Electronic version of an article published in **Industrie Management, Ausgabe 6/2006, pp. 59-61**

http://www.gito.de/
http://www.industrie-management.de/

Durchgängige Qualität von Unternehmenssoftware

Andrea Herrmann, Barbara Paech, Carsten Binnig, Timea Illes, Universität Heidelberg, Stefan Kirn, Daniel Weiß, Universität Hohenheim, Donald Kossmann, ETH Zürich, Günter Müller, Maike Gilliot und Lutz Lowis, Universität Freiburg

Dr. Andrea Herrmann forscht am Lehrstuhl Software Engineering der Universität Heidelberg im Bereich Anforderungen und deren Schnittstellen zu anderen Softwareentwicklungsaktivitäten.

Prof. Dr. Barbara Paech ist Inhaberin des Lehrstuhl Software Engineering der Universität Heidelberg. Sie ist Sprecherin des Fachbereichs Softwaretechnik und der Fachgruppe Requirements Engineering der Gesellschaft für Informatik.

M.Sc. Carsten Binnig und **Dipl.-Inf. Timea Illes** sind wissenschaftliche Mitarbeiter am Lehrstuhl Software Engineering der Universität Heidelberg.

Prof. Dr. Stefan Kirn ist Inhaber des Lehrstuhls Wirtschaftsinformatik II und Vorstandsvorsitzender des Forschungszentrums Innovation und Dienstleistung (FZID) der Universität Hohenheim.

Dipl.-Kfm. techn. Daniel Weiß ist Mitarbeiter am Lehrstuhl Wirtschaftsinformatik II der Universität Hohenheim.

Prof. Dr. Donald Kossmann leitet das Institut für Informationssysteme an der ETH Zürich.

Prof. Dr. Günter Müller leitet die Abteilung Telematik des Instituts für Informatik und Gesellschaft an der Universität Freiburg.

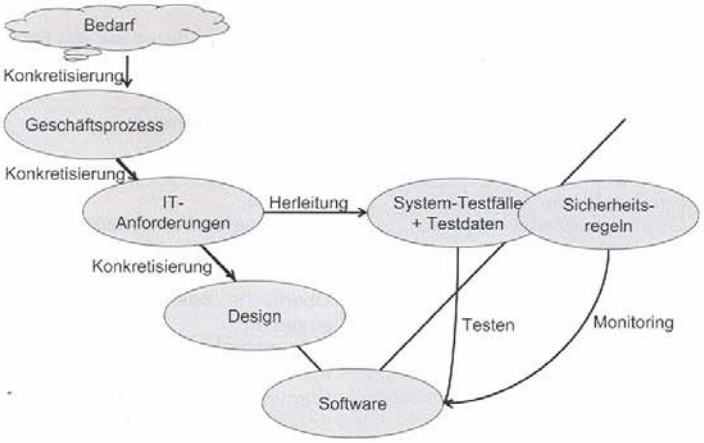
Dipl.-Inform. Maike Gilliot und **Dipl.-Inform. Lutz Lowis** sind als wissenschaftliche Mitarbeiter in der Abteilung Telematik des Instituts für Informatik und Gesellschaft an der Universität Freiburg tätig.

IT-Systeme, die in einem dynamischen Umfeld betrieben werden, müssen sich ständig an neue betriebliche Anforderungen anpassen. Insbesondere im Rahmen neuer Compliance-Regularien wie Sarbanes-Oxley-Act oder BASEL II ist eine vollständige Transparenz der Geschäftsprozesse und der IT-Unterstützung auch nach den Änderungen gefordert. IT-Abteilungen und Software-Hersteller müssen deshalb die Qualität des IT-Systems klar definieren und durchgängig in der Entwicklung und im Betrieb gewährleisten.

Kontakt:
Dr. Andrea Herrmann
Institut für Informatik
Neuenheimer Feld 326, 2. Stock
69120 Heidelberg
Tel.: 06221 / 54 5816
E-Mail:
andrea.herrmann@informatik.uni-heidelberg.de

Zunehmend werden im betrieblichen Umfeld IT-Systeme eingesetzt, die die Leistungserbringung in der gesamten Wertschöpfungskette über Unternehmensgrenzen hinweg unterstützen. Die komplexe Unternehmensumwelt führt dabei zu ständigen Veränderungen der Anforderungen an die IT-Systeme: Änderungen der inner- und zwischenbetrieblichen Geschäftsprozesse, der Organisationen oder deren Ziele, aber auch Änderungen an den IT-Systemen selbst. Wichtig ist nicht nur, diese Änderungen mit möglichst geringem Aufwand durchführen zu können, sondern auch, dass die Qualität des Systems auf hohem Niveau bleibt und die Transparenz der Prozesse und der IT-Unterstützung gewährleistet ist. Compliance-Regularien wie Sarbanes-Oxley-Act (SOX), BASEL II oder KonTragG (Gesetz zur Kontrolle und Transparenz im Unternehmensbereich) fordern die Quantifizierung von IT-Risiken. Als Voraussetzung dafür wurde im Projekt SIKOSA (Sichere kollaborative Softwareentwicklung und Anwendung) explizit eine Methodik entwickelt, mit

Bild 1: Darstellung der durch die SIKOSA-Methodik unterstützten Aktivitäten.



der definiert werden kann, was Qualität für Geschäftsprozesse bedeutet, wie sie durch das IT-System sichergestellt wird und wie dessen Qualität gemessen und geprüft wird. Diese durchgängige Nachverfolgbarkeit der Qualitätsanforderungen ist die Grundlage für ein erfolgreiches Änderungsmanagement und den Compliance-Nachweis. Sie wird bisher weder in der Praxis noch in der Forschung verwirklicht, da an den Schnittstellen immer noch Methodenbrüche zutage treten. Darum arbeiten in SIKOSA Experten aus den unterschiedlichen Bereichen zusammen und haben erstmals eine durchgängige Methodik entwickelt. Diese besteht aus Methoden, die gemeinsam oder modular eingesetzt werden können.

Überblick über die SIKOSA-Methodik

Bild 1 zeigt schematisch anhand des V-Modells auf, welche Aktivitäten des Softwareentwicklungsprozesses die SIKOSA-Methodik unterstützt, um Durchgängigkeit zu erreichen: die Erstellung von Geschäftsprozessmodellen, Ableitung von Geschäftsprozessanforderungen und deren Konkretisierung in Form von IT-Anforderungen und im Design sowie eine möglichst automatische Erzeugung von Systemtestfällen, Systemtestdaten und Sicherheitsregeln aus den IT-Anforderungen. Wichtig ist nicht nur die einmalige Übergabe und Weiterverwendung von Produkten an spätere Aktivitäten, sondern auch eine ständige Nachverfolgbarkeit, die auch bei Änderungen an den Dokumenten und der Software erhalten bleibt. Diese Nachverfolgbarkeit erlaubt es, die Quellen und die Weiterverwendung von Informationen nachzuvollziehen sowie den Fortschritt und die Vollständigkeit einer Aktivität in Bezug auf ein Vorgängerdokument zu prüfen und die Auswirkungen von Änderungen zu analysieren. Die einzelnen Aktivitäten der SIKOSA-Methodik werden im Folgenden beschrieben.

Geschäftsprozess- und Anforderungsanalyse

Um eine Anwendung zu entwickeln, die Geschäftsprozesse unterstützt, analysiert man zunächst diese Prozesse mit ProQAM [1] und leitet hieraus mit MOQARE (Misuse oriented quality requirements engineering) [2] die IT-Anforderungen an die Anwendung her. Prozesse und IT-Anforderungen werden nicht nur mit verschiedenen Notationen beschrieben, sondern ihre Qualität wird auch mit unterschiedlichem Maß gemessen, z.B. die des Prozesses anhand des geschöpften Werts und die der Anwendung anhand von Antwortzeiten auf eine Anfrage. Eine zentrale Rolle spielen hierbei Szenarien, sowohl die Beschreibung erwünschter Systemnutzung (Use Cases) als auch unerwünschter Nutzung (Misuse Cases). Um Qualitätsanforderungen zu konkretisieren analysiert MOQARE Misuse Cases, deren Ursachen, Folgen und Gegenmaßnahmen, die diesen Misuse Case entdecken, verhindern oder abschwächen. Auf diese Weise gelangt man zu konkreten, realisierbaren und testbaren Anforderungen, wie neue oder erweiterte Funktionalitäten (z.B. ein Datenbereinigungsprozess), Qualitätsanforderungen an Funktionalitäten oder Architekturanforderungen.

Soll beispielsweise das Geschäftsziel „Kundenzufriedenheit steigern" erreicht werden, kann das u.a. bedeuten, das IT-Qualitätsziel „Dauer des Use Case Bestellung aufgeben von unter 8 Minuten" zu erreichen. Einige Misuse Cases und deren Gegenmaßnahmen enthält Bild 2.

Bild 2: Einige Misuse Cases und Gegenmaßnahmen in Bezug auf das Qualitätsziel „Dauer des Use Case Bestellung aufgeben von unter 8 Minuten".

Misuse Case	Gegenmaßnahme(n)
Besondere Last führt zu Geschwindigkeitsverlust	Lasttests; Monitoring der Antwortzeiten; Beschränkung der Benutzerzahlen
Verzögerung bei der Eingabe einer Bestellungdurch Benutzerfehler	Benutzerfreundlichkeit der Oberfläche „Bestellung eingeben"
Performanzverlust des Systems durch Wartungsfehler	Dokumentation performanz - relevanter Entscheidungen; Wartungshandbuch; Schulung der Wartungsmitarbeiter

Design

Auf dem Weg zum Systementwurf müssen oft Anforderungskonflikte gelöst werden. Hierbei sind die Machbarkeit, Kosten und Abhängigkeiten zwischen Anforderungen Entscheidungskriterien. Diese können allerdings nur aufgrund eines ersten Architekturentwurfs realistisch beurteilt werden. In der SIKOSA-Methodik wird deshalb durch die Methode ICRAD (Integrated Conflict Resolution and Architectural Design) [3] das Lösen von Konflikten zwischen Anforderungen und das Erstellen des Architekturentwurfs viel enger miteinander verzahnt als üblich.

Testen

Beim Testen wird ein System darauf hin getestet, ob und wie weit es die spezifizierten Anforderungen erfüllt. Um dies zu erreichen wird eine Menge von Testfällen entwickelt, die das zu testende System mit unterschiedlichen Eingabewerten ausführen. Die Definition dieser Testfälle ist schwierig und zeitaufwändig. In SIKOSA wird mithilfe des PAT³-Ansatzes [4] unter Verwendung von Mustern aus Use Cases eine Testspezifikation hergeleitet, die das Verhalten des Systems aus Sicht der Benutzer testet.

Die Systemtestfälle werden ergänzt durch die nötigen Testdaten (Eingabedaten der Anwendung und Tabellen einer Datenbank). Auf Basis der Vorbedingungen der Testspezifikation werden Testdaten automatisiert generiert [5].

Ebenso wird die automatisierte Testfallausführung unterstützt [6]. Beides ermöglicht insbesondere bei komplexeren Datenbankanwendungen eine Kostenersparnis bei erhöhter Qualität. Außerdem können nach Änderung der Anforderungen schnell angepasste Testfälle automatisiert hergeleitet und ausgeführt werden.

Sicherheitsmonitoring

Zentral für die Sicherheit von Unternehmensdaten und -ressourcen ist die Zugriffskontrolle. Dazu werden die Zugriffsregeln *vor* der Ausführung der Unternehmenssoftware aus den Sicherheitsanforderungen abgeleitet. Jedoch sind die Konsequenzen von Softwareänderungen aufgrund der Komplexität der Softwaresysteme nur mit großem Zeitaufwand überprüfbar. Eine statische Beschreibung der Zugriffsrechte ex ante ist nicht mehr ausreichend. Daher müssen Zugriffsentscheidungen *während* der Durchführung des Geschäftsprozesses getroffen werden, indem von einem adaptiven Sicherheitsmonitor über Zugriffsanfragen entschieden wird.

Der Sicherheitsmonitor (Bild 3) betrachtet neben den statischen Sicherheitsregeln auch situationsbedingte sicherheitsrelevante Informationen. Diese laufzeitabhängigen Sicherheits-informationen sind zum einen verfeinerte und formalisierte Schwachstellen und zum anderen potenzielle Angriffe (Misuse Cases).

Bei einer Zugriffsanfrage können sich folgende Situationen ergeben:

- Die Zugriff ist gemäß der statischen Sicherheitsregeln nicht erlaubt: Dann wird der Zugriff abgelehnt.
- Die statischen Sicherheitsregeln sind erfüllt und es sind keine sicherheitskritischen Schwachstellen bezüglich dieses Zugriffs bekannt: Der Monitor gewährt Zugriff ohne Rückfrage.
- Die statischen Sicherheitsregeln sind erfüllt, jedoch wird der Zugriff aufgrund der Schwachstellen als sicherheitskritisch eingestuft. Die Anfrage wird zur Entscheidung an einen Verantwortlichen weitergegeben.

Zusammenfassung und Ausblick

Die SIKOSA-Methodik sichert die durchgängige Nutzung und Verwaltung von Anforderungen in den Aktivitäten des Softwareentwicklungsprozesses. Dies ist eine notwendige Bedingung für das Änderungsmanagement und den Compliance-Nachweis.

Da in diesem Artikel nur die groben Grundzüge der SIKOSA-Methodik dargestellt werden können, empfehlen wir Ihnen die unter www.sikosa.org bereitgestellten Veröffentlichungen. Sie können uns auch gerne persönlich kontaktieren.

Es gibt bereits praktische Erfahrungen mit den SIKOSA-Methoden in realen Projekten. Die gesamte Methodik wird zurzeit bei einem mittelständischen Unternehmen der Automobilindustrie eingesetzt. Wir bieten auch anderen Unternehmen Unterstützung bei der Anwendung an.

Literatur

[1] Dietrich, A., Otto, S., Kirn, S.: Simulationsmodell für logistische Prozesse in Mass-Customization-Szenarien. In: Kirn, S. u.a.: Kundenzentrierte Wertschöpfung mit Mass Customization. Kundeninteraktion, Logistik, Simulationssystem und Fallstudien. Magdeburg 2005.

[2] Herrmann, A., Paech, B.: Quality Misuse. REFSQ Workshop 2005, S.193-199.

[3] Herrmann, A., Paech, B., Plaza, D.: ICRAD: An Integrated Process for Requirements Conflict Solution and Architectural Design. In: IJSEKE 16 (2006) 5, im Druck.

[4] Illes, T., Paech, B.: Workshop „From „V" to „U" or: How Can We Bridge the V-Gap Between Requirements and Test?". Software & Systems Quality Conferences. Düsseldorf 2006.

[5] Binnig, C., Kossmann, D., Lo, E.: Testing database applications. In: ACM SIGMOD, 2006.

[6] Haftmann, F., Kossmann, D., Kreutz, A.: Efficient Regression Tests for Database Application Systems. CIDR 2005, S. 95-106.

Schlüsselwörter:
Softwareentwicklung, Qualität, Änderungsmanagement, Compliance, Unternehmenssoftware

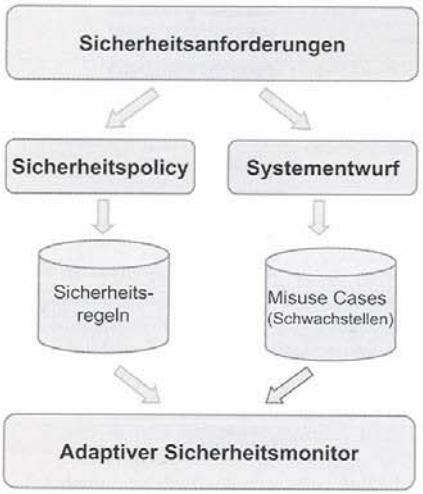


Bild 3: Eingaben des Adaptiven Sicherheitsmonitors.

Integrated Quality of Corporate Software

IT systems which operate in a dynamic environment must adapt to permanently changing business requirements. In particular new compliance rules like Sarbanes-Oxley-Act or BASEL II demand complete transparency of business processes and IT-support, also after changes. IT-departments and software providers need to clearly define and continuously assure the quality of the IT system during development and operation.

Keywords
software development, quality, change management, compliance, corporate software