

# System-Integrationstest

DI Dr. techn. Mohsen Ekssir-Monfared  
ANECON Software Design und Beratung  
[m.ekssir-monfared@anecon.com](mailto:m.ekssir-monfared@anecon.com)

## Kurzfassung:

Um die Funktionalität von mehreren Systemen, die im Verbund in Betrieb gehen, zu garantieren ist ein systematischer System-Integrationstest unabdingbar. Bei diesem Test muss das Zusammenspiel der Systeme unter einander sowie ihre Kommunikation nach Außen über die gemeinsame spezifizierte Schnittstelle erprobt und bestätigt werden. Für Unternehmen, die solche Verbundsysteme betreiben, ist der System-Integrationstest ein essentieller qualitätssichernder Faktor für die ordnungsgemäße Funktionalität ihrer IT und ein Garant für die Zuverlässigkeit, Sicherheit und Performanz ihrer Geschäftsprozesse.

Schlüsselbegriffe: *System-Integrationstest, gemeinsame Schnittstelle, Konzerttest, heterogenes Umfeld, Big Picture, Zwiebelschalen-Prinzip, Teilintegrations- und Gesamtintegrationstest*

## 1. System-Integration

Die Aufgaben, die heute in der IT-Welt zu lösen sind, verlangen integrierte Lösungsansätze. Keiner allein ist in der Lage die komplexe Aufgaben zu bewältigen. Die verschiedenen Geschäftsprozesse sind mit einander verwoben und greifen über Unternehmensgrenzen hinaus auf andere Systeme (siehe e-Business). Um bestimmte Anforderungen zu erfüllen werden immer öfter Funktionalität, bzw. Information, verschiedener Systeme zusammengeführt. Ein gutes Beispiel dafür sind Mashups.

Mashups [1], führen die Ergebnisse zwei oder mehr getrennter Systeme zusammen um ein zusammengestelltes Response auf ein Request zu bilden. Dadurch werden bestehende Webapplikationen kombiniert um eine neue dynamische Web-Applikation mit zusätzlichen Diensten zu schaffen. Dies ist ein „sichtbares“ Beispiel für Systemintegration. Sichtbar aus dem Grund, da der End-User nur das Endergebnis zu sehen bekommt aber nicht die zahlreichen Zwischenergebnisse. Die durch das Zusammenwirken und die innere Beziehungen der integrierten Systeme zustande kommen. In der Regel treten die Wechselwirkungen der integrierten Systeme im Hintergrund auf. Daher werden sie vom Endbenutzer gar nicht wahrgenommen.

Ein weiteres Beispiel sind die auf Service-orientierten Architektur (SOA) basierten Systeme. Diese bestehen ebenso aus mehreren kollaborierenden Teilsystemen. In diesem Zusammenhang spricht man in der SOA-Welt von Komponierung, Orchestrierung und Choreografie der Geschäftsprozesse [2], [3]. Die Responses zahlreicher Services werden gesammelt und zusammengeführt. In der Regel ist es für eine Integration solcher Systeme notwendig, ihre in Kommunikation tretenden Schnittstellen auf beiden Seiten anzupassen oder neue gemeinsame Schnittstellen zu entwickeln. Es werden Schichten von Web Service Schnittstellen gebildet um die Responses zu filtern und zu aggregieren.

Es versteht sich, dass die Störanfälligkeit solcher zusammengesetzter Systeme wesentlich höher als die von einfachen Standalone Systeme sind. Ein Problem in einem System eskaliert sich schnell zu einem Problem in allen betroffenen Systemen. Daher bedarf die Integration mehrerer Systeme einer höheren Qualität und Präzision. Dies zu sichern ist die Aufgabe des System-Integrationstests.

## 2. System-Integrationstest

Der System-Integrationstest soll dazu dienen Inkonsistenzen zwischen den verbundenen Systemen aufzudecken, Inkonsistenzen die erst ans Tageslicht treten wenn die Systeme erst zusammenwirken. Da die Entwickler eines bestimmten Subsystems oft nicht wissen, was die Entwickler eines anderen Subsystems annehmen, sind Missverständnisse aufgrund falscher Annahmen an der Tagesordnung. Jeder wird behaupten seine Ergebnisse sind korrekt aber in der Vereinigung mit den Ergebnissen der Anderen sind sie falsch. Deshalb ist das Risiko eines Fehlschlages beim „Scharfschalten“ der Systeme besonders groß. Kein Unternehmen kann es sich leisten, seine Systeme mit ihren vielen gegenseitigen Abhängigkeiten zueinander ohne einen System-Integrationstest in Betrieb zu nehmen. Bei diesem Test komme es darauf an, mehrere unabhängig von einander entwickelte Software-Systemen mit einer Vielzahl gemeinsamer Schnittstellen zu erproben. Dabei können auch unterschiedlicher Technologien ins Spiel kommen, z.B. verschiedener Webbrowser. Es gilt zu bestätigen, dass diese Technologien wirklich in allen Aspekten gegenseitig verträglich sind.

### 2.1 Einbindung des Tests in der frühen Phase der Produkterstellung

Es ist bekannt, dass die meisten Fehler, laut der Teststatistik über 40% [4], bei der Entwicklung eines neuen Produktes ihre Ursachen in der Spezifikation der Anforderungen haben. Deshalb muss darauf geachtet werden, dass die Anforderungen präzise und genau formuliert werden, so dass sie nicht unterschiedlich interpretiert werden können. Besonders die Systemschnittstellen müssen genau festgelegt werden, wenn möglich gleich formal mit Fallbeispielen.

Die Einbindung des Tests in die frühen Phasen der Entwicklung sorgt dafür, dass keine vagen und unscharfen Aussagen in das Lastenheft einfließen. Die Aufgabe des Testspezialisten in dieser Phase der Produktentwicklung ist es, die Formulierungen so zu gestalten, dass sie keinen Spielraum für Interpretationen lassen. Jede Aussage muss auf ihre Kontrollierbarkeit geprüft werden. Notfalls muss der Tester in Abstimmung mit dem Anwender die Anforderungen umformulieren. Bezüglich der Integration hat der Tester darauf zu achten, dass die Interaktionen zwischen den Systemen transparent sind. Die Interaktion der Systeme sollte möglichst mit Zustandsdiagrammen modelliert werden. D.h. hier wird empfohlen punktuell modellbasiert zu arbeiten.

Ein weiterer wichtiger Punkt ist, dass der Testspezialist auch bei der Schnittstellendefinitionen eine aktive Rolle spielen soll. Schnittstellendefinitionen in IDL oder WSDL sind einer Prüfung zu unterziehen ob sie auch konsistent und vollständig sind. Dies kann mit Hilfe einer automatisierter Schnittstellen-Audit geschehen.

### 3.2 Testdesign und -realisierung

Bei einem integrierten Verbundsystem ist es zunächst erforderlich, um eine bessere Einsicht in die Abhängigkeiten zu bekommen sowie einen Überblick über die Gesamtheit aller integrierten Systeme zu erhalten, ein Integrationsmodell, bzw. einen **System-Stadtplan**, zu erstellen. Dieses vermittelt ein Bild der Systemabhängigkeiten und ihrer gemeinsamen Schnittstellen und kann nach Bedarf mit unterschiedlichem Informationsgehalt dargestellt werden. Informationen wie die Art der Schnittstelle, unidirektionale oder bidirektionale, synchron oder asynchron Kommunikation usw. können in diesem Modell berücksichtigt werden. Der System-Stadtplan muss natürlich wie jedes lebendes Dokument während des Testprozesses aktualisiert und präzisiert werden.

### 3.2.1 Voraussetzungen für die Durchführung des Integrationstests

Um mit der Testrealisierung und -durchführung der System-Integrationstests beginnen zu können, müssen bereits einige Voraussetzungen erfüllt sein.

Die folgenden Punkte sind u. a. als **Eingangskriterien** für den Beginn der Testrealisierung und -durchführung zu sehen:

- Es müssen bis zu diesem Zeitpunkt alle technischen Dokumente, Schnittstellenbeschreibungen, Pflichtenhefte sowie Administrations-Handbücher fertig erstellt sein. Die Tester sollten bei dem Erstellungsprozess (und Reviewprozess) der Dokumentationen eingebunden sein.
- Für die einzelnen integrierten Systeme sollte der standalone Systemtest schon vor der Integration durchgeführt worden sein. Bei dem **Systemtest** der einzelnen Systeme werden u. a. auch dessen Schnittstellen getestet, allerdings nur aus der Sicht des einen Systems. Die Testprotokolle jedes einzelnen Systems sollten zur Verfügung stehen.
- Es muss sichergestellt werden, dass keine **Heterogenität** im Bezug auf die Qualität der gelieferten SW-Systeme besteht. Die Systeme, die zum Integrationstest vorgelegt werden, sollten alle in etwa den gleichen Qualitätsstand haben. Es ist daher sinnvoll gewisse **Annahmekriterien** für die Übernahme der Systeme seitens des Testteams, festzulegen.
- Zur Erreichung einer notwendigen „Eindringtiefe“ in die Kette der zusammenschalteten SW-Systeme, bzw. Testlinge, müssen bereits vor dem Beginn der System-Integrationstests geeignete **Testdaten** bereitgestellt werden. Bei der Erstellung der Testdaten muss dafür gesorgt werden, dass durch die Tests alle relevanten und kritischen Teile des Systemverbunds erreicht werden.

Die Erstellung der Testfälle sowie der Aufbau einer geeigneten Testumgebung (Test-Infrastruktur) sind u. a. die weiteren Bedingungen für den Beginn der Durchführung des System-Integrationstests.

### 3.2.2 Test-Vorgehen

Der System-Integrationstest beginnt mit der Integration zweier Systeme und reicht bis zur vollständigen Integration sämtlicher Geschäftsprozesse mit allen möglichen Geschäfts- und Testscenarien und aller beteiligten Systeme. Dieser Test muss oft für mehrere Testzyklen durchgeführt werden. Die System-Integrationstests bestehen hauptsächlich aus zwei großen Teststufen:

- Teilintegrationstests und
- Gesamtintegrationstests

#### 3.2.2.1 Teilintegrationstest

Hier ist zu klären, welche Systeme und deren Schnittstellen als erste getestet werden müssen. Dabei können verschiedene Faktoren, wie die zeitliche Verfügbarkeit der Systeme, eine wesentliche Rolle spielen. In Abstimmung mit den Fachbereich-Mitarbeitern wird festgelegt, welche Systeme samt ihrer Schnittstellen, integriert testbar sind. Dabei müssen die Systeme oder Funktionen genommen werden, die unabhängig von den anderen Systemen getriggert und getestet werden können. Diese **Teilintegrationen** werden, nach dem **Zwiebelschalenprinzip**, sukzessive um weitere Systeme bis zu ihrem vollen Umfang erweitert und iterativ getestet.

Ein großer Vorteil bei Anwendung dieses Ansatzes ist es, dass durch die vielen Iterationen die gefundenen Abweichungen leichter zu lokalisieren und somit zu beheben sind. Die Unterschiede zwischen Soll- und Ist-Zustand der Daten sind auch leichter zu erkennen, wenn der Abstand zwischen den Zuständen nicht zu groß ist.

Für die Teilintegrationstests müssen sowohl **funktionale** als **nicht-funktionale** Testfälle spezifiziert und durchgeführt werden. Dabei dürfen besonders die **Performancetests** nicht außer Acht gelassen werden um die Bottlenecks in der Kette der zusammenschalteten Systeme zu finden. Hier gilt, dass „eine Kette nur so stark ist wie ihr schwächstes Glied“. Nur durch ein End to End Test kann das Performanz eines Verbundsystems wirklich gemessen werden [5].

Für den Fall, dass alle Systeme nicht gleichzeitig zur Verfügung stehen und **(zeitliche) Abhängigkeiten** bestehen, müssen die Teilintegrationstests für jene Systeme geplant und durchgeführt werden, die früher verfügbar und testbar sind. Die Schnittstellen zu den fehlenden Systemen müssen durch Stubs simuliert werden.

### 3.2.2.2 Gesamtintegrationstest

Nach erfolgreichem Abschluss der Teilintegrationstests beginnt die Teststufe der **Gesamtintegrationstests**. Die Gesamtintegrationstests sind komplexer als die Teilintegrationstests und dürfen erst dann durchgeführt werden, wenn die Systeme und ihre Anbindungen eine gewisse Stabilität aufweisen. Daher müssen die in den Teilintegrationstests gefundenen Fehler mit höheren Prioritäten auf jeden Fall bereits vor dem Beginn der Gesamtintegrationstests behoben sein.

Beim Gesamtintegrationstest geht es um den „**Konzerttest**“ aller integrierten Systeme. Dabei wird das Zusammenspiel der Systeme miteinander sowie ihre angeforderte und spezifizierte Kommunikation über die gemeinsame Schnittstelle getestet. Beim Konzerttest wird überprüft, wie harmonisch die einzelnen Systeme zusammenspielen.

Für die Gesamtintegrationstests müssen - genauso wie bei den Teilintegrationstests - sowohl **funktionale** als auch **nicht-funktionale** Testfälle spezifiziert und durchgeführt werden. Hier spielen auch besonders die **Last- und Performancetests** sowie **Performancemessungen** eine große Rolle, um die Bottlenecks in der gesamten Kette der zusammenschalteten System-Landschaft zu finden [6]. Hier gilt das Prinzip, dass „eine Kette nur so stark ist wie ihr schwächstes Glied“.

## 4. Zusammenfassung

Der System-Integrationstest ist bei der Einführung eines Verbunds mehrerer SW-Systemen unerlässlich. Man darf den Aufwand dafür nicht unterschätzen. Dieser ist ein Vielfaches von dem was der Test aller einzelnen Systeme verursacht. Denn ein Verbundsystem ist mehr als nur die Summe aller Mitgliedssysteme. Es ist die Summe aller einzelnen Systeme, plus die Summe aller Beziehungen zwischen diesen Systemen, plus alle Zustände die durch die Zusammenwirkung der Systeme entstehen. Durch den System-Integrationstest wird eine höhere Stabilität, Zuverlässigkeit und Verfügbarkeit der spezifizierten Funktionalitäten über Systemgrenzen hinweg gewährleistet. System-Integrationstest führt zu geringeren Kosten in der Wartung und Weiterentwicklung und somit zu einer geringeren Amortisationszeit der Einzelprodukte.

### Literatur:

[1] Volker Hinzen, Mashup von Web-Daten wie News, Videos oder Fotos: Mehrwert fürs Web, Die Medien-Mischung macht's!, magnus.de, Technik. Trends. Entertainment, am 27.01.2010

- [2] Nicolai Josuttis, SOA in der Praxis, dpunkt .verlag, Heidelberg, 2008, S. 121-123
- [3] Stefan Riedl, Wie SOA und SaaS ERP-Software für den Mittelstand umkrempeln, IT-Business, Das Informationsportal für den IT-Markt, am 03.03.2008  
<http://www.it-business.de/news/produkte/software/unternehmenssoftware/articles/110652/>
- [4] Harry M. Sneed, Martin Hasitschka, und Maria-Theresia Teichmann, Software-Produktmanagement, dpunkt .verlag, Heidelberg, 2004, S. 358
- [5] Nguyen, Hung Q., Johnson, R., Hackett, M.: Testing Applications on the Web, Wiley Publishing, Indianapolis, 2003, S. 479
- [6] Sneed, H., Baumgartner, M., Seidl, R.: Der Systemtest, Hanser Verlag München, 2008, S. 153