Metriken - ein unverzichtbarer Begleiter für Software-Prozess-Verbesserungen

Dipl.-Math. Hermann Will

QADVICE | Software+System Qualität
Jamnitzerstr. 2, 81543 München

hermann.will@qadvice.de

Zusammenfassung. Im vorliegenden Beitrag werden auf Basis praktischer Erfahrung aus der Qualitätssicherung von Data Center Lösungen Architektur und Aufbau eines Metriksystems sowie ausgewählte Metriken zur Bewertung von Software-Entwicklungsprozessen und zur Steuerung von Prozessveränderungen vorgestellt.

1. Einleitung

Ständige Veränderung der Märkte und Technologien und zunehmende Komplexität der Systeme in der IT-Branche erfordern die kontinuierliche Überprüfung und Anpassung der Prozesse für die Entwicklung von Software und Software basierten Systemen. Damit diese Prozessanpassungen nicht zu einem Blindflug werden, ist ein Metriksystem erforderlich, mit dem die Qualität der Prozesse und ihre Verbesserung objektiv gemessen werden können.

Im vorliegenden Beitrag wird auf Basis langjähriger Erfahrungen aus der Qualitätssicherung von Data Center Lösungen dargestellt, welche Anforderungen ein solches Metriksystem erfüllen muss, wie es aufgebaut werden kann und welche Metriken die Veränderung von Prozessen begleiten sollten.

2. Anforderungen an das Metriksystem

Software-Metriken – im weiteren Verlauf dieses Beitrags nur noch als Metriken oder auch Kennzahlen bezeichnet – bilden Eigenschaften von Software-Produkten, -Projekten und -Prozessen auf Zahlenwerte ab, um deren Qualität beurteilen zu können. Sie müssen ohne großen Aufwand zeitnah ermittelbar sein und der Algorithmus für ihre Erhebung muss für alle Stakeholder nachvollziehbar und reproduzierbar sein. Metriken müssen in nachweisbarem funktionalen Zusammenhang zu den von ihnen gemessenen Eigenschaften stehen. Um diese letzte Anforderung erfüllen zu können, benötigt man in der Regel mehr als eine Metrik für eine Eigenschaft. So kann man z.B. aus hoher Fehlerdichte einer Softwarekomponente nicht notwendigerweise auf schlechte Qualität schließen: Gut strukturierter Code kann durchaus eine höhere Fehlerdichte aufweisen als sog. Bandwurm-Code für dieselbe Funktionalität. Ergänzend

zur Fehlerdichte müssen weitere Kennzahlen - in diesem Fall u.a. Code-Qualitätskennzahlen - zur Bewertung der Qualität herangezogen werden.

3. Aufbau eines Metriksystems

Um die Beurteilung der Qualität von Prozessen im IT-Bereich Metrik basiert durchführen zu können, werden Kosten- und Termindaten, Daten aus dem Configuration Management zum Produkt selbst und zu den zugehörigen Artefakten, Daten zu den Anforderungen sowie Test und Fehler bezogene Daten benötigt. Hier exemplarisch einige dieser sog. direkten Metriken, die man aus den genannten Verfahren gewinnen kann:

Verfahren	Metriken
Kostenmanagement	▪ Ausbildungskosten ▪ Mitarbeiterkosten nach Tätigkeitsarten ▪ Kosten für IT-Infrastruktur oder Lizenzen
Projektmanagement	▪ Plan- und Ist-Termine für Meilensteine ▪ Dauern und Aufwände für Projektphasen
Configuration Management	▪ Umfang bzw. Änderungsumfang von Code und Dokumenten ▪ Code-Kennzahlen wie z.B. Komplexität oder Code-Cloning-Rate
Anforderungsmanagement	▪ Priorität, Aufwand und Laufzeit von Anforderungen
Testmanagement	▪ Anzahl der Testfälle differenziert nach Priorität und Aufwand ▪ Anzahl automatisierter/manueller Testfälle
Fehlermanagement	▪ Anzahl der Fehler, differenziert nach Priorität, Analyse-Ergebnis, korrigierte Komponente, Laufzeit

Die zu ermittelnden Metriken mit ihren zugeordneten Attributen sind sorgfältig zu planen, da diese Festlegungen auch Auswirkungen auf die Auswahl und das Customizing der Lifecycle-Management-Werkzeuge haben. So kann man z.B. nur dann die Fehlerqualität in der Qualitätssicherung ermitteln, also den Anteil der gemeldeten Fehler, die zu einer Korrektur führen, wenn die im Bug-Tracking geführten Daten dies auch ermöglichen.

Durch geeignete Kombination der direkten Metriken zu sog. abgeleiteten Metriken kann ein umfassendes Metriksystem aufgebaut werden. Ein Beispiel für eine abgeleitete Metrik ist die oben schon erwähnte Fehlerdichte, also die in einer bestimmten Prüfphase entdeckte Fehlerzahl bezogen auf den Umfang des Prüflings. Um den Aufwand für die Ermittlung der Daten zu minimieren, sollten die Metriken Tool ge-

stützt ermittelt und zur leichten Kombination der Daten sowie zur Erfahrungsdatensicherung in einer „Metrik-Datenbank“ abgelegt werden (siehe Abb. 1).

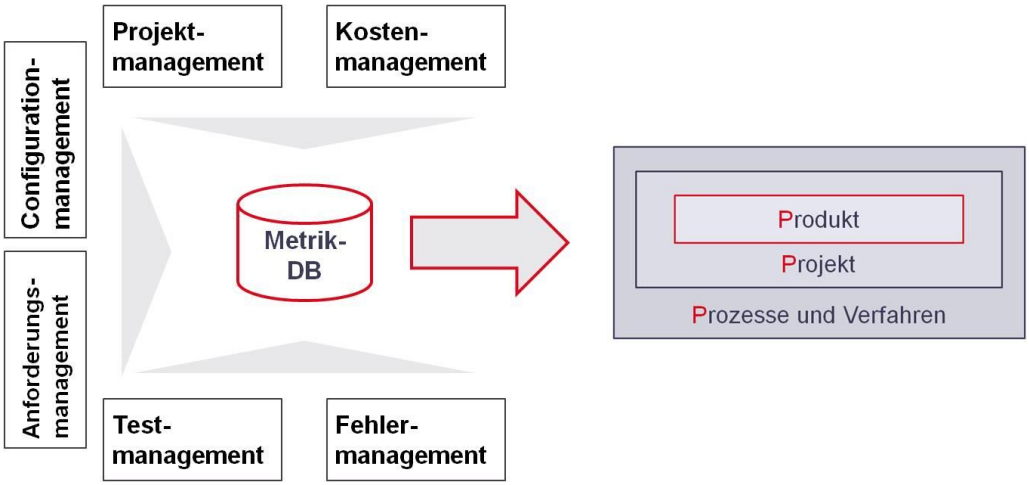


Abb. 1: Architektur eines umfassenden Metriksystems

Der Weg zu einem Metriksystem sollte dem in Abb. 2 skizzierten Prozess folgen. Zunächst sind alle Stakeholder und deren Ziele zu identifizieren. Ziele können z.B. die Verkürzung der Durchlaufzeit von Anforderungen oder die Verbesserung der Termintreue sein. Mit Hilfe von GQM (Goal Question Metric) oder einer ähnlichen Methode werden Metriken festgelegt, mit deren Hilfe die Zielerreichung beschrieben und kontrolliert werden kann. Die Standards und die Messmethoden zur Ermittlung der Metriken sind festzuschreiben, so dass die oben erhobenen Anforderungen an Metriken wie Reproduzierbarkeit und Nachvollziehbarkeit erfüllt sind. Wenn z.B. die Termintreue im Fokus ist, muss unmissverständlich definiert werden, wie die Zeitpunkte für den Start und das Ende eines Projekts festgelegt sind. Zu den Metriken sind anschließend realistische Zielwerte oder auch Schwellenwerte festzulegen wie z.B.: 80% Fehlerfindungsrate vor dem Systemtest und 15% im Systemtest. Damit Kennzahlen kein Selbstzweck werden, müssen Sie in das Berichtswesen integriert werden, und sie müssen als Erfahrungsdaten für nachfolgende Projekte und insbesondere für Prozessverbesserungsmaßnahmen weiter verwendet werden.

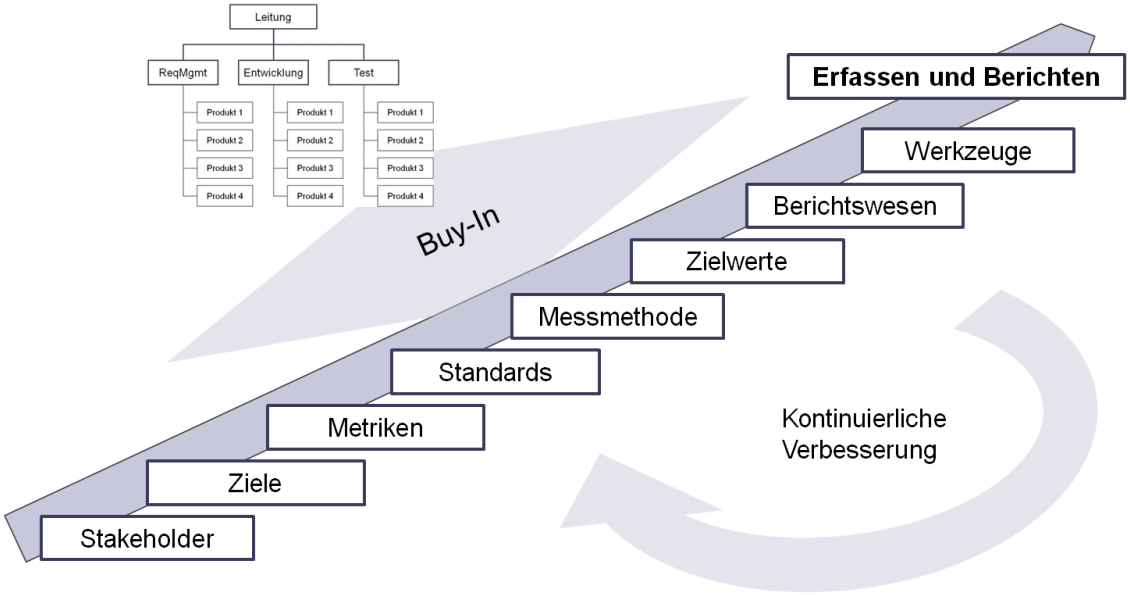


Abb. 2: Prozess zum Aufbau eines Metriksystems

4. Metriken zur Bewertung von Prozessen und zur Begleitung von Prozessveränderungen

Die Qualität eines Software-Entwicklungsprozesses manifestiert sich in Termin-, Budget- und Funktionstreue sowie in der Qualität der auf Basis dieses Prozesses entwickelten Produkte. Praxiserprobte Metriken, die diese Prozessmerkmale abbilden, sind die folgenden Metriken:

- Termintreue:
  Kann definiert werden durch „Anteil der Projekte, die ihren Zieltermin um weniger als 10% verfehlen“. Wichtig dabei ist, den Meilenstein für den Projektstart für alle Projekte eindeutig festzulegen. Für agile Methoden mit ihrem Timeboxing ist die Termintreue per se gegeben.
- Anforderungstreue:
  Eine mögliche Definition ist: „Anteil der Projekte, die ihre Anforderungen zu mindestens 90% realisieren“. Bei den schwergewichtigen Entwicklungsprozessen steht im Gegensatz zu den agilen Methoden die Anforderungstreue im Vordergrund.
- Aufwände und Kosten:
  Hier sind unterschiedliche Detaillierungen und Messmethoden möglich, wie z.B. Aufwände/Kosten pro Function Point oder Aufwände/Kosten pro Line of Code oder auch Aufwände/Kosten pro Fehler.
- Fehlerstrom:
  Diese Metrik erfasst die Verteilung der Fehlerfindung über die Phasen des Entwicklungsprozesses mit der Zielsetzung, einen großen Teil der Fehler in den frühen, entwicklungsnahen Phasen zu finden.

- Code-Qualität:
  Mit geeigneten Code-Kennzahlen wie z.B. Komplexität, Duplikatsrate und Komponentengröße ist die interne Qualität zu messen und bei Prozessveränderungen, wie z.B. Veränderung von Code-Ownerschaft im Rahmen agiler Vorgehensweisen, zu verfolgen.
- Feldfehler:
  Die Zahl der Feldfehler, also die Fehler nach allgemeiner Freigabe ist ein wichtiger Indikator für die externe Qualität und sollte natürlich kontinuierlich - insbesondere aber bei Veränderung von Prozessen - verfolgt werden.

Für reife Organisation sind diese 6 Metriken zur Bewertung von Prozessen und zur Steuerung von Prozessveränderungen unverzichtbar.

Literatur

[1] V. R. Basili, G. Caldiera, H. D. Rombach
THE GOAL QUESTION METRIC APPROACH
Encyclopedia of Software Engineering. John Wiley & Sons, 1994,
S. 528–532

[2] S. H. Kan
METRICS AND MODELS IN SOFTWARE QUALITY ENGINEERING
Addison-Wesley, Boston, MA, second edition, 2002

[3] K. H. Möller, D.J. Paulisch
SOFTWARE METRICS
Chapman & Hall Computing Series, 1993

[4] A. Spillner, T. Roßner, M. Winter, T. Linz
PRAXISWISSEN SOFTWARETEST - TESTMANAGEMENT
dpunkt.verlag 2008

[5] L. Westfall
12 STEPS TO USEFUL SOFTWARE METRICS
The Westfall Team, 2005

[6] H. Will
METRIKEN AUS DER PRAXIS FÜR DIE PRAXIS
Postproceedings zur ASQT 2011, bislang nicht veröffentlicht

QADVICE Software+System Qualität Beratung und Training

19. GI-WIVM Workshop: Qualitätsmanagement und Vorgehensmodelle

Metriken - ein unverzichtbarer Begleiter für Software-Prozess-Verbesserungen

Hermann Will

Düsseldorf, 12.09.2012

hermann.will@qadvice.de

Agenda

Einführung

Prozessmetriken

Aufbau eines Metriksystems

QADVICE Software+System Qualität Beratung und Training

Einführung

Prozessmetriken

Aufbau eines Metriksystems

Metrik = quantifizierte Beschreibungen von Eigenschaften

Je höher die
Fehlerdichte,
um so schlechter ist der Code.

Je höher der
Testautomatisierungsgrad,
um so effektiver ist der
Testprozess.

Je höher das
Fehleraufkommen im Test,
um so besser die Qualität der Software nach Auslieferung.

QADVICE Software+System Qualität Beratung und Training

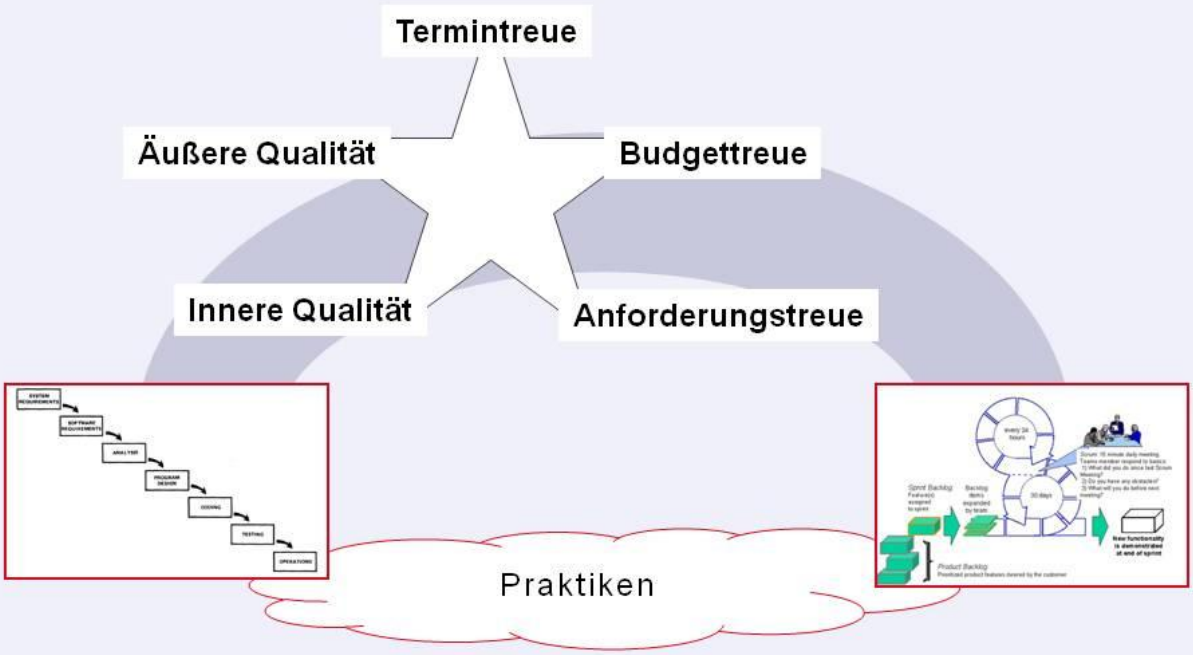
Einführung

Prozessmetriken

Aufbau eines Metriksystems

QADVICE Software+System Qualität Beratung und Training

Prozess-Qualitätsindikatoren



.....treue

- Termintreue:
  Anteil der Projekte, die ihren Zieltermin um weniger als 10% verfehlen
- Anforderungstreue:
  Anteil der Projekte, die ihre Anforderungen zu mindestens 90% realisieren
- Budgettreue:
  Anteil der Projekte, die ihre Budgetziele um weniger als 10% verfehlen

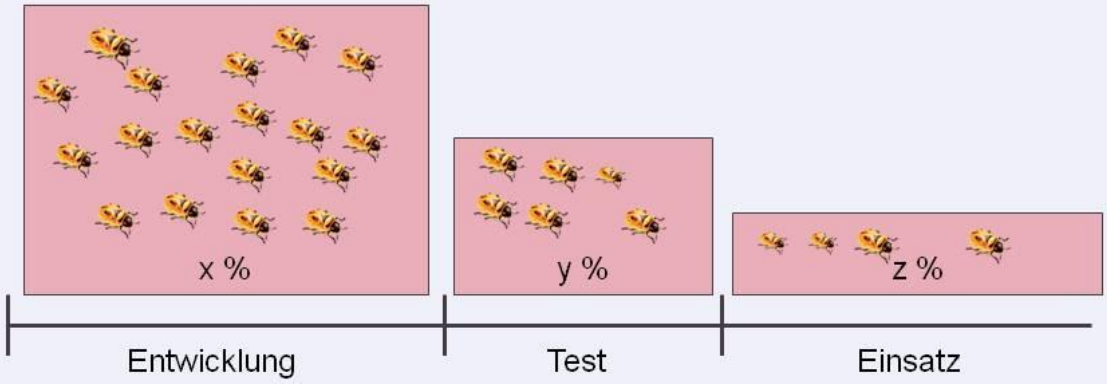
Innere Qualität

- Qualität des Codes
  - Komplexität
  - Clone-Rate
  - Anteil „Gott-Dateien“
  - .....
- Qualität der Architektur

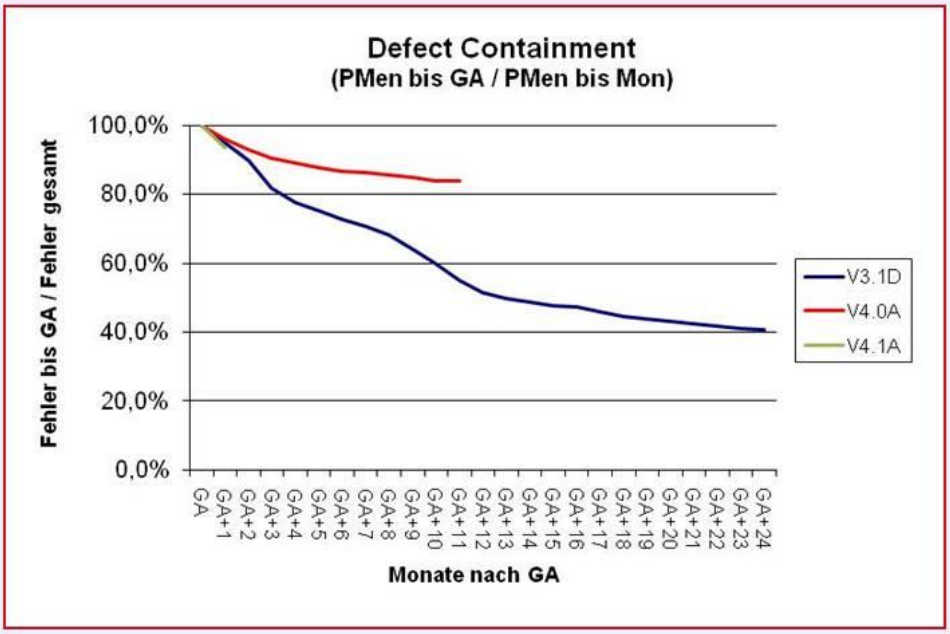
Äußere Qualität

- Effizienz
- Zuverlässigkeit

Fehlerstrommodell

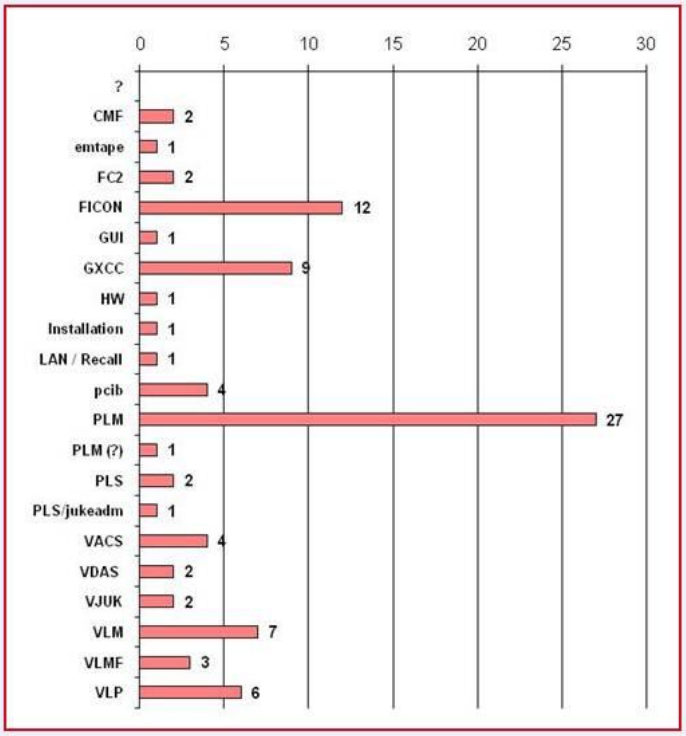


Defect Containment



(GA: General Availability)

Defect Prevention durch Escape-Analysen



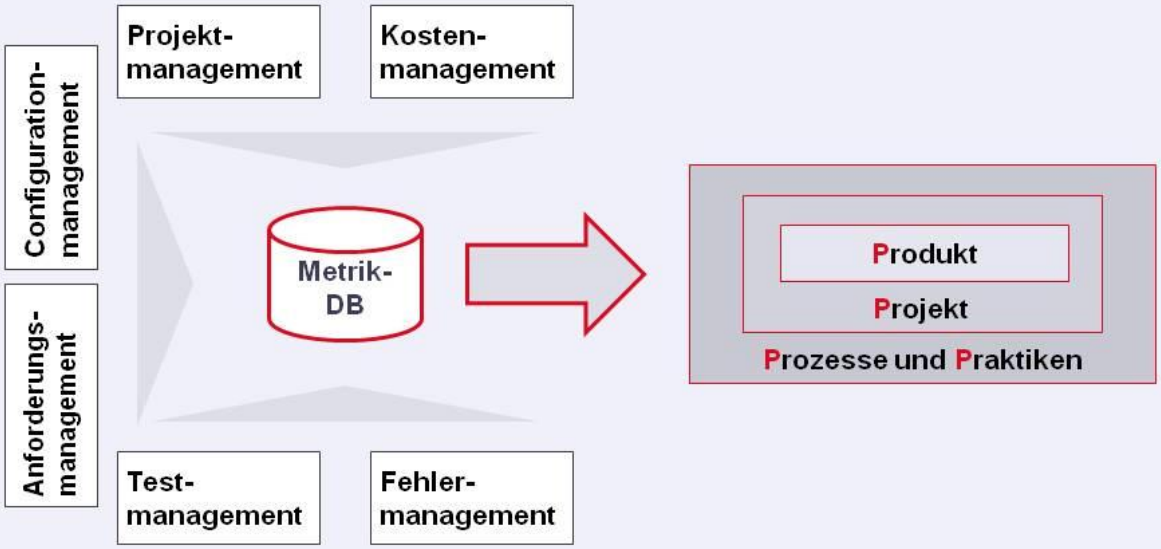
Einführung

Prozessmetriken

Aufbau eines Metriksystems

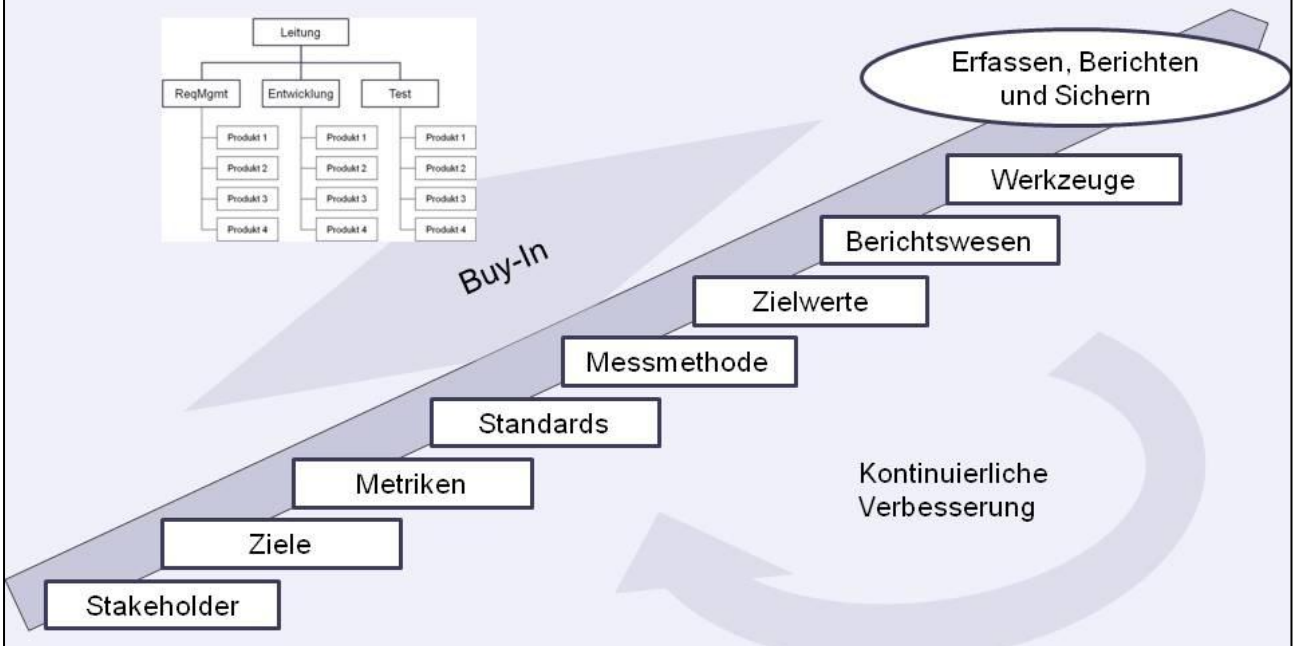
QADVICE Software+System Qualität Beratung und Training

Metriksystem Architektur



QADVICE Software+System Qualität Beratung und Training

Entwicklungsstufen für ein Metriksystem




QADVICE
Software+System Qualität
Beratung und Training
Vielen Dank für Ihre Aufmerksamkeit!
Hermann Will
hermann.will@qadvice.de
www.qadvice.de
12.09.2012
Copyright Hermann Will 2012
Folie 15