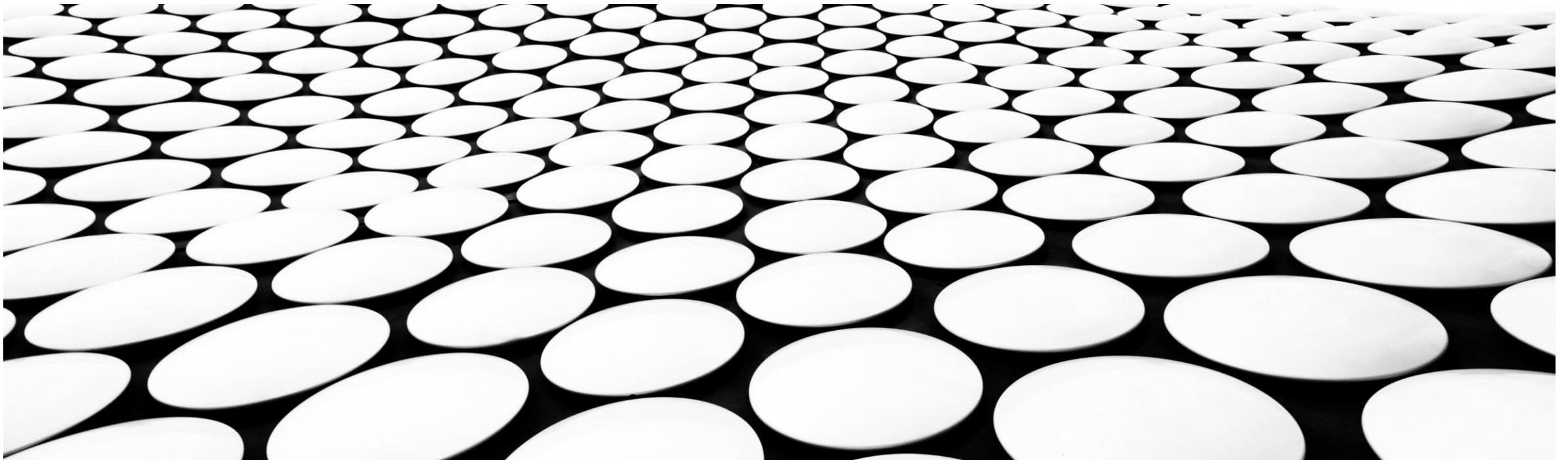


---

# SOFTWARE ENGINEERING

WAS IST SOFTWARE ENGINEERING?





# WAS IST SOFTWARE ENGINEERING

- Software Engineering (deutsch: Softwaretechnik) ist die praktische Anwendung wissenschaftlicher Erkenntnisse, um Software wirtschaftlich zu entwickeln, einzusetzen und zu warten.



# ZIELE VON SOFTWARE ENGINEERING

- Erstellung von Qualitätssoftware
- Fehler vermeiden
- Aufwände minimieren
- Kundennutzen maximieren

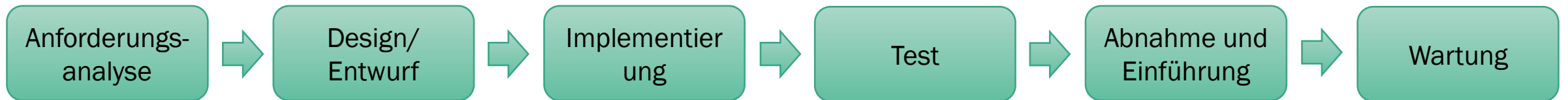


# **SOFTWARE LEBENSZYKLUS**



# WAS IST DER SOFTWARE LEBENSZYKLUS?

- Jedes Softwaresystem durchläuft einen Software Lebenszyklus
  - Muss bei Softwareentwicklung beachtet werden.





# **REQUIREMENTS ENGINEERING**



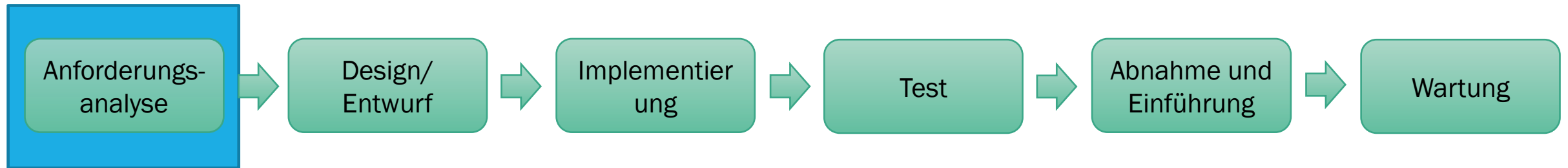


# ANALYSEPHASE

- In der Analysephase werden:
  - Die Anforderungen ermittelt.
  - Die Anforderungen im Lastenheft festgehalten.

# REQUIREMENTS ENGINEERING

- Findet in der Anforderungsanalyse statt





# REQUIREMENTS ENGINEERING

- Requirements Engineering ist ein Prozess, in dem Kundenanforderungen für ein zu erstellendes System ermittelt, dokumentiert, überprüft und verwaltet werden
- **Ziel:** Anforderungen ermitteln, darüber nachdenken und verstehen
- **Prozess:**
  - Anforderungen ermitteln
  - Anforderungen dokumentieren
  - Anforderungen überprüfen und abstimmen
  - Anforderungen verwalten

# ANFORDERUNGSARTEN

- Funktionale Anforderungen (Beschreiben, was ein System tun soll)
  - z.B. „Die Software soll basierend auf einer Formel die Kreditwürdigkeit berechnen“
  - z.B. „Die Software soll einen Mechanismus zur Identifizierung der Nutzer vorsehen“
- Nichtfunktionale Anforderungen(Einschränkende Bedingungen)
  - z.B. „Die Software soll dem Anwender innerhalb von 1 Sekunde antworten“



# HÄUFIGE PROBLEME

- Stakeholder können Anforderungen oft nicht korrekt ausdrücken
- Fachsprache
- Widersprüchliche Sichten auf den Ist-Zustand
- Unklare Zielvorstellung der zu erstellenden Software
- Implizites und unbewusstes Wissen



# ANFORDERUNGSANALYSE

DIE ANFORDERUNGSANALYSE IST IN DER INFORMATIK EIN TEIL DES SYSTEMENTWICKLUNGSPROZESSES





# **VOLERE**

VORGEHENSMODELL ENTWICKELT VON DEN ROBERTSONS



# NACH VOLERE

In dem von den Robertsons entwickelten Vorgehensmodell Volere existieren

- Anforderungsspezifikation,
- Stakeholder-Analyse,
- Bedarfsanalyse,
- Analyse der Priorisierung und
- Aufzeichnung der elementaren Anforderungen.
- Die Volere-Templates konnten bis zum Jahr 2008 kostenlos heruntergeladen werden.  
Seit 2009 ist es kostenpflichtig.  
Sie dürfen gegen Entrichtung einer Spende und unter Angabe der Quelle für Anforderungsanalyse in eigenen Projekten eingesetzt werden.  
<https://www.volere.org/>
- Neben einigen weiteren Beteiligten stehen vor allem Suzanne und James Robertson als treibende Kräfte hinter Volere.  
Der Name Volere wurde in Anlehnung an das italienische Verb volere – wollen, wünschen – gewählt.



# IIBA

INTERNATIONAL INSTITUTE OF BUSINESS ANALYSIS



# NACH IIBA

Das International Institute of Business Analysis führt zu diesem Thema im Business Analysis Body of Knowledge drei Kapitel auf:

- Anforderungserhebung:
  - Anforderungen der Stakeholder ermitteln,
- Anforderungs-Management & Kommunikation:
  - Anforderungen verwalten und kommunizieren
  - wiederverwendbare Anforderungen identifizieren
  - Anforderungen zusammenstellen
  - Anforderungen zur Genehmigung vorbereiten
  - Anforderungsänderungen managen,
- Anforderungsanalyse:
  - Anforderungen priorisieren, strukturieren,
  - Anforderungen in Textform dokumentieren,
  - Anforderungen mit Grafiken/Modellen dokumentieren,
  - auf inhaltliche Qualität prüfen,
  - auf Übereinstimmung mit den Zielen prüfen.





# IEEE

I E E E ODER EI TRIPLE I



# **NACH IEEE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS**

Laut IEEE kann das requirements engineering unterteilt werden in:

- Anforderungserhebung (requirements elicitation),
- Anforderungsanalyse (requirements analysis),
- Anforderungsspezifikation (requirements specification) und
- Anforderungsbewertung (requirements validation)

Diese Tätigkeiten überlappen einander und werden oft auch mehrfach – iterativ – durchgeführt.



**CMMI**





# NACH CMMI

Das Software Engineering Institute (SEI) der Carnegie Mellon Universität unterscheidet in ihrem Capability Maturity Model Integration

- die Entwicklung der Anforderungen und
- das Management von Anforderungen.



**IREB**



# NACH IREB

Das International Requirements Engineering Board listet zu diesem Thema im Lehrbuch für die Zertifizierung zum Certified Professional for Requirements Engineering vier Kapitel auf:

- Ermitteln: Beim Ermitteln der Anforderungen werden verschiedene Techniken genutzt, um die Anforderungen der Stakeholder und anderer Quellen zu gewinnen, zu detaillieren und zu verfeinern.
- Dokumentieren: Durch die Dokumentation werden erarbeitete Anforderungen adäquat beschrieben.
- Prüfen und abstimmen: Dokumentierte Anforderungen müssen frühzeitig geprüft und abgestimmt werden, um zu gewährleisten, dass sie allen geforderten Qualitätskriterien genügen.
- Verwalten: Die Anforderungsverwaltung geschieht flankierend zu allen anderen Aktivitäten und umfasst alle Maßnahmen, die notwendig sind, um Anforderungen zu strukturieren, für unterschiedliche Rollen aufzubereiten sowie konsistent zu ändern und umzusetzen.



# VORGEHEN



# VORGEHEN

- In allen genannten Modellen existieren die folgenden Schritte in der einen oder anderen Form. Dabei werden Anforderungen gesammelt (englisch elicitation); durch Analyse soll ein gemeinsames Verständnis hergestellt werden; die Anforderungen werden textlich oder in Modellen dokumentiert, d. h. spezifiziert. Danach wird üblicherweise geprüft, ob das Ganze noch stimmig ist (englisch validation). Rund um diese Schritte existiert Verwaltung und Management des Prozesses.



# ERMITTLUNG, ANALYSE

Beim Sammeln der Anforderungen (engl. elicitation) ist der Übersetzungsprozess zwischen Fachseite und Entwickler von besonderer Bedeutung. Folgende Kriterien sind zu erfüllen:

- **Vollständig**  
Alle Anforderungen des Kunden müssen explizit beschrieben sein, es darf keine impliziten Annahmen des Kunden über das zu entwickelnde System geben.
- **eindeutig definiert / abgegrenzt**  
Präzise Definitionen helfen, Missverständnisse zwischen Entwickler und Auftraggeber zu vermeiden.
- **verständlich beschrieben**  
Damit sowohl der Auftraggeber als auch der Entwickler mit vertretbarem Aufwand die gesamten Anforderungen lesen und verstehen kann.
- **Atomar**  
Es darf nur eine Anforderung pro Abschnitt oder Satz beschrieben sein. Das Kriterium für ein „Atom“ sollte die Entscheidbarkeit einer Anforderung sein.
- **Identifizierbar**  
Jede Anforderung muss eindeutig identifizierbar sein (z. B. über eine Kennung oder Nummer).
- **einheitlich dokumentiert**  
Die Anforderungen und ihre Quellen sollten nicht in unterschiedlichen Dokumenten stehen oder unterschiedliche Strukturen haben.
- **Nachprüfbar**  
Die Anforderungen sollten mit Abnahmekriterien verknüpft werden, damit bei der Abnahme geprüft werden kann, ob die Anforderungen erfüllt wurden. Testfälle werden aus den Abnahmekriterien abgeleitet. Siehe auch Verifizierung.
- **rück- und vorwärtsverfolgbar**  
Es muss nachverfolgbar sein, ob eine Anforderung vollständig erfüllt wurde (vorwärts). Ebenso soll für jede implementierte Funktionalität kontrollierbar sein, aufgrund welcher Anforderungen sie erstellt wird (rückwärts), um Überflüssiges zu vermeiden. Siehe Rückverfolgbarkeit (Anforderungsmanagement).
- **Konsistent**  
Die definierten Anforderungen sind untereinander widerspruchsfrei.

Das Ergebnis der Anforderungsaufnahme ist eine Liste mit Anforderungen. Diese kann z. B. in ein **Lastenheft** überführt werden.

# STRUKTURIERUNG UND ABSTIMMUNG

Nach der Erfassung muss eine Strukturierung und Klassifizierung der Anforderungen vorgenommen werden. Damit erreicht man, dass die Anforderungen übersichtlicher werden. Dies wiederum erhöht das Verständnis der Beziehungen zwischen den Anforderungen. Kriterien sind hierbei:

- **Abhängig**  
Anforderungen müssen daraufhin überprüft werden, ob eine Anforderung die Voraussetzung für eine andere ist, sie sich gegenseitig bedingen oder sich unabhängig voneinander realisieren lassen.
- **Zusammengehörig**  
Anforderungen, die fachlich-logisch zusammengehören, sollen nicht allein realisiert werden.
- **Rollenbezogen**  
Jede Benutzergruppe hat ihre eigene Sicht auf die Anforderungen, die damit unterstützt werden soll, siehe Benutzerrolle.

Weitere Strukturierungsmöglichkeiten sind funktionale und nichtfunktionale Anforderungen sowie fachlich motivierte (fachliche und technische) und technisch motivierte (nur technische) Anforderungen. Die so strukturierten Anforderungen müssen dann zwischen Kunde und Entwickler abgestimmt werden. Diese Abstimmung kann gegebenenfalls zu einem iterativen Prozess werden, der zur Verfeinerung der Anforderungen führt.

# PRÜFUNG UND BEWERTUNG

Nach der Strukturierung, zum Teil auch parallel dazu, erfolgt die Qualitätssicherung der Anforderungen nach Qualitätsmerkmalen:

- **Korrekt**  
Die Anforderungen müssen untereinander widerspruchsfrei sein. Siehe Korrektheit.
- **Machbar**  
Die Anforderung muss realisierbar sein. Siehe Machbarkeit.
- **Notwendig**  
Was nicht vom Auftraggeber gefordert wird, ist keine Anforderung.
- **Priorisiert**  
Es muss erkennbar sein, welche Anforderungen die wichtigsten sind. Ziel der Priorisierung ist es, häufig benötigte oder dem Kunden besonders wichtige Funktionen vor den weniger häufig benötigten bereitzustellen. Man erreicht es über eine Quantifizierung der Funktionszweige.
- **nutzbar, nützlich**  
Auch bei teilweiser Realisierung soll bereits ein produktives System entstehen.

Das Ergebnis der Prüfung stellt die Basis für das **Pflichtenheft** dar. Die Bewertungen stehen teilweise in Konkurrenz zueinander. Eine Realisierung von nur als hoch priorisierten Aufgaben erbringt nicht automatisch ein produktives System. Bei der Bewertung ist nicht nur die Einzelfunktion für sich, sondern auch ihr Wirken im Gesamtsystem zu betrachten.



# LASTENHEFT





# LASTENHEFT

- Im Lastenheft definiert der Auftraggeber seine Erwartungen und die konkreten Anforderungen an die IT-Lösung.
  - Wird oft gemeinsam von Auftraggeber und Auftragnehmer erarbeitet.

# INHALTE EINES LASTENHEFTES

Einleitung und Ausgangssituation	Liefergegenstände und Metriken	Anforderungen	Unterstützung im Betrieb
<ul style="list-style-type: none"><li>• Kurzbeschreibung geforderte Lösung</li><li>• Projektziele</li><li>• Geplanter Ablauf Implementierung und Betrieb</li><li>• Kurzvorstellung Auftraggeber</li><li>• Projekt-Umfeld</li><li>• Bestehende IT-Infrastruktur</li><li>• Glossar</li></ul>	<ul style="list-style-type: none"><li>• Hardware</li><li>• Software</li><li>• Cloud-Services</li><li>• Dokumentation</li><li>• Implementierungsleistungen</li><li>• Abgrenzung: Eigenleistungen</li><li>• Useranzahl</li><li>• Datenmenge</li><li>• Verfügbarkeit</li><li>• Max. Ausfallzeit</li><li>• Anzahl Endgeräte</li><li>• Usw.</li></ul>	<ul style="list-style-type: none"><li>• Datenstruktur und Datenzugriff</li><li>• Funktionale Anforderungen</li><li>• Daten-Schnittstellen</li><li>• IT-Security und Datenschutz</li><li>• Nicht-Funktionale Anforderungen, wie Testsystem, Performance, Usability, Regelwerke</li></ul>	<ul style="list-style-type: none"><li>• Software-Updates</li><li>• Support</li><li>• Anpassung der IT-Lösung</li></ul>

# LASTENHEFT

## Wie detailliert sollte das Lastenheft sein?

Grob spezifizierte Anforderungen erlauben, dass der Lieferant eine möglicherweise unbrauchbare IT-Lösung liefert und der Vertrag aus rechtlicher Sicht trotzdem als erfüllt gilt.

Im Lastenheft sollten zumindest folgende Punkte im Detail spezifiziert sein:

- Geforderte Liefergegenstände und Leistungen
- Zu unterstützende Geschäftsprozesse
- Zu verwaltende Daten (Daten-Anforderungen)
- Wichtige/kritische Funktionen
- Daten-Schnittstellen



# LASTENHEFT

## Muss und Kann-Kriterien

Sie sollten bereits im Lastenheft die Kritikalität der Anforderungen, Liefergegenstände usw. definieren:

- Ein Muss-Kriterium (Must-have) ist für die vorgesehene Anwendung unbedingt erforderlich. Erfüllt das Angebot ein Muss-Kriterium nicht, ist die Lösung für den geplanten Einsatzzweck nicht brauchbar.
- Die Erfüllung eines Kann-Kriteriums (Nice-to-have) verbessert ein Angebot, ist aber verzichtbar.



# LASTENHEFT - EINLEITUNG

## Einleitung zum Lastenheft

In der Einleitung beschreiben Sie in Kurzform:

- Die geforderte Lösung
- Die Projektziele
- Den geplanten Ablauf der Implementierung sowie der Betriebsphase

Erstellen Sie ein Glossar für die verwendeten Fachbegriffe, um Missverständnisse zu vermeiden.

# LASTENHEFT - AUSGANGSSITUATION

## Kurzvorstellung Auftraggeber

Stellen Sie Ihr Unternehmen kurz vor:

- Unternehmenszweck
- Größe Ihres Unternehmens
- Standorte

# LASTENHEFT - AUSGANGSSITUATION

## Ausgangssituation

### Bestehende IT-Infrastruktur

Beschreiben Sie die relevante IT-Infrastruktur für das Projekt:

- Bestehende IT-Systeme, welche im Haus betrieben werden
- Genutzte IT-Lösungen, welche außer Haus betreiben werden (Cloud-Services)
- Ggf. Name/Kurzbeschreibung der IT-Lösung, welche abgelöst werden soll

# LASTENHEFT - AUSGANGSSITUATION

## Geforderte Liefergegenstände und Leistungen

Hier zählen Sie alle Produkte und Ergebnisse auf, welche vom Anbieter zu liefern bzw. zu erbringen sind:

- Softwarelizenzen (Produkt, Anzahl)
- Hardware (Produkt, Anzahl)
- Schulung (Anwenderschulung, Administrationsschulung, Online-Training, Trainingsmaterial)
- Handbücher (Anwender-Handbuch, Administrations-Handbuch)
- Cloud-Services, Hosting-Leistungen
- Support und Software-Updates für den Betrieb

# LASTENHEFT - AUSGANGSSITUATION

## Geforderte Liefergegenstände und Leistungen

Weiter nennen Sie Leistungen, welche für die Implementierung zu erbringen sind:

- Projektmanagement
- Konkretisierung der Anforderungen, Pflichtenheft-Erstellung (Fein-Spezifikation)
- Erstellen einer Schnittstellen-Spezifikation
- Installation und Konfiguration
- Unterstützung beim Systemtest
- Datenmigrationsleistungen (Daten von der alten auf die neue IT-Lösung übernehmen)
- Intensiv-Support vor Ort bei der Einführung des Systems (Go-Live)

# LASTENHEFT - AUSGANGSSITUATION

## Software Lizenz-Modelle

Es gibt drei übliche Software-Lizenzmodelle für die Anzahl der User:

### Fixed/named license

Eine Lizenz pro Installation am PC oder pro User

### Concurrent license

Eine Lizenz pro User für die gleichzeitige Nutzung (alle eingeloggten User werden gezählt)

### Site license

Lizenz für die Installation bei einem Unternehmen und Standort

Die Anzahl der User und Arbeitsplätze pro Lizenz ist unlimitiert



# LASTENHEFT - AUSGANGSSITUATION

## Software Lizenz-Modelle

Weiters bestimmt die Lizenz-Art eine zeitliche Limitierung des Nutzungsrechts:

### Unlimitiert (perpetual)

Lizenzen müssen nur einmal gekauft werden und können zeitlich unbefristet verwendet werden

### Wiederkehrend (subscription)

Monatliche, quartalsweise, jährliche Zahlung der Lizenzkosten

# LASTENHEFT - AUSGANGSSITUATION

## Eigenleistungen

Nennen die Leistungen, welche Sie als Auftraggeber selbst erbringen wollen.

Eigenleistungen könnten sein:

- Bereitstellung von IT-Infrastruktur (Netzwerk, Server, Storage, Verkabelung) nach den Vorgaben des Anbieters
- Bauliche Anpassungen
- Bereitstellung Klimatisierung, Elektrizität
- Projektmanagement
- Schulung der End-User durch die Key-User (Train-the-Trainer Prinzip)
- Stammdaten-Erfassung
- Konfigurationsarbeiten
- Testplanung, Testmanagement, Testdurchführung



# LASTENHEFT - ANFORDERUNG AN DIE GEWÜNSCHTE LÖSUNG

## Metriken der geforderten IT-Lösung

Typische Metriken sind:

- Geforderte Verfügbarkeit der IT-Lösung
  - z.B. rund-um-die-Uhr (7×24) oder während der Geschäftszeiten (Mo. – Fr. 8-18 Uhr)
  - Wenn die Lösung in mehreren Ländern verwendet wird, bedenken Sie unterschiedliche Feiertage und die Zeitverschiebung
- Für kritische IT-Lösungen: Max. Ausfallzeit
  - Wie lange darf die IT-Lösung maximal ausfallen, bevor es kritisch für den Betriebsablauf wird?
- User Anzahl
  - Named User (Anzahl User Accounts)
  - Concurrent User (gleichzeitige Nutzer)
- Datenmenge
  - Initial: Welche Datenmenge wird mit der Lösung von Beginn an verwaltet?
  - Datenwachstum: Mit welchem Wachstum ist über den Betriebszeitraum zu rechnen?
- Anzahl Endgeräte (PCs, mobile Geräte.) für die Verwendung der IT-Lösung

# LASTENHEFT - ANFORDERUNG AN DIE GEWÜNSCHTE LÖSUNG

Daten-Anforderungen

Datenstruktur

Hierzu beschreiben Sie die geforderte Datenstruktur mit einer Liste von Datenobjekten und Attributen.

Zum Beispiel:

<i>Datenobjekte</i>	<i>Attribute</i>
Kunden/Interessenten	Nummer Name Adresse Telefonnummer E-Mail Status (Kunde, Interessent, gesperrt)

# LASTENHEFT - ANFORDERUNG AN DIE GEWÜNSCHTE LÖSUNG

## Zugriffsrechte und Rollen

Definieren Sie die User-Rollen und Zugriffsrechte.

Beispiel:

- Rolle Mitarbeiter
  - kann alle Daten außer Finanzdaten einsehen
  - keine Änderungen erlaubt außer bei der Zeiterfassung
- Rolle Auftragssachbearbeiter
  - kann Auftragsdaten einsehen und ändern
  - keine Stammdatenänderungen außer Kundendaten
  - keine User-Verwaltung

# LASTENHEFT - ANFORDERUNG AN DIE GEWÜNSCHTE LÖSUNG

## Zugriffsrechte und Rollen

- Rolle Buchhalter
  - kann Buchhaltungs- und Finanzdaten einsehen und ändern
  - keine Stammdatenänderungen
  - keine User-Verwaltung
- Rolle Stammdaten-Administrator
  - kann Stammdaten ändern
  - keine User-Verwaltung
- Rolle Administrator
  - User-Verwaltung erlaubt
  - darf Stammdaten nicht ändern

# LASTENHEFT - ANFORDERUNG AN DIE GEWÜNSCHTE LÖSUNG

## Zugriffsrechte und Rollen

Spezifizieren Sie das gewünschte Konzept zur Zuordnung der Berechtigungen:

- Ein User hat mehrere Rollen: Damit bekommt der User die Zugriffsrechte von allen zugeordneten Rollen.
- Ein User hat eine Rolle: Alle verwendeten Kombinationen an Berechtigungen müssen mit einer Rolle abgedeckt werden.
- Das Konzept „ein User, mehrere Rollen“ ist leichter zu administrieren als „ein User, eine Rolle“.

Alternativ können Sie vom Anbieter eine Beschreibung fordern, wie mehrere Rollen pro User abgebildet werden (Frageliste).



# **FUNKTIONALE ANFORDERUNGEN**



# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Funktionale Anforderungen

### Zu unterstützende betriebliche Abläufe (Prozesse)

Die IT-Lösung muss die relevanten betrieblichen Abläufe unterstützen. Beschreiben Sie deshalb die von der IT-Lösung betroffenen betrieblichen Abläufe, um den Anbietern ein Bild vom Gesamtsystem zu geben. Nennen Sie auch seltene Ereignisse, wie z.B. den Jahresabschluss.

Der Anbieter soll in seinem Angebot beschreiben, wie die angebotene IT-Lösung die betrieblichen Abläufe unterstützt (Frageliste).

Wenn Sie diese Prozesse bereits dokumentiert haben, dann legen Sie Prozessdokumentation dem Lastenheft bei.



# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Funktionale Anforderungen

### Use Cases (Anwendungsfälle)

Sollten Sie keine Prozessbeschreibungen haben, dann beschreiben Sie die typischen Anwendungsfälle (Use Cases) für die geforderte IT-Lösung.

Dafür eignet sich am besten die Darstellung als Anwendungsfall-Diagramm oder eine Darstellung als Aktivitäts-Diagramm. Die Beschreibung kann aber auch einfach in Textform erfolgen.



# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Dateneingabe und Bearbeitung

Typische Anforderungen dazu sind:

- Methoden der Dateneingabe z.B. via Touchscreen, Scan von Barcodes/QR-Codes
- Datenimport von Datei, Datenbank etc.
- Datenübernahme von Geräten z.B. Scanner, Sensoren
- Unterstützung der Ersterfassung von Daten z.B. das bestimmte Muss-Datenfelder anfangs leer gelassen werden dürfen, da noch nicht bekannt
- Änderung und Löschung von bestehenden Daten z.B. Einschränkung auf bestimmte User-Rollen
- Korrektur von Falscheingaben nach dem Speichern, z.B. wie lange ist eine Änderung möglich

# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Datenverarbeitung und Auswertungen

Typische Anforderungen sind:

- Berechnungen z.B. automatische Planung
- Generierte Dokumente z.B. Rechnung
- Daten-Visualisierungen und Statistiken z.B. Dashboard
- Berichte z.B. Monatsbericht
- Format-Vorgaben für Dokumente (Corporate Design)
- Ausgabemöglichkeiten für Dokumente und Berichte z.B. Drucker, PDF, E-Mail Versand, Fax
- Workflow-Support z.B. Freigabeprozess
- Protollierung, Nachvollziehbarkeit von Änderungen (Audit-Trail)

# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Daten-Schnittstellen

Spezifizieren Sie die Datenschnittstellen, welche zum Datenaustausch benötigt werden:

- Details zu den involvierten IT-Systemen
- Ausgetauschte Daten-Objekte (wie unter Daten Anforderungen spezifiziert)
- Bestimmung des Senders und Empfängers für die ausgetauschten Daten-Objekte
- Authentifizierung und Autorisierung zwischen den IT-Systemen
- Datentransfer: Protokolle, Verschlüsselung
- Vorgangsweise bei Ausfall eines der IT-Systeme (Zwischenspeicherung, Wiederholung der Übertragung)
- Schnittstellen-Protokollierung (Logging)

# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Anforderungen zu IT-Security und Datenschutz

Datenschutz- und IT-Security Anforderungen haben meist einen gesetzlichen oder vertraglichen Hintergrund. Weiter sollten die Maßnahmen für Elementarschäden (Feuer, Wasser) und Katastrophen vorsorgen.

Im Folgenden finden Sie ein paar generelle Anforderungen, weitere spezifische IT-Security Anforderungen finden Sie hier:

[IT-Sicherheitshandbuch für KMU](#) der WKO

[IT-Grundschutz-Kataloge](#) des BSI

Die Sicherheit einer IT-Lösung ist besonders kritisch, wenn der Zugriff aus dem Internet möglich ist (Hacking-Gefahr). In diesem Fall sollten Sie einen IT-Security Experten für das Design und die Auswahl der IT-Lösung hinzuziehen bzw. IT-Security-Expertise vom Anbieter einfordern.



# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Schutz der Daten vor Hardware-Defekten oder Elementarschäden (Feuer, Wasser, etc.)

Eine betriebskritische IT-Lösung benötigt eine redundante Datenhaltung. Damit ist die Speicherung der Daten auf mehreren Festplatten bzw. mehreren Storage-Systemen zum Schutz vor Datenverlust bei Hardware-Defekten gemeint.

Redundante Datenhaltung ist dabei nicht gleich Datensicherung (Backup). Da Backups typischerweise nicht permanent laufen, kommt es bei einem Hardware-Defekt ggf. zum Verlust der Änderungen seit dem letzten Backup.

# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

Fordern Sie eine Beschreibung im Angebot (Frageliste), wie das System mit dem Ausfall einer oder mehrerer Disks umgeht:

- Viele Storage-Systeme haben die Funktionalität zur redundanten Datenspeicherung bereits eingebaut (RAID)
- Auch eine Alarm-Funktion bei einem Disk-Defekt ist wichtig, um reagieren zu können bevor es zu weiteren Defekten und damit zu einem Komplettausfall kommt.

Handelt es sich um ein besonders kritisches System, so ist die Datenhaltung in verschiedenen Rechenzentren sinnvoll:

- Die konkrete technische Umsetzung hängt von der verwendeten Anwendung und der Entfernung zwischen den Rechenzentren ab
- Manche Applikationen/Datenbanken bieten Standard-Funktionen zur parallelen Speicherung auf mehreren Storage-Systeme
- Auch Virtualisierungslösungen können mit Datensynchronisierung realisiert werden.

# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Backups

Weiter spezifizieren Sie Details zum Backup, wenn die Datensicherung vom Bieter eingerichtet werden soll:

- Mit welcher Methode und wie oft die Daten gesichert werden müssen
  - vollständige oder inkrementelle Datensicherung
  - mehrmals am Tag, täglich, wöchentlich
  - Offline oder online: Für intensiv genutzte und betriebskritische Systeme sollte die Datensicherung mehrmals am Tag erfolgen, jedoch ohne damit Verfügbarkeit der Lösung einzuschränken (Online-Backup).
- Wie lange Backups aufbewahrt werden müssen
  - Die Rücksicherung von Daten sollte zumindest über den Zeitraum von einem Monat nach der Sicherung möglich sein.
  - Denken Sie dabei an den Fall einer versehentlichen Datenlöschung, welche erst nach einiger Zeit entdeckt wird.

# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Security-Anforderungen für Software

- Schutz vor unberechtigttem Applikations- und Datenzugriff (Login mit Username/Password, Zwei-Faktor-Authentifizierung)
- Berechtigungskonzept für die Beschränkung des Datenzugriffs für eingeloggte User (siehe Datenzugriff)
- Verschlüsselte Speicherung aller oder nur kritischer Daten (z.B. Login-Namen, Passwörter, personenbezogene Daten)
- Verschlüsselter Datenaustausch mit anderen IT-Systemen
- Security-Updates zeitnah nach Veröffentlichung mit nach Möglichkeit automatischer Installation



# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Security-Anforderungen für Server/Arbeitsplatzrechner, Betriebssysteme

- Systemzugriff ist nur nach erfolgreicher Anmeldung möglich
- Automatische Sperre nach längerer Inaktivität
- Anti-Viren/Anti-Malware Software
- automatische Installation von sicherheitsrelevanten Updates
- System-Redundanz

# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Security-Anforderungen für IT-Infrastruktur

- Zentrale Komponenten (Netzwerk-Geräte, Server) befinden sich in einem versperrten Raum und nur Befugte haben Zutritt
- Die Komponenten sind vor Gefahren wie Feuer, Wasser geschützt und eine Überhitzung der Komponenten wird vermieden (Klimaanlage)
- Eine Notstromversorgung (USV) überbrückt kürzere Stromausfälle und bei einem längeren Stromausfall erfolgt ein ordentlicher Shutdown
- Ordentliche und gut beschriftete Verkabelung

# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## Security-Anforderungen für Cloud-Services

Der Anbieter ist zertifiziert bzw. erfüllt:

- [ISO 27001](#) (Informationssicherheit)
- [ISO 27018](#) (Datenschutz)
- [SSAE 16](#) (alt: [SAS 70](#), Service Organisation Audit)
- [Eurocloud Star Audit](#)
- Einhaltung der DSGVO
- Das BSI bietet einen eigenen [Anforderungskatalog für Cloud-Services](#).



# LASTENHEFT – FUNKTIONALE ANFORDERUNGEN

## IT-Security muss laufend angepasst werden

Da laufend neue Bedrohungsszenarien auftauchen, muss die IT-Security auch laufend überwacht und ggf. angepasst werden.

Wenn Sie diese Aufgabe nicht selbst durchführen können, so holen Sie sich dafür einen kompetenten Dienstleister. Um sicherzustellen, dass der Dienstleister diese Aufgabe auch ernst nimmt, sollten Sie eine entsprechende Schadenshaftung vereinbaren.



# **NICHT-FUNKTIONALE ANFORDERUNGEN**



# LASTENHEFT – NICHT-FUNKTIONALE ANFORDERUNGEN

## Nicht-Funktionale Anforderungen:

- System
  - Testsystem
  - System-Redundanz
  - Performance
  - Hardware-Anforderungen
  - Ergonomie, Usability
- Regelwerke
  - Gesetze
  - Branchenregelwerke
  - Interne Vorgaben

# LASTENHEFT – NICHT-FUNKTIONALE ANFORDERUNGEN

## Testsystem

Für z.B. eine Test-Installation eines Updates oder das Nachvollziehen von Problemen sollte ein Testsystem implementiert werden, um dafür das Produktivsystems nicht zu beeinträchtigen.

Testsysteme werden gerne als virtuelles System betrieben (Virtualisierungslösung). Da Testsysteme eher selten in Gebrauch sind, kann damit der Hardware-Bedarf reduziert werden.

Für ein Testsystem benötigen Sie auch Software-Lizenzen. Manche Anbieter bieten diese kostenlos bzw. günstiger im Bundle mit den Lizenzen für Produktivsysteme an.

# LASTENHEFT – NICHT-FUNKTIONALE ANFORDERUNGEN

## Hardware-Anforderungen

Wenn Sie die Auswahl der Hardware nicht komplett dem Anbieter überlassen wollen, spezifizieren Sie Ihre Anforderungen:

- Hardware-Ausstattung (CPU, Memory, Storage, Anschlüsse etc.)
- Schutzklasse (Staub/Wasser/Erschütterung/Einbruch)
- Abmessungen bzw. Einbauart (Rack)
- Limitierung Energieverbrauch
- Limitierung Kühlungsbedarf
- Zu unterstützende Betriebssysteme
- Funktionen zur Hardware-Konfiguration (z.B. Remote Hardware-Management)
- Software-Ausstattung
  - Betriebssystem
  - Standard-Software wie Virens Scanner, Java etc.



# LASTENHEFT – NICHT-FUNKTIONALE ANFORDERUNGEN

## System-Redundanz, Virtualisierung

Diese Anforderung resultiert aus den geforderten Metriken „Verfügbarkeit“ und „Max. Ausfallzeit“.

Ein redundant ausgelegtes System wird heutzutage gerne mit einer Virtualisierungslösung realisiert, was Vorteile und Nachteile hat:

- Der Betrieb von virtuellen Servern ermöglicht die Wartung, den Austausch oder die Erweiterungen der physischen Hardware ohne/mit geringen Ausfallzeiten für die IT-Lösung
- Auf der Virtualisierungslösung können mehrere virtuelle Server gleichzeitig betrieben werden, womit die physische Hardware besser ausgelastet wird
- Bei Bedarf können die Hardware-Ressourcen (CPU, Memory) zwischen den virtuellen Servern einfach und schnell verschoben werden
- Die Virtualisierungstechnologie erhöht die Komplexität und damit den Administrationsaufwand und die Fehlermöglichkeiten
- Die Virtualisierungstechnologie kostet etwas Hardware-Leistung
- Die virtuellen Server könnten sich gegenseitig negativ beeinflussen (Performance, Verfügbarkeit, Security-Incidents)
- Die Virtualisierungslösung verursacht auch Kosten (Lizenzen, Support)
- In der Praxis überwiegen die Vorteile der Virtualisierungstechnologie

# LASTENHEFT – NICHT-FUNKTIONALE ANFORDERUNGEN

## Performance

Wenn die Reaktions- und Verarbeitungsgeschwindigkeit der IT-Lösung kritisch ist, sollten Sie dafür konkrete Anforderungen spezifizieren.

Dafür spezifizieren Sie konkrete Tests, um die Erfüllung später objektiv bewerten zu können, z.B.:

- Zeit bis zur Bereitschaft für die Eingabe/Auswahl nach dem Start der Anwendung
  - Die Anwendung muss nach dem Start in max. 30 sec bereit für die Auswahl/Eingabe sein
- Max. Dauer für die Generierung eines bestimmten Berichts
  - Die Umsatzstatistik muss binnen 1 min verfügbar sein (Umsatzdaten für 200 Filialen über 12 Monate).

# LASTENHEFT – NICHT-FUNKTIONALE ANFORDERUNGEN

## Ergonomie, Usability

Spezifizieren Sie besondere Anforderungen der User hinsichtlich Bedienung, z.B. aufgrund körperlicher Beeinträchtigung, Schutzbekleidung.

Beispiele:

- Die Schriftgröße muss für jeden User individuell einstellbar sein.
- Die Schrift am Bildschirm hat mindestens 10pt und kann auf 20pt vergrößert werden.
- Durch die Vergrößerung der Schrift darf kein Text verschwinden bzw. darf die Bedienung nicht beeinträchtigt werden.
- Die Bedienung muss auch mit wärmeisolierten Handschuhen auf dem Touchscreen einfach möglich sein.
- Die Elemente auf dem Bildschirm müssen so platziert sein, dass die Auswahl eines Elements für die meisten User mit Handschuhen ohne unabsichtliche Mehrfachauswahl möglich ist.



# **ANFORDERUNGEN ZUR UNTERSTÜTZUNG IM BETRIEB**



# LASTENHEFT – UPDATES UND SUPPORT

## Software-Updates

Jeder Software-Hersteller hat einen definierten Modus, wie Kunden Software-Updates bekommen:

- „Minor Updates“, Patches, Fixes sind in der Regel für einen gewissen Zeitraum kostenlos. Typischerweise entspricht das den Ansprüchen aus der Gewährleistung.
- Update-Vertrag: alle Software-Updates sind im Rahmen der Laufzeit des Update-Vertrages kostenlos. Sollte der Vertrag beendet und später wieder aktiviert werden, werden Aufschläge verrechnet (Upgrade-Kosten).
- „Major Updates“ bzw. Upgrades bei Bedarf: Sie kaufen die neue Software-Version und kommen in der Regel einen Upgrade-Rabatt.



# LASTENHEFT – UPDATES UND SUPPORT

Jede Software-Version erreicht nach einer definierten Zeit (herstellerabhängig) den Status „End of Life“. Damit werden keine neuen Updates mehr bereitgestellt. Um die Software weiter aktuell zu halten, ist dann ein Upgrade auf eine neuere „Major Release“ erforderlich. Damit sind bei einem Nutzungszeitraum über 10 Jahre in der Regel mehrere Software-Upgrades notwendig.

Upgrades haben typischerweise größere Auswirkungen auf das IT-System, womit diese am besten im Rahmen eines IT-Projekts implementiert werden.

Um Kosten zu sparen, können Sie im Support-Vertrag auf kostenlose Upgrades verzichten (herstellerabhängig).

# LASTENHEFT – UPDATES UND SUPPORT

## IT-Support

Für die zu erbringenden Support-Leistungen spezifizieren Sie:

- Welche Art von Support der Anbieter zu erbringen hat
  - Support-Level (First-Level, Second-Level, Third-Level) bzw. Leistungen darüber hinaus
  - Telefon-Support, Vor-Ort-Support
- Wie die Kommunikation mit dem Support erfolgen muss
  - Welche Sprachen müssen die Support-Mitarbeiter sprechen?
  - Meldung per Telefon, E-Mail, Support-Ticket?

# LASTENHEFT – UPDATES UND SUPPORT

## IT-Support

Für die zu erbringenden Support-Leistungen spezifizieren Sie:

- Welche Verfügbarkeit und Reaktionszeit vom Support gefordert sind
  - Zu welchen Zeiten muss der Support erreichbar sein?
  - In welchem Zeitraum muss sich ein Support-Mitarbeiter melden?
  - In welchem Zeitraum muss sich ein Experte für die IT-Lösung melden?
- Welche Problembehebungszeiten garantiert werden müssen
  - In welchem Zeitraum muss ein schweres bzw. kritisches Problem behoben werden?
  - In welchem Zeitraum muss ein normales Problem behoben werden?





# LASTENHEFT – UPDATES UND SUPPORT

## Spätere Anpassungen der IT-Lösung

Eine IT-Lösung anzupassen oder zu erweitern kann teuer werden: Manche Lieferanten nutzen die Abhängigkeit des Kunden aus und verlangen deutlich mehr, als für eine vergleichbare Leistung in einer Erst-Implementierung.

Deshalb sollten Sie bereits bei der Auswahl der IT-Lösung eine Kostenerhebung für eine mögliche Erweiterung durchführen und diese bei der Auswahl der IT-Lösung berücksichtigen. Idealerweise vereinbaren Sie dafür eine Preisbindung im Liefervertrag.

# LASTENHEFT – UPDATES UND SUPPORT

## Spätere Anpassungen der IT-Lösung

Beschreiben Sie einige Szenarien zur zukünftigen Anpassung der IT-Lösung, wie z.B.:

- Neuen Bericht bzw. neue Statistik implementieren
- Bestehendes Formular um ein Datenfeld erweitern
- Nicht mehr benötigtes Datenfeld ausblenden
- Erweiterung des Speicherplatzes
- System in einer weiteren Abteilung/Niederlassung verwenden
- Anpassung der System-Schnittstellen nach Änderung eines angebundenen IT-Systems
- Anpassung an geänderte gesetzliche Vorgaben bzw. Standards



# **ANFORDERUNGEN KATEGORISIEREN (KANO MODELL)**





# WARUM ANFORDERUNGEN KATEGORISIEREN

- Bei sehr kleinen Projekten nicht notwendig
- Bei komplexen Projekten sehr sinnvoll
  - Welche Anforderungen haben welchen Einfluss auf die Kundenzufriedenheit?



# KANO MODELL

- Von Noriaki Kano (Professor an der Universität von Tokio) entwickelt
- Beschreibt Zusammenhang zwischen dem Erreichen bestimmter Eigenschaften eines Produkts und der zu erwarteten Zufriedenheit von Kunden

# KANO MODELL

## Basismerkmale

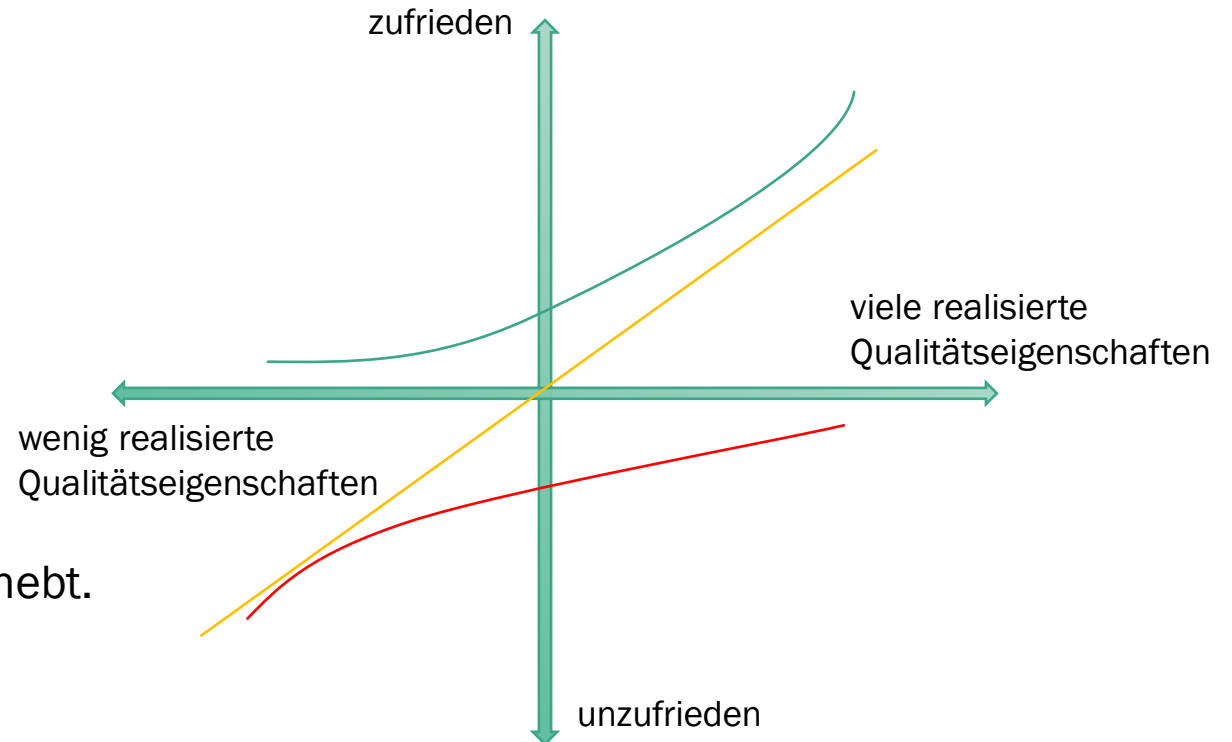
Merkmale, bei denen der Kunde davon ausgeht, dass sie vorhanden sind z.B. Toolleiste für Formatierungen bei einem Texteditor

## Leistungsmerkmale

z.B. verschiedene Layouts für Texteditor

## Begeisterungsmerkmale

Nutzenstiftende Merkmale, mit denen der Kunde nicht unbedingt rechnet, Mit der sich das Produkt von der Konkurrenz abhebt.  
z.B. Übersetzungsfunktion im Texteditor





# AUFWANDSSCHÄTZUNG



# WARUM AUFWANDSSCHÄTZUNG

- Softwareentwicklung kostet Geld
  - Personalkosten
  - Sachkosten
- Kosten müssen bekannt sein, um die Wirtschaftlichkeit zu prüfen
- Durch Aufwandsschätzung können Termine bestimmt werden



# AUFWANDSSCHÄTZUNG METHODEN

- Expertenschätzung
  - Einzelschätzung
  - Mehrfachbefragung
  - Delphi-Methode
  - Schätz-Klausur
- Algorithmische Schätzung
  - Cocomo (Constructive Cost Modell)
  - Function Point Methode



# DELPHI-METHODE





# DELPHI-METHODE

- Systematisches, mehrstufiges Schätzverfahren
- Wird verwendet, wenn Prognosen aufgestellt werden müssen
- 1963 von der RAND-Corporation(amerikanische Denkfabrik) entwickelt

# DELPHI-METHODE VORGEHENSWEISE

- Es werden erfahrene Experten gesucht
- Jedem Experten werden die Anforderungen vorgelegt
- Jeder Experte soll einzeln für sich eine Prognose für den Aufwand abgeben
- Danach werden alle Ergebnisse ausgewertet
- Auswertungen werden den einzelnen Experten wieder vorgelegt
  - **Wichtig:** Auswertungen sind anonymisiert, damit dominante Persönlichkeiten nicht zuviel Einfluss haben
- Jeder Experte kann eigene Prognose nun wieder anpassen
- Schritte werden so lange wiederholt, bis in einem gewissen Toleranzbereich ein Konsens entsteht



# **COCOMO AUFWANDSSCHÄTZUNG**





# WAS IST COCOMO

- COCOMO: Constructive Cost Model In der Softwareentwicklung zur Kosten- und Aufwandsschätzung verwendet
- 1981 von Barry W. Boehm entwickelt
- Algorithmisches Kostenmodell, dass auf einer Kombination von Gleichungen, statischen Modellen und Schätzungen von Parametervariablen beruht

# RAHMENBEDINGUNGEN

- 17 identifizierte Kostentreiber, die in Algorithmen eingeflossen sind
- **Primärer Kostentreiber:** Delivered Source Instructions (DSI)
- **Beginn:** Entwurf des Produktdesigns
- **Ende:** Abschließende Tests durch den Kunden
- Anforderungsspezifikation darf sich im Laufe des Entwicklungsprozesses nicht mehr gravierend ändern

# GROBER ABLAUF

- **Schritt 1:** Anzahl der auszuliefernden Codezeilen ermitteln
  - Einheit: KDSI (1000 Delivered Source Instructions)
- **Schritt 2:** Komplexität bestimmen
  - Organic Mode (kleine bis mittelgroße Projekte)
  - Semidetached Mode(mittelgroße Projekte) 50-350 KDSI
  - Embedded Mode(große Projekte) ab 350 KDSI
- **Schritt 3:** Aufwand und Projektdauer berechnen
  - **Einheit vom Aufwand:** Personenmonate (19 Arbeitstage mit jeweils 8 Arbeitsstunden)