

4.	EDV-Systementwicklung	4
4.1	Projektmanagement von Informatikprojekten	4
4.1.1	Spezielle Aspekte der Projektorganisation 2	4
4.1.1.1	Aufgabenträger	4
4.1.1.2	Konstituieren Projektorganisation und Ausserkraftsetzen	4
4.1.1.3	Instanzen im Projektablauf	4
4.1.2	Aufgaben und Verantwortungen des Organisators 2	4
4.1.2.1	Formulierung Auftrag/Initialisierung Projekt	4
4.1.2.2	Aufwandschätzungen/Ressourcenplanung	4
4.1.2.3	Termin- und Budgetplanung	4
4.1.2.4	Berichtswesen	4
4.1.2.5	Überwachung und Kontrolle	4
4.1.3	Spezielle Aspekte der Vorgehensmodelle 2	4
4.1.3.1	Empirisches Vorgehensmodell	4
4.1.3.2	Phasen des Organisationsprozesses, Organisationszyklus	4
4.1.3.3	Prototyping, Varianten, Versionen	4
4.1.3.4	Top-Down-Approach	4
4.1.3.5	Evaluation von Standardsoftware	4
4.1.4	Zentrale/Denzentrale Informatikprojekte	4
4.1.4.1	Entwicklung, Betrieb, Unterhalt von Verarbeitung und Datenhaltung	4
4.1.5	Projektdokumentation 2	5
4.1.5.1	Allgemeine Dokumentation	5
4.1.5.2	Phasenbezogene Dokumentation	5
4.2	Informatik-Controlling, Qualitätssicherung und EDV-Revision	5
4.2.1	Informatik-Controlling	5
4.2.1.1	Definition (Abgrenzung zu Qualitätssicherung und EDV-Revision)	5
4.2.1.2	Standardisierung	5
4.2.1.3	Normen und Messgrößen	5
4.2.1.4	Planungs-, Steuerungs- und Kontrollinstrumente	5
4.2.1.5	Eingliederung in Informatik-Strategie und Leitbild	5
4.2.1.6	Methoden des Informatik-Controlling	5
4.2.1.7	(Alternativensuche, Bewertung, Projektplanung, Kontrolle, Accounting, Monitoring, Benchmarking)	5
4.2.2	Qualitätssicherung 1	5
4.2.2.1	Metriken	5
4.2.2.2	Software-Ergonomie	5
4.2.2.3	Effektivität	5
4.2.2.4	Qualitätsmerkmale von Daten und Dokumentationen (Änderbarkeit, Aktualität, Vollständigkeit...)	5
4.2.2.5	Projektplan	5
4.2.2.6	Freigabewesen	5
4.2.2.7	Gestaltung von Support und Schulung	5
4.2.2.8	Qualitätsprüfung, (Reviews, Tests, Konfigurationsmanagement, Verifikation, Validierung)	5
4.2.2.9	Versionen-Verwaltung	6
4.2.3	EDV-Revision 1	6
4.2.3.1	Methoden und Massnahmen (Prüfpfadkonzept, Hilfsmittel, Tools)	6
4.3	Standardisierung 3	6
4.3.1	Ziele der Standardisierung	6
4.3.1.1	Verinfachung	6
4.3.1.2	Kontinuität	6
4.3.1.3	Einheitlichkeit	6
4.3.1.4	Austauschbarkeit (Bausteine, Module, Komponenten)	6
4.3.1.5	Investitionsschutz	6
4.3.2	Standardisierungsgebiete 3	6
4.3.2.1	Qualitätssicherung / Controlling	6
4.3.2.2	Telekommunikation	6
4.3.2.3	Programmierung	6
4.3.2.4	Testorganisation	6
4.3.2.5	Methoden/Verfahren	6
4.3.2.6	EDV-Produktion	6
4.3.2.7	Dokumentation	6

4.3.3	Standards für die Informatik 3.....	6
4.3.3.1	Vorgehensmodell.....	6
4.3.3.2	Projektbearbeitung	6
4.3.3.3	Entwicklungsumgebung.....	6
4.3.3.4	Hardware-Standard	6
4.3.3.5	Software-Stnadard.....	6
4.3.3.6	Applikationsübergreifend (z.B. EDIFACT).....	6
4.4	System-Design.....	6
4.4.1	Design-Methoden, Design-Techniken 1	6
4.4.1.1	CASE	6
4.4.1.2	Workflow (Geschäftsprozess, funktions- und objektorientiert)	6
4.4.1.3	Daten-Design und Informationsfluss	6
4.4.1.4	Normalisierung, Kommunikations-Analyse.....	6
4.4.2	Lösungsformen 1	7
4.4.2.1	Batch.....	7
4.4.2.2	Dialog.....	7
4.4.2.3	Zentral.....	7
4.4.2.4	Dezentral	7
4.4.2.5	DDP (Distributed Data Processing).....	7
4.4.2.6	CSA (Client/Server-Architektur).....	7
4.4.2.7	IC (Information Center-Konzept, Datenextrakt).....	7
4.4.2.8	Enduser-Computing.....	7
4.4.3	Bearbeitungsgrundsätze 1	7
4.4.3.1	Hardest First	7
4.4.3.2	Top Down	7
4.4.3.3	Bottom Up.....	7
4.4.3.4	Middle Out	7
4.5	Daten-Speicherung und -Verwaltung.....	7
4.5.1	Daten-Management 1	7
4.5.1.1	Datenspeicherung (zentral, verteilte)	7
4.5.1.2	CSA (Client-Server-Architektur)	7
4.5.1.3	Datenbank-Modelle	7
4.5.1.4	Datenbanktheorie	7
4.5.2	Daten-Administration 1	7
4.5.2.1	Data-Dictionary, Repository.....	7
4.5.2.2	Datennutzung	7
4.5.2.3	Speicherungsformen und Zugriffsmethoden	7
4.5.2.4	Auswertugen.....	7
4.5.2.5	Reorganisation	7
4.5.2.6	Abfragesprachen	7
4.6	Applikationsentwicklung	7
4.6.1	Programmorganisation 1	7
4.6.1.1	Denken in Prozessen	7
4.6.1.2	Ablaufstrukturen (Dialog- und Batchdesign).....	7
4.6.1.3	Datenstrukturen	7
4.6.1.4	Normierte Programmierung.....	7
4.6.1.5	Programm- und Modulbildung	7
4.6.1.6	Interprogramm-Kommunikation, Schnittstellen	7
4.6.2	Programmiersprachen, Programmierhilfen 1.....	8
4.6.2.1	Arten und Aufbau.....	8
4.6.2.2	Einsatzgebiete, Vorteile und Nachteile.....	8
4.6.2.3	Compiler	8
4.6.2.4	Interpreter	8
4.6.2.5	Generatoren.....	8
4.6.2.6	Macroprozessoren.....	8
4.6.2.7	Viert-/Fünfte-Generationssprachen	8
4.6.2.8	Objektorientierte Programmierung	8
4.6.3	Programmbeschreibung 1	8
4.6.3.1	Kurzbeschreibung.....	8
4.6.3.2	Programmablaufplan	8
4.6.3.3	Strukturierte/formalisierte Beschreibung	8
4.6.4	Standardisierte Applikationssoftware	8

4.6.4.1	Einsatzgebiete/Beurteilungskriterien 1	8
4.6.4.2	Auswahl- und Bewertungsverfahren 2	8
4.6.5	Applikationsschnittstellen 1	8
4.6.6	Applikationsdokumentationen 1	8
4.7	Test	8
4.7.1	Ziel und Zweck 2	9
4.7.1.1	Für Analyse, Programmierung und Betriebsorganisation	9
4.7.1.2	Für das Rechenzentrum	9
4.7.1.3	Für den Benutzer	9
4.7.2	Testarten 2	9
4.7.2.1	Review/Walk Through	9
4.7.2.2	Modultest/Programmeinzeltest	9
4.7.2.3	Code-Inspection (z.B. Performance-Aspekte)	9
4.7.2.4	Schreibtischtest (Trockentest)	9
4.7.2.6	Integrationstest (technischer und betrieblicher)	9
4.7.2.7	Systemtest (Verbundtest, Regressionstest)	9
4.7.2.8	Katastrophentest	9
4.7.2.9	Abnahmetest	9
4.7.2.10	Parallellauf	10
4.7.2.11	Pilotinstallation	10
4.7.3	Testdaten, Testfälle 2	10
4.7.3.1	Auswahl und Beschaffung (wer, was)	10
4.7.3.2	Testfallbestimmung wie z.B. Reihenfolge	10
4.7.3.3	Normalfälle, Spezialfälle, Grenzwerte (inhaltlich und volumenbezogen)	10
4.7.4	Testverfahren, Testvorgehen 2	10
4.7.4.1	Prinzipien der Teilung, der Wiederholung, der Messbarkeit	10
4.7.4.2	Teststrategien	10
4.7.4.3	Testplanung, -verarbeitung, -durchführung, -ausführung, -abnahme, -verwaltung..	10
4.7.5	Testdokumentation 2	10
4.8	Einführung/Schulung	10
4.8.1	Einführungsvorgehen 3	10
4.8.1.1	Formen (schlagartig, stufenweise, parallel)	10
4.8.1.2	Vorbereitung	10
4.8.1.3	Betreuung der Benutzer	10
4.8.2	Schulungsplanung und -organisation 3	10
4.8.2.1	Zielpublikum	10
4.8.2.2	Methode (Unterricht, schriftlich, on the job)	10
4.8.2.3	Vorbereitung 8Aufbau, Administration, Hilfsmittel usw.)	10
4.8.3	Benutzerhandbuch 3	10
4.8.3.1	Inhalt	10
4.8.3.2	Art der Speicherung, Medien	10
4.8.3.3	Zugriff, Verfügbarkeit	10
4.8.3.4	Verwaltung und Archivierung	10
4.9	Betrieb/EDV-Produktion	10
4.9.1	Rechenzentrum 1	11
4.9.1.1	Organisationsformen	11
4.9.1.2	Aufbauorganisation	11
4.9.1.3	Ablauforganisation	11
4.9.1.4	Hardwarebedienung und -betreuung (Steuerung und Überwachung)	11
4.9.1.5	Betriebsplanung und -abwicklung	11
4.9.1.6	Systemverwaltung und -betreuung	11
4.9.1.7	Schnittstelle zu Fachbereichen und Applikationsentwicklung	11
4.9.1.8	Infrastruktur	11
4.9.1.9	Sicherheit	11
4.9.1.10	RZ-Dokumentaion	11
4.9.2	Planung, Accounting und Berichtswesen 1	11
4.9.2.1	Volumenentwicklung, Leistungsbedarf, Nutzkapazität	11
4.9.2.2	Leistungserfassung, Preisfindung, Verrechnung	11
4.10	Unterhalt der Applikation	11
4.10.1	Applikationserhaltung 2	11
4.10.1.1	Applikations-Controlling	11
4.10.1.2	Organisationsregelungen	11

4.10.1.3	Dokumentation	11
4.10.1.4	Benutzerergonomie, Anwendbarkeit	11
4.10.1.5	Betriebliche Gegebenheiten	11
4.10.1.6	Release Manangement	11
4.10.1.7	Migration, Konversion (Aufwärtskompatibilität, Rollback)	11
4.10.2	Nachkalkulation und Ausserkraftsetzung 2	11
4.10.2.1	Projekt abrechnung	11
4.10.2.2	Leistungsnachweis und -verrechnung	11
4.10.2.3	Ausserkraftsetzungsplanung	11
4.10.2.4	Benutzerbetreuung und -berechtigung	11
4.10.2.5	Gesetzeskonformität	11
4.10.2.6	Revisionsanforderungen	11
4.10.2.7	Aufbewahrung	11
4.10.2.8	Physikalische Lösungen	11
4.11	Datensicherheit und Datenschutz	11
4.11.1	Risikobereiche 2	11
4.11.1.1	Benutzer	11
4.11.1.2	EDV (Entwicklung, Produktion und Wartung)	11
4.11.1.3	Kommunikationsnetz	11
4.11.1.4	Kontrollmechanismen (Authentifikation, Ownership)	11
4.11.2	Sicherheitsmassnahmen, Kontrollmechanismen 2	12
4.11.2.1	Sicherstellungs-Konzept	12
4.11.2.2	Technische Massnahmen (Logging, Restart, Recovery, Check Points etc.)	12

4. EDV-Systementwicklung

4.1 Projektmanagement von Informatikprojekten

4.1.1 Spezielle Aspekte der Projektorganisation 2

4.1.1.1 Aufgabenträger

4.1.1.2 Konstituieren Projektorganisation und Ausserkraftsetzen

4.1.1.3 Instanzen im Projektablauf

4.1.2 Aufgaben und Verantwortungen des Organisators 2

4.1.2.1 Formulierung Auftrag/Initialisierung Projekt

4.1.2.2 Aufwandschätzungen/Ressourcenplanung

Methoden der Aufwandschätzung :

- Algorithmische Methoden
- Vergleichsmethoden
- Kennzahlenmethoden

4.1.2.3 Termin- und Budgetplanung

4.1.2.4 Berichtswesen

4.1.2.5 Überwachung und Kontrolle

4.1.3 Spezielle Aspekte der Vorgehensmodelle 2

4.1.3.1 Empirisches Vorgehensmodell

4.1.3.2 Phasen des Organisationsprozesses, Organisationszyklus

4.1.3.3 Prototyping, Varianten, Versionen

4.1.3.4 Top-Down-Approach

4.1.3.5 Evaluation von Standardsoftware

4.1.4 Zentrale/Denzentrale Informatikprojekte

4.1.4.1 Entwicklung, Betrieb, Unterhalt von Verarbeitung und Datenhaltung

4.1.5 Projektdokumentation 2

4.1.5.1 Allgemeine Dokumentation

Anforderung an Dokumentationen:

änderbar / aktuell / eindeutig / indentifizierbar / normenkonform / verständlich / vollständig / widerspruchsfrei / etc

4.1.5.2 Phasenbezogene Dokumentation

4.2 Informatik-Controlling, Qualitätssicherung und EDV-Revision

4.2.1 Informatik-Controlling

4.2.1.1 Definition (Abgrenzung zu Qualitätssicherung und EDV-Revision)

4.2.1.2 Standardisierung

4.2.1.3 Normen und Messgrößen

4.2.1.4 Planungs-, Steuerungs- und Kontrollinstrumente

4.2.1.5 Eingliederung in Informatik-Strategie und Leitbild

4.2.1.6 Methoden des Informatik-Controlling

4.2.1.7 (Alternativensuche, Bewertung, Projektplanung, Kontrolle, Accounting, Monitoring, Benchmarking)

4.2.2 Qualitätssicherung 1

4.2.2.1 Metriken

4.2.2.2 Software-Ergonomie

4.2.2.3 Effektivität

4.2.2.4 Qualitätsmerkmale von Daten und Dokumentationen (Änderbarkeit, Aktualität, Vollständigkeit...)

Anforderung an Dokumentationen:

änderbar / aktuell / eindeutig / indentifizierbar / normenkonform / verständlich / vollständig / widerspruchsfrei / etc

4.2.2.5 Projektplan

Inhalte :

- Projektorganisation (Aufstellung eines Organigramms, Festlegen der Verantwortlichkeiten)
- Projektbeschreibung (Arbeitsumfang, Annahmen, Lieferumfang, Abnahmeverfahren)
- Entwicklungsprozess (Phasen der Entwicklung, Arbeitsergebnisse, Integration der Prüfungen, Qualitätssicherung, Konfigurationsmanagement)
- Entwicklungsplan (Arbeitspakete, Terminpläne, Kostenplan, kritische Punkte, Risiken)
- Anforderungen an die Umgebung (Rechnersystem, Software, Leistung anderer)

4.2.2.6 Freigabewesen

4.2.2.7 Gestaltung von Support und Schulung

4.2.2.8 Qualitätprüfung, (Reviews, Tests, Konfigurationsmanagement, Verifikation, Validierung)

Konfigurationsmanagement = Verfahren zur SW-Kennzeichnung und systematischen Überwachung.

Ein **Review** ist ein geregeltes, dokumentiertes Verfahren einer Prüfung von Resultaten, d.h. der Projektergebnisse. Ein Review überprüft und kommentiert die in schriftlicher Form vorliegenden Projektergebnisse. (KG 22).

Prüfobjekte von Reviews:

- Anforderungsspezifikation
- Entwurf
- Code
- Testpläne
- Testfälle
- Testergebnisse
- Benutzerhandbuch

Validierung (Validation):

- Unter Validation versteht man die Prüfung, ob das Produkt den Erwartungen/Bedürfnissen des Benutzers gerecht wird (mache ich das richtige?). In der Regel handelt es sich um die Prüfung der Anforderungsspezifikationen durch den Auftraggeber und die Abnahme.

Mit der Validation wird versucht:

- die richtige Basis für die Entwicklung zu legen
- die Gebrauchstauglichkeit des Produktes zu beurteilen

4.2.2.9 Versionen-Verwaltung

4.2.3 EDV-Revision 1

4.2.3.1 Methoden und Massnahmen (Prüfpfadkonzept, Hilfsmittel, Tools)

4.3 Standardisierung 3

4.3.1 Ziele der Standardisierung

4.3.1.1 Verinfachung

4.3.1.2 Kontinuität

4.3.1.3 Einheitlichkeit

4.3.1.4 Austauschbarkeit (Bausteine, Module, Komponenten)

4.3.1.5 Investitionsschutz

4.3.2 Standardisierungsgebiete 3

4.3.2.1 Qualitätssicherung / Controlling

4.3.2.2 Telekommunikation

4.3.2.3 Programmierung

4.3.2.4 Testorganisation

4.3.2.5 Methoden/Verfahren

4.3.2.6 EDV-Produktion

4.3.2.7 Dokumentation

4.3.3 Standards für die Informatik 3

4.3.3.1 Vorgehensmodell

4.3.3.2 Projektbearbeitung

4.3.3.3 Entwicklungsumgebung

4.3.3.4 Hardware-Standard

4.3.3.5 Software-Standard

4.3.3.6 Applikationsübergreifend (z.B. EDIFACT)

4.4 System-Design

(für Informations- und Funktionsstruktur)

4.4.1 Design-Methoden, Design-Techniken 1

4.4.1.1 CASE

4.4.1.2 Workflow (Geschäftsprozess, funktions- und objektorientiert)

4.4.1.3 Daten-Design und Informationsfluss

4.4.1.4 Normalisierung, Kommunikations-Analyse

4.4.2 Lösungsformen 1

4.4.2.1 Batch

4.4.2.2 Dialog

4.4.2.3 Zentral

4.4.2.4 Dezentral

4.4.2.5 DDP (Distributed Data Processing)

4.4.2.6 CSA (Client/Server-Architektur)

4.4.2.7 IC (Information Center-Konzept, Datenextrakt)

4.4.2.8 Enduser-Computing

4.4.3 Bearbeitungsgrundsätze 1

4.4.3.1 Hardest First

4.4.3.2 Top Down

4.4.3.3 Bottom Up

4.4.3.4 Middle Out

4.5 Daten-Speicherung und -Verwaltung

4.5.1 Daten-Management 1

4.5.1.1 Datenspeicherung (zentral, verteilte)

4.5.1.2 CSA (Client-Server-Architektur)

4.5.1.3 Datenbank-Modelle

4.5.1.4 Datenbanktheorie

4.5.2 Daten-Administration 1

4.5.2.1 Data-Dictionary, Repository

4.5.2.2 Datennutzung

4.5.2.3 Speicherungsformen und Zugriffsmethoden

4.5.2.4 Auswertungen

4.5.2.5 Reorganisation

4.5.2.6 Abfragesprachen

4.6 Applikationsentwicklung

4.6.1 Programmorganisation 1

4.6.1.1 Denken in Prozessen

4.6.1.2 Ablaufstrukturen (Dialog- und Batchdesign)

Prinzipien der Dialogabläufe:

- Einzelaufgabenorientiert
- Geschäftsfallorientiert (case processing)

4.6.1.3 Datenstrukturen

4.6.1.4 Normierte Programmierung

4.6.1.5 Programm- und Modulbildung

4.6.1.6 Interprogramm-Kommunikation, Schnittstellen

4.6.2 Programmiersprachen, Programmierhilfen 1

4.6.2.1 Arten und Aufbau

4.6.2.2 Einsatzgebiete, Vorteile und Nachteile

4.6.2.3 Compiler

4.6.2.4 Interpreter

4.6.2.5 Generatoren

4.6.2.6 Macroprozessoren

4.6.2.7 Viert-/Fünfte-Generationssprachen

4. Generation : Abfagesprachen und Anwendungsgeneratoren

Mit der Verbreitung von Datenbanken und dem Zunehmen der Datenverarbeitung entstand das Bedürfnis, Datenbanken auf einfache Weise abfragen, manipulieren und auswerten zu können. Diesem Bedürfnis entsprechend wurde eine Reihe von Softwareprodukten auf den Markt gebracht, die die verlangten Fähigkeiten mehr oder weniger zur Verfügung stellen. Kein einheitliches Beurteilungsverfahren für Programmiersprachen dieser Generation. Viele sogenannte 4GL Systeme erlauben lediglich Abfragen von Datenbeständen, was in vielen Fällen nicht genügt. Erweiterte Fähigkeiten ermöglichen zusätzlich das Verändern der Datenbestände und - wenn sie eine Entscheidungshilfe sein sollen - auch eine Auswertung, zum Beispiel in der Form von Tabellenkalkulations-Systemen. Derart umfassende Systeme bezeichnen Hersteller oft als "Anwendungsgeneratoren".

Beispiele : ORACEL, AS, MAPPER, FOCUS, IFPS, ICU/GDDM

5. Generation : Sprachen zur Wissensverarbeitung

Die Sprachen der 1. bis 2. Generation verarbeiten als Information Zahlen und Texte. Für die Anwendungen im Gebiet der Künstlichen Intelligenz bei Expertensystemen geht es um etwas anderes: Dort ist Wissen herzuleiten und zu verarbeiten. Für diese Aufgabe werden neue Programmiersprachen entwickelt, die Algorithmen der Wissensbildung (etwa nach dem Gesetzen der Logik) zu implementieren gestatten. Vertreter dieser Generation, die erst am Anfang der Entwicklung stehen sind:

PROLOG (Programmin in Logic), eine prädikative Programmiersprache. Das eigene Wissen über ein Problem kann formuliert werden, und der Computer versucht damit, Antworten auf neue Fragestellungen zu finden.

LISP (List Processing Language), eine funktionale Programmiersprache, die auf der Theorie linearer Listen aufbaut und unter anderem im Umgang mit Formeln zum automatischen Beweisen eingesetzt werden kann.

4.6.2.8 Objektorientierte Programmierung

4.6.3 Programmbeschreibung 1

4.6.3.1 Kurzbeschreibung

4.6.3.2 Programmablaufplan

4.6.3.3 Strukturierte/formalisierte Beschreibung

4.6.4 Standardisierte Applikationssoftware

4.6.4.1 Einsatzgebiete/Beurteilungskriterien 1

4.6.4.2 Auswahl- und Bewertungsverfahren 2

4.6.5 Applikationsschnittstellen 1

4.6.6 Applikationsdokumentationen 1

4.7 Test

4.7.1 Ziel und Zweck 2

Als Test bezeichnet man die systematische Prüfung vom Programmabläufe und Programmen insbesondere auf korrekte Funktionserfüllung, Ressourcenbeanspruchung, Zeitverhalten etc. mit dem Ziel, noch vor der Übergabe der Lösung Fehler und Schwachstellen zu finden und zu beheben. Es geht beim Testen ausdrücklich nicht darum, zu zeigen, daß das Programm fehlerfrei läuft. Denn dann könnte der Test ja nach dem ersten fehlerfrei Testfall abgebrochen werden, was natürlich sehr oberflächlich wäre.

Das Programm wird definitiv auf logische Fehler untersucht

4.7.1.1 Für Analyse, Programmierung und Betriebsorganisation

4.7.1.2 Für das Rechenzentrum

4.7.1.3 Für den Benutzer

4.7.2 Testarten 2

4.7.2.1 Review/Walk Through

- Review: ist ein geregeltes, dokumentiertes Verfahren einer Prüfung von Resultaten, d.h. der Projektergebnisse. Ein Review überprüft und kommentiert die in schriftlicher Form vorliegenden Projektergebnisse. (KG 22).

Prüfobjekte von Reviews:

- Anforderungsspezifikation
- Entwurf
- Code
- Testpläne
- Testfälle
- Testergebnisse
- Benutzerhandbuch

Referenzunterlagen eines Reviews:

- Checklisten (meist Standard-Checklisten, die für einen bestimmten Review mit speziellen Fragen ergänzt werden).
- Richtlinien und Weisungen (z.B. betriebliche Standards für die Gestaltung von Systemen)
- Vorgaben des Auftraggebers (z.B. Projektauftrag, Pflichtenheft, etc.)

4.7.2.2 Modultest/Programmeinzeltest

4.7.2.3 Code-Inspection (z.B. Performance-Aspekte)

4.7.2.4 Schreibtischtest (Trockentest)

Der Programmablaufplan wird "trocken" getestet, d.h. am Schreibtisch überprüft. Dabei werden Testdaten angenommen, und mit dem Bleistift in der Hand wird im Ablaufplan den entsprechenden Wegen nachgefahren. Ausserdem wird geprüft, ob alle Varianten und Möglichkeit bieten noch entsprechende Programmzweige behandelt **werden**.

4.7.2.5 Kettentest (Funktionstest)

Kettentest :

Man konzentriert sich auf das Testen der Modulschnittstellen und das Zusammenwirken der Module (Modulkommunikation (Ketten/Integrationstest))

4.7.2.6 Integrationstest (technischer und betrieblicher)

Vorgehen der Methode "nichtinkrementellen" Testens (auch big bang Test genannt):

- Zuerst wird jedes der einzelnen Module dem Modultest unterzogen, indem jedes Modul als eigenständige Einheit behandelt wird.
- Die Module können gleichzeitig oder nacheinander getestet werden.
- Anschliessend werden die Module zu Programmen zusammengesetzt und integriert.

Inkrementelltest:

Die Module werden nicht separat getestet, sondern man kombiniert das nächste, noch nicht getestete Modul mit den anderen, bereits getesteten Modulen

4.7.2.7 Systemtest (Verbundtest, Regressionstest)

Man testet die gesamte Software, d.h. alle unteren Hierarchiestufen sind vorhanden und getestet. Die funktionale (und Grenze der) Leistungsfähigkeit wird getestet.

4.7.2.8 Katastrophentest

4.7.2.9 Abnahmetest

Parallellauf und Pilotinstallation sind genau genommen keine Testmethoden weil:

- Weil das Ziel dieser beiden "Testmethoden" nicht das Fehlerfinden, sondern die Verminderung des Einführungs-Risikos ist.

Testen = Ziel ist, Fehler zu finden!

- Einführen = Ziel ist, die Umstellung (Einführung) abzusichern!

4.7.2.10 Parallellauf

4.7.2.11 Pilotinstallation

4.7.3 Testdaten, Testfälle 2

4.7.3.1 Auswahl und Beschaffung (wer, was)

4.7.3.2 Testfallbestimmung wie z.B. Reihenfolge

4.7.3.3 Normalfälle, Spezialfälle, Grenzwerte (inhaltlich und volumenbezogen)

4.7.4 Testverfahren, Testvorgehen 2

4.7.4.1 Prinzipien der Teilung, der Wiederholung, der Messbarkeit

4.7.4.2 Teststrategien

4.7.4.3 Testplanung, -verarbeitung, -durchführung, -ausführung, -abnahme, -verwaltung

4.7.5 Testdokumentation 2

Die 4 Teile des Testberichts sind:

- Testzusammenfassung
- Testprotokoll
- Liste der Probelemmeldungen
- Liste der SW-Einheiten

4.8 Einführung/Schulung

4.8.1 Einführungsvorgehen 3

4.8.1.1 Formen (schlagartig, stufenweise, parallel)

4.8.1.2 Vorbereitung

4.8.1.3 Betreuung der Benutzer

4.8.2 Schulungsplanung und -organisation 3

4.8.2.1 Zielpublikum

4.8.2.2 Methode (Unterricht, schriftlich, on the job)

4.8.2.3 Vorbereitung (8Aufbau, Administration, Hilfsmittel usw.)

4.8.3 Benutzerhandbuch 3

4.8.3.1 Inhalt

4.8.3.2 Art der Speicherung, Medien

4.8.3.3 Zugriff, Verfügbarkeit

4.8.3.4 Verwaltung und Archivierung

4.9 Betrieb/EDV-Produktion

4.9.1 Rechenzentrum 1

4.9.1.1 Organisationsformen

4.9.1.2 Aufbauorganisation

4.9.1.3 Ablauforganisation

4.9.1.4 Hardwarebedienung und -betreuung (Steuerung und Überwachung)

4.9.1.5 Betriebsplanung und -abwicklung

4.9.1.6 Systemverwaltung und -betreuung

4.9.1.7 Schnittstelle zu Fachbereichen und Applikationsentwicklung

4.9.1.8 Infrastruktur

4.9.1.9 Sicherheit

4.9.1.10 RZ-Dokumentation

4.9.2 Planung, Accounting und Berichtswesen 1

4.9.2.1 Volumenentwicklung, Leistungsbedarf, Nutzkapazität

4.9.2.2 Leistungserfassung, Preisfindung, Verrechnung

4.10 Unterhalt der Applikation

4.10.1 Applikationserhaltung 2

4.10.1.1 Applikations-Controlling

4.10.1.2 Organisationsregelungen

4.10.1.3 Dokumentation

Anforderung an Dokumentationen:

- änderbar / aktuell / eindeutig / identifizierbar / normenkonform / verständlich / vollständig / widerspruchsfrei / etc

4.10.1.4 Benutzerergonomie, Anwendbarkeit

4.10.1.5 Betriebliche Gegebenheiten

4.10.1.6 Release Management

4.10.1.7 Migration, Konversion (Aufwärtskompatibilität, Rollback)

4.10.2 Nachkalkulation und Ausserkraftsetzung 2

4.10.2.1 Projektabrechnung

4.10.2.2 Leistungsnachweis und -verrechnung

4.10.2.3 Ausserkraftsetzungsplanung

4.10.2.4 Benutzerbetreuung und -berechtigung

4.10.2.5 Gesetzeskonformität

4.10.2.6 Revisionsanforderungen

4.10.2.7 Aufbewahrung

4.10.2.8 Physikalische Lösungen

4.11 Datensicherheit und Datenschutz

4.11.1 Risikobereiche 2

4.11.1.1 Benutzer

4.11.1.2 EDV (Entwicklung, Produktion und Wartung)

4.11.1.3 Kommunikationsnetz

4.11.1.4 Kontrollmechanismen (Authentifikation, Ownership)

4.11.2 Sicherheitsmassnahmen, Kontrollmechanismen 2

4.11.2.1 Sichernsstellungs-Konzept

**4.11.2.2 Technische Massnahmen (Logging, Restart, Recovery, Check Points
etc.)**