

SECURITY:

• Angriffe abwehren: Was ist rechtens?



Server

Netzwerk

Security

Desktop

Administration

Entwicklung

» Perl

» Java

» Python

» Web

» System

» Tools

» Anwendungen

» weitere...

Hardware

Szene

Special

Heftarchiv

» 2009

» 2008

» 2007

» 2006

» 2005

» 2004

» 2003

» 2002

» 2001

» Listings

» Inlay Jahres-CD

» Sonderheft

» Technical Review

Service

Bestellen

Kontakt

Live-Streaming

Firmensuche

Stellenangebote

Technical Review

Anzeige

idealo

Sie suchen eine interessante Herausforderung? Idealo bietet Ihnen diese

» **Jobangebote bei Idealo**



Home » Heft & Abo » Heftarchiv » 2009 » 01 » **Bruchlos modelliert**

Drucken Empfehlen Kommentieren

Software-Modellierung mit Unicase

Bruchlos modelliert

von Bernd Brügge, Jonas Helming, Maximilian Kögel
Erschienen im Linux-Magazin 2009/01

Weder Anhänger des traditionellen Programmierparadigmas noch Verfechter agiler Methoden bestreiten bei Softwareprojekten die Notwendigkeit einer bruchlosen Modellierung. Das auf Eclipse aufsetzende Unicase stellt hierfür ein einheitliches Projektmodell zur Verfügung.

Inhalt

100 | Vertrauensselig

So manche Webanwendung übernimmt immer noch ungeprüft Benutzerdaten. Die Flugauskunft des Münchner Airports ist ein prominentes Beispiel.

104 | Zellkultur

Anwendungen für die Cell-Architektur gedeihen unter Linux dank eines SDK inklusive Simulator prächtig.

110 | Perl-Snapshot: Stolperfallen

Perl-Syntax ist wunderbar effizient - Einsteiger bremsst sie aber auch mal aus.

Informatiker vernetzen die Welt einerseits durch Kommunikationstechnologien. Andererseits neigen sie innerhalb ihres eigenen Fachgebiets zur Lagerbildung. Scheinbar unversöhnlich prallen die Gegensätze bei den Vorgehensmodellen für Software-Entwicklung zwischen dem traditionellen Lager und Anhängern agiler Methoden wie Scrum oder Extreme Programming aufeinander.

Andererseits schlägt Royce bereits 1973 in seinem Wasserfallmodell vor - lange vor dem Aufkommen des agilen Paradigmas -, die Machbarkeit in Softwareprojekten durch frühe Prototypen zu evaluieren, statt sich lange mit Spezifikationen aufzuhalten. Dieser Vorschlag wirkt, als käme er aus dem agilen Lager. Agile Publikationen fordern hingegen, verbreiteten Vorurteilen entgegen, keinesfalls völligen Verzicht auf Modellierung.

Eine durchgängige Modellierung der Anforderungen, des zu entwickelnden Systems und des konkreten Vorgehens ist also unabhängig vom gewählten Modell erstrebenswert. Ein neues Vorgehen einzuführen oder dieses gar während der Projektklaufzeit zu modifizieren bereitet Produktmanagern jedoch regelmäßig Kopfzerbrechen. Viel Forschungsaufwand fließt in das Change-Management an bestehenden Systemen. Dabei geht es jedoch selten um Veränderungen, die den Entstehensprozess selbst betreffen.

Durchgängig modelliert

In der Praxis ist eine bruchlose Modellierung schwer zu erreichen. Weder die zur Verfügung stehenden Modelle selbst noch die unterstützenden Tools erweisen sich dazu als leistungsfähig genug. Die Zahl der Toolgrenzen und Medienbrüche, mit denen eine durchgängige Modellierung Software-Ingeneure täglich konfrontieren würde, ist zu groß.

Zahlreiche Forschungsansätze versuchen die Zahl der Bruchstellen zu verringern - bisher mit mäßigem Erfolg. Daher sollte an die Stelle einer nachträglichen Integration ein von Anfang an flexibler und ganzheitlicher Ansatz treten. Die Open-Source-Praxis hält dazu wertvolle Einsichten parat: Dort entwickeln sich einzelne Tools oft erst nachträglich zu einem zentralen Repository - in der Open-Source-Gemeinde oft um den Bugtracker herum -, obwohl sie nicht explizit darauf ausgelegt sind.

Uniforme Plattform

Eclipse [1] ist in mehrfacher Hinsicht ein gutes Beispiel, wie ein flexibler und ganzheitlicher Ansatz für Software-Entwicklung aussehen kann. Zum einen integrieren verschiedene Hersteller ihre gemeinsamen Grundaufwände. Von allen benötigte Funktionalität entwickeln sie so quelloffen in der Form eines Framework. Zum anderen integriert die IDE Eclipse durch ausgereifte Plugins die Aufgaben des Entwicklers immer mehr zu einem kohärenten Ganzen. Was jedoch fehlt, ist ein zentrales, durchgehendes Projektmodell.

Unicase [2] ist ein Projekt auf Eclipse-Basis, das Jonas Helming und Maximilian Kögel an der TU München leiten. Es versucht sämtliche für Projektmodellierung nötigen Funktionen zu integrieren. Das Open-Source-Projekt bietet ein flexibles und anpassungsfähiges Repository für sämtliche Typen von Projekt-Artefakten. Forscher wie Firmen können es an ihre spezifischen Anforderungen anpassen. Beziehungen zwischen Artefakten bilden sie über Links ab. Unicase bindet zahlreiche existierende Eclipse-Komponenten wie das Graphical Modelling Framework (GMF) an die zentrale Datenbasis an und bietet Ansichten und Werkzeuge für alle Teilnehmer eines Softwareprojekts, vom Projektmanager bis zum Entwickler.

**OPEN SOURCE
SPEZIAL**

Jetzt am Kiosk
oder
online
bestellen



Themen-Special
SPAM

Alles was Sie zum
Thema Spam
wissen müssen:

- » **Filterung**
- » **Abwehr**
- » **Software**
- » **Hardware**

Dabei integriert Unicas auch Management und Organisation in die Modellierung. Entwickler können so beispielsweise von ihrer Aufgabeliste in einen damit verknüpften Teil der Spezifikation springen. Ein Projektmanager hingegen erhält Antworten auf Fragen wie "Welche Aufgaben sind bis zur Umsetzung der Anforderung X offen?" (*pk*)

Infos
[1] Eclipse: [http://www.eclipse.org]
[2] Unicas: [http://www.unicas.org]

Der Autor

Dr. Bernd Brügge ist Professor für angewandte Software-Entwicklung an der TU München. Er beschäftigt sich mit Software Engineering, insbesondere mit Tools, die die Teamarbeit bei Projekten koordinieren.

 Drucken  Empfehlen  Kommentieren



Ähnliche Artikel	
Ins Gesicht geblickt	Gesichtserkennungs-SDKs OpenCV, Verilook und FaceVACS
Android: Der Rivale	Googles offene Plattform für mobile Endgeräte - der erste Eindruck
Von Java zu Ajax	Google Web Toolkit zum Erstellen dynamischer Webapplikationen
Mischmasch	Mit JRuby on Rails die Vorteile von Java und Ruby nutzen
Freundlicheres Wesen	Gipfeltreffen: Erstes Eclipse Summit Europe 2006
Verwandlungskünstler	C- und C++-Entwicklung mit Eclipse

Whitepaper
Mythos und Wahrheit über Open-Source-Sicherheit Ist es riskant, Open-Source-Software in geschäftskritischen Infrastrukturen einzusetzen? Warum sollten wir für einen Open-Source-Anbieter zahlen, wenn es sich doch um "freie" Software handelt? Bedeutet der Wechsel zu Open Source, dass die Komplexität der IT-Infrastruktur zunimmt? Dieses Whitepaper von Astaro räumt mit Mythen über Open-Source-Sicherheit auf. Download PDF (Registrierung erforderlich)
Bereichern Sie Ihre Anwendungen mit Webkomponenten Wenn Sie mit der Entwicklung eines Softwareprojekts beginnen, stehen Sie vor der grundsätzlichen Frage: kann ich dieses Programm auch für die Darstellung im Web aufbereiten? Die Standardantwort lautet immer häufiger „Ja“. Finden Sie selbst heraus, wie einfach es mittlerweile ist, Webinhalte mittels der WebKit Rendering Engine direkt in ihre herkömmlichen Anwendungen einzufügen. Dank Trolltech ist dies fast ein Kinderspiel. Download PDF (Registrierung erforderlich)
Virtualisierung im Hosting-Umfeld Hohe Leistung und Steigerung der Energieeffizienz rücken stärker in den Vordergrund. Attraktive Hosting-Angebote sind ohne Virtualisierungstechnologie nicht mehr denkbar. Welche Aspekte müssen bei der Wahl der passenden Hosting-Lösung berücksichtigt werden? Download PDF (Registrierung erforderlich)

Kommentare (0)
Neuer Kommentar

Dieser Online-Artikel kann Links enthalten, die auf nicht mehr vorhandene Seiten verweisen. Wir ändern solche "broken links" nur in wenigen Ausnahmefällen. Der Online-Artikel soll möglichst unverändert der gedruckten Fassung entsprechen.

Impressum | Datenschutzerklärung | © 2009 Linux New Media AG
Partner-Sites

Deutschland: [LinuxUser] [EasyLinux] [Linux-Community]
Europa: [EasyLinux Polen] [Linux Magazine Polen] [Linux Magazine Spanien]
International: [Linux Magazine International] [Linux Pro Magazine] [Linux Magazine Brasilien] [EasyLinux Brasilien]