

Erfolgsbarrieren in Software-Projekten

Nach den Erfahrungen von Actinium Consulting müssen die Unternehmen regelmäßig einen hohen Aufwand in die Nachbearbeitung ihrer neuen oder modifizierten Software-Lösungen investieren, weil sich im Produktivbetrieb vielfältige Probleme zeigen. Die Schwierigkeiten reichen von Inkonsistenzen unter Praxisbedingungen bis zur unzureichenden Revisionsfähigkeit. Klaus Hüttl, Geschäftsführer des Beratungshauses, hat deshalb einige der aus Praxissicht bedeutendsten Erfolgsbarrieren von Softwareprojekten skizziert:

- **Schlechte Softwareverwaltung/Dokumentation:** Sie resultiert nicht zuletzt daraus, dass beim Einsatz der üblichen Programme zur Source- oder Versionsverwaltung wie CVS, Subversion, Git oder Mercurial nur Abhängigkeiten auf der Source-Ebene Berücksichtigung finden. Diese Systeme können deshalb nur begrenzt helfen, Inkonsistenzen zu vermeiden. Notwendig ist vielmehr eine Softwareverwaltung, die ebenso Querbezüge - beispielsweise auch zwischen Programmen und Skripten - analysieren und verwalten kann. Sie muss zudem durchgängig in den gesamten Entwicklungsprozess integriert sein.
- **Unklare Prozesse in der Softwareentwicklung:** Eine geringe Transparenz der Abläufe erzeugt Unklarheiten, verunsichert die Entwickler, erhöht den Kommunikationsbedarf und erzeugt dadurch anschließend in unterschiedlichem Umfang Fehlverhalten. Dies geht nicht nur zulasten der Effizienz, sondern die nachträglichen manuellen Korrekturen bergen immer auch eine höhere Fehleranfälligkeit in sich, da selbst bei der Anpassung eines Skripts oder dem Kopieren einer Datei intransparente Entwicklungen mit hohem Folgerisiko entstehen.
- **Programmierstandards werden nicht genau beachtet:** In der Regel sind die Programmierstandards in internen Handbüchern hinterlegt, die in der Praxis aber selten in die Hand genommen werden. Als Konsequenz nutzt jeder Entwickler eigene

Interpretationen der Standards. Durch die Nutzung unterschiedlicher Methoden entstehen später jedoch mitunter erhebliche Diskrepanzen in der Anwendung. Dies führt bestenfalls zu einer nachträglichen Bearbeitung, es ist aber auch möglich, dass methodische Bugs trotz Testing noch eine längere Zeit im Programm schlummern. Für kritische Softwareobjekte ist deshalb eine konsequente Orientierung an dem Generierungsprinzip einer durchgängig genutzten Entwicklungsmethodik wie beispielsweise Scrum zu empfehlen. Es vermeidet die Probleme bereits in ihrer Entstehung und macht die Ergebnisse portabel.

- **Personelle Fluktuation in der Entwicklung:** Je unklarer die Prozesse in der Softwareentwicklung und die festgelegten Standards sind, umso negativer wirken sich personelle Veränderungen im Verlauf von Entwicklungsprojekten aus. Dann bedeutet jede Fluktuation den Verlust von Know-how und spezifischer Erfahrung mit der Applikation, ohne dass ein ausreichender Transfer auf einen neuen Mitarbeiter gewährleistet werden kann. Diese Probleme verschärfen sich sogar, wenn häufig wechselnde externe Ressourcen eingesetzt werden.
- **Fachliche Defizite der Mitarbeiter:** Gerade komplexe Softwareprojekte benötigen ein über alle beteiligten Entwickler hinweg ähnliches fachliches Niveau. Eine solche Situation ist in der Praxis aber vielfach nicht anzutreffen, vielmehr sind häufig fachliche Defizite einzelner Mitarbeiter festzustellen. In einem klar definierten Entwicklungsprozess mit festgelegten Standards werden Mitarbeiter hingegen so gesteuert, dass fachliche Schwächen kompensiert werden können. Denn solch eine integrierte Softwareverwaltung versorgt sie mit allen Informationen, die sie für die tägliche Entwicklungsarbeit benötigen.