

IaaS – Infrastructure as a Service – Es werden ausschließlich fundamentale Rechenressourcen angeboten, wie z. B. Rechenleistung, Speicherkapazität, Netzwerkverbindung. Hier kann der Nutzer beliebige Software (Betriebssysteme, Anwendungen) einsetzen. Mit IaaS gestalten sich Nutzer frei ihre eigenen virtuellen Computer-Cluster und sind daher für die Auswahl, die Installation, den Betrieb und das Funktionieren ihrer Software selbst verantwortlich.

PaaS – Platform as a Service – Der Nutzer hat die Möglichkeit, in der Cloud-Infrastruktur selbst erstellte oder erworbene Anwendungen zu verwenden, die auf Programmiersprachen, Bibliotheken, Diensten und Tools basieren, die der Dienstleister bereitstellt. Der Nutzer ist nicht für die zugrundeliegende Infrastruktur einschließlich Netzwerk, Servern, Betriebssystemen und Speicher verantwortlich. Er hat aber die Kontrolle über die bereitgestellten Anwendungen und die Konfigurationseinstellungen der Anwendungen. Mit PaaS entwickeln Nutzer ihre eigenen Softwareanwendungen innerhalb einer Softwareumgebung, die vom Dienstanbieter (Service Provider) bereitgestellt und unterhalten wird, die sie dort auch ausführen lassen. Force.com ist ein PaaS-Angebot von Salesforce.com.

SaaS – Software as a Service – Der Dienstleister stellt die Nutzung einer oder mehrerer Anwendungen zur Verfügung. Dies ist ein stark standardisierter Service für ein bestimmtes Einsatzszenario. Der Nutzer benötigt keine lokale Software, um auf den Service zuzugreifen. Rechnerwolken bieten Nutzungszugang von Softwaresammlungen und Anwendungsprogrammen. SaaS-Dienstleister offerieren eine spezielle Menge an Software, die auf ihrer Infrastruktur läuft. SaaS wird auch als Software on demand (Software bei Bedarf) bezeichnet. Bekannte Beispiele: Microsoft Office 365, Google Apps, Salesforce.com, Dropbox, Apple iCloud, Google Drive u. v. m.

4.1 Vor- und Nachteile von Cloud-Computing

Tab. 1. Vor- und Nachteile von Cloud-Computing.

Vorteile des Cloud-Computing	Nachteile des Cloud-Computing
Der Nutzer von Cloud-Computing kann erhebliche Investitionskosten und die damit verbundene langfristige Kapitalbindung vermeiden.	Die Nutzung von Cloud-Diensten führt zu einer Abhängigkeit vom Dienstleister, dem Service des Dienstleisters und der Qualität der internen Prozesse des Dienstleisters.
Die Skalierbarkeit der Cloud-Dienste – abhängig vom Nutzungsgrad – macht es möglich, kurzfristig Nutzungsspitzen auszugleichen. Andernfalls müsste der Nutzer in Phasen hoher Auslastung entweder Performanceverluste akzeptieren oder entsprechende Kapazitäten vorhalten.	Die Daten und Dienste in der Cloud sind nur verfügbar, wenn die Internetanbindung mit ausreichender Bandbreite verfügbar ist.
Die Betriebskosten für Cloud-Dienste sind für den Nutzer i. d. R. günstiger als die Betriebskosten für intern bereitgestellte Dienste.	Durch die Auslagerung von IT-Diensten in die Cloud verliert der Nutzer interne IT-Kompetenz.
Der Cloud-Dienstleister kann seine Ressourcen einfacher auf dem aktuellen Stand der Technik halten.	Bei einem kurzfristigen Ausfall des Cloud-Anbieters (z. B. durch Insolvenz) ergeben sich Risiken in Bezug auf die Verfügbarkeit der Daten.

Forts. Tab. 1 nächste Seite

Vorteile des Cloud-Computing	Nachteile des Cloud-Computing
Der Cloud-Dienstleister bietet den Nutzern i. d. R. eine höhere Ausfallsicherheit, als dies bei vergleichbaren Kosten für intern angebotene Dienste möglich ist.	Risiken in Bezug auf den Datenschutz müssen betrachtet und dabei auch gesetzliche Regelungen berücksichtigt werden.

Die Vorteile, v. a. Reduzierung von Kosten und höhere Flexibilität, machen die Auslagerung von IT-Diensten an Cloud-Anbieter für alle Unternehmen sehr attraktiv. Es stellt sich aber die Frage, was das für Unternehmen im regulierten Umfeld (pharmazeutische Industrie, Medizintechnik) bedeutet. Wie kann der Nutzer von Cloud-Diensten die in diesen Bereichen von Behörden geforderte Qualität sicherstellen und den dokumentierten Nachweis dafür erbringen, dass die Qualität sichergestellt ist?

5. Anforderungen an die Validierung

Im Folgenden werden grundsätzliche Anforderungen an die Validierung von Cloud-basierten Systemen im regulierten Umfeld aufgelistet:

1. Für die Validierung eines Cloud-basierten Systems gelten die gleichen Anforderungen, die auch für die Validierung eines konventionellen computergestützten Systems gelten:

- Der Nutzer/Betreiber muss den dokumentierten Nachweis erbringen, dass er das System unter Kontrolle hat.
- Der Nutzer/Betreiber muss den dokumentierten Nachweis erbringen, dass das System 'fit for intended use' ist.

2. Der Validierungsansatz ist risikobasiert. Das heißt, der Validierungsaufwand ergibt sich aus der Betrachtung der mit der Nutzung des Systems verbundenen Risiken. Dabei müssen die besonderen Risiken berücksichtigt werden, die sich daraus ergeben, dass das System ein Cloud-basiertes System ist.

3. Leistungen können an den Dienstleister ausgelagert werden. Dabei muss sichergestellt sein, dass der Dienstleister in der Lage ist, diese Leistungen in der geforderten Qualität zu erbringen. Auch muss ggf. in einer Inspektion der dokumentierte Nachweis geliefert werden, dass diese Leistungen erbracht wurden.

Bei der Nutzung von Cloud-basierten Diensten ist es i. d. R. nicht zu vermeiden, Teile der Validierung an den Dienstleister auszulagern (Ausnahme: Nutzung einer privaten Cloud, die vom Nutzer selbst betrieben wird). Dadurch ergeben sich zusätzliche Anforderungen an den Dienstleister, die im Auswahlprozess berücksichtigt werden müssen:

- Im Auswahlprozess muss der Nutzer prüfen, ob der Anbieter in der Lage ist, diese Anforderungen in der benötigten dokumentierten Qualität zu erfüllen, z. B. durch ein vorhandenes Qualitätsmanagementsystem.
- Die Verantwortlichkeiten bzgl. der Implementierung und auch für den Betrieb müssen zwischen Nutzer und Anbieter eindeutig definiert werden (z. B. Validierungsplan, Service Level Agreements).

6. Rollen und Verantwortlichkeiten

Die Rollen und Verantwortlichkeiten, die sich bei der Validierung und dem Betrieb von computergestützten Systemen ergeben, müssen bei der Validierung Cloud-basierter Systeme angepasst werden. Durch die notwendige Einbeziehung des Dienstleisters muss abhängig vom gewählten Servicemodell definiert werden, welche Rolle vom Nutzer im regulierten Umfeld wahrgenommen wird und welche Rolle an den Dienstleister delegiert wird.

Die zu betrachtenden Rollen, die sich aus GAMP® 5 ergeben, sind: Prozesseigner, Systemeigner, Fachexperten, Qualitätssicherung und Endanwender. Dazu kommt die Rolle Servicemanager.

Der Prozesseigner ist verantwortlich dafür, dass das computergestützte System und sein Betrieb über den gesamten Lebenszyklus konform zu den Regularien sind. Diese Rolle ist unabhängig vom verwendeten Servicemodell immer beim Nutzer.

Der Systemeigner (nach GAMP® 5) ist dafür verantwortlich, dass das System gemäß gültiger SOPs unterstützt und instandgehalten wird. Für das Servicemodell SaaS liegt diese Rolle beim Cloud-Dienstleister. Für die anderen Servicemodelle (PaaS, IaaS) liegt diese Rolle beim Nutzer.

Bei der Verwendung von Cloud-Dienstleistungen ist es zu empfehlen, dass zusätzlich zu den in GAMP® 5 definierten Rollen sowohl auf Seiten des Nutzers als auch auf Seiten des Dienstleisters ein Servicemanager installiert wird, die beide dann für die Definition und Überwachung von Service Level Agreements verantwortlich sind.

Fachexperten, die z. B. die fachliche Verantwortung für IT-Infrastruktur, IT-Applikation und Systemadministration übernehmen, müssen auf beiden Seiten festgelegt sein.

Auf Seiten des Dienstleisters und des Nutzers muss eine Qualitätssicherung etabliert sein, die beide für die Implementierung und Einhaltung der Qualitätsvorgaben in ihrem Bereich verantwortlich sind.

Die Rolle des Endanwenders, der für die Nutzung des Systems gemäß den Vorgaben (SOPs) verantwortlich ist, liegt naturgemäß beim Nutzer.

In Vereinbarungen zwischen Nutzer und Dienstleister (Service Level Agreements) muss eindeutig festgelegt werden, wer für welche Aktivität welche Rolle übernimmt, d. h.:

- Wer ist der Zuständige, der die Aktivität umsetzen muss?
- Wer hat die formale Verantwortlichkeit dafür, dass die Aktivität korrekt umgesetzt wird?
- Wer sollte unterstützend mitwirken?
- Wer soll über die Aktivität informiert werden?

Schon bei der Planung einer Cloud-basierten Lösung soll man sich Gedanken dazu machen, was bei der Stilllegung des Systems zu tun ist. Eine Stilllegung des Systems kann aus verschiedenen Gründen notwendig werden. Beispiele sind:

- der Nutzer will den Anbieter wechseln
- der Anbieter stellt die Dienstleistung nicht mehr länger zur Verfügung (Insolvenz, Änderung des Geschäftsmodells)

Im Rahmen der Bewertung des Cloud-Anbieters soll untersucht werden, wie hoch das Risiko ist, dass dieser kurzfristig nicht mehr verfügbar ist. Abhängig

1. Einleitung

Aktuelle Umfragen und Studien, wie z. B. die Studie ‚Status Quo Agile‘ der Hochschule Koblenz – University of Applied Sciences, zeigen, dass das Thema rund um agile Methoden nach wie vor aktuell und von großem Interesse ist. Bereits zum zweiten Mal wird bei dieser Studie untersucht, wie der Status zu Verbreitung und Nutzen agiler Methoden bei verschiedenen Unternehmen aussieht. Die erste Umfrage zu diesem Thema fand bereits 2012 statt und führte u. a. zu folgenden Erkenntnissen:

- Scrum wird von durchgängig agilen Nutzern zu 100 % als gut oder sehr gut bewertet
- Anwender agiler Methoden bewerten die von ihnen genutzten Praktiken in allen Kriterien besser als die Anwender klassischer Projektmanagementmethoden
- die Nutzung agiler Methoden hat seit 2008 einen sehr starken Aufschwung genommen
- nur 5 % der Nutzer agiler Methoden sehen keine Verbesserungen bei Ergebnissen und Effizienz

Alle Ergebnisse können auf der entsprechenden Webseite [1] nachgelesen werden.

Dies macht deutlich, dass agile Methoden nicht nur weiter im Trend liegen, sondern auch sehr positiv von den Unternehmen bewertet werden.

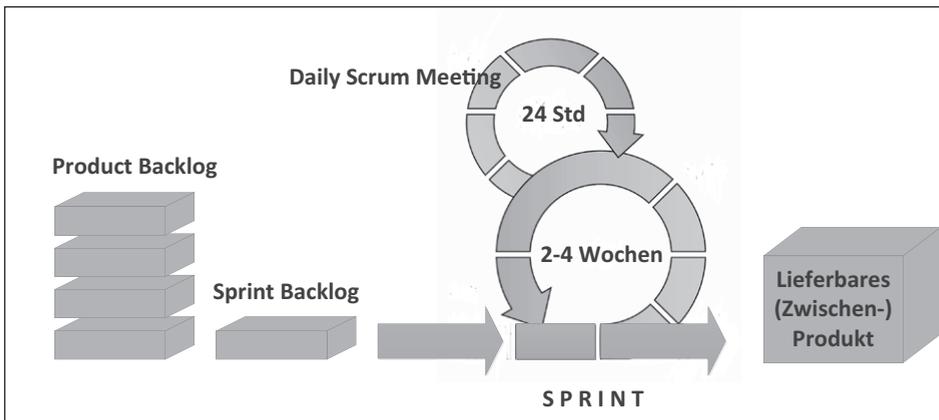


Abb. 1. Aufbau und Einteilung der Entwicklung nach der Scrum-Methodik.

Eine der am häufigsten genutzten agilen Methoden ist Scrum. Scrum hatte seine Anfänge 1990, geprägt durch Jeff Sutherland. Die Prinzipien dieser Vorgehensweise sind im „Manifesto for Agile Software Development“ [2] festgeschrieben. Typische Merkmale von Scrum sind u. a. der Aufbau und die Einteilung der Entwicklung in kurze Zyklen, sog. „Sprints“, wie in Abb. 1 ersichtlich. Ein Sprint dauert üblicherweise zwischen 2–4 Wochen, an dessen Ende ein auslieferbares (Zwischen-)Produkt vorliegt. Diese Vorgehensweise ermöglicht die Ablieferung eines lauffähigen Softwarestands in kurzen, regelmäßigen Abständen. Die Entwicklerteams organisieren sich weitgehend selbst, ein regelmäßiger Austausch mit dem Kunden ist notwendig.

Vorteile dieser agilen Vorgehensweise sind u. a.:

- eine erhöhte Entwicklungsgeschwindigkeit

- kurze Entwicklungszyklen
- verbesserte Mitarbeiterzufriedenheit
- kontinuierliche Nutzengenerierung
- frühe Einbeziehung des Kunden

2. Agile und klassische Methode – eine Gegenüberstellung

Häufig wird behauptet, dass die Anforderungen im Bereich der Validierung nicht mit der Methodik von Scrum vereinbar sind.

Gemäß der klassischen Methode steht die Validierung i. d. R. ganz im Zeichen des V-Modells. Das darin streng geregelte Vorgehen gibt vor, dass zuerst die Anforderungen definiert sein müssen, welche dann phasenweise abgearbeitet werden. Der Ansatz von Scrum geht im Gegensatz zu diesem streng geregelten Vorgehen von einem inkrementellen und iterativen Ansatz aus. Auf den ersten Blick sind diese beiden Vorgehensweisen also nicht wirklich miteinander vereinbar. Betrachtet und vergleicht man jedoch einmal die klassische Methode mit der agilen Vorgehensweise, so stellt man fest, dass es durchaus die Möglichkeit gibt, Scrum im regulierten Umfeld einzusetzen.

	Klassisches Vorgehensmodell	Scrum
Projektmanagement		
Projektplanung	Projektplan / Meilensteine	Product Backlog
Aufgabenplanung	Arbeitspakete	Sprint Backlog
Statusverfolgung	durch PL / Meilensteine	Täglich, am White Board
Auslieferung	Je Release oder Projektende	Am Sprint-Ende
Änderungsmanagement	Gemäß CR	Per Backlog Update
Qualitätsmanagement		
Prozessverbesserung	Auditierungen, Prozessverbesserungsprogramme	Sprint Retrospektiven
Verifikation / Test	in Testphasen gegenüber Spezifikation	kontinuierlich im Sprint
Validierung / Abnahme	Abnahmetest am Projektende	User Demo am Sprint Ende
Entwicklung		
Techniken	abhängig vom Modell	Keine Vorgaben, oftmals Test First Programming
Personalführung		
Werte / Prinzipien	Projektmanagement, kontinuierliche Verbesserung (ISO 9000)	Agiles Manifest, Scrum Werte
Organisation	Team-, Projektleitung	Scrum Master, Scrum Team, Product Owner

Abb. 2. Gegenüberstellung klassisches Vorgehensmodell und Scrum.

Gemäß GAMP® 5 gibt es keine Einschränkung mehr, welchem Lebenszyklusmodell gefolgt werden soll. Das bedeutet, dass der Einsatz des V-Modells nicht zwingend notwendig ist. Einzig die Dokumentation nach jedem größeren Zyklus oder Meilenstein wird gefordert. Stellt man dies nun der Vorgehensweise nach Scrum gegenüber, wird ersichtlich, dass dies sehr gut miteinander vereinbar ist. So wird z. B. im Gegensatz zu der oftmals falschen Behauptung, dass Scrum komplett auf Dokumentation verzichtet, sehr wohl auch bei dieser Methode eine Dokumentation erwartet. Zwar hat das Produkt einen höheren Stellenwert als

die Dokumentation: „*Working software over comprehensive documentation*“ [2]. Jedoch bedeutet dies nicht, dass auf diese verzichtet werden muss.

Ein Vergleich beider Vorgehensmodelle mit allen wichtigen Projektphasen und Anforderungen zeigt, dass zu jedem Punkt aus Sicht der klassischen Validierung auch ein Punkt aus Scrum-Sicht gegenübergestellt werden kann (Abb. 2).

Dies verdeutlicht, dass Scrum sehr wohl auch im Bereich der Computervalidierung eingesetzt werden kann, v. a. wenn eine sog. hybride Vorgehensweise genutzt wird, welche beide Modelle miteinander kombiniert.

3. Scrum und Dokumentation

Betrachtet man die Dokumentation beider Vorgehensweisen miteinander, wird man schnell feststellen, dass sich die verschiedenen Dokumente bei beiden Methoden unterscheiden.

Maßgebende Dokumente nach V-Modell sind v. a. die einzelnen Spezifikationen, Risikoanalyse, Testpläne sowie Testprotokolle, wie in Abb. 3 ersichtlich. Bei Scrum sind Product und Sprint Backlog Hauptartefakte der Dokumentation.

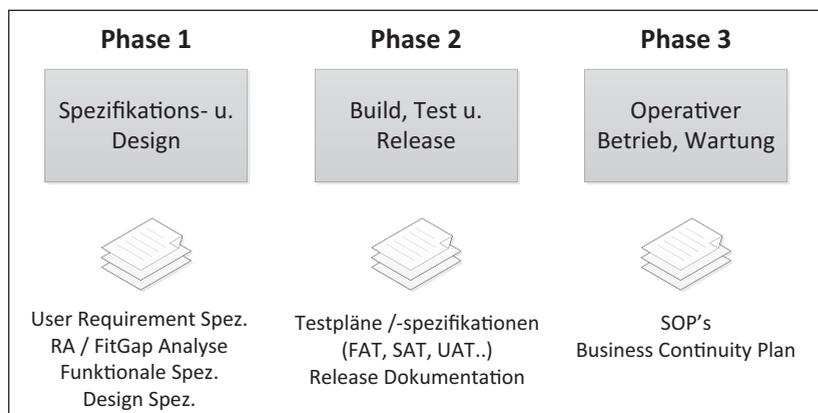


Abb. 3. Beispiele klassischer Dokumente bei Vorgehensweise nach V-Modell.

Das Product Backlog umfasst und definiert alle Anforderungen eines Projekts. Dem Product Backlog untergeordnet ist das Sprint Backlog, welches alle Aufgaben innerhalb eines bestimmten Sprints beschreibt. Diese Aufgaben werden als User Stories definiert und enthalten Informationen über Anforderungen, Rollen, Nutzen und Funktionalität. Ebenso geht aus dem Sprint Backlog hervor, in welchem Arbeitsstatus sich die einzelnen User Stories befinden. Wie in Abb. 4 aufgezeigt, sind die typischen Status „To Do“, „In Arbeit“ und „Done“.