

Architekturzentriertes Testen

Editorial



Warum noch eine Test-Methodik?

Testmethoden und Standards gibt es doch nun schon genug. Oft werden nicht mal die gängigen Standards im Testen eingehalten!

Warum sollten wir uns also mit noch einer neuen Methode beschäftigen? Weil die bisherigen Ansätze zwar gut sind, im Gesamtkontext jedoch bisher noch eine Lücke offen ließen: Die Architektur des Systems.

Es fokussieren zwar auch bisher der Integrations- und Systemtest zu einem gewissen Teil auf die Architektur des zu testenden Systems. Die Wechselwirkung zwischen Architektur und Testen und auch die damit verbundenen Kompromisse und Kommunikationsanforderungen zwischen Architekt und Testmanager wurden bislang jedoch nicht ausreichend berücksichtigt.

Mit dem „Architekturzentrierten Testen“ wird diese Lücke geschlossen!

Das „Architekturzentrierte Testen“ ist eine konstruktive Technik im Testprozess mit dem Ziel, möglichst frühzeitig im Sinne einer guten Qualität Einfluss auf die Produktentstehung zu nehmen.

Dadurch werden sowohl der Architektur- als auch der Testprozess effektiver und effizienter, weil man Probleme und Abhängigkeiten bereits sehr zeitnah lösen kann.

Die Architektur eines Systems ist ja eines der wesentlichsten und kritischen Elemente im Entstehungsprozess einer Software und nachträgliche Änderungen haben meist sehr weitreichende Auswirkungen.

Durch das „Architekturzentrierte Testen“ bekommt der Architekt mit einem entsprechend qualifizierten Testmanager einen Sparringpartner auf Augenhöhe. Dies hilft, einen hohen Qualitätsstandard in der Entwicklung sicherzustellen.

Dipl.-Ing. Johannes Bergmann

Berater, Trainer, Ziviltechniker für Informatik

Testen und Qualitätssicherung haben sich in den letzten Jahren immer weiter entwickelt, um begleitend das Entstehen von Qualität zu unterstützen.

War das Testen früher noch darauf ausgelegt, analytisch bzw. nachgelagert stichprobenartig zu prüfen, ob Anforderungen durch das Testobjekt erfüllt sind, so hat sich dies immer mehr in Richtung einer konstruktiven Qualitätssicherung verlagert. Das bedeutet, dass das Testen vermehrt frühzeitig einsetzt, um Fehler gleich im Vorfeld zu vermeiden.

Als Ansatzpunkt dienen oft die Anforderungen. Mit dem „Architekturzentrierten Testen“ wird dies nun in Bezug auf Architektur erweitert, denn die Architektur liefert bereits sehr frühzeitig sehr detaillierte Informationen, damit die Teststrategie optimal ausgerichtet werden kann und wie Fehler im Vorfeld vermieden werden können.

Der Architekturprozess

Das Erstellen einer Softwarearchitektur folgt einem Prozess. Natürlich hat letztlich jedes System eine Architektur, auch wenn diese nicht innerhalb des Architekturprozesses geplant und spezifiziert wurde. Folgt man jedoch einem Prozess, erhält man eine Architektur, die mit allen Stärken und Schwächen bewusst so entschieden wurde. Wichtige Elemente bzw. Phasen dieses Prozesses sind:

▣ Systemabgrenzung

Hierbei wird untersucht, welche Teile des Systems gestaltbar sind und wo Randbedingungen liegen, die als gegeben hingenommen werden müssen.

▣ Wechselwirkungen

Aus den Randbedingungen, den Anforderungen und den Architekturzielen ergeben sich in Summe die Einflussfaktoren. Diese stehen meist in Wechselwirkungen zueinander. Die Kenntnis dieser Wechselwirkungen ist relevant, da Einflussfaktoren sich gegenseitig verstärken oder abschwächen können. Da alle Einflussfaktoren aufgrund der Wechselwirkungen zueinander oft nicht 100% erfüllt werden können, stellt eine Architektur immer auch einen Kompromiss dar. Dieser Kompromiss (oder auch die Kompromisse) führen letztlich zu Risiken, die pro-aktiv gesteuert werden müssen.

▣ Fachliche Architektur

In diesem Schritt wird das Problem zunächst fachlich in Bausteine zerlegt. Diese Zerlegung ist wichtig, um die Komplexität des Problems beherrschbar zu machen. Bei dieser fachlichen Zerlegung werden technische Aspekte (noch) nicht oder nur sehr wenig berücksichtigt. Vielmehr ist das Ziel, bereits in der Fachlichkeit zusammenhängende Teile zu identifizieren und entsprechend ihrer fachlichen Nähe zu gruppieren. Im Gesamtbild stehen diese Bausteine dann über ihre gegenseitigen Abhängigkeiten in Zusammenhang. Schon diese fachliche Modellierung gibt erste wichtige Hinweise über die spätere Architektur:

- ▣ Von welchen fachlichen Bausteinen sind viele andere abhängig? Dies lässt erste Risikoassessungen zu.
- ▣ Ist das System ausgewogen geschnitten?
- ▣ Wo werden Schnittstellen liegen?
- ▣ Konnten Architekturprinzipien und Heuristiken eingehalten werden? (z.B. Kohäsion)

Technische Architektur

Die technische Architektur zeigt die Umsetzung. Während die fachliche Architektur mehr dazu dient, das "Was" näher zu untersuchen, zeigt die technische Architektur das "Wie". Die Bausteine der fachlichen Architektur werden also näher untersucht, es werden die Wechselwirkungen herangezogen und der Architektur einigt sich mit den Stakeholdern auf einen Kompromiss, welche der Einflussfaktoren in welchem Maße berücksichtigt werden und übersetzt diese (oftmals unter Verwendung von Architekturmustern) in eine technische Architektur.

Der Architekt

Wichtig ist, dass zum Prozess auch die Rolle des Architekten gehört. Er hält die Fäden in der Hand und triggert, dass die entsprechenden Tätigkeiten auch gemacht werden. Da eine Architektur einen Kompromiss darstellt, ist es seine Aufgabe, diesen mit den Stakeholdern abzustimmen und die verschiedenen Sichten und Erwartungen zu einer gemeinsamen Win-Win Situation zusammenzuführen.

Nachdem nun der Architekturprozess grob umrissen wurde, wird im Folgenden gezeigt, wie das Wissen um die Architektur die Ausrichtung des Testens positiv beeinflussen kann (und umgekehrt).

Systemabgrenzung

Bei der Systemabgrenzung verfolgen Testmanager und Architekt ähnliche Ziele. Beiden muss klar sein, wie sie in Bezug auf Fremdsysteme mit ihren Ansätzen reagieren müssen. Die Systemabgrenzung ist auch der erste Ansatzpunkt des Testmanagers, Anforderungen an die Architektur einzubringen.

Dazu gehört z.B., welche Testwerkzeuge und Technologien des Testrahmens von der Architektur bezüglich ihrer Schnittstellen berücksichtigt werden müssen. Der Testmanager stellt somit einen Stakeholder des Architekten dar.

Beide müssen zudem ein konsistentes Bild hinsichtlich des Systemkontextes haben. Z.B. könnte der Testmanager andere Quellen, aus denen sich Anforderungen ergeben, gegen die die Tests spezifiziert werden, berücksichtigen. Wird dies nicht in gleichem Maße auch von dem Architekten berücksichtigt, arbeiten beide aneinander vorbei auf unterschiedliche Ziele hin.

Das Mittel, um hier eine konsolidierte Sicht herzustellen sind Reviews. Auch Stakeholderlisten und deren Prioritäten sollten gegenseitig geprüft und konsolidiert werden. Desgleichen gilt für alle anderen Formen von Einflussfaktoren, die sich aus dem Systemkontext ergeben.

Die Grenze muss von beiden Rollen gleich definiert sein. Die Grenze zum Systemkontext ist gerade zu Anfang noch breit und eher unscharf. Sie wird im weiteren Verlauf durch viele Diskussionen und Erkenntnisse konkreter. In jedem Fall darf sie vom Testmanager und vom Architekten nicht unterschiedlich gezogen sein. Zieht der Testmanager sie z.B. enger, so wird er

beim Systemtest manche Teile nicht berücksichtigen, die vom Architekten jedoch bei der Architektur mitgestaltet wurden und getestet werden sollten.



Abbildung 1: Systemabgrenzung und Testen im Systemkontext

Wechselwirkungen

Nichtfunktionale Anforderungen und Einflussfaktoren stehen in Wechselwirkung zueinander. Manche beeinflussen sich in negativen Sinne.

Security wirkt z.B. negativ auf die Bedienbarkeit, wenn ständig vom Benutzer Sicherheitsabfragen durchgeführt werden müssen. Performance wirkt oft negativ auf die Wartbarkeit, wenn der Code sehr auf das Zeitverhalten zugeschnitten ist.

Diese Wechselwirkungen sollte auch der Testmanager kennen. Er sollte wissen, wo der Architekt die Schwerpunkte bei den Kompromissen gelegt hat, da nicht alle Einflussfaktoren gleich stark berücksichtigt werden können. Aus den Kompromissen resultieren dann Risiken, die wiederum dem Testmanager helfen, seine Teststrategie effektiv darauf auszurichten.

Auch die Testbarkeit gehört zu den nicht funktionalen Anforderungen. Ist diese sehr hoch bewertet, kann sich hier der Testmanager frühzeitig einbringen, um eine Ausrichtung des Architekturkompromisses in Richtung der Testbarkeit zu bewirken.

Entgegen einer einfachen Analyse der Anforderungen (die in der Frühphase oft mehr als Wunsch anzusehen sind), wird der Testmanager beim architekturzentrierten Testen seine Strategie daher an der tatsächlichen Entscheidung in Bezug auf den Architekturkompromiss ausrichten.

Er wird also für alle Einflussfaktoren schauen, wie (in welcher Testtiefe) sie berücksichtigt werden müssen, ja nachdem wie relevant sie in der Architektur berücksichtigt sind.

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.



Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter <http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/>

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Letters von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.

Software Quality Days — Die größte Konferenz zum Thema „Software Qualität“ in Europa!



Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungsergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com

