

## Ausgabe 2007 / 2

Erscheinungsart: ca. 4 x jährlich in elektronischer Form

# Software-Validierung

### 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.

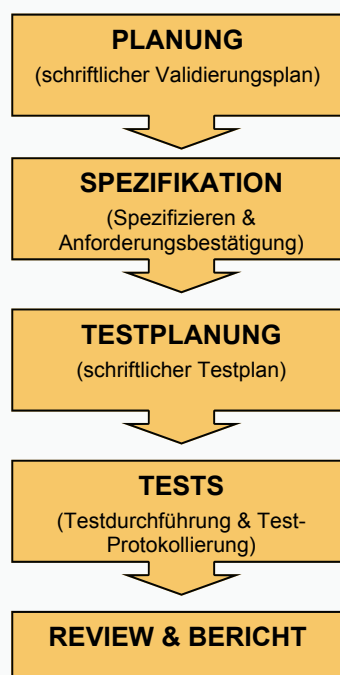


Abb. 1: Allgemeiner Ablauf der Validierungsaktivitäten  
Quelle: GAMP4-Leitfaden



**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](mailto: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 Projektteil 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**  
Diese Tabelle zeigt den grundsätzlichen Ablauf der Validierung.

Validierungsphase	Aktivitäten			Validierungsphase
	in Projekten	in der Produkt-Entwicklung	Anmerkung	
Planung / Entwurf	GMP Risk Assessment (GPA) leitet Projekt für die projektleitenden Teile	GMP Risk Assessment (GPA) für das Produkt insgesamt (einmalig bzw. bei Änderungen)	nur intern aus Sicht des Herstellers, erst danach können GMP-Risk-Assessments	Planung
	User Requirements Specification (URS) Projektplan	User Requirements Specification (URS) Releaseplan		
Spezifikation / Design	Traceability Matrix	Process Risk Assessment (PRA)	Identifiziert die Integrität der Spezifikation (FD, D) des Systems	Spezifikation
	Functional Specification (FS)	Software Design Specification (SDS)		
Umsetzung	System Package Specification (SPS)	Release SOP		Umsetzung
	Test Specifications	Functional Risk Assessment (FRA)	Identifiziert die Testintelligenz bzw. Teststrategie in der Herstellerentwicklung	
Validierung	Code-Reviews	Code-Reviews	nur die herstellereigenen Teile	Validierung
	D-D-Reviews	D-D-Reviews		
Validierung abschließen	O-D-Tests	O-D-Tests	nur die herstellereigenen Teile	Validierung abschließen
	P-D-Tests	Bereitstellen aller Installationsrelevanten Unterlagen	nur die herstellereigenen Teile, die im Rahmen der Herstellerentwicklung angedacht sind	
Validierung abschließen	ID-Tests		nur die herstellereigenen Teile (z.B. Admin-Benutzerhandbuch, Software ...)	Validierung abschließen
	X-D-Tests		nur die herstellereigenen Teile, die im Rahmen der Herstellerentwicklung angedacht sind z.B. Installier-, Betriebssystem-, Updates ...	
			Da nur die herstellereigenen Teile getestet werden, wird auf eine Trennung im Review zwischen dem X-D-Phasenprogramm und einem Review unterzogen	
		Validierungs-Abschlussbericht		

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

(Fortsetzung auf Seite 3)

(Fortsetzung von Seite 2)

- ⇒ Im GAMP-Leitfaden sind unterschiedliche Software-Kategorien definiert, die auch eine abgestufte und unterschiedlich detaillierte Vorgehensweise im Rahmen der Validierung ermöglichen.

### **Einbindung und „Validierung“ von Lieferanten**

In vielen Fällen wird die Software oder zumindest ein Teil davon von einem externen Lieferanten entwickelt, angepasst oder auch als Standard-Produkt zugeliefert.

In diesem Fall sind die zugelieferten Systeme ebenfalls einer entsprechenden Validierung sowie die Lieferanten einer systematischen Auditierung zu unterziehen.

#### **🔴 Probleme:**

- ⇒ Die Lieferantenaudits werden zu oberflächlich durchgeführt. Man verlässt sich zu sehr auf die Selbstdarstellung und den ersten Eindruck des Lieferanten.
- ⇒ Der Systembetreiber, der ja für die ordnungsgemäße Validierung verantwortlich ist, meint, um die Validierung könne sich auch der Lieferant kümmern und mit seinem Produkt auch gleich die fertige Validierung mitliefern.
- ⇒ Es wird kein passendes oder ausreichend definiertes Vorgehensmodell für die Zusammenarbeit vereinbart.
- ⇒ Testprozeduren werden nicht oder unzureichend vereinbart.
- ⇒ Inhalts- und Qualitätskriterien für die zu erstellenden Dokumente (z.B. Anforderungsspezifikationen, Testspezifikationen, Testprotokolle, Change-Requests, Benutzerdokumentation, Systemdokumentation, ...) werden nicht definiert.

#### **✅ Tipps:**

- ⇒ Im GAMP-Leitfaden wird dem Lieferantenaudit ein großer Stellenwert eingeräumt und es ist in den Anhängen sehr ausführlich beschrieben. Der System-Betreiber sollte sich daher VOR der Beauftragung eines Lieferanten durch entsprechende Audits von der Eignung des Lieferanten überzeugen und dies auch nachweislich dokumentieren.
- ⇒ Der Systembetreiber kann seine Validierungsverantwortung nicht abschieben. Sehr wohl aber kann der Lieferant einen wesentlichen Beitrag zur Validierung leisten (z.B. entsprechende Nachweise über ein konformes Vorgehensmodell, Protokollierung der Testdurchführung, ... erbringen).
- ⇒ Die oben bei den Problemen genannten Punkte müssen unbedingt klar definiert und Vertragsbestandteil sein.

### **Anforderungsspezifikation**

Im Rahmen der System-Beschaffung, -Entwicklung oder des Customizings sind entsprechende Anforderungsspezifikationen zu erstellen. Im Vorgehensmodell von GAMP wird dies typischerweise als URS (User-Requirements-Specification) bezeichnet.

#### **🔴 Probleme:**

- ⇒ Der Kunde macht sich weniger eigene Gedanken und überlässt die Erstellung der URS dem Lieferanten.
- ⇒ Es wird zwar eine URS erstellt, weil dies von GAMP so gefordert wird, jedoch wird das System anschließend ohne

weitergehende Berücksichtigung der URS (z.B. durch prototyping-orientiertes Vorgehen) entwickelt.

#### **✅ Tipps:**

- ⇒ Die Erstellung der URS kann durchaus durch iterative und agile Methoden und Prototyping unterstützt werden. Auf den Zusammenhang und die Konsistenz mit den anderen Phasen und Dokumenten muss jedoch geachtet werden.
- ⇒ Die URS soll hauptverantwortlich durch den Kunden selbst erstellt werden (ev. mit Lieferanten-Unterstützung).

### **Testen**

Das Testen ist ein wesentlicher Aspekt der Validierung. Sowohl der System-Betreiber als Verantwortlicher als auch eventuelle Lieferanten haben entsprechende Test-Aufgaben und -Verpflichtungen.

#### **🔴 Probleme:**

- ⇒ Das Testen wird durch den System-Betreiber an den Lieferanten delegiert, was jedoch in der Praxis nicht funktioniert.
- ⇒ Der Testaufwand wird massiv unterschätzt.
- ⇒ Es wird keine Testautomatisierung durchgeführt.
- ⇒ Unzureichende Testplanung und Testspezifikation.
- ⇒ Testen wird erst kurz vor dem Echtbetrieb berücksichtigt.
- ⇒ ...

#### **✅ Tipps:**

- ⇒ Klare Vorgaben für die Testdurchführung sowohl für den Systembetreiber als auch für Lieferanten definieren.
- ⇒ Die Tests systematisch planen und spezifizieren (hier ist wiederum eine gute Anforderungs- und Anwendungs-Spezifikation eine wichtige Basis).
- ⇒ Da gerade im Rahmen der Aufrechterhaltung der Validierung über den gesamten Lebenszyklus sehr große Aufwände durch Regressionstests zu erwarten sind, rechnet sich die frühzeitige Investition in Testautomatisierung in den meisten Fällen. Das Hauptproblem ist hier meist, das entsprechende Bewusstsein dafür bei den Entscheidungsträgern zu schaffen.

### **Weitere wesentliche Themen**

Im Rahmen der Validierung gibt es noch eine Vielzahl von weiteren wichtigen Themenbereichen, deren Erörterung den Rahmen dieses Newsletters sprengen würde. Daher werden einige wesentliche Punkte nachfolgend nur kurz aufgelistet:

- ⇒ Erstellung von Anwendungs- und Entwurfsspezifikationen
- ⇒ Change-Management, Risiko-Management
- ⇒ Datenmigration, Schnittstellen
- ⇒ Dokumentenqualität- und Dokumentenmanagement
- ⇒ IT-Sicherheits-Aspekte
- ⇒ Betrieb der validierten Systeme
- ⇒ Aufrechterhaltung der Validierung in Betrieb und Wartung
- ⇒ Erstellung des Validierungsberichts
- ⇒ Gesetzliche/rechtliche Aspekte (z.B. Haftung)
- ⇒ ...

# GAMP - Ein Standard zur System- und Software-Validierung

von Dipl.-Ing. Johannes Bergmann

Eine Software-Validierung kann nach unterschiedlichsten Standards und Vorschriften durchgeführt werden. Die Wahl des Standards ist primär abhängig vom Branchenbereich und von den zuständigen Aufsichts-Behörden.

Ein bekannter Standard zur System- und Software-Validierung ist GAMP4, der vor allem im medizinischen und pharmazeutischen Bereichen angewendet wird und sich dort praktisch als Defacto-Standard etabliert hat.

In diesem Artikel wird der Standard GAMP in seinen Grundzügen im Überblick vorgestellt.

Anmerkung: Dieser Artikel bezieht sich auf den Standard GAMP 4. GAMP 5 ist derzeit in Vorbereitung.

## Allgemeines und Historie

Regulatorien und Richtlinien für automatisierte Systeme in der Pharmaherstellung gibt es schon sehr lange. Jedoch waren diese bislang weniger genau als für konventionelle Bereiche. Durch den verstärkten Computereinsatz entwickelte sich seit Mitte der 90er-Jahre ein neuer Standard.

### ⇒ GAMP = Good Automated Manufacturing Practice

GAMP zielt darauf ab, Systeme, die Validierungsvorschriften unterliegen, effizient und effektiv entsprechend einer anerkannten guten Industriepraxis zu realisieren.

GAMP wurde von einer industriellen Gruppe („GAMP-Forum“) entwickelt, das aus Vertretern der Pharma-Industrie, Lieferanten und auch Inspektoren der Behörden besteht.

Das GAMP-Forum ist heute ein technisches Unterkomitee der ISPE (International Society of Pharmaceutical Engineering).

Ein vorrangiger Zweck von GAMP ist die Bereitstellung von Richtlinien für die Lieferanten von automatisierten Systemen.

Der Leitfaden liefert einen allgemeinen und verlässlichen Ansatz zur Validierung automatisierter Systeme.

## Struktur des GAMP-Leitfadens

### GAMP-Prinzipien und Rahmenwerk:

Dieser Teil besteht aus

- ⇒ Einführung und Übersicht zur Validierung
- ⇒ Lebenszyklus der Validierung
- ⇒ Management-Leitfaden für Lieferanten
- ⇒ Nutzen der Validierung
- ⇒ „Good-Practice“-Definitionen
- ⇒ ...

Zusätzlich dazu gibt es 3 Anhangs-Kapitel, die jeweils methodisch und prozess-orientiert in der Form von Good-Practice-Anleitungen verschiedene Themenbereiche beschreiben:

### Management-Anhänge:

Diese Anhänge enthalten Richtlinien für ...

- ⇒ Validierungsplanung
- ⇒ Lieferantenaudits
- ⇒ Risikobewertung
- ⇒ Software- und Hardware-Kategorien
- ⇒ Designprüfungen (Reviews) und Traceability-Matrix
- ⇒ Qualitäts- und Projektplanung
- ⇒ Erstellung des Validierungsberichts

- ⇒ Change-Control
- ⇒ Konfigurationsmanagement
- ⇒ Dokumentenmanagement

### Entwicklungs-Anhänge:

Diese Anhänge enthalten Beispiele für Vorgehensweisen und Richtlinien für ...

- ⇒ Erstellung eines Anwender-Lastenhefts (Anforderungsspezifikation)
- ⇒ Erstellung eines Pflichtenhefts (Anwendungsspezifikation)
- ⇒ Erstellung einer Hardware-Entwurfsspezifikation
- ⇒ Erstellung einer Software-Entwurfsspezifikation
- ⇒ Kodierung, Kontrolle und Prüfung von Software
- ⇒ Test eines automatisierten Systems

### Betriebs-Anhänge:

Diese Anhänge enthalten Beispiele für Vorgehensweisen und Richtlinien für ...

- ⇒ Periodische Prüfungen
- ⇒ Service-Vereinbarungen
- ⇒ Betriebliches Change-Control
- ⇒ System-Leistungsüberwachung
- ⇒ Aufbewahrung von Aufzeichnungen
- ⇒ Sichern und Wiedereinspielen von Software und Daten
- ⇒ Kontinuitätsplanung
- ⇒ ...

## Die Validierung nach GAMP

Die Validierungsvorgehensweise orientiert sich grundsätzlich an den in der System- und Software-Entwicklung bekannten Phasenmodellen.

Ergänzend zum klassischen Entwicklungszyklus wird jedoch bei der Validierung noch eine verbindliche Validierungsplanung vorangestellt.

Grundsätzlich betrifft die Validierung den gesamten Entwicklungs-Lebenszyklus und sollte daher auch schon von Beginn eines Projekts berücksichtigt werden!

Schwerpunkte der Validierung sind jedoch die sogenannten Qualifizierungs-Phasen:

(Fortsetzung auf Seite 5)

(Fortsetzung von Seite 4)

### Design-Qualification (OQ)

In dieser Phase wird der Nachweis erbracht, dass der vorgeschlagene Entwurf des Systems für den vorgesehenen Gebrauch adäquat ist.

### Installation-Qualification (IQ)

In dieser Phase muss der Nachweis erbracht werden, dass das System gemäß schriftlicher und genehmigter Spezifikation installiert wurde.

### Operational-Qualification (OO)

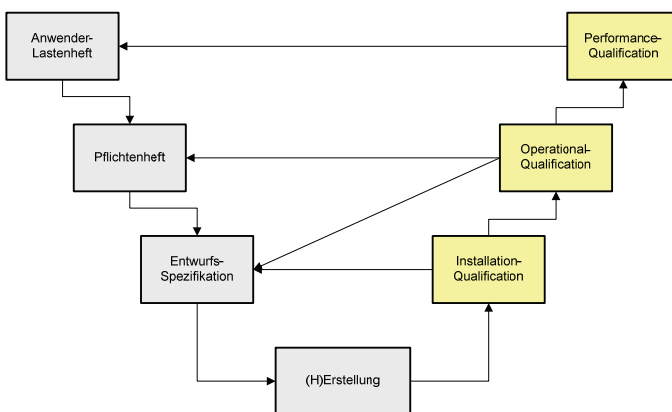
In dieser Phase wird der Nachweis erbracht, dass das System gemäß schriftlicher und genehmigter Spezifikation innerhalb der vorgesehenen Betriebsbereiche arbeitet.

### Performance-Qualification (PQ)

In dieser Phase wird der Nachweis erbracht, dass das System in der Lage ist, die auszuführenden bzw. zu kontrollierenden Prozessvorgänge gemäß schriftlicher und genehmigter Spezifikation auch auszuführen bzw. zu kontrollieren.

Grundsätzlich geht man heute bei der Validierung davon aus, dass die Qualifizierungsdokumente nur dann zufriedenstellend erstellt werden können, wenn die entsprechenden Spezifikationsdokumente vorhanden sind.

In GAMP ist das grundlegende Modell zur Spezifizierung und Qualifizierung vom V-Modell abgeleitet:



Die Ergebnisspezifikation einer Phase ist immer die Vorgabe für die nächste Phase.

Zu jeder Ebene der Spezifikation gibt es auf gleicher Ebene eine Testspezifikation, die überprüft, ob die Ergebnisse dieser Phase die Anforderungen erfüllen.

*Hinweis: Grundsätzlich ist anzumerken, dass das Validierungsmodell in GAMP kein reines Software-Validierungsmodell ist, sondern sich auf eine Gesamtsystem-Validierung bezieht (inklusive Hardware-Systeme). Daher weichen auch manche Bezeichnungen und Methoden gegenüber den bekannten Software-Modellen ab.*

Im konkreten Projekt sollten daher vor Beginn der Validie-

rung die Validierungsgrenzen klar definiert werden und im Validierungsplan das Vorgehensmodell und die Begrifflichkeiten entsprechend definiert werden.

Ein sehr wichtiger Punkt in GAMP ist die Nachvollziehbarkeit. Dies gilt nicht nur für die Protokollierung von Vorgängen und Phasen sowie für das Dokumentenmanagement (Erstellung, Prüfung, Freigabe).

Wesentlich ist auch die Nachverfolgbarkeit der System-Entstehung, Qualifizierung und Aufrechterhaltung der Validierung über den gesamten Lebenszyklus.

Dies bedeutet, dass hier besondere Anforderungen an die Traceability von Anforderungen, Testfällen, ... gestellt werden.

Die Traceability-Matrix wird als Technik in diesem Bereich explizit erwähnt.

*Hinweis: Der praktische Nutzen einer Traceability-Matrix auf Basis von Text- oder Tabellen-Dokumenten — vor allem in großen Projekten — kann bezweifelt werden, da diese Dokumente grundsätzlich meist inkonsistent sind. Wenn man dies tatsächlich als Werkzeug verwenden möchte, so ist ein enormer Aufwand notwendig.*

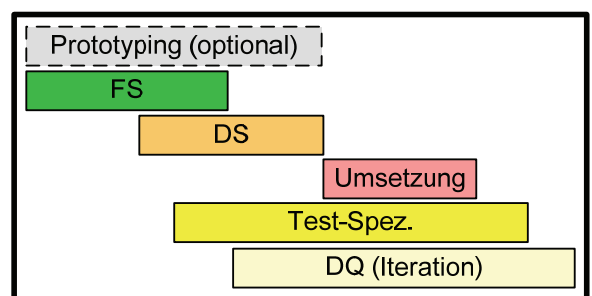
*Eine effiziente Handhabung ist praktisch nur durch den Einsatz einer durchgängigen Toolkette mit einem entsprechenden Repository möglich.*

### Prototyping

Aufgrund der geschilderten Vorgehensweisen und Best-Practices könnte man meinen, dass iterative Vorgehensweisen oder Prototyping nicht oder nur sehr schlecht möglich sind.

Dies ist jedoch nicht der Fall. GAMP enthält sogar explizit Hinweise, wie Prototyping im Rahmen der Validierung als Methode und Vorgehensweise angewendet werden kann.

Beispielsweise kann Prototyping angewendet werden, um Anwender-Anforderungen zu klären oder Risikobereiche zu beurteilen.



Die Absichten und Ziele des Prototypings müssen jedoch schon vorab im Qualitäts- und Projektplan definiert werden.

Wichtig ist dabei, dass die durch Prototyping gewonnenen Informationen und Erkenntnisse kontrolliert in die Spezifikation und Erstellung des endgültigen Produkts aufgenommen werden.

Womit auch klar ist, dass mit dem Prototyping in GAMP nur bedingt das evolutionäre Prototyping (der Prototyp „mutiert“ langsam zum endgültigen Produkt) gemeint ist, son-

(Fortsetzung auf Seite 6)

dem primär das experimentelle Prototyping, in dem eine klare Trennung zwischen Prototyping und endgültigem Produkt gegeben ist.

### Lieferanten

Die Einbindung und das Management von Lieferanten ist ein wesentlicher Faktor für die Qualität des Produkts und auch für die Durchführung der Validierung.

Einerseits kann die Validierung durch Einbindung von Lieferanten, die sich schon an „Best-Practices“ orientieren und strukturierte Prozesse anwenden sowie nachvollziehbare Dokumentationen vorweisen können, wesentlich erleichtert werden.

Andererseits ist es jedoch sehr wichtig und auch eine Forderung von GAMP, dass der Auftraggeber sehr sorgfältig bei der Lieferantenauswahl vorgeht (z.B. durch Lieferantenaudits vor der Auftragvergabe) und auch nach der Auswahl die Prozesskette klare Qualitätsvorgaben einhält (z.B. Übergabe von klaren Spezifikationen an den Lieferanten und Übernahme eines nachweislichen getesteten und mit den Anforderungen übereinstimmenden Produkts).

### Abschluß der Validierung im Projekt

Die Validierung wird typischerweise mit einem detaillierten Validierungsbericht abgeschlossen, in dem die Verantwortlichen die Einhaltung des Validierungsplans und der definierten Vorgehensweisen nochmals überprüfen und bestätigen.

### Erhaltung des validierten Zustands

Im Gegensatz zu vielen anderen Vorgehens- und Prozessmodellen, die sich primär um die Entstehungs-Prozesse eines Systems kümmern, sind bei GAMP auch wesentliche Vorgaben bezüglich der Erhaltung des validierten Zustands enthalten.

Folgende Punkte werden in GAMP in diesem Bereich beispielsweise explizit angeführt:

- ⇒ Betriebliche Pläne und Vorgehensweisen
- ⇒ Training von Anwendern
- ⇒ Testspezifikation des Auftragnehmers
- ⇒ Problemmanagement und Problemlösung
- ⇒ Servicevereinbarungen
- ⇒ Datensicherung und Wiedereinspielung
- ⇒ Konfigurationsmanagement
- ⇒ Betriebliche Änderungslenkung
- ⇒ Sicherheitsmanagement
- ⇒ Leistungsüberwachung
- ⇒ Aufbewahrung von Aufzeichnungen
- ⇒ Geschäftsprozesskontinuitätsplanung
- ⇒ Periodische Prüfungen und Überwachungen
- ⇒ System-Außerbetriebsetzung

Weitere Informationen auf Anfrage.

### Literatur und Links

- ⇒ *GAMP-Leitfaden Validierung automatisierter Systeme in der pharmazeutischen Herstellung*, ISPE, 2001
- ⇒ <http://www.ispe.org>  
Web-Site der International Society for Pharmaceutical Engineering.

### Leistungen

Im Rahmen der **System- und Software-Validierung** unterstützen die Experten von Software Quality Lab als neutrale Berater durch folgende Leistungen:

- ⇒ Unterstützung beim Erstellen von **Validierungsplänen**.
- ⇒ Unterstützung beim Erstellen von ausreichend **detaillierten Anforderungs- und Anwendungs-Spezifikationen (URS, FS, ...)**.
- ⇒ Erstellen von **Qualifizierungs- und Testplänen** sowie den entsprechenden **Testspezifikationen**.
- ⇒ Aufsetzen von passenden **Testorganisationen**.
- ⇒ **Durchführen von Tests**
- ⇒ **Test-Automatisierung**
- ⇒ Begleitendes **Projekt-Controlling** und **Qualitätsmanagement**.
- ⇒ **Coaching von Validierungsbeauftragten** und Projektleitern.
- ⇒ Unterstützung beim Erstellen von im Rahmen der Validierung benötigten **Procedures, SOP's, VA's, ...**
- ⇒ Unterstützung beim Erstellen von **SLA's und OLA's im Rahmen der zu validierenden Betriebsprozesse**.
- ⇒ ...

### Zitate

„Es ist von entscheidender Bedeutung für die Qualität, dass sie integraler Bestandteil der Entwicklung und Inbetriebnahme neuer Computersysteme ist und dass diese unter Einhaltung der Vorschriften betrieben und instand gehalten werden.“  
*GAMP4-Leitfaden*

„Die Erfahrung ist ein teurer Lehrer, aber der Narr wird durch nichts anderes lernen.“  
*Poor Richard's Almanach*

„Es ist nur natürlich, sich eine Methode vorzunehmen und sie auszuprobieren. Wenn es schief geht, gibt es offen zu und versuche etwas anderes. Die Hauptsache ist, dass du überhaupt etwas versuchst.“  
*Franklin D. Roosevelt*

„Einen guten Handwerker erkennt man an seinem Werkzeug.“  
*Altes Sprichwort*