

Gute Softwarequalität ist billiger

sieber & partners 

ICT-Beschaffungskonferenz
Bern, 24. August 2016

Fakten zur Firma sieber&partners

- Gründungsjahr: 2000
- Mitarbeitende: 30
- Geschäftsleitung:



Dr. Pascal Sieber



Marc André Hahn



Andreas Dietrich

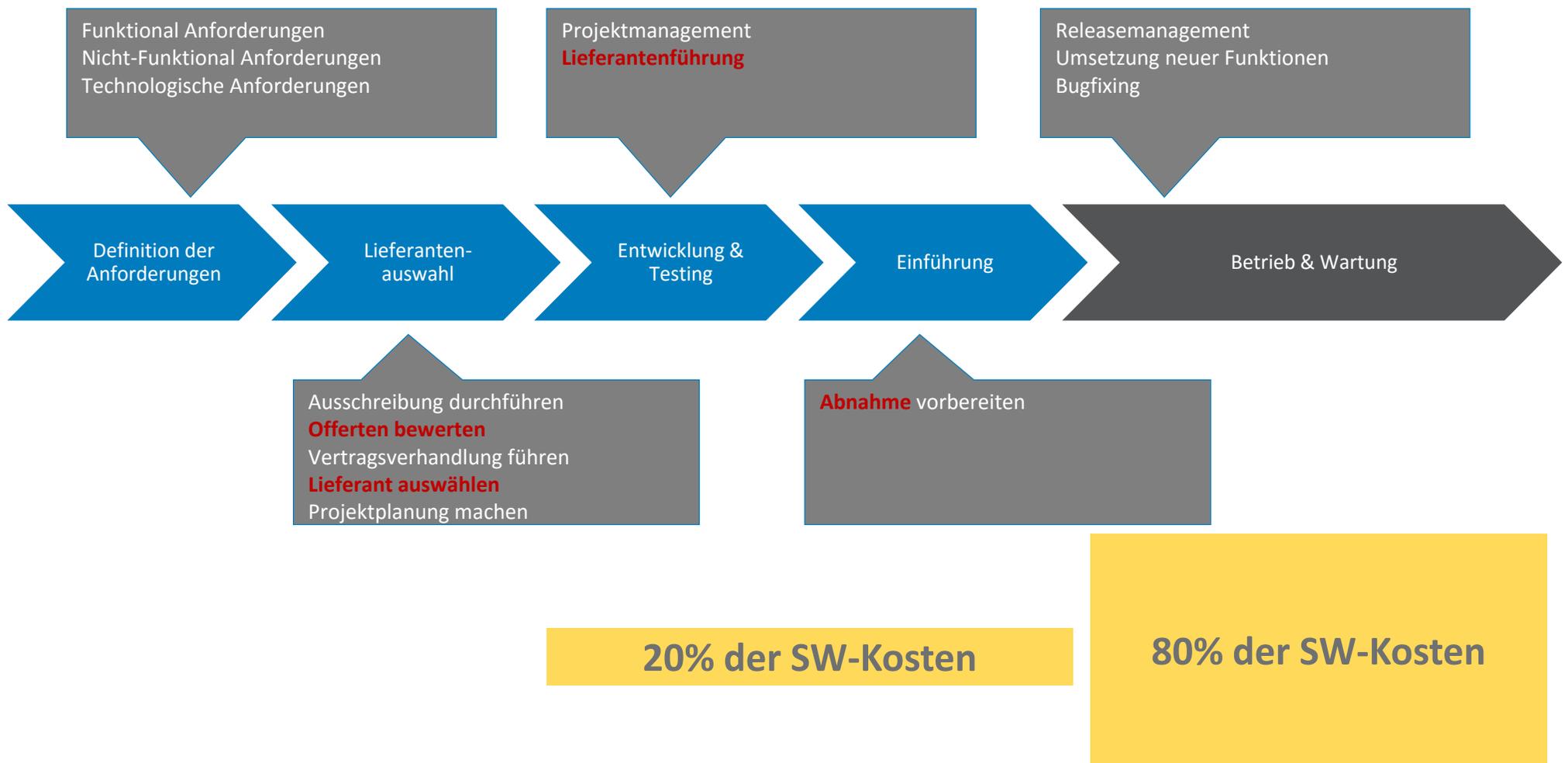
- Standorte: Zürich, Bern
- Tochterunternehmen:
 - Software Improvement Group AG
 - Innovationsplattform.ch GmbH
 - Schweizerisches Produktivitätsinstitut AG

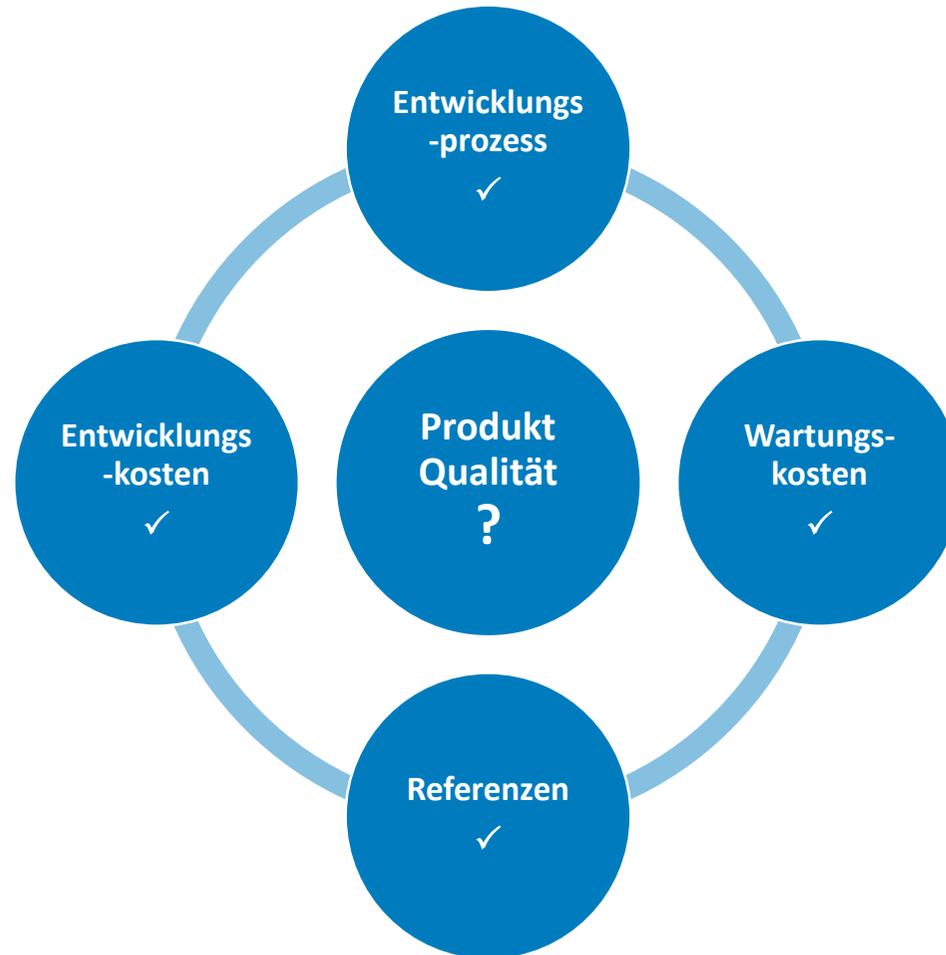
Inhalte

1. Das Beschaffungsdilemma
2. Softwarequalität und Auswirkung auf Kosten
3. Grösse – Qualität – Kosten
4. Beispiel für Umsetzung

Das Dilemma

Beschaffungsprozess





Entwicklungskosten

- Sind oft die wichtigste Entscheidungsgrundlage
- Machen 20% der Kosten im Lebenszyklus aus
- Bei öffentlichen Ausschreibungen sehr kleiner Spielraum

Entwicklungsprozess

- Standards wie CMMI werden beachtet
- Sind nur auf die Organisation bezogen, nicht auf das Produkt
- Können Hinweis auf schlechte Qualität, aber nicht auf gute Qualität geben
- Garantieren keine gute Qualität

Wartungskosten

- Machen den grössten Teil der Kosten eines System im gesamten Lebenszyklus aus (bis zu 80%)
- In der Regel sind nur Kosten und nicht Leistungen im Blick
- Sagen nichts über die Angemessenheit aus

Referenzen

- Geben Kundenzufriedenheit wieder
- Zeigen Kompetenzen und Erfahrungen auf
- Sagen nichts über Qualität und Angemessenheit des Preises aus

Qualität des Endprodukts

- Qualität des Sourcecodes wird häufig nicht geprüft!
 - Abnahmekriterium
 - Entwicklungskontrolle / Projektfortschritt
 - Risikomanagement

Folgen:

- Scheinbar tiefe Entwicklungskosten werden mit vielfach höheren Wartungskosten bezahlt
- Höhere Entwicklungskosten
- Höhrere Wartungskosten
- Nur Kosten werden gemessen – keine Leistungen
- Keine Lieferantensteuerung
- Vendor Lock-In

- Die Qualität ist wichtiger als der Tagessatz!

Softwarequalität

ISO 25010 Standard für Software Qualität



ISO 25010: Wartbarkeit

	Volume	Duplication	Unit size	Unit complexity	Unit interfacing	Module coupling	Component balance	Component independence
Analysability	X	X	X					X
Modifiability			X		X		X	
Testability	X				X			X
Modularity							X	X
Reusability				X		X		

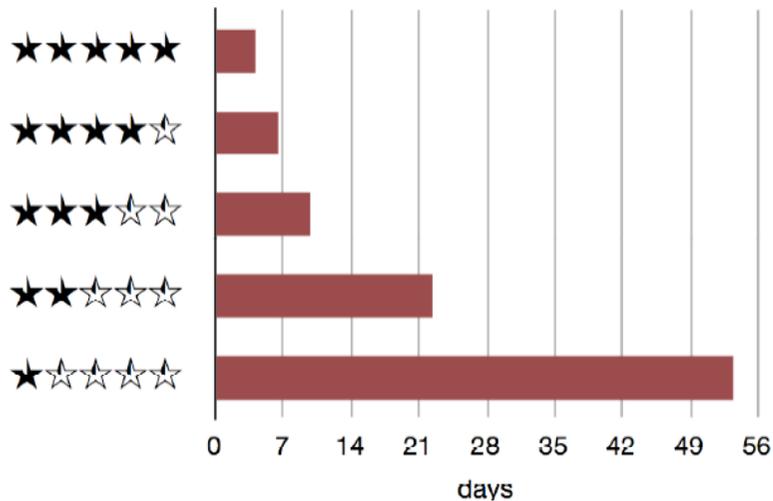
Technische Qualität von Software – Zertifizierung nach TÜVIT-Modell

Rating	Branchen Benchmark
★★★★★	Top 5 % der Branche
★★★★☆	Zwischen 65%-95% der Branche
★★★☆☆	Zwischen 35%-65% der Branche
★★☆☆☆	Zwischen 5%-35% der Branche
★☆☆☆☆	Schlechteste 5%

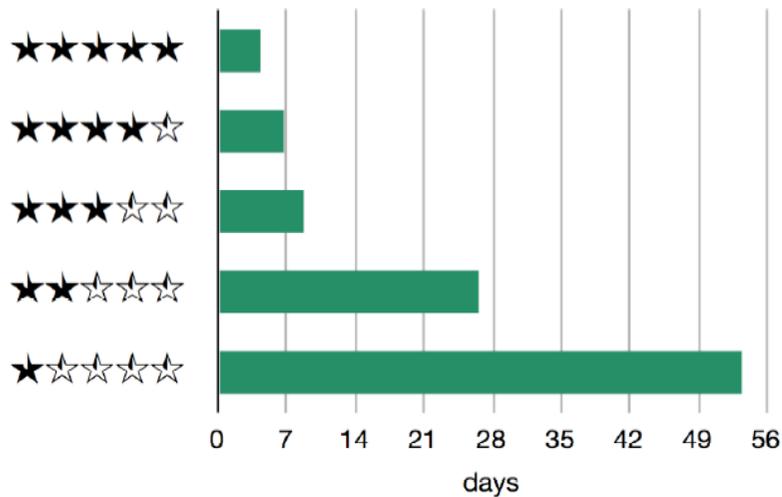
- Von★ (1-Stern) bis ★★★★★ (5-Sterne) Bewertung basierend auf Benchmarks
- Das Rating ist dynamisch
- Rating basiert auf den wissenschaftlichen Modellen der Softwareentwicklung
- Eine vollständige, automatische Source Code Analyse liefert die Kennzahlen
- Der TÜVIT zertifiziert anhand der vollständigen, automatischen Source Code Analyse

Technische Qualität und Lebenszykluskosten bei Software

Defect Resolution Time

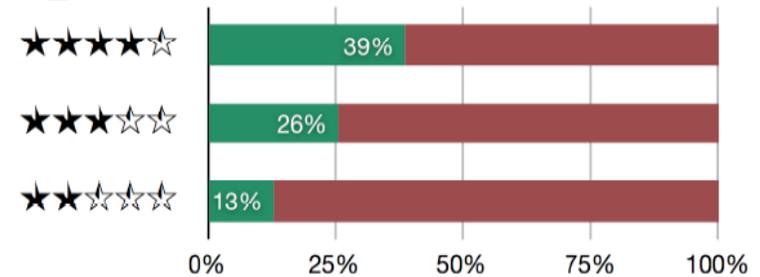


Enhancement Resolution Time

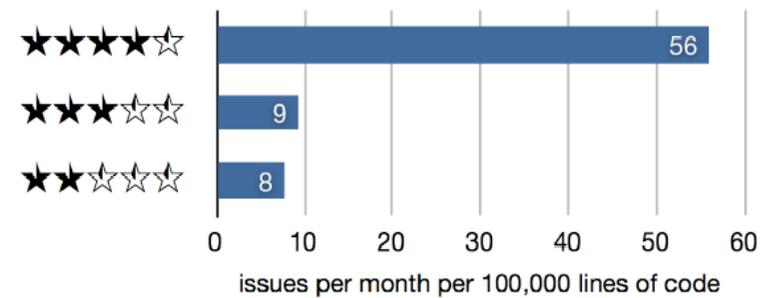


■ enhancements
■ defects

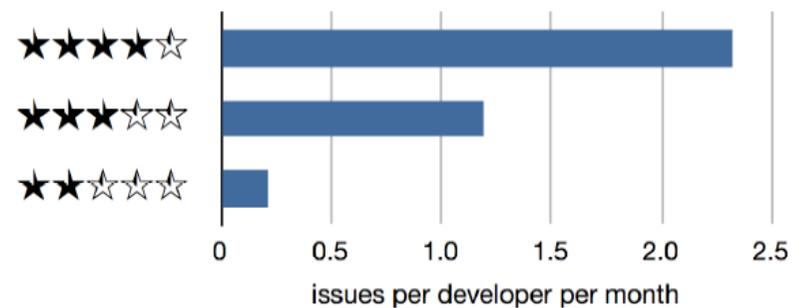
Efficiency



Throughput



Productivity



Auswirkungen von Qualität

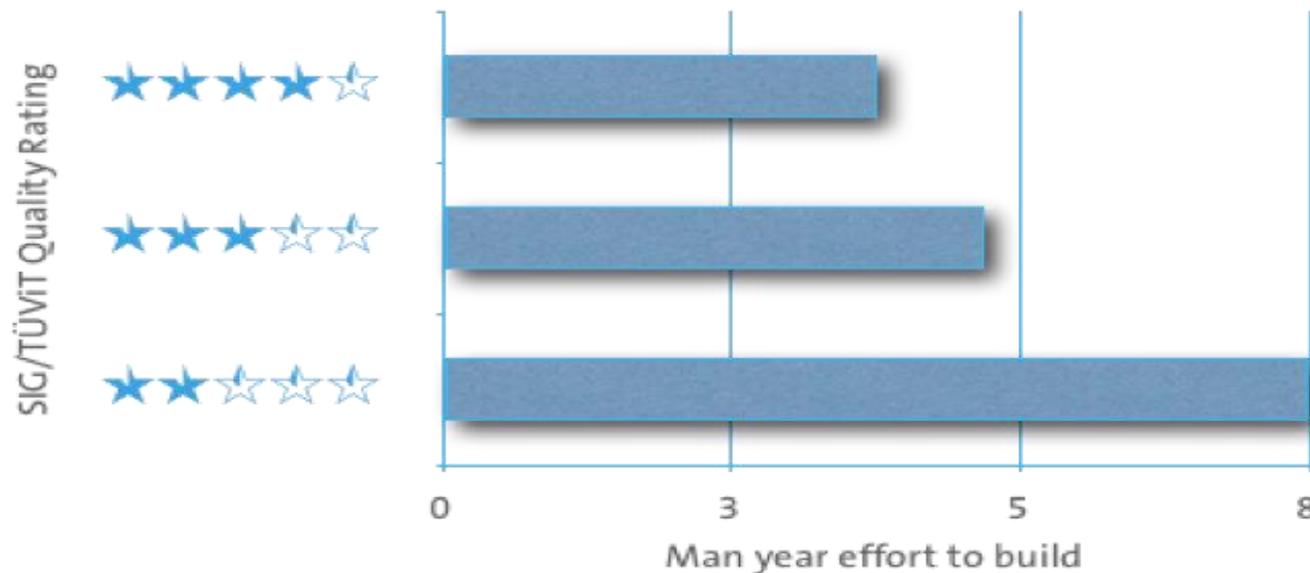
- Fehler werden im 4-Sterne-System **3-mal schneller** behoben als in einem 2-Sterne-System
- Durchsatz an Fehlerkorrekturen und Erweiterungen ist **7-mal höher**
- Produktivität steigt nahezu um den **Faktor 11**

- Durchsatz in diesem Kontext bedeutet Anzahl behobener Fehler und Erweiterungen pro Monat pro 100'000 Zeilen Code
- Produktivität bezieht sich auf die Anzahl Fehler und Erweiterungen (Issues) pro Entwickler pro Monat

Qualität – Grösse - Kosten

Qualitativ gute Software ist günstiger zu entwickeln

- Qualitativ gute Software hat ein besseres Softwaredesign, benötigt weniger doppelte Arbeiten, kann kostengünstiger getestet werden und ist leichter zu verändern, wenn sich die Anforderungen verändern.
- Qualitativ gute Software erfordert kleinere Entwicklerteams, kleine Entwicklerteams haben weniger Overhead

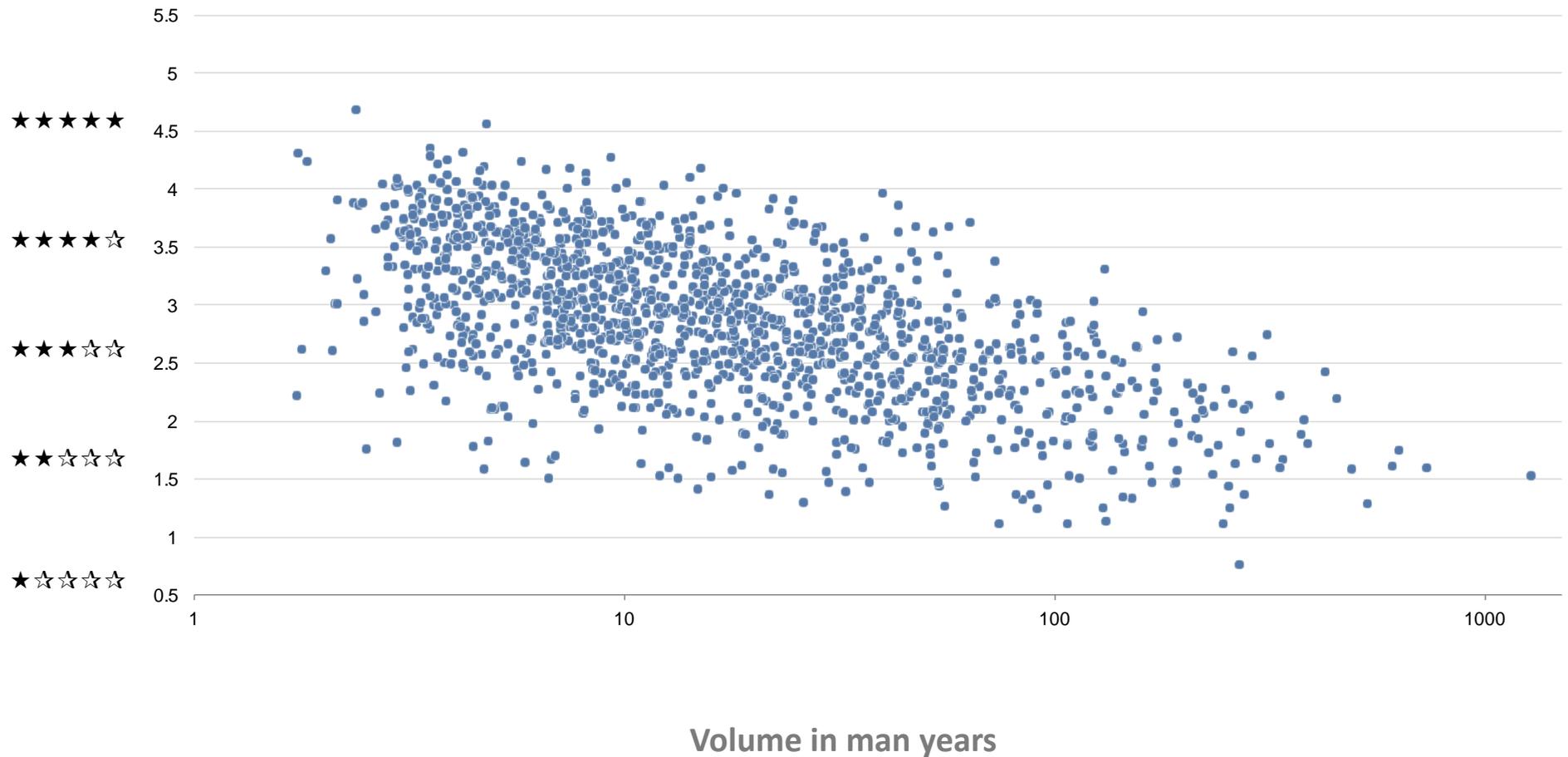


(Quelle: Benchmarkdaten für ein 1000 Function-Point System in Java oder .net)

Systemgrösse und –qualität stehen in Korrelation

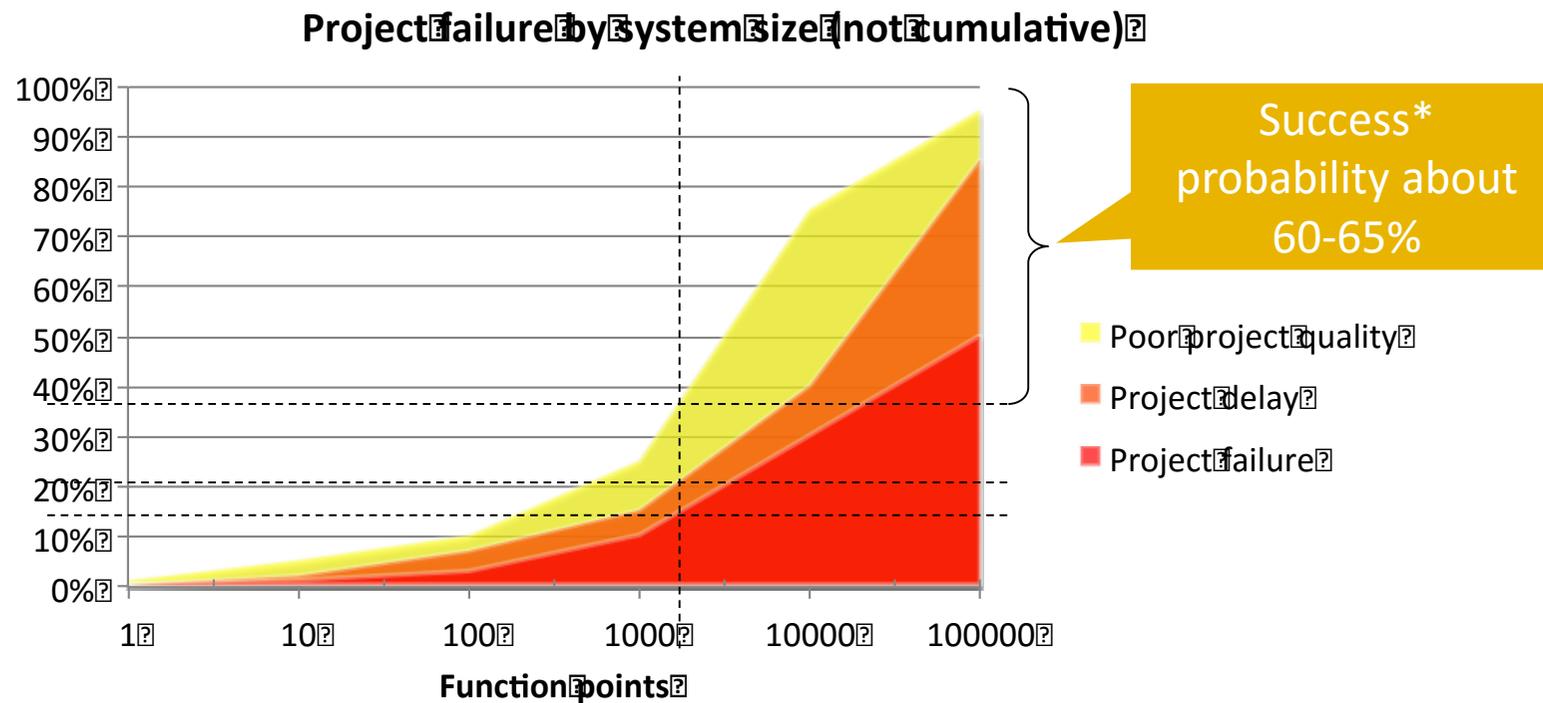
Gemäss SIG Benchmark, sollte ein 4-Sterne System nicht grösser als 20-Mannjahre sein

Maintainability
rating



Systemgrößen und Projektscheitern - Statistik

Systemgröße wird mit 2'300 Function Points geschätzt auf Basis der Gesamtgröße

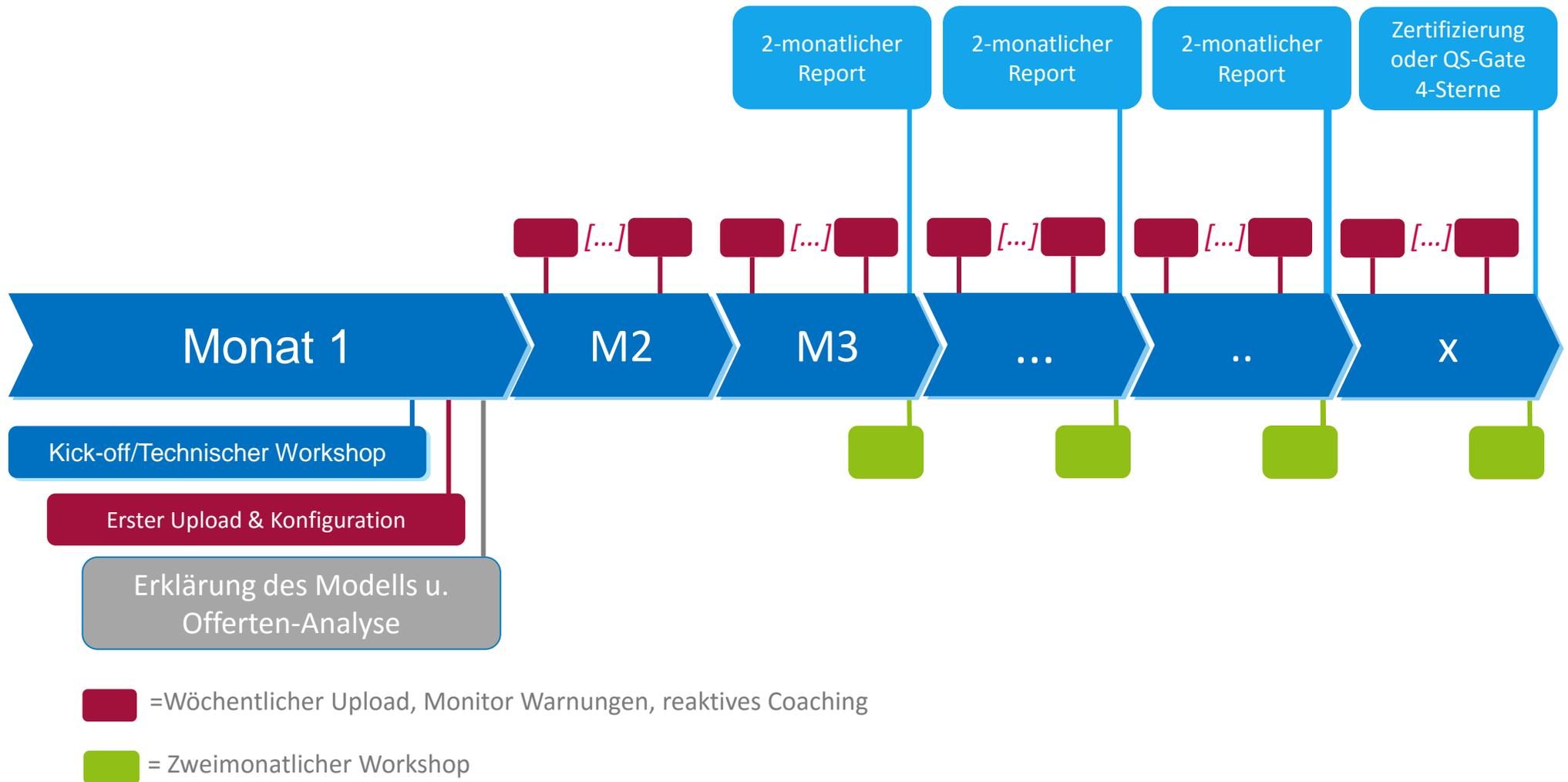


*Success=Good quality, on time, in budget

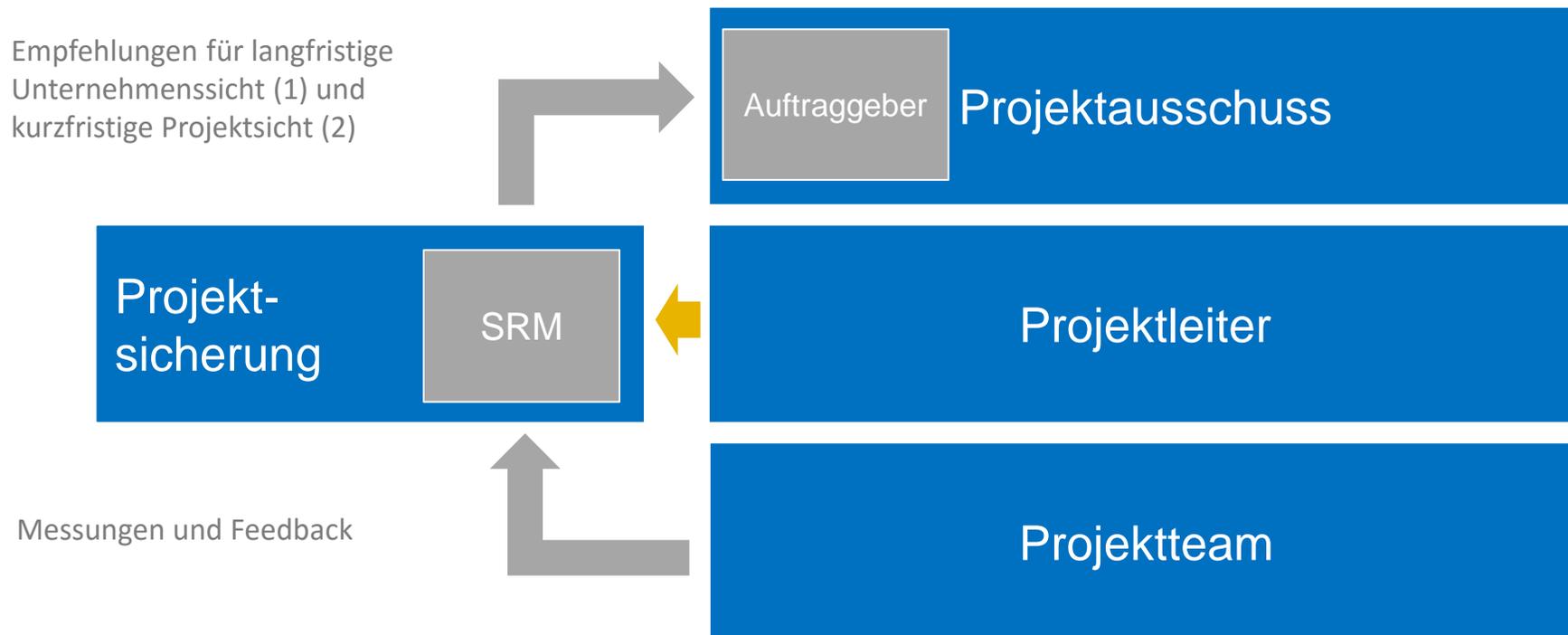
Source: The Economics of Software Quality by Capers Jones (2012)

Beispiel für ein Monitoring

Beispiel für einen Monitoring Prozess



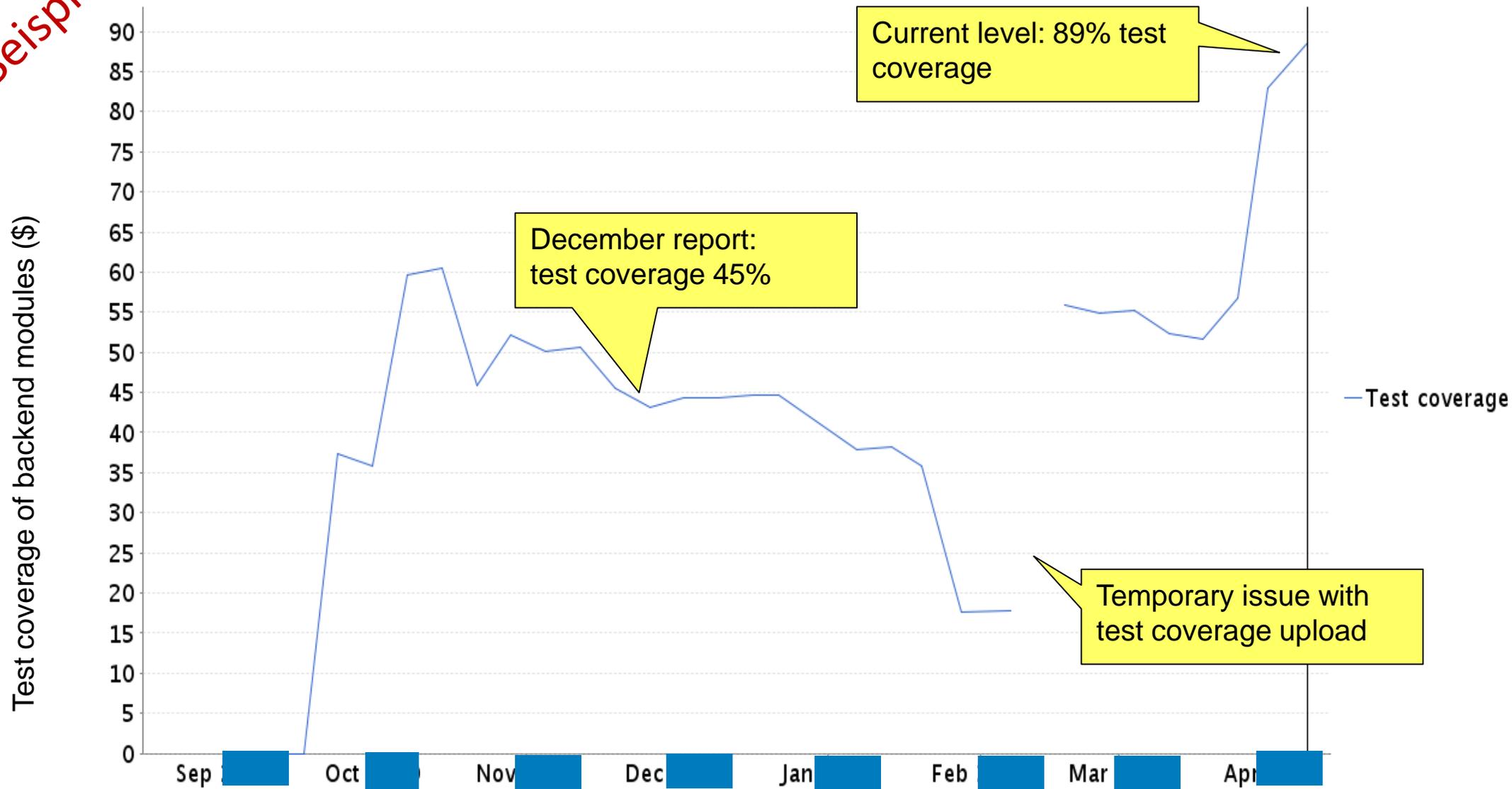
Wer ist der Kunden des Monitorings?



(Angelehnt an PRINCE 2 Projektorganisationsmodell)

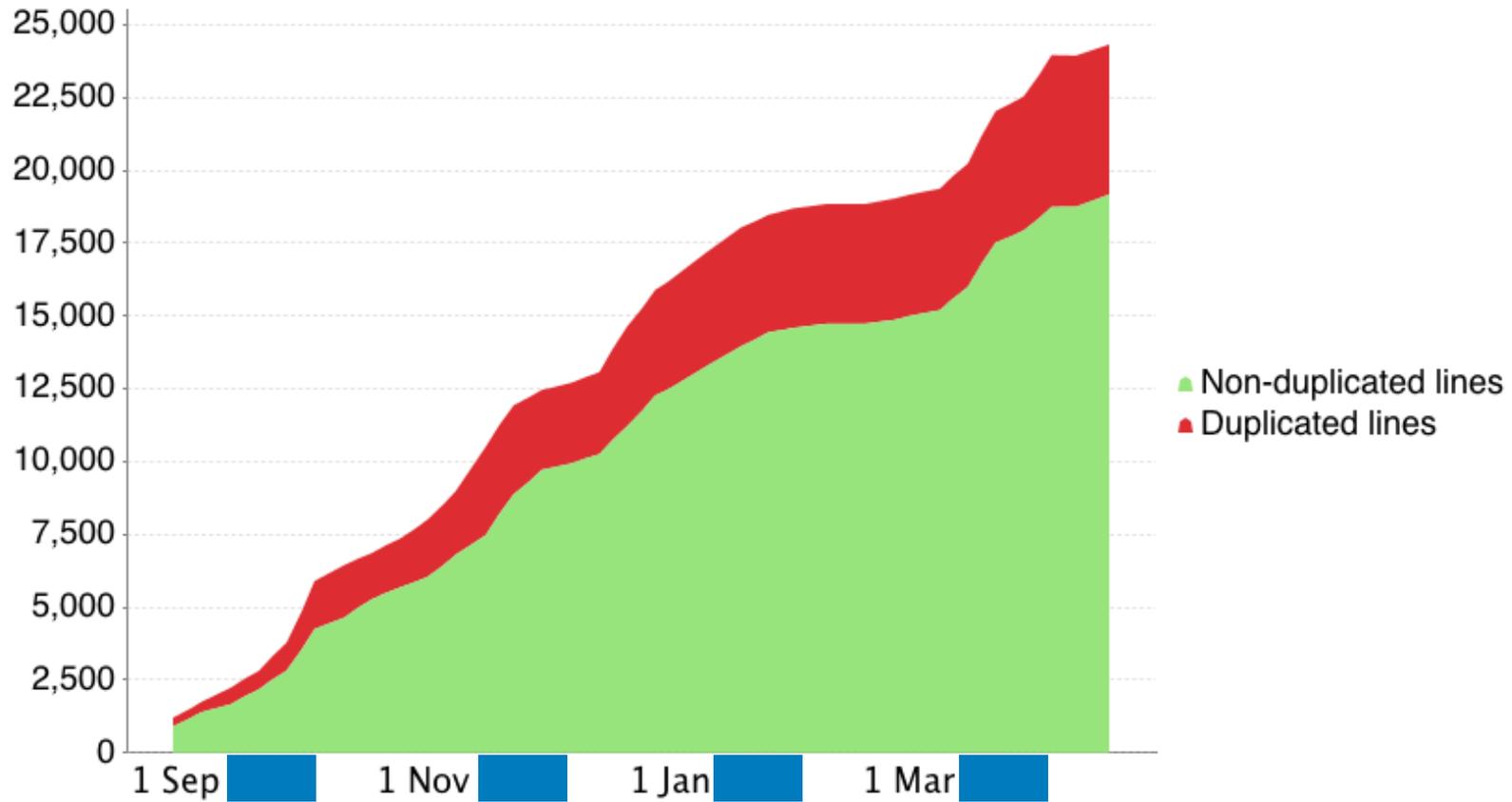
Test coverage of the back-end modules has improved from 45% to 89%

Beispiel



Duplication over time

Beispiel



Fazit

- Qualität kann gemessen werden
- Gute Qualität ist günstiger – die Qualität ist wichtiger als der Tagessatz
- Die Qualität hat massgeblich Einfluss auf die Entwicklungskosten
- Die Qualität hat massgeblich Einfluss auf die Wartungskosten
- Die Qualitätsmessung während des Projekts erlaubt es, die Risiken im Beschaffungsprozess zu reduzieren und Kosten zu senken

Beispiele von Anwendungen:

- Abnahmen in Abhängigkeit der Qualität der Lieferung (4-Sterne)
- Effektiv bezahlte Preise in Abhängigkeit der Qualität der Lieferung
- Boni in Abhängigkeit der Wartbarkeit
- Projektsteuerung in Abhängigkeit entdeckter Risiken und objektivem Projektfortschritt
- Benchmarks zu vergleichbaren Systemen
- u.v.m.

Vielen Dank für Ihre Aufmerksamkeit!

Marc André Hahn

marc.hahn@sieberpartners.com

078 686 85 16

sieber & partners 

Bern Zürich Triesen Gerra Verzasca

