

Testkonzept – Transition von Klassisch zu Agil

Ralf Somplatzki

GEBIT Solutions GmbH

Oststraße 10, 40211 Düsseldorf, Germany

1 Motivation

Können langfristig angelegte Projekte, die im klassischen Sinne (wasserfall- oder iterativ) aufgesetzt sind, in ein agiles Modell transformiert werden? Was bedeutet das für das etablierte Testkonzept und somit die entwicklungsbegleitenden Testaufgaben?

Dieser Artikel adressiert die Herausforderungen an eine möglichst reibungs- und verlustfreie Transition. Dabei sollen etablierten Artefakten und Best Practices ebenso Rechnung getragen werden wie den Gewinnen durch ein agiles Projekt-Setup.

An Hand eines konkreten Fallbeispiels werden die Herausforderungen benannt und ein konkreter Lösungsansatz aufgezeigt.

2 Ausgangslage

Das Softwareentwicklungsprojekt wird seit mehr als sechs Jahren nach einem iterativen Prozessmodell gelebt. Das Team ist über die Jahre auf ca. 40 Mitglieder gewachsen. Die Software wird nach Lieferung kundenseitig mit den Produkten zweier weiterer Lieferanten zusammengeführt und systemgetestet. Jede dritte Iteration stellt plangemäß einen Release-Kandidaten dar. Auf der Zeitachse bedeutet das ca. 18-21 Wochen.

Die resultierenden, hohen Laufzeiten für Korrekturen wie auch die Wartezeiten für die Umsetzung von Feature-Requests haben sowohl auf Kunden- wie auch Lieferantenseite dazu geführt, eine Transition des laufenden Projektes in ein agiles Modell zu planen.

3 Zielsetzung

Aufgrund der Teamgröße fiel die Wahl auf ein skaliertes SCRUM-Modell [1]. Das gewählte Konzept im Sinne von „Scrum of Scrums“ [2] unterstützt den Gedanken der kleineren, agilen Teams in einem Rahmen, der dem größeren Projektrahmen Rechnung trägt.

Konkrete Zielsetzung für die Transition des Testkonzeptes war, neben der reibungsfreien Unterstützung des SCRUM-Vorgehens, Bewährtes (erworbene Skills, Methoden und Artefakte) nicht zu verlieren, sondern zu integrieren.

Um „auf dem Weg“ eine Erwartungskontrolle und ggf. Kurskorrekturen vornehmen zu können, wurde die Umstellung nicht als „Big Bang“, sondern in mehreren Abschnitten geplant.

4 Herausforderungen

Die, rein faktischen, Anforderungen an das geänderte Team-Setup und die gewünschten, deutlich kürzeren Lieferzyklen stellen nur die Spitze des Eisberges dar, die es im Rahmen einer solchen Verfahrensumstellung zu beachten gilt.

Wie auch an zahlreichen anderen Projektbeispielen abzulesen, ist der Erfolg einer Weichenstellung in Richtung Agilität von viel weicheren Faktoren abhängig:

- Kompensation fehlender agiler Erfahrung bei Teammitgliedern
- Kommunikation der agilen Methodik in das Team
- Gemeinsames Verständnis, dass es sich nicht nur um einen alternativen Prozess handelt, sondern um einen Kulturwandel [3]

Gerade der letzte Punkt wird besonders deutlich, wenn es heißt, dass im Team alle Mitglieder „gleichrangig“ sind und „das Team“ für ihr Produkt verantwortlich zeichnet und sich dabei selbst organisiert. Dies hat zweifellos deutliche Auswirkungen auf die Disziplinen Entwicklung und Test.

Als weitere, projektspezifische und herausfordernde Problematik ist die Tatsache zu sehen, dass kundenseitig nur sehr eingeschränkt und bei den beteiligten Drittlieferanten zunächst gar nicht agil gearbeitet wird. Die Konsequenz ist

- Asynchrone Lieferungen der Drittanbieter-Produkte
- Verbindlich vorgegebene Release-Termine, die sich nicht am Sprint-Rhythmus orientieren

Die Herausforderung für das neue Testkonzept ist es dabei, sowohl der teaminternen Sprint-Orientierung als auch der kundenseitigen Release-Terminplanung Rechnung zu tragen. Und dies ohne die SCRUM-Teams in ihrer Autonomie zu stark einzuschränken.

5 Lösungsansatz und Ergebnisse

Um den, eingangs geschilderten, kulturellen und mentalen Herausforderungen zu begegnen, wurden SCRUM-erfahrene Mitarbeiter, teils aus anderen Projekten, sowohl auf Kunden- wie auch auf Herstellerseite als Coaches und Reviewer für das neue Projekt-Setup gewonnen. Für die einzelnen Diszipli-

nen wie Anforderungsmanagement, Entwicklung und Test wurden Taskforces gebildet, die entsprechende Konzepte ausgearbeitet haben. Damit haben sich die Teammitglieder von Anfang an mit dem anstehenden Kulturwandel auseinandergesetzt und an den Methoden mitgearbeitet, die anschließend ihren Alltag bestimmen werden.

5.1 Testkonzept

Das neue Testkonzept ist von Grund auf im Hinblick auf agile Teams, Sprints und Stories ausgerichtet. Dabei wird, im Hinblick auf Skill-relevante Elemente wie Semantik, Best Practices und Dokumentationsstrukturen, großer Wert auf eine Kompatibilität zu bisherigen Erfahrungen gelegt.

Ein erfolgreicher „Bestandsschutz“ von mehr als 4.500 logischen Testfällen bedeutet auch Investitionsschutz und ist als vertrauensbildende Maßnahme gegenüber dem Kunden nicht zu unterschätzen.

Die gravierenden Unterschiede zwischen dem bisherigen und dem neuen Testvorgehen sind in der Tabelle am Seitenende aufgeführt.

5.2 Anpassung Prozessvorgehen

Um der speziellen Herausforderung gerecht zu werden, dass nicht alle Projektbeteiligte im SCRUM-Rhythmus „ticken“, ist der Testprozess so gestaltet, dass er ein asynchrones Vorgehen unterstützt.

Dazu wird berücksichtigt, dass das Programmmanagement auf Kundenseite langfristig vorlaufende Release-Termine festlegt, die als unverrückbar gelten und sich an keinerlei Sprint-Rhythmus halten.

Im Gegensatz dazu steht das Sprint-orientierte Vorgehen, hier im gewählten 3-Wochen-Takt. Am Ende jedes Sprints steht per Definition ein nutzbares Produkt.

Die Herausforderung ist, beide parallelaufenden

Projektvarianten im Sprint-Rhythmus in Deckung zu bringen.

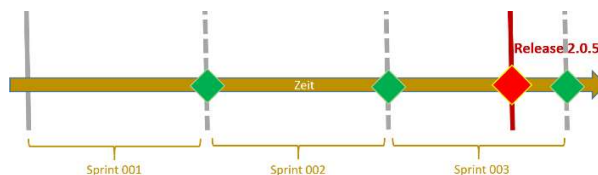


Abbildung 1: Sprint-Rhythmus vs. Release-Meilenstein

Dazu werden in dem speziellen Sprint, der auf einem geplanten Release Termin abzielt, spezielle Stories (Integrationstest-Stories) eingeplant (in Abbildung 2 grün dargestellt). Für das SCRUM-Team stellt sich der Sprint methodisch wie jeder andere Sprint dar. In der hier vorliegenden Dokumentation wollen wir ihn „Integrations-Sprint“ nennen.

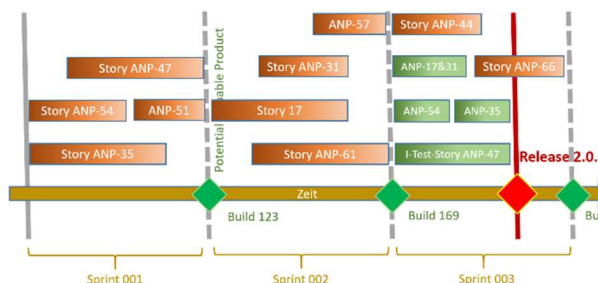


Abbildung 2: Reguläre Sprints und Integrations-Sprint

In der Praxis bedeutet dies, dass innerhalb dieses Sprints die Produkte der weiteren Entwicklungshäuser (somit aus asynchronen Prozessen) und die bisherigen Sprint-Artefakte zusammengebracht und integrativ getestet werden. Faktisch weicht dieses Vorgehen nicht vom Release-Testzyklus ab, der durchgeführt wurde, als noch alle Lieferanten deckungsgleich, iterativ entwickelt und geliefert haben. Gleichzeitig bietet die Einplanung der Integrationstest-Stories über das Backlog die Möglichkeit, dass die SCRUM-Teams ohne Medien- und Methodenbrüche arbeiten können.

Kategorie	Bisher	Heute	Optimierung
Aufwands-optimierung im Testentwurf	Schwerpunkt im Testfallentwurf auf Anwendungsfall oder Testobjektbasierte Unittests.	Testfallentwurf auf Story-Ebene mit Schwerpunkt auf die Akzeptanzkriterien.	Bei Stories mit „Change“-Charakter müssen keine neuen Tests entwickelt werden, sofern im Bestand bereits existierende Tests das Szenario abdecken.
Testmanagement/ Testabdeckungskriterien	Neben anderen Kriterien wurde die Testabdeckung gegen die Anwendungsfall-Spezifikationen gemessen. Dabei wurde in Abdeckungsgrade für manuelle Tests und automatisierte Regressionstests unterschieden.	Testabdeckung wird gegen die einzelnen Sprint-Produkte (auf Story-Basis) gemessen. Unterschieden wird in automatisierte Regressionstests, manuelle Tests zum Sprintergebnis und manuelle Tests finaler Release-Versionen.	Die Testfallspezifikationen für die manuellen Sprinttests können häufig, nahezu 1:1 als Release-Tests verwendet werden

Tabelle 1: Änderungen im Testvorgehen

Diese spezielle Methodik erlaubt es, dass nicht das gesamte Projektteam auf einen Schlag auf die SCRUM-Methodik umgestellt werden muss. Denn ebenso wie die externen Lieferanten liefern mit diesem Vorgehen auch SCRUM-Teams und klassische Teams ihre Entwicklungsergebnisse in den Integrations-Sprint ein.

5.3 Anpassung Artefakt Strukturen

Der Testprozess im Projekt ist weitestgehend an die Nomenklatur des ISTQB-Syllabus angelehnt. Dies sollte auch nach der Umstellung so bleiben.

Im Laufe des Projektes wurde das eingesetzte Spezifikationswerkzeug (TREND/Analyst [5]) für jede Teststufe [6] entsprechend erweitert. Durch Nutzen des modellbasierten Ansatzes werden so Testfälle und Spezifikationsartefakte ohne Medienbrüche miteinander verbunden und eine bidirektionale Traceability [7] von der Anforderung bis zum Testfall ermöglicht.

Das bestehende Dokumentationsmodell kann beibehalten werden, da der bisher gelebte Testprozess und die Testartefakte am fundamentalen Testprozess [8] ausgerichtet sind.

Lediglich für die hinzugekommenen sprint-spezifischen Testmaßnahmen wurde eine neue Teststufe „Sprinttests“ eingeführt.

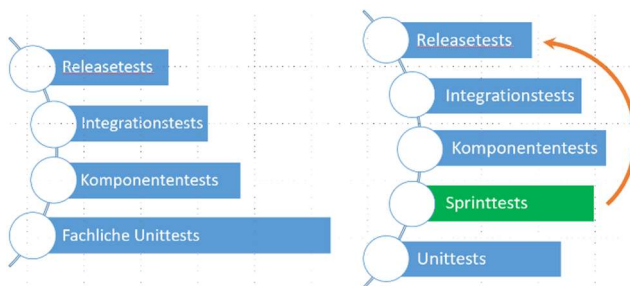


Abbildung 3: Teststufen erweitert um Sprinttests

Sprinttests umfassen Tests der jeweiligen Sprintergebnisse. Sie bilden damit gleichzeitig die Grundlage für die zukünftigen Tests, die im Rahmen einer Release-Lieferung durchzuführen sind.

Diese Anpassung erlaubt die Erstellung von Testfällen an Stories und die Realisierung sowie die Messung der Testabdeckung im Rahmen einzelner Sprints.

Da sich alle Artefakte in einem Namensraum bewegen, ist eine Wiederverwendung in allen Dimensionen möglich:

- a) Innerhalb der Story-Entwicklung werden Testfälle wiederverwendet, die den spezifizierten Use Case zugeordnet sind.
- b) Für Use Case übergreifende Szenarien wird auf Testfall-Spezifikationen in den Teststufen Komponenten- und Integrationstest zurückgegriffen.

- c) Testfälle aus den Sprints werden im Rahmen kommender Release-Tests wiederverwendet.

Die Verknüpfung der Artefakte untereinander erlaubt neben der Traceability auch eine reportunterstützte Auswertung im Rahmen des Testmanagements

5.4 Testanalyse-Systematik

Um im Rahmen der umfassenden Teamgröße ein einheitliches Vorgehen bei der Testanalyse zu unterstützen, wurden in der Vergangenheit systematische Vorgehen etabliert und geschult. So wird ein Großteil der Regressionstests als „fachliche Unittests“ auf Grundlage der spezifizierten Anwendungsfälle [9] ermittelt. Als Analysewerkzeug kam dazu eine Cause/Effect Matrix [10]/[11] zum Einsatz

Dieses, bereits im Projekt etablierte Verfahren, wird im Zuge der Transition nun auf die Stories im SCRUM-Kontext angewandt. Diese Adaption minimiert den Schulungsaufwand im Team und erlaubt weiterhin den Einsatz bewährter Tools und Denkmethode ohne gravierenden Migrationsaufwand nach sich zu ziehen.

5.5 Rollen und Verantwortlichkeiten

Die einzelnen SCRUM-Teams sind standardgemäß als Teams gleichwertiger Mitglieder aufgesetzt. Die Teammitglieder unterscheiden sich lediglich in ihren Skills (Entwickler/Tester) nicht durch ihre Kompetenz. Alle gemeinsam zeichnen verantwortlich für die jeweiligen Sprintergebnisse.

Da im skalierten SCRUM mehrere Teams für das gesamte Projektergebnis stehen, sind „Querschnittsrollen“ etabliert, wobei für den Testbereich die Rolle „Test Manager“ eingeplant ist. Ganz im kollaborativen Geiste des agilen Ansatzes hat der Testmanager eher eine moderierende Rolle. Seine Aufgabe besteht darin, die Erkenntnisse der Teams zur Verbesserung des Qualitätsmanagements zu moderieren und die Teammitglieder im Bedarfsfall zu coachen.

Im hier beschriebenen Beispiel wurde die Rolle etabliert, indem der Testmanager das beschriebene Testkonzept initial vorgeschlagen und mit dem ersten etablierten SCRUM-Team ausgearbeitet und feingeschliffen hat. Dadurch wurden die Teammitglieder gleich in den Gestaltungsprozess involviert und verstehen im Nachgang die Qualitätsvorgaben, insbesondere „Definition of Ready“ (DoR) und „Definition of Done“ (DoD), als ihre eigenen.

Für den Austausch zwischen den Teams ist als zentrale Instanz eine „Community of Practice“ (CoP Test/QA) [12] etabliert worden.

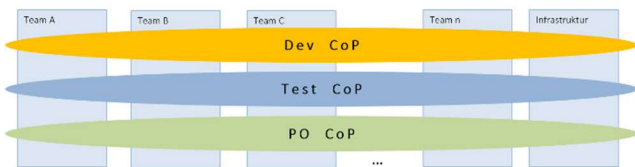


Abbildung 4: Organisatorisches Setup m. Querschnittsrollen

Diese Plattform ermöglicht (insbesondere durch regelmäßige Treffen) allen interessierten Teammitgliedern, bzw. Vertretern aus den SCRUM-Teams, ihre Interessen und Anregungen auszutauschen, Lösungen anzufragen und zu diskutieren, sowie gemeinsame Standards im gesamten Team weiter zu entwickeln.

5.6 Gelebter Prozess

Das Ergebnis gestattet den Projektverantwortlichen einen gleitenden Übergang in die neue Welt. Dabei ist es möglich, das Tempo selbst zu bestimmen und im Bedarfsfall zu variieren. Im hier zitierten Fall wurde zunächst ein SCRUM-Team (Team „A“) etabliert. Nach einer Erprobungsphase von drei Sprints wurde Team „B“ aufgesetzt. Nach einem Wissenstransfer wurde mit dem fünften Sprint das nächste Team („B“) etabliert. Nach insgesamt neun Sprints bewertete der Kunde die Transition als erfolgreich und zielführend und es wurden die verbliebenen Teammitglieder in weitere zwei SCRUM-Teams aufgeteilt.

Damit ist die Übergangsphase abgeschlossen und das gesamte Projekt arbeitet in dreiwöchigen Sprints. Ausstehend ist noch die Verkleinerung der Teams durch Einsatz eines fünften SCRUM-Teams. Dies ist eine Frage der Zeit und vor allem verfügbarer Ressourcen auf Kundenseite.

6 Fazit und Ausblick

Der Fakt, dass agile Modelle neben der Methodik vor allem einen Kulturwandel bedeuten wirkt sich auch auf das Testvorgehen aus. Das Denken und Handeln im Team bezieht sich nicht nur auf den Erstellungsvorgang der Software, sondern auch auf die Qualitätssicherung. Die Integration der Tester in die Teams führt dazu, dass die Betrachtung der Qualität auch in Augen der Entwickler eine neue Dimension bekommt. Nicht nur das Produkt, sondern auch seine Qualität wird als Gesamtergebnis und damit Leistung des Teams wahrgenommen. Der kausale Zusammenhang zwischen Qualitätsbewusstsein, Testmethodik und Teamergebnis wird für alle Beteiligten unmittelbar erlebbar.

Die Adaption des Testvorgehens aus der Iterativen Welt und insbesondere die Einführung der „Integrations-Sprints“, hat es ermöglicht, die Transition gleitend und frei bestimmbar zu gestalten.

Da aktuell das Transitionsergebnis noch recht jung ist, wird erwartet, dass in der nächsten Zeit ein kontinuierlicher Lern- und Verbesserungsprozess

durchlebt wird. Hier haben insbesondere die Querschnittsrollen und die „Community of Practice Test/QA“ die Aufgabe, Lernergebnisse aus den einzelnen SCRUM-Teams im Hinblick auf das gesamte Projektteam zu moderieren.

Literatur

- [1] Bradley, Charles: Resurrecting the Much-Maligned Scrum of Scrums: www.scrum.org/resources/blog/resurrecting-much-maligned-scrum-scrums
- [2] von Bergen, Katja: www.computerwoche.de/a/agile-methoden-loesen-keine-probleme-machen-sie-aber-sichtbar,3331014
- [3] Eckstein, Jutta: Agile Softwareentwicklung in großen Projekte, 2. Auflage 2012, D-Punkt-Verlag, ISBN 978-3-89864-790-8
- [4] Fowler, Martin: martinfowler.com/bliki/TestPyramid.html.
- [5] TREND/Analyst: Model Driven Requirement Engineering: www.gebit.de/loesungen/technische-loesungen/trend-analyst-requirements.html
- [6] Witte F. (2016) Teststufen. In: Testmanagement und Softwaretest. Springer Vieweg, Wiesbaden
- [7] International Software Testing Qualifications Board, Founcaion Syllabus 2018, S. 22: Traceability between the Test Basis and Test Work Products
- [8] Spillner, A.undT. Linz: Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester; Foundation Level nach ISTQB-Standard. dpunkt-Verlag, 2005.
- [9] Software Testing Help: Use Case Testing www.softwaretestinghelp.com/use-case-testing/
- [10] Copeland, Lee: A Practitioner's Guide to Software Test Design, Artech House, 2003, ISBN 1-58053-791-X
- [11] Wingens, Silke: test.silke-wingens.de/2011/02/14/dynamische-analyse-black-box-test-ursache-wirkungs-darstellung-und-entscheidungstabellen/
- [12] Simons, Kai: www.ksimons.de/2013/12/scrum-skalieren-communities-of-practice-cop-verstaendlich-erklaert/