

Checklisten

Software Engineering

für Teamprojekte im kommerziellen Bereich

Copyright © 1998, 2001
Stefan Biffel, Wolfgang Zuser
Institut für Softwaretechnik

Inhaltsverzeichnis	
EINLEITUNG	1
CHECKLISTE <PHASE> – <DOKUMENT>	1
GENERELLE CHECKLISTE FÜR DOKUMENTE	1
CHECKLISTEN ANALYSE	2
DATEN	2
ANFORDERUNGEN	2
ANALYSEMODELL	4
REVIEW DER ANALYSE	4
VORBEREITUNGEN ZUM UMSETZEN DER ANALYSE	5
CHECKLISTEN ENTWURF	7
INHALT SYSTEMENTWURF	7
REVIEW DES ENTWURFS	8
VORBEREITUNGEN FÜR DIE IMPLEMENTIERUNG	8
ENTWERFEN VON KOMPONENTEN	9
ENTWERFEN EINER OBJEKT-METHODE	9
CHECKLISTEN TESTEN	10
VORBEREITUNGEN	10
TESTPLAN	10
INTEGRATIONSBERICHT	10
TESTBERICHT ZU EINEM (TEIL)SYSTEM	11
FEHLERAUFZEICHNUNGEN	11
CHECKLISTEN ANWENDERDOKUMENTATION	12
ANWENDERHANDBUCH UND ONLINE-HILFE	12
CHECKLISTEN PROJEKTMANAGEMENT	13
PROJEKTSTART	13
PROJEKTPLAN	13
PROJEKTMAPPE	13
PROJEKTENDE	15

Einleitung

Eine der größten Herausforderungen im Rahmen eines Software-Entwicklungsprojektes ist die Übersetzung der Probleme und Lösungen in die jeweilige Fachsprache der beteiligten Personengruppe (Anwender, Entwickler, Management). Dokumente dienen im Entwicklungsprozeß als Schnittstelle zwischen Phasen und Projektmitarbeitern, über die die notwendige Information möglichst vollständig transportiert werden soll.

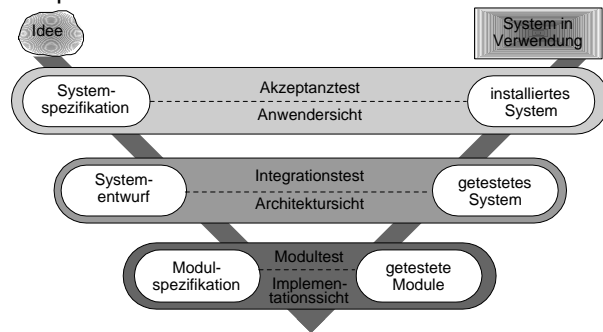


Abb. 1: Schichten des V-Modells

Die Checklisten enthalten folgende gestaltende Elemente.

Checkliste <Phase> – <Dokument>

Nach der Überschrift folgt oft ein erklärender Einleitungstext.

<Dokumentabschnitt>

<Überschrift>. Ein Text in diesem Format symbolisiert eine Inhaltsangabe. Im Dokument soll unter dieser <Überschrift> der entsprechende Inhalt folgen.

- In diesen Punkten stehen Tips und Fragen, die der Verfasser des Dokuments als Denkanstoß

verwenden kann. Ob der Tip oder die Frage für das Dokument relevant ist, hängt vom konkreten Projekt ab.

Generelle Checkliste für Dokumente

Alle aktuellen Dokumente werden in einer geeigneten Projektmappe (am besten einem Ringordner) gesammelt und durch Zwischenblätter mit Indizes zum schnelleren Auffinden getrennt. Die vollständige Projektmappe ist bei jeder Review mitzubringen.

Grundlegende Überlegungen

- Beginnen Sie mit einfachen, auch für Uneingeweihte gut verständlichen Inhalten.
- Legen Sie besonderen Wert auf eine einfache Sprache.
- Verwenden Sie zur Übersicht Diagramme.

Planung

- Je mehr Personen am Dokument arbeiten, desto genauer sollten Zusammenbau und Überprüfung einer Version geplant werden.
- Wer wird welches Kapitel verfassen?
- Welche Werkzeuge stehen den einzelnen Personen zur Verfügung?

Format und Layout

- Am Deckblatt sind Teambezeichnung, Dokumentbezeichnung, Projektbezeichnung (ev. Logo), Dokumentversion, Datum und Autor(en) (der Dokumentverantwortliche zuerst) beschrieben.
- Ab zehn Seiten Umfang ist ein Inhaltsverzeichnis mit Seitennummern zu erstellen.
- Überlegen Sie sich, welche Schriftsätze und grundlegende Layoutvorschriften für alle Dokumente gelten sollen. Fassen Sie diese Vorgaben in einem allgemeinen Teil am Anfang der Dokumentationsrichtlinien zusammen.

- Überlegen Sie sich Richtlinien zur Darstellung von Tabellen und Grafiken. Vorgaben für Grafiken sollten Sie mit einer Beispielgrafik versehen.
- Stellen Sie sicher, daß ihr Layout zum Drucker paßt, auf dem Sie drucken werden.

Inhalt

- Stellen Sie wichtige, allgemeine Inhalte an den Anfang, etwa "wichtige Begriffe", "Wer sollte dieses Dokument lesen?" oder "Wie ist dieses Dokument aufgebaut?".
- Begriffe, die für das Verständnis eines Dokumentes wichtig sind, sollen einheitlich verwendet werden.
- Vorgaben für ein Dokument sollten auf eine A4-Seite passen, damit sie auch gelesen werden.

Hinweise für Entwurfsversionen

- Die Version bzw. Aktualität soll klar sein.
- Papier nur einseitig beschreiben, das Dokument bleibt so leichter teilbar und kopierbar.
- Seitennummer und Dokumentname sollten auf jeder Seite zu finden sein, für den Fall, daß einmal ein Stoß Papier vom Tisch fällt oder die Blätter beim Lesen durcheinandergeraten.
- Genügend Platz für Korrekturen lassen.
- Die Einarbeitung von Änderungen gegenüber der letzten Version genau überprüfen bzw. abhaken, bevor die alte bemalte Version entsorgt wird. Oft werden einige Korrekturen übersehen, was leicht zu einem Versionenchaos und zu Frust bei den Korrekturlesern führt ('Das habe ich doch schon in der vorletzten Version ausgebessert...').
- Eine Seitennumerierung in der Mitte der Seite erspart Überraschungen bei doppelseitigem Kopieren.

Checklisten Analyse

Die Analyse beschreibt den Softwareanteil einer Lösung zu einem übergeordneten Problem, dessen Einbettung in die Gesamtlösung (technisch und wirtschaftlich bzw. organisatorisch). Es kann im Zuge einer allgemeineren Lösung zum Einbeziehen von generellem Domänenwissen kommen. Im Hinblick auf eine leichtere Änderbarkeit oder Wiederverwendbarkeit von Teilen des Systems ist es sinnvoll, in der Analyse eine Unterteilung des gesammelten und modellierten Wissens nach dem Grad der Allgemeinheit (Anwendungsbereich bzw. Branche oder Domäne – System – Softwarelösung – spezifische Technologie) einzuhalten.

Eine Checkliste für Reviews der Analyse bildet einerseits die Struktur des zu prüfenden Dokuments ab und zeigt andererseits Kategorien von typischen Anforderungsfehlern auf.

Voraussetzung für die Durchführung einer Analyse: Es gibt einen geeigneten Kontakt zu Personen, die zum Anwendungsbereich Auskunft geben können und wollen.

Diese Checkliste beinhaltet jene Aspekte, die den Teil des Entwicklerteams betreffen, der für die Übersetzung der Kundenwünsche in eine technisch machbare und validierbare Spezifikation zuständig ist.

Daten

Domänenmodelldiagramm

Domänenmodelldiagramm. In einem Überblicks-Klassendiagramm sind alle Daten der Anwendungsdomäne und deren Beziehungen darzustellen, die zur Beschreibung der fachlichen Lösung notwendig sind. In den einzelnen Klassen werden alle wesentlichen Attribute dargestellt.

Datenübernahme

Datenbestände. Anhand von Dokumenten wie etwa Akten, Formularen, Karteien, Listen, Dateien oder Arbeitsanweisungen ist ein Überblick zu den verwendeten Datenbeständen zusammenzufassen.

- Wie hoch ist die Integrität der Daten in existierenden Dokumenten und Datenbanken? Gibt es Datenverluste bzw. Fehler, die auf inkonsistente Daten hinweisen?
- Wieviel der derzeitigen Datenbank wird aktiv genutzt? Von wem?

Übernahme existierender Daten. Beschreibung welche und wie viele Daten zu übernehmen sind.

- Sollen umfassende Datenarchive von der existierenden Datenbasis konvertiert werden?
- Wie schwierig wird es sein die derzeitige Datenbank in eine neue umzuwandeln? Wieviel Testaufwand wird notwendig sein, um eine saubere Umwandlung zu erreichen?

Anforderungen

Systembeschreibung

Ausgangssituation. Kurzer Überblick über den Kunden und die aktuelle Situation im Unternehmen.

Zielsetzung. Was soll mit dem Projekt in bezug auf die Ausgangssituation erreicht werden?

Wissensquellen. Zu Beginn der Analyse macht es Sinn, mehr oder weniger naheliegende Informationsquellen zu sammeln, die im Zuge der Analyse existierendes Wissen zugänglich machen.

- Ist für diesen Anwendungsbereich einschlägige, aktuelle Fachliteratur verfügbar?
- Wurden bereits Analysearbeiten in diesem Anwendungsbereich durchgeführt?
- Gibt es Handels- oder Industrievereinigungen, die Informationen im Bereich sammeln?

Organisation. Als Grundlage für die Erhebung ist ab einer nicht-trivialen Firmengröße die organisatorische Aufteilung in Geschäftsbereiche und die räumliche Verteilung geeignet zu beschreiben.

Existierende Systeme in der Branche. Eine kurze Zusammenfassung über den Stand der Technik bei Vorgehensweisen und kommerziellen Produkten, die als Vorbild dienen können.

- Gibt es Standardsoftwarepakete, die dieses Anwendungsgebiet abdecken?
- Können andere Unternehmen bezüglich ihrer Vorgehensweisen kontaktiert werden?

Existierende Systeme im Unternehmen. Falls im Unternehmen bereits Software-Systeme oder Prototypen im Einsatz sind, die auf das zukünftige System Einfluß haben können, so sind diese zu beschreiben.

Gesetzlicher Rahmen. Zusammenfassung von im Rahmen der Spezifikation anzuwendenden gesetzlichen Regelungen oder behördlichen Auflagen.

Schwachstellen existierender Systeme. Eine Beschreibung der wesentlichen Fehler, Kosten und Begrenzungen in Betrieb stehender Systeme als Basis für Verbesserungs- und Erweiterungsvorschläge.

- Sind berechtigte Beschwerden der Anwender über das derzeitige System dokumentiert? Sind die Auswirkungen der Schwachstellen dokumentiert?
- Welche Entwicklungs-, Wartungs- und Einsatzkosten sind mit dem derzeitigen System verbunden, inklusive der Zeitaufwände der Anwender?
- Erwarten die Anwender einige bestimmte Vorteile von dem neuen System? Sind diese realistisch erfüllbar?

Abgrenzung des Systems. Überblick, welche Teile des Unternehmens vom neuen System betroffen sein werden.

Begriffsverzeichnis

Verzeichnis der Fachausdrücke aus dem Anwendungsbereich. Alle möglicherweise mißverständlichen oder mehrdeutigen Begriffe müssen präzise definiert werden.

Aktorenliste

Rollen. Die unterschiedlichen Funktionen der Anwender(gruppen) im Unternehmen, das sind Rollen, sind mit ihren Verantwortlichkeiten und Kompetenzen bzw. Berechtigungen geeignet zu beschreiben.

Beschreibung der zu berücksichtigenden Abläufe im Unternehmen - Anwendungsfälle

Anwendungsfälle. Alle Anwendungsfälle, die im zukünftigen System abgebildet werden sollen, sind gemeinsam in einem *Anwendungfalldiagramm* und einzeln im Detail zu beschreiben. Die *Anwendungsfallbeschreibung* soll Auskunft geben, welche Rollen mit dem Ablauf zu tun haben, welche Ein- und Ausgabedaten wie bearbeitet werden und unter welchen Umständen andere Anwendungsfälle angestoßen werden. Hier sind alle normalen Abläufe und auch alternative Abläufe sowie Fehlerfälle zu berücksichtigen.

- Sind alle Anforderungen bekannt, die die Anwender durchführen möchten?
- Ist das Anwendungsfalldiagramm übersichtlich und aussagekräftig? Ist die Benennung der Anwendungsfälle intuitiv verständlich?
- Kann jeder Anforderung einer Gruppe von Anwendern zugeordnet werden (Aktoren)?
- Ist für jede Anforderung klar, welche Daten dafür notwendig sind, und welche Daten dabei entstehen? Sind diese Daten im Domänenmodell enthalten?

- Ist die Beschreibung der Anwendungsfälle zur Erstellung eines ersten Prototypen und des Analysemodells ausreichend?
- Sind alle nichtfunktionalen Anforderungen bekannt (Qualitätsanforderungen, Leistungsanforderungen, Fehlerverhalten)?

Nichtfunktionale Anforderungen an das zukünftige System

Die nichtfunktionalen Anforderungen können als Teil der Anwendungsfälle dokumentiert werden, oder als eigenes Dokument ausgeführt werden. Jedenfalls sollten folgende Punkte eindeutig beschrieben sein.

Qualitätsanforderungen. Die drei bis fünf wesentlichsten Qualitäten, die das zukünftige System haben soll, sind in absteigender Rangfolge zu nennen und anhand von Beispielen zu erläutern.

- Üblicherweise erwünschte Qualitäten sind Korrektheit, Verlässlichkeit, Anwenderfreundlichkeit, Genauigkeit, Sicherheit, Robustheit.

Leistungsanforderungen. Für das System bzw. einzelne Betriebsabläufe sind jene Leistungsanforderungen überprüfbar zu beschreiben, die für den Systemerfolg wesentlich sind. Hier werden etwa Antwortzeiten, Durchsatz für Daten und typische Aufgaben oder Genauigkeiten erläutert.

- Ist die vom Anwender erwartete Antwortzeit für die wesentlichen Aufgaben bekannt?
- Ist die geforderte Leistung unter Berücksichtigung der anderen Systeme, die mit dem Software-System zusammenarbeiten, realistisch erreichbar?
- Wurden Leistungskriterien für das System als Ganzes und für die Teile im einzelnen definiert? Sind diese Leistungskriterien mit der gegenwärtigen Technologie machbar? Sind sie sinnvoll? Gibt es Prioritäten, etwa Kosten/Nutzen-Überlegungen?

Fehlerverhalten. Eine Zusammenfassung des akzeptablen Systemverhaltens in Ausnahmeständen. Für das gesamte System soll einheitlich beschrieben werden, welche Reaktionen das System in Krisensituationen zeigen darf; etwa Dialoge mit dem Anwender, Schreiben auf eine Logdatei, Verstärken des Systemoperators, Umschalten auf Backup-Systeme oder Ähnliches.

- Gibt es Überlegungen, was im Fall eines Softwareversagens passieren kann, welche Information davor jedenfalls geschützt werden muß, und welche Strategie zur Fehlererkennung und -behebung verwendet werden soll?

- Ist der Initialzustand des Systems definiert?

Dokumentation und Änderungsmanagement. Im Rahmen des Projekts soll aus der Spezifikation eine Implementierung entstehen; dabei ist es besonders wichtig, daß die einzelnen Anforderungen der Spezifikation *verfolgbar* sind, d.h. nachvollziehbar in den Entwurf und einzelne Komponenten einfließen, die diese Anforderung erfüllen. Da ein erfolgreiches System im Lauf seines Lebenszyklus an Änderungswünsche anzupassen ist, ist hier zu beschreiben, welche Dokumente für *Anwender, Betrieb und Wartung* herzustellen sind und wie über die Durchführung und Reihung von Änderungswünschen entschieden werden soll.

Ausblick auf den Lebenszyklus des Systems. Eine kurze Zusammenfassung über die geplante Lebensdauer des Systems sowie absehbare Erweiterungen bzw. Leistungsverbesserungen.

- Gibt es Überlegungen in bezug auf die Wartbarkeit des Systems, wahrscheinliche Änderungen der Funktionalität, der Schnittstellen oder der Betriebsumgebung?

Analysemodell

Analysemodellidiagramm

Analysmodellidiagramm. Als Übersicht ist die Aufteilung des Systems als Klassendiagramm mit den Stereotypen Schnittstelle, Kontroller und Entität zu erstellen.

Schnittstellen des zukünftigen Systems

Datenfluß im Überblick. Alle Informationen, die in das System hinein bzw. aus dem System heraus fließen sollen, sind geeignet anhand der beteiligten Fremdsysteme darzustellen.

Datenaustausch mit anderen organisatorischen Systemen und Anwendungen: Existierende Formulare und Berichte sind eine gute Basis für Ein- und Ausgaben, mit denen das System arbeiten wird.

- Sind für alle *Eingaben* in das System ihre Herkunft und Typ (Format, Wertebereich, ev. Genauigkeit) bekannt?
- Sind für alle *Ausgaben* des Systems ihr Ziel, Typ (Format, Wertebereich, ev. Genauigkeit) bekannt? Sind die Berichtsformate spezifiziert?

Anwenderschnittstellen. Als wesentliche Schnittstellen zu den Benutzern des Systems werden deren Aussehen und die wesentliche Funktionalität der Anwenderschnittstellen beschrieben.

Anforderungen an die Anwenderschnittstelle. Diese sind nach den Bedürfnissen des Publikums der jeweiligen Schnittstelle zu beschreiben, ohne den Entwurf unnötig einzuschränken. Hier haben sich Skizzen, verbale Beschreibungen oder Verweise auf (Papier-)Prototypen bewährt.

Umsetzung der Anforderungen an das zukünftige System - Kontroller

Beschreibung der Kontroller. Für jeden Kontroller gibt es eine eindeutige Bezeichnung, eine Liste

seiner Attribute samt Typenbeschreibung und eine Liste seiner Leistungen bzw. Services bzw. Methoden samt Parametern und Beschreibung.

- Werden alle Anwendungsfälle durch die Leistungen eines Kontrollers abgedeckt? Gibt es eine einfache Zuordnung von

Persistentes Datenmodell

Datenmodell. Alle permanent zu speichernden Objekte und Daten werden in einem Klassendiagramm (falls vom Kunden gewünscht auch EER-Modell) beschrieben. Das Klassendiagramm ist als Verfeinerung und Erweiterung des Domänenmodells zu verstehen (gemäß den Anforderungen des Analysmodells). Zu jeder Klasse bzw. Entität sind die Attribute und Typen anzugeben.

Beschreibung der Klassen. Für jede Klasse gibt es eine eindeutige Bezeichnung, eine Liste seiner Attribute samt Typenbeschreibung.

- Sind alle permanent zu speichernden Klassenattribute im Datenmodell zu finden?

Integritätsbedingungen. Wo sinnvoll, werden für einzelne Attribute spezielle Wertebereiche bzw. Formate beschrieben, Abhängigkeiten zwischen Attributen von einer oder mehr Entities werden als logische Bedingungen formuliert.

Wachstum der Datenbestände. Für jede Klasse und Beziehung im Datenmodell ist tabellarisch die Größenordnung existierender Daten und eine grob geschätzte jährliche Wachstumsrate anzugeben.

Review der Analyse

Prinzipien

- Die Analyse beschreibt nur das extern wahrnehmbare Systemverhalten (Black-Box).
- Eine Analyse ist in den Details nie 100% fertig, aber gleichzeitig im Grundsatz solide.

Qualitätskriterien

Folgende Qualitätskriterien für die Analyse sind eine allgemeine Checkliste für die System-Definitionsphase. Die Analysedokumente sollten gegen jedes Kriterium geprüft werden:

- *Verständlich.* Die Analyse ist in einer klaren einfachen Sprache abgefaßt.
- *Leicht zu ändern.* Die Analyse ist so verfaßt, daß sie relativ leicht an organisatorische, politische oder technische Änderungen angepaßt werden kann.
- *Vollständig* Alles, was der Kunde verlangt, wird abgedeckt.
- *Relevant.* Nur die Dinge, die der Kunde verlangt, werden behandelt. Eine einfache Kontrolle durch eine unabhängige Person wird unnötige Punkte in der Spezifikation an den Tag fördern. In der Analyse soll keine Hintergrundinformation, keine Geschichte, keine Anekdote enthalten sein. Wenn unbedingt notwendig, kann man diese in den Anhang schreiben.
- *Testbar.* Es ist nicht sinnvoll, eine Anforderung aufzustellen, die nicht getestet werden kann. Worte wie "zuverlässig", "effizient" und "flexibel" sind im allgemeinen ein Kennzeichen von nicht testbaren Anforderungen. Manchmal kann man solche, nicht-quantifizierbare und nicht-testbare Anforderungen umgehen, indem sie durch bestimmte Tests ersetzt werden, die das System überstehen muß, damit es für einen bestimmten Bereich annehmbar ist.
- *Eindeutig.* Es darf nur einen Weg geben, wie die Analyse interpretiert werden kann. Konkret bedeutet das, absolut genaues Definieren und konsistentes Gebrauchen von Ausdrücken.
- *Korrekt.* Es ist offensichtlich, daß die funktionale Analyse korrekt sein soll, aber das bedeutet mehr als die Anforderungen der Anwender korrekt auszudrücken. Eine Gefahrenquelle stellen

einzelne Funktionen dar, die zwar in sich korrekt sind aber zu ungünstigen oder ungeeigneten Kombinationen führen. D.h. die funktionale Analyse ist in all ihren Facetten und Auswirkungen vollständig durchzuarbeiten – etwa mit Hilfe von Prototypen.

- **Konsistenz.** Ausdrücke müssen im ganzen Dokument die gleiche Bedeutung haben. Darüber hinaus sollten sich zwei Anforderungen nicht widersprechen, weder direkt noch indirekt. Das letztere ist natürlich viel schwieriger zu überprüfen und bedeutet, daß über die Auswirkungen jeder Anforderung wirklich nachgedacht werden muß.
- **Verfolgbar** in den Entwurf. Die Analyse sollte Möglichkeiten bieten, um gezielt auf einzelne Anforderungen Bezug nehmen zu können, sodaß in der Entwurfsdokumentation eindeutig auf diese Anforderungen verwiesen werden kann. Das dient als Überprüfung, ob alle Anforderungen implementiert wurden.
- **Realistisch.** Es muß sicher sein, daß alle Anforderungen, mit zu Verfügung stehenden Werkzeugen, Techniken, Personal und Budget implementiert werden können. Deswegen bedeutet Durchführbarkeit sowohl technische als auch logistische Durchführbarkeit.

Wenn die Review abgeschlossen ist, wird die Analyse vom Kunden und vom Entwickler abgesegnet. Die Analyse wird zu einem Vertrag für die Software-Entwicklung.

Vorbereitungen zum Umsetzen der Analyse

Vorbereitungen für den Entwurf

Verstehen und Überprüfen der Analyse

- Hat ein Anwender eine ihm selbst verständliche Version der Analyse überprüft, etwa ein Konzept zum Anwenderhandbuch oder einen Prototypen?
- Sind die Anforderungen in einer klaren, für Anwender und Software-Ingenieure verständlichen, Sprache verfaßt? Sind auch die Anwender dieser Meinung?
- Sind die Anforderungen verständlich für die Entwickler, die den Entwurf durchführen müssen? Sind die Anforderungen so klar, daß sie an eine unabhängige Organisation zur Umsetzung vergeben werden könnten?
- Sind die verwendeten Diagramme verständlich; kann jedes für sich alleine, ohne unterstützenden Text, stehen?
- Sind die Systemziele präzise definiert? Müssen sie interpretiert bzw. später definiert werden?

Verfolgbarkeit

- Sind die Anforderungen vollständig insofern, daß ein Produkt, das alle Anforderungen erfüllt, jedenfalls auch ein erfolgreiches Produkt ist?
- Wurden ev. Entwurfseinschränkungen aus dem Anwendungsbereich bestimmt? Sind die Anforderungen überspezifiziert?
- Gibt es Anforderungen, die wahrscheinlich nicht implementierbar sind und nur aus 'politischen' Überlegungen in die Analyse aufgenommen worden sind?
- Sind alle Anforderungen bis zur Systemebene bzw. zu einzelnen Personen(gruppen) zurück verfolgbar? Ist die Verfolgbarkeit der einzelnen Anforderungen im Entwurf sichergestellt?

Umsetzbarkeit

- Hängen Problembereiche oder Anforderungen an Systemteile stark zusammen? Widersprechen Sie einander? Gibt es einen Trade-off zwischen der Realisierung verschiedener Anforderungen?

- Ist die vorgeschlagene Lösung mit der heutigen Technologie wenig riskant umsetzbar? Sind die Entwurfseinschränkungen realistisch?
- Liegt der Umfang der in der Analyse genannten Ziele im Rahmen des Projektauftrags? Steht ein vorläufiger Plan samt Kostenabschätzung für die Umsetzung des Systems zur Verfügung?
- Sind die Anforderungen an die Ausrüstung für das neue System abgeschätzt? Wie stark kann sich der Technologiefortschritt während des Projekts auswirken?

Auswirkungen

- Wird das neue System die Verwendung von Technologie im Unternehmen stark verändern?
- Wird in die Arbeitssituation von Anwendern substantiell eingegriffen?

Überlegungen bei mehreren Systemvarianten

- Welche Kriterien sind besonders wichtig für die Bewertung der Varianten?
- Wie werden die Varianten bewertet? Ist eine deutlich besser als alle anderen?

Vorbereitungen für den Abnahmetest

- Gibt es zu jeder spezifizierten Funktion praktisch überprüfbare Mindestabnahmekriterien?
- Ist die Erfüllung der Anforderungen durch unabhängige Dritte entscheidbar und überprüfbar? Alle Anforderungen die unklar oder mehrdeutig sind, sollten in einer eigenen Liste zur weiteren Klärung gesammelt werden, da bekanntlich Mißverständnisse während der Analyse oft weitreichende und unter Umständen kostspielige Folgen haben.

Vorbereitungen für Betrieb und Wartung

- Wurden Anforderungen für spätere Erweiterungen spezifiziert und sind diese Anforderungen speziell gekennzeichnet?

- Wurden die notwendigen Fähigkeiten und Ausbildungsbedürfnisse des Bedienungs- und Wartungspersonals spezifiziert?
- Sind die Aufgaben der Anwenderunterstützung erkannt und geplant? Wissen die Anwender davon?
- Sind die Anforderungen mit dem Plan, den Ressourcen und dem Budget konsistent?

Die Analyse bestimmt oft Funktion und Leistung vieler verschiedener Systemelemente. Deshalb müssen in die Review des Systementwurfs mehrere Kundenkreise, von denen sich jeder auf sein Interessensgebiet konzentriert, miteinbezogen werden.

Reviews der Softwareanforderungsanalyse konzentrieren sich auf die Nachvollziehbarkeit der Systemanforderungen und die Konsistenz und Korrektheit der Darstellung.

Checklisten Entwurf

Reviews des Softwareentwurfs konzentrieren sich auf die Datenstrukturen, die Programmstruktur und den Ablauf. Grundsätzlich werden zwei Arten von Entwurfsreviews durchgeführt. Die Review des vorläufigen Entwurfs bezieht sich auf die Übersetzung der Anforderungen in den Entwurf und konzentriert sich dabei auf die Softwarearchitektur. Die zweite Review, meist mit Entwurfsüberprüfung bzw. -durcharbeitung bezeichnet, konzentriert sich auf prozedurale Richtigkeit der Algorithmen, die in den einzelnen Programmkomponenten implementiert werden.

Inhalt Systementwurf

Einleitung mit kurzer Beschreibung der Aufgabe und Hinweis auf wichtige Vorgängerdokumente.

Architektur und Subsysteme

Architektur. Anhand der gewählten Technologievariante wird die Zerlegung des Systems in Subsysteme beschrieben. Den einzelnen Subsystemen werden Gruppen von Anforderungen zugeordnet.

Überblicksdiagramme. Die gewählte Architektur und die Aufteilung des Systems in Teile wird mit geeigneten UML-Diagrammen dargestellt.

- Spiegeln sich die Softwareanforderungen im Architekturentwurf wider? Passen die Teile der Architektur konzeptuell zusammen?
- Gibt es Vorgaben zum minimalen Grad an Robustheit, Verlässlichkeit und Sicherheit? Gibt es Begründungen für wichtige Entscheidungen?
- Wurden vorhandene Softwarekomponenten für die Wiederverwendung identifiziert?
- Beschreibt die Architektur, wie wiederverwendeter Code an die anderen Vorgaben der Architektur angepaßt wird?

- Sind Grundinformationen zum Thema 'Kaufen oder Entwickeln' von Teilen bekannt?
- Wurde die Software so weit unterteilt, daß eine Zuweisung an einzelne Personen zur Implementierung erfolgen kann? Wurden die Verantwortlichkeiten für Konsistenzüberprüfungen festgelegt?

Spezifikation der Komponenten

Subsysteme und Komponenten. Jedes Subsystem kann bei Bedarf weiter in Komponenten zerlegt werden. Jede Komponente ist als compilierbares Modul zu modellieren. Die Komponente enthält die *interne Schnittstellenbeschreibung* mit den Objekten des Entwurfs, die den *Funktionen* und *externen Schnittstellen* der Spezifikation geeignet zugeordnet werden. Für jedes Objekt werden die *Attribute* mit Wertebereich und Integritätsbedingungen angeführt. Für jede *Methode* werden die Parameter und allfällige nicht-triviale Vorbedingungen sowie Normal- und Sonderfälle beschrieben.

- Werden alle Anwendungsfälle aus der Analyse durch eine oder mehrere Komponenten abgedeckt?
- Werden die externen Schnittstellen in Komponenten abgebildet?
- Wurden alle Schnittstellen zwischen den Komponenten festgelegt?
- Sind alle wichtigen Objekte beschrieben und überprüft?

Anwenderschnittstelle

Anwenderschnittstelle. Wird in Form eines Prototypen hergestellt, sodaß alle Dialoge anzeigbar sind. Fehlermeldungen bestehen zumindest aus einer Fehlerbezeichnung, einer Kurzbeschreibung des Fehlers und Möglichkeiten zur Fortsetzung der Arbeit in der Sprache des Anwenders.

- Gibt es eine *Strategie* zur Behandlung der Anwendereingaben? Sind die zentralen Aspekte der Anwenderschnittstelle definiert? Sind die Anwendereingaben auf ein Minimum beschränkt?
- Werden *Fehlermeldungen* einheitlich beschrieben, sodaß Fehler dem Anwender über eine saubere Schnittstelle präsentiert werden?
- Sind die *Layouts* der Anwenderschnittstelle einheitlich?
- Sind Bildschirmmasken nicht mit Informationen überladen? Sind die Bildschirmausgaben übersichtlich? Ist die Anwenderführung ausreichend?

Klassendiagramm

Klassendiagramm. Es gibt ein Diagramm, in dem neben den Klassen aus der Analyse alle weiteren benötigten Klassen mit ihren Beziehungen zueinander aufscheinen. Die Klassen können üblicherweise in *fünf Schichten* geordnet werden; zuoberst *Schnittstellen, Controller*, dann die Objekte des *Datenmodells*, zum Teil wiederverwendbare *Werkzeugobjekte* und zuunterst *Infrastruktur und Betriebssystemobjekte* aus einer Klassenbibliothek.

- Gibt es eine einheitliche Strategie zur Fehlerbehandlung?

Interne Datenstrukturen und Algorithmen

Datenstrukturen und Algorithmen. Die Klassen werden um die Definition der internen Datenstrukturen und Algorithmen (als Pseudocode in Kommentar) ergänzt.

- Sind alle wesentlichen Datenstrukturen beschrieben, konsistent mit dem Anwendungsbereich und den Softwareanforderungen?
- Werden die wichtigen Datenstrukturen durch Zugriffsroutinen abgeschirmt?
- Sind alle nicht-trivialen Algorithmen beschrieben und überprüft? Erfüllt jeder Algorithmus die geforderte Funktion?

- Ist der Algorithmus logisch korrekt? Ist die logische Komplexität angemessen?

Datenbank

Datenbank. Falls im Entwurf Ergänzungen zum persistenten Datenmodell der Analyse notwendig werden, sind sie, wie für die Analyse verlangt, zu beschreiben.

- Wurde das Datenmodell geprüft? Ist es in sich und mit dem Klassendiagramm konsistent?
- Sind die wesentlichen Integritätsbedingungen explizit angegeben?

Infrastruktur

Im Rahmen der *Infrastruktur* werden grundlegende Themen behandelt, wie die systemweite Behandlung von Fehlern, die Vergabe von Berechtigungen auf Daten und Ressourcen und die Initialisierung des Systems. Es gibt eine Liste der Fehlertypen, auf die das System planmäßig reagiert.

- Gibt es eine Strategie zur Vergabe der Systemressourcen?
- Welche Anforderungen werden an das Betriebssystem gestellt?
- Gibt es definierte Systeminitialisierungs-, -start- und -wiederanlaufprozeduren?
- Wurde ein Prinzip festgelegt, nach dem die interne Überprüfung stattzufinden hat und welche Wiederherstellungsverfahren im Falle des Auftretens eines Fehlers verwendet werden sollen?

Projektstandards

Projektverwaltung. Es gibt eine aktuelle Liste mit allen Dateigruppen des Projekts samt Erklärung der Zusammenhänge. Die Struktur und Verwendung der gemeinsamen Projektumgebung wird festgelegt.

Review des Entwurfs

Prinzipien

- *Abstraktion.* Die Anwendungsebene wird von der technischen Ebene der Maschine durch eine Hierarchie mehrerer Ebenen abstrakter virtueller Maschinen getrennt.
- *Modularität.* Das System wird in mehrere unabhängig bearbeitbare Teile unterteilt.
- *Kohäsion.* Jeder Teil bearbeitet einen gut abgegrenzten Problembereich.
- *Koppelung.* Die Teile hängen so wenig wie möglich von einander ab.
- *Information Hiding.* Daten werden durch Zugriffsfunktionen gekapselt.
- *Schrittweise Verfeinerung.* Der Entwurf geht über die Schritte Architektur, Subsysteme, Komponenten, Datenstrukturen und Algorithmen vom Allgemeinen zum Speziellen vor.

Qualitätskriterien

- *Verständlich.* Die Notation der Entwurfselemente ist so einfach und klar wie möglich.
- *Namensgebung.* Bedeuten die Namen etwas, die in der Komponente verwendet wurden? Bedeutungsvolle Namen spiegeln die Namen von Eigenschaften der Realität wider.
- *Dokumentation.* Ist die Komponente so gut dokumentiert, daß der Zusammenhang zwischen den „Real-world“-Eigenschaften und der Komponente klar ist? Sind die Gedankengänge für diese Zusammenhänge dokumentiert?
- *Anpaßbar.* Der Entwurf ist an wahrscheinliche Änderungen leicht anpaßbar.
- *Verfolgbar.* Die Anforderung ist in den Entwurf verfolgbar, der Entwurf ist in die Implementierung verfolgbar, sodaß nachprüfbar ist, ob die Implementierung die Anforderungen umsetzt.

- *Robust.* Die Anwendung reagiert auf außergewöhnliche Umstände in definierter Art und Weise.
- *Komplexität.* Die Datenstrukturen und Algorithmen sind nicht komplexer als das Problem selbst.
- *Wartbar.* Verständlich, änderbar und testbar.

Vorbereitungen für die Implementierung

Vorbereitung der Implementierung

- Wurde der Entwurf von allen Betroffenen, etwa Entwicklern, Testern und Wartungsprogrammierern überprüft und für gut geheißen?
- Wurde bestätigt, daß der Entwurf die Anforderungen erfüllt?
- Wurden benötigte Werkzeuge zur Entwicklung (Software und Hardware) beschafft oder wurde ihre Entwicklung geplant?
- Wurden Pläne für die Implementierungsphase, betreffend Arbeitspakete, Qualitätsmanagement und Konfigurationsmanagement erstellt oder überprüft?

Vorbereitung für Integration und Systemtest

- Wurde bestätigt, daß die Abnahmekriterien verwendbar sind?
- Gibt es Überlegungen zu einer sinnvollen Integrationsreihenfolge?
- Gibt es ein Konzept für das Vorgehen zur Installation einer Systemkopie?
- Welche Voraussetzungen müssen Testdaten zumindest erfüllen?
- Wurden geeignete Simulationen zur Überprüfung des Zeitverhaltens und der Leistung geplant?

Vorbereitung für Betrieb und Wartung

- Ist die Architektur für die Umsetzung wahrscheinlicher Änderungen geeignet?
- Wurden genügend Diagnosehilfen bedacht?
- Wurden Konsistenzüberprüfungsmechanismen für die Datenbanken bedacht?
- Wurden geeignete Wartungswerkzeuge eingeplant, z.B. Wartungsumgebungen, Konfigurationskontrollsysteme?
- Sind die verschiedenen Systemressourcen (etwa Speicher, Plattenspeicher, Prozessorzeit) sowohl für jetzige wie auch für zukünftige Zwecke ausreichend groß dimensioniert?

Entwerfen von Komponenten

Hohe Kohäsion – innerer Zusammenhalt

- Die Komponente hat einen klaren Schwerpunkt und bietet ein entsprechendes Spektrum an Leistungen.
- Die Komponente bearbeitet eine zusammengehörende Menge Datenstrukturen.
- Sind die Leistungen der Komponente komplett, sodaß andere Komponenten nicht in die internen Datenstrukturen dieser Komponente eindringen müssen?

Lose Koppelung – geringe Abhängigkeit von außen

- Ist die Schnittstelle konsistent mit dem Architekturentwurf?
- Erfordert der Komponentenentwurf zusätzlich Voraussetzungen oder Einschränkungen, die Auswirkungen an anderen Stellen im System haben?
- Die Komponente hängt über wohldefinierte schmale Schnittstellen von anderen Modulen ab.

Die Komponente ist so weit wie möglich unabhängig von anderen Komponenten.

- Kann man die Komponente als Black-Box verwenden?

Implementierung

- Wurden für die einzelnen Komponenten Ziele für Zeit- oder Speicherverbrauch festgesetzt?
- Wurden die lokalen Datenstrukturen geeignet definiert?
- Wurden ausschließlich objektorientierte bzw. strukturierte Programmkonstrukte verwendet?
- Sind die Designdetails mit der gewählten Programmiersprache umsetzbar?
- Wurden betriebssystemunabhängige oder sprachunabhängige Funktionen verwendet?
- Wurden Fehlerbehandlung und -vorbeugung bzw. defensives Programmieren spezifiziert?

Entwerfen einer Objektmethode

Grundlegende Überlegungen

- Die Methode löst genau ein wohldefiniertes Problem.
- Die Methode hat einen klaren, verständlichen Namen (Verb und Objekt).
- Die Methode hat eine klare, verständliche Schnittstelle.
- Die Methode verwendet als Daten im Normalfall nur Parameter und lokale, initialisierte Variable.
- Die Methode überprüft Eingabeparameter auf Gültigkeit und Sonderfälle; die Methode reagiert klar und sinnvoll auf Sonderfälle.
- Teile, die sich als sinnvolle Routinen eignen, werden auch als eigene Routinen beschrieben.
- Die Methode ist testbar.
- Die Methode ist leicht verständlich. Haben Sie sich in einschlägigen Büchern über geeignete Standardalgorithmen informiert?

- Haben Sie allfällige Annahmen als Kommentar beschrieben?

Defensives Programmieren

- Schützt sich die Methode selbst vor ungültigen Eingabedaten?
- Wurden die Prinzipien des Information Hiding, der losen Koppelung und der Überprüfung von Daten verwendet, um die Auswirkungen von Fehlern auf andere Programmteile zu minimieren?
- Werden die Rückgabewerte von Funktionen immer ausgewertet?
- Werden Annahmen über den Zustand von Variablen durch Abfragen im Code überprüft?
- Gibt es Diagnosehilfen zum Finden von Fehlern, und können diese Hilfen ohne besondere Umstände ein- und ausgeschaltet werden?
- Wird der Diagnosecode in einem ausgelieferten System dem Anwender bei Problemen helfen, oder ist er nur für eingeweihte Entwickler nutzbringend und entsprechend (de)aktivierbar?

Checklisten Testen

Vorbereitungen

Analyse der Systemstruktur

Verfolgbarkeitsanalyse. Die funktionalen Anforderungen an das System werden den Klassen der Analyse bzw. des Entwurfs zugeordnet.

- Sind die in der Analyse gestellten Anforderungen an das System testbar? Müssen Teile der Analyse ergänzt oder konkretisiert werden?
- Gibt es Klassen, denen keine Anforderungen zugeordnet werden können?
- Gibt es Anforderungen an das System, die keiner Klasse zugeordnet werden können?

Einteilung des Systems in Schichten. Das System wird in aufeinander aufbauende Schichten von Klassen eingeteilt. Selbständige Teilsysteme werden identifiziert.

- Gibt es Klassen, die zyklisch aufeinander aufbauen?

Einteilung in Integrationsschritte

Überblick über die Integrationsschritte. Ein kurzer Überblick über die Integration, Teile und ihre Abhängigkeiten.

- Die Integrationsschritte bauen auf einander auf.
- *Beschreibung der einzelnen Integrationsschritte.* Die Integrationsschritte werden etwas ausführlicher beschrieben (betroffene Klassen/Module, Abhängigkeiten, vom Teilsystem erfüllte Anforderungen).
- Werden nach dem letzten Integrationsschritt alle Anforderungen erfüllt?
- Bestehen weitere, bisher unerkannte Abhängigkeiten zwischen Klassen oder Subsystemen?

Integrationsvorgehen

Einleitung. Ein kurzer Überblick über das Integrationsvorgehen.

Durchführung der Integrationsschritte. Beschreibt die Rahmenbedingungen für die Durchführung der Integration (Integrationsplattform etc.).

- Welche Besonderheiten unterscheiden die Integrationsplattform von der Entwicklungs- bzw. der Zielplattform?
- Worauf ist bei der Durchführung der Integration besonders zu achten?
- Wie wird bei Modifikationen an bereits integrierten Klassen/Modulen vorgegangen?

Integrationsbericht. Kurze Beschreibung des Integrationsberichts (Aufbau, Inhalt).

Testplan

Einleitung

Ein kurzer Überblick über das Testvorgehen.

Testrahmen

Testrahmen. Beschreibt die Rahmenbedingungen für die Durchführung der Systemtests (Testplattform, Reihenfolge, Einbettung in Integration, Vorbedingungen für den Test).

- Welche Besonderheiten unterscheiden die Testplattform von der Integrationsplattform?
- Wie wird bei Modifikationen an bereits getesteten Klassen/Modulen vorgegangen?
- Wie werden die Ergebnisse der Testfälle überprüft?

Teststrategie

- Ist der Zweck der Tests klar?
- Gibt es eine geeignete Teststrategie zur Erfüllung des Zwecks? Wurden Prioritäten verteilt?

Testfälle

Einleitung. Kurze Einleitung zur Beschreibung der Testfälle.

Testfälle nach Integrationsschritten. Pro Anwendungsfall müssen genügend Testfälle zur Überprüfung aller Normal-, Sonder und Fehlerfälle vorhanden sein. Die einzelnen Testfälle werden nach den Integrationsschritten gruppiert, nach deren Abschluß sie durchgeführt werden sollen, und detailliert beschrieben.

- Werden alle Anforderungen an das System getestet?
- Der Ablauf jedes Testfalls ist so zu beschreiben, daß der Tester weiß, was er konkret tun soll.
- Wie kann der Tester das tatsächliche Ergebnis vom erwarteten Ergebnis unterscheiden?

Testzeitplan

Testzeitplan. Pro durchzuführenden Test sind in Abstimmung mit dem Projektplan Termine festzulegen.

- Werden alle Integrationsschritte berücksichtigt?
- Wurde der Plan unter Testern, Programmierern und der Projektleitung abgestimmt?
- Wurde genügend Zeit für Fehlerkorrektur und Regressionstests eingerechnet?

Integrationsbericht

Einleitung

Überblick zum Dokument. Was ist wo zu finden.

- Gibt es strukturelle Unterschiede zu der im Systemtestplan festgelegten Struktur?

Integrationsvorgehen

Abweichungen und Ergänzungen zum geplanten Integrationsvorgehen sollen hier detailliert festgehalten werden.

- Waren alle benötigten Klassen/Module in der Endfassung verfügbar?
- Wurden zusätzliche Klassen/Module benötigt?

Integrationschritte und Systemtests

Zusammenfassung der Ergebnisse der Integrationschritte und Testberichte der Teilsysteme zu den jeweiligen Testsitzungen.

- Wurden alle Ergebnisse und Vorkommnisse des Integrationschrittes dokumentiert?
- Wurden die Daten der Integrationssitzung (Datum, Dauer, beteiligte Personen etc.) festgehalten?
- Wurde das Ziel des Integrationschrittes erreicht? Welche weiteren Aktivitäten sind geplant?
- Wurde der korrespondierende Systemtestbericht beigelegt?

Testbericht zu einem (Teil)System

Einleitung

In eigener Sache. Hier können besondere Vorkommnisse und Erfahrungen dokumentiert werden. *Rahmenbedingungen.* Wann wurde was von wem getestet, mit welchem Zeitaufwand.

Testvorgehen

Identifizieren der Version des getesteten Systems und der verwendeten Testdatenbank.

Abweichungen und Ergänzungen zum geplanten Integrationsvorgehen sollen hier detailliert festgehalten werden.

- Wurde die geplante Testkonfiguration verwendet?

Testfälle

Dokumentation der durchgeführten Testfälle und ihrer Ergebnisse. Der Schwerpunkt liegt dabei auf Fehlersituationen, in denen das tatsächliche Er-

gebnis vom erwarteten abweicht.

- Welche Eingabedaten wurden konkret verwendet (exakte Auflistung)?
- Wurde bei der Durchführung des Tests ein unerwartetes Systemverhalten beobachtet?
- Wurde das Testziel erreicht? Welche Probleme sind aufgetreten?
- Welche Korrekturen müssen vorgenommen werden?

Fehleraufzeichnungen

Beim Erstellen der Fehlerstatistik ist es vernünftig eine Klassifikation der Fehlertypen zu erstellen, um Gebiete in denen Schwachstellen auftreten einzugrenzen. Die folgende Liste kann als Basis für eigene Aufzeichnungen dienen.

Anforderungen

- Vergessen einer Funktion, obwohl sie in der Funktionalen Spezifikation vorhanden ist.
- Fehlinterpretation einer festgelegten Anforderung.

Schnittstellen

- Fehlerhafte Schnittstellen zwischen Komponenten.
- Fehlerhafte Schnittstellen mit der Außenwelt (Peripheriegeräte, Anwender, Betriebssystem, Softwarepakete, usw.).

Leistung

- Probleme beim Zeitverhalten.
- Speicherplatzprobleme (primärer oder sekundärer Speicher).

Datenstrukturen

- Fehlerhafte oder inkonsistente Datenstrukturen, fehlende Deklaration oder Spezifikation (bei Programm oder Daten).

- Falsche Benennung von Variablen, inkonsistente Datenbehandlung (z. B. Typ, Feldgröße, Genauigkeit, etc.), Fehler bei Grenzwerten.
- Initialisierungsfehler, Terminierungsfehler (z. B. Variable in falschem Zustand zurückgelassen)

Algorithmen

- Syntaxfehler, fehlerhafte Kommentare.
- Fehlerhafte Auswertung arithmetischer Ausdrücke, fehlerhafte boolesche Ausdrücke.
- Falsche Reaktion auf Auswertung logischer Bedingungen.
- Falsche Schleifenbehandlung (z. B. vorzeitiges oder zu spätes Abbrechen).
- Vergessen von Fällen (bei CASE).
- Falsche Reihenfolge von Aktionen, fehlende Aktionen.
- Indizierungsfehler.
- Fehlerhafte oder fehlende Wiederanlaufmechanismen.

Checklisten Anwenderdokumentation

Anwenderhandbuch und Online-Hilfe

Das Anwenderhandbuch dient dem neuen Anwender zum Erlernen des Systems und dem Fortgeschrittenen zum Nachschlagen. Es kann grob schon nach der Spezifikation geschrieben werden, Einzelheiten werden nach dem Systemtest hinzugefügt.

Das Anwenderhandbuch sollte in zwei, klar von einander getrennte Bereiche unterteilt sein – den Lehrbuchteil und den Nachschlageteil. Die beiden Bereiche unterscheiden sich in Gliederung, Sprache und Stil.

Lehrbuchteil

Der Lehrbuchteil ist ein in sich geschlossener Text, der durchgehend gelesen wird, d.h. Füllstellen sowie Redundanz in Form zusammenfassender Wiederholungen sind sinnvoll; Beispiele sind in den Text integriert und der gesamte Aufbau ist auf die Anforderungen des Lesers als Schüler ausgerichtet.

Einleitung. Kurze Beschreibung des Systems, Anwendungsbereich, Anwendungsvoraussetzungen (z.B. Geräte, Software, Datenbanken, Benutzerberechtigungen), Systemüberblick

Systemfunktionen. Welche Funktionen bietet das System an? Was kann das System, was kann es nicht? Wo liegen die Systemgrenzen (Speicher, Daten, Zeit etc.)?

- Die anzusprechenden Anwendergruppen werden identifiziert (Anwendungswissen, EDV-Hintergrund, Häufigkeit der Computerbenutzung).
- Alle Soll-Betriebsabläufe werden im Handbuch erläutert.

Benutzerschnittstellen. Darstellung der tatsächlich implementierten Benutzerschnittstelle (Ein- und Ausgaben, Ausdrücke) mit erklärenden Beispielen. Zuordnung zu den Funktionen, Hinweise auf Besonderheiten und Fehlermöglichkeiten.

- In der Benutzerschnittstelle des laufenden Programmes wird auf die entsprechende Seite im Handbuch verwiesen.
- Es gibt Beispiele zur Auflockerung und Erläuterung des Textes.
- Es gibt Diagramme und Tabellen für einen besseren Überblick.

Nachschlageteil (Referenz)

Der Nachschlageteil besteht aus vielen, stark voneinander unabhängigen Teilen, die jeweils allein gelesen werden können müssen. Die dabei entstehende Redundanz darf daher nicht zusammenfassend sein, sondern muß aus wörtlichen Wiederholungen bestehen. Im Nachschlageteil kann man das im Lehrbuchteil vermittelte Wissen voraussetzen. Durch Querverweise kann man den Leser zu Stellen navigieren, die das Verständnis fördern.

Der Nachschlageteil kann als Online-Hilfe implementiert werden. Alle Hilfestellungen zur Installation des Systems und zum Zugriff auf die Online-Hilfe müssen sinnvollerweise auf Papier vorliegen.

- Es gibt ein herausnehmbares Übersichtsblatt mit den wichtigsten Systembefehlen, der Systemstruktur o.ä.
- Maßnahmen für Notfälle werden erläutert (einschließlich der notwendigen Kontaktpersonen)
- Für Abkürzungen und Fachausdrücke gibt es ein Glossar.
- Es gibt einen Index.
- Es gibt ein Verzeichnis der Tabellen bzw. Abbildungen.

- Das Handbuch wurde einmal von Anfang bis Ende verwendet, um es auf Korrektheit und Verwendbarkeit zu überprüfen; mit anderen Worten: Das Handbuch wurde getestet.

Stichworte, Glossar

Durch die Anwendung eines gut zusammengestellten Stichwortverzeichnisses kann man dem Leser zusätzlich helfen, rasch die gewünschte Information zu erhalten, ohne bereits Bekanntes lesen zu müssen.

Fehlerliste

Eine Liste aller schwerwiegenden Systemfehler, die Wartungsarbeiten erfordern, mit Bezug zu Fehlermeldungen am Bildschirm, Erklärung und Hinweis auf weitere Schritte (Anleitung zur Fehlerbehebung durch den Benutzer, Informationen zu Hotline usw.).

Online-Hilfe

Die Online-Hilfe soll so strukturiert sein, daß sie den Anfänger von den Grundlagen zu den spezielleren Teilen führt. In der Online-Hilfe soll nach Stichworten aus der Anwenderschnittstelle gesuchte werden können. Insbesondere Begriffe aus Betriebsabläufen, Dialogen und Fehlermeldungen sollen in der Hilfe zu finden sein. Die Online-Hilfe ergänzt durch ihren Nachschlagecharakter das Anwenderhandbuch, das vor allem zum Einlesen zu Beginn und bei grundlegenden Problemen mit der Installation bzw. dem Einstieg ins Programm genutzt wird.

Checklisten Projektmanagement

Alle aktuellen Dokumente werden in einer *Projektmappe* (am besten Ringordner) gesammelt und durch Zwischenblätter zum schnelleren Auffinden getrennt. Die vollständige Projektmappe ist bei jeder Review abzugeben.

Projektstart

Projektdeckblatt

Das Deckblatt enthält eine Tabelle mit Namen, Telefonnummern (wann erreichbar) aller Beteiligten. Darüber hinaus identifiziert es das Team, Projektbezeichnung, Koordinator und Administrator sowie deren Stellvertreter. Einige Zeilen zur Kurzbeschreibung des Projekts.

- Für alle Teammitglieder gibt es Kürzel (für die verschiedenen Listen und Formulare).

Dokumentationsrichtlinien

Einheitliches Aussehen von Dokumenten im Team. Kurzes Dokument, in dem für jeden zu erzeugenden Dokumenttyp eindeutige Vorgaben für die Erstellung enthalten sind.

- Es gibt eine einheitliche Vorgabe für Schrift, Absatz, Abschnitt und Dokumentformatierung.
- Das Aussehen von Quelltext und dem darin enthaltenen Kommentar ist vorgegeben.
- Es gibt ein Beispieldokument auf Diskette, das Überschriften-, Text- und Tabellenformatierung enthält.
- Es gibt eine einheitliche Vorgabe für Schreib- und Zeichenprogramme.

Ressourcenliste

Überblick über vorhandene Ressourcen (Betriebs-

mittel) im Team.

Wer kann was zur Verfügung stellen? Was kostet das? Wie lange vorher muß der Bedarf bekannt sein? Welche externen Ressourcen können als Ersatz dienen?

- Der Zugriff im Team auf Computer, Drucker, Kopierer, wichtige Programme, Dokumentation, E-Mail etc. wird beschrieben.

Projektplan

Work Breakdown Structure (WBS)

Tabelle aller Aktivitäten des Projektes inkl. Verantwortlichem und Mitarbeitern.

- Für jede Aktivität ist der Aufwand (Tage geplant, bisher getan, noch zu tun) beschrieben.
- Jede Aktivität hat eine Priorität (ABC) zugeordnet und ist hierarchisch numeriert.
- Das notwendige Ausmaß an Training (z.B. Einarbeitung in neue Werkzeuge) für Mitarbeiter ist im Zeitrahmen und Projektbudget berücksichtigt.
- Für den abgeschätzten Systemumfang sind entsprechende Entwurfs-, Codier- und Testaktivitäten in der WBS vorgesehen.

Netzplan

- Netzplan und Balkendiagramm sind zusammengesetzt.
- Netzplan und Balkendiagramm haben eine erklärende Legende.
- Für jede Aktivität sind der späteste Start, das späteste Ende und die Dauer ersichtlich.
- Alle Abhängigkeiten zwischen Aktivitäten sind überprüft und eingetragen.
- Alle Aktionen der WBS auf unterster Ebene sind mit Verantwortlichem im Netzplan vorhanden.

Balkendiagramm

- Für alle Aktionen im Balkendiagramm werden die Mitarbeiter, der späteste Start und das späteste Ende dargestellt.
- Ist-Balkendiagramm: Die tatsächlichen Arbeitstermine im Laufe des Projektes sind (händisch) eingezeichnet.
- Alle Aktionen der WBS sind mit Verantwortlichem und Dauer im Balkendiagramm ersichtlich.

Projektmappe

Projekttagbuch

Liste mit den Spalten: Datum, Beginn, Dauer, Ereignis, Personen, Verweis auf Dokumente; Was ist wann passiert, wer war daran beteiligt, wo finde ich nähere Informationen darüber?

Der Teamkoordinator führt das Tagebuch auf Projektebene, jeder Dokumentverantwortliche führt über die Arbeiten an seinem Dokument ein Tagebuch, und jeder, der bei einem Dokument mitarbeitet, führt Buch über seine Arbeitszeit (Stundenliste).

- Für jedes Ereignis sind Beginn, Verantwortlicher und ev. ein Verweis auf ein Dokument mit den Details (z.B. Besprechungsprotokoll) vorhanden.
- Aus dem Tagebuch gehen Datum, Dauer und Kurzbeschreibung aller teamweiten Ereignisse hervor.

Besprechungsprotokolle

Was war der Zweck der Besprechung? Welche Fragen und Probleme wurden tatsächlich besprochen? Was ist dabei herausgekommen? Was soll im weiteren passieren? Wer war bei der Entscheidungsfindung anwesend?

- Das Protokoll enthält Teambezeichnung und Projektbezeichnung.

- Im Protokoll stehen Datum, Dauer, Anlaß der Besprechung. Darüber hinaus werden die entsprechenden Entscheidungen und durchzuführende Aktionen (wer, bis wann) aufgezeichnet.
- Das Protokoll enthält Ort, Beginn und Anwesenheit der Besprechung sowie Verweise auf Dokumente mit weiteren Details.
- Es enthält zu jeder Frage bzw. zu jedem Problem eine entsprechende Entscheidung.
- Größere Aktivitäten werden sofort in die WBS übernommen bzw. in die nächste Version des Projektplanes.

Versionsmanagement – Dokumenttagebuch

Zu jeder Zeit muß klar sein, welche Versionen eines Dokumentes aktuell sind. Die aktuelle Version jedes Dokumentes muß rechtzeitig vor der Review verfügbar sein. Falls notwendig, muß es möglich sein, auf eine frühere Version eines Dokumentes zurückzugreifen (Archiv).

- Für jedes Dokument ist die aktuelle Version notiert.
- Es gibt ein Abhängigkeitsdiagramm für alle Projektdokumente.
- Die Sicherung und Verteilung wichtiger Daten ist geplant (Archivprogramme, passende Datenformate etc.).
- Das Dokumenttagebuch enthält Verweise auf alle Dateien, die für die Herstellung und Dokumentation des Systems benötigt werden.
- Das Dokumenttagebuch enthält für alle Dokumente eine Liste mit der aktuellen Versionsnummer und ev. Hinweise, was gegenüber der letzten Version verändert wurde.

Entwicklungsumgebung

Definition einer einheitlichen Entwicklungsumgebung für das Team. Planung der Zielkonfiguration, in der das System endgültig laufen soll.

Hardware. Computertyp (Mindestanforderung), Massenspeicher (z.B. Diskettenlaufwerk, Hard-disk), Drucker, sonstige Hardwareanforderungen
Software. Ordnerstruktur, Suchpfade (samt Reihenfolge), Inhalt von Konfigurationsdateien, Treiber, Programme, Virenschutz, Archivprogramme

- Welche Software darf *nicht* installiert sein (falls bekannt)?

Testbares (Teil)System

Ein funktionsfähiger Teil des Systems oder das ganze System ist in der Entwicklungsumgebung funktionsfähig installiert. Das System durchläuft folgende Stadien: *raw* – die einzelnen Module sind übersetzbar mit den Schnittstellen aus dem Systementwurf vorhanden, können aber nicht integriert werden; *medium* – bestimmte Modulgruppen sind mit eingeschränkter Funktion ausführbar, aber nicht stabil; *done* – das (Teil)System ist funktional ausgereift und robust einsetzbar.

- Das (Teil)System ist wohldefiniert (was ist drin, was nicht). Es gibt eine Liste aller Dateien, die zum (Teil)System gehören; diese Dateien sind auch vorhanden.
- Die zum Testen notwendigen Testdaten sind vorhanden.
- Das Navigieren zwischen *allen* Masken des aktuellen Teilsystems ist möglich.
- Es treten keine ernstesten unerwünschten Seiteneffekte auf (Absturz, Datenzerstörung oder -verfälschung).
- Alle bisher integrierten Services wurden verwendet (auch in Grenz- und Fehlersituationen).

Projektstatusbericht

Wie sieht der Stand des Projektes zur Zeit aus?

Der Koordinator faßt die Statusberichte der aktiven Teammitglieder zusammen.

Was ist seit dem letzten Bericht passiert?

Was läuft gerade?

Wo gibt es Probleme?

Was passiert als Nächstes?

Entspricht der Projektplan noch dem Stand des Projektes? Falls nicht, ist er zu aktualisieren.

- Das Projekttagbuch ist am letzten Stand (inkl. Stundenlisten der Teammitglieder).
- Es gibt ein aktuelles Ist-Balkendiagramm (die tatsächliche Arbeit laut Stundenliste wird manuell im Soll-Balkendiagramm eingezeichnet).
- Die bisher erfaßbaren Daten für die Projektstatistik werden in einem eigenen Abschnitt dokumentiert und erläutert.

Reviewbericht

Welches Ziel hatte die Review? Welches Material wurde der Review unterzogen? Wer war in welcher Rolle beteiligt? Was ist das Ergebnis der Review?

Mängelliste. Dokument/Ort des Mangels, Mangel, Beheber, Behebungstermin

Mögliche Ergebnisse der Review: In Ordnung, Re-Review, Projektanhaltung und Projektabbruch.

In Ordnung: Alles OK.

ReReview. Bei wichtigen Dokumenten wurden Mängel gefunden; eine weitere Review ist daher notwendig, diese wird am __.__.__ stattfinden. Die nächste Phase kann trotz der Mängel begonnen werden.

Projektanhaltung: Bei kritischen Dokumenten wurden schwerere Mängel gefunden; das Projekt wird bis zur Behebung der folgenden Mängel angehalten: ..., die Re-Review findet am __.__.__ statt.

Projektabbruch: Schwere Mängel an kritischen Dokumenten konnten bis zur ReReview nicht behoben werden; das Projekt wird wegen ... abgebrochen.

- Datum, Anlaß, reviewte Dokumente, Ort und Beschreibung gefundener Mängel, Entschei-

dungen und zu planende Aktionen (wer, bis wann) der Review sind protokolliert.

- Das Gesamtergebnis der Review (ev. ein Termin für die ReReview) ist festgehalten.
- Das Protokoll identifiziert Beginn, Dauer und Anwesende der Review.

Projektende

Projektpräsentation

Das Produkt und das Projekt werden in einer Kundenpräsentation etwa 20 bis 30 Minuten vorgestellt.

- Es gibt eine Broschüre, der die Vorteile des Produktes für den Kunden zur Geltung bringt.

Projektendbericht

Was war das Projektziel? Wurde es erreicht? Wenn ja, mit welchem Aufwand? Wenn nein, warum nicht? Was heißt das für zukünftige Projekte der Teammitglieder?

Die Ergebnisse des Projektes (entstandene Software, Dokumente, Zeitaufwand, Einnahmen/ Ausgaben, gewonnene Erfahrung (Probleme und Lösungen)) werden beschrieben.

Welche Techniken (virtuelle Maschine (HW und Systemsoftware), SE-Methoden und Werkzeuge für alle Phasen, Teamstruktur, Kontakt zu Kunden) wurden verwendet? Waren sie erfolgreich? Wenn ja, in welcher Weise? Wenn nein, warum nicht? Hatten Sie das Gefühl, das Projekt immer gut im Griff zu haben? Welche Punkte könnten/sollten zu den Checklisten hinzugefügt werden?

Der Bericht enthält Feedback zur Projektumgebung: Was war besonders gut/ eher weniger gut. Was könnte in Zukunft besser gemacht werden?

Verteilung

Wer erhält welche Versionen von welchen Dateien auf welchem Medium? Das System wird für späte-

re Untersuchungen archiviert.

- Es gibt eine Verteilerliste, falls es mehrere Versionen des Systems und Kunden gibt.
- Es gibt eine passende Installationsanleitung auf der Diskette
- Alle notwendigen Testdatenbanken sind auf der Diskette.
- Eine lauffähige Version des Systems befindet sich auf der Diskette.
- Es gibt eine ausreichend kommentierte Packliste des Disketteninhaltes.
- Die Dateien sind in entpacktem Zustand auf Freiheit von Viren geprüft.

Installation

Hier stehen alle Informationen, die benötigt werden, um das System in einer Zielumgebung zu installieren bzw. technische Informationen, die nicht im Anwenderhandbuch stehen.

Für jede Zielumgebung:

- Generierung des Systems, Erstinstallation, benötigte Hardware und Software.
- Möglichkeiten zur Hilfe (Hotline).
- Datensicherung und Archivierung.
- Logbuch.
- Restart nach Fehlern.

Vorgangsweise:

- Welche Personen werden benötigt?
- Einrichten von Dateien und Datenbanken.
- Systembefehle, Installationsbefehle, erwartete Ausgaben.
- mögliche Fehler, mögliche Ursachen, weitere Schritte.
- Tests zur Überprüfung der erfolgreichen Installation.
- Laufwerk und Ordner für die Installation sollen vom Installierer wählbar sein.