

JavaTMmagazin

Internet & Enterprise Technology

XML
extra included

Komponenten

Architekturen EJB, CORBA, .NET, Apache Avalon

Struts-Workshop

Struts-Entwicklung am Beispiel erklärt

XML-Wissensdatenbank

MVC mit EJB, Struts, Cocoon

J2EE-Patterns

MVC 2-Architektur verstehen

JUnit-Tests automatisieren

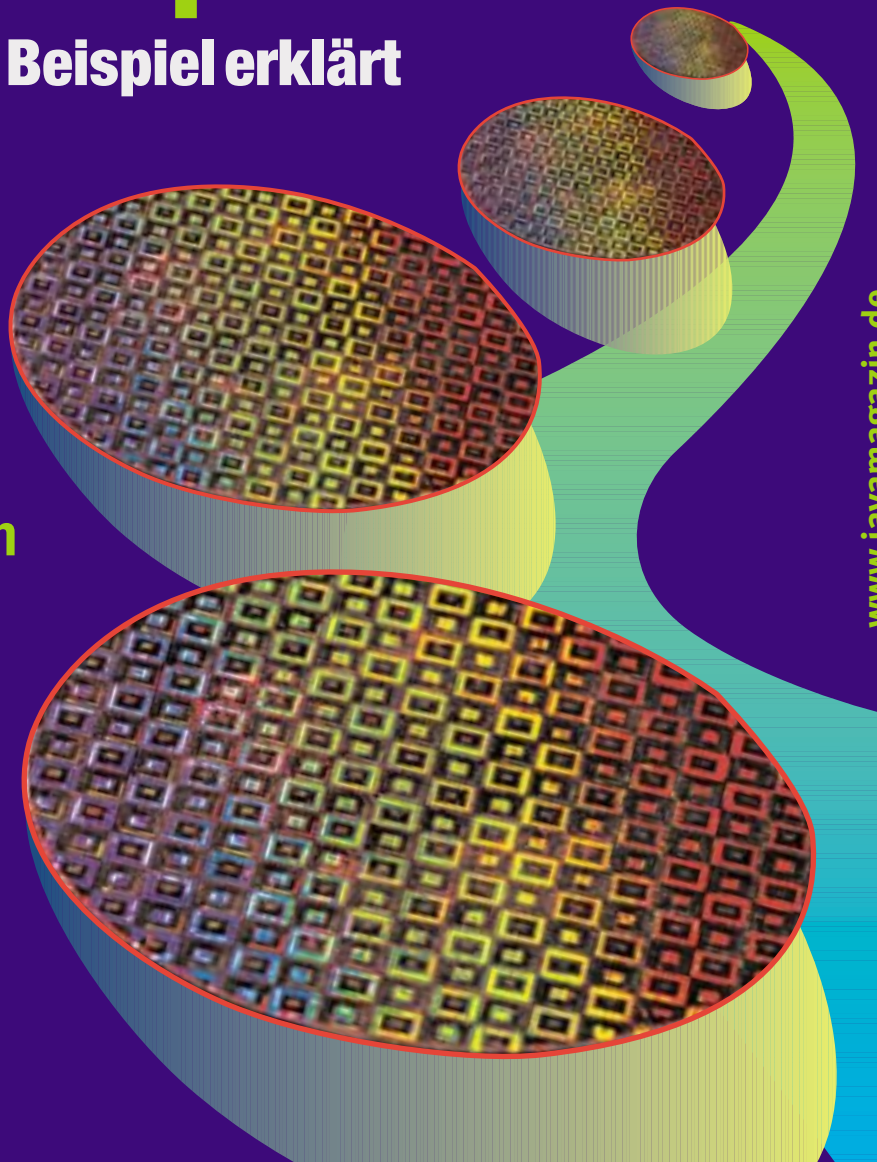
JUnitDoclet generiert Testklassen

Mobile Web Services

SOAP und XML-RPC auf
mobilen Geräten

Benutzereingaben prüfen

Framework für Plausibilisierung &
Validierung



www.javamagazin.de



mit CD!

D 45 86 7



Prozessmodelle angewandt, Teil 2: XX Process – Prozess für die Entwicklung von Individualsoftware

von Andreas Lux

Extrem rational

Prozessmodelle wie der Rational Unified Process (RUP) oder eXtreme Programming (XP) lassen sich in puristischer Form in der Praxis nur schwer anwenden. Der eXXcellent Process hat sich die Vorteile der einzelnen Modelle zueigen gemacht und ihn für den Einsatz bei der Erstellung von kundenspezifischer Individualsoftware optimiert.

Im letzten *Java Magazin* (Ausgabe 12. 2002) wurden, als Vertreter moderner Prozessmodelle, RUP und XP mit ihren Stärken und Schwächen im Hinblick auf den täglichen Projekteinsatz vorgestellt. Wie wir dort erläutert haben, ist der Einsatz vom RUP oder XP im reellen Projektalltag sehr aufwändig bzw. sehr schwer durchführbar. Die einzelnen Prozessmodelle müssen vor dem Einsatz an die Gegebenheiten des Unternehmens, das Einsatzgebiet und das Umfeld des Projekts, sowie die Vorstellungen der am Projekt beteiligten Mitglieder angepasst werden.

Im zweiten Teil dieses Artikels wird der eXXcellent Process (XX Process) vorgestellt. Dieser soll lediglich als Stellvertreter einer möglichen Implementierung eines beliebigen, eigenen Prozesses fungieren.

Die Motivation ...

Reflektieren wir noch einmal die Anforderungen, die wir an einen Prozess stellen. Ein Prozess soll als Leitfaden für die Erstellung eines Softwaresystems dienen. Es soll uns leiten und einschränken, wo es für die Aufgabe nützlich ist und soll uns Freiräume erhalten, wo dies möglich ist. Er soll uns Dinge bereitstellen, die uns helfen, das gesteckte Ziel zu erreichen, soll uns aber nicht mit zusätzlicher Arbeit belasten. Er soll uns an das aktuelle Problem heranfüh-

ren, wenn wir neues Terrain betreten und uns sehr konkrete Hilfen bieten, wenn wir uns mit Details beschäftigen.

Die Idee ...

Wie aber lässt sich nun ein Prozess definieren, der all diese Anforderungen erfüllt? Damit der Prozess als Leitfaden und als eine Unterstützung für die Durchführung eines Softwareprojekts verstanden werden kann, haben wir uns dafür entschieden, den Prozess als anforderungsgetriebenen Prozess zu formulieren. Anforderungsgetrieben bedeutet hierbei, dass der Mitarbeiter einen Lösungsvorschlag (Mikroprozess) für ein konkretes Problem (Anforderung) vom Prozess erwartet.

Damit der Prozess uns für alle in einem Projekt auftretenden Situationen die entsprechende Hilfe und Unterstützung geben kann, muss er zwingend eine iterative und inkrementelle Vorgehensweise unterstützen. Die jeweilige Unterstützung ist natürlich abhängig davon, in welcher Rolle sich der Hilfesuchende gerade befindet. Somit ist es unabdingbar, ebenfalls alle am Prozess beteiligten Rollen zu definieren. Hierzu gehören nicht nur die Rollen, die das Entwicklungsteam inne hat, sondern natürlich auch die Rollen, die die Mitarbeiter des Kunden im aktuellen Projekt einnehmen.

Um dem Prozess keinen allzu großen Ballast, wie beim RUP, aufzulasten, ist es erforderlich, dessen Inhalte sehr stark an das eigene Umfeld anzupassen. Dies sind sowohl die Strukturen und Gepflogenheiten des eigenen Unternehmens als auch die Rahmenbedingungen des Projekts. Im Falle des XX Process bezieht sich dieses auf die Erstellung von Individualsoftware im Kundenauftrag mittels objektorientierter Vorgehensweisen.

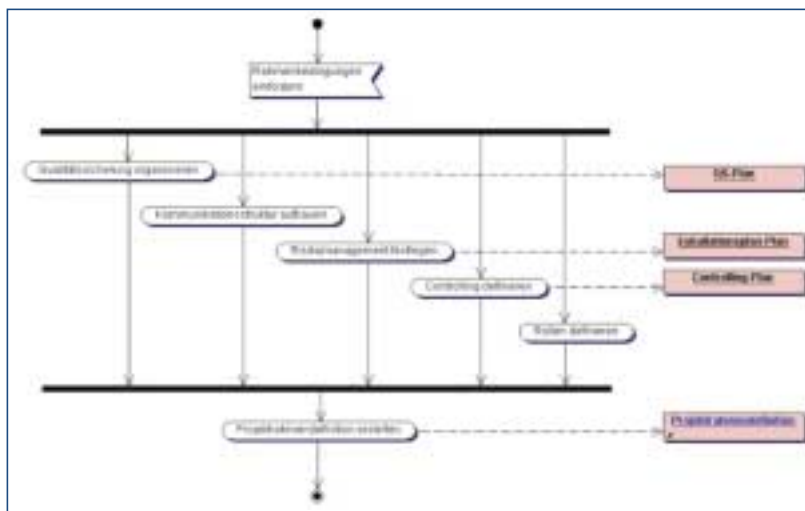
Das Tool ...

Nachdem wir geklärt haben, was der Prozess beinhalten soll, stellt sich nun die Frage, wie er umgesetzt werden soll. Wir haben uns dafür entschieden, den Prozess selbst mit dem UML-Case-Tool Together zu erstellen. Dies bringt folgende Vorteile mit sich:

- Die Autoren des Prozesses arbeiten mit einem Werkzeug und einer Notation, die ihnen sehr vertraut ist. Die Vorteile sind einfache Pflege und einheitliche Symbolik.
- Der Prozess selbst kann als eigenes Projekt mit eigenen Anforderungen und eigenen Umsetzungen gesehen werden.
- Der objektorientierte Gedanke wird durch Modularisierung im Prozess selbst bereits gelebt.

- Mittels Together können aus dem UML-Modell des Prozesses sehr leicht unterschiedliche Arten von Dokumentationen (HTML, RTF, PDF) mit unterschiedlichen Detaillierungsgraden generiert werden.
- Durch die Verlinkungsmöglichkeiten innerhalb des UML-Modells ist es in Together auf einfache Weise möglich, sich eine Navigation durch den Prozess generieren zu lassen.
- Durch den Modellierungsansatz ist es (im Gegensatz zu einer rein textuellen Beschreibung) nun auch möglich, den Prozess auf syntaktische Fehler selbst zu prüfen, da dieser in strukturierter Form vorliegt.

Abb. 1:
Projekt
aufsetzen



Der Aufbau ...

Als maßgeblicher Faktor für den erfolgreichen Einsatz des Prozesses hat sich dessen Aufbau und Strukturierung erwiesen. Aufbau und Strukturierung sind deshalb so wichtig, da die Arbeit mit der Prozessbeschreibung dem Mitarbeiter keine zusätzliche Last aufbürden darf. Aus diesem Grund haben wir uns dazu entschieden, die Prozessanforderungen als Einstiegspunkt in den Prozess zu definieren. Die Anforderungen werden als *UseCases* notiert und haben eine Verknüpfung zu den zugehörigen Abarbeitungsvorschriften, im Weiteren Mikroprozesse genannt.

Die Mikroprozesse sind als Aktivitätsdiagramme notiert und beschreiben die *Workflows* für die jeweiligen Anforderungen. Jeder Aktivität des Mikroprozesses können Dokumente zugewiesen werden, die helfen, die aktuelle Aktivität durchführen zu können. Diese können sein:

- Checklisten
- Guidelines
- Best Practices
- Vorlagen
- Beispiele
- Toolbeschreibungen

Um die Benutzbarkeit des Prozesses zu erhöhen, ist es auch möglich, (sehr wichtige) Dokumente direkt den *UseCases* zuzuweisen.

Die Disziplinen ...

Da die Probleme, die bei der Erstellung von Individualsoftware auftreten können,

sehr vielseitig sind, sind natürlich auch die Anforderungen an den Prozess sehr umfangreich. Um die Komplexität des Prozesses zu reduzieren, wurde er in 3 + 1 Disziplinen unterteilt.

Jede einzelne Disziplin befasst sich mit einem Kernproblem, das bei der Erstellung von Individualsoftware auftritt. Unserer Auffassung nach sind das:

- Projektmanagement
- Anforderungsanalyse
- Umsetzung
- (Prozessänderung)

In den einzelnen Disziplinen gibt es unterschiedliche Aufgabenbereiche, die jedoch nicht zwingend in allen Aufgabenbereichen vorkommen müssen. So besteht üblicherweise jede Disziplin aus einer oder mehreren analysierenden, planenden, erzeugenden, prüfenden und dokumentierenden Aufgaben.

Die einzelnen Aufgabenbereiche sind zwar grundsätzlich getrennt zu betrachten, je nach Disziplin und Projekt kann es aber auch Überschneidungen geben. Und natürlich gibt es für die Aufgaben, je nach Disziplin, unterschiedliche Gewichtungen.

Projektmanagement

Als Kerndisziplin für die Erstellung von Individuallösungen verstehen wir das Projektmanagement. Der unbeliebteste Teil des Projektmanagements ist wohl die Projektplanung. Hierbei werden Meilensteine definiert, deren Zeitlichkeit und

Inhalt mit dem Kunden abgestimmt werden. An den festgelegten Terminen werden später die vereinbarten Dokumente oder Funktionalitäten dem Kunden übergeben. Der Projektleiter erarbeitet zusammen mit dem Kunden einen Eskalationsplan, der Anwendung findet, falls ein Meilenstein nicht eingehalten werden kann.

Der Projektplan wird in dieser frühen Phase noch nicht mehr beinhalten als die Meilensteine und das beabsichtigte Projektende. Einen detaillierten Iterationsplan hingegen wird es zu diesem Zeitpunkt noch nicht geben. Dieser wird passend zu jedem Meilenstein vom Projektleiter zusammen mit dem Team erstellt. Somit wird vermieden, dass sich die Entwickler mit unrealistischen Iterationsplänen auseinandersetzen müssen.

Dem Projektmanagement ordnen wir aber deutlich mehr Aufgaben zu als lediglich das Erstellen eines Projektplanes und die Definition von Meilensteinen. In diese Disziplin fallen unserer Meinung nach auch Themen wie Qualitätssicherung, Risikomanagement, Controlling und nicht zuletzt die Pflege der Kommunikationskultur sowie die Definition der im Projekt vertretenen Rollen.

All diese Vereinbarungen sind notwendig, um eine Projektrahmendefinition erstellen zu können. Dieses Dokument stellt die Basis der gesamten Projektorganisation und -kommunikation dar und steht sowohl dem eigenen Team als auch dem Auftraggeber zur Verfügung. Wie dieses Dokument erstellt wird, welche

können und wie anschließend das Ergebnis dieser Kontrolle wieder an das Analyse-Team zurückgeführt wird.

Das Ergebnis der Anforderungsanalyse liegt nun in Form eines Spezifikationsmodells vor. Da ein Großteil der Kunden zur Überprüfung der fachlichen Korrektheit das Modell aber in ausgedruckter Form einfordert, muss aus dem Modell eine druckfähige Dokumentation erstellt werden können.

Damit der Prozess eine echte Hilfe wird, muss dessen Beschreibung sehr detailliert sein. Beispielsweise wird beim Generieren des Spezifikationsdokuments genau erklärt, mit welchen Dokumentenvorlagen und Generatoren gearbeitet werden muss, welche Probleme beim Generierungsprozess auftreten bzw. wie bekannte Probleme gelöst werden können.

Als Qualitätssicherungsmaßnahmen bietet der eXXcellent-Prozess Guidelines und Checklisten an, die es erlauben, die Anforderungen an die Applikation exakt zu beschreiben. Da das Analyseergebnis als Modell vorliegt, ist es nun möglich, syntaktische Prüfungen über das Modell auszuführen, die auf widersprüchliche oder unvollständige Inhalte hinweisen.

Umsetzung

Unter Umsetzung verstehen wir die Realisierung einer definierten Funktionalität als ausführbare Software. Hierzu gehören auch Prototypen, die die Machbarkeit der Applikation unter Zuhilfenahme der verwendeten Technologien, neuen Bedienkonzepten und ähnliches beweisen.

Nachdem die Machbarkeit durch diverse Prototypen bewiesen wurde, gibt es mindestens eine Iteration, in der die Architektur der Applikation unter Verwendung möglichst aller verwendeter Technologien definiert wird.

Auf der Basis der Architektur wird nun die spezifizierte Funktionalität in mehreren Iterationen abgewickelt. Die Rahmendaten jeder Iteration werden vor Iterationsbeginn zwischen dem Projektleiter und dem Entwicklungsteam abgestimmt. Zu den Rahmendaten gehören unter anderem der zu implementierende Umfang, Ansprechpartner, Teamgröße, Auslieferungsdatum, Abnahmekriterien und Eskalationsschema.

Das Eskalationsschema beispielsweise tritt dann in Kraft, wenn absehbar ist, dass die verabredeten Inhalte der Iteration nicht erbracht werden können. Bei den Abnahmekriterien wird festgelegt, welche Tests für die aktuelle Iteration relevant sind (Business-tests, Integrationstests und Unit-Test), wie die Tests angewandt und auf welche Weise sie protokolliert werden müssen.

Im eXXcellent-Prozess wird auf der einen Seite versucht, eine Iteration so kurz wie möglich zu terminieren. Dies gewährleistet, dass der Kunde in relativ kurzen Abständen eine lauffähige Version der Software bekommt. Diese kann er nun auf fachliche Belange und auf deren Bedienbarkeit hin prüfen.

Auf der anderen Seite sollte aber eine Iteration fachlich zusammenhängende Aspekte möglichst geschlossen behandeln. Beide Aspekte lassen sich nach unserer Erfahrung mit mittleren Iterationslängen zwischen zwei und drei Monaten optimal vereinbaren.

Am Ende jeder Iteration steht eine Integration, die das Ergebnis der Iteration in das Gesamtsystem einfügt. Somit sind wir in der Lage, dem Auftraggeber zu den vorab definierten Meilensteinen jeweils ein funktionsfähiges System liefern zu können. Dies ist besonders wichtig, da jede Erweiterung des Systems zur Meinungsbildung des Kunden beiträgt. Auf diese Weise lassen sich Missverständnisse und Richtungswechsel bei der Funktionalität früh erkennen und das Vertrauen des Kunden in die Entwicklung stärken.

Als Qualitätssicherungsmaßnahmen für die Umsetzung dienen Guidelines und Checklisten für die Architektur und die Designs. Für die Erhöhung der Qualität der Implementierung selbst liefert der Prozess Empfehlungen für die Durchführung von Reviews und Walkthroughs. Des Weiteren wird für jede Iteration versucht, die Implementierungen für automatische Tests so früh wie möglich zu implementieren. Dieses Vorgehen hat sich bei der Identifikation von widersprüchlichen Spezifikationen als sehr hilfreich erwiesen.

Prozessänderung

Die vierte Disziplin hat mit der Softwareerstellung selbst direkt eigentlich nichts zu tun. Für die Steigerung der Software-

qualität ist es aber unabdingbar, die Qualität des Prozesses ständig zu erhöhen. Änderungen am Prozess sind abhängig von den Erfahrungen aller Projektteilnehmer. Aus diesem Grund muss ein Mechanismus geschaffen werden, der es auf unbürokratische Weise erlaubt, konstruktive Anregungen zu sammeln und zu kanalisieren. Dies muss unabhängig davon geschehen können, ob die Projektteilnehmer am selben Tisch oder an (weltweit) verteilten Standorten arbeiten.

Der eXXcellent-Prozess delegiert diese Aufgabe an ein Web-basiertes Request Tracking Tool. Die gesammelten Anregungen werden hierbei sowohl kategorisiert als auch priorisiert. Die so gesammelten Vorschläge werden regelmäßig von einem Gremium beurteilt und bei Bedarf in den Prozess aufgenommen. Somit ist sichergestellt, dass der Prozess sich an eine sich stetig verändernde IT- und Kundenlandschaft anpassen kann.

Schlussbemerkung

Der eXXcellent-Prozess wurde bewusst sehr konkret auf die Anforderungen bei der Erstellung von Individualsoftware zugeschnitten. Denn nur durch diese Spezialisierung ist es überhaupt möglich, dass er allen Beteiligten eine effektive Unterstützung bietet. Er formuliert den Anspruch, die Lücken zu schließen, die RUP oder XP bei dem Versuch hinterlassen, ein möglichst breites Anwendungsgebiet zu erfassen. ■

Andreas Lux (a.lux@excellent.de) ist Projektleiter und Gesellschafter der eXXcellent solutions GmbH aus Ulm (www.excellent.de). Er befasst sich seit vielen Jahren mit der Erstellung von Individualsoftware für Businessapplikationen im Umfeld Java- und Internettechnologien und ist Trainer und Consultant für UML und OO-Themen.

Links & Literatur

- Andreas Lux, „Angewandte Prozessmodelle, Teil 1“, *Java Magazin* 12.2002
- www.excellent.de/
- www.togethersoft.de/
- www.rational.com/products/rup/
- www.extremeprogramming.org/