

SWT-Praktikum: Prozesshandbuch



1 Inhaltsverzeichnis

2 [Abkürzungsverzeichnis](#)

3 [Zweck](#)

4 [Zusammenfassung](#)

5 [Modell der Software-Entwicklung](#)

5. 1 [Rollen der Software-Entwicklung](#)

5. 2 [Phasen der Software-Entwicklung](#)

5. 2.1 [Machbarkeitsanalyse](#)

5. 2.2 [Analyse](#)

5. 2.3 [Planung](#)

5. 2.4 [Entwurf](#)

5. 2.5 [Implementierung](#)

5. 2.6 [Test](#)

5. 2.7 [Projektabschluss](#)

5. 3 [Produkte der Phasen der Software-Entwicklung](#)

6 [Gliederung und Inhalt der Dokumente](#)

7 [Glossar](#)

8 [Literatur](#)



SWT-Praktikum: Prozesshandbuch



2 Abkürzungsverzeichnis

KM	Konfigurationsmanagement
PM	Projektmanagement
QS	Qualitätssicherung
SE	Software-Entwicklung / System-Erstellung
IT	Informationstechnik



SWT-Praktikum: Prozesshandbuch



3 Zweck

Dieses Dokument soll die Studierenden in die Lage versetzen, die geforderte Dokumentation im Praktikum Software-Technik in der gewünschten Form zu erstellen und die [Software-Entwicklung](#) vorgabengemäß durchzuführen.



Das Konfigurationsmanagement ist dafür verantwortlich, alle Zwischenprodukte und sonstige Dokumente zu speichern. Dabei werden auch unterschiedliche Versionen vorgehalten und der jeweilige Status jedes Dokumentes notiert. Typische Dokumentationszustände sind: geplant, in Bearbeitung, vorgelegt, akzeptiert.

SWT-Praktikum: Prozesshandbuch



4 Zusammenfassung

Ein allgemein anwendbares Modell zur [Software-Entwicklung](#) wird vorgestellt. Die [Phasen](#), in die die Bearbeitung zerlegt wird, sowie die [Rollen](#), die von den Bearbeitern eingenommen werden und die zu erstellenden Zwischenergebnisse, die sogenannten [Produkte](#), werden beschrieben.

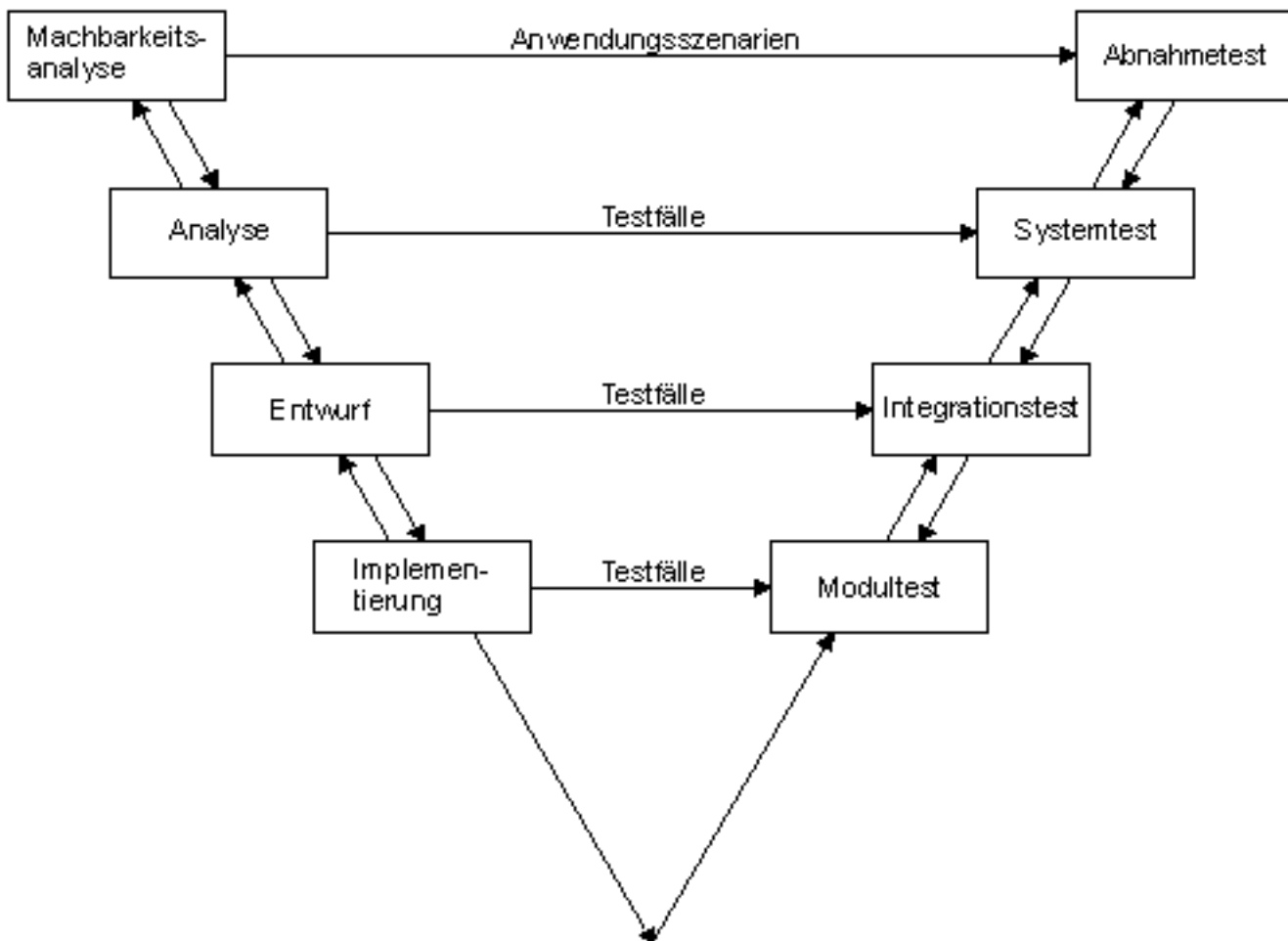
Dieses Modell wird auf die Software-Entwicklung übertragen und die dabei auftretenden speziellen Phasen, Rollen und Produkte werden beschrieben. Zu den zu erstellenden Dokumenten werden Mustergliederungen vorgegeben.



SWT-Praktikum: Prozesshandbuch

5 Modell der Software-Entwicklung

Im Folgenden wird ein Modell der Software-Entwicklung dargestellt. Es umfasst [Rollen](#), die unterschiedliche Verantwortungsbereiche widerspiegeln, [Phasen](#) in die sich der gesamte [Prozess](#) unterteilen lässt und [Produkte](#), die während des Entwicklungsprozesses entstehen und im Praktikum abzugeben sind.



Das V-Modell verdankt seinen Namen der grafischen Darstellung in V-Form. Seine [Produktiv-Phasen](#) entsprechen denen des Wasserfallmodells. Sie sind im absteigendem Zweig des Vs dargestellt. Ergänzend befinden sich im aufsteigendem Zweig die dazugehörigen Testphasen.



SWT-Praktikum: Prozesshandbuch



5.1 Rollen der Software-Entwicklung

Der Begriff **Software** steht in diesem Umfeld für ein Programm und alle zugehörigen Dokumente um das Programm herum.

Im Vorgehensmodell /KBSt92/ /VMod97/ des Bundes werden vier Verantwortungsbereiche für verschiedenen Aktivitäten in IT-Vorhaben definiert. Es sind:

- System-Erstellung (SE)
- Projektmanagement (PM)
- Qualitätssicherung (QS)
- Konfigurationsmanagement (KM)

Die **System-Erstellung** umfasst die Erstellung aller Dokumente auf dem Weg zum auslieferungsfähigen Endprodukt einschließlich des Programms. Daher werden die System-Ersteller auch Autoren genannt, insbesondere wenn die Zwischenprodukte Dokumente sind und keine Programme.

Das **Projektmanagement** plant und überwacht Umfang und Termin des Einsatzes von Personal, Maschinen, Räumen und sonstigen Hilfs- und Betriebsmitteln. Gegenüber dem Management werden regelmäßig Statusberichte zum [Projekt](#) gegeben. Sie enthalten insbesondere einen Soll-Ist-Vergleich der Aufwände und Termine. Der Repräsentant des Projektmanagement ist der Projektleiter.

Die **Qualitätssicherung** überprüft die erstellten [Produkte](#) und Zwischenprodukte der Software-Erstellung gemäß den Vorgaben des angewendeten [Prozesses](#). Die Erstellung von Testprogrammen gehört dazu.

Das **Konfigurationsmanagement** ist dafür verantwortlich, alle Zwischenprodukte und sonstige Dokumente zu speichern. Dabei werden auch unterschiedliche Versionen vorgehalten und der jeweilige Status jedes Dokumentes notiert. Typische Dokumentationszustände sind: geplant, in Bearbeitung, vorgelegt, akzeptiert.

In großen Projekten werden die Aufgaben von mehreren Personen wahrgenommen. Beispielsweise beim Projektmanagement kann aber letztendlich nur einer verantwortlich sein, nämlich der Projektleiter. Im Praktikumsumfeld sind die Aufgaben nicht so groß, dass hier mehrere Personen beteiligt sind. PM, QS und KM sollen deshalb neben den Verantwortungsbereichen gleichzeitig die entsprechende verantwortliche Person bezeichnen. SE steht in dem Sinne auch für System-ErstellerIn oder auch mehrere SystemerstellerInnen.



SWT-Praktikum: Prozesshandbuch



5.2 Phasen der Software-Entwicklung

Teile der folgenden Ausführung sind nicht für sich allein verständlich, sondern setzen die Kenntnis der entsprechenden Kapitel der Vorlesung Software-Technik voraus.

Vor einer Software-Entwicklung werden grundsätzliche Überlegungen darüber angestellt, ob und wie die Anforderungen des [Kunden](#) zu erfüllen sind. Die geschieht in der Machbarkeitsanalyse. Es folgt die Analyse, in der festgelegt wird, was das zu erstellende System alles umfassen soll. Im Entwurf wird festgelegt, wie das System aufgebaut ist und wie es die Anforderungen des [Auftraggebers](#) erfüllen soll. Darauf folgen noch die Implementierung und der Test.

In allen [Phasen](#) werden in Übereinstimmung mit den Zielen des Praktikums die benötigten Bearbeitungszeiten und gefundene Fehler protokolliert. Beobachtungen über das Zusammenwirken mit Kollegen und dem Umfeld werden notiert. Für Phasen, in denen wichtige Produkte entstehen, planen die Projektgruppen gesonderte Überprüfungsphasen ein.

Zu jeder Phase, in der produktiv am System gearbeitet wird, kann eine Qualitätssicherungs-Phase vorgesehen werden. In manchen Beschreibungen taucht diese zugehörige Phase gar nicht mehr auf, da sie grundsätzlich in jede [Produktiv-Phase](#) integriert ist. Eine ausdrückliche 1:1-Zuordnung von Produktiv- zu Qualitätssicherungs-Phase wird wegen der besseren Sichtbarkeit der QS im Praktikum empfohlen.

Die Phasen werden im Folgekapitel dargestellt. Die Beschreibung beschränkt sich in den ersten Abschnitten auf die allgemeine Zielsetzung einer Phase. Stehen einem Bearbeiter noch keine speziellen Methodenkenntnisse zur Verfügung, sollte diese Anleitung als grobe Vorgabe für die Strukturierung einer [Software-Entwicklung](#) ausreichen. Es folgt dann in den meisten Fällen eine Beschreibung, die sich an der Vorlesung orientiert. In späteren Kapiteln folgen dann noch für jede Phase die [Vorbedingungen](#) und [Nachbedingungen](#), die Ablaufbeschreibung und die Gliederungen der zu erzeugenden [Produkte](#).

Für große Software-Projekte werden die Phasen nicht genau einmal durchlaufen, sondern es können Teilsysteme definiert werden, die nach einer gemeinsamen Analyse die restlichen Phasen unabhängig voneinander durchlaufen und zum Schluss zusammengefügt werden.

Die evolutionäre Software-Entwicklung beginnt mit den Teilen, die dem Kunden mit dem geringsten Aufwand den größten Nutzen bringen. Dafür wird ein Kern-System entwickelt. Im Einsatz kann der Kunde seine Anforderungen an das Gesamtsystem problemlos aktualisieren und die weiteren Funktionalitäten werden in Zyklen, die jeweils die unten angegebenen Phasen enthalten realisiert.



SWT-Praktikum: Prozesshandbuch



5.2.1 Machbarkeitsanalyse

5.2.1.1 Überblick

Zunächst muss der [Auftragnehmer](#) die gestellte Aufgabe grob analysieren, um sicher zu gehen, dass er die [Anforderungen](#) vollständig und richtig verstanden hat. Dazu sind Rückfragen zur Beseitigung von Unklarheiten oder die Beschaffung von Hintergrundinformation erforderlich. Es ist zu klären, welche kritischen [Qualitätsmerkmale](#) zum Scheitern des [Projektes](#) bzw. zur Weigerung des [Kunden](#), das Endergebnis bzw. [Produkt](#) zu akzeptieren, führen können. Zu all diesen Punkten sind grundsätzliche Lösungsideen zu skizzieren und zu bewerten sowie das Risiko abzuschätzen, ein Qualitätsmerkmal nicht zu erfüllen. Auf dieser Basis wird die Entscheidung getroffen, ob das Projekt überhaupt machbar ist.

Das Produkt der Machbarkeitsanalyse ist die Machbarkeitsstudie.

Im Gegensatz zur später folgenden Analyse konzentriert sich die Machbarkeitsanalyse auf die Qualitätsmerkmale. Die Funktionalität des zu erstellenden Systems wird nur sehr grob beschrieben. Bezogen auf die Funktionalität kann die Machbarkeitsstudie als Grobanalyse und die spätere Analyse als Feinanalyse betrachtet werden.



SWT-Praktikum: Prozesshandbuch



5.2.1 Machbarkeitsanalyse

5.2.1.1 Überblick

Zunächst muss der [Auftragnehmer](#) die gestellte Aufgabe grob analysieren, um sicher zu gehen, dass er die [Anforderungen](#) vollständig und richtig verstanden hat. Dazu sind Rückfragen zur Beseitigung von Unklarheiten oder die Beschaffung von Hintergrundinformation erforderlich. Es ist zu klären, welche kritischen [Qualitätsmerkmale](#) zum Scheitern des [Projektes](#) bzw. zur Weigerung des [Kunden](#), das Endergebnis bzw. [Produkt](#) zu akzeptieren, führen können. Zu all diesen Punkten sind grundsätzliche Lösungsideen zu skizzieren und zu bewerten sowie das Risiko abzuschätzen, ein Qualitätsmerkmal nicht zu erfüllen. Auf dieser Basis wird die Entscheidung getroffen, ob das Projekt überhaupt machbar ist.

Das Produkt der Machbarkeitsanalyse ist die Machbarkeitsstudie.

Im Gegensatz zur später folgenden Analyse konzentriert sich die Machbarkeitsanalyse auf die Qualitätsmerkmale. Die Funktionalität des zu erstellenden Systems wird nur sehr grob beschrieben. Bezogen auf die Funktionalität kann die Machbarkeitsstudie als Grobanalyse und die spätere Analyse als Feinanalyse betrachtet werden.

5.2.1.2 Anforderungen

Alle [funktionalen Anforderungen](#), [Qualitätsmerkmale](#) und [Randbedingungen](#) sind anzugeben. Sie erhalten Namen, deren erster Buchstabe auf den Typ der Anforderung hinweist. Qualitätsmerkmale werden mittels PLANGUAGE spezifiziert /Gilb88/, d.h. es werden folgende Rubriken spezifiziert:

Name	Name des Qualitätsmerkmals
Erklärung	kurze Erklärung des Qualitätsmerkmals
Maßeinheit	Maßeinheit, in der das Qualitätsmerkmals gemessen wird
Test	stichwortartige Beschreibung des Tests der Einhaltung des Qualitätsmerkmals
Ist	heutiger Wert des Qualitätsmerkmals
Soll	geplanter Wert des Qualitätsmerkmals
Schlechtestens	schlechtester akzeptierter Wert des Qualitätsmerkmals ist der Wert schlechter, ist das Projekt gescheitert
Traumwert	Wert des Qualitätsmerkmals in den kühnsten Träumen
Quellenhinweise	woher die angegebenen Zahlen stammen

5.2.1.3 Lösungen

Zu jedem Qualitätsmerkmal wird mindestens eine Lösung angegeben. Jede Lösung erhält einen Namen und wird kurz beschrieben.

5.2.1.4 Lösungs-Bewertung

Die Lösungen werden hinsichtlich der Wechselwirkung auf die anderen Qualitätsmerkmale bewertet. Dazu werden die Qualitätsmerkmale den Lösungen in einer Matrix gegenüber gestellt. Im Kreuzungspunkt wird der Einfluss der Lösung auf die Erfüllung des Qualitätsmerkmals notiert. Eine Notation verwendet die Symbole --, -, o, +, ++, eine andere Prozentzahlen in 5er-Schritten zwischen -100 und +100.

5.2.1.5 Risiko- Bewertung

Für jedes Qualitätsmerkmal wird notiert welches Risiko der Erfüllung im Wege stehen könnte und was ggf. dagegen unternommen werden kann.

5.2.1.6 Qualitätssicherung der Machbarkeitsanalyse

In einem geeigneten Verfahren ist das Produkt der [Phase](#) zu überprüfen. Eine [Formale Inspektion](#) wird empfohlen.

5.2.1.7 Eingangs- und Ausgangsprodukte

	Produkte	Verantwortliche
Eingangsprodukt	Aufgabenbeschreibung	AuftraggeberIn
durchzuführende Tätigkeiten	<ul style="list-style-type: none"> ● Funktionale Anforderungen ● Qualitätsmerkmale ● Randbedingungen ● Lösungen ● Lösungsbewertung ● Risikoanalyse ● Risikobewertung ● Identifizierung von Testfällen für den Abnahmetest 	Softwareentwicklung (SE)
Ausgangsprodukt	Machbarkeitsstudie	Projektmanagement (PM)

Nachbedingungen	Qualitätssicherung der Machbarkeitsstudie mit geeignetem Verfahren. Formale Inspektion empfohlen. Machbarkeitsstudie im Konfigurationsmanagment abgeben.	Qualitätssicherung (QS) Konfigurationsmanagment (KM)
-----------------	---	---



SWT-Praktikum:

Prozesshandbuch



5.2.2 Analyse

5.2.2.1 Überblick

In der Analyse geht es darum, mit dem [Kunden](#) genau abzustimmen welche Aufgaben das System erledigen soll, welche Daten in welchen Formaten und Zusammenhängen auftreten und welche Funktionalitäten erwartet werden. Gegenüber dem späteren Entwurf wird hier die Frage nach dem **Was** gestellt und die Beantwortung der Frage nach dem **Wie** bewusst dem Entwurf überlassen. Informationen über Datenvolumina oder zu erwartende Aufrufhäufigkeiten sind für den späteren Entwurf ebenfalls von Bedeutung. Häufig neigen Ingenieure dazu Lösungen zu präsentieren, bevor die Aufgabenstellung wirklich klar ist. Vorsicht!

Die Analyse beschreibt also das bestehende Umfeld bzw. das geplante System bezüglich der

- auftretenden Daten
- erforderlichen Funktionalitäten

unter Berücksichtigung

- der erwarteten Datenmengen
- der erwarteten Aufrufhäufigkeiten bzw.
- der geforderten Bearbeitungsdauer
- verschiedener Randbedingungen

Zur Befragung des Kunden werden Interviews durchgeführt und unterschiedliche Befragungstechniken angewendet. Ein wichtiger Aspekt dabei ist immer die Rückkopplung, in der überprüft wird, ob das was der [Auftragnehmer](#) verstanden hat und in seiner Darstellung dem Kunden präsentiert, noch dem entspricht was der Kunde gemeint hat. [Prototypen](#), dienen ebenfalls der Klärung der [Anforderungen](#).

Je nach der gewählten Analysemethode stehen verschiedene Aspekte im Vordergrund und unterschiedliche Diagrammarten werden verwendet:



SWT-Praktikum: Prozesshandbuch



5.2.2 Analyse

5.2.2.1 Überblick

In der Analyse geht es darum, mit dem [Kunden](#) genau abzustimmen welche Aufgaben das System erledigen soll, welche Daten in welchen Formaten und Zusammenhängen auftreten und welche Funktionalitäten erwartet werden. Gegenüber dem späteren Entwurf wird hier die Frage nach dem **Was** gestellt und die Beantwortung der Frage nach dem **Wie** bewusst dem Entwurf überlassen. Informationen über Datenvolumina oder zu erwartende Aufrufhäufigkeiten sind für den späteren Entwurf ebenfalls von Bedeutung. Häufig neigen Ingenieure dazu Lösungen zu präsentieren, bevor die Aufgabenstellung wirklich klar ist. Vorsicht!

Die Analyse beschreibt also das bestehende Umfeld bzw. das geplante System bezüglich der

- auftretenden Daten
- erforderlichen Funktionalitäten

unter Berücksichtigung

- der erwarteten Datenmengen
- der erwarteten Aufrufhäufigkeiten bzw.
- der geforderten Bearbeitungsdauer
- verschiedener Randbedingungen

Zur Befragung des Kunden werden Interviews durchgeführt und unterschiedliche Befragungstechniken angewendet. Ein wichtiger Aspekt dabei ist immer die Rückkopplung, in der überprüft wird, ob das was der [Auftragnehmer](#) verstanden hat und in seiner Darstellung dem Kunden präsentiert, noch dem entspricht was der Kunde gemeint hat. [Prototypen](#), dienen ebenfalls der Klärung der [Anforderungen](#).

Je nach der gewählten Analysemethode stehen verschiedene Aspekte im Vordergrund und unterschiedliche Diagrammartentypen werden verwendet:

5.2.2.2 Objektorientierte Analyse

Das Analysemodell der objektorientierten Analyse enthält die folgenden Diagramme und kommentiert sie.

- Anwendungsfall-Diagramm
- Klassen-Diagramm
- Interaktions-Diagramm
- evtl. Zustands- Diagramm

Die Anwendungsfälle werden gruppiert und zu [Paketen](#) zusammengefasst. Verschiedene Pakete können durch verschiedene Personen bzw. Gruppen realisiert werden.

Für die häufig gewünschte grafische Oberfläche wird ein [Prototyp](#) erstellt, an Hand dessen der Kunde seine Anforderungen präzisieren kann.

In einem [Data Dictionary](#) werden die Beschreibungen der Klassen zusammengestellt.

5.2.2.3 Qualitätssicherung der Analyse

In einem geeigneten Verfahren ist das Produkt der Phase zu überprüfen.

5.2.2.4 Eingangs- und Ausgangsprodukte

	Produkte	Verantwortliche
Vorbedingungen	Machbarkeitsstudie vom Konfigurationsmanagement akzeptiert.	Konfigurationsmanagement (KM)
Eingangsprodukt	Machbarkeitsstudie	Projektmanagement (PM)
durchzuführende Tätigkeiten	Objektorientierte Analyse <ul style="list-style-type: none"> ● Anwendungsfall-Diagramm ● Klassen-Diagramm ● Interaktions-Diagramm ● Zustandsdiagramm (falls erforderlich) ● Identifizierung von Testfällen für den Systemtest 	Softwareentwicklung (SE)
Ausgangsprodukt	Analysemodell	Projektmanagement (PM)
Nachbedingungen	Qualitätssicherung des Analysemodells mit geeignetem Verfahren. Analysemodell im Konfigurationsmanagement abgeben.	Qualitätssicherung (QS) Konfigurationsmanagement (KM)



SWT-Praktikum: Prozesshandbuch



5.2.3 Planung

5.2.3.1 Überblick

Die Planung steht hier hinter der Analyse, da die Aufwandsabschätzung ein Mindestmaß an Kenntnis über das zu erstellende System erfordert. Bei viel Erfahrung und in Abstimmung mit dem Detaillierungsgrad der Funktionsbeschreibung der Machbarkeitsstudie kann die Aufwandsschätzung bereits nach der Machbarkeitsanalyse erfolgen. Wir wollen auf Nummer sicher gehen und keine Versprechungen machen, ohne eine Basis zu haben, der wir selber trauen.

In der Planung wird die Bearbeitung der gesamten Aufgabe in die vorgesehenen [Phasen](#) zerlegt. Die Zerlegung folgt dem hier vorgegebenen Muster. Es ist festzulegen welche Qualitätssicherungsphasen vorgesehen sind. Für jede Phase werden die zu erstellenden [Produkte](#) sowie die Verantwortlichkeiten festgelegt und der Aufwand geschätzt. Für die gesamte Entwicklung wird ein Terminplan erstellt.

Die Planung kann vor Beendigung der Analyse anlaufen, sobald in der Analyse genügend detaillierte Informationen für eine seriöse Planung vorliegen.



SWT-Praktikum: Prozesshandbuch



5.2.3 Planung

5.2.3.1 Überblick

Die Planung steht hier hinter der Analyse, da die Aufwandsabschätzung ein Mindestmaß an Kenntnis über das zu erstellende System erfordert. Bei viel Erfahrung und in Abstimmung mit dem Detaillierungsgrad der Funktionsbeschreibung der Machbarkeitsstudie kann die Aufwandsschätzung bereits nach der Machbarkeitsanalyse erfolgen. Wir wollen auf Nummer sicher gehen und keine Versprechungen machen, ohne eine Basis zu haben, der wir selber trauen.

In der Planung wird die Bearbeitung der gesamten Aufgabe in die vorgesehenen [Phasen](#) zerlegt. Die Zerlegung folgt dem hier vorgegebenen Muster. Es ist festzulegen welche Qualitätssicherungsphasen vorgesehen sind. Für jede Phase werden die zu erstellenden [Produkte](#) sowie die Verantwortlichkeiten festgelegt und der Aufwand geschätzt. Für die gesamte Entwicklung wird ein Terminplan erstellt.

Die Planung kann vor Beendigung der Analyse anlaufen, sobald in der Analyse genügend detaillierte Informationen für eine seriöse Planung vorliegen.

5.2.3.2 Die Planung im Praktikum

Für die oben aufgezählten Aufgaben werden unterstützende Formblätter zur Verfügung gestellt.

5.2.3.3 Qualitätssicherung der Planung

In einem geeigneten Verfahren ist das Produkt der Phase zu überprüfen.

5.2.3.4 Eingangs- und Ausgangsprodukte

	Produkte	Verantwortliche
Vorbedingungen	Analysemodell vom Konfigurationsmanagement akzeptiert.	Konfigurationsmanagement (KM)
Eingangsprodukt	Machbarkeitsstudie, Analysemodell	Projektmanagement (PM)

durchzuführende Tätigkeiten	<ul style="list-style-type: none"> ● Aufwandsabschätzung ● Terminplanerstellung ● Verantwortlichkeiten festlegen ● zu erstellende Produkte festlegen 	Projektmanagement (PM)
Ausgangsprodukt	Projektplan	Projektmanagement (PM)
Nachbedingungen	<p>Qualitätssicherung des Projektplans mit geeignetem Verfahren.</p> <p>Projektplan im Konfigurationsmanagement abgeben.</p>	<p>Qualitätssicherung (QS)</p> <p>Konfigurationsmanagement (KM)</p>



SWT-Praktikum: Prozesshandbuch



5.2.4 Entwurf

5.2.4.1 Überblick

Im Entwurf geht es nun darum, ein System zu konstruieren, das all die geforderten [Funktionalitäten](#) und [Qualitätsmerkmale](#) aus der Machbarkeitsstudie und der Analyse erfüllt. Hier kann und soll der Entwerfer seine Kreativität und sein handwerkliches Geschick anwenden um festzulegen, wie das System die [Anforderungen](#) erfüllen soll. Endlich kann gestaltet werden.

Zunächst sind einige grundsätzliche Fragen zu beantworten, die die Systemarchitektur grundsätzlich beeinflussen.

- Sollen Programmteile parallel ausführbar sein?
- Ist das System für einen oder mehrere gleichzeitige Nutzer auszulegen?
- Sollen Teilsysteme auf verschiedene Rechner verteilt werden?
- Sind Echtzeit-Anforderungen zu erfüllen?
- Ist das System interaktiv oder läuft es im Stapelbetrieb?
- Werden Datenstrukturen im Hauptspeicher, in Dateien oder in einer Datenbank gehalten?
- Haben die geforderten Qualitätsmerkmale Einfluss auf die Systemarchitektur?

Daraus resultiert eine Strukturierung des Systems in [Schichten](#) und [Partitionen](#).



SWT-Praktikum: Prozesshandbuch



5.2.4 Entwurf

5.2.4.1 Überblick

Im Entwurf geht es nun darum, ein System zu konstruieren, das all die geforderten [Funktionalitäten](#) und [Qualitätsmerkmale](#) aus der Machbarkeitsstudie und der Analyse erfüllt. Hier kann und soll der Entwerfer seine Kreativität und sein handwerkliches Geschick anwenden um festzulegen, wie das System die [Anforderungen](#) erfüllen soll. Endlich kann gestaltet werden.

Zunächst sind einige grundsätzliche Fragen zu beantworten, die die Systemarchitektur grundsätzlich beeinflussen.

- Sollen Programmteile parallel ausführbar sein?
- Ist das System für einen oder mehrere gleichzeitige Nutzer auszulegen?
- Sollen Teilsysteme auf verschiedene Rechner verteilt werden?
- Sind Echtzeit-Anforderungen zu erfüllen?
- Ist das System interaktiv oder läuft es im Stapelbetrieb?
- Werden Datenstrukturen im Hauptspeicher, in Dateien oder in einer Datenbank gehalten?
- Haben die geforderten Qualitätsmerkmale Einfluss auf die Systemarchitektur?

Daraus resultiert eine Strukturierung des Systems in [Schichten](#) und [Partitionen](#).

5.2.4.2 Objektorientierter Entwurf

Der objektorientierte Entwurf unterteilt sich in den Systementwurf und den Objektentwurf. Beim Systementwurf sind die im Überblick aufgelisteten grundlegenden Entscheidungen zu treffen.

Beim Objektentwurf werden unter Berücksichtigung der Laufzeit- und Speichereffizienz

- die Klassendefinitionen vervollständigt die Algorithmen der Methoden ausgewählt und nicht triviale Algorithmen in Pseudo-Code oder einer ähnlichen, geeigneten Methode ausformuliert
- geeignete Behälterklassen ausgewählt
- Interne Klassen zur Implementierung definiert
- die Implementierung der Assoziationen festgelegt

5.2.4.3 Qualitätssicherung des Entwurfs

In einem geeigneten Verfahren ist das Produkt der Phase zu überprüfen. Eine [Formale Inspektion](#) wird empfohlen.

5.2.4.4 Eingangs- und Ausgangsprodukte

	Produkte	Verantwortliche
Vorbedingungen	Analysemodell vom Konfigurationsmanagement akzeptiert.	Konfigurationsmanagement (KM)
Eingangsprodukt	Analysemodell	Projektmanagement (PM)
durchzuführende Tätigkeiten	<ul style="list-style-type: none"> ● Systemarchitektur bestimmen (z.B. Parallelität, Multiuser, verteiltes System, Echtzeitanforderungen, Interaktivität, Datenbankanbindung) ● Objektorientierter Entwurf ● Identifizierung von Testfällen für den Integrationstest 	Softwareentwicklung (SE)
Ausgangsprodukt	Entwurfsdokumentation	Projektmanagement (PM)
Nachbedingungen	<p>Qualitätssicherung der Entwurfsdokumentation mit geeignetem Verfahren. Formale Inspektion wird empfohlen.</p> <p>Entwurfsdokumentation im Konfigurationsmanagement abgeben.</p>	<p>Qualitätssicherung (QS)</p> <p>Konfigurationsmanagement (KM)</p>



SWT-Praktikum: Prozesshandbuch



5.2.5 Implementierung

5.2.5.1 Überblick

Der Entwurf des Systems wird in der vorgesehenen Programmiersprache implementiert.
[Programmierrichtlinien](#) sind dabei zu beachten.



SWT-Praktikum:

Prozesshandbuch



5.2.5 Implementierung

5.2.5.1 Überblick

Der Entwurf des Systems wird in der vorgesehenen Programmiersprache implementiert. [Programmierrichtlinien](#) sind dabei zu beachten.

5.2.5.2 Qualitätssicherung der Implementierung

In einem geeigneten Verfahren ist das Produkt der Phase zu überprüfen.

5.2.5.3 Eingangs- und Ausgangsprodukte

	Produkte	Verantwortliche
Vorbedingungen	Entwurfsdokumentation vom Konfigurationsmanagement akzeptiert.	Konfigurationsmanagement (KM)
Eingangsprodukt	Entwurfsdokumentation	Projektmanager (PM)
durchzuführende Tätigkeiten	<ul style="list-style-type: none"> ● Implementierung des Programms unter Einhaltung der Programmier-Richtlinien. (tippeditippeditipp) ● Identifizierung von Testfällen für den Modultest 	Softwareentwicklung (SE)
Ausgangsprodukt	Programm	Projektmanager (PM)
Nachbedingungen	<p>Qualitätssicherung des Programms mit geeignetem Verfahren.</p> <p>Programm im Konfigurationsmanagement abgeben.</p>	<p>Qualitätssicherung (QS)</p> <p>Konfigurationsmanagement (KM)</p>



SWT-Praktikum: Prozesshandbuch



5.2.6 Test

5.2.6.1 Überblick

Der Test setzt lauffähige Programmmodule voraus. Ein Modul kann also erst nach seiner Implementierung getestet werden. Es muss jedoch nicht die komplette Umgebung eines Moduls existieren, bevor es getestet wird. Diese Umgebung kann durch Testprogramme ersetzt werden. Es ist möglich, bezogen auf den zu testenden Modul aufrufende als auch aufgerufene Module durch Ersatzmodule zu simulieren.

Erhält der Tester ein Programm zum Test, erhebt sich die Frage, was nun alles zu testen ist. Je nach dem, ob der Tester das Programm einsehen kann oder nur seine Schnittstelle kennt, unterscheidet man [Black-Box-Tests](#) und [Glas-Box-Tests](#).

In jeder [Produktiv-Phase](#) von der Analyse bis zum detaillierten Entwurf gibt es andere Schwerpunkte der Entwicklung. Es ist nun nahe liegend, bereits im Entwicklungsprozess Testfälle zu bestimmen und zu dokumentieren, die dann später durchgeführt werden können.

Durchgeführte Tests müssen nachvollziehbar und nachweisbar sein. Deshalb sind sie zu dokumentieren.



SWT-Praktikum: Prozesshandbuch



5.2.6 Test

5.2.6.1 Überblick

Der Test setzt lauffähige Programmmodule voraus. Ein Modul kann also erst nach seiner Implementierung getestet werden. Es muss jedoch nicht die komplette Umgebung eines Moduls existieren, bevor es getestet wird. Diese Umgebung kann durch Testprogramme ersetzt werden. Es ist möglich, bezogen auf den zu testenden Modul aufrufende als auch aufgerufene Module durch Ersatzmodule zu simulieren.

Erhält der Tester ein Programm zum Test, erhebt sich die Frage, was nun alles zu testen ist. Je nach dem, ob der Tester das Programm einsehen kann oder nur seine Schnittstelle kennt, unterscheidet man [Black-Box-Tests](#) und [Glas-Box-Tests](#).

In jeder [Produktiv-Phase](#) von der Analyse bis zum detaillierten Entwurf gibt es andere Schwerpunkte der Entwicklung. Es ist nun nahe liegend, bereits im Entwicklungsprozess Testfälle zu bestimmen und zu dokumentieren, die dann später durchgeführt werden können.

Durchgeführte Tests müssen nachvollziehbar und nachweisbar sein. Deshalb sind sie zu dokumentieren.

5.2.6.2 Testplanung

In den Produktiv-Phasen können bereits Testfälle identifiziert werden, die besonders kritische Fälle darstellen oder sicherstellen sollen, dass die Funktionalität vollständig und korrekt erbracht wird. Jeder geplante Testfall ist mit folgenden Informationen zu dokumentieren

- Testidentifikation
- Testziel
- Testbeschreibung inkl. Eingabedaten
- Testumgebung
- erwartete Testausgabe

Wichtig ist, die erwartete Testausgabe wirklich vor der Testdurchführung zu dokumentieren, da sonst die Gefahr besteht, dass man zu unkritisch mit den gelieferten Testergebnissen ist.

Die Dokumentation der Eingabedaten kann sich auch in Dateien befinden, die unmittelbar zum Lauf des Tests verwendet werden. Die Nachvollziehbarkeit muss gewährleistet sein.

5.2.6.3 Testdurchführung

Die Tests werden wie geplant durchgeführt. Die tatsächlichen Ausgaben und/oder Reaktionen werden

dokumentiert und mit den erwarteten Ausgaben verglichen.

5.2.6.4 Eingangs- und Ausgangsprodukte

	Produkte	Verantwortliche
Vorbedingungen	Programmmodule vom Konfigurationsmanagement akzeptiert.	Konfigurationsmanagement (KM)
Eingangsprodukt	Programmmodule	Projektmanager (PM)
durchzuführende Tätigkeiten	<ul style="list-style-type: none"> ● Testplanung, wichtig: erwartete Testergebnisse vor der Testdurchführung dokumentieren! ● Testdurchführung 	Softwareentwickler (SE)
Nachbedingungen	Testdokumentation im Konfigurationsmanagement abgeben.	Konfigurationsmanagement (KM)
Ausgangsprodukt	Testdokumentation	Projektmanager (PM)



SWT-Praktikum: Prozesshandbuch



5.2.7 Projektabschluss

Im Projektabschluss werden die Aufwands- und Fehlerzahlen zusammengetragen. Gemeinsam mit den persönlichen Notizen zum Umfeld werden sie analysiert und bewertet und falls erforderlich Maßnahmen zur Verbesserung des Prozesses oder des Umfeldes eingeleitet.



SWT-Praktikum: Prozesshandbuch



5.3 Produkte der Phasen der Software-Entwicklung

Die Ergebnisse einer Phase werden Produkte genannt. Zu den vorher beschriebenen allgemeinen Phasen gibt es folgende Zwischenergebnisse:

Grundsätzlich sind für die Phasen Machbarkeitsanalyse bis einschließlich Test jeweils folgende Produkte zu erstellen bzw. zu erweitern.

- Aufwandsstatistiken, in denen die Beteiligten ihre aufgewendeten Zeiten notieren (Formular Zeiterfassung)
- Fehlerprotokolle, in denen die aufgetretenen Fehler, deren Art und die Dauer der Behebung notiert werden (Formular Fehlerstatistik)
- Umfeldnotizen, in denen Sie auftretende Probleme, Verbesserungsvorschläge und sonstige Anmerkungen notieren können.
- Projektstatus: mit Hilfe der Formulare Termin- und Aufwandsplanung soll der aktuelle Projektstatus darlegbar sein.

Phase	Produkt
Machbarkeitsanalyse	Machbarkeitsstudie
	Testplan (Abnahmetest)
Planung	Projektplan
Analyse	Analysemodell
	Testplan (Systemtest)
Entwurf	Entwurfs-Dokumentation
	Testplan (Integrationstest)
Implementierung	Programm
	Testplan (Modultest)
Test	Testdokumentation <ul style="list-style-type: none"> ● Testbeschreibungen ● Testprotokolle
Abnahmetest	Abnahmeprotkoll
Formale Inspektion	Inspektionsbericht

Projektabschluss	
alle	Produktablage

- Jedes Dokument muss im Konfigurationsmanagement abgeliefert werden.
- Die Qualitätssicherung ist für die Tests (Testpläne und Testdurchführung) verantwortlich.
- Für die Vollständigkeit und die Abgabe der erforderlichen Produkte ist **uns gegenüber** immer der Projektleiter / das Projektmanagement (PM) verantwortlich.





SWT-Praktikum: Prozesshandbuch

6 Gliederung und Inhalt der Dokumente

Die Produkte aus dem vorherigen Kapitel sind bis auf das Programm selbst, alles Dokumente.

Auflistung der einzelnen Dokumente und ihre Gliederung:

- Zu **jedem** Dokument gehören
 - Titelblatt
 - Inhaltsverzeichnis
 - Abkürzungsverzeichnis
 - Glossar
- **Aufgabenbeschreibung**
- **Machbarkeitsanalyседokument**
 - Anforderungen
 - Funktionale Anforderungen
 - Qualitätsmerkmale
 - Randbedingungen
 - Lösungen
 - Identifizierung der Lösungen
 - Bewertung der Lösungen
 - Risiken
 - Identifizierung der Risiken
 - Bewertung der Risiken
- **Analysedokument**
 - Anwendungsfalldiagramm
 - Klassen
 - Klassenbeschreibung (je Klasse 1 - 2 Sätze)
 - Klassendiagramm
 - Interaktionsdiagramm
 - Zustandsdiagramm
 - Oberflächenprototyp

- **Entwurfsdokument**

- Systemarchitektur
 - Art des Systems
 - Struktur des Systems
 - Teilsysteme
 - Verteilung auf Prozessoren und Prozesse
 - Datenspeicherung
 - globale Betriebsmittel / Synchronisation
 - Steuerung
- Objektentwurf
- Entwurfsoptimierung
- Entwurf von Assoziationen

- **Implementierungsdokument**

- Implementierung des Programms unter Einhaltung der Programmierrichtlinien.

- **Testdokument**

- In der Machbarkeitsanalyse identifizierte Testfälle für den Abnahmetest
- In der Analyse identifizierte Testfälle für den Systemtest
- Im Entwurf identifizierte Testfälle für den Integrationstest
- In der Implementierung identifizierte Testfälle für den Modultest
- Testplanung. **Wichtig:** erwartete Testergebnisse vor der Testdurchführung dokumentieren!
- Testdurchführung

- **Managementdokument**

- zu erstellende Produkte festlegen
- Verantwortlichkeiten festlegen
- Aufwandsabschätzung
- Terminplanerstellung
- Aufwandsstatistiken
- Fehlerprotokolle
- QS-Berichte mit Fehlerprotokollen z.B. von den formalen Inspektionen
- Umfeldnotizen
- Projektstatus

- **Handbuch**



SWT-Praktikum: Prozesshandbuch



8 Literatur

Gilb88	<p>Gilb, T. Principles of Software Engineering Management Addison Wesley 1988 ISBN 0-201-19246-2</p>
KBSt92	<p>Koordinierungs- und Beratungsstelle der Bundesregierung für Informationstechnik in der Bundesverwaltung Planung und Durchführung von IT-Vorhaben in der Bundesverwaltung - Vorgehensmodell (V-Modell) Schriftenreihe der KBSt, Band 27/1 und 27/2 August 1992</p>
VMod97	<p>Entwicklungsstandard für IT-Systeme des Bundes, Vorgehensmodell, Teil 1: Regelungsteil, Teil 3: Handbuchsammlung, Allgemeiner Umdruck Nr. 250/1, Juni 1997, BWB IT 15, Koblenz</p>
Balz96	<p>Balzert, Helmut Lehrbuch der Software-Technik, Software-Entwicklung Spektrum Akademischer Verlag GmbH 1996 ISBN 3-8274-0042-2</p>
Balz98	<p>Balzert, Helmut Lehrbuch der Software-Technik, Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung Spektrum Akademischer Verlag GmbH 1998 ISBN 3-8274-0065-1</p>
Dumk93	<p>Reiner Dumke Software Engineering Vieweg 1993 ISBN 3-528-153355-5</p>



SWT-Praktikum:

Prozesshandbuch



7 Glossar

Anforderungen	Unterscheidung in funktionale Anforderungen , die vom entwickelten System erfüllt sein müssen, Qualitätsmerkmale , die quantifiziert werden und einen Grad der Erfüllung haben und die Randbedingungen , die vorgegeben sind (Hardware, Software u.a.)
Auftraggeber	Der Kunde / Auftraggeber ist derjenige, der das Softwareprodukt in Auftrag gibt und es später auch bezahlt, sofern er zufrieden ist. Er beschreibt, was er von dem Produkt erwartet (Anforderungen).
Auftragnehmer	Der Auftragnehmer erfüllt die Anforderungen des Auftraggebers. Über die Lösungen beschreibt er, wie er die Anforderungen erfüllen wird.
Black-Box-Test	Im Black Box Test werden die Module getestet, ohne deren Programmtext zu kennen. Sie sind rein funktionsorientiert. Es wird geprüft, ob sich aus den Eingabedaten die erwarteten Ausgabedaten erzeugen lassen.
Data Dictionary	Data Dictionary ist ein Verzeichnis, das Informationen über die Struktur von Daten, ihre Eigenschaften sowie ihre Verwendung enthält. Datenstrukturen und -elemente werden in der Backus-Naur-Form beschrieben.
Formale Inspektion	Formale Inspektion ist eine Methode der Qualitätssicherung, nach der in genau definierten Phasen mehrere Inspektoren in verschiedenen Rollen ein vorliegendes Dokument unter Einhaltung der Vorgaben detailliert hinsichtlich Vollständigkeit, Korrektheit und Konsistenz begutachten.

Funktionale Anforderungen	Funktionale Anforderungen beschreiben eine Funktion, die ein Produkt erfüllen muss, kann auch als Inhaltsanforderung bezeichnet werden und bezieht sich auf den Umfang des Produktes Ein System verfügt über eine Funktion oder nicht. Einen Grad des Vorhandenseins gibt es nicht!
Glas-Box-Test	Im Glas-Box-Test werde die Module mit Kenntnis des Programmtextes getestet. Der Test ist kontroll- und datenflußorientiert.
Konfigurationsmanagement	Das Konfigurationsmanagement ist dafür verantwortlich, alle Zwischenprodukte und sonstige Dokumente zu speichern. Dabei werden auch unterschiedliche Versionen vorgehalten und der jeweilige Status jedes Dokumentes notiert. Typische Dokumentationszustände sind: geplant, in Bearbeitung, vorgelegt, akzeptiert.
Kunde	Der Kunde / Auftraggeber ist derjenige, der das Softwareprodukt in Auftrag gibt und es später auch bezahlt, sofern er zufrieden ist. Er beschreibt, was er von dem Produkt erwartet (Anforderungen).
Nachbedingung	Die Nachbedingung einer Phase umfasst die Liste der in dieser Phase zu erstellenden Produkte, sowie die Kriterien, die diese Produkte erfüllen müssen. z.B. die anzuwendende Art der Qualitätssicherung auf die Produkte.
Paket	Ein Paket ist eine Sammlung von Klassen, die zur Implementierung eines oder mehrere logisch zusammengehöriger Anwendungsfälle verwendet werden. In der Analysephase werden diese zusammengehörigen Anwendungsfälle ebenfalls als Paket bezeichnet.
Partition	Partitionen sind die vertikale Anordnung der Teilsysteme eines Systems. Partitionen teilen ein System in mehrere unabhängige oder lose gekoppelte Teilsysteme, von denen jedes eine Art Dienst bereitstellt.

Phase	<p>Die Softwareentwicklung wird in einzelne Phasen unterteilt. Machbarkeitsanalyse, Analyse, Planung, Entwurf, Implementierung, Test, Projektabschluss. Unterteilt man das Projekt in einzelne Teilbereiche (grafische Oberfläche, Berechnung u.a.), so werden die Phasen für jeden dieser Teilbereiche durchlaufen.</p>
Produkt	<p>In jeder Phase entstehen Produkte die zum Teil als Eingabeprodukte für die nächste Phase benötigt werden.</p>
Produktiv-Phase	<p>Als Produktiv-Phase bezeichnet man die Phasen in der Software-Entwicklung ohne die organisatorischen und Qualitätssicherungsphasen. Machbarkeitsanalyse, Analyse, Planung, Entwurf, Implementierung.</p>
Projekt	<p>Ein Projekt ist ein einmaliges Vorhaben begrenzter Dauer, das zur Erreichung eines definierten Zieles mit begrenzten Personal- und Sachmitteln durchgeführt wird.</p>
Projekthandbuch	<p>Ist die Konkretisierung des Prozesshandbuches für ein vorliegendes Projekt.</p>
Projektmanagement	<p>Das Projektmanagement plant und überwacht Umfang und Termin des Einsatzes von Personal, Maschinen, Räumen und sonstigen Hilfs- und Betriebsmitteln. Gegenüber dem Management werden regelmäßig Statusberichte zum Projekt gegeben. Sie enthalten insbesondere einen Soll-Ist-Vergleich der Aufwände und Termine. Der Repräsentant des Projektmanagement ist der Projektleiter.</p>
Prototyp	<p>Prototyp ist ein provisorisches Software-System, das während der Produktdefinition erstellt wird, um Anforderungsfragen zu klären oder Anforderungen zu veranschaulichen. Man unterscheidet das vertikale Prototyping, in dem nur einige Funktionen vollständig implementiert werden, sowie das horizontale Prototyping, bei dem man die ganze Funktionalität betrachtet, diese aber nur andeutungsweise (Bsp: Grafische Oberfläche) realisiert.</p>

Prozess	Ein Prozess beschreibt Aktivitäten, Methoden und Verfahren, die zur Entwicklung und Überprüfung von Software benötigt werden. /Balz-SM/
Prozesshandbuch	Das Prozesshandbuch beschreibt die erforderlichen Phasen zur Softwareentwicklung, inklusive der zu erstellenden Produkte.
Pseudo-Code	Pseudo-Code ist eine textuelle semiformale Darstellung von Kontrollstrukturen in Anlehnung an problemorientierte Programmiersprachen. /Balz-SE/
Qualitätsmerkmal	Qualitätsmerkmale werden auch Eigenschaftsanforderungen genannt. Sie müssen vor dem ersten Versuch einer Lösungsspezifikation spezifiziert und quantifiziert sein.
Qualitätssicherung	Die Qualitätssicherung überprüft die erstellten Produkte und Zwischenprodukte der Software-Erstellung gemäß den Vorgaben des angewendeten Prozesses. Die Erstellung von Testprogrammen gehört dazu.
Randbedingungen	Randbedingungen charakterisieren die Umgebung, in der das Produkt erstellt wird. Dazu gehören können: Hardware, Software, Betriebssystem, Entwicklungsumgebung, Programmiersprache, Entwicklungsstandards und Gesetze.
Rolle	Eine Rolle beschreibt die notwendigen Erfahrungen, Kenntnisse und Fähigkeiten, über die ein Mitarbeiter verfügen muss, um Aktivitäten durchzuführen. /Balz-SM/
Schicht	Gliederung einer Softwarearchitektur in hierarchische Schichten. Jede Schicht besteht aus Systemkomponenten, z.B. Benutzeroberfläche, eigentliche Anwendung, Datenhaltung.
Software-Entwicklung/System-Erstellung	Die Software-Entwicklung / System-Erstellung umfasst die Erstellung aller Dokumente auf dem Weg zum auslieferungsfähigen Endprodukt einschließlich des Programms. Daher werden die System-Ersteller auch Autoren genannt, insbesondere wenn die Zwischenprodukte Dokumente sind und keine Programme.

Softwareprodukt

Ein Softwareprodukt ist die Gesamtheit von Softwarekomponenten (Programmen, Dokumentationen usw.), die als Ganzes entwickelt, vertrieben, angewendet und gewartet werden. Es besteht aus:

- der Anwendungsbeschreibung (user manual), die die Handhabung der Software einschließlich der organisatorischen und technischen Voraussetzungen beschreibt.
- den eigentlichen Programmen, die zur Umsetzung der Funktionalität in verschiedenen Formen vorhanden sind.
- dem Installationsprogramm zur "Einrichtung" aller code-basierten Produktbestandteile auf einem konkreten Computer
- der Entwicklerdokumentation, die alle im Verlauf der Produktentwicklung entstandenen Dokumente, wie Skizzen, Diagramme, Prüflisten, Testdateien, Hilfsprogramme, Qualitätsberichte, Änderungsanleitungen usw., zusammenfasst.
- ein Demonstrationsprogramm, das den potentiellen Anwender über den Leistungsumfang und die Handhabung informiert.

/Dumk93/

Vorbedingung

Die Bedingungen, die vor Aufgabenbeginn erfüllt sein müssen. Eingabeprodukte.

