

MAJA_OBJECTS

ProjectForge

OpenDevelopment Platform

WhitePaper

ABSTRAKT

Projektmanagement im Bereich Softwareentwicklungen unterliegt besonderen Bedingungen. Strukturierte Kommunikation ist der Schlüsselparame-ter für eine erfolgreiche Projektabwicklung.

AUTOR

Dipl.Ing. Roger René Kommer (kommer@micromata-objects.de)

VERSION

1.0

DATUM

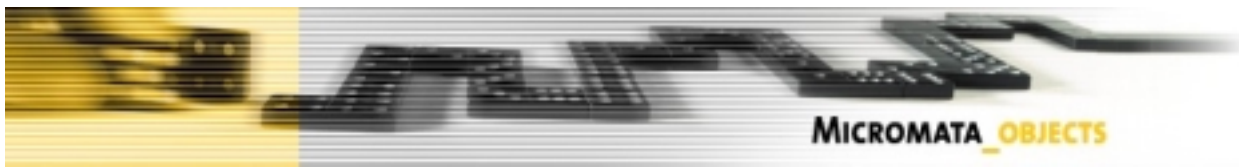
Kassel, den 24.07.2000

SEITEN

8

KONTAKT

Micromata_Objects GmbH
Ochshäuser Strasse 45
D-34123 Kassel
Tel.: 0561-57063-0
Fax.: 0561-57063-19
<http://www.micromata-objects.de>
<mailto:kommer@micromata-objects.de>



Project Management in Software Development

– Webbased OpenDevelopment Platform

Im Rahmen eines Vortrages einer Veranstaltung der Deutschen Bank „Open Exchange Deutsche Bank & Guests am 2000-05-23 fand ein Symposium zu dem Thema “Managing Software Development through Key Performance Indicator´s” mit Beiträgen von Juergen Krusch (Deutsche Bank), Bernd Will (Oracle Deutschland), Dipl.-Ing. Günter Glanzmann (IBM), Dipl.-Ing. Roger René Kommer (Micromata_Objects) statt.

Der Vortrag von Micromata_Objects bietet die Grundlage für Betrachtungen, die als Hintergrund zu dem Produkt ProjectForge zu verstehen sind.

Traditionelle Rolle des Projektmanagement

Die Traditionelle Organisation von Projekten besteht aus einem hierarchisches Gefüge, wobei die jeweils übergeordnete Ebene die Steuerung und Kontrolle der jeweils untergeordneten Ebene darstellt.

Durch diese Steuerung sollen folgende wesentlichen Projektziele erreicht werden:

Innere Ziele:

- Effektivität der Softwareentwicklung
 - Zuverlässigkeit der Softwareentwicklung
- = Qualität der Entwicklung

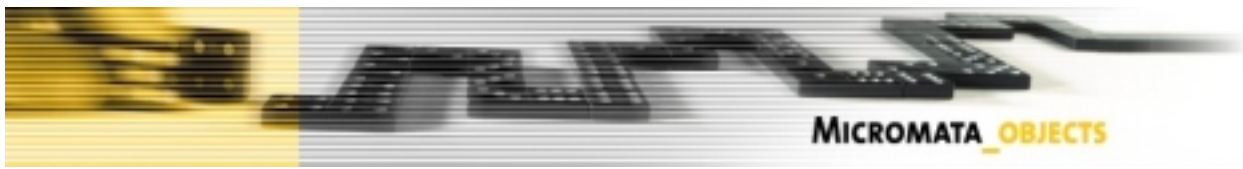
Äussere Ziele:

- Erfüllung von Spezifikationen / Pflichtenhefte / Verträge
 - Kalkulierbarkeit der Softwareentwicklung
 - Erfüllung des Zeitplanes
 - Kostenminimierung
 - Maximierung der Kundenzufriedenheit
- = Marktfähigkeit der Entwicklung

Je nach Markt und Branche gestaltet sich das konkrete Projektgeschäft innerhalb dieser Parameter. Das Projektmanagement muss diese Parameter messen und steuern.

Neue Rollen des Projektmanagement

Seit Jahrzehnten gibt es die Rede von dem Anwendungsstau – womit man ausdrücken will, dass in Unternehmen stets eine längere Wunschliste von Anwendungen, die noch nicht erstellt worden sind. Nun, „Jammer“ gehört ebenso zum Geschäft. In den letzten Jahren – im Rahmen von E-Commerce



und Enterprise Application Integration können fehlende Anwendungen das schnelle Aus für Unternehmungen werden.

Aus diesem Grund sind folgende Aspekte in der Softwareentwicklung besonders wichtig:

- Schnelligkeit der Entwicklung
- Kontrolle der Entwicklung
- Reaktionsfähigkeit der Entwicklung auf das Marktgeschehen (Time to Market)

Für Häuser, die ihr Geld mit der Erstellung von Software verdienen, ist in diesem Kontext auch der Punkt

- Nachhaltigkeit der Softwareentwicklung

im Sinne einer Produktgenese aus dem Projektgeschäft von wesentlicher Bedeutung.

Bedeutung des Manament in Softwareprojekten

- Arbeit und Ressourcen planen
- Arbeit und Ressourcen steuern
- Leiten von Kommunikation

Neue Ansätze im Softwareentwicklungsbereich

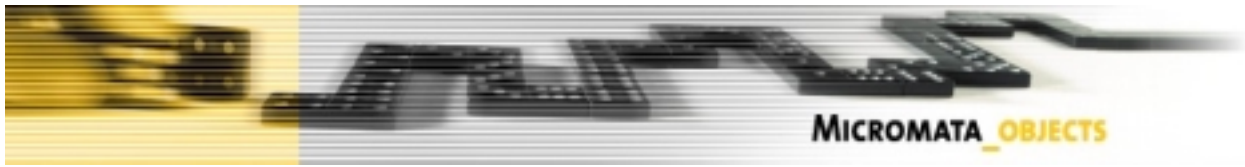
"The Cathedral and the Bazaar"

Im Bereich der OpenSource-Bewegung, also im Umfeld von Linux, Apache, KDE und Samba (um nur ein paar zu nennen) scheint die Welt der Softwareentwicklung gänzlich anders zu verlaufen.

Hierbei operieren die Entwickler dieser doch sehr komplexen Software unter erschwerten Bedingungen:

- Keine oder kaum Leitung
- Knappe finanzielle Ausstattung
- Entwickler sind über die gesamte Welt verstreut
- Entwickler sind nicht immer verfügbar.

Eric Raymond nahm diese Beobachtungen zum Anlass in seinem Essay "The Cathedral and the Bazaar" von einem neuen Paradigma in der Softwareentwicklung zu sprechen. Er stellt die klassische Entwicklung eines grossen Softwareprojektes - das wie der Bau einer Kathedrale organisiert ist - gegen das Prinzip in der OpenSource-Szene das eher wie ein Basar organisiert.



Wichtige Regeln für das Projektmanagement und den Erfolg von Softwareprojekten im OpenSource Bereich identifiziert Raymond:

- Schnelle und viele Freigaben
- Kommunikation ist die Schlüssel Qualifikation für den Projekterfolg

"Extreme Programming"

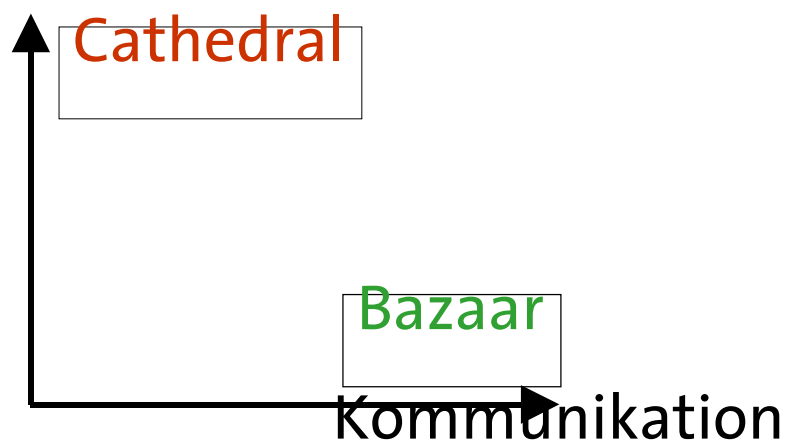
Kent Beck hat mit seinem Buch "Extreme Programming" die wichtigen Erfolgsparameter für Softwareprojekte zusammenfassen. Eine Auszug hiervon ist:

- Schnelle und viele Freigaben
- Design, Implementation und Tests sind kein linearer sondern ein zyklischer Prozess.
- Den Entwicklern muss die Kontrolle über das Projekt gegeben werden.
- Kommunikation ist die Schlüssel Qualifikation für den Projekterfolg

Organisation und Kommunikation

Wenn wir die von Raymond eingeführten Modelle Cathedral / Bazaar näher betrachten so fällt folgender Zusammenhang auf:

Organisation



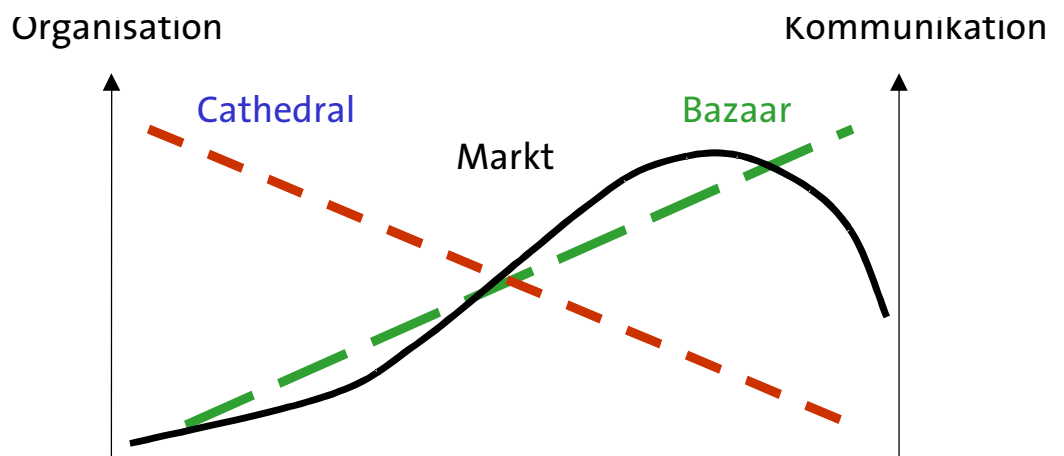
Das Modell Cathedral ist Organisationsgesteuert während das Modell Bazaar Kommunikationsgesteuert ist.

Marktmechanismen bei der Projektabwicklung

Klassische Projekte operieren plangesteuert, während die Anforderungen in den Felder E-Commerce und EAI oft keinen längerfristigen Plan, da die Softwareentwicklung der zeitnah zum Marktgeschehen gesteuert werden muss.

Nach Berichten aus Grossunternehmen mit größeren Projekten, stellt sich das Problem bei gescheiterten Projekten nicht so sehr, dass die außerhalb des gesetzten Zeit- und Kostenrahmen laufen, sondern vielmehr, dass die einerseits die Projekte nicht die grundsätzlichen technologischen Anforderungen erfüllen können, oder im Laufe eines Projektes – vor allem bei längerfristigen Projekten – die ursprünglichen Intensionen der Applikationsentwicklung völlig deformiert oder gar obsolet wurde, da die Unternehmen selbst – für diese Anwendungen erstellt werden sollen – selbst hoch dynamische Systeme geworden sind.

Aus diesem Grund ist sinnvoll auch im Projektgeschäft von einem Marktgeschehen und den damit verbundenen Mechanismen auszugehen. In den Zusammenhang von organisations- und kommunikationsgesteuerten Projekten lässt sich in etwa folgenden Zusammenhang herstellen:

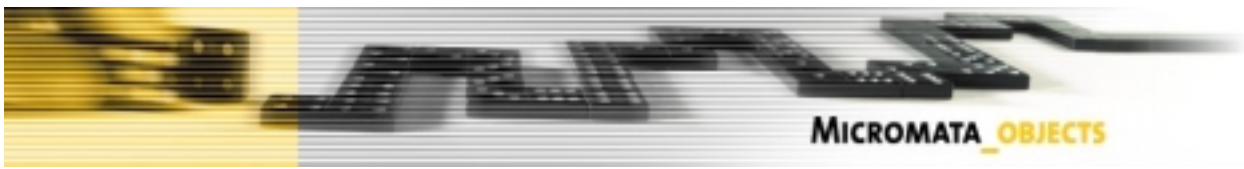


Frühzeitige und viele Releases

Der Sinn von einem engen Raster an Releasezyklen, die auch schon in relativ frühen Projektphasen gestartet werden ist darin begründet, dass Qualität als Eigenschaft erst durch ein Veröffentlichung wirklich zu messen ist, auf der anderen Seite es aber für ein Projekt schädlich ist, diese Qualität erst spät in das zu erstellende Produkt einzupflegen.

Weiterhin haben erst durch den Zwang zur Veröffentlichung das Management eine Möglichkeit den Projektverlauf zu überschauen und Verzögerungen frühzeitig zu erkennen (Programmierer unterschätzen – in der Regel sogar unterschlagen vollständig – den Aufwand für Dokumentation und Distribution).

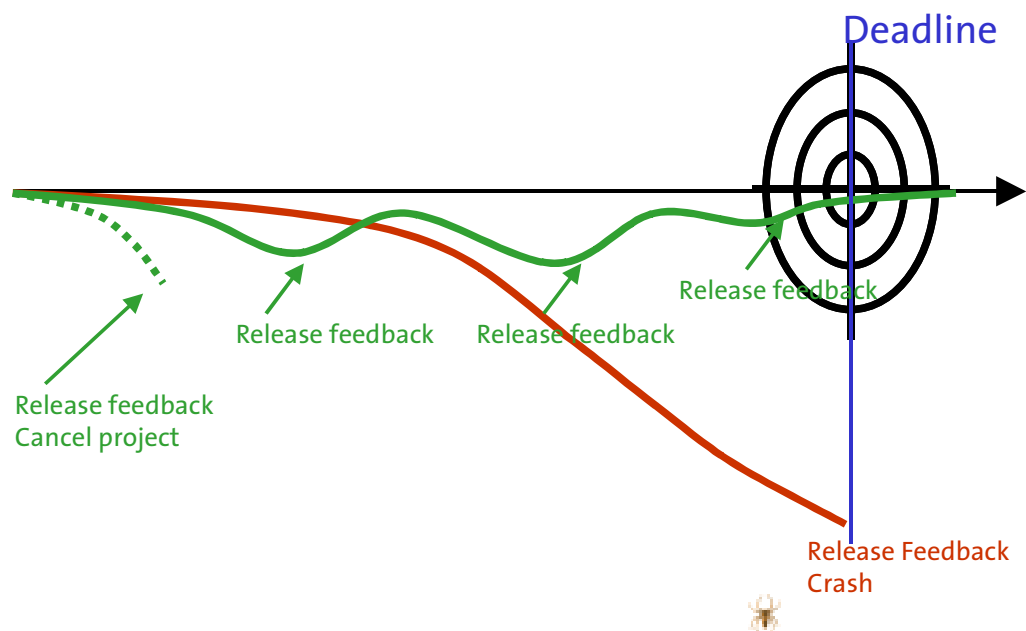
Weiterhin gibt frühe Freigaben der Software auch den Kunden die Möglichkeit den Projektfortschritt und vor allem auch die Anwendungsfälle innerhalb des



Kundenunternehmen kritisch zu untersuchen. Leider gibt es viel zu viele Projekte, bei denen im Projektverlauf der Auftraggeber, dass er die Software im beauftragten Umfang eigentlich gar nicht braucht, da sich sein Geschäft in der Zwischenzeit geändert hat. Er wird durch Mäkeln am Projekt versuchen möglichst wenig Geld dem Auftraggeber zu zahlen, während das Entwicklungsbüro die Software bis zum bitteren Ende durchprogrammiert, da erst durch die Abnahme (an welcher der Auftragnehmer das Interesse verloren hat) eine weitere Inrechnungsstellung möglich ist.

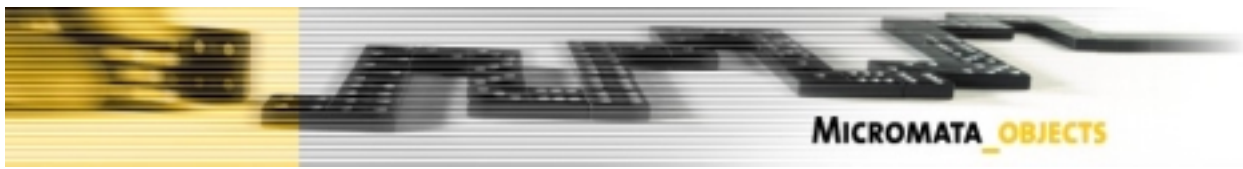
Zyklische Entwicklung

Natürlich müssen Erkenntnisse über die Entwicklung eines Projektes konstruktiv in technischen oder administrativen Maßnahmen umgesetzt werden. Aus diesem Grunde sollte die Softwareentwicklung nicht einem strengen Wasserfallmodell entsprechen, sondern vielmehr zyklisch organisiert werden. Re-Design sollte ein permanenter Prozess sein – und nicht eine Maßnahme, die ergriffen wird, wenn das Projekt eigentlich schon gescheitert ist. Eine Überprüfung der inneren Projektmarktsituation sind nur dann sinnvoll, wenn allen Beteiligten des Projekts die Möglichkeit geboten wird falsche Richtungen zu korrigieren.



Kommunikation

Ein Problem der Offenlegung des Projektverlaufes durch frühe Releases ist oft auch, dass der Kunde wenig Verständnis für halbfertige Software zeigt. Hierfür gibt es zwei Lösungen.



Die Releases werden nicht nach dem Grad der Fertigstellung der Software definiert sondern vielmehr nach dem Scope, der Ausbaustufe des Produktes. Dazu müssen die ersten Versionen der Software nur Kernfunktionen erfüllen, diese jedoch schon konkrete Anhaltspunkte für die Brauchbarkeit und Qualität der Gesamtsoftware.

Der Ansatz funktioniert jedoch nicht, wenn nicht eine offene Kommunikation zwischen dem Kunden und den Entwicklern herrscht.

Eine durchaus pädagogische Arbeit sowohl an den Entwicklern, den Projektleitern und Geschäftsführern von Auftragnehmer und Auftraggeber sowie den Endanwender muss geleistet werden.

Kommunikation

Einige Thesen:

- Viele Projekte scheitern aus Kommunikationsfehlern
- Der schlimmste Kommunikationsfehler ist das Unterlassen von Kommunikation
- Man kann leider nichts aus dem Scheitern von solchen Projekten lernen.
- Angst in vielfältiger Form – bezogen auf der eigenen Stellung innerhalb der Firma oder auch bezogen auf die Aussenwirkung eines Unternehmens auf den Partner – ist einer der Hauptverhinderer von Kommunikation
- Nicht-Kommunikation erhöht dieses Angstpotential und vice versa.
- Alle Beteiligten wissen oder erahnen zumindest, wie es wirklich um das Projekt bestellt ist.
- Alle Beteiligten sind in solchen Projekt hoch frustriert.

Die wichtigste Erkenntnis aus diesen Erfahrungen ist:
Kommunikation muss offen sein!

OpenDevelopment Platform

In der Architektur gibt es ein wesentlichen Erkenntnisse: Man kann Häuser nicht dazu benutzen den Menschen eine gewisse Art vorzugeben, wie sie zu leben haben – außer diese Häuser menschenfeindlich. Häuser können vielleicht die Phantasie der Menschen anregen, ihr Leben in anderen Bahnen zu verfolgen.

Ein Ähnliches ließe sich über Werkzeuge sagen, die Unternehmen helfen können, ihre Projekte besser abzuwickeln.

Eine OpenDevelopment Platform sollte solch eine Versuchung und Aufforderung enthalten.

Hierzu braucht es folgende grundlegende Eigenschaften:



- allgemeine Verfügbarkeit
- einfache Nutzbarkeit
- Senkung der Hemmschwellen
- Möglichkeit unterschiedliche Kanäle und Methoden der Kommunikation abzubilden
- Möglichkeit verschiedene Projektrollen abzubilden
- Möglichkeit verschiedene Sichten auf das Projekt abzubilden
- Keine reinen Managertools (welcher Entwickler hat Microsoft Project installiert) keine reinen Entwicklertools (welcher Manager wählt sich durch Code um die noch offenen Baustellen zu bewerten).
- Integrativer Ansatz für Controlling, innere Projektverwaltung, Customer Care, etc.
- Gestufter Zugang für Entwickler / Manager / Kunde / Endnutzer