

Collaborative Software Development im praktischen Einsatz

von Frank Sterkmann

Entwicklung global

Bei räumlich verteilten Teams gibt es keinen Platz mehr für individuelle „Hacker“. Die Kommunikation der Teammitglieder, die Verständlichkeit und Qualität der Arbeitsergebnisse sowie die Einhaltung definierter Prozesse gewinnen höchste Priorität in der Zusammenarbeit. Diese Art der Softwareentwicklung stellt das Management vor neue Herausforderungen. Die Steuerung geht weit über die internetbasierte Verwaltung und Versionierung von Sourcecode hinaus. Wie man mit einem erweiterten Fokus auf internetbasierte Kommunikation, Prozesse, Controlling und Verwaltung von Skills erfolgreiche Softwareprojekte realisieren kann, wird im Folgenden exemplarisch dargestellt.



Die Softwareentwicklung ist ein interessantes Beispiel für die Dualität von Kommunikation und Konzentration. Auf der einen Seite ist die Kommunikation zum Austausch von Informationen mit den Teammitgliedern erfolgsentscheidend, auf der anderen Seite ist die Konzentration auf die eigentliche Softwareproduktion (dieser Begriff deckt den ganzen Softwareerstellungsprozess von Analyse, Design, Implementierung, Integration und Test ab) die Basis für ein fertiges Softwareprodukt.

Da die Softwareentwicklung unbestritten eine kreative Tätigkeit ist, die hohe Konzentration erfordert, sollten Maßnahmen zur Erhöhung der Produktivität

in diesem Bereich ansetzen. Störungen und andere Nebentätigkeiten müssen bekanntlich auf ein Mindestmaß reduziert werden und die persönliche Konzentration der Mitarbeiter auf die Softwareproduktion muss gefördert werden. Wenn man dieses Thema konsequent weiterverfolgt, wird man schnell bemerken, dass die Produktivität von vielen verschiedenen individuellen Faktoren abhängen kann.

Warum verteilte Teams?

Es werden viele Wünsche nach individuellen, unabhängigen, autarken Arbeitsplätzen auf uns zukommen. Wün-

sche nach Arbeitsplätzen, die ohne Reisezeiten erreicht werden können. Da dies durch unsere mobile Gesellschaft vorangetrieben wird, wird dies sicherlich zu immer mehr verteilten Teams führen. Die breite Verfügbarkeit des Internets bietet uns heute optimale technische Möglichkeiten, die Softwareentwicklung in verteilten Teams durchzuführen. Die allseits vorhandenen schnellen Anschlüsse bewirken das ihre.

Aus der Sicht von Software- und Dienstleistungsunternehmen gibt es noch einen weiteren Grund, der verstärkt zu verteilten Teams führen wird. Dies ist die intensivere Zusammenarbeit mit Kunden

und Sublieferanten. Um die Vorteile der objektorientierten Softwareentwicklung nutzen zu können, ist die Einbindung aller Stakeholder in das Projektgeschehen ein wesentlicher Erfolgsfaktor. Die Vorteile und Probleme sind mit denen der Heimarbeitplätze vergleichbar. Die Fähigkeit der optimalen Unterstützung von verteilten Teams stellt für Software- und Dienstleistungsunternehmen zukünftig einen wichtigen Wettbewerbsfaktor dar.

Die Schaffung der technischen Voraussetzungen ist für die Realisierung erfolgreicher Softwareprojekte bei weitem nicht ausreichend. Es müssen Werkzeuge und Vorgehensweisen entwickelt werden, die auch über die Softwareentwicklung hinausgehende Bedürfnisse erfüllen. Diese sind übers Internet nutzbar und erfordern keinen Installationsaufwand auf dem gerade genutzten Arbeitsplatz, um die Vorteile einer räumlichen Verteilung nicht den Nachteilen zu opfern.

Die Voraussetzung: Collaborative Project

Aus Sicht des Projektverantwortlichen stellen verteilte Teams eine große Herausforderung dar. Er hat die ehrenvolle Aufgabe den Erfolg des Projektes zu gewährleisten und muss mit allen Mitteln darauf hin wirken, dass die Vorteile der verteilten Entwicklungsteams nicht durch andere Nachteile aufgeessen werden. Wie soll er mit den Teammitgliedern oder Entwicklungsgruppen kommunizieren? Wie sollen die Teammitglieder untereinander kommunizieren? Wie soll er die Aufgaben aufbereiten und verteilen? Wie soll er die Arbeitsergebnisse überwachen und wie soll er die Softwareproduktion koordinieren?

Viele Hersteller von Werkzeugen für die Softwareentwicklung haben den Trend der Zeit erkannt und bieten heute Werkzeuge an, die eine Zusammenarbeit von verteilten Softwareteams über das Internet ermöglichen sollen. Collaborative Software Development ist dabei das Marketing-Stichwort. Wenn man das Wörterbuch nach der Übersetzung dieses Begriffes befragt, erhält man die erwartete positive Interpretation wie „mitarbeiten“, „zusammenarbeiten“ oder „behilflich sein“. Im politischen Sinne heißt es

aber, mit dem Feind zusammenzuarbeiten und genießt nicht gerade positives Ansehen.

Instinktiv haben wir es wahrscheinlich alle gedacht: Wir öffnen uns nach außen, zu unseren Kunden, externen Mitarbeitern und Sublieferanten. Wir sind auf einen Schlag sehr transparent und verletzlich. Wie wollen wir sicherstellen, dass unsere Daten und Arbeitsergebnisse nicht missbräuchlich verwendet werden? Wie können wir sicherstellen, dass bei der Übertragung von vertraulichen Daten keine Sicherheitslücken entstehen? Mit diesen objektiven und teilweise subjektiven Themen müssen wir uns bei der verteilten Softwareentwicklung beschäftigen.

Die Voraussetzungen für Softwareprojekte sind der gemeinsame Zugriff und die Verwaltung von Sourcen und Informationen. Dies trifft für alle Projekte zu, stellt aber für verteilte Projekte eine erste Herausforderung dar. Für diejenigen unter uns, die ein professionelles Projekt realisieren wollen, kommt hier zwingend die Versionierung der Sourcen und Dokumente hinzu. Die Werkzeuge wie CVS, PVCS, MKS, ClearCase, Continuum etc. bieten umfangreiche Funktionalitäten an, die mit webbasierten Frontends eine kom-

fortable Verwaltung und Versionierung von Projektdokumenten ermöglichen.

Spannender wird es, wenn wir die weiteren Funktionalitäten wie Organisation, Kommunikation und Qualitätssicherung abdecken wollen. Um diese Funktionalität für verteilte Projektteams abzudecken, stehen nicht allzu viele Werkzeuge zur Verfügung, am bekanntesten mag wohl SourceForge sein (siehe unser Beitrag in dieser Ausgabe), aber auch Borland hat jetzt mit seinem TeamSource DSP erste Gehversuche in dieser Richtung unternommen. SourceForge deckt sämtliche Funktionsbereiche, bis auf die Qualitätssicherung und Suchmaschine, ab. Ein größeres Problem stellt für Softwareunternehmen die Zutrittsberechtigung dar. Da die Kunden nicht sehr begeistert sein werden, wenn ihre strategisch wichtigen Projekte als Open Source öffentlich zur Verfügung stehen, muss eine dezidierte Zutrittskontrolle her, die eine Registrierung der Nutzer pro Projekt ermöglicht, bevor die internen Informationen der Projekte angezeigt werden. Dies soll laut SourceForge mit der so genannten Professional Edition abgedeckt sein.

Eine interessante Alternative stellt das in Deutschland entwickelte Produkt CodeBeamer dar. CodeBeamer kommt ur-

Anforderungen an Collaborative Development Tools

Voraussetzungen:

- Verwaltung von Rechten und Anwendern (User, Accounts)
- Vergabe der Zugriffsrechte pro Projekt
- Datensicherheit bei der Übertragung im Internet (SSL)
- Zentrale Sourcecode- und Dateiverwaltung
- Versionierung

Organisation:

- Verwaltung von Fehlermeldungen (Bugtracking)
- Verwaltung von Änderungsanfragen (Change Requests)
- Verwaltung von Supportanfragen (Support Requests)
- Verwaltung von Projektanforderungen (Requirements-Management)
- Verwaltung von Aufgaben (Task Management)

Kommunikation:

- Kommunikation, eMail-Verteiler, Chat, News
- Notification bei Änderung oder Neuanlage von Aufgaben

Qualitätssicherung und Suchmaschine:

- QS-Analysen und Metriken
- Statistiken und grafische Darstellungen (HTML & Office Export)
- Suchmaschine für Sourcecode (Cross-Referencing, Grep, Powersearch)
- Abhängigkeiten (Dependencies) und Impact Analyse

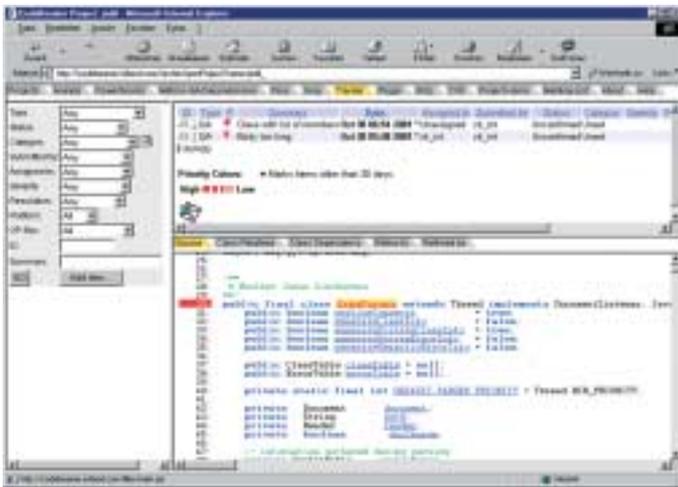


Abb. 1: Issue Tracker mit direktem Link zum Sourcecode

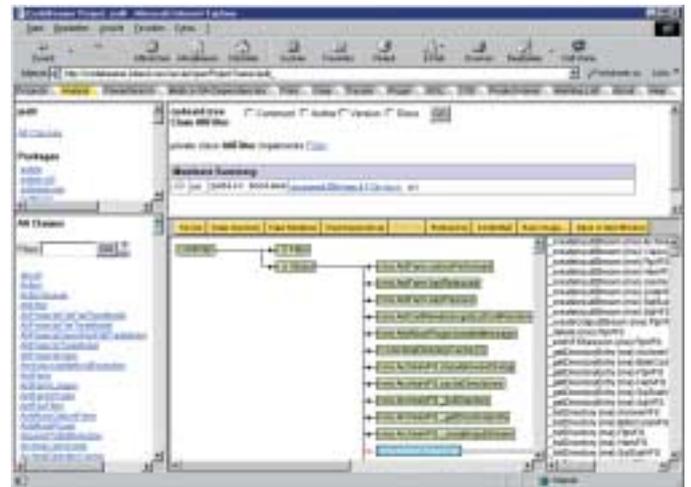


Abb. 2: Grafische Darstellung des Sourcecodes mit Cross Referenzen

sprünglich aus dem Bereich der Qualitätssicherung und Sourcecode-Analyse. Es wurde in der neuesten Version um die Funktionalitäten in den Bereichen Organisation und Kommunikation erweitert und stellt heute ein leistungsfähiges Collaborative Development Portal dar. Der Leistungsumfang der Bereiche Organisation und Kommunikation ist mit SourceForge vergleichbar, stellt aber nur einen kleinen Teil des Funktionsumfangs des Produktes dar.

Beim Anlegen eines neuen Projekts kann der Administrator entscheiden, ob es ein öffentliches oder vertrauliches Projekt sein soll. Die öffentlichen Projekte sind ohne Registrierung übers Internet sichtbar. Erst wenn aktiv Informationen in das Projekt eingebracht werden sollen, muss eine Registrierung erfolgen, was natürlich in jedem Falle sinnvoll ist, um die neuen Informationen und evtl. Änderungen einem Nutzer zuordnen zu können.

Außerdem können beim Anlegen eines Projektes der Ort und die Art der Quelle angegeben werden. So können z. B. CVS-verwaltete Projekte für die zukünftige Bearbeitung direkt in CodeBeamer importiert werden. Während das Projekt angelegt wird, analysiert ein Parser den vorhandenen Sourcecode (gemischter Java, C++ und C Sourcecode möglich) und baut ein umfassendes Repository mit allen Symbolen wie Klassen, Methoden, Attributen und deren Abhängigkeiten auf. Dieses Repository wird bei Änderungen des

Sourcecodes automatisch im Hintergrund aktualisiert, ohne dass der Nutzer etwas davon bemerkt. Dieser stets aktuell aufbereitete Sourcecode bringt uns im weiteren Verlauf des Softwareprojektes sehr viele Vorteile.

So können wir mit dem Tracker einen Fehler mit dem Typ *Bug* neu anlegen und die zugehörige Information und verantwortlichen Personen eintragen. Zusätzlich können wir einen direkten Link zum Sourcecode eintragen. CodeBeamer ermöglicht die Verknüpfung von Funktionen aus den Bereichen Organisation und Kommunikation mit dem Sourcecode. Somit hat der Verantwortliche sofort die notwendige Information verfügbar, um die Entscheidungen für die Behebung des Fehlers zu treffen. Bei Neuanlage oder Änderungen von Einträgen im Tracker wird per eMail eine Notification an den Verantwortlichen versendet. Diese enthält die Informationen über die Änderungen und ebenfalls den direkten Link. Auch im Sourcecode werden existierende Links als Markierung angezeigt und können zum Anzeigen der zugehörigen Bemerkungen oder Fehlermeldungen genutzt werden.

Wie bereits erwähnt, bringt uns der aufbereitete Sourcecode sehr viele Vorteile. Die sehr schnelle und leistungsfähige Suche nach allen im Sourcecode vorhandenen Symbolen und die grafische Darstellung der Abhängigkeiten (Cross Referenzen) ermöglichen einen schnellen

Überblick über die Software und unterstützen, dass sich Mitarbeiter schnell in einem Projekt zurechtfinden können. Untersuchungen haben ergeben, dass Softwareentwickler im Schnitt 47 Prozent ihrer Arbeitszeit mit der Analyse des Sourcecodes verbringen (Quelle: „Software Quality: Producing Practical, Consistent Software“, Mordecai Ben-Menachem & Garry S., 1997).

Besonders bei verteilten Projekten bringt die grafische Darstellung wertvollen Nutzen, da die Kommunikation zu den anderen Teammitgliedern aufwändiger ist. Somit sind die Mitarbeiter darauf angewiesen, möglichst selbständig mit dem Sourcecode zurecht zu kommen.

Ein wichtiges Feature stellt ein herunterladbares Plug-In dar, welches als Java-Anwendung direkt auf dem Client ausgeführt wird. Dies bringt zwei wichtige Vorteile: Die Datenübertragung über das Internet ist bis heute ein Engpass, so müssen bei interaktiven Analysen wie Browsing und beim Durchhangeln von Dependencies teilweise eine Menge Daten übertragen werden. Das Plug-In holt sich die notwendigen Daten vom Server und übernimmt selbständig die Aufbereitung und Darstellung. Im Intranet ist damit eine Arbeitsweise wie bei einer herkömmlichen Desktop Anwendung möglich. Ebenfalls eingeschränkt ist die Interaktivität bei HTML-Clients. Ein Plug-In ermöglicht eine komfortablere und effizientere Bedienung der Anwendung.

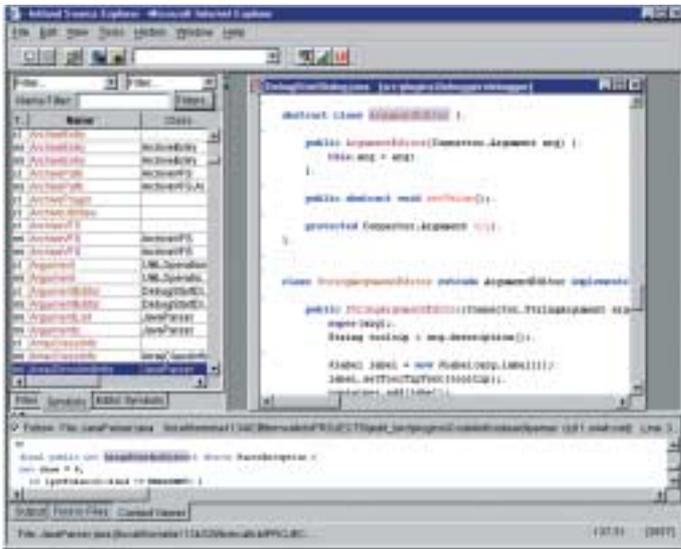


Abb. 3: Plug-In für den CodeBeamer-Client

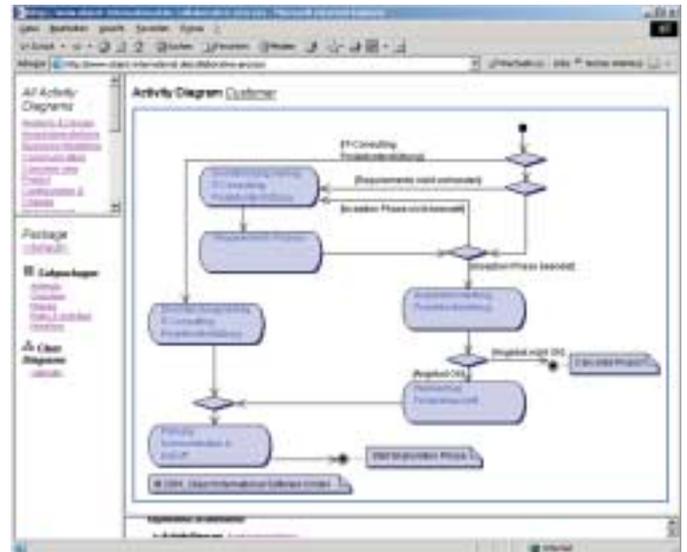


Abb. 4: Softwareentwicklungsprozess im Intranet am Beispiel des „Customer Process“

Die Ergänzung: Collaborative Process

Wesentlich stiefkindlicher wird der Bereich der Softwareentwicklungsprozesse unterstützt. Werkzeuge zur Unterstützung der technischen Belange sind wesentlich ausgereifter und mehr verbreitet. Das Bewusstsein für Softwareentwicklungsprozesse ist teilweise nicht stark ausgeprägt, dabei bergen Prozesse sehr viel Potenzial zur Verbesserung der Softwarequalität. Alle bekannten Zertifizierungs- und Qualitätsmerkmale setzen auf die Wiederholbarkeit von Abläufen und Vorgehensweisen in der Entwicklung.

In verteilten Teams ist eine gemeinsame Vorgehensweise aller Projektmitarbeiter besonders wichtig, da die andauernde individuelle Synchronisation einzelner Mitarbeiter untereinander oft entfällt. Dies kann nur über klar strukturierte und allen bekannten Abläufe ersetzt werden. Ein Softwareentwicklungsprozess dient auch als Wissensbasis für die Mitarbeiter. Er beantwortet Fragen nach Programmierrichtlinien, nach Vorlagen für Protokolle, Kalkulationen, Projektpläne etc. und erleichtert die Arbeitsweise der einzelnen Teammitglieder. Entscheidend bei verteilten Teams ist jedoch der Einsatz der gleichen Abläufe und Vorgaben.

Als optimales Medium für die Abbildung und Publikation eines Entwicklungsprozesses bietet sich ein internetba-

siertes Portal an. Man kann damit gleich zwei Probleme lösen: Erstens steht die Information auch bei verteilten Projekten allen Mitarbeitern zur Verfügung und zweitens ist die Information stets aktuell. Der zweite Punkt ist sehr wichtig, da ein wichtiges Kriterium eines Prozesses die Verbesserung ist. Wenn man erreichen kann, dass alle Mitarbeiter ihre aktuellen Erfahrungen und Vorschläge in den Prozess einfließen lassen, kann damit eine stetige Verbesserung und Wiederholbarkeit der zukünftigen Projekte erreicht werden.

Die Prozesse können z. B. mit Aktivitätsdiagrammen beschrieben werden, wie sie in der Unified Modeling Language (UML) definiert sind. Im abgebildeten Customer Process werden die Aktivitäten beschrieben, die in der Zusammenarbeit mit dem zukünftigen Kunden durchgeführt werden müssen, bevor es überhaupt einen Auftrag zur Realisierung eines Projektes gibt. Diese Abläufe sind für den späteren Verlauf der meisten Projekte erfolgsentscheidend. So wird zum Beispiel geprüft, ob ein Pflichtenheft vorhanden ist. Wenn ja wird die Vollständigkeit des Pflichtenhefts anhand der Ergebnisse der Projektvorbereitungsphase (Inception Phase) überprüft. Sind nicht alle geforderten Ergebnisse vorhanden, werden diese als Dienstleistungsauftrag erarbeitet. Erst wenn alle geforderten Ergebnisse vorhanden sind, wird die Erstel-

lung eines Festpreisangebotes begonnen. Die Aktivitäten der Prozesse werden in einer nächsten Stufe detaillierter als Workflow Detail dargestellt (siehe Abb. 5).

In den Workflow-Details werden die einzelnen Aktivitäten und die verantwortlichen Rollen dargestellt. Ferner müssen die Ergebnisse anderer Aktivitäten vorhanden sein, damit die Endergebnisse erstellt werden können. Die Ergebnisse können Dokumente, Diagramme, Datenbankeinträge, Protokolle etc. sein und werden laut, Unified Software Development Process (USDP) als Artifacts bezeichnet.

Am Beispiel wird gezeigt, welche Unterlagen vorhanden sein müssen, um eine Aufwandsschätzung durchzuführen. Diese Aufwandsschätzung ist dann die Voraussetzung für die Angebotserstellung. Der Sinn dieses Diagramms ist eigentlich nicht der Ablauf, da dieser sehr einfach ist, sondern die Möglichkeit, dieses Diagramm als Checkliste und als Wissensbasis zu nutzen.

Alle Elemente des Diagramms können angeklickt werden, um so innerhalb der Prozesse navigieren zu können. So können anhand der aktuellen Problemstellung sehr einfach die zugehörigen Vorlagen oder Richtlinien gefunden werden, da sie über die Hierarchie des Prozess, Workflow Detail und Artifact schnell lokalisiert werden können.

Der Erfolg: Collaborative Controlling

Erfolgreiche Softwareprojekte müssen nicht nur die fachlichen und technischen Ziele erreichen sondern auch wirtschaftliche Rahmenbedingungen einhalten. Diese werden meist zu Beginn des Projektes kalkuliert und müssen während des Projektes überwacht werden. Die Überwachung der

Aufwendungen und somit der Kosten muss zeitnah durchgeführt werden. Die iterativen inkrementellen Vorgehensweisen und Best Practices aus Extreme Programming bringen die Qualität der fachlichen und technischen Ergebnisse weit nach vorne, machen aber das Controlling von Projekten schwierig, da die Aufwen-

dungen durch viele kleine (oder größere) Änderungen oder Erweiterungen in vielen kleinen Stufen kontinuierlich anwachsen. Wenn dies erst am Ende des Projektes bemerkt wird, ist eine erfolgreiche Argumentation gegenüber dem Kunden nahezu unmöglich. Abweichungen von Budgets müssen schnellstmöglich erkannt werden.

Zusätzlich steigert sich die Brisanz des Controllings bei verteilten Projekten, da der Projektmanager nicht mehr auf die Kommunikation seiner Mitarbeiter mit dem Kunden einwirken kann und somit Aufwendungen nicht mehr zentral kontrolliert werden können. Die Erfassung der täglichen Projektzeiten in einer internetbasierten Projekt-Controlling-Anwendung ermöglicht die zeitnahe Analyse von Planabweichungen und somit erfolgreiche Steuerungsmaßnahmen für ein wirtschaftlich erfolgreiches Projekt, auch bei verteilten Teams.

Die Mitarbeiter: Collaborative Skills

Die Auswahl und Weiterentwicklung von Skills der Mitarbeiter sind für technologieorientierte Dienstleistungs- und Softwareunternehmen die Basis für den zukünftigen Erfolg. Um der erschwerten persönlichen Kommunikation der Führungskräfte in verteilten Projekten Rechnung zu tragen, bietet sich auch in diesem Bereich der vorteilhafte Einsatz internetbasierter Skill Management Anwendungen an. Diese dürfen und sollen keinesfalls das regelmäßige Mitarbeitergespräch ersetzen, können aber eine wertvolle Unterstützung dafür bieten.

Resümee

Die Unterstützung von verteilten Projekten wirft einige Probleme auf und ist nicht mit einer gemeinsamen Sourcecode-Basis erledigt. Allerdings ist dies die Basis, ohne die eine verteilte Entwicklung gar nicht vernünftig funktionieren kann. Also sollte mit der Einführung bei Werkzeugen für diesen Bereich begonnen werden. Zur Verbesserung der weiterführenden organisatorischen Abläufe wie Prozesse, Controlling und Mitarbeitersteuerung müssen ebenfalls Maßnahmen ergriffen werden, um Projekte wiederholbar und erfolgreich durchführen zu können. ■

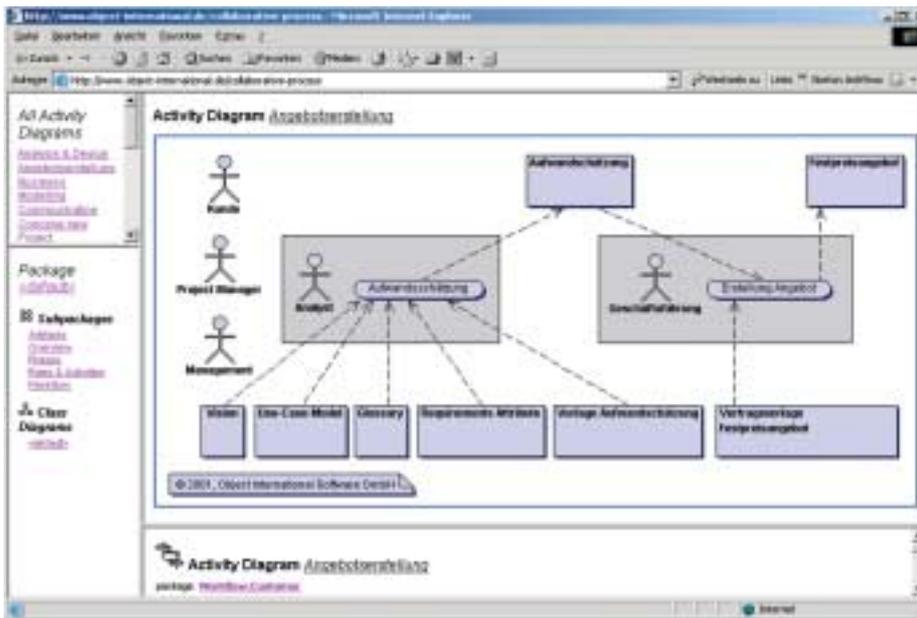


Abb. 5: Workflow Detail „Angebotserstellung“

Anforderungen an ein Collaborative Software Development Portal

Projektverwaltung, Organisation, Kommunikation und QS:

Collaborative Development Portal für verteilte Softwareprojekte. Verwaltung von Sourcecode und Informationen, Versionierung, Supportanfragen, Bugtracking, Change Requests, Feature Requests, Kommunikation, Notification, Cross-Referencing, Impact Analysis, Dependencies, Quality Assurance, Metrics, Power Search. Beispiel unter www.object-international.de/collaborative-project.

Softwareentwicklungsprozess:

Portal zur Unterstützung einer unternehmensweiten Vorgehensweise in Softwareprojekten. Der abgebildete unternehmensspezifische Softwareentwicklungsprozess ist eine Kombination aus USDP (Unified Software Development Process), XP (Extrem Programming) und spezifischen Erweiterungen, um ISO 9000 und CMM SW Level 2-3 Qualitätsanforderungen erfüllen zu können. Das Portal bietet Vorgaben für Artifacts, Protokolle, Richtlinien für Analyse, Design und Softwareentwicklung, Architekturen, Projektplanung, Kalkulation, Konfigurationsmanagement usw.

Controlling und Leistungsabrechnung:

Software- und Dienstleistungsunternehmen müssen die erbrachten Leistungen erfassen und abrechnen können. Dazu ist eine Anwendung zur Erfassung, Auswertung und Abrechnung von Projektzeiten notwendig. Wenn die Mitarbeiter in räumlich verteilten Teams, z. B. beim Kunden, arbeiten, muss die Anwendung internetfähig sein. Ferner sind diese Zahlen für das Projektcontrolling notwendig, um die geplanten Aufwendungen mit den tatsächlichen Aufwendungen vergleichen zu können.

Skill Management:

Dienstleistungsunternehmen erbringen mit dem Know-how der Mitarbeiter Dienstleistungen für ihre Kunden. Somit ist die Suche und Auswahl von Personen mit bestimmten Skills die Basis ihrer Arbeit. Dazu ist eine Anwendung zur Erfassung und Verwaltung von Skill-Profilen von Mitarbeitern, Bewerbern und Teammitgliedern hilfreich. Beispiel unter www.oisoft.com/mitarbeiter/eignungstest.htm