



**Fachgruppe Test,
Analyse und
Verifikation von
Software (TAV)**

Test- und Qualitätsmetriken - aus der Praxis für die Praxis

Hermann Will, 18.11.2011

hermann.will@qadvice.de

Agenda

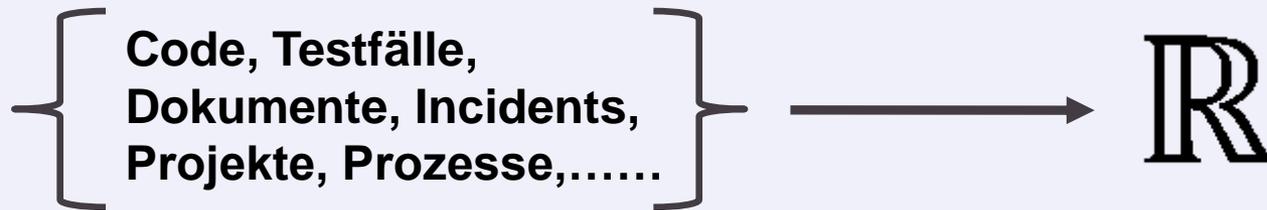
Einführung

Aufbau Metriksystem

Beispiele

Do's und Don'ts

Metrik = quantifizierte Beschreibungen von Eigenschaften

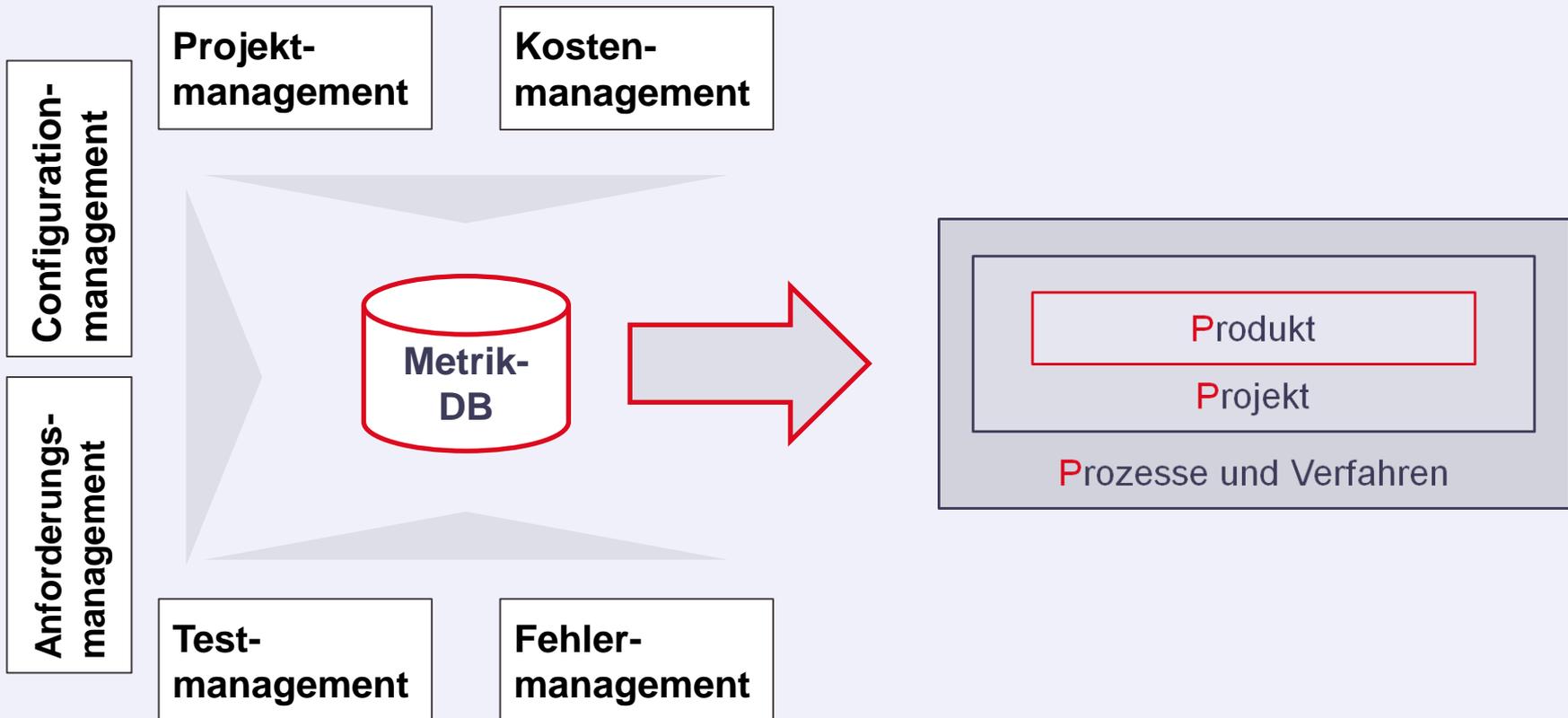


Je höher die **Fehlerdichte**, um so schlechter ist der Code.

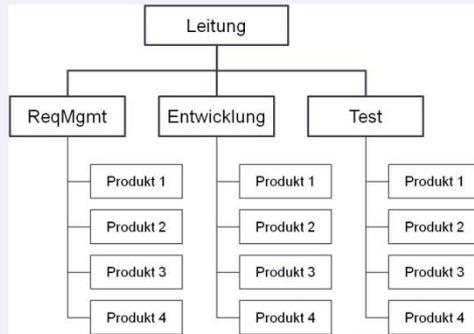
Je höher der **Testautomatisierungsgrad**, um so effektiver ist der Testprozess.

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

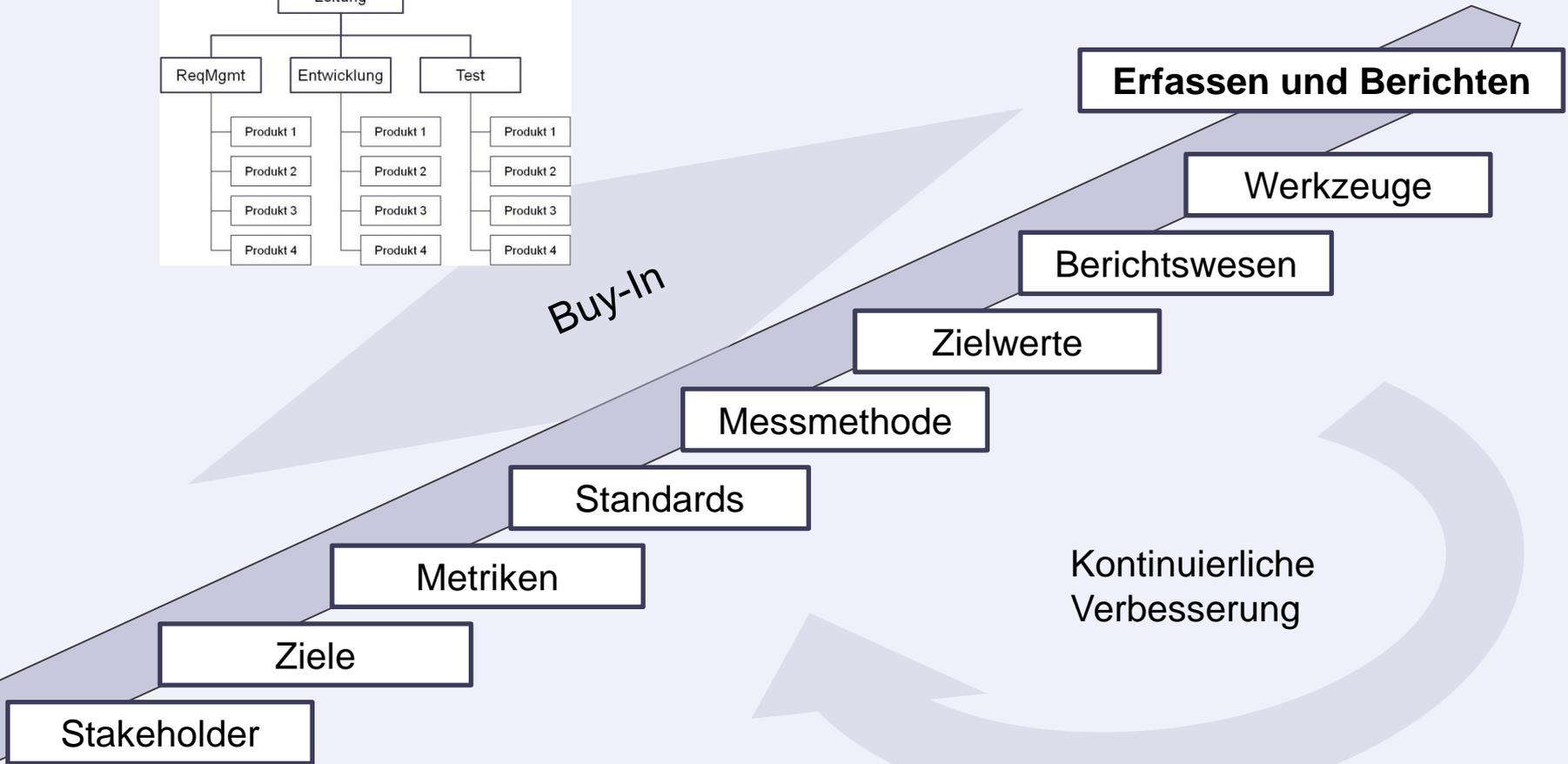
Metriksystem Architektur



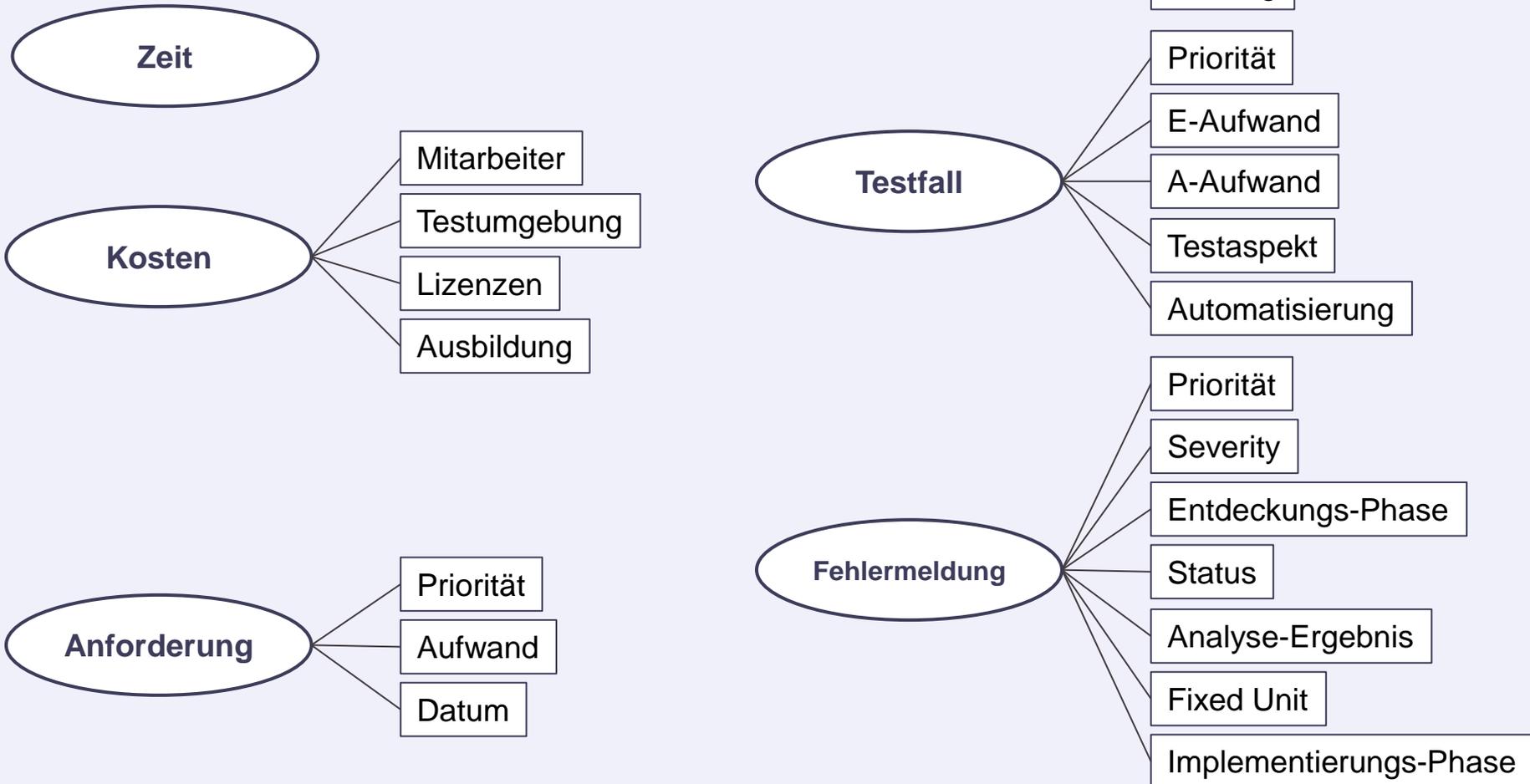
Entwicklungsstufen für ein Metriksystem



Buy-In



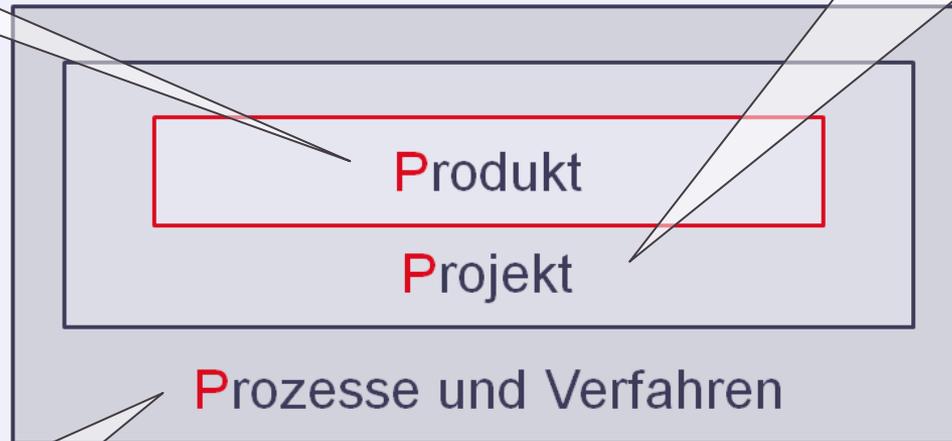
Direkte Metriken



P-P-P Metrikset

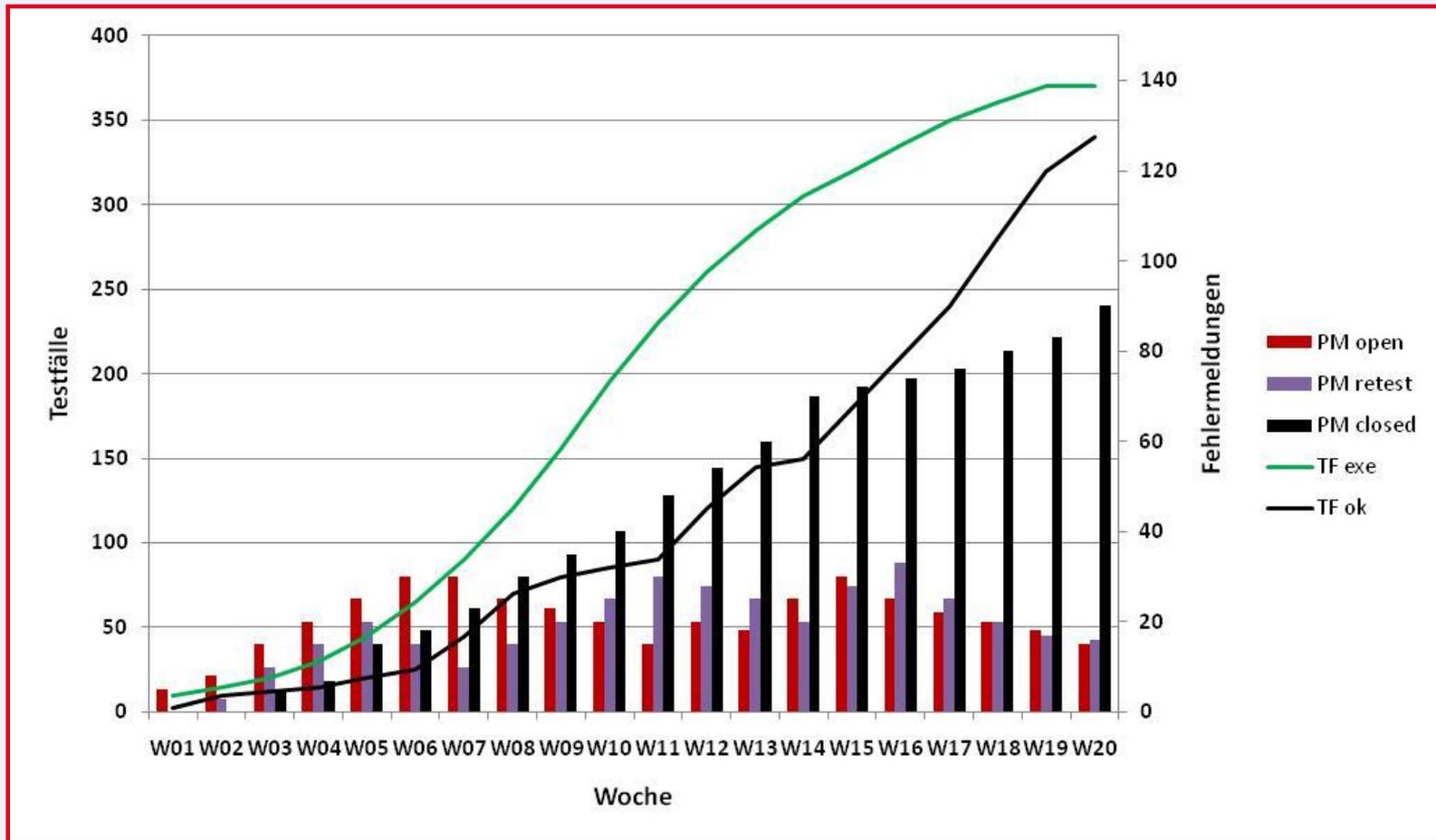
- Code-Qualität
- Feldfehler
- Testabdeckung

- Änderungsumfang
- Änderungslokalität
- Testfallentwicklung
- Testfallabwicklung
- Fehlermeldungseingang
- Fehlermeldungslaufzeit
- Fehlerqualität

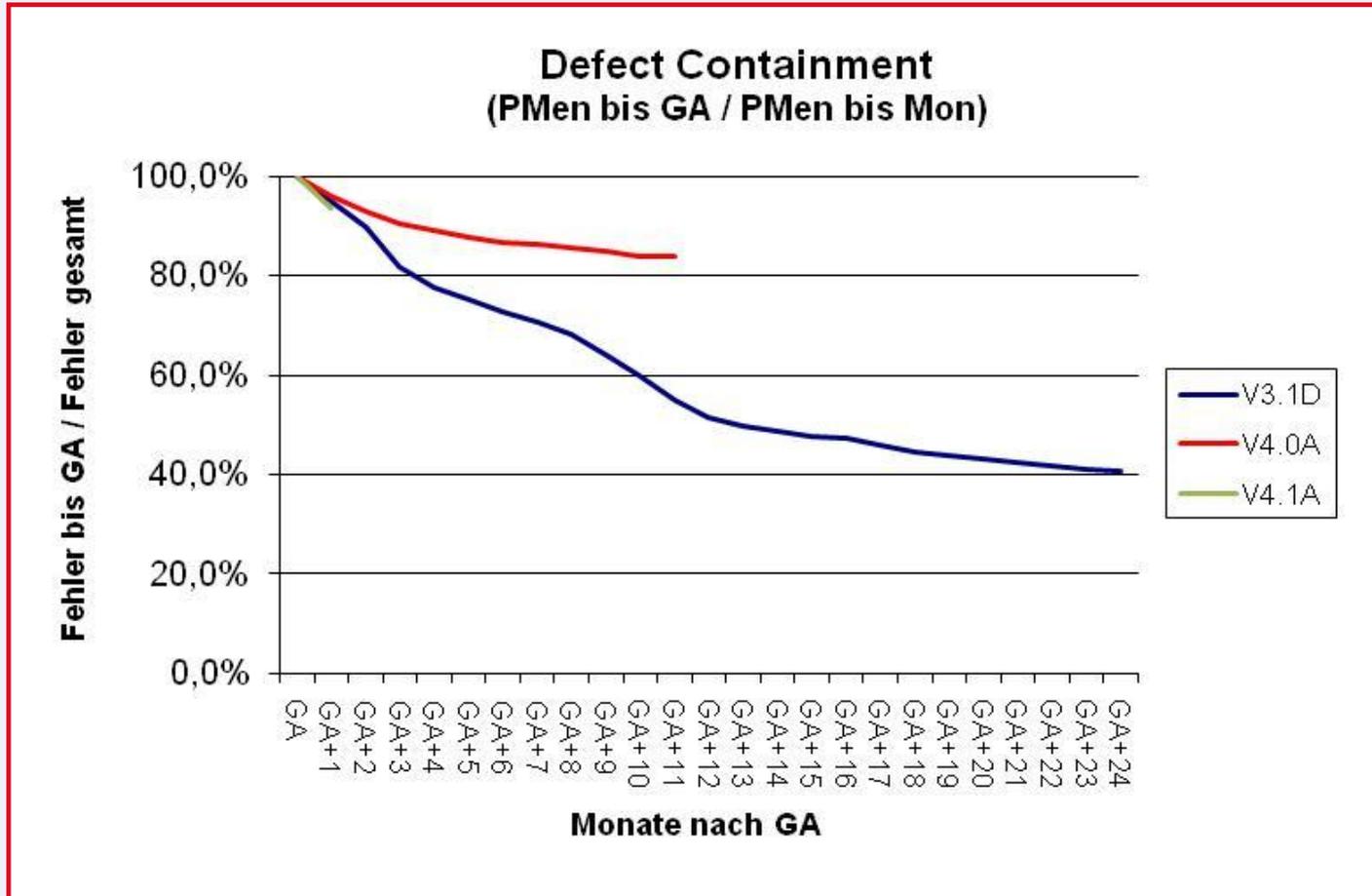


- Fehlerstrom
- Termintreue
- Anforderungstreue
- Anforderungslaufzeit
- Kosten pro Fehler

Testfortschritt

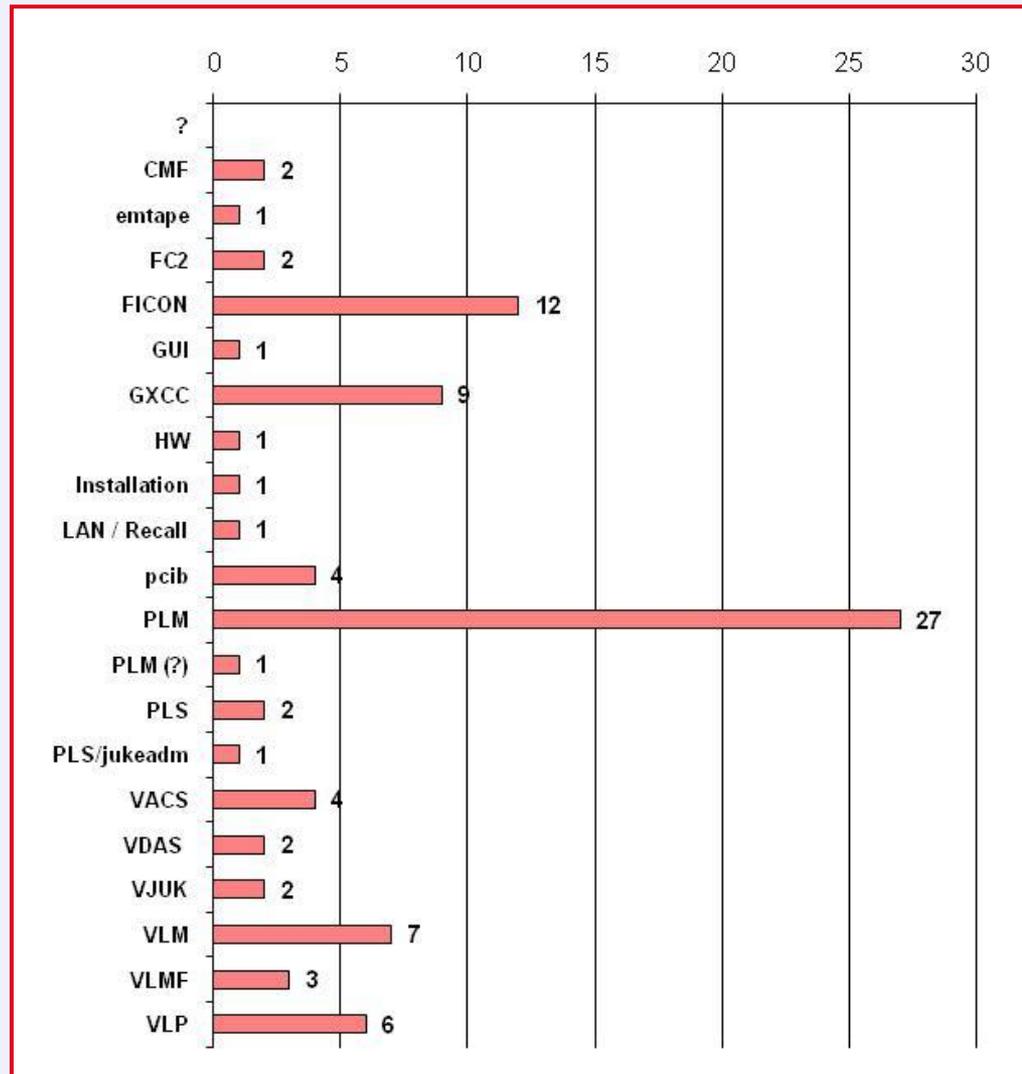


Delivered Defects

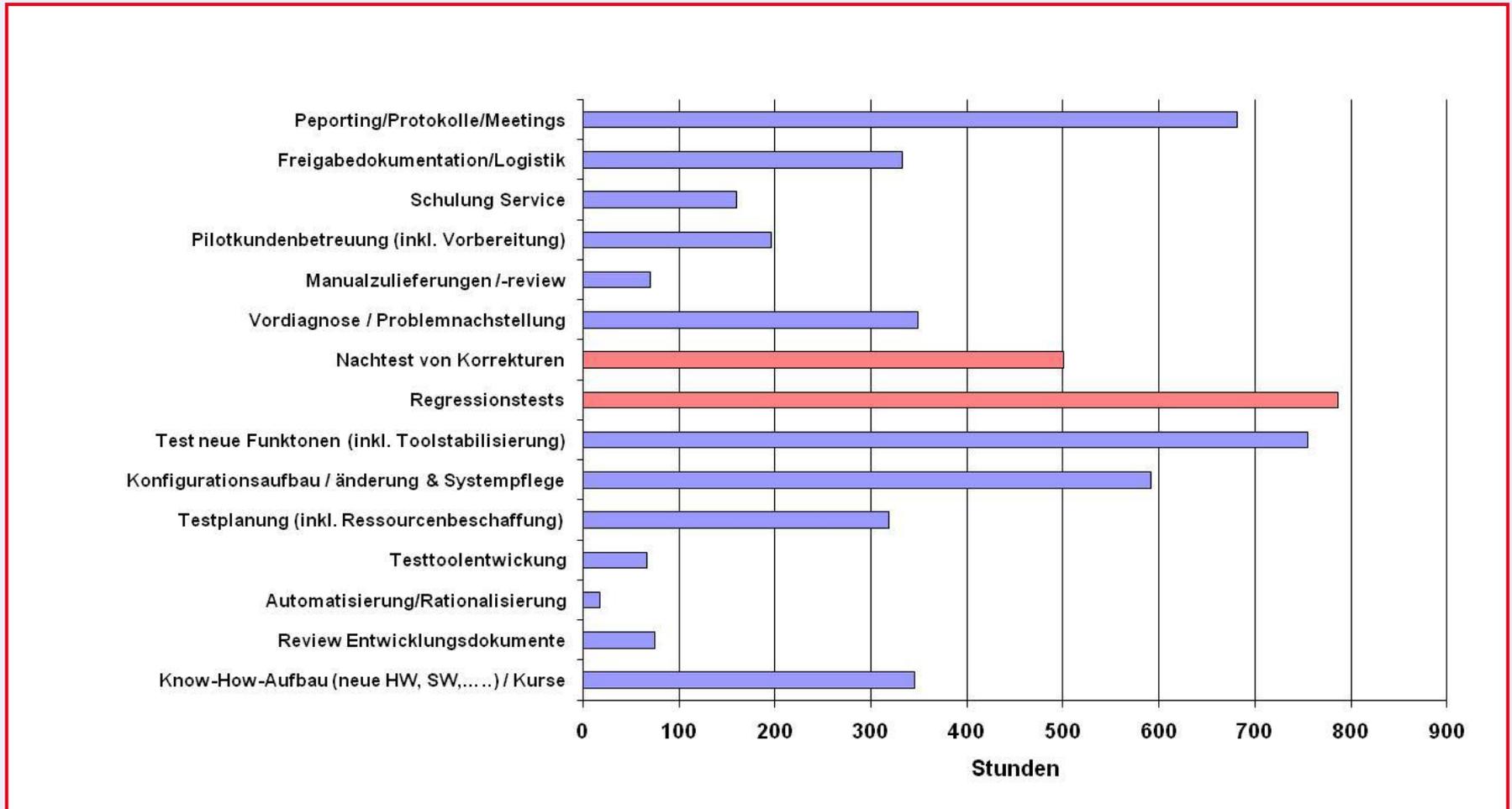


(GA: General Availability)

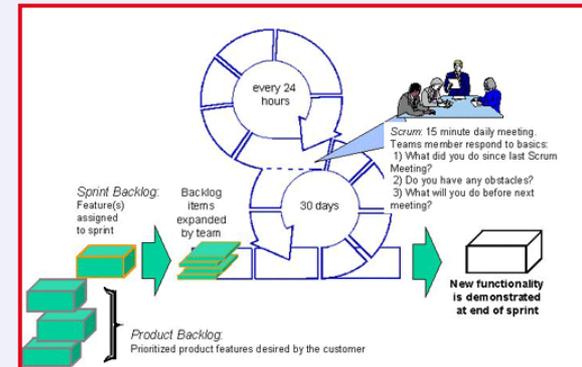
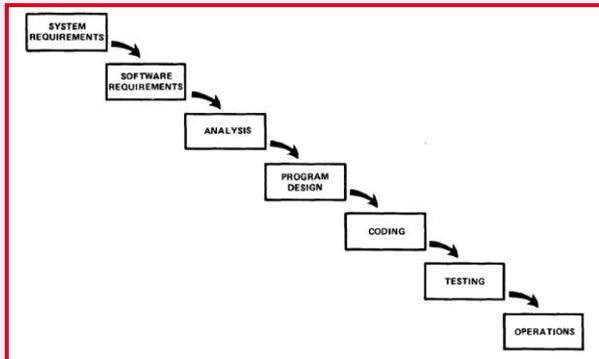
Escape Analysen



Kosten/Nutzen Testautomatisierung ?



Wir werden agil



- Termintreue
- Anforderungstreue
- Aufwände & Kosten
- Fehlerstrom
- Code-Qualität
- Feldfehler



Do's und Don'ts

No single metric can provide wisdom.

(Quelle: M. Bundschuh, „Aufwandsabschätzung von IT-Projekten“)



Weniger ist mehr.



Not everything that counts can be counted and not everything that can be counted counts.

(Albert Einstein)



Metriken dürfen nicht Personen bezogen sein.



Metriken in das Berichtswesen integrieren und Erfahrungsdatenbank aufbauen.





Vielen Dank für Ihre Aufmerksamkeit!

Hermann Will

hermann.will@qadvice.de

www.qadvice.de