

2015-01-24

## 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

24. Januar 2015

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

Daniel Schmidt

24. Januar 2015

# Inhalt

- 1 Einleitung
- 2 Anpassungsfähigkeit
- 3 Metriken
  - AAS und RAS
  - MAAS und MRAS
  - LSA
- 4 Adapt<sup>+</sup> und Adapt<sup>-</sup>
- 5 Beispiel
- 6 Analyse des Ansatzes
- 7 Beschränkungen
- 8 TODOs

2015-01-24

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

└─ Inhalt

### Inhalt

- Einleitung
- Anpassungsfähigkeit
- Metriken
  - AAS und RAS
  - MAAS und MRAS
  - LSA
- Adapt<sup>+</sup> und Adapt<sup>-</sup>
- Beispiel
- Analyse des Ansatzes
- Beschränkungen
- TODOs

# Einleitung

2015-01-24

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

└ Einleitung

└ Einleitung

Einleitung

- 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

An adaptable software system can tolerate changes in its environment without external intervention.

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

2015-01-24

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

└ Anpassungsfähigkeit

└ Anpassungsfähigkeit

Anpassungsfähigkeit

### Definition

An adaptable software system can tolerate changes in its environment without external intervention.

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

- Quantifizierung des Grads der Anpassungsfähigkeit wichtig
- Korrelation mit Ansatz von "Yang et al. (2009)", welche eine Abwägungsanalyse zwischen Qualitätsattributen und Anpassungsfähigkeit darstellt. Dabei nimmt dieser Ansatz Änderungen des Kontextes mit auf und die Entscheidung eine Anpassungsstrategie zur Laufzeit auszuführen wenn das System den aktuellen Kontext kennt
- Über heuristische Verfahren kann eine automatische Anpassung der Architektur erfolgen, hin zu einer Architektur, welche die Qualitätsmerkmale erfüllt oder nah dran ist
- Die Grundsätze des hier gewählten Ansatzes sind ähnlich derer in "Ägyed and Wile (2006)", obwohl das Ziel divergiert. Im Gegensatz zu diesen Verfahren wird die Menge der möglichen Architekturen reduziert, indem Ansätze, welche eine Einschränkung nicht erfüllen gelöscht werden



# Beispiel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ne tum quidem te respicies et cogitabis sibi quemque natum esse et suis voluptatibus? Dicam, inquam, et quidem discendi causa magis, quam quo te aut Epicurum reprehensum velim. Cum id fugiunt, re eadem defendunt, quae Peripatetici, verba. Egone non intellego, quid sit don Graece, Latine voluptas? Aliter enim nosmet ipsos nosse non possumus. Quis, quaeso, illum negat et bonum virum et comem et humanum fuisse? Duo Reges: constructio interrete.

2015-01-24

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

- └ Anpassungsfähigkeit
- └ Beispiel

Beispiel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ne tum quidem te respicies et cogitabis sibi quemque natum esse et suis voluptatibus? Dicam, inquam, et quidem discendi causa magis, quam quo te aut Epicurum reprehensum velim. Cum id fugiunt, re eadem defendunt, quae Peripatetici, verba. Egone non intellego, quid sit don Graece, Latine voluptas? Aliter enim nosmet ipsos nosse non possumus. Quis, quaeso, illum negat et bonum virum et comem et humanum fuisse? Duo Reges: constructio interrete.

# Metriken

## Definition ( $UC_i$ )

$UC_i$  = Komponenten die den Service i bereitstellen

## Definition ( $C_i$ )

$C_i$  = Komponenten die den Service bereitstellen können

2015-01-24

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

└ Metriken

└ Metriken

Metriken

Definition ( $UC_i$ )

$UC_i$  = Komponenten die den Service i bereitstellen

Definition ( $C_i$ )

$C_i$  = Komponenten die den Service bereitstellen können

# Metriken

- AAS und RAS
- MAAS und MRAS
- LSA

2015-01-24

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

└ Metriken

└ Metriken

- Der Ansatz basiert auf einer Component-and-Connector Ansicht, da sie allgemein verwendet wird um über die Qualitätswerte zur Laufzeit zu reden.

Metriken

- AAS und RAS
- MAAS und MRAS
- LSA

# AAS und RAS

Definition (Absolute adaptability of a service)

$$AAS_i = |UC_i|$$

2015-01-24

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

└ Metriken

└└ AAS und RAS

└└└ AAS und RAS

AAS und RAS

Definition (Absolute adaptability of a service)

 $AAS_i = |UC_i|$



# AAS und RAS

## Definition (Absolute adaptability of a service)

$$AAS_i = |UC_i|$$

## Definition (Relative adaptability of a service)

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

2015-01-24

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

└ Metriken

└ AAS und RAS

└ AAS und RAS

AAS und RAS

Definition (Absolute adaptability of a service)

$$AAS_i = |UC_i|$$

Definition (Relative adaptability of a service)

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

- **AAS:** misst die Anzahl der benutzten Komponenten, welche gewisse Dienste bereitstellen.
- **RAS:** misst die Anzahl der verwendeten Komponenten, welche einen gegebenen Service bereitstellen in hinsicht auf die Anzahl der Komponenten, die tatsächlich solchen Service anbieten.



# MAAS und MRAS

Definition (Mean of absolute adaptability of service)

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

2015-01-24

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

└─ Metriken

└─ MAAS und MRAS

└─ MAAS und MRAS

Definition (Mean of absolute adaptability of service)

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

# MAAS und MRAS

## Definition (Mean of absolute adaptability of service)

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

## Definition (Mean of relative adaptability of service)

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

2015-01-24

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

└ Metriken

└ MAAS und MRAS

└ MAAS und MRAS

MAAS und MRAS

Definition (Mean of absolute adaptability of service)

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

Definition (Mean of relative adaptability of service)

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

- **\*\*MAAS\*\*** (Mean of absolute adaptability of service): misst die durchschnittliche Nummer der genutzten Komponenten pro Dienstleistung.
- **\*\*MRAS\*\*** (Mean of relative adaptability of service): misst den Durchschnitt des RAS (Relative Adaptability of a service).

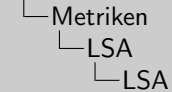
# LSA

## Definition (Level of system adaptability)

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

2015-01-24

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



LSA

Definition (Level of system adaptability)

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

- **\*\*LSA\*\*** (Level of system adaptability): misst die Anzahl der Komponenten die benutzt werden um das System abzubilden im Verhältnis zu der Anzahl der Komponenten die die anpassungsfähigste Architektur nutzen würde

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

2015-01-24

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

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

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

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

- Um die Bedeutung von Adapt<sup>+</sup> und Adapt<sup>-</sup> zu erklären wird sich auf die Abbildung Fig.4 bezogen: 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.

## Beispiel

2015-01-24

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

Beispiel

└ Beispiel

└ Beispiel

- 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<sup>+</sup> und Adapt<sup>-</sup> für beide Qualitätsattribute einzeichnen, so entstehen (vielleicht) Bereiche in denen beide Anforderungen erfüllt sind, nur einer erfüllt ist oder keiner erfüllt ist.

# Analyse des Ansatzes

2015-01-24

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

Analyse des Ansatzes

└ Analyse des Ansatzes

└ Analyse des Ansatzes

- 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.
- Der Ansatz dauert länger als andere Ansätze, die sich auf das Finden einer Architektur mit den höchsten Nutzen für die konkreten Systemvoraussetzungen. Allerdings sind die Erkenntnisse aus den anderen Ansätzen nutzlos sobald sich die Anforderungen ändern und die Analysen müssen wiederholt werden. Bei dem hier gewählten Ansatz muss lediglich die Asymptote der Anforderungen neu gezeichnet werden und dann die neuen Komponenten entsprechend ausgewählt werden.
  - Die Umgebung stellt eine neue Komponente bereit: Der Ansatz kann angewendet werden, da es neue Möglichkeiten gibt
  - Die Umgebung zerstört eine Komponente: Wenn die Komponente in der aktuellen Architektur nicht verwendet wird

# 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

2015-01-24

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

### └ Beschränkungen

### └ 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

- 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)
- 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 durchgehenderen 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 enviroment 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>.

2015-01-24

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

└ Beschränkungen

└ 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>.

# TODOs

- Notizen sind unter <https://github.com/DanielMSchmidt/software-architecture-presentation> zu finden
- Pro Thema entscheiden was an text auf die Folie soll und wie viele Folien zu dem Thema gehören
- Graphiken rendern / finden
- Beispiel übernehmen oder selbst überlegen
- Aufhänger überlegen

2015-01-24

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

└ TODOs

└ TODOs

### TODOs

- Notizen sind unter <https://github.com/DanielMSchmidt/software-architecture-presentation> zu finden
- Pro Thema entscheiden was an text auf die Folie soll und wie viele Folien zu dem Thema gehören
- Graphiken rendern / finden
- Beispiel übernehmen oder selbst überlegen
- Aufhänger überlegen