

## **Summary: Noser Way of Testing**

### **Warum unterstützt Testen das Management?**

- Steuerungsinstrument für nachhaltige Systemqualität
- Probleme und unnötige Kosten vermeiden
- Fehler früh und rechtzeitig erkennen
- TestmanagerIn = kompetente Ansprechperson für Management
- TestmanagerIn erarbeitet verlässliche, faktenbasierte Entscheidungsgrundlagen
- Schafft Transparenz
- Beschleunigt Entscheidungsprozess
- Sichert Entscheidungen ab

### **Der Nutzen vom Testen**

- Testbudget kann eingeplant werden
- Testen spart Geld
- Bessere Beschreibung der Anforderungen
- Bessere Anforderungen führt zu schnellerer Entwicklung
- Besser Unterscheidung
  - Fehler
  - Änderungen
  - Neuanforderungen

### **3 Erfolgsfaktoren fürs Testing**

- Tools
- Testmanager/in
- Prozess

### **Kriterien für gute Testing-Tools**

- Einfach zu bedienen
- Gut gepflegt
- Zugewiesene Verantwortlichkeit (für Pflege)
- Von allen Projektbeteiligten akzeptiert
- Pro Projekt individuell anpassbar
- Benötigen offene und einfache Schnittstellen
- Einfach dem Prozess anpassbar

## Kriterien für gute Test-Prozesse

- Einfach
- Verständlich
- Übersichtlich
- Muss gelebt und gepflegt werden
- Zugewiesene Verantwortlichkeit (Pflege, Schulung, Durchsetzung)
- Breit akzeptiert
- Nötige Freiräume geben
- Wichtige Details definieren
- Projekt unterstützen, nicht behindern

## Kriterien für gute Testmanager/in

- gute, solide Ausbildung in Anwendung und Entwicklung
- Erfahrung in Projektleitung
- Fundierte Testausbildung / Erfahrung
- Softskills
- Leidenschaft und Motivation

## “Agiles” Testen

- keine fixen Rollen verteilt
- Team organisieren sich selbst und handeln interdisziplinär
- ganzes Team für Testaspekte verantwortlich
- z.B. Entwickler auch verantwortlich das Klassen Unit-Tests haben
- Personen mit Testwissen sollen in SCRUM-Teams eingebettet werden

## Wo beginnt das Testen?

- bereits bei Abnahme der Anforderungen
- Testteam überprüft, ob gewünschter Endzustand klar, vollständig und widerspruchsfrei beschrieben, messbar und somit testbar ist
- Testteam stellt Testbarkeit der Anforderungen sicher

## Testing und die Wasserfall-Methode

- Testplanung
  - Viel Zeit
  - Testen als Phase
- Entwicklung vs. Testteam
  - Zwei Welten

- Testinfrastruktur
  - Viel Zeit für Vorbereitung
- Change Request
  - Grosser Impact
  - CR kann verloren gehen
- Testkonzept
  - Hat Raum und Zeit

## Testing und die RUP-Methode

- Rational Unified Process
- Testplanung
  - Eigene Disziplin
- Entwicklung vs. Testteam
  - Getrennt
- Testinfrastruktur
  - Kann vorbereitet werden
- Change Request
  - Iterativ
- Testkonzept
  - Vorgesehenes Artefakt

## Testing und die SCRUM-Methode

- Testplanung
  - Rollend
- Entwicklung vs. Testteam
  - Integriert
- Testinfrastruktur
  - Gleichzeitiges Testen
  - Infrastruktur bauen
- Change Request
  - Gut verwaltet (Backlog)
  - Gut kommuniziert
  - Testtasks laufend eingebracht
- Testkonzept
  - Kommt öfters zu kurz

## Strategie und Testkonzept

- Strategie bestimmt, wie stark Risiko durch Testen minimiert werden soll
- Folgendes ergibt sich aus Strategie
  - WAS soll getestet werden

- WIE muss getestet werden
- WIE intensiv muss getestet werden
- WANN sind Meilensteine angelegt
- WIE gross ist (Zeit-)Budget

## **Wozu dient die Teststrategie**

- Sich vorgängig Gedanken zum Testvorhaben zu machen und diese verbindlich festzuhalten

## **Wozu dient das Testkonzept**

- Ermöglicht es, Testen risikobasiert zu Priorisieren und Gruppieren
- Bildet Basis für erfolgreiches Testen
- Gibt Auskunft, wie umfassend und mit welchen Schwerpunkten getestet werden soll
- Beschreibt, wie sich mit zunehmendem Entwicklungsfortschritt der Testfokus verändert

## **Testkonzept/-strategie in 15min**

1. WAS: Welche Testaktivitäten sind notwendig?
  - Qualitätsanforderungen
  - Ableitung relevanter Testaufgaben
2. WIE: Wie intensiv muss getestet werden?
  - Kritikalität
  - Bewertung und Ableitung angemessener Testmethoden
3. WANN: Wie viel Zeit/Budget steht zur Verfügung?
  - Termin/Budget
  - Planung der Tests

## **Erste Schritte beim Erstellen von Testkonzept/-strategie**

- Akzeptierbares Risiko bestimmen
- Prioritäten festlegen
- Einzelne Testobjekte definieren / abgrenzen
- Notwendiges Testvorgehen vorgeben
- Pro Testobjekt erwartete Testtypen und -arten definieren
- Erforderliche Testtiefe festlegen
- Führungskonzepte etablieren
- Angepasste Testorganisation mit verbindlichen Verantwortlichkeiten festlegen

## **SCRUM und Testkonzept**

- Testkonzept kommt oft zu kurz
- Testkonzept trotzdem notwendig
- Testaspekte müssen vorgängig mit Product Owner geklärt werden
- Testmanagement als Ansprechperson für Product Owner und SCRUM-Master
- Vor Sprint überprüfen, ob an SCRUM angepasstes Testkonzept besteht

## **Experimentieren vs. Testen**

- Testen: Es existiert eine Erwartung, die belegt werden soll
- Experimentieren: Ergebnis offen oder wird nur vermutet

## **Wichtige Fragen beim Erstellen von Testkonzept**

- Soll zugunsten einer kurzen Testdurchlaufzeit auf tiefes Testen verzichtet werden?
- Wie relevant ist die Sicherheit, benötigt dieser Aspekt zusätzliche Testfälle?
- Wie wichtig sind Bedienbarkeit, Ergonomie sowie Aufbau und Farbgebung?
- Benötigt es zusätzliche “sehr kreative Spezialfälle”?
- Wie wichtig sind Performance, Geschwindigkeit, Wartezeiten, Verfügbarkeit, etc.?
- Was passiert im Fehlerfall, was ist der “WORST CASE”?
- Wie ist die Haftung und Garantie geregelt, was passiert im Fehlerfall?
- Wie lange “lebt” das Produkt?
- In welchem Umfeld wird das Produkt eingesetzt?
- Wie “komplex” ist das Produkt?
- Was kostet das Produkt?
- Gibt es Gesetze, Vorschriften, Normen?
- Welchen “Standard” haben vergleichbare Produkte?

## **Lösungsansatz für zunehmender Testaufwand pro Iteration**

- Testautomatisierung

## **Qualitätsmerkmale nach ISO 25000**

- Funktionale Eignung
- Zuverlässigkeit

- Benutzbarkeit
- Leistungseffizienz
- Wartbarkeit
- Übertragbarkeit
- Sicherheit
- Kompatibilität
- Eselsbrücke: Für seine Brüder würde Lukas über Zäune klettern.

### **Funktionale Eignung nach ISO 25000**

- Angemessenheit
- Richtigkeit
- Interoperabilität
- Ordnungsmässigkeit
- Ist die geforderte Funktionalität in der Software gegeben?

### **Zuverlässigkeit nach ISO 25000**

- Reife
- Fehlertoleranz
- Wiederherstellbarkeit
- Wie zuverlässig arbeitet Software?

### **Benutzbarkeit nach ISO 25000**

- Verständlichkeit
- Erlernbarkeit
- Bedienbarkeit
- Ist die Software einfach bedienbar?

### **Leistungseffizienz nach ISO 25000**

- Zeitverhalten
- Verbrauchsverhalten
- Wie effizient arbeitet die Software?

### **Wartbarkeit nach ISO 25000**

- Analysierbarkeit
- Modifizierbarkeit
- Stabilität

- Prüfbarkeit
- Anpassbarkeit
- Wie leicht lässt sich die Software modifizieren?

## **Übertragbarkeit nach ISO 25000**

- Anpassbarkeit
- Installierbarkeit
- Konformität
- Austauschbarkeit
- Wie leicht lässt sich die Software auf ein anderes System portieren?

## **Sicherheit nach ISO 25000**

- Zugriffssicherheit
- Datenverschlüsselung
- Wie sicher sind unsere Daten und Programme vor nicht autorisiertem Zugriff?

## **Kompatibilität nach ISO 25000**

- Austauschbarkeit
- Erweiterbarkeit
- Abwärtskompatibilität
- Wie kompatibel ist die Software beim Austausch und der Verarbeitung von Daten mit und von anderen Systemen

## **Wie weiss man was getestet werden soll?**

- Testobjekte bilden und nach verschiedenster Kriterien gruppieren

## **Typische Kriterien, um Testobjekte zu gruppieren**

- Funktionale Aspekte
- Architektur-Layer
- Aspekte der Zuständigkeit von Schlüsselpersonen
- Abhängigkeiten von Umsystemen
- Gruppen von unterschiedlichen nichtfunktionalen Tests
- Manuelle und automatische Tests
- Grenzwert- und Negativtests
- Quick- oder Smoketests

## Was macht gute Testobjekte aus?

- Untereinander ausgewogen
- Unabhängig voneinander testbar