

Testmanagement unter dem Einfluss der Software-Industrialisierung

Andreas Birk

Software.Process.Management
Gutenbergstraße 99
70197 Stuttgart
andreas.birk@swpm.de

Abstract: Die Software-Entwicklung unterliegt seit einigen Jahren einem Trend zur Industrialisierung, der auf mehreren Ebenen abläuft. So berührt er den Aufbau und die Technologien der Software-Systeme ebenso wie die Vorgehensweise bei Software-Entwicklung und IT-Betrieb. Das Testmanagement ist ein Teilbereich des Software-Managements, das aktuell in besonderem Maße dem Einfluss der Software-Industrialisierung ausgesetzt ist.

Dieser Artikel erfasst die Auswirkungen der Software-Industrialisierung auf das Testmanagement und identifiziert die Herausforderungen, die sich daraus ergeben. Er stellt Maßnahmen vor, mit denen das Testmanagement auf die Herausforderungen reagieren kann. Fallbeispiele veranschaulichen und belegen, wie die Maßnahmen in der Praxis von IT und Software-Entwicklung eingesetzt werden.

1 Einleitung

Industrialisierungsbestrebungen treiben schon seit Längerem Entwicklungen in der Software- und IT-Branche voran. Versteht man Struktur und Auswirkungen der Software-Industrialisierung, kann man die aktuellen und künftigen Anforderungen an die Branche besser überblicken und die Entwicklungen optimal mitgestalten. Dazu gilt es folgende Fragen zu beantworten: Welche Facetten besitzt die Industrialisierung? Welche Herausforderungen bringt sie für die Software-Praxis? Wie reagieren Software-Entwicklung und -Management darauf?

Das Testmanagement ist den Industrialisierungstendenzen zurzeit besonders stark ausgesetzt. Als überschaubarer Teilbereich des Software-Managements bietet es eine geeignete Arena, um die Auswirkungen der Industrialisierung näher zu analysieren.

Dieser Artikel gibt zunächst einen Überblick über die Facetten der Software-Industrialisierung (Kapitel 2). Danach identifiziert er die Herausforderungen der Industrialisierung für das Testmanagement (Kapitel 3) und zeigt, mit welchen Maßnahmen das Testmanagement darauf reagiert (Kapitel 4). Kapitel 5 illustriert die Maßnahmen mit Fallbeispielen. Die abschließende Diskussion (Kapitel 6) fasst die Beobachtungen zusammen und überträgt sie auf das Software-Management allgemein. Die beiden Unterkapitel dieser Einleitung möchten zunächst das Phänomen der Software-Industrialisierung und seine Relevanz für das Testmanagement näher fassen.

Das zentrale Anliegen des Artikels ist es, die Industrialisierungstendenzen rund um das Testmanagement zusammen zu tragen und die Maßnahmen zu identifizieren, mit denen die Praxis auf die Entwicklungen reagieren kann. Es handelt sich um eine erste explorative Untersuchung auf Grundlage von Literaturstudien und der langjährigen Erfahrungen des Autors in verschiedenen Industriedomänen. Weitergehende Studien sollten künftig die identifizierten Tendenzen weiter belegt und untersuchen sowie zusätzliche Maßnahmen identifizieren und ihre Wirksamkeit belegen.

1.1 Software-Industrialisierung

Industrialisierung bezeichnet den Einsatz von Arbeitsteilung und Spezialisierung sowie maschinellen, automatisierten und rationalisierten Verfahren zur Erstellung und Distribution von Gütern und Dienstleistungen [Me]. Es steht im Gegensatz zu den ursprünglich bestehenden handwerklichen und manuellen Verfahren, aus denen es sich schrittweise entwickelt hat. Die Software- und IT-Branche wendet zwar von Beginn an automatisierte Verfahren zur Datenverarbeitung an. Ihre Anfänge hatten jedoch eher handwerkliche als industrielle Züge. Ihr industrielles Potenzial hat sich erst allmählich und schrittweise entwickelt.

Heute, so scheint es, ist die Software-Industrialisierung so weit fortgeschritten, dass zumindest ihr Potenzial dem Niveau anderer etablierter Industriebereiche entspricht - beispielsweise dem Maschinenbau. Zugleich ist die Branche noch intensiv damit beschäftigt, dieses Potenzial auf breiter Front und nachhaltig zu erschließen. Von den Erfolgen zeugen beeindruckend komplexe und leistungsfähige Systeme wie die Verfahren zur Zahlungsabwicklung im Bankenbereich oder die Software-Steuerung der großen Telekommunikationsnetze. Die nach wie vor verbreiteten Herausforderungen und Schwierigkeiten werden klar, wenn man sich die (Miss-)Erfolgsraten von Software-Projekten anschaut oder Nachrichten verfolgt, wie kürzlich über das anfängliche Scheitern der Gepäckverteilung im neuen Terminal 5 des Flughafens London-Heathrow.

Die Industrialisierung im Software- und IT-Bereich weist einige Besonderheiten auf, die man zunächst kennen muss, um ihr Potenzial und ihre Herausforderungen besser zu verstehen. Der grundlegende Unterschied zur herkömmlichen Industrialisierung ist, dass in der Software-Industrie Produktion und Distribution eine untergeordnete Rolle spielen. Im Kern ist die Software-*Herstellung* ein kreativer Problemlösungs- und Entwicklungsprozess. Somit können herkömmliche Verfahren der Industrialisierung von Produktion und Distribution nur eingeschränkt auf Software und IT übertragen werden. Die Industrialisierung der Branche bewegt sich weitgehend auf neuem Terrain.

Recherchiert man in Literatur und Internet, so findet man recht verschiedene, wenn auch miteinander verwandte Ausprägungen der Software-Industrialisierung bzw. ähnlicher Begriffe und Konzepte:

- Software-Industrialisierung [BBT05];
- IT-Industrialisierung, insbesondere Industrialisierung des IT-Betriebs [FS07];
- Prozessverbesserung in der Software-Entwicklung und im IT-Betrieb (z. B. CMMI, ITIL V3 und CobIT) [OCG];
- Produkttechnologien auf industriellem Niveau (z. B. hochintegrierte ERP-Systeme sowie Service-orientierte Architektur, SOA);
- Produktlinien-Entwicklung und Software-Produktmanagement [CN02]; [HSH06];
- Komponentenbasierte Verfahren und Software Factory [Cu91]; [GS04].

Es gibt also Ansätze, die sich auf Software-Entwicklung konzentrieren, während andere den IT-Betrieb betrachten. Teils geht es um Produkttechnologie, teils um Prozessaspekte. Manche Verfahren widmen sich technischen, implementierungsnahen Fragen. Andere adressieren vorwiegend strategische Management-Aspekte.

1.2 Software-Management und Testmanagement

Testmanagement ist ein Teilbereich des Software-Managements. Es umfasst die Planung und Steuerung der Testaktivitäten [Sp06]. Damit hat es besonderen Einfluss auf die Qualität des erstellten Produktes sowie auf die Einhaltung der Zeit- und Kostenvorgaben eines Projektes. Komplexe Testvorhaben besitzen heute häufig den Charakter eines Teilprojektes. Somit decken sich die Aufgaben des Testmanagements stark mit denen des allgemeinen Software-Managements. Alle neun Knowledge-Areas des PMI [PMI] finden eine Entsprechung in Aufgaben des Testmanagements.

Dennoch ist das Testmanagement spezifischer und einfacher zu fassen als das allgemeine Software-Management. Dies macht das Testmanagement zu einem geeigneten Kandidaten, um die Auswirkungen der Software-Industrialisierung näher zu untersuchen. Anschließend können die Erkenntnisse gut auf das Software-Management allgemein übertragen werden (siehe Kapitel 6).

2 Trends in der Software-Industrialisierung

Die Einleitung hat aufgezeigt, dass sich hinter dem Begriff *Software-Industrialisierung* recht unterschiedliche Phänomene verbergen. Um die Auswirkungen auf das Testmanagement besser verstehen zu können, fächert dieses Kapitel die Facetten der Software-Industrialisierung auf. Die Trends fallen in zwei große Gruppen:

- Trends bei den erstellten Software-Systemen (Produktspekte);
- Trends in Software-Entwicklung und -Betrieb (Prozess-/Organisationsaspekte).

Die nachfolgenden Abschnitte untersuchen diese Trends näher. Ausschlaggebend für diese Trends sind die Geschäftsziele und -anforderungen der Software-erstellenden Organisationen. Sie bestimmen beispielsweise die steigenden Produkthanforderungen sowie die Erwartungen an die Entwicklungsproduktivität. Die Geschäftsanforderungen geben auch die Richtung für technische Innovationen vor. Die Innovationen schaffen dann neue Anwendungsfelder der Software und induzieren neue Geschäftsanforderungen.

2.1 Trends rund um die Software-Systeme

Dimension der Systeme: Der Scope der Systeme, die man entwickelt und betreibt, hat sich stetig ausgeweitet. In der Anfangszeit der Informatik waren Software-Systeme für einen einzelnen Rechner konzipiert. Später bestand ein System aus vernetzten Teilsystemen. Heute entwickelt man Netze aus vernetzten Teilsystemen. Früher widmete sich der IT-Betrieb dem Netzwerk einer Unternehmensabteilung. Heute geht es mitunter um weltumspannende Anwendungslandschaften.

Größe der Systeme: Innerhalb einer Klasse von Systemen wird die Software stetig umfangreicher. Zum Beispiel der Code eines PC-Betriebssystems ist heute um viele Zehnerpotenzen größer als zu den Anfängen der Personalcomputer.

Fachliche und funktionale Komplexität: Die Systeme sind nicht nur aufgrund besserer Bedienbarkeit und bunterer Bildschirme gewachsen. Fast überall, insbesondere bei Unternehmensapplikationen, wächst auch stetig die Komplexität der darin verborgenen Abläufe und Daten. Kurz: Die Systeme können immer mehr.

Technische Komplexität: Es gibt immer mehr Möglichkeiten, mit denen Systeme entwickelt werden können. So ist die Vielfalt der technischen Konzepte, die einem gewöhnlichen Anwendungssystem zugrunde liegen, überwältigend - beispielsweise die in modernen Programmierframeworks wie Java Enterprise und .NET enthaltenen Mechanismen und vordefinierten Klassen. Die Software-Industrialisierung treibt die Entwicklung in diesem Bereich immer weiter voran.

Qualitätsvorgaben: Die Qualitätsvorgaben an die Systeme, die sogenannten *Nichtfunktionalen Anforderungen*, sind zunehmend anspruchsvoller geworden. Standen in der Anfangszeit der Datenverarbeitung Benutzbarkeits- und Performance-Anforderungen noch hinter den funktionalen Eigenschaften der Systeme zurück, kann ein modernes Internetsystem oder eingebettete Software darauf nicht mehr verzichten. Weitere Beispiele sind Sicherheits- und Wartbarkeitsvorgaben. Manche dieser Anforderungen gehen direkt auf die Software-Industrialisierung zurück (z. B. Wartbarkeit und Portabilität). Andere sind eine Folge der durch die Industrialisierung erreichten weiten Verbreitung von Software-basierten Systemen und den daraus entstandenen Erwartungen der Benutzer und Anwender.

Standardisierung: Produktstandards - zum Beispiel Branchenarchitekturen oder domänenspezifische Modellierungssprachen - sind eine Reaktion auf die überwältigende Konzeptvielfalt. Zugleich erhöhen sie in anderer Hinsicht die Komplexität: Man muss die Standards erlernen, stringent anwenden und ihre Einhaltung überprüfen (siehe unten: Trends rund um System-Erstellung und Betrieb).

Vernetzung und Integration: Vernetzung ist bereits oben als Triebfeder für die immer komplexer werdenden Größendimensionen angesprochen worden. Als grundlegendes Phänomen verdient sie eine Einzelnennung. Sie ist konzeptuell eng verknüpft mit den Bestrebungen zur Integration von Systemen.

2.2 Trends rund um System-Erstellung und -Betrieb

Größe der Entwicklungs- und Betriebsorganisation: Mit der Größe der entwickelten Systeme wächst meist auch die Größe der Organisationen, die sie entwickeln und betreiben. Ein weiterer Grund für das Wachstum der Organisationen ist die Konsolidierung über Unternehmenseinheiten hinweg: Hatten früher viele Bereiche ihre eigene Software-Entwicklung und IT, so gibt es heute nur noch eine große unternehmensweite Zentralabteilung. Waren Entwicklungs- und IT-Abteilungen früher regional oder global verteilt, konzentrieren sich die Aufgaben heute in wenigen großen Zentren, z. B. am Unternehmenssitz oder in Ländern mit besonders günstiger Kostenstruktur.

Komplexität der Entwicklungs- und Betriebsaufgaben: Die Aufgaben, die Entwicklung und Betrieb zu erfüllen haben, werden immer komplexer. Ein Grund dafür ist die steigende technische Komplexität der Produkte.

Produktbezogene Entwicklungstechnologien: Es entstehen ständig neue und immer komplexer werdende Konzepte zur Systementwicklung, die mit dem Aufbau der entwickelten Produkte verbunden sind, und mit denen sich die Software-Entwicklung auseinandersetzen muss. Klassische Beispiele sind komponentenbasierte Verfahren und objektorientierte Entwicklung. Aktuelle Beispiele sind modellgetriebene Entwicklung und Aspekt-orientierte Programmierung.

Prozessbezogene Entwicklungstechnologien: Viele technologische Konzepte spielen sich auf der Prozessebene ab. Sie streben an, die Vorgehensweisen bei der Entwicklung zu verbessern und ihr Industrialisierungsniveau zu erhöhen. Beispiele sind Agile Methoden und Produktlinien-Entwicklung.

Produktstandardisierung: Produktstandardisierung wirkt sich, wie oben bereits erwähnt, auch auf die Vorgehensweise aus. Manchmal fordern sie bestimmte Vorgehensweisen mit dem Ziel, spezielle Produkteigenschaften zu gewährleisten. In jedem Fall muss der Prozess Sorge dafür tragen, dass ein Produktstandard in der Entwicklung berücksichtigt und umgesetzt worden ist.

Prozessstandardisierung: Viele Prozessstandards zielen explizit auf die Industrialisierung von Software-Entwicklung und IT-Betrieb ab. Beispiele sind CMMI und ITIL V3.

Zeit- und Kostenvorgaben, Produktivität: Eine wesentliche Triebfeder der Software-Industrialisierung ist die Einhaltung von Zeit- und Kostenvorgaben, die immer enger werden und auf kontinuierliche Produktivitätssteigerung ausgerichtet sind. Dies kann oft nur durch Maßnahmen im Bereich des Vorgehens und der Organisation von Software-Entwicklung und IT-Betrieb erreicht werden.

Nachweisführung und „Compliance“: Vorgaben wie ISO 9000 in der Entwicklung von Gütern, FDA-Vorgaben im Gesundheitswesen oder Sarbanes-Oxley im Finanzbereich fordern, dass die betriebsunterstützenden oder leistungserbringenden IT-Systeme spezielle Nachweise führen. Sie schreiben Verfahren vor, mit denen die Einhaltung der Vorgaben überwacht werden („Compliance“). Dies sind stark ausgeformte Verfahren zur Qualitätssicherung im Zuge der Software-Industrialisierung, die sich stark auf die Prozesse in Software-Entwicklung und Betrieb auswirken können.

Verteilte Entwicklung und Outsourcing: Im Rahmen der Kostenreduktion und der (globalen) Arbeitsteilung fassen verteilte und ausgelagerte Leistungserbringung in Software-Entwicklung und IT-Betrieb immer stärker Fuß.

3 Herausforderungen für das Testmanagement

Wie wirken sich die Industrialisierungsentwicklungen speziell auf das Testmanagement aus? Welche Herausforderungen bringen sie mit sich? Dieses Kapitel listet diese Bereiche und Fragestellungen im Testmanagement auf. Grundsätzlich wirken die Industrialisierungstrends in zweierlei Hinsicht:

- Auswirkungen auf die zu testenden Systeme sowie auf das Entwicklungs- und Betriebsumfeld des Testens; dadurch wirken sie sich indirekt auf das Testen aus.
- Auswirkungen auf die Testtechniken und Testmanagement-Aktivitäten direkt.

Zur ersten Gruppe zählen vorwiegend die Steigerungen von Größe und Komplexität der Systeme sowie der Entwicklungs- und Betriebs-Organisationen, technologische Entwicklungen (Produkttechnologien), Standardisierungsentwicklungen und Compliance-Bestrebungen.

Die folgenden Unterkapitel beschreiben die verschiedenen Auswirkungen und Herausforderungen für das Testmanagement.

Testdokumente und -informationen: Ein Kernbereich des systematischen Testens ist die explizite Definition von Testkonzepten, Teststrategien und Testfällen, also wichtiger Testdokumente und -informationen. Im Zuge der Software-Industrialisierung steigt die Komplexität der zu testenden Produkte, so dass beim Testen eine Vielzahl von Aspekten berücksichtigt werden müssen, die häufig untereinander interagieren. Dies führt zu besonderen Herausforderungen für:

- Größe und Umfang von Teststrategien, Testfall-Repositories und Testplänen;
- Aufwand zur Testvorbereitung;
- Strukturierung von Testfall-Repository;
- Verwaltung von Testdaten.

Integration von Testarten und Teststufen: Im Rahmen der Komplexitätssteigerung von Software-Systemen steigt der Bedarf an unterschiedlichen Testarten. Traditionell waren funktionale Tests sowie Last- und Performance-Tests erforderlich. Für interaktive Benutzerschnittstellen benötigt man heute oft auch Usability-Tests. Web-Anwendungen wecken den Bedarf für das Security-Testing von Web-Applikationen.

Auch die Integration der Teststufen (typischerweise Modultest, Integrationstest, Systemtest und Abnahmetest) schreitet voran. Sie wird getrieben von Bestrebungen zur Steigerung der Testproduktivität. Das Ziel ist, redundante Tätigkeiten zu vermeiden und Synergien zwischen den Teststufen zu nutzen. Ein Beispiel hierfür ist die Verwendung einer einheitlichen Testfallbasis für System- und Abnahmetests, sowie das teilweise Ersetzen von Abnahmetests durch den Review von Systemtest-Ergebnissen.

Die wesentlichen Herausforderungen bestehen hier darin, die Testvorgehensweise, die Testkompetenzen im Team und die Testinfrastruktur so auszugestalten, dass die angestrebte Integration ohne Einbußen der Testeffektivität gelingt.

Testen mehrstufiger Architekturen: Produktlinien-Entwicklung und andere Verfahren der Komponenten- oder Plattform-basierten Entwicklung erfordern meist ein mindestens zweistufiges Testverfahren. Zunächst muss die Plattform getestet werden, später die auf der Plattform aufsetzenden oder von ihr abgeleiteten Systemvarianten. Beide Teststufen besitzen inhaltliche Bezüge (z. B. teils die gleichen Testfälle) während redundante Tests vermieden werden sollen. Ferner ist es schwierig, für die Plattform relevante Testfälle zu identifizieren, da die Einsatzszenarien - und somit die konkreten Testfälle - erst für die abgeleiteten Produktvarianten konkret bestimmt werden können.

Testen bei verteilter Entwicklung: Der Trend zur Arbeitsteilung und Auslagerung von Aufgaben macht auch vor dem Testen nicht halt. Er fordert das Testmanagement insbesondere in dreierlei Hinsicht: (1) Testen und Abnahme von Zulieferungen, (2) Outsourcing des Testens und (3) Koordination verteilter Testaktivitäten. Alle diese Bereiche sind aufwändiger und anspruchsvoller umzusetzen als die traditionelle Entwicklung als Teil eines Entwicklungsprojektes, das zusammenhängend und an einem Ort durchgeführt wird.

Testen bei Agiler Entwicklung: Agile Software-Entwicklung weist dem Testen eine besonders zentrale Rolle in der Software-Entwicklung zu (z. B. Erstellung von Testfällen vor der Entwicklung, Testfälle als teilweiser Ersatz von Anforderungen). Dadurch sind neue Testtechnologien entwickelt worden, insbesondere für den Unit-Test. Eine besondere Herausforderung bietet die Einbindung komplexer System- und Abnahmetests in die kurzen Entwicklungszyklen der Agilen Entwicklung. Wenn diese Tests umfangreiche Vorbereitungen erfordern, können sie nur schwierig mit dem hohen Änderungstempo der Iterationen in Einklang gehalten werden.

Testcenter: Wenn speziell das Testen ausgelagert wird und an einem anderen Ort oder in einer anderen Organisation erfolgt als die Entwicklung, hat man einen Spezialfall des verteilten Testens vor sich, ein sogenanntes *Testcenter*. Anders als beim Testen bei verteilter Entwicklung (siehe oben) erwartet man von dem Ergebnis des Testcenters nicht, dass man es noch näher prüfen muss. Vielmehr liefert das Testcenter die (weitgehend) verbindliche Qualitätsaussage über das getestete System.

Testkoordination: Mit der Größe der Testvorhaben und der Heterogenität der Testorganisation steigt auch der Koordinationsbedarf. In großen verteilten Projekten, mitunter unter Einbeziehung verschiedener Auftragnehmer und Zulieferer, sind die folgenden Koordinationsaspekte besonders anspruchsvoll: (1) Statusermittlung und Statusdarstellung, (2) Aktivitätsplanung, (3) Aufwandsschätzung sowie (4) Auswahl und Implementierung von Metriken für das Monitoring.

Aufbau und Unterhalt komplexer Testinfrastrukturen: Mit der Komplexität der Systeme steigt auch die Komplexität der erforderlichen Testinfrastrukturen. Beispielsweise erfordern vernetzte oder eventgesteuerte Systeme häufig spezielle Simulatoren als Teil der Testumgebung. Ansonsten können zumindest einzelne Teilsysteme vor Fertigstellung des Gesamtsystems nicht sinnvoll getestet werden. Die Herausforderungen bestehen insbesondere darin, dass die Erstellung geeigneter Testinfrastrukturen rares Expertenwissen erfordern und sehr aufwändig sein kann.

Etablierung, Konsolidierung und Unterhalt von Testmanagement-Werkzeugen: Software-Industrialisierung fordert Automatisierung und Werkzeugunterstützung. Aber viele Werkzeug-Lösungen für das Testmanagement weisen noch Lücken auf. Es ist oft auch schwierig, Tools, die sich inhaltlich ergänzen, technisch wirklich gut zu integrieren. Dadurch entsteht die Herausforderung, einigen Aufwand speziell in die Erstellung und Pflege der Testmanagement-Werkzeuge zu stecken. Ein Teilbereich davon ist die Integration von Defect-, Change- und Requirements-Management.

Einsatz von Testautomatisierung: In engem Zusammenhang mit den Testinfrastrukturen stehen die Herausforderungen für die Testautomatisierung - so sind Simulatoren eine spezielle Form der Testautomatisierung. Außerdem ruft der Bedarf für effizientere Tests nach einem Ausbau der Testautomatisierung. Zugleich erschweren manche neuen Produkttechnologien die Testautomatisierung. Ein Beispiel ist das automatisierte Testen von Web-Oberflächen, das durch generative Verfahren zur dynamischen Erstellung von Webseiten erschwert wird, da sich zwischen Testfallaufzeichnung und -ausführung allein durch Auswirkungen der Generierung die Seiten unterscheiden können und die Tests fehl schlagen, auch wenn sich am eigentlichen Inhalt der Seiten nichts geändert hat.

Aufbau und Pflege von Kompetenzbereichen: Viele der genannten Herausforderungen bringen eine weitere Herausforderung mit sich: Man benötigt kompetente Personen, die die anstehenden Aufgaben erfüllen und die nötige Qualifikation aufweisen. Zunächst müssen die erforderlichen Kompetenzen überhaupt in ausreichendem Maß vorhanden sein. Dann müssen sie auch effektiv miteinander integriert werden. Denn komplexe Aufgaben rufen nach dem gemeinsamen, gut abgestimmten Vorgehen mehrerer Personen. Schließlich muss das Testen attraktive Karrierepfade und Entwicklungsmöglichkeiten für hochqualifizierte Mitarbeiter bieten.

Prozessintegration über das Testen hinaus: Das Testen steht in Wechselwirkung mit vielen anderen Aufgaben der Software-Entwicklung und des IT-Betriebs. Exemplarisch seien das Requirements-Management und das Change- und Defect-Management genannt. Mit der Software-Industrialisierung wird es zunehmend wichtiger, die Schnittstellen zwischen diesen Aktivitäten effizient zu gestalten. Neben die Herausforderungen, das Testen selbst gut auszugestalten, tritt die neue Herausforderung, das Testen auch effizient mit den anderen Aufgabenbereichen zu verzahnen.

Effizientes Testen: Im Rahmen der allgemeinen Produktivitätssteigerungen muss auch das Testen seinen Beitrag liefern. Besondere Herausforderungen im Testmanagement sind dabei: (1) Frühzeitiger Beginn der Testvorbereitungen, (2) Priorisierung von Testfällen, (3) Planung des Testens, (4) Prozesssteuerung der Testaktivitäten und (5) Kontinuierliche Verbesserung des Testens.

4 Maßnahmen des Testmanagements

Welche Maßnahmen ergreift das Testmanagement, um auf die Herausforderungen der Software-Industrialisierung zu reagieren? Dieses Kapitel stellt aktuelle methodische und technologische Entwicklungen im Testmanagement vor und setzt sie in Beziehung zur Software-Industrialisierung. Die Fallbeispiele in Kapitel 5 illustrieren die Maßnahmen näher und belegen ihre Relevanz für die Software- und IT-Praxis. Die abschließende Diskussion in Kapitel 6 untersucht, inwiefern die Maßnahmen im Testmanagement beispielhaft sind für vergleichbare Maßnahmen im Software-Management allgemein, um die Herausforderungen der Software-Industrialisierung in den Griff zu bekommen.

Die Maßnahmen des Testmanagements lassen sich in fünf Gruppen einteilen, die im folgenden näher beschrieben sind: (1) Spezialisierte und zentralisierte Organisationseinheiten, (2) Integration und Anreicherung von Aufgabenbereichen, (3) verteilte Organisation und Outsourcing, (4) Ausbau von Prozesssteuerung und Prozessverbesserung sowie (5) umfassende Werkzeugunterstützung.

4.1. Maßnahme 1: Spezialisierte und zentralisierte Organisationseinheiten

Traditionell war das Testen Aufgabe eines jeden einzelnen Entwicklungsprojektes. Vereinzelt wurden bestimmte Testarten auch von Spezialisten der IT durchgeführt, insbesondere Last- und Performance-Tests. Doch dies war eher eine Nebenaufgabe der IT als dass dafür eine eigene Organisationseinheit eingerichtet worden wäre.

Seit einiger Zeit etablieren immer mehr Unternehmen spezialisierte Testcenter, die ihre Dienstleistungen und Fachexpertise rund um das Testen als interne Auftragnehmer anbieten. Das geht darauf zurück, dass zum Einen das Bewusstsein für die Wichtigkeit des Testens für einen durchgängigen und systematischen Entwicklungs- und Betriebsprozess steigt. Zum Anderen lassen unternehmensweit standardisierte Anwendungsarchitekturen und der hohe Vernetzungsgrad der Applikationen gar keine andere Wahl, als auch die Tests zentral zu koordinieren und durchzuführen.

Beides sind Phänomene der Software-Industrialisierung. Beispiele für zentralisierte Organisationseinheiten im Testmanagement sind:

- Testcenter als unternehmensinterne Dienstleister für Testen und Testkoordination, insbesondere bei funktionalen Systemtests und Abnahmetests;
- Spezialisierte Teams für Testautomatisierung und Verwaltung von Regressionstest-Infrastrukturen;
- Spezialisierte Teams für Last- und Performance-Tests als Teile der Qualitätsmanagement- oder IT-Betriebsorganisationen;
- Usability-Beauftragte und -Gruppen, die Usability-Tests als zentrale Dienstleistungen;
- Zentrales Kompetenzzentrum für Sicherheitstests von Web-Applikationen.

4.2 Maßnahme 2: Integration und Anreicherung von Aufgabenbereichen

Ursprünglich wurde Testen in vielen Entwicklungsprojekten stiefmütterlich als lästiges Übel betrachtet, dem man sich gegen Ende eines Projektes mehr oder weniger widmet, je nach dem wie viel Zeit und Budget noch vorhanden ist.

Inzwischen haben viele Entwicklungsorganisationen gelernt, dass systematisches Testen einen hohen Beitrag für den Projekterfolg leistet. Entsprechend müssen Testvorbereitungen früh im Entwicklungsprozess beginnen, das Testen muss mit anderen Entwicklungsaktivitäten eng verzahnt werden (Stichworte: Anforderungsbasiertes Testen, Risikobasiertes Testen, Issue- und Fehlermanagement) und die Inhalte der Testaktivitäten werden umfangreicher (z. B. neue Testarten adressieren wie das Security-Testing von Web-Applikationen). All dies ist eng verknüpft mit dem Trend zur Software-Industrialisierung - insbesondere mit Prozessstandardisierung, gestiegenen Qualitäts- und Sicherheitsanforderungen bzw. Erwartungen der Auftraggeber und Benutzer sowie die höhere Komplexität der Systeme, die zum Beispiel neue Sicherheitslücken zur Folge hat.

4.3 Maßnahme 3: Verteilte Organisation und Outsourcing

Früher wurden Entwicklungsprojekte - somit auch die Tests - vorwiegend von einem Team an einem einzigen Ort durchgeführt. Inzwischen haben die meisten Unternehmen eine differenzierte Organisationsstruktur für Software-Entwicklung, IT-Betrieb und somit auch für das Testen etabliert. Die Gründe können wiederum klar auf die Software-Industrialisierung zurück geführt werden: Aufgabenstandardisierung und Arbeitsteilung sowie Kostenreduzierung.

Die Ausprägungen dieser Aufgabenverteilung im Testen sind:

- Spezialisierte Testdienstleister vor Ort, im Entwicklungsteam oder als separates Testteam;
- Spezialisierte zentrale Testcenter an einem anderen Standort des Unternehmens als dem, in dem das Entwicklungsprojekt angesiedelt ist;
- Testlabs (intern oder extern) in Ländern mit niedrigerer Kostenstruktur (Offshore-Outsourcing).

4.4 Maßnahme 4: Ausbau von Prozesssteuerung und Prozessverbesserung

Testen wird auch vielfach heute noch lediglich als eine Phase der Software-Entwicklung betrachtet und dementsprechend im Rahmen der Gesamtprojektmanagements geplant und gesteuert. Die Spezialisierung (Maßnahme 1), Integration und Anreicherung (Maßnahme 2) und vielmehr noch die verteilte Organisation und das Outsourcing im Testbereich (Maßnahme 3) erfordern aber eine sehr ausgeprägte Prozesssteuerung speziell für das Testen. Deshalb tun Software- und IT-Organisationen gut daran, Testen als ein eigenes (Teil-)Projekt im Rahmen der Software-Entwicklung zu behandeln und mit einer eigenen Management-Struktur auszustatten. Diese muss freilich eng mit dem Gesamtprojektmanagement verzahnt sein.

Da die umfangreichen und mitunter hoch komplexen Testprozesse für viele Organisationen noch neu sind, und teils die methodischen Grundlagen noch nicht genügend ausgereift sind, muss die Testprozesssteuerung um ein besonderes Augenmerk auf die kontinuierliche Prozessverbesserung im Testen ergänzt werden.

Der Ausbau von Prozesssteuerung und Prozessverbesserung im Testen sind indirekte Folgen der Software-Industrialisierung, wobei die Prozessverbesserung selber von dem erhöhten Bedarf nach Prozesssteuerung getragen wird.

Ausprägungen und Belege für diese Entwicklung hin zu stärkerer Prozesssteuerung und Prozessverbesserung sind:

- Testkoordination als ein zentraler Leistungsbaustein im Angebot der unternehmensinternen Testcenter;
- Partner- und Lieferantenmanagement für Testdienstleistungen (bei Outsourcing und/oder Offshore-Leistungen);
- Methoden, Beratungs- und Schulungsangebote für Testqualifikation und Testprozessverbesserung (z. B. TMap Next, TPI, ISTQB Certified Tester).

4.5 Maßnahme 5: Umfassende Werkzeugunterstützung

Nach wie vor sind Textverarbeitungsprogramme und Tabellenkalkulationen die am meisten verbreiteten Software-Werkzeuge zur Unterstützung der Testphase: Textdokumente definieren Testkonzepte, Teststrategien und Testfälle. Tabellenkalkulationen dienen ebenfalls der Testfalldefinition. Außerdem werden sie zur Planung und Verfolgung der Testdurchführung verwendet.

Dennoch ist das Testen derjenige Bereich in der Software-Entwicklung (nach den unabdingbaren Entwicklungsumgebungen für die Implementierung), für den spezialisierte Werkzeugunterstützung am stärksten verbreitet ist: Große IT-Projekte kommen schon lange nicht mehr ohne Testmanagement-Tools aus, Fehlermanagement ebenso wie Load- und Performancetests sind ohne spezielle Tools kaum vorstellbar, und Testautomatisierung für User-Interface-Tests ist weithin etabliert. Bei Prozess- und Methodenfragen wird in kaum einem anderen Bereich so schnell und konkret die Frage nach Werkzeugunterstützung gestellt.

Auch die Anzahl kommerziell und öffentlich verfügbarer Werkzeuge für die unterschiedlichen Teilaufgaben des Testens ist höher als die der Tools für Requirements-Management, Modellierung und Projektmanagement. Ferner zeigen die Konsolidierungsentwicklungen auf dem Werkzeugmarkt - insbesondere der Trend hin zur übergreifenden Unterstützung des Application Lifecycle Managements - die große Bedeutung, die der durchgängigen Toolunterstützung beigemessen wird.

In Hinblick auf den Einfluss der Software-Industrialisierung rührt der Bedarf für eine umfassende Werkzeugunterstützung einerseits direkt her von der erhöhten Komplexität der Produkte und Abläufe, von den Standardisierungsbestrebungen für Produkttechnologien und Abläufe sowie von den Bestrebungen zur Kostenoptimierung und Produktivitätssteigerung.

Andererseits verstärken die anderen beschriebenen Maßnahmen den Bedarf für umfassende Tool-Lösungen: Insbesondere die Integration und Anreicherung von Aufgabenbereichen (Maßnahme 2) schafft klare Anforderungsprofile für den Werkzeugeinsatz. Der Ausbau der Prozesssteuerung (Maßnahme 4) erfordert zentrale Datenhaltung, die Toolgestützte Konsolidierung von Statusinformationen und effiziente Reporting-Funktionen.

5 Fallbeispiele

Dieses Kapitel stellt drei Fallbeispiele für die Testpraxis in Unternehmen vor, die die oben vorgestellten Maßnahmen illustrieren. Der Autor hat die Fallbeispiele im Rahmen seiner Tätigkeiten in der Software-Industrie kennen gelernt. Sie belegen auch die Zusammenhänge zwischen den Herausforderungen der Software-Industrialisierung und den Maßnahmen. Jedes Unterkapitel beschreibt zunächst den jeweiligen Fall und erläutert anschließend die Zusammenhänge zwischen den Herausforderungen und den Maßnahmen.

5.1 Fallbeispiel 1

Ein Finanzdienstleister mit einem starken Anteil von Internet-basiertem Geschäft (u. a. Online-Banking) verfügt über eine durchgängige Applikationslandschaft aus IT-Anwendungen, die stark miteinander interagieren und von der zentralen IT eng aufeinander abgestimmt worden sind. Dies ist insbesondere deshalb erforderlich, um eine hohe Verfügbarkeit der Anwendungen zu gewährleisten - eine wichtige und unternehmenskritische Anforderung an die Systeme. Auftraggeber und Eigner der Anwendungen sind die Fachabteilungen.

Traditionell sind die Fachabteilungen für die Applikationsgestaltung zuständig und somit auch für die Abnahmetests der Anwendungen. Mit der durchgängigen Applikationslandschaft und aufgrund der hohen Verfügbarkeitsanforderungen hat das Unternehmen jedoch die Zuständigkeit für umfangreiche Last- und Performance-Tests bei der zentralen IT-Qualitätsabteilung angesiedelt. Der Erfolg dieser Last- und Performance-Tests ist eine wesentliche Voraussetzung für die Freigabe eines neuen Releases der Applikationslandschaft.

In diesen Last- und Performance-Tests hat sich die IT-Qualitätsabteilung einen sehr guten Ruf erarbeitet und wird im Unternehmen als wichtiger Garant für die Qualität der Applikationen und den darauf aufsetzenden Geschäftsprozessen betrachtet. Aufgrund dieser positiven Erfahrungen hat die IT-Qualitätsabteilung auch die Aufgabe erhalten, die funktionalen Tests der Fachabteilungen zentral zu verwalten und schrittweise zu automatisieren. Sie wird dadurch zum umfassenden Test-Dienstleister im Unternehmen.

Das Beispiel belegt, wie die Industrialisierungstendenzen Technologie- und Architekturstandardisierung sowie hohe Qualitätsanforderungen den Bedarf für fortgeschrittene Testpraktiken verstärken. In der Folge sind die Organisationseinheiten zentralisiert und mit besonderen Aufgaben versehen worden (Maßnahme 1), der Aufgabenbereich der IT-Qualitätsabteilung ist schrittweise ausgeweitet worden (Maßnahme 2) und eine umfassende Werkzeugunterstützung (Maßnahme 3) ist eingeführt worden.

5.2 Fallbeispiel 2

Ein großer Hersteller von Industriegütern mit mehreren weitgehend eigenständigen Unternehmensteilen unterhält eine zentrale IT-Abteilung. Die IT-Abteilung hat eine einheitliche IT-Anwendungslandschaft definiert, deren Kern ein gemeinsames Applikations-Framework ist. Das Applikations-Framework wird in allen Unternehmensteilen eingesetzt, wobei jeder Unternehmensteil spezifische Anpassungen (Änderungen von Funktionen in Teilen des Applikations-Frameworks) und Erweiterungen (zusätzliche Applikationen) besitzt. Für jeden Unternehmensteil betreibt die zentrale IT eine eigene Instanz des Applikations-Frameworks.

Vor der Einführung des zentralen Applikations-Frameworks hat jeder Unternehmensteil seine eigenen IT-Anwendungen definiert, beauftragt und getestet (Abnahmetest). Durch das zentrale, von der IT verwaltete Applikations-Framework haben sich die Anforderungen an das Testen maßgeblich geändert: Die IT muss zentral Systemtests für die übergreifenden Anforderungen aller Unternehmensteile durchführen. Die anschließenden Abnahmetests der Unternehmensteile überlappen sich sowohl untereinander als auch mit den Systemtests der IT. Daher besteht Bedarf, die Tests aufeinander abzustimmen und Möglichkeiten zur Wiederverwendung von Testfällen zu nutzen.

Die IT des Unternehmens hat ein Testcenter eingerichtet, um die Tests zu koordinieren und zu unterstützen. Es tritt auf als interner Dienstleister, der von den Unternehmensteilen (Eigner der Applikationen) beauftragt werden kann, aber nicht einbezogen werden muss.

Zu den wesentlichen Angeboten des Testcenters gehört die Bereitstellung von Testmanagern als Experten zur Koordination der Tests sowie eine zentrale Werkzeugplattform für das Testen (v. a. Testfall-Repository, Testplanung, manuelle Testdurchführung, Fehlermanagement und Status-Reporting). Das Testcenter hat sich innerhalb kurzer Zeit als Testdienstleister etabliert und unterstützt die großen Testvorhaben im Umfeld des zentralen Applikations-Frameworks.

In diesem Fall wirken verschiedene Industrialisierungstendenzen: Komplexe und zugleich standardisierte IT-Technologien und -Architekturen (z. B. komponentenorientierte Produkttechnologie im Applikations-Framework mit Zügen der Produktlinien-Entwicklung), hohe Komplexität der Software-Entwicklung (zweistufige Entwicklung mit Applikations-Framework und Anpassungen) sowie zunehmend engere Zeit- und Kostenvorgaben.

Das Beispiel veranschaulicht gut, wie eine spezialisierte, zentrale Organisationseinheit (Maßnahme 1) das Testen in einer verteilten Organisation (Maßnahme 3) koordiniert. Ein wichtiger Baustein ist der Ausbau der Prozesssteuerung (Maßnahme 4) in Form von Testmanagement-Experten, Fehlermanagement und Status-Reporting. Das Testcenter kann diese Leistungen so nur dank einer fortgeschrittenen Werkzeugunterstützung erbringen (Maßnahme 5).

5.3 Fallbeispiel 3

Ein Industrieunternehmen entwickelt und pflegt die Software-Systeme für seine Fertigungssteuerung unter Einbeziehung mehrerer Zulieferer, teils mit Offshore-Anteilen. Die Zulieferer führen Teile der Systemtests für ihre entwickelten Systemteile auf einer zentralen Testumgebung in Eigenregie durch. Der Auftraggeber - das Industrieunternehmen - koordiniert die Tests und überprüft die Testergebnisse.

Um diese verteilten Tests effizient abzuwickeln, hat das Unternehmen die verpflichtenden Testprozesse der Zulieferer detailliert definiert, durch eine einheitliche Werkzeugumgebung unterstützt und die Rolle eines Testkoordinators geschaffen. Die Werkzeugunterstützung umfasst insbesondere das Fehlermanagement. Dazu mussten Werkzeugbrücken zwischen dem System des Auftraggebers und den Lösungen der Zulieferer geschaffen werden.

In diesem Beispiel tritt insbesondere die Industrialisierungstendenz der komplexen Entwicklungsprozesse (samt Organisation) zu Tage. Es belegt das Zusammenwirken von zentraler Testkoordination (Maßnahme 1) und Prozesssteuerung (Maßnahme 4) in einer verteilten Organisation mit Outsourcing (Maßnahme 3).

6 Zusammenfassung und Diskussion

Dieser Artikel stellt die Facetten der Software-Industrialisierung dar und leitet daraus Herausforderungen für das Testmanagement ab. Auf dieser Grundlage identifiziert er fünf Maßnahmen, mit denen das Testmanagement auf die Herausforderungen reagiert. Die Maßnahmen stellen aktuelle Schwerpunkte des Testmanagements dar, für die weitere methodische Unterstützung besonders benötigt wird. Drei Fallbeispiele illustrieren die Maßnahmen und belegen ihre Relevanz.

Die identifizierten Maßnahmen des Testmanagements haben Entsprechungen im Software-Management allgemein: Dort besteht ebenso Bedarf für spezialisierte und zentralisierte Organisationseinheiten wie Project Management Offices, wenn auch die Aufgabendifferenzierung nicht so ausgeprägt ist wie im Testmanagement. Integration und Anreicherung von Aufgabenbereichen findet sich im Software-Management, zum Beispiel bei der Anwendung spezialisierter Projektmanagement-Methoden. Die verteilte Entwicklung wirkt auf das Software-Management insgesamt. Gleiches gilt für Prozesssteuerung und Werkzeugunterstützung.

Die identifizierten Maßnahmen zeigen, wie das Testmanagement auf die Herausforderungen der Software-Industrialisierung reagieren kann. Empirische Studien sollten auf den beschriebenen Faktoren aufsetzen und die Effektivität der Testmanagement-Maßnahmen näher untersuchen. Bislang gibt es hierzu kaum gesicherte Erkenntnisse.

Literaturverzeichnis

- [BBT05] Bode, A; Broy, M; Taubner, D;: "Software-Industrialisierung," Informatik-Spektrum, vol. 28, Aug. 2005, S. 269.
- [CN02] Clements, P.; Northrop, L. M.: Software Product Lines: Practices and Patterns. Addison Wesley, Upper Saddle River, NJ (2002)
- [Cu91] Cusumano, M.A: Japan's Software Factories, Oxford Univ Pr, 1991.
- [FS07] Fröschle, H.; Strahinger, S.: IT-Industrialisierung, Dpunkt Verlag, 2007.
- [GS04] Greenfield, J.; Short, K.: Software Factories, Wiley & Sons, 2004.
- [HSH06]Helferich, A.; Schmid, K.; Herzwurm, G.: "Product management for software product lines: an unsolved problem?," Commun. ACM, vol. 49, 2006, S. 66-67.
- [Wi] "IT-Industrialisierung – Wikipedia"; <http://de.wikipedia.org/wiki/IT-Industrialisierung>.
- [Me] Meyers Online Lexikon, <http://lexikon.meyers.de/meyers/Industrialisierung>.
- [OCG] OGC: ITIL V3"; <http://www.ogc.gov.uk/index.asp?id=2261>.
- [PMI] Project Management Institute, A Guide to the Project Management Body of Knowledge: PMBOK Guide, B&T, 2004.
- [Sp06] Spillner A. et al.: Praxiswissen Softwaretest - Testmanagement: Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB-Standard, Dpunkt Verlag, 2006.