

- 01 Einführung
- 02 Prozessmodelle
- 03 Konfigurationsmanagement
- 04 Requirements Engineering
- 05 Modellierung
- 06 Qualitätsmanagement



Nach dieser Vorlesungseinheit ...

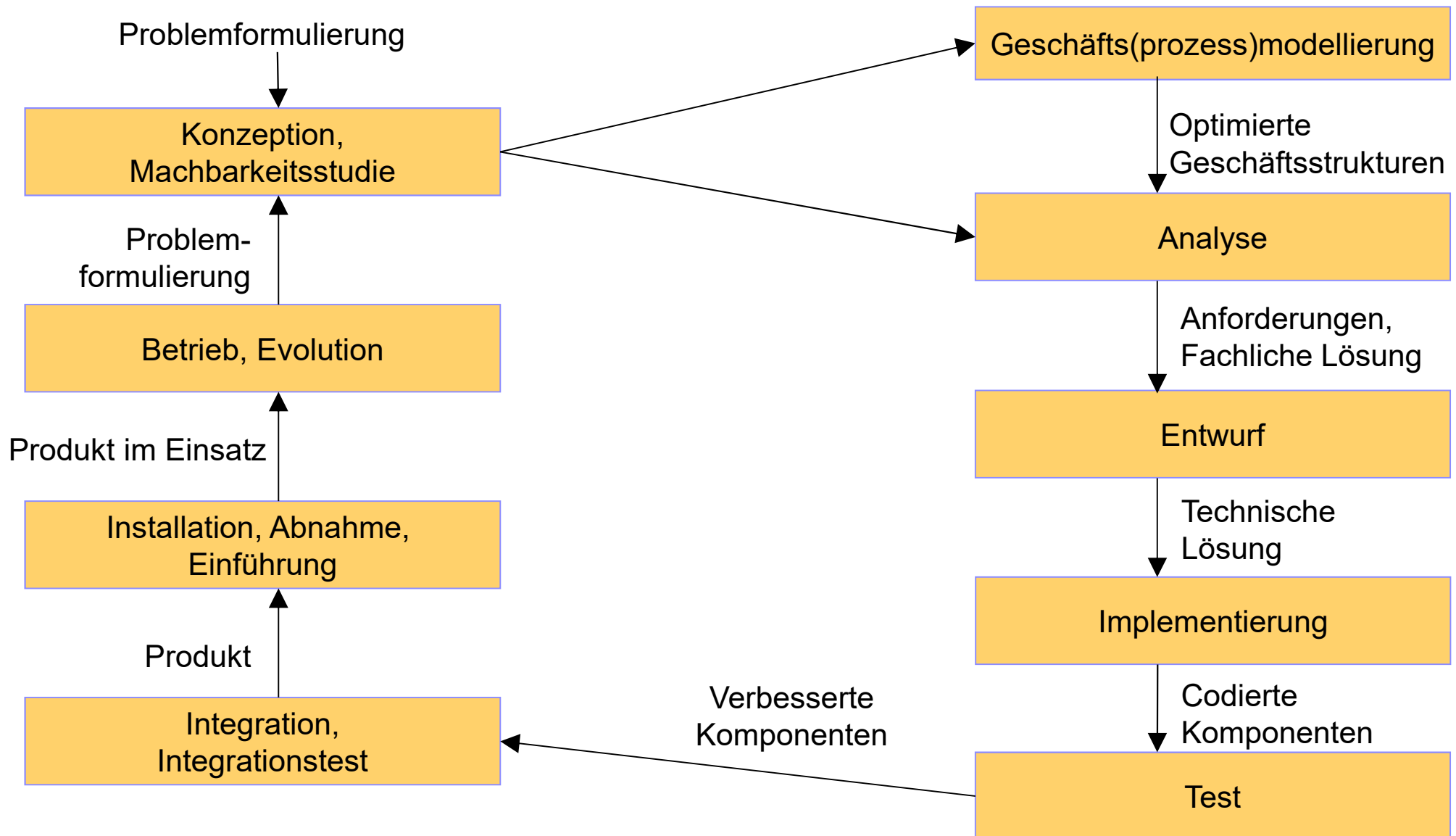
- ... verstehen Sie den **typischen Ablauf (Softwarelebenszyklus)** von Softwareentwicklungsprojekten, kennen die wichtigen **Aufgaben** und **Ergebnisse** sowie unterstützende **Teildisziplinen**
- ... wissen Sie was **Vorgehensmodelle** sind
- ... können Sie **verschiedene** in der Praxis auftretende **Vorgehensmodelle** charakterisieren
- ... wissen Sie was Sie bei der **Auswahl und Anpassung** eines Vorgehensmodell **für ein konkretes Projekt beachten müssen**



- 01 Einführung
- 02 Prozessmodelle
 - 02.1 Softwarelebenszyklus
 - 02.2 Basis-Vorgehensmodelle
 - 02.3 Monumentale Vorgehensmodelle
 - 02.4 Agile Vorgehensmodelle
- 03 Konfigurationsmanagement
- 04 Requirements Engineering
- 05 Modellierung
- 06 Qualitätsmanagement

Softwareentwicklung – Softwarelebenszyklus

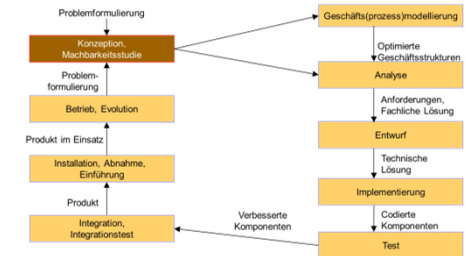
02 Prozessmodelle / 02.1 Softwarelebenszyklus



Konzeption und Machbarkeitsstudie

02 Prozessmodelle / 02.1 Softwarelebenszyklus

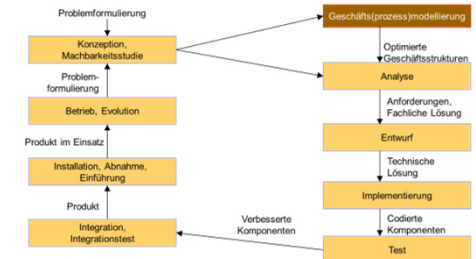
- Abschätzen von **Kosten** und **Nutzen** der geplanten Softwareentwicklung
- Aufgaben
 - Problem eingrenzen, informell und abstrahiert beschreiben
 - Verschiedene Lösungsansätze erarbeiten und evaluieren
 - Technologischer Proof of Concept („Geht das überhaupt?“)
 - Zu erwartenden Aufwand und Ertrag abschätzen
 - Angebot erstellen
- Ergebnisse
 - Lastenheft = (sehr) grobes Pflichtenheft
 - Projektkalkulation und Projektplan
 - Angebot an Auftraggeber



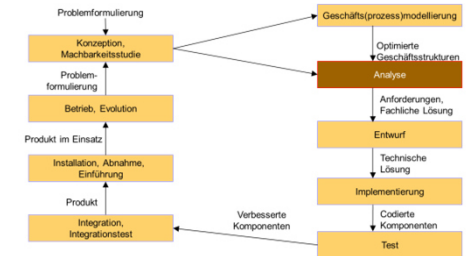
Geschäftsprozessmodellierung, Business Engineering

02 Prozessmodelle / 02.1 Softwarelebenszyklus

- Optimieren der zu unterstützenden **Geschäftsprozesse** und **-strukturen**, Ausrichten auf die softwaretechnischen Möglichkeiten
- Aufgaben
 - Bestehende Geschäftsprozesse und -strukturen erfassen
 - Schwachstellen ermitteln und analysieren
 - Soll-Konzept erstellen
 - Erste Verbesserungsprojekte aufsetzen
 - Culture Change einleiten
- Ergebnisse
 - Modell des Ist-Zustands der Prozesse und der Organisation
 - Modell des Soll-Zustands, ggf. teilweise umgesetzt
 - Priorisierung der Veränderungsmaßnahmen



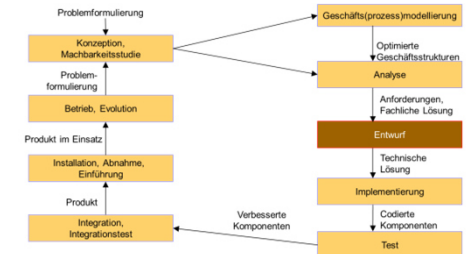
- Exakt festlegen, **was** die Software leisten soll, aber nicht, wie diese Leistungsmerkmale erreicht werden
- Aufgaben
 - Vorstellungen der Kunden ermitteln, konsolidieren und konkretisieren
 - Zu realisierende Systemeigenschaften präzise beschreiben: z.B. Funktionalität, Performanz, Benutzungsschnittstelle, Portierbarkeit, ...
 - Testfälle bestimmen
 - Erforderliche Dokumentation festlegen
- Ergebnisse
 - **Anforderungsspezifikation** (Pflichtenheft) und **Produktmodell** (fachliche Lösung)
 - Plan für Akzeptanztest



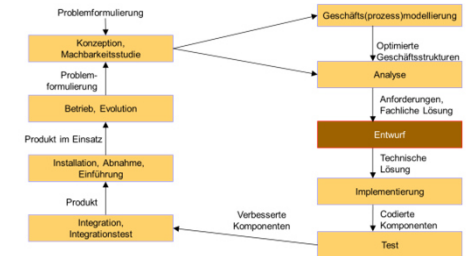
Entwurf (Design, Programming in the Large)

02 Prozessmodelle / 02.1 Softwarelebenszyklus

- Exakt festlegen, **wie** die Software aufgebaut ist (Softwarearchitektur) und wie die Leistungsmerkmale der Software realisiert werden (**technische Lösung**)
- Aufgaben
 - Geeignete Technologie auswählen
 - Programmieren im Großen: Bauplan entwickeln (**Architektur**)
 - System in Teilsysteme bzw. Module zerlegen (**Grobentwurf**)
 - Bestehende Softwarebibliotheken, Rahmenwerke etc. auswählen
 - Modulschnittstellen und Algorithmen vorgeben (**Feinentwurf**)
- Ergebnisse
 - Entwurfsdokumente mit Architektur, Grob- und Feinentwurf
 - Detaillierte(re) Testpläne



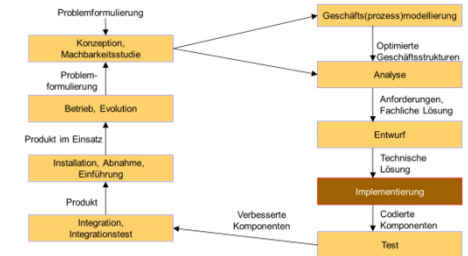
- Geeignete technologische Plattform auswählen
- Ggf. bereits bestehende Systeme einbinden
- Unterstützende Softwareprodukte auswählen (Application Server, Datenbank, ...)
- Strukturierung der Software in (logische) Schichten
- Komponenten, Module, Pakete
- Schnittstellen
- Verbindungen zwischen Modulen
- Verhalten



Implementierung (Programmieren im Kleinen)

02 Prozessmodelle / 02.1 Softwarelebenszyklus

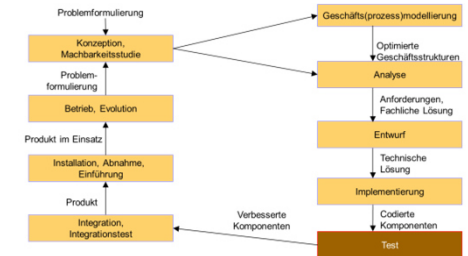
- **Realisieren** der einzelnen Module in einer bestimmten Reihenfolge
- **Aufgaben**
 - Implementieren der einzelnen Module
 - Programmierrichtlinien einhalten
 - Quelltext dokumentieren
 - Ggf. weitere Dokumentation erstellen
 - Für kritische Module Code-Inspektionen durchführen
- **Ergebnisse**
 - Menge der realisierten Module
 - Implementierungsbericht (Abweichungen von Entwurf, Zeitplan, etc.)
 - Dokumentation



Test, Validierung und Verifikation

02 Prozessmodelle / 02.1 Softwarelebenszyklus

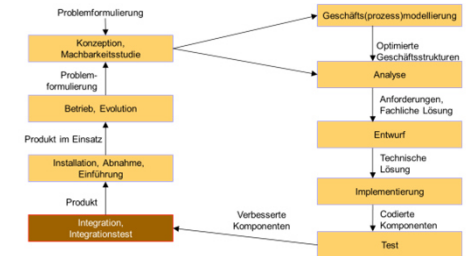
- **Prüfen** der einzelnen realisierten Module auf isolierte Weise
- Aufgaben
 - Realisieren von Testfällen
 - Erstellte Module testen
 - Ggf. Verifikation des Moduls gegen die (formale) Spezifikation
- Ergebnisse
 - Testprotokolle
 - Ggf. detaillierte Fehlerberichte



Integration und Integrationstest

02 Prozessmodelle / 02.1 Softwarelebenszyklus

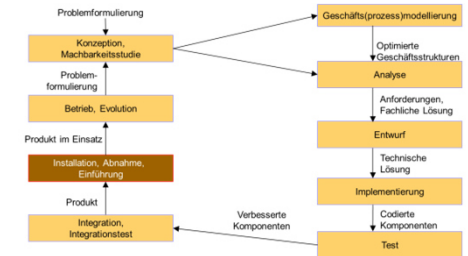
- Schrittweises Zusammenbauen der einzelnen Module zu einem Gesamtsystem und Testen desselben
- Aufgaben
 - Module zusammenbauen
 - Gesamtsystem testen (durch Entwicklungsorganisation; **α -Test**)
 - Dokumentation fertig stellen
- Ergebnisse
 - Fertiges System
 - Benutzerhandbuch
 - Technische Dokumentation
 - Testprotokolle



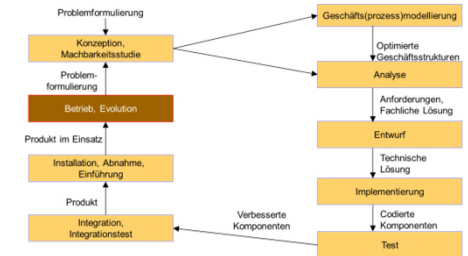
Installation, Abnahme und Einführung

02 Prozessmodelle / 02.1 Softwarelebenszyklus

- Installation des Softwaresystems beim Kunden entspricht Auslieferung; Inbetriebnahme meist in zwei Phasen
- Aufgaben
 - Auslieferung an ausgewählte Benutzer (**β-Test**)
 - Probebetrieb durchführen
 - Auslieferung an alle Benutzer
 - Benutzer schulen
- Ergebnisse
 - Fertiges System im produktiven Einsatz
 - Dokumentation der Akzeptanztests



- Wartung (Maintenance) und Weiterentwicklung des Softwaresystems, ggf. durch neuen Entwicklungszyklus



- Aufgaben
 - Ca. 20-25% Fehler beheben (Corrective Maintenance)
 - Ca. 20-25% Anpassungen durchführen (Adaptive Maintenance)
 - Ca. 50-60% Verbesserungen vornehmen (Perfective Maintenance)
- Ergebnisse
 - Problembereiche für die Software (Bug Reports)
 - Änderungsvorschläge für die Software
 - Neue Softwareversionen (Software Releases, Service Packs)

Wichtige unterstützende Teildisziplinen

02 Prozessmodelle / 02.1 Softwarelebenszyklus

- Projektindividuell
 - Projektmanagement
 - Qualitätsmanagement
 - Konfigurationsmanagement
 - Versionierung, Datensicherung, Integration bei Entwicklung im Team, Verwaltung konsistenter Entwicklungsstände
 - Problem- und Änderungsmanagement
- Projektübergreifend
 - Kontinuierliche Prozessverbesserung
 - Systematisches Lernen aus den Erfahrungen von laufenden/abgeschlossenen Projekten für zukünftige Projekte
 - Portfolio Management
 - Projektübergreifende Planung des Bedarfs an bzw. des Einsatzes von (personellen) Ressourcen, Know-how, ...

