



# Architecture Tradeoff Analysis Method<sup>SM</sup> – ATAM

---

- Motivation
- Elemente von ATAM
- Ablauf einer ATAM-Bewertung
- Beispiel
- Zusammenfassung
- Quellen

*Architecture Tradeoff Analysis Method* and *ATAM* are service marks of Carnegie Mellon University.





ATAI

About the SEI

Areas of Work

Work with Us

Products & Services

Publications

● [Course Offerings](#)

■ [By Category](#)

□ [By Date](#)

□ [By Name](#)

□ [By Location](#)

○ [Prices](#)

○ [Locations, Travel, and Lodging](#)

○ [Courses FAQ](#)

○ [Registration](#)

○ [Contact Information](#)

○ [Credentials Program](#)

○ [SEI Certification](#)

## ATAM Leader Training

Dates	2007 Prices (USD)
<p><b>2007 Dates</b>            April 24-25, 2007 (SEI Pittsburgh, PA)            October 9-10, 2007 (SEI Pittsburgh, PA)</p> <p><b>Course Registration</b>            Software Engineering Institute            Carnegie Mellon University            Pittsburgh, PA 15213-3890            Phone: 412 / 268-7388            FAX: 412 / 268-7401            Questions: <a href="mailto:courseregistration@sei.cmu.edu">courseregistration@sei.cmu.edu</a>            To Register: <a href="#">2007 Click Here</a></p> <p>This course may also be offered by arrangement at customer sites. E-mail <a href="mailto:course-info@sei.cmu.edu">course-info@sei.cmu.edu</a> or call +1 412-268-7622 for details.</p>	<p><b>U.S. Course Fee:</b>            Industry: \$2640            Government: \$2120            Academic: \$2120</p> <p><b>International Course Fee:</b>            \$3960</p> <p><b>Please review prerequisites before registering below.</b></p>

### Course Description

Leading a software architecture evaluation using the SEI Architecture Tradeoff Analysis Method<sup>®</sup> (ATAM<sup>®</sup>) requires technical expertise in both software architecture and quality attributes, a working knowledge of the ATAM, and superior facilitation skills. The social dimension of leading the ATAM is often overlooked, but there are considerable facilitation challenges for the ATAM lead evaluator:

- How do you manage multiple people with diverse viewpoints?

Fol  
Vo



# Voraussetzung

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ Ablauf
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

Architekturen werden  
weitgehend durch  
*nicht-funktionale / Qualitäts-*  
*Anforderungen* geprägt



ATAM<sup>SM</sup>



- Bei jedem System-Design sind *tradeoffs* (= Kompromisse) notwendig
- Wichtig ist, diese *tradeoffs* explizit zu machen  
→ nur so lassen sich potenzielle Probleme / Risiken (früh!) erkennen

**Tradeoff points** sind Stellen in der Architektur, die für verschiedene Qualitätsattribute gleichzeitig relevant sind



# Ziele von ATAM

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ Ablauf
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- ATAM soll Problem-/Risikobereiche in *einer* Architektur aufdecken
- Es soll *nicht* versucht werden, Qualitätseigenschaften konkret vorherzusagen
- Gesucht sind Korrelationen zwischen **Parametern für die Architektur** und Qualitätseigenschaften  
→ *Parameter* können beeinflusst / geändert werden!

## Beispiele

- Bitzahl für den Schlüssel in einer Verschlüsselungskomponente: Sicherheit vs. Performance
- Möglichkeit, Aufgaben auf parallele Threads zu verteilen: Ausbaufähigkeit vs. Performance in "kleinen" Systemen
- Solche parametrisierbaren Stellen können/sollten dann weiter untersucht werden:  
detailliertes Design, andere Analysen, Prototyping etc.



# Positive Effekte von ATAM

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ Ablauf
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Klären von Anforderungen
- Verbesserte Architektur-Dokumentation
- Dokumentierte Basis für Entscheidungen auf Architektur-Ebene

→ Bessere Architekturen...



# Qualitätsattribute

ATAM<sup>SM</sup>



- Qualitätsattribute, die von ATAM beobachtet werden (können)
  - Modifizierbarkeit (wie SAAM)
  - Performance
  - Sicherheit (*security*)
  - Zuverlässigkeit
  - etc.
- Was "mitemledigt" werden kann
  - Vollständigkeit von Anforderungen
  - Testbarkeit
  - etc.



ATAM<sup>SM</sup>

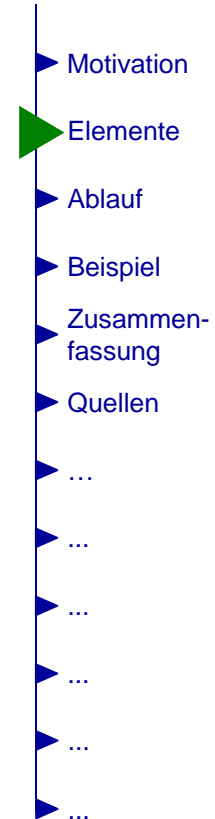


- Ausgewählt werden Stakeholders mit konkretem Interesse am System
- Beispiele
  - Architekt (!)
  - Entwickler
  - Tester
  - Wartungsentwickler
  - System-Administrator
  - Kunde
  - Anwender
  - etc.



# Darstellung der Architektur 1/2

ATAM<sup>SM</sup>



- Die Architektur wird in verschiedenen Sichten (*structures*) dargestellt
- Beispiele
  - static (e.g., objects, procedures, calls)
  - data flow
  - control flow
  - physical (e.g., computers, devices, networks)
  - process
  - uses
  - execution (e.g., trace of activations)



## Darstellung der Architektur 2/2

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ Ablauf
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Verschiedene Sichten ermöglichen die Diskussion von Änderungen (Verbesserungen?)
- Die Auswahl der Sichten wird von den zu untersuchenden Qualitätsattributen bestimmt
  - Modifizierbarkeit      static, uses, data flow
  - Sicherheit              data flow
  - Performance            process, physical, execution
  - Zuverlässigkeit        physical, data flow
- Stakeholders bestimmen, welche Qualitätsattribute relevant sind



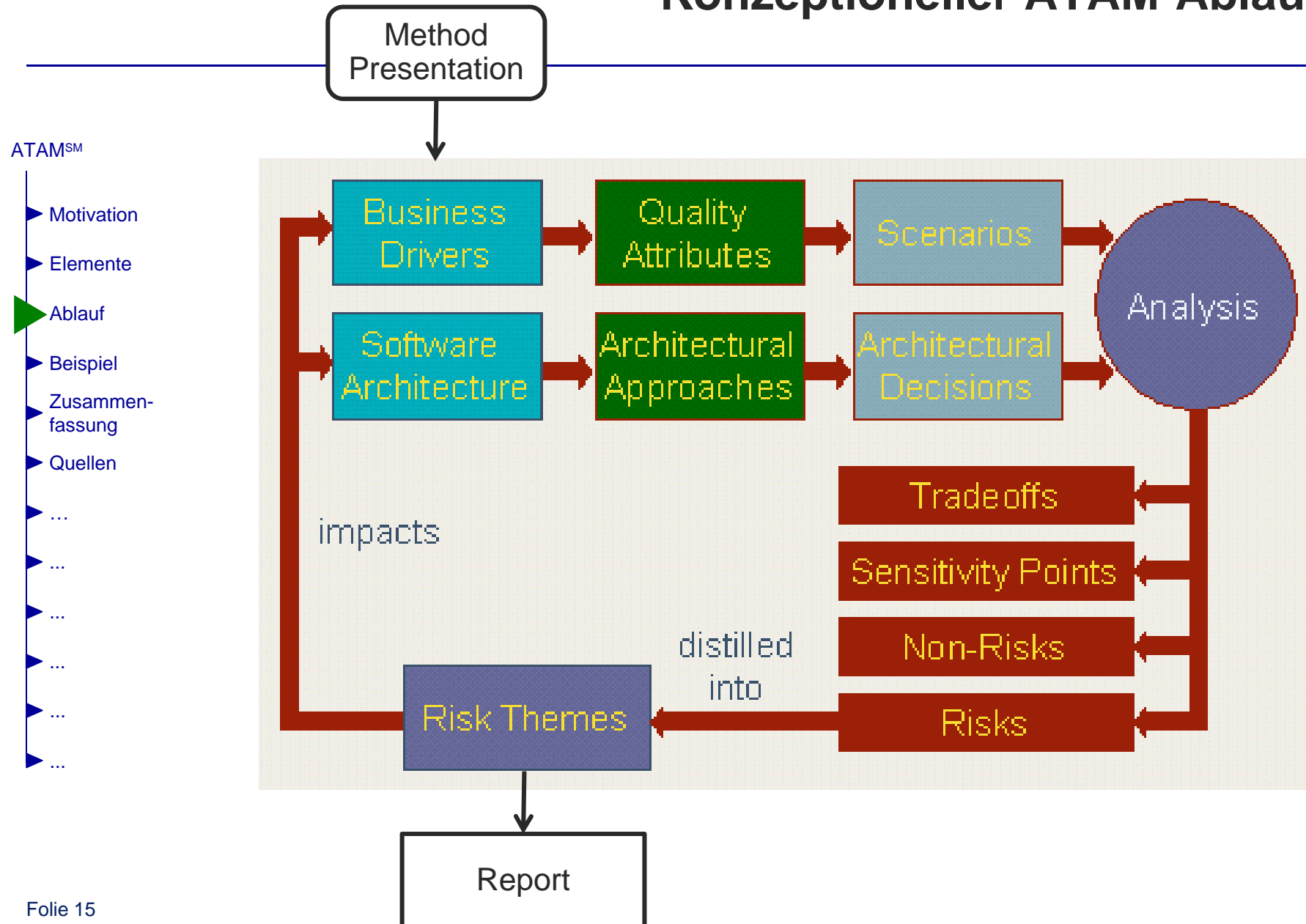
ATAM<sup>SM</sup>



- Wie für SAAM werden Szenarien benutzt, um Qualitätsattribute konkret zu beschreiben
- Ein Szenario beschreibt konkret, wie das System
  - benutzt werden soll,
  - geändert werden soll,
  - einer besonderen Last ausgesetzt werden soll,
  - zum Absturz gebracht werden soll,
  - etc.aus Sicht von einem oder mehreren (bestimmten!) Stakeholders



# Konzeptioneller ATAM-Ablauf





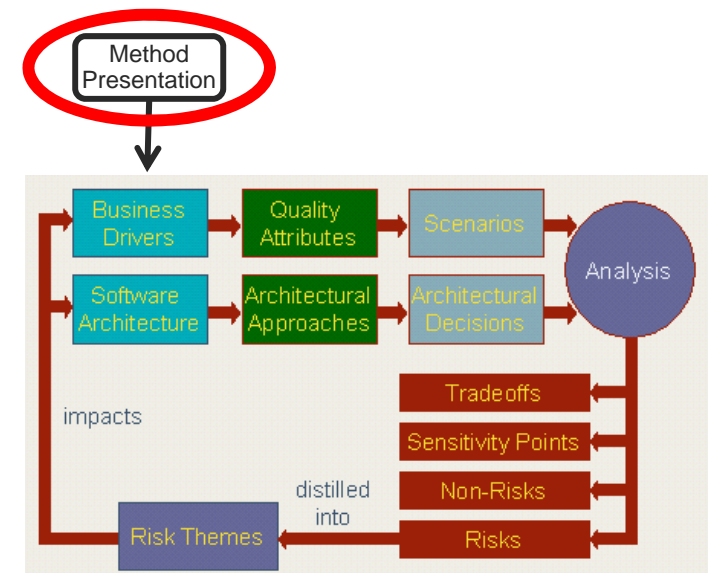
# 1. Present the ATAM

## Erklären von ATAM für alle Stakeholder

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Prozessschritte
  - Techniken für die Informationsgewinnung
  - Techniken für die Informationsanalyse
  - Szenarien, deren Priorisierung, Utility Tree
  - Ergebnisse, Form / Dokumentation der Ergebnisse
- Kontext
- Erwartungen
- Verwendung der gewonnenen Ergebnisse





## 2. Present Business Drivers

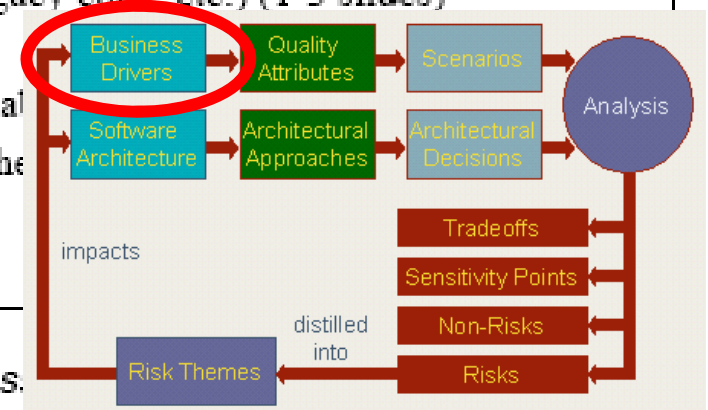
### Projektmanager: System-Übersicht aus der Geschäftsperspektive

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

#### Business Context/Drivers Presentation (~ 12 slides; 45 minutes)

- Description of the business environment, history, market differentiators, driving requirements, stakeholders, current need and how the proposed system will meet those needs/ requirements (3-4 slides)
- Description of business constraints (e.g., time to market, customer demands, standards, cost, etc.) (1-3 slides)
- Description of the technical constraints (e.g., COTS, interoperation with other systems, required hardware or software platform, reuse of legacy code, etc.) (1-3 slides)
- Quality attributes desired (e.g., performance, availability, operability, integrability) and what business needs the system to deliver (1-3 slides)
- Glossary (1 slide)





# 3. Present Architecture

## Chef-Architekt: Architektur-Übersicht

- Technische Randbedingungen: OS, Hardware, Middleware etc.
- Vorhandene Systeme (→ Interaktion)
- Maßnahmen (*approaches*), um die Qualitätsanforderungen zu erfüllen (vgl. nächste Folie)

## Der Detailgrad

- hängt von Zeit, vorhandenem Material etc. ab,
- bestimmt die Qualität der / einiger Analysen und
- muss je nach Bedarf verfeinert werden

ATAM<sup>SM</sup>

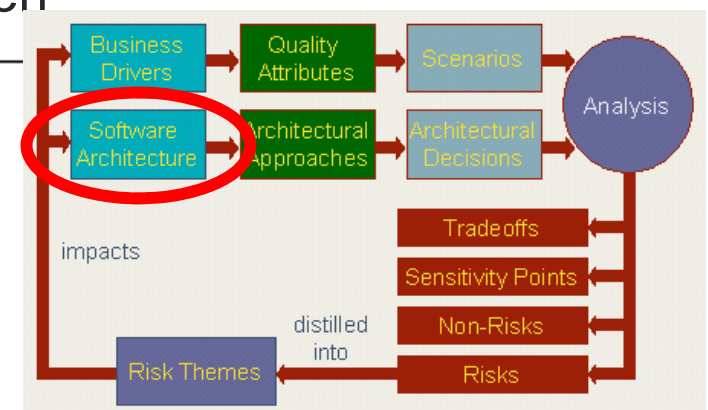
- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

### Architecture Presentation (~20 slides; 60 minutes)

- Driving architectural requirements (e.g., performance, availability, security, modifiability, interoperability, integrability), the measurable quantities you associate with these requirements, and any existing standards/models/approaches for meeting these (2-3 slides)

Folie 18

Vorlesung SWA | Prof. Dr. Peter Knauber | HS MA, SS 2013





## 4. Identify Architectural Approaches

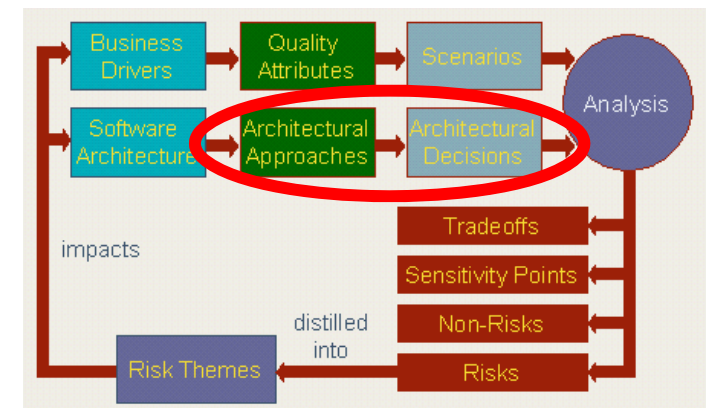
ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Der Architekt stellt die Maßnahmen/seine Überlegungen vor, die sicherstellen sollen, dass die (ursprünglichen) Qualitätsanforderungen erfüllt werden

"We look for approaches and styles because not all architects are familiar with the language of *architectural styles*, and so may not be able to enumerate a set of styles used in the architecture. But every architect makes architectural decisions, and the set of these we call '*approaches*'. These can certainly be elicited from any conscientious architect."

→ Nur Identifizieren / Präsentieren, nicht Analysieren!





# 5. Generate Quality Attribute Utility Tree

ATAM<sup>SM</sup>

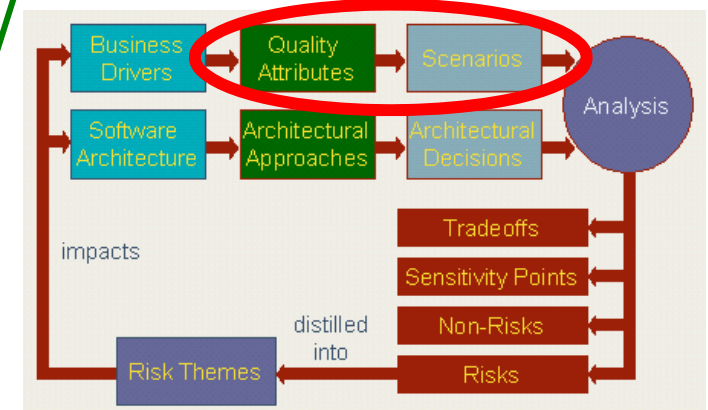
- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

## Generieren von Szenarien, um die Architektur zu bewerten

	Utility Trees	Facilitated Brainstorming
Stakeholders	Architects, project leader	All stakeholders
Typical Group size	2 evaluators; 2-3 project personnel	4-5 evaluators; 5-10 project-related personnel
Primary Goals	Elicit, concretize and prioritize the driving quality attribute requirements. Provide a focus for the remainder of the evaluation.	Foster stakeholder communication to validate quality attribute goals elicited via the utility tree.
Approach	<u>Top-down</u> (general to specific)	<u>Bottom-up</u> (specific to general)

Bekannt von SAAM

Dieser Schritt kommt später!





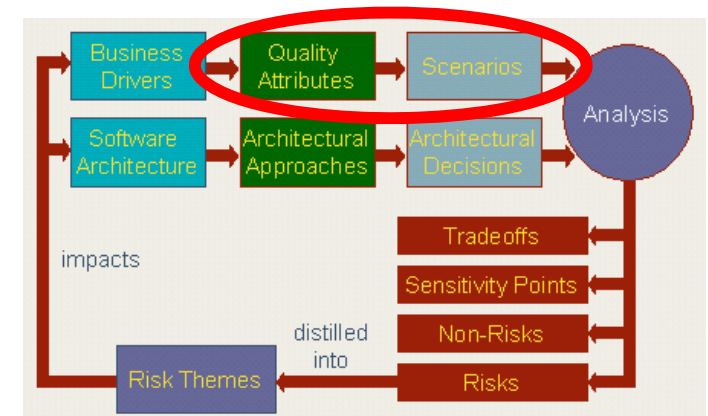
# Generieren von Szenarien

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Mögliche relevante Qualitätsattribute
  - Änderungsszenarien
  - Bedrohungsszenarien
  - Szenarien für Ausnahmebedingungen
  - Benutzungsszenarien
  - etc.
- Das sind mehr/andere Kategorien als in SAAM
- Top-Down vs. Bottom-Up
  - Verbesserte Chancen auf "Vollständigkeit"
  - Gegenseitige Kontrolle der Bedeutung (für die Priorisierung)

**Beispiel**





## 6. Analyze Architectural Approaches

ATAM<sup>SM</sup>

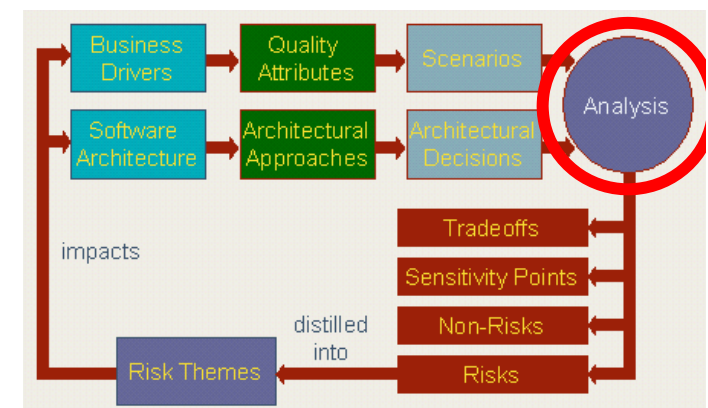
- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

Prüfen, ob die Architekturstile bzw. -*Tactics* wie erwartet die Qualitätsanforderungen erfüllen:

- Sind sie richtig angewendet (*instantiated*)?
- Sind sie konsequent angewendet?
- Welche Anforderungen betreffen welche "Stellen" in der Architektur?
- Gibt es "Stellen", die von mehreren Anforderungen betroffen sind?

Vorgehen:

- Der Architekt wendet die Szenarien nacheinander auf die Architektur an und erklärt die *Antwort/Reaktion* der Architektur darauf
- *Reaktionen* können zur Laufzeit oder zur Entwicklungszeit stattfinden
- Die Analysten protokollieren die Antworten des Architekten



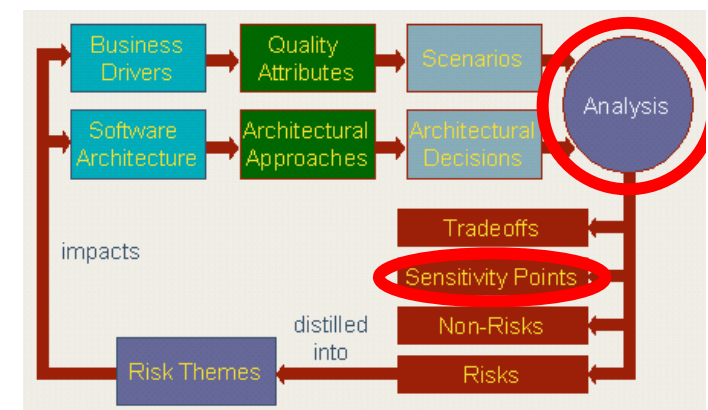


## 6. *Sensitivity-Points* identifizieren

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Im Dialog mit dem Architekten werden Qualitätsattribut-spezifische *Sensitivity-Points* in der Architektur identifiziert
- Definition:  
Ein **Sensitivity-Point** ("sensible Stelle") ist ein Parameter in der Architektur, dessen Wert bestimmt, ob eine Qualitätsanforderung besser oder schlechter erfüllt wird
- Die Bewerter können präzise Modelle für nachfolgende Analysen skizzieren, s. nächste Folie



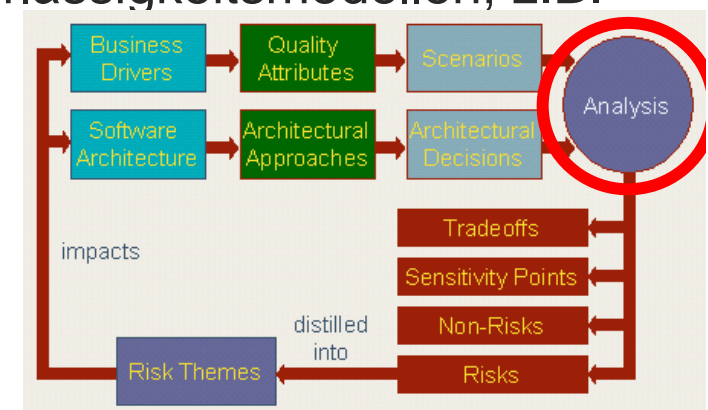


# Beispiele für Attribut-spezifische Analysen

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- **Modifizierbarkeit:**  
(SAAM-)Analysen identifizieren betroffene Elemente und den potenziellen Aufwand für Änderungen
- **Performance:**  
Entwerfen von Performance-Modellen des Systems für spätere genaue Analysen, z.B. Queuing- oder Scheduling-Modelle
- **Sicherheit (*security*):**  
Identifizieren potenzieller Schwachstellen bei Bedrohungen
- **Zuverlässigkeit:**  
Entwickeln von Verfügbarkeits-/Zuverlässigkeitsmodellen, z.B. mittels Markov-Modell



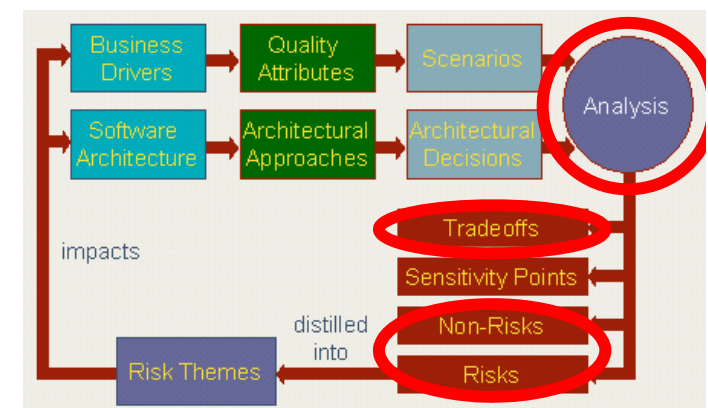


## 6. *Tradeoff-Points* identifizieren

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ Ablauf
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Identifizieren von *Tradeoff-Points*, Risiken und Nicht-Risiken
  - Definition:  
ein **Tradeoff-Point** ("Kompromiss-Stelle") ist eine Stelle in der Architektur, die für mehr als ein Qualitätsattribut sensitiv ist
- Ermitteln, ob die betroffenen Qualitätsattribute "parallel" zueinander beeinflusst werden oder gegenläufig





## 6. Tradeoff-Points identifizieren

ATAM<sup>SM</sup>

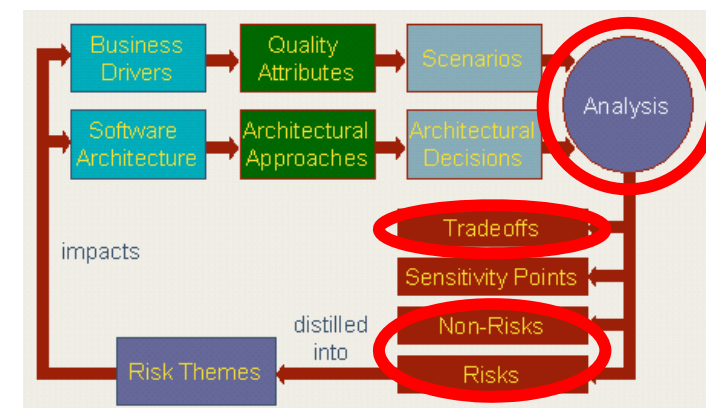
- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

### Risiken:

- Als (negative) Konsequenz aus der Verwendung bestimmter Stile
- Als Konsequenz aus der Interaktion / Überlagerung verschiedener Stile
- Mit Referenz in den *Utility Tree* (→ *Business Driver*)

### Nicht-Risiken:

- Risiken, die durch die Architektur entschärft werden





# 7. Brainstorm and Prioritize Scenarios

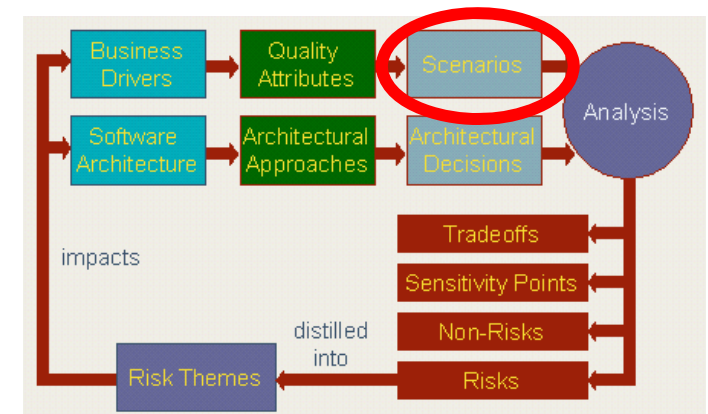
## (1. Schritt des Kontrollzyklus)

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

	Utility Trees	Facilitated Brainstorming
Stakeholders	Architects, project leader	All stakeholders
Typical Group size	2 evaluators; 2-3 project personnel	4-5 evaluators; 5-10 project-related personnel
Primary Goals	Elicit, concretize and prioritize the driving quality attribute requirements. Provide a focus for the remainder of the evaluation.	Foster stakeholder communication to validate quality attribute goals elicited via the utility tree.
Approach	<u>Top-down</u> (general to specific)	<u>Bottom-up</u> (specific to general)

Bekannt von SAAM





# 7. Beispiel für ein Szenario-Ranking

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

Scenario	#Votes	Quality Attributes
4	28	Performance
27	26	Performance, Modifiability, Availability
10	23	Integrability
12	13	Performance
14	12	Modifiability

Figure 10: Highly Ranked Scenarios with Quality Attribute Annotations



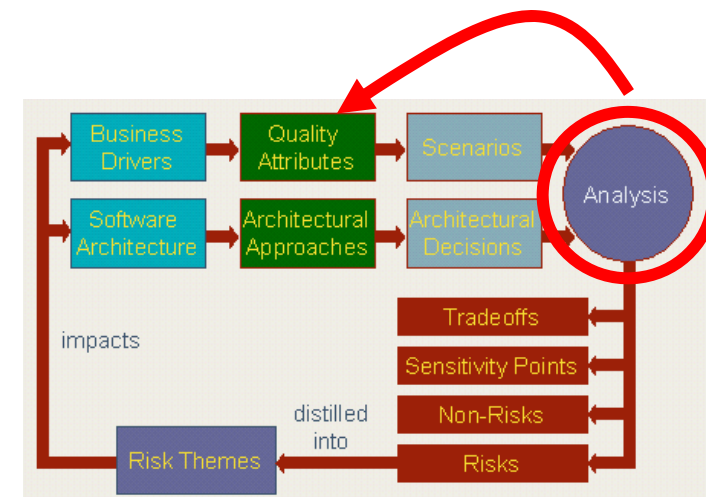
# 8. Analyze Architectural Approaches

## (2. Schritt des Kontrollzyklus)

ATAM<sup>SM</sup>

- Motivation
- Elemente
- Ablauf
- Beispiel
- Zusammenfassung
- Quellen
- ...
- ...
- ...
- ...
- ...
- ...

- Gleiches Vorgehen wie in Schritt 6 mit der Hoffnung, nichts Neues mehr zu finden...
- Falls doch:  
Zurück zu Schritt 4, *Utility Tree Generation*, um „benachbarte“ Szenarien zu entwickeln/zu analysieren





## 9. Present Results

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

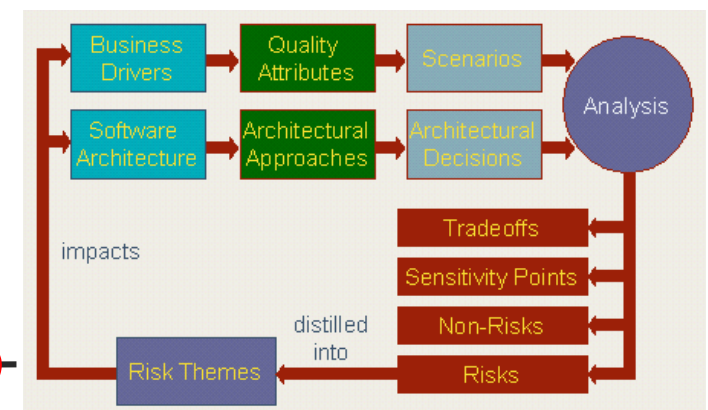
Aufgabe des ATAM-Teams:

- Mündlicher Bericht mit Folien
- Eventuell (nachträglich) schriftlicher Bericht

Inhalte:

- Dokumentierte Anwendung von Architekturstilen
- Szenarien, Fragen, *Utility Tree*
- Dokumentierte Risiken
- Dokumentierte Nicht-Risiken
- Dokumentierte *Sensitivity Points*, *Tradeoff Points*

Verbesserungsvorschläge können,  
*müssen aber nicht* gemacht werden!





# Hausaufgaben, Nacharbeiten

---

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ **Ablauf**
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- Entwickeln neuer Design-Alternativen und Bewertung ihrer Konsequenzen
- Verfeinern / Analysieren spezifischer Modelle
- Modelle erstellen und dokumentieren, die nachträglicher Verfeinerungen oder Analysen bedürfen



ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ Ablauf
- ▶ **Beispiel**
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

# Beispiel

# (Rick Kazzman)





# Zusammenfassung

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ Ablauf
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- ATAM ist eine Methode, um Architekturen bzgl. unterschiedlicher Qualitätsattribute (viele/mehrere gleichzeitig) zu bewerten
- Architekturbewertungen sind ein Mittel, um die Risiken zu vermeiden, die aus schlechten Architekturentwürfen entstehen können
  - Sie können früh im Entwicklungsprozess durchgeführt werden
  - Sie sind relativ billig (ca. 3 Tage, 30-70 Arbeitstage)
- ATAM sucht nach *Trends*, versucht nicht, präzise Analysen zu machen
  - Trend = Korrelation von Qualitätsattribut und Architekturelement
- ATAM benötigt
  - Klar formulierte Qualitätsanforderungen
  - Aktive Teilnahme aller Stakeholder
  - Aktive Teilnahme des Architekten
  - Für die (nachfolgende) Analyse: Übung im Umgang mit Analysemodellen



ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ Ablauf
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ **Quellen**
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

- R. Kazman, M. Klein, P. Clements: *ATAM – A Method for Architecture Evaluation*, Technical Report CMU/SEI-2000-TR-004, ESC-TR-2000-004, August 2000; verfügbar unter <http://www.sei.cmu.edu/publications/documents/00.reports/00tr004.html>, Mai '06
- Software Architecture Technology Initiative des SEI: <http://www.sei.cmu.edu/architecture/>, Mai '06



## Evaluating a Software Architecture

[P. Clements, R. Kazman, M. Klein]

ATAM<sup>SM</sup>

- ▶ Motivation
- ▶ Elemente
- ▶ Ablauf
- ▶ Beispiel
- ▶ Zusammenfassung
- ▶ Quellen
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...
- ▶ ...

In general, when can an architectural evaluation be held?

As soon as there is enough of an architecture to justify it.

Different organizations may measure that justification differently, but a good rule of thumb is this:

Hold an evaluation when development teams start to make decisions that depend on the architecture and the cost of undoing those decisions would outweigh the cost of holding an evaluation.