

ESE Kongress 2018

Vortragsskript:

Qualitätsanforderungen an Embedded-Software

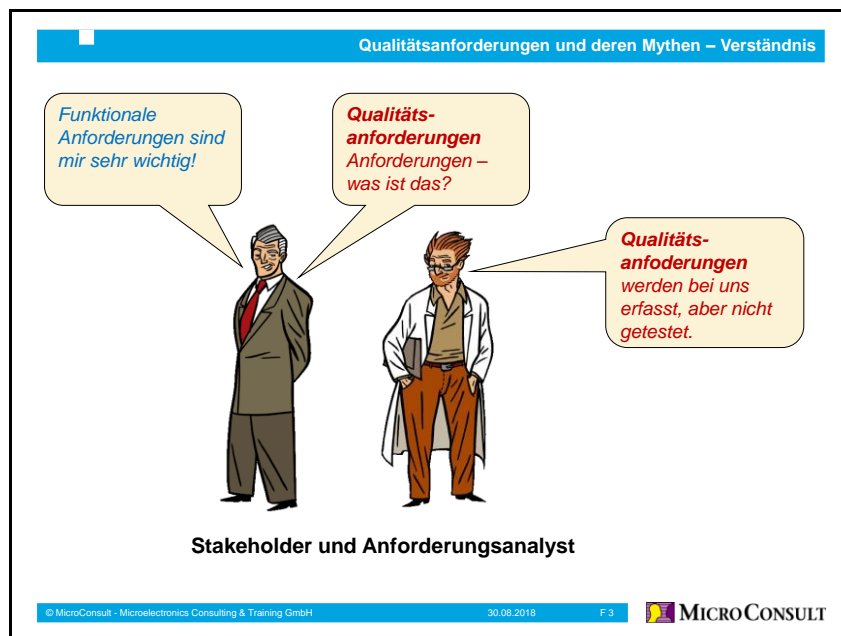
Die Kunst der hieb- und stichfesten Qualitätsanforderungen

Thomas Batt, MicroConsult GmbH

Anforderungen zu erfassen und zu verwalten ist ein wesentlicher Schlüssel zum Projekterfolg. Die Embedded-Software-Funktionalität lässt sich einfacher in Anforderungen beschreiben als die Qualitätsmerkmale. Dennoch: Qualitätsmerkmale lassen sich nicht am Ende einfach "hineintesten". Je abstrakter sie sind, desto aufwendiger ist ihre Erfassung. Welche Herausforderung stellt das für ein Projekt dar?

Qualitätsanforderungen und deren Mythen

Verständnis




Wichtige Voraussetzungen für die erfolgreiche Erfassung von Anforderungen:

- Der Anforderungsanalyst hat **alle Stakeholder identifiziert** und bezieht sie mit ein.
- Stakeholder und Anforderungsanalyst sind sich der **Wichtigkeit** der nichtfunktionalen Anforderungen für den Systemerfolg **bewusst**.
- Der Anforderungsanalyst lebt eine **strukturierte Vorgehensweise** zur Erhebung, Definition und Dokumentation von Anforderungen.
- Der Stakeholder **macht keine impliziten Annahmen**, die der Anforderungsanalyst nicht kennt.
- Erfahrungsbasierend fällt es dem Stakeholder und Anforderungsanalysten leicht, die Qualitätsanforderungen (nichtfunktionale Anforderungen) für ein System zu erheben und sie zu **quantifizieren** □ Entschärfung der abstrakten Qualitätsmerkmale zur Gewährleistung der Umsetzbarkeit und Testbarkeit.
- Verschiedene Stakeholder haben **ähnliche (gleiche) Sichtweisen** auf die Qualität.
- Verschiedene Stakeholder haben ein **ähnliches (gleiches) Verständnis** von Qualität.


Formulierung

Qualitätsanforderungen und deren Mythen – Formulierung

Der Kaffeevollautomat muss einen **niedrigen Standby-Leistungsverbrauch** aufweisen. 


↓ **Konkretes System-Qualitätsmerkmal** ↓

Der Kaffeevollautomat muss einen **Standby-Leistungsverbrauch < 5W** aufweisen.

Die Kaffeevollautomat-Software muss **wiederverwendbar** sein. 

↓ **Abstraktes Software-Qualitätsmerkmal** ↓

Die Kaffeevollautomat-Software muss **mindestens mit dem Grad 4 wiederverwendbar** sein.

© MicroConsult - Microelectronics Consulting & Training GmbH
30.08.2018
F 4



Konkrete Qualitätsmerkmale lassen sich sehr einfach durch eine konkrete Mengenangabe quantisieren, z.B. der Stromverbrauch eines Systems. Auch die Verifikation durch einen Testfall ist dabei sehr einfach.

Abstrakte Qualitätsmerkmale müssen vor ihrer Verwendung quantisiert werden. Das bedeutet, dass die Umsetzung und Verifikation für jede Quantisierung jedes abstrakten Qualitätsmerkmals einmalig spezifiziert sind. Z.B. sind für die Wiederverwendbarkeit Grad 4 alle für die Umsetzung relevanten Dinge und für die Verifikation relevanten Dinge spezifiziert.

Exemplarische Qualitätsmerkmale

Qualitätsanforderungen und deren Mythen – Exemplarische Qualitätsmerkmale

Qualitätsmerkmal	Subqualitätsmerkmal
Funktionalität	Angemessenheit, Richtigkeit, Interoperabilität, Ordnungsmäßigkeit
Zuverlässigkeit	Reife, Fehlertoleranz, Wiederherstellbarkeit
Benutzbarkeit	Verständlichkeit, Erlernbarkeit, Bedienbarkeit
Effizienz	Zeitverhalten, Verbrauchsverhalten
Änderbarkeit	Analysierbarkeit, Modifizierbarkeit, Stabilität, Prüfbarkeit
Übertragbarkeit	Anpassbarkeit, Installierbarkeit, Konformität, Austauschbarkeit, Wiederverwendbarkeit
Sicherheit	Funktionale Sicherheit, Angriffssicherheit

© MicroConsult - Microelectronics Consulting & Training GmbH
30.08.2018
F 5


ISO/IEC 9126-1:2001 Software Engineering – Product Quality – Part 1: Quality Model

Dieser Standard ist abgelöst durch: ISO/IEC 25010:2011 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuARE) – System and Software Quality Models

Wichtige Fakten zu Qualität:

- Unterscheidung: Produkt- und Prozessqualität – beide beeinflussen sich gegenseitig
- Unterscheidung: Innere (Entwicklersicht) und äußere Qualität (Kundensicht)
- Qualität kann nicht später hineingetestet werden
- Qualitätsmerkmale müssen vor Entwicklungsbeginn bestimmt sein
- Manche Qualitätsmerkmale unterstützen sich gegenseitig; z.B. beeinflussen sich Änderbarkeit und Wiederverwendbarkeit gegenseitig positiv
- Manche Qualitätsmerkmale sind gegensätzlich; z.B. beeinflussen sich Performance und Zuverlässigkeit gegenseitig negativ
- Richtlinie: Vier aus sechs Qualitätsmerkmalen zur Realisierung auswählen

Stellen wir uns ein Beispiel einer Anwendung und ihre Umsetzung vor: Schaffen Sie es, einen Formel-1-Rennwagen so zu realisieren, dass er als Drei-Liter- Auto und zum Containertransport geeignet ist?

Zuverlässigkeit

Die Zuverlässigkeit des Systems wird durch eine Menge von Submerkmalen beschrieben, die es ermöglichen, das System ausfallsicher zu betreiben.

Reife

Die Reife des Systems wird durch eine Menge von Merkmalen beschrieben, die die Fähigkeit des Systems spezifizieren, ein Versagen aufgrund von Fehlerzuständen im System zu vermeiden und die Fähigkeit/das Vermögen einer Organisation spezifizieren, die Abläufe mit effizienten und effektiven Prozessen zu gestalten.

Wiederherstellbarkeit

Die Wiederherstellbarkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die Fähigkeit/das Vermögen eines Softwareprodukts, die sich auf die Möglichkeit beziehen, bei einem Versagen das spezifizierte Leistungsniveau des Systems wiederherzustellen und die direkt betroffenen Daten wiederzugewinnen [nach ISO 9126, DIN 66272]

Fehlertoleranz

Die Fehlertoleranz des Systems wird durch eine Menge von Merkmalen beschrieben die bestimmen, wie tolerant das System gegenüber intern und extern auftretenden Fehlern ist.

Verfügbarkeit

Die Verfügbarkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die bestimmen, wie lange das System bezogen auf einen definierten Zeitraum im fehlerfreien Betrieb ist.

Benutzbarkeit

Die Benutzbarkeit des Systems wird durch eine Menge von Submerkmalen beschrieben, die sich auf den Aufwand beziehen, der zur Benutzung des Systems erforderlich ist, sowie auf die individuelle Bewertung einer solchen Benutzung durch eine festgelegte oder vorausgesetzte Benutzergruppe.

Verständlichkeit

Die Verständlichkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand für den Benutzer beziehen, um das Konzept und die Anwendung zu verstehen.

Erlernbarkeit

Die Erlernbarkeit wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand für den Benutzer ergeben, um das System zu erlernen.

Bedienbarkeit

Die Bedienbarkeit wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand für den Benutzer bei der Bedienung beziehen.

Effizienz

Die Effizienz des Systems wird durch eine Menge von Submerkmalen beschrieben, die ausdrücken, wie schnell das System bei welchem Ressourcenverbrauch arbeitet.

Zeitverhalten

Das Zeitverhalten des Systems wird durch eine Menge von Merkmalen beschrieben, die das Systemverhalten über die Zeit spezifizieren.

Echtzeitverhalten

Das Echtzeitverhalten des Systems wird durch eine Menge von Merkmalen beschrieben, die das Systemverhalten über der Zeit und die Einhaltung der Zeitvorgaben beschreiben. Bei hartem Echtzeitverhalten müssen in 100% aller Fälle alle Zeitvorgaben eingehalten werden, bei weicher Echtzeit weniger als 100%.

Verbrauchsverhalten

Das Verbrauchsverhalten des Systems wird durch eine Menge von Merkmalen beschrieben, die den Verbrauch von Ressourcen (z.B. Datenspeicher, Programmspeicher, CPU-Rechenleistung, ...) spezifizieren.

Änderbarkeit

Die Änderbarkeit des Systems wird durch eine Menge von Submerkmalen beschrieben, die sich auf den Aufwand für die Durchführung vorgegebener Änderungen des Systems beziehen.

Analysierbarkeit

Die Analysierbarkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand beziehen, um Mängel oder Ursachen von Fehlfunktionen und Versagen des zu erstellenden Systems zu diagnostizieren oder um änderungsbedürftige Teile zu bestimmen.

Modifizierbarkeit

Die Modifizierbarkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand beziehen, der zur Ausführung von Verbesserungen, zur Fehlerbeseitigung oder zur Anpassung an Umgebungsveränderungen notwendig ist.

Stabilität

Die Stabilität des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf das Risiko unerwarteter Auswirkungen von Änderungen beziehen.

Prüfbarkeit

Die Prüfbarkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand beziehen, der zur Prüfung des geänderten Systems notwendig ist.

Übertragbarkeit (Portabilität)

Die Übertragbarkeit des Systems wird durch eine Menge von Submerkmalen beschrieben, die sich auf den Aufwand beziehen, ein System (Software) in verschiedenen Umgebungen (Hardware) zu betreiben.

Installierbarkeit

Die Installierbarkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand beziehen, um das System in spezifizierten Umgebungen installierbar zu machen.

Konformität

Die Konformität des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand beziehen, der bewirkt, dass das System anwendungsspezifische Normen oder Vereinbarungen zur Übertragbarkeit erfüllt.

Austauschbarkeit

Die Austauschbarkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand beziehen, um das System oder Teile davon anstelle eines anderen spezifizierten Systems oder Teilsystems zum selben Zweck in der gleichen Umgebung nutzen zu können.

Anpassbarkeit

Die Anpassbarkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand beziehen, um das System an verschiedene festgelegte Umgebungen anzupassen, wenn nur Schritte oder Tätigkeiten unternommen oder Mittel eingesetzt werden, die für diesen Zweck für das betrachtete System vorgesehen sind.

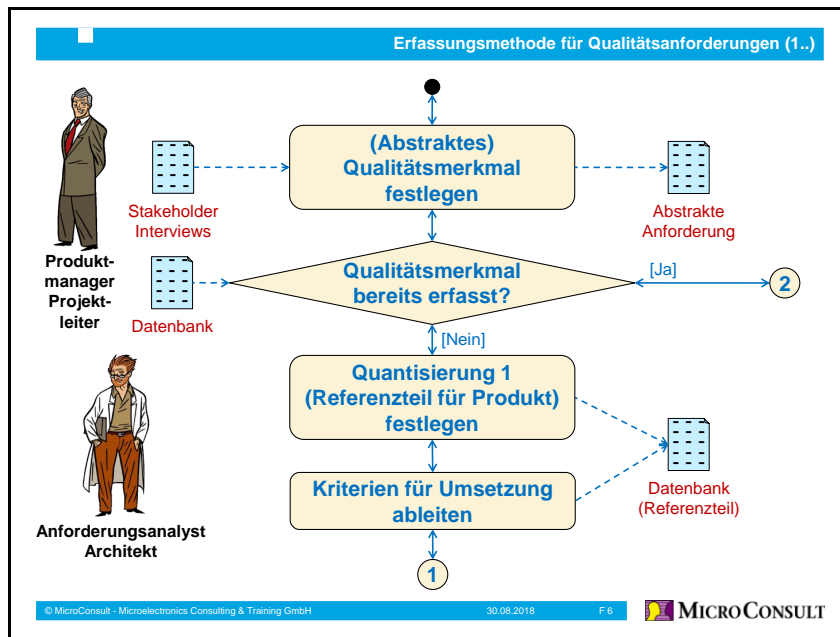
Wiederverwendbarkeit

Die Wiederverwendbarkeit des Systems wird durch eine Menge von Merkmalen beschrieben, die sich auf den Aufwand beziehen, um eine Wieder- und Weiterverwendung des Systems oder einzelner Systemteile für andere Systeme zu ermöglichen.

Sicherheit

Die funktionale Sicherheit des Systems wird durch eine Menge von Merkmalen beschrieben, die verhindern, dass Mensch und Umwelt durch ein Fehlverhalten oder eine Fehlbedienung des Systems gefährdet werden. Für die Sicherheit ist die Risikoklassifizierung des Systems sehr wichtig. [DIN V 19250: Risikoklassen 1-8; IEC61508: SIL1-4; EN 954-1 Safety Category 1-4]

Erfassungsmethode für Qualitätsanforderungen



Abstraktes Qualitätsmerkmal festlegen:

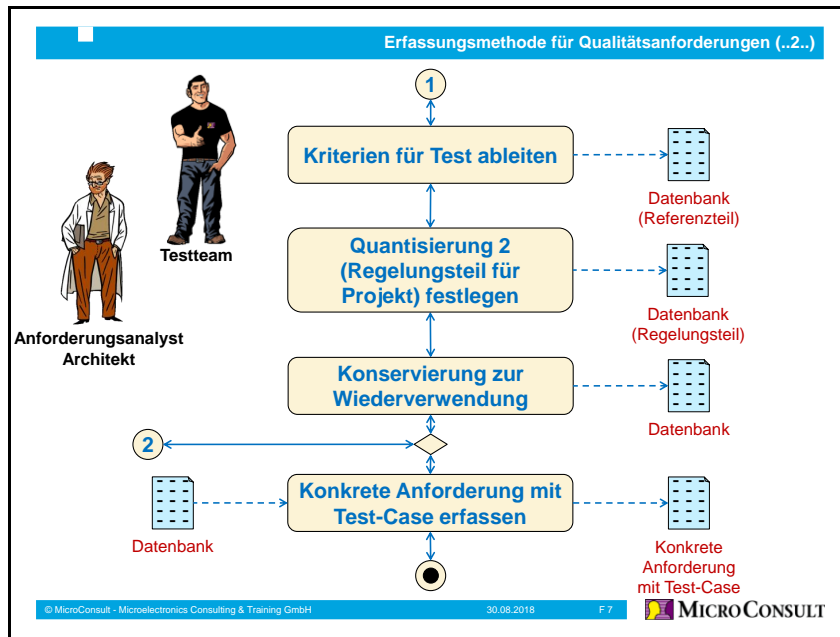
Zu jedem Qualitätsmerkmal formuliert der Anforderungsanalyst **erschließende Fragen** und zu jedem Subqualitätsmerkmal weiterführende Fragen. Daraus ergeben sich zunächst abstrakte Anforderungen. Er kann in der Datenbank prüfen, ob das Qualitätsmerkmal bereits quantisiert ist. Wenn ja, kann der Anforderungsanalyst direkt die konkrete Anforderung formulieren.

Quantisierung 1 (Referenzteil für Produkt) festlegen:

Der Anforderungsanalyst erarbeitet zusammen mit Fachexperten ein Quantisierungsschema.

Kriterien für die Umsetzung ableiten:

Aus dem Quantisierungsschema leiten sich Kriterien und Fakten für die Umsetzung der Anforderung ab.



Kriterien für den Test ableiten:

Aus dem Quantisierungsschema leiten sich Kriterien und Fakten für die Verifikation der Anforderung ab.

Quantisierung 2 (Regelungsteil) für Projekt festlegen:

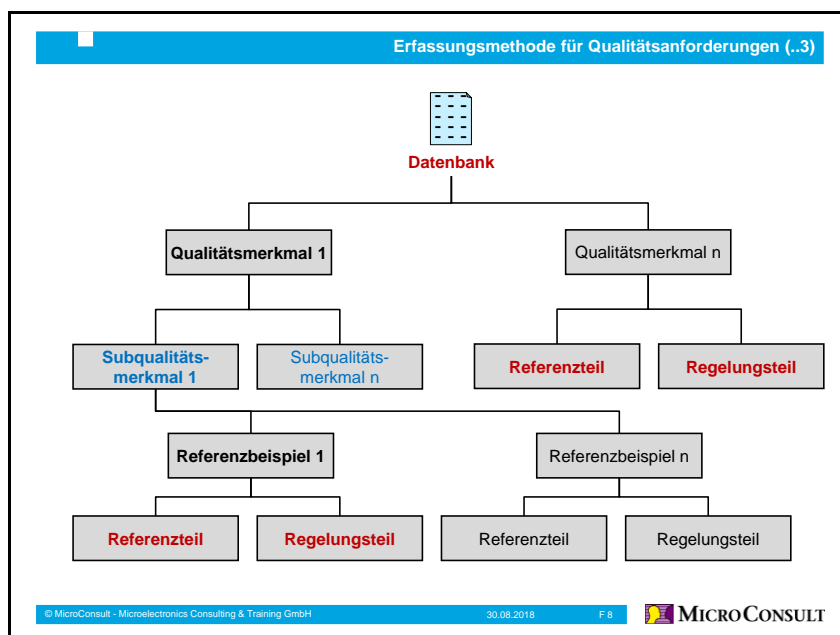
Viele Quantisierungsschemata erlauben dem Anforderungsanalysten eine projektspezifische Anpassung von Werten und oder Zusammenhängen.

Konservierung zur Wiederverwendung:

Neu entwickelte Quantisierungsschemata zu Qualitätsmerkmalen pflügt der Anforderungsanalyst zur Wiederverwendung in die Datenbank ein.

Konkrete Anforderung erfassen:

Auf Basis der bereits vorhandenen abstrakten Anforderung zusammen mit dem Quantisierungsschema für das betrachtete Qualitätsmerkmal erfasst der Anforderungsanalyst eine umsetzbare und nachweisbare Anforderung.



Die Referenzdatenbank ist möglichst generisch zu halten, damit sie den stetig wachsenden und immer wieder wechselnden Anforderungen aus Theorie und Praxis gerecht wird.

Allgemeiner Teil:

- Gültig und zu berücksichtigen bei der Erhebung aller Qualitätsmerkmale
- Allgemeine und merkmalsübergreifende Informationen und Anmerkungen
- Fachliche Ansprechpartner
- Weiterführende Literatur- und interne Linkhinweise

Referenzbeispiel:

Für die in der Datenbank enthaltenen Lösungen werden Referenzbeispiele erstellt. Zu jedem Referenzbeispiel gehören ein Referenzteil und ein Regelungsteil.

Referenzteil:

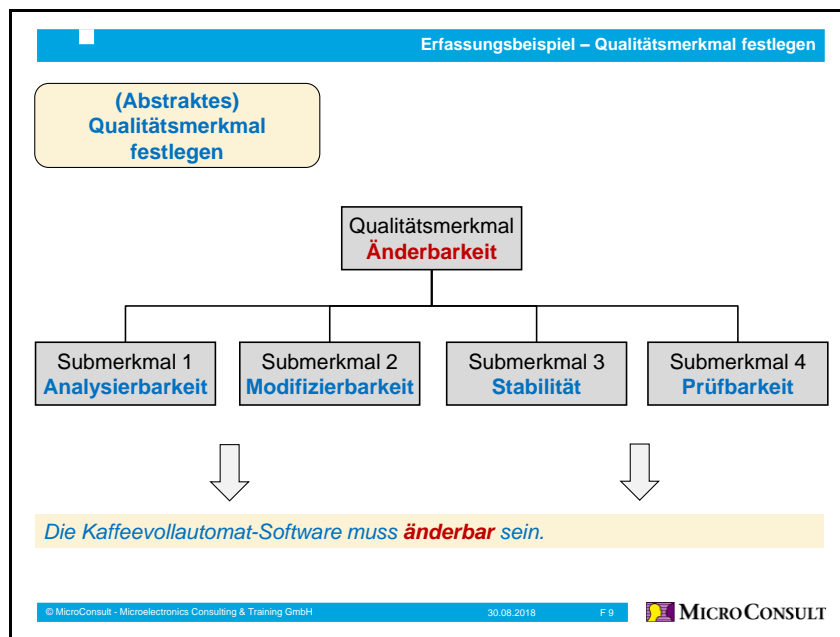
Der Referenzteil beinhaltet Kenngrößen und mögliche Wertebereiche zur Berechnung der entsprechenden Submerkmale.

Regelungsteil:

Im Regelungsteil wird der Lösungsweg zur konkreten Lösung anhand von Beispielen beschrieben. Am Ende des Regelungsteils kann die konkrete Anforderung für das entsprechende Merkmal stehen.

Erfassungsbeispiel

Qualitätsmerkmal festlegen



Zu jedem Qualitätsmerkmal formuliert der Anforderungsanalyst **erschließende Fragen**:

Änderbarkeit: Wollen Sie Kenngrößen zu den Merkmalen definieren, die sich auf den Aufwand beziehen, der zur Durchführung vorgegebener Änderungen des zu entwickelnden Systems notwendig sind?

Zu jedem Subqualitätsmerkmal formuliert der Anforderungsanalyst **weiterführende Fragen**:

Analysierbarkeit: Wollen Sie Kenngrößen für die Merkmale definieren, die sich auf den Aufwand beziehen, um Mängel oder Ursachen von einem Versagen des zu erstellenden Systems zu diagnostizieren oder um änderungsbedürftige Teile zu bestimmen?

Modifizierbarkeit: Wollen Sie Kenngrößen für die Merkmale definieren, die sich auf den Aufwand beziehen, der zur Ausführung von Verbesserungen, zur Fehlerbeseitigung oder zur Anpassung an Umgebungsveränderungen notwendig ist?

Stabilität: Wollen Sie Kenngrößen für die Merkmale definieren, die sich auf den Aufwand beziehen, der zur Prüfung der geänderten Hardware/Software/... notwendig ist?


Prüfbarkeit: Wollen Sie Kenngrößen für die Merkmale definieren, die sich auf den Aufwand beziehen, der zur Prüfung der geänderten Hardware/Software/... notwendig ist?

Quantisierung 1 (Referenzteil) festlegen

Erfassungsbeispiel – Quantisierung 1 Referenzteil festlegen (1..)

**Quantisierung 1
(Referenzteil für Produkt)
festlegen**

Metrik	Erklärung	Wertebereich	
		Min	Max
NSTAT	Anzahl der ausführbaren Anweisungen	1	50
DCAL	Anzahl der direkt aufgerufenen Komponenten	0	7
VG	Komplexität einer Funktion nach McCabe (Cyclomatic Number)	1	15
UJMP	Anzahl der direkten Sprünge (z.B. goto)	0	0
NOUT	Anzahl der Austritte (z.B. return)	1	1
LEVL	Anzahl der Ebenen (Schachtelungstiefe + 1)	1	5
COMF	Anzahl der Kommentare dividiert durch die Anzahl der Anweisungen	0.2	1
VOCF	Vokabel-Frequenz nach Halstead (Indikator für Redundanz)	1	4
AGS	Durchschnittliche Länge einer Anweisung nach Halstead	1	10

© MicroConsult - Microelectronics Consulting & Training GmbH
30.08.2018
F 10


Das folgende Beispiel basiert auf Veröffentlichung der DSF Deutsche Flugsicherung GmbH und Brandenburg Technische Universität Cottbus: „Pieper: Definition von Software-Qualitätsmerkmalen“ und „Bennicke, Dörr: New Metric Application“.

Die oben dargestellten Metriken sind ein Beispiel für einen Auszug aus der Referenzdatenbank zum Qualitätsmerkmal **Änderbarkeit**. Die Metriken und Grenzwerte können zur Bestimmung der Subqualitätsmerkmale **Analysierbarkeit**, **Modifizierbarkeit**, **Stabilität** und **Prüfbarkeit** herangezogen werden.


Erfassungsbeispiel – Quantisierung 1 Referenzteil festlegen (..2..)

**Quantisierung 1
(Referenzteil für Produkt)
festlegen**

Zuordnung der Metriken zu den Subqualitätsmerkmalen und deren Gewichtung:

Analysierbarkeit	= VG ^{GW} + NSTAT ^{GW} + COMF ^{GW} + AVGS ^{GW}
Modifizierbarkeit	= AVGS ^{GW} + LEVL ^{GW} + UJMP ^{GW} + VOCF ^{GW}
Stabilität	= NOUT ^{GW} + DCAL ^{GW} + UJMP ^{GW} + VOCF ^{GW}
Prüfbarkeit	= LEVL ^{GW} + UJMP ^{GW} + VG ^{GW}

GW = Gewichtungsfaktoren

© MicroConsult - Microelectronics Consulting & Training GmbH
30.08.2018
F 11


Nach Bestimmung der **Metriken** müssen diese den Submerkmalen **zugeordnet** werden:

Analysierbarkeit = a + b + c + ...
 Modifizierbarkeit = a + b + c + ...
 Stabilität = a + b + c + ...
 Prüfbarkeit = a + b + c + ...

Nach der Zuordnung der **Metriken** müssen diese **gewichtet** werden:

Analysierbarkeit = a * x + b * x + c * x + ...
 Modifizierbarkeit = a * x + b * x + c * x + ...
 Stabilität = a * x + b * x + c * x + ...
 Prüfbarkeit = a * x + b * x + c * x + ...

Die Summe aller Gewichtungsfaktoren pro Subqualitätsmerkmal muss 100 ergeben.

Beim Errechnen der Änderbarkeit wird eine Metrik mit „1“ bewertet, wenn der Messwert innerhalb des Wertebereichs liegt. Ist der Wertebereich nicht erfüllt, wird die Metrik mit „0“ bewertet.

Für das konkrete System werden im Regelungsteil die Wertebereiche der einzelnen Metriken und deren Gewichtung angepasst.

Erfassungsbeispiel – Quantisierung 1 Referenzteil festlegen (...3)

Quantisierung 1
(Referenzteil für Produkt)
festlegen

Bewertungsschema für jedes (Sub-) Qualitätsmerkmal:

Summe pro (Sub-) Qualitätsmerkmal	Erfüllungsgrad (Quantisierung)
100	3
67 – 99	2
34 – 66	1
0 – 33	0

Zuordnung der Subqualitätsmerkmale zum Qualitätsmerkmal:

Änderbarkeit

= Analysierbarkeit
 + Modifizierbarkeit
 + Stabilität
 + Prüfbarkeit

© MicroConsult - Microelectronics Consulting & Training GmbH 30.08.2018 F 12 MICROCONSULT

Im Referenzteil ist auf Basis der berechenbaren Ergebnisse die Quantisierung in [Erfüllungsgrad] festgelegt.

Beim Subqualitätsmerkmal ist die Bedeutung der Quantisierung eindeutig.

Definition: Beim Qualitätsmerkmal muss sich der Grad der Erfüllung auf jedes Subqualitätsmerkmal übertragen. Ist also der Erfüllungsgrad 4 auf Qualitätsmerkmalebene gefordert, so müssen alle Subqualitätsmerkmale ebenfalls diesen Grad erfüllen.

Quantisierung 2 (Regelungsteil) festlegen

Erfassungsbeispiel – Quantisierung 2 Regelungsteil festlegen (1..)

**Quantisierung 2
(Regelungsteil für
Projekt) festlegen**

Metrik	Erklärung	Wertebereich	
		Min	Max
NSTAT	Anzahl der ausführbaren Anweisungen	1	50
DCAL	Anzahl der direkt aufgerufenen Komponenten	0	7
VG	Komplexität einer Funktion nach McCabe (Cyclomatic Number)	1	15
UJMP	Anzahl der direkten Sprünge (z.B. goto)	0	0
NOUT	Anzahl der Austritte (z.B. return)	1	1
LEVL	Anzahl der Ebenen (Schachtelungstiefe + 1)	1	5
COMF	Anzahl der Kommentare dividiert durch die Anzahl der Anweisungen	0.2	1
VOCF	Vokabel-Frequenz nach Halstead (Indikator für Redundanz)	1	4
AGS	Durchschnittliche Länge einer Anweisung nach Halstead	1	10

© MicroConsult - Microelectronics Consulting & Training GmbH
30.08.2018
F 13

Im Regelungsteil erfolgt die projektspezifische **Anpassung** der **Wertebereiche** der **Metriken**.

Erfassungsbeispiel – Quantisierung 2 Regelungsteil festlegen (..2)

**Quantisierung 2
(Regelungsteil für
Projekt) festlegen**

Gewichtung der Subqualitätsmerkmale festlegen:

Analysierbarkeit = $VG*25 + NSTAT*25 + COMF*25 + AVGS*25$

Modifizierbarkeit = $AVGS*25 + LEVL*25 + UJMP*25 + VOCF*25$

Stabilität = $NOUT*25 + DCAL*25 + UJMP*25 + VOCF*25$

Prüfbarkeit = $LEVL*33 + UJMP*33 + VG*34$

Erfüllungsgrad (Quantisierung) festlegen:

Summe pro (Sub-) Qualitätsmerkmal	Erfüllungsgrad (Quantisierung)
67 – 99	2

© MicroConsult - Microelectronics Consulting & Training GmbH
30.08.2018
F 14

Im Regelungsteil erfolgt die projektspezifische **Anpassung** der **Gewichtung** der **Metriken** für jedes Qualitätsmerkmal.

Der Erfüllungsgrad ist in diesem Beispiel für alle Subqualitätsmerkmale identisch.

Konkrete Anforderung erfassen

Erfassungsbeispiel – Konkrete Anforderung erfassen

Konkrete Anforderung mit Test-Case erfassen

Die Kaffeevollautomat-Software muss mindestens mit dem Grad 2 analysierbar sein.

Die Kaffeevollautomat-Software muss mindestens mit dem Grad 2 modifizierbar sein.

Die Kaffeevollautomat-Software muss mindestens mit dem Grad 2 stabil sein.

Die Kaffeevollautomat-Software muss mindestens mit dem Grad 2 prüfbar sein.

Die Kaffeevollautomat-Software muss mindestens mit dem Grad 2 änderbar sein.

© MicroConsult - Microelectronics Consulting & Training GmbH
30.08.2018
F 15

Auf Basis dieser konkreten Software-Anforderung weiß der Softwareentwickler, welche Metriken er wie bei der Implementierung zu beachten hat. Der Softwaretester kann die Metriken ermitteln und damit verifizieren, ob eine Änderbarkeit von mindestens 2 erreicht ist.

Konkreten Test-Case erfassen

Erfassungsbeispiel – Konkreten Test-Case erfassen

Konkrete Anforderung mit Test-Case erfassen


REQ-ID	Requirement
SW-REQ-00x	Die Kaffeevollautomat-Software muss mindestens mit dem Grad 2 <u>änderbar</u> sein.
TC-ID	Test-Case
SW-TC-00y	Testvorbedingungen: <ul style="list-style-type: none"> ▪ Das Release der Kaffeevollautomat-Software liegt vor.
	Testauslösendes Ereignis: <ul style="list-style-type: none"> ▪ Die Analyse bezüglich Änderbarkeit der Kaffeevollautomat-Software erfolgt.
	Testergebnis erwartet: <ul style="list-style-type: none"> ▪ Die Kaffeevollautomat-Software ist mit dem Grad 2 <u>änderbar</u>.

© MicroConsult - Microelectronics Consulting & Training GmbH
30.08.2018
F 16

Auf Basis dieser konkreten Software-Anforderung weiß der Softwareentwickler, welche Metriken er wie bei der Implementierung zu beachten hat. Der Softwaretester kann die Metriken ermitteln und damit verifizieren, ob eine Änderbarkeit von mindestens 2 erreicht ist.

Resümee

Resümee



Bewusstsein und Verständnis
von Qualitätsanforderungen **schaffen**.


Die Kaffeefüllautomat-Software muss wiederverwendbar sein.

↓


Abstraktes Software-Qualitätsmerkmal

↓

Die Kaffeefüllautomat-Software muss mindestens mit dem Grad 4 wiederverwendbar sein.



Qualitätsanforderungen **implementierbar**
und **testbar formulieren**.

© MicroConsult - Microelectronics Consulting & Training GmbH
30.08.2018
F 17


Referenzierte und weiterführende Links

[1] MicroConsult Trainings auf www.microconsult.de:

- [Requirements Engineering und Management für Embedded-Systeme](#)
- [Funktionale Sicherheit \(Safety\) von Elektronik und deren Software](#)
- [Security: Sicherheit von Embedded-Systemen und ihr Kontext zur funktionalen Sicherheit](#)
- [Usability: Produkte benutzerfreundlich entwickeln](#)
- [SysML: Systemanalyse und Systemdesign mit der Systems Modeling Language](#)
- [UML-Praxis-Workshop: Praktischer Einsatz für Embedded- und Echtzeit-Softwareentwicklung](#)

[2] V-Modell: www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html

[3] Agiles Manifest: <http://agilemanifesto.org/iso/de/manifesto.html>

[4] Object Management Group (OMG): <http://www.omg.org>

[5] STEP (Standard for the Exchange of Product Model Data) ISO 10303: <http://www.iso.org>

[6] Neuro-Linguistisches Programmieren: https://de.wikipedia.org/wiki/Neuro-Linguistisches_Programmieren

[7] **Fachliteratur:** Requirements Engineering und Management, SOPHIST GmbH
[HANSER Verlag](#), Print-ISBN: 978-3-446-43893-4, E-Book-ISBN: 978-3-446-44313-6

Autor



Dipl.-Ing. (FH) Thomas Batt studierte nach seiner Ausbildung als Radio- und Fernstechniker Nachrichtentechnik. Seit 1994 arbeitet er kontinuierlich in verschiedenen Branchen und Rollen im Bereich Embedded-/Real-Time Systementwicklung. 1999 wechselte er zu MicroConsult. Dort verantwortet er heute als zertifizierter Trainer und Coach die Themenbereiche Systems /Software Engineering für Embedded-/Real-Time-Systeme sowie Entwicklungsprozess-Beratung.
t.batt@microconsult.de