



Denken Sie gerade an Ihr IT-System?

# »HISTORISCH GEWACHSEN ...«

**Erfolgreiche Software** lebt häufig viel länger als ursprünglich gedacht. Kontinuierliche Erweiterungen führen zu einem Zustand, den Praktiker gerne als »historisch gewachsen« bezeichnen. Das äußert sich in inkonsistenter User Experience, leidender Qualität, steigenden Wartungskosten und mangelnder Innovationsfähigkeit. Somit stellt sich fast jede Softwarefirma irgendwann die Frage, wie die eigene Software erfolgreich renoviert werden kann und wie tiefgreifend der Eingriff werden muss oder darf, um auch in Zukunft die Geschäftsziele zu erreichen. Um die Weichen der Renovierung richtig zu stellen, ist es notwendig, den tatsächlichen Zustand der Software genau zu analysieren, vor allem weil dieser erfahrungsgemäß deutlich von früheren Plänen und Dokumenten abweicht. Ein Softwaresystem erfolgreich zu renovieren erfordert eine integrierte Betrachtung des Zielzustands und des Migrationspfades, das behutsame Vertrautmachen von Entwicklern und Nutzern mit neuen Konzepten und das kontinuierliche Management von Risiken und Zielen. Softwarerenovierung betrifft somit alle, die auch in Zukunft mit ihrer Software Maßstäbe setzen wollen.

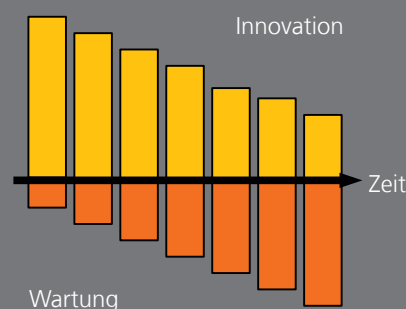
## INNOVATION STATT WARTUNG

Software ist DAS Mittel, mit dem heute Innovation geschaffen wird. Leider wird es über die Lebenszeit eines Softwaresystems immer schwieriger, innovativ zu bleiben und auf neue Anforderungen und Gegebenheiten zu reagieren. Software wird immer weiterentwickelt und neue Features werden hinzugefügt, auch wenn sie nicht in das ursprüngliche Konzept passen. So altert Software spürbar und bringt vielfältige Probleme mit sich. Für die Nutzer ist die User Experience einerseits nicht mehr einheitlich und andererseits nicht mehr zeitgemäß. Für die Entwickler wird die Software immer schwerer zu verstehen und Änderungen werden fehleranfälliger, wodurch mehr und mehr Zeit für Wartungsaktivitäten aufgebracht werden muss. In der Konsequenz bleibt immer weniger Zeit für die Entwicklung von innovativen Features und gerade bahnbrechende Innovation ist nahezu unmöglich.

Unternehmenssoftware überdauert häufig mehrere Jahrzehnte. Die Weiterentwicklung von Technologien und die zunehmende Durchdringung des beruflichen und privaten Lebens durch IT in dieser Zeit ist enorm. Noch immer laufen sehr viele Anwendungen in Unternehmen auf Mainframes und verursachen immense Wartungs- und Betriebskosten. Ganz klar ist deshalb die Tendenz, Software auf kostengünstigere und aktuelle Standardhardware zu portieren, zumal immer weniger Entwickler für alte Technologien zur Verfügung stehen. Alle Softwarefirmen stehen früher oder später vor dem Problem, »historisch gewachsene« Software zu renovieren. Die Herausforderungen reichen von unpassender Architektur und Code-Qualität über veraltete und ausgestorbene Technologien bis hin zu unrealisierbaren Anforderungen. Das Ziel ist es immer, innovativ und wettbewerbsfähig zu bleiben.

### »SOFTWARE ALTERT DOCH!«

Hartnäckig hält sich das Gerücht, Software könne nicht altern. Das gilt höchstens für den Code als solchen. Spätestens die Weiterentwicklung von Technologien und Erwartungen im Umfeld eines Softwaresystems führen dazu, dass die Software gefühlt altert. Vor allem kontinuierliche Weiterentwicklung mit vielen Kompromissen führt im Lebenszyklus einer Software meist zu einer Situation, in der mehr und mehr Aufwand für Wartung statt für Innovation aufgewendet werden muss.





# Beim Renovieren gibt es immer Überraschungen

»Erfolgreiche Softwareprodukte werden zu Treibern der ›Digitalen Transformation‹ von Unternehmen. Sie bleiben aber nur dann erfolgreich, wenn sie kontinuierliche Innovation erlauben.«

Werner Weiss  
CEO, Insiders Technologies

»Softwarerenovierung ist wie die Renovierung eines weiterhin genutzten Gebäudes, in dem die Bewohner nicht beeinträchtigt werden dürfen.«

Timo Rihtnieni  
Manager Product Architecture  
Tekla

## NEUBAU, UMBAU ODER NUR NEU STREICHEN?

Die Renovierung eines Softwaresystems kann je nach Ausgangszustand und Zielsetzung sehr unterschiedlich ausfallen. Viele Firmen denken zunächst über ein Refactoring nach, was relativ kostengünstig und lokal durchzuführen ist. Leider sind auch die Verbesserungseffekte überschaubar, weil globale Herausforderungen so nicht gelöst werden können. Deshalb stellt sich dann häufig die Frage, ob ein System mit einer Neuausrichtung der Architektur umgebaut werden soll oder ob es sich sogar lohnt, ein komplett neues System zu entwickeln. Die komplette Neuentwicklung ist in der Praxis oft aber auch keine

Option, weil die Entwicklungsmannschaft nicht gleichzeitig das alte System warten und das neue entwickeln kann.

Softwarerenovierung ist oft unumgänglich und mit Risiken verbunden. Viele Entscheidungen über das zukünftige Produkt und den Entwicklungspfad dorthin sind notwendig. Diese Entscheidungen reichen von Features und Interaktionskonzept über die zukünftige Architektur bis hin zur Art der Qualitätssicherung. Deshalb ist eine umfangreiche Analyse der Vorgeschichte und eine gute Planung der Renovierung notwendig.

### »SOFTWARE IST NICHT SOFT!«

Eine der größten Errungenschaften von Software ist, dass man sie ändern kann, ohne physikalische Änderungen vornehmen zu müssen. Das hat dazu beigetragen, dass alle Aspekte, bei denen Änderungen an einem System vorgenommen werden müssen, heute soweit wie möglich über Software abgebildet werden. Das gilt für Software in Unternehmen genauso wie für Software im Auto.

Obwohl Software an sich leicht und in nahezu jede Richtung geändert werden kann, ist dies in der Praxis meist nicht möglich. Viele Änderungen erstrecken sich über weite Teile des Systems, haben große und unerwartete Nebeneffekte und machen es sehr schwer, wieder einen Zustand hoher Qualität herbeizuführen.

## KEINE RENOVIERUNG OHNE ANALYSE DER VORGESCHICHTE


»Unsere Software muss nach der Renovierung mindestens das Gleiche können wie jetzt!« ist ein häufiger Satz zu Beginn einer Renovierung. Das liegt einerseits daran, dass diese Anforderung sehr leicht zu formulieren ist und niemand exakt sagen kann, was die Software wirklich tut. Andererseits fällt es vielen Firmen sehr leicht, neue Features hinzuzufügen, aber sehr schwer, auf existierende Features zu verzichten. Die Erfahrung zeigt aber, dass eine Renovierung immer auch mit einer Konsolidierung der Features einhergehen sollte.

Die Rekonstruktion von umgesetzten Anforderungen und Interaktionskonzepten ist aufwändige manuelle Arbeit, die Domänenverständnis erfordert. Die existierende Code-Basis ist oft groß und schwer zu durchschauen. Mithilfe von Reverse Engineering Tools kann der Code teilautomatisiert untersucht werden. Erst das Festhalten der gefundenen Informationen in einer redokumentierten Architektur schafft eine Abstraktionsebene, auf der die Komplexität des zu renovierenden Systems beherrscht werden kann.

Fast nie kann eine Renovierung auf einer konsistenten Dokumentation der Software aufsetzen. Das heißt, dass eine Analyse nicht nur den aktuellen Zustand der Software untersuchen, sondern meist zuerst auch rekonstruieren muss. Dabei ist der Source Code oft die einzige verlässliche Wahrheit und Quelle.

Es ist wichtig, die Analyse des existierenden Systems als Investition für die erfolgreiche Renovierung zu begreifen. Die Analyse muss dabei ganzheitlich erfolgen und alle Aspekte rund um Nutzung, Betrieb und Entwicklung des Systems methodisch abdecken.





# Komplexe Renovierungen benötigen Engineering Tools

»Eine durchdachte und möglichst stringent umgesetzte Architektur erspart kostspielige Reparaturen. Der nicht zu vermeidenden, natürlichen Degeneration wirkt man dann im Rahmen wertbringender Umbaumaßnahmen entgegen.«

Dr. Dirk Muthig  
Head of Product and Systems Design  
Lufthansa Systems

## BAUEN KANN JEDER. RENOVIEREN KÖNNEN NUR SPEZIALISTEN.

Viele Renovierungen von Softwaresystemen scheitern, obwohl sie notwendig wären. Während eine Neuentwicklung aus vielerlei Gründen oft schnell ausgeschlossen wird, erscheint eine Renovierung als gangbarer und steuerbarer Weg, um ein Softwaresystem wieder in die richtige Spur zu bringen. Dies führt aber oft zu einer zu niedrigen Priorität und zu halbherzigem Vorgehen. Dann erscheint manchmal schon die Analyse als zu große Hürde.

Softwarerenovierung muss als strategische Aufgabe angegangen werden und erfordert den Einsatz von fachlichen und methodischen Spezialisten. Aufbauend auf der Analyse der Vorgeschichte des Softwaresystems wird der neue Zielzustand konstruiert. Während eine Neukonstruktion mit bedeutend weniger Einschränkungen arbeiten kann, muss eine Renovierung auf das existierende Softwaresystem Rücksicht nehmen. Somit gibt es eine ständige Abwägung zwischen den Umbaukosten und dem neu entstehenden Nutzen, die schwer zu quantifizieren ist.

Ganzheitlich müssen Features und Qualitätsanforderungen aus den Blickwinkeln Benutzung, Betrieb und Entwicklung berücksichtigt werden. Renovierung muss die externe Gestaltung eines Softwaresystems im Sinne von Interaktionsdesign und visuellem Design integriert mit der internen Gestaltung im Sinne der Softwarearchitektur betrachten. Die Entscheidungen über zukünftige Features, Oberflächen und Interaktionen betreffen viele Stakeholdergruppen und sollten nicht einseitig getroffen werden (z.B. nur durch den Vertrieb). Insbesondere sollten sie durch Fakten untermauert sein, wie z.B. durch die Messung der tatsächlichen Nutzung.

Die Herausforderung liegt nicht nur im Design eines Zielzustandes für das Softwaresystem, sondern auch im darauf abgestimmten Design eines Migrationspfades. Renovierung findet fast immer parallel zur Weiterentwicklung des Systems statt und deshalb müssen beide Aktivitäten koordiniert werden, sodass eine schrittweise Renovierung möglich wird. Renovierung beinhaltet auch viel Change Management. Nicht nur die Software ändert sich, es gibt auch Auswirkungen auf die Nutzer, die Entwickler, das Betriebspersonal und den Vertrieb. Änderungen, auch wenn sie Verbesserungen darstellen, werden oft als negativ wahrgenommen. Deshalb müssen diese Stakeholdergruppen rechtzeitig involviert werden und brauchen abhängig von der Softwareänderung auch neue Qualifikationen oder zumindest Training. Stetige und zielgerichtete Migration stellt sehr hohe Anforderungen an die Qualitätssicherung und erfordert insbesondere automatisierte Tests, um die Auswirkungen von Änderungen überprüfen zu können.

Die Renovierung eines Softwaresystems ist immer komplex und individuell und es gibt dafür keinen Königsweg. Das Fraunhofer IESE baut daher auf Erfahrung aus über 100 Renovierungsprojekten und stellt einen gut gefüllten Werkzeugkasten an Methoden und Tools zur Verfügung.

Das Fraunhofer IESE hat viele Softwarefirmen bei der Renovierung ihrer Softwaresysteme unterstützt und forscht an weiteren Verbesserungen von Methoden und Tools für die Softwarerenovierung.

■ Marcus Trapp, Matthias Naab

### »NACH DER RENOVIERUNG IST VOR DER RENOVIERUNG!«

Auch nach einer erfolgreichen Renovierung eines Softwaresystems dreht sich die Welt weiter, es kommen neue Anforderungen hinzu und neue Technologien stehen zur Verfügung. Deshalb sollte Renovierung nicht als einmalige Aktion, sondern vielmehr als kontinuierliche Aktivität aufgefasst werden, die je nach benötigtem Grad entsprechend ausgestaltet werden kann. Trotzdem sollte auch die Option einer kompletten Neuentwicklung nicht immer kategorisch ausgeschlossen werden, weil eine Renovierung eine Renovierung bleiben sollte und nicht den Umbau auf ein komplett anderes System verfolgen sollte.