

Integrationen und Schnittstellen – "Wie es Euch gefällt"

(Link zum Artikel: <http://www.it-republik.de/dotnet/artikel/3135>)

Eine Collaboration-Plattform muss einen aktiven Nutzen für jedes Teammitglied, vor allem für die Entwickler, haben, um nicht in der Ecke zu verstauben und als lästiger "Mehraufwand" empfunden zu werden. Dazu ist die nahtlose Anbindung zwischen Entwicklungsumgebung (IDE) und Plattform auf allen Ebenen unabdingbar.

Die IDE ist die persönliche Entwicklungsumgebung eines jeden Programmierers. Diese Entwicklungsumgebung ist so eingerichtet, dass er effizient arbeiten kann. Es ist sozusagen sein persönliches Heiligtum. Es gibt eine Vielzahl von Softwareprodukten, kommerziell oder Open Source, mit denen, je nach persönlicher Vorliebe des Entwicklers, diese bestückt werden kann. Eine sehr beliebte IDE bei Java-Programmierern ist z. B. das Open-Source-Produkt Eclipse. Eine optimal eingerichtete Entwicklungsumgebung beschleunigt den Programmierprozess. Es gilt also, diese sehr private Sphäre ständig zu optimieren. Dazu ist der Einsatz der besten Tools erforderlich.

Die Einführung einer Collaboration-Plattform, zusätzlich zur IDE, bedeutet für den Entwickler auf den ersten Blick einen Mehraufwand. Einarbeitung ist erforderlich. Es besteht zu Recht die Sorge, dass dadurch für Implementierung jedes Features ein paar Minuten für Buchung und Verwaltung der Aufgaben zu investieren sind. Um dies zu minimieren, muss die Plattform nahtlos an die IDE angebunden sein. Hierfür ist ein ständiges Lernen und Optimieren notwendig. Die Anbindung von verschiedenen Clients, mit denen Daten direkt in die Plattform eingetragen werden können, ist ein wichtiges Kriterium für die Akzeptanz der Werkzeuge bei den Entwicklern. Wenn es als lästige Aufgabe empfunden wird, den letzten Entwicklungsstand eines Bugs im Bug Tracker zu dokumentieren, dann ist die Collaboration-Plattform nicht lange auf dem neuesten Stand und Daten und Metriken sind wertlos. Es bricht ein Kampf zwischen dem Projektleiter und den Entwicklern aus.

Jede gute Collaboration-Plattform stellt ein Application Programming Interface (API) zur Verfügung, mit dem Zusatzprogramme auf die Daten ihrer Datenbank zugreifen können. So können nicht nur Tools von anderen Herstellern angebunden werden, sondern auch eigene Programme für die Erweiterung der Plattform erstellt werden. Mithilfe einem guten API kann fast jedes externe Tool eingebunden werden. Das schafft Freunde. Auf diesem Wege stehen für die Verbindung von IDE und Collaboration-Plattform aller bekannten Hersteller Produkte kostenlos zur Verfügung.

Das Wichtigste ist die Anbindung des Sourcecode-Archivs an die IDE. Ein Check-in des Sourcecodes sollte direkt von der IDE aus möglich sein. Eclipse und Netbeans sind zum Beispiel weitverbreitete Open Source IDEs. Subversion ist ein bekanntes Sourcecode-Archiv, das von vielen Plattformen verwendet wird. Für Eclipse gibt es Plug-ins wie Subclipse (Tigris) oder Subversive (Polarion), die als Open-Source-Produkte zur Verfügung stehen. Das ist schon die halbe Miete.

Aufgabenorientiertes Arbeiten ist dann möglich, wenn der Entwickler auf Tracker wie zum Beispiel den Bug Tracker zugreifen kann. Plug-ins, wie Mylyn für Eclipse, erlauben den direkten Zugriff auf Workitems in verschiedenen Plattformen wie z. B. Polarion, Collab.net

und Codebeamer. So kann der Entwickler, ohne die Umgebung zu wechseln, den Status seiner erledigten Aufgaben verändern und sich mit neuen Aufgaben versehen. Das spart Zeit.

Der für Entwickler relevante Teil der IBM-Collaboration-Plattform JAZZ/RTC ist direkt mit Eclipse -Technologie umgesetzt und ein Musterbeispiel der Integration von IDE und Plattform. So ist auch die Anbindung an alle IBM Rational Tools gesichert.

Projekte, bei denen die Entwickler im .NET-Umfeld von Microsoft arbeiten, brauchen eine gute Anbindung an die Microsoft-Welt. Visual Studio und Microsoft Office sollten nahtlos an die Collaboration-Plattform angebunden werden. Hier empfiehlt es sich, den Team Foundation Server (TFS) von Microsoft zu verwenden. Es gibt viele Produkte von unabhängigen Herstellern, die eine Integration in Produkte wie MS Word und MS Office produzieren. MS-Project-Daten können in Workitems des TFS umgewandelt werden.

Uta Kapp arbeitet als freiberufliche IT-Beraterin und systemischer Coach. Mit einer Kombination aus Fach- und Prozessberatung für Softwareprojekte hilft sie Entwicklungsteams bei der Bewältigung der ständig steigenden Komplexität. Hier kommen die agile Softwareentwicklungsmethode Scrum und Organisationsaufstellungen zum Einsatz. Der Einsatz von Collaboration-Plattformen ist ein weiterer Schwerpunkt.

Jean Pierre Berchez ist Geschäftsführer der [HLSC UG](#) und beschäftigt sich seit mehr als 15 Jahren mit den Themen Projektmanagement, Software Engineering und objektorientierte Softwareentwicklung. In den letzten Jahren liegt sein Interesse insbesondere auf den Themengebieten agile Entwicklung mit Schwerpunkt Scrum sowie "Collaborative" Software Development. Neben seiner Tätigkeit als Geschäftsführer ist Jean Pierre Berchez auch als Lehrbeauftragter an den BAs Stuttgart und Heidenheim sowie an der Hochschule Liechtenstein für die Themen "Anforderungsmanagement", "Scrum" und "Collaborative Software Development" tätig. Er organisiert unter anderem den Community-Event "[Scrum-Day](#)" in Deutschland.

Original Artikel auf:

<http://it-republik.de/dotnet/artikel/Integrationen-und-Schnittstellen-%96-Wie-es-Euch-gefaellt-3135.html>

andere Artikel dieser Serie

- [Kollaborieren mit Plattform](#)
- [Bughunting](#)
- [Was macht ein Open Source-Projekt erfolgreich?](#)
- [Metriken, Statistiken, Dashboards](#)
- [Kommunikation zwischen Entwicklern, Benutzern und Management](#)
- [Wikinomics](#)
- [Kollabieren oder Kollaborieren](#)
- [Integration statt Brückenbau](#)
- ["Projekt Gold" oder: Der Weg führt zum Ziel](#)