improvement and Capability Determination
Kombination von ISO 12207 und CMM

CMM

Capability Maturity Model
Qualität des Software-Entwicklungsprozesses

Reifegrad-Stufe	Prozess	Kosten, Termine, Qualität
1. initial	ad hoc	nicht vorhersehbar
2. wiederholbar	intuitiv isoliert	schwanken, von Erfahrung abhängig
3. definiert	standardis., dokumentiert	nicht vollständig vorhersehbar
4. verwaltet	gesteuert	vorhersehbar, durch zuverlässige Metriken gesteuert
5. optimierend	rückgekoppelt	kontinuierliche Verbesserung durch Feedback-Verarbeitung

TMM

Test Maturity Model
Qualität des Software-Testprozesses

Reifegrad-Stufe	Merkmale
1. unsystematisch	Tests von Programmierer abhängig
2. organisiert	einzelne Projekte haben Testpläne
3. kostensenkend	Reviews, externe Testgruppen, Testplanmuster, stabile Software
4. systematisch	teilweise automatisierte Tests
5. optimiert	kontinuierliche Suche nach Verbesserung und Optimierung

ANSI / IEEE

American National Standards Institute
Institute of Electrical and Electronics Engineers (USA)

ANSI/IEEE 1012:
Standard für Software Verifikation und Validation

ANSI/IEEE 1059:
Guide for Software Verification and Validation Plans

ANSI/IEEE 1028:
Standard for Software Reviews and Audits

DIN

Deutsche Industrie-Norm

DIN 66285:
Qualitätsanforderungen und Prüfbestimmungen für Anwendungssoftware
geringe Bedeutung in der Praxis

EFQM

European Foundation for Quality Management
Business Excellence-Modell: Einbezug der Ergebnisse

Enabler-Faktoren (vom Management beeinflussbar):

- Führung (10%)
- Mitarbeiter (9 %)
- Politik und Strategie (8 %)
- Partnerschaft und Ressourcen (9 %)
- Prozesse (14 %)

Ergebnisse (nur über Enabler beeinflussbar):

- mitarbeiterbezogene Ergebnisse (9 %)
- kundenbezogene Ergebnisse (20 %)
- gesellschaftsbezogene Ergebnisse (6 %)
- Geschäftsergebnisse (15 %)

Qualitätsdimensionen

QUAL

ISO/IEC 9126

Qualitätsmodell mit sechs Kategorien zur Messung der Softwarequalität.

Verwendung:

- Anforderungsspezifikationen
- Evaluation von Softwarequalität

Funktionalität (Functionality)

Der Funktionsumfang des IT-Systems ist den Aufgaben angemessen und führt diese richtig und in der gewünschten Genauigkeit aus.

- Angemessenheit (Suitability): weder zuviel noch zuwenig
- Richtigkeit (Accuracy): korrekt und fehlerfrei
- Interoperabilität (Interoperability): Verknüpfung zu anderen Anwendungen
- Konformität (Compliance): Übereinstimmung mit dem Gesetz, mit internen Regeln und mit der Corporate Identity
- Sicherheit (Security): Schutz vor unberechtigtem Zugriff

Zuverlässigkeit (Reliability)

Das IT-System zeichnet sich durch Robustheit und Reife aus. Ausfälle kommen nur in seltenen Fällen und für kurze Zeit vor.

- Reife (Maturity): wenig Ausfälle
- Fehlertoleranz (Fault tolerance): Überbrückung von Fehlersituationen
- Wiederherstellbarkeit (Recoverability): letzter konsistenter Zustand

Benutzbarkeit (Usability)

Die Benutzer erlernen die Bedienung intuitiv und rasch. Anwenderfreundlichkeit und Bedienbarkeit werden durch GUI-Konformität sichergestellt.

- Verständlichkeit (Understandability): einheitlich, gemäss GUI-Standards
- Erlernbarkeit (Learnability): kontextsensitive Hilfetexte
- Bedienbarkeit (Operability): intuitiv

Effizienz (Efficiency)

Das System kann die unterschiedlichen Belastungsspitzen ohne nennenswerte Performanceeinbußen bewältigen.

- Zeitverhalten (Time behaviour): Online: Antwortzeiten < 3 Sekunden
- Verbrauchsverhalten (Resource behaviour): Batch: Ressourcenverbrauch

Wartbarkeit (Maintainability)

Auftretende Softwarefehler müssen innert angemessener Frist behoben werden. Sich ändernde Anforderungen sollten über Parametrierung oder Softwareerweiterungen vorgenommen werden können.

- Analysierbarkeit (Analysability): einfach zu verstehen
- Modifizierbarkeit (Changeability): einfach anzupassen, parametrierbar
- Stabilität (Stability): keine Nebenwirkungen bei Änderungen
- Prüfbarkeit (Testability): gut testbar

Übertragbarkeit (Portability)

Die Software soll einfach auf den Zielplattformen installierbar und wieder deinstallierbar sein. Ein Einsatz auf verschiedenen Betriebssystemplattformen ist in den meisten Fällen erwünscht.

- Anpassbarkeit (Adaptability): verschiedene Systemumgebungen
- Installierbarkeit (Installability): einfach installierbar und deinstallierbar
- Konformität (Conformance): gemäss Normen zur Portierbarkeit
- Austauschbarkeit (Replaceability): verschiedene Betriebssysteme

Statische Prüfverfahren

QUAL

Definition

Manuelle oder maschinelle inhaltliche Prüfung von Testobjekten.

Ziel: Fehler in einer möglichst frühen Phase zu erkennen und zu bereinigen.

Prüfaspekte:

- Vollständigkeit
- Lesbarkeit / Verständlichkeit
- formale Korrektheit
- inhaltliche / funktionale Korrektheit
- Einhalten von Standards

Vorteile / Nachteile

Vorteile:

- polyvalent, alle Entwicklungsergebnisse prüfbar
- Herstellungsweise wird mitgeprüft
- Förderung des allgemeinen Qualitätsgedankens
- bessere Einhaltung von Dokumentations- und Coding-Richtlinien
- einheitlichere Ergebnisse
- gegenseitiger Wissensaustausch
- gezielte Massnahmen durch statistische Auswertung

Nachteile:

- grosser Aufwand
- kann zu Konflikten führen

Informeller Review

- ad hoc auf Initiative des Autors
- Sitzung oder Zustellung mit der Bitte um Stellungnahme

Peer Rating:

- Zustellung mit Checkliste und Beurteilungsbogen

Walkthrough:

- gemeinsames Durchgehen des Prüfobjekts
- prüfobjektorientierte Checkliste
- Lösungsdiskussion

Round Robin Review

Gutachter versuchen, die anderen von der Qualität des Testobjekts zu überzeugen.

Technischer Review

formelles Verfahren:

- Projektleiter: Anordnung und Ressourcenplanung
- Moderator: Review-Planung, Einladung
- Autor: Verteilung des Prüfobjekts
- Gutachter: Vorbereitung mit Anforderungen / Richtlinien
- Alle: max. 2 Std. Gruppenmeeting
- Moderator: Check-, Verständnisfragen (sonst Abbruch)
- Autor: nur Fragenbeantwortung
- Alle: evtl. 3. Std. Lösungsdiskussion
- Protokollführer: Mängel-, Beschlussprotokoll
- Gutachter: Empfehlung über Freigabe
- Projektleiter: Entscheid über Freigabe
- Autor: evtl. Überarbeitung
- Moderator: Protokollversand, evtl. Planung zweiter Review

Inspektion

- Formelles Verfahren, analog Technischer Review, z.B. Fagan Code Inspection
- zusätzlich Einführungssitzung
- Checklisten mit Aspekten (z.B. Codekonventionen, Ressourcengebrauch, Schnittstellenkonformität) pro Gutachter
- Inspektions-Team ist Entscheidungsinstanz

Audit

Vom Management angeordnete Überprüfung der Einhaltung und Wirksamkeit von vorgegebenen Qualitätsregeln mittels Assessments (Dokumentenstudium, Interviews, Sitzungen).

- Systemaudit
- Prozessaudit
- Projektaudit

Definition

Tests an lauffähigen Testobjekten

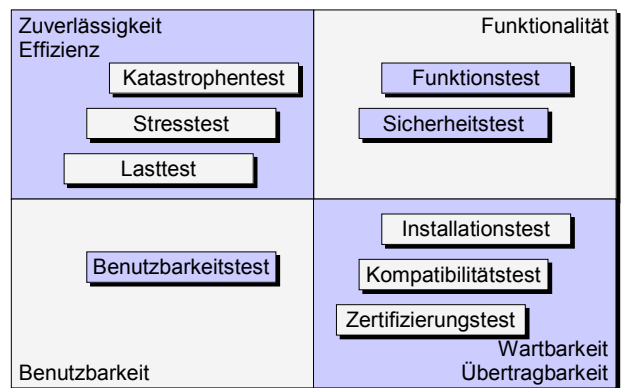
Vorteile:

- reproduzierbar
- Zielumgebung wird mitgeprüft
- Systemverhalten wird sichtbar gemacht

Nachteile:

- Ursache wird nicht aufgedeckt

Einteilung in Qualitätsdimensionen



Zuverlässigkeit / Effizienz

Katastrophentest:

Crashtest, Recoverytest, Simulation von Extremsituationen wie Stromausfall, Disc Crash, Netzwerkausfall, Test auf automatischen Wiederanlauf beim letzten konsistenten Zustand, organisatorische und bauliche Massnahmen

Stresstest:

Systemlast wird bis zum Zusammenbruch erhöht, Test auf regulierende Massnahmen, z.B. Eskalationsmeldungen ans Operating

Lasttest:

Volumentest, Performancetest, Antwortzeitverhalten, Transaktionsdurchsatz, Datendurchsatz, Skalierbarkeit

Funktionalität

Funktionstest:

Test der Use Cases auf Anforderungen, Vollständigkeit, Richtigkeit

Sicherheitstest::

- **Datensicherheit:**
Datenkonsistenz, Logging, Rollbackverfahren, Backupverträglichkeit
- **Datenschutz:**
Persönlichkeitsschutz, Passwortschutz, Verschlüsselungsmechanismen, Rollenkonzept, Firewalls, Ethical Hacking zum Sicherheitslücken aufdecken

Benutzbarkeit

Benutzbarkeitstest:

Softwareergonomie, GUI-Regeln, logische und intuitive Benutzerführung, einfache Erlernbarkeit, kontextsensitive Hilfe Benutzerhandbuch

Wartbarkeit / Übertragbarkeit

Installationstest:

unterschiedliche Betriebssysteme, Deinstallation, Upgrades

Kompatibilitätstest:

Hardware, Netzwerkkomponenten, Betriebssoftware, verknüpfte Anwendungssoftware, Normenkonformität

Zertifizierungstest::

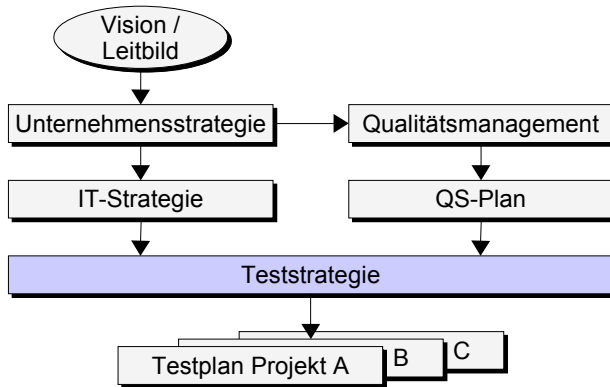
für bestimmte Softwarearten, Schnittstellen zu Standard-Hardware und -Betriebssystemen

Teststufen	QUAL
<p style="text-align: center;">Grafik Teststufen</p>	
<p style="text-align: center;">Unittest</p> <p>Laufende Prüfung abgrenzbarer Funktionseinheiten während der Implementation.</p> <p>Testschwerpunkte:</p> <ul style="list-style-type: none"> • funktionale Korrektheit und Vollständigkeit • korrekte Schnittstellen • extrapolierte Performanceaspekte 	<p style="text-align: center;">Integrationstest</p> <p>Prüfung des Zusammenspiels der einzelnen Komponenten und der Schnittstellen zu den umgebenden Systemen.</p> <p>Testschwerpunkte:</p> <ul style="list-style-type: none"> • Komponentenintegration • Schnittstellenintegration • Benutzerschnittstelle • Hardwarekompatibilität • Betriebssystemverträglichkeit • Netzwerkverträglichkeit • Durchsatz • Sicherheitsaspekte
<p style="text-align: center;">Systemtest</p> <p>Prüfung des gesamten Systems auf bestimmte Qualitätsmerkmale.</p> <p>Testausführung:</p> <ul style="list-style-type: none"> • Testspezialisten • Netzwerkspezialisten • Systemspezialisten • Anwender 	<p style="text-align: center;">Abnahmetest</p> <p>Prüfung kritischer Bereiche und vorhandener Dokumente durch Vertreter des Auftraggebers.</p> <p>Grundlagen:</p> <ul style="list-style-type: none"> • schriftlich festgehaltene Abnahmekriterien • protokollierte Systemtests
<p style="text-align: center;">Regressionstest</p> <p>Durchführung eines Subsets der Systemtests für ein neues Softwarerelease zur Vermeidung von Side-Effects.</p> <p>Testschwerpunkte:</p> <ul style="list-style-type: none"> • risikoorientierte Auswahl der Testfälle • hoher Grad an Automatisierung <p>Smoke-Test: Subset mit den wichtigsten automatisch ablaufenden Tests für Zwischenreleases.</p>	<p style="text-align: center;">Live-Test</p> <p>Prüfung im produktiven Betrieb.</p> <ul style="list-style-type: none"> • Parallelbetrieb: neues System läuft parallel zum alten aufwändig, nur für kritische Projekte • Pilotbetrieb: Vorabführung in einer Abteilung oder Filiale nur wenn keine Integrationsschwierigkeiten bestehen • Beta-Test: Test einer Vorabversion durch einzelne Benutzer

Testmethoden		QUAL
<p align="center">Definition Testmethode</p> <p>Planmässiges, auf einem Regelwerk aufbauendes Vorgehen zur Herleitung oder Auswahl von Testfällen.</p> <p>methodisches Vorgehen:</p> <ul style="list-style-type: none"> • Abdeckung der Fälle ist berechenbar • Testfortschritt ist messbar <p>exploratives Vorgehen:</p> <ul style="list-style-type: none"> • aufgrund persönlicher Erfahrung • vermehrt bei Fehlerkonzentration • Error Guessing: intuitiv • Guerrilla-Testen: alles Nicht-Erlaubte 	<p align="center">Whitebox Test</p> <p>Glassbox Test, Strukturtest, beim Unittest aufgrund der inneren Struktur eines Programms</p> <p>Testabdeckungsgrad:</p> <ul style="list-style-type: none"> • Anweisungsabdeckung: jede Anweisung wird mindestens einmal ausgeführt • Zweigabdeckung: jeder Programmzweig wird mindestens einmal durchlaufen, true und false, Schleifen mindestens einmal • Bedingungsabdeckung (Termabdeckung): pro Bedingung alle möglichen logischen Kombinationen • Pfadabdeckung (in der Regel nicht möglich): jeder mögliche Pfad von Anfang bis zum Schluss, Schleifen 0, 1 und mehrmals 	
<p align="center">Greybox Test</p> <p>beim Integrationstest einzelne Programme als Blackbox Zusammenspiel der Komponenten als Whitebox</p> <p>Testabdeckungsgrad:</p> <ul style="list-style-type: none"> • Komponentenabdeckung: jede Komponente mindestens einmal getestet • Ablaufabdeckung: alle möglichen Programmverkettungen 	<p align="center">Blackbox Test</p> <p>Funktionstest, beim Systemtest aufgrund des erwarteten Funktionsumfangs und der definierten Schnittstellen</p> <p>Testabdeckungsgrad:</p> <ul style="list-style-type: none"> • Funktionsüberdeckung • Eingabeüberdeckung: gemäss Typ, Länge, Wertebereich • Ausgabeüberdeckung <p>Testfallermittlung:</p> <ul style="list-style-type: none"> • Use Cases (Anwendungsfälle) • Äquivalenzklassenbildung • Grenzwertanalyse • zustandsbezogen: Zustandsdiagramm, Übergangsbaum 	
<p align="center">Äquivalenzklassenbildung</p> <p>Bildung repräsentativer Testfälle aus Daten mit identischem funktionalem Verhalten.</p> <p>Ziel: hohe Fehlerentdeckungswahrscheinlichkeit mit minimaler Anzahl Testfälle</p> <ul style="list-style-type: none"> • gültige Äquivalenzklassen: Funktionstests • ungültige Äquivalenzklassen: Stabilitätstests z.B. negative Ganzzahlen, Dezimalzahlen, Null, Blank, alphabetische Werte, Sonderzeichen, keine Eingabe 	<p align="center">Grenzwertanalyse</p> <p>Ergänzung zur Äquivalenzklassenbildung, Testfälle für die Werteübergänge.</p> <p>Grund: Randbereiche sind besonders fehlerintensiv</p> <ul style="list-style-type: none"> • erster / letzter Wert in einer Äquivalenzklasse • erster / letzter Wert ausserhalb Äquivalenzklasse 	
<p align="center">Diversifizierende Methoden</p> <ul style="list-style-type: none"> • Back-to-Back-Test: Vergleich unabhängiger Realisierungen aufgrund derselben Spezifikation • Mutationstest: Bewusster Einbau von Fehlern zur Ermittlung von schwer zugänglichen Fehlverhalten • Error Seeding: Bewusster Einbau von Fehlern zur Überprüfung der Qualität des Testteams 		

Testtools	QUAL
<p style="text-align: center;">Definition</p> <p>Computer Aided Software Testing CAST: Durch Software unterstütztes und automatisiertes Testen</p> <p>Einsatz lohnt bereits ab der dritten Wiederholung</p>	
<p style="text-align: center;">Tools zur Testprozessunterstützung</p> <p>Testmanagementsoftware:</p> <ul style="list-style-type: none"> • Überblick über Arbeitsfortschritte • Verwaltung der Testfälle und Problemmeldungen <p>Testinformations-Repository:</p> <ul style="list-style-type: none"> • Input-Datenbestände • Dokumente, Protokolle, Berichte • Informationsaustausch 	<p style="text-align: center;">Tools für statische Prüfungen</p> <p>Static Measurement Tools</p> <p>Code Analyzer:</p> <ul style="list-style-type: none"> • Syntaxfehler • statistische Werte: Verschachtelungstiefe, Codezeilen • Konformität mit Programmierrichtlinien • unerreichbarer Programmcode • nicht deklarierte oder falsch verwendete Variablen <p>Complexity Analyzer:</p> <ul style="list-style-type: none"> • Messwerte zu Anzahl Parameter, Variablen, Bedingungen, Schleifen usw.
<p style="text-align: center;">Tools zur Testvorbereitung</p> <ul style="list-style-type: none"> • Spezifikationsbasierte Testdatengeneratoren: aus formal definierten Anforderungen aus CASE-Tools, mit Solldaten • Codebasierte Testdatengeneratoren: aus Programmcode, nach gewünschtem Abdeckungsgrad Nachteil: nicht implementierte Funktionen werden nicht entdeckt • Schnittstellenbasierte Testdatengeneratoren: aus Interfaces (Parameter, Eingabefelder), mittels Äquivalenzklassen- und Grenzwertanalyse • Massendaten-Generatoren: für Performancetests 	<p style="text-align: center;">Tools zur Testdurchführung</p> <ul style="list-style-type: none"> • Stubs (Dummies, Teststümpfe): für Top down-Verfahren • Drivers (Treiberprogramme): für Bottum up-Verfahren • Simulatoren: simulieren Zielsystem • Capture- / Playbacktools (Testroboter): Eingaben werden einmal aufgezeichnet und nachher laufen gelassen • Performance Analyzer: Statistik über Systemverhalten • Coverage Monitor: prüft Abdeckungsgrad • Debugger: für schwierig zu erzeugende Testsituationen • Testresult Comparator: für Massentests, erstellt Abweichungsbericht

Teststrategie im Unternehmen



Inhalt Teststrategie

- Anwendung der Richtlinien in Projekten
- Rollen mit AKV
- Workflow des Testprozesses im Vorgehensmodell
- Testmethoden für Testfall- und Testdatenauswahl
- Testarten pro Phase und Testobjekt
- Automatisierungssoftware
- Anforderungen an das Testberichtswesen
- Unternehmensinternes QS- und Testglossar

Testverfahren

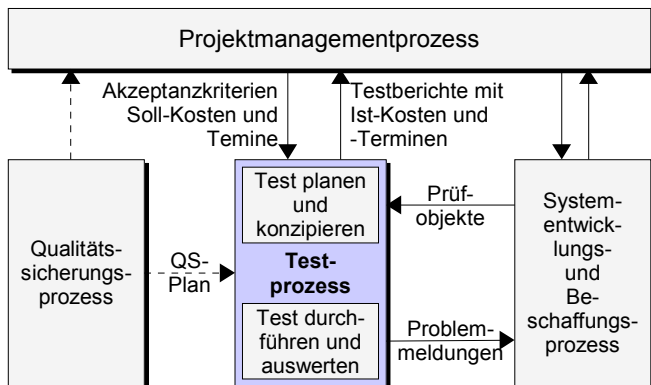
inhaltliche Vorgehensweise der Testaktivitäten
v.a. beim Integrationstest, Integrationsstrategien

- Top down: für Prototypen, einzelne Anwendungsfälle, in frühem Stadium, braucht Stubs
- Bottom up: für Module, bestehende Funktionalität, in späterem Stadium, braucht Drivers
- Big Bang: gesamtes System, in separater Testphase
- ad hoc: nach Verfügbarkeit
- hardest first: risikoorientiert, Priorisierung komplexer und risikoreicher Komponenten
- funktionsorientiert
- geschäftsprozessorientiert

Prüfobjekte

- Anforderungsspezifikationen:
Use-Case-Modelle, Prozessmodelle, Datenmodelle
- Design:
Eingabemasken, Architektur, Prozessbeschreibungen
- Programme im Source Code:
Modul, Klasse, Komponente, Programm, Applikation
- Lauffähige Anwendungssoftware:
inkl. Benutzer- und Betriebshandbücher,
Installationsanweisungen
- Betriebs- und Systemsoftware
- Datenbankmanagementssoftware DBMS
- Hardware: Workstation, Server, Netzwerkkomponenten
- Netzwerk: LAN, WAN, VPN

Testprozess im Unternehmen

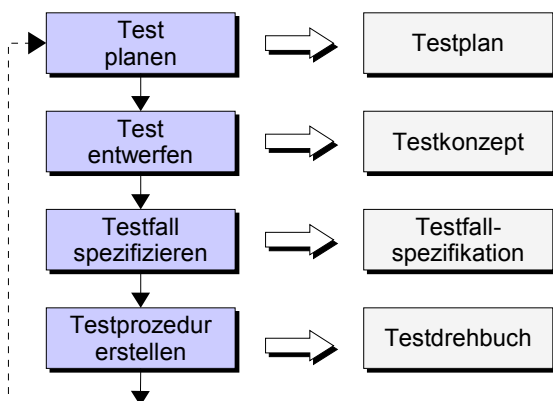


ANSI/IEEE 829

Standard für Software Test Dokumentation

<http://www.softwaretesting.de/article/view/1/1/7/>
<http://www.coleyconsulting.co.uk/IEEE829.htm>

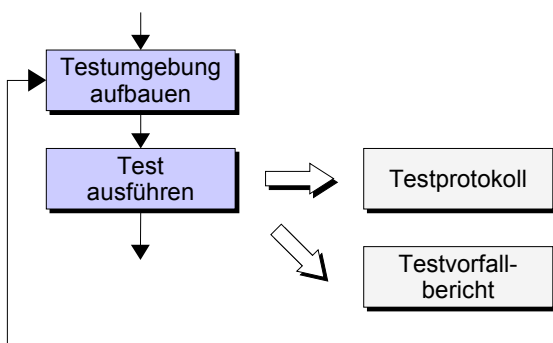
Testvorbereitung



Preparation of Tests

- Test plan: plan how the testing will proceed
- Test design specification: decide what needs to be tested
- Test case specification: create the tests to be run
- Test procedure specification: describe how the tests are run
- Test item transmittal report: Specify the items released for testing (Regelung der Software-Übergabe)

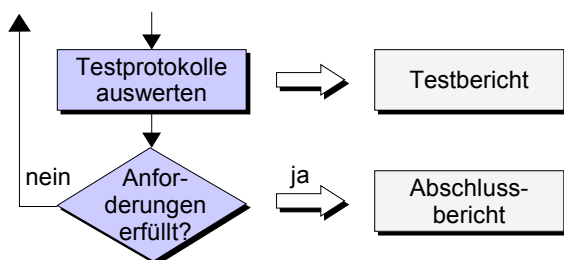
Testdurchführung



Running the tests

- Test log: record the details of tests in time order
- Test incident report: record details of events that need to be investigated

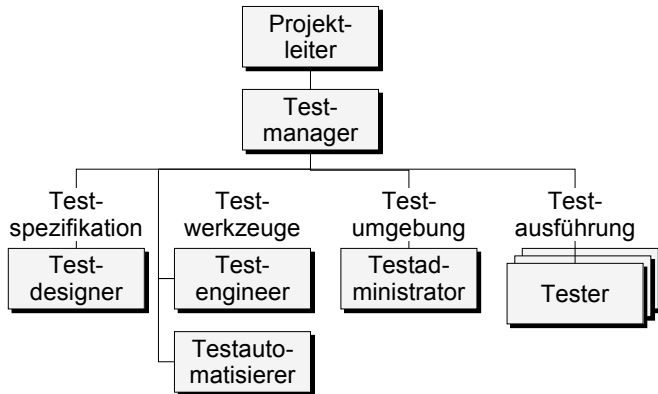
Testauswertung



Completion of testing

- Test summary report: summarise and evaluate tests

Organigramm Testteam



Testmanager

- führt die Mitarbeiter des Testteams
- erstellt (parallel zum Design) den Testplan
- plant, überwacht und steuert den Testprozess
- publiziert empfangergerecht den Testfortschritt
- rapportiert periodisch an den Projektleiter
- wertet die Testresultate aus (Triage, Aussortierung)
- schlägt Korrekturmaßnahmen vor
- erstellt den Testabschlussbericht

Testdesigner

- erstellt (parallel zum Design) die Testspezifikationen: Testkonzept, Testfallbeschreibungen, Testdrehbücher
- erstellt Vorgaben für den Testengineer
- erstellt Arbeitsanweisungen für den Testautomatisierer und den Testadministrator
- instruiert die Tester

Testengineer

- erstellt Testsoftware anhand der Spezifikationen des Testdesigners
- erstellt automatisierte Testskripts

Testautomatisierer

- parametriert die Testwerkzeuge (Testtools)
- erstellt Skripts in der Toolumgebung
- instruiert den Testadministrator in Bezug auf die Installation und den Betrieb
- instruiert die Tester über den Gebrauch

Testadministrator

- erstellt die Testumgebung
- installiert die notwendigen Testwerkzeuge
- installiert Testgeschirr (Umgebungssoftware: Testtreiber und Teststümpfe)
- definiert Benutzerrechte
- stellt Testdaten bereit
- stellt Testobjekte bereit
- stellt Messgeräte und Analysetools bereit
- sichert den Betrieb

Tester

- führt die Tests gemäss Testspezifikationen aus
- erstellt das Testprotokoll
- erstellt Testvorfallberichte

Testdokumente		QUAL
<p style="text-align: center;">Verwaltung Testdokumente</p> <ul style="list-style-type: none"> zentrale projektbezogene Ablage Standard-Templates und Mustervorlagen bereitstellen eindeutige Identifikation aller Dokumente Versionenführung Testdokumentations-Datenbank für anfallende Informationen während des Testens 	<p style="text-align: center;">Vorteile Testdokumentation</p> <ul style="list-style-type: none"> Planung und Steuerung Förderung von Kommunikation / Information Vermeidung von Missverständnissen Absicherung des Testteams gegenüber Auftraggeber Nachvollziehbarkeit von Fehlern 	
<p style="text-align: center;">Testplan (Prüfplan)</p> <ul style="list-style-type: none"> Übersicht Testobjekte des Projekts mit Referenzierung QS-Plan, Teststrategie, Projektplan, Anforderungsdokumente, Richtlinien zu testende und nicht zu testende Komponenten und Funktionen mit Referenzierung Designdokumente Testvorgehen mit gewünschtem Qualitätsstandard allgemeine Pass- und Fail-Kriterien, Testendekriterien, Testabbruchkriterien zu erstellende Dokumente Testumgebung und Testwerkzeuge (evtl. zu beschaffen) Verantwortlichkeiten und Testorganisation Zeitplan mit Milestones Risiken mit Notfallmassnahmen Genehmigungen durch Unterschrift 	<p style="text-align: center;">Testkonzept</p> <ul style="list-style-type: none"> Auflistung Testobjekte mit Referenzierung Testplan zu testende Funktionen mit Referenzierung Anforderungs- und Designdokumente konkrete Testverfahren pro Funktion Auflistung der Testfälle Zusammenstellung zu Testszenarien detaillierte Pass- und Fail-Kriterien, Akzeptanzkriterien für zu testende Qualitätsdimensionen, Kriterien für Testabbruch und Testfortsetzung 	
<p style="text-align: center;">Testfallspezifikation</p> <p>Pro Testfall:</p> <ul style="list-style-type: none"> Beschreibung Testobjekte und Funktionen mit Referenzierung Testkonzept / Designdokumente Eingabedaten mit Referenzierung Speicherort erwartete Ausgaben (Testorakel) mit Referenzierung Anforderungsdokumente benötigte Testumgebung Vorbedingungen, spezielle Ausführungsvorschriften Abhängigkeiten zu anderen Testfällen 	<p style="text-align: center;">Testdrehbuch</p> <p>Testskript, Testprozedur, möglichst wiederverwendbar</p> <p>Pro Testszenario (Gruppierung von Testfällen):</p> <ul style="list-style-type: none"> Beschreibung mit Testziel und Referenzierung Testfälle testfallübergreifende Voraussetzungen, Anforderungen an Tester und Testumgebung Einzelschritte inkl. Vorbereitung, Start, Durchführung, Prüfkriterien, Verhalten bei Abbruch, Neustart, Stopp, Abschlusskriterien, Aufräumen, Unvorhergesehenes Anweisungen zur Protokollierung / Logführung 	

Testprotokoll

Pro Testszenario (gemäss Testdrehbuch):

- Identifikationsdaten: Datum und Uhrzeit, Tester, Testumgebungsparameter, Softwarerelease
- Übersicht getestete Fälle mit passed / failed
- allgemeiner Eindruck des Testers
- pro Testfall durchgeführte Schritte und gelieferte Resultate
- vorgenommene Änderungen in der Testumgebung
- unerwartete Ereignisse
- Referenzierung Testvorfallbericht (Problemmeldungen)
- evtl. Referenzierung automatisch erstelltes Testlog

Testvorfallbericht / Problemmeldungen

- Problembeschreibung: Identifikationsdaten und durchgeführte Schritte analog Testprotokoll
- Fehlverhalten: erwartetes und eingetretenes Ergebnis
- Fehlerwirkung: Auswirkungen
- Hinweise zur Reproduzierbarkeit des Fehlers
- evtl. Workaround
- Vermutungen über Fehlerursache
- Einschätzung und Gewichtung

Testbericht

Periodische Dokumentation des Testfortschritts an die Projektleitung.

- allgemeine Lage
- spezielle Vorkommnisse
- Ist-Werte, Soll-Werte, Abweichungen (mittels Testmetriken)
- vorgeschlagene Massnahmen
 - Testunterbruch
 - Anpassung der Ressourcen
 - Terminverschiebung

Testabschlussbericht

Pro Teststufe:

- Kurzbeschreibung der getesteten Objekte mit Referenzierung aller vorgängigen Dokumente
- Ressourcenverbrauch
- Zusammenfassung der Testergebnisse
- Abweichungen mit Begründung
- Bewertung in Bezug auf die gewünschten Qualitätskriterien mit Einschätzung der Fehlerrisiken
- Freigabeempfehlung für nächste Stufe
- Genehmigungen durch Unterschrift

Testmetriken

- Testeffektivität (Testabdeckungsgrad): Coverage Monitor für Whitebox-Tests, Anwendungsfallabdeckung, Komponentenabdeckung
- Testeffizienz (Testwirksamkeit): Verhältnis zwischen Testaufwand (Kosten, Ressourcen, Zeit) und Testnutzen (gefundene Fehler)
- Fehlertrendstatistik (Severity Trend): Testeffizienz-Diagramm mit Schweregrad-Klassifizierung
- Fehlerdichte: Verhältnis zwischen der Anzahl Fehler und der Grösse des Systems (in Codezeilen)
- Testfortschrittsübersicht: Zusammenfassung der Soll- und Ist-Fertigstellungswerte

Fehlertrendstatistik

Pareto-Prinzip: 80% der Fehler mit 20 % Testaufwand finden

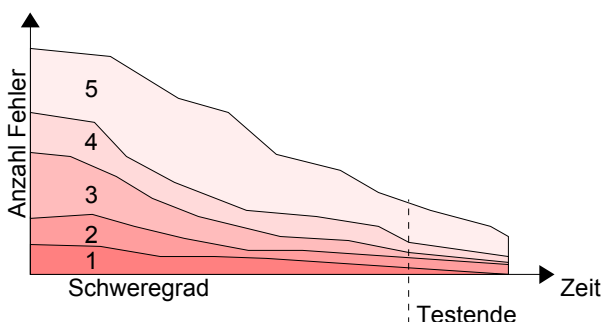
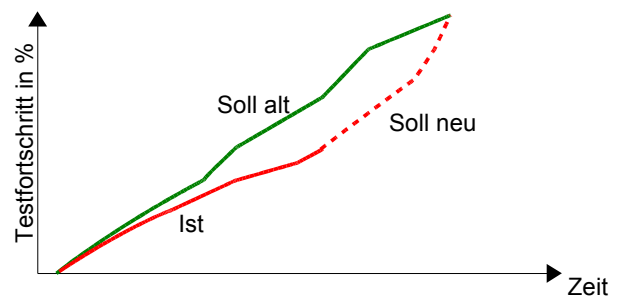


Diagramm Testfortschritt



Problembearbeitung

QUAL

Ablauf Problembearbeitung

- Tester macht eine erste Einschätzung
- Testmanager nimmt Klassifizierung nach Problemart, Schweregrad und Priorität im Hinblick auf das gesamte Projekt vor
- Testmanager weist Problemstatus zu
- Problemstatus wird je nach Bearbeitungsschritt von der zuständigen Person nachgeführt

Problemart

Fehler	Lösung
Fehlbedienung	Benutzerdokument anpassen
Fehler in den Testdaten / Testumgebung	Testumgebung anpassen
Softwarefehler	Fehler in der Umsetzung beheben
Konzeptionsfehler	Fehler in der Konzeption beheben
Fehlende / geänderte Anforderung	Change Request an das Change Management

Schweregrad

- 1 Show-Stopper, keine weiteren Tests möglich
- 2 Wesentliche Funktionen sind fehlerhaft, System ist nur mit grossen Einschränkungen einsetzbar
- 3 Funktionale Abweichung, System kann mit Einschränkungen genutzt werden
- 4 Geringfügige Abweichung
- 5 Schönheitsfehler

Priorität

- A Höchste Priorität, Lösung ist im nächstmöglichen Patch (ausserplanmässige Softwarekorrektur) zu liefern
- B Mittlere Priorität, Lösung ist im nächsten regulären Softwarerelease zu liefern
- C Niedrige Priorität, Lösung ist bei Gelegenheit zu liefern

Problemstatus

- Neu: Einschätzung aus Sicht des Testers
- Offen: Klassifizierung durch Testmanager
- Analyse: Ermittlung der Fehlerursache durch Entwickler
- Beobachtung: Warten auf weitere Infos
- Korrektur: Fehlerkorrektur durch Entwickler
- CR-Antrag: durch Testmanager, bei Anforderungsfehlern
- Abgewiesen: durch Testmanager oder CR-Board
- Change Request: Freigabe CR-Board zur Realisierung
- Testwiederholung: Freigabe Entwickler nach Unittest
- Wieder Offen: ungenügendes Testresultat
- Erledigt: Testresultat ok

