

# Systems Engineering (SE)

© Copyright 2020

Document Name: Systems Engineering.docx

Last update: 11.06.2020

Autor: A. Balogh

## Content

<i>Einführung</i> .....	7
<i>EA - ENTERPRISE ARCHITECTURE</i> .....	7
<i>EA - FRAMEWORKS</i> .....	7
<b>TOGAF</b> .....	7
<b>Zachman Framework</b> .....	8
<b>SABSA - Sherwood Applied Business Security Architecture</b> .....	9
<b>FEAF - Federal Enterprise Architecture Framework</b> .....	9
CRM - Consolidated Reference Model .....	10
<b>DoDAF - Department of Defense Architecture Framework</b> .....	10
<b>UML - Unified Modeling Language</b> .....	10
<b>BPMN - Business Process Model &amp; Notations</b> .....	10
<i>Systems Engineering (nach G. Pacher)</i> .....	11
<b>System, Elemente und Beziehungen</b> .....	11
<b>Systemmodellierung</b> .....	11
<b>Systembetrachtung</b> .....	13
<i>Informationssystem-Management (ISM) (nach G. Pacher)</i> .....	14
<b>Ziele des ISM:</b> .....	14
<b>Aufgaben des ISM:</b> .....	14
<b>Die Komponenten eines Informationssystems</b> .....	14
<b>Die Lebensdauer der Elemente</b> .....	15
<b>Kategorien der Informationssysteme</b> .....	15
<b>IS-Architektur</b> .....	16
<b>IS-Projektportfolio</b> .....	16
<b>Wasserfallmodell nach Boehm</b> .....	17
<b>Waterfall Model according Winston Royce</b> .....	17
<b>Spiral Model</b> .....	18
<b>Information Engineering Methode (IEM, nach James Martin)</b> .....	19
Strategische Informationsplanung (ISP) .....	19
Analyse der Geschäftsgebiete (BAA) .....	19

Fachlicher Systementwurf (BSD) .....	19
Technisches Design (TD).....	19
<b>Prototyping .....</b>	<b>19</b>
Exploratives Prototyping.....	20
Experimentelles Prototyping .....	20
Evolutionäres Prototyping.....	20
<b>Standardsoftware .....</b>	<b>21</b>
Standardsoftwaregruppen.....	21
<i>Analyse und Design (nach E. Fuchs).....</i>	<i>22</i>
<b>Prinzipien der Analyse und des Designs.....</b>	<b>22</b>
<b>Objektorientierte Analyse (OOA).....</b>	<b>22</b>
<b>Unified Modeling Language (UML) .....</b>	<b>23</b>
<i>Systems Engineering (nach Walter F. Daenzer).....</i>	<i>24</i>
<b>Systemdenken .....</b>	<b>24</b>
Systembegriffe .....	25
Systemabgrenzung .....	25
Modell und Modellmethode .....	25
<b>Vorgehensmodell.....</b>	<b>26</b>
Lebensphasen des Systems .....	26
Problemlösungszyklus.....	30
<b>Systemgestaltung.....</b>	<b>31</b>
Situationsanalyse.....	31
Zielformulierung .....	32
Synthese & Analyse .....	33
Bewertung .....	33
Entscheidung.....	34
<b>Projektmanagement im SE.....</b>	<b>35</b>
PM-Funktional .....	35
PM-Institutionell .....	35
<i>Strukturierte Systemanalyse (nach Tom DeMarco).....</i>	<i>36</i>
<b>Grundlagen .....</b>	<b>36</b>
<b>Inkarnation eines Systems.....</b>	<b>37</b>
<b>Essenz der Systementwicklung .....</b>	<b>37</b>
Essentieller Systementwicklungsprozess .....	37
<b>Systemmodellierung.....</b>	<b>37</b>
Modelldarstellungstechnik .....	37
Die vier Grundprinzipien der Systemmodellierung .....	38
<b>Sytemzerlegung .....</b>	<b>38</b>
<i>Testen.....</i>	<i>39</i>
<b>Bottom-Up Testing.....</b>	<b>39</b>
<b>Regression Test.....</b>	<b>39</b>
<b>Smoke Test.....</b>	<b>39</b>
<b>DAST - Dynamic Application Security Testing.....</b>	<b>39</b>
<b>SAST - Static Analysis Security Testing.....</b>	<b>39</b>
<i>Darstellungstechniken .....</i>	<i>41</i>
<b>Aktivitätendiagramm.....</b>	<b>41</b>
<b>Blackbox-Betrachtung .....</b>	<b>41</b>

<b>Entscheidungstabellen .....</b>	<b>41</b>
<b>Datenflussplan (DFP).....</b>	<b>42</b>
<b>Datenstrukturdiagramm (DSD).....</b>	<b>42</b>
<b>Entscheidungsmatrix .....</b>	<b>43</b>
<b>Entity-Relationship-Modell (ERM) .....</b>	<b>43</b>
<b>Ereignisgesteuerte Prozesskette (EPK) .....</b>	<b>43</b>
<b>Flussdiagramm .....</b>	<b>43</b>
<b>Mengengerüst-Tabelle .....</b>	<b>44</b>
<b>Projektauftrag .....</b>	<b>45</b>
<b>SWOT-Analyse (Potentialanalyse) .....</b>	<b>46</b>
<b><i>DEFINITIONEN</i> .....</b>	<b>47</b>
<b>ACID - Atomicity, Consistency, Isolation, and Durability .....</b>	<b>47</b>
<b>ALM - Application Lifecycle Management.....</b>	<b>47</b>
<b>Angular .....</b>	<b>47</b>
<b>Ansible.....</b>	<b>47</b>
<b>Agile Softwareentwicklung.....</b>	<b>47</b>
<b>Apache Maven .....</b>	<b>47</b>
<b>AS3 - Application Services 3 Extension.....</b>	<b>47</b>
<b>BIG-IP Local Traffic Manager (LTM) .....</b>	<b>48</b>
<b>BIG-IP Access Policy Manager (APM) .....</b>	<b>48</b>
<b>Bitbucket.....</b>	<b>48</b>
<b>Business Analyst / Business Analysis .....</b>	<b>49</b>
<b>BPaaS - Business Process as a Service.....</b>	<b>49</b>
<b>BPM - Business Process Management.....</b>	<b>49</b>
<b>BRMS - Business Rules Management System .....</b>	<b>49</b>
<b>BI - Business Intelligence.....</b>	<b>49</b>
<b>Blockchain.....</b>	<b>49</b>
<b>CaaS - Container as a Service .....</b>	<b>50</b>
<b>CBTC - Communication Based Train Control.....</b>	<b>50</b>
<b>CD - Continuous Delivery/Deployment .....</b>	<b>51</b>
<b>CI - Continuous Integration .....</b>	<b>51</b>
<b>Cisco HyperFlex .....</b>	<b>51</b>
<b>Cisco ISE - Identity Services Engine.....</b>	<b>51</b>
<b>Cisco NSX .....</b>	<b>51</b>
<b>Citrix .....</b>	<b>51</b>
Citrix XenApp.....	52
Citrix XenDesktop .....	52
Citrix Netscaler .....	52
<b>CMM - Capability Maturity Model.....</b>	<b>52</b>
<b>CNI - Container Network Interface.....</b>	<b>54</b>
<b>CobiT.....</b>	<b>54</b>

<b>CONTAINERS</b> .....	<b>54</b>
<b>CORBA - Common Object Request Broker Architecture</b> .....	<b>55</b>
<b>Coreo Flannel</b> .....	<b>55</b>
<b>COSO - Committee of Sponsoring Organizations of the Treadway</b> .....	<b>56</b>
<b>COTS - Commercial of the Shelf</b> .....	<b>56</b>
<b>Commission</b> .....	<b>56</b>
<b>DMS - Dokumentenmanagement</b> .....	<b>56</b>
<b>Docker - Container</b> .....	<b>56</b>
<b>Drools -</b> .....	<b>57</b>
<b>DTU - Database Transaction Unit</b> .....	<b>57</b>
<b>ECM - Enterprise-Content-Management</b> .....	<b>57</b>
<b>Enterprise Collaboration</b> .....	<b>57</b>
Microsoft Teams .....	57
<b>EOS - End of Support</b> .....	<b>57</b>
<b>Foreman</b> .....	<b>58</b>
<b>Funktion</b> .....	<b>58</b>
<b>Galvanische Trennung</b> .....	<b>58</b>
<b>GitHub</b> .....	<b>58</b>
<b>GitLab</b> .....	<b>58</b>
<b>GMC - Grandmaster Clock</b> .....	<b>59</b>
<b>Grafana</b> .....	<b>59</b>
<b>Grid Computing</b> .....	<b>59</b>
<b>IaaS - Infrastructure as-a-Service</b> .....	<b>59</b>
<b>IACS - Industrial Automation and Control Systems</b> .....	<b>59</b>
<b>IDaaS - Identity-as-a-Service</b> .....	<b>60</b>
<b>IPaaS - Integration Platform as a Service</b> .....	<b>60</b>
<b>IPAM - IP Address Management</b> .....	<b>60</b>
VitalQIP .....	60
BT Diamond IP .....	60
Infoblox .....	60
BlueCat .....	60
Men&Mice .....	60
SolarWinds IP Address Manager .....	60
LightMesh .....	60
<b>IDEAL Model</b> .....	<b>60</b>
<b>IRC - Internet Relay Chat</b> .....	<b>61</b>
<b>ITSM - IT Service Management</b> .....	<b>61</b>
BMC Remedy Service Management Suite.....	61
Cherwell Service Management .....	62
Freshservice .....	62
OTRS .....	62
KIX .....	62
Servicenow .....	62
<b>ITIL - IT Infrastructure Library</b> .....	<b>62</b>
<b>JBoss EAP - JBoss Enterprise Application Platform</b> .....	<b>63</b>

<b>Jenkins</b> .....	<b>63</b>
<b>KUBERNETES</b> .....	<b>63</b>
<b>MDM - Mobile Device Management</b> .....	<b>63</b>
Microsoft Intune.....	63
<b>Microservices</b> .....	<b>63</b>
<b>MTBF - Mean Time Between Failures</b> .....	<b>64</b>
<b>MTTF - Mean Time to Failure</b> .....	<b>64</b>
<b>MTTR - Mean Time to Repair</b> .....	<b>64</b>
<b>MTR - Mean Time to Recovery</b> .....	<b>64</b>
<b>Nexus</b> .....	<b>64</b>
<b>Nginx</b> .....	<b>65</b>
<b>NMock2</b> .....	<b>65</b>
<b>NVMe</b> .....	<b>65</b>
<b>Octopus Deploy</b> .....	<b>65</b>
<b>OODB - Object-Oriented Database</b> .....	<b>66</b>
<b>OpenShift - Container</b> .....	<b>66</b>
<b>OVF - Open Virtualization Format</b> .....	<b>66</b>
<b>PaaS - Platform-as-a-Service</b> .....	<b>66</b>
<b>PLM - Product Lifecycle Management</b> .....	<b>66</b>
<b>PSP - Personal Software Prozess</b> .....	<b>66</b>
<b>Puppet</b> .....	<b>67</b>
Declarative .....	67
Imperative .....	68
<b>RAD - Rapid Application Development</b> .....	<b>68</b>
<b>RFB - Request for Bid</b> .....	<b>68</b>
<b>RFI - Request for Information</b> .....	<b>68</b>
<b>RFP - Request for Proposal</b> .....	<b>68</b>
<b>RFQ - Request for Quote</b> .....	<b>68</b>
<b>RUP - Rational Unified Process</b> .....	<b>68</b>
<b>SaaS - Software-as-a-Service</b> .....	<b>68</b>
<b>Satellite - Red Hat Satellite</b> .....	<b>68</b>
<b>SCCM - System Center Configuration Manager</b> .....	<b>69</b>
<b>SCOM - System Center Operations Manager</b> .....	<b>69</b>
<b>SCVMM - System Center Virtual Machine Manager</b> .....	<b>69</b>
<b>SDLC - System Development Life Cycle</b> .....	<b>69</b>
Phase: System Investigation.....	69
Phase: System Analysis.....	69
Phase: Design.....	69
Phase: Environments.....	70
Phase: Testing .....	70
Phase: Training and transition.....	70
Phase: Operations and Maintenance.....	70
Phase: Evaluation.....	70
<b>SERVERLESS COMPUTING</b> .....	<b>70</b>

<b>Sharding</b> .....	<b>70</b>
<b>Slack</b> .....	<b>70</b>
<b>SOA - Serviceorientierte Architektur</b> .....	<b>70</b>
<b>SPICE - Software Process Improvement and Capability Determination</b> .....	<b>71</b>
<b>System-Architect</b> .....	<b>71</b>
<b>TDD</b> .....	<b>71</b>
<b>Terraform</b> .....	<b>71</b>
<b>TFS - Team Foundation Server</b> .....	<b>71</b>
<b>VDI - Virtualized Desktop Infrastructure</b> .....	<b>71</b>
<b>Virtual Kubelet</b> .....	<b>71</b>
<b>VMWare ESXi</b> .....	<b>72</b>
<b>UML - Unified Modeling Language</b> .....	<b>72</b>
<b>XP - eXtreme Programming</b> .....	<b>73</b>
<b>YAML - Ain't Markup Language</b> .....	<b>73</b>
<i>TOOLS</i> .....	<i>74</i>
<i>Abbildungsverzeichnis</i> .....	<i>75</i>
<i>Index</i> .....	<i>76</i>

## Einführung

Literatur: Systems Engineering

(Walter F. Daenzer, )

System-Entwicklung in der Wirtschaftsinformatik

(R. Böhm E. Fuchs, G.Pacher)

Strukturierte Systemanalyse

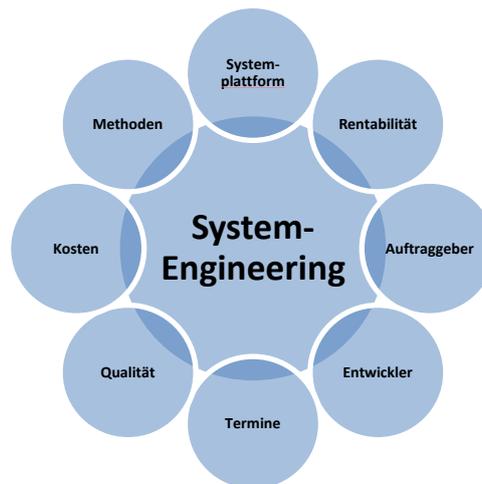
(Stephen M. McMenamin, John F. Palmer, Tom DeMarco)

Structured Analysis and Design Technique "SADT"

(Douglas T. Ross, Kenneth E. Schoman)

Methode und Techniken der Organisation

(Götz Schmid)



## EA - ENTERPRISE ARCHITECTURE

- Enterprise architecture supports planning and decision-making through documentation and information that provides an abstracted view of an enterprise at various levels of scope and detail.
- Enterprise architecture provides a framework to define business requirements, business rules, and business case that will lead the organization to accomplish its mission, vision, and purpose successfully.
- The purpose of enterprise architecture is to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy.
- This is valuable for all organizations, but is critical in large, complex environments.

A great architecture starts with a solid foundation built on four pillars:

- Security
- Performance and scalability
- Availability and recoverability
- Efficiency and operations

## EA - FRAMEWORKS

### TOGAF

- Das **The Open Group Architecture Framework (TOGAF)** bietet einen Ansatz für Entwurf, Planung, Implementierung und Wartung von Unternehmensarchitekturen.
- TOGAF is a business-centric enterprise architecture framework that provides flexibility for formal definition of business activities.

- TOGAF provides a structured approach for organizations seeking to organize and manage the implementation of technology in the enterprise.
- However, organizations and activities within groups can focus on specific subsets of the framework to accomplish their goals.
- The flexibility of TOGAF makes this framework effective for activities that range **from small development projects to comprehensive enterprise resource planning (ERP) systems**.

PART I: Introduction  
 PART II: Architecture Development Method  
 PART III: ADM Guidelines and Techniques  
 PART IV: Architecture Content Framework  
 PART V: Enterprise Continuum & Tools  
 PART VI: TOGAF Reference Models  
 PART VII: Architecture Capability Framework

## Zachman Framework

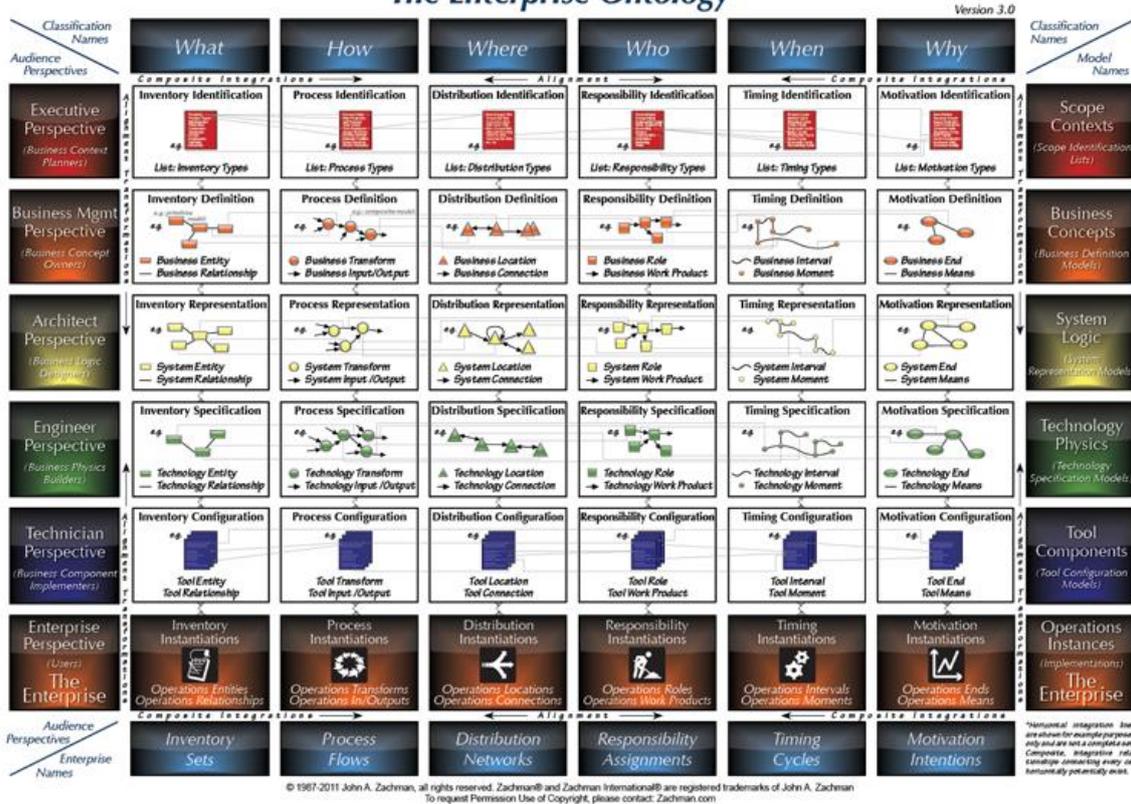
- The Zachman Framework is an ontology providing a classification system that produces a “working model of entities and interactions that create an agreed-upon vocabulary for exchanging information when specified in a particular way”.
- Because of the complexity and the level of detail, the Zachman Framework is a tool best suited for **large enterprise environments**.
- Many versions of enterprise architecture have roots in the efforts completed by Zachman to develop the Zachman Framework.
- Using the matrix in the Zachman Framework to answer the questions of what, how, where, who, when, and why, as they relate to key attributes of the enterprise, an organization can formalize a definition of the enterprise architecture to support governance activities related to this detailed enterprise classification.
- The Zachman Framework typically is depicted as a bounded 6 x 6 “matrix” with the Communication Interrogatives as Columns and the Reification Transformations as Rows.
- The Framework classifications are represented by the Cells, that is, the intersection between the Interrogatives and the Transformations.
- This matrix would necessarily constitute the total set of descriptive representations that are relevant for describing something...anything: in particular, an enterprise.

	DATEN <i>Was</i>	FUNKTION <i>Wie</i>	NETZWERK <i>Wo</i>	PERSONEN <i>Wer</i>	ZEIT <i>Wann</i>	MOTIVATION <i>Warum</i>
<b>Zielsetzung/Bereich</b> <i>(Kontextabhängig)</i> → Rolle: <i>Planer</i>	Liste von wichtigen Faktoren im Geschäft	Liste von Kernprozessen	Liste von Geschäftsstellen	Liste von wichtigen Organisationen	Liste von Ereignissen	Liste von Geschäftszielen/Strategien
<b>Unternehmensmodell</b> <i>(Konzeptionell)</i> → Rolle: <i>Besitzer</i>	Konzeptionell Datenmodell/ Objektmodell	Geschäftsprozessmodell	Geschäftslogistiksystem	Arbeitsablaufmodell	Ablaufplan	Geschäftsplan
<b>Systemmodell</b> <i>(Logisch)</i> → Rolle: <i>Designer</i>	Logisches Datenmodell	Systemarchitekturmodell	Distributed Systems Architecture	Human-Interface-Architektur	Prozessstruktur	Geschäftsregelmodell
<b>Technologiemodell</b> <i>(Physisch)</i> → Rolle: <i>Builder</i>	Physische Daten/ Klassenmodell	Technologie-designmodell	Technologiearchitektur	Darstellungsarchitektur	Kontrollstruktur	Regeldesign
<b>Detaillierte Darstellung</b> <i>(Aus dem Kontext heraus)</i> → Rolle: <i>Programmierer</i>	Datendefinitionen	Programm	Netzwerkarchitektur	Sicherheitsarchitektur	Zeitplan	Regelspezifizierung
<b>Unternehmen</b> → Rolle: <i>Nutzer</i>	Nutzbare Daten	Anwendungszweck	Nutzbare Netzwerk	Arbeitsorganisation	Eingeschlossener Zeitplan	Arbeitsweise

Source: Wikipedia

# The Zachman Framework for Enterprise Architecture™

## The Enterprise Ontology™

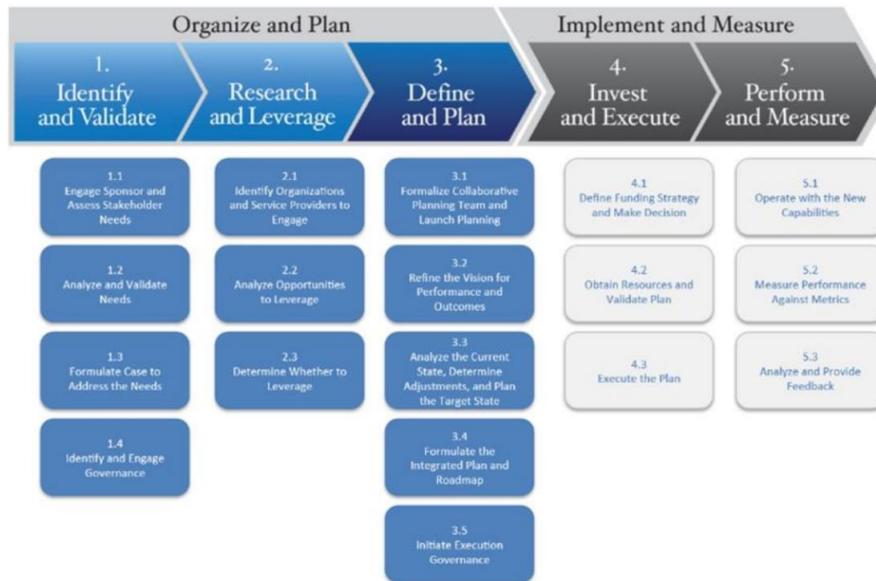


## SABSA - Sherwood Applied Business Security Architecture

- SABSA, the Sherwood Applied Business Security Architecture, is a framework and methodology used to meet a variety of enterprise needs:
  - risk management
  - information assurance
  - governance
  - continuity management
- SABSA is valuable because the framework ensures that security services are designed, delivered, and supported as an integral part of your business and IT management infrastructure.
- The goal is that the security architecture is fully integrated in the enterprise architecture.

## FEAF - Federal Enterprise Architecture Framework

- The Federal Enterprise Architecture Framework (FEAF) is a set of tools that help U.S. federal agencies define the structure of enterprise architecture within each organization.
- It helps the agencies describe the current enterprise in appropriate detail and identify how the enterprise should look in the future so each organization can make plans to transition from the current state to the future state.
- The FEAF also provides a mechanism for agencies to map the architecture to the strategic plan defined by each organization.



### CRM - Consolidated Reference Model

- Part of FEAF

### DoDAF - Department of Defense Architecture Framework

- The complexity of the Department of Defense Architecture Framework (DoDAF) highlights the importance of selecting an enterprise architecture framework that is appropriate for the needs of the organization.
- DoDAF and similar frameworks, like the U.K. Ministry of Defense Architecture Framework (MoDAF), define requirements that support activities and concerns related specifically to military enterprises--in particular, information sharing across defense organizations.

### UML - Unified Modeling Language

- Tool
- UML 2.0 defines thirteen types of diagrams, divided into three categories.
- Six diagram types represent static application structure.
- three represent general types of behavior
- four represent different aspects of interactions

### BPMN - Business Process Model & Notations

- A standard Business Process Model and Notation (BPMN) will provide businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner. Furthermore, the graphical notation will facilitate the understanding of the performance collaborations and business transactions between the organizations.
- This will ensure that businesses understand themselves and participants in their business, as well as enable organizations to adjust to new internal and B2B business circumstances quickly.

## Systems Engineering (nach G. Pacher)

Das **Systemdenken** fördert das Denken in geordneter Form.

Das **Systemdenken** zur Problemlösung hilft das System Top-Down systematisch in Teilprobleme zu zerlegen die noch immer Teil des Ganzen sind. Damit lassen sich Teillösungen erarbeiten, die zur Gesamtlösung des Problems führen.

Das **systematische Denken** beinhaltet deshalb sowohl analysierendes wie auch integriertes Denken.

**Probleme** werden dadurch mittelbar und beurteilbar.

## System, Elemente und Beziehungen

Jene Komponenten/Teile eines Systems, welche nicht mehr aufgelöst resp. weiter unterteilt werden, sind die **Elemente** eines Systems.

Die **Wechselbeziehungen** (Relationen) zwischen den Elementen eines Systems oder zwischen Systemen repräsentieren **Strömungsgrößen** (Wirkungspotentiale).

*Strömungsarten:*

- Materialfluss
- Informationsfluss
- Energiefluss

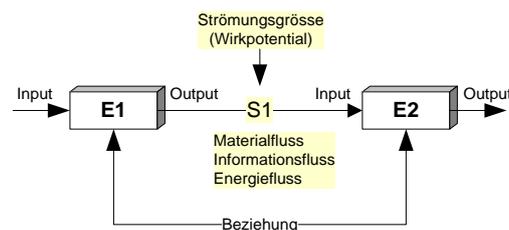


Abbildung 1: Strömungsgrößen

## Systemmodellierung

**Systemmodell** entwickeln unter Berücksichtigung der wirkungsbezogenen Betrachtung. (Blackbox-Betrachtung)

Der **Abstraktionsgrad** ist dabei so zu wählen, dass das konzeptionelle Modell für einen grossen Kreis von Personen eine gute Basis für die Kommunikation schafft.

Durch bewusstes **Wechseln der Betrachtungsebene** können wir ein System analysieren oder in ein grösseres Ganzes integrieren.

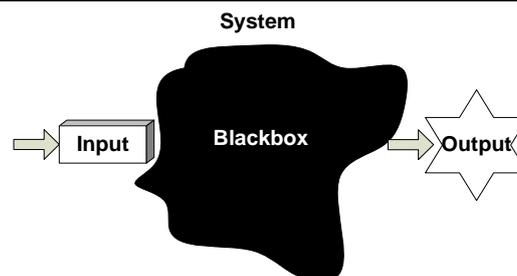


Abbildung 2: Blackbox-Betrachtung

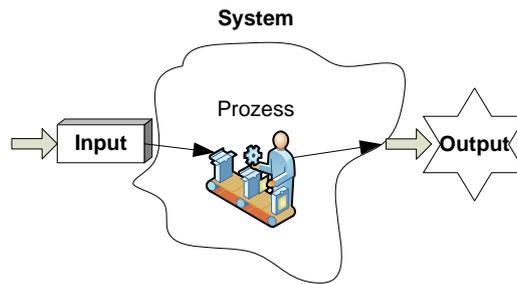


Abbildung 3: Prozessanalyse

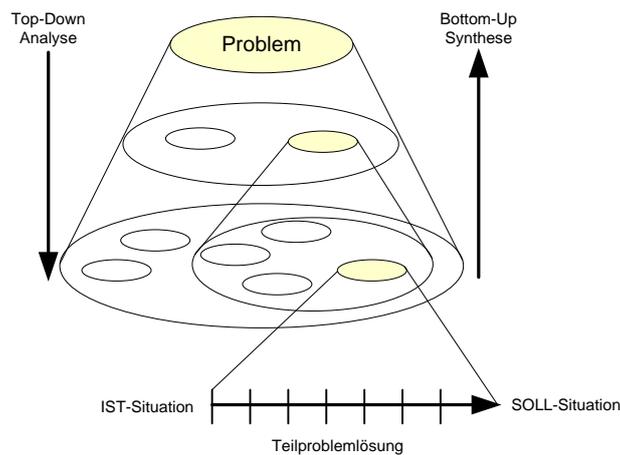


Abbildung 4: Teilproblemlösung

**Modelle** sind reale oder irrealen Systeme, die andere Systeme so darstellen, dass eine experimentelle Manipulation der abgebildeten Strukturen und Zustände möglich wird.

**Zitat: P. Valéry**

"Die einfachen Modelle sind falsch; die anderen unbrauchbar."

- Landschaftsmodell
- Gebäudemodell
- Flugzeugmodell, Schiffsmodell, Fahrzeugmodell



Abbildung 5: Zinnmodelle

**Das Prinzip der inneren Bindung.**

Die Anzahl der Beziehungen der Elemente im System gegenüber den Beziehungen der Elemente zwischen System und Aussen soll grösser sein.

**Geschlossene Systeme** ohne Interaktion zu anderen Systemen gibt es nur in der Theorie.

Ein **Problem** ist die Diskrepanz zwischen der vorhandenen sowie feststellbaren IST-Situation und der SOLL-Situation.

## Systembetrachtung

## Informationssystem-Management (ISM) (nach G. Pacher)

Die 4 Produktionsfaktoren der Volkswirtschaftslehre:

- Kapital
- Boden
- Arbeit
- Informationen

**Die Potentiale** der Ressourcen "Information und Informationstechnologie" zu erkennen und diese in unternehmerische Lösungen umzusetzen, bildet im weitesten Sinne den Verantwortungsbereich des Informationssystem-Managements.

In allen Unternehmen existiert eine **Diskrepanz** zwischen dem benötigten und dem verfügbaren Informationssystem. Es ist Aufgabe des **ISM** diese Diskrepanz zwischen **IST- und SOLL-Vorstellung** klein zu halten.

### Ziele des ISM:

- Information zur Richtigen Zeit am richtigen Ort
- Informationsverarbeitung zur Steuerung von Wirtschaftlichen Nutzung
- Einbindung Informationsverarbeitung und Unternehmensführung
- Unternehmensunterstützung zur strategischen Erfolgsposition (SEP)

### Aufgaben des ISM:

- Architekturplanung
- Integration
- Einbindung in die Unternehmensführung
- Einbindung des Fachbereiches
- Zentralisierung / Dezentralisierung
- Verbindung von Organisation und Informationssystem
- Projektportfoliomanagement
- Change Management
- Umsetzung

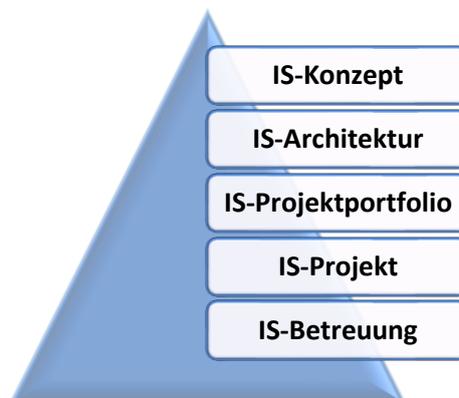


Abbildung 6: Die Stufen des ISM

## Die Komponenten eines Informationssystems

- Daten

- Funktionen  
Betriebswirtschaftliche Funktionen / IT-Funktionen
- Organisatorische Einheiten
- Technische Geräte

Im Unterschied zu den **Daten**, die auch ohne IT-Funktionen einen Sinn ergeben oder reale Zustände beschreiben, sind **IT-Funktionen** sehr stark von den Daten abhängig.

**IT-Funktionen ohne Daten ergeben betriebswirtschaftlich keinen Sinn!**

**Funktionsorientierte Entwicklungen** führen zu Insellösungen.

Die **IS-Architektur** beschreibt auf konzeptioneller Ebene den Aufbau der gesamten Kommunikationsinfrastruktur eines Unternehmens.

## Die Lebensdauer der Elemente



Abbildung 7: Lebensdauer der Elemente

## Kategorien der Informationssysteme

- Administrationssysteme
- Dispositionssysteme
- Informationssysteme
- Planungssysteme

Als **Geschäftsfelder** kommen solche Bereiche der Unternehmung in Frage, die einen bestimmten Produkt-/Marktbereich bearbeiten und für die es sinnvoll erscheint, eigenständige, von anderen Bereichen der Unternehmung unabhängige Strategien zu formulieren und durchzusetzen.

## IS-Architektur



Abbildung 8: IS-Architektur

Pro Geschäftsfeld ist eine IS-Architektur zu bilden. Die Praxis hat gezeigt, dass das "Mamut-Vorhaben" eine IS-Architektur für das Gesamtunternehmen selten erfolgreich ist. Deshalb wird die IS-Architektur auf Stufe Untersystem gebildet. Es ist Aufgabe des ISM dafür zu sorgen, dass der passende **Integrationsbereich** gefunden wird.

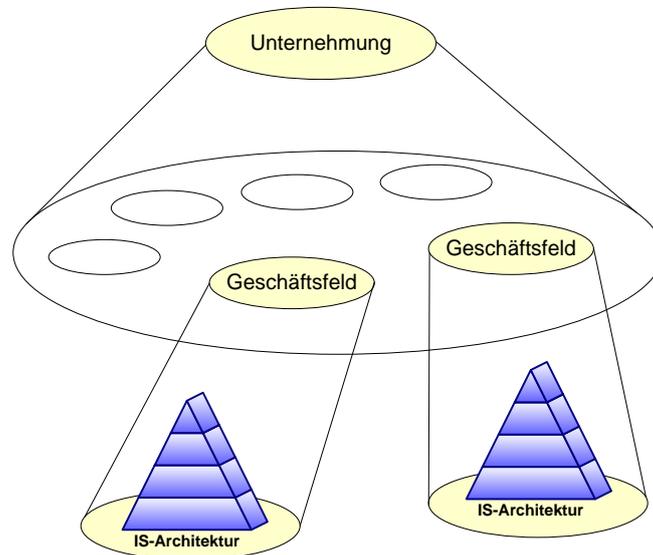


Abbildung 9: Geschäftsfeld/IS-Architektur

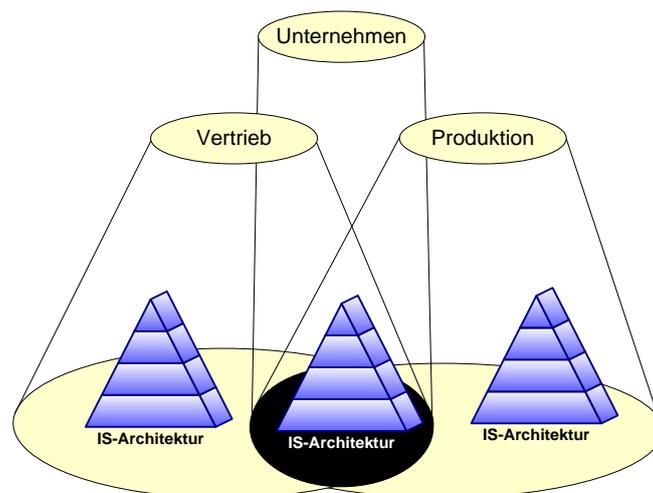


Abbildung 10: IS-Architektur des Integrationsbereiches

## IS-Projektportfolio

- **Welche** Projekte sind durchzuführen

- **Wann** mit welchen Projekten begonnen werden kann
- **Wann** ein Projekt abgeschlossen ist
- **Welche** Projekte parallel zu anderen Projekten realisiert werden können
- In **welcher** Reihenfolge realisiert wird
- **Wie** lange das gesamte Realisierungsvorhaben dauert
- **Wann** welche Investitionen getätigt werden müssen
- **Wie** hoch der jährliche Realisierungsaufwand in Wochen ist

## Wasserfallmodell nach Boehm

(Waterfall Model)

Flussorientierter Ablauf.

**Besonderheit:** Die Lösungssuche wird sofort abgebrochen, sobald eine funktionstüchtige Lösung gefunden wurde. Deshalb wird diese Strategie auch "**Nicht optimierte Suchstrategie**" genannt.

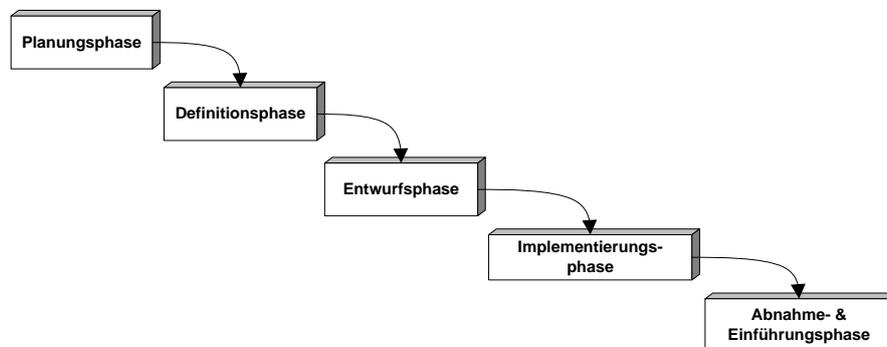


Abbildung 11: Wasserfallmodell

## Waterfall Model according Winston Royce

- Originally developed by **Winston Royce** in 1970, the waterfall model seeks to view the systems development life cycle as a series of iterative activities.
- The traditional waterfall model has **seven stages**.
- As each stage is completed, the project moves into the next phase.

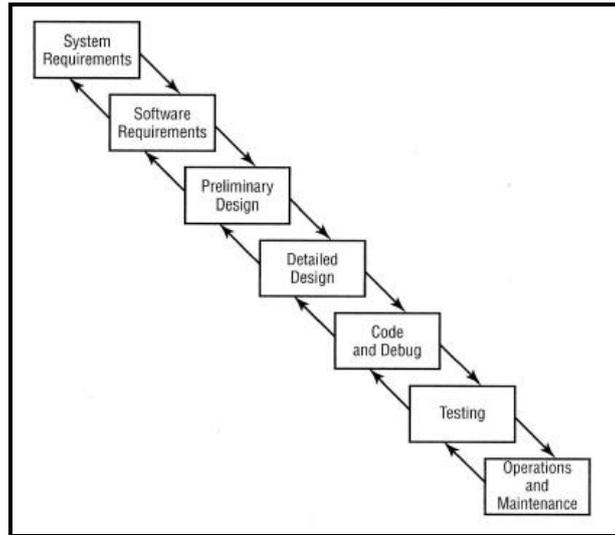


Figure 1: Waterfall life cycle model

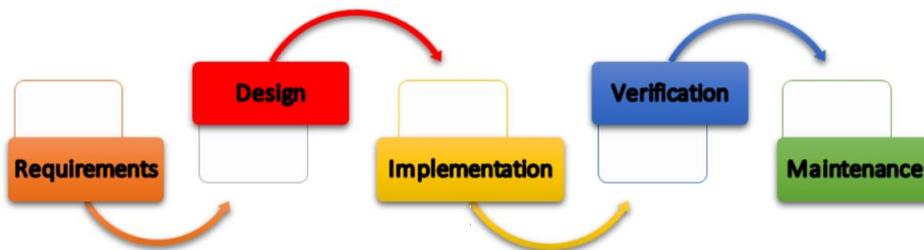


Figure 2: Waterfall Model

## Spiral Model

- Proposed by **Barry Boehm** in 1988.
- **Software development model**, that incorporates a number of the software development models.
- The **spiral model** encapsulates a number of iterations of another model (the waterfall model), it is known as a **metamodel**, or a **"model of models"**.
- Boehm's spiral model provides a solution to the major criticism of the waterfall model.

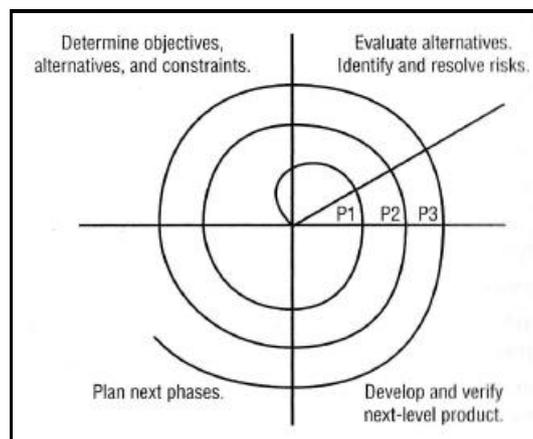


Figure 3: Spiral life cycle model

## Information Engineering Methode (IEM, nach James Martin)

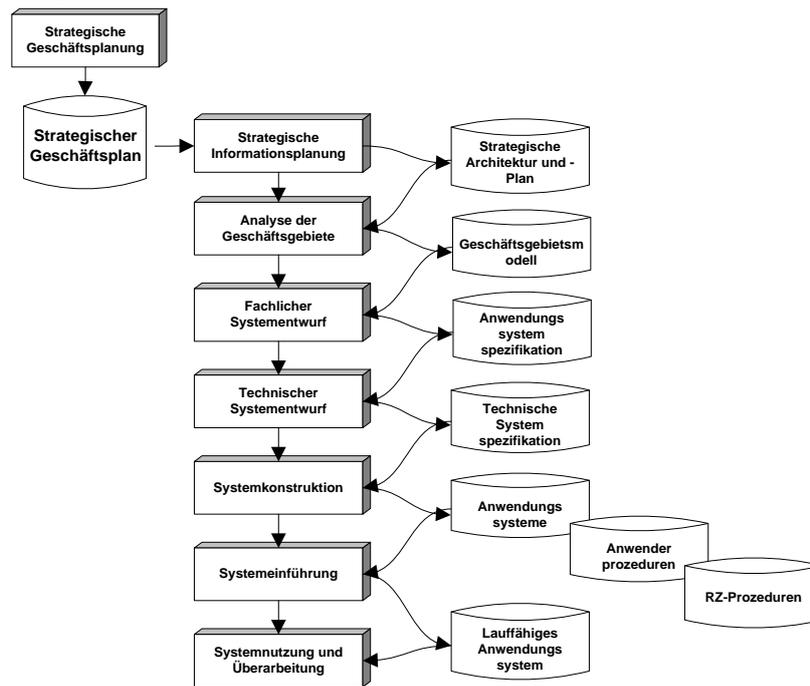


Abbildung 12: Phasen des IEM

### Strategische Informationsplanung (ISP)

### Analyse der Geschäftsgebiete (BAA)

### Fachlicher Systementwurf (BSD)

### Technisches Design (TD)

### Prototyping

Das Prototyping ist eine mögliche Methode zur Systementwicklung, bei der so früh wie möglich eine erste vereinfachte Version (Prototyp) der angestrebten Anwendung realisiert wird. Dieser Prototyp wird als Kommunikationsgegenstand für den Interessenausgleich zwischen Auftraggeber und Entwickler verwendet. Damit werden **Rückkopplungen** erreicht, die zu einer zyklischen **Systementwicklung** führen.

Ein solches Vorgehen ist attraktiv, weil die Erfahrung zeigt, dass die Anforderungen an eine Informatiklösung nur in den seltensten Fällen ein und für allemal festgelegt werden können. Damit können Erfahrungen gesammelt werden. Den die anfallenden Entwicklungsdokumentationen beim konventionellen Phasenmodell sind sehr stark entwicklungsorientiert und bilden keine geeignete Grundlage für die Kommunikation zwischen Entwicklern und Anwendern.

**Prototypen** sind zu unterscheiden von **Modellen**. Ein Modell enthält nicht die Leistungsmerkmale des fertigen Produkts, es bildet sie nur ab.

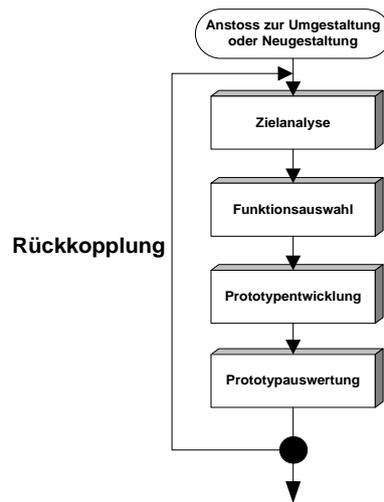


Abbildung 13: Prototypingmodell

Es wird unterschieden zwischen, **vertikalem**, **horizontalem** und **diagonalem** Prototyping.

Ist die **Qualität** der zur Verfügung gestellten Prototypen zu hoch, verführt dies die Anwender leicht dazu, den Prototypen bereits als fertiges Anwendungssystem zu sehen. Der Erfolg eines Prototyping-Projektes hängt stark von der **Kommunikationsfähigkeit** der Projektbeteiligten ab.

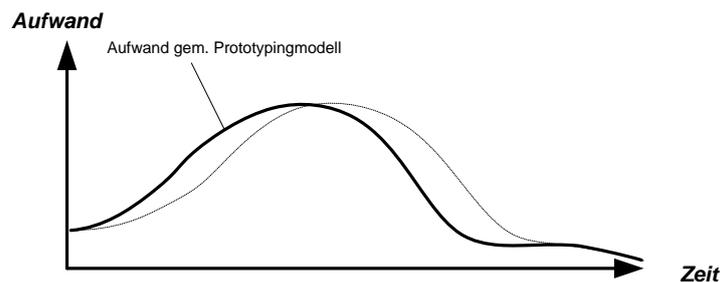


Abbildung 14: Verlagerung des Aufwandes

### Exploratives Prototyping

Primäre Zielsetzung beim explorativen Prototyping ist die rasche Klärung fachlicher Anforderungen. Wird auch **Rapid Specification Prototyping** genannt. Es werden verschiedene "Throw-away-Prototypen" erstellt.

### Experimentelles Prototyping

Die Zielsetzung des experimentellen Prototyping ist der Nachweis der Tauglichkeit vorgeschlagener Problemlösungen, bevor in eine umfangreiche Implementation investiert wird. Das experimentelle Prototyping baut immer auf einer bereits vorhandenen Anforderungsspezifikation oder einem Software-Entwurf auf.

### Evolutionäres Prototyping

Verschiedene Entwurfszyklen werden sukzessive Durchlaufen und das System wird inkrementell vom explorativen, sehr simplen Prototyp bis zum fertigen Produkt entwickelt.

## Standardsoftware

### Standardsoftwaregruppen

- Applikationen/Standardanwendungssoftware
- Betriebssoftware
- Tools

Die Anforderungen an eine Standardsoftware werden nach **Muss-** und **Soll-Anforderungen** unterteilt.

Werden die Eigenschaften der Standardsoftware akzeptiert, so sind **aufbau-** und **ablauforganisatorische Anpassungen** vorzunehmen.

#### Vorteile:

- Wirtschaftlichkeit
- Ergonomie und Dokumentation
- Zeitvorteil
- Personalressourcen
- Standard Ablauforganisation (positiv)

#### Nachteile:

- Anpassungsprobleme
- Abhängigkeit
- Schnittstellenprobleme
- Standard Ablauforganisation (negativ)
- Qualitätsanforderungen
- Unternehmens IS-Architektur

## Analyse und Design (nach E. Fuchs)

**Analyse** bedeutet die systematische Untersuchung eines Systems hinsichtlich aller Komponenten und der Beziehungen zwischen diesen Komponenten. Sie beginnt mit der Projektauftragserteilung und endet mit der Detailstudie.

- Aus welchen Teilen besteht das System?
- Was gehört alles dazu?
- Was gehört nicht dazu?
- Wie funktioniert es heute?
- Wie muss es in Zukunft angelegt sein?
- Welchen Anforderungen muss es genügen?

Unter **Design** verstehen wir allgemein einen zeichnerischen oder plastischen Entwurf, angefangen von einer Skizze bis hin zu einem kompletten Konzept oder Modell.

Bei Informationssystemen beinhaltet das unter anderem:

- Verarbeitungsformen
- Gestaltung der Schnittstellen
- Definition der Transaktionen

### **Risiko der Analyse und des Designs**

Durch die rasante Entwicklung in der Geschäftswelt besteht die Gefahr, dass bevor Analyse und Design beendet sind, die Konzeption bereits nicht mehr mit der realen Welt übereinstimmt.

### Empfohlene Reihenfolge der Betrachtung

1. Daten-Orientierte Sicht
2. Funktionale Sicht
3. Zeit-Orientierte Sicht

## Prinzipien der Analyse und des Designs

- Top-Down-Prinzip
- Bottom-Up-Prinzip
- Prinzip der minimalen Modelle
- Prinzip der Strukturierung
- Prinzip der Verständlichkeit
- Prinzip der Neutralität
- Prinzip der 3-Sichten  
Funktionen, Daten, Zeit

## Objektorientierte Analyse (OOA)

### Hauptaktivitäten:

- Auffinden der Klasse und Objekte
- Identifizierung von Strukturen
- Identifizierung von Subjekten
- Definieren der Attribute
- Festlegen der Services (Methoden)

### OOA-Sichten:

- Subject-Layer
- Class- & -Object layer
- Structure layer
- Attribute layer
- Service layer

## **Unified Modeling Language (UML)**

Zur Notation von OOA dient vorzugsweise UML

## Systems Engineering (nach Walter F. Daenzer)

**Systems Engineering (SE)** ist eine auf bestimmten Denkmodellen und Grundprinzipien beruhende Methodik zur zweckmässigen und zielgerichteten Gestaltung komplexer Systeme.

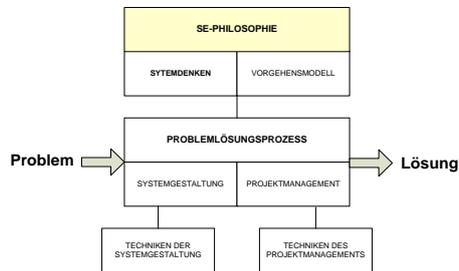


Abbildung 15: SE-Philosophie

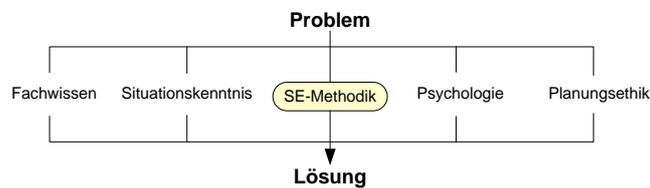


Abbildung 16: SE-Methodik



Abbildung 17: Problem

## Systemdenken

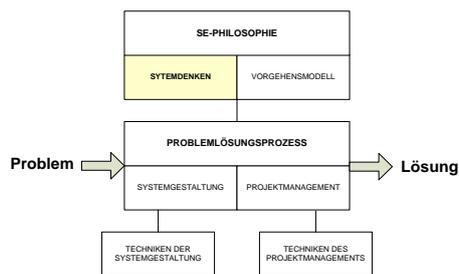


Abbildung 18: Systemdenken

Als **System** wird die Gesamtheit von Elementen verstanden, die miteinander durch Beziehungen verbunden ist.

Das **Systemdenken** stellt eine stufenweise Betrachtungsart dar, die eine Betrachtung vergleichbar mit einem "Zoom-Objektiv" ermöglicht.

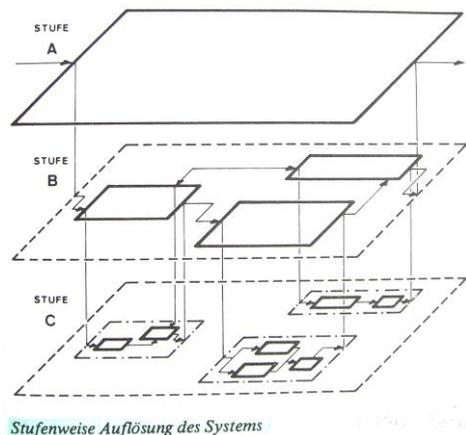


Abbildung 19: System - Zoom

## Systembegriffe

Beispiel: **System** → Europa  
**Untersysteme** → Schweiz, Deutschland, Frankreich, ...  
**Teilsysteme** → Energieverbund, Handelsverkehr, Post, ...

Beispiel: **System** → Menschlicher Körper  
**Untersysteme** → Kopf, Arme, Beine, Gehirn  
**Teilsysteme** → Nervensystem, Blutkreislauf, Knochengestüt

## Systemabgrenzung

Elemente, welche ausserhalb des Systems liegen, aber zum System selbst Beziehungen aufweisen.

Ein wichtiges Instrument im Zusammenhang mit der Systembetrachtung sind **Ablaufdiagramme**.

- Lineare Ablaufdiagramme
- Datenflussdiagramme (DFD)
- Nassi-Shneiderman
- Jackson Diagramm
- Programmablaufplan (PAP)

## Modell und Modellmethode

Ein **Modell** dient dazu, die Realität abzubilden. Es wird damit vor allem die Absicht verfolgt, Erkenntnisse über die bestehende Realität zu gewinnen oder die künftige gewollte Realität darzustellen.

Ein **Modell** kann leichter analysiert und diskutiert werden.

Ein **Modell** stellt eine Analogie zwischen realem Objekt und Modell dar.

Die **wirkungsbezogene- / Blackbox-Betrachtung** des Systems interessiert nicht wie das Ergebnis zustande kommt, sondern lediglich die Wirkung.

## Vorgehensmodell

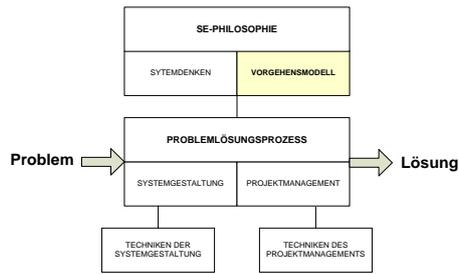


Abbildung 20: Vorgehensmodell

**Grundsatz** Vom Groben zum Detail!

Mit den Methoden des **SE** sollen Probleme gelöst werden, die in sich komplex sind und eine starke Verflechtung mit der Umwelt aufweisen und nicht solche, bei denen die gegenseitige Beeinflussung offensichtlich ist und daher keiner näheren Untersuchung und Überlegungen mehr benötigen.

## Lebensphasen des Systems

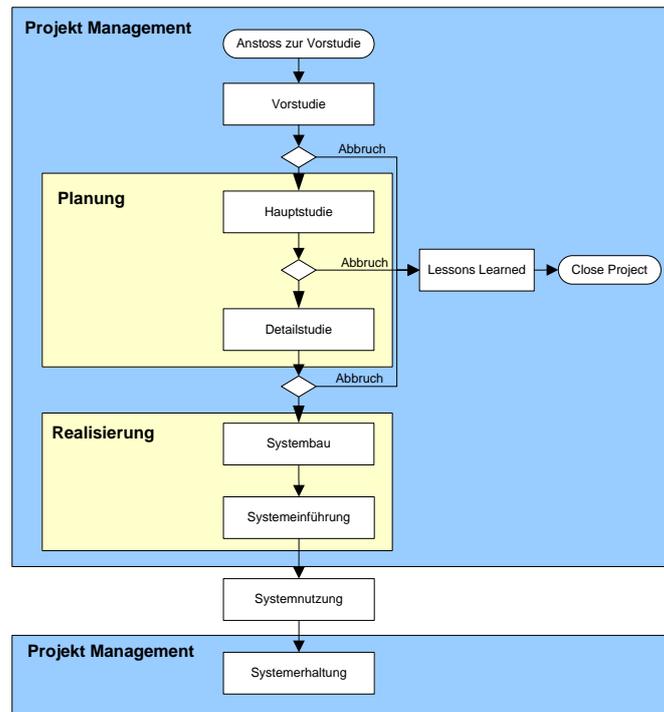


Abbildung 21: Lebensphasen eines Systems

### Anstoss zur Vorstudie

Am Anfang des Projektes steht die **Idee**.

## Vorstudie

Input: Projektauftrag

- Wird das richtige Problem angegangen.
- Besteht ein Bedürfnis nach einem neuen System.
- Das System wird abgegrenzt.
- Anforderungen definieren.
- Vergleich der Lösungsvarianten
- Identifikation und Klassifizierung der Stakeholder (Evtl. Delphi-Methode anwenden)
- Die zu erwartenden Kosten und Zeitplan um das Konzept zu realisieren.
- Wirtschaftlichkeit: Kosten-Nutzen

Output: Lösungsvarianten → Antrag für weiteres Vorgehen

Aktivitäten Vorstudie	Ergebnis	Darstellungsmethode
Analyse	Projektauftrag (ergänzt, angepasst)	Netzplan, Balkendiagramm
Projektauftrag		
Phasenplanung	Arbeitsplan (Mitarbeiter, Termine)	
Situationsanalyse durchführen	Systemübersicht mit Systemgrenzen	Bubble Chart
	Ist-System Datenfluss	DFD
	Mengengerüst	Mengengerüst-Tabelle
	Schwachstellen-Katalog	Tabelle
Ziele definieren	Zielstruktur, operationelle Ziele, Zielkonflikte, Zielpräferenzen, Rahmenbedingungen, Qualitätsmerkmale	Formulare, Texte
Lösungsvarianten entwickeln	Pro Variante grob: Datenmodell	ERM
	Geschäftsfunktionsmodell	GFHD, DFD
	Organisations-Modell	Organigramm, GFVM
	Konfigurations-Modell	Konfigurationsschema
	Realisierungsplan	
Kosten-/Nutzen-Analyse	Pro Variante: - Investitionskosten - Betriebskosten	
Lösungsvarianten analysieren	Zielerfüllung, Risikoanalyse, Konsequenzen	
Lösungsvarianten bewerten	Bewertungskriterien, Erfüllungsgrad pro Variante	
Antrag für weiteres Vorgehen	Detailplan der nächsten Phase, Antrag für die nächste Phase	Netzplan, Balkendiagramm

## Hauptstudie

Es sollte nur noch der erfolgversprechendste Lösungsweg der Vorstudie untersucht werden.

- Investitionsentscheidungen ermöglichen
- Detaillierte Ermittlung der Anforderungen
- Definition der Teilprojekte (Groblösungen) ermöglichen

Output: Gesamtkonzept entwickeln, dass es ermöglicht, die weitere Entwicklung und Realisierung in einen geordneten Rahmen zu stellen.

Aktivitäten Hauptstudie	Ergebnis	Darstellungsmethode
Analyse Projektauftrag Phasenplanung	Arbeitsplan (Mitarbeiter, Termine)	Netzplan, Balkendiagramm
Datenanalyse durchführen Funktionsanalyse durchführen	Datenmodell verfeinert Funktionsmodell verfeinert Datenmodell überprüfen, verfeinern	ERM GFHD, GFAD, DFD, DSET, GFDSM
Geschäftsereignis-Analyse Entitätstypen-Lebenszyklen definieren	Funktionsmodell verfeinert Datenmodell	GEB, GEA, GFA EZM, ELH
Anforderungen an Datensicherheit/Datenschutz festlegen	Anforderungskatalog	Texte
Organisationsmodell erstellen Applikationen und Datenbanken identifizieren und priorisieren	Organisations-Modell Design Areas	Organigramm, GFVM
Schnittstellen definieren Kosten-/Nutzen-Analyse	Schnittstellenspezifikation Investitionskosten, Betriebskosten	Formulare
Realisierungsplan erstellen	Aufwandschätzung Realisierungsstufen, Einführungsplanung	
Antrag für weiteres Vorgehen	Detailplan der nächsten Phase, Antrag für die nächste Phase	Netzplan, Balkendiagramm

### Detailstudie

- Detaillierte Lösungskonzepte werden entworfen.
- Die neuen Unter- und Teilsystem müssen in die Systemlandschaft integriert werden.
- Einzel- und Untersysteme werden soweit konkretisiert, dass sie gebaut werden können.
- Redaktionsschluss (PSP/WBS)

Aktivitäten Detailstudie	Ergebnis	Darstellungsmethode
Analyse	Arbeitsplan (Mitarbeiter, Termine)	Netzplan, Balkendiagramm
Phasenauftrag, Phasenplanung		
Logische Transaktionen identifizieren	Katalog der logischen Transaktionen	KLT
Ablauf der logischen Transaktionen festlegen	Ablauf der logischen Transaktionen	BAT
Datendesign fertigstellen	Datenmodell: Zeitaspekte, Historisierung, Schlüsselfestlegung	
Verarbeitungslogik der logischen Transaktionen definieren	Transaktions-Logik-Diagramm	ETM, TLD
Organisations-Modell verfeinern	Matrix Org.-Einheit	LTOM
Online-Dialog spezifizieren	Logische Transaktion Menu-Struktur, Dialog-Ablauf, Bildschirmformulare, Funktionstasten, Listen/Formulare	MEN, DLGD, BFB
Batch-Verarbeitung spezifizieren	Datenfluss-Diagramm	
Programm-Vorgaben erstellen	Pseudo-Code Beschreibung, Fehlermeldungen	PCB
Antrag für weiteres Vorgehen	Detailplan der nächsten Phase, Antrag für die nächste Phase	Netzplan, Balkendiagramm

### Systembau

- Einführung planen:  
Schlagartige Einführung, stufenweise Einführung, parallellaufende Einführung.
- Präsentationen vorbereiten.
- Service Desk vorbereiten.
- Schulungen vorbereiten.
- Ziel: "Fertig installiertes, betriebsbereites System liegt vor."

### Systemeinführung

- Eskalationsprozesse definieren.
- Schulungen durchführen.

### Systemnutzung

- Beobachtung, Protokollierung und Nachbesserung des Systems.

### Checkliste Projektabschluss

- Formelle Produktabnahme
- Projektinhalt und -umfang dokumentieren per Abschluss
- Lessons Learned dokumentieren

## Problemlösungszyklus

Der Problemlösungszyklus ist iterativ für die Phasen:

- Vorstudie
- Hauptstudie
- Detailstudie

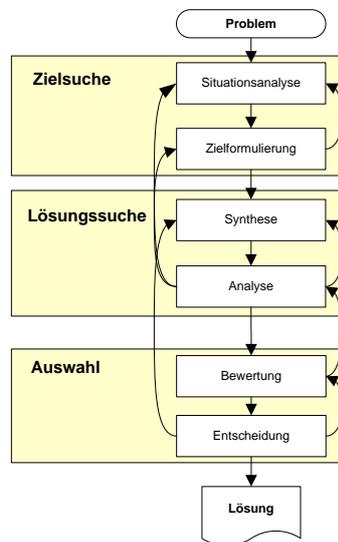


Abbildung 22: Problemlösungszyklus

## Systemgestaltung

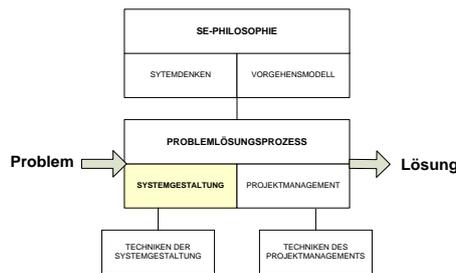


Abbildung 23: Systemgestaltung

Der Zielsucheprozess muss bei der **Vorstudie**, **Hauptstudie** und der **Detailstudie** jeweils am **Anfang** und am **Ende** durchlaufen werden.

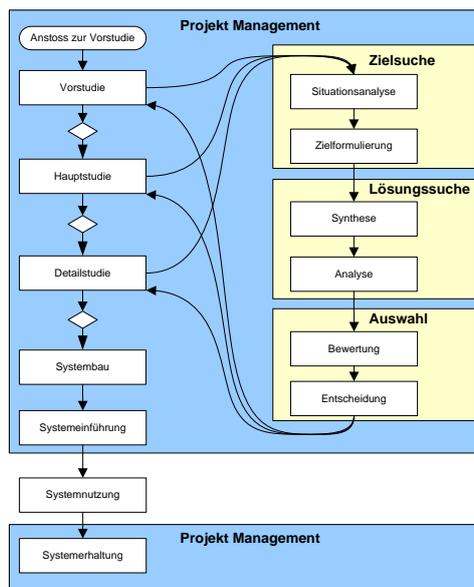


Abbildung 24: Vorgehensmodell

### Situationsanalyse

Unter Situationsanalyse verstehen wir die systematische Durchleuchtung (Analyse) einer intuitiv als problematisch empfundenen Erscheinung oder eines im Auftrag angegebenen Sachverhaltes (Situation) zu Beginn der Planungstätigkeit.

Den betrachteten Bereich derart zu strukturieren, dass die Problemsituation erkennbar und eine **Problemdefinition** ermöglicht wird.

#### Fokus:

- Zielformulierung ermöglichen
- Synthese und Analyse unterstützen

- **Systemorientierte Betrachtung**  
Elemente, Beziehungen, Blackbox-Betrachtung  
Über- und Untersysteme, Teilsysteme
- **Ursachenorientierte Betrachtung (diagnostisch)**  
Beschreibung der Chancen und Risiken
- **Lösungsorientierte Betrachtung (therapeutisch)**  
Beschreibung realistischer Ziele

- **Zeitorientierte Betrachtung**  
Trendanalysen, Prognosetechniken
- Informationsbeschaffung

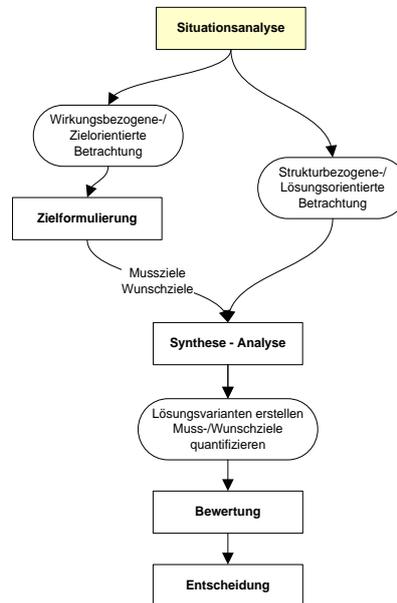


Abbildung 25: Situationsanalyse zur Zielformulierung und Synthese/Analyse

#### Arten von Situationsanalysen

- Wirkungsanalysen
- Strukturanalysen
- Einflussgrößenanalysen
- Funktionsanalysen
- Mittelanalysen

#### Zielformulierung

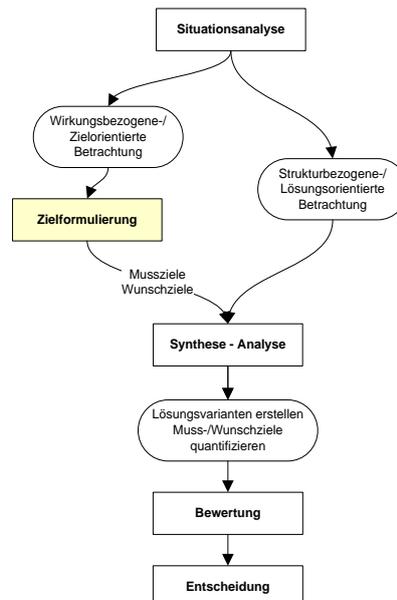


Abbildung 26: Zielformulierung

Was soll erreicht (vermieden) werden?

Zielvorstellungen, die sich teilweise bereits in der Situationsanalyse ergeben haben, zu bereinigen, systematisch zu strukturieren, auf Vollständigkeit zu prüfen, zu ergänzen und schliesslich in verbindlicher Form festzuhalten.

- Lösungsneutral
- Vollständig
- Präzise und verständlich
- Realistisch
- Muss- und Wunschziele unterscheiden
- Zielkonflikte bewältigen
- Definition der kurz-, mittelfristige- und langfristige-Ziele
- Ziele messbar machen

## Synthese & Analyse

### Fokus:

Möglichst umfassenden Überblick über **Lösungsmöglichkeiten** verschaffen.

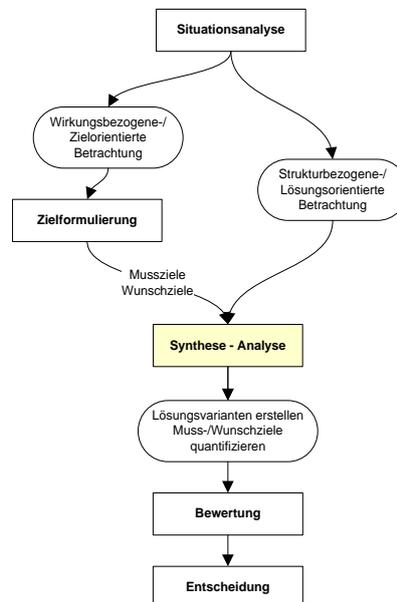


Abbildung 27: Synthese & Analyse

### Synthese

- Variantenkreation unter Anwendung verschiedener Kreativitätstechniken.

### Analyse

- Mussziele
- Konzeptprüfung
- Wirkungsanalyse

### Bewertung

- Nutzwertanalyse
- Kosten/Nutzen-Analyse
- Kosten/Wirksamkeitsanalyse
- Vorbereiten der Ergebnispräsentation
- etc.

## Entscheidung

- Ergebnispräsentation

## Projektmanagement im SE

Der Begriff Projektmanagement steht für alle planenden, überwachenden und steuernden Vorgehensmassnahmen.

**SE** definiert den Problemlösungsprozess.

**PM** definiert den Vorgehensprozess.

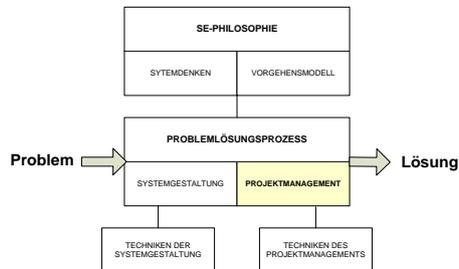


Abbildung 28: Projekt-Management im SE

### PM-Funktional

- **Projektplanung**  
Etappenziele, Ablauforganisation, Aufbauorganisation, Aufwand, Terminierung, Budgetierung, Projektinformations- und -dokumentationssystem
- **Projektmanagement Implementierung**  
Aufbauorganisation, Bereitstellung der Mittel und Räume usw.
- **Projektsteuerung**  
Zuordnung von Aufgaben, Kompetenzen und Verantwortungen  
Anleitungen, Motivierung, Abschirmung von Mitarbeitern  
Überwachen des Ablaufs des Problemlösungsprozesses
- **Projektkontrolle**  
Aussagefähige Zusammenfassungen über die zu überwachenden Teilaufgaben erstellen.  
Bewertung des Projektstandes um den noch zu erwartenden Aufwand zu quantifizieren.

### PM-Institutionell

- Fokussiert auf die Projektorganisation.

WAS	Aufgabeninhalt qualitativ und quantitativ
WANN	Zeitfolgen, -dauer, -punkte
WER	personelle Arbeitszuordnung
WIE	Verfahren
WO	Ort
WOMIT	Sachmittel, Art und Menge
WOHER	Beschaffung von Personen und Mitteln
WOHIN	Verwendung von Personen und Mitteln nach erfolgtem Leistungsvollzug

# Strukturierte Systemanalyse (nach Tom DeMarco)

## Grundlagen

Als **Essenz** eines Systems gelten die Aspekte, die unabhängig von allen technologischen Randbedingungen gelten. Ergo die Summe der wahren Anforderungen an ein System.

Die **essentiellen Aktivitäten** (Reaktionen) sind alle Aufgaben, die das System ausführen müsste wenn es mit perfekter Technologie implementiert wäre. Diese Aktivitäten werden im Systemmodell als Knoten dargestellt.

**Datenflussdiagramme** (DFD) sind die abstrakte Darstellung von essentiellen Aktivitäten in Mini-Spezifikationen.

**Ereignisse** werden im Systemmodell als Pfeile dargestellt.

### Gefahr!

In ihrem Bestreben, vollständige Spezifikationen zu erstellen und angesichts der Tatsache, dass ihnen die Fähigkeit zur Erkennung wahrer Anforderungen fehlt, erforschen und dokumentieren Systemanalytiker alles, um der Gefahr zu entgehen, etwas zu vergessen.

Der Begriff **Prozessor** wird in einer sehr allgemeinen Weise verwendet. Er beschreibt alles was eine Reaktion auf ein Ereignis ausführt.

*Dies kann sein:*

- Mensch, mechanisches Gerät, Digitalgerät oder sonst irgendwas.

Prozessoren bestehen aus vier Elementen:

- Sensoreinrichtung
- Verarbeitungseinheit
- Speicher
- Schnittstelleneinheit

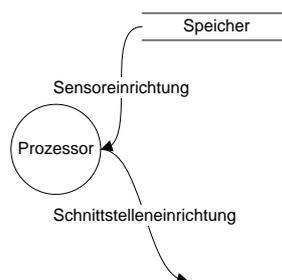


Abbildung 29: Prozessor

DeMarco verwendet den Begriff "**Policy**" für die logische Version des Systemmodells.

Zweck der (essentiellen) **Systemanalyse** ist es, ein Dokument der wahren Anforderungen an ein zu erstellendes System zu produzieren.

## System-Archäologie

1. Suchen der Fragmente aller essentiellen Aktivitäten in den bestehenden Inkarnationen.  
Entfernen aller physikalischen Charakteristika. Klassifizieren der Fragmente.
2. Rekonstruieren der essentiellen Aktivitäten und erstellen eines Modells.
3. Integrieren der einzelnen Modelle zu einem Gesamtmodell.

Gemäss *DeMarco* reagieren interaktive Systeme **spontan** oder **geplant** auf Ereignisse. Ziel ist es die spontanen Reaktionen zu reduzieren/eliminieren.

## Inkarnation eines Systems

Der Begriff "**Inkarnation**" wird für die Verkörperung eines Konzeptes verwendet.

Die **Technologie** legt uns vier Schranken auf:

Kosten, Kapazität, funktionelle Leistungsmerkmale, Fehlerhafte Systeme.

## Essenz der Systementwicklung

Definition eines gut entworfenen Systems:

Vollständigkeit, Verständlichkeit und Portabilität.

## Essentieller Systementwicklungsprozess

Die folgenden **drei** Prozesse findet man in jedem Systementwicklungsprojekt!



Abbildung 30: Systementwicklungsprozess

## Systemmodellierung

### Modelldarstellungstechnik

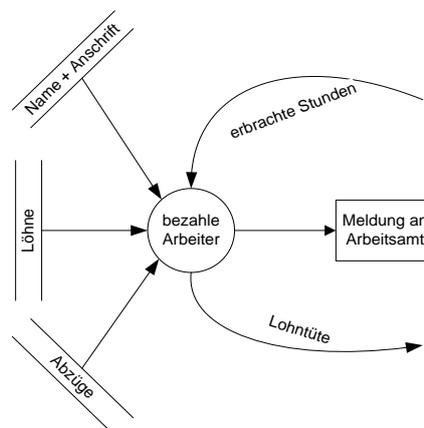


Abbildung 31: Datenspeicher

- Die benannten parallelen Linien werden **Datenspeicher** genannt.
- Knoten sind **essenzielle Aktivitäten**
- Rechtecke zeigen Meldungen über die **Systemgrenze** hinweg.

- Die Pfeile zeigen **Ereignisse**

## Die vier Grundprinzipien der Systemmodellierung

- Grad der Komplexität
- Technologische Neutralität
- Perfekte interne Technologie
- Minimale essentielle Modelle

### *Grad der Komplexität*

Die menschlichen Schranken bei der Verarbeitung von Informationen sind der Gradmass für die Komplexität.

Eine Mini-Spezifikation soll nicht länger als eine Seite sein.

Ein Datenflussdiagramm (DFD) soll nicht mehr als sieben Aktivitäten darstellen.

Die Bezeichnungen sollen sorgfältig gewählt werden und präzise sein.

### *Technologische Neutralität*

Es dürfen keine technologischen Details in das logische Modell aufgenommen werden.

Der Zweck eines logischen Modells ist es, das, was die Aktivitäten tun sollen, von dem zu trennen, wie es mit einer bestimmten Technologie getan werden kann.

### *Perfekte interne Technologie*

Bei der Anwendung des Konzepts der perfekten Technologie lassen Sie die Eigenschaften der Technologie ausser acht, die zur Implementierung Ihres Systems benutzt werden könnten. Es werden nur innerhalb des Modells perfekte Technologien vorausgesetzt es gibt keine Annahmen über Korrektheit, Geschwindigkeit, Fähigkeiten, Speichervermögen und Kosten der Technologien ausserhalb des Modells. Das ist das Prinzip der perfekten internen Technologie.

### *Minimale essentielle Modelle*

Hat man mehrere Möglichkeiten, eine essentielle Aktivität auszudrücken, so soll man die auswählen, die die wenigsten Aktivitäten und Speicher hat. Auch innerhalb jeder Aktivität soll man die geringste Komplexität wählen. Auf diese Art stellen Sie sicher, dass Die die Auswahl der Inkarnation mit einer präzisen Anforderungsdefinition beginnen können.

## Sytemzerlegung

Die Zerlegung von **Reaktionen** in Systemteile nennen wir "**ereignisorientierte Zerlegung**".

Die "**objektorientierte Zerlegung**" benutzen wir zur Zerlegung des **essentiellen Speichers** in Objekte, denen dann jeweils die Datenelemente zugeordnet werden, die das jeweilige Objekt beschreiben.

### **Ziele:**

Essentielle Aktivitäten kommunizieren ausschliesslich über Speicher miteinander.

Nach Ausführung der Aktivität muss das System stillstehen, bis ein neues Ereignis eintrifft. Wenn dies nicht der Fall ist, dann ist die Aktivität nicht vollständig.

## Testen

- Intuition
- Grenzwertanalyse
- Äquivalenzklassenbildung

Pulttest

Walk-Through

Code Inspection

Peer Rating

NMock2

NUnit

## Bottom-Up Testing

- Is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher-level components.
- The process is repeated until the component at the top of the hierarchy is tested.

### **Advantage:**

- With this approach, errors in critical modules are detected earlier.

## Regression Test

- Type of Quality Assurance (QA) testing.
- Regression testing means that after a change to a system takes place, you retest to ensure functionality, performance, and protection.

## Smoke Test

- Type of Quality Assurance (QA) testing.
- A smoke test is a quick run through of a site.
- It focuses on critical functionality to ensure the site can perform basic features.
- The primary features are often called red routes in the software industry.
- It only takes a couple of minutes to complete, up to ten minutes at most.
- What is great about smoke tests is you can perform them either daily or every other day.
- Smoke testing came to software testing from a similar hardware test, where the device passed if it did not catch fire (or smoked) the first time it was turned on!

### **Example for a hotel reservation site:**

- In this smoke test example, the tester would ensure the user will be able to **sign up**, **change password**, **create a booking**, and **be notified**.

## DAST - Dynamic Application Security Testing

- This is a testing technique that simulates activities of a malicious attacker, using a wide range of tools and techniques to do so.
- The analyst actively tries to compromise the system.

## SAST - Static Analysis Security Testing

- This is an **examination of source code**, performed either manually or by automated code review tools.
- Automated software testing tools can quickly identify potential issues and, in some cases, provide information to fix code flaws.

- It provides higher assurance that the code or system will not be exploited once deployed.
- The tools can also validate that secure coding practices have been followed.
- Automated tools can greatly reduce the work force necessary to review source code.

## Darstellungstechniken

### Aktivitätendiagramm

Ein **Aktivitätendiagramm** besteht aus einer Folge von **Aktivitäten** und **Zustandsübergängen** und kann für die Beschreibung von **Geschäftsprozessen** und **Anwendungsfällen** bzw.

**Prozessvarianten** sowie für **Prozeduren** und **Operationen** verwendet werden.

Die Darstellungstechnik ist **wichtiger Bestandteil** der standardisierten Modellierungssprache "**Unified Modelling Language (UML)**". Eine Variante des Aktivitätendiagramms "**Swimlanes**" erlaubt eine Darstellung bei der Aktivitäten bestimmten Organisationseinheiten zugeordnet werden können.

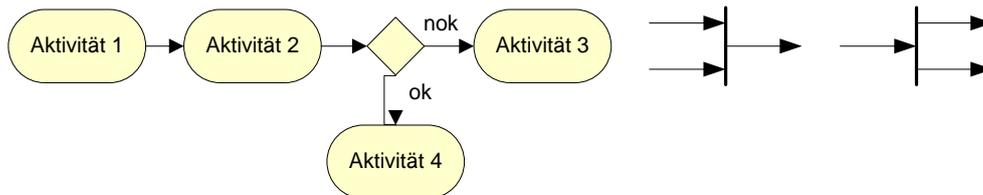


Abbildung 32: Aktivitätendiagramm

### Blackbox-Betrachtung

Wirkungsanalyse

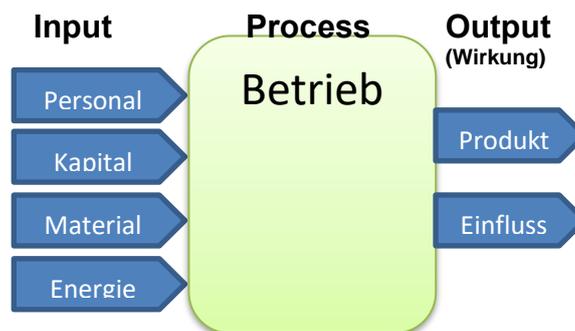


Abbildung 33: Blackbox-Betrachtung

### Entscheidungstabellen

Entscheidungstabellen sind Hilfsmittel zur Beschreibung von Entscheidungssituationen. Mit ihrer Hilfe lassen sich sehr transparent und eindeutig komplexe Abhängigkeiten dokumentieren. Sie dienen insbesondere für präzise Vorgaben für die Programmierung.

Bestellprozess		Regel1	Regel2	Regel3	Regel4
Bedingung1	Ist bereits Kunde	J	J	N	N
Bedingung2	Bonität ist OK	J	N	J	N
Aktion1	Liefern	X		X	
Aktion2	Negativ Meldung		X		X

Abbildung 34: Entscheidungstabelle

Die Anzahl der Regeln ergibt sich (unverdichtet) aus der Anzahl der Bedingungen(n)  $2^n = 4$ . Die Tabelle kann nach dem vollständigen Aufbau verdichtet werden (z.B. Regel1 & Regel3 plus Regel2 & Regel4).

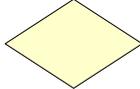
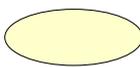
- J = Bedingung erfüllt
- N = Bedingung ist nicht erfüllt
- = Bedingung ist unerheblich
- X = Aktion wird ausgelöst
- (Blank) = Aktion wird nicht ausgelöst

Bestellprozess		Regel1&3	Regel2&4
Bedingung1	Ist bereits Kunde	-	-
Bedingung2	Bonität ist OK	J	N
Aktion1	Liefern	X	
Aktion2	Negativ Meldung		X

Abbildung 35: Entscheidungstabelle verdichtet

## Datenflussplan (DFP)

(DIN 66001)

	Rechteck	Allgemeine Operation bzw. Datenverarbeitung
	Trapez	Manuelle Operation bzw. Datenverarbeitung
	Rhombus (Raute)	Verzweigung der Operation bzw. Datenverarbeitung
	Ellipse	Anfang, Zwischenhalt oder Ende einer Operation bzw. Datenverarbeitung
	Kreis	Verbindungsstelle einer Operation bzw. Datenverarbeitung (zu einer anderen Verbindungsstelle mit gleicher Beschriftung)
	Parallelogramm	Allgemeine Daten
	Gewölbtes Rechteck	Maschinell zu verarbeitende Daten
	Dokument	Daten auf Schriftstück (z.B. auf Papiereleg oder in elektronischer Form auf Mikrofilm)

## Datenstrukturdiagramm (DSD)

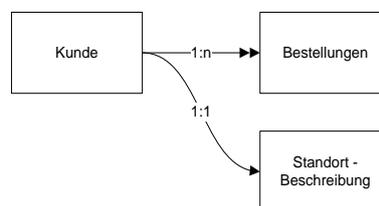


Abbildung 36: Datenstrukturdiagramm (DSD)

## Entscheidungsmatrix

Kriterien	Gewichtung 1 - 10	Lösung I		Lösung II		Lösung III	
Note 1-10		Note	Wert	Note	Wert	Note	Wert
Parameter I	10	5	50	8	80	9	90
Parameter II	8	4	32	5	40	7	56
Parameter III	5	5	25	6	30	9	45
<b>Summe</b>			<b>107</b>		<b>150</b>		<b>191</b>

Abbildung 37: Entscheidungsmatrix

## Entity-Relationship-Modell (ERM)



Abbildung 38: Entity-Relationship-Modell (ERM)

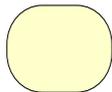
## Ereignisgesteuerte Prozesskette (EPK)

Zentraler Bestandteil der **ARIS-Methode** und des **ARIS-Toolsets**. Wurde mit **SAP** zusammen entwickelt.



Ereignis

Gibt einen Zustand oder eine Zustandsänderung an. Am Beginn und am Ende eines Prozesses oder Prozesszweigs steht ein Ereignis.



Funktion

Tätigkeiten bzw. Aufgaben, die im Verlauf eines Prozesses durchgeführt werden.



Linie (Kante)

Verbindung zwischen Ereignissen und Funktionen, die die zeitlich logische Abfolge des Prozessablaufs darstellen.

Der Prozess verzweigt nach folgenden logischen Regeln:

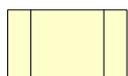


Verzweigung

UND: Alle Prozesszweige werden parallel durchlaufen.  
 XOR (exclusives Oder): Nur ein Prozesszweig wird durchlaufen  
 ODER: Ein oder mehrere Prozesszweige werden durchlaufen  
 Abteilung bzw. Stelle, die für eine Funktion zuständig ist.



Organisationseinheit



Anwendungssystem

ICT-System, das eine Funktion unterstützt.



Datenspeicher oder Datenbank

Input oder Output einer Funktion



Dokument

Input oder Output einer Funktion

## Flussdiagramm

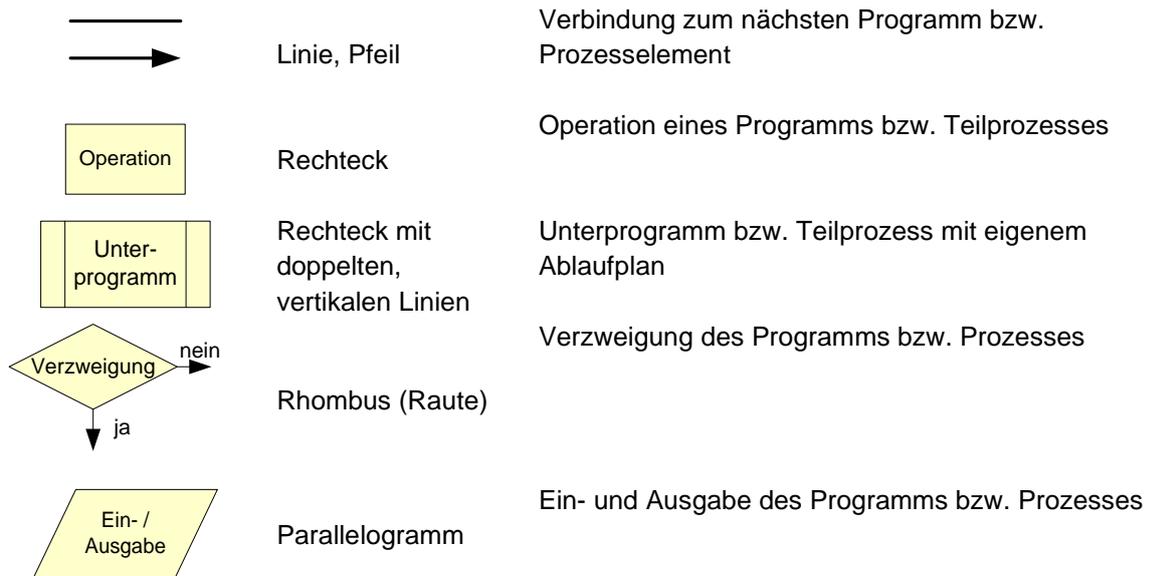
(DIN 66001)



Start/Stopp

Ellipse

Start, Ende des Programms bzw. Prozesses



## Mengengerüst-Tabelle

	Bestand	Mutationen	Zugänge	Abgänge
Bestellungen		150/T		
Kunden	900	30/Mt	10/J	10/J
Artikel	20			
Rechnungen		700/Mt		
Touren	35	7/T		
Chauffeure	8			
Lastwagen	8			
Beifahrer	5			
Speditionsmitarbeiter	4			

Abbildung 39: Mengengerüst-Tabelle

## Projektauftrag

Projekt Nr.

Projekt-Name:

1. Grundlage:

2. Ziele:

3. Aufgabenstellung:

4. Gestaltungsbereich:

5. Restriktionen:

6. Projektorganisation: - Auftraggeber  
- Ausschuss  
- Projektleiter  
- Team  
- Berater

7. Termine: Gesamt Beginn:  
Ende:  
Vorstudie Beginn:  
Ende:

8. Genehmigung des  
Auftraggebers

9. Budget

10. Kommunikations-  
richtlinien

Abbildung 40: Projektauftrag

## **SWOT-Analyse (Potentialanalyse)**

Die SWOT-Analyse (S-Strengths - Stärken, W-Weaknesses - Schwächen, O-Opportunities - Chancen, T-Threats - Risiken) ist ursprünglich ein Instrument des strategischen Managements und dient dazu, aus den Stärken und Schwächen einer Organisation (interne Sicht) und den Chancen und Risiken der Umwelt (externe Sicht) geeignete strategische Lösungsalternativen für die Erreichung der Ziele der Organisation abzuleiten.

## DEFINITIONEN

Atomisieren     Unteilbar/kann nicht mehr weiter zerlegt werden.

### ACID - Atomicity, Consistency, Isolation, and Durability

- **Atomicity** means a transaction must execute exactly once and must be atomic; either all of the work is done, or none of it is. Operations within a transaction usually share a common intent and are interdependent.
- **Consistency** ensures that there is consistent data before the transaction, and data is consistent after the transaction.
- **Isolation** ensures that one transaction is not impacted by another transaction.
- **Durability** means that the changes made due to the transaction are permanently saved in the system. Committed data is saved by the system so that even in the event of a failure and system restart, the data is available in its correct state.

### ALM - Application Lifecycle Management

- See: HP ALM

### Angular

- **AngularJS** (zur Abgrenzung vom Nachfolger Angular (Version 2 und höher) auch AngularJS 1 oder Angular 1 bezeichnet) ist ein **clientseitiges JavaScript-Webframework** zur Erstellung von Single-Page-Webanwendungen nach einem Model-View-ViewModel-Muster.
- Die Softwareentwicklung und das Komponententesten können damit vereinfacht werden.
- Es wird als Open-Source-Framework vom US-amerikanischen Unternehmen Google Inc. entwickelt.

### Ansible

- Open-source IT automation engine.
- Ansible is known for its simplicity.
- Provisioning, configuration management and application deployment.
- Ansible doesn't require an agent.
- Ansible uses YAML
- Ansible does require Python on Linux hosts
- Ansible requires PowerShell 3 on Windows hosts

### Agile Softwareentwicklung

See: **Scrum Master Handguide.docx**

### Apache Maven

- See: <https://mvnrepository.com>
- Maven Repository.
- Maven ist ein **Build-Management-Tool** der Apache Software Foundation und basiert auf **Java**.
- Mit ihm kann man insbesondere Java-Programme standardisiert erstellen und verwalten.

### AS3 - Application Services 3 Extension

- See: **F5**
- AS3 is a flexible, low-overhead mechanism for managing **application-specific configurations** on a BIG-IP system.

- The Application Services 3 Extension uses a declarative model, meaning you send a declaration file using a single Rest API call.

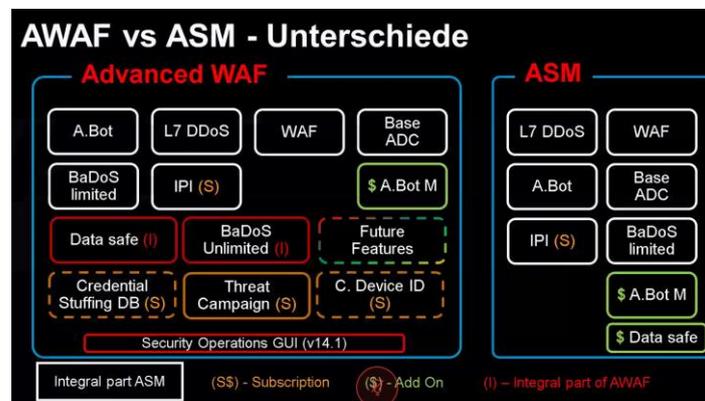
## BIG-IP Local Traffic Manager (LTM)

See: **F5**

See: <https://clouddocs.f5.com/>

- BIG-IP LTM ermöglicht einen intelligenten **Lastenausgleich** (Load Balancer).
- Da **BIG-IP LTM** ein vollständiger **Proxy** ist, können Sie den in Ihrem Netzwerk eingehenden und ausgehenden Datenverkehr inspizieren, verwalten und darüber berichten.
- Datenverkehr auf der Grundlage der angefragten **URL** umleiten oder bestimmte Serverantworten in einem Berichtssystem protokollieren.
- Verschlüsselung aller Inhalte zwischen Client und Server mit **SSL**.

**Trainings:** <https://f5.com/education/super-netops-training>  
<https://www.f5.com/c/emea-2020/webinar/f5-shape-security-virtual-workshop>  
<https://academy.westconcomstor.com/ch/vendors/F5+Networks>



## BIG-IP Access Policy Manager (APM)

- F5's BIG-IP product family comprises hardware, modularized software, and virtual appliances that run the **F5 TMOS** operating system.
- Sorgt für einen sicheren und differenzierten Zugriff auf Anwendungen, Daten, Netzwerk und Cloud basierend auf Benutzeridentität und Kontext.
- So können Sie zentral festlegen, wer auf Ihr Netzwerk oder Ihre Cloud zugreifen kann, auf welche Anwendungen zugegriffen werden kann und von welchen Geräten und Standorten aus der Zugriff auf diese Apps möglich ist.
- Kurz gesagt, BIG-IP APM vereinheitlicht und erzwingt eine **identitätsbasierte** und kontextbezogene dynamische richtliniengesteuerte Steuerung des **Zugriffs auf Anwendungen**, unabhängig vom Standort des Benutzers oder der Anwendung.

## Bitbucket

Vendor: Atlassian

- Open Source
- SW distributed **version control system** that makes it easy to collaborate with your team.
- The only collaborative **Git solution** that massively scales.
- Profitiere mit "Pull Requests" von einer effizienteren Genehmigung von Code-Reviews.
- Führe Diskussionen mithilfe von Inline-Kommentaren direkt im Quellcode.

- Sichere dir mit dem Data Center einen störungsfreien Zugang und eine schnelle, vom Standort unabhängige Leistung.

## Business Analyst / Business Analysis

- ???

## BPaaS - Business Process as a Service

- ???

## BPM - Business Process Management

**Business Process Management (BPM)**, Geschäftsprozessmanagement, beschäftigt sich mit dem Herausfinden, Gestalten, Dokumentieren und Verbessern von Geschäftsprozessen. Geschäftsprozessmanagement synchronisiert die Bereiche Planung, Entwurf, Konstruktion, Produktion, Instandhaltung, Nachverfolgung und Anpassung in einer Organisation. „Wer macht was, wann, wie und womit?“ ist eine zentrale Fragestellung. Zur Verbesserung und Steuerung werden entsprechende Kennzahlen verwendet. Diese Kennzahlen können z. B. in einer **Balanced Scorecard** dargestellt werden.

## BRMS - Business Rules Management System

- ???

## BI - Business Intelligence

Der Begriff **Business Intelligence (BI, Geschäftsanalytik)** bezeichnet Verfahren und Prozesse zur systematischen Analyse (Sammlung, Auswertung und Darstellung) von Daten in elektronischer Form. Ziel ist die Gewinnung von Erkenntnissen, die in Hinsicht auf die Unternehmensziele bessere operative oder strategische Entscheidungen ermöglichen. Dies geschieht mit Hilfe analytischer Konzepte und IT-Systeme, die Daten über das eigene Unternehmen, die Mitbewerber oder Marktentwicklung im Hinblick auf den gewünschten Erkenntnisgewinn auswerten. Mit den gewonnenen Erkenntnissen können Unternehmen ihre Geschäftsabläufe, Kunden- und Lieferantenbeziehungen profitabler machen, Kosten senken, Risiken minimieren und die Wertschöpfung vergrößern.

In der Praxis versteht man in den meisten Fällen unter „Business Intelligence“ die Automatisierung des Berichtswesens (Reporting). Die in den ERP-Systemen anfallenden Unternehmensdaten werden genutzt, um unter verschiedenen Blickwinkeln die Situation des Unternehmens zu analysieren und ggf. zu bewerten. Die Analyse erfolgt nicht in den ERP-Systemen, sondern in einer davon getrennten Datenbasis, dem Data-Warehouse (DWH).

## Blockchain

- A **blockchain** — originally **block chain** — is a distributed database that maintains a continuously-growing list of records called blocks secured from tampering and revision.
- Each block contains a **timestamp** and a **link to a previous block**.
- The blockchain is a technology that underlies **bitcoin**—conceived in 2008 and first implemented in 2009—where it serves as the public ledger for all transactions.
- In the bitcoin case, every compatible client can connect to the network, send new transactions to it, verify transactions, and take part in the competition to create new blocks.
- The competition creating new blocks is known as mining.
- The bitcoin design has been the inspiration for other applications.

Source: Wikipedia

## CaaS - Container as a Service

- Die Abkürzung CaaS steht für Container as a Service.
- Es handelt sich um ein Cloud-Computing-Modell, mit dem sich Leistungen rund um die Container-basierte Virtualisierung als Service aus der Cloud nutzen lassen.
- Anwender von CaaS benötigen keine eigene Infrastruktur für die Container-Virtualisierung und müssen keine eigene Container-Plattform betreiben und managen.
- Sämtliche Ressourcen für die Virtualisierung wie Rechenleistung, Speicherplatz, Container-Engine und Orchestrierungssoftware stellt der Cloud-Service-Provider zur Verfügung.
- Es ist möglich, CaaS als Service aus einer Public Cloud oder aus einer Private Cloud zu beziehen.
- Was das Cloud-Computing-Modell angeht, liegt Container as a Service zwischen den Modellen Infrastructure as a Service (IaaS) und Platform as a Service (PaaS).
- Zentrale Komponenten von CaaS sind die Container, die die virtualisierten Applikationsumgebungen bereitstellen.
- Im Gegensatz zu einer virtuellen Maschine beinhaltet die Virtualisierungsebene kein eigenes Betriebssystem.
- Die Verwaltung der Container und das Hochladen von Daten erfolgt beim CaaS über eine Web-Schnittstelle oder eine spezielle API (Application Programming Interface).
- Das Kostenmodell für die Services ist in der Regel ein nutzungsabhängiges Mietmodell.
- Grosse Public-Cloud-Anbieter wie Microsoft, Google oder Amazon bieten ihren Kunden CaaS-Leistungen an.

## CBTC - Communication Based Train Control

- Automatische Zugbeeinflussungssysteme.
- Benutzt das «**Moving Blocks Signaling**» System anstelle des «**Fixed Block Signaling**» systems.
- Dabei erfolgt eine **externe Steuerung** des Triebfahrzeuges durch **Zugsteuerungs- und Zugsicherungssysteme**.
- Die Übermittlung der Fahrerlaubnis und Steuerungsbefehle erfolgt nicht durch äussere optische Signale an einen Triebfahrzeugführer, sondern über **Datenkommunikation zwischen Schienenfahrzeug und Streckenausrüstung**.
- Die Grundfunktion der Bewegung des Schienenfahrzeuges kann um die bidirektionale Übermittlung von Betriebsdaten des Fahrzeuges oder des Transportgutes wie Türsteuerung oder Fahrgastinformationen erweitert werden.
- Die Anforderungen und die allgemeine Systemarchitektur von CBTC-Systemen sind in mehreren **IEEE-Standards** definiert.
- CBTC-Systeme wurden bisher überwiegend bei Nahverkehrssystemen wie **U-Bahnen** eingesetzt.
- Innerhalb von ETCS (Level 2 und 3) werden für den Eisenbahnbetrieb nach offenem Industriestandard Funktionen bereitgestellt, die der Funktion von CBTC entsprechen.

Ein CBTC-System kann unterschiedliche Stufen abdecken:

- **ATO** - Automatic Train Operation / Vollautomatischer Zugbetrieb
- **ATP** - Automatic Train Protection / Zugbeeinflussung
- **ATS** - Automatic Train Supervision / Zugleitsystem

Siehe auch:**OCTYS**

### **Techniques:**

WiFi 802.11 Frequency Hopping (Spread Spectrum, FHSS)

Dedicated 4G LTE

Shared Public 4G LTE

**Products:**

SelTrac (Thales)

Urbalis

**Einsatz:**

USA: U-Bahn New York

Spanien: Metro Madrid

Kanada: Toronto, Vancouver

Singapur: MRT Singapur

Frankreich: Metro Paris

Dänemark: S-Bahn Kopenhagen

## CD - Continuous Delivery/Deployment

- Continuous Delivery (CD) bezeichnet eine Sammlung von Techniken, Prozessen und Werkzeugen, die den **Softwareauslieferungsprozess** verbessern.

## CI - Continuous Integration

- Techniken wie Continuous Integration (CI), **Testautomatisierung** und **kontinuierliche Installation** erlauben in Kombination mit agilen Methoden die Entwicklung qualitativ hochwertiger Software.
- The cost and effort required to fix a bug increases exponentially if found in later stages of the Software Design Life Cycle (SDLC), while also increasing risk. It is therefore crucial to find and fix bugs as soon as they are introduced.
- Have you been in a situation where multiple code changes have been committed to your application and something then breaks? How do you find out which change caused the problem? Usually, this scenario leads to multiple exchanges with the bug being passed from developer to developer trying to detect where the bug was introduced.
- The best way to address this problem is with a software development best practice called Continuous Integration (CI) which helps improve software quality without slowing down releases. Every code commit is immediately integrated into a common shared repository and is automatically built and tested. The developer making the code change gets immediate feedback and can fix bugs easily if required since every bug can be tracked down to a commit.

**Tools:**

- Shippable [www.shippable.com](http://www.shippable.com)
- Jenkins

## Cisco HyperFlex

- Management, Provisioning and Optimization.

## Cisco ISE - Identity Services Engine

## Cisco NSX

## Citrix

- Thin Client solution  
Terminal-Server

## Citrix XenApp

## Citrix XenDesktop

## Citrix Netscaler

- Application Delivery Controller (ADC)
- **Loadbalancing** and **monitoring**.
- Acts as **proxy** for the client.

## CMM - Capability Maturity Model

- Introduced by **Software Engineering Institute (SEI)** at Carnegie Mellon University.
- Also called **SW-CMM** or **SCMM**.

**Capability Maturity Model** (CMM®, Reifegradmodell) ist ein Reifegradmodell zur Beurteilung der Qualität ("Reife") des Softwareprozesses (Softwareentwicklung, Wartung, Konfiguration etc.) von Organisationen sowie zur Bestimmung der Massnahmen zur Verbesserung desselben.

CMM wurde Ende 2003 durch **Capability Maturity Model Integration** (CMMI®) ersetzt, um dem Wildwuchs diverser CM-Modelle (jede Entwicklungs-Disziplin entwickelte ein eigenes Modell) entgegenzuwirken und ein neues, modulares und vor allem vereinheitlichtes Modell zu erstellen.

Andere populäre Modelle sind **ITIL** für die Infrastruktur und **S3M** für die Software-Wartung.

Bei **CMM / CMMI** wird die Qualität mit einer von **fünf Stufen** bewertet, wobei die Qualität mit jeder Stufe steigt.

1. **Initial:**

Dies ist der Grundzustand, den jede Organisation erreicht, auch ohne dass ein Prozess für die Softwareentwicklung definiert und umgesetzt wird.  
Kosten, Zeiten und Qualität sind nicht vorhersehbar.

2. **Repeatable / Managed:**

Ein grundlegender Prozess existiert. Die Planung neuer Projekte erfolgt anhand der Erfahrungen mit vergangenen Projekten.  
Zeiten sind einigermaßen kontrollierbar. Kosten und Qualität unterliegen starken Schwankungen.

3. **Defined:**

In der Organisation ist ein typischer Software-Entwicklungs- und -Wartungsprozess eingeführt und dokumentiert (Standard-Software-Prozess).  
Eine spezielle Organisationseinheit ist für die Umsetzung verantwortlich.  
Kosten und Zeiten sind hier einigermaßen zuverlässig bewertbar. Qualität ist immer noch Schwankungen ausgesetzt.

4. **Managed / Quantitatively Managed:**

Sowohl für das Produkt als auch für den Prozess werden quantitative Ziele vorgegeben, ihre Erreichung gemessen und überwacht.  
Zeiten, Kosten und Qualität sind zuverlässig kontrollierbar.

5. **Optimizing:**

Die gesamte Organisation konzentriert sich auf das Finden von Schwächen und die weitere Verbesserung des Prozesses.

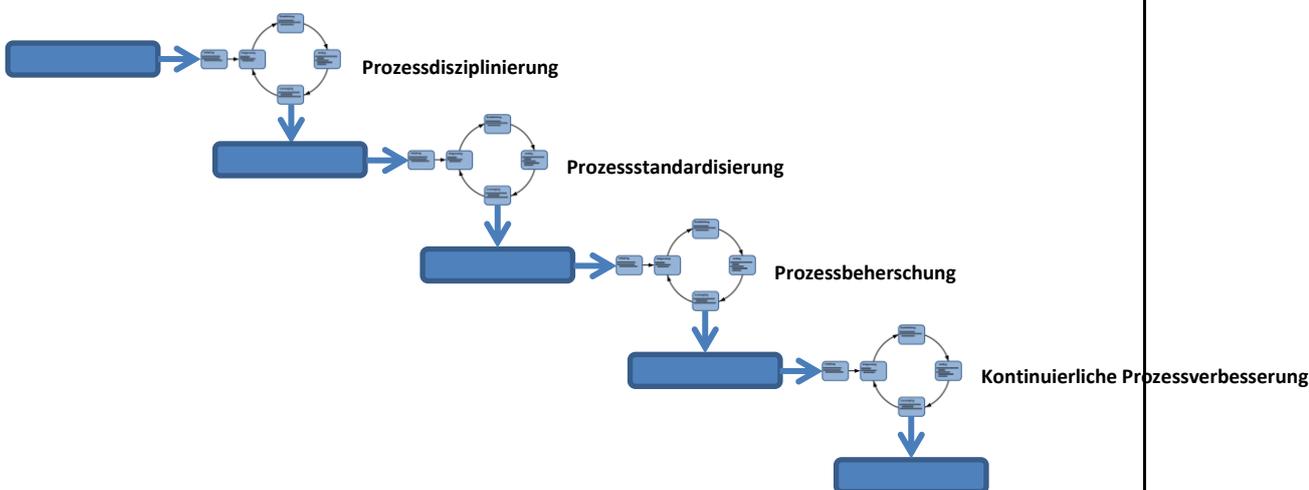
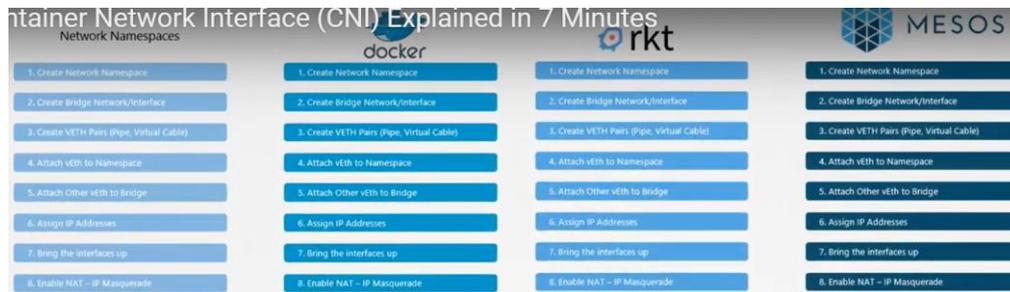


Abbildung 41: CMM + IDEAL

**The stages of SW-CMM:**

- Level 1: **Initial**
- Level 2: **Repeatable**
- Level 3: **Defined**
- Level 4: **Managed**
- Level 5: **Optimizing**

## CNI - Container Network Interface



## CobiT

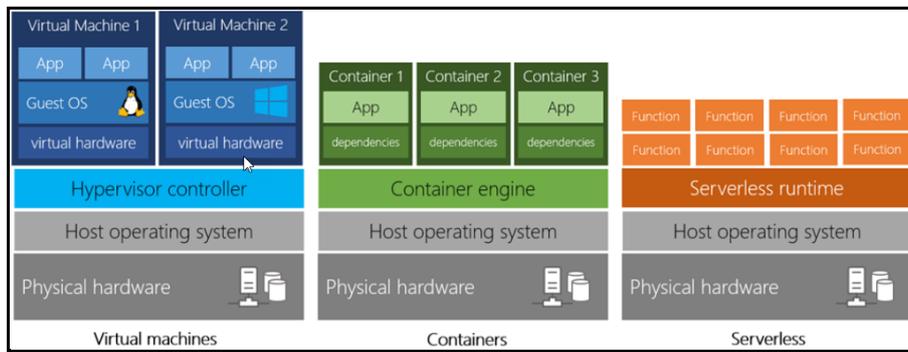
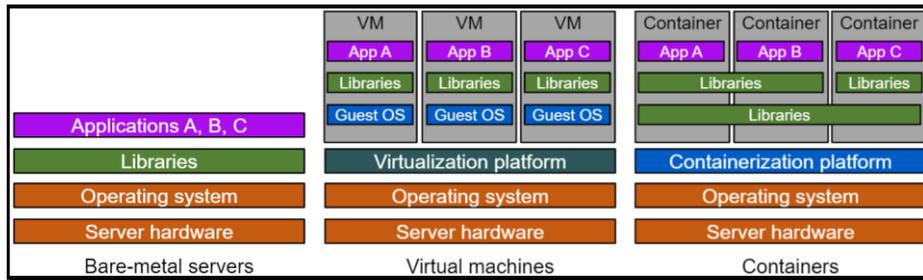
**CobiT** (*Control Objectives for Information and Related Technology*) ist das international anerkannte Framework zur IT-Governance und gliedert die Aufgaben der IT in Prozesse und Control Objectives (*Steuerungsvorgaben*). CobiT definiert hierbei nicht vorrangig, *wie* die Anforderungen umzusetzen sind, sondern primär darauf, *was* umzusetzen ist.

## CONTAINERS

- Examples: **Docker, OpenShift**
- Containers are a virtualization environment for running applications.
- Do not require a **guest Operating System**.
- Just like virtual machines, containers are run **on top of a host operating system**.
- But unlike VMs, **containers don't include an operating system** for the apps running inside the container.
- Instead, containers bundle the **libraries** and **components** needed to run the application and use the existing host OS running the container.
- For example, if five containers are running on a server with a specific Linux kernel, all five containers and the apps within them share that same Linux kernel.
- A container doesn't use **virtualization**, so it doesn't waste resources simulating virtual hardware with a redundant OS.

### Some Definitions

- **Container Host:** Also called the Host OS. The Host OS is the operating system on which the Docker client and Docker daemon run. In the case of Linux and non-Hyper-V containers, the Host OS shares its kernel with running Docker containers. For Hyper-V each container has its own Hyper-V kernel.
- **Container OS:** Also called the Base OS. The base OS refers to an image that contains an operating system such as Ubuntu, CentOS, or windows-server-core. Typically, you would build your own image on top of a Base OS image so that you can take utilize parts of the OS. Note that windows containers require a Base OS, while Linux containers do not.
- **Operating System Kernel:** The Kernel manages lower level functions such as memory management, file system, network and process scheduling.



## What is a Container?

- Lightweight:** alternative for app development
- Cloud-native:** easy to build into Public or Private cloud environments
- Portability:** Easy lift and shift to clouds or vice versa

### CONTAINERS VS VMS

Unlike VMs, containers can share the same OS, virtualizing at the OS level rather than the hypervisor level.

## CORBA - Common Object Request Broker Architecture

Die **Common Object Request Broker Architecture (CORBA**, engl. Allgemeine Architektur für Vermittler von Objekt-Anforderungen) ist eine Spezifikation für eine objektorientierte Middleware, deren Kern ein sog. **Object Request Broker**, der ORB bildet, und die plattformübergreifende Protokolle und Dienste definiert. Sie wird von der Object Management Group (OMG) entwickelt. CORBA-konforme Implementierungen vereinfachen das Erstellen verteilter Anwendungen in heterogenen Umgebungen.

## Coreo Flannel

- See also **CALICO**
- **flannel** is a virtual network that gives a subnet to each host for use with container runtimes.
- Platforms like Google's Kubernetes assume that each container (pod) has a unique, routable IP inside the cluster.
- The advantage of this model is that it reduces the complexity of doing port mapping.
- Maintains IP address range per node and writes IP address range to file (for integration with Docker or CNI)

- It's a pluggable backend for networking, usually VxLAN.

## COSO - Committee of Sponsoring Organizations of the Treadway

### COTS - Commercial of the Shelf

- Products are **packaged solutions** which are then adapted to satisfy the needs of the purchasing organization, rather than the commissioning of custom-made, or bespoke, solutions.
- A related term, Mil-COTS, refers to COTS products for use by the U.S. military.

Source: Wikipedia

### Commission

Das **COSO (Committee of Sponsoring Organizations of the Treadway Commission)** ist eine freiwillige privatwirtschaftliche Organisation in den USA, die helfen soll, Finanzberichterstattungen durch ethisches Handeln, wirksame interne Kontrollen und gute Unternehmensführung qualitativ zu verbessern.

### DMS - Dokumentenmanagement

**Dokumentenmanagement** -Systeme (**DMS**) dienen der datenbankgestützten Verwaltung elektronischer Dokumente. Verwaltung von Dateien mit Checkin/Checkout, Versionierung und anderen Funktionen.

Gilt als eine Komponente des ECM.

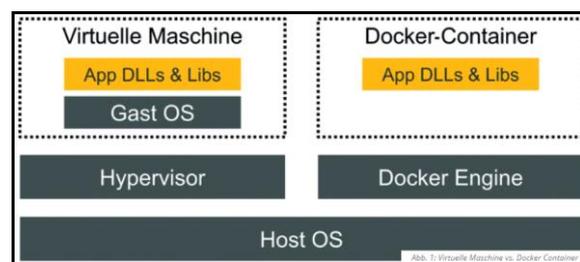
**Enterprise Application Integration (EAI)** bzw. **Unternehmensanwendungsintegration (UAI)** ist ein Konzept zur unternehmensweiten Integration der Geschäftsfunktionen entlang der Wertschöpfungskette, die über verschiedene Applikationen auf unterschiedlichen Plattformen verteilt sind, und die im Sinne der Daten- und Geschäftsprozessintegration verbunden werden können.

Die unterschiedlichen Methoden sind:

Datenintegration	➔	Enterprise Bus,
Anwendungsintegration	➔	Message Broker
Prozessintegration	➔	Prozessmanagement-Tool

### Docker - Container

- Docker ist eine Freie Software zur Isolierung von Anwendungen mit **Containervirtualisierung**.
- Docker vereinfacht die Bereitstellung von Anwendungen, weil sich Container, die **alle nötigen Pakete enthalten**, leicht als Dateien transportieren und installieren lassen.
- Container gewährleisten die Trennung und Verwaltung der auf einem Rechner genutzten Ressourcen.
- Das beinhaltet laut Aussage der Entwickler: **Code, Laufzeitmodul, Systemwerkzeuge, Systembibliotheken** – alles was auf einem Rechner installiert werden kann.



## Drools -

- Drools is a Business Rules Management System (BRMS) solution.
- It provides a core Business Rules Engine (BRE), a web authoring and rules management application (Drools Workbench), full runtime support for Decision Model and Notation (DMN) models at Conformance level 3 and an Eclipse IDE plugin for core development.
- Drools is open source software, released under the Apache License 2.0.
- It is written in 100% pure Java™, runs on any JVM and is available in the **Maven Central repository** too.

## DTU - Database Transaction Unit

- ???

## ECM - Enterprise-Content-Management

**Enterprise-Content-Management (ECM)** umfasst die Technologien zur Erfassung, Verwaltung, Speicherung, Bewahrung und Bereitstellung von Content und Dokumenten zur Unterstützung von organisatorischen Prozessen im Unternehmen.

- Revisionssichere Archivierung
- Sicherstellung der langfristigen Lesbarkeit
- Digitale Aktenbildung
- E-Mail Archivierung
- Versionsmanagement

## Enterprise Collaboration

- See: Circuit by Unify, IBM Connect, Beezy, Slack, Microsoft Teams

## Microsoft Teams

- Organisiert in «**Channels**».
- Externe können eingeladen werden, wenn diese ein **Live ID** haben.

## EOS - End of Support

- End of support means there will be **no further updates**, including regular **security updates**.
- With the increasing sophistication and frequency of cyberattacks, running apps and data on unsupported versions of Windows Server and SQL Server can create significant security and compliance risks.
- Moving to **Azure**, will provide free extended security updates for the Azure-hosted versions for **up to 3 years** after the end of support for the stand-alone versions.

**OS: Windows Server 2008 and 2008 R2**

End of support 14 January 2020

**OS: Microsoft Windows Server 2003 (W2K3)**

End of support July 2015

**IBM Websphere**

End of support September 2019

**SQL Server 2008 and 2008 R2**

End of support 9 July 2019

## Foreman

- Foreman (also known as The Foreman) is an open source complete life cycle systems management tool for provisioning, configuring and monitoring of physical and virtual servers.
- Foreman has deep integration to configuration management software, with **Ansible**, **Puppet**, **Chef**, **Salt** and other solutions through plugins, which allows users to automate repetitive tasks, deploy applications, and manage change to deployed servers.
- Foreman provides provisioning on **bare-metal** (through managed DHCP, DNS, TFTP, and PXE-based unattended installations), virtualization and cloud. Foreman provides comprehensive, auditable interaction facilities including a web frontend, a command line interface, and a robust REST API.

## Funktion

**Prozess** → Instanz  
**Funktion** → Klasse

Eine Funktion ist der Bauplan eines Prozesses. Ein Prozess ist die Ausführung der Funktion.

Betrieblich gesehen ist die Funktion der Aufbauorganisation zuzuordnen. Der Prozess der Ablauforganisation.

Meine Funktion als Vater ist die Erziehung, aber der Prozess ist ein schwieriger.

Ich habe eine Funktion, ich stelle aus den Teilen X und Y das Teil Z her. Sind die Teile X und Y da, so führe ich den Prozess aus und erstelle Z.

## Galvanische Trennung

- Unter galvanischer Trennung (auch galvanische Entkopplung) versteht man das Vermeiden der elektrischen Leitung zwischen zwei Stromkreisen, zwischen denen Leistung oder Signale ausgetauscht werden sollen.

## GitHub

- GitHub ist ein netzbasierter Dienst zur **Versionsverwaltung für Software-Entwicklungsprojekte**.
- Namensgebend war das Versionsverwaltungssystem **Git**.
- Das Unternehmen GitHub, Inc. hat seinen Sitz in San Francisco in den USA.
- Ähnliche Dienste sind **GitLab** und **Bitbucket**.
- Seit 26. Dezember 2018 gehört das Unternehmen zu **Microsoft**.
- Executes **Code Scanning/Vulnerability Scans!**

## GitLab

- GitLab entstand als Webanwendung zur **Versionsverwaltung** für Softwareprojekte auf Git-Basis.
- Hinzu kamen später Funktionen zur **Fehlersuche**, mit GitLab CI ein System zur kontinuierlichen Integration (CI), die Nutzerverwaltung, Runner, Wiki, Kanban-Board, Container-Registry, Sicherheitsscanner für Container und Sourcecode, Multi-Cluster-Verwaltung und -Überwachung sowie einige Funktionen mehr.
- GitLab ist in Produkte für Entwickler integrierbar und selbst über eine API fernsteuerbar.
- GitLab ist in den Programmiersprachen **Ruby** und **Go** entwickelt.

## GMC - Grandmaster Clock

- Grandmaster clocks receive UTC-based time information from an external time reference, most commonly a GNSS satellite source.
- This is then distributed downstream to other clocks.
- When successfully receiving a reference signal, the grandmaster derives accurate time from the reference.

### Products:

See: PTP <https://www.oscilloquartz.com>

## Grafana

- Grafana is the open source analytics and monitoring solution for every database.

## Grid Computing

- Grid computing is a form of parallel distributed processing that loosely groups a significant number of processing nodes to work towards a specific processing goal.

## IaaS - Infrastructure as-a-Service

Infrastructure as a Service is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. The client typically pays on a per-use basis.

- Takes the PaaS model yet another step forward and provides not just on-demand operating solutions but complete outsourcing options.
- Infrastructure as a Service is sometimes referred to as Hardware as a Service (**HaaS**).
- **IaaS** is one of three main categories of cloud computing service.

Characteristics and components of **IaaS** include:

- Utility computing service and billing model.
- Automation of administrative tasks.
- Dynamic scaling.
- Desktop virtualization.
- Policy-based services.
- @Internet connectivity.

### Examples IaaS:

- Amazon EC2 and S3
- Amazon Web Services (AWS)
- Terremark Enterprise Cloud
- Microsoft Azure
- Windows Live
- Windows SkyDrive
- Google Compute Engine
- Go grid
- Sungrid

## IACS - Industrial Automation and Control Systems

- Includes control systems used in **manufacturing** and **processing plants** and **facilities**, **building environmental control systems**, geographically dispersed operations such as utilities (i.e., electricity, gas, and water), pipelines and petroleum production and distribution facilities, and other industries and applications such as **transportation networks**, that use automated or remotely controlled or monitored assets.

## IDaaS - Identity-as-a-Service

- IDaaS provides **SSO** for the cloud.
- Microsoft provides **IDaaS for Office 365**.
- **Atos Evidian IDaaS** – available from the cloud on a subscription basis.

## IPaaS - Integration Platform as a Service

- In der Praxis ist iPaaS im Grunde genommen eine Teilmenge oder eine spezielle Form von PaaS.
- IPaaS beschreibt eine allgemeine Klasse von Cloud-Integrationsdiensten.
- Diese iPaaS-Angebote arbeiten alle zusammen, um Workloads und Datensätze von unterschiedlichen Tools und Plattformen, die lokal oder in der Cloud bereitgestellt werden, nahtlos zu verbinden.

## IPAM - IP Address Management



## VitalQIP

- Alcatel-Lucent

## BT Diamond IP

## Infoblox

## BlueCat

## Men&Mice

## SolarWinds IP Address Manager

## LightMesh

## IDEAL Model

- The Software Engineering Institute (SEI) also developed the IDEAL model for software development, which implements many of the **SW-CMM** attributes.

The five phases of IDEAL:

1: Initiating

- 2: Diagnosing
- 3: Establishing
- 4: Acting
- 5: Learning

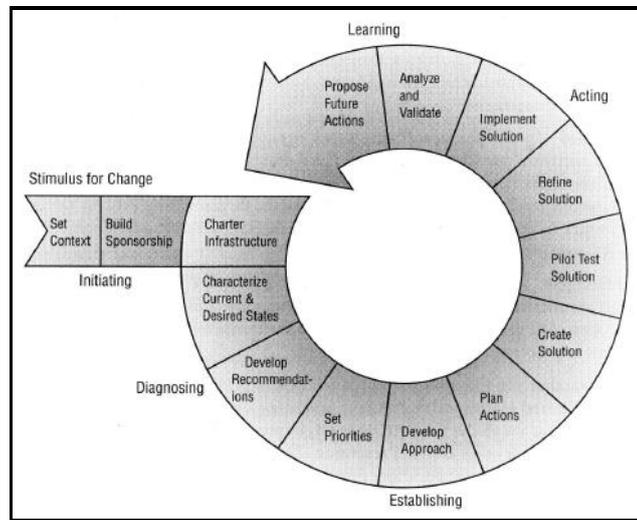


Figure 4: IDEAL Model

## IRC - Internet Relay Chat

- Wurde in den 1980er Jahren von dem finnischen Studenten Jarkko Oikarinen entwickelt.
- Er benötigt eigene **Chat-Server**, diese Server sind meistens untereinander vernetzt.
- Zudem wird eine **Client-Software** benötigt, die entweder auf den Rechnern der chattenden Personen installiert ist oder aber über einen Browser gestartet wird, z. B. ein Java-Web-Client.
- Für die Steuerung des Clients werden spezielle **IRC-Kommandos** verwendet.

## ITSM - IT Service Management

An ITSM Product should cover:

- Incident-Management (z.B. Printer Druckt nicht)
- Problem-Management (Ursachenforschung: z.B. Applikationsausfall, mehrere User betroffen)
- Change-Management (z.B. FW-Rule Changes)
- Operation-Management (Processes, z.B. HR Onboarding/Offboarding)
- Self-Service Portal
- Service-Catalog incl. SLAs
- Asset Management (CMDB/CI)
- License-/Contract Management
- Knowledgebase
- Reporting
- Dashboards
- Virtual Chat
- Cloud or On-Premises Solution

## BMC Remedy Service Management Suite

- Smart Reporting
- Incident & Problem Management
- Knowledge Management
- Change Management

- Release Management
- Asset Management
- Configuration Management Database (CMDB)
- Virtual Chat

## Cherwell Service Management

- Incident, Problem, and Change Management
- Self-Service Portal
- Service Catalog
- Knowledge Base
- Reporting and Dashboards

<https://www.youtube.com/watch?v=7Nh--Qw3I9w>

<https://www.youtube.com/watch?v=24dN86qywGw&feature=youtu.be>

<https://www.youtube.com/watch?v=nDQWla9Tglk&feature=youtu.be>

[https://www.youtube.com/watch?time\\_continue=36&v=RwrtgMA3uNU&feature=emb\\_logo](https://www.youtube.com/watch?time_continue=36&v=RwrtgMA3uNU&feature=emb_logo)

### Vetriebspartner:

PLAIN IT AG, Militärstrasse 5, CH-3600 Thun

Kontakt: [inbox@plain-it.ch](mailto:inbox@plain-it.ch) T +41 33 224 01 24

## Freshservice

- Cloud Solution
- Incident Management
- Asset Management
- Release Management
- Service Catalog

### Agents:

- Ein Vollzeit-Agent ist jemand, der Teil Ihres zentralen Service-Desk-Teams ist..

### Assets:

- Jedes von der Sonde oder dem Agenten entdeckte Asset wird als verwaltetes Asset betrachtet.
- Dasselbe gilt für Assets, die mit anderen Methoden hinzugefügt und später mit dem Discovery-Agent aktualisiert werden.

## OTRS

## KIX

## Servicenow

## ITIL - IT Infrastructure Library

Die **IT Infrastructure Library (ITIL)** ist eine Sammlung von *Best Practices* in einer Reihe von Publikationen, die eine mögliche Umsetzung eines IT-Service-Managements (ITSM) beschreiben und inzwischen international als De-facto-Standard hierfür gelten.

## JBoss EAP - JBoss Enterprise Application Platform

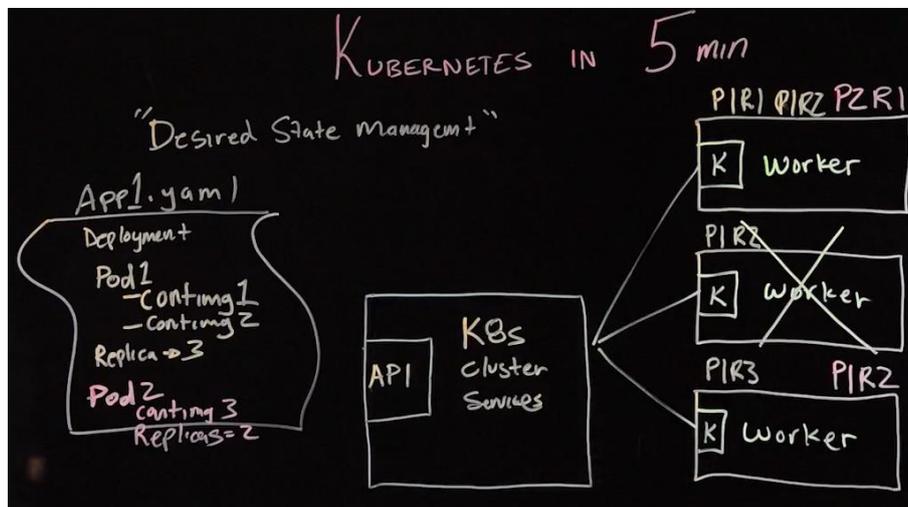
- **JBoss EAP** is a subscription-based/open-source Java EE-based application server runtime platform used for building, deploying, and hosting highly-transactional Java applications and services.

## Jenkins

- Jenkins ist ein erweiterbares, webbasiertes Software-System zur **kontinuierlichen Integration (CI)** von Komponenten zu einem Anwendungsprogramm.

## KUBERNETES

- Ursprünglich entworfen von **Google**.
- **Open Source** zur Automatisierung der Bereitstellung, Skalierung und **Verwaltung von Container Anwendungen** einschliesslich **Docker**.
- Master/Slave Architektur
- Auch als "**K8s**" bezeichnet
- Kubernetes orchestriert sogenannte «**Pods**» als **kleinste deploybare Einheit**.



## MDM - Mobile Device Management

- See: MobileIron, MS Intune Mobile

## Microsoft Intune

- Uses Azure Active Directory (Azure AD) as a directory store for identity, and it can integrate with local management infrastructures such as Microsoft Endpoint Configuration Manager.
- Intune is especially useful for devices that are beyond the management scope of Group Policy, such as mobile phones, devices that are not Active Directory Domain Services (AD DS) domain members, or Windows 10 devices that are joined to Azure AD.

## Microservices

- In a microservices architecture, the application is composed of small, independent services.
- Each microservice executes a single business capability.
- A microservice is small enough that a single small team of developers can write and maintain it.
- Microservices run in separate processes, communicating through well-defined APIs or messaging patterns.
- Microservices do not share data stores or data schemas.
- Each microservice is responsible for managing its own data.

- Microservices have separate code bases, and do not share source code.
- They may use common utility libraries, however.
- Each microservice can be deployed and updated independently of other services.

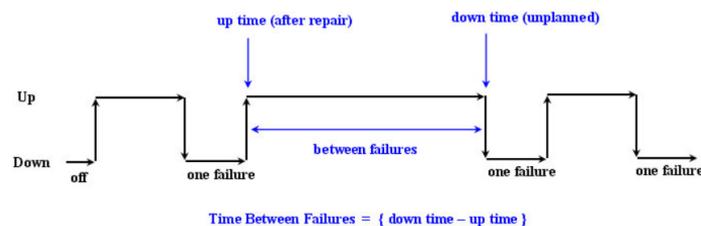
### Deutsch

- Microservices sind ein Architekturmuster der Informationstechnik, bei dem komplexe Anwendungssoftware aus unabhängigen Prozessen komponiert wird, die untereinander mit sprachunabhängigen Programmierschnittstellen kommunizieren.
- Die Dienste sind weitgehend entkoppelt und erledigen eine kleine Aufgabe.
- So ermöglichen sie einen modularen Aufbau von Anwendungssoftware.

## MTBF - Mean Time Between Failures

- Mittlere Zeit bis zum Defekt.
- Nach **IEC 60050** (191): Der Erwartungswert der Betriebsdauer zwischen zwei aufeinanderfolgenden Ausfällen.
- Ein **Mass** für die Zuverlässigkeit von Einheiten (Baugruppen, Geräten oder Anlagen).
- Je **höher** der MTBF-Wert, desto zuverlässiger ist das Gerät.

**Formel:**  $MTBF = (\text{Gesamtbetriebszeit}) / (\text{Anzahl der Ausfälle})$



## MTTF - Mean Time to Failure

- Die mittlere Betriebsdauer bis zum Ausfall und wird auch als **mittlere Lebensdauer** bezeichnet.
- Je **höher** der MTTF-Wert, desto zuverlässiger ist das Gerät.
- Siehe auch: **EN ISO 13849-1**

## MTTR - Mean Time to Repair

- Die mittlere **Reparaturzeit** nach einem Ausfall eines Systems.
- Je **niedriger** der MTTR-Wert, desto besser.

**Formel:**  $MTTR = (\text{Gesamtauszeit}) / (\text{Anzahl der Ausfälle})$

## MTR - Mean Time to Recovery

- ???

## Nexus

- See: <https://de.sonatype.com/>
- Der weltweit führende **Repository Manager**.
- Eine einzige verlässliche Informationsdatenquelle für all Ihre Komponenten, Binärdateien und Build-Artefakte.
- Verteilen Sie Komponenten und Container effizient an Entwickler.

## Nginx

- Nginx is a **web server** which can also be used as a **reverse proxy, load balancer, mail proxy** and **HTTP cache**.
- The software was created by **Igor Sysoev** and first publicly released in 2004.
- A company of the same name was founded in 2011 to provide support and Nginx plus paid software

## NMock2

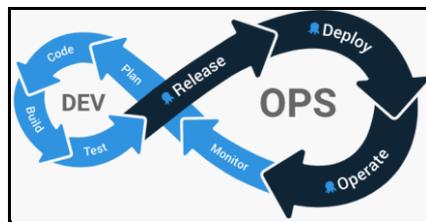
**NMock2** is a library for assisting test driven development of .NET code by providing a dynamic Mock Object framework for .NET interfaces.

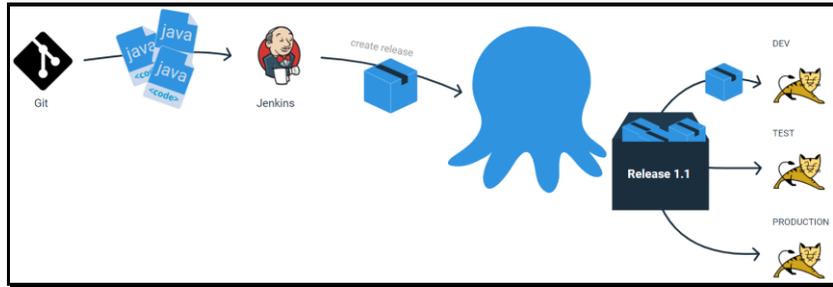
## NVMe

- NVM Express (kurz NVMe) ist eine erstmals im Jahr 2011 veröffentlichte Softwareschnittstelle (also ein Protokoll), um SSD, also nichtflüchtige Massenspeicher (engl. nonvolatile memory, kurz NVM), über PCI Express zu verbinden, ohne dass dafür herstellerspezifische Treiber nötig wären.
- Sie soll besonders bei parallelen Zugriffen, wie sie bei Multithreading häufig vorkommen, die Geschwindigkeit erhöhen, indem die Latenz und der Overhead durch die Befehle verringert werden.
- NVMe ist damit bei den Massenspeicherschnittstellen in einer Reihe mit **SCSI** und **AHCI** zu nennen, da sie aus der Sicht des Computers generische Softwareschnittstellen sind, um unterschiedliche Massenspeicher ohne eigene Treiber anzusprechen.
- Treibende Kraft hinter dem Standard ist Intel, daneben sind die Hersteller von Speicherlösungen wie NetApp, Dell, Cisco, EMC, Marvell und die Speicherhersteller SanDisk, Avago Technologies, Micron Technology, HGST, Samsung und Seagate involviert.
- Ursprünglich wurde NVMe vornehmlich für Server entwickelt, mittlerweile wird die Technik auch bei Arbeitsstationen genutzt. Deshalb sind mittlerweile sehr unterschiedliche Bauformen von NVMe-Geräten erhältlich:

## Octopus Deploy

- See: <https://octopus.com>
- Release management
- Deployment automation
- Operations runbooks





## OODB - Object-Oriented Database

- An OODB is more dynamic than a relational database as it stores data as objects.
- It allows object-oriented programming (OOP) code, to manipulate the objects.
- This makes the reuse of code possible.

## OpenShift - Container

- OpenShift ist ein Produkt von **Red Hat**.
- You must install it on Red Hat Linux other Linux distributions are not supported
- Container-Anwendungsplattformen für Cloud-Computing.

## OVF - Open Virtualization Format

- Is an open standard for **packaging** and **distributing virtual appliances** or, more generally, software to be run in virtual machines.

## PaaS - Platform-as-a-Service

- The primary attraction of PaaS is the avoidance of having to purchase and maintain high-end hardware and software locally.
- **PaaS** is defined as a computing platform being delivered as a service. Typically, the service facilitates the deployment of applications, application development, testing, and also supports the building, testing and hosting of web applications.

### Examples:

- Microsoft Azure
- Force.com
- Google App Engine
- Intel MashMaker

## PLM - Product Lifecycle Management

**Product Lifecycle Management (PLM, „Produktlebenszyklusmanagement“)** bezeichnet ein IT-Lösungssystem, mit dem alle Daten, die bei der Entwicklung, Produktion, Lagerhaltung und dem Vertrieb eines Produkts anfallen, einheitlich gespeichert, verwaltet und abgerufen werden. Im Idealfall greifen alle Bereiche bzw. Systeme, die mit einem Produkt in Berührung kommen, auf eine gemeinsame Datenbasis zu: Von der Planung (PPS/ERP), Konstruktion (CAD), Berechnung (CAE) und Fertigung (CAM) bis zum Controlling, Vertrieb und Service.

## PSP - Personal Software Prozess

Der **Team Software Prozess (TSP) / Personal Software Prozess (PSP)** ist eine Methode für Softwareentwicklungsteams zur Selbstoptimierung.

Sie wurde von Watts S. Humphrey am Software Engineering Institute (SEI) an der Carnegie Mellon University/Pittsburgh entwickelt, um den Software-Entwicklern eine Methode an die Hand zu geben, mit der sie die Anforderungen des Capability Maturity Model (CMM) konkret umsetzen können.

## Puppet

- **DevOps** automation for a multi-cloud world.
- Puppet spans across your hybrid environments to automate infrastructure and application workflows and ongoing management.
- Puppet provides the ability to define **which software** and **configuration a system requires** and then maintain a specified state after an initial setup.
- You use a declarative Domain Specific Language (DSL) that is **similar to Ruby** to define configuration parameters for a specific environment or infrastructure.

### Versions

- **Puppet Enterprise** (For Free)
- **Puppet Enterprise Standard**  
Every license includes Standard support from our team during local business hours, typically from the team in your geo. This plan gives you access to hundreds of current knowledge base articles and the ability to open support cases at our support portal.

Standard support includes:

Open to 5 support cases per month at our support portal  
Business hour service level agreement (SLA) response times\*:  
Priority 1: 1 hour  
Priority 2: 4 hours  
Priority 3: 12 hours  
Priority 4: Next business day  
Business hours: Monday – Friday 8am – 6pm in your time zone  
Up to 4 technical contacts

- **Puppet Enterprise Premium**  
Premium support is the best choice if your business is global, spans multiple geos, or is critical and requires around the clock support. This plan gives you access to hundreds of current knowledge base articles and the ability to open support cases at our support portal.

Premium support includes everything available in Standard support plus extras:

Open unlimited cases at our support portal  
24 x 7 x 365 service level agreement (SLA) response times\*:  
Priority 1: 1 hour  
Priority 2: 4 hours  
Priority 3: 12 hours  
Priority 4: Next business day  
24 x 7 x 365 phone support for Priority 1 issues  
Unlimited technical contacts  
Training vouchers: Two vouchers for Puppet Enterprise customers with under 500 nodes, four vouchers for Puppet Enterprise customers with 500 nodes and above

## Declarative

- Tell the system **WHAT** you would like to happen and let it figure out **HOW** to do it.

## Imperative

- Tell the system HOW to do something and as a result WHAT you want to happen.

## RAD - Rapid Application Development

- The Rapid Application Development model is a software development model or methodology that relies on the use of rapid prototyping and enables organizations to develop strategically important systems faster while reducing development costs and maintaining quality.

## RFB - Request for Bid

### RFI - Request for Information

- Provides the opportunity to collect information about the capabilities of suppliers.
- This approach is used when an organization lacks the details required to document an RFP.
- Instead, the organization uses the RFI process to learn about the capabilities of vendors and then chooses a vendor who has the capability to work with the organization to define the solution.

### RFP - Request for Proposal

- The request for proposal (RFP) is generally regarded as a **best practice** designed to assist in the procurement of services.
- RFP is beneficial because it facilitates a bidding process that may drive down prices or increase value as vendors compete to win the bid.
- Processes used to create RFPs should outline the bidding process, contract terms, and provide guidance on how the bid should be formatted and presented.
- This presents a good opportunity for the organization facilitating the RFP to include security requirements in the proposal and evaluate the capability of vendors to meet those requirements.

## RFQ - Request for Quote

## RUP - Rational Unified Process

Der **Rational Unified Process** (RUP) ist ein objektorientiertes Vorgehensmodell zur Softwareentwicklung und ein kommerzielles Produkt der Firma Rational Software, die seit 2002 Teil des IBM Konzerns ist. IBM entwickelt den RUP und die zugehörige Software weiter. Die 9. Version ist die derzeit (2006) aktuelle Version. Der RUP benutzt die Unified Modeling Language (UML) als Notationssprache. Der RUP wurde von Philippe Kruchten in seiner Urform erstmals 1996 vorgestellt.

## SaaS - Software-as-a-Service

- SaaS provides on-demand online access to specific software applications or suites without the need for local installation.

### **Examples**

- Google Docs
- Calendar
- Salesforce CRM

## Satellite - Red Hat Satellite

- Infrastructure Configuration Management

## SCCM - System Center Configuration Manager

- Ist ein Software-Produkt aus der System Center 2012 Suite von Microsoft. Es löst frühere Versionen mit dem Namen Systems Management Server (SMS) ab.
- Unter (IT) Systems Management versteht man allgemein die zentralisierte Verwaltung von Hard- und Software innerhalb eines Unternehmens.
- Systems Management bildet die Obermenge des Client-Managements.

**Folgende Aufgaben können von SCCM automatisiert für eine sehr grosse Anzahl von Clients durchgeführt werden:**

- Inventarisierung
- Softwareverteilung
- System-Update-(Patch-)Management
- Fernwartung
- Lizenzüberwachung
- Reporting
- Anti-Schad-Software-Management

## SCOM - System Center Operations Manager

- SCOM can record and analyze performance information about physical computers and virtual machines for both consolidation and virtualization.

## SCVMM - System Center Virtual Machine Manager

The SCVMM console is an administrative console that helps you manage the Windows virtual machine infrastructure efficiently. Using the console, creating and configuring virtual machines becomes simpler. Another important aspect to consider when you create a virtual environment is to place the virtual machines at optimal locations. Also, after the virtual environment is configured, you need to follow guidelines that help you implement management tasks, such as using SCOM for measuring performance data of virtual machines and implementing pilot virtualization on the virtual environment.

SCVMM also uses Windows PowerShell giving administrators the flexibility to manage the virtual network by using either the GUI of VMM Administrator Console or the Windows PowerShell command-line interface

Physical-to-virtual (P2V) machine conversions

## SDLC - System Development Life Cycle

- The SDLC is a **methodology** that forms the framework for planning and controlling the creation, testing, and delivery of an information system.

### Phase: System Investigation

### Phase: System Analysis

### Phase: Design

- Product Design.
- The design stage takes as its initial input the requirements identified in the approved requirements document, this would include security specifications.

## Phase: Environments

## Phase: Testing

## Phase: Training and transition

## Phase: Operations and Maintenance

## Phase: Evaluation

## SERVERLESS COMPUTING

- Serverless computing is a **cloud-hosted execution environment** that runs your code but completely abstracts the underlying hosting environment.
- You create an instance of the service, and you add your code; no infrastructure configuration or maintenance is required, or even allowed.

## Sharding

- Is a technique to distribute large amounts of identically **structured data** across several independent databases.

### RECAP

**Structured Data** is organized data that neatly fits into rows and columns in tables.

**Semi-Structured Data** is still organized and has clear properties and values, but there's variety to the data.

**Unstructured data** doesn't fit neatly into tables, nor does it have a schema.

## Slack

- Enterprise Collaboration
- Slack ist ein webbasierter **Instant-Messaging-Dienst** des US-amerikanischen Unternehmens Slack Technologies zur Kommunikation innerhalb von Arbeitsgruppen (Groupware).
- Mit Slack können **Nachrichten ausgetauscht werden**, und das **Chatten** mit Einzelpersonen oder in einer Gruppe wird ermöglicht.
- Die gemeinsame **Bearbeitung von Dokumenten** ist nur möglich, wenn man bei weiteren externen Diensten wie **Dropbox**, **Google Drive** oder **GitHub** ein Konto und eine aktivierte Anmeldung hat.
- Direkt innerhalb von Slack ist eine gemeinsame Dokumentenbearbeitung (nativ) nicht möglich.
- Zusätzlich zur Webanwendung gibt es **Clients** für die Betriebssysteme Windows, macOS, Linux, Android und iOS.

## SOA - Serviceorientierte Architektur

**Serviceorientierte Architektur (SOA)**, engl. **service-oriented architecture**, auch **dienstorientierte Architektur**, ist ein Ansatz der Informationstechnik aus dem Bereich der verteilten Systeme, um Dienste von Mitarbeitern und Organisationen zu strukturieren und zu nutzen. Eine besondere Rolle spielt dabei die Orientierung an **Geschäftsprozessen**. Vereinfacht könnte man SOA als Methode ansehen, die vorhandenen EDV-Komponenten wie Datenbanken, Server und Websites so in Dienste zu **kapseln** und dann zu koordinieren („Orchestrierung“), dass ihre Leistungen zu höheren Diensten zusammengefasst und anderen Organisationsabteilungen oder Kunden zur Verfügung gestellt werden können.

## SPICE - Software Process Improvement and Capability Determination

**SPICE** (Software Process Improvement and Capability Determination) oder ISO/IEC 15504 ist ein internationaler Standard zur Durchführung von Bewertungen (Assessments) von Unternehmensprozessen mit Schwerpunkt auf der Softwareentwicklung.

## System-Architect

### System-Architect in SE

- Schnittstelle zwischen den Benutzern, Sponsoren und allen Stakeholdern um Anforderungen aufzunehmen
- Systemanforderungen definieren auf hohem Niveau und definieren der Bedingungen, Kosten und Zeitrahmen
- Sicherstellen das die Anforderungen Konsistent, korrekt und operationell klar sind
- Kosten-/Nutzen Analysen erstellen um make or buy Entscheidungen treffen zu können, dabei alle bereits vorhandenen Komponenten nutzen
- Systemprozessgruppen definieren die einzelnen Engineering Teams zugewiesen werden können
- Problemkoordination zwischen den verschiedenen System-Engineering Gruppen.
- Eine stabile, robuste I-Architektur gewährleisten und entwickeln
- Testkonzepte entwickeln

## TDD

**TDD**: Als **testgetriebene Entwicklung** (auch *testgesteuerte Programmierung*, engl. *test first development* oder *test-driven development (TDD)*, manchmal auch scherzhaft Extreme Testing) ist eine Methode, die häufig bei der agilen Entwicklung von Computerprogrammen eingesetzt wird. Bei der testgetriebenen Entwicklung erstellt der Programmierer Software-Tests konsequent vor den zu testenden Komponenten. Die dazu erstellten Unit-Tests sind Grey-Box-Tests statt klassischer White-Box-Tests.

## Terraform

- Infrastructure automation tool

## TFS - Team Foundation Server

- ???

## VDI - Virtualized Desktop Infrastructure

- ???

## Virtual Kubelet

- Is an open-source Kubernetes kubelet implementation that masquerades as a kubelet.
- This allows Kubernetes nodes to be backed by Virtual Kubelet providers such as serverless cloud container platforms.
- You can use Virtual Kubelet to connect Azure Container Instances into your Kubernetes environment, including AKS.
- With Virtual Kubelet, when your Kubernetes cluster demands additional container instances, those demands can be met from ACI.
- Since ACI is serverless, there is no need to have reserved capacity.
- You can therefore take advantage of the control and flexibility of Kubernetes scaling with the per-second-billing of serverless.

- At time of writing, the Virtual Kubelet is described as **experimental software** and should not be used in production scenarios.

## VMWare ESXi

Link: [www.vmware.com](http://www.vmware.com)

- Serving virtual computers

## UML - Unified Modeling Language

Die **Unified Modeling Language**, kurz **UML** (zu deutsch „Vereinheitlichte Modellierungssprache“), ist eine von der Object Management Group (OMG) entwickelte und standardisierte Sprache für die Modellierung von Software und anderen Systemen. Sie ist auch über die ISO standardisiert (ISO/IEC 19501). Im Sinne einer Sprache definiert die UML dabei Bezeichner für die meisten Begriffe, die für die Modellierung wichtig sind, und legt mögliche Beziehungen zwischen diesen Begriffen fest. Die UML definiert weiter graphische Notationen für diese Begriffe und für Modelle von statischen Strukturen und von dynamischen Abläufen, die man mit diesen Begriffen formulieren kann .

## XP - eXtreme Programming

### **eXtreme Programming**

eXtreme Programming (XP) stellt die bekannteste agile Methode dar und bietet sowohl Managementtechniken als auch Techniken auf der Team- und Programmierenebene. Hervorzuheben ist insbesondere die testgetriebene Entwicklung als Teil des inkrementellen Designs und das Programmieren in Paaren.

## YAML - Ain't Markup Language

- Is a relatively new **data language** that's growing quickly in popularity in part due to its human friendliness.
- The data structure is defined by line separation and indentation, and reduces the dependency on structural characters like parentheses, commas and brackets.

# TOOLS

## Abbildungsverzeichnis

Strömungsgrößen .....	11
Blackbox-Betrachtung .....	11
Prozessanalyse .....	12
Teilproblemlösung .....	12
Zinnmodelle .....	12
Die Stufen des ISM.....	14
Lebensdauer der Elemente .....	15
IS-Architektur .....	16
Geschäftsfeld/IS-Architektur.....	16
IS-Architektur des Integrationsbereiches .....	16
Wasserfallmodell .....	17
Phasen des IEM .....	19
Prototypingmodell.....	20
Verlagerung des Aufwandes .....	20
SE-Philosophie .....	24
SE-Methodik .....	24
Problem .....	24
Systemdenken .....	24
System - Zoom .....	25
Vorgehensmodell.....	26
Lebensphasen eines Systems.....	26
Problemlösungszyklus.....	30
Systemgestaltung .....	31
Vorgehensmodell.....	31
Situationsanalyse zur Zielformulierung und Synthese/Analyse .....	32
Zielformulierung.....	32
Synthese & Analyse .....	33
Projekt-Management im SE.....	35
Prozessor.....	36
Systementwicklungsprozess .....	37
Datenspeicher.....	37
Aktivitätendiagramm .....	41
Blackbox-Betrachtung .....	41
Entscheidungstabelle .....	41
Entscheidungstabelle verdichtet.....	42
Datenstrukturdiagramm (DSD) .....	42
Entscheidungsmatrix .....	43
Entity-Relationship-Modell (ERM) .....	43
Mengengerüst-Tabelle.....	44
Projektauftrag .....	45
CMM + IDEAL.....	53

## Index

- Ablaufdiagramme**, 25
- agile Methode, 73
- ARIS-Methode**, 43
- BI*, 49
- Blackbox-Betrachtung**, 25
- BPM*, 49
- CMM, 53
- CMMI, 53
- CobiT**, 54
- CORBA**, 55
- COSO, 56
- Datenflussdiagramme**, 36
- Datenflussplan, 42
- DFD, 38
- DFP, 42
- DMS, 56
- DSD, 42, 43
- EAI**, 56
- ECM, 57
- Elemente**, 11
- EPK, 43
- Ereignisgesteuerte Prozesskette, 43
- essentiellen Aktivitäten**, 36
- Essenz**, 36
- eXtreme Programming, 73
- HaaS, 59
- Inkarnation**, 37
- IS-Architektur**, 15
- ISM, 14
- ISO/IEC 15504, 71
- ITIL, 53, 62
- Modell**, 25
- Modelle**, 12, 19
- NMock2**, 65
- OOA, 22
- PaaS, 66
- PLM**, 66
- Policy**, 36
- Potentialanalyse, 46
- PowerShell, 69
- Problem**, 13
- Prototypen**, 19
- Prozessor**, 36
- PSP, 66
- Rapid Specification Prototyping**, 20
- RUP, 68
- S3M**, 53
- SaaS, 68
- SAP**, 43
- SCCM, 69
- SCOM, 69
- SCVMM, 69
- SEP, 14
- SOA, 70
- SPICE, 71
- Strömungsgrößen**, 11
- Swimlanes**, 41
- SWOT-Analyse, 46
- System**, 24, 25, 27, 29
- Systemanalyse**, 36
- System-Archäologie**, 37
- System-Architect, 71
- TDD**, 71
- Teilsystem, 25, 29
- Top-Down, 11
- TSP, 66
- UAI**, 56
- UML, 23, 41, 72
- Untersystem, 25
- Wechselbeziehungen**, 11
- Wirkungspotentiale, 11