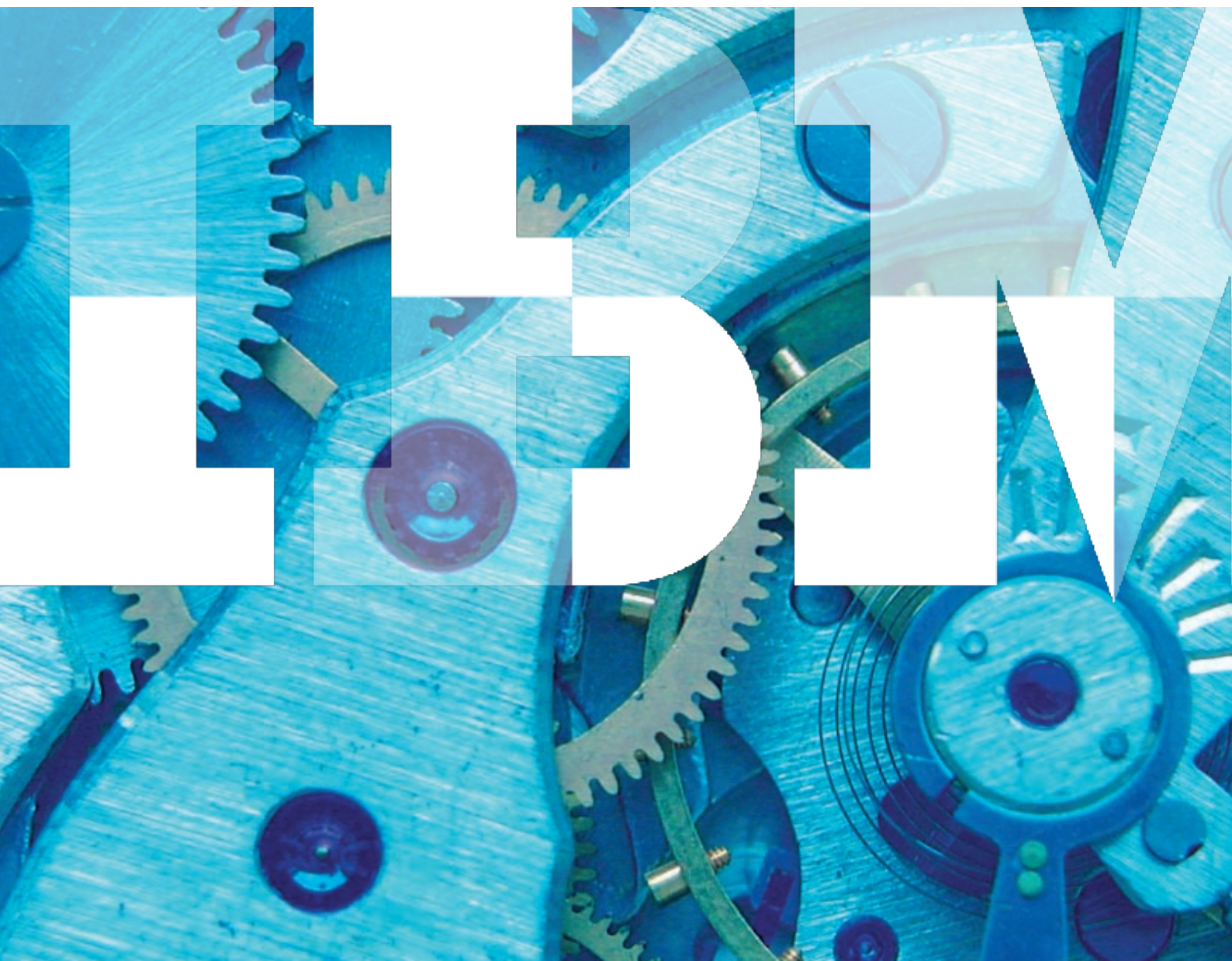


Interdisziplinäre Software-Entwicklung

oder ‚es wächst zusammen was zusammen gehört‘



Über viele Jahre war die Entwicklung von Software geprägt von Individualisten: kreative Köpfe und hochspezialisierte Teams entwarfen und implementierten intelligente Softwarelösungen für spezifische einzelne Anwendungen. Der Umfang dieser Anwendungen wurde schnell größer, die prinzipielle Herangehensweise änderte sich lange Zeit nur wenig: einzelne Teams konzentrierten sich auf die Lösung einer klar umgrenzten Aufgabe, und die Integration verschiedener Module zu einem funktionierenden System wurde als nachgelagerte Aktivität angesehen. Die resultierenden Probleme sind bekannt.

Integrationstests entwickelten sich zu einem signifikanten Zeit- und Kostenfaktor, der unglücklicherweise bei der Projektplanung kaum realistisch einzuschätzen war. Es dauerte eine Weile bis sich die Erkenntnis durchsetzte, dass das Problem weder durch Ignorieren noch durch straffes Projektmanagement und mehr Ressourcen für die Integrationstests aus der Welt verschwand. In der Folge wurden neue Methoden für einen ganzheitlichen Entwurf zusammengesetzter Systeme unter besonderer Berücksichtigung der Schnittstellen entwickelt, welche die Risiken der Integration deutlich minimieren.

Heute steht die Entwicklung an einem ähnlichen Wendepunkt: Die heutige Hardware und Software erschließt nahezu täglich neue Anwendungsbereiche; wir sind in der Lage, Software zu erstellen und zu implementieren, die noch vor wenigen Jahren undenkbar war und wir können Geräte mit softwaregesteuerten Funktionen ausstatten, die noch vor kurzem in unserer Vorstellung nicht einmal existierten. Webbasierte Dienste werden laufend erweitert, ergänzt oder neu entwickelt, und die Rechenleistung der ca. 50 Prozessoren, die heute in einem Mittelklassewagen versteckt ist, beanspruchte früher ein Mehrfaches des Platzbedarfs des ganzen Autos. Die Vielzahl unterschiedlicher Anwendungsdomänen und die preiswert zur Verfügung stehenden Prozessoren und Speicher haben zu einer ausgesprochen vielfältigen Landschaft an Software- und Architekturframeworks geführt. Traditionelle, isoliert entwickelte Anwendungen, die nicht in diese neueren Frameworks integriert werden können, haben außer ihrem eingeschränkten Anwendungsbereich einen weiteren entscheidenden Nachteil: Sie führen in eine Sackgasse, an deren Ende eine Weiterentwicklung teurer wäre als eine Neuentwicklung im Kontext des gewählten Frameworks. Die Softwareentwicklung ist geprägt von einer rasant verlaufenden Evolution, in der man leicht Gefahr läuft, fundierte Erkenntnisse über Wirtschaftlichkeit, Integration und Qualitätsmanagement leicht aus dem Blick zu verlieren.

“Um klar zu sehen, genügt ein Wechsel der Blickrichtung.”

– Antoine de Saint-Exupéry

Schaut man sich die Charakteristika traditioneller Fertigungsunternehmen an, stellt man fest, dass präzise definierte interdisziplinäre Abläufe, eindeutige Schnittstellen zwischen Expertenteams und Unternehmensbereichen sowie bewährte Architekturen als Rahmenwerk für die Konstruktion unabdingbar sind für langfristigen Erfolg und qualitativ hochwertige Produkte. Die Entwicklung und Bereitstellung von Software wird zunehmend geprägt von Hersteller/Zuliefererkonstellationen, heterogenen Systemen, vernetzten Funktionen und Architektur-Frameworks. Daher gibt es keinen Grund anzunehmen, dass hier andere Regeln gelten!

In dem Maße, wie Software die Unternehmensbereiche durchdringt, wird die Zusammenarbeit aller Unternehmensbereiche immer wichtiger. Die Dienste und Funktionen einer Internet-Plattform für den Einzelhandel bestehen nicht nur aus dem, was vordergründig auf der Ebene der Interaktion mit dem Kunden sichtbar ist. Den weitaus größeren Teil macht die dahinter liegende Einbindung der Oberflächenfunktionen in den operativen Betrieb aus. Ein vollständiger Bestellvorgang setzt sich zusammen aus einzelnen Funktionen wie Auftragsannahme, Prüfung der Verfügbarkeit, Auftragsbestätigung, Zuordnung der Bestellung zu einem bestimmten Auslieferungslager, ggfs. Kontaktaufnahme mit dem Lieferanten und Nachbestellung, Kommunikation mit dem Spediteur, und möglicherweise noch weitere, je nach Branche.

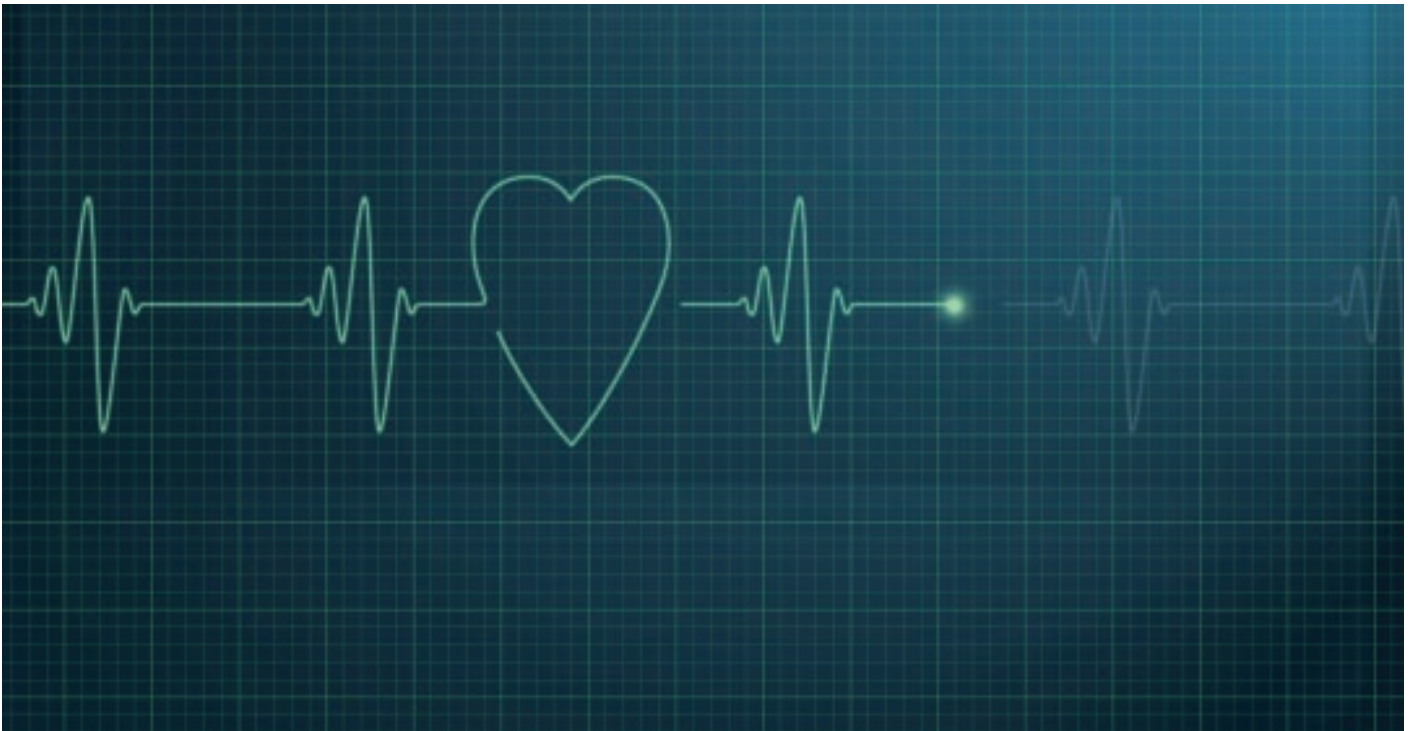
Die Entwicklung von Software für ein Fahrzeug ist heute keineswegs eine isolierte Teilaufgabe für einen einzelnen Experten. Hier müssen in den Teams Spezialisten aus allen Unternehmens- und Fachbereichen zusammenarbeiten, wie Elektronik, Mechanik, Mechatronik, für Antrieb, Chassis, Infotainment, aber auch Marketing, Produktplanung, Produktion und Logistik sowie Vertriebsorganisation, um nur einige zu nennen. Eine weitere Dimension der Interaktion ergibt sich aus der stark vermaschten Hersteller/Zuliefererstruktur in der Automobilindustrie.

Aus diesen Beispielen wird deutlich, dass die Entwicklung und Bereitstellung von Software heute eine unternehmensweite Aktivität ist, die weit mehr umfasst als das klassische Thema ‚Entwurf und Implementierung‘. Es geht um einen ganzheitlichen Blick auf das Projekt, sowohl horizontal – von der Idee bis zum Auslaufen des Produktes oder Abschalten des Dienstes – als auch vertikal – in jeder Phase verzahnt mit allen involvierten Unternehmensbereichen und Projektpartnern.

Entwicklung als Geschäftsprozess

Nachhaltiger geschäftlicher Erfolg von Softwareentwicklungsprojekten setzt daher voraus, dass Softwareentwicklung selbst als Geschäftsprozess begriffen und gestaltet wird. Aus dieser Perspektive ergeben sich einige Schlussfolgerungen für erfolgreiche Entwicklung:

- Die Steuerung von Projekten ist Bestandteil der Corporate Governance .
- Präzise definierte und lebbare Entwicklungsprozesse entlasten von fehleranfälligen Routineaufgaben.
- Einbindung der Entwicklung in das operative Umfeld des Unternehmens sorgt dafür, dass IT und Entwicklung im Einklang mit den Unternehmenszielen agieren.
- Die interdisziplinäre Zusammenarbeit reduziert Risiken aufgrund mangelnder Kommunikation.
- Eine geeignete unternehmensübergreifende Plattform ist notwendig um die Ausführung der Prozesse und die Zusammenarbeit zu erleichtern.



Softwareentwicklung ist heute hochkomplexe Präzisionsarbeit – winzige Fehler können Riesenprojekte zum Scheitern bringen. Gleichzeitig spielen beim Entwicklungsprozess immer mehr geschäftskritische Faktoren eine Rolle, wie schnellere Innovationszyklen, Compliance-Anforderungen, global verteilte und interdisziplinäre Teams oder heterogene IT-Landschaften. Unternehmen brauchen daher Werkzeuge, mit denen sie schnell, aber mit kontrolliertem Risiko stabile Software und Systeme entwickeln können. Das bietet IBM Rational.

Die Begriffe ‚Steuerung‘ und ‚Governance‘ werden oft mit der Einhaltung von Vorschriften assoziiert: Einhaltung von Vorschriften des Sarbanes-Oxley-Gesetzes (Sarbanes-Oxley Act) oder des Health Insurance Portability and Accountability Act (HIPAA). Gleichzeitig liegt die Überlegung nahe, welche unerfreulichen Konsequenzen bei Nichteinhaltung drohen. Das Netz der Governance deckt im Bereich der Softwareentwicklung und -bereitstellung jedoch viel mehr ab als nur die Durchsetzung vorhandener rechtlicher Vorgaben. Governance ist nicht nur ein Framework zum Zweck der Aufsicht, sondern ein Satz von Standards, deren Umsetzung ein Softwareteam fördern kann, statt es zu behindern. Zum Beispiel hat das Framework der Projektgovernance, das in den Arbeitsergebnissen des Eclipse-Projekts sichtbar ist, eine der kreativsten und am schnellsten wachsenden Open-Source-Communities maßgeblich gefördert. Die Einhaltung von Vorschriften entsteht dann aus erfolgreicher Governance und nicht aus der Orientierung an den Maßnahmen zur Softwarebereitstellung. Ob C, Java, das Internet, XML, Linux, Open-Source-Software oder SOA – nichts kann ohne Governance existieren. Alle diese Technologien, Architekturen und Frameworks haben sich gerade deshalb so gut entwickelt, weil Standards vorhanden sind, Normen, die dem Verhalten, der Diskussion und der Implementierung Grenzen vorgeben.

Wer Governance zum eigenen Vorteil nutzen möchte, muss sich mit Anderen auf einen Standard, ein Framework oder ein Produkt einigen, um unnötigen Aufwand zu minimieren und die implizite Kontrolle nutzen, um ein Projekt in die richtige Richtung zu lenken. Governance ist ein entscheidender Bestandteil jeder erfolgreichen Softwarebereitstellung oder Systementwicklung. Sie ist ein entscheidender Bestandteil jedes Aspekts bei der Erstellung, Implementierung und Pflege von Software.

Eine wirksame Governance kann durch eine einheitliche Datenstruktur unterstützt werden, die eine gemeinsame, echtzeitfähige Nutzung der Informationen ermöglicht – anwendungs- und plattformübergreifend. Es ist eine Möglichkeit zur Integration der Kommunikation: Projektteammitglieder können im Zusammenhang mit dem aktuellen Projekt und den verwendeten Anwendungen untereinander Nachrichten austauschen – von den Unternehmensanalysten bis zu den Spezialisten für die Produktion. Es ist die Möglichkeit zur Softwareentwicklung in weltweit verteilten Teams und zur nahtlosen Integration dieser Produkte. Es ist die Möglichkeit, staatliche Prüfungen ohne zusätzlichen Aufwand zu bestehen, da die Einhaltung von Vorschriften bereits im Framework enthalten ist. Es ist eine grundlegende Verschiebung von rein rollenorientiertem zu teamorientiertem Denken. Der Schlüssel ist die stärkere Beteiligung der Teams aus allen involvierten Unternehmensbereichen an allen Phasen des Projekts.

Konvergenz der Fachbereiche – „Es wächst zusammen ...“

Der interdisziplinäre Charakter der Entwicklungsprojekte zeigt sich unter anderem darin, dass eine Vielzahl unterschiedlicher Komponenten und Subsysteme am Ende ein lauffähiges Gesamtsystem ergeben müssen, das den definierten Anforderungen entspricht. Dabei sind die einzelnen Teile in der Regel heterogen; je nach Branche gibt es Elektronik-, Software-, Mechatronik-, elektrische und mechanische Komponenten, die von verschiedenen Geschäftsbereichen des Unternehmens oder auch von Zulieferern beigelegt werden. Die Komponenten und Module existieren bereits, oder sie werden neu entwickelt, oder sie werden durch Anpassung oder Weiterentwicklung aus vorhandenen Komponenten hergeleitet.

In dem bereits erwähnten Mittelklassewagen versehen ca. 50 Prozessoren ihren Dienst. Dahinter stehen Entwicklungsmannschaften, die in der ganzen Welt verteilt sind. Trotzdem sollen die verschiedenen Teilsysteme perfekt zusammenarbeiten. Die Anforderungen an die Organisation zur Realisierung solcher Vorgaben sind anspruchsvoll. Firmen wie IBM Rational bieten Softwaresysteme an, die Planer, Projektmanager und Entwickler immer mit aktuellen Daten versorgen, denn in der Regel ändern sich im Laufe der Entwicklung Pflichtenhefte und Anforderungen.

Die Entwicklung industrieller Produkte ist gekennzeichnet durch rasant zunehmende Komplexität. Oft handelt es sich – wie in einem Flugzeug – um ein „System von Systemen“, zusammengesetzt aus vernetzten Subsystemen verschiedener Funktionsbereiche. Beschreibung, Modellierung und Implementierung solcher vernetzten, Domänen übergreifenden Systeme sind mit herkömmlichen Lösungen für Einzeldisziplinen nicht mehr sinnvoll zu bewerkstelligen.

Reibungsverluste können überall dort entstehen, wo Phasen oder Disziplinen des Projektes aneinander angrenzen. Viele Projektrisiken mit hohem Verlustpotential verstecken sich in den Schnittstellen zwischen den Expertenteams, Produktmanagement, Projektmanagement und anderen Projektbeteiligten. Die ersten Missverständnisse entstehen typischerweise bereits bei der konsistenten Aufnahme der Anforderungen, und bei ihrer Weitergabe an die Systemarchitekten oder Softwaredesigner. Mit jeder weiteren Übergabe von Entwicklungsfakten wird die Unschärfe bezogen auf den Zielkorridor größer. Spätestens bei der Systemintegration wird das ganze Ausmaß der Abweichungen und Konsistenzbrüche deutlich. Die Entwickler befassen sich in der Folge mit Ursachenforschung und mühsamen Korrekturen statt ihre Kreativität den nächsten Aufgaben zu widmen.



Als einziger Hersteller bietet IBM Rational ein Portfolio, das den gesamten Entwicklungsprozess abbildet und diesen außerdem um Themen wie Unternehmensarchitektur, Geschäftsprozesse oder strategische Produktplanung erweitert – bis hin zu den Insights für die Chefetage.

Der Vollständigkeit halber sei darauf hingewiesen, dass der Aufwand für die Fehlerkorrektur um ein Vielfaches steigen kann, je näher Fehlererkennung und -korrektur am Projektende und dem Produktionsstart liegen.

Zu diesen Risiken trägt, neben der teilweise unzureichenden Kommunikation zwischen den verschiedenen Teams und Funktionsdomänen, vor allem die isolierte Verwendung von Werkzeugen bei, die sich nicht zu einer gemeinsamen Plattform verbinden lassen. Proprietäre Formate verhindern die leichte Austauschbarkeit der Resultate, und der Prozesskontext fehlt. Gerade eine durchgängige und automatisierte Prozesskonformität trägt in hohem Maße zu einer gesteigerten Effizienz und

Qualität bei der Ausführung bei. Voraussetzung ist die optimal unterstützende Anpassung des Prozesses an die spezifischen Aufgabenstellungen der jeweiligen Entwicklungsteams.

Mit den Softwarewerkzeugen, die oft unter dem Begriff Lifecycle Management zusammengefasst werden, verbindet man die Projektphasen und die beteiligten Teams über das ganze Unternehmen hinweg und stellt eine Art Infrastruktur für den automatischen Austausch von Informationen, für die Synchronisation von Projektdaten über das gesamte Projektteam und für jederzeit aktuelle Reports über den Status des Projektes zur Verfügung.

Gemeinsame Plattform erleichtert die Zusammenarbeit

Ein wichtiger Aspekt für eine optimierte Zusammenarbeit im Team ist eine gemeinsame Plattform, die nicht an Abteilungsgrenzen halt macht und die den kompletten Lebenszyklus der Entwicklung stützt. Punkt-zu-Punkt Schnittstellen zwischen einzelnen Werkzeugen helfen nur begrenzt weiter. Weitaus tragfähiger ist die Bereitstellung einer Datenkonsistenzschicht zur verbesserten Austauschbarkeit und schnellen Verfügbarkeit von Artefakten aus verschiedenen Werkzeugen sowie die Möglichkeit zur Verkopplung mit den Entwicklungsprozessen und der interaktiven Ausführung von bereitgestellten Prozessmodellen.

Damit können Artefakte im Kontext der aktuellen Prozessphase bearbeitet werden und über die Laufzeit der Entwicklung in ihrer Relation mit anderen Artefakten verknüpft werden und dies auch bleiben. Dabei interagieren sie mit den verschiedenen Phasen des Entwicklungsprozesses: der Prozess initiiert die Bearbeitung von Artefakten, gleichzeitig kann aber auch die Statusänderung eines Artefaktes neue Prozessphasen und Workflows auslösen.

Ein Beispiel: Ein Ticketautomat befindet sich in der Entwicklung. Entgegen der ursprünglichen Planung soll der Automat nicht nur Kreditkarten sondern auch Euroscheckkarten zur Zahlung akzeptieren. Dazu formuliert das Marketing den Änderungswunsch und speist ihn ins System ein. Das für solche Entscheidungen zuständige Gremium oder der Produktmanager prüfen den Wunsch und akzeptieren ihn (oder lehnen ab). Die neue Anforderung "Euroscheckkarte" wird erzeugt oder eine existierende Anforderung wird geändert (statt "Kreditkarte" nun "Kreditkarte oder Euroscheckkarte"). Um Missverständnissen vorzubeugen wird eine Notation gewählt, die den Interpretationsspielraum minimiert, dabei können formale Sprache, Graphiken, Tabellen oder Modelle hilfreich sein. Im nächsten Schritt werden die zuständigen Designer und Entwickler automatisch von der Änderung unterrichtet und können sie umsetzen. Die Tester erhalten ebenfalls Kenntnis davon und können geeignete Tests vorbereiten, die die Akzeptanz von Kreditkarten und Euroscheckkarten überprüfen.

Die Abfolge der einzelnen Schritte, die Entscheidungspunkte und die Übergaben sind im Prozess definiert, deren Ausführung durch die Entwicklungsplattform unterstützt wird. Eine explizite Auswahl des Verantwortlichen für den nächsten Schritt liegt nicht in der Verantwortung der einzelnen Teammitglieder sondern wird durch die Plattform sichergestellt.

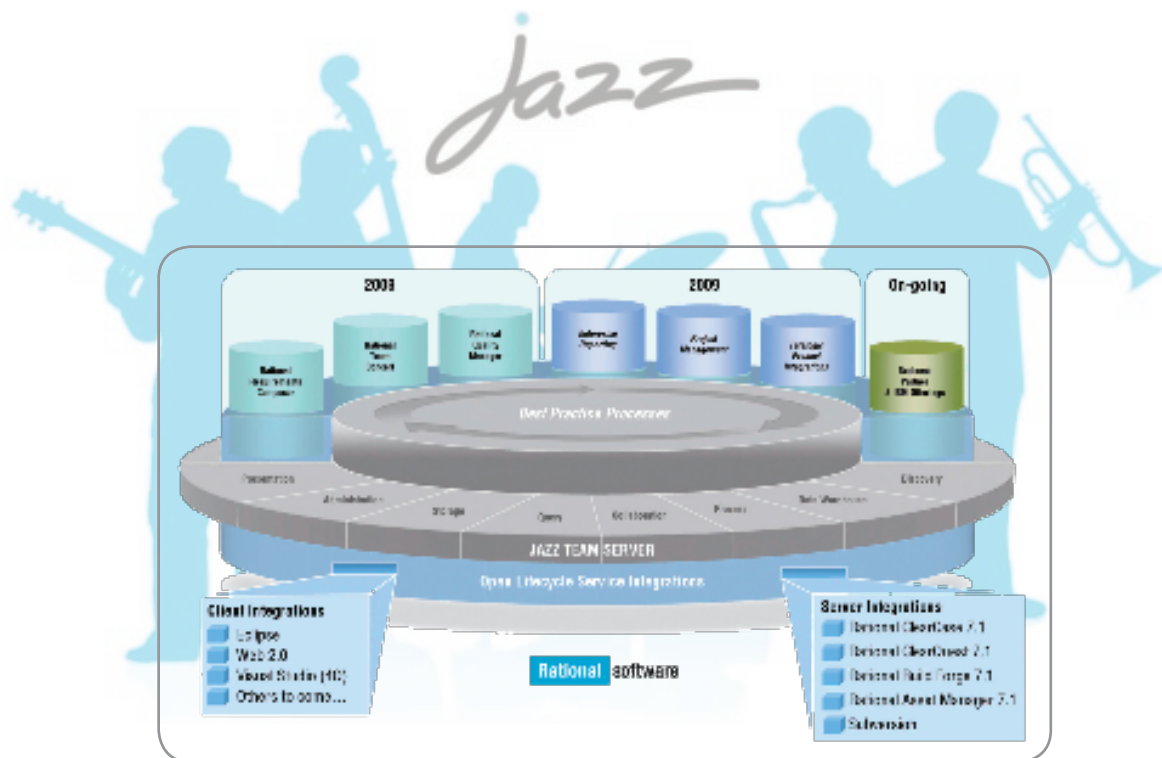
Da alle Daten zentral vorgehalten werden, ist es zum Beispiel für den Projektmanager leicht möglich, sich in Echtzeit über den Fortschritt bei der Implementierung der Änderung zu informieren und zu sehen, welche Tests geplant sind, um die korrekte Ausführung zu verifizieren. Im weiteren Verlauf kann er auf derselben Plattform prüfen, wie die Tests verlaufen sind, und welche Erkenntnisse die Experten der Qualitätssicherung gewonnen haben. Abschließend für diesen Vorgang wird die technische Dokumentation aus den Daten, die im System vorliegen, automatisch generiert.

Entwicklung als unternehmensweiter Geschäftsprozess

Mit dem Jazz-Framework von IBM (www.jazz.net) steht eine Plattform zur Verfügung, welche diese neue Sichtweise von Entwicklung als unternehmensweiter Geschäftsprozess praxistauglich macht. Jazz integriert Teams und deren Aufgaben in den gesamten Bereitstellungszyklus für Software und Systeme. Organisationen, die das Jazz-Framework verwenden, können einschätzen, wie sich die Änderung einer Anforderung auf Builds auswirkt, und können dadurch noch genauer bestimmen, was durch wen geändert werden muss. Mit dieser Integration in den gesamten Lebenszyklus wird eine werkzeuggestützte Prozessführung erreicht, bei der die Werkzeuge selbst den Entwicklungsprozess verstehen, für den sich das Team entschieden hat.

Durch Automatisierung kann mit der Integration in den Lebenszyklus sichergestellt werden, dass alle Teammitglieder dem Prozess jederzeit folgen können ohne selbst zum Experten für einzelne Teilaufgaben werden zu müssen. Um den vollen Nutzen aus einer teamorientierten Governanceplattform für den gesamten Lebenszyklus ziehen zu können, müssen die Organisationen außerdem flexible modulare Komponenten auf der Basis offener Standards bereitstellen, die auf der Plattform ausgeführt werden. Das Eine geht nicht ohne das Andere. Dekomposition und Aufteilung der vorhandenen Merkmale und Funktionen in Komponenten sind parallele und gleichermaßen wichtige Arbeitsaufgaben.

Diese Technologie ist skalierbar für Entwicklungsumgebungen jeglicher Größe, von sehr kleinen Teams bis zu großen, dezentralen Organisationen. Die Organisationen haben die Möglichkeit, Werkzeuge und Technologien für die Softwareentwicklung und -bereitstellung nach Bedarf zusammenzustellen, um in jeder Phase der Transformation ihrer Entwicklungsumgebungen die optimale Lösung zu erreichen.



Die Idee der Jazz-Plattform ist es, Entwicklern und Projektmanagern ein Instrumentarium an die Hand zu geben, mit dem sie ihre Arbeit genau so ungezwungen und kreativ verbinden können, wie die Solisten einer Jazz-Band.

Offene Standards wie die Eclipse Plattform leisten weitreichende Unterstützung für die Werkzeug- und Prozessintegration. Dabei macht die Kollaborationsplattform Jazz genau dort weiter, wo Eclipse seine Grenzen hat: bei der Einbeziehung des Teams und der Koordination von Best Practices und prozessbedingten Abläufen, über die lokale Integration auf der Workstation hinaus.

Der Plattformansatz von Jazz basiert auf der Open Source Technologie Eclipse und etabliert ein Client-Server Konzept. Damit wird der nächste Level der Eclipse Technologie-Ansätze erreicht. Ein erstes Produkt basierend auf Jazz wird von IBM unter dem Produktnamen Rational Team Concert (RTC) ausgeliefert.

In dieser automatisierten Prozessumgebung werden Arbeitspakete zugeordnet und verwaltet, mit einer weitgehend automatisierten Versionskontrolle und Konfigurationsmanagement, sowie einer derzeit auf Softwareentwicklung abgestimmten Projektmanagement-Funktionalität. Basisfunktionen für Änderungsmanagement und Fehlerverfolgung sind ebenfalls enthalten.

Wie oben postuliert, wird in dieser Umgebung das Team in der Ausführung der Prozesse transparent unterstützt und der Austausch projektrelevanter Informationen wird über die gemeinsamen Datenstrukturen sicher gestellt. Das Ziel hierbei ist, die Kreativität im Team weitgehend unbeeinflusst zu lassen. Vordefinierte Prozesse (z.B. Eclipse Way, Scrum, Open Up) werden mitgeliefert, die flexibel auf die konkrete Projektsituation abgestimmt werden können und sogar die Differenzierung nach Arbeitsgruppen innerhalb eines Teams erlauben.

Die Entwicklung der Jazz Plattform selbst ist über die Web-community auf www.jazz.net in allen Phasen offengelegt und lädt zur Interaktion sowie zur Artikulation von weitergehenden Anforderungen ein. Für die Entwicklung der Jazz Implementierung Rational Team Concert wurde die Technologieplattform selbst bereits im frühen Prototypenstadium verwendet (self hosting development).

Ausblick

Die Jazz Kollaborationsplattform ist geeignet, den Ansprüchen der moderner Softwareentwicklung gerecht zu werden. Sie ermöglicht die Realisierung einer durchgängigen, prozessgestützten Entwicklungsumgebung, die alle Aspekte einer Produktentwicklung adressiert. Alle Entwicklungsartefakte werden durchgängig verknüpft und konsistent, ohne Datenbrüche von den ersten Anforderungen und Konzeptbeschreibungen eines Systems über die Detailumsetzung bis zur Testphase weitergereicht. Dies ermöglicht eine umfassende Traceability.

Das Team kommuniziert über die Fachbereichs- und sogar Unternehmensgrenzen hinweg und wird durch einen hohen Grad an Automatisierung in der Prozessanwendung unterstützt. Dies steigert die Effizienz der Projekte, und die Kosten sowie Time-to-Market werden reduziert bei höherer Qualität der Ergebnisse.

Auf dieser Plattform kommen talentierte Entwickler zusammen mit Experten anderer Unternehmensbereiche sowie Best-in-Class Werkzeugen und erprobten Best Practices – eine Kombination, die alle Voraussetzungen für erfolgreiche Entwicklung und Bereitstellung von Software erfüllt.

Über die Autorin

Dipl.-Inform. Renate Stücka ist als Senior Program Manager bei IBM Rational für Marketing und Kommunikation mit Schwerpunkt für komplexe Systeme und Software verantwortlich. Sie kam im Rahmen der Akquisition von Telelogic zu IBM. In vorherigen Positionen bei Telelogic war sie verantwortlich für Marketing, Business Development, Partnermanagement und Kommunikation.

Renate Stücka

Senior Program Manager Rational Geo Marketing,
EMEA, IBM Deutschland

E-Mail: renate.stuecka@de.ibm.com



IBM Deutschland GmbH
IBM-Allee 1
71139 Ehningen
ibm.com/de

IBM Österreich
Obere Donaustrasse 95
1020 Wien
ibm.com/at

IBM Schweiz
Vulkanstrasse 106
8010 Zürich
ibm.com/ch

Die IBM Homepage finden Sie unter:
ibm.com

IBM, das IBM Logo, ibm.com und Rational sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite „Copyright and trademark information“ unter ibm.com/legal/copytrade.shtml

Die vorliegende Veröffentlichung dient ausschließlich der allgemeinen Information.

Vertragsbedingungen und Preise erhalten Sie bei den IBM Geschäftsstellen und/oder den IBM Business Partnern. Die Produktinformationen geben den derzeitigen Stand wieder. Gegenstand und Umfang der Leistungen bestimmen sich ausschließlich nach den jeweiligen Verträgen.

© Copyright IBM Corporation 2009
Alle Rechte vorbehalten.
