

Risikomanagement in der Software-Entwicklung

Risikomanagement ist eine der besten Methoden, um den Aufwand in einer Software-Entwicklung effizient und an den richtigen Stellen einzusetzen. Damit lässt sich auf einfache Art und Weise viel Geld sparen!

Welche Themen im Software-Entwicklungsprozess besondere Risiken bergen und wie man einfaches Risikomanagement betreiben kann, beschreibt dieser Artikel.

Ineffiziente Software-Entwicklung

Wir wissen, dass ein Großteil der Software-Entwicklungsprojekte nicht so läuft, wie man es sich am Anfang vorstellt. Das hat auch wenig damit zu tun, ob man nun agil oder plangetrieben vorgeht, sondern kommt in beiden Welten vor.

Interessanterweise kennen wir aber auch eine Unmenge an Methoden und Prinzipien, wie man schlechte Projektabwicklung vermeiden könnte. Diese werden auch in vielen Ausbildungen gelehrt. Also muss man sich die berechtigte Frage stellen:

„Wieso scheitern die SW-Projekte nach wie vor?“

Diese Frage zu beantworten ist nicht einfach, da es eine Unzahl von Einflussfaktoren gibt, angefangen von schlechten Requirements über mangelnde Kommunikation bis hin zu unzureichender Qualitätssicherung.

Leider werden Methoden und Prinzipien guter Software-Entwicklung meist nicht effektiv angewendet und wenn sie zur Anwendung kommen, oft als Vorgabe oder Richtlinie gleichmäßig über den gesamten Prozess „ausgegossen“. Z.B. „Kommuniziere intensiv mit deinem Kunden.“, „Teste die User Stories mit den Testmethoden xyz“. Dieses Gießkannenprinzip ist nicht effektiv.

Eine sehr gute Möglichkeit, die Aufwände in der Software-Entwicklung in die richtigen Bahnen zu lenken, ist Risikomanagement. Dadurch lassen sich die Aufwände gezielt an den Stellen intensivieren, an denen es notwendig ist und an anderen Stellen kann bewusst gespart werden.

Risikomanagement in der Software-Entwicklung

Es tauchen im Zusammenhang mit Risikomanagement in der SW-Entwicklung jedoch auch einige Fragen auf:

Wie finde ich überhaupt die Risiken im Projekt / Entwicklung?

Eine sehr einfache und effektive Methode ist es, einige wenige Experten, die in ihrer Zusammenstellung sowohl die fachliche als auch die technische Seite gut abdecken können, in einem Risiko-Meeting die Risiken zu erheben und einstufen zu lassen. Bis auf die Projektsituationen, in denen eine detaillierte Risikoanalyse z.B. zur Haftungsminimierung und zur Einhaltung von anzuwendenden Standards erforderlich ist, wird es meist ausreichen, die Risiken grob in 3 Ebenen (High, Medium, Low) zu erfassen und zu bewerten.



Software-Entwicklung
und Testen für
Erwachsene!

"Risikomanagement ist Projektmanagement für Erwachsene" ist die Meinung der beiden Software-Entwicklungs-Gurus Tom DeMarco und Timothy Lister. Leider ist das Thema systematisches Risikomanagement in der Software-Entwicklung nicht besonders weit verbreitet bzw. in Vergessenheit geraten.

Intuitiv haben zwar alle Software-Entwicklungs-Beteiligten auch ein Risiko-Verständnis. Aber durch eine systematischere Herangehensweise und Anwendung könnte die Effizienz im Software-Entwicklungsprozess meist noch deutlich gesteigert werden.

Stellen Sie sich vor, die Produktverantwortlichen, Entwickler und Tester würden ganz gezielt nur diejenigen Software-Teile sehr detailliert in allen Varianten, Sonderfällen, etc. dokumentieren und testen, die auch ein sehr hohes Risiko haben und „Low-Risk“-Teile werden z.B. nur mit einem Standard-Durchlauf beschrieben und getestet.

Dies lenkt den Aufwand gezielt auf die Teile der Software, für die es notwendig ist und senkt damit auch das Schadens- und Haftungsrisiko systematisch.

Nutzen Sie daher das Potential, das in einem guten und systematischen Risikomanagement steckt, für Ihren Software-Entwicklungsprozess und betreiben Sie „Software-Entwicklung für Erwachsene“.

Dipl.-Ing. Johannes Bergsmann

Berater, Trainer, Ziviltechniker für Informatik

Der Quality Knowledge Letter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit dem Schwerpunkt SW-Qualität.

Weitere Infos zu diesem und anderen Themen finden Sie auf www.software-quality-lab.com

Welcher Detaillierungsgrad soll angewendet werden?

Bei der Analyse der Risiken muss man sich die Frage stellen, auf welcher Ebene der Software und in welchem Detaillierungsgrad man dies durchführt. So könnte man die Software als Ganzes risikomäßig bewerten oder auch jedes einzelne Requirement und jede Funktion in der Software.

Ein möglicher Weg ist ein mehrstufiges Top-Down-Vorgehen.

Dabei wendet man die Risikobewertung zuerst auf die oberste Zerlegungsebene an (meist größere Software-Module oder Funktionsblöcke). Diese werden nach der Expertenbewertung in die 3 Risikoklassen H, M, L eingeteilt.

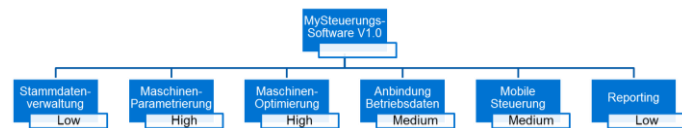


Abb. 1 - Einstufung der ersten Software-Zerlegungs-Ebene

Danach werden die Blöcke mit der **Einstufung H** zerlegt und in der nächsten Ebene nach demselben Schema bewertet. Darin dann wiederum alle H bewerteten Teile und so fort, bis man auf einer Ebene ist, an der eine weitere Zerlegung keinen Mehrwert mehr bringt (das könnte z.B. die Ebene der User-Stories oder Features sein). Bei einer Einstufung H muss auch mindestens 1 Element der nächsten Ebene mit H eingestuft sein.

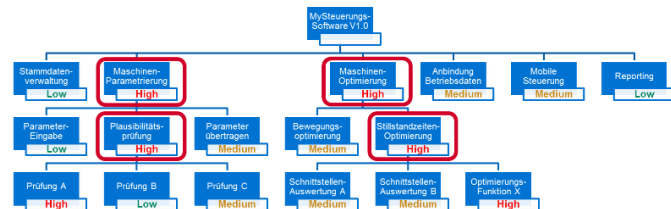


Abb. 2 - Weitere Zerlegung der mit H eingestuften Elemente

Die mit der **Einstufung M** versehenen Teile werden ebenfalls in der nächsten Ebene nach demselben Schema bewertet. Darin dann wiederum alle M bewerteten Teile und so fort, bis man auf einer Ebene ist, an der eine weitere Zerlegung keinen Mehrwert mehr bringt (das könnte z.B. die Ebene der User-Stories oder Features sein).

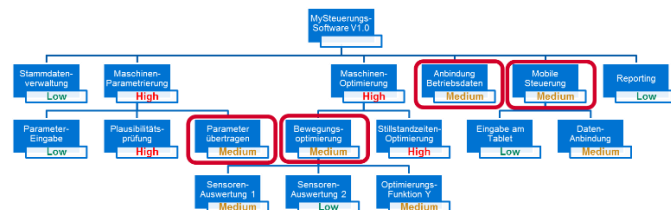


Abb. 3 - Weitere Zerlegung der mit M eingestuften Elemente

Die mit der **Einstufung L** versehenen Teile werden nicht weiter zur Risikoanalyse zerlegt und analysiert.

Mit diesem Ansatz kommt man relativ schnell zu einer Risikoübersicht des gesamten Software-Systems.

Was ist die passende Methode zur Risikoeinstufung bzw. -bewertung?

Wie schon gesagt ist die Experten-Einstufung eine sehr einfache und effektive Methode.

Es reicht meist aus, die Risiken grob in 3 Ebenen (High, Medium, Low) zu erfassen und zu bewerten (siehe auch Abb. 4).

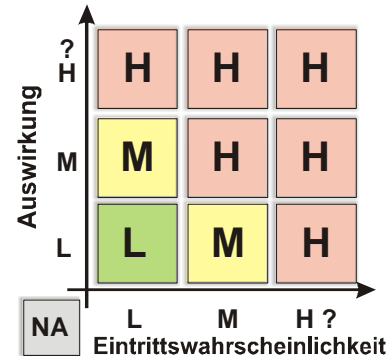


Abb. 4 - Einfaches Risikobewertungs-Schema

Wenn man es noch etwas feiner machen möchte kann ev. noch eine Klasse **Critical** dazu genommen werden.

Ob man die Auswirkung und die Eintrittswahrscheinlichkeit bei der Analyse extra auswertet oder gleich einen Gesamtrisikowert bildet, ist theoretisch zwar relevant, in der Praxis in den meisten Situationen jedoch nicht relevant. Ein Mittelweg ist wie folgt möglich: Wenn eine Einstufung in der Runde der Experten nicht von allen gleich gesehen wird, dann kann in einer zweiten Runde die Zerlegung in Auswirkungshöhe und Eintrittswahrscheinlichkeit erfolgen, was dann ev. zu einer besseren Einschätzung führt.

Was tun mit der erfolgten Risikoeinstufung?

Wenn man nun eine Risikobewertung des Systems und seiner Komponenten bis zu einer sinnvollen Detaillierungsebene vorgenommen hat, dann ist damit das Risikomanagement natürlich nicht zu Ende. Leider wird dies aber in der Praxis oft so gemacht und damit wird die Chance auf eine Effizienzsteigerung im weiteren Entwicklungsprozess vergeben.

Es ist sinnvoll, entsprechend der Risikoeinstufung eine Aufwandsverteilung im weiteren Entwicklungsprozess vorzunehmen. Risikowerte eignen sich sehr gut dazu, die Detailliertheit und den Aufwand für folgende Aktivitäten zu steuern:

- Detailliertheit der Requirements-Spezifikation
- Umfang der Architektur-Analyse und Spezifikation
- Umfang der inline Source-Code-Dokumentation
- Anzahl der Testfälle für den jeweiligen Software-Teil

Es könnte z.B. definiert werden (im agilen Umfeld in einer Definition of Done oder im plangetriebenen Vorgehen in einem Quality-Gate), dass es für die mit L eingestuften Features ausreicht, den Standardfall als Testfall durchzuführen.

Für die mit M eingestuften Features müssen z.B. der Standardfall sowie die Sonderfälle als Test durchgeführt werden und es

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.



Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter <http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/>

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Letters von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.

Software Quality Days — Die größte Konferenz zum Thema „Software Qualität“ in Europa!



Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungsergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com

