



Modernisierung

Modernisierung versus Neuentwicklung

4A Solutions GmbH
Dachauer Straße 42
80335 München
Tel. +49 (0)89 / 189 126 30
Fax +49 (0)89 / 189 126 32
info@4a-solutions.de
www.4a-solutions.de

Inhaltsverzeichnis

1. Ist-Stand	2
2. R & R	3
3. Vorgehen	3
4. Modernisierung und Neuentwicklung im Überblick	5

1. Ist-Stand

In zahlreichen Unternehmen sind Software-Systeme im Einsatz, die seit Jahren, wenn nicht Jahrzehnten, ihren bewährten Dienst tun. Trotz oder gerade wegen des langgedienten erfolgreichen Einsatzes haben diese Systeme meist ein Problem: sie altern.

Software-Systeme altern aus verschiedenen Gründen:

- Durch geänderte Rahmenbedingungen kommen neue Anforderungen auf sie zu. Erweiterungen müssen geschaffen werden. Neue Funktionalität muß integriert werden, neue Sicherheitsbestimmungen müssen erfüllt werden, neuer Code kommt hinzu.
- Neue Schnittstellen müssen unterstützt werden. Die Umgebung der Systeme bleibt nicht statisch. Die Systeme müssen mit der Umgebung kommunizieren.
- Neue Technologien müssen unterstützt werden.
- Neue Hardware kommt zum Einsatz. Dies kann zum einen die Hardware, auf der das Software-System selbst läuft, sein, zum anderen zugehörige Peripheriegeräte wie neue Ein-/Ausgabegeräte (graphische Benutzeroberflächen, Sprachschnittstellen, ...), Kommunikationshardware (neue Netzwerke, ...) u. ä..
- Schließlich tun noch Versionswechsel von Betriebssystemen, Bibliotheken, Datenbanken, etc., die unterstützt werden müssen, ein übriges, das Änderungen an der eingesetzten Software erfordert.

Diese Änderungen werden in den seltensten Fällen mit der gleichen Vorsicht, Sorgfalt und dem gleichen Aufwand an Zeit und Ressourcen eingearbeitet, wie es bei Erstellung des Systems der Fall war.

Durch die Weiterentwicklung verschlechtert sich dadurch die Qualität der Software. Die Struktur der Software und ihre Architektur sind nicht mehr tragfähig, da sie nie für die neu hinzugekommenen und weiter hinzukommenden Aufgaben entwickelt wurde. In Extremfällen drohen die Systeme unter der neu hinzugekommenen Last auseinander zu brechen.

In jedem Fall werden Erweiterungen und die Behebung von Fehlern immer aufwendiger, immer teurer und immer gefährlicher. Das Projektrisiko steigt von mal zu mal. Die Performanz des eingesetzten Systems sinkt.

Es scheint an der Zeit für eine Neuentwicklung. Nur eine neue Generation der Software kann die neuen Aufgaben zufriedenstellend lösen.

Eine Neuentwicklung bedeutet jedoch i. allg. immens hohe Kosten. Gleichzeitig stellt eine Neuentwicklung ein sehr hohes Projektrisiko dar, da es sich um ein großes Projekt handelt.

Wird die alte Software obendrein weiter eingesetzt, gewartet oder gar vertrieben, besteht die Gefahr, mit der Neuentwicklung einem "moving target", einem beweglichen Ziel, hinterher zu arbeiten, bei dem die Neuentwicklung das Altsystem bezüglich Funktionalität und Komfort möglicherweise nie einholt.

Die Unternehmen befinden sich derzeit immer noch in einer Phase der Konsolidierung, bestenfalls der vorsichtigen Expansion. Das Geld sitzt nicht mehr so locker wie noch vor zwei oder drei Jahren. Kosten müssen weiter eingespart oder zumindest vermieden werden.

Trotzdem wollen und müssen die Unternehmen ihre Wettbewerbsfähigkeit erhalten und verbessern. Ein

Beharren auf den maroden Altsystemen können sich nur wenige erlauben.

Die Antwort auf dieses Dilemma lautet Reengineering und konkret Refactoring der Software.

2. R & R

Reengineering bedeutet ein schrittweises Umbauen der alten Software indem einzelne Teile durch Neuentwicklungen ersetzt werden. Refactoring bedeutet, dass die Qualität des Quellcodes der alten Software durch erprobte Maßnahmen verbessert wird, um ihn für zukünftige Erweiterungen tragfähig zu machen und der normalen Verschlechterung durch Änderungen vorzubeugen. Durch die Kombination beider Maßnahmen (Reengineering und Refactoring: R & R) kann alte Software in handhabbaren Schritten inkrementell umgebaut und schließlich komplett durch ein Neusystem ersetzt werden, ohne jemals die Neuentwicklung als einzelnes riesiges und gefährliches Projekt angehen zu müssen. Das System bleibt während der gesamten Umbauphase einsatzfähig.

R & R ist nicht simpel, schließlich entspräche es analog in der Automobilwelt etwa der Umstellung eines alten BMW 7er-Modells auf die neue 7er-Baureihe während das Auto fährt.

Trotzdem bietet R & R eine Reihe unbestreitbarer Vorteile, die besonders in der heutigen Zeit eine große Rolle spielen:

- Bei R & R kommen die Anwender der Software schnell in den Genuß neu entwickelter Features. Gleichzeitig erhalten die Entwickler viel früher Feedback zu ihrer Software aus der Praxis und aus Anwendersicht, da die Software früher eingesetzt wird.
- Die Kosten verteilen sich über die gesamte Projektlaufzeit und können direkt an den meßbaren Projekterfolg der Software im Einsatz gekoppelt werden.
- Das Projektrisiko sinkt gegenüber einer Neuentwicklung i. allg. dramatisch, da Fehlentwicklungen sehr früh direkt im Zusammenhang mit dem vollständigen Produkt erkennbar werden.

Dennoch stellt R & R nicht die Lösung aller Softwareprobleme dar. Es gibt auch eine Reihe von Nachteilen und Gefahren für R & R-Projekte:

- R & R ist technisch sehr schwierig und anspruchsvoll.
- Bei der In-house vorhandenen Entwicklungsmannschaft muss i. allg. das vorhandene Potential für die effiziente Durchführung von R & R-Projekten erst noch geweckt werden.
- Von Seiten der Entwickler, vor allem der weniger erfahrenen, gibt es oft heftige Widerstände gegen R & R-Projekte, da sie technisch weniger interessant erscheinen und damit weniger Spaß versprechen als Neuentwicklungen.
- Ggf. müssen Werkzeuge, Prozesse und Dokumentation im Projekt für das Altsystem nachgezogen werden. Hier lauern versteckte Kosten und Fehlerquellen.

3. Vorgehen

Um konkret für ein Projekt beurteilen zu können, welcher Ansatz (Neuentwicklung oder R & R) sinnvoller und vielversprechender ist, sind viele Details im Einzelfall zu prüfen. R & R ist immer noch wenig bekannt. Da es obendrein wenig Experten für R& R gibt, wird dieser Ansatz oft gar nicht ernsthaft als Alternative zu einer Neuentwicklung in Betracht gezogen.

Verantwortliche Manager und Projektleiter, die jedoch Probleme wie die zu Beginn dieses Artikels beschriebenen aus eigener Erfahrung und von Software aus dem eigenen Haus kennen, sollten dennoch R & R sehr ernsthaft in Erwägung ziehen und sich ggf. mit einem Software- oder Consulting-Haus, dass auf R & R spezialisiert ist, in Verbindung setzen.

Was sind nun die konkreten einzelnen Schritte bei einem R & R Projekt:

Bestandsaufnahme

Bei einer Bestandsaufnahme sind unterschiedliche Bereiche zu untersuchen. Es kann nie nur die Technologie des bestehenden Systems analysiert werden. Auch die Organisation und insbesondere die Menschen, die in ihr arbeiten sind zu betrachten. Demnach möchten wir folgende Unterteilung für eine Bestandsaufnahme vorschlagen:

1. Technische Faktoren:
 - Architektur
 - Dokumentation
 - Hardware
 - Architektur- und Design-Patterns
 - Fremdsysteme (Datenbank, Betriebssystem, Frameworks, ...)
2. Organisatorische Faktoren:
 - Skills der Entwicklungsmannschaft
 - Entwicklungsprozess
 - Entwicklungsumgebung (Werkzeuge)
 - Konfigurationsmanagement-System
 - Qualitätssicherungsprozess und Qualitätswerkzeuge (z.B. zum Testen)
 - Release-Prozess
3. Anforderungen:
 - Sind die Anforderungen strukturiert erfasst?
 - Liegt eine Priorisierung der Anforderungen vor?
 - Sind bekannte Fehler und Probleme strukturiert erfasst und priorisiert?
4. Produktroadmap:
 - Was ist die Zielarchitektur?
 - Welche High Level Anforderungen sollen bis wann umgesetzt sein?
 - Identifikation der kritischen Komponenten und Ersetzungsstrategie
 - Kostenaufstellung/Risiko (kritische Erfolgsfaktoren) Neuentwicklung vs. R & R

Nach der durchgeführten Bestandsaufnahme und dem Abwägen aller Ergebnisse kann es durchaus sein, dass in gewissen Fällen eine Neuentwicklung doch vorzuziehen ist.

Bei dieser Entscheidung spielen insbesondere Kosten und Risiken eine entscheidende Rolle. Demnach müssen alle oben genannten Punkte sorgfältig, gewissenhaft und mit einem hohen Maß an Selbstkritik analysiert werden, um eine geeignete Entscheidungsgrundlage zu bieten.

4. Modernisierung und Neuentwicklung im Überblick

Tabelle 1. Gegenüberstellung von Modernisierung und Neuentwicklung im Überblick

Neuentwicklung	Modernisierung
Höhere Kosten, da i. d. R. nach Fertigstellung mehrere Releasezyklen stattfinden, um die bis dahin versteckten und unbekanntem Anforderungen zu ergänzen	Geringere Kosten, da kleine Änderungen viel überschaubarer sind und damit i. d. R. sofort genau auf die Anforderungen der Anwender zugeschnitten werden können
Kosten fallen geballt an	Kosten verteilen sich
Alles oder nichts Prinzip	Jederzeit ein Exit möglich, ohne die bisherige Investition verwirkt zu haben
Umstellung auf neue Technologie dauert so lange, bis das Gesamtsystem fertig ist	Integration neuer Technologie ins aktuelle System mit bestehenden Technologien aufwendig, kommen dennoch an den entscheidenden Punkten früher zum Einsatz
Alle Probleme können auf einmal beseitigt werden	Probleme werden nach und nach beseitigt
Entweder Stillstand der Entwicklung am alten System bis das neue System komplett fertig ist (im allg. monatelang) oder Neuentwicklung mit einem "moving target", d.h. permanentes Hinterherlaufen der Neuentwicklung hinter dem Altsystem, das auch noch weiterentwickelt wird. Das kann im Extremfall dazu führen, dass die Neuentwicklung nie fertig wird.	Verbesserungen werden schrittweise in das Produktionssystem übernommen, dadurch profitieren die Nutzer frühzeitig und die Entwicklung kann nicht auseinanderlaufen. Die Wahrscheinlichkeit eines termingerechten Projekterfolgs ist hoch.
Profitiert kaum von Altsystem	Sicherung der Investition ins Altsystem
Relativ hohes Projektrisiko	Relativ geringes Projektrisiko