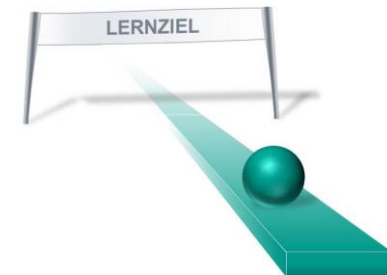


IT-Projektmanagement und Software Engineering:
1 Grundlagen

Dr. Holger Kramer

1	Grundlagen
2	Unternehmens- und Projektorganisation
3	Projekt- und Produktlebenszyklus von Software
4	Vorgehensmodelle in der Softwareentwicklung
5	Übergreifende Aufgaben des Managements
6	Projektentstehung
7	Projektdefinition
8	Projektdurchführung
9	Projektabschluss
10	Metriken und Messung
11	Reifegradmodelle und Prozessverbesserung
12	Werkzeuge

Ziel dieses Kapitels ist, die wesentlichen Aufgaben der Organisation und des Managements von Softwareprojekten und den Zusammenhang zu den unternehmerischen, fachlichen, methodischen und softwaretechnischen Aufgaben in Softwareprojekten zu beschreiben.



► **Definition (Software Engineering)**

Das Software Engineering stützt sich auf klassische Gebiete der Informatik und auch auf Aufgabenfelder der Betriebswirtschaftslehre [82]. Die zentralen Inhalte des Software Engineerings können dabei grob in die folgenden zwei Bereiche unterteilt werden:

1. **Fachliches, Technik und Methodik:** Zu den fachlichen, technischen und methodischen Aufgaben des Software Engineerings gehören im Einzelnen: Problemanalyse (Systemanalyse), Anforderungsspezifikation, Systementwurf, Softwarearchitektur, Implementierung, Integration, Qualitätssicherung und Test, Installation, Wartung und Weiterentwicklung.
2. **Organisation und Management von Softwareprojekten:** Dies umfasst: Projektorganisation und Projektmanagement, darunter insbesondere Projekt und Qualitätsplanung, Kostenschätzung und -kontrolle, Vertragsgestaltung, Vermarktung, sowie die Projekt und Qualitätsüberwachung und -steuerung.

Erkenntnis – Menschliche Faktoren sind dominant

Nicht selten scheitern Softwareprojekte nicht aus technischen sondern aus organisatorischen Gründen. Ungeachtet der geschickten Projektorganisation bleibt der wohl wichtigste und entscheidende Faktor für den Projekterfolg die Kompetenz und Professionalität der Mitarbeiter auf allen Ebenen der Hierarchie. Der Auswahl der Mitarbeiter und Führungskräfte (Einstellungspolitik), ihrer Motivation, Expertise, Weiterbildung und Schulung kommt nach DeMarco [58] eine zentrale Bedeutung zu.

Grundlagen

Aufbau und Kapitelstruktur des Basistexts

Teil 1: Grundlagen & Begriffserklärung	Teil 2: Management im Projektlebenszyklus	Teil 3: Weiterführende Themen	Anhang
Grundlagen	Übergreifende Aufgaben des Managements	Metriken und Messung	Projektunterlagen Code & Talk
Unternehmens- und Projektorganisation	Projektentstehung	Reifegradmodelle und Prozessverbesserung	Ergänzungen zu COCOMO/COCOMO II
Projekt- und Produktlebens-zyklus von Software	Projektdefinition	Werkzeuge	Ergänzungen zu CMMI
Vorgehensmodelle in der Software- entwicklung	Projektdurchführung		
	Projektabschluss		

Das Modul befasst sich mit der Organisation und dem Management von Softwareprojekten. Behandelt werden insbesondere die einzelnen Aufgaben während der

- Projektentstehung
- Projektdefinition
- Projektdurchführung und
- Projektabschluss

Diese Systematik kann zu typischen Standards im Bereich PM in Beziehung gesetzt werden:

PMBOK	Buch
Initializing	Kapitel 6 Projektentstehung
Planning	Kapitel 7 Projektdefinition
Executing & Controlling	Kapitel 8 Projektdurchführung
Closing	Kapitel 9 Projektabschluss

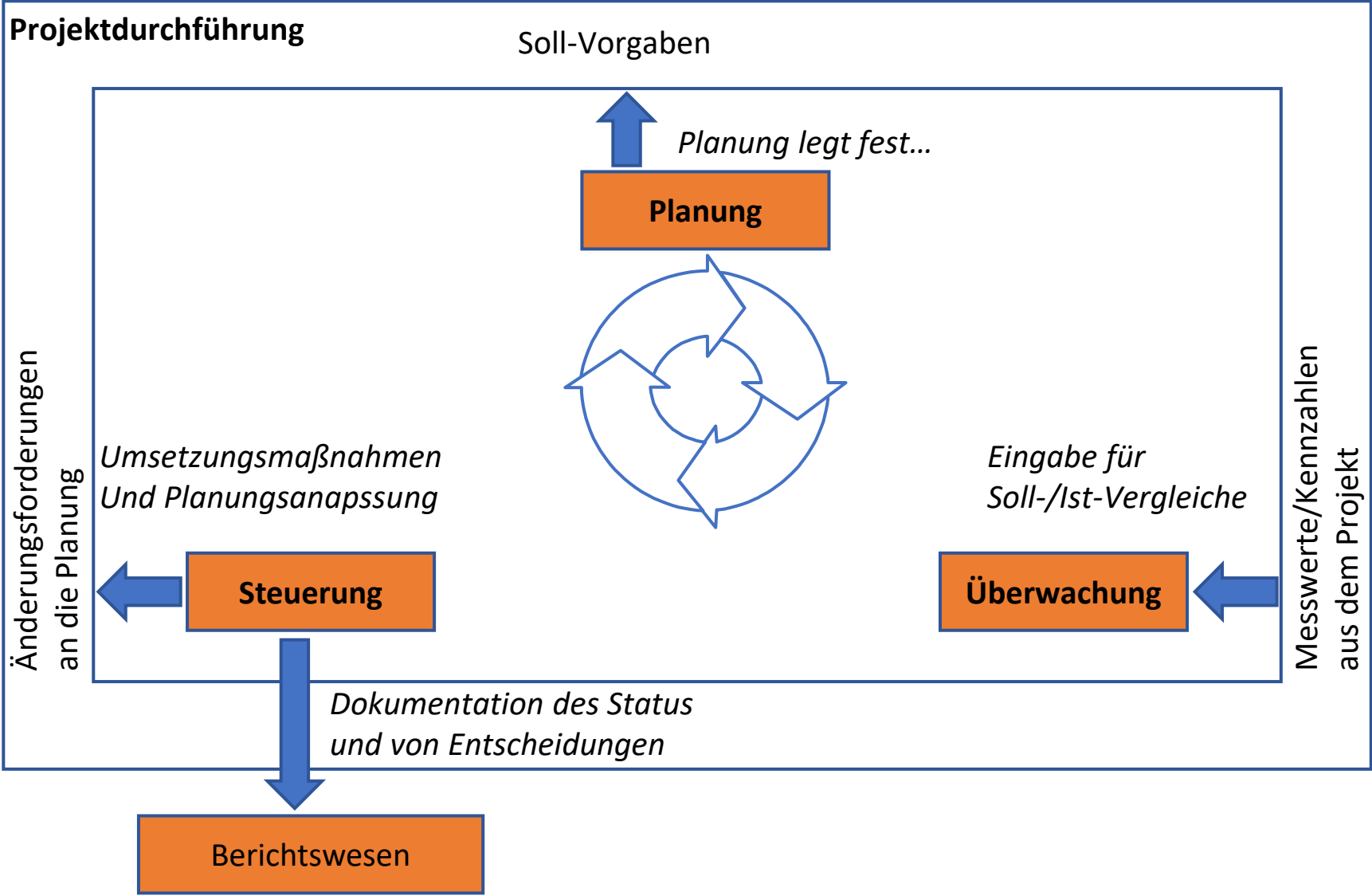
Broy, M., & Kuhrmann, M. (2013). Projektorganisation und Management im Software Engineering. Berlin, Heidelberg, New York: Springer, S.8

► **Definition (Management)**

Management (engl. to manage → italienisch: maneggiare, „handhaben“) kann sowohl Leitungsaufgaben in Projekten und Unternehmen bezeichnen, als auch die Gruppe der Personen, die diese Aufgaben ausüben und entsprechende Managementkompetenzen benötigen. Typische Aufgaben des Managements sind: Planung, Delegation, Organisation, Führung und Kontrolle (im Sinne von Fortschritts- und Erfolgskontrolle).

Die Gesamtheit aller Tätigkeiten, die mit der erfolgreichen Abwicklung eines Projektes zusammenhängen, mündet im sogenannten Regelkreis des Projektmanagements zur Überwachung und Steuerung von Projekten. Dieser Regelkreis wird in einem Projekt kontinuierlich durchlaufen.

Grundlagen
Regelkreis des Projektmanagements



Broy, M., & Kuhrmann, M. (2013). Projektorganisation und Management im Software Engineering. Berlin, Heidelberg, New York: Springer, S.10

► **Definitionen für „Projekt“**

► **Definition (Projekt, ISO)**

Ein Projekt ist nach [64] ein Vorhaben, bei dem innerhalb einer vorgegebenen Zeitspanne ein spezifiziertes Ziel erreicht werden soll und das sich dadurch auszeichnet, dass es im Wesentlichen ein einmaliges (individuelles) zeitlich begrenztes Vorhaben ist.

► **Definition (Projekt, PRINCE2)**

Ein Projekt ist nach [147] eine für einen befristeten Zeitraum geschaffene Organisation, die mit dem Zweck eingerichtet wurde, eine oder mehrere Erzeugnisse in Übereinstimmung mit einem vereinbarten Business Case zu liefern. Es zeichnet sich darüber hinaus durch eine fortlaufende wirtschaftliche Rechtfertigung aus.

► **Definition (Projekt, PMBOK)**

Ein Projekt ist nach [158] eine zeitlich beschränkte Anstrengung zur Erzeugung eines einmaligen Erzeugnisses (Produktes) oder Dienstes.

Verkürzt und vereinfacht ausgedrückt lässt sich ein Projekt wie folgt charakterisieren:

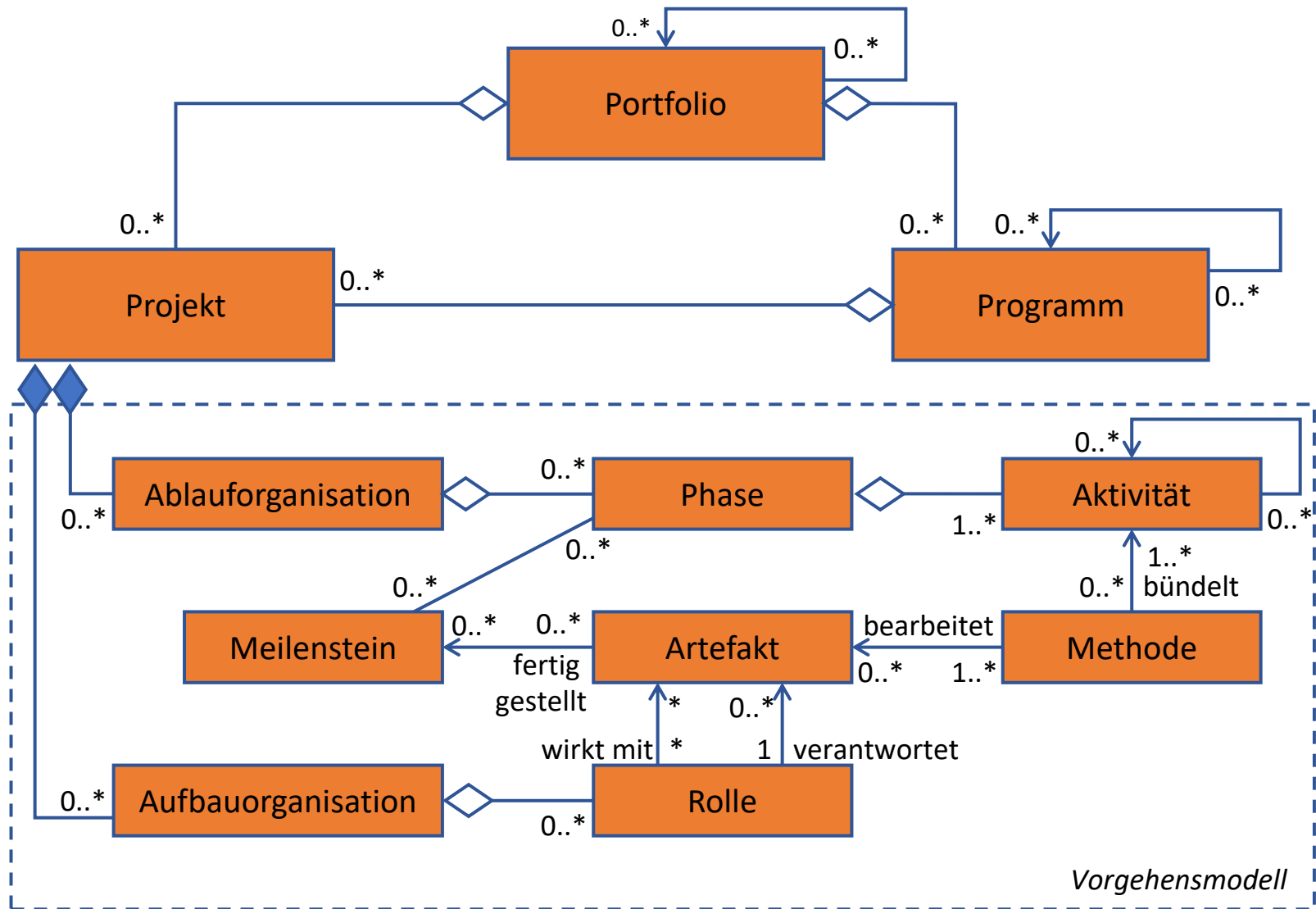
- Ein Projekt zielt auf die Lösung eines eigenständigen Problems.
- Ein Projekt zielt auf die Erarbeitung eines Ergebnisses.
- Ein Projekt hat einen Anfang und ein Ende – es ist zeitlich begrenzt.
- Ein Projekt hat ein klares Ziel.
- Ein Projekt ist ein einmaliges Vorhaben.
- Ein Projekt grenzt sich von anderen Vorhaben ab.
- Einem Projekt steht eine begrenzte Menge von Ressourcen zur Verfügung.
- Ein Projekt benötigt einen organisatorischen Rahmen, der alle Aufgaben unterstützt, um das Projektziel zu erreichen.

► **Definition (Artefakt)**

Ein Artefakt ist ein primäres oder sekundäres Arbeitsergebnis, das in einem Projekt durch bestimmte Projektaktivitäten erstellt, bearbeitet oder genutzt

Grundlagen

Ordnungsrahmen (Ontologie) für Projekte in Form eines Metamodells



Broy, M., & Kuhrmann, M. (2013). Projektorganisation und Management im Software Engineering. Berlin, Heidelberg, New York: Springer, S.12

Ein Artefakt kann durch folgende Aspekte gekennzeichnet werden

- sein Inhalt, Zweck
- durch wen und durch welche Aktivitäten es geschaffen wird
- durch wen und in welchen Aktivitäten es genutzt wird
- wie es repräsentiert wird (Papier, Datei, Inhalt einer Datenbank)
- wie es strukturiert ist (Gliederung)
- mit welchen Mitteln sein Inhalt dargestellt wird (Text, Formel, Diagramm)

Software Engineering ist eine Ingenieurdisziplin

Das Software Engineering weist mit dem Ziel der Entwicklung nutzbarer Produkte und Systeme alle Merkmale einer Ingenieurdisziplin auf, nämlich der Erreichung formalisierter technischer Zielsetzungen unter Einhaltung technischer und ökonomischer Rahmenbedingungen.

Allerdings gibt es eine Reihe signifikanter Unterschiede. Der wahrscheinlich wichtigste betrifft die Immaterialität von Software. Die Folgen dieses wesentlichen Unterschieds sind:

1. Es gibt nur den Entwicklungsprozess und die Inbetriebnahme und keinen wirklichen Produktions- bzw. Fertigungsprozess.
2. Es gibt Unterschiede in den Kostenstrukturen.
3. Vertriebs- und Wartungsstrukturen unterscheiden sich deutlich.
4. Software greift tief in menschliche, betriebswirtschaftliche und technische Prozesse ein und erfordert ein umfassendes Verständnis der Anwendungsfelder und der Kundenbedürfnisse.
5. Es gibt zusätzliche Chancen und Risiken durch zeitige Kundeneinbindung und Änderung von Kundenanforderungen während der Entwicklung.
6. Software ist ein abstraktes Artefakt. Die Erarbeitung und Analyse von Kundenanforderungen ist oft nur iterativ (manchmal auch nur experimentell) möglich.
7. Software beschreibt dynamisches Verhalten und unterliegt einer ständigen Evolution, womit sie eine inhärent hohe Komplexität aufweist.

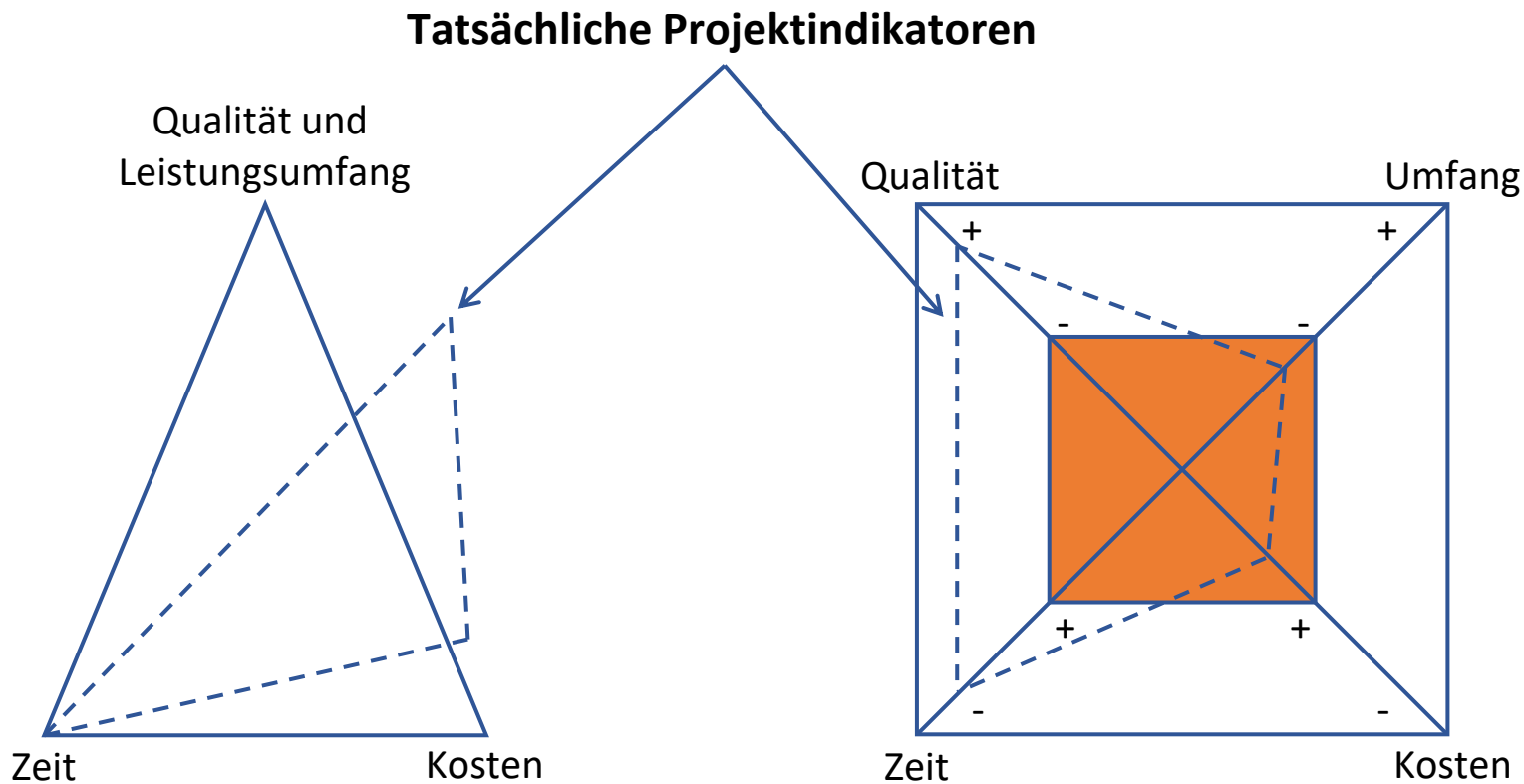
Ziele des Managements im Software Engineering

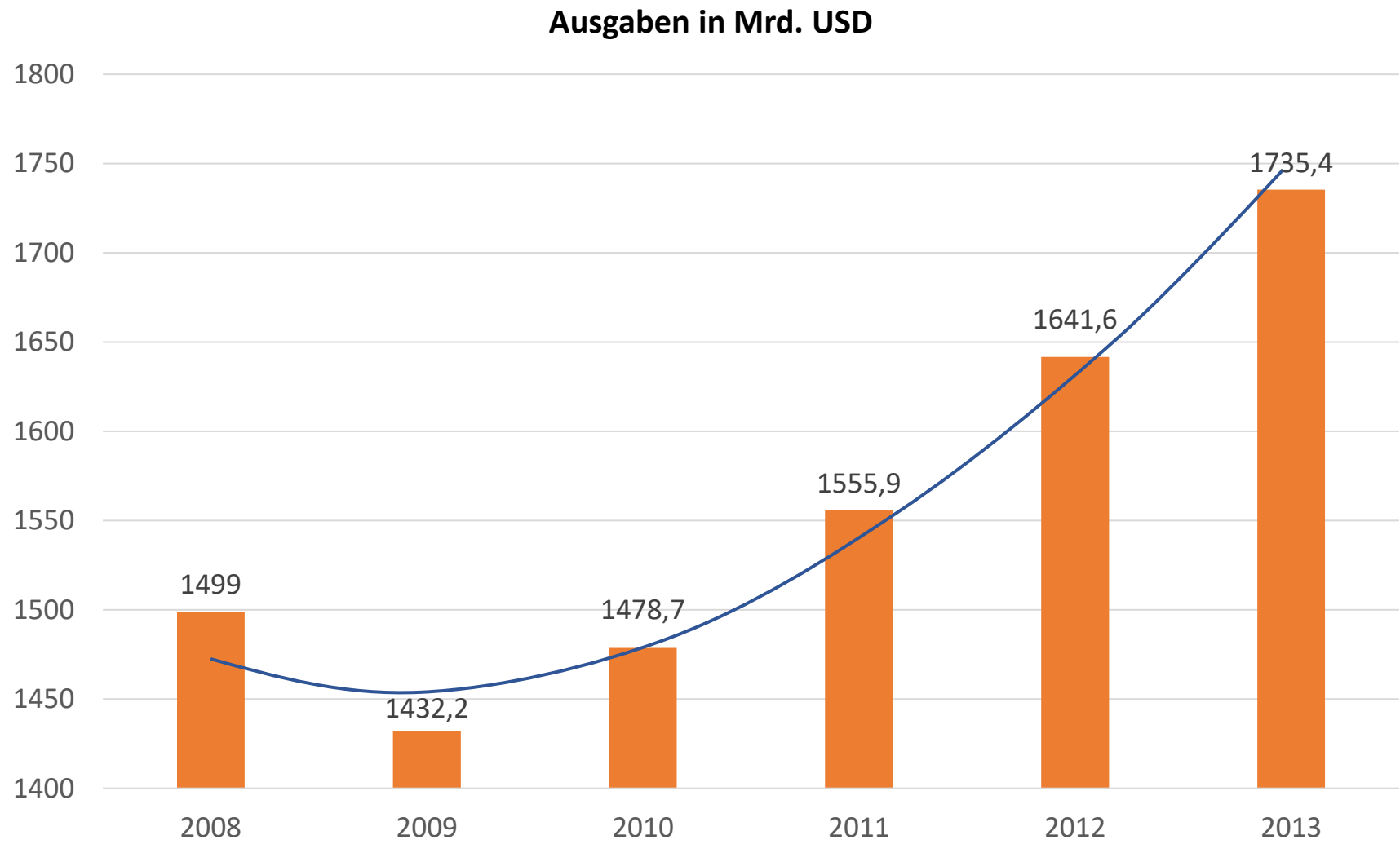
- Beherrschung von Zeit und Kosten
- Erreichung von Qualitätszielen
- Sicherung von Investitionen
- Einhaltung von Rahmenbedingungen

Grundlagen

„Magische Dreieck des Projektmanagements“ und „Teufelsquadrat“

Das „Magische Dreieck des Projektmanagements“ und das „Teufelsquadrat“ nach H. Sneed. Beide Darstellungen illustrieren die wechselseitigen Abhängigkeiten zwischen Zeit, Umfang, Qualität und Kosten.





Broy, M., & Kuhrmann, M. (2013). Projektorganisation und Management im Software Engineering. Berlin, Heidelberg, New York: Springer, S.16

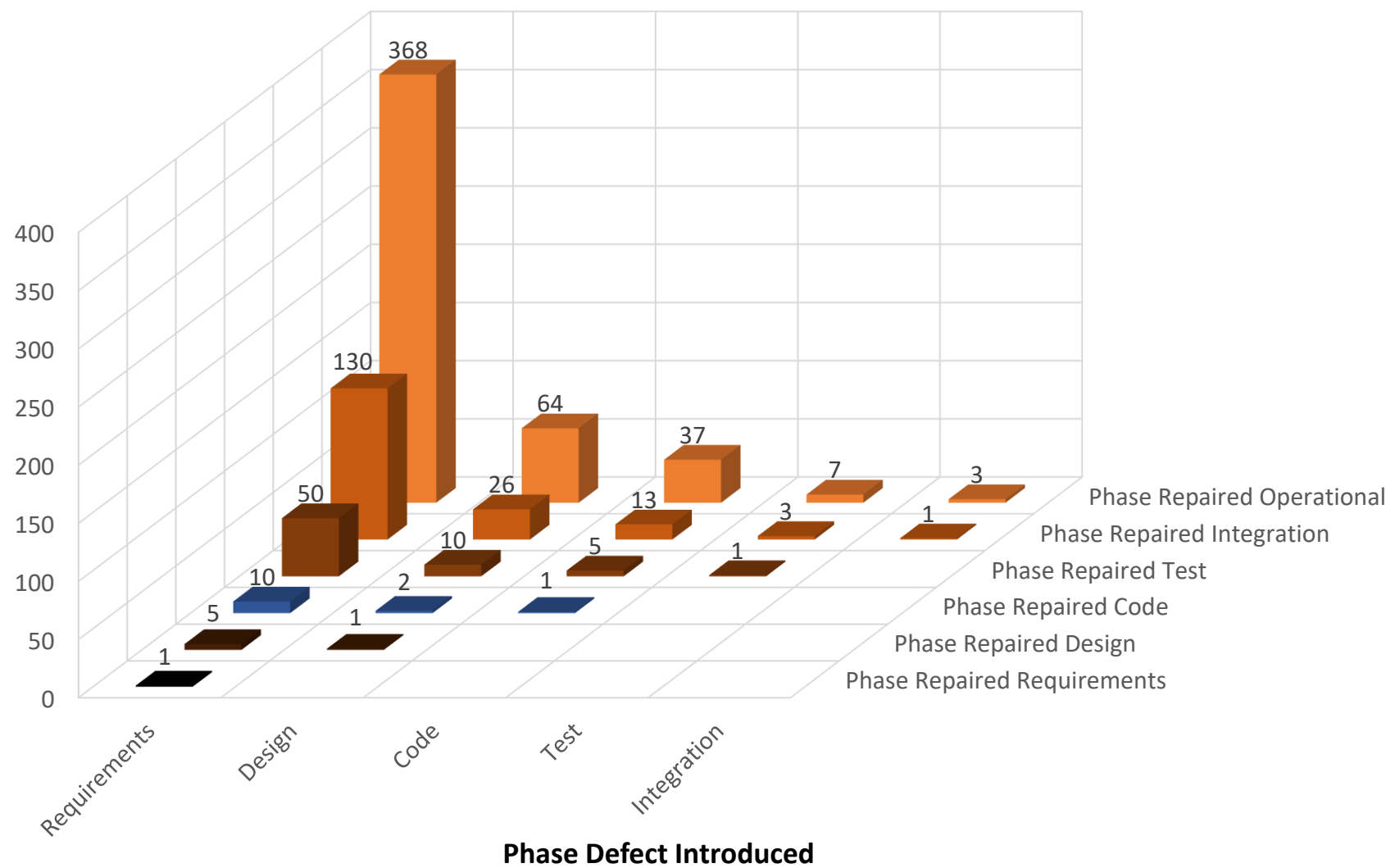
I. D. C. (IDC). IT-Umsatz erholt sich langsam, 03 (2010)

1. Kostenschätzungen für Softwareprojekte erweisen sich immer wieder um Faktoren zu niedrig. Als Folge können Projekte oft nur bei signifikanter Überschreitung des Budgets zum Abschluss gebracht werden.
2. Terminschätzungen erweisen sich oft als zu optimistisch. Infolge dessen wird Software oft verspätet ausgeliefert oder auf den Markt gebracht.
3. Die Qualität und Funktionalität der Software entspricht nicht immer den Erwartungen und Erfordernissen. Die Software enthält in manchen Fällen katastrophale Fehler, in anderen Fällen zeigt sie sich instabil oder arbeitet nicht mit der erforderlichen Performanz.

- **Kosten**
Die Kostenentwicklung in Softwareprojekten ist besonders eindrucksvoll, wenn man sie mit der Kostenentwicklung bei Hardware vergleicht. Die Kosten für Hardware fallen im Verhältnis zu ihrer Leistungsfähigkeit seit Jahrzehnten exponentiell.
- **Behandlung von Änderungen**
Eines der großen Probleme der Softwareentwicklung ist sicherlich der ständig erforderliche Änderungsaufwand.

Grundlagen

Bugfixing-Kosten nach Bennett und Wennberg [31]



Für eine gute Projektorganisation sind folgende Prinzipien unabdingbare Voraussetzung:

1. Klare Projektziele und transparente, realistische Projektpläne
2. Einbindung aller relevanten Interessengruppen (Stakeholder)
3. Klar geregelte, durchschaubare Verantwortungsstrukturen (Rollenkonzept) mit definierten Entscheidungsbefugnissen und festgelegten, transparenten Entscheidungsprozessen
4. Unabhängige Kontrollinstanzen für die Qualitäts- und Fortschrittssicherung der Arbeiten im Projekt mit einer klaren, effektiven Eskalationsstruktur
5. Offene Kommunikationskultur und festgelegte Schnittstellen zwischen den Interessengruppen zur Minimierung (unproduktiver) Kommunikation
6. Hohe Motivation der Mitarbeiter
7. Guter Ausbildungsstand der Mitarbeiter
8. Gute Arbeitsbedingungen und Arbeitsumgebung
9. Erfahrene Projektmanager und Entwickler

- Mitarbeiter
- Kundenorientierung
- Prozessorientierung
- Dokumentation und Artefakte – Ergebnisorientierung
- Modularisierung und Wiederverwendung

Zu den wichtigsten strategischen Erfolgsfaktoren zählen im Allgemeinen:

- Ausrichtung auf die Kunden
- Realistische Ziele für das Unternehmen und seine zukünftige Entwicklung
- Entwicklung von handhabbaren und aussagekräftigen Metriken zur Messung des Grads der Zielerreichung
- Klare Fokussierung auf das Kerngeschäft
- Effizienter Werkzeug- und Methodenbaukasten für die Softwareentwicklung
- Etablierung kontinuierlicher Verbesserungsprozesse

Unternehmenspositionierung und –strategie

- Unternehmensgeschäftsfelder
- Kunden
- Produkte
- Partner
- Außendarstellung

Personal

Wissensmanagement

Prozessentwicklung und -verbesserung

► **Definitionen Projektportfolio und Programm**

► **Definition (Projektportfolio)**

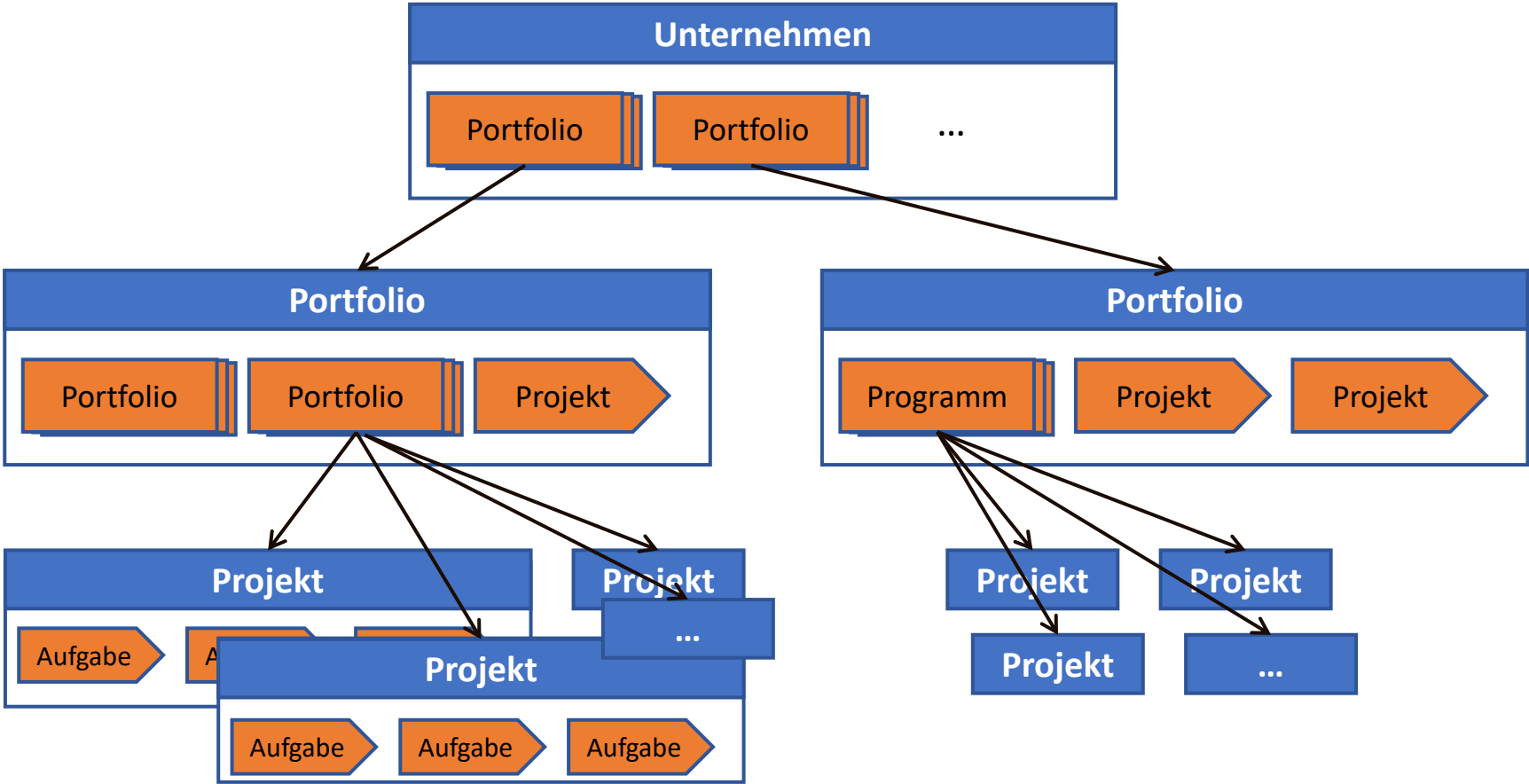
Ein Projektportfolio ist eine Menge von Projekten, die übergreifend koordiniert werden. Ziel ist die optimierte Nutzung von Ressourcen durch ein übergreifendes Management.

► **Definition (Programm)**

Ein Programm umfasst eine Menge von inhaltlich zusammengehörenden Projekten (und gegebenenfalls zugehöriger Einzelaufgaben), die dem Erreichen strategischer Unternehmensziele dienen und deren Durchführung für sich allein genommen wenig wirtschaftlich oder nicht projektwürdig sind.

Grundlagen

Positionierung von Projekten, Portfolios und Programmen



Broy, M., & Kuhrmann, M. (2013). Projektorganisation und Management im Software Engineering. Berlin, Heidelberg, New York: Springer, S.29

- Recherchieren Sie den „Software Engineering Body of Knowledge“ (SEBoK). Verschaffen Sie sich einen Überblick, welche Themengebiete insgesamt zur Disziplin Software Engineering gehören.
- Warum verlaufen Software-Entwicklungsprojekte häufig problematisch? Welche Punkte werden besonders häufig genannt? Was sind Gegenmaßnahmen?
Beispiele für prominente IT-Projekte/Produkte:
 - Toll-Collect (Autobahn-Maut)
 - Adonis (Austrian Digital Operating Network for Integrated Services)
 - Denver International Airport Baggage Handling System (Denver International Airport)
 - „Passion for Performance“ (P4P) SAP-Integration bei OTTO
 - FISCUS (vereinheitlichte IT für die Steuerverwaltung)
 - Dialba2020 (Raiffeisenbanken Schweiz)
 - Inpol Neu (BKA)
 - Gesundheitskarte
 - "Rollenbasierte Oberfläche“ (RobasO) der Bundesagentur für Arbeit (BA)
 - A2LL („Arbeitslosengeld II – Leistungen zum Lebensunterhalt“) der BA
 - Therac-25 der Atomic Energy of Canada Limited (AECL)
 - Schiaparelli EDM Lander