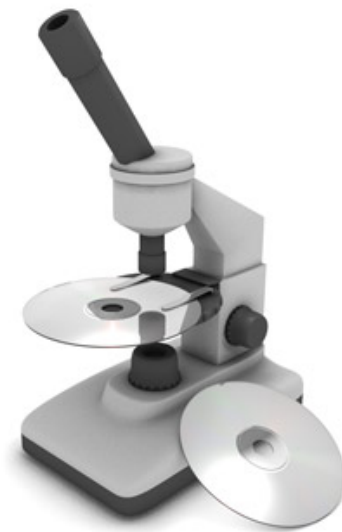


Hinweis:

Diese Druckversion der Lerneinheit stellt aufgrund der Beschaffenheit des Mediums eine im Funktionsumfang stark eingeschränkte Variante des Lernmaterials dar. Um alle Funktionen, insbesondere Animationen und Interaktionen, nutzen zu können, benötigen Sie die On- oder Offlineversion. Die Inhalte sind urheberrechtlich geschützt.
©2018 Beuth Hochschule für Technik Berlin

ANA - Analyse



Lernziele und Überblick



Lernziele

- Erst einmal sollen alle wesentlichen Aspekte der Analysephase vorgestellt werden. Sie sollen verstehen, welche Stufen und Aufgaben es in der Analysephase gibt und was alles zu Beginn und zum Ende der Analyse getan werden muss. Pflichten- und Lastenheft sollen Sie ebenso einordnen können wie Use-Cases und Requirements.
- Als Übung werden Sie ein eigenes Konzept einer Analyse zusammenzustellen
- Ihr eigenes Konzept sollen Sie dann in einer Diskussion mit der Kursbetreuung darstellen können und somit ihr Wissen und Anwendung vertiefen.

Die in der Lerneinheit vorgestellten Punkte der Analyse müssen also unbedingt zu einem eigenen Konzept zusammengestellt und selbst validiert werden.



Zeitbedarf und Umfang


- Für das aktive Lesen dieser Einheit benötigen Sie 2 Stunden.
- Für das Aufstellen und Anwenden einer eigenen Analyse brauchen Sie mindestens 4-5 Stunden.



Hinweis

Literatur


• Systemanalyse

 [Kra96] KRALLMANN, H. : Systemanalyse in Unternehmen.
München; Wien: Oldenburg Verlag 1996

 [RS04] SOPHIST GROUP / CHRIS RUPP: „Systemanalyse kompakt“

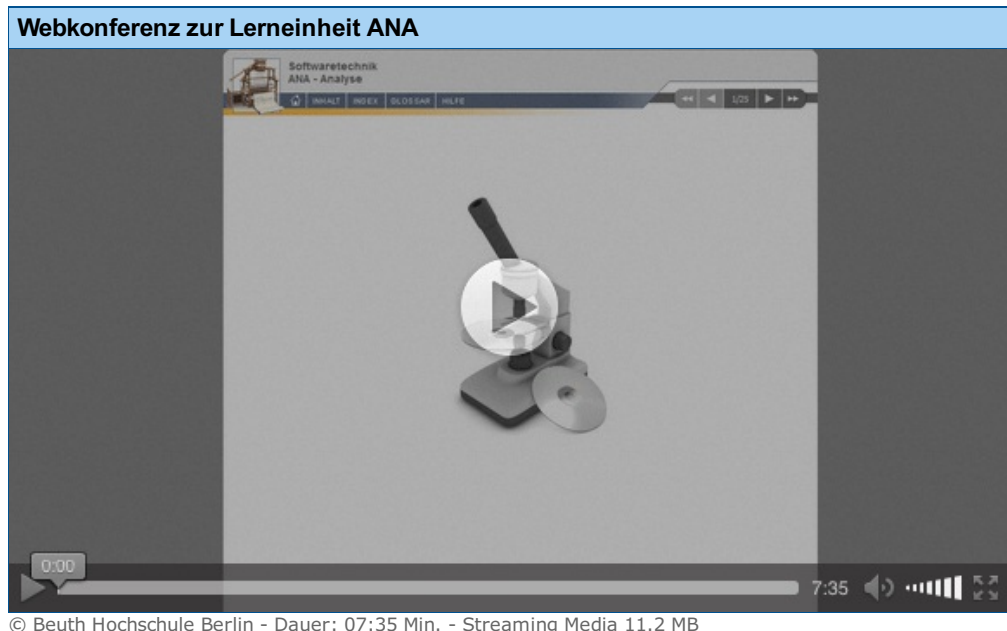
• Analyse

BALZERT, HELMUT Softwaremanagement: Lehrbuch der Softwaretechnik,
Spektrum Akademischer Verlag, ISBN-13: 978-3827411617

 [Oes98] OESTERREICH, BERND, „Objektorientierte Softwareentwicklung.
Analyse und Design mit UML“,
Oldenbourg, ISBN-13: 978-3486579260

OOSE INNOVATIVE INFORMATIK GMBH:

 <http://www.oose.de/oep/>



Die Hinweise auf klausurrelevante Teile beziehen sich möglicherweise nicht auf Ihren Kurs.
Stimmen Sie sich darüber bitte mit ihrer Kursbetreuung ab.

1 Analyse-Phase

Wichtige Punkte in der
Frühphase

In dieser Lerneinheit geht es nicht nur streng um alle Punkte der Analyse. Neben den klassischen Aufgaben der Analyse gibt es im Allgemeinen in den frühen Phasen weitere wichtige Punkte, die während der Analyse berücksichtigt werden sollten. Dazu gehören auch Überlegungen bezüglich des Vorgehensmodells, dem Projektmanagement oder der Projektkalkulation.

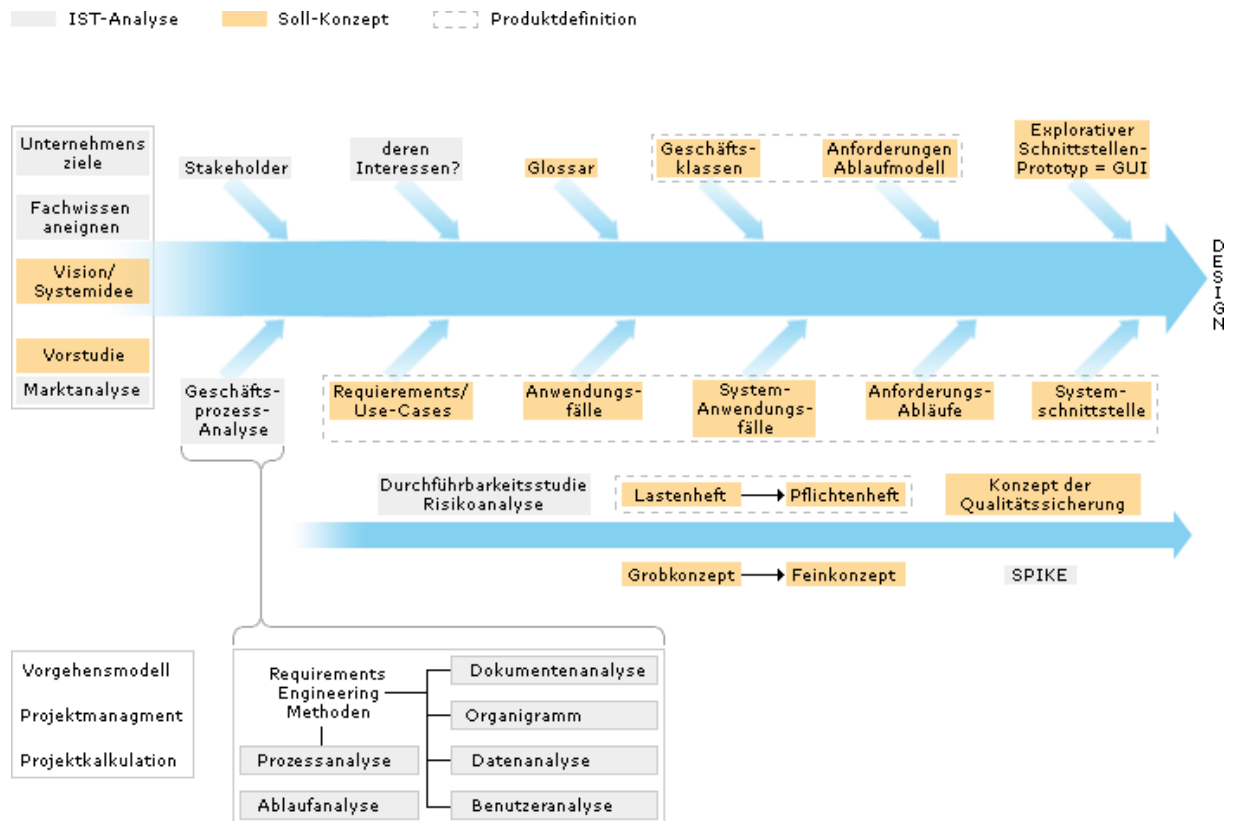


Abb.: Analyse-Phase

Anfangsphase

In der Anfangsphase finden sich wie bereits erwähnt:

- Beschreibung der Ausgangssituation
- Ressourcenplanung (personell, finanziell, zeitlich und technisch)
- erste Projektplanung
- Projektkalkulation mit einer Aufwandschätzung (hier gibt es viele Methoden wie Analogiemethode, Relationsmethode, Multiplikatormethode, Gewichtungsmethode, Methode der parametrischen Gleichungen, Prozentsatzmethode, etc., die leicht nachgeschlagen werden können.
- Die Gesamtorganisation des Projektes und das Vorgehensmodell

Unterscheidung in der Analyse

Man unterscheidet alle in der Analyse aufgeteilten Punkte in:

- die Problemanalyse / Geschäftsprozess-Analyse, die oft auch als **IST-Analyse** bezeichnet wird, betrachtet vorrangig die Gegenwart.
- das **SOLL-Konzept** der Analysephase betrachtet hingegen die Zukunft und gibt Aufschluss darüber, was entwickelt werden soll.

Alles was in den frühen Phasen oder Analysephasen das zukünftige Produkt beschreibt, wird unter dem Oberbegriff **Produktdefinition** zusammengefasst. Dies kann das Pflichtenheft (oder ein Teil daraus), Anforderungen, GUIs, Geschäftsklassen, etc. sein.

2 Unternehmensziele

Unternehmensziel

Die Unternehmensziele sind klar durch die Fragen definiert, was will das Unternehmen (mit dem Produkt) erreichen? Dabei ist zu beachten: dass es entweder ein externes oder firmeninternes Produkt sein kann, welches verkauft wird.

- Mehr Gewinn
- Marktanteile erhöhen
- Interner Datenfluss verbessern
- Besserer Kontakt zum Kunden
- Höherer Umsatz

Marktanalyse

Marktanalyse

- Wie sieht der Markt aus?
- Kann das Ziel erreicht werden?
- Wer sind die wichtigsten Konkurrenten?
- Wie kann man sich positionieren?
- Gibt es bereits Produkte auf dem Markt?

Oft verschwimmen oder verändern sich die Ziele mit der Zeit - sogar noch während der Entwicklung.

3 Fachwissen

Kommunikations- schwierigkeiten

Einige Projekte scheitern, weil die Auftragnehmer oder die Berater (Consultants) nicht die „gleiche Sprache“ sprechen wie der Auftraggeber bzw. das Auftrag gebende Unternehmen. Man versteht sich nicht und dies hat zur Folge, dass die Kommunikation unter- und miteinander ausbleibt, wodurch das Projekt letztendlich scheitern kann.

Was kann man dagegen tun?

- Literatur lesen
- Interviews führen
- versuchen die Sprache der Kunden oder deren Anspruchsgruppen (Stakeholder) zu verstehen und zu sprechen
- ein Praktikum machen
- daneben sitzen und zuschauen
- Requirements Engineering Methoden anwenden



4 Vision und Systemidee

Visionsskizze

Nichts ist wichtiger als alle Projektmitarbeiter zu **begeistern** und die Idee in einer kurzen Visionsskizze darzustellen. Dies sollte in **wenigen Sätzen** möglich sein.

Analog der bekannten Aussage von **Antoine de Saint-Exupéry**:

„Wenn Du ein Schiff bauen willst, so trommle nicht Männer zusammen, um Holz zu beschaffen, Werkzeuge vorzubereiten, Aufgaben zu vergeben und die Arbeit einzuteilen, sondern lehre die Männer die Sehnsucht nach dem weiten endlosen Meer.“

Dazu gehört auch ein prägnanter, intuitiver und kurzer **Projektname**. So wird beispielsweise aus der Adresse der Webseite „www.wohin-heute.net“ (die übrigens einer ihrer Online-Studentenkollegen erstellt hat) sofort ersichtlich, worum es geht. Oder nicht?



Formulieren

Übung ANA-01

Nomen est omen

Bevor Sie sich die Seite <http://www.wohin-heute.net> anschauen, versuchen Sie sich vorzustellen, wie Sie eine Seite mit diesem Namen bauen würden und welchen Inhalt oder welche Dienste Sie anbieten würden.

Schauen Sie sich erst danach die Seite an und vergleichen Sie ob Sie mir Ihrer Einschätzung richtig lagen.

Bearbeitungszeit: 20 Minuten

Viele Projekte verzichten auf einen kurzen prägnanten Namen. Dabei sind gute Projektnamen wichtige Selbstläufer, die sich schnell in den Köpfen vieler Personen einnisten.



Beispiel

Bosporus

Ein Projekt, welches zwei Gebiete wie Grafik und Robotik miteinander verbindet, könnte z. B. einfach **Bosporus** heißen. Dieses Projekt gab es wirklich und es war innerhalb von einem Monat in einer ganzen Hochschule bekannt.

5 Vorstudie und Marktanalyse

Vorstudie

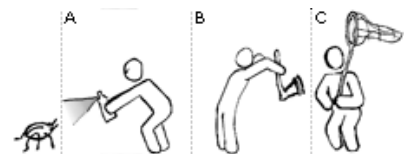
Die Vorstudie bildet den gesamten Analyseprozess schon mal vorher „im kleinen“ Rahmen ab!

- Generelle **Zielsetzung** festlegen. Zusammen mit dem Auftraggeber wird die generelle Zielsetzung und die zu beachtenden Randbedingungen festgelegt. Ergebnisdokumente: Gesprächsprotokolle u.ä.
- **Informationsquellen**. Es werden Unterlagen eingesehen und vorhandene Ideen zu dem Problem aufgenommen. Wie wird derzeit mit dem Problem umgegangen? Ergebnisdokumente: Verzeichnis der vorhandenen Informationsquellen und Ansprechpartner.
- **Ist-Zustand**. Durch Fragenkataloge und Interviews wird der Ist-Zustand in groben Zügen erhoben.

Anforderungen klären

In Gesprächen mit den Fachabteilungen werden die generellen funktionalen und nichtfunktionalen Anforderungen und Zielvorstellungen der Betroffenen und Beteiligten ermittelt.

- Problemfeld **abgrenzen**. Was gehört nicht mehr zu der Aufgabe? Wo existieren Schnittstellen zur Außenwelt?
Ergebnisdokument: Aufgabenbeschreibung, Kontextdiagramm.
- **Akteure** identifizieren. Wer macht was, d. h. welche Anwender haben welche Aufgaben und Kompetenzen im Kontext des Problembereiches.
Ergebnisdokumente: Akteurdiagramm, ggf. Kompetenzschema, Berechtigungskonzept
- **Umfeld** beschreiben. Wie sieht die Umgebung aus, in der das zu entwickelnde System einmal laufen soll?
Ergebnisdokument: Ein Infrastrukturmodell, das die Hardware-, Software-, Netzwerk- und Middlewaretechnologie sowie -architektur beschreibt.
- **Grobe Systemanwendungsfälle** identifizieren. Wie heißen die Systemanwendungsfälle die das System unterstützen sollen? Eine namentliche Auflistung der Fälle liefert bereits eine erste Orientierung über die Anforderungen. Ggf. wird jeder Systemanwendungsfall bereits mit einem Satz oder einigen Stichworten kurz erläutert.
Ergebnis: Systemanwendungsfallübersicht.
- **Prioritäten**. Es werden Schwerpunkte definiert und Prioritäten vergeben. Welche Probleme sind am Dringendsten zu lösen, welche haben etwas mehr Zeit?
Ergebnisdokument: Rahmenkonzept.
- **Lösungsalternativen**. Mögliche Lösungsansätze werden erarbeitet und gegenübergestellt: Welche Alternativen sind denkbar? Bei ganz dringenden Problemen stellt sich die Frage nach möglichen Sofortmaßnahmen.
Ergebnisdokument: Darstellung der Lösungsalternativen.
- **Empfehlung**. Die Lösungsalternativen werden (nach zuvor definierten Kriterien) bewertet, gegebenenfalls Wirtschaftlichkeitsbetrachtungen durchgeführt.
Ergebnisdokument: Empfehlung.
- **Weiteres Vorgehen**. Abschließend wird das weitere Vorgehen für die empfohlene bzw. vom Auftraggeber gewünschte Alternative geplant.
Ergebnisdokument: Empfehlungen zum weiteren Vorgehen.



Informationen zur Marktanalyse finden Sie unter:

 <http://www.unternehmerinfo.de>

6 Grob- und Feinkonzept

Konzeptphasen

In der Industrie spricht man oft von den beiden Konzeptphasen Grobkonzept und Feinkonzept.

Grobkonzept


Das Grobkonzept kommt insbesondere dann zum Einsatz, wenn kein agiles Modell angewendet wird. Beim **Grobkonzept** der Analysephase gibt es starke Überschneidungen mit dem, was sonst im Pflichtenheft oder in der Vorstudie vorgenommen werden würde. Auch hier sollen alle Anforderungen in einem Projektantrag formuliert und die Ziele festgehalten werden.

Feinkonzept

Das **Feinkonzept** ist die ausführliche und fertig dokumentierte Analyse. Meistens muss diese Beschreibung so ausführlich sein, dass sie zur Überprüfung der erbrachten Leistung herhalten kann. Sie muss daher auch realisierbar sein. Aus diesem Grund sind zumindest die folgenden Punkte vollständig enthalten:

- Benutzerschnittstelle
- Objektmodell
- Schnittstellensysteme

Der Informationsmanager **ANDREAS PFUND** beschreibt ein solches Vorgehen für die Erstellung von Webseiten auf seiner Homepage. Bitte schauen Sie sich diese Seiten gut an und stimmen Sie sich mit der Kursleitung ab, ob die Inhalte dieser Webseite ggf. prüfungsrelevant sind.

 www.andreas-pfund.de



7 Lastenheft und Pflichtenheft

Orientiert man sich bei der Problemanalyse bereits daran, die wesentlichen Anforderungen an das zu entwickelnde Softwareprodukt zu definieren, dann kann als Ergebnisform für die Untersuchungen auch ein Lastenheft verwendet werden. Das Lastenheft enthält eine Zusammenfassung aller fachlichen Basisanforderungen, die das zu entwickelnde Softwareprodukt aus der Sicht des Auftraggebers erfüllen muss. Das Pflichtenheft enthält dann genau alle zu realisierenden Anforderungen.

Reihenfolge

Das Lastenheft kommt vor dem Pflichtenheft. Als Gedächtnisstütze ist es hilfreich, sich an der alphabetischen Reihenfolge zu orientieren (L vor P). Oder man merkt sich, dass man erst die Last hat, zu formulieren was zu tun ist und danach die Pflicht dies auch umzusetzen.

Lastenheft

Das Lastenheft wird in der DIN 69901-1 bis 69901-5 spezifiziert und sollte vom Auftraggeber formuliert werden. In der Praxis wird es aber meist zusammen mit dem Auftragnehmer erstellt. Die Kosten für die Erstellung werden daher oftmals auf den Auftragnehmer abgewälzt.



Das Lastenheft bildet die Angebotsgrundlage für das Vorhaben und umfasst somit die erste Spezifikation des zu erstellenden Produkts (die „Last“). Ebenso enthält es bereits erste Anforderungen an das Produkt bei seiner späteren Verwendung (z. B. 1000 Hits pro Sekunde) sowie Rahmenbedingungen für Produkt und Leistungserbringung (Hardware, Infrastruktur, etc.).

Weiterhin sind dort die vertraglichen Konditionen wie z. B. das Erbringen von Teilleistungen, Gewährleistungsanforderungen, Risikomanagement usw. festgehalten. Die Anforderungen an den Auftragnehmer (z. B. Zertifizierungen) sowie an das Projektmanagement des Auftragnehmers (z. B. Projektdokumentation, Controlling-Methoden) sind ebenso im Lastenheft formuliert.

Die häufigste Einteilung ist:

1. **Zielbestimmung:** Hier wird beschrieben, welche Ziele durch den Einsatz des Produktes erreicht werden sollen.
2. **Produkteinsatz:** Es wird festgelegt, für welche Anwendungsbereiche und für welche Zielgruppen das Produkt vorgesehen ist.
3. **Produktfunktionen:** Die Hauptfunktionen des Produktes werden aus Auftragsgebersicht beschrieben. Es ist darauf zu achten, dass die Kernfunktionen und nicht sekundäre Funktionen beschrieben werden. Auf Detailbeschreibungen ist zu verzichten. Jede Funktionsanforderung ist durch eine vorangesetzte Zahl mit vorangesetztem LF (Lastenheft-Funktion), eingeschlossen in Schrägstrichen, zu markieren (z. B. /LF 10/), um eindeutig referenzieren zu können.
Hinweis: Die Anforderungen werden in 10er-Schritten durchnummeriert, um später zusätzliche Anforderungen einfügen zu können.
4. **Produktdaten:** Die Hauptdaten des Produktes, die permanent gespeichert werden müssen, werden festgelegt (/LDnn/).
5. **Produktleistungen:** Liegen Leistungsanforderungen bzgl. Zeit, Datenumfang oder Genauigkeit an einzelnen Hauptfunktionen und -daten vor, dann werden sie hier aufgeführt und mit /LLnn/ markiert.
6. **Qualitätsanforderungen:** Die wichtigsten Qualitätsanforderungen sollten hier aufgeführt werden, wie gute Zuverlässigkeit, gute Benutzbarkeit, normale Effizienz usw.
7. **Ergänzungen:** Hier werden Ergänzungen oder spezielle Anforderungen beschrieben, z. B. außergewöhnliche Anforderungen an die Benutzerschnittstelle.



Definition

Pflichtenheft

Das Pflichtenheft ist eine **vertraglich** bindende und genaue Beschreibung des zu entwickelnden IT-Systems.

Inhalt

Die Anforderungen an das Pflichtenheft sind - wie die des Lastenhefts auch - in der DIN 69901-1 bis 69901-5 festgelegt. Die Norm enthält Detailbeschreibungen aller Lösungsansätze sowie in der Regel das Lastenheft.



Beispiel

Gliederung eines Pflichtenhefts nach Balzert

 [Bal98]

1. Zielbestimmung
 - Grenzkriterien
 - Wunschkriterien
 - Abgrenzungskriterien
2. Produkteinsatz
 - Anwendungsbereiche
 - Zielgruppen
3. Produktkonfiguration der Software, Hardware, Orgware
4. Produktumgebung: Schnittstellen, Betriebsbedingungen
5. Produktfunktionen
 - Funktion 1
 - Funktion 2 usw.
6. Produktleistungen
7. Benutzerschnittstelle
8. Qualitäts-Zielbestimmung
9. Globale Testfälle
 - Testfall 1
 - Testfall 2 usw.
10. Entwicklungskonfiguration der Software, Hardware, Orgware
11. Ergänzungen

8 Durchführbarkeitsstudie und Risikoanalyse

Fragen zu
Projektbeginn

Zu Beginn eines Projektes sollte man mindestens die folgenden Fragen beantworten:

- Ist das Budget ausreichend?
- Ist das System technisch realisierbar?
- Sind die Randbedingungen erfüllbar?
- Wird das Projekt ein wirtschaftlicher Erfolg?
- Können wir die Erfolgskriterien bestimmen und festlegen?
- Kann man messbare Nachweiskriterien für deren Eintritt ermitteln?
- Kann man die Projektrisiken ermitteln und bewerten?
- Worin bestehen die Projektrisiken?
- Können wir die Eintrittswahrscheinlichkeit und die Schadenshöhe schätzen?

Risikoanalyse

Als Antwort zur letzten Frage sollten Maßnahmen zur Reduzierung der Eintrittswahrscheinlichkeit und zur Schadensminimierung festgelegt werden. Zudem sollte man das Risiko nach Eintrittswahrscheinlichkeit und Schadenshöhe bewerten sowie alle Projektrisiken in einer Liste sammeln.

9 Konzept der Qualitätssicherung

Anwendung der
Qualitätsverfahren

Qualität ist ein sehr umfassender Bereich. Als Projektverantwortlicher sollten Sie alle Ebenen kennen, auf die Qualitätsverfahren angewendet werden müssen.

- Es müssen **Produktreviews** / Codereviews festgelegt werden. Beispielsweise: Bei Google muss jedes Codestück 5-6 Reviews passieren, bevor es in Produktion geht.
- **Entwicklungsrichtlinien** festlegen und prüfen. Beispielsweise Style-Guides, Tools, Verfahren, Wikis, etc.
- **Testkonzepte**: Auf welchen Testebenen (siehe Lerneinheit TST - Testen) muss getestet werden.
- **Testpläne**: Werden Tests ausreichend dokumentiert?
- **Arbeitsaufträge** checken. Wie werden Arbeiten notiert, verteilt und geprüft? Vieles gibt hier das Vorgehensmodell und der Prozessleitfaden vor.
- **Meilensteine** überprüfen. Wer überwacht die Meilensteine? Gibt es Kundenabnahmen für Meilensteine oder „User Stories“ wie in Scrum?

Qualitätsmanagement

Die genannten Verfahren sind Teil eines kompletten Qualitätsmanagements und werden in Großprojekten von einzelnen Personen oder auch von einer Gruppe von Personen (Qualitätsbeauftragten) durchgeführt und gesteuert.

Üblicherweise versucht man auch die Qualität automatisiert zu messen und **von vornherein** fest **in den Buildzyklus** zu **integrieren**. Dazu können die Tools und die folgenden Ebenen gehören:

- Stylechecker und Bugfinder
- Unit-Tests
- Architekturmetriken (Abhängigkeitsgraphen, Zyklen, etc.)



Die Erfahrung zeigt, dass Projekte die dies initial tun, viel besser laufen. Die Codequalität ist besser und das später groß wachsende System wird besser beherrscht.

10 Technischer Prototyp / Durchstich (Spike)

Technischer Durchstich

Es ist sinnvoll, bereits in der frühen Phase einen technischen Prototypen zu erstellen.

Dieser wird auch als **technischer Durchstich** bezeichnet - im englischen „*Spike*“ genannt. Weiterhin kann er evtl. bei geringem Budget oder bei Zeitmangel auch mit einem allgemeinen Prototypen verbunden werden.

Meistens ist dieser Prototyp jedoch dazu da, zu zeigen, ob bestimmte Technologien funktionieren. Auch für nichtfunktionale Anforderungen ist ein technischer Prototyp meistens gut geeignet: Beispielsweise kann gezeigt werden, dass ein SOAP oder REST Client bei 1000 Anfragen pro Sekunde nicht performant ist. Oder ein RMI Client eben doch.

Sinn des Prototyps

Sinn dieses Prototyps ist es also einerseits schnell zu beweisen oder zu widerlegen, dass die **Technologie beherrscht** wird, angemessen und reif genug ist. Zum anderen besteht der Nutzen darin, ein schnelles **Feedback** zu erhalten. Sei es in Form von Erfahrung von der Technik oder in Form von Feedback vom Auftraggeber.

Anzumerken ist hier, dass in einigen Vorgehensmodellen der *Spike* die erste Version der Software ist. Das heißt, man versucht sofort die einfachste Funktionalität abzubilden und das in der Umgebung, wie sie auch zum Ende vorliegen wird. Typisch sind hier einfache Prototypen z. B. in Ruby on Rails, die einfache Grundgerüste (Scaffolds) zur Verfügung stellen.

11 Geschäftsprozessanalyse und Systemanalyse

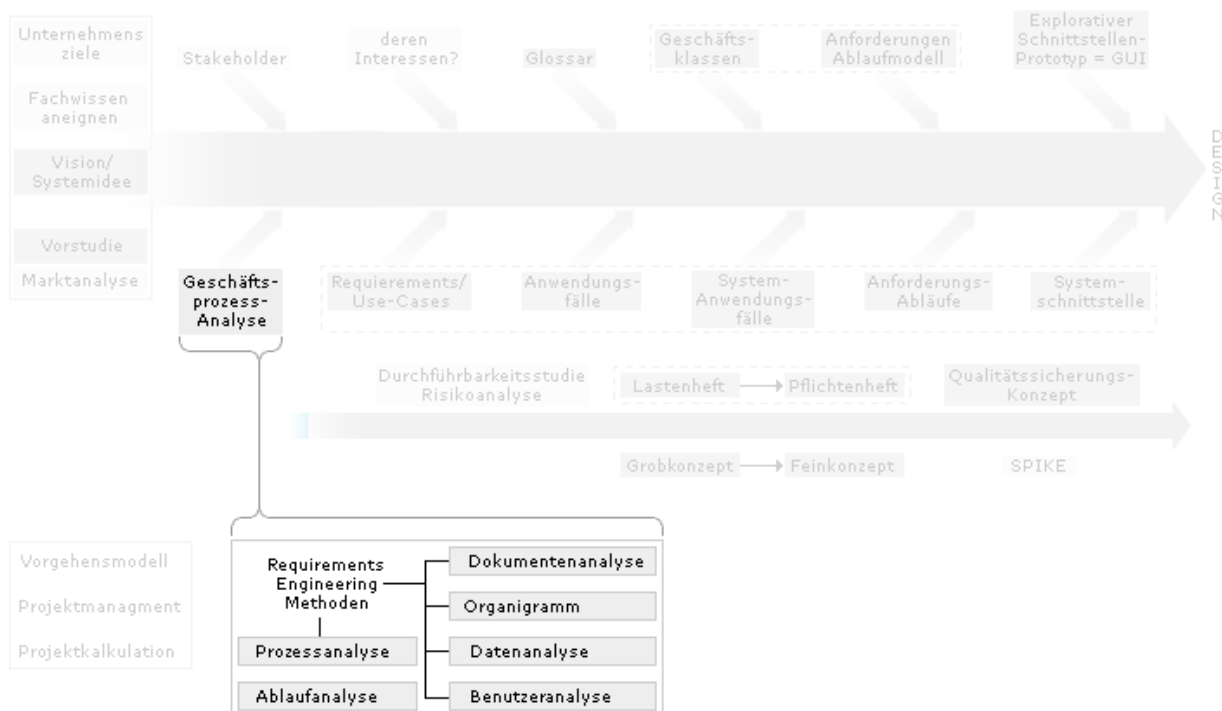


Abb.: Geschäftsprozessanalyse

Die Geschäftsprozessanalyse umfasst in der Regel grob:

- Requirements-Engineering Methoden
- Dokumente studieren
- Organigramme bekommen oder aufstellen
- Daten-Analyse
- Prozess-, Ablauf-, -Fluss-Analyse
- Benutzer-Analyse

Detaillierte Analyse

In vielen Fällen muss ein bestehendes System / Unternehmen oder eine Produktionsstruktur analysiert werden. Daher können die folgenden Teile detaillierter untersucht und dokumentiert werden:

1. Organisationsstruktur
 - Aufbauorganisation
 - Beschreibung der Organisationseinheit / Geschäftsbereiche
2. Ergebnisse der Geschäftsprozessenerhebung
 - Logistik
 - Beschaffung / Einkauf
 - Fertigung
 - Lager
 - Qualitätssicherung
3. Ergebnisse der Dokumentenerhebung
4. Analyse des vorhandenen Anwendungssystems (PPS-System)
 - Menüstruktur
 - Anwendungsfunktionalität
 - Datenstrukturen
 - Engpässe des verfügbaren Anwendungssystems
5. Vorschlag zum Aufbau des künftigen Anwendungssystems (Grobübersicht)

Web 2.0 Unternehmen

Nicht immer geht es um die Fertigungsindustrie, sondern vermehrt auch um Unternehmen im Web 2.0 Umfeld. In diesem Fall gibt es keine Fertigung und das jeweilige Produkt oder die Dienstleistung muss entsprechend untersucht werden.

12 Vorgehens- und Prozessmodell

Vorgehensmodell

Viele Dinge wie Projektmanagement oder die Festlegung des Vorgehensmodells fallen nicht direkt in die Analysephase. Dennoch ist dieser Punkt hier aufgenommen, um die Gesamtheit aller Tätigkeiten in den frühen Phasen zu erfassen. (Wie beispielsweise auch Projektmanagement oder Projektkalkulation, die bereits an anderer Stelle in der Analyse oder in der entsprechenden Lerneinheit erwähnt wurden)

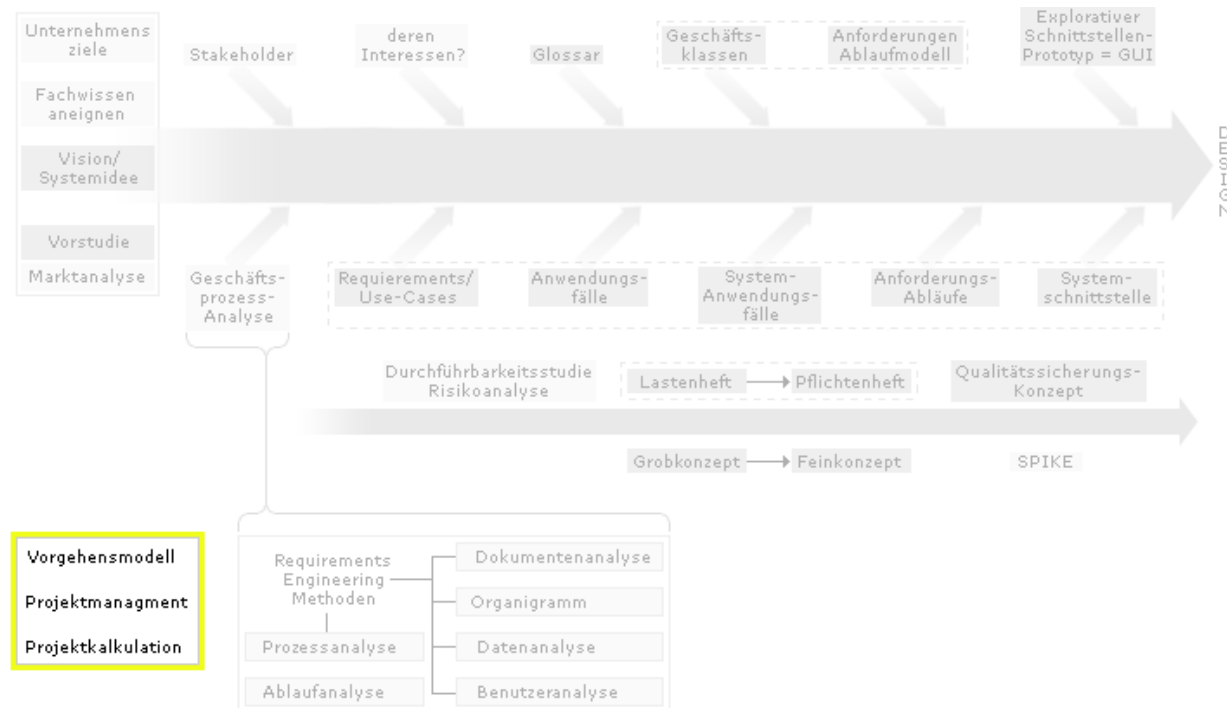


Abb.: Vorgehensmodell in der Analysephase

Entscheidungen

Es liegt in der Regel am Projektmanager initial Entscheidungen zu treffen: Ein eigenes Vorgehensmodell zu verwenden oder ein bestehendes Vorgehensmodell geeignet anzupassen.

Bekannte Vorgehensmodelle sind in diesem Zusammenhang:

- Wasserfallmodell
- Spiralmodell
- V-Modell
- **RUP / Rational Unified Process**
- **XP**
- **Scrum**
- **Crystal Clear**

Die vier zuletzt genannten sind die aktuellsten und damit die heutzutage relevanten Modelle. Die folgenden Dinge sollten Ihre Wahl beeinflussen:

- Größe und Art des Projektes
- Kultur und Bereitschaft des Unternehmens
- Wissen und Bereitschaft aller Beteiligten

Fragen Sie einmal in Unternehmen nach, mit welchen Modellen dort gute Erfahrungen gemacht wurden!

13 Projektmanagement und Projektkalkulation

Die Themen Projektmanagement und Projektkalkulation werden im entsprechenden Online-Kurs behandelt. Aufgrund ihrer Wichtigkeit hier jedoch ein paar kurze Punkte.



Definition

Projektmanagement

Nach DIN-Norm 69901 versteht man unter Projektmanagement:

„Projektmanagement ist die Gesamtheit von Führungsaufgaben, -organisation, -techniken und -mitteln für die Abwicklung eines Projektes“.

Projektmanagement

Als Projektmanager/in müssen Sie:

- Die Projektvorgaben und Ziele kommunizieren und überwachen
- Die Struktur des Projektes festlegen
- Die Projektphasen begleiten und festlegen
- Ein Risikomanagement betreiben
- Verantwortung delegieren und Teilergebnisse prüfen
- Termine und Kosten managen (s. u.)
- Das Qualitätsmanagement durchführen oder aufsetzen
- Personal verwalten (Krankheiten, Kündigungen?!)
- Kommunikationsmanagement durchführen (einer der zeitaufwendigsten und wichtigsten Punkte im Projektmanagement!)

Den Anspruchsgruppen (Stakeholdern) gegenüber müssen Sie das magische Dreieck im Auge behalten.

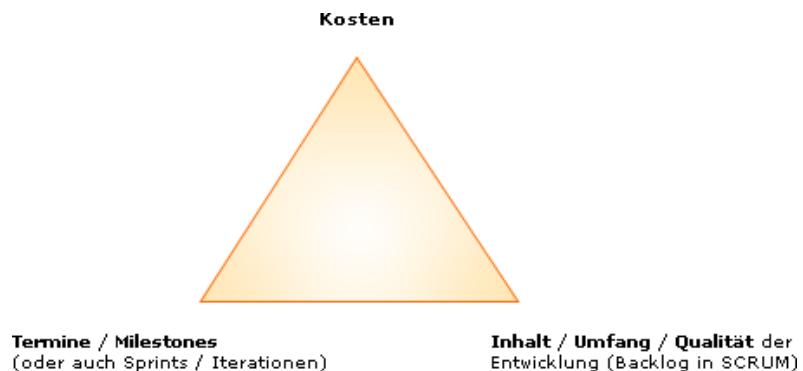


Abb.: Magisches Dreieck

Es gilt also ständig zu entscheiden, an welchem dieser Stellschrauben man drehen kann.



Definition

Projektkalkulation

Die Projektkalkulation ist nach DIN 69905 die „*Ermittlung der voraussichtlichen kostenwirksamen Projektleistungen und ihre Bewertung*“.

Projektleitung

Als Projektleiter/in müssen Sie:

- initial den Projektaufwand abschätzen
- im Projekt den Ist-Aufwand berechnen
- die Abweichung Soll-Ist bestimmen und daraus Maßnahmen ableiten

Entscheidend für die Projektkalkulation sind die folgenden Verfahren:

- **Expertenschätzung:** Die beliebteste Methode. Experten haben vielleicht viele ähnliche Projekte durchgeführt und große Erfahrung.
- **Delphi-Methode:** Ist die Gewichtung von Expertenschätzungen
- **Function-Point Verfahren:** Ein Function-Point ist entweder eine Eingabe, eine Ausgabe, eine Anfrage, eine Schnittstelle oder ein Datenbestand. All dies wird ermittelt, gewichtet und in Geld oder Mannmonate umgerechnet.
- **COCOMO (I+II):** steht für Constructive Cost Model und berechnet ein KDSI. Dies steht für *Kilo Lines of Devlivered Source Instructions* also für 1000 Zeilen Instruktionen im Sourcecode. Es sind hier - ähnlich der McCabe Cyclomatic Complexity - Dinge wie Schleifen, Verzweigungen oder Zuweisungen gemeint. In COCOMO II werden dann weitergehende Konzepte berücksichtigt wie:
 - Wiederverwendung von Code
 - Komposition von Programmen
 - Automatisch generierter Programmcode
 - Objektorientierte Ansätze

14 Stakeholder und deren Interessen

Anteilseigner

Stakeholder sind **Anteilseigner** für die etwas auf dem Spiel steht. Der Begriff kommt aus dem Englischen. *Stake* ist der Einsatz und Stakeholder sind **Anspruchsberechtigte** oder **Interessenvertreter**.



Wichtig zu Beginn eines Projektes ist es daher immer, all diese Personen ausfindig zu machen, dies zu dokumentieren und sie richtig einzuordnen. Problematisch ist dabei oft in der Praxis, dass man an wichtige Chefs (die das Projekt mal eben umlenken können) oft nicht herankommt.

Wichtige Fragen

Wichtige Fragen sind also:

- Wer kann alles zu den Requirements beitragen?
- Wer sind die Ansprechpartner?
- Wer sind die Endkunden?
- Wer sind die Fachabteilungen?
- Welche Interessen haben die Stakeholder?
- Gibt es politische Interessen?
- Gibt es egoistische Interessen?
- Welche Sichtweisen haben die Fachabteilungen?
- Welche Probleme oder Wünsche haben die Stakeholder?

15 Glossar

Ein global erreichbares Glossar dient sowohl dazu, die Begriffe aller Dokumente zu erklären als auch dazu eine generelle Fachsprache bekannt zu machen und sie zwischen den Beteiligten zu etablieren.

Leider verwenden Consultants, Projektmanager und Firmenmitarbeiter in der Praxis oft ein unterschiedliches Vokabular. Banker sprechen beispielsweise von „Roller Coster Swaps“, Biologen von „Hydrogenase“, etc.

Anforderung

Ein Glossar sollte:

- Leicht **erreichbar** sein
- **Editierbar** sein (z. B. als Wiki)
- initial etwas gefüllt sein (erhöht die Akzeptanz)



Beispiel

Beispiel für Glossareinträge

Hier ein Beispiel für Glossareinträge, wieder mal für ein Schachprogramm:

1. *pondering mode* - Analysemodus - (Beschreibungen, Auftreten, Link, ...)
2. *UCI* - Universal Chess Interface - (Beschreibungen, Auftreten, Link, ...)
3. *transposition table* - Tabelle für Zugumstellungen - (Beschreibungen, Auftreten, Link, ...)

Glossareinträge

In diesem Beispiel sind zufällig alle Glossareinträge englische Wörter; was natürlich nicht sein muss aber oft vorkommt.

Hilfreich ist es auch, wenn weitere Referenzen, Beschreibungen oder Weblinks zu einem Glossar angegeben werden.

16 Requirements und Use Cases

Requirements und
Use Cases

In einer frühen Phase der Analyse muss auch festgehalten werden:

- Was sind die Requirements = Anforderungen?
- Welche Use-Cases gibt es?

Auf die Details beider Punkte und deren Überschneidungen / Unterschiede wurde bereits in der Lerneinheit REQ - Requirements-Engineering eingegangen.

Use Cases

Die Use Cases überdecken meistens die **funktionalen Anforderungen**. Sie beantworten die Frage: Was tut das System?

Sonstige
Requirements

Die sonstigen Requirements decken oft auch zusätzlich die **nicht funktionalen Anforderungen** ab, wie beispielsweise:

- Benutzbarkeit / Usability-Requirements
- Performance-Requirements (Effizienz)
- Reliability-Requirements und
- Supportability-Requirements wie: Änderbarkeit, Wartbarkeit, Administrierbarkeit

17 Anwendungsfälle und Abläufe

Erstellung einfacher
Geschäftsprozesse

Nach der Vorbereitung kann nun mit der Erstellung der ersten **einfachen Geschäftsprozesse** begonnen werden. Diese können der Einfachheit halber, zu Beginn auch als Anwendungsfälle notiert werden. Das heißt, alle Akteure werden berücksichtigt. Zu jedem Geschäftsprozess können dann etwas genauere Aktivitätsdiagramme erstellt werden.

Wohlgemerkt, wir sind hier nicht in der UML-Design Phase! Dennoch lassen sich bereits in der Analyse auch produktive UML-Werkzeuge und passende Diagramme einsetzen.

Die erfassten wichtigsten Geschäftsanwendungsfälle werden oft als Business Use-Cases bezeichnet. Dabei beschreibt man:

- Diese in Kartenform / Textform / Katalogform
- Ob diese „stabile“ sind - sich wahrscheinlich nicht mehr ändern werden
- Die Auslöser des konkreten Anwendungsfalls
- Die Vorbedingungen zu einem Anwendungsfall
- Die Randbedingungen zu einem Anwendungsfall
- Unklare Bereiche, die noch nicht erfasst werden können.

Systemanwendungsfälle
identifizieren

Angepasst auf das konkret zu erstellende System wird jetzt detaillierter ausgearbeitet:

- Entscheidung für jeden vorliegenden Geschäftsanwendungsfall, ob er systemtechnisch umgesetzt werden soll.
- Ergänzung von ggf. fehlenden fachliche Ausnahmen und Varianten, beispielsweise fachlich sinnvolle Abbruchmöglichkeiten.
- Zerlegung der systemtechnisch umzusetzenden Geschäftsanwendungsfälle in zeitlich kohärente Systemanwendungsfälle.
- Erstellen eines Systemanwendungsfallmodells.

Beispielsweise kann das Ausleihen eines Videos auf verschiedene Arten geschehen. Hier müssen also verschiedene (IT-)Systemanwendungsabläufe gestaltet werden.

18 Geschäftsklassen

Geschäftsklassen

Bereits in der Analysephase sollte man wissen, welche wesentlichen Geschäftsklassen es gibt. Diese werden meist auch **Businessklassen** (oder Domainklassen) genannt. Hier kann man bereits Assoziationen und Multiplizitäten erarbeiten und erste Zusammenhänge oder Probleme erkennen.

Natürlich gehört ein ausführliches Klassendiagramm aller Komponenten in die Designphase. Dann aber auch meist mit allen Feldern und Methoden. In der Analysephase ist jedoch auch meist schon ein grober Überblick (ohne Felder und Methoden) sehr sinnvoll.

Basisklassen

Üblicherweise sind dies Basisklassen wie: **Person**, **Adresse**, **Konto**, **Produkt**, **Kunde**, etc.

Im Beispiel der Schachprogrammierung sind die wichtigsten Basisklassen:

- **UciHandler** - Kommuniziert mit dem Anwender
- **MoveGenerator** - Erzeugt die Züge
- **OpeningManager** - Managt die Eröffnungsbücher
- **GameInfo** - Information über die aktuelle Partie
- **Board** - Die Brettrepräsentation
- **Engine** - die Logik, die Spielen kann

Echte Businessklassen sind in diesem Beispiel nur **GameInfo** und **Board**, um alle anderen Klassen kümmert man sich üblicherweise erst nach der Analysephase.

19 Systemschnittstellen

Systemschnittstellen identifizieren

Zu jedem Anwendungsfall müssen häufig Systemschnittstellen identifiziert werden.

In kleinen Projekten sind diese meist nicht sofort erkennbar. In größeren Projekten wird jedoch schnell klar, dass ein Projekt technisch gesehen nicht isoliert zu betrachten ist. Das System interagiert mit anderen IT-Systemen oder mit der realen Welt.

Stellen Sie sich das Beispiel „Toll-Collect“ vor. Es gibt dort Satelliten, Autobahnbrücken mit Sendern, Terminals an Raststätten, den normalen Postweg und noch sehr vieles mehr. All diese Schnittstellen müssen berücksichtigt werden. Oft gibt es mit derartigen Schnittstellen die meisten Probleme.



Systemschnittstellen

Beispiele für Systemschnittstellen sind:

- **Dialogschnittstellen** (gute / schlechte GUIs?!)
- **Ausgabeerzeugnisse** (Briefe, Reports, etc.)
- **Datenschnittstellen** (Altsysteme, Web-Services, REST, etc.)
- oder rein funktionale Schnittstellen zu externen Systemen

20 Explorativer Schnittstellenprototyp / GUI

Grobentwurf

Es sollte schon in der Analysephase ein erster **grober Entwurf** der grafischen Benutzeroberflächen, der GUIs (Graphical User Interface) vorliegen, der am besten mit Kollegen oder den Stakeholdern schon einmal am Whiteboard skizziert wird.

Ein detailliertes GUI ist dann allerdings erst in der Designphase zu erstellen.

Feedback

Besonders bei agilen Modellen ist es wichtig, dass - ggf. auch schon durch den **Prototypen** - die Ergebnisse der **Benutzeranalyse** früh mit einfließen.

Ein frühzeitiges Feedback hilft Designfehler zu vermeiden, die später teuer werden können.

An dieser Stelle wäre die Analyse beendet. In größeren Projekten sollten auf jeden Fall alle die in dieser Lerneinheit erwähnten Schritte durchgearbeitet werden. Was nun folgt ist die Designphase. Dort geht es ebenfalls um UML, aber insbesondere deren Anwendung und um Modellierungs- und Architekturwissen.

Zusammenfassung

- Die in der Analyse aufgeführten Stufen und Aufgaben werden in Ist-Analyse und Soll-Konzept unterteilt.
 - Was in den frühen Phasen oder Analysephasen das zukünftige Produkt beschreibt, wird unter dem Oberbegriff **Produktdefinition** zusammengefasst.
 - Neben der Betrachtung der Unternehmensziele ist eine Marktanalyse wichtig.
 - Die mangelnde Kommunikation zwischen den Beteiligten kann zum Scheitern des Projektes führen.
 - Visionsskizzen dienen der Motivation und sollen begeistern.
 - Vorstudien bilden den Analyseprozess im geringeren Umfang ab.
 - Das Feinkonzept sollte zur Überprüfung der erbrachten Leistung taugen.
 - Das Pflichtenheft ist wie das Lastenheft in der DIN 69905 festgelegt.
 - Die Machbarkeit wird mittels Durchführbarkeitsstudie und Risikoanalyse überprüft.
 - Die Qualitätssicherung umfasst mehrere Ebenen und die verwendeten Verfahren sind Teil des Qualitätsmanagements.
 - Prototypen zeigen, dass die Technologie beherrscht wird, und diese angemessen und reif genug ist.
 - Die Geschäftsprozessanalyse umfasst zahlreiche weitere Analysen deren Ergebnisse Einfluss auf das Projekt haben können.
 - Ein Glossar dient dazu eine generelle Fachsprache bekannt zu machen und zwischen den Beteiligten zu etablieren.
 - Use-Cases decken funktionale Anforderungen ab, die Sonstigen decken auch die funktionalen Requirements ab.
 - Die wichtigsten Geschäftsanwendungsfälle werden als Business Use-Cases bezeichnet.
 - Die wesentlichen Geschäftsklassen werden meist auch **Businessklassen** (oder Domainklassen) genannt.
 - Designfehler können durch Benutzeranalysen anhand von Prototypen und GUI-Entwürfen vermieden werden.
-

Wissensüberprüfung



Freitext

Übung ANA-02

Analysephase

In welche zwei Bereiche wird die Analysephase grob eingeteilt?

-Analyse

-Konzept

? Test wiederholen Test auswerten



Freitext

Übung ANA-03

Technischer Prototyp

Wie lautet die englische Bezeichnung für einen „technischen Durchstich“?

? Test wiederholen Test auswerten

Einsendeaufgabe

Fragen Sie bei Ihrer Kursbetreuung nach, wie die Dateien eingereicht werden sollen. Möglich ist ein Hochladen im Lernraumsystem oder die Zusendung per E-Mail.



Einsendeaufgabe

Einsendeaufgabe ANA-E1

Konzeption und Durchführung einer Analyse

1. Erstellen Sie aus den in der Lerneinheit vorgestellten Punkten Ihren eigenen Fahrplan für die Analyse. Begründen Sie Ihre Entscheidungen
2. Führen Sie eine komplette Analyse für Ihr Softwaretechnik-Gesamtprojekt durch.

Bearbeitungszeit: 120 Minuten

Umfang ca. 4 A4 Seiten