



Ausgabe 2011 / 2

Erscheinungsart: ca. 4 x jährlich in elektronischer Form

SPRINT - Software PRocess INTegration

In dieser Ausgabe:

- Das Problem mit der Integration von Tools & Prozessen
- \Rightarrow Die verschiedenen Integrations-Ansätze
- Der Weg zur voll integrierten SW-Entwicklungs-Organisation

Fiktion wird Realität

von Klaus Veselko

Betrachten wir die heute vielfältig eingesetzten Applikationen und Tools und die sich daraus ergebenden IT-Landschaften etwas genauer, so zeigt sich ein trübes Bild: Automatisierung von Abläufen ist – wenn überhaupt – nur in Teilbereichen vorhanden. Die Automatisierung von Prozessen durch periodisch wiederkehrende Batchjobs (Stapelverarbeitung) hat schon im letzten Jahrhundert begonnen und kann als nicht mehr zeitgemäß betrachtet werden. Die Automatisierung in einzelnen Prozess-Bereichen wie z.B. der Codierung oder der Testautomatisierung ist derzeit in vollem Gange.

Manche Organisationen sind dabei schon weiter, andere wiederum haben noch großen Nachholbedarf.

Die übergreifende Automatisierung von Prozessen und komplexen Arbeitsabläufen und Tool-Landschaften mit Anforderungsmanagement, Entwicklung, Test, Support, etc. findet man jedoch äußerst selten. Warum ist das so?

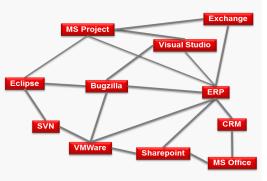


Abb.3 - Beispiel einer vernetzten Tool-Landschaft

Gerade in einem komplexen und heterogenen Aufgabengebiet wie der SW-Entwicklung sind auch die unterstützenden Tools meist sehr heterogen und zumeist als best-of-breed Insellösungen im Einsatz. Fehlende Integration mit im Prozess vorgelagerten, begleitenden bzw. nachgelagerten Werkzeugen stellt sich meist als sehr aufwändig heraus und man befürchtet, dass bei Releasewechsel oder gar Toolwechsel die bis dahin getätigten Integrationsaufwände verloren sind.

Es ist an der Zeit, dass in diesem Bereich zukunftsorientierte und flexible Ansätze realisiert werden. Mit SPRINT (Software PRocess INTegration) bringt Software Quality Lab ein Dienstleistungspaket inkl. einer Architektur und technologischen Plattform, mit der es einfach möglich ist, Prozesse in der Entwicklungsorganisation zu automatisieren, vorhandene Tools entsprechend der definierten Prozesse einzubinden und auch in Zukunft neue Werkzeuge schrittweise zu integrieren. Daraus resultieren eine Reihe von Vorteilen wie z.B. Toolunabhängigkeit, Investitionssicherheit, mehr Qualität, Effizienz und Effektivität durch Automatisierung im System- und Software-Engineering.

Völlig verschiedene, unabhängige Werkzeuge werden zu einem Komplettpaket zusammen geführt, die sich für den Anwender als voll integrierte Gesamtlösung darstellt. Die Fiktion wird damit zur Realität.



Wie ... noch mal funktioniert denn nun unser Entwicklungs-Prozess?

Unlängst habe ich mit dem Entwicklungsleiter eines Kunden über Prozess-Automation gesprochen.

Die handelnden Personen dort haben durchaus Verständnis für Tools und einen durchgängigen Entwicklungsprozess. Sie haben dies auch schon teilweise umgesetzt, indem sie passende Tools beschafft haben (z.B. für Anforderungsmanagement, Testmanagement, Bug-Tracking, etc.).

Es wurde auch erkannt, dass Tool-Inseln nicht der Prozess-Weisheit letzte Schluss sein können. Daher haben leidgeplagte Entwickler die Initiative ergriffen und die Tools in mühevoller Kleinarbeit durch selbst programmierte Schnittstellen miteinander vernetzt.

Obwohl dies gut gemeint war hat dies doch gravierende Nachteile wie z.B.:

- Intransparente Prozesse (kaum einer außer dem Entwickler der Schnittstellen weiß, wie der Prozess wirklich funktioniert)
- Inflexibel und fehleranfällig (durch ndirektionale Vernetzung der Tools entsteht eine Komplexität, die kaum beherrschbar ist). Wegen jeder kleinen Prozess-Änderung muss man zum Entwickler gehen, damit der Prozess Code geändert wird.
- eigentlichen Aufgabe, sondern mit Tool-Entwicklung — das macht vielen Entwicklern natürlich Spaß, ist aber höchst ineffizient für das Unternehmen.

Der Entwicklungsleiter klagte mir sein Leid (siehe Titel oben) und wünschte sich eine Plattform für eine transparente und flexible Prozessdarstellung und Automatisierung. Dass dies realisierbar ist, zeigt dieser Knowledge-Letter.

Dipl.-Ing. Johannes Bergsmann Staatl. befugter und beeideter Ingenieurkonsulent für Informatik

Der Quality-Knowledgeletter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit den Schwerpunkten IT-Qualitätsmanagement, Projekt- und Prozess-Management.

Inhalt: fachliche Beiträge und Schwerpunktthemen, Vorstellung neuer Produkte und Leistungen, neue wissenschaftliche Erkenntnisse und andere Fachbeiträge aus unseren Themenbereichen.

Aktuelle Fach- und Forschungsbeiträge sind willkommen Einsendungen an info@software-quality-lab.at

Weitere Infos zu diesem und anderen Themen finden Sie auf http://www.software-quality-lab.at.







SPRINT - Software PRocess INTegration

Software Quality Lab

Ausgangssituation und Problembereiche

In vielen Firmen sind die eingesetzten Werkzeuge für die Softwareentwicklung nur teilweise bzw. mangelhaft miteinander verbunden.

Die Datenübergabe basiert oft noch auf manuellem Kopieren von Datensätzen von einem Tool ins Nächste, manchmal werden Datenübergaben durch Batchfiles abgearbeitet, die auch meist keine angemessene Form des Datenaustausches und der Prozess-Steuerung ermöglichen. In weiter fortgeschrittenen Entwicklungsorganisationen ist manchmal schon eine etwas integriertere Form der Tool-Vernetzung zu finden, hier haben sich des manuellen Datenaustauschs überdrüssige Mitarbeiter selbst schon Schnittstellen zwischen den verschiedenen Tools programmiert.

Ganz abgesehen davon, dass in diesen Fällen die Komplexität der Schnittstellen durch die m:n-Verbindung unter den Tools meist exponentiell steigt, ist der Prozessablauf für die teilnehmenden Personen meist völlig intransparent und nur schwer zu überblicken — das Know-How über den Prozess steckt ja vergraben im Source-Code der Tool-Schnittstelle.

Zusätzlich sind solche Konstrukte aufgrund der hohen Komplexität auch **fehleranfälliger** — auf keinen Fall etwas editieren oder am Prozess verändern, sonst klappt nichts mehr.

Kritische Firmenprozesse und Entscheidungen hängen oft von solchen Lösungen und Daten ab, deren Zusammenstellung teilweise nicht nachvollziehbar ist oder auf Daten basiert, deren Werte teilweise schon veraltet sind.

Blickt man auf die Tool-Landschaften von Software-Entwicklungsbetrieben oder -abteilungen, so stellen sich diese gegenwärtig auch als sehr unreif dar. Dies betrifft sowohl die Prozesse, über die Werkzeuge miteinander verbun-

den sind, als auch die Datenmodelle auf denen der Datenaustausch zwischen den Tools basiert.

Die meisten Unternehmen besitzen in ihren SW-Entwicklungsabteilungen ein Konglomerat aus verschiedenen Werkzeugen, Daten und Services. Diese sind oft über Jahre oder Jahrzehnte gewachsen, die Unternehmensprozesse sind vielfach auf die Werkzeuge abgestimmt und somit stellt der Austausch oder die Neuanschaffung eines Werkzeugs ein unberechenbar hohes Risiko und einen hohen Kostenfaktor dar. Des weiteren ist anzumerken, dass sich die verwendeten Werkzeuge oft in deren Basistechnologien, Plattformen und verwendeten Frameworks unterscheiden und eine Gesamtintegration eine große, technologische Herausforderung darstellt.

Aufgrund der Tatsache, dass fast alle Software-Organisationen auf individuelle Tool-Einzellösungen setzen, ergeben sich oft folgende Probleme:

- Kaum jemand hat mehr den Überblick, wie die Prozesse im Detail wirklich laufen.
- Die heterogene Tool-Landschaft ist schwierig zu administrieren.
- Es treten oft Daten-Redundanzen und mit ihnen Daten-Inkonsistenzen auf.
- Die mangelnde Integration hat einen erhöhten Aufwand bei den Nutzern der Systeme zur Folge, da Prozess-Brüche vorhanden sind und Daten teilweise doppelt eingegeben werden müssen.
- Die Nutzer des Systems haben mangelnden Überblick über die Gesamtheit der Daten und Prozesse im System. Oft werden deshalb wichtige Daten nur isoliert ausgewertet. Manchmal sind wichtige Daten aufgrund fehlender Integration gleich von der Datenanalyse und von der täglichen Einbindung in die Prozesse ausgeschlos-
- Die losen Verknüpfungen zwischen den Werkzeugen macht eine Automatisierung von Abläufen innerhalb des Software-Entwicklungsprozesses sehr schwer bis unmöglich.

Die Idee von SPRINT

Um bereits vorhandene bzw. etablierte sowie zukünftige Werkzeuge in einer Organisation auf möglichst einfache Art und flexibel miteinander zu integrieren, ist es sinnvoll, bereits seit Jahren etablierte und als Standard-Produkt am Markt verfügbare Werkzeuge aus dem Workflow- und Enterprise Content Integration-Umfeld einzusetzen. Diese ermöglichen eine werkzeuggestützte Integration von verschiedens-

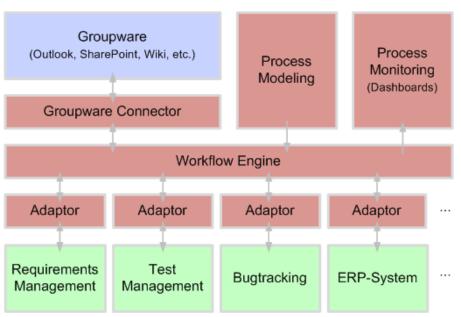


Abb.1 - Der schematische Aufbau des SPRINT Frameworks







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 Lettern 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 Forschungs-ergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com

