

Zusammenfassung SWT

The Noser Way of Testing

pascal.kiser@stud.hslu.ch

19.06.2019

Inhaltsverzeichnis

1 Grundlagen Testing	2
1.1 Argumente fürs Testen	2
1.2 Die drei Erfolgsfaktoren	3
1.3 Generischer Vier-Schritte-Ablauf	3
2 Die Vier Phasen	4
2.1 Analysieren	4
2.2 Vorbereiten	4
2.3 Durchführen	5
2.3.1 Fehlerklassifizierung	5
2.3.2 Incident Management	5
2.4 Auswerten	5
3 Projektmethodik und Testen	7
4 Strategie und Konzept	8
4.1 Bastelbogen Teststrategie	8
4.2 Testziele	8
4.2.1 Qualitätsmerkmale nach ISO 25000	10
4.2.2 Gruppierung von Testobjekten	11
4.2.3 Testvielfalt	11
4.3 Testaufwand	11
4.4 Risiko Prioritäts Index (RPI)	12
5 Testautomatisierung	12
6 Testkonzept nach ISO 29119	14

1 Grundlagen Testing

Die wichtigsten Begriffe im Zusammenhang mit Testing sind die folgenden:

Testidee, Teststrategie & Testkonzept

- *Teststrategie*: Eine Dokumentation, die die generischen Anforderungen an das Testen in einem oder mehreren Projekten innerhalb einer Organisation beschreibt, einschließlich Details darüber, wie das Testen durchgeführt werden soll, und die an der Testrichtlinie ausgerichtet ist.
- *Testkonzept*: Die Dokumentation der Testziele sowie der Maßnahmen und Zeitplanung, um diese zu erreichen, zum Zweck der Koordination von Testaktivitäten.
- *Testidee*: `~_(_)_/~`

<http://glossar.german-testing-board.info/v3.21/>

Für Interessierte gibts dazu Lektüre im *Internet*, für alle anderen die Noser-Broschüre bzw. diese Zusammenfassung davon.

- [ISO-Norm 29119](#):
 1. **Concepts and definitions**
 2. **Test processes**
 3. **Test documentation**
 4. **Test techniques**
 5. **Keyword-driven testing**: build test automation based on keywords
- [ISTQB](#)

1.1 Argumente fürs Testen

Das Testen hat einige Vorteile. Ein paar davon sind:

- Unerwartete Katastrophen (Personenschaden, Betriebsausfall) verhindern
- Haftung / Erfüllung von Sorgfaltspflicht
- Kontrolle: *Vertrauen ist gut, Kontrolle ist besser*
- Garantie: Abnahmetests
- Qualität: Schnellere Entwicklung, bessere Dokumentation
- Liefert Kennzahlen (KPI) zum Führen
- Gezielt & geplant: *Agieren statt Reagieren*

1.2 Die drei Erfolgsfaktoren

Gemäss Broschüre gibt es genau drei Faktoren, die den Erfolg des Testens beeinflussen. Diese sind:

1. **Tools:**

- Einfach zu bedienen, gut gepflegt
- Verantwortlichkeiten klar
- Akzeptiert von allen Beteiligten
- Individuell pro Projekt
- Offene & einfache Schnittstellen

2. **Prozess:**

- Einfach, verständlich, übersichtlich
- gelebt & gepflegt
- Verantwortlichkeiten klar
- Akzeptiert von allen Beteiligten
- Nötige Freiräume
- Wichtige Details definieren
- Projekt unterstützen, nicht behindern

3. **Testmaster / Manager:**

- Gute Ausbildung in Fach und Entwicklung
- Erfahrung in Projektleitung
- Fundierte Testausbildung (ISTQB) & Erfahrung
- Softskills, Kommunikation und Menschenkenntnis
- Leidenschaft & Motivation

1.3 Generischer Vier-Schritte-Ablauf

Die portmann'sche Variante des *Generischen Vier-Schritte-Ablaufs* besteht aus folgenden Phasen:

1. Analysieren
2. Vorbereiten
3. Durchführen
4. Auswerten

2 Die Vier Phasen

Zur Repetition:

Die vier Phasen des Testings sind:

1. Analysieren
2. Vorbereiten
3. Durchführen
4. Auswerten

2.1 Analysieren

Beim Analysieren der Anforderungen müssen zwei wichtige Aussagen getroffen werden:

1. Kosten / Risiken: Wie viele Testszenarien (vorgegeben durch Testkonzept)
2. Abdeckung: Notwendige Testabdeckung (Grundlage für Anzahl Testfälle)

Je nach Bedarf kann die Abdeckung in drei Stufen gegliedert werden:

- **Schnell und grob:** Pro Anforderung (mind.) einen Testfall
- **Schnell und optimistisch:** Anzahl Anforderungen $\cdot 3$
- **Vorsichtig:** Anzahl Anforderungen $\cdot \frac{20}{RPI}$

Die Testabdeckung kann also **einfach** (*sind alle Anforderungen durch einen Testfall abgedeckt?*) oder **genau** (*ist die Abdeckung risikogerecht?*) beurteilt werden.

2.2 Vorbereiten

Testbereitschaft untersuchen und dokumentieren. Folgendes sollte nachher klar sein:

- Wie viele Testszenarien / Testfälle?
- Wie viel Vorbereitungszeit?
- Ist die Testumgebung bereit?

=> **Sind wir bereit zum Testen?**

Besonders bei agilen Projekten ist es wichtig, dass sämtliche Anforderungen der "Definition of Done" entsprechen. Test- und Review-Tasks müssen in der Planung berücksichtigt werden.

2.3 Durchführen

Abläufe & Prozesse einhalten, damit Kennzahlen, Information usw. korrekt erhoben werden.

Es soll vor allen der *Fortschritt* und der *Reifegrad* gemessen werden:

- **Fortschritt:** Wie viele Tests sind erfolgreich, wie viele Tests müssen noch durchgeführt werden? Wie lange dauert es noch?
- **Reifegrad:** Summe aller Fehler pro Fehlerklasse, Anzahl erfolgreich getesteter Anforderungen

2.3.1 Fehlerklassifizierung

Auswirkung / Schweregrad:

Grad	Beschreibung
1 Low	leichter Fehler, Funktion in Grunde gewährleistet
2 Medium	Störend, Systemfunktion nicht beeinträchtigt
3 High	Schwerer Fehler, Funktion nicht gewährleistet
4 Urgent	Fataler Fehler, Auswirkung auf ganzes System

Beobachtungsgüte / Reproduzierbarkeit:

Grad	Beschreibung
A	Eindeutig feststellbar, belegbar & reproduzierbar
B	Wiederholt auftretend
C	Nicht reproduzierbar

2.3.2 Incident Management

Die Abbildung "Incident Management" beschreibt das Incident Management mit den verschiedenen Zuständen eines Incidents.

2.4 Auswerten

Vorhandene Testresultate in *management-tauglichen* Report zusammenfassen, grafisch darstellen.

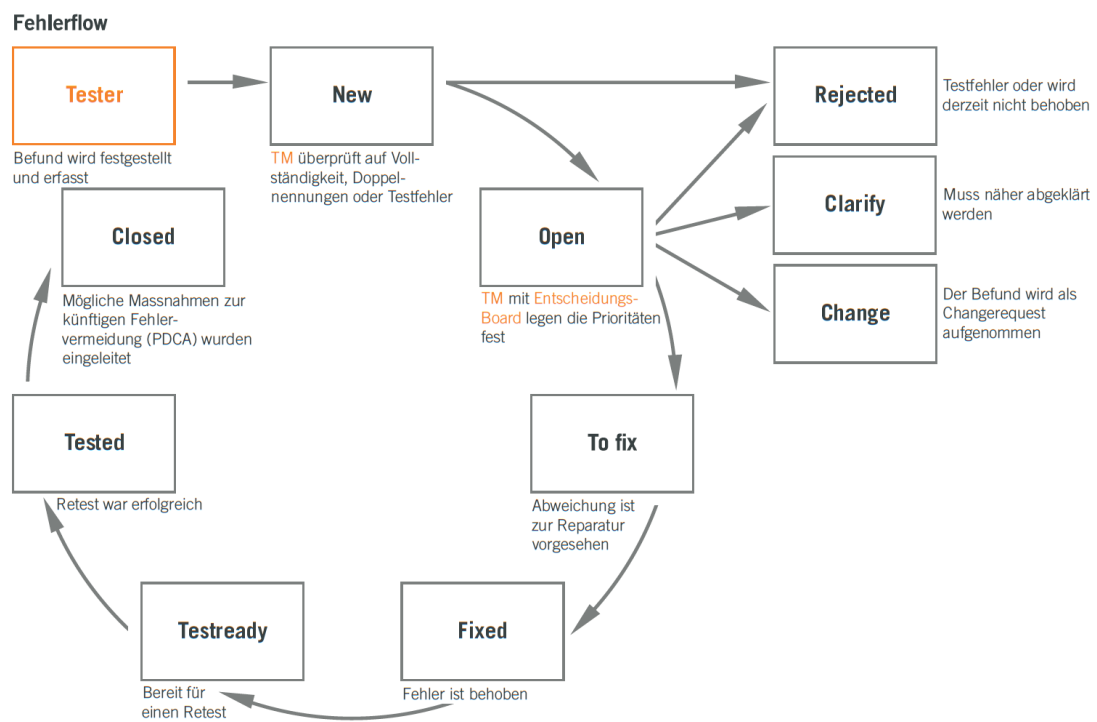


Abbildung 1: Incident Management

Dieser *Testreport* dient als Entscheidungsgrundlage: **GO** oder **NO GO**.

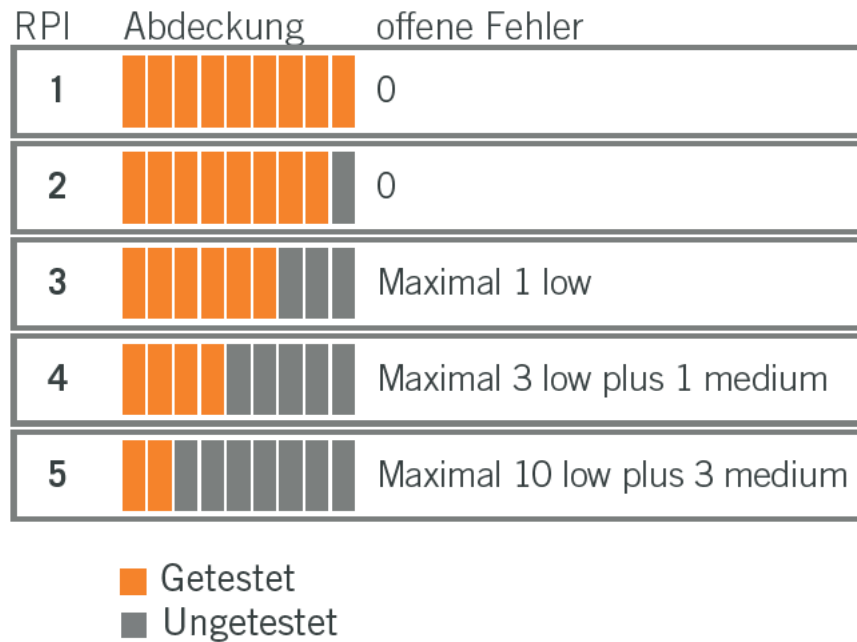


Abbildung 2: Wagen wir es?

3 Projektmethodik und Testen

Die Wahl der Projektmethodik hat Einfluss auf den Stellenwert und Zeitpunkt vom Testen innerhalb des Projekts:

1. Wasserfall:

- Viel Zeit fürs Testen & Vorbereitung, eigene Phase
- Stark von der Entwicklung getrennt

2. RUP:

- Viel Zeit fürs Testen
- Getrennt von der Entwicklung

3. SCRUM:

- Kontinuierliches Testen, kommt oft zu kurz
- Integriert in die Entwicklung

4 Strategie und Konzept

- Die *Strategie* bestimmt, wie weit das Risiko durch Testen minimiert werden soll.
- Das *Testkonzept* ermöglicht das Priorisieren und Gruppieren von Tests => Basis für erfolgreiches Testen.

4.1 Bastelbogen Teststrategie

Es folgt eine Anleitung zum Erstellen einer Noser-konformen Teststrategie in 15 Minuten. Man stelle sich zuerst folgende Fragen:

1. **WAS?** => Qualitätsanforderungen
2. **WIE?** => Kritikalität, wie intensiv
3. **WANN?** => Termin, Budget

Man befolge anschliessend folgende allgemeine Schritte:

- Akzeptierbares Risiko bestimmen
- Prioritäten festlegen
- Testobjekte definieren und abgrenzen
- Testvorgehen definieren
- Testarten / -typen pro Testobjekte definieren
- Erforderliche Testtiefe festlegen
- Führungskonzepte etablieren
- Testorganisation mit Verantwortlichkeiten

4.2 Testziele

Testkonzept:

- Testdurchlaufzeit vs Tiefe
- Relevanz von Sicherheit, Performance
- Wichtigkeit von Ergonomie & Bedienbarkeit
- Worst Case
- Gesetzliche Vorschriften, Normen
- Lebensdauer des Produkts

Was? Welche Test- aktivitäten sind notwendig?	Wie? Wie intensiv muss getestet werden?	Wann? Wie viel Zeit/Budget steht zur Verfügung?
Qualitätsanforderungen Qualitätsmerkmale/ Qualitätsziele, welche messbar sein sollen	Kritikalität Projekt, Teilprojekt, Testobjekt, Risiko- analyse	Termin/Budget Dringlichkeit, Gesamt- budget, Ressourcen
Ableitung relevanter Testaufgaben	Bewertung und Ableitung angemessener Testmethoden	Planung der Tests

Abbildung 3: Testkonzept in 15 Minuten

4.2.1 Qualitätsmerkmale nach ISO 25000

- **Funktionale Eignung**
 - Angemessenheit
 - Richtigkeit
- **Zuverlässigkeit**
 - Reife
 - Fehlertoleranz
 - Wiederherstellbarkeit
- **Benutzbarkeit**
 - Verständlichkeit
 - Erlernbarkeit
 - Bedienbarkeit
- **Leistungseffizienz**
 - Zeitverhalten
 - Verbrauchsverhalten
- **Wartbarkeit**
 - Analysierbarkeit
 - Modifizierbarkeit
 - Stabilität
 - Prüfbarkeit
 - Anpassbarkeit
- **Übertragbarkeit**
 - Anpassbarkeit
 - Installierbarkeit
 - Konformität
 - Austauschbarkeit
- **Sicherheit**
 - Zugriffssicherheit
 - Datenverschlüsselung
- **Kompatibilität**
 - Austauschbarkeit
 - Erweiterbarkeit
 - Abwärtskompatibilität

4.2.2 Gruppierung von Testobjekten

Testobjekte können auf verschiedene Arten gruppiert werden:

- Funktionale Aspekte
- Architektur-Layer
- Aspekte der Zuständigkeiten
- Abhängigkeiten von Umgebungen
- Verschiedene nichtfunktionale Tests
- Manuelle und automatische Tests
- Grenzwert- und Negativtests
- Quick- oder Smoketests

Testfälle sollten *ausgewogen* sein und *unabhängig* voneinander ausführbar.

4.2.3 Testvielfalt

Folgende *Testaspekte* müssen berücksichtigt werden und in die *Testplanung* einfließen:

- **Formale Eintrittskriterien:** Voraussetzungen zum Starten der Tests (Vorbereitungen, Daten vorhanden)
- **Vitalität:** Einfache, schnelle Tests zum Prüfen ob das System “lebt”, grundsätzlich prüfbar ist
- **Neue Funktionen:** Neue Funktionen / Technologien zeitnah testen
- **Gefixte Bugs:** Schnelle Rückmeldung bei gefixten Bugs
- **Retest / Regression:** Regelmässige Regressionstests
- **“Nur leicht modifiziert”:** Sicherstellen, dass bereits getestete Funktionen stabil bleiben
- **Monkey-, Negativ-, Grenzwertest:** Erweiterung der Schönwetter-Testfälle
- **Unvorhergesehenes:** Reserven einplanen

4.3 Testaufwand

Mit jeder Iteration steigt der Testaufwand:

- Mit jeder Iteration kommen neue Features (+Tests) dazu
- Zusätzliche Tests für bestehende Features
- Wiederholung bestehender Tests
- Automatisierung von Test

	1 Iteration	2 Iteration	3 Iteration	4 Iteration	5 Iteration
Automatisiertes Testen					
					Test A
				Test A	Test B
			Test A	Test B	Test C
Zusätzlich Negativ- und Grenzwerttest		Test A	Test B	Test C	Test D
Neue Features werden getestet (Schönwetter)	Test A	Test B	Test C	Test D	Test E

Ungetestet
 Retest

Neue Tests, neue Funktionen
 Getestet, automatisiertes Testing

Abbildung 4: Testaufwand

4.4 Risiko Prioritäts Index (RPI)

Der Testaufwand wächst *exponentiell*, deshalb ist es wichtig, die Testfälle richtig zu priorisieren. Bei der systematischen Priorisierung mit RPI werden Anforderungen nach drei Kriterien bewertet:

- **Business Relevanz:** Auswirkung im Fehlerfall (1-3)
- **Auffindbarkeit:** Wie offensichtlich und schnell (1-3)
- **Komplexität:** Umsetzung & Realisierung (1-3)

Das Produkt dieser drei Kriterien ergeben den (*RisikoPrioritätsIndex™*). Dieses ausgeklügelte Führungsinstrument bildet die Grundlage für die Bestimmung der *Testtiefe*, des *Testumfangs* und *Testreihenfolge*.

5 Testautomatisierung

Automatisierung ist ein breites Feld. Im Minimum sollte folgendes im Testkonzept geklärt sein:

- Ziel der Automatisierung
- Relevanz der Zeitersparnis / Testdurchlaufzeit
- Pflege Tools, Ergebnisse, Daten
- Return of Investment

Produkt	Priorität	RPI	Testumfang, Testtiefe
27	very high	1	alles in jeder Iteration, sehr gründlich und sehr tief
18			alles pro zwei Iterationen, gründlich und tief
12	high	2	alles pro zwei bis drei Iterationen, wechselnd gründlich/tief
9			wesentliche Abdeckung, in mehreren Iterationen, gründlich und tief
8			wesentliche Abdeckung über mehrere Iterationen, Wechsel Tiefe/Breite
6	medium	3	wesentliche Abdeckung verteilt auf mehrere Iterationen
4			Abdeckung wesentlicher Pfade, alternative Pfade wechselnd
3			Abdeckung wesentlicher Pfade, alternative Pfade stichprobenartig, Wechsel
2	low	4	nur Abdeckung, wesentliche Pfade
1	very low	5	stichprobenartige Abdeckung wesentlicher Pfade nicht testen

Abbildung 5: RPI

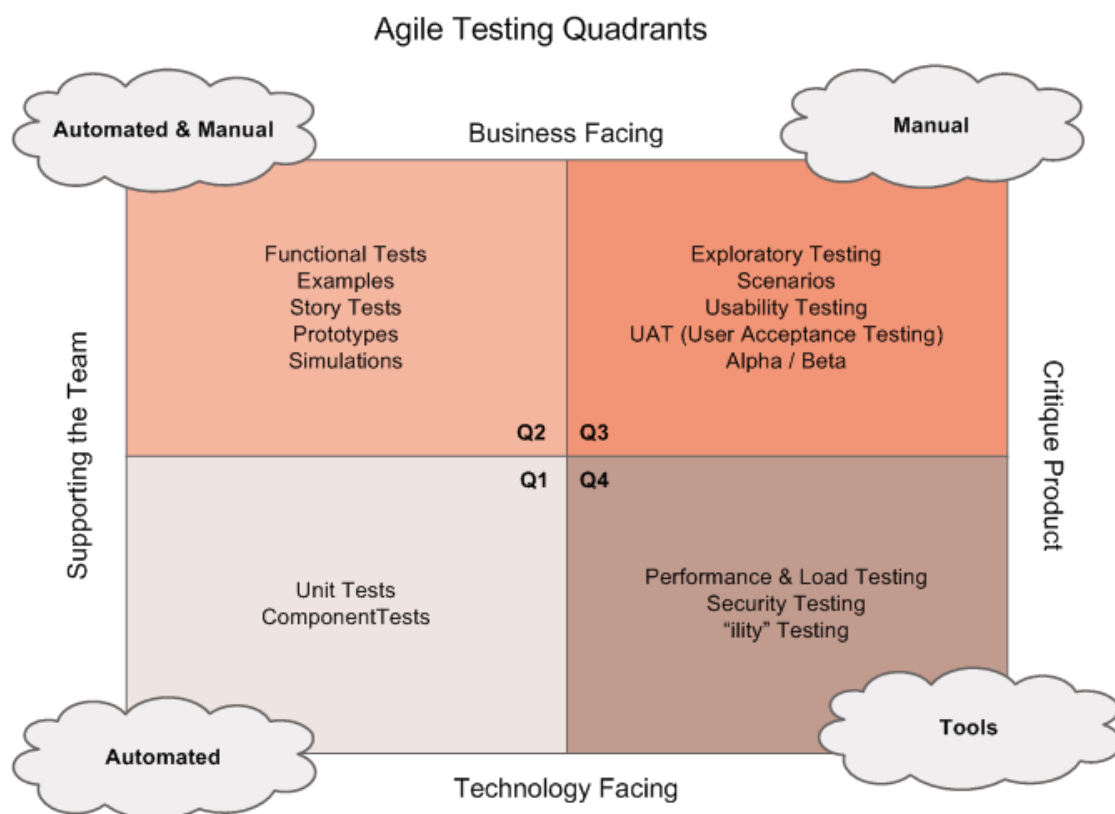


Abbildung 6: Agile Testquadranten

6 Testkonzept nach ISO 29119

Das *TESTKONZEPT*® ist das Fundament für ein erfolgreiches Testprojekt. Die Struktur gemäss ISO-Norm 29119:

- **Einleitung**
 - Einleitung
 - Ziel und Zweck
 - Geltungsbereich
 - Geltende Dokumente / Referenzen
- **Testobjekte**
 - Scope: Welche Komponente / Umsysteme werden getestet
 - Constraint: Was ist explizit **nicht** Bestandteil
- **Testumfang**
 - Qualitätsmerkmale (Funktionalität, Zuverlässigkeit, Wartbarkeit...)
 - Qualitätskriterien (Richtigkeit, Angemessenheit, Stabilität, Performance...)
- **Kommunikationswege**
 - Mitglieder / Rollen
- **Risikobetrachtung**
 - Risiken nach Auffindbarkeit, Schweregrad, Komplexität
 - Adäquate Massnahmen
- **Teststufen**
 - Komponententests
 - Integrationstest
 - Systemtests
 - Abnahmetests
- **Lieferobjekte**
 - Notwendige Artefakte (Testkonzept, Testdaten, Reports...)
 - Verantwortlichkeiten bei den Artefakten
 - Empfänger der Artefakte
- **Komponenten- / Unit-Tests**
- **Planung**
 - Terminplanung
 - Zeitachse, GANT-Charts

- **Abnahmetest**
 - Durchführung & Freigabe
 - Abnahmeprotokolle
- **Testabbruch**
 - Kriterien
 - Wer veranlasst dies?
 - Tätigkeiten für Wiederaufnahme
- **Testaktivitäten**
 - Analyse, Vorbereitung, Durchführung, Auswertung, Abschluss
- **Personal**
 - Rollen (Testmanager, Tester, BUG(!)-Team)
- **Testdesigntechniken**
 - Wie werden Testfälle hergeleitet?
- **Testendkriterien**
 - Metriken
 - Status: “passed” / “failed”
- **Anforderungen an die Testdaten**
- **Anforderungen an die Testumgebung**
- **Regressionstests und Fehlernachtests**