Safety als Fundus – Sicherheit als Roadmap

Sicherheitskritische Vorgaben als Goldgrube der Prozessverbesserung

Auf die Gesamtanzahl der Softwareentwicklungen bezogen, betreffen sicherheitskritische Vorgaben eine überschaubare Projektlandschaft. Betrachtet man aber die gültigen Standards, Normen und Best Practices als sorgsam und oft aus bitterer Erfahrung zusammengetragenes Prozess- und Produktwissen, öffnet sich ein wahrer Fundus an Qualitätsverbesserungsmaßnahmen.

Wen betreffen die Vorgaben

Aus Sicht der Qualitätssicherung stellt sich die Frage ob ein Produkt konform zu definierten Prozesslandschaften entwickelt werden muss oder nicht, eher selten. In klar regulierten Bereichen wie Automotive, Railway, Aviation oder Medical ist dies im Zuge einer Entwicklung entweder grundsätzlich bereits vorgesehen oder kann im Rahmen etablierter Prozesse verhältnismäßig rasch festgestellt werden.

Für den Großteil der Projekte sind sicherheitskritische Vorgaben aber kein wirkliches Thema. Zwar sind vertrags- und haftungsrelevante Vorgaben zu erfüllen (siehe Newsletter), jedoch ergibt sich nur selten die tatsächliche Notwendigkeit sich mit dem Thema auseinander zusetzen. Warum also die Büchse der Pandora öffnen?

„Warum gerade Safety als Vorlage!"

Sicherheitsrelevante Vorgaben haben einen alten Erfahrungsschatz aus unterschiedlichen Ingenieurwissenschaften auch außerhalb der Softwareentwicklung und sind branchentypisch eher träge. Viele Entwicklungen die sich in der SW-Entwicklung beobachten lassen (z.B. verstärktes Testen, nachvollziehbares Anforderungsmanagement, verstärkter Architekturfokus), sind typische Vorgaben der sicherheitskritischen Produktentwicklung.

Abb. 1: Safety Vorgaben und Maßnahmen lassen sich auf grundlegende Konzepte zurückführen

Viele grundlegenden Konzepte bieten sich außerdem sehr gut für Verbesserungen an, da primär ein nachvollziehbarer und risikobasierter Ansatz verfolgt wird (Abb. 1).

„Darf's ein bisschen mehr sein?"

Die Projektdurchlaufzeiten werden kürzer, die Anforderungen ändern sich ständig und alte Produkte müssen nebenbei trotzdem langfristig weiterentwickelt und betrieben werden können. Das Ganze im Rahmen der klassischen Projektvorgaben von Umfang, Zeit und Geld.

Was liegt näher als unnötige Prozessschritte wegzulassen, Rollen und Verantwortungen zusammen zu führen und den Kunden so nah wie möglich ins Boot zu holen?

Statt sich auf den möglichst schlanken Prozess zu fokussieren, stellt dieser Artikel jedoch die Frage ob es nicht sinnvoll wird bekannte Prozessschritte hinzuzufügen, Rollen zu trennen und klare Schnittstellen zu schaffen. Und dazu werfen wir bewusst einen Blick auf Brachen die sich typische Vereinfachung nicht leisten können, weil die gestellten Anforderungen weit über den klassischen Rahmenbedingungen für „normale" Softwareprojekte liegen.

Der sicherheitskritische Bereich ist, aus gutem Grund, nur schwer in der Lage auf Prozessschritte zu verzichten. Geltende Vorgaben sollen sicherstellen das keine Gefährdungen durch das hergestellte Produkt auftreten. Dieser Nachweis gestaltet sich aber gerade für Softwaresystem nicht trivial, benötigt daher viele aufeinander aufbauende Maßnahmen.

Fehler und Ausfälle sind aber auch in der klassischen Software-Entwicklung teuer und mit hohen Kosten verbunden, was liegt also näher als sich mit sicherheitsrelevanten Vorgaben auseinander zusetzen, um davon zu lernen.

Frei nach dem Motto: „Würde unser Unternehmen Flugzeuge bauen, würde ich mich trauen mitzufliegen ? Und Warum?"

Mag. Simon-Alexander Zerawa

Senior Quality Consultant und Trainer

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

Inhalt: fachliche Beiträge und Schwerpunktthemen, Vorstellung neuer Produkte und Leistungen, neue wissenschaftliche Erkenntnisse und andere Fachbeiträge aus unseren Themenbereichen.

Aktuelle Fach- und Forschungsbeiträge sind willkommen. Einsendungen an info@software-quality-lab.com

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

Freiwillig sicherheitskritisch ?!

Warum sollte man sich aufwändigeren Prozessvorgaben unterwerfen ohne rechtlichen Grundlagen unterworfen zu sein? Nicht nur aus der Sicht der Prozessverbesserung ist die Auseinandersetzung mit Vorgaben aus dem regulierten Bereich sinnvoll. In vielen derzeit noch nicht regulierten Bereichen ist langfristig eine zunehmende Regulierung absehbar und es macht schon aus strategischer Sicht Sinn, sich mit den Basis-Konzepten vertraut zu machen. Dies ist umso wichtiger, wenn zukünftig vorhandene oder neue Produkte im regulierten Umfeld angeboten werden sollen. In diesem Fall müssen sämtliche Rahmenbedingungen sorgfältig betrachtet werden. Typischerweise werden die Aufwände für die Prozessänderungen unterschätzt. Ein großer Teil des Entwicklungsaufwandes ist dann allein der Erfüllung geltender Vorgaben geschuldet, unabhängig von der technischen Produktentwicklung selbst.

„Wir sind nicht sicherheitskritisch!“

Die grundsätzliche Argumentation vieler Unternehmen die Software-Entwicklung betreiben geht in die Richtung „Wir entwickeln keine sicherheitskritische Software, können uns daher viele formale Prozessschritte und Methoden ersparen“. Es ist zwar nachvollziehbar, dass eine möglichst effiziente Zielerreichung angestrebt wird, dieses Vorgehen führt langfristig aber eher zum Aufbau technischer Schuld als tatsächlich zu Kostenersparnis. In vielen Fällen führen beispielsweise fehlende Risiko- und Prioritätsbetrachtungen, unzulängliche Dokumentation oder unzureichende Stakeholder-Einbindung zu gravierenden und ungeplanten Mehraufwänden.

Der steigende Stellenwert des Softwaretests ist dabei ein gutes Beispiel. Softwarefehler sind in den meisten Entwicklungsprojekten nicht unmittelbar mit Schaden für Leib und Leben verbunden, trotzdem lassen sich Ansätze aus den sicherheitskritischen Vorgaben zumindest in Ansätzen im nicht-regulierten Bereich anwenden bzw. wiederfinden. Gerade wenn Fehler teuer werden oder potentieller Imageschaden droht. Die schrittweise Übernahme von testrelevanten Prozessschritten und Methoden in der klassischen Software-Entwicklung sprechen genau für diesen Trend. Aus subjektiver Sicht ist dies auch in der zunehmenden Verbreitung von Zertifizierungsausbildungen wie dem Certified Tester der ISTQB-Zertifizierungen nachvollziehbar.

Ähnliche Beobachtungen lassen sich auch im Bereich Requirements und Architektur anstellen. Je nachhaltiger und langfristiger Projekteergebnisse von Kunden und Auftraggeber eingefordert werden, umso stärker spielen Dokumentation, Prozesskonformität und Nachweisbarkeit ein Rolle. Und genau hier schlägt sich die Brücke zu sicherheitskritischen Überlegungen.

Risikomanagement

Ausgehend von einer nachvollziehbaren Risikoanalyse der zu realisierenden Anwendung unterscheiden sich die meisten Safety-Normen in unterschiedlichen Risiko-Klassen. In diesem Kontext wird der Safety Integrity Level (SIL) oder im Automotive-Sektor der Automotive Safety Integrity Level (ASIL), meist begleitet durch eine entwicklungsexterne Instanz, bestimmt. Ausgehend von diesem Level ergeben sich vorgeschriebene oder teilweise optionale Maßnahmen(-gruppen) die auf eine Risiko-minimierung auf ein tolerierbares Restrisiko abzielen.

Während diese Schritte im sicherheitsrelevanten Bereich sehr transparent und dokumentiert ablaufen müssen, findet sich im nicht-regulierten Kontext ein sehr unterschiedliches Vorgehen. So werden zwar oft Risikoüberlegungen angestellt aber durchaus nicht unter Berücksichtigung aller relevanten Stakeholder oder teilweise nur auf sehr abstrakter Projekt- oder Produktebene. Auch die wiederholte und auf konkrete Maßnahmen ausgerichtete Durchführung wird zum Teil vernachlässigt.

Für viele Entwicklungstätigkeiten stellt eine fundierte Risikobetrachtung jedoch eine maßgebliche Unterstützung dar. Aus Sicht der Prozessverbesserung eignet sich daher gerade auch im nicht-regulierten Bereich die Etablierung eines transparenten Risikomanagements. Ein allgemeiner Ansatz für einen Risiko-Management-Prozess bestehend aus Identifikation, Bewertung, Steuerung und Controlling verringert die Gefahr übersehener und unbehandelter Risiken (Abb. 2).

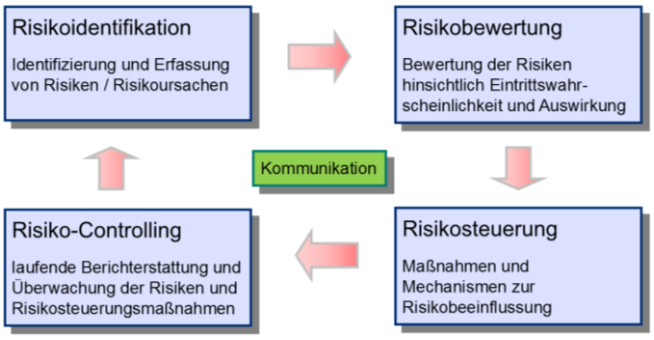

Abb. 2: Allgemeiner Risiko-Management-Prozess

Prozesskonforme Entwicklung

Eine der fundamentalen Annahmen im Kontext von Softwarequalität geht davon aus, dass die Qualität des entwickelten Produktes durch Tests allein nicht ausreichend nachgewiesen werden kann, sondern sich auch die Sorgfalt bei der Entwicklung maßgeblich auf die Qualität auswirkt. Da diese Sorgfalt im Nachhinein nicht am Software-Produkt selbst „gemessen“ werden kann, ist im regulierten Bereich hierfür ein separater Nachweis einer Prozesskonformität erforderlich, typischerweise als schriftliche Dokumentation. Ähnliche Vorgaben finden sich

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 Lettern 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 Forschungs-ergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com