

2015-02-04

# On the relationships between QoS and software adaptability at the architectural level

On the relationships between QoS and software adaptability at the architectural level

Daniel Schmidt

17.02.2015

## On the relationships between QoS and software adaptability at the architectural level

Daniel Schmidt

17.02.2015

2015-02-04

# On the relationships between QoS and software adaptability at the architectural level

## └ Inhalt

### Inhalt

- 1 Einleitung
- 2 Anpassungsfähigkeit
- 3 Metriken
  - AAS und RAS
  - MAAS und MRAS
  - LSA
- 4 Ansatz
  - Adapt<sup>-</sup> und Adapt<sup>+</sup>
  - Mehrere Anforderungen
  - Beziehungen der QoS zur Anpassungsfähigkeit
- 5 Analyse des Ansatzes

## Inhalt

- 1 Einleitung
- 2 Anpassungsfähigkeit
- 3 Metriken
  - AAS und RAS
  - MAAS und MRAS
  - LSA
- 4 Ansatz
  - Adapt<sup>-</sup> und Adapt<sup>+</sup>
  - Mehrere Anforderungen
  - Beziehungen der QoS zur Anpassungsfähigkeit
- 5 Analyse des Ansatzes

# Einleitung

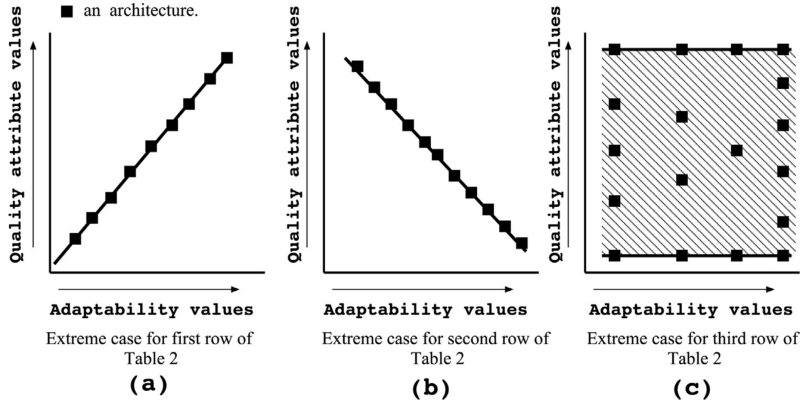


Abbildung : Zusammenhang zwischen Anpassungsfähigkeit und QoS

2015-02-04

On the relationships between QoS and software adaptability at the architectural level

└ Einleitung

└ Einleitung

Einleitung

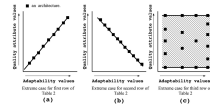


Abbildung : Zusammenhang zwischen Anpassungsfähigkeit und QoS

- Es geht immer um garantierte Anpassungsfähigkeit
- Garantierte Anpassungsfähigkeit von Software kann andere Qualitätsattribute wie Geschwindigkeit, Verlässlichkeit und Wartbarkeit beeinflussen.
- Ansatz ist bei einem wechselnden Kontext nützlich, er wird benutzt um zu testen ob die ausgewählten Komponenten die Voraussetzungen des Systems erfüllen.

# Anpassungsfähigkeit

## Definition (Anpassungsfähiges Software System)

Ein anpassungsfähiges Software System kann Änderungen in der Umwelt ohne einen externen Eingriff vertragen.

Lawrence Chung Nary Subramanian. *Metrics for Software Adaptability*. URL: <https://www.utdallas.edu/~chung/ftp/sqm.pdf>

- Quantifizierung des Grads der Anpassungsfähigkeit wichtig
- Über heuristische Verfahren kann eine automatische Anpassung der Architektur erfolgen, hin zu einer Architektur, welche die Qualitätsmerkmale erfüllt oder nah dran ist
- Die Ziele des Papers sind:
  - Eine erweiterte Menge von architekturellen Metriken die zur Evaluierung der Anpassungsfähigkeit des Systems verwendet werden können
  - Der Ansatz benutzt diese Metriken um die Beziehung zwischen Anpassungsfähigkeit und Qualitätswerten zu definieren, damit hilft dieser Ansatz bei der Begründung des Designs
  - Ein Hilfsmittel bereitstellen um den Ansatz zu benutzen

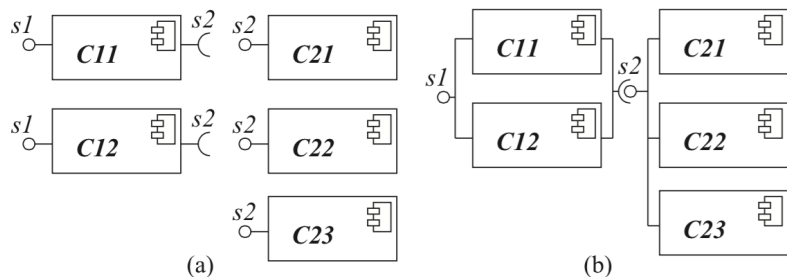
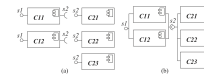


Abbildung : Beispiel Component-and-Connector Ansicht

- Der Ansatz basiert auf einer Component-and-Connector Ansicht, da sie allgemein verwendet wird um über die Qualitätswerte zur Laufzeit zu reden.
- Sockets für benötigte und angebotene Dienste
- Gemeinsame Linien zeigen an dass mehrere Komponenten den gleichen Dienst anbieten oder benötigen.



# Metriken

## Definition ( $UC_i$ )

Komponenten, die den Dienst i bereitstellen

## Definition ( $C_i$ )

Komponenten, die den Dienst i bereitstellen können

2015-02-04

On the relationships between QoS and software adaptability at the architectural level

└ Metriken

└ Metriken

Metriken

Definition ( $UC_i$ )

Komponenten, die den Dienst i bereitstellen

Definition ( $C_i$ )

Komponenten, die den Dienst i bereitstellen können

# Metriken

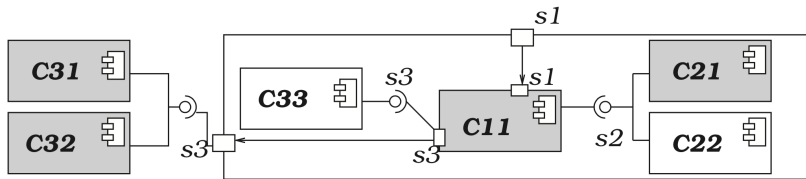


Abbildung : Beispielarchitektur

2015-02-04

On the relationships between QoS and software adaptability at the architectural level

└ Metriken

└ Metriken

Metriken



Abbildung : Beispielarchitektur



# Metriken

- AAS und RAS
- MAAS und MRAS
- LSA

2015-02-04

On the relationships between QoS and software adaptability at the architectural level

└─ Metriken

└─ Metriken

Metriken

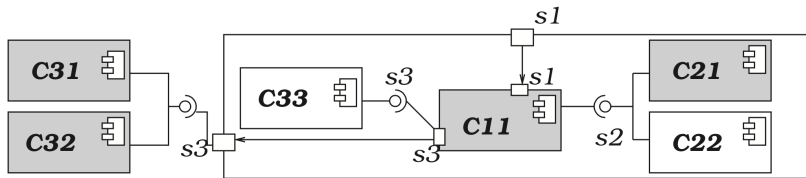
- AAS und RAS
- MAAS und MRAS
- LSA



## AAS

## Definition (Absolute adaptability of a service)

$$AAS_i = |UC_i|$$



2015-02-04

## On the relationships between QoS and software adaptability at the architectural level

- └ Metriken
  - └ AAS und RAS
    - └ AAS

AAS

Definition (Absolute adaptability of a service)

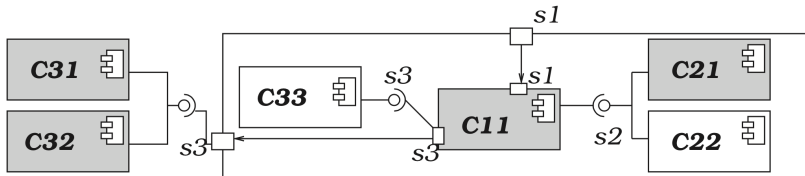
 $AAS_i = |UC_i|$ 

- **AAS** misst die Anzahl der benutzten Komponenten, welche gewisse Dienste bereitstellen.
- Lösung: [1,1,2]

## RAS

## Definition (Relative adaptability of a service)

$$RAS_i = \frac{|UC_i|}{|C_i|}$$



2015-02-04

## On the relationships between QoS and software adaptability at the architectural level

- └ Metriken
  - └ AAS und RAS
    - └ RAS

RAS

Definition (Relative adaptability of a service)

$$RAS_i = \frac{|UC_i|}{|C_i|}$$

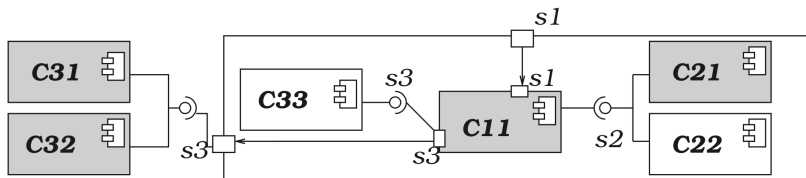


- **RAS** misst die Anzahl der verwendeten Komponenten, welche einen gegebenen Service bereitstellen in hinsicht auf die Anzahl der Komponenten, die tatsächlich solchen Service anbieten.
- Lösung: [1,0.5, 0.6]

# MAAS

## Definition (Mean of absolute adaptability of service)

$$MAAS = \frac{\sum_{i=1}^n AAS_i}{n}$$



2015-02-04

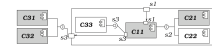
On the relationships between QoS and software adaptability at the architectural level

- └ Metriken
  - └ MAAS und MRAS
    - └ MAAS

MAAS

Definition (Mean of absolute adaptability of service)

$$MAAS = \frac{\sum_{i=1}^n AAS_i}{n}$$

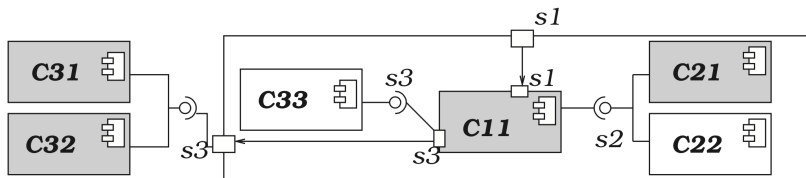


- **MAAS** misst die durchschnittliche Anzahl der genutzten Komponenten pro Dienstleistung. Komponenten, die tatsächlich solchen Service anbieten.
- Lösung:  $4/3 = 1.3$

## MRAS

## Definition (Mean of relative adaptability of service)

$$MAAS = \frac{\sum_{i=1}^n RAS_i}{n}$$



2015-02-04

## On the relationships between QoS and software adaptability at the architectural level

- └ Metriken
  - └ MAAS und MRAS
    - └ MRAS

MRAS

Definition (Mean of relative adaptability of service)

$$MAAS = \frac{\sum_{i=1}^n RAS_i}{n}$$

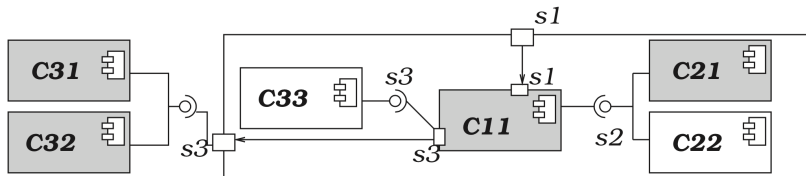


- **MRAS** misst den Durchschnitt des RAS (Relative Adaptability of a service).
- Lösung:  $(1 + 0.5 + 0.6) / 3 = 0.7$

## LSA

## Definition (Level of system adaptability)

$$LSA = \frac{\sum_{i=1}^n AAS_i}{\sum_{i=1}^n |C|}$$



2015-02-04

## On the relationships between QoS and software adaptability at the architectural level

└ Metriken

└ LSA

└ LSA

LSA

Definition (Level of system adaptability)

$$LSA = \frac{\sum_{i=1}^n AAS_i}{\sum_{i=1}^n |C|}$$



- **LSA** bezeichnet das Verhältnis zwischen der Anzahl an Komponenten aus denen ein System besteht und der Anzahl die das Anpassungsfähigste nutzen würde.
- Lösung:  $4 / (1 + 2 + 3) = 0.66666$

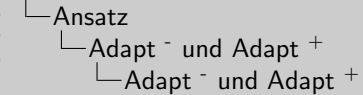
# Adapt<sup>-</sup> und Adapt<sup>+</sup>

## Definition (*Adapt<sup>-</sup>*)

Das niedrigste  $A_i$  für welches man eine Architektur finden kann, welche die Anforderungen erfüllt.

2015-02-04

## On the relationships between QoS and software adaptability at the architectural level

Adapt<sup>-</sup> und Adapt<sup>+</sup>

### Definition (*Adapt<sup>-</sup>*)

Das niedrigste  $A_i$  für welches man eine Architektur finden kann, welche die Anforderungen erfüllt.

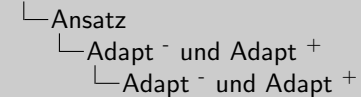
- Anforderungen werden durch Architekten gewählt und beziehen sich auf QoS des Systems.
- $A_i$  sind zunehmende Werte für die gewählte Metrik der Anpassungsfähigkeit.

Adapt<sup>-</sup> und Adapt<sup>+</sup>Definition (*Adapt*<sup>+</sup>)

Das niedrigste  $A_i$  für dessen Grenzen  $Q_{A_iU}$  und  $Q_{A_iL}$  die Anforderungen erfüllen.

2015-02-04

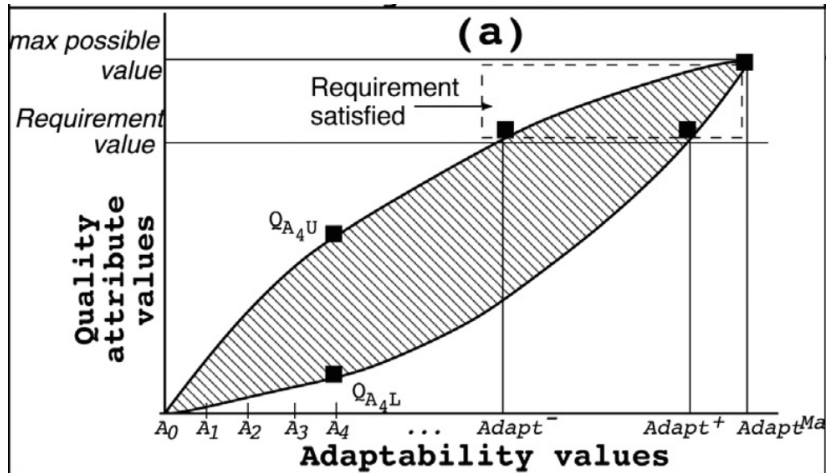
## On the relationships between QoS and software adaptability at the architectural level



- $Q_{A_iU}$  ist der höchste Qualitätswert den eine Architektur für ein Anpassungsfähigkeitsniveau erreichen kann.
- $Q_{A_iL}$  ist entsprechend der niedrigste.

Adapt<sup>-</sup> und Adapt<sup>+</sup>Definition (*Adapt*<sup>+</sup>)

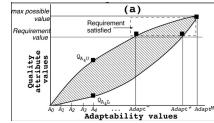
Das niedrigste  $A_i$  für dessen Grenzen  $Q_{A_iU}$  und  $Q_{A_iL}$  die Anforderungen erfüllen.

Adapt<sup>-</sup> und Adapt<sup>+</sup>

2015-02-04

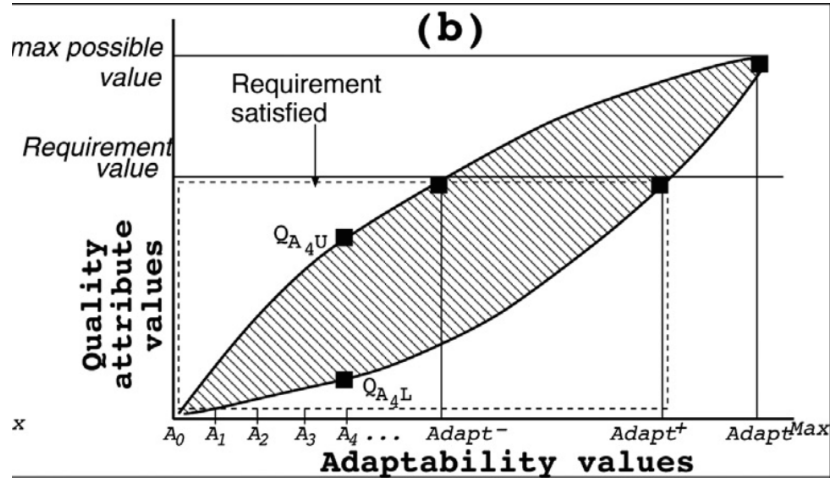
On the relationships between QoS and software adaptability at the architectural level

- └ Ansatz
  - └ Adapt<sup>-</sup> und Adapt<sup>+</sup>
    - └ Adapt<sup>-</sup> und Adapt<sup>+</sup>

Adapt<sup>-</sup> und Adapt<sup>+</sup>

- In (a) und (d) ist Adapt<sup>-</sup> das niedrigste  $A_i$  für welches man eine Architektur finden kann, welche die Anforderungen erfüllt. Adapt<sup>+</sup> ist das niedrigste  $A_i$ , dessen Grenzen  $Q_{A_iU}$  und  $Q_{A_iL}$  die Anforderungen erfüllt.
- Die Werte zeigen, dass die Erfüllung der Anforderungen eine Anpassungsfähigkeit von Adapt<sup>-</sup> voraussetzen und, dass jede Architektur die mindestens Adapt<sup>+</sup> hat die Anforderungen auch erfüllt. Für Anpassungsfähigkeit dazwischen gibt es Architekturen, die die Anforderungen erfüllen und solche die es nicht tun.

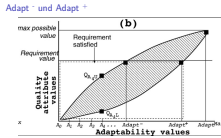


Adapt<sup>-</sup> und Adapt<sup>+</sup>

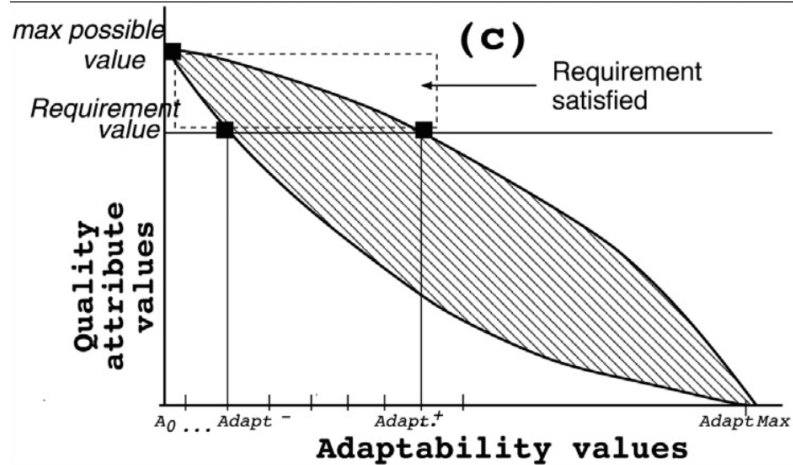
2015-02-04

On the relationships between QoS and software adaptability at the architectural level

- Ansatz
  - Adapt<sup>-</sup> und Adapt<sup>+</sup>
    - Adapt<sup>-</sup> und Adapt<sup>+</sup>



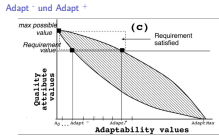
- In (a) und (d) ist Adapt<sup>-</sup> das niedrigste  $A_i$  für welches man eine Architektur finden kann, welche die Anforderungen erfüllt. Adapt<sup>+</sup> ist das niedrigste  $A_i$ , dessen Grenzen  $Q_{A_iU}$  und  $Q_{A_iL}$  die Anforderungen erfüllt.
- Die Werte zeigen, dass die Erfüllung der Anforderungen eine Anpassungsfähigkeit von Adapt<sup>-</sup> voraussetzen und, dass jede Architektur die mindestens Adapt<sup>+</sup> hat die Anforderungen auch erfüllt. Für Anpassungsfähigkeit dazwischen gibt es Architekturen, die die Anforderungen erfüllen und solche die es nicht tun.

Adapt<sup>-</sup> und Adapt<sup>+</sup>

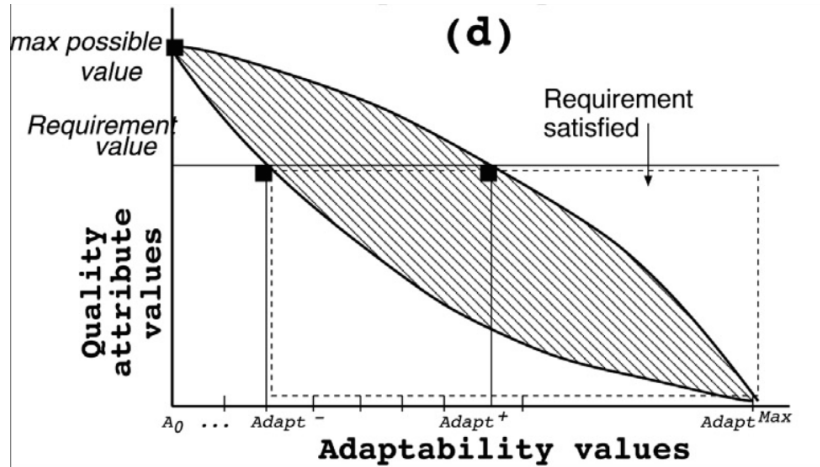
2015-02-04

## On the relationships between QoS and software adaptability at the architectural level

- └ Ansatz
  - └ Adapt<sup>-</sup> und Adapt<sup>+</sup>
    - └ Adapt<sup>-</sup> und Adapt<sup>+</sup>



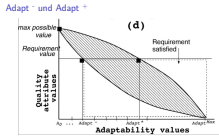
- In (a) und (d) ist Adapt<sup>-</sup> das niedrigste  $A_i$  für welches man eine Architektur finden kann, welche die Anforderungen erfüllt. Adapt<sup>+</sup> ist das niedrigste  $A_i$ , dessen Grenzen  $Q_{A_i U}$  und  $Q_{A_i L}$  die Anforderungen erfüllt.
- Die Werte zeigen, dass die Erfüllung der Anforderungen eine Anpassungsfähigkeit von Adapt<sup>-</sup> voraussetzen und, dass jede Architektur die mindestens Adapt<sup>+</sup> hat die Anforderungen auch erfüllt. Für Anpassungsfähigkeit dazwischen gibt es Architekturen, die die Anforderungen erfüllen und solche die es nicht tun.

Adapt<sup>-</sup> und Adapt<sup>+</sup>

2015-02-04

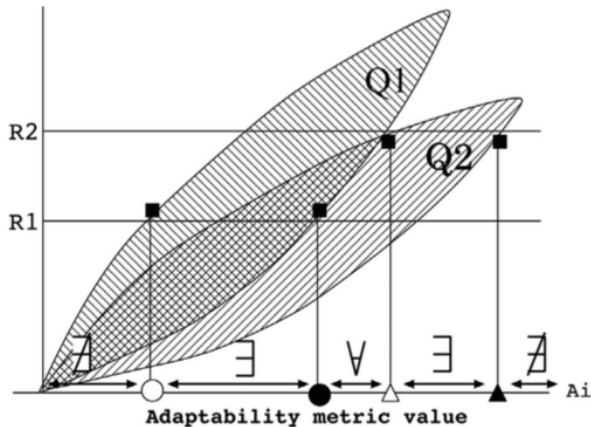
## On the relationships between QoS and software adaptability at the architectural level

└ Ansatz

└ Adapt<sup>-</sup> und Adapt<sup>+</sup>└ Adapt<sup>-</sup> und Adapt<sup>+</sup>

- In (a) und (d) ist Adapt<sup>-</sup> das niedrigste  $A_i$  für welches man eine Architektur finden kann, welche die Anforderungen erfüllt. Adapt<sup>+</sup> ist das niedrigste  $A_i$ , dessen Grenzen  $Q_{A_i,U}$  und  $Q_{A_i,L}$  die Anforderungen erfüllt.
- Die Werte zeigen, dass die Erfüllung der Anforderungen eine Anpassungsfähigkeit von Adapt<sup>-</sup> voraussetzen und, dass jede Architektur die mindestens Adapt<sup>+</sup> hat die Anforderungen auch erfüllt. Für Anpassungsfähigkeit dazwischen gibt es Architekturen, die die Anforderungen erfüllen und solche die es nicht tun.

## Mehrere Anforderungen



2015-02-04

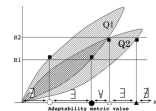
On the relationships between QoS and software adaptability at the architectural level

└ Ansatz

└─ Mehrere Anforderungen

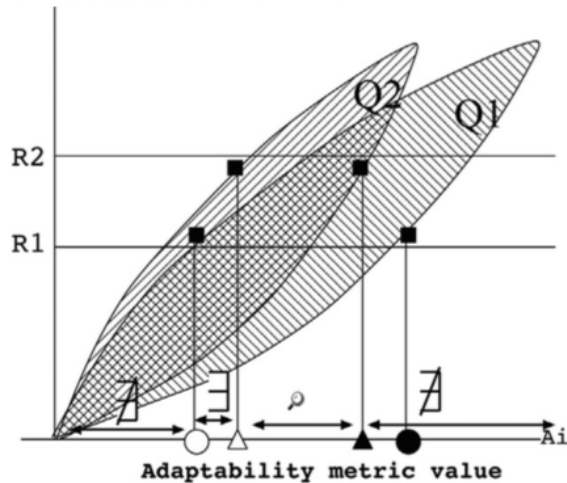
└─ Mehrere Anforderungen

Mehrere Anforderungen



- Es lassen sich bei Nutzung der gleichen Metrik zwei QoS in einen Graphen einzeichnen. Hierbei wird eine Fläche eingezeichnet, die die Werte bei allen möglichen Architekturen anzeigt. Es lassen sich Adapt+ und Adapt- für beide Qualitätsattribute einzeichnen, so entstehen (vielleicht) Bereiche in denen beide Anforderungen erfüllt sind, nur einer erfüllt ist oder keiner erfüllt ist.

## Mehrere Anforderungen

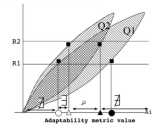


2015-02-04

On the relationships between QoS and software adaptability at the architectural level

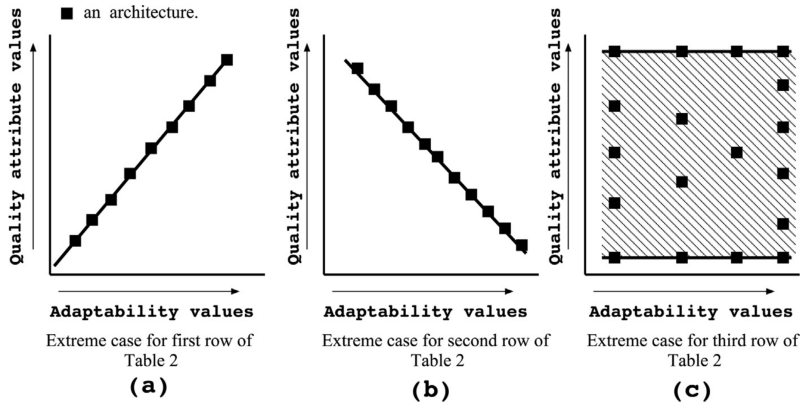
- └ Ansatz
  - └ Mehrere Anforderungen
    - └ Mehrere Anforderungen

Mehrere Anforderungen



- Es lassen sich bei Nutzung der gleichen Metrik zwei QoS in einen Graphen einzeichnen. Hierbei wird eine Fläche eingezeichnet, die die Werte bei allen möglichen Architekturen anzeigt. Es lassen sich Adapt+ und Adapt- für beide Qualitätsattribute einzeichnen, so entstehen (vielleicht) Bereiche in denen beide Anforderungen erfüllt sind, nur einer erfüllt ist oder keiner erfüllt ist.

# Beziehungen der QoS zur Anpassungsfähigkeit

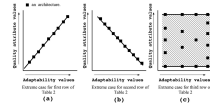


2015-02-04

On the relationships between QoS and software adaptability at the architectural level

- Ansatz
  - Beziehungen der QoS zur Anpassungsfähigkeit
    - Beziehungen der QoS zur Anpassungsfähigkeit

Beziehungen der QoS zur Anpassungsfähigkeit



- Für jedes System gelten unterschiedliche Beziehungen zwischen QoS und Anpassungsfähigkeit
- Wissen über die Beziehungen ermöglicht es den besten Kompromiss zu finden zwischen Anpassungsfähigkeit und Zielerfordernis
- Ziel der Analyse ist es zu zeigen, dass es eine Reihe von Möglichkeiten gibt ein System durch die Anwendung des Ansatzes zu entwerfen, welches die Anforderungen erfüllt und manchmal auch die gesamte Qualität und / oder Anpassbarkeit verbessert
- SOLAR (SOftware quaLities and Adaptability Relationships) ist ein Programm, welches den Ansatz umsetzt. Es hat jedoch performance probleme (bei 30 komponenten bis zu 20 minuten)

## Vorteile

- hilft die Architekturentscheidung zu rechtfertigen.
- dauert länger als bisherige Verfahren, aber das Resultat ist auch bei Änderungen weiterhin nutzbar.

2015-02-04

## On the relationships between QoS and software adaptability at the architectural level

### └ Analyse des Ansatzes

### └ Vorteile

- **Ziel:** Zu zeigen, dass es eine Reihe von Möglichkeiten gibt mithilfe des Ansatzes ein System zu entwerfen, welches die Anforderungen erfüllt und manchmal auch die gesamte QoS und / oder Anpassbarkeit zu verbessern.
- dauert länger als andere Ansätze, aber Erkenntnisse aus den anderen Ansätzen nutzlos sobald sich die Anforderungen ändern, hier nicht.
- Es muss lediglich die Asymptote der Anforderungen neu gezeichnet werden und die neuen Komponenten entsprechend ausgewählt werden.
  - neue Komponente: Ja, da es neue Möglichkeiten gibt
  - Komponente zerstört: Ja, wenn in Architektur
  - Komponente ändert QoS: Wenn es in der Architektur ist bei Verschlechterung Ja, ansonsten nein. Falls es nicht in der Architektur ist sollte er angewendet werden.
  - Die Anforderungen ändern sich: Wenn die Anforderungen

Vorteile

- hilft die Architekturentscheidung zu rechtfertigen.
- dauert länger als bisherige Verfahren, aber das Resultat ist auch bei Änderungen weiterhin nutzbar.

## Beschränkungen

- Weicher Erfüllungsgrad kann mit dem aktuellen Ansatz nicht vereint werden, da *Adapt*<sup>+</sup> und *Adapt*<sup>−</sup> in einem durchgehenden Erfüllbarkeitsschema nicht existieren würden
- Keine Gewichtung von Komponenten & Services
- Fehlendes Wissen über die tatsächliche Umgebung und die Schwierigkeit bei der Definition architektureller Parameter

2015-02-04

## On the relationships between QoS and software adaptability at the architectural level

### └ Analyse des Ansatzes

### └ Beschränkungen

#### Beschränkungen

- Weicher Erfüllungsgrad kann mit dem aktuellen Ansatz nicht vereint werden, da *Adapt*<sup>+</sup> und *Adapt*<sup>−</sup> in einem durchgehenden Erfüllbarkeitsschema nicht existieren würden
- Keine Gewichtung von Komponenten & Services
- Fehlendes Wissen über die tatsächliche Umgebung und die Schwierigkeit bei der Definition architektureller Parameter

- Es wird für den Ansatz generell nur eine binäre Erfüllung der Anforderungen genutzt (erfüllt, nicht erfüllt). Eine weichere Form kann mit dem aktuellen Ansatz nicht vereint werden, da *Adapt*<sup>+</sup> und *Adapt*<sup>−</sup> in einem durchgehenden Erfüllbarkeitsschema nicht existieren würden
- Bisher gibt es keine Gewichtung in der einige Komponenten, bzw Services wichtiger sein können als andere (WIP).
- Normale Probleme (lack of knowledge about the real world execution environment and consequently the difficulty in defining architecture parameters)



# Literatur



José Merseguer Diego Perez-Palacin Raffaella Mirandola. “On the relationships between QoS and software adaptability at the architectural level”. In: *The Journal of Systems and Software* (2013).



Lawrence Chung Nary Subramanian. *Metrics for Software Adaptability*. URL:  
<https://www.utdallas.edu/~chung/ftp/sqm.pdf>.

2015-02-04

On the relationships between QoS and software adaptability at the architectural level

└ Analyse des Ansatzes

└ Literatur

Literatur

📄 José Merseguer Diego Perez-Palacin Raffaella Mirandola. “On the relationships between QoS and software adaptability at the architectural level”. In: *The Journal of Systems and Software* (2013).

📄 Lawrence Chung Nary Subramanian. *Metrics for Software Adaptability*. URL:  
<https://www.utdallas.edu/~chung/ftp/sqm.pdf>.