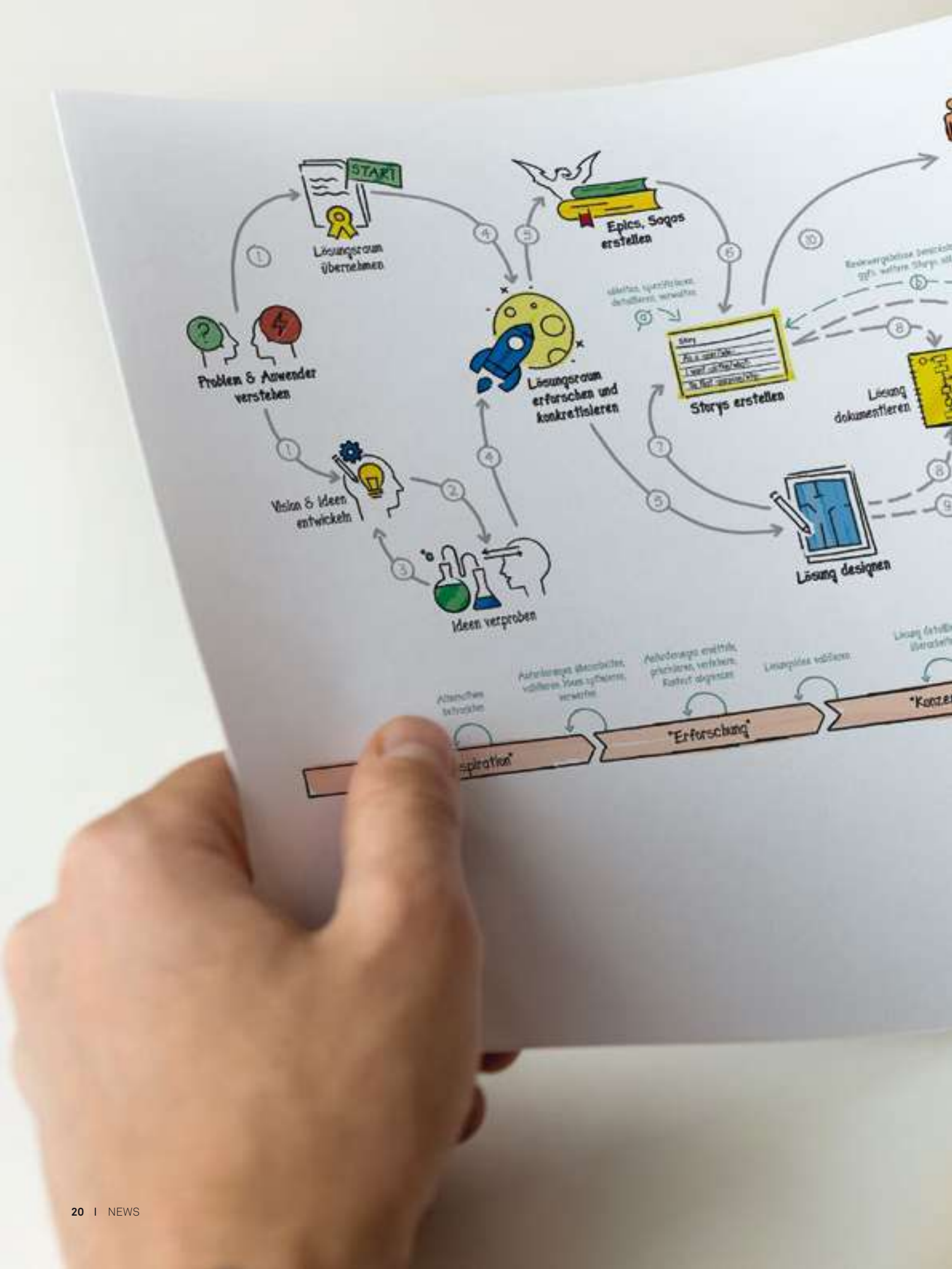
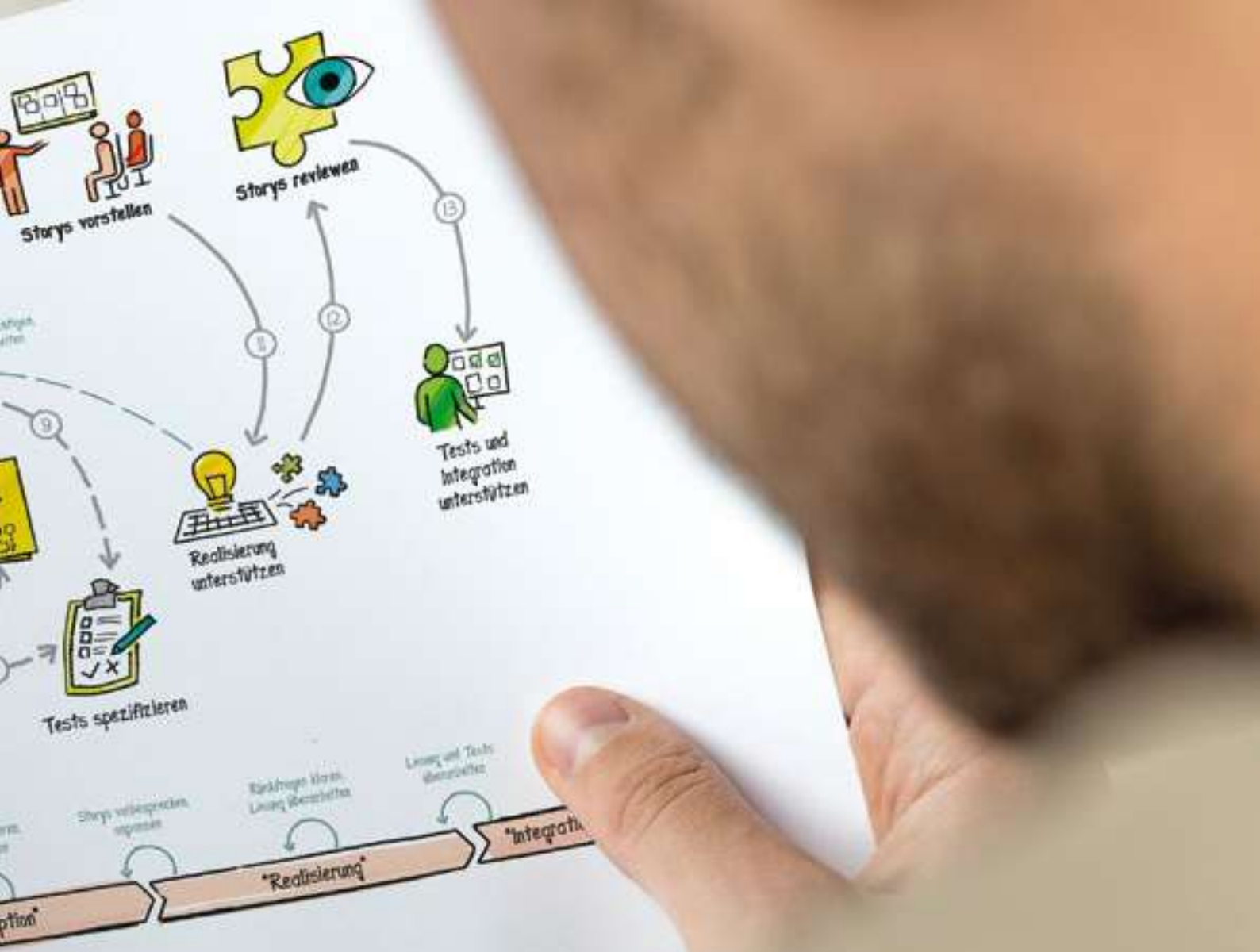

START
Lösungsraum übernehmen
Epics, Sagas erstellen
Problem & Anwender verstehen
Lösungsraum erforschen und konkretisieren
Storys erstellen
Lösung dokumentieren
Vision & Ideen entwickeln
Ideen verproben
Lösung designen
"Erforschung"



Oliver Lukas

Requirements Engineering

Aufgabenportfolio im hybriden Projektumfeld

» Requirements Engineering ist tot – Lang lebe Requirements Engineering!

Dieser Satz enthält all die Widersprüche der Disziplin der Anforderungsanalyse und deren Management, wenn man versucht, diesbezügliche Meinungen, Erfahrungen und Trends im Kontext agiler Softwareentwicklung zu erfassen. Das „agile Manifest", als Mantra für idealisierte, agile Vorgehen, wie beispielsweise eXtreme Programming oder Scrum, befasst sich maßgeblich mit Entwicklungsaufgaben und delegiert die Ermittlung und Dokumentation von Anforderungen an die Rolle des Product Owners. Es ist jedoch wenig verwunderlich, dass diese zentrale Rolle bereits bei mittelgroßen Vorhaben in der Regel mit den komplexen Analyse- und Strukturierungsaufgaben überlastet ist. Auch in agilen Projekten besteht der Bedarf, fachliche Anforderungen vorab zu analysieren und so aufzubereiten, dass sie dann vom Entwicklerteam widerspruchsfrei verstanden und effizient umgesetzt werden können.

Originäre Aufgaben des Requirements Engineerings (RE), wie sie zum Beispiel von der IREB.org formuliert sind, finden sich vermehrt auch in modernen agilen Projekten wieder. Zeitpunkt, Intensität und Umfang des RE-Anteils sind dabei variabel und werden je nach Projekt individuell justiert. Werden Aspekte und Ergebnistypen des klassischen RE in angemessenem Umfang mit einem agilen Vorgehen kombiniert, spricht man auch von einem „hybriden" RE-Projektvorgehen. Ob die RE-Aufgaben dabei innerhalb des Teams oder als separate Zuarbeit für den Product Owner außerhalb des Teams erledigt werden, ist dabei unwesentlich. Beide Varianten haben ihre Vor- und Nachteile und können entsprechend der Beauftragungssituation und Teamstruktur adaptiert werden.

Dennoch ist RE im agilen Kontext nicht mehr das gleiche RE, das von klassischen Projektvorgehen bekannt ist. Lange Vorab-Analysephasen weichen zeitnahen Explorationssprints kurz vor der Implementierung, Bücherwände voll konzeptioneller Ergebnisse werden durch minimalistische „lebende Dokumente" in Kollaborationsplattformen ersetzt. Und umfassende Anforderungssammlungen werden von „schlanken Storys" abgelöst, die ein Wertversprechen gegenüber dem Kundensegment oder Stakeholder abgeben.

Motiviert durch Vorhaben der digitalen Transformation erweitert RE zudem seinen Horizont und ergänzt sich mit ursprünglichen Themenfeldern aus der User-Experience (UX). RE ist somit nicht mehr nur Sammler und Strukturgeber von Anforderungen, sondern beteiligt sich gegebenenfalls auch an der Ideenfindung für spätere Anforderungen. Der Kunde beziehungsweise Systemnutzer, als zentrale Quelle für Bedarfe, muss dabei umfassend verstanden und von Anfang an in die Anforderungserhebung integriert werden. Die Berücksichtigung und Fokussierung hinsichtlich des zugesicherten „Wertversprechens" ist dabei die Basis für die fachlich orientierte Analyse und Lösungsarchitektur.

Sowohl Abbildung 1 als auch die nachfolgende Beschreibung geben einen exemplarischen und subjektiven Blick auf übliche RE-Aufgaben im Kontext agiler Softwareprojekte, deren reale Aufgaben und Phasen in den konkreten Projekten durchaus abweichen können. Gleiches gilt auch hinsichtlich der Vollständigkeit und Rückkopplung der skizzierten Übergabepunkte zwischen den Aufgaben, die sich schon aufgrund des verwendeten agilen Vorgehensmodells unterscheiden können. Ziel dieser „Landkarte" sind die grundsätzliche Einordnung und Verortung von Aufgaben des Requirements Engineers oder Business-Analysten (nachfolgend RE/BA genannt) und seiner Berührungspunkte zu einem agilen Projektvorgehen, jedoch nicht die Festlegung eines allgemeingültigen Sollprozesses.

AGILE BUSINESS-ANALYSE UND AGILE SOFTWAREENTWICKLUNG

Dieser veränderten und erweiterten RE-Perspektive geschuldet, wird dieses Aufgabenspektrum in einem agilen Vorgehen auch als „Agile Business-Analyse" bezeichnet. Sie definiert den Inhalt des Projekts (WAS, WARUM) und realisiert es durch die „Agile Softwareentwicklung" (WIE, WANN). Die nahtlose und enge Zusammenarbeit beider Disziplinen und Sichtweisen ist dabei ein entscheidender Erfolgsfaktor. Damit die „Agile Softwareentwicklung" eine effiziente und wertschöpfende Realisierung gewährleisten kann, ist es das primäre Ziel der „Agilen Business-Analyse", einen angemessenen Reifegrad der benötigten Informationen für das Umsetzungsteam zu erarbeiten. Agilität versteht sich als kontinuierlicher und iterativer Lern- und Verbesserungsprozess. Dies gilt grundsätzlich auch für die RE-Anteile innerhalb eines agilen Vorgehens. Nichtsdestotrotz ergeben sich aufgrund von Kenntnisstand, Rahmenbedingungen und Fortschritt zeitliche Schwerpunkte in Bezug auf Intensität und Detaillierung der RE-Aufgaben. Dies wird in Abbildung 1 durch die Phasen „Inspiration", „Erforschung", „Konzeption", „Realisierung" und „Integration" ausgedrückt. In der Praxis existieren Überlappungen dieser Phasen sowie eine iterative Entwicklung deren Ergebnisse, was zielführend und beabsichtigt ist. Die folgende Tabelle erklärt die einzelnen RE-Aufgabenfelder innerhalb der skizzierten Evolutionsphasen aus Abbildung 1 und referenziert dabei die Übergänge zwischen den Aufgaben durch (n).

Evolutionsphase „Inspiration“

Problem und Anwender verstehen

Das Problem ist das Problem. Bevor nicht die inhaltliche Problemstellung, der Bedarf der zukünftigen Anwender, die Potenziale und Möglichkeiten für die betroffenen Kundensegmente und Stakeholder verstanden wurden, sollte die Lösungsfindung nicht beginnen. Sich dessen bewusst werden und den erforderlichen Freiraum hierfür zu schaffen, ist die Voraussetzung für die zielführende Ermittlung von werthaltigen Anforderungen, also dem gewünschten „Outcome“ einer Lösung. Für den RE/BA ist dies eine stetig begleitende Disziplin, um alle Schritte der Lösungsfindung immer wieder gegenüber der zentralen Problemstellung zu validieren und zu priorisieren. Initial muss sich der RE/BA mit systemischer Empathie in die Anwendersituation hineindenken und gegebenenfalls sogar exemplarisch übernehmen, um für eine nachfolgende Visions- und Ideenentwicklung beziehungsweise Lösungsfindung (1) gerüstet zu sein.

Vision und Ideen entwickeln

Nur in Ausnahmefällen sind Innovationen spontane Ideen, die morgens beim Duschen entstehen. Innovationen schlummern oft in den Köpfen von Spezialisten (und auch Nichtspezialisten) und müssen explizit motiviert und gefunden werden. Beispielsweise werden im Rahmen eines Design-Thinking-Prozesses verschiedene Kreativitätsmethoden eingesetzt, um für ein Geschäftsszenario evolutionäre oder auch disruptive Ideen zu finden und zu entwickeln. Ein RE/BA ist in dieser Phase ein Sparringspartner für Ideen anderer, Ideengeber in seiner fachlichen Domäne und zugleich Moderator und Manager für die gefundenen Ideen. Wesentlich ist hierbei, dass die Ideenentwicklung dabei nicht eingeschränkt, sondern lediglich strukturiert wird. Idealerweise wird bereits zu diesem Zeitpunkt ein „Wertversprechen“ formuliert, das den Mehrwert der Idee repräsentiert und aufzeigt, welches „Problem“ mit dieser Option gelöst oder welches Potenzial gehoben werden soll.

Die vermeintlich besten Ideen (2) werden für eine Verprobung und Validierung vorbereitet. Vermeintlich deshalb, weil eine objektive Ideenbewertung zu diesem Zeitpunkt nur selten möglich ist und als „Best Guess“, also nach bestem Wissen, für die nächste Runde der Ideenverprobung nominiert wird.

Ideen verproben

Die beste oder auch mehrere beste Ideen haben gegebenenfalls noch nicht die Reife, gleich in die konkrete Umsetzung zu gelangen. Je disruptiver oder von der Regel abweichender die Idee, umso sinnvoller sind eine Vorabprüfung der grundsätzlichen Tragfähigkeit und die Validierung auf positive Resonanz. Die Komplexität des Vorhabens ist maßgeblicher Treiber für den Aufwand, den man in dessen Plausibilisierung stecken sollte. Ideen verproben bedeutet auch, dass diese in anderer Form aufbereitet werden können, als sie später tatsächlich zur Umsetzung kommen. So könnten beispielsweise Teilaspekte einer App auch nur als Cartoon skizziert und potenziellen Nutzern vorgestellt werden. Das heißt, die Ideen werden nicht als lauffähiger Prototyp umgesetzt, sondern nur soweit, dass aussagekräftige Rückmeldungen möglich werden. Der RE/BA ist maßgeblich an der Konzeption der Verprobung beteiligt und begleitet die Validierung, um deren Erkenntnisse für nachfolgende Tätigkeiten zu strukturieren und zu dokumentieren.

Stellt sich die Idee als nicht zielführend heraus, so ist ein Rücksprung zum Punkt (3) „Ideen entwickeln“ sinnvoll, um diese zu überarbeiten und mit einer anderen Idee wieder in die Verprobung zu starten. Hat sich die Idee hingegen als wertvoll bestätigt, wird sie im nächsten Schritt als vorgegebener Lösungsraum (4) weiter erforscht und konkretisiert.

Lösungsraum übernehmen

Im Projektalltag zeigt sich allerdings oft, dass RE/BAs erst dann mit ins Boot geholt werden, wenn der Lösungsraum bereits einen gewissen Fokus und Rahmen hat. Zudem beginnen nicht alle Vorhaben auf der grünen Wiese und mit grundsätzlich neuen Ideen, sondern haben eher einen evolutionären als revolutionären Charakter. Das heißt, anstatt durch „Ideen entwickeln und verproben“ den Lösungsraum erst zu ermitteln, wird dieser bereits vorgegeben, indem beispielsweise schon Vorgaben zu den relevanten Geschäftsszenarien und den betroffenen Umsystemen existieren. Idealerweise erlaubt der Lösungsraum jedoch entsprechende Freiheiten, um dogmatische Weiterentwicklungen alter Prinzipien zu verwerfen und stattdessen ein werthaltiges modernes Softwaresystem zu konzipieren. Für den RE/BA bedeutet das, ein Verständnis des Lösungsraums aufzubauen, was oftmals nur anhand übergebener Dokumente oder von referenzierten Regularien oder Gesetzen erfolgen muss.

Basierend auf diesen Rahmenbedingungen und dem erreichten inhaltlichen Verständnis erfolgt im nächsten Schritt die Konkretisierung des Lösungsraums (4).

Evolutionsphase „Erforschung“

Lösungsraum erforschen und konkretisieren

Die kreativen Freiräume werden aufgrund von Detaillierungen und Konkretisierungen im Sinne der Ziellösung sukzessive reduziert. Das heißt, man nähert sich durch Analysen des Umfelds, der technischen Abhängigkeiten sowie der relevanten Geschäftsprozesse und Objekte immer mehr der fachlichen Zielarchitektur. Die Detaillierung wird jedoch auf ein sinnvolles Minimum beschränkt, sodass die Ziellösung erkennbar wird und zugleich ein Wertversprechen formuliert und validiert werden kann. Mit der Verfeinerung des initial ermittelten Lösungsraums erfolgt auch die Definition, was nicht Teil der Lösung sein wird – also der Abgrenzung des Systemkontexts.

Wird während der Analyse festgestellt, dass der skizzierte Lösungsraum nicht tragfähig ist, erfolgt idealweise ein Rücksprung zur Ideenverprobung, eine Anpassung des Lösungsraums oder, im schlimmsten Fall, der Abbruch des Vorhabens. Der RE/BA ist maßgeblich an der Strukturierung und Erarbeitung des „Big-Pictures“ der fachlichen Zielarchitektur beteiligt oder gar dafür verantwortlich. Das konzipierte „Big-Picture“ und die erarbeiteten funktionalen und qualitativen Rahmenbedingungen sind Basis für die Erstellung agiler Anforderungsartefakte (5) sowie für die weitere Detaillierung des Lösungsdesigns (5).

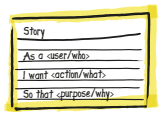
Epics, Sagas erstellen

Die erarbeitete Grobstruktur und gefundenen Prozesse des Lösungsraums sind Quelle für die Erstellung der agilen Artefakte im Product Backlog. Aufgrund des noch abstrakten Reifegrads der Lösung werden Epics und Sagas erstellt, die in der Regel für eine Umsetzung noch zu wenig detailliert und eventuell nicht die passende Größe haben. Sie dienen jedoch der Priorisierung der Themenfelder und umfassen den zu erstellenden groben funktionalen Umfang der Ziellösung. Der RE/BA unterstützt hierbei den Product Owner in der Erstellung der Epics, indem er diese inhaltlich und semantisch aufbereitet und funktional gruppiert, damit sie den Lösungsrahmen vollständig abdecken und zugleich für die weitere Verfeinerung geeignet sind.

Vorabgespräche mit dem Feature-Team oder den Softwarearchitekten ermöglichen eine Bewertung hinsichtlich der Realisierbarkeit des Epics und einer groben Einschätzung dessen Größenordnung, zum Beispiel in T-Shirt Größen (XS, S bis XXL). Je nach Projektvorgehen können Epics auch vom Team selbst „refined“ werden, um ein umfassendes Verständnis hierzu zu erlangen und gleichzeitig einen passenden Story-Schnitt zu erarbeiten. Die erforderliche Detaillierung und Umfang der Epics und Sagas werden projektspezifisch vereinbart. Die Qualitätskriterien an diese Artefakte werden durch eine spezifische „Definition of Ready (DoR)“ fixiert.

Die einzelnen Epics, Sagas und/oder Themes werden ins Product Backlog integriert und anschließend in detailliertere und sprintkonforme Storys verfeinert (6).

Evolutionsphase „Konzeption“



Storys erstellen

Storys sind der Treibstoff für den agilen Entwicklungsmotor. Sie werden in passender Größe bezüglich der benötigten Funktionalität abgeleitet, verfeinert, geschnitten und justiert (a) und enthalten alle Informationen, die das Umsetzungsteam für das umfassende Verständnis der Aufgabe benötigt. An dieser Stelle wird bewusst auf den gängigen Begriff der User-Story verzichtet, da in mittleren und großen Systemen letztendlich nicht wirklich alle Storys einen konkreten User-Mehrwert liefern. Es gibt viele Storys, die vielleicht erst in ihrer Gesamtheit und auf höherer Ebene tatsächlich etwas mit einem User zu tun haben, aber dennoch für ein funktionierendes System unverzichtbar sind. Das bedeutet nicht, dass eine Story kein konkretes Wertsprechen aufzeigt, aber eben nicht jede Story einen unmittelbaren und für den Kunden spürbaren Mehrwert generiert.

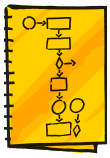
Die Storys werden zudem im Product Backlog geführt, priorisiert und in einer Story Map zu einem Gesamtbild des zu realisierenden Systems orchestriert. Repräsentieren Storys eine umfassendere Komplexität, als diese durch ein grundsätzliches Wertversprechen „Ich als <x> möchte <y> damit <z>“ auszudrücken ist, so ist der RE/BA der maßgebliche Treiber der Story-Konkretisierung, in Vertretung des Product Owners. Die Story wird dann mit ergänzenden Informationen wie beispielsweise einem User-Interface-Design oder Prozessbeschreibungen angereichert. Zusätzliche Inspiration und Detaillierung erfahren Storys in hybriden Vorgehensmodellen, durch konzeptionelle Artefakte des Lösungsdesigns (7), die von der Story referenziert werden.

Lösung designen

In hybriden Vorgehensszenarien erfolgt das Lösungsdesign oftmals parallel oder kurz vor der Erstellung der Storys. Domänenmodelle, Prozessabläufe, Use Cases, Business Rules etc. werden hierbei entwickelt, um die fachliche Architektur zu formulieren.

Ein Lösungsdesign konkretisiert und formt die fachliche Struktur des Systems, wohingegen die Storys eine Ausrichtung in Bezug auf Kundennutzen und Team Velocity haben. Eine bidirektionale Beziehung zwischen Storys und Lösungsdesign zu erreichen, ist aufgrund der divergierenden Perspektive nicht trivial, birgt aber einen erheblichen Mehrwert. Konzepte wie Use-Case-Slices bieten hier gegebenenfalls Teillösungen, jedoch besteht der eigentliche Vorteil nicht in der 100-prozentigen Abbildung einer Story auf ihre Konzeptionsausschnitte. Das Potenzial liegt vielmehr in der bedarfsorientierten Nutzung der mächtigen Dokumentationsarten im Lösungsdesign (wie beispielsweise UML oder SysML). Die Kunst ist hierbei, eine angemessene Balance zwischen Detailgrad und Umfänglichkeit der möglichen Ergebnistypen und dem tatsächlichen Informationsbedarf zu erreichen. Die erstellten Ergebnisse können dann über die Story (7) referenziert werden.

Die Konzeption und Beschreibung ganzheitlicher Prozesse, die Definition des statischen und dynamischen Verhaltens des Systems, seine visuelle Präsentation und die Abstimmung der Schnittstellen zur Systemumwelt sind die Kernaufgaben, die ein RA/BA hier zu leisten hat. Operative Abhängigkeiten zur Realisierung bezüglich Größe und Schnitt werden dabei nicht berücksichtigt. Nicht zuletzt haben die Art des Systems und die damit verbundene Branche (Medizin, Luftfahrt, Finanzdienstleistung etc.) Einfluss auf den benötigten Detailgrad und Umfang des Designanteils, um gegebenenfalls rechtlichen oder regulatorischen Vorgaben gerecht zu werden (8).



Lösung dokumentieren

Die konsequente Fortschreibung des Lösungsdesigns [to-be] (8) durch den RE/BA führt im optimalen Fall zu einer finalen Systemdokumentation [as-is]. Sie beschreibt die fachlichen Funktionen der Lösung in knapper und präziser Form, manchmal unterstützt durch Modellierungen oder Grafiken, um Prozesse, Objekte und Abhängigkeiten möglichst kompakt, verständlich und widerspruchsfrei zu erklären. Im Gegensatz zu traditionellen Vorgehensmodellen mit ihren oftmals prosaischen und üppigen Dokumentationen wird diszipliniert und minimalistisch dokumentiert. Typischerweise nutzen moderne Projekte Kollaborationsplattformationen für eine lebende Dokumentation, die eine verbesserte Verlinkung und Navigation der Inhalte ermöglichen, um damit auch Redundanzen in den Beschreibungen zu minimieren und zugleich immer aktuell zu sein.

Dokumentation kann aber auch aus den Storys abgeleitet (8) werden, um vor allem neue Erkenntnisse aus der Sprint-Realisierung festzuhalten. Unabhängig davon, ob ein paralleles Lösungsdesign als konsolidierte Dokumentation fortgeschrieben wird oder lediglich die Epics, Storys in Verbindung mit dem Programmcode als Systeminformation vorliegen – der Bedarf für eine fachliche Beschreibung wächst mit dem Umfang sowie der Komplexität, Kritikalität, Variantenanzahl und Lebensdauer der Lösung.

Evolutionsphase „Realisierung“

Tests spezifizieren

Sind die formulierten Akzeptanzkriterien in den Storys nicht ausreichend, um Progressions- und Regressionstests effizient abzuleiten, so müssen Testszenarien und Testfälle spezifiziert werden. Primär stellen die Akzeptanzkriterien der Storys (9) die Basis für die Tests dar, um in erster Linie eine Abnahme der jeweiligen Story zu erreichen. Akzeptanzkriterien der Epics oder Sagas dienen gegebenenfalls als Vorlage für umfassendere Progressions- und Regressionstestfälle.

Aber auch Use Cases und Business Rules, als Artefakte des Lösungsdesigns (9), können als Quelle für entsprechende Testfälle herangezogen werden. Im Idealfall sogar mit einem hohen Automatisierungsgrad, indem die Normal- und Alternative-Flows der parametrisierten Use Cases den potenziellen Testdurchlauf definieren und möglicherweise mit Testdaten anreichern.

Testmanager, Tester und RE/BA arbeiten hier Hand in Hand mit den Entwicklern, um eine adäquate Testabdeckung der Ziellösung zu erreichen.

Storys vorstellen

Die intensivste Phase der Interaktion zwischen „Agiler Business-Analyse“ und „Agiler Softwareentwicklung“ erfolgt im Rahmen der Vorstellung, Verhandlung und Abnahme der Storys mit dem Umsetzungsteam. Storys werden bereits vorab mit dem Team oder dessen technischem Lead bezüglich Reifegrad, Machbarkeit, Komplexitätsbewertung und Umsetzbarkeit diskutiert und Verbesserungen iterativ wieder in die Storys zurückgeführt (b). Der vereinbarte Reifegrad der Storys bezüglich der „Definition of Ready“ (DoR) muss vorab erreicht werden, damit eine Story für einen Sprint überhaupt nominiert werden kann.

Im Sprint Planning werden die vorab plausibilisierten Storys dem Team offiziell vorgestellt (10), entweder durch den Product Owner selbst oder dem verantwortlichen RE/BA der Story. Rückmeldungen des Teams bezüglich Größe, Comittment etc. werden wiederum in die Storys eingearbeitet. Idealerweise sind die vorgestellten Storys bereits vorab soweit „refined“, dass deren Vorstellung im übergreifenden Sprint Planning keine großen Überraschungen mehr hervorruft.

Realisierung unterstützen

Die Realisierung der Ziellösung ist durch die agilen Vorgehensprinzipien (Agile Softwareentwicklung) definiert und innerhalb des hier skizzierten Lebenszyklus nur ein Platzhalter und RE-Anknüpfpunkt für die verschiedenen Vorgehensmodelle wie Scrum, SaFe, Less etc. Innerhalb dessen wirkt das Umsetzungsteam mit all seinen Rollen, Regelterminen und Ergebnistypen. Wichtig im Zusammenhang mit den Aufgaben der RE/BA ist die Bereitstellung der und Hilfestellung zu den Artefakten (11) für das Umsetzungsteam und die Unterstützung im Falle von Rückfragen oder bei zusätzlichem Klärungsbedarf.

Erkenntnisse aus der Realisierung fließen während oder auch nach Sprintende wieder zurück in die verschiedenen RE-Aufgabenpakete und führen gegebenenfalls zu neuen oder veränderten Storys (b). Die Systemdokumentation muss anhand der konkreten Realisierung aktualisiert werden, und Tests bedürfen gegebenenfalls einer erneuten Überarbeitung, wenn die Implementierung vom geplanten Vorgehen abweicht.

Als wesentlichen Ergebnistyp liefert der Sprint ein oder mehrere Produktinkremente, gemäß den umgesetzten Storys, die für eine Abnahme dem Product Owner und weiteren Stakeholder zur Verfügung gestellt werden (12).

Storys reviewen

Am Sprintende werden die erreichten Ergebnisse im Review dem Product Owner zur gemeinsamen Überprüfung und Abnahme präsentiert. Vertretend kann auch in diesem Fall der verantwortliche RE/BA eine Vorabprüfung anhand der Akzeptanzkriterien durchführen (12), um eine Empfehlung für die Abnahme auszusprechen und das Review – gerade in großen Projekten – auf die wesentlichen Abnahmeelemente zu beschränken. Die erstellten Produktinkremente sowie weitere vereinbarte Ergebnisse zu den umgesetzten Storys werden hinsichtlich der „Definition of Done“ (DoD) überprüft und Folgeaktivitäten vereinbart. Die DoD kann auch Kriterien bezüglich zu erstellender Dokumentation, Programmcodequalität oder Testerreichung beinhalten – sie umfasst den abgestimmten Anforderungskatalog des Product Owners an eine erledigte Story oder ein Epic.

Evolutionsphase „Integration“

Tests und Integration unterstützen

Aufgrund der Detailkenntnisse zu den fachlichen Anforderungen sind RE/BA per se für die Unterstützung im Rahmen von Integrationstests geeignet (13). Das bedeutet jedoch nicht, dass RE/BA diese originären Testaufgaben vollständig übernehmen, die idealerweise anhand strukturierter Testszenarien eine hohe Testabdeckung garantieren. Vielmehr sind RA/BA ideale Sparringspartner, um die oftmals umfangreichen End-to-end-Abhängigkeiten und -Prozesse zu validieren und Testergebnisse zu plausibilisieren.

Gewonnene Erkenntnisse aus den Integrationstests können wiederum in die Testspezifikationen einfließen.

Wie bereits zu Beginn des Artikels gesagt, stellt dieser Lebenszyklus eine beispielhafte Sicht auf die realen Aufgaben eines RE/BA im agilen beziehungsweise hybriden Projektumfeld dar. Pragmatismus in der Ausgestaltung der konkreten Zusammenarbeit mit dem Product Owner und dem Team ist hierbei wichtiger als das dogmatische Verfolgen von idealisierten und vordefinierten agilen Prinzipien. Mit Augenmaß praktiziertes methodisches Vorgehen, ausbalancierte detailbewusste Konzepte sowie geförderte Kreativität und evolutionär wachsende fachliche Architekturen sind Merkmale des heutigen Requirements Engineerings, die agilen Vorgehensweisen nicht im Wege stehen, sondern vielmehr Grundlage für ihren Erfolg sind. Ein allgemeingültiges und garantiertes Erfolgsrezept gibt es auch in der agilen Welt nicht, aber durch die geschickte Einbettung der RE/BA-Expertisen in einen hybriden RE-Lebenszyklus werden das WARUM und WAS (entwickelt werden soll) eines „Agilen Projekts" mit der angemessenen Transparenz, Eindeutigkeit, Nachhaltigkeit und Qualität bestimmt. ■

Center of Competence
Requirements
Engineering

Oliver Lukas
Leiter CoC Requirements Engineering
oliver.lukas@msg-gillardon.de

msgGillardon

SAVE THE DATE

Sparkassenkonferenz 2020

3. bis 4. November 2020
Frankfurt a. M.