

sd&m

A Company of Capgemini



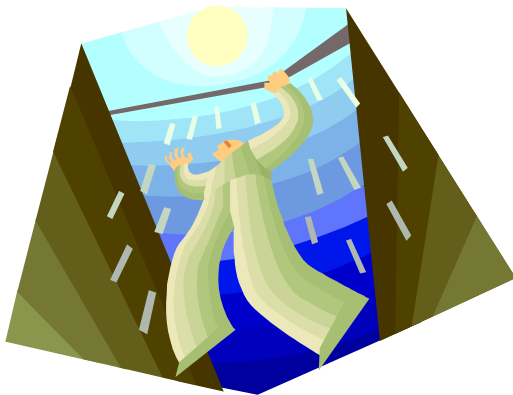
Testen in Großprojekten

Erfahrungen aus der Praxis

sd&m AG
software design & management
Carl-Wery-Str. 42
81739 München
Telefon 089 63812-0
www.sdm.de

Andreas Kellerstraß
Bad Honnef, 5. Mai 2006

These: Die Probleme des Testens in Großprojekten sind nicht konzeptionell



Die Probleme des Testens sind nicht konzeptioneller Natur (hierzu ist fast alles gesagt), der Test reibt sich an den vorgefundenen Projektbedingungen.

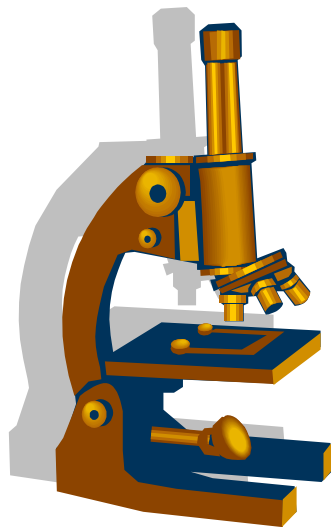
Das Testmanagement muss lernen ungünstige Umstände zu akzeptieren und proaktiv an die Erfordernisse des Testens anzupassen.

Insbesondere in Großprojekten* ist der Druck aus den Gegebenheiten das Beste zu machen am größten.

- **80% aller Großprojekte scheitern**, nicht zuletzt auch wg. Defiziten im Test.
- Der offizielle **Handlungsspielraum für den Test ist in Großprojekten am kleinsten**, z. B. aufgrund
 - der erzwungenen Übernahme von Altlasten (aus x Vorprojekten)
 - des fehlenden direkten Zugriffs auf die Entwicklungskette
 - der Abhängigkeiten von vielen externen Zulieferern und Partnern
- Aufgrund der **großen, absoluten Zahlen** treten viele prinzipielle **Testprobleme klarer** zu Tage, als in kleineren Projekten.
- Die **Organisation von Großprojekten ist stark ausdifferenziert**.
Der Test ist ein eigenes (Teil-)Projekt mit expliziter Testorganisation abseits der „Fachbereiche“.
- Großprojekte unterliegen nicht zuletzt einem **sehr hohen Druck** z. B. durch
 - *Budget-Deckelung* (als Reaktion auf projektweite Kostenexplosion)
 - fixe *Endtermine* (z. B. aufgrund gesetzlicher Rahmenbedingungen)
 - hohen *Erfolgsdruck* nach (üblicherweise) x Neuanläufen
 - unternehmensinterne *Umstrukturierung* wie z. B.
 - Bildung neuer Linienorganisation spätestens gegen Ende des Projektes
 - Neuausrichtung des Betriebs - z. B. nach ITIL - oder Outsourcing
- Die Laufzeit beträgt in der Regel mehrere Jahre, d. h. das **Rahmenbedingungen sind einem erheblichen Wandel unterworfen**. Beispiel: Fluktuation der Mitarbeiter.

* Projekte mit > 100 MA und/oder einem Volumen > 10 M €

Das Betrachten von Testen in Großprojekten ist wie der Blick durch eine (Zeit-)lupe.



Der hohe Differenzierungsgrad und die lange Laufzeit in Großprojekten schaffen einen Blick auf Testen wie durch eine (Zeit-)Lupe.

Dadurch werden die Kernthemen sichtbar in denen das Testen seine größten Reibungsverluste erfährt.

Das Ziel des Vortrages ist ...

- ← **nicht** eine erneute Auflistung von klassischen Testproblemen wie
- Ressourcenmangel
 - Missbrauch von Testzeiten als Puffer für die Entwicklung
 - Mangelnde Testbarkeit und Automatisierungsprobleme
 - Probleme mit Testdaten
 - etc.

die sich in Großprojekten „nur“ quantitativ dramatischer auswirken.

→ **Sondern**

die Darstellung von oft beobachteten **grundlegenden Defiziten** die auf dem Testing lasten



die zur „**zwangsläufigen**“ (?) **Übernahme von testfremden Aufgaben führen** (... den letzten beißen bekanntlich die Hunde)


mit der Idee, daraus **eine grundsätzlich andere Einbettung des Testing** in ein (Groß-)Projekt abzuleiten und damit insgesamt die Erfolgchancen zu erhöhen.

Bei den Zulieferern für den Test liegen testfremde und chronisch unterschätzte Aufwände des Testmanagements

An der Schnittstelle zu folgenden Gebieten fallen die Zusatzaufwände des Testmanagements besonders groß aus:

- **IT-Service-Management (Betrieb)**
 - *Wg. Mängeln bei Bereitstellung, Betrieb und Test der Infrastruktur (Zeiten, Stabilität, Verfügbarkeit)*
- **Anforderungs- und Changemanagement**
 - *Wg. Spezifikationsmängeln (Priorisierung, Stabilität), und Mängeln im Changeprozess*
- **Integrationsmanagement**
 - *Wg. ungeplanter fachlicher und technischer Integration und Konfiguration*
- **Releasemanagement**
 - *Wg. mangelhafter Synchronisation Entwicklung – Test (Auslieferungen, Bug-Fixing, etc.)*
- **Projektmanagement**
 - *Wg. Genereller Missachtung des Testaufwandes, falscher Zeitplanung, Fokussierung auf frühe Phasen, etc.*


IT-Service-Management (Betrieb)

- Hoher Kostenfaktor für die Bereitstellung adequater Testumgebungen => zu wenige und nicht adequate Umgebungen, chronische Budgetnot
 - „parasitierende“ Interessengruppen auf den Testsystemen => eingeschränkte Verfügbarkeit
 - Testumgebung steht bereit und kann nicht genutzt werden (aufgr. Verzögerung) => Zunehmender Unwillen des Betriebes
 - Manuelle Installationsprozesse, fehlendes betriebl. Konfigmanagement => hohe Rüstzeiten, Umgebung fehleranfällig, eingeschränkte Reproduzierbarkeit
 - Testumgebungen werden nicht als Produktivsysteme gefahren => eingeschränkte Verfügbarkeit, Instabilität
 - Der Betrieb versteht sich als reiner „Befehlsempfänger“ => kein „Mitdenken“,
 - Test verlangt ein hohes Maß an Flexibilität und ad hoc-Verfügbarkeit => schwer kalkulierbar für Betrieb.
 - „Der Betrieb macht keine Fehler“
 - applikatorische Mängel behindern Umgebungstests bzw. destabilisieren die Umgebung
- 
- „**produktionsnahe**“ **Testumgebungen** inkl. Betrieb bereits zu Projektbeginn anfordern.
 - Vereinbarung von **Betriebsservices und SLAs** mit dem Betrieb. Installation eines SPOC zum Betrieb.
 - **Betrieb in Testplanung einbeziehen.** Vermittlung des eigentlichen Zielbedarfs des Testing.
 - **Hoher Automatisierungsgrad** des Installationsprozesses und betriebliches Konfigmanagement bereits im Testbetrieb.
 - Definition und Durchführung eines **produktionsnahen Testbetriebs**, inkl. Überwachung.
 - Definition eines eigenen **Testteams im Betrieb**, der Testumgebungen und Betriebsprozesse testet.
 - **Einbinden eines versierten technischen Architekten** des Lieferanten in betriebliche Tests der Testumgebungen insbesondere der künftigen Produktionsumgebung.

Anforderungs- und Changemanagement

Fixierung, Konsolidierung und laufende Aktualisierung

- Fachbereiche haben Hemmungen eine komplexe und umfangreiche Spezifikation abzunehmen => Fehlen der Testgrundlage
- Unkonsolidierte und teilw. Widersprüchliche Menge von Vertrags- und Spezifikationselementen aus verschiedenen Projektphasen und unterschiedlichsten Entstehungszeitpunkten. => kein gemeinsames Verständnis der Abnahmekriterien
- Die Absolutzahl offener Punkte oder Spezifikationsfehler ist nach Ende der offiziellen Spezifikationsphase sehr hoch. Teile wurden bewusst nicht näher detailliert. => hoher Klärungs-, Interpretations- und Änderungsaufwand
- Detaillierungen und Ergänzungen durch Entwicklung und Test werden nicht in der Spezifikation nachgezogen. => keine verlässliche Dokumentation


- 
- **Triggern der Abnahme** der Spezifikation um eine ausreichende formale Testgrundlage zu besitzen. (Die Entwicklung geht meist pragmatischer damit um).
 - **Extraktion und Konsolidierung** aller Requirements aus bestehender Dokumentenlage. Gemeinsames Verständnis mit Dienstleister erreichen.
 - Detaillierte Nach-QS und Klärung von restlichen **offenen Punkten** bei Testfallerstellung und -durchführung.
 - **Verwaltung aller Spezifikationsänderungen** und Triggern der Aktualisierung der betroffenen Spezifikationselemente.

Anforderungs- und Changemanagement


Priorisierung und Risikobewertung, weiße Flecken

- Es existiert keine mit dem Management und dem Dienstleister vorgängig abgestimmte Priorisierung von Funktionalität anhand einer Risikobetrachtung => Testfallerstellung ohne fachl. Schwerpunkte, keine gemeinsame Bewertung des Projektstatus
 - Eine hohe Fehlerzahl vor einem fixen Endtermin => Strategie möglichst effektiver Fehlerbehebung erforderlich
 - Teile der Spezifikation wurden falsch umgesetzt oder die Fehlerzahl ist zu hoch => Backupszenarien erforderlich
 - Änderungen an Randsystemen und andere weiße Flecken wurden übersehen => Umsetzung erfolgt zu spät und verhindert Gesamttest
- 
- Nachgelagerte **fachliche Priorisierung** durch Risiko-Klassifizierung der Testfälle und der gefundenen Fehler parallel zur Bewertung des Schweregrades eines Fehlers.
 - Durchführung von „**Fehler-Clearings**“ mit dem Lieferanten zur Erreichung eines gemeinsamen Verständnisses über die „Kritikalität des Fehlers und des Status der umgesetzten Funktionalität.“
 - Vorgängige Identifikation von **goldenen Henkeln** und möglichen Einschränkungen des Lieferungsumfanges.
 - Vorgängige Identifikation und ggf. Festschreiben möglicher **Workarounds**.
 - Vorgängige Veröffentlichung und Abstimmung eines **Integrationsplanes** für den Test.


Integrationsmanagement

- Integration beschränkt sich auf Schnittstellen => kein Verantwortlicher mit Gesamtüberblick über das System und übergreifende Prozesse.
 - Fachliche Gesamtkonfiguration (-Parametrierung) auf Detailebene undefiniert (Rechte, Rollen, Vorlagen, etc.) => fehlende oder inkonsistente Gesamtkonfiguration
 - Unterschiedlichste Testumgebungen der Rand- und Teilsysteme => keine übergreifend konsistenten Testdaten
 - Datenmigration als separates Teilprojekt und nicht unter Integrationsmanagement. Spezifikation und Umsetzung verschleppt. => keine rechtzeitige Bereitstellung konsistenter Migrationsdaten.
 - Lange Entwicklung gegen eine definierte Schnittstelle unter Verwendung von Emulatoren. Eigentliche Integration der „Echtsysteme“ findet sehr spät und erst auf den Testsystemen des Auftraggebers statt => Tragfähigkeit der Schnittstellen zeigt sich zu spät.
 - Sonderthema Druck: Outputmanagement ist komplex und sensibel und wird chronisch unterschätzt
- 
- Erteilung der **Abnahme** erst nach Nachweis eines funktionierenden Gesamtsystems. Frühzeitige Definition und Abstimmung vollständiger, Teilsystemübergreifender Testketten. Übernahme von **Gesamtsystemverantwortung**.
 - Etablierung eines **fachlichen Konfigurationsmanagements** im Test.
 - Etablierung eines fachlichen **Datenmanagements**, zur konsistenten Bereitstellung der Testdaten aus
 1. den unterschiedlichen Randsystemen
 2. den zeitlich abgestimmten Migrationsbeständen.Integration der Datenmigration in ein übergreifendes Datenmanagement.
 - Frühzeitiger Durchstich an allen Schnittstellen zur **Verifikation der Tragfähigkeit**. Beteiligung versierter Architekten an den Schnittstellentests.
 - Etablierung eines eigenen **Outputmanagements**, das den gesamten Prozess von der Druckdatenerzeugung bis zur Couvertierung kontrolliert.

Releasemanagement

- Fehlerbehebung des Dienstleisters unter Termindruck erfolgt
 - ohne Rücksprache mit Fachbereich oder Testteam und
 - orientiert sich an der Komplexität der Fehlerbehebung
 - erfolgt nach Vorgaben wie: Reduktion um 5 Fehler / Stunde => hohes Risiko das Fehlerbehebung nicht zur eigentlichen Zielerreichung führt.
 - Das Schneiden neuer Release (inkl. Bugfixes) wird von der Entwicklung, ausschließlich nach Ihrem Fortschritt, diktiert. => Testvoraussetzungen werden ignoriert und führen zu Verzögerungen.
 - Keine aktive Steuerung von Releases durch den Fachbereich auf Basis einer Priorisierung.
 - Keine Abstimmung von Teilprojekten untereinander im Sinne einer ganzheitlichen Fehlerbehebung, insbesondere fachlich und technisch.
 - Lieferungen kommen unkontrolliert auf Testumgebungen => Gesamtüberblick, wo was getestet wird geht verloren.
- 
- Etablierung eines **teilprojektübergreifenden fachlichen Releasemanagements** zur aktiven Steuerung von Release-Inhalten orientiert an
 - fachlicher Priorisierung
 - technischer Problembehebung
 - an Testerfordernissen
 - **Starke Verzahnung von Entwicklung- und Testplanung.** Releases müssen die Abhängigkeiten im Test beachten.
 - **Einbezug der Fachbereiche** in die Definition der Release-Inhalten auch im Rahmen des Bug-Fixing.
 - Verhindern von chaotischen Lieferprozessen durch Definition und **Einüben des Auslieferungsprozesses**, von der Annahme und Vorprüfung eines Releases bis zur Übernahme in den eigentlichen Test (Quasiproduktion).

Projektmanagement

- Fehleinschätzung des Testaufwandes und der Sinnhaftigkeit, Test „produziert nichts und kostet nur“ => Test erhält zu geringes Budget, Aufwand wird chronisch gedrückt.
 - Starke Fokussierung auf frühe Projektphasen und Ihre „Probleme“ => keine „Einmischung“ in frühen Phasen möglich.
 - Geringe Kompetenzen für den Test in ausdifferenzierter Projektorganisation => hohe Abhängigkeiten ohne ausreichende Möglichkeit zur Einflussnahme
 - Starke Fokussierung auf KPIs (Durchführungsrate, Fehlerbehebungsrate, etc.) => Messung/Bewertung des Testens statt Verwendung des Tests als Messinstrument.
 - Wille zur Zielerreichung (Einführung) übersteuert Bedenken des Testmanagement => Test als Bedenkenträger und Erfolgsverhinderer
- 
- Vorgängig Transparenz schaffen für Auswände im Test vor allem für **testfremde, chronisch unterschätzte Aufwände** im Test
 - Etablierung von Testen als **Partner des Qualitätsmanagements** während der gesamten Projektlaufzeit, nicht alleine gebunden an den Testgegenstand „Software“.
 - Abhängigkeiten in der Projektorganisation Transparent machen, Qualität der „Lieferanten“ einschätzen und **Kompetenzen zur Einflussnahme** vorgängig einfordern.
 - Ausführliches Reporting aus dem Test auch mit dem Ziel eines **Selbstmarketing**.
 - Etablierung des Tests als „**Stethoskop des Projektmanagements am Patienten Projekt**“
 - Vorgängige Verknüpfung von Zielerreichung (Geschäftsplanung) und **Risikobetrachtung** im Test.

Fazit

Testen in großen Projekten kann gelingen, wenn in den Kernbereichen

- IT-Service-Management,
- Anforderungs- und Changemanagement,
- Integrationsmanagement
- Releasemanagement
- Projektmanagement

die Erfordernisse des Tests ausreichend berücksichtigt werden.

Das Projektmanagement muss die Interessen des Tests in diesen Kernbereichen vertreten und ggf. Prozessverbesserungen durchsetzen.

Testfremde Aufgaben sind zu reduzieren.

Für absehbare Mehraufwände im Test aufgrund Reibungsverlusten und schlechter Zulieferungen ist dem Test entsprechendes Budget zuzugestehen.

Das Testmanagement muss unabhängig von der Software als operativer Partner des projektübergreifenden Qualitätsmanagements betrachtet werden und eine stärkere Rolle direkt unterhalb des Projektmanagements spielen.