



□ Markus Adam

[E-Mail: markus.adam@sogeti.de]

arbeitet seit 1995 im Bereich Qualitätssicherung für Software, er ist seit 2003 zertifizierter SPiCE[®] Assessor und war in vielen Projekten für die Bewertung und Definition von QS-Prozessen zuständig. Seit August 2008 ist er für Sogeti Deutschland als Senior Consultant tätig.

Agile Entwicklung und Testmodelle: Integration/Kooperation ist kein Widerspruch

In den letzten Jahren wurden IT-Professionals von einem neuen „Hype“ hinweggefegt, benennen wir ihn mal den „Agilen Hype“: Die Revolution am Softwareentwicklungsmarkt.

In letzter Zeit rückten eine Menge an agilen SW-Entwicklungsmodellen/Methoden in den Blickpunkt der SW-Branche, wie XP, TDD, Scrum, um nur einige zu nennen.

Scrum als Beispiel definiert ein Management Modell für agile Softwareprojekte, wobei von der Anforderung mit dem Kunden über die Planungsphase bis zu den Sprints selbst ein Prozessmodell vorhanden ist. Ziel ist, den Gesamtanforderungskatalog in viele kleine beherrschbare Einheiten zu unterteilen, die in Iterationen von 14 – 30 Tagen umgesetzt werden. Und das inklusive Test und Dokumentation, mit einer Besonderheit, genannt Retrospective, die zusätzlich zum Abschlussreview durchgeführt wird. Behandelt wird in dieser Runde das vorhandene Verbesserungspotenzial und seine Umsetzung.

Software schneller und ohne Qualitätseinbußen produzieren zu können, liegt ja durchaus im Interesse des Managements und auch der Entwickler.

Zugleich werden Modelle wie CMMi, ISO/IEC 15504 SPiCE[®] immer bedeutender, vor allem in der Automotive Branche aber auch in der Medical Software und Financial Software. Diese Modelle stehen im Ruf, auf sogenannten traditionellen SW-Entwicklungsansätzen und Modellen wie Wasserfall und V-Modell zu basieren. Beide

Modelle sind als Prozessbewertungsmethoden entstanden, das SW-Entwicklungsmodell stand bei beiden nicht im Vordergrund. Dennoch werden sowohl SPiCE[®] als auch CMMi für die Definition von Prozessen in der Softwareproduktion verwendet.

Auch die QS- und Testprozesse der Softwarebranche rückten in den letzten Jahren mehr in den Fokus der Softwarebranche, Review und methodisches Testen sind in den letzten Jahren eher die Regel als die Ausnahme. Zu den Vertretern auf diesem Zweig zählen ISTQB[®], TMap Next[®], und weitere proprietäre Modelle verschiedenster Anbieter. ISTQB[®] definiert einen Bausatz und Begrifflichkeiten für Testprozesse, der Prozess im speziellen ist nicht im Fokus der Basisausbildung. TMap Next[®] geht einen Schritt weiter und definiert sowohl ein Ablaufmodell, sowie einen kompletten Prozess mit Bausatz für die Abläufe in den Aktivitäten.

Die Frage, die sich stellt: können diese Trends auch miteinander kooperieren? Kann zum Beispiel in einer an SPiCE[®] angelegten Prozesslandschaft Scrum sinnvoll als Entwicklungsmodell eingesetzt werden? Kann ein Scrum Testprozess auf ISTQB[®] oder TMap Next[®] basieren?

Das folgenden Beispiel aus verschiedenen Praxiserfahrungen könnte darauf eine Antwort geben.

Wir finden bei einem Softwareproduzenten folgende Umgebung:

- Die Definition der Software-Lifecycle-Prozesse basiert auf dem ISO/IEC 12207 Standard, der auch als Prozess-Referenz-Modell für SPiCE[®] dient.
- Die verwendete agile Methode ist Scrum mit XP als Entwicklungsmodell.
- Die Testprozesse basieren auf TMap Next[®], auch die innerhalb der Sprints.
- Gegeben sei ein Überblick über die einzelnen verwendeten Modelle.

ISO/IEC 15504 SPiCE[®]: Ist der Prozessbewertungsstandard der ISO, SPiCE[®] definiert einen ordnenden Rahmen, dem Assessments folgen sollten, sowie eine Definition SPiCE konformer Prozessreferenzmodellen (PRM's) und Prozess - Assessment - Modellen (PAM's). Das PRM definiert die Prozesse des Modells, und auch die Resultate einer erfolgreichen Implementation des Prozesses.

Im Prozess Assessment Modell finden wir, unter anderem, Indikatoren wie prozesstypische „Workproducts“, Best Practises, ein Mapping mit dem Reifegradmodell etc., die als Richtlinien für die Prozesse in der Praxis dienen können. SPiCE[®] sagt allerdings nichts darüber aus, wie diese Aktivitäten

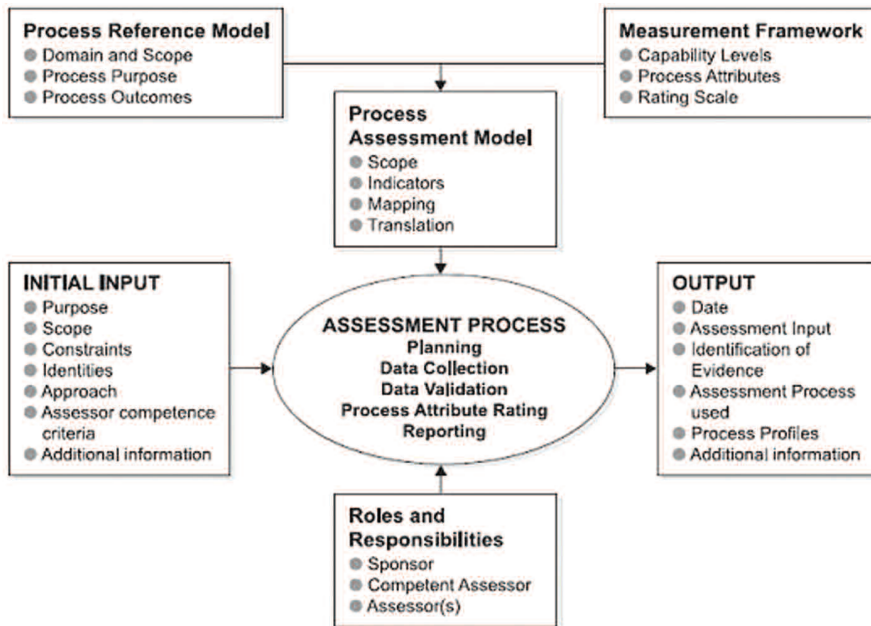


Abbildung 1: SPiCE® Modellübersicht. Quelle: Hörmann, Dittmann, Hindel, Müller: SPiCE in der Praxis, dpunkt-Verlag, 2006

durchgeführt werden sollen. Beschreibungen von Reviewprozeduren oder SW-Entwicklungsmodellen sind nicht Bestandteil von SPiCE®.

Scrum kann als agile SW-Entwicklungsmanagement-Methode beschrieben werden. Scrum beschreibt den SW-Entwicklungsprozess von der Anforderungssammlung über die Planung, bis zur Durchführung und „Auslieferung“. Dabei werden die Anforderungen an die Software in einem „Product Backlog“ definiert, der vom „Product

Owner“ - ein Vertreter des Kunden im Idealfall verwaltet wird. Aus dieser Sammlung werden in der „Sprintplanung“ im sogenannten „Sprint Backlog“ Aufgaben für den Sprint generiert. Diese werden innerhalb des Sprints abgearbeitet, inklusive Design, QS, Test, und Dokumentation.

Scrum teilt also die zu erstellende Software in leichter steuerbare kleine Einheiten, die in sogenannten Sprints erstellt werden. Dabei können, wenn notwendig, mehrere Teams parallel arbeiten. Der Vorteil liegt in

der besseren Steuerbarkeit kleinerer Einheiten aus Projektmanagementsicht und in der selbstorganisierenden Arbeit des Scrum-Teams und der Teams untereinander, wenn mehrere vorhanden sind. Zusätzlich bietet Scrum die Möglichkeit, sich schneller an sich ändernde Umstände anpassen zu können. Während in „klassischen“ Modellen jede Änderung mit einem großen zeitlichen Aufwand zu bezahlen ist, sollte Scrum bereits im nächsten Sprint reagieren können. Auf diese Weise kann z. B. zu einem recht frühen Zeitpunkt die Undurchführbarkeit einer Anforderung oder des gesamten Projektes erkannt werden. Das spart Zeit und Kosten.

TMap Next® wiederum definiert einen modularen Baukasten an Prozessaktivitäten, die eine hohe Reife des Testprozesses ergeben, wenn sie sinnvoll eingesetzt und an die Umstände angepasst werden. Dabei werden Review-Strategien, Testdesign-Techniken, Hinweise zur Infrastruktur etc. angeboten.

TMap Next® basiert auf dem BDTM (Business Driven Test Management)-Ansatz, der eine vollständige Begleitung des Entwicklungsprozesses durch den Testprozess vorsieht. Bereits in der Planungsphase der Software kann über Testbarkeitsreviews und Risikoanalysen eine Auswahl an geeigneten Testdesignverfahren getroffen, die Anforderungen an die Infrastruktur definiert werden usw. Ziel ist es, die Testaktivitäten soweit wie möglich mit den Entwicklungsaktivitäten parallel laufen zu lassen, um Zeit und Ressourcen zu sparen. So wenig Testaktivitäten wie möglich sollen den „kritischen Pfad“ belasten. Was für den einzelnen Testprozess geeignet ist, kann vom Benutzer individuell bestimmt werden. Von Risikoanalysen bis zur Infrastrukturcheckliste bietet TMap® einen Baukasten für nahezu alle Anforderungen eines Testprozesses.

Kommen wir zum Projekt, in dem diese Ansätze gemeinsam verwendet werden. Folgende Umgebung liegt vor:

Das Projekt ist die Entwicklung des Business Logic Layers und die Persistierung der Business Objects, für eine J2EE basierte SW-Lösung. Der Auftrag stammt von einem Kunden, der noch andere Zulieferer beschäftigt, die ebenso Teile für dieses Produkt liefern. Aufgrund der vertraglichen Situation der einzelnen Zulieferer mit dem Auftraggeber läuft die Abstimmung und Koordination der einzelnen Zulieferer zentral, das schließt die Abgabetermine für Tests etc. mit ein.

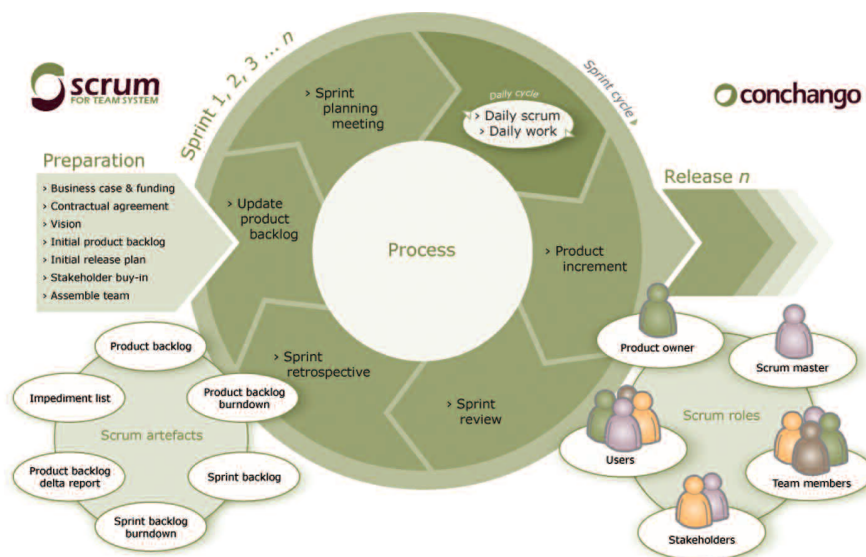


Abbildung 2: Scrum Prozessübersicht. Quelle: Conchango

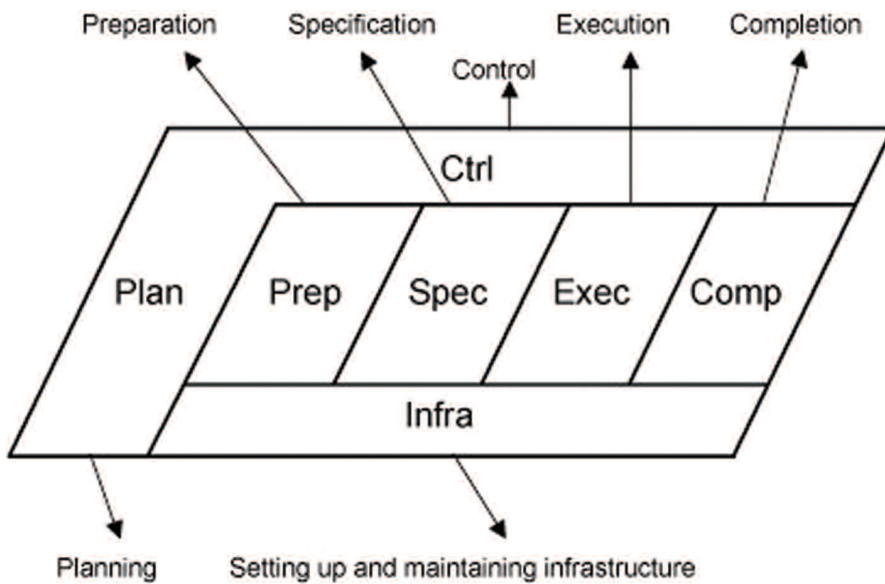


Abbildung 3: Prozessaktivitäten in TMap®. Quelle: Sogeti Nederland B.V.

Die Prozesse der Firma sind an ISO/IEC 15504 angelegt. Die Definition der Prozesse folgt der ISO/IEC 12207 Software Lifecycle Processes. Das Entwicklungsmodell für den Prozess ENG 6: Software Construction ist Scrum. Um die Forderungen von SPiCE® Level 3 abzudecken, wird innerhalb der SW – Entwicklung TMap Next® als Basis für die Testprozesse verwendet.

Da die anderen Zulieferer Teile wie GUI, Kundendaten etc. liefern, kann innerhalb der Scrum-Sprints nicht der gesamte Integrationstest erfolgen, da man auf die Lieferung der anderen Zulieferer angewiesen ist.

Aufgrund der Organisation des Auftraggebers ist eine Steuerung der anderen Beteiligten nur beschränkt möglich.

Das Projekt ist sehr umfangreich, es sind also eine Anzahl an Sprints notwendig, um die gesamte Software zu erstellen. Aufgrund des Umfangs haben wir mehrere Scrum-Teams im Einsatz. Hierbei ist anzumerken, dass die Resultate der Sprints einen Unit- und Modultest durchlaufen haben. Ein Gesamtintegrationstest steht noch aus. Da sich dieser nur schwer in den Sprint integrieren lässt, wird ein zusätzlicher Integrationstest, der die Ergebnisse der Sprints

und des Zulieferers in testbare Module zusammenfasst, außerhalb der Sprints notwendig. Dieser kann alle Anforderungen von SPiCE®, TMap® etc. abbilden, ohne die Sprints zu beeinträchtigen (abgesehen von der Behebung gefundener Fehler).

Hier können im Bereich des Integrationstests beliebige Testmodelle verwendet werden, in dem hier verwendeten Beispiel TMap®.

Der Integrationstest, der alle 5 Sprints verfolgt, kann durchaus so geplant und vorbereitet werden wie in TMap® definiert. Die Zeit dafür ist in den vorliegenden Intervallen der Lieferungen gegeben. Als Testbasis können die Use Cases und die Klassenmodelle dienen. Im vorliegenden Beispiel werden als „Zwischenschritt“ am Ende der Sprints zentral Klassenmodelle erstellt, die in ein Knowledge-Management System eingepflegt werden.

So hat das Integrationstest-Team Zugriff auf die Information, ohne die Sprints zu stören. Der Integrationstest selbst basiert auf TMap® mit allen Schritten, die innerhalb des Modells definiert werden, wie Planung, Vorbereitung der Infrastruktur, Testbarkeitsreview, Testdesign, Dokumentation der Testfälle etc. Als „Nebenprodukt“ wird auf diese Weise die gesamte Software dokumentiert. Diese kann bei Bedarf in der Sprintplanung Verwendung finden, zum Beispiel im Design.

Die Testfälle des Integrationstests können auch, wenn genügend Ressourcen finanzieller und personeller Natur vorhanden sind, für nachfolgende Regressionstests der gesamten Software also auch der Module des Zulieferers, automatisiert werden. Vor allem in der Betriebs- und Wartungsphase ist ein Regressionstest unerlässlich.

Diese Vorgehensweise steigert die Effizienz des Integrationstests, da die Durchführungsdauer minimiert wird und der Integrationstest den kritischen Pfad weniger „belastet“.

Solche oder ähnliche Kombinationen bieten in der Praxis eine gemeinsame Basis für agile Methoden und Prozess- bzw. Testmodelle.

Fazit:

Daraus folgt auch, dass eine Kombination von agilen Methoden und sogenannten traditionellen Prozess- und Testmodellen durchaus sinnvoll ist und konsequent umgesetzt zum Erfolg führen kann.

Es sei festgehalten, dass agile Entwicklung und agiles Management durchaus eine Berechtigung in der Softwareindustrie ha-

Level	Attributes
5 "Optimising"	<ul style="list-style-type: none"> Process innovation Continuous optimization
4 "Predictable"	<ul style="list-style-type: none"> Process measurement Process control
3 "Established"	<ul style="list-style-type: none"> Process definition Process deployment
2 "Managed"	<ul style="list-style-type: none"> Performance management Work product management
1 "Performed"	<ul style="list-style-type: none"> Process performance

Abbildung 4: SPiCE® Reifegrade. Quelle: Kugler Maag CIE

ben. Um jedoch eine optimale „Ausbeute“ zu erzielen, wäre eine Kombination mit traditionellen Methoden/Modellen angebracht und sinnvoll, vor allem im Bereich Qualitätssicherung und Test. Selbst innerhalb der Sprints können Modelle, wie SPiCE[®] und TMap Next[®] als Guideline fungieren, um die Frage „was habe ich zu tun?“ zu beantworten. Über das „wie“ und „in welchem Ausmaß“ muss das Team oder die Teamleitung selbst entscheiden. Ziel müsste jedoch sein, alle Aktivitäten des SW-Entwicklungsprozesses und des dazugehörigen Testprozesses innerhalb des Sprints abuarbeiten. Dass der Testprozess dennoch auf TMap[®] oder ISO 9000ff. basiert, sollte kein Problem darstellen. Die nötigen Anpassungen an agile Umgebungen sollten leicht durchzuführen sein.

Innerhalb und außerhalb der Sprints können Modelle wie TMap Next[®], vor allem im Integrationstest, z. B. in Projekten mit externen Zulieferern, eine hochwertige Ergänzung zu Scrum darstellen, um die Produktion qualitativ hochwertiger Produkte abzusichern. ■

□ Literatur

ISBN 0-387-23182-X: Process Assessment and Improvement: A practical guide for managers, quality professionals and assessors von Han van Loon

ISBN 0-387-23172-2: Process Assessment and ISO/IEC 15504 A Reference Book von Han van Loon

ISBN-10: 3898644618: TMap[®] Next - Ein praktischer Leitfaden für ergebnisorientiertes Softwaretesten von Tim Koomen (Autor), Leo van der Aalst (Autor), Bart Broekman (Autor), Michiel Vroon (Autor)

ISBN-10: 3866456433: Scrum im Unternehmen von Ken Schwaber von Microsoft Press Deutschland

ISBN-10: 3898644731: Agile Softwareentwicklung: Werte, Konzepte und Methoden von Wolf-Gideon Bleek und Henning Wolf von dpunkt-Verlag

TMap[®] und TMap Next[®] sind eingetragene Warenzeichen der Sogeti Nederland B.V.