

Der Architekt kommt nach Baubeginn.

Die Bedeutung des requirements engineering in der Softwareentwicklung.

Niemand kann sich heute noch vorstellen, daß ein Bauherr die Handwerker mit dem Hausbau beginnen läßt, bevor die Pläne von Architekten und Statiker vorliegen. Auf die Reihenfolge Plan/Ausführung achtet auch die Baubehörde, um die allgemeine öffentliche Sicherheit zu gewährleisten. Die Zeiten sind vorbei, in denen die Bauherren ihr Haus nach eigenen Vorstellungen in Do it yourselfe Verfahren intuitiv ohne Aufzeichnungen ausführen.

Sie fragen sich, was dieses Beispiel mit der Softwareerstellung zu tun hat? Auch die Softwareerstellung unterliegt den Entwicklungsmethoden, die für alle Produkte gilt:

in einem ersten Schritt ist das Produkt, das erstellt werden soll, zu entwerfen und zu beschreiben (requirements engineering) und

in einem zweiten Schritt beginnt auf der Grundlage des Entwurfes und der Pläne die Erstellung des Produktes.

Diese grundsätzliche Vorgehensweise gilt sowohl für einmalige als auch für Serienprodukte.

Erstaunlich ist, daß diese allgemeine Regel im Bereich der Softwareentwicklung nicht konsequent eingehalten wird.

Die Softwareentwickler sind der Meinung, daß ihre Tätigkeit mit keiner anderen zu vergleichen ist. Sie fühlen sich in dieser Haltung so sicher, daß sie nicht versuchen, aus den Erfahrungen anderer Bereiche Vorteile für sich selber zu ziehen. Sie haben in der Vergangenheit z.B. sehr wirkungsvolle Programme für den industriellen Fertigungsbereich erstellt, ohne darüber nachzudenken, ob sie mit diesen Produkten nicht auch den eigenen Bereich unterstützen können. Aber auch die in den Softwareprodukten abgebildeten Methoden haben sie nicht für sich verinnerlicht. So findet man im Bereich der Softwareentwicklung nur in seltenen Fällen die organisatorische, personelle und verantwortliche Trennung zwischen Produktplanung und -erstellung.

In Gesprächen mit Informatikern ist festzustellen, daß sie der Meinung sind, sie üben eine "Kreative Tätigkeit" aus, die schwer zu beschreiben ist. Woher kommt diese Einschätzung der eigenen

Tätigkeit? Ist sie berechtigt oder soll mit dieser Aussage eine künstliche Abgrenzung geschaffen werden und was sind die Folgen?

Softwareerstellung bedeutet, eine Abbildung von realen Vorgängen, Zuständen und Regeln zu machen und dieses Abbild in einer Form zu dokumentieren, die von Computern interpretiert und ausgeführt werden kann. Zwei unterschiedliche Tätigkeiten müssen also nacheinander ausgeführt werden: Erstellen des Abbildes in Form von detaillierten Organisations- und/oder Arbeitsanweisungen und deren Übersetzung aus unserer Sprache in die Sprache der Computer. Da die Informatik allgemein als Wissenschaft von der systematischen Verarbeitung von Informationen, insbesondere der automatischen Verarbeitung mit Hilfe von Computern definiert wird, beschäftigen sich die Informatiker in den Unternehmen weniger mit der Beschreibung der realen Grundlagen (Aufgabe) der Programme als mit deren technischer Umsetzung. Die Informatiker betätigen sich mehr als Bauhandwerker, denn als Architekt. So wie sich der Mauerer mehr darum bemüht, senkrechte und glatte Mauern zu erstellen, und sich weniger mit dem Zuschnitt der Räume beschäftigt, so vernachlässigen die Mehrzahl der Informatiker die Aufgabenstellung für die Programme.

Sie beginnen mit dem Schreiben der Programme, bevor deren Inhalt in Details definiert ist.

In der Praxis der Softwareentwicklung führt diese Situation zu allgemein bekannten Folgen:

1. Aufwand- und Kostenschätzungen, die zu Beginn der Projekte den Planungen zu Grunde gelegt wurden, werden nur in seltenen Fällen eingehalten. Diese negative Erfahrung wird von den Informatikern uneingeschränkt bestätigt. Sie sehen in dieser Tatsache eine der Besonderheiten für ihren Arbeitsbereich und nicht eine falsche Arbeitsmethode. Niemand kann aber den Aufwand für eine Arbeit nennen, wenn nicht bekannt ist, was im einzelnen zu machen ist.

Als Entschuldigung für den Beginn der Programmerstellung auf unzureichenden Unterlagen wird einhellig angeführt, daß vorgegebene Termine keine Zeit lassen, um vorweg detaillierte Vorgaben zu erstellen.

2. Die Methode der Fixierung des Auftragsgegenstandes im Verlauf der Produkterstellung führt zwangsweise dazu, daß die gesetzten Termine zu eng werden. Die Folge ist, daß unvollständige Produkte abgeliefert werden. Unter dem Begriff Wartung und zu deren Lasten werden dann später diese Mängel beseitigt.

3. Die Erfahrung zeigt, daß bei Terminschwierigkeiten die notwendigen Tests nicht sorgfältig genug durchgeführt werden. Die Folgen haben die Anwender zu ertragen.

4. Um den Termindruck zu mildern neigen Informatiker wie jeder andere dazu, zuerst die Arbeiten zu vernachlässigen bzw. auszulassen, die man ungern macht. An erster Stelle ist hier für den Informatiker die Dokumentation zu nennen. Fehlende oder unvollständige Dokumentationen aber verursachen im Verlauf der Nutzung der Produkte hohe Wartungskosten. So wäre z.B. der Aufwand für das Problem 2000 weniger hoch, wenn für alle Programme gute Dokumentationen vorliegen würde.

Die Unterschätzung der Bedeutung des requirements engineering ist ein, wenn nicht das Grundübel der Softwareentwicklung. Das requirement management muß ausgebaut werden. Der Aufwand für die Definition der zu realisierenden Funktionalität ist realistisch mit ca. 30 - 35 % des Gesamtaufwandes vor dem Beginn technischen Umsetzung einzuplanen. Die Anforderungen an das Softwaresystem sind ingenieurmäßig festzulegen, bevor die Realisierung beginnt. Die Vorstellung, daß der Termindruck dies nicht zuläßt, führt nur dazu, daß dieser gegen Ende des Projektes noch größer wird als bereits zu Anfang angenommen.

Viele Probleme in der Softwareentwicklung werden sich vereinfachen, wenn dem requirements engineering die gleiche Bedeutung zugemessen wird wie der systemtechnischen Realisierung.

© Robert Hürten
April 2000