

Agiler Softwareentwicklungsprozess bei der SAP SE

Am Beispiel der ERP-Suite SAP S/4HANA Cloud

Felix Breitsch, Peter Preuss

Bei der agilen Softwareentwicklung erstellen selbstorganisierte Teams in einer iterativen Vorgehensweise Softwareprodukte. Ein wesentlicher Vorteil dieses Vorgehens besteht darin, dass regelmäßig lauffähige Produktinkremente erstellt werden und so die Möglichkeit besteht, schneller auf neue oder geänderte Anforderungen reagieren zu können. Ziel dieses Beitrages ist es, den Softwareentwicklungsprozess bei der SAP SE anhand der Cloud-basierten ERP-Suite SAP S/4HANA Cloud vorzustellen und zu untersuchen, wie so ein komplexes Softwareprodukt von mehreren agilen Teams kontinuierlich weiterentwickelt werden kann. Zuerst wird der klassische S/4HANA Cloud-Release-Zyklus überblicksartig beschrieben. Danach wird die darin enthaltene Entwicklungsphase aus der Perspektive eines agilen Entwicklungsteams beleuchtet. Der Beitrag schließt mit einer Diskussion der seit 2022 verwendeten Continuous-Delivery-Strategie.

Die SAP SE gehört zu den weltweit führenden Anbietern von Enterprise Application Software und sie ist nach Marktkapitalisierung eines der größten Technologieunternehmen Europas. Das Unternehmen hat nach eigenen Angaben eines der umfangreichsten Portfolios von Unternehmenssoftware. Die über 100 Anwendungen unterstützen ein breites Spektrum an betrieblichen Prozessen, wie zum Beispiel „recruit to retire“, „lead to cash“, „source to pay“ oder „design to operate“. Angeboten werden die unterschiedlichen Softwarelösungen als Lizenz-, Cloud- oder Hybridmodell.

Bekannt wurde die Marke SAP vor allem durch die ERP-Suites SAP R/2 und SAP R/3. Das aktuelle ERP-Kernprodukt der SAP ist SAP S/4HANA. Gegenüber seinen Vorgängern zeichnet sich dieses ERP-System u. a. durch eine höhere Geschwindigkeit und bessere Integrationsmöglichkeiten aus. Das wird erreicht, indem die Anwendungslogik vereinfacht und die harmonisierten Datenmodelle

in der SAP-eigenen In-Memory-Datenbank SAP HANA abgebildet werden. Weitere Charakteristika von SAP S/4HANA sind die webbasierte Oberfläche auf HTML5-Basis (SAP Fiori) und die dazugehörigen Webservices und Protokolle. Neben SAP S/4HANA ist die SAP Business Technology Platform (SAP BTP) ein weiteres wichtiges Produkt der SAP. Hierbei handelt es sich um eine Plattform-as-a-Service-Lösung, die unter anderem für die Anwendungsentwicklung, das Datenmanagement, (kundeneigene) Integrationsszenarien und die Bereitstellung von Analytics-Funktionen genutzt werden kann.

Agile Softwareentwicklung bei der SAP

Die erste Version von SAP S/4HANA wurde bereits 2010 angekündigt und erstmals 2015 auf den Markt gebracht. Ungefähr zeitgleich hat die SAP in größerem Umfang agile Entwicklungsmethoden eingeführt. Obwohl die agile Softwareentwicklung bei der SAP schon so viele Jahre im Einsatz ist, gibt es immer wieder neue Herausforderungen, die es zu bewältigen gilt. Eine Schwierigkeit besteht sicherlich darin, dass die Softwareprodukte der SAP historisch gewachsen sind und teilweise immer noch auf monolithischen Architekturen basieren, die sukzessive auf neuere Technologien und feingliedrige Komponenten umgestellt werden müssen. Vor diesem Hintergrund ist es nicht überraschend,

Lesen Sie

- wie sich der agile Softwareentwicklungsprozess bei der SAP SE gestaltet
- wie die SAP SE die Continuous-Delivery-Strategie für SAP S/4HANA Cloud umsetzt

Wesentliche Akteure	Ausgewählte Aktivitäten
Phase 1: Planung	
Product Owner Development Teams Zentrale Teams (z. B. Front-End, Architektur, Accessibility etc.)	<ul style="list-style-type: none"> • Definition des Entwicklungsumfangs • Erstellung erster Rapid Prototyps/ Mockups • Priorisierung und Ausgestaltung des Backlogs • Enablement zu zentralen Entwicklungsvorgaben
Phase 2: Entwicklung	
Development Teams Zentrale Teams (z. B. Data Landscape Management, Konfigurationsteams, Quality Teams, Security Teams)	<ul style="list-style-type: none"> • Schwerpunktmäßige Entwicklungsarbeit (testgetrieben) • Begleitende Test- und Qualitätssicherungsaktivitäten („Clean-Desk“-Prinzipien, Testpläne, Design Gates) • Initiierung und Überwachung der Auslieferung (Transportschienen, Dienste und Systeme)
Phase 3: Korrektur	
Quality Engineers Development Teams Content/Build Developer	<ul style="list-style-type: none"> • Komplettierung des Scopes und der Strukturen (Apps, Dokumentation etc.) • Überprüfung und Komplettierung der Testabdeckung • Finalisierung der Qualitätsarbeiten
Phase 4: Auslieferung	
Lines of Businesses Content/Build Developer Scenario Owner	<ul style="list-style-type: none"> • Management-Reporting zum Release • Komplettierung Dokumentation/ Kundenkommunikation • Auslieferungsbezogene Aktivitäten (Packaging)

Tabelle 1: Typische Phasen des SAP S/4HANA Cloud-Release-Zyklus (vereinfacht).

dass die SAP ihre kommende ERP-Generation als „composable ERP-System“ beschreibt, also als ein flexibles, modulares System, bei dem sogar Kernfunktionen beliebig ergänzt, erweitert und in unterschiedlichen Umgebungen betrieben werden können. Auf dem Weg dahin gilt es nicht nur, die beständig entstehenden technischen Schulden der Vergangenheit zu beseitigen. Es müssen auch kontinuierlich innovative Software-Features und neue Lösungen ausgeliefert werden. Nicht allein um den steigenden und sich verändernden Anforderungen der Kunden Rechnung zu tragen, sondern um neue Geschäftsprozesse und -modelle zu ermöglichen. Erschwerend kommt hinzu, dass eine sinnvolle Balance zwischen den Standardisierungswünschen der SAP und den Individualisierungsbestrebungen der Kunden gefunden werden muss. Da es sich bei einem ERP-System um eine unternehmenskritische Anwendung handelt, haben die Anwender auch

hohe Ansprüche an die Verfügbarkeit und die Zuverlässigkeit der Software.

der Chief Product Owner und die Product Owner der beteiligten Entwicklungsteams legen in dieser Phase den Release-Umfang fest (Entwicklungsscope) und priorisieren die Items des Release-Backlogs.

Aus der Perspektive eines Entwicklungsteams liegt der Schwerpunkt in der Planungsphase auf der Vorstellung der Prioritäten des Programm- und Produktmanagements. Die ungefähr sechswöchige Planungsphase startet direkt im Anschluss an den Entwicklungsschluss des vorherigen Releases. Obwohl die Entwickler zu diesem Zeitpunkt noch mit Nacharbeiten des vorherigen Releases beschäftigt sind, arbeiten sie dennoch aktiv an der Planung und den Vorarbeiten des neuen Releases mit. Hiermit wird dem agilen Gedanken Rechnung getragen, dass die besten Produkte nur dann entstehen, wenn selbstorganisierte Teams daran arbeiten. Gegen Ende der Planungsphase steht die zentrale

SAP S/4HANA Cloud-Release-Zyklus

Neue Software-Features stellt die SAP in regelmäßigen Release-Zyklen zur Verfügung. Bei SAP S/4HANA Cloud dauerte die Bereitstellung eines solchen Updates (bisher) üblicherweise drei bis vier Monate. Neben den eigentlichen Entwicklungsarbeiten enthält ein solcher Zyklus auch alle notwendigen Vor- und Nacharbeiten, wodurch sich die zur Verfügung stehende Nettoentwicklungszeit entsprechend reduziert. Der klassische Release-Zyklus kann vereinfacht in den in Tabelle 1 beschriebenen Phasen Planung, Entwicklung und Korrektur- sowie Auslieferungsphase unterteilt werden, wobei diese nicht wasserfallartig abgearbeitet werden, sondern in Form von „Mikrophasen“ ineinander verschränkt sind. Das bedeutet, dass beispielsweise die Planung kontinuierlich gemacht wird und auch parallel zu den Entwicklungsarbeiten stattfindet.

In der Planungsphase liegt einer der Schwerpunkte auf dem Programm- und Produktmanagement. Der Chief Product

Entscheidung an, ob mit der Entwicklung formal begonnen werden kann. Das hängt insbesondere von der Qualität des vorherigen Releases ab. Falls hier noch Nacharbeiten erforderlich sind oder die Produktqualität nicht den internen Anforderungen genügt, wird mit der Entwicklung des neuen Releases so lange gewartet, bis die Nacharbeiten an dem Vorgänger-Release abgeschlossen sind.

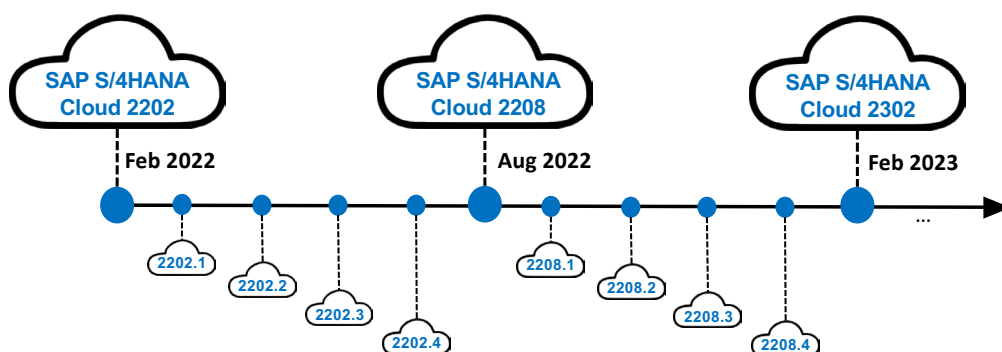


Bild 1: Major Releases und Continuous-Delivery-Updates zwischen den Releases.

Die Entwicklungsphase ist in mehrere zweiwöchige Takte unterteilt. Obwohl die agilen Entwicklungsteams eigenständig entscheiden können, wie lange ihre Sprints sind und wann sie welche Features aus dem Entwicklungsscope umsetzen, sorgt der Takt für eine Synchronisierung der beteiligten Entwicklungsteams vor dem Hintergrund agiler Prinzipien und mit dem Ziel, die Produktinkremente und letztlich das ganze Produkt mit dem passenden Umfang, in der gewünschten Zeit und mit der entsprechenden Qualität liefern zu können. Dazu wird im Rahmen der Takte geprüft, welcher neue Umfang geschaffen wurde (viable new scope), dass eine Regressionsfreiheit besteht und dass die beteiligten Teams die internen Clean-Desk-Prinzipien umsetzen. Die Entwicklungsphase endet typischerweise mit umfangreichen Funktions-, Integrations- und Akzeptanztests, bei denen jedes Team von einem eigenen Quality Engineer begleitet wird.

Direkt im Anschluss an die Entwicklungsphase startet die drei Wochen dauernde Korrektur- und Release-Phase. In dieser Phase führen die Entwicklungsteams weitere Tests durch und finalisieren die Konfiguration, die Dokumentation und die ausgelieferten Vorlagen. Der Release-Zyklus mündet schließlich in die abschließende Auslieferungphase. Die Entwicklungsteams unterstützen hier nur noch punktuell, da es spezialisierte Teams gibt, die für diese erfolgskritische Phase verantwortlich sind.

Entwicklungsphase aus Sicht eines agilen Entwicklungsteams

Wie im vorherigen Abschnitt beschrieben, werden in der Entwicklungsphase feedbackgestützte Inkremente erstellt, mittelbar in Sprints mit teamindividueller Dauer, aber synchronisiert durch zweiwöchige Abschnitte (Takte), die eine teamübergreifende Qualitätssicherung gewährleisten. Jedes Entwicklungsteam ist für den Erfolg seiner

Sprints verantwortlich. Sie werden bei ihrer Entwicklungsarbeit organisatorisch von einem Scrum Master unterstützt. Zu Beginn eines jeden Sprints formuliert das Team, unter Berücksichtigung der vom Product Owner festgelegten Prioritäten, das Sprint Goal, das dann im Nachgang vom gesamten Entwicklungsteam im Sprint Backlog, welcher sich aus dem Product Backlog speist, weiter verfeinert wird. Das Sprint Backlog enthält aber nicht nur neu zu entwickelnde Features, sondern auch Backlog-Items, die den Umbau der bestehenden Applikationslogik für den Cloud-Betrieb zum Ziel haben oder neuere Innovationen bzw. zentrale Anforderungen betreffen (z. B. die SAP Fiori-Oberfläche). Der Product Owner verantwortet gegenüber dem Chief Product Owner die Umsetzung des Product Backlog Items. Der Quality Engineer eines Entwicklungsteams verwendet das Sprint Backlog, um erste Testschwerpunkte zu identifizieren und Testpläne zu entwickeln, die an den der zweiwöchigen Taktung ausgerichtet sind. Die technischen Aktivitäten eines Sprints werden mit funktionalen Tests, die von den Entwicklungsteams durchgeführt werden, abgeschlossen. Der Sprint endet mit einem gemeinsamen Review Meeting. In dieser Veranstaltung wird der Entwicklungsfortschritt begutachtet und anhand selbst definierter Qualitätsregeln entschieden, was als Inkrement wirklich fertig ist („Ready“ im Sinne der „Definition of Done“).

Wie der Product Owner ist auch der Quality Engineer eine wichtige Schnittstelle des Entwicklungsteams nach „außen“, weil er in enger Abstimmung mit dem Delivery- und Programmmanagement die qualitativen Anforderungen festlegt. Er koordiniert und überwacht die Testaktivitäten während eines Sprints. Diese Aktivitäten beziehen sich sowohl auf die Benutzerschnittstelle als auch auf die Backend-Funktionen des SAP S/4HANA-Systems. Hierbei gilt, dass die meisten Testskripts automatisiert in dezidierten Testsystemen von Testrobotern durchgeführt werden.

Von der klassischen Release-Strategie hin zu Continuous Delivery

Bis vor Kurzem wurden neue Funktionen von SAP S/4HANA Cloud in einem vierteljährlichen Release-Rhythmus ausgeliefert. Das hat sich 2022 geändert. Seit diesem Jahr wird eine Continuous-Delivery-Strategie verfolgt, bei der es nur noch zwei halbjährliche Releases gibt, die dann um acht kleinere Updates pro Jahr ergänzt werden (Bild 1). Hiermit verfolgt die SAP zwei Ziele: Zum einen soll der Aufwand für die Release-Upgrades reduziert und zum anderen sollen neue Software-Features in kürzeren Zeitabständen angeboten werden. Wichtig hierbei ist, dass die Updates weniger Aufwand und keine Unterbrechungen bzw. Störungen verursachen.

Die Release-Planung für 2022 sah beispielsweise vor, dass es im Februar und im August ein Haupt-Release gibt (SAP S/4HANA Cloud Release 2202 und 2208). Zusätzlich sollen von Februar bis August 2022 und von August 2022 bis Februar 2023 jeweils vier Continuous-Delivery-Updates bereitgestellt werden (Updates 2202.1 bis 2202.4 und 2208.1 bis 2208.4). Aus Sicht der agilen Entwicklungsteams ergibt sich dabei die Notwendigkeit, einen kontinuierlichen Planungs-, Entwicklungs- und Auslieferungsprozess zu etablieren, was einerseits die Anforderungen an die Teams erhöht, aber andererseits auch einen Anreiz bietet, mehr Eigenverantwortung und Gestaltungsspielraum zu gewinnen. Aus Kundensicht gilt, dass jede Funktion, die in ei-

nem solchen Update enthalten ist, automatisch mit dem darauffolgenden Haupt-Release ausgeliefert wird. Es ist daher nicht zwingend notwendig, jedes Feature bei der Bereitstellung nutzen zu müssen. Die Continuous-Delivery-Strategie bietet den Kunden vielmehr die Möglichkeit, selbst zu entscheiden, ob sie die ausgelieferten Features nutzen („freischalten“) oder vorerst beim aktuellen Funktionsumfang bleiben möchten.

Fazit

Mit SAP S/4HANA Cloud bietet die SAP ein ERP-System an, das als Software-as-a-Service-Lösung über das Internet genutzt werden kann. 2022 wurde der Entwicklungsprozess für dieses Cloud-Produkt umgestellt. Anstatt vier Releases pro Jahr gibt es nun nur noch zwei Major Releases, ergänzt um weitere Continuous-Delivery-Updates zwischen jeweils zwei aufeinanderfolgenden Releases. Dieser Continuous-Delivery-Prozess ermöglicht es der SAP nicht nur ihre Kunden stärker und zeitnah in den Entwicklungsprozess einzubinden und so relevante neue Anforderungen bereitstellen zu können. Die neue Lieferstrategie motiviert auch die Entwicklungsteams, weil sie mitentscheiden, was, wann und wie ausgeliefert wird. Hierbei besteht allerdings die Herausforderung, dass in sehr kurzen Zeitabständen neue Funktionen zur Verfügung angeboten werden, die auf Kundenseite eine intensive Auseinandersetzung mit den gelieferten Ergänzungen und Änderungen erfordern. Hinzu kommt, dass, auch wenn der Funktionsumfang eines solchen Updates viel kleiner als bei den halbjährlichen Major Releases ist, trotzdem jedes Mal ein (kleiner) vollständiger Planungs-, Entwicklungs- und Auslieferungsprozess durchlaufen werden muss.

Literatur

- [1] Brüggenkamp, J. / Preuss, P. / Renk, T. (2021): Der Scrum-Reiseführer – Herausforderungen in der Praxis meistern, UTB-Verlag.
- [2] Geldmacher, C. (2022): How to adopt a new functionality delivered via SAP S/4HANA Cloud Updates, in HTML-Dokument: <https://blogs.sap.com/2022/02/01/how-to-adopt-a-new-functionality-delivered-via-sap-s-4hana-cloud-updates/>, abgerufen am 17.07.2022.
- [3] SAP (2021): SAP Integrated Report 2021, in HTML-Dokument: <https://www.sap.com/investors/en.html?pdf-as-set=903be721-1b7e-0010-bca6-c68f7e60039b&page=338>, abgerufen am 17.07.2022.
- [4] SAP (2022): continuous delivery for SAP S/4HANA Cloud, in HTML-Dokument: <https://community.sap.com/topics/s4hana-cloud/continuous-delivery>, abgerufen am 17.07.2022.
- [5] Saueressig, T. / Gilg, J. / Betz, O. / Homann, M. (2021): SAP S/4HANA Cloud: An Introduction, Rheinwerk Verlag.



Felix Breitsch arbeitet als Softwareentwickler im Bereich SAP S/4HANA Finance bei der SAP SE. Er studiert berufsbegleitend Wirtschaftsinformatik an der FOM Hochschule für Oekonomie & Management in Mannheim.



Prof. Dr. Peter Preuss lehrt Wirtschaftsinformatik an der FOM Hochschule für Oekonomie & Management in Stuttgart. Neben seiner Lehrtätigkeit ist Peter Preuss geschäftsführender Gesellschafter der Unternehmensberatung People Consolidated GmbH.

Prof. Dr. Peter Preuss
FOM Hochschulstudienzentrum Stuttgart
Rotebühlstraße 121
70178 Stuttgart
www.fom.de

Stichwörter:

Agile Softwareentwicklung, SAP S/4HANA Cloud, Continuous Delivery, ERP-System