



### Integration PM/TM

**Integration des TM in das PM**

Das Testmanagement-Framework (TMF) gliedert sich nahtlos in das Projekt-Management-Framework (PMF) ein und bildet die Grundlage für die Testaktivitäten innerhalb des Softwareentwicklungsprojektes. Das Testmanagement unterstützt die Testaktivitäten in den Projekten durch die grundsätzliche Festlegung der Testmethodik, hilfreiche Anleitungen sowie einheitliche Verfahrensweisen und Definitionen. Um eine klare Unterscheidung zwischen den Aufgaben und Verantwortungen des PM und des TM zu erhalten, wird eine Zuständigkeitsmatrix erstellt, die anhand einer „RACI“ (siehe PMI) Zuordnung die verschiedenen Rollen und Verantwortungen genau definiert und beschreibt.

**Zuständigkeitsmatrix über die verschiedenen Rollen im Projekt**

	Projektleiter	Test Manager	Test Consultant	Tester	Test Auditor	Developer
1 Test-initialisierung	Verantwortlich	Durchführung	Durchführung	Mitwirkung	Mitwirkung	Mitwirkung
2 Test-vorbereitung	Durchführung	Durchführung	Durchführung	Durchführung	Mitwirkung	Mitwirkung
3 Test-durchführung	Durchführung	Durchführung	Durchführung	Durchführung	Mitwirkung	Mitwirkung
4 Test-abnahme	Durchführung	Durchführung	Durchführung	Mitwirkung	Mitwirkung	Mitwirkung
5 Test-abschluss	Durchführung	Durchführung	Durchführung	Mitwirkung	Mitwirkung	Mitwirkung

Legende: Verantwortlich (blau), Durchführung (rot), Information (grün), Mitwirkung (orange)

### Reporting

**Reportingebenen**

Fachbereich: über-greifend, Programm, Programm/Release, Projekt/Release, Teilprojekt, Arbeitspaket, Testfunktion/Service

IT

Die Standardisierung des Testmanagements gewährleistet die effiziente Aggregation der für einen Teststatus benötigten Informationen. Die unterschiedlichen Reportinganforderungen und Reportingansichten können hierdurch bis hin zu einer tagesaktuellen Statusmeldung abgedeckt werden. Das Reporting unterstützt hiermit die Koordination und Steuerung von IT-Projekten erheblich.

**Toolunterstützung**

- Mobile Apps
- Webservices
- MS-Report Services
- u. a. HP-QC/ALM

**Präsentationsebene**

- Mobile Endgeräte
- Präsentation
- Berichte
- Live-Analysen

### Ressourcenplanung

**Grundlagen der Planung**

Infrastruktur (Backend), Tester (IT), Tester (Fachbereich), Testfälle, Testdaten, Projektplan

**Ressourcenplanung**

Im Rahmen der Ressourcenplanung für ein Testprojekt sind einige zentrale Punkte zu beachten, damit der Test erfolgreich durchgeführt werden kann:

- Bei der Auswahl der Tester müssen alle erforderlichen Skills berücksichtigt werden
- Testinfrastruktur muss verfügbar sein
- Alle Testphasen (Integrationstest, Gesamtintegrationstest, Abnahmetest, ...) müssen bei der Planung berücksichtigt werden
- Komplexe Testketten (Prozesse) müssen berücksichtigt werden
- Geplante Softwarelieferungen sind zu berücksichtigen

**Testereinsatzplanung**  
Testfalldurchführungsplanung

**Finales Ergebnis:**

### Testautomatisierung

**„Prozessketten“:** Ein automatisierter Test von Prozessketten mit entsprechender Toolunterstützung (z. B. LoadRunner®) ermöglicht einen möglichst hohen Grad an Vergleichbarkeit und Messbarkeit der Testergebnisse.

**„Stress-/Performancetest“:** Um das Lastverhalten eines Systems zu testen, bieten automatisierte Stress-/Performancetests einen strukturierten und wiederholbaren Ansatz.

**„Regressionstest“:** Um auch Testobjekte testen zu können, die nicht direkt verändert wurden, die aber durch Seiteneffekte von Änderungen betroffen sein könnten, ermöglichen automatisierte Regressionstests eine effiziente Art, auch die Qualität dieser mit zu testen.

**„Unittest“:** Automatisierte Unittests (z. B. JUnit) ermöglichen, schon während der Entwicklung erste Fehler zu erkennen und diese vor dem eigentlichen Test zu beheben, was in den folgenden Testphasen die Stabilität der Testobjekte erhöht.

**„Testdaten“:** Durch eine automatisierte Testdatenerstellung erreicht man für jeden Test einen definierten Ausgangszustand, auf den alle weiteren Tests strukturiert aufsetzen können.

**Toolunterstützung (beispielhaft):**

- LoadRunner
- Rational Functional Tester
- app.test
- JUnit

Steigende Komplexität aber auch steigender Effizienzgewinn

### 1 Testinitialisierung

Innerhalb der Testinitialisierungsphase wird der Grundstein eines erfolgreichen späteren Tests gelegt. Hierzu werden sowohl Testinhalt und -umfang als auch die den Test umgebenden Rahmenbedingungen evaluiert, festgelegt und vorbereitet.

**Aktivitäten**

- Überprüfung Startfaktoren
- Initialisierung Teststeckbrief
- Initialisierung Action Item List für den Test
- Initialisierung Milestone-Planung
- Initialisierung Ressourcen- und Kapazitätsplanung
- Initialisierung Aufwands- und Budgetschätzung
- Initialisierung Risikomanagement
- Phasenabschluss-präsentation zur Testinitialisierung
- Abgleich und Überprüfung Endfaktoren

**Ergebnis-objekte**

- Teststeckbrief
- Action Item List
- Milestone-Planung
- Ressourcen- und Kapazitätsplanung
- Aufwands- und Budgetschätzung
- Risikomatrix Test
- Abschluss-präsentation zur Testinitialisierung

### 2 Testvorbereitung

Um die Testdurchführung erfolgreich zu gestalten, wird im Rahmen der Testvorbereitungsphase die organisatorischen Prämissen geschaffen. Wichtige Voraussetzung für das optimale Erreichen der festgelegten Testziele ist eine präzise Testvorbereitung.

Dabei kann die Testvorbereitungsphase nochmals in die Segmente *Organisatorisches*, *Testplanung*, *Testdesign*, *Testereinsatzplanung* und *Testinfrastruktur* gegliedert werden. Dies unterstützt die jeweiligen Verantwortlichen bei der strukturierten Bewältigung der unterschiedlichen Aufgaben, Herausforderungen und Risiken sowie der zielgerichteten Ressourceneinsatzplanung.

Eine effektive Testvorbereitung beinhaltet ebenfalls ein vollumfängliches Risikomanagement, dass eng mit der Gesamtprojektleitung koordiniert werden muss. Nur so kann proaktiv und rechtzeitig auf Risiken reagiert und Gegenmaßnahmen eingeleitet werden.

**Aktivitäten**

- Überprüfung Startfaktoren
- Einrichtung Aufbauorganisation
- Durchführung Kickoff TP Test
- Definition/Detailierung Teilprojekte
- Supplier Management Externe
- Bereitstellung Testerhandbuch
- Anbahnung Kickoff „Teststart“
- Konkretisierung Milestone-Planung
- Konkretisierung Ressourcen- und Kapazitätsplanung
- Festlegen Testmanagement-Team
- Erstellung RACI-Matrix
- Konkretisierung Aufwands- und Budgetschätzung
- Anfertigung testspezifisches Reporting
- Abnahmekriterien festlegen
- Entwicklung Testkonzept
- Abnahme Testkonzept
- Detaillierte Anforderungen
- Analyse der Anforderungen, Prioritäten und Risiken
- Identifikation der betroffenen Prozesse und Systeme
- Identifizieren der Testscenarien
- Erstellen der Testfälle
- Definition der Testdaten
- Definition Anforderung an Testumgebung
- Identifikation Regressions-aufwand
- Bewerten der Testabdeckung
- Entwicklung Testfälle
- Testereinsatzplanung
- Anforderung und Zuordnung Ressourcen
- Entwicklung Testereinsatzplanung
- Beauftragung Testerschulung
- Aufbau Testinfrastruktur
- Planung und Beauftragung Testdaten
- Installation und Einrichtung Testtool
- Planung und Beauftragung Testlabor
- Planung und Beauftragung Berechtigungen
- Abgleich und Überprüfung Endfaktoren
- Konkretisiertes Testfallportfolio

**Ergebnis-objekte**

- Testorganigramm
- TP Test Kickoff-Unterlagen
- Test-Steckbrief
- Testerhandbuch
- Unterlagen Kickoff „Testbeginn“
- Risikoliste
- Detailierung Risikomanagement
- Testplan
- Milestone-Planung
- Ressourcen- und Kapazitätsplanung
- RACI-Matrix
- Aufwands- und Budgetschätzung
- Testkonzept
- Testkonzept
- Testkonzept
- Testkonzept
- Überblick-Präsentation
- Liste priorisierte Anforderungen
- Beziehungsmatrix Anforderungen, Prozesse und Systeme
- Katalogisierte Testfälle
- Spezifizierte Testfälle
- Testdaten-datenbank
- Regressions-testsuite
- Abdeckungs-matrix
- Testprotokoll
- Ressourcen-planung Tester
- Testfallmaster
- Ressourcen-planung Tester
- Infrastruktur-anforderung
- Infrastruktur-anforderung
- Infrastruktur-anforderung
- Berechtigungs-planung
- Abgleich und Überprüfung Endfaktoren

**Legende:**

- Aktivität (blau)
- Kritische Aktivität (Kritikalität = Priorität, Zeit oder Komplexität) (rot)
- Startpunkt (blau)
- Endpunkt (orange)

### 3 Testdurchführung

Die Testdurchführung beinhaltet alle Aktivitäten, die im Rahmen der Bearbeitung der vorbereiteten Testfälle durchzuführen sind.

Hierzu zählt unter anderem die Einarbeitung und Schulung der Tester, ein proaktives Abweichungsmanagement und der gesamte Bereich Reporting und Dokumentation.

**Aktivitäten**

- Überprüfung Startfaktoren
- Kickoff Teststart
- Annahmetest
- Durchführung Test
- Steuerung Test
- Kontinuierliches Abweichungsmanagement
- Laufende Testdokumentation
- Laufendes Statusreporting des Tests
- Management Testinfrastruktur
- Abgleich und Überprüfung Endfaktoren

**Ergebnis-objekte**

- Kickoff Teststart Unterlagen
- Testprotokoll
- Testprotokoll
- Testprotokoll
- Statusreport, Abweichungs-status

### 4 Testabnahme

Eine Überprüfung von Abnahmekriterien und Erfüllungsgrad in Bezug auf in der Testinitialisierung fixierte Ziele erfolgt während der Testabnahme.

Nachdem sichergestellt wurde, dass sowohl fachliche als auch technische Ansprüche an die Software für einen Go-Live erfüllt sind, kann an dieser Stelle über einen Abschluss der Projektaktivitäten entschieden werden.

**Aktivitäten**

- Überprüfung Startfaktoren
- Auswertung Testergebnisse
- Bewertung Testergebnisse
- Testabnahme
- Übergabe Restanten für Rollout
- Beauftragung Testerschulung
- Abgleich und Überprüfung Endfaktoren

**Ergebnis-objekte**

- Übersicht Testfälle
- Bewertung Testergebnisse
- Abnahme-erklärung
- Offene Abweichungen

### 5 Testabschluss

Innerhalb der Testabschlussphase sind sämtliche Aktivitäten aufgeführt, die Voraussetzung für eine erfolgreiche Beendigung des Tests sind.

Das umfasst neben der Bereitstellung eines Abschlussberichts auch die letztmalige Durchführung eines Projektabschluss-meetings. Weitere Aktivitäten beinhalten die Archivierung von generierten Projektdokumenten sowie den Wissenstransfer in Form von Lessons Learned and Feedback.

**Aktivitäten**

- Überprüfung Startfaktoren
- Bereitstellung Projektabschlussbericht
- Aufsetzen Lessons Learned
- Abgabe Testergebnisse
- Ab- und Rückbau Testinfrastruktur
- Abgleich und Überprüfung Endfaktoren

**Ergebnis-objekte**

- Testabschlussbericht

### Rollenkonzept

Rolle <sup>1)</sup>	Projektgröße			Funktion
	large	medium	small	
Test Manager	Verantwortlich	Durchführung	Durchführung	Übernahme der Verantwortung für den gesamten Test (als Teilprojektleiter Test) oder eines definierten Teils des Tests
Test Analyst (fachlich/technisch)	Durchführung	Durchführung	Durchführung	Übernahme der operativen Aktivitäten im Test (alle Phasen)
Tester	Durchführung	Durchführung	Durchführung	Übernahme der operativen Aktivitäten in der Testdurchführung
Test Auditor	Information	Durchführung	Durchführung	Fachliche Abnahme der Testfälle
Test Infrastructure Coordinator	Durchführung	Durchführung	Durchführung	Übernahme der Aktivitäten hinsichtlich der Testumgebungen
Test Consultant	Durchführung	Durchführung	Durchführung	Beratung des Testmanagements
Defect Manager	Durchführung	Durchführung	Durchführung	Verwaltet gemeldete Abweichungen, Schnittstellen zur Realisierung von Testseite
Dispatcher	Durchführung	Durchführung	Durchführung	Schnittstelle zum Test auf Realisierungsseite, verteilt Defects auf Entwickler
Developer	Durchführung	Durchführung	Durchführung	Stellt Software bereit und löst Defects

<sup>1)</sup> Eine Person kann auch mehrere Rollen gleichzeitig wahrnehmen.

Legende: Verantwortlich (blau), Durchführung (rot), Information (grün), Mitwirkung (orange)

### Zielsetzung

**Motivation für die Standardisierung**

Die Anforderungen an den Software-Application-Life-Cycle sind unverändert hoch. Allerdings nimmt im Zuge steigender Ansprüche an schnelle und rechtzeitige Softwareproduktionen bzw. -einführungen und sich permanent ändernde Entwicklungsprozesse die Notwendigkeit eines strukturierten Testens immer mehr zu. Während z. B. früher einfache, sequentielle Programmfunktionen das Testvorgehen bestimmte haben, beherrschen mittlerweile Webservices, Scrum-Vorgehensmodelle und völlig neue Anwendungsarchitekturen mit sich ändernden Sourcingmodellen die Anforderungen an den Test. Z. B. wurden Geschäftsprozesse über Applikationen hinweg „end-to-end“ getestet, wofür heute neue Techniken, Verfahren, Methoden & Tools notwendig sind, die ein lose gekoppeltes Konstrukt an Webservices, vielleicht sogar in einer Public-Cloud, testen können. „Testen“ heißt heute Standardisierung und Komplexitätsbeherrschung.

**Zielsetzung des Testmanagements**

Zielsetzung des Testmanagement-Framework (TMF) ist es, einen verbindlichen und praxisorientierten Rahmen für das Testen bereitzustellen. Dieser Rahmen soll die Softwarequalität sicherstellen, als Leitfaden für die Projektarbeit im Testmanagement dienen und alle Testphasen strukturiert unterstützen. Wesentliche Bestandteile sind:

- Ein standardisiertes Testvorgehen inkl. Vereinheitlichung von Begriffen
- Definitionen aller für das Testen durchzuführenden Aktivitäten
- Sicherstellung der u. a. für revisionstechnische Belange erforderlichen Dokumentation
- Effizienter und effektiver Ressourceneinsatz
- Durchgängige Toolunterstützung für alle Testaktivitäten mit der Möglichkeit einer Aggregation der Testinformationen
- Definition des Testrahmens für Testoutsourcing (Near-Off-Shore, TestFactories, interne Testorganisationen bzw. Qualitätsmanagement-Einheiten)

### Ansprechpartner

Stefan Lauritzen Partner stefan.lauritzen@bearingpoint.com	Wolfgang Correnz Partner wolfgang.correnz@bearingpoint.com
--	--

Version: 2.1 Stand 30.06.2015