Gerald Heller

IN ETAPPEN ZUM ZIEL: INKREMENTELLE ENTWICKLUNG MIT HP QUALITY CENTER

DIE NUTZUNG DES RELEASE- UND CYCLE-MANAGEMENTS

Agenda

- Release und Cycle im Projektkontext
- Aufbau eines Releasebaumes
- Nutzung im Anforderungsmodul
- Nutzung für Tests
- Die Kunst des effektiven Filterns
- Reporting
- Endlich: „Pinned Tests“ in QC 10

Begriff: Release

- Ein Software Release bezeichnet die zur Verfügungstellung einer bestimmten Konfiguration eines Software Produktes
- Die Erstellung eines Software Releases findet typischerweise innerhalb eines Projektes statt
- Die Zielgruppe kann intern oder extern sein

Releaseinhalt

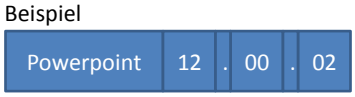
- Einem Software Release sind zugeordnet
  - Anforderungen
  - Spezifikationen
  - Lauffähige Software
  - Tests
  - Fehler
  - Dokumentation
  - …

Releasetaxonomie

- Ein Release wird näher festgelegt durch einen Releasebezeichner
- Elemente können sein
  - Namensbezeichner
  - Hauptversion
  - Unterversion
  - Patchversion
  - Servicepack
  - Kompilierversion
  - Betriebssystem
  - Hardwarearchitektur

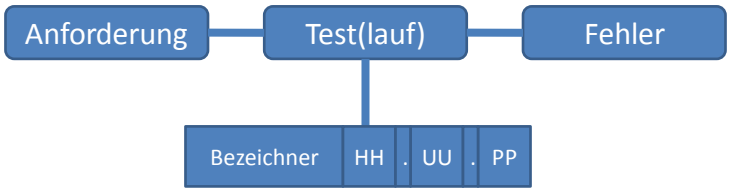
Beispiel:
Aufbau eines Releasebezeichners

Beispiel



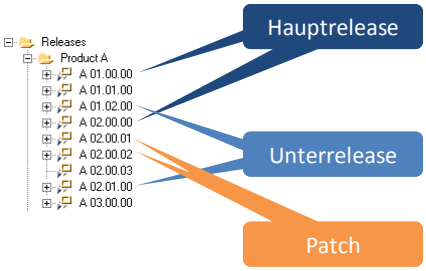
Entscheidend ist der Zweck des Ganzen!

Was ist das Interesse?

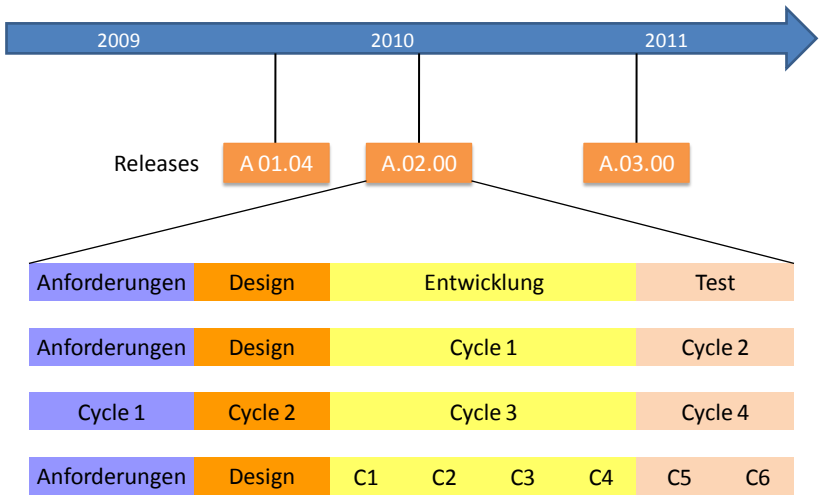


Aus Qualitätsperspektive ist es wichtig zu wissen, welche Qualitätsmaßnahmen für einen spezifischen Softwarestand vorgenommen wurden

Beispiel: Ein Releasebaum in QC



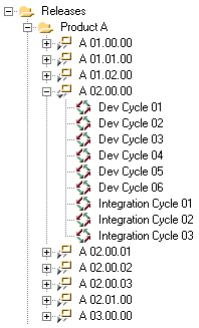
Projektlebenszyklen



Quality Center Cycle

- Ein Cycle in Quality Center bezeichnet einen Zeitraum mit definiertem Anfang und Ende
- Einem Cycle können Quality Center Artefakte zugeordnet werden
  - Requirement
  - Testset
  - Defect

Cycle Struktur

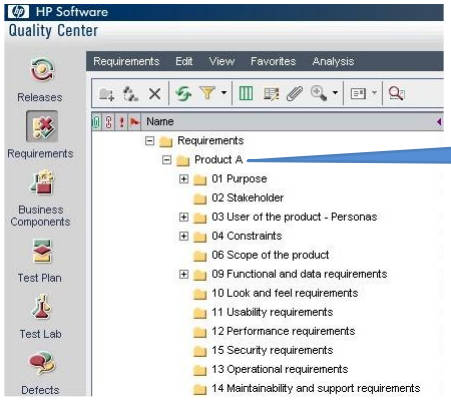


Je nach Einsatz können Cycles ausschließlich für Tests genutzt werden oder aber für alle Entwicklungsabschnitte

In diesem Beispiel wird das gesamte Produkt über mehrere Releases repräsentiert

Nicht jeder Cycle muss auch für QA relevant sein!
(z.B. Dev Cycle 01)

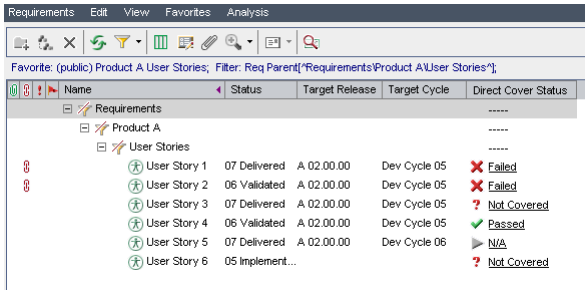
Anforderungsmodul



Produkt !
Nicht Projekt

Seit Version 9.2 bietet Quality Center eine vollständige Anforderungsmanagement Lösung

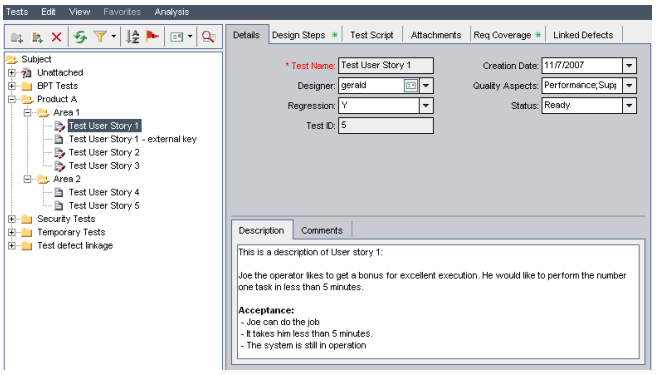
Anforderungen sortiert nach cycle



- Anforderungen werden gemäß ihrer Businessfunktionalität im Baum angeordnet.
- Filtern erfolgt gemäß Cycle Wert
- Agile Teams bevorzugen Ordnerstrukturen gemäß den Cycles

Testplanstruktur

Die Testplanstruktur orientiert sich an der Business- oder Architekturstruktur

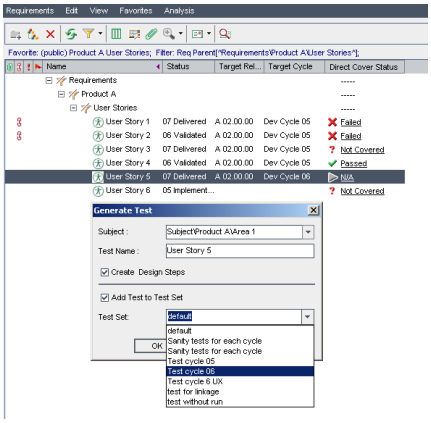


Tests und Cycles

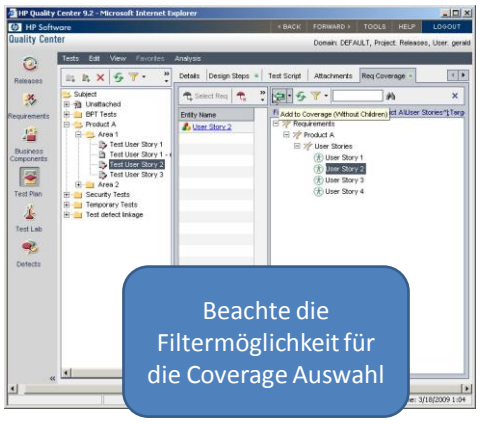
- Tests im Testplanmodul werden zu Anforderungen im Requirementsmodul verknüpft
- Damit erhalten sie automatisch eine Beziehung zu dem Cycle, welcher zur Anforderung verknüpft wurde

Verknüpfung von Anforderung zu Test

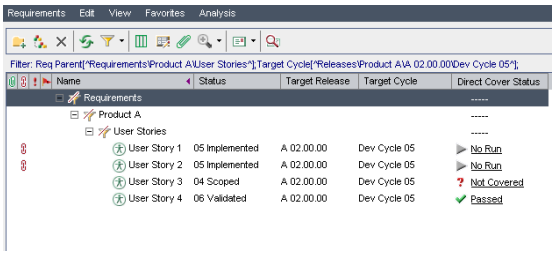
A) Erzeuge Test aus Anforderung



B) Erzeuge Test, verlinke zu Anforderung



Coverage Berichte



- Stand der Testentwicklung
- Stand des Testens

Nutzung von Cycles bei Fehlern

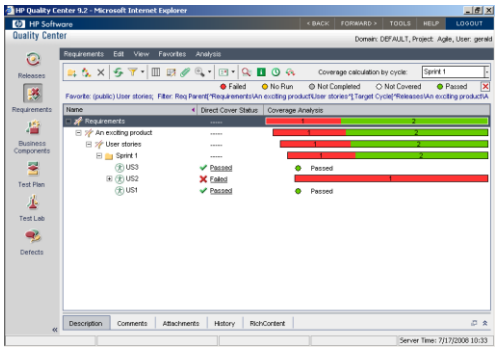
- Wenn während der Tests ein Fehler auftritt wird die Zuordnung zu dem Cycle automatisch eingefügt
- Zusätzlich empfiehlt es sich die Buildversion hinzuzufügen

Die Kunst des effektiven Filterns

- Um qualitative Aussagen über Cycle coverage zu bekommen, ist es notwendig die Filter synchronisiert zu nutzen

Cycles: Der erste Cycle

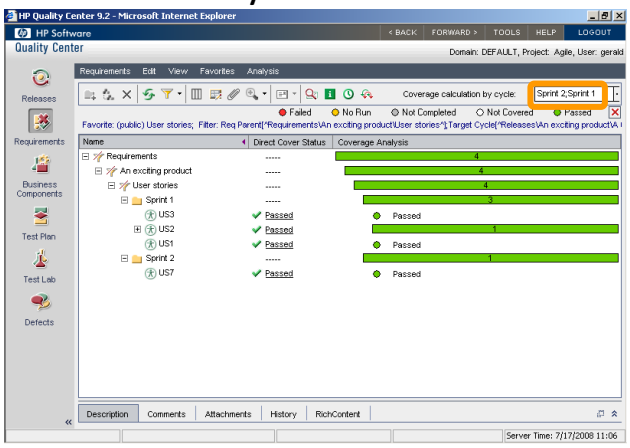
- Ergebnis nach dem ersten Cycle: Test für US2 nicht erfolgreich



Cycles: Der zweite cycle

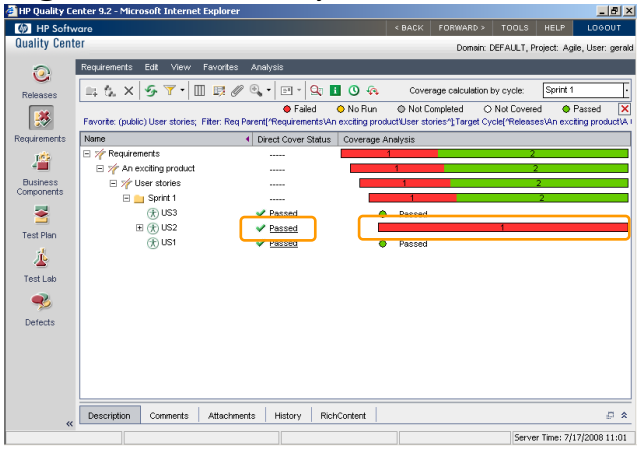
Ergebnis nach dem zweiten Cycle: Test für US2 ist nun o.k.

Die zwei Filter sind nun aufeinander abgestimmt:
Coverage
und
View Filter



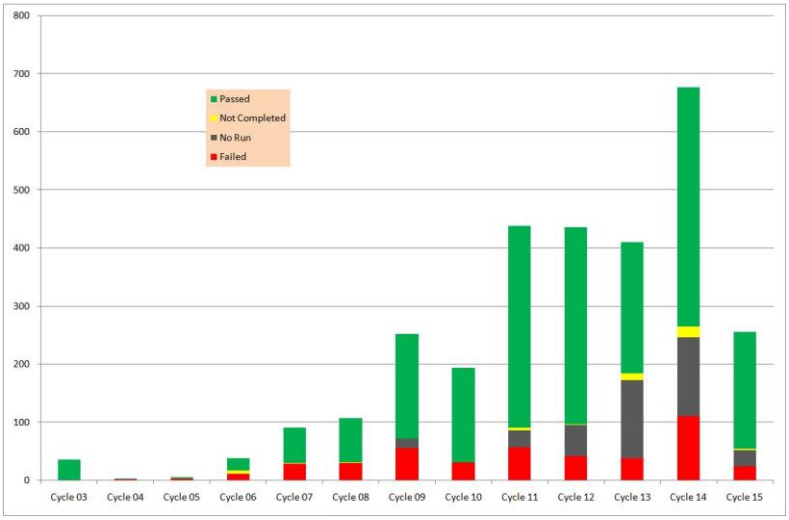
Cycles: Rückblick auf cycle 1

Ergebnis nach 2. cycle: Ansehen von cycle 1



2 Filter müssen koordiniert sein:
Coverage
und
View Filter

Cycle Reporting



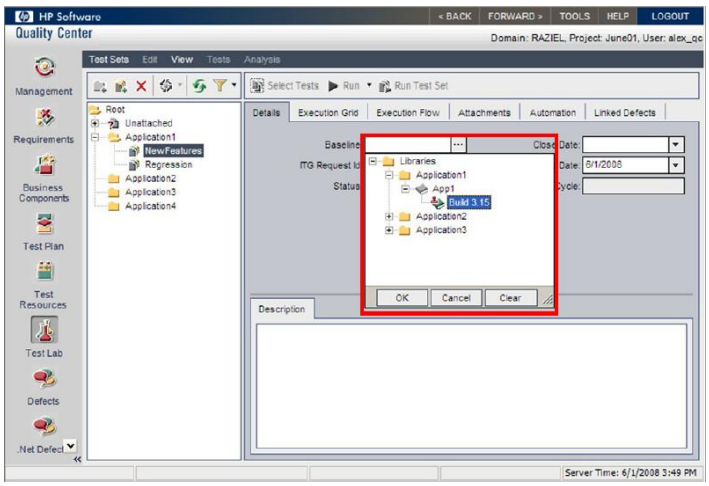
QC 10 und pinned Tests

- Ausgangsszenario
  - Fixieren des Entwicklungsstand X samt Tests Auslieferung an Kunde
  - Weiterentwicklung des Produktes auf Stand Y
  - Problem bei X gefunden, Patch wird gebaut
  - Erneutes Testen von Entwicklungsstand X
  - Tests sollen nur einen Effekt auf Stand X haben

Pinned tests

- Ein Test-Set kann zu einer Baseline gebunden werden
- Beim erneuten Ausführen des Tests wird dann exakt die Version genommen, die zum Zeitpunkt der Baselineerstellung aktuell war
- Testläufe von pinned tests beeinflussen den aktuellen Stand nicht

Pinned test GUI



Danke!

Fragen, Kommentare?

gerald.heller@swq4all.de