

Qualitätskennzahlen für IT-Verfahren in der öffentlichen Verwaltung – Lösungsansätze zur Beschreibung von Metriken nach V-Modell XT

Von Stefan Bregenzer

Der Autor arbeitet im Bereich Softwaretest und beschäftigt sich als Qualitätsbeauftragter mit Themen zu Qualitätssicherung und IT-Sicherheit im Rahmen eines großen IT-Verfahren der Bundesfinanzverwaltung.

Inhaltsverzeichnis

1	Allgemeines.....	1
1.1	Begriffe und Abkürzungen	1
1.2	Definitionen	1
1.2.1	Messen.....	1
1.2.2	Metrik.....	1
1.2.3	Kennzahl	1
1.2.4	Messdatentypen	1
1.2.5	Messdaten.....	1
1.3	Grundvoraussetzungen.....	2
2	Metriken.....	3
2.1	Softwarequalitätsveränderung.....	3
2.1.1	Name der Metrik	3
2.1.2	Zielsetzung.....	3
2.1.3	Erläuterung.....	3
2.1.4	Zielgruppe	3
2.1.5	Definition	3
2.1.6	Messdatentypen	3
2.1.7	Auswertung	4
2.1.8	Verantwortliche	4
2.1.9	Verwendung.....	4
2.1.10	Darstellung	4
2.2	Softwarequalitätsveränderung - gewichtet.....	4
2.2.1	Name der Metrik	4
2.2.2	Zielsetzung.....	4
2.2.3	Erläuterung.....	4
2.2.4	Zielgruppe	5
2.2.5	Definition	5

2.2.6	Messdatentypen	6
2.2.7	Auswertung	6
2.2.8	Verantwortliche	6
2.2.9	Verwendung	6
2.2.10	Darstellung	6
2.3	Testqualität.....	6
2.3.1	Name der Metrik	6
2.3.2	Zielsetzung.....	6
2.3.3	Erläuterung.....	6
2.3.4	Zielgruppe	7
2.3.5	Definition	7
2.3.6	Messdatentypen	7
2.3.7	Auswertung	7
2.3.8	Verantwortliche	7
2.3.9	Verwendung	7
2.3.10	Darstellung	8
2.4	Testqualität - gewichtet	8
2.4.1	Name der Metrik	8
2.4.2	Zielsetzung.....	8
2.4.3	Erläuterung.....	8
2.4.4	Zielgruppe	8
2.4.5	Definition	8
2.4.6	Messdatentypen	9
2.4.7	Auswertung	9
2.4.8	Verantwortliche	9
2.4.9	Verwendung	9
2.4.10	Darstellung	9
3	Beschreibung der Messdatentypen	10
3.1	T_{ER}	10
3.1.1	Beschreibung	10
3.1.2	Messzeitpunkte	10
3.1.3	Datenquelle	10
3.1.4	Ablagestruktur für Messdaten.....	10
3.1.5	Verantwortlicher für die Erfassung und Ablage	10
3.2	$T_{ER}(K_n)$	10
3.3	T_{EBR}	10
3.4	$T_{EBR}(K_n)$	10
3.5	T_{FR}	11

3.5.1	Beschreibung	11
3.5.2	Messzeitpunkte	11
3.5.3	Datenquelle	11
3.5.4	Ablagestruktur für Messdaten.....	11
3.5.5	Verantwortlicher für die Erfassung und Ablage	11
3.6	$T_{FR}(K_n)$	11
3.7	T_{FBR}	11
3.8	$T_{FBR}(K_n)$	12
3.8.1	Beschreibung	12
3.9	T_{AR}	12
3.9.1	Beschreibung	12
3.9.2	Messzeitpunkte	12
3.9.3	Datenquelle	12
3.9.4	Ablagestruktur für Messdaten.....	12
3.9.5	Verantwortlicher für die Erfassung und Ablage	12
3.10	$T_{AR}(K_n)$	12
3.11	T_{ABR}	12
3.12	$T_{ABR}(K_n)$	13
4	Glossar	14
5	Anhang	15
5.1	Kriterien zur Klassifikation von Softwarefehlern	15

1 Allgemeines

In diesem Dokument werden Methoden vorgestellt, mit denen Qualität in einem IT-Projekt gemessen werden kann. Das Messen ist eine Grundvoraussetzung für die Steigerung der Qualität, da nur so eine objektive Aussage über die Auswirkung von projektspezifischen Veränderungen auf die Qualität getroffen werden kann.

1.1 Begriffe und Abkürzungen

In der Tabelle unter 4 Glossar sind die in diesem Dokument verwendeten Abkürzungen sowie die Bedeutung spezieller Begriffe aufgeführt.

1.2 Definitionen

1.2.1 Messen

Messen heißt **Zuordnung** von Zahlen oder Symbolen zu Eigenschaften von Dingen der realen Welt. Dazu sind konsistente Regeln erforderlich.

1.2.2 Metrik

Diese konsistenten Regeln stellen die sog. Metriken dar. Eine **Softwaremetrik**, oder kurz **Metrik**, ist eine (meist mathematische) Funktion, die eine Eigenschaft von Software in einem Zahlenwert, auch **Maßzahl oder Kennzahl** genannt, abbildet. Hierdurch werden formale Vergleichs- und Bewertungsmöglichkeiten geschaffen¹. Die Metrik ist also die Rechenregel zur Ermittlung einer Kennzahl.

Man unterscheidet hierbei in **direkte und indirekte Metriken**. Die direkte Metrik (z.B. Kosten, Lines of Code, Performance, Fehlerhäufigkeit) kann konkret - **ohne weitere Interpretation** - gezählt werden. Indirekte Metrik wie Qualität, Funktionalität, Effizienz, Zuverlässigkeit, hingegen erfordert eine **interpretierende Umsetzung der Messdaten** durch eine Funktion.

Im V-Modell XT ist die Metrik im Vorgehensbaustein „Messung und Analyse“ angesiedelt.

1.2.3 Kennzahl

Eine Kennzahl ist das **Ergebnis einer Metrik**.

1.2.4 Messdatentypen

Messdatentypen sind die Eingangsdaten, die zur Berechnung der Metriken benötigt werden. Sie sind die **Variablen oder Platzhalter in der mathematischen Funktion einer Metrik**.

1.2.5 Messdaten

Die **konkret gemessenen Daten** werden als Messdaten bezeichnet, während unter Messdatentypen die Definition verstanden wird.

¹ www.wikipedia.de Stichwort: Softwaremetrik

1.3 Grundvoraussetzungen

Die **Planungsphase einer Messung** sollte folgende Punkte berücksichtigen:

1. Die Regeln des Zählens werden festgelegt, bevor die Sammlung von Daten beginnt.
2. Die Metriken werden vorher klar definiert. Die Metrik muss:
 - ✓ einfach und berechenbar
 - ✓ empirisch und intuitiv überzeugend
 - ✓ konsistent und objektiv und
 - ✓ unabhängig von der Realisierung (Programmiersprache) sein.

Bei der anschließenden **Durchführung der Messung**, d.h. bei der Erhebung der Messdaten, sollte um Fehlerquellen zu vermeiden folgendes beachtet werden:

1. möglichst automatisierter Messvorgang
2. Einsatz statistischer Überprüfungen

2 Metriken

2.1 Softwarequalitätsveränderung

2.1.1 Name der Metrik

Softwarequalitätsveränderung (Δ SQ)

2.1.2 Zielsetzung

Die Kennzahl Softwarequalitätsveränderung (Δ SQ) gibt Aufschluss über die **Entwicklung der Softwarequalität eines Release in Bezug auf das Vorgängerrelease bzw. eines vorher festgelegten Release.**

2.1.3 Erläuterung

Nimmt die Softwarequalität ab, so wird die Kennzahl negativ. Steigt hingegen die Softwarequalität, so ist sie positiv.

2.1.4 Zielgruppe

Testteam

2.1.5 Definition

$$\Delta SQ = \frac{T_{BR} - T_R}{T_{BR}} [\%]$$

$$T_R = T_{FR} + T_{AR} + T_{ER}$$

Unter T_R wird die Anzahl der Tickets verstanden, die zu einem ausgewählten Release erfasst worden sind. Diese stammen entweder aus dem Funktionstest (Ticket-Attribut „Herkunft“ beim Incident-Ticket : Funktionstest), aus dem Abnahme- bzw. Bugfixtest (Ticket-Attribut „Herkunft“ beim Incident-Ticket : Abnahme) oder dem Echtbetrieb (Ticket-Attribut „Herkunft“ beim Incident-Ticket : Echtbetrieb).

$$T_{BR} = T_{FBR} + T_{ABR} + T_{EBR}$$

Das Bezugsrelease ist das Release, mit dem das gewählte Release verglichen wird. Die Summe der im Funktionstest, Abnahmetest, Bugfixtest und Echtbetrieb erfassten Tickets bildet die Anzahl der Tickets des Bezugsrelease (T_{BR}).

Die Kennzahl kann entweder bei einem **kompletten Release** (z.B. BEISPIELPROJEKT 8.2) oder **bezogen auf eine bestimmte Softwarekomponente** ermittelt werden. Dementsprechend müssen die Messdatentypen konkretisiert werden.

2.1.6 Messdatentypen

1. T_{ER}

2. T_{FR}
3. T_{AR}
4. T_{EBR}
5. T_{FBR}
6. T_{ABR}

2.1.7 Auswertung

Die Auswertung für das zweitneueste Release erfolgt jeweils nach der Übergabe des aktuellsten Release an den Echtbetrieb. Beispielsweise kann das Release BEISPIELPROJEKT_8.1.1 erst mit Übergabe des Release BEISPIELPROJEKT_8.2.1 an den Echtbetrieb überprüft werden.

2.1.8 Verantwortliche

QS-Beauftragte(r) und Vertreter(in) Testteam

2.1.9 Verwendung

Ein Bericht ist zurzeit nicht vorgesehen. Das Ergebnis wird als Managementwerkzeug der Leitung des Testteam zur Verfügung gestellt und dient der effektiven Steuerung.

2.1.10 Darstellung

Die Ergebnisse der Metrik werden in einer entsprechenden EXCEL-Arbeitsmappe mit Diagrammen ausgewertet und mit dem Titel „jjjjmmtt_qualitaetskennzahlen.xls“ gespeichert.

2.2 Softwarequalitätsveränderung - gewichtet

2.2.1 Name der Metrik

Softwarequalitätsveränderung - gewichtet (ΔSQ_G)

2.2.2 Zielsetzung

Die Kennzahl Softwarequalitätsveränderung - gewichtet (ΔSQ_G) gibt Aufschluss über die Entwicklung der Softwarequalität eines Release in Bezug auf das Vorgängerrelease bzw. eines vorher festgelegten Release. Der Unterschied zur Metrik Softwarequalitätsveränderung (ΔSQ) unter 2.1 ist, dass die Kennzahl nicht nur die Anzahl der Tickets, sondern auch andere Kriterien, wie z.B. die Fehlerklasse umfasst. Von ausschlaggebender Bedeutung für die Qualität der Software ist, ob gehäuft schwerwiegende Fehler auftreten oder ob lediglich „Schönheitsfehler“ gemeldet werden, die marginale Auswirkungen haben (vgl. 5 Kriterien zur Klassifikation von Softwarefehlern).

2.2.3 Erläuterung

Siehe 2.1.3

2.2.4 Zielgruppe

Siehe 2.1.4

2.2.5 Definition

$$\Delta SQ_G = \frac{T_{BRG} - T_{RG}}{T_{BRG}} [\%]$$

$$T_{RG} = T_{FRG} + T_{AR_G} + T_{ER_G}$$

Hierunter wird die gewichtete Anzahl der Tickets verstanden, die zu einem ausgewählten Release erfasst worden sind. Diese stammen entweder aus dem Funktionstest (Ticket-Attribut „Herkunft“ beim Incident-Ticket : Funktionstest), aus dem Abnahme- bzw. Bugfixtest (Ticket-Attribut „Herkunft“ beim Incident-Ticket : Abnahme) oder dem Echtbetrieb (Ticket-Attribut „Herkunft“ beim Incident-Ticket : Echtbetrieb). Die gewichtete Anzahl wird wie folgt ermittelt.

$$T_{FRG} = T_{FR}(K_n) \times \text{Gewichtung}_{K_n}$$

$$T_{AR_G} = T_{AR}(K_n) \times \text{Gewichtung}_{K_n}$$

$$T_{ER_G} = T_{ER}(K_n) \times \text{Gewichtung}_{K_n}$$

Dabei steht K_n für die verschiedene Ausprägungsformen eines Kriteriums (K). Beispielsweise kann K_1 die Fehlerklasse 1, K_2 die Fehlerklasse 2 und K_3 die Fehlerklasse 3 sein. Alle drei sind die möglichen Ausprägungsformen des Kriteriums Fehlerklasse (K) im Bugtrackingsystem, d.h. $(K_1, K_2, K_3, \dots, K_n) \in K$.

Die Gewichtung_{K_1} , Gewichtung_{K_2} und Gewichtung_{K_3} sind vorher vom Qualitätsbeauftragten bzw. dem Leiter des Teams-Abnahme festgelegte Prozentwerte, mit denen die Tickets, die das entsprechende Kriterium erfüllen, gewichtet werden. Die Anzahl der Tickets mit Fehlerklasse 1 kann z.B. die Gewichtung 100% und die Anzahl der Tickets mit Fehlerklasse 3 die Gewichtung 10% erhalten. Dies bedeutet, dass man 10 Tickets der Fehlerklasse 3 benötigt, um 1 Ticket der Fehlerklasse 1 aufzuwiegen.

$$T_{BRG} = T_{FBRG} + T_{ABRG} + T_{EBRG}$$

$$T_{FBRG} = T_{FBR}(K_n) \times \text{Gewichtung}_{K_n}$$

$$T_{ABRG} = T_{ABR}(K_n) \times \text{Gewichtung}_{K_n}$$

$$T_{EBRG} = T_{EBR}(K_n) \times \text{Gewichtung}_{K_n}$$

Das Bezugsrelease ist das Release, mit dem das gewählte Release verglichen wird. Die Summe der im Funktionstest, Abnahmetest, Bugfixtest und Echtbetrieb erfassten Tickets bildet die Anzahl der Tickets des Bezugsrelease (T_{BR}).

Die Kennzahl kann entweder bei einem **kompletten Release** (z.B. BEISPIELPROJEKT 8.2) oder bezogen **auf eine bestimmte Softwarekomponente** ermittelt werden. Dementsprechend müssen die Messdatentypen konkretisiert werden.

2.2.6 Messdatentypen

1. $T_{FR}(K_n)$
2. $T_{AR}(K_n)$
3. $T_{ER}(K_n)$
4. $T_{FBR}(K_n)$
5. $T_{ABR}(K_n)$
6. $T_{EBR}(K_n)$

2.2.7 Auswertung

Siehe 2.1.7

2.2.8 Verantwortliche

Siehe 2.1.8

2.2.9 Verwendung

Siehe 2.1.9

2.2.10 Darstellung

Siehe 2.1.10

2.3 Testqualität

2.3.1 Name der Metrik

Testqualität (TQ)

2.3.2 Zielsetzung

Die Kennzahl Testqualität (TQ) zeigt auf, **wie genau das Testteam** im Funktions-, Abnahme- und Bugfixtest **getestet** hat.

2.3.3 Erläuterung

Je **weniger Fehler** durch die Anwender im Echtbetrieb festgestellt werden, wenn das entsprechende Release nach durchgeführten Tests an den Betrieb übergeben worden ist, **desto höher** ist die Testqualität (TQ).

Es geht also in den jeweiligen Tests nicht darum, möglichst viele Fehler zu finden und somit möglichst viele Tickets zu erstellen. Dies wäre z.B. der Fall, wenn mit

den Entwicklern keine Rücksprache über die geplanten Prüfspezifikationen gehalten wird. Ziel ist es vielmehr **so zu testen**, das im **Echtbetrieb möglichst keine Softwarefehler** auftreten und mithin möglichst wenig Tickets aus dem Echtbetrieb gemeldet werden.

Es gilt: **Je höher** die Testqualität ist, **desto niedriger ist der Aufwand für Nachtests** in Form von Bugfixen. Mithin kann die Arbeitskraft in das Erstellen von Prüfspezifikationen für neue Anwenderforderungen bzw. ÄKDs gelegt werden.

2.3.4 Zielgruppe

Testteam

2.3.5 Definition

$$TQ = \frac{T_{FR} + T_{AR}}{T_{ER} + T_{FR} + T_{AR}} [\%]$$

Die Kennzahl kann entweder bei einem **kompletten Release** (z.B. BEISPIELPROJEKT 8.2) oder bezogen **auf eine bestimmte Softwarekomponente** ermittelt werden. Dementsprechend müssen die Messdatentypen konkretisiert werden.

2.3.6 Messdatentypen

1. T_{ER}
2. T_{FR}
3. T_{AR}

2.3.7 Auswertung

Die Auswertung für das zweitneueste Release erfolgt jeweils nach der Übergabe des aktuellsten Release an den Echtbetrieb. Beispielsweise kann das Release BEISPIELPROJEKT_8.1.1 erst mit Übergabe des Release BEISPIELPROJEKT_8.2.1 an den Echtbetrieb überprüft werden.

2.3.8 Verantwortliche

QS-Beauftragte(r) und Vertreter(in) Testteam

2.3.9 Verwendung

Ein Bericht ist zurzeit nicht vorgesehen. Das Ergebnis wird als Managementwerkzeug der Leitung des Testteam zur Verfügung gestellt und dient der effektiven Steuerung.

2.3.10 Darstellung

Die Ergebnisse der Metrik werden in einer entsprechenden EXCEL-Arbeitsmappe mit Diagrammen ausgewertet und mit dem Titel „jjjjmmtt_qualitaetskennzahlen.xls“ gespeichert.

2.4 Testqualität - gewichtet

2.4.1 Name der Metrik

Testqualität gewichtet (TQ_G)

2.4.2 Zielsetzung

Die Kennzahl Testqualität - gewichtet (TQ_G) zeigt auf, **wie qualitativ hochwertig das Abnahmeteam** im Funktions-, Abnahme- und Bugfixtest **getestet** hat.

Hierbei werden auch bestimmte Kriterien berücksichtigt und gewichtet, z.B. ob im Echtbetrieb ein schwerwiegender Fehler der Fehlerklasse 1 oder nur ein marginaler Fehler der Fehlerklasse 3 gefunden wurde.

2.4.3 Erläuterung

Siehe 2.3.3

2.4.4 Zielgruppe

Siehe 2.3.4

2.4.5 Definition

$$TQ_G = \frac{T_{FR_G} + T_{AR_G}}{T_{ER_G} + T_{FR_G} + T_{AR_G}} [\%]$$

Die Gewichtung erfolgt so:

$$T_{FR_G} = T_{FR_G}(K_1) \times Gewichtung_{K_1} + T_{FR_G}(K_2) \times Gewichtung_{K_2} + T_{FR_G}(K_3) \times Gewichtung_{K_3}$$

Allgemein:

$$T_{FR_G} = T_{FR}(K_n) \times Gewichtung_{K_n}$$

$$T_{AR_G} = T_{AR}(K_n) \times Gewichtung_{K_n}$$

$$T_{ER_G} = T_{ER}(K_n) \times Gewichtung_{K_n}$$

Wobei $K_1, K_2, K_3, \dots, K_n$ die verschiedenen Ausprägungsformen eines Kriteriums (K) sind. So kann K_1 die Fehlerklasse 1, K_2 die Fehlerklasse 2 und K_3 die Fehlerklasse 3 sein. Alle drei sind die möglichen Ausprägungsformen des Kriteriums Fehlerklasse (K) im Bugtrackingsystem, d.h. $(K_1, K_2, K_3, \dots, K_n) \in K$.

Die $Gewichtung_{K_1}, Gewichtung_{K_2}$ und $Gewichtung_{K_3}$ sind vorher festgelegte Prozentwerte, mit denen die Tickets, die das entsprechende Kriterium erfüllen,

gewichtet werden. Beispielsweise kann die Anzahl der Tickets mit Fehlerklasse 1 die Gewichtung 100% und die Anzahl der Tickets mit Fehlerklasse 3 die Gewichtung 10% erhalten. Dies bedeutet, dass man 10 Tickets der Fehlerklasse 3 im Testteam erfassen muss, um 1 Ticket der Fehlerklasse 1 im Echtbetrieb aufzuwiegen.

Die Kennzahl kann entweder bei einem **kompletten Release** (z.B. BEISPIELPROJEKT 8.2) oder bezogen **auf eine bestimmte Softwarekomponente** ermittelt werden. Dementsprechend müssen die Messdatentypen konkretisiert werden.

2.4.6 Messdatentypen

1. $T_{FR}(K_n)$
2. $T_{AR}(K_n)$
3. $T_{ER}(K_n)$

2.4.7 Auswertung

Siehe 2.3.7

2.4.8 Verantwortliche

Siehe 2.3.8

2.4.9 Verwendung

Siehe 2.3.9

2.4.10 Darstellung

Siehe 2.3.10

3 Beschreibung der Messdatentypen

3.1 T_{ER}

3.1.1 Beschreibung

Hierunter wird die Anzahl der Tickets aus dem betrachteten Release verstanden, die für dieses im Echtbetrieb erfasst wurden. Diese haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Echtbetrieb“.

3.1.2 Messzeitpunkte

Die Auswertung für das zweitneueste Release erfolgt jeweils nach der Übergabe des aktuellsten Release an den Echtbetrieb. Beispielsweise kann das Release BEISPIELPROJEKT_8.1.1 erst mit Übergabe des Release BEISPIELPROJEKT_8.2.1 an den Echtbetrieb überprüft werden.

3.1.3 Datenquelle

Abfrage aus unserem Bugtrackingsystem ASPE, einer modifizierten Form von [IBM Rational Change](#)

(cvtype='problem') and (aem_herkunft='Echtbetrieb') and ((aem_release='%1') or (aem_release match '%2'))

3.1.4 Ablagestruktur für Messdaten

Die Ablage erfolgt vorerst in der EXCEL-Arbeitsmappe „jjjmmmtt_testqualitaet.xls“.

3.1.5 Verantwortlicher für die Erfassung und Ablage

QS-Beauftragte(r) und Vertreter(in) Testteam

3.2 $T_{ER}(K_n)$

weiterhin gilt sinngemäß 3.1.2, 3.1.3, 3.1.4 und 3.1.5

Hierunter wird die Anzahl der Tickets aus dem betrachteten Release verstanden, die für dieses im Echtbetrieb erfasst wurden. Sie haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Echtbetrieb“. Weiterhin entsprechen sie der n. Ausprägungsform des Kriteriums (K). Beispielsweise haben sie alle die Fehlerklasse 1.

3.3 T_{EBR}

weiterhin gilt sinngemäß 3.1.2, 3.1.3, 3.1.4 und 3.1.5

Hierunter wird die Anzahl der Tickets aus dem Release verstanden, mit dem ein Vergleich erfolgt und die für dieses im Echtbetrieb erfasst wurden. Sie haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Echtbetrieb“.

3.4 $T_{EBR}(K_n)$

weiterhin gilt sinngemäß 3.1.2, 3.1.3, 3.1.4 und 3.1.5

Hierunter wird die Anzahl der Tickets aus dem Release verstanden, mit dem ein Vergleich erfolgt und die für dieses im Echtbetrieb erfasst wurden. Sie haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Echtbetrieb“. Weiterhin entsprechen sie der n. Ausprägungsform des Kriteriums (K), d.h. sie haben alle die Fehlerklasse 1.

3.5 T_{FR}

3.5.1 **Beschreibung**

Hierunter wird die Anzahl der Tickets aus dem betrachteten Release verstanden, die für dieses im Funktionstest erfasst wurden. Diese haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Funktionstest“.

3.5.2 **Messzeitpunkte**

Die Auswertung für das zweitneueste Release erfolgt jeweils nach der Übergabe des aktuellsten Release an den Echtbetrieb. Beispielsweise kann das Release BEISPIELPROJEKT_8.1.1 erst mit Übergabe des Release BEISPIELPROJEKT_8.2.1 an den Echtbetrieb überprüft werden.

3.5.3 **Datenquelle**

Abfrage aus unserem Bugtrackingsystem ASPE, einer modifizierten Form von [IBM Rational Change](#)

(cvtype='problem') and (aem_herkunft='Funktionstest') and ((aem_release='%1') or (aem_release match '%2'))

3.5.4 **Ablagestruktur für Messdaten**

Die Ablage erfolgt vorerst in der EXCEL-Arbeitsmappe „jjjjmmtt_testqualitaet.xls“.

3.5.5 **Verantwortlicher für die Erfassung und Ablage**

QS-Beauftragte(r) und Vertreter(in) Testteam

3.6 $T_{FR}(K_n)$

weiterhin gilt sinngemäß 3.5.2, 3.5.3, 3.5.4 und 3.5.5

Hierunter wird die Anzahl der Tickets aus dem betrachteten Release verstanden, die für dieses im Echtbetrieb erfasst wurden. Sie haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Funktionstest“. Weiterhin entsprechen sie der n. Ausprägungsform des Kriteriums (K), d.h. sie haben sie alle die Fehlerklasse 1.

3.7 T_{FBR}

weiterhin gilt sinngemäß 3.5.2, 3.5.3, 3.5.4 und 3.5.5

Hierunter wird die Anzahl der Tickets aus dem Release verstanden, mit dem ein Vergleich erfolgt und die für dieses im Funktionstest erfasst wurden. Diese Gruppe hat beim Attribut „Herkunft“ im Incident-Ticket den Wert „Funktionstest“.

3.8 $T_{FBR}(K_n)$

weiterhin gilt sinngemäß 3.5.2, 3.5.3, 3.5.4 und 3.5.5

3.8.1 **Beschreibung**

Hierunter wird die Anzahl der Tickets aus dem Release verstanden, mit dem ein Vergleich erfolgt und die für dieses im Funktionstest erfasst wurden. Diese haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Funktionstest“. Weiterhin entsprechen sie der n. Ausprägungsform des Kriteriums (K). Beispielsweise haben sie alle die Fehlerklasse 1.

3.9 T_{AR}

3.9.1 **Beschreibung**

Hierunter wird die Anzahl der Tickets aus dem betrachteten Release verstanden, die für dieses im Abnahme- bzw. Bugfixtest erfasst wurden. Diese haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Abnahme“.

3.9.2 **Messzeitpunkte**

Die Auswertung für das zweitneueste Release erfolgt jeweils nach der Übergabe des aktuellsten Release an den Echtbetrieb. Beispielsweise kann das Release BEISPIELPROJEKT_8.1.1 erst mit Übergabe des Release BEISPIELPROJEKT_8.2.1 an den Echtbetrieb überprüft werden.

3.9.3 **Datenquelle**

Abfrage aus unserem Bugtrackingsystem ASPE, einer modifizierten Form von [IBM Rational Change](#)

(cvtype='problem') and (aem_herkunft='Abnahme') and ((aem_release='%1') or (aem_release match '%2'))

3.9.4 **Ablagestruktur für Messdaten**

Die Ablage erfolgt vorerst in der EXCEL-Arbeitsmappe „jjjmmmtt_testqualitaet.xls“.

3.9.5 **Verantwortlicher für die Erfassung und Ablage**

QS-Beauftragte(r) und Vertreter(in) Testteam

3.10 $T_{AR}(K_n)$

weiterhin gilt sinngemäß 3.9.2, 3.9.3, 3.9.4 und 3.9.5

Hierunter wird die Anzahl der Tickets aus dem betrachteten Release verstanden, die für dieses im Abnahme- bzw. Bugfixtest erfasst wurden. Diese haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Abnahme“. Weiterhin entsprechen sie der n. Ausprägungsform des Kriteriums (K). Beispielsweise. haben sie alle die Fehlerklasse 1.

3.11 T_{ABR}

weiterhin gilt sinngemäß 3.9.2, 3.9.3, 3.9.4 und 3.9.5

Hierunter wird die Anzahl der Tickets aus dem Release verstanden, mit dem ein Vergleich erfolgt und die für dieses im Abnahme- bzw. Bugfixtest erfasst wurden. Diese haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Abnahme“.

3.12 $T_{ABR}(K_n)$

weiterhin gilt sinngemäß 3.9.2, 3.9.3, 3.9.4 und 3.9.5

Hierunter wird die Anzahl der Tickets aus dem Release verstanden, mit dem ein Vergleich erfolgt und die für dieses im Abnahme- bzw. Bugfixtest erfasst wurden. Diese haben beim Attribut „Herkunft“ im Incident-Ticket den Wert „Abnahme“. Weiterhin entsprechen sie der n. Ausprägungsform des Kriteriums (K). So haben z.B. alle die Fehlerklasse 1.

4 Glossar

Abkürzung	Bedeutung
ΔSQ	Softwarequalitätsveränderung
ΔSQ_G	Softwarequalitätsveränderung - gewichtet
TQ	Testqualität
TQ_G	Testqualität - gewichtet
K_n	Ausprägungsform eines Kriteriums K
$Gewichtung_{K_n}$	Gewichtung der Ausprägungsform des Kriteriums K in Prozent
T_R	Gesamte Anzahl der Tickets aus dem betrachteten Release
T_{R_G}	Gesamte Anzahl der Tickets aus dem betrachteten Release , die nach einem bestimmten Kriterium gewichtet ist
T_{BR}	Gesamte Anzahl der Tickets aus dem Bezugsrelease
T_{BR_G}	Gesamte Anzahl der Tickets aus dem Bezugsrelease , die nach einem bestimmten Kriterium gewichtet ist
T_{ER}	Anzahl der Tickets aus dem Echtbetrieb des betrachteten Release
T_{ER_G}	Anzahl der Tickets aus dem Echtbetrieb des Bezugsrelease , die nach einem bestimmten Kriterium gewichtet ist
T_{EBR}	Anzahl der Tickets aus dem Echtbetrieb des Bezugsrelease
T_{EBR_G}	Anzahl der Tickets aus dem Echtbetrieb des Bezugsrelease , die nach einem bestimmten Kriterium gewichtet ist
T_{FR}	Anzahl der Tickets aus dem Funktionstest des betrachteten Release
T_{FR_G}	Anzahl der Tickets aus dem Funktionstest des betrachteten Release , die nach einem bestimmten Kriterium gewichtet ist
T_{FBR}	Anzahl der Tickets aus dem Funktionstest des Bezugsrelease
T_{FBR_G}	Anzahl der Tickets aus dem Funktionstest des Bezugsrelease , die nach einem bestimmten Kriterium gewichtet ist
T_{AR}	Anzahl der Tickets aus dem Abnahme- und Bugfixtest des betrachteten Release
T_{AR_G}	Anzahl der Tickets aus dem Abnahme- und Bugfixtest des betrachteten Release , die nach einem bestimmten Kriterium gewichtet ist
T_{ABR}	Anzahl der Tickets aus dem Abnahme- und Bugfixtest des Bezugsrelease
T_{ABR_G}	Anzahl der Tickets aus dem Abnahme- und Bugfixtest des Bezugsrelease , die nach einem bestimmten Kriterium gewichtet ist

5 Anhang

5.1 Kriterien zur Klassifikation von Softwarefehlern

Fehlerklasse	Auswirkungen Software	Indikator
1	schwere Fehler: Beeinträchtigung der Arbeit, Showstopper, in der Regel technische Fehler	alternative Kriterien: a) Fehler, die dazu führen, dass die Software nicht genutzt werden kann. b) Fehler in zentralen Funktionen, die zum Abbruch der gesamten Anwendung führen. c) Fehler welche die Prüfung von weiteren Teilbereichen ausschließen (z.B. wenn die Anwendung ZB nicht funktioniert, kann auch ASF dort nicht getestet werden).
2	mittlere Fehler: Erschwerung der Arbeit	kumulative Kriterien: ✓ Fehler in der Anwendung, die nicht zum Abbruch führen, ✓ die nicht zur Fehlerklasse 1 gehören, ✓ die gleichwohl so erheblich sind, dass eine Echtbetriebsübergabe nicht zumutbar ist und ✓ die nicht mit organisatorischen Mitteln umgangen werden können.
3	leichte Fehler: Schönheitsfehler	kumulative Kriterien: ✓ Fehler, die nicht zur Fehlerklasse 1 und 2 gehören, ✓ die keine bedeutsame Auswirkung auf Funktionalität und Nutzbarkeit haben und ✓ bei denen jedoch trotzdem eine Fehlerkorrektur vor dem Echtbetriebsübergang wünschenswert, jedoch nicht zwingend ist.