

In dieser Ausgabe:

- ⇒ Was ist Software-Validierung
- ⇒ Durchführung der Software-Validierung (Fallstricke und Tipps)
- ⇒ GAMP4 - Ein Standard zur Software-Validierung
- ⇒ Literatur und Zitate

Was ist Software-Validierung?

Die Validierung von Systemen und Software (als ein Teil des Systems oder auch als eigenständiges System betrachtet) ist heute vor allem in (personen-) sicherheitskritischen Prozessen und Umgebungen gefordert und etabliert.

Viele Branchenbereiche wie chemische und pharmazeutische Industrie, Medizin, aber auch Automobilbau, Luftfahrt, ... praktizieren die Software-Validierung bereits seit vielen Jahren.

In der „klassischen“ und „kommerziellen“ Software-Entwicklung ist das Thema Software-Validierung jedoch kaum etabliert.

In diesem Zusammenhang sei auch angemerkt, dass die in diesem Newsletter erläuterte „Software-Validierung“ (in Form eines systematischen Vorgehensmodells) zu unterscheiden ist von dem Begriff „Validierung“ wie er z.B. in CMMI, SPICE oder anderen bekannten Modellen der Software-Entwicklung verwendet wird (dort bedeutet Validierung: „Das RICHTIGE zu tun.“ - im Vergleich zu Verifizierung: „Die Dinge richtig zu tun.“). Die Validierung in diesem Fall wird hauptsächlich durch Feedback-Techniken und Methoden wie Reviews, Kundengespräche, ... umgesetzt und ist mehr als kommunikatives Element bzw. Aktivität in den Vorgehensmodellen (zur besseren Einbindung des Kunden) zu sehen.

Die Software-Validierung, die in diesem Newsletter beschrieben wird, hat zum **Ziel**, den dokumentierten Nachweis zu erbringen, dass alle Teile eines Systems nach dessen Inbetriebnahme mit hoher Wahrscheinlichkeit dauerhaft korrekt arbeiten.

Dies wird einerseits durch klassische Nachweis-**Techniken** wie z.B. Testen oder methodische Reviews dokumentiert.

Andererseits liegt bei der Software-Validierung auch ein sehr großer Schwerpunkt im Bereich der konstruktiven Qualitätstechniken.

Vor allem auf eine systematische und frühzeitige Planung der gesamten Validierungsaktivitäten sowie auf eine durchgängige und im Software-Entstehungsprozess nachvollziehbare Spezifikationskette wird großer Wert gelegt.

In diesem Knowledge-Letter werden auszugsweise einige wesentliche Aspekte der Software-Validierung aufgezeigt.

Weiterführende Informationen erhalten Sie gerne auf Anfrage.

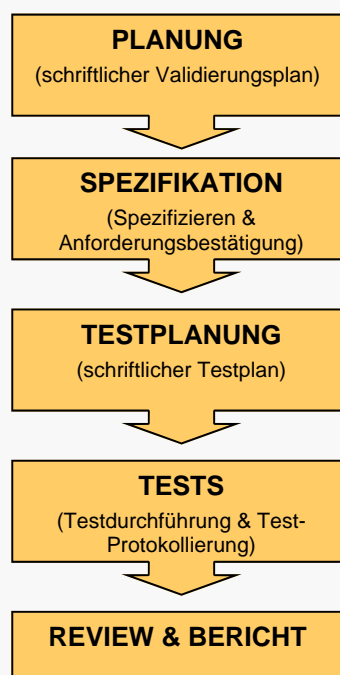


Abb.1: Allgemeiner Ablauf der Validierungsaktivitäten
Quelle: GAMP4-Leitfaden



Gezungen zur guten Projekt-Praxis?

Bei der Durchführung von Software-Projekten, in denen die (rechtlich zwingend vorgeschriebene) SW-Validierung ein Thema ist, stöhnen die Verantwortlichen oft unter der Last, die Ihnen durch die Behörden aufgebürdet wird und schwärmen, wie einfach die Projekte doch in anderen Bereichen sind.

Doch was fordert die SW-Validierung Außergewöhnliches?

- ⇒ Nachvollziehbare Projektplanung
- ⇒ Eine gute und mit dem Kunden abgestimmte Anforderungsspezifikation
- ⇒ Eine Risikoanalyse, um Aufwände gezielt auf risikoreiche Bereiche zu lenken
- ⇒ Eine systematische Ableitung des SW-Designs
- ⇒ Kontrollierte Implementierung
- ⇒ Methodische u. nachvollziehbare Tests
- ⇒ Kontrolliertes Change-Management
- ⇒ Kontrollierte Lieferanteneinbindung

Was ist das Besondere an diesen Forderungen, das nicht auch in jedem anderen Projekt sinnvoll wäre?

Grundsätzlich ist aus meiner Sicht anzumerken, dass Software-Validierung meist nichts Außergewöhnliches und zusätzlich Belastendes ist, sondern eigentlich (primär durch die Aufsichtsbehörde) einem Projekt aufgezwungene gute Projektpraxis, wie sie eigentlich in den meisten „normalen“ Projekten auch angewandt werden sollte, jedoch mangels Zeit und Einsatz der Beteiligten meist nicht gemacht wird (mit den bekannten Auswirkungen!).

Dipl.-Ing. Johannes Bergmann

Staatl. befugter und beideter Ingenieurkonsulent für Informatik

Der Quality-Knowledgeletter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit den Schwerpunkten IT-Qualitätsmanagement, Projekt- und Prozess-Management.

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.at.

Weitere Infos zu diesem und anderen Themen finden Sie auf <http://www.software-quality-lab.at>.

Durchführung der Software-Validierung

von Dipl.-Ing. Johannes Bergmann

Durch die zuständigen Aufsichtsbehörden wird in verschiedenen - primär in personen-sicherheitskritischen - Bereichen bei Einführung oder Veränderung von Software-Systemen eine Validierung der Software durch den Systembetreiber gefordert.

Leider wird dies auf Seiten der Systembetreiber oft zu spät oder methodisch falsch angegangen, so dass in derartigen Projekten oft im Nachhinein große nicht kalkulierte Aufwände zur Erfüllung der Validierungs-Anforderungen auf den Betreiber zukommen oder die Validierung nur mehr retrospektiv möglich ist.

Der nachfolgende Beitrag zeigt, welche Fallstricke bei Software-Projekten, in deren Rahmen eine Validierung notwendig ist, auftauchen können und wie sie vermieden werden können.

Projekt-Planung

Das Projekt wird nach bestem Wissen geplant und aufgesetzt. Auch Validierungsaufwände werden eingeplant, oft jedoch als „Software-Validierung“ am Ende des Projekts oder im Rahmen der Testphasen des Projekts.

Probleme:

- ⇒ Es wird übersehen, dass die SW-Validierung nicht nur aus dem Testen der Software besteht, sondern eigentlich den kompletten Lebenszyklus des Projekts umfasst.
- ⇒ Daher fehlen dann am Schluss des Projekts wesentliche Validierungs-Aktivitäten und –Nachweise, die nur mehr mit großem Aufwand retrospektiv erstellt werden können.

Tipps:

- ⇒ Planen Sie die Validierung nicht als eigenes „Validierungs-Projekt“, sondern als integralen Bestandteil des Projekts.
- ⇒ Im Idealfall sollte es gar nicht als eigenständiger Projekttitel in der Projektplanung aufscheinen, sondern in jeder geplanten Aktivität des Projekts von vornherein entsprechend berücksichtigt werden.
- ⇒ Der Validierungsplan sollte so gestaltet werden, dass er sich „nahtlos“ in die Projekt-Struktur und -planung einfügt.
- ⇒ Es sollte kein eigenes „Validierungs-Projekt“ oder eine parallel zum Projektteam existierendes „Validierungs-Team“ aufgesetzt werden. Der (Gesamt-)Projektleiter sollte auch die Verantwortung für die Validierung haben und die mit der Validierung befassten Personen sollten entsprechend in die Projektorganisation integriert sein.

Validierungs-Plan

Es wird ein Validierungsplan erstellt, der eigenständig die Software-Validierungs-Aktivitäten abdecken soll und sich an bestimmten Standards und regulativen Vorgaben orientiert.

Probleme:

- ⇒ Der Validierungsplan wird vielfach nicht mit dem Projektplan abgestimmt bzw. in diesen integriert.
- ⇒ Im Validierungsplan fehlen wesentliche Aspekte, die für eine Software-Validierung notwendig sind.
- ⇒ Der Validierungsplan übernimmt alle Aspekte der geltenden oder gewählten Standards, ohne die jeweilige Organisations- und Projekt-Situation zu berücksichtigen.

Tipps:

- ⇒ Der Validierungsplan ist die zentrale „Drehscheibe“ der Validierung.
- ⇒ Er sollte alle Validierungsanforderungen und –aktivitäten berücksichtigen.
- ⇒ Um eine effektive Validierung durchführen zu können, muss der Validierungsplan mit dem Projektplan abgestimmt und in diesen integriert sein.
- ⇒ Die Validierung hat zwar Schwerpunkte im Bereich des Testens / Abnahme, jedoch sollte die Validierung ebenfalls schon mit dem Projektstart beginnen.
- ⇒ Validierungs-Standards (wie z.B. GAMP) sind typischerweise Good-Practice-Modelle, die eine Fülle von Vorschlägen und Anleitungen enthalten, wie Projekte umgesetzt werden *sollten*, aber nicht unbedingt *müssen*. Wesentlich für der Validierungsplanung ist, dass bezogen auf die jeweilige Projektsituation entsprechende Anpassungen und ev. auch Ausschlüsse der zugrunde liegenden Standards definiert werden, um neben einer dem Stand der Technik entsprechenden Umsetzung auch die Wirtschaftlichkeit des Projekts zu beachten.

| 8.3. Validierungslebenszyklus allgemein | | Validierungsphase |
|---|--|---|
| Diese Tabelle zeigt den grundsätzlichen Ablauf der Validierung. | | |
| Validierungsphase | Aktivitäten | Anmerkung |
| Planung / Entwurf | in Projekten in der Produkt-Entwicklung | |
| Planung | GMP Risk Assessment (GPA) / GPR Projekt I für die projektleisenden Teile User Requirements Specification (URS) Projektplan | nur intern aus Sicht des Herstellers, erst bei nicht-kunden GMP-Risk-Assessment |
| | Process Risk Assessment (PRA) Traceability Matrix Functional Specification (FS) Software Design Specification (SDS) Release SP | Identifiziert die Integrität der Spezifikation (FS, SDS) fest |
| Spezifikation / Entwurf | System Package Specification (SPS) Release SP | |
| | Functional Risk Assessment (FRA) Test Spezifikationen Parametrierung, Customizing | Identifiziert die Testintelligenz bzw. Teststrategie in Parametrierung, Customizing |
| | Code-Reviews DQ-Reviews | nur die hersteller-lesbaren Teile |
| Umsetzung | OO-Tests | nur die hersteller-lesbaren Teile |
| | PO-Tests Bereits stellen aller Installationsrelevanten Umfragen | nur die hersteller-lesbaren Teile, die im Rahmen der Hersteller-Entwicklung möglich sind |
| Validierung | ID-Tests xQ-Test Review | nur die hersteller-lesbaren Teile (z.B. Admin-Benutzermaschine, Software, ...) nur die hersteller-lesbaren Teile (z.B. Benutzer, Betriebssystem, Updates, ...) |
| | Validierungs-Abschlussbericht | Nur die hersteller-lesbaren Teile genehmigt werden, wird auf keine Trennung im Review-Bereich für die xQ-Phase eingegangen |

Abb.2: Ausschnitt aus einem Validierungsplan, Quelle: Software Quality Lab

(Fortsetzung auf Seite 3)

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

