2 LeSS

Es gibt zwei Möglichkeiten für die Umsetzung eines [Designs]: Eine Möglichkeit ist es, die Umsetzung so einfach zu gestalten, dass es offensichtlich keine Mängel gibt, die andere Möglichkeit besteht darin, es so kompliziert zu machen, dass es keine offensichtlichen Mängel gibt.

C.A.R. Hoare

Scrum mit einem Team

Scrum ist ein Framework zur Entwicklung mittels empirischer Prozesskontrolle, in dem ein cross-funktionales, selbstgeführtes *Team* ein Produkt iterativ-inkrementell entwickelt.[1] In jedem zeitlich begrenzten (Timebox) *Sprint* wird ein *potenziell auslieferbares Produktinkrement* erstellt und idealerweise ausgeliefert. Ein *Product Owner* (ein einziger!) ist verantwortlich dafür, den Wert des Produkts zu maximieren, die *Einträge* im *Product Backlog* zu priorisieren und basierend auf konstantem Feedback und Lernen adaptiv zu entscheiden, was das Ziel eines jeden Sprints ist. Ein kleines *Team* ist gemeinsam dafür verantwortlich, das Sprint-Ziel zu liefern; es gibt hier keine limitierenden einzelnen Spezialistenrollen. Ein *Scrum Master* vermittelt, warum Scrum eingesetzt wird und wie man wertvolle Arbeit damit erzielen kann. Er coacht den Product Owner, das Team und die Organisation, wie man Scrum anwenden kann, und agiert als Spiegel. Es gibt keinen Projektleiter oder Teamleiter.

Empirische Prozesskontrolle braucht möglichst vollständige *Transparenz*, die durch die kurzen Entwicklungszyklen entsteht und die Begutachtung der auslieferbaren Produktinkremente. Es wird Wert gelegt auf kontinuierliches Lernen, Inspektion und Adaption hinsichtlich des

1. Bitte lesen Sie das Vorwort, um zu verstehen, warum jedes Kapitel mit diesem Abschnitt beginnt, die sich wiederholende, übergeordnete Struktur in jedem Kapitel, die Definition von einigen Schlüsselbegriffen und stilistische Punkte.

Produkts und der Art und Weise, wie es erstellt worden ist. Scrum basiert auf dem Verständnis, dass die Entwicklung zu komplex und dynamisch ist für detaillierte, schablonenhafte Prozesskochrezepte, die eher dem Infragestellen, dem Engagement und der Verbesserung im Weg stehen.

Im *Scrum Guide* und dem *Scrum Primer* liegt die Betonung auf einem Team; der Fokus liegt nicht auf vielen Teams, die zusammen an einem Produkt arbeiten. Und das führt natürlich dazu, dass man über *Large-Scale* Scrum, also Scrum im Großen, nachdenkt.

LeSS

> LeSS ist Scrum, angewandt auf viele Teams, die gemeinsam an einem Produkt arbeiten.

Siehe Kapitel ***Einführung****, S. 61*

LeSS ist Scrum – Large Scale Scrum (LeSS)[2] ist kein neues und verbessertes Scrum. Und es ist nicht *Scrum auf unterer Ebene für jedes Team mit etwas anderem oben draufgepackt*. Es geht eher darum, herauszufinden, wie man die Prinzipien, den Zweck, die Elemente und die Eleganz von Scrum in einem skalierten Kontext anwendet, und das so einfach wie möglich. Genau wie Scrum und andere, wirklich agile Frameworks ist LeSS bewusst eine »eher unvollständige Methode«, um große Auswirkungen zu erzielen.

> Skaliertes Scrum ist kein spezielles Skalierungsframework, das zufälligerweise Scrum nur auf Teamebene enthält. Echtes skaliertes Scrum ist Scrum skaliert.

Siehe Kapitel ***Organisiere nach Kundennutzen****, S. 87*

... angewandt auf viele Teams
Cross-funktionale, komponentenübergreifende, »Full-Stack«-Feature-Teams von drei bis neun Personen, die auf Lernen fokussiert sind und alles machen – von UX über Code bis hin zu Videos –, um Einträge komplett fertigzustellen und ein auslieferbares Produkt zu erzeugen.

Siehe Kapitel ***Koordination & Integration****, S. 309*

... die zusammenarbeiten
Die Teams arbeiten zusammen, da sie ein gemeinsames Ziel haben: ein gemeinsames, auslieferbares Produkt am Ende eines gemeinsamen Sprints auszuliefern. Und jedes Team kümmert sich darum, da sie Feature-Teams sind, die verantwortlich für das *Ganze* sind, nicht nur für einen Teil.

---

2. LeSS schlägt beides vor: Large-Scale Scrum und Vereinfachung beim Runterskalieren.

... an einem Produkt
Was für ein Produkt? Eine umfassende, komplette, von Anfang bis Ende gedachte kundenzentrierte Lösung, die von echten Kunden benutzt wird. Ein Produkt ist keine Komponente, Plattform, Schicht oder Bibliothek.

Siehe Kapitel ***Produkt****, S. 169*

Vorgeschichte

2002, als Craig *Agile & Iterative Development* geschrieben hat, haben viele geglaubt, dass agile Entwicklung nur für kleine Teams gedacht war. Wie auch immer, wir beide (Craig und Bas) haben angefangen, uns dafür zu interessieren, wie Scrum in großen, verteilten und Offshore-Entwicklungen angewandt werden kann – und wir erhielten auch immer mehr Anfragen hierzu. So kam es, dass wir uns 2005 zusammengetan haben, um mit Kunden daran zu arbeiten, Scrum zu skalieren. Heute sind beide LeSS-Frameworks (LeSS und LeSS Huge) in großen Gruppen weltweit in unterschiedlichen Domänen eingeführt worden:

- Telekommunikationsequipment – Ericsson & Nokia Networks[3]
- Investment- und Handelsbanken – UBS
- Handelssysteme – ION Trading
- Marketingplattformen und Markenanalyse – Vendasta
- Videokonferenzsysteme – Cisco
- Onlinespiele (Wetten) – bwin.party
- Offshore Outsourcing – Valtech India[4]

Was ist in Bezug auf *groß* ein typischer Fall einer LeSS-Einführung? Vielleicht fünf Teams an einem oder zwei Standorten. Wir waren in Einführungen von solcher Größenordnung involviert, in solche mit ein paar Hundert Leuten bis hin zu einem LeSS-Huge-Fall von weit über tausend Personen, viel zu vielen Entwicklungsstandorten, vieler Zehnmillionen Zeilen C++-Code mit kundenspezifischer Hardware.

Weitere Informationen zu LeSS

Um Menschen bei ihrer Weiterentwicklung zu unterstützen, haben wir, basierend auf unseren Erfahrungen mit Kunden, 2008 und 2010 zwei Bücher über die Skalierung von agiler Entwicklung mit dem LeSS-Framework veröffentlicht:

3. Nokia Networks ist nicht das Handy-Unternehmen, das von Microsoft übernommen worden ist.
4. Siehe auch die Fallstudien auf *less.works* für weitere Beispiele.

1. *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum* – erklärt das Denken, die Führung und Veränderungen am organisatorischen Design.
2. *Practices for Scaling Lean & Agile Development: Large, Multi-site & Offshore Product Development with Large-Scale Scrum* – lässt teilhaben an Hunderten konkreten Experimenten für LeSS, basierend auf unserer Erfahrung mit Kunden; Experimente im Produktmanagement, Architektur, Planung, verteilte Umgebungen, Offshore, Verträge und mehr.

Dieses Buch – *Large-Scale Scrum: Scrum erfolgreich skalieren mit LeSS*[5] – ist das dritte in der LeSS-Serie, ein Vorgänger und Grundlagenbuch. Dieses Buch fasst zusammen, erklärt und hebt hervor, was am wichtigsten ist.

Neben diesem Buch findet man online auf *http://less.works* weitere Quellen zum Lernen (inklusive Buchkapitel, Artikel und Videos), Kurse und Coaching.

Experimente, Wegweiser, Regeln und Prinzipien

In den ersten beiden LeSS-Büchern wurde bereits hervorgehoben, dass es *Dinge wie »Best Practices« in der Produktentwicklung nicht gibt. Es gibt nur Praktiken, die in einem bestimmten Kontext passend sind.*

Praktiken sind situativ. Fröhlich zu behaupten, sie seien das »Beste«, trennt sie von der Motivation und dem Kontext, die sie jedoch erst zu dem machen, was sie sind. Ohne das werden sie zu Ritualen. Und sogenannte »Best Practices« hervorzuheben, tötet eine Kultur des Lernens, des Infragestellens, des Engagements und der kontinuierlichen Verbesserung. Warum würden Menschen das Beste *anfechten?*

Daher haben die früheren LeSS-Bücher die Leser an *Experimenten* teilhaben lassen, die wir und unsere Kunden ausprobiert haben. Wir bestärkten – und bestärken nach wir vor – diese Geisteshaltung. Allerdings beobachteten wir mit der Zeit auch zwei Probleme der Nur-Experimente-Geisteshaltung:

- Unerfahrene Gruppen treffen ungeschickte Entscheidungen zu ihrem Nachteil und führen LeSS in einer Art und Weise ein, für die LeSS nicht gedacht war, mit offensichtlichen Problemen. Zum Beispiel gibt es Gruppen, die Requirement Areas mit nur einem Team erstellt haben. Autsch!

5. Originaltitel: Large-Scale Scrum: More with LeSS.

- Unerfahrene Gruppen fragen: »Wo fangen wir an? Was ist das Wichtigste?« Verständlicherweise können sie die wesentlichen Grundlagen nicht sehen.

Aufgrund dieses Feedbacks haben wir nachgedacht und sind noch mal zum *Shu-Ha-Ri*-Modell des Lernens zurückgekommen: *Shu* – folge den Regeln, um die Grundlagen zu erlernen. *Ha* – breche die Regeln, um den Kontext zu erkunden. *Ri* – meisterhaftes Können und finde deinen eigenen Weg. In einer LeSS-Einführung auf Shu-Ebene sind nur wenige *Regeln* nötig für ein *gerade mal ausreichendes* Framework, um empirische Prozesskontrolle und einen ganzheitlichen Produktfokus anzukurbeln.[6] Diese Regeln definieren die beiden LeSS-Frameworks, die schon bald vorgestellt werden.

Um später darauf aufzubauen und um es einmal zusammenzufassen, sind nachfolgend die Bestandteile von LeSS aufgeführt:

- **Regeln**
  Ein paar wenige Regeln genügen, um loszulegen und um das Fundament zu schaffen. Sie definieren die wesentlichen Elemente des LeSS-Frameworks, die vorhanden sein sollten, um empirische Prozesskontrolle und einen ganzheitlichen Produktfokus zu unterstützen, beispielsweise *führe in jedem Sprint eine Gesamtretrospektive durch*.
- **Wegweiser**
  Ein angemessener Satz an Wegweisern hilft dabei, die Regeln effektiv einzuführen, und gilt für eine Untermenge von Experimenten, die es wert sind, ausprobiert zu werden, da sie auf jahrelanger Erfahrung mit LeSS basieren. Wegweiser enthalten *Tipps*. In der Regel sind sie hilfreich und ein Bereich für kontinuierliche Verbesserung, beispielsweise die *Drei Einführungsprinzipien*.
- **Experimente**
  Viele Experimente sind sehr situativ und sind es nicht unbedingt wert, ausprobiert zu werden, beispielsweise der Versuch, bei der Entwicklung von multilingualen Anwendungen den Übersetzer im Scrum-Team zu verorten. Dieser Versuch könnte sich im Verhältnis Kosten zu Nutzen als eher nicht praktikabel herausstellen. Doch es gilt: Probieren geht über Studieren!
- **Prinzipien**
  Im Mittelpunkt steht ein Satz von Prinzipien – herausgezogen aus der Erfahrung mit LeSS-Einführungen –, die die Regeln, Wegweiser und Experimente prägen, beispielsweise *ganzheitlicher Produktfokus*.

---

6. Scrum hat ebenfalls nur wenige Regeln für sein Framework, aus den gleichen Gründen wie LeSS.

Die LeSS-Wegweiser und Experimente sind optional. Die Wegweiser werden vermutlich hilfreich sein und wir empfehlen, diese auszuprobieren. Solche, die nicht passen oder die weitere Verbesserung eher einschränken, können Sie übergehen oder weglassen.

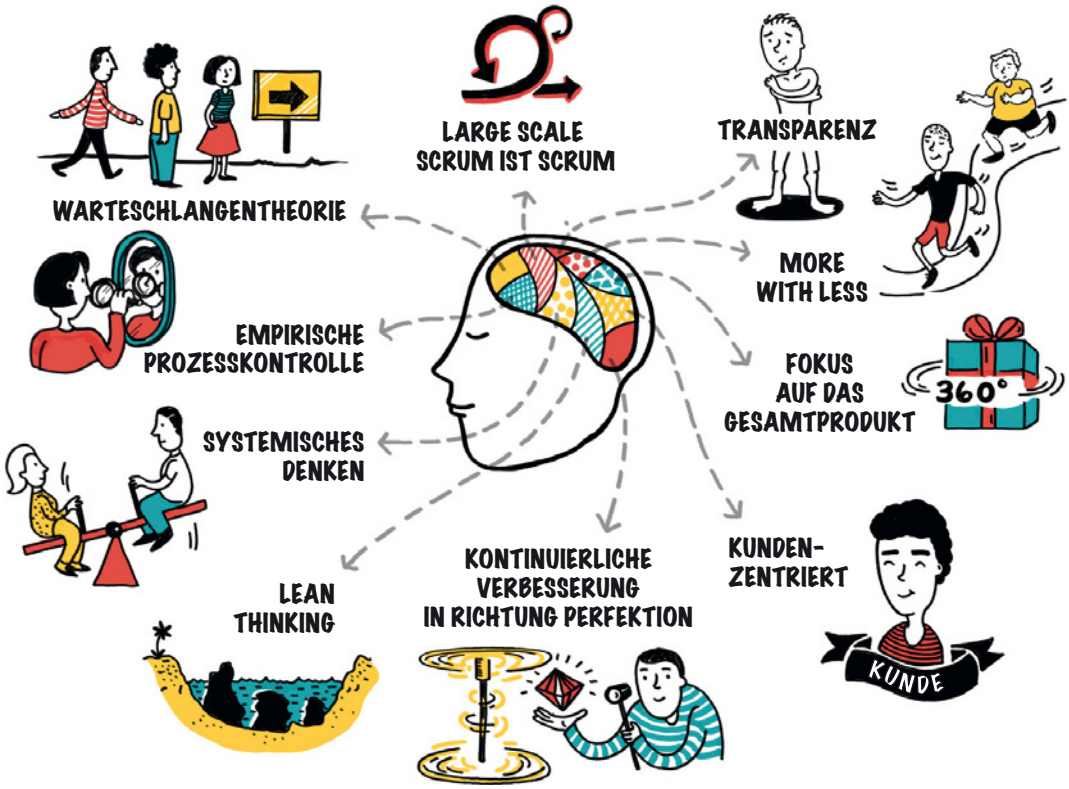
Ein guter Weg, LeSS zu betrachten, ist im *LeSS-Gesamtbild* visualisiert:

Das LeSS-Gesamtbild gibt den Weg vor, wie wir LeSS einführen:

1. LeSS-Prinzipien – kommen als Nächstes
2. LeSS-Frameworks (definiert durch die Regeln) – im Rest des Kapitels
3. LeSS-Wegweiser – in den nachfolgenden Kapiteln dieses Buches
4. LeSS-Experimente – stehen schon mit den ersten beiden LeSS-Büchern zur Verfügung

LeSS-Prinzipien

Die LeSS-Regeln definieren das LeSS-Framework. Diese Regeln sind jedoch minimalistisch und beantworten nicht die Frage, wie LeSS in einem spezifischen Kontext angewendet wird. Die LeSS-*Prinzipien* liefern die Basis für solche Entscheidungen.



Large-Scale Scrum ist Scrum
Es ist kein neues und verbessertes Scrum. Viel eher geht es in LeSS darum, herauszufinden, wie die Prinzipien, Regeln, Elemente und der Zweck von Scrum in einem großen, skalierten Kontext angewendet werden können, und zwar so einfach wie möglich.

Transparenz
Basiert auf greifbaren »fertigen« Einträgen, kurzen Zyklen, Zusammenarbeit, gemeinsamen Definitionen und dem Vertreiben von Angst am Arbeitsplatz.

More with less (mehr durch weniger)
Wir wollen nicht noch mehr Regeln aufstellen, da mehr Rollen zu weniger Verantwortlichkeit der Teams führen. Wir wollen nicht noch mehr Artefakte, da mehr Artefakte zu einer größeren Distanz zwischen Teams und Kunden führen. Wir wollen nicht noch mehr Prozess, da dies zu weniger Lernen und Teamhoheit über den Prozess führt. Statt dessen wollen wir mehr verantwortliche Teams durch weniger (LeSS-) Regeln. Wir wollen mehr kundenorientierte Teams, die sinnvolle Produkte bauen, indem sie weniger Artefakte haben. Wir wollen mehr

Hoheit der Teams über den Prozess und mehr sinnhafte Arbeit durch einen weniger definierten Prozess. *Wir wollen mehr durch weniger – »More with LeSS«.*

Ganzheitlicher Produktfokus
Ein Product Backlog, ein Product Owner, ein auslieferbares Produkt, ein Sprint – egal, ob drei Teams oder 33 Teams. Kunden wollen wertvolle Funktionalität in einem zusammenhängenden Produkt, keine technischen Komponenten in einzelnen Teilen.

Kundenzentriert
Die Konzentration liegt darauf, die wirklichen Kundenprobleme zu verstehen und zu lösen. Identifizieren Sie Wert und Nutzlosigkeit aus Sicht des bezahlenden Kunden. Reduzieren Sie die Wartezeit aus seiner Perspektive. Erhöhen und verstärken Sie Feedbackschleifen mit echten Kunden. Jeder muss verstehen, wie sich seine tägliche Arbeit direkt auf den zahlenden Kunden auswirkt und wie dieser davon profitiert.

Kontinuierliche Verbesserung in Richtung Perfektion
Hier ist ein Perfektionsziel: Erstellen und liefern Sie ein Produkt möglichst jederzeit ohne zusätzliche Kosten und Fehler, das Kunden erfreut, zur Verbesserung der Umwelt beiträgt und das Leben lebenswerter macht. Seien Sie immer bescheiden und führen Sie radikale Verbesserungsexperimente durch, bis dieses Ziel erreicht ist.

Lean Thinking
Bauen Sie ein organisatorisches System auf, dessen Fundament aus Managern als Lehrer besteht, die Lean Thinking anwenden und vermitteln, die führen, um zu verbessern, die »Stop & Fix« fördern und die »Go & See«[7] praktizieren. Fügen Sie die beiden Säulen, Respekt für den Menschen und Geisteshaltung hin zu einer Verbesserung durch kontinuierliches Infragestellen des Status quo, hinzu – alles zur Erreichung der Perfektion.

Systemisches Denken
Betrachten, verstehen und optimieren Sie das gesamte System[8] (und nicht einzelne Teile davon) und verwenden Sie die Systemmodellierung, um Systemdynamiken zu erforschen. Vermeiden Sie lokale Teiloptimierungen durch Konzentration auf die Effizienz oder Produktivität von einzelnen Individuen oder Teams. Kunden interessieren sich für die Dauer und den Fluss des gesamten Zyklus »vom Konzept zu Cash«,

7. Siehe Wegweiser: Go & See, S. 137.
8. Das System besteht aus allem und jedem, vom Konzept bis zum Geld, und allen Dynamiken in Bezug auf Zeit und Raum, insbesondere aus der Perspektive von Kunden und Nutzern.

nicht für individuelle Schritte, und *lokale* Optimierungen führen meist zur einer Verschlechterung des *Gesamten*.

Empirische Prozesskontrolle
Das bedeutet kontinuierliches Inspizieren und Adaptieren des Produkts, der Prozesse, der Verhaltensweisen, des organisatorischen Designs und der Praktiken, sich in situativ angemessenen Wegen zu entwickeln. Tun Sie dies, statt einem vorgeschriebenen Satz sogenannter Best Practices zu folgen, die den Kontext ignorieren, zu einem ritualisierten Befolgen führen, Veränderungen und Lernen behindern und den Sinn der Menschen für Engagement und Eigentum zermalmen.

Warteschlangentheorie
Verstehen Sie, wie Systeme mit Warteschlagen sich in einem Forschungs- & Entwicklungsumfeld verhalten, und setzen Sie diese Erkenntnisse ein, um die Größen von Warteschlangen, Work-in-Progress(WIP)-Limitierungen, Multitasking, Arbeitspakete und Varianz zu managen.

Zwei Frameworks: LeSS & LeSS Huge

Large-Scale Scrum hat zwei Frameworks:

- **LeSS**
  zwei bis acht Teams
- **LeSS Huge**
  mehr als acht Teams

Das Wort *LeSS* bedeutet zweierlei: Es meint sowohl Large-Scale Scrum im Allgemeinen als auch das kleinere LeSS-Framework.

Die magische Zahl Acht

Eigentlich ist die *Acht* keine magische Zahl, und wenn Ihre Gruppe das kleinere LeSS-Framework mit mehr als acht Teams erfolgreich anwenden kann, wunderbar! Aber das haben wir nicht erlebt ... noch nicht. Diese Obergrenze hat sich einfach nur durch empirische Beobachtung ergeben. Und in einigen Fällen, z.B. unterschiedlich komplexen Zielen in verteilten Umgebungen mit unerfahrenen Teams, die jeweils nur ihre eigene Sprache sprechen, könnte es sein, dass diese Grenze bei weniger als acht liegt.

Es gibt auf jeden Fall einen Zeitpunkt,

- an dem ein einziger Product Owner nicht mehr länger den Überblick über das gesamte Produkt erfassen kann,
- der Product Owner nicht mehr den Ausgleich zwischen externem und internem Fokus schaffen kann und
- das Product Backlog so groß ist, dass es für eine Person schwierig wird, damit zu arbeiten.

Trifft die Gruppe auf diesen Wendepunkt, könnte es Zeit werden für den Wechsel vom kleineren LeSS-Framework zum LeSS-Huge-Framework. Auf der anderen Seite schlagen wir vor, erst einmal zu versuchen, besser, kleiner und einfacher zu werden, bevor man *größer* wird.

Gemeinsamkeiten beider Frameworks

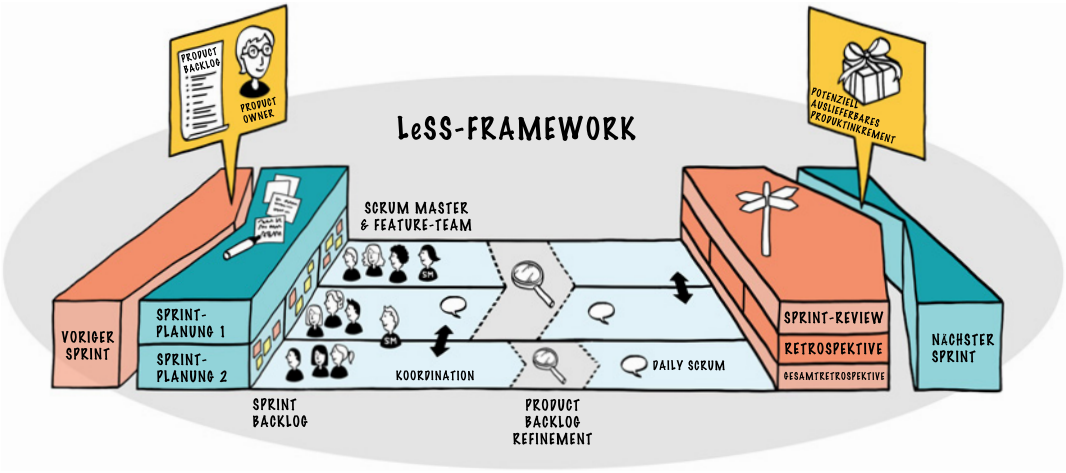
Das LeSS- und das LeSS-Huge-Framework weisen gemeinsame Elemente auf:

- Ein Product Owner und ein Product Backlog
- Ein gemeinsamer Sprint über alle Teams
- Ein auslieferbares Produktinkrement

Die folgenden zwei Abschnitte dieses Kapitels gehen mehr ins Detail der beiden Frameworks. Zunächst wird das kleinere LeSS-Framework erläutert, anschließend folgt LeSS Huge auf Seite 39.

LeSS-Framework

LeSS-Framework – Zusammenfassung



Das kleinere LeSS-Framework ist ausgelegt für einen (und nur für einen) Product Owner, dem das Produkt gehört und der ein Product Backlog verwaltet, an dem Teams in gemeinsamen Sprints daran arbeiten, das Gesamtprodukt zu verbessern. Die Elemente des LeSS-Frameworks sind ungefähr die gleichen wie bei Scrum mit einem Team:

Rollen
Ein *Product Owner*, zwei bis acht *Teams*, ein *Scrum Master* für ein bis drei Teams. Es ist entscheidend, dass diese Teams Feature-Teams sind – echte cross-funktionale, komponentenübergreifende »Full-Stack«-Teams, die zusammen an einer gemeinsamen Codebasis arbeiten, wo jedes Team *alles* macht, um *fertige* Einträge zu erstellen.

Artefakte
Ein potenziell auslieferbares Produktinkrement, ein Product Backlog und ein eigenes Sprint Backlog für jedes Team.

Events
Ein gemeinsamer Sprint für das gesamte Produkt. Dieser inkludiert alle Teams und endet in einem potenziell auslieferbaren Produktinkrement. Details hierzu kommen in den anstehenden Geschichten und in eigenen Kapiteln zur Sprache.

Regeln & Wegweiser
Regeln für ein gerade mal ausreichendes Skalierungsframework für empirische Prozesskontrolle und ganzheitlichen Produktfokus. Wegweiser *könnten* helfen.

LeSS-Geschichten

LeSS lernen
Ein Weg, etwas zu lernen, besteht darin, tiefgehende Erklärungen zu lesen, und Leser, die das bevorzugen, können problemlos alles bis zur Vorstellung des *LeSS-Huge*-Frameworks (S. 39) überspringen und von dort aus weiterlesen. Andere, die Geschichten mögen, können hier weitermachen.

Einfache Geschichten
Diese Geschichten handeln nicht von den Komplexitäten großer, skalierter Entwicklungen – von der Politik bis zur Priorisierung –, die wir durch Beratung erfahren. Wir widmen uns in späteren Kapiteln diesen Dingen. An dieser Stelle haben wir bewusst klare und einfache Geschichten ausgewählt, um nur die Grundlagen eines LeSS-Sprints vorzustellen. Wer auf der Suche nach einem aufregenden Dialog oder einem Drama ist, der sollte ein *Lean*-Buch lesen.

Regeln & Wegweiser
Bei den Geschichten wird Ihnen auffallen, dass die Randnotizen auf die dazugehörigen LeSS-Regeln und Wegweiser referenzieren, um für mehr Klarheit zu sorgen und Verbindungen herzustellen.

Zwei Perspektiven
Nachfolgend sind zwei zusammenhängende Geschichten mit zwei grundlegenden Perspektiven aufgeführt, um einige Abläufe einfacher vorzustellen:

1. Der Ablauf von *Teams* innerhalb eines LeSS-Sprints
2. Der Fluss von *kundenzentrierten Einträgen (Features)*

LeSS-Geschichte: Der Ablauf von Teams

Diese Geschichte konzentriert sich auf den Ablauf von *Teams* in einem Sprint und weniger auf den Fluss von *Arbeitspaketen*. In der Realität verbringt man den größten Teil der Zeit in einem Sprint mit entwicklungsbezogenen Aufgaben und weniger in *Meetings*. *Wie auch immer, diese Geschichte betont die Meetings und die Interaktionen, da es das Ziel dieser Geschichte ist, zu verstehen, wie mehrere Teams in den LeSS-Events zusammenarbeiten und wie sie sich tagtäglich koordinieren.*

Mark betritt den Raum, in dem sein Team (»Handel«) arbeitet, und sieht Mira[9]. »Guten Morgen!«, sagt sie, »nur zur Erinnerung: Wir sind die Stellvertreter unseres Teams in diesem Sprint und Sprint-Planung 1 beginnt in 10 Minuten.« »Richtig«, antwortet Mark, »Ich treffe dich gleich im großen Raum.«

Tipp
Wechsle die Stellvertreter in jedem Sprint.

Sprint-Planung 1

(Wegweiser: Sprint-Planung 1, S. 298)

Es ist Zeit für die gemeinsame Sprint-Planung 1. In einem großen Raum haben sich die zehn Stellvertreter von den fünf Teams aus dieser Produktgruppe verteilt. Alle arbeiten am Flagschiffprodukt zum Handel mit Anleihen und Derivaten. Sam, der Scrum Master der Teams »Handel« und »Marge«, ist ebenfalls anwesend. Er plant, das Event zu beobachten und ggf. zu coachen, sollte das nötig werden.

Regel
Es gibt einen produktübergreifenden Sprint und nicht verschiedene Sprints für jedes Team.

Früher, vor vielen Sprints war jedes Teammitglied aus allen Teams bei der Sprint-Planung 1 dabei. Das war zu dem damaligen Zeitpunkt auch wesentlich nützlicher, da die Gruppe nicht sehr gut darin gewesen war, Arbeitspakete aus dem Product Backlog klar und »fertig« zu bekommen oder überhaupt ein tieferes Wissen in den verschiedenen Teams entstehen zu lassen. In jenen Tagen wurde die Sprint-Planung 1 dazu verwendet, Antworten auf viele der wesentlichen Fragen zu liefern, die auch jeder hören musste. Die Sprint-Planung 1 wurde immer weiter verbessert, und verglichen mit damals wurde bis heute eine deutliche Verbesserung erzielt. Die Gruppen experimentieren mit dem Entsenden von wechselnden Stellvertretern, was dazu führte, dass die Sprint-Planung 1 ein einfaches und kurzes Meeting wurde, in dem nur ein paar unwesentliche Fragen geklärt werden, die immer mal wieder auftauchen. Sollte dieser neue Ansatz nicht so gut funktionieren, wird das mit Sicherheit in einer der Gesamtretrospektiven thematisiert werden, woraufhin dann ein neues Experiment zur Verbesserung der Sprint-Planung angegangen wird.

Regel
Die Sprint-Planung besteht aus zwei Teilen: Sprint-Planung 1 ist für alle Teams gemeinsam, wohingegen Sprint-Planung 2 typischerweise jedes Team für sich macht. Wenn man mehrere Teams hat, die an eng zusammenhängenden Dingen arbeiten, führt man mit diesen Teams eine gemeinsame Sprint-Planung 2 in einem gemeinsamen Raum durch.

9. Um es Ihnen als Leser einfacher zu machen, sich an Charaktere und Rollen zu erinnern, verwenden die Namen eine Alliteration. Zum Beispiel ist Mira ein Mitglied des Entwicklungsteams, Sam ist ein Scrum Master und Paolo ein Product Owner.

Regel

An der Sprint-Planung 1 nehmen der Product Owner, die Teams oder deren Stellvertreter teil. Gemeinsam werden sie probeweise die Dinge auswählen, an denen jedes Team im kommenden Sprint arbeiten wird.

Paolo betritt den Raum und begrüßt alle mit einem »Hi!«. Er ist der Product Owner und der leitende Produktmanager.[10] Paolo verteilt 22 Karten auf einem Tisch und sagt: »Hier sind die großen Themen: der Deutsche Markt, Auftragsverwaltung und ein paar regulatorische Berichte. Ich habe sie entsprechend meiner Priorisierung geordnet und ausgelegt. Ich denke, jeder hier versteht, warum die Prioritäten so sind, da wir das häufig genug in den Product Backlog Refinements diskutiert haben. Sollte jedoch etwas unklar sein, fragt bitte noch mal.«

Tipp

Teams wählen ihre Arbeitspakete selbst aus.

Mira und Mark gehen gemeinsam mit den anderen Stellvertretern um den Tisch herum und nehmen sich zwei Karten heraus, die mit Dingen zu tun haben, die den deutschen Anleihenmarkt betreffen. In den letzten beiden Sprints hat ihr Team diese Dinge durch Workshops zu Product Backlog Refinement (PBR) in Einzelteams im Detail geklärt.

Wegweiser

Multiteam-PBR, S. 273

Sie nehmen auch noch zwei weitere Karten, die mit der Auftragsverwaltung zu tun haben und die sowohl das Team »Handel« als auch das Team »Marge« relativ gut verstehen. Beide Teams haben in Multiteam-PBR-Workshops an diesen Einträgen zusammengearbeitet. Warum? Die Teams wollten so spät wie möglich in einer zukünftigen Sprint-Planung entscheiden, welches Team an welchem Eintrag arbeitet. Das erhöht die Agilität der Gruppen, einfach auf Veränderungen zu reagieren, und das tiefere Gesamtproduktwissen begünstigt selbstorganisierte Koordination.

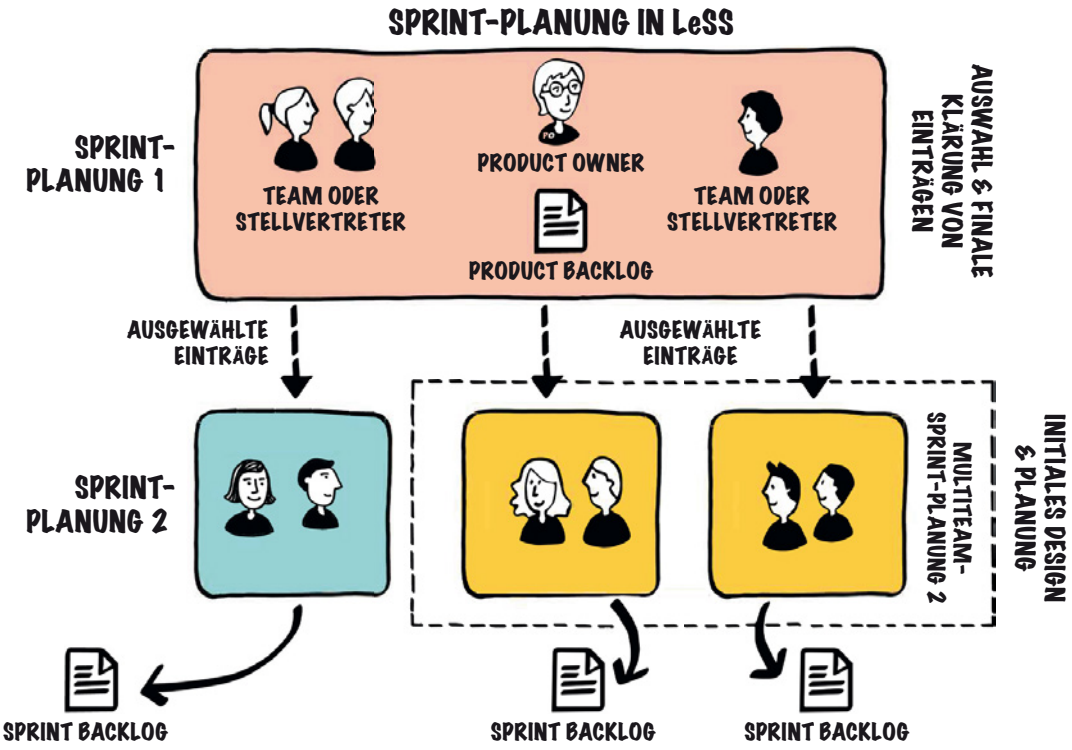
Tipp

Treffe keine Vorentscheidungen, welches Team an welchen Einträgen arbeiten soll.

Eine Minute später überfliegt Mary von Team »Marge« die Karten eines anderen Teams und fragt deren Stellvertreter: »Würde es euch etwas ausmachen, wenn wir uns um diesen Bericht kümmern? Wir haben etwas sehr Ähnliches im letzten Sprint gemacht, und ich könnte wetten, dass wir das hier schnell fertig bekommen. Könnten wir das gegen diesen Eintrag hier für den deutschen Markt tauschen?« Sie stimmen zu.

---

10. In Firmen, die ein Produkt herstellen, konzentrieren sich die Rollen des Produktmanagements oder auch des Produktmarketings auf die Vision und die Ausrichtung, regen Innovationen an, analysieren Konkurrenten und erforschen Kunden und die Bedürfnisse des Markts sowie Trends. In internen Entwicklungsabteilungen könnte diese Rolle durch einen Hauptnutzer ausgefüllt werden, der in einem operativen Geschäftsteil des Unternehmens sitzt. Der Product Owner – derjenige, dem das Produkt »gehört« – kommt in Scrum und in LeSS üblicherweise aus einer dieser Rollen, so wie Paolo beispielsweise, der leitende Produktmanager, der als Product Owner dient (siehe auch das Kapitel Product Owner für weitere Informationen).



Nach ein paar Minuten sind die Teams fertig mit der Auswahl und dem Tauschen gemäß ihren Interessen, Stärken und Vorlieben bezogen auf den Gruppenschwerpunkt.

Sam (der Scrum Master) sagt: »Mir ist aufgefallen, dass Team ›Marge‹ die vier wichtigsten Einträge ausgewählt hat. Könnte das zu einem Problem werden?« Es folgt eine kurze Diskussion, in der die Gruppe realisiert, dass die Möglichkeit besteht, dass einer der vier Einträge nicht fertig werden könnte, wenn die Sachen für Team »Marge« nicht reibungslos laufen. Sie entscheiden sich, ein paar dieser hoch priorisierten Einträge über mehrere Teams zu verteilen (abhängig davon, welches Team welchen Eintrag kennt), was es deutlich wahrscheinlicher macht, dass alle wichtigsten Arbeiten auch fertiggestellt werden können.

Wegweiser
Fünf Scrum-Master-Werkzeuge, S. 153

Tipp
Verteile wichtige Dinge.

Die Stellvertreter haben insgesamt 18 Karten ausgewählt. Die unwichtigsten vier Karten bleiben auf dem Tisch liegen. Paolo betrachtet die vier übriggebliebenen Karten, nimmt zwei davon in die Hand und meint: »Diese beiden sind ziemlich wichtig für mich in diesem Sprint. Vielleicht hätte ich ihnen zu Anfang eine höhere Priorität zukommen lassen sollen, was ich nicht habe, und nun würde ich gern meine Meinung ändern. Lasst uns einen Weg finden, diese beiden Kar-

ten mit zweien zu tauschen, die ihr schon ausgewählt habt. Und sollte ein Team Glück haben und vorzeitig fertig werden, kann es sich natürlich die liegengebliebenen Dinge ziehen.«

Regel
Die Teams identifizieren Möglichkeiten der Zusammenarbeit, und finale Fragen sind geklärt.

Nachdem das gelöst ist, sagt Paolo: »Okay, lasst uns ein wenig Zeit nehmen, um noch offene Fragen zu klären. Wie ihr wisst, konzentriere ich mich eher auf die Priorisierung und die meisten von euch kennen die Details dieser Einträge wesentlicher besser als ich, aber schauen wir mal, welche kleineren Punkte wir noch gemeinsam klären können.«

Tipp
Diskutiere abweichende Meinungen, um zu klären.

Gleichzeitig denken Mira, Mark und die anderen angestrengt darüber nach, welche Punkte noch zur Klärung für ihre Teams offen sind, und schreiben einige Fragen auf Flipchart-Papiere, die auf allen Wänden im Raum verteilt sind. Paolo geht von einem Bereich zum anderen und diskutiert mit den anderen. Alle verteilen sich und arbeiten mit. Nach ungefähr 30 Minuten sind alle Fragen, die zur Klärung anstanden, beantwortet.

Die Gruppe bildet stehenderweise einen Kreis, um alles zusammenzufassen. Niemand bringt Koordinationsthemen zur Sprache. Daraufhin bemerkt Sam: »Mir ist aufgefallen, dass die Teams ›Handel‹, ›Marge‹ und ›Nicht-Derivate‹ Einträge bearbeiten, die sehr stark miteinander zusammenhängen.« Mira sagt: »Hey, lasst unsere Teams ›Handel‹, ›Marge‹ und ›Nicht-Derivate‹ zu einer gemeinsamen Sprint-Planung 2 zusammenkommen. So haben wir die Gelegenheit, zusammenzuarbeiten.« Dem stimmen alle zu und das Meeting ist beendet.

Team- und Multiteam-Sprint-Planung 2

*(**Wegweiser**: Multiteam-Sprint-Planung 2, S. 302)*

Regel
Jedes Team hat sein eigenes Sprint Backlog.

Nach einer Pause gehen zwei der fünf Teams in ihr eigenes Sprint-Planung-2-Meeting (SP2), um ihre eigenen Sprint Backlogs zu erstellen und damit ihre Arbeit für den Sprint zusammenzustellen und zu planen.

Regel
Führe ein Multiteam-SP2 in einem gemeinsamen Raum durch, wenn die Teams an eng zusammenhängenden Einträgen arbeiten.

Im Gegensatz dazu gehen die Teams »Handel«, »Marge« und »Nicht-Derivate« gemeinsam zu einer Multiteam-Sprint-Planung 2 in einen großen Raum, da sie stark zusammenhängende Inhalte implementieren, die zuvor gemeinsam in einem Multiteam-PBR geklärt worden sind, und sie einen Mehrwert darin erkennen, eng zusammenzuarbeiten.

Tipp
Eine Sitzung für die ganze Gruppe für Design und gemeinsame Arbeit

Sie reden zusammen in einer 10-minütigen Sitzung, um durch die Identifizierung der gemeinsamen Arbeit (gemeinsamen Aufgaben) und der Designthemen den Weg für die nächsten Schritte zu bereiten. Dazu starten sie eine 30-Minuten-Timebox zum Thema Design und einigen sich darauf, möglichst viel zu *visualisieren*: mehr Skizzen auf dem Whiteboard und weniger Reden ohne Zeichnen. Während dieser Zeit

entdecken die Teams noch mehr gemeinsame Arbeit und schreiben dies ebenfalls aufs Whiteboard.

Dong! Nach 30 Minuten bleibt noch eine Menge ungeklärter Details bestehen, aber die Teams machen trotzdem weiter mit dem nächsten Schritt. Jedes Team geht in eine andere Ecke des Raumes und beginnt konzentriert mit seiner eigenen Sprint-Planung 2, in der mehr über Designthemen im Detail gesprochen wird und das Sprint Backlog mit Karten erstellt wird. Weitere Koordinierung wird durch eine fortgeschrittene Variante der *Just-talk*-Technik in LeSS adressiert: *Just scream*.

Wegweiser
Keine Softwarewerkzeuge für das Sprint Backlog, S. 304

Während des Gesprächs kommen die Teams zu dem Schluss, dass ein gründlicherer Multiteam-Design-Workshop nötig ist. Sie vereinbaren, sich im Laufe des Tages dazu noch einmal zu treffen.

Wegweiser
Einfach reden, S. 311

Multiteam-Design-Workshop

*(**Wegweiser**: Multiteam-Design-Workshop, S. 325)*

Nach der Sprint-Planung und einer weiteren Pause gehen Mira und Mark vom Team »Handel« und ein paar Leute aus dem Team »Marge« und Team »Nicht-Derivate« in einen Multiteam-Design-Workshop mit einer Timebox von einer Stunde, um tiefer in ein gemeinsames und konsistentes Design für ihre Arbeit einzutauchen. Auf einem großen Whiteboard skizzieren sie ihre Überlegungen und unterhalten sich, um mehr Klarheit und Einigung hinsichtlich des Designansatzes der gemeinsamen technischen Aufgaben zu bekommen. Glücklicherweise haben die Ergebnisse keine ernsthaften Auswirkungen auf die bestehenden Sprint-Pläne. Allerdings fühlen sie sich mit ihrem Prozess nicht sehr wohl, da sie den Klärungsbedarf hinsichtlich solch großer Designfragen früher hätten feststellen können.

Entwicklungsaktivitäten, die die Koordinierung und kontinuierliche Auslieferung unterstützen

Nach der Sprint-Planung tauchen die Teams in die Entwicklung ein mit dem Schwerpunkt, mittels *Code zu kommunizieren*. Alle Teams *integrieren kontinuierlich*. Die kontinuierliche Integration des ganzen Codes aller Teams sorgt für die Möglichkeit, zu kooperieren, indem überprüft wird, wer noch Änderungen an der Komponente vorgenommen hat, an der gearbeitet wurde. Das ist hilfreich, da die Gruppe das Integrieren als Weg benutzt, andere zu informieren, und als Unterstützung zur Koordination.

Wegweiser
Kommuniziere mittels Code, S. 316

Wegweiser
Integriere kontinuierlich, S. 317

Regel
Bevorzuge dezentrale und informelle Koordination vor zentralisierter Koordination

Wegweiser
Einfach reden, S. 311

Beispielsweise zieht Mark, seines Zeichens Entwickler im Team »Handel«, früh am Morgen des zweiten Tages des Sprints die letzte Version lokal auf seinen Rechner und überfliegt schnell die letzten Änderungen, die im Zusammenhang mit der Komponente stehen, an der er gerade arbeitet. Er entdeckt dabei Änderungen, die mit Code zusammenhängen, den Maximilian vom Team »Marge« hinzugefügt hat. Mark weiß, dass Team »Marge« an etwas arbeitet, was sehr stark mit seiner Komponente zusammenhängt, sodass ihn die Änderungen von Maximilian nicht wirklich überraschen. Jetzt, da der Code im Versionskontrollsystem gelandet ist, besteht die Notwendigkeit zur Koordination, und es ist klar, wer mit wem reden muss. Daher macht sich Mark auf den Weg und trifft sich mit Team »Marge« am anderen Ende des Flurs. Sie unterhalten sich darüber, wie sie zusammenarbeiten wollen, damit sie von der Arbeit der anderen profitieren.

Für den Eintrag, den das Team »Handel« gerade entwickelt, tatsächlich aber auch für jeden Eintrag jedes anderen Teams, wurden automatisierte Akzeptanztests geschrieben, *bevor* man überhaupt mit dem Programmieren der Lösung begonnen hat. Dementsprechend integrieren sie ebenfalls die automatisierten Tests in Ergänzung zum kontinuierlichen Integrieren des Quellcodes. Diese Akzeptanztests lassen die Teammitglieder regelmäßig laufen und sobald ein Test fehlschlägt, wird den Teams signalisiert, dass Koordinationsbedarf besteht. Der Code sagt ihnen: »Hey! Es gibt da ein Problem! Ihr müsst miteinander reden und das lösen.«

Regel
Das Perfektionsziel ist es, die Definition of Done dahingehend zu verbessern, dass nach jedem Sprint ein potenziell auslieferbares Produkt vorliegt (oder zumindest häufiger als zurzeit).

Ein weiterer, großer Vorteil, dass die Gruppe kontinuierlich integriert, automatisiert testet und jedes Mal, wenn der Build bricht, die Arbeit liegen lässt, um den Build zu fixen, ist natürlich auch, dass das Produkt mehr oder weniger kontinuierlich bereit ist, in Produktion zu gehen. Es gibt keine speziellen Integrations- oder Testteams, die für zusätzliche Verzögerungen, Übergaben und Komplexität sorgen würden.

Gesamtretrospektive

*(**Wegweiser**: Gesamtretrospektive, S. 344)*

Am zweiten Tag im Sprint kommen Sam und die anderen Scrum Master, der Product Owner Paolo, ein Standortleiter und Stellvertreter der meisten Teams für eine Gesamtretrospektive zusammen, die auf 90 Minuten angesetzt ist und mit dem *letzten* Sprint im Zusammenhang steht.

Regel
Eine Gesamtretrospektive findet im Anschluss an die Retrospektiven der Teams statt, um team- und systemübergreifende Themen zu diskutieren und Experimente zur Verbesserung zu beschließen. Der Product Owner, die Scrum Master, Stellvertreter aus den Teams und Führungskräfte (wenn überhaupt) nehmen daran teil.

Warum führen sie die Gesamtretrospektive nicht durch, bevor der neue Sprint startet? Sie hätten es machen können, aber normalerweise endet ein Sprint an einem Freitag und ein neuer beginnt am Montag (im Gegensatz zu Sams Vorschlag, einen Mittwoch-Donnerstag-Übergang zu versuchen). Und am letzten Freitag haben sie sowohl das Sprint-Review als auch die Retrospektiven auf Teamebene abgehalten. Anschließend hatten sie nicht mehr die Energie, am Ende des Tages noch eine fesselnde Gesamtretrospektive abzuhalten. Und so haben Sam und die anderen eine Gesamtretrospektive früh am Anfang des nächsten Sprints gewählt. Insgeheim denkt Sam nicht, dass diese Verzögerung eine gute Idee war – er hätte eher die Sprint-Planung ein wenig später nach dieser Retrospektive durchgeführt, aber er möchte, dass die Gruppe das selbst herausfindet.

Wegweiser
Verbessere das System, S. 346

Sie konzentrieren sich auf ein systemweites Thema und dessen Verbesserung: Wie laufen die Koordination, das Teilen von Informationen und das Lösen von Problemen mit der ganzen Gruppe während des Sprints? Bisher hat man Scrum-of-Scrum-Meetings ausprobiert und das fanden sie nicht sehr effektiv. Sam erklärt das Prinzip von Open Space und sie vereinbaren, das in diesem Sprint auszuprobieren.

Aktivitäten zur Koordination

*(Siehe Kapitel **Koordination & Integration**, S. 309)*

Regel
Teamübergreifende Koordination wird von den Teams entschieden.

Der vierte Tag zeigt eine Vielzahl von Ideen zur Koordination in LeSS:

In LeSS hält jedes Team ein Daily Scrum ab wie üblich. Um die Koordination zwischen den Teams »Handel« und »Marge« zu unterstützen, macht Mira sich auf den Weg zum Daily Scrum von Team »Marge«, um das Meeting als *Scout* zu beobachten. Anschließend kommt sie zurück und berichtet dem Team, was sie gelernt hat. Und jemand aus dem Team »Marge« macht das Gleiche.

Wegweiser
Scouts, S. 332

Wie in der Gesamtretrospektive beschlossen, halten die Gruppen ein 45-minütiges *Open-Space*-Meeting zur Koordination und zum Lernen ab, das mit Getränken und Snacks eröffnet wird. Sam agiert hier als Moderator, um den Gruppen beizubringen, wie man ein Open-Space-Meeting durchführt. Jeder ist herzlich eingeladen, aber die meis-

Wegweiser
Open Space, S. 330

ten Teams haben entschieden, nur ein paar Stellvertreter zu schicken. Mira und Mark vom Team »Handel« sind ebenfalls dabei. Gemeinsam planen sie, einen Open Space pro Woche zu versuchen.

Wegweiser
Communitys, S. 319

Die Test-*Community*, die die größte Menge an Teilnehmern aus den meisten Teams hat, kommt für eine halbe Stunde zusammen, um Marys Vorschlag bezüglich eines neuen Werkzeugs für automatisierte Akzeptanztests zu hören. Enthusiastisch stimmen sie zu und Mary schlägt vor, dass ihr Team »Marge« das Werkzeug im nächsten Sprint ausprobieren wird, da alle sehr interessiert daran sind, das Werkzeug kennenzulernen.

Tipp
Etabliere eine Architektur-Community

Mira ist Mitglied der Design/Architektur-*Community*. In diesem Sprint ist kein Design-Workshop hinsichtlich der Gesamtarchitektur nötig, aber sie möchte gern im kommenden Sprint einen halbtägigen *Spike* machen, um eine neue Technologie zu erkunden. Sie veröffentlicht ihre Idee auf dem Werkzeug, das die Community zur Kollaboration einsetzt, und schlägt vor, den Spike zusammen mittels Mob Programming durchzuführen, um ihr gemeinsames Lernen zu verstärken.

Tipp
Stoppe und behebe Fehler, wenn es Probleme gibt.

Das Build-System scheint einen äußerst merkwürdigen Fehler zu haben. Zeit, die Arbeit zu *stoppen und den Fehler zu beheben*! In diesem Sprint ist Team »Handel« dafür verantwortlich, und das ist eine von Marks weiteren Spezialitäten. Daher meldet er sich freiwillig, den Fehler zu beheben, und fragt ein anderes Teammitglied, ob dieses mit ihm paarweise arbeiten möchte. So kann Mark seinem Kollegen helfen, mehr über das Build-System zu lernen.

Tipp
Experten unterrichten andere.

Regel
Klärung findet idealerweise zwischen den Teams und den Nutzern und anderen Stakeholdern statt.

Später besuchen Mira und ein paar weitere Teammitglieder die Gruppe, die sich um Kundenbetreuung und Training kümmert und somit sehr eng mit den Anwendern zusammenarbeitet. Miras Team hat gerade sein erstes Product Backlog Item (PBI) fertiggestellt und will frühzeitiges Feedback von denen bekommen, die näher am Anwender dran sind. Einer der Trainer hat Zeit und spielt mit dem neuen Feature herum. Nach einer gewissen Zeit gehen die Mitglieder von Team »Handel« wieder mit ein paar Ideen, wie sie das Feature verbessern können, an die Arbeit zurück.

Tipp
Frühes Feedback

Wegweiser
Kommuniziere mittels Code, S. 316

Wegweiser
Integriere kontinuierlich, S. 317

Im weiteren Tagesverlauf arbeiten Mark und der Rest des Teams »Handel« an ihrem zweiten PBI. Mark hat gerade einen 10-minütigen TDD-Zyklus hinter sich gebracht und hat sauberen, stabilen Code nach einer minimalen Änderung vor sich. Und wieder einmal stellt Mark eine kleine Änderung unter Versionskontrolle in das zentrale, gemeinsame Repository ein (ganz oben im Hauptentwicklungszweig, dem Trunk). Das geschieht ungefähr alle 10 Minuten und so integrieren Marks Team und alle anderen auch *kontinuierlich*. Mark schaut hinüber zum großen, deutlich sichtbaren Rot-Grün-Monitor und sieht,

wie das Build-System erfolgreich alle Tests durchläuft – nicht nur die des Teams, sondern die der gesamten Gruppe.

Gesamt-Product-Backlog-Refinement

*(**Wegweiser**: Arten von Product Backlog Refinements, S. 269)*

Regel

Führe Multiteam- und/oder ein Gesamt-PBR durch, um das gemeinsame Verständnis zu erhöhen, und nutze die Gelegenheit zur Koordination, wenn es eng zusammenhängende PBIs gibt oder das Bedürfnis nach breiterem Input oder Lernen besteht.

Am fünften Tag nehmen Mark und Mira zusammen mit Vertretern der anderen Teams und Paolo, dem Product Owner, an einem *Gesamt-PBR-Workshop* teil. Paolo eröffnet den Workshop, in dem er seine gegenwärtige Meinung, wie die Ausrichtung des Produktes ist, allen mitteilt. Außerdem gibt er einen Ausblick darüber, wohin es kurzfristig gesehen als Nächstes mit dem Produkt hingeht und vor allem warum. Um dem Rest der Anwesenden dabei zu helfen, seine Gründe zu verstehen, schaut er noch einmal sein Priorisierungsmodell mit der Gruppe an, das folgende Faktoren berücksichtigt: Auswirkung auf den Profit, Auswirkung auf den Kunden, unternehmerisches Risiko, technisches Risiko, Verzögerungskosten und noch einige mehr.

Tipp

Rotiere die Stellvertreter in jedem Sprint.

Wegweiser

Priorisierung über Klärung, S. 195

Paolo fragt nach Feedback und Ideen aus der Gruppe hinsichtlich der kommenden Produktausrichtung, und die Gruppe diskutiert, welche PBIs als Nächstes verfeinert werden müssen. Obwohl Paolo weiß, dass er das letzte Wort hinsichtlich der Priorisierung hat, arbeitet er hart daran, dass die Teams selbstständig versuchen, seine Denkweise nachzuvollziehen. Dabei lernt er die Denkweise der Teams kennen, von der er ebenfalls lernen kann. Paolo will erreichen, dass die Teams sich am *Produktbesitz* beteiligen und sich selbst entsprechend einbringen.

Wegweiser

Fünf Beziehungen, S. 197

Tipp

Der PO beteiligt das Team am Produktbesitz.

Die Gruppe teilt ein paar große, neue Einträge auf, klärt diese vorerst relativ grob (die weiter Klärung kommt später) und schätzt diese mittels Planning Poker, um dadurch mehr über diese Einträge zu *lernen* und weniger, um Schätzungen abzugeben.

Wegweiser

Schneiden, S. 281

Die Stellvertreter von drei Teams (inkl. »Handel« und »Marge«) beschließen, später ein Multiteam-PBR durchzuführen, um das gemeinsame Verständnis für diese Einträge zu erhöhen, da diese stark zusammenhängen. Die Stellvertreter der anderen beiden Teams wählen Einträge aus, auf die sie sich dann unabhängig voneinander in separaten PBR-Sitzungen konzentrieren können.

Wegweiser

Schätzung skalieren, S. 291

Multiteam-PBR und Team-PBR

*(**Wegweiser**: Multiteam-PBR, S. 273)*

Am sechsten Tag kommen alle Mitglieder der drei Teams zu einem *Multiteam-PBR-Workshop* in einem großen Raum zusammen.

Obwohl das Hauptgeschäft des Unternehmens das Entwickeln und Verkaufen seiner Handelslösung ist, hat die Firma eine kleine Gruppe von Anleihen-Händlern, die das Produkt ebenfalls nutzen – mit verhältnismäßig kleinen Positionen, um sie bei der Stange zu halten, aber ohne Risiko. Auf diesem Weg hat die Firma einen besseren Einblick in Markttrends und auch die Möglichkeit, dass sehr fachkundige Nutzer problemlos mit den Entwicklungsteams reden können.

Regel

Die gesamte Priorisierung geht über den Product Owner, aber die Klärung der Inhalte erfolgt so weit wie möglich direkt zwischen den Teams und den Kunden/Nutzern und anderen Stakeholdern.

Tanya und Ted sind die beiden Händler, die Paolo von einem Trend erzählt haben, was zu den Einträgen führte, die jetzt in der Multiteam-PBR-Sitzung verfeinert werden sollen. Also nehmen auch beide an der Sitzung teil, da sie die Experten sind, die die Teams unterstützen und die neuen Einträge klären können.

Die anderen beiden Teams, die ebenfalls in der Diskussion mit ein paar anderen Händlern sind, halten separate PBR-Workshops ab, um ein paar anfänglich verfeinerte PBIs final zu klären und bei einigen neuen PBIs mit der Klärung zu beginnen. Außerdem nimmt einer der drei Anwälte der Firma mit Schwerpunkt »Finanzielle Regularien und Regelüberwachung« an einem der beiden Workshops teil, um bei der Klärung zu unterstützen.

Wegweiser

Werkzeuge für große Product Backlogs, S. 231

Tipp

Benutze ein Wiki für die Details der PBIs

Als einer der letzten Schritte in den PBR-Meetings machen die Teilnehmer Fotos von allem, was während der Meetings an den Wänden und Whiteboards entstanden ist. Diese Bilder fügen sie dann in den entsprechenden Seiten im Wiki ein, die dazu verwendet werden, alles für jeden Eintrag aufzuzeichnen. Dabei werden die Texte und Tabellen aktualisiert und aufgeräumt, die schnell mal während der Diskussionen hinzugefügt worden sind.

Eine Unterhaltung über Backlogs auf Teamebene und Product Owner

Nach dem Multiteam-PBR-Workshop sieht Mike, der erst vor Kurzem in der Firma angefangen hat, Sam an der Kaffeemaschine stehen und nutzt die Gelegenheit für einen kleinen Plausch. Mike sagt: »Hey Sam, mich interessiert deine Meinung zu folgendem Thema: Im Refinement-Workshop, den wir gerade beendet haben, ist mir aufgefallen, dass wir direkt mit einigen Händlern zusammengearbeitet haben, um ein paar Dinge zu klären. Aber ist das nicht eher ineffizient? In meiner letzten Firma hatte jedes Team seinen eigenen Product Owner. Der hat Stories geschrieben, Wireframes und Spezifikationen erstellt und hat uns diese

zur Implementierung gegeben. Dadurch konnten wir uns voll auf das Programmieren konzentrieren. Und jedes Team hatte sein eigenes Product Backlog, das vom entsprechenden Product Owner priorisiert wurde. All das sehe ich hier nicht. Warum macht man das hier anders?«

Sam antwortet: »Interessante Fragen. Hast du was dagegen, wenn ich dir ein paar Fragen stelle, um unsere Herangehensweise näher zu erkunden?«

»Nein, leg' los.«

»Lass' uns als Erstes ein Product Backlog versus viele Backlogs auf Teamebene betrachten. Angenommen, jedes Team hat sein eigenes Backlog. Wie einfach und effektiv ist es für einen wirklichen Gesamt-Product-Owner, den Überblick zu behalten? Und wie viel kann ein Team über die Anforderungen und die Designs von Einträgen aus dem Backlog eines anderen Teams wissen?«

Mike erwidert: »Das kann ich dir ganz klar durch die Erfahrung in meiner letzten Firma beantworten: nicht viel.«

Sam fährt fort: »Nun lass' uns annehmen, es gibt acht Teams und acht Team-Backlogs. Was, wenn aus Unternehmenssicht oder aus Produktperspektive aus irgendeinem Grunde die Einträge in zwei der acht Team-Backlogs aktuell die bei Weitem wichtigsten Dinge sind oder die mit der höchsten Priorität? Vielleicht verändert sich der Markt dahingehend, dass es zu dieser Situation kommt. Hier mal ein paar Fragen für dich: Können die sechs Teams, die an den Backlogs mit der niedrigeren Priorität arbeiten, einfach wechseln und mit der Arbeit an den hoch priorisierten Einträgen aus den anderen beiden Backlogs beginnen? Und wie wahrscheinlich ist es, dass die Gruppe dieses Problem überhaupt erkennt, wenn alle in ihrem eigenen Team mit ihrem eigenen Backlog und den lokalen Prioritäten eingeschlossen sind?«

Mike antwortet: »Unsere Teams in der alten Firma haben nur an ihrem eigenen Team-Backlog gearbeitet. Sie konnten nicht zu anderen wechseln. Aber warum sollten sie auch? Ist das nicht ineffizient?«

Sam beantwortet das: »Also, aus Unternehmensperspektive arbeiten die Teams nur deshalb ‚effizient', da ihr eng gefasstes Wissen dadurch erzeugt wurde, dass jedes Team sich auf ein anderes Team-Backlog fokussiert und weil die Gesamtpriorität und der Gesamtüberblick nicht sichtbar sind. Lass' mich dir einige Fragen stellen: Erscheint das unflexibel oder flexibel – agil? Und wird dadurch die Möglichkeit optimiert, dass Personen an den Dingen arbeiten können, die die größten Auswirkungen aus Unternehmensperspektive haben?«

Regel
Es gibt nur einen Product Owner und nur ein Product Backlog für das komplette auslieferbare Produkt.

Mike hält einen Moment inne. »Oh! Ich glaub', ich hab's. Es ist eigentlich gar nicht agil, auch wenn unsere Gruppe gesagt hat, wir seien agil. Wir waren insgesamt überhaupt nicht empfänglich für gewinnbringendste Änderungen. Und der Product Owner meines alten Teams sagte, er priorisiere unser Team-Backlog nach dem höchsten Wert. Aber jetzt sehe ich, dass mein Team eigentlich nur damit beschäftigt war, effizient daran zu arbeiten, was aus höherer Perspektive betrachtet einen sehr geringen Wert haben könnte.«

Sam sagt: »Genau. Das ist einer von vielen Gründen, warum wir hier *ein* Product Backlog haben und keine Team-Backlogs, auch wenn wir viele Teams haben. Kurzum, es unterstützt die Konzentration auf das Gesamtprodukt, die Systemoptimierung und Agilität. Und natürlich ist es einfacher und man bekommt leichter mit, was in der gesamten Gruppe los ist.«

»Also«, kommentiert Mike, »ich habe schon bemerkt, dass es in meiner alten Firma für alle Teams viel härter war, wirklich zur gleichen Zeit *zusammenzuarbeiten*, da wir an unterschiedlichen Zielen in asynchronen Sprints gearbeitet haben. Hier fühlt es sich an, als haben alle Teams eher einen gemeinsamen Fokus und eine gemeinsame Ausrichtung, zusammen in einem Sprint.«

»Exakt!«, erwidert Sam, bevor er fortfährt.

Regel
Der Product Owner sollte nicht allein an der Verfeinerung des Product Backlog arbeiten; er wird unterstützt durch mehrere Teams, die direkt mit Kunden/Nutzern und anderen Stakeholdern arbeiten.

»Ich habe hierzu noch ein Frage: Wenn es nur ein Product Backlog gibt und nur einen wahren Product Owner, der es priorisiert, aber jedes Team seinen eigenen sogenannten Product Owner hat, der per Definition kein Team-Backlog priorisiert – da es ja keines gibt –, was machen diese Product Owner dann den ganzen Tag?«

Mike erwidert: »Nun, in meiner letzten Firma war die Aufgabe des Product Owners auf Teamebene, mit den Nutzern zu sprechen und für das Team Stories zu schreiben, sodass diese sich auf effizientes Programmieren konzentrieren konnten. Während der entsprechende Product Owner daran gearbeitet hat, Anforderungen zu sammeln und zu schreiben.«

Regel
Die gesamte Priorisierung geht durch den Product Owner, aber die Klärung der Inhalte erfolgt so weit wie möglich direkt zwischen den Teams und den Kunden/Nutzern und anderen Stakeholdern.

Daraufhin fragt Sam: »Mike, bevor du Scrum-Begriffe wie ‚Product Owner' oder so kennengelernt hast, wie hast du den Mittelsmann zwischen Entwicklern und echten Kunden genannt – denjenigen, der Anforderungen gesammelt hat und sie dann den Entwicklern gegeben hat?«

»Ich habe in meiner letzten Firma vor der Scrum-Einführung angefangen«, antwortet Mike, »und damals gab es eine Gruppe von Business-Analysten, die das gemacht haben. Nachdem wir Scrum eingeführt haben, wurden wir gebeten, diese Gruppe Product Owner zu nennen.«

»Und heute in unserem PBR-Workshop«, fragt Sam, »hast du dich da mit den Händlern unterhalten, die anwesend waren?«

»Lass' mich nachdenken«, antwortet Mike, »ja, ich habe mich mit Tanya über ihre Idee unterhalten, den Handel mit russischen Geschäftsanleihen zu analysieren. Es wirkte auf mich eher ein bisschen verwirrend, darum fragte ich: warum? Sie erklärte mir, dass die Sorgen, Geld würde in Überseekonten gewaschen werden, der Grund seien. Nun, sie wusste nicht, dass wir vor Kurzem an einigen anderen Features gearbeitet haben, die neue regulatorische Datenbanken für EU und USA integriert haben, um genau das auswerten zu können. Also habe ich ihr einen anderen Ansatz vorgeschlagen, von dem ich denke – und sie stimmte mir zu –, dass er das Problem besser löst.«

»Nun, wo ich darüber nachdenke«, reflektiert er, »hätte so etwas in meiner alten Firma eher nicht stattgefunden, da wir nur äußerst selten direkt mit Nutzern gesprochen haben.«

Weiteres zur Entwicklung

Von Minute zu Minute und Tag für Tag entwickeln die Teams Code, integrieren diesen kontinuierlich in Verbindung mit voller Testautomatisierung. Wenn der Build bricht, halten die Teams an und beheben das Problem – sie arbeiten in Richtung ihres Perfektionsziels, ein fertiges, auslieferbares Produkt zu erstellen, das sie kontinuierlich an die Kunden ausliefern können. Aus diesem Grund gibt es zum Sprint-Ende hin, wo die Teams sich auf die Teilnahme am Sprint-Review vorbereiten, kein verrücktes Gehetze auf den letzten Drücker, um einen großen Batzen Code zu integrieren und zu testen – der Code ist ja schon lange integriert und getestet.

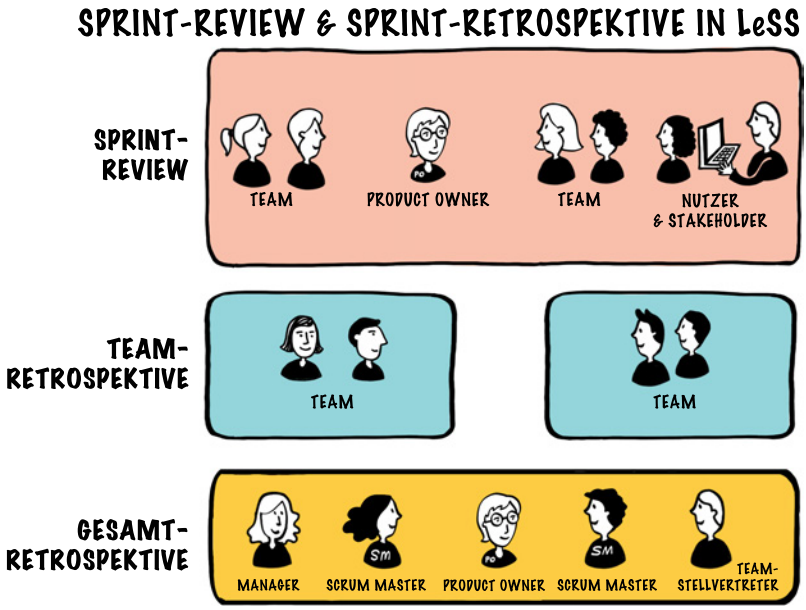
Sprint-Review

*(Siehe Kapitel **Review & Retrospektive**, S. 339)*

Schließlich kommt der letzte Tag des Sprints und es wird Zeit für ein gemeinsames Sprint-Review. Wer ist dabei? Paolo (Product Owner und leitender Produktmanager), alle internen Anleihenhändler, ein paar Trainer und Personen aus der Kundenbetreuung, ein paar Leute aus dem Vertrieb und vier Nutzer von externen Kunden, die geringere, jährliche Lizenzkosten zahlen und im Gegenzug dazu regelmäßig an den Reviews teilnehmen. Außerdem sind alle Teammitglieder dabei – von sämtlichen Teams.

Regel

Es gibt nur ein für alle Teams gemeinsames, produktbezogenes Sprint-Review.



Wegweiser
Review-Basar, S. 342

Aufgrund der Tatsache, dass es so viele Dinge zu erkunden gibt, beginnt die Gruppe mit einem einstündigen *Basar* – so etwas wie eine Wissenschaftsmesse – mit vielen Geräten im Raum. Jedes ist so eingerichtet, dass es für die Erkundung eines unterschiedlichen Satzes an fertiggestellten Einträgen zur Verfügung steht. Einige Teammitglieder bleiben an bestimmten Stellen im Raum stehen, um Feedback einzusammeln, während die anderen die neuen Features benutzen und diskutieren.

Nach einer Stunde kommt die Gruppe wieder zusammen und diskutiert unter Leitung von Paolo Fragen und Feedback. Anschließend besprechen sie die zukünftige Ausrichtung. Paolo teilt den Anwesenden mit, wie sich der Markt und die Konkurrenten entwickeln, und seine Gedanken dazu, wohin es als Nächstes für das Produkt geht, und bittet um Ratschläge.

Tipp
Diskutiere die Ausrichtung der kommenden Sprints.

Teamretrospektive

Nach einer Pause hält das Team »Handel« (und alle anderen Teams ebenfalls) eine eigene, separate Sprint-Retrospektive auf Teamebene ab. Sie beschließen, dass das Abhalten eines Multiteam-Design-Workshops mit dem Team »Marge« nach der Sprint-Planung (anstatt vorher) in diesem Fall weit vom Ideal entfernt war, da wichtige Themen bis zur letzten Minute unbesprochen zurückgeblieben sind – Themen, die durchaus die Entwicklung ernsthaft hätten blockieren oder verkomplizieren können. Also entscheiden sie, dass sie während der PBR-Sitzungen im nächsten Sprint versuchen, Einträge zu identifizieren, bei denen wichtige Kernfragen in Bezug auf das Design noch offen sind und andere Teams mitdiskutieren sollten. Wenn solche Dinge identifiziert werden, wird ein Multiteam-Design-Workshop so schnell wie möglich abgehalten.

Regel
Jedes Team hat seine eigene Sprint-Retrospektive.

Das Ende

Der Sprint ist geschafft! Sam lädt das Team »Handel« ein, ihn und Mira zu ihrer Lieblingskneipe – dem Belgischen Pub am Ende der Straße – zu begleiten, um ihren Geburtstag zu feiern.

Wegweiser
Belgisches Starkbier

Zusammenfassung

Einige wichtige Punkte aus der Geschichte:

- Sie hebt das Vorgehen der Menschen und Teams durch einen Sprint in LeSS hervor.
- Sie verbindet Geschichtselemente mit spezifischen Wegweisern und Regeln aus LeSS.
- Für einen Leser, der Scrum kennt, sollten die Events vertraut sein.
- Die Geschichte zeigt einen ganzheitlichen Produktfokus, selbst mit vielen Teams.
- Die Aktivitäten heben teambasiertes Lernen und Koordination hervor.
- Einträge werden durch kontinuierliche Integration entwickelt, sodass Kommunikation durch Code die dezentrale Koordination

und das »einfach miteinander Reden« (*Just Talk*) unterstützt in Ergänzung zu kontinuierlicher Auslieferung.
- Teams beseitigen Unklarheiten direkt mit Nutzern und Kunden, um Übergaben zu reduzieren und Verständnis, Empathie und Eigenverantwortung wachsen zu lassen.

LeSS-Geschichte: Der Fluss von Einträgen

Diese Geschichte konzentriert sich mehr auf den Fluss der *Einträge (Features)* durch einen Teil des Sprints, in erster Linie während der Verfeinerung und Entwicklung.

Portia beendet ihr Meeting mit der staatlichen Regulierungsbehörde und macht sich auf in Richtung Flughafen – und dann ab nach Hause. Sie ist ebenfalls Produktmanager, spezialisiert auf die Entwicklung von Regularien und Audits[11] – und darin hilft sie Paolo.

Portia trifft sich später mit Paolo. Sie schreibt die Zusammenfassung der neuen Regeln, die Auswirkungen auf ihr Produkt haben werden, auf ein paar Karten. Außerdem schreibt sie dazu, welche Kunden welche Features ihrer Meinung nach zuerst haben wollen. Paolo zeigt auf die fünf Karten und fragt: »Soweit du weißt, deckt das also die gesamte Arbeit ab?« Portia lächelt und sagt: »Das sind Regularien. Die sind *nie* fertig oder eindeutig.«

Wegweiser
Helfer für Product Owner, S. 196

Paolo fragt: »Kannst du diese Dinge bitte für mich im Product Backlog eintragen? Es reicht, wenn sie ungeordnet ganz unten landen.«

»Sicher.«

11. Zusätzlich zu einem leitenden Produktmanager, der oft als Product Owner dient, haben viele große Gruppen ein paar unterstützende Produktmanager, von denen jeder auf ein Hauptmarktsegment oder einen Kundenbereich spezialisiert ist.

Eine Woche später erzählt Paolo Portia: »Ich möchte demnächst mit der Auslieferung einiger Teile dieser großen regulatorischen Anforderung im Kontext ›Anleihen und Derivate‹ beginnen. In den Workshops zum Product Backlog Refinement des nächsten Sprints werde ich darum bitten, dass einige Teams sich darauf konzentrieren. Da du die Person mit dem größten Wissen in diesem Bereich bist, möchte ich dich bitten, beim Gesamt-PBR dabei zu sein und auch bei den Refinement-Workshops, bei denen die Teams dich dabei haben wollen. Könntest du auch eine Seite im Wiki einrichten, auf der man Links zur Dokumentation der neuen Regularien findet, und diese dann mit den Teams teilen? Das wäre klasse.«

»Bereits erledigt«, antwortet Portia.

Wegweiser
Werkzeuge für große Product Backlogs, S. 231

Tipp
Benutze eine Tabelle und ein Wiki für ein großes Product Backlog.

Gesamt-PBR

Paolo beginnt einen kurzen Gesamt-PBR-Workshop. »Wir haben eine Menge Arbeit mit den neuen Regularien. Und wir müssen demnächst damit anfangen, Teile, die damit in Verbindung stehen, auszuliefern. Es gibt nämlich eine gesetzliche Frist: das Ende des Geschäftsjahres. Wir werden mehr wissen, wenn wir erst mal einige Sachen geschnitten und geschätzt haben. Aber es würde mich nicht überraschen, wenn die Umsetzung dieser Dinge letztlich drei oder mehr Teams erfordert – und eine Menge Zeit.«

Wegweiser
Arten von Product Backlog Refinements, S. 269

Die Gruppe schneidet den neuen, gigantischen Eintrag in nur wenige, größere Teile, um die wesentlichen Elemente kennenzulernen. Das weitere Schneiden erfolgt später in Einzelteam- oder Multiteam-PBR-Sitzungen. Portia geht zum Whiteboard. Auf die linke Seite schreibt sie »Regularien für Anleihen und Derivate«. Im weiteren Gesprächsverlauf skizziert sie gemeinsam mit der Gruppe eine Baumstruktur mit vier Zweigen, wobei jeder Zweig einen der vier Schnitte in Unterelemente darstellt. Sie müssen auch gar nicht tiefer in das Thema eintauchen, denn so vermeiden sie eine übermäßige Analyse.

Wegweiser
Schneiden, S. 281

Als Nächstes erstellt die Gruppe vier Karten für die neuen Einträge und schätzt diese gemeinsam mittels Planning Poker und relativen Punkten. Als Referenz dienen bereits bestehende, wohlbekannte Einträge im Product Backlog. Das Hauptziel ist nicht, Schätzungen zu bekommen, sondern gemeinsam mit Portia Fragen aufzudecken und darüber zu diskutieren.

Wegweiser
Schätzung skalieren, S. 291

Dann fragt Paolo: »Also, Portia, dies sind die großen vier. Welches kommt als Erstes?«

Sie zeigt auf die zweite Karte. »Freiverkäufliche, exotische Anleihen und Derivate.«

Paolo sagt: »Wir müssen schnellstmöglich damit anfangen, einiges davon auszuliefern. Es gehört viel höher ins Product Backlog. Also hätte ich gern, dass sich ein Team ein Stückchen davon ›abbeißt‹ und sich damit im nächsten Sprint auseinandersetzt. Ist jemand interessiert?«

Team »Handel« meldet sich freiwillig.

Schließlich entscheiden die Teammitglieder aus drei anderen Teams, einen Multiteam-PBR-Workshop für verwandte Einträge abzuhalten.

Team-PBR: Reinbeißen

Am nächsten Tag führt das Team »Handel« mit Portia einen PBR-Workshop auf Teamebene durch. Sie haben nur einen der vier gigantischen Einträge, auf den sie sich konzentrieren: »Neue Regularien für freiverkäufliche (OTC), exotische Anleihen und Derivate.« Sam (ihr Scrum Master) ist ebenfalls anwesend. Portia sagt: »Dies ist ein unglaublich komplexes Element in einem Bereich, der offen gesagt niemandem wirklich bekannt ist. Es wird dauern, den Eintrag zu schneiden, ihn wirklich zu verstehen und gut zu spezifizieren.«

Daraufhin fragt Sam: »Müssen wir ihn wirklich *in Gänze* verstehen? Und wird uns dieses ganze Rumanalysieren mehr Erkenntnisse bringen oder könnte das uns nicht sogar beim Lernen *behindern*?«

Wegweiser
Nimm nur ein Stückchen, S. 223

Er prüft mit Portia und dem Team die Idee, »ein Stückchen zu nehmen«: Schneide ein einziges, winziges Fragment heraus, verstehe es richtig und implementiere es schnell. Sam fasst zusammen: »Ihr wisst, dass Diagramme nicht abstürzen können und Dokumente nicht laufen.«

Gemeinsam mit Portia schneidet das Team ein winzig kleines Stückchen in Form eines eher dünnen, kundenzentrierten, ganzheitlichen Elements heraus.

Tipp
Spezifikation am Beispiel, siehe Abschnitt »Klären« auf S. 275

Ab jetzt konzentriert sich das Team nur auf dieses kleine Stückchen, klärt und implementiert es. Erst nach der Implementierung mit anschließendem Feedback werden sie zu weiterem Schneiden und Verfeinern zurückkommen. Den Rest des Tages verbringen Portia und das Team damit, durch Spezifikation am Beispiel auf ihrem Stückchen »herumzukauen«.

Multiteam-PBR: Rotierende Verfeinerung

Wegweiser
PBR mit verteilten Standorten, S. 274

Ein Ergebnis des Gesamt-PBR war die Entscheidung, dass das Team »Handel« »ein Stückchen« nimmt. Ein anderes bestand in der Entscheidung dreier Teams, einen Multiteam-PBR-Workshop für verwandte Einträge durchzuführen, um das Lernen und die Agilität meh-

rerer Teams dadurch zu erhöhen, dass diese die gleichen Einträge kennen und über die gleichen Elemente nachdenken.

Zusätzlich zu den Teammitgliedern der drei Teams nehmen auch die internen Händler Tanya, Ted und Travis an dem Meeting teil, um den Teams dabei zu helfen, mit der Klärung von ungefähr einem Dutzend neuer Einträge zu beginnen.

Zunächst bilden sie drei temporäre, gemischte Gruppen aus jedem Team. Diese gemischten Gruppen beginnen damit, verschiedene Elemente in getrennten Bereichen des Raumes zu klären. Jeder Bereich verfügt über ein Whiteboard, eine große, freie Wandfläche, einen Laptop und einen Beamer. Tanya ist bei einer Gruppe dabei, Ted bei einer anderen und Travis bei der dritten.

Dann führen sie eine sogenannte *Rotierende Verfeinerung* durch: Nach 30 Minuten endet ein Timer: Dong! Das ist das Zeichen, dass eine Gruppe zum Bereich einer anderen geht und umgekehrt, nur Tanya, Ted und Travis bleiben stehen. Der Timer wird neu gestartet, die Händler erläutern die bisherigen Ergebnisse den Neuankömmlingen und fahren mit der Klärung fort.

Abb. 2–1
Multiteam-PBR

Im Laufe des Tages werden verschiedene Einträge relativ klar oder es bleiben einige Fragen offen, die später weiter behandelt werden müssen. Dabei tauchen aber auch neue Elemente in einigen Arbeitsbereichen auf. Einige der größeren, neuen Einträge werden in zwei oder drei kleinere unterteilt.

Es passiert ein paar Mal während dieses Tages, dass die Gruppen ihre Klärungsarbeiten unterbrechen und eine Schätzung durchführen, meist um mehr zu erfahren und eine Unterhaltung anzuregen. Sie nutzen relative (Story) Points, die immer im gleichen Verhältnis zu einer allgemeinen Referenz stehen, und stimmen sich gegenüber bereits fertiger, wohlbekannter Einträge aus dem Product Backlog ab.

Wegweiser
Schätzung skalieren, S. 291

Product Backlog und Product Owner auf dem Laufenden halten

Wegweiser
Helfer für Product Owner, S. 196

Am Tag nach den PBR-Workshops

- aktualisieren Portia und ein paar Teammitglieder das Product Backlog, indem sie die neuen, geteilten Einträge, die von den Originaleinträgen abgeleitet wurden, einfügen und die Originaleinträge löschen.
- Sie fügen Links zu den neuen Seiten im Wiki hinzu, die während des PBR-Workshops entstanden sind und mehr Details zu den Einträgen enthalten.
- Sie tragen neue Schätzungen ein und fügen Einträge hinzu, die zur Umsetzung bereit sind.

Wegweiser
Der Umgang mit Elternelementen, S. 225

Später treffen sich Portia und diese Teammitglieder mit Paolo, um die Änderungen am Product Backlog durchzusehen und seine Fragen zu beantworten.

Das Ende

Einige wichtige Punkte aus der Geschichte:

- Nehmen Sie nur ein Stückchen von einem gigantischen Feature und lernen Sie daraus, etwas Kleines auszuliefern, um voreilige und übertriebene Analyse zu vermeiden.
- Führen Sie Multiteam-PBRs für Einträge durch, um gemeinsames Wissen über Teamgrenzen hinweg zu erlangen, was die organisatorische Agilität erhöht sowie ganzheitliches Produktwissen und selbstorganisierte Koordination fördert.
- Streben Sie die Fokussierung auf das Gesamtprodukt an, auch mit vielen Teams.

Weiter
Der nächste Abschnitt wechselt zum *LeSS-Huge*-Framework, das für große Gruppen und viele Teams genutzt wird.

LeSS-Huge-Framework

Requirement Areas

Mit 1000 oder auch mit nur 100 Personen, die gemeinsam an einem Produkt arbeiten, scheint ein Teile-und-herrsche-Ansatz aufgrund der Komplexität von so vielen Anforderungen und so vielen Menschen unvermeidbar. Üblicherweise unterteilt man größere Entwicklungsumgebungen nach folgenden Ansätzen:

- Einzelne Funktionsgruppen (Analysegruppe, Testgruppe, ...)
- Architektonische Komponentengruppen (Gruppe für die UI-Schicht, Gruppe für die Serverseite, Gruppe für den Datenzugriff, ...)

Dieses organisatorische Design führt zu einer langsamen, unflexiblen Entwicklung mit

1. einem hohen Maß an Verschwendung (z.B. Inventar, Work-in-Progress, Übergaben, verstreute Informationen, ...),
2. einem stark verzögerten ROI,
3. komplexer Planung und Koordination,
4. mehr unnötigem Management und
5. schlechtem Feedback und Lernen.

Und es ist mehr nach *innen* gerichtet organisiert – um einzelne Fertigkeiten, Architektur und Management herum – und weniger nach außen auf den Mehrwert für den Kunden.

Im LeSS-Huge-Framework hingegen, das dann angewendet wird, wenn es mehr als acht Teams gibt, erfolgt die Aufteilung in *wichtige Bereiche der Kundenbelange*, die **Requirement Areas** genannt werden. Das reflektiert das LeSS-Prinzip der *Kundenorientierung*.

Die magische Zahl Acht, S. 15

Größe
Eine Requirement Area ist *groß*, üblicherweise besteht sie aus *vier bis acht Teams*, nicht ein oder zwei. Der nachfolgende Abschnitt über *Area-Feature-Teams* erklärt warum.

Dynamik
Requirement Areas sind *dynamisch*. Im Laufe der Zeit wird sich die Bedeutung einer Area ändern, was dazu führt, dass sie wächst oder schrumpft, indem Teams der Area beitreten oder hinausgehen – höchstwahrscheinlich von oder zu einer anderen bestehenden Area.

Beispiel
Betrachtet man ein *Wertpapierprodukt* (zum Handeln mit Aktien), könnten folgende Bereiche einige der wichtigen Bereiche aus Kundensicht – Requirement Areas – sein:

- Geschäftsabwicklung (von der Preisfindung über die Erfassung bis hin zur Abrechnung)
- Anlagenservice (z. B. Abwicklung einer Aktienaufteilung, Dividenden)
- Markteinführung (z. B. Nigeria)

Vom Konzept her wird dem Product Backlog ein Attribut für die Requirement Area hinzugefügt und jeder Eintrag einer Area und nur einer Area zugeordnet:

Eintrag	Requirement Area
B	Markteinführung
C	Geschäftsabwicklung
D	Anlagenservice
F	Markteinführung
...	...

Dann können Leute sich auf ein **Area Product Backlog** konzentrieren (vom Konzept her eine *Blickweise* auf ein Product Backlog), wie z. B. die *Markteinführung*-Area:

Eintrag	Requirement Area
B	Markteinführung
F	Markteinführung

Ein gemeinsamer Sprint
Arbeitet jede Requirement Area unabhängig voneinander in ihrem eigenen Sprint, mit verzögerter Integration bis zu einem Datum in weit entfernter Zukunft? Nein.

In LeSS Huge wird kontinuierlich in einem gemeinsamen Sprint integriert.

Es gibt einen produktweiten Sprint, keine unterschiedlichen Sprints für jede Requirement Area. Er endet in einem integrierten Gesamtprodukt. Alle Teams aus allen Requirement Areas streben nach kontinuierlicher Integration über das gesamte Produkt hinweg.

Area Product Owner

In LeSS Huge wird eine neue Rolle eingeführt. Jede Requirement Area besitzt einen **Area Product Owner,** der sich auf diesen Bereich spezialisiert und sich auf das Area Product Backlog konzentriert.

Große Produktgruppen haben in der Regel mehrere unterstützende Produktmanager, die auf verschiedene Kundenbereiche spezialisiert sind, und einige von ihnen könnten durchaus als der Area Product Owner fungieren. Gelegentlich kommt es auch vor, dass ein Product Owner eine Doppelrolle bekleidet und auch als Area Product Owner für einen Bereich tätig wird. Das ist aber eher bei kleineren *LeSS-Huge*-Gruppen der Fall!

Area-Feature-Teams

Area-Feature-Teams arbeiten innerhalb einer Requirement Area (z.B. Anlagenservices) mit einem Area Product Owner und konzentrieren sich auf die Einträge in ihrem Area Product Backlog. Aus Teamsicht *gleicht die Arbeit der im kleineren LeSS-Framework*. Man arbeitet mit dem Area Product Owner genau so zusammen, als ob er der Product Owner wäre.

Die Teammitglieder lernen das Kundenumfeld für diesen Bereich gut kennen. Und erfreulicherweise decken die Einträge aus einer Requirement Area eine nahezu vorherbestimmbare Teilmenge der gesamten Codebasis ab, wodurch die Menge der Inhalte, die das Team erlernen muss, trotz des großen Gesamtprodukts recht gering ausfällt.

Kernpunkt bezüglich der Größe: *Viele* Feature-Teams arbeiten in einer Requirement Area.

> Eine Requirement Area hat normalerweise *vier bis acht* Teams.
> Das bedeutet, dass eine Requirement Area groß ist.

Die magische Zahl Vier

Zunächst stellt sich die Frage, warum eine Requirement Area eine empfohlene Obergrenze von acht Teams hat? Siehe *Die magische Zahl Acht,* S. 15.

Wie sieht es mit der Untergrenze von *vier* Teams aus? Warum nicht ein oder zwei Teams? Natürlich ist vier keine magische Zahl, aber sie schafft ein Gleichgewicht, sodass die Produktgruppe nicht aus vielen winzigen Requirement Areas besteht.

Worin besteht das Problem mit vielen winzigen Bereichen? Sie reduzieren die Übersicht über die Prioritäten der Gesamtproduktsicht und erhöhen die Anzahl von lokalen Optimierungen, den Koordinationsaufwand und die Komplexität. Sie benötigen mehr Rollen, produzieren Teams, die sich mehr und mehr spezialisieren, und die Flexibilität (Agilität) fehlt, die aufkeimenden, wertvollsten Features aus Sicht des Unternehmens umzusetzen. Des Weiteren wird der Area Product Owner in einem kleinen Bereich zusehends zu einem Business-Analysten zwischen den Nutzern und einem oder zwei Teams.

Gibt es keine vernünftigen *Ausnahmen* für die Untergrenze von vier? Doch:

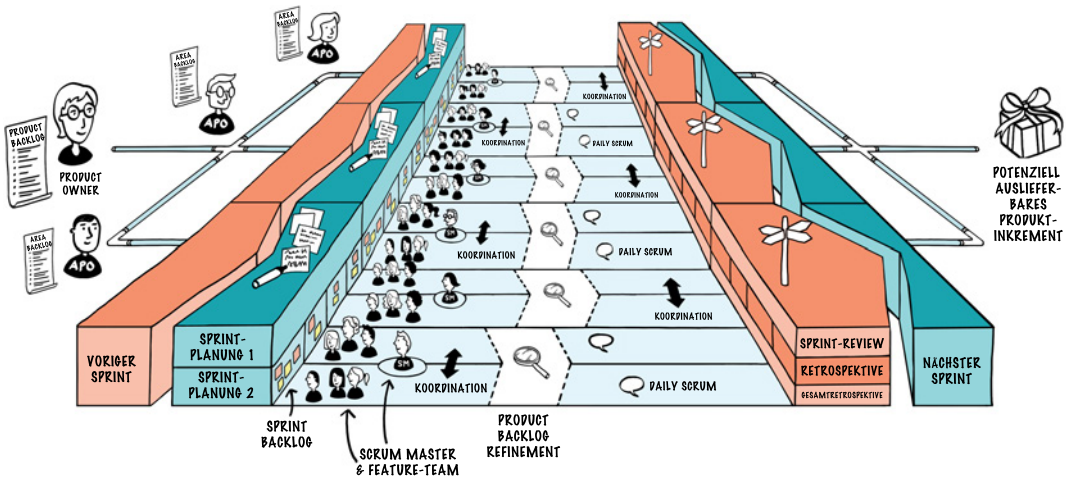
- In einer frühen Phase des Übergangs, wenn die Gruppe schrittweise einen neuen Bereich aufbaut, der in seiner Endausbaustufe aus vier oder mehr Teams bestehen wird. In diesem Fall starten Sie klein und einfach mit einem Team.
- Wenn man Teams ausgleichen muss, da in einem Bereich der Bedarf sinkt, während er in einem anderen steigt, und damit ein Bereich von vier auf drei Teams reduziert wird, dann ist es definitiv empfehlenswert, zwei kleine Bereiche zu einem großen zusammenzulegen.

Beispiel Requirement Areas und Teams

Zusammenfassend lässt sich sagen, ein *Wertpapier*-Produkt könnte

- einen Product Owner und drei Area Product Owner haben, diese bilden das Product-Owner-Team,
- sechs Feature-Teams im Bereich Geschäftsabwicklung,
- vier Feature-Teams im Bereich Markteinführung und
- vier Feature-Teams im Bereich Anlagenservice.

LeSS-Huge-Framework – Zusammenfassung



Jede Requirement Area arbeitet wie eine (kleinere) LeSS-Implementierung. Jede arbeitet parallel im Rahmen eines Gesamt-Sprints. Manchmal beschreiben wir einen Sprint in LeSS Huge auch als *einen Stapel von LeSS*.

> Aus der Sicht eines Teams in einer Requirement Area sieht LeSS Huge hinsichtlich der Events aus wie (das kleinere) LeSS.

Wie bei LeSS gibt es **Regeln** und optionale **Wegweiser** für LeSS Huge. Diese werden in den folgenden Geschichten beschrieben und in den späteren Kapiteln ausgearbeitet.

Rollen
Die gleichen Rollen wie bei LeSS, zusätzlich aber zwei oder mehrere **Area Product Owner** und vier bis acht Teams in jeder Requirement Area. Der *eine* **Product Owner** (der sich auf die Optimierung des Gesamtprodukts konzentriert) und mehrere Area Product Owner bilden das **Product-Owner-Team**.

Artefakte
Genau wie bei LeSS, zusätzlich ein *Requirement-Area*-Attribut im Product Backlog und demnach eine **Area-Product-Backlog**-Ansicht für jeden Bereich.

Events
Es gibt nach wie vor nur einen gemeinsamen Sprint für das Produkt. Er umfasst alle Teams und endet in einem gemeinsamen potenziell auslieferbaren Produktinkrement.

LeSS-Huge-Geschichten

LeSS Huge erlernen
Leser, die eine Erläuterung bevorzugen, können diese Geschichten bequem überspringen und direkt zu den nachfolgenden Kapiteln gehen.

Einfache Geschichten
Es sind absichtlich schlichte und einfache Geschichten, um die Grundlagen von LeSS Huge darzustellen.

Zwei Themen
Im Folgenden werden zwei Geschichten mit unterschiedlichen Themen behandelt:

1. Erstellen und ausbauen einer neuen Requirement Area, um sich mit einem neuen, gigantischen Anforderungsbereich zu befassen.
2. Arbeiten mit standortübergreifenden Teams. (Dies kommt beim kleineren LeSS-Framework auch vor, trifft aber vorwiegend auf LeSS Huge zu.)

LeSS-Huge-Geschichte: Eine neue Requirement Area

Wegweiser
LeSS Huge Product Owner, S. 212

Priti begrüßt Portia an ihrem ersten Tag in ihrem neuen Job.[12] Sie ist Führungskraft im mittleren Management für den Betrieb des Bereichs Sicherheit innerhalb des großen Handelsunternehmens. Aber auch in ihrer Rolle als Product Owner der internen *Sicherheitssysteme* ist Priti verantwortlich für die Suche und das Binden von Talenten an ihr Product-Owner-Team, bestehend aus mehreren Area Product Ownern. Und sie denkt, dass Portia eine fantastische neue Kollegin ist, weil sie genau die Expertise bietet, die für den Umgang mit einigen neuen riesigen Anforderungsbereichen erforderlich ist.

Während des kürzlich stattgefundenen Vorstellungsgesprächs war Portia noch Produktmanagerin, spezialisiert auf Regulierungsfragen bei einer Firma, die ein System für Handelsanleihen anbietet. Priti hatte ihr dabei die Situation folgendermaßen dargelegt: »Portia, nach

12. Zur Erinnerung: Die Benennung der Personen bedient sich einer Alliteration, um sich die Rollen besser merken zu können. Priti ist Product Owner, Portia ist Area Product Owner, Susan ist Scrum Master und Mario ist Teammitglied.

dem letzten Bankencrash sind die Regulierungsbehörden sehr strikt und sie verlangen von uns, dass wir die Maßnahmen der Dodd-Frank-Reform einhalten. Im Moment wissen wir nicht genau, was das genau bedeutet oder welche Auswirkungen das auf unser System haben wird. Du hast ein unglaubliches Wissen in diesem Umfeld und großartige, berufliche Verbindungen zu den Regulierungsbehörden. Ich würde mich freuen, wenn du zu unserer Gruppe dazustoßen würdest und uns hilfst, herauszufinden, wie man damit umgeht.«

Eine große Überraschung

Ein paar Tage später ... Priti begrüßt Portia, Peter und Susan in ihrem Büro. Peter ist Area Product Owner für Markteinführung und Susan ist Scrum Master aus dem Bereich Geschäftsabwicklung.

Priti sagt: »Wie ihr alle wisst, steht Dodd-Frank vor der Tür, und es ist ein riesiger Brocken. Was ihr nicht wisst, ist, dass heute Morgen die Regulierungsbehörde angerufen hat, und sie wollen, dass wir *sofort* handeln. Meine Annahme war falsch, dass wir erst nächstes Jahr beginnen könnten. Wir werden also in großem Maßstab anpassen müssen.

Ich glaube nicht, dass irgendjemandem klar ist, was das im Detail bedeutet – auch nicht der Regulierungsbehörde. Und wir wissen nicht, wie es sich auf unser System auswirken wird und wie viel Arbeit das kosten wird. Eins weiß ich aber schon: Es wird eine Menge sein! Aber jetzt ist ja Portia bei uns und obwohl unsere Systeme für sie völlig neu sind, hat sie ein besseres Verständnis von diesen Dingen als jeder andere. Also, wie können wir ihr helfen, mit diesem Berg von Arbeit zu beginnen?«

Susan fragt: »Ihr kennt doch alle das Bild von den legasthenischen Zombies?«

Peter und Priti nicken. Jeder kennt sie – und nicht nur ihren Namen. Die legasthenischen Zombies[13] haben wahrscheinlich die umfassendste Erfahrung von allen Teams. Sie gibt es schon seit Jahren und sie haben bei der LeSS-Einführung echt genervt. Das Team bestand aus zwei Mitgliedern der nun verwaisten Architekturabteilung und ein paar Leuten, die an dem System seit über fünfzehn Jahren gearbeitet hatten. Der Widerstand dieser Mitarbeiter gegen die LeSS-Einführung war legendär, da sie befürchteten, sie würden den »Überblick über das Gesamtsystem« verlieren. Zu ihrer Überraschung passierte genau das Gegenteil! Aufgrund ihrer tiefen Kenntnisse erhalten sie immer wieder schwierige Aufgaben. Und sie nehmen regelmäßig als

13. Ja, das war wirklich ihr Name in Lissabon!

Experten an Architektur-Workshops mit neuen Mitarbeitern teil. Mario, einer der früheren PowerPoint-Architekten, ist nun Koordinator für die Architektur-Community. Nach einer ausreichend großen Menge Bier würde er sicher zugeben, dass die engere Arbeit mit dem Code und den Tests zu einem *realistischeren* Verständnis des Systems geführt hat.

Susan fährt fort: »Wenn es ein Team gibt, das Portia helfen kann, den Umfang und die Auswirkungen von Dodd-Frank auf unser System zu verstehen, sind es die Zombies. Außerdem haben sie die Arbeiten am Sarbanes-Oxley Act vor ein paar Jahren geleitet. Morgen findet ihr PBR-Meeting statt. Sie müssen nur noch ein Feature abschließen. Warum lenken wir das Meeting nicht in die Richtung, über Dodd-Frank zu diskutieren und das Team zu fragen, ob sie sich in Vollzeit darum kümmern können?«

Verfeinern mit den Zombies

Am nächsten Tag im Refinement-Meeting mit den Zombies erklärt Portia die Situation: »Ihr habt sicher alle schon von der Dodd-Frank-Verordnung gehört. Was uns allerdings überrascht hat: Die Regulierungsbehörde hat uns angewiesen, dass wir ›sofort‹ tätig werden und bis zum Jahresende eine spürbare Einhaltung der Verordnung demonstrieren müssen. Sonst könnten sie unsere Handelsaktivitäten beschränken.«

Die Zombies sind sichtlich überrascht. Sie hatten zwar Gerüchte darüber gehört, aber sind nicht davon ausgegangen, dass die Zeit so drängt!

Mario sagt: »O.k., Portia, gib uns eine kurze Zusammenfassung, was dies bedeutet. Und wie unterscheidet es sich vom Sarbanes-Oxley Act?«

Portia nimmt einen Stift und beginnt, auf einem Whiteboard zu skizzieren. Nach etwa 45 Minuten ist sie mit der Übersicht fertig und die Zombies schauen ein wenig verblüfft.

»Ende des Jahres, haben sie gesagt?«, fragt Mario. »Selbst wenn die *ganze Gruppe* heute starten würde, würde es nicht fertig werden. Der Aufwand ist riesig!«

Er nimmt einen Stift und beginnt am Whiteboard mit einer groben Skizze ihres Systems, während er mit den restlichen Zombies die möglichen Auswirkungen bespricht.

Er sagt: »Portia, lass uns die Gelegenheit auch nutzen, dir das System besser verständlich zu machen. Frag nur!«

Portia sagt: »Kannst du kurz warten? Ich würde das gerne auf Video aufnehmen, um mich daran erinnern zu können.«

Michelle, ein altgedientes Teammitglied, wendet ein: »Wir sollten besser gleich mit der konkreten Entwicklung beginnen und dadurch mehr lernen, da wir ansonsten nur Zeit mit dem Analysieren verbringen. Ich hab das schon mehrfach erlebt.«

Susan, ihr Scrum Master, sagt: »Das erinnert mich an Folgendes ... Tom DeMarco hat einmal gesagt, dass der Grund für jedes fehlgeschlagene Projekt darin liegt, dass es zu spät gestartet ist.« Alle lachen und sie fährt fort: »Hier ein Vorschlag: *Nimm nur ein Stückchen*.«

Wegweiser
Nimm nur ein Stückchen, S. 223

Erstellen einer neuen Requirement Area

Am nächsten Tag trifft sich Portia mit Priti und dem Rest des Product-Owner-Teams. Portia gibt eine Zusammenfassung des Aufgabenbereichs, wie sie ihn nun versteht.

Priti sagt: »Das Ganze ist ja noch größer, als ich erwartet hatte. Wir müssen den Regulierungsbehörden innerhalb der nächsten Monate erste Fortschritte vorlegen und bereits ein großes Stück vor Ende des Geschäftsjahres – das sind sieben Monate ab jetzt. Nur um das Offensichtliche auszusprechen: Sie sind nun berechtigt, mehr von uns zu verlangen, und können uns den Laden zumachen. Wie ihr wisst, hat unser CEO erst letzten Monat unmissverständlich ausgeführt, dass neue regulatorische Anforderungen Vorrang vor allen anderen Anliegen haben. Ich habe gute Erfahrung damit gemacht, wenn wir unseren guten Willen und unsere Flexibilität gegenüber den Aufsichtsbehörden zeigen, indem wir ihnen erste Ergebnisse früh liefern, transparent sind und reagieren können. Lasst uns also so vorgehen.«

Wegweiser
Eine neue Area für eine gigantische Anforderung, S. 242

Priti fährt fort: »Es scheint mir, dass wir einen neuen Bereich für diese große Überraschung benötigen. Und natürlich hat das großen Einfluss auf unsere bestehenden, hoch priorisierten Ziele, da wir einige Teams verschieben müssen. Bereiten wir uns also auf eine ausgiebige Diskussion über die Gesamtpriorisierung innerhalb der nächsten Tage vor. Aber jetzt freue ich mich auf eure Ideen, wie wir den neuen Bereich aufziehen können.«

Nach einer kurzen Diskussion wird jedem klar, wie wichtig es ist, einen neuen Bereich zu schaffen.

Priti sagt dann: »Portia, ich weiß, du bist neu bei uns, aber fühlst du dich in der Lage, die Verantwortung des Area Product Owners dafür zu übernehmen?«

Portia nickt.

Wegweiser
Ein führendes Team, S. 333

Priti fährt fort: »Peter, glaubst du, dass die Zombies mit dieser Arbeit beginnen könnten? Sie müssen sich in das Thema Dodd-Frank einarbeiten und die Auswirkungen auf unser System herausfinden, bevor wir weitere Teams hinzufügen können.«

Peter sagt: »Ich glaube nicht, dass wir eine andere Wahl haben.«

Priti sagt: »O.k., Portia, also derzeit haben wir ein paar Dinge in Peters Area Backlog. Zum einen diesen riesigen Eintrag, den du – glaube ich – Rest von Dodd-Frank' genannt hast, und dann noch den winzigen Eintrag, den du und die Zombies davon abgetrennt haben. Sprich mal Peter an. Er soll dir zeigen, wie man einen neuen Bereich im Product Backlog anlegt und die Einträge dorthin verschiebt.«

Priti wendet sich weiter an die Gruppe: »Der nächste Sprint startet in drei Tagen. Wir verschieben die Zombies in euren Bereich und legen

mit diesem Monster los. In ein paar Sprints werden wir euren Bereich um ein weiteres Team ergänzen müssen und dann wahrscheinlich auch so weit sein. Leute, bitte denkt über zwei wichtige Punkte nach: Erstens müssen wir ein Meeting zu den Konsequenzen der Umpriorisierung in den nächsten Tagen vorbereiten. Und zweitens, welche anderen Teams wären gute Kandidaten für den neuen Bereich.«

Sprint-Planung in der neuen Requirement Area

Jede Requirement Area führt eigene Sprint-Planungen durch, alle mehr oder weniger gleichzeitig. Portia beginnt die Sprint-Planung in ihrem neuen Bereich, indem sie den Zombies zwei unbekannte Gesichter vorstellt.

Sie sagt: »Gillian und Zak hatten regelmäßig Kontakt mit den Aufsichtsbehörden und sie werden uns helfen, diesen Brocken auszugestalten. Sie haben sich einverstanden erklärt, uns sowohl bei der Planung zu helfen als auch bei den PBR-Meetings und – sofern möglich – im Tagesgeschäft während der nächsten Sprints.«

Sie fährt fort: »Hier ist mein vorläufiger Plan, wie wir die nächsten beiden Sprints in Angriff nehmen können: Im ersten Schritt müssen wir gemeinsam mehr über Dodd-Frank erfahren und darüber hinaus das Thema in einige wichtige und überschaubare Stücke aufteilen, damit wir den Nebel etwas lichten können und ein besseres Gespür für die Prioritäten bekommen.

Im zweiten Schritt setzen wir das erste kleine Stückchen in diesem Sprint um. Das gibt uns mehr Informationen über die tatsächliche Arbeit und die Auswirkungen auf unser Produkt. Und wir sehen gleich konkrete Fortschritte.

Im dritten Schritt bereiten wir uns auf weitere Teams vor, die zu unserem Bereich dazustoßen werden. Was haltet ihr von diesem Ansatz? Habt ihr andere Vorschläge?«

Während der kurzen Diskussion sagt Mario zu seinem Team: »Lasst mich euch ein bisschen mehr Kontext geben. Ich habe unser Team im jüngsten Product-Owner-Team-Meeting mit den anderen Area Product Ownern und Priti vertreten. Zunächst liegt es an uns, das Thema zu beginnen. Wir gehen in Vorleistung mit einer frühen Implementierung, bekommen dadurch einen ersten Überblick über das Thema und verstehen damit die Gesamtauswirkung auf unsere Architektur.«

Wegweiser
Ein führendes Team, S. 333

Michelle unterbricht: »Wie ein Tiger-Team, das an einem neuen Produkt arbeitet?«

»Ja, genau so«, sagt Mario. »Denkt an die Unterstützung von Dodd-Frank wie an ein neues Produkt, das kontinuierlich in den Rest des Produkts integriert werden muss. Das Problem ist: Wir sind in Eile und es ist eine Menge Arbeit, sodass ein weiteres Team in ein paar Sprints dazustoßen wird und kurz danach wahrscheinlich zwei weitere Teams. Wir werden zwar weiter entwickeln, aber wir werden das *Führungsteam* sein, was bedeutet, dass wir die anderen Teams zum Fliegen bringen und das gesamte Produkt im Auge behalten müssen.«

Michelle sagt: »Das hört sich für mich an, als wären wir das Architektur- und Projektmanagementteam!«

Mario lacht: »Nein. Damit bin ich durch. Wir sind immer noch ein normales Feature-Team, aber neben der Entwicklung konzentrieren wir uns halt auf die Betreuung der neuen Teams und helfen ihnen, so schnell wie möglich Fahrt aufzunehmen. Aber wir behalten im Hinterkopf: Teamkoordination und Management liegen noch immer in der Verantwortung jedes einzelnen Teams.«

Der erste Sprint in der neuen Requirement Area

Wegweiser
Nimm nur ein Stückchen, S. 223

Wegweiser
Umgang mit gigantischen Anforderungen, S. 243

Ihr erster Sprint ist zwar eine ungewöhnliche Mischung aus Klärung von Sachverhalten und tatsächlicher Entwicklung, aber in dieser Extremsituation dennoch sehr nützlich. Sie verbringen fast die Hälfte des Sprints gemeinsam mit Portia, Gillian und Zak mit der Klärung der Aufgaben. Das liegt vor allem daran, dass es selbst für dieses kleine Stückchen notwendig ist, vieles zu untersuchen, sich einzulesen, zu diskutieren und mit Außenstehenden zu kommunizieren, um zu verstehen, was im obskuren Reich der neuen Verordnung gewünscht ist, ohne einen direkten Zugriff auf die Politiker und die Verfasser der Verordnung zu haben. Sie gehen davon aus, dass in den kommenden Sprints der Anteil an Zeit für Klärungen auf die typischen 10 % oder 15 % fallen wird.

Und so wenden sie auch nur etwa die Hälfte des Sprints für die Entwicklung *eines* kleinen Features auf. Aber die Diskussionen und das Lernen durch und über das frühe Coding zahlen sich aus. Langsam, aber sicher beginnen sie, Dodd-Frank auseinanderzudröseln – zumindest die Teile, die sie verstehen können.

Bei der Umsetzung dieses kleinen Features, das sie sich zuerst geschnappt haben, verbringen sie einen Großteil der Zeit gemeinsam am Whiteboard, um die Auswirkungen auf das Gesamtdesign des Systems zu diskutieren. Das Team wechselt häufig hin und her zwischen dem Code und der Wand.

Sprint-Review in der neuen Requirement Area

Die gesamte Wertpapierproduktgruppe arbeitet zusammen in einem Sprint, um ein finales auslieferbares Produktinkrement zu erstellen. Allerdings hält jede Requirement Area ihr eigenes Sprint-Review ab, alle mehr oder weniger parallel.

Während des Reviews in ihrem Bereich schauen sich Portia, Gillian und Zak die eine »fertige« Funktionalität an, die die Zombies fertiggestellt und in das Gesamtprodukt integriert haben. Sie hatten zwar ursprünglich zwei Einträge prognostiziert, aber Portia ist dennoch beeindruckt, dass sie einen geschafft haben, vor allem wenn man bedenkt, wie kurzfristig ihnen diese neue Aufgabe übergeben wurde.

Der zweite Sprint

Im zweiten Sprint kommen sie mit der Feature-Entwicklung etwas besser voran, obwohl sie immer noch viel Zeit damit verbringen, gemeinsam mit Portia, Gillian und Zak Fachlichkeiten zu klären.

In der Mitte des Sprints halten sie gemeinsam mit dem zweiten Team, das dem Bereich demnächst beitreten soll, einen PBR-Workshop ab. Dabei führen sie das neue Team in das Thema Dodd-Frank ein. Außerdem findet ein Einführungsworkshop zur aktuellen Architektur statt, um dem neuen Team die wichtigsten, vorhandenen Gestaltungselemente vorzustellen.

Wegweiser
Workshop zur aktuellen Architektur, S. 327

Die Zombies wissen, wie groß die anstehende Arbeit ist, und freuen sich auf weitere Unterstützung.

Product-Owner-Team-Meeting

Nach ein paar Sprints ist es mal wieder Zeit für das auf den nächsten Sprint konzentrierte Product-Owner-Team-Meeting. Sie nutzen es einerseits, um sich zwischen den verschiedenen Area Product Ownern gegenseitig abzustimmen und zu koordinieren, und andererseits, damit Priti die Richtung vorgeben kann.

Wegweiser
Product-Owner-Team-Meeting, S. 305

Die Area Product Owner tauschen sich über ihre jeweilige Situation und ihre nächsten Ziele aus. Als Portia an der Reihe ist, sagt sie: »Was keinen von uns überraschen wird, ist, dass der Fortschritt nur gering und die Überraschungen enorm sind. Aber der Nebel lichtet sich und die Teams und ich können uns langsam in die Arbeit eindenken. Gillian und Zak waren hier eine enorme Hilfe.«

Pablo, der Area Product Owner für Anlagenservice, kommentiert, dass er bei einigen Punkten enge Verflechtungen zwischen ihren beiden Bereichen sähe. Portia findet es auch sinnvoll, sich im Anschluss mit Pablo und einigen Teamstellvertretern darüber auszutauschen.

Priti fragt: »Portia, wie sehen deine Ziele für unseren bevorstehenden Sprint aus?«

Hinzufügen eines dritten Teams

Beim Koordinationstreffen der Product Owner zwei Sprints später sagt Priti: »Wie ihr wisst, hat Portias Bereich immer noch nur zwei Teams. Mir ist klar, dass Pablo seine sechs Teams gerne beim Thema Anlagenservice behalten möchte, aber Dodd-Frank ist mir in diesem Jahr einfach zu wichtig. Wir werden also ein Team von Pablos Bereich in Portias verschieben. Pablo, bitte such nach einem Team aus deiner Gruppe, das Lust dazu hätte, und gib Portia und mir Bescheid.«

Das Ende

Einige wichtige Punkte aus der LeSS-Huge-Geschichte:

- Der Product Owner ist verantwortlich für die Besetzung der Area Product Owner und für die Weiterentwicklung ihrer Fähigkeiten.
- Der Product Owner ist verantwortlich für die Entscheidung, ob Requirement Areas neu eingerichtet werden, wachsen oder reduziert werden sollen.
- Requirement Areas sind groß und erfordern normalerweise zwischen vier und acht Teams, aber während der Startphase können die Requirement Areas auch etwas kleiner sein, vor allem dann, wenn mit einem Team gestartet wird, das dem »Nimm ein Stückchen«-Ansatz folgt.
- Ein führendes Team arbeitet alleine, um sich an einen riesigen Brocken heranzutasten, bis es das Umfeld und die Entwicklung versteht, um dann weitere dazustoßende Teams zu coachen, die dabei helfen sollen, die restliche Arbeit zu erledigen.

Standortübergreifende Teams: Begriffe und Tipps

Als Nächstes folgt eine LeSS-Huge-Geschichte mit standortübergreifenden Teams. Aber zunächst sind einige klärende Definitionen erforderlich, da die geläufige Bezeichnung *verteilte Teams* mehrere Bedeutungen hat. Die zu klärenden Begriffe lauten wie folgt:

- **Verstreutes Team**
  Ein Team aus (z.B. sieben) Personen, die auf verschiedene Orte, verschiedene Räume, Gebäude oder Städte verstreut sind.
- **Zusammenhängendes Team**
  Ein Team, das buchstäblich am selben Tisch sitzt.

- **Standortübergreifende Teams**
  Ein zusammenhängendes Team befindet sich an einem Standort und ein weiteres an einem anderen Standort.

Zudem gibt es noch folgende Beobachtung und Empfehlung:

- Ein verstreutes Team ist selten ein *echtes Team*; es ist vielmehr eine lose miteinander verbundene Gruppe von Individuen. Die Kommunikations- und Koordinationsverluste sind höher, und die Mitglieder fühlen sich selten als Team.
- Wenn Ihre Produktgruppe 50 oder 500 Personen umfasst, sind *verstreute Teams nicht unbedingt notwendig*. Jedes Team mit einer Größe von ungefähr sieben Mitgliedern kann problemlos gemeinsam am gleichen Ort untergebracht werden. Möglicherweise befinden sich jedoch einige Teams an verschiedenen Standorten, sodass die Produktgruppe *standortübergreifende Teams* hat. Verstreute Teams sind in der Regel das Ergebnis schlechter, organisatorischer Entscheidungen und beruhen auf der Unkenntnis darüber, welche Kosten verursacht werden, wenn man keine Teams hat, die gemeinsam an einem Standort sitzen.

Regel
Jedes Team ist
1. selbstverwaltend,
2. cross-funktional,
3. zusammenhängend und
4. beständig.

LeSS-Huge-Geschichte: Standortübergreifende Teams

Portia ist Area Product Owner für eine neue Requirement Area in einem Wertpapierhandelssystem. Dieser neue Bereich startete aus Gründen der Fokussierung und Einfachheit mit nur einem Team. Ein paar Sprints später nimmt Portias Bereich ein drittes Team auf. Ihre ersten beiden Teams sitzen gemeinsam mit ihr in London. Aber ihr neues, drittes Team, *HouseDraculesti*, sitzt in Cluj in Rumänien, einem Hauptentwicklungsstandort des Unternehmens.

Warum nimmt man nicht einfach ein drittes Team vom Londoner Standort auf? Damit hätten viel Ärger und Einbußen in der Effizienz vermieden werden können, die mit der Entwicklung eines Bereichs an verteilten Standorten einhergehen. Die damit zusammenhängenden Kosten sind unter Umständen so hoch, dass das *Hinzufügen eines Teams* im Grunde genommen einem *Entfernen eines Teams* gleichkommt.

Positiv betrachtet ist Cluj lediglich zwei Zeitzonen von London entfernt, und jeder dort spricht gutes Englisch. Darüber hinaus handelt es sich um gute Entwickler mit einem abgeschlossenen Informatikstudium in einer Stadt, die viel Wert auf die langjährige, angewandte Exzellenz im Ingenieurbereich legt. Zusätzlich geht es um einen internen, dedizierten Entwicklungsstandort des Unternehmens, sodass man

erwarten kann, dass es sich um erfahrene Teams handelt, die ein fundiertes Wissen über das Produkt und sein Umfeld haben.

Und unterm Strich wollte Priti, der Product Owner, keins der anderen Londoner Teams von deren aktuellen Bereich abziehen.

Priti weiß, dass die Situation mit standortübergreifenden Teams für Portia neu ist, und rät ihr von daher beim nächsten Meeting: »Bitte frage deinen Scrum Master, ob er sich mit Sita unterhalten kann, und bitte Sita, euch bei einigen eurer Events zu coachen. Sie ist Scrum Master im Team ›Anlagenservice‹ und sammelt seit ein paar Jahren Erfahrungen in ihrem Bereich mit der Entwicklung an verteilten Standorten. Sie weiß, wie wichtig es ist, dass Scrum Master bei ihren Teams sitzen, und sie hat bereits viele Meetings über mehrere Standorte begleitet.«

Priti führt des Weiteren aus: »Da wir ein super erfolgreiches Jahr hatten, stelle ich dir und dem Zombies-Team – zumindest denjenigen, die reisen können – die nötigen Mittel bereit, so früh wie möglich einen Sprint in Cluj zu verbringen. Arbeitet eng zusammen, alle in einem Raum. Natürlich könnte das Team aus Cluj hierher nach London kommen, aber es wäre ein starkes Signal an die Personen am dortigen Standort, dass *sie* wichtig sind und ihr Standort auch. Versuche den Eindruck zu vermeiden, dass London wichtiger ist als Cluj. Ah, und du solltest sie mindestens alle paar Monate mal besuchen.«

Standortübergreifende Sprint-Planung 1

Wegweiser
Sprint-Planung 1, S. 298

Ein paar Sprints später betritt Portia den Raum. Es gibt einen Laptop mit angeschlossenem Beamer, über den der Raum in Cluj zu sehen ist. Das gesamte Team in Cluj sitzt dort zusammen und wartet. Sita schlägt vor, dass während der ersten paar Monate, in denen das Team noch neu im Bereich ist, das gesamte Team in Cluj an standortübergreifenden Meetings teilnimmt, um den Lernprozess und das Engagement zu fördern.

Alle Stellvertreter der Teams haben Tablets oder Laptops dabei.

Portia eröffnet das Meeting: »Herzlich willkommen! Lasst uns anfangen. Mein Vorschlag für die Einträge dieses Sprints sind in der bereits verteilten Tabelle markiert. Könnt ihr das alle sehen? Ich gehe davon aus, dass ihr wisst, weshalb diese Einträge wichtig sind und im Mittelpunkt stehen – wir haben sie ja im letzten PBR besprochen und ich habe unsere gemeinsamen Punkte eingearbeitet. Aber fragt gerne nochmal nach, wenn euch etwas unklar ist. Wenn es keine Fragen gibt, dann würde ich euch bitten, eure Teamnamen neben den Einträgen einzugeben, die ihr für euer Team mitnehmen wollt.«

Danach verbringt die Gruppe einige Zeit damit, letzte Verständnisfragen bezüglich der Einträge zu klären. Die Stellvertreter der Londoner Teams beginnen, die Fragen auf Flipchart-Papier aufzuschreiben. Die Teammitglieder in Cluj tragen ihre Fragen in einer Tabelle ein, auf die alle Personen zugreifen können. Portia nimmt sich die Zeit, die Fragen auf den verschiedenen Flipcharts zu besprechen, Antworten zu diskutieren und auf Papier zu skizzieren. Darüber hinaus beantwortet sie die Punkte aus der Tabelle und hält dies elektronisch für das Team in Cluj fest, während sie sich per Video mit ihnen darüber unterhält.

Nach rund 30 Minuten sind die einzelnen Fragen beantwortet und Portia ruft alle wieder zusammen. Sie fragt: »Hat jemand noch Fragen, die wir gemeinsam besprechen müssen, bevor wir die Punkte zusammenfassen?«

Standortübergreifendes Gesamt-PBR

Die Teilnehmer betreten den Workshopraum in London. Zwei Beamer sind aufgebaut – einer zeigt das Videobild des Workshopraums in Cluj, der andere einen Browser auf Portias Rechner.

Portia sagt: »Lasst uns loslegen. Ich möchte mich auf das Schneiden einiger Einträge konzentrieren. Ich habe Zak eingeladen, der sich sehr gut damit auskennt.«

Zak beginnt einige Verzweigungen in einem grafischen, browserbasierten Mindmapping-Programm anzulegen, während er mit der Gruppe diskutiert.

Wegweiser
Arten von Product Backlog Refinements, S. 269

Wegweiser
PBR mit verteilten Standorten, S. 274

Im Anschluss daran diskutieren sie anhand einer für alle geteilten Tabelle und legen für jeden neu erstellten Themenblock einen Eintrag an, damit alle Beteiligten auf beiden Seiten ein möglichst leichtgewichtiges, aber konkretes Verständnis über die Details bekommen. Später schätzt die Gruppe die neuen Einträge, indem sie extra große Planning-Poker-Karten verwendet, die man auch trotz Kamera und Video am anderen Standort gut sehen kann, wenn man sie hoch hält.

Das Ende

Ein paar der wesentlichen Punkte aus der LeSS-Huge-Geschichte über verteilte Standorte:

- Bei Teams an verteilten Standorten entstehen häufig sowohl offensichtliche als auch subtile Spannungen und Kosten, die in ihrer negativen Auswirkung überraschend groß sind.
- Punkte, die die Reibung mit anderen Standorten reduzieren können, sind die geografische Lage in einer ähnlichen Zeitzone, dass der Standort zum Unternehmen gehört oder dediziert an dem Produkt arbeitet, sich alle Entwickler fließend in der gleichen Sprache unterhalten können und eine Kultur, die auf langfristige und praktische Exzellenz der Entwickler Wert legt.
- Ein Scrum Master muss mit seinen Teams zusammen sitzen.
- Jeder Standort muss sich ebenbürtig fühlen – nicht als Standort zweiter Klasse.
- Die verschiedenen Standorte müssen regelmäßig besucht werden und sich gegenseitig befruchten.
- Meetings sollten von Angesicht zu Angesicht mithilfe von Videoprogrammen durchgeführt werden.
- Die Verwendung von Werkzeugen zum gemeinsamen Bearbeiten von Dokumenten über Standortgrenzen hinweg macht es für jeden einfach, gleichzeitig und gemeinsam daran zu arbeiten.

Ausblick

Frage nicht: »Wie können wir im Großen *agil arbeiten* in unserer komplexen und schwierigen Organisation?« Stelle eine andere, tiefgreifendere Frage: »*Wie können wir die Organisation vereinfachen und **agil sein**, anstatt agil zu arbeiten*?« Und da eine ernsthafte Skalierung von Scrum damit beginnt, die Organisation zu ändern, anstatt Scrum zu ändern, konzentriert sich der nächste Teil des Buches auf das Verständnis und die Einführung einer einfacheren, kundenorientierten LeSS-*Organisation*.

Darauf folgen weitere Abschnitte, die sich mit einem kundenorientierteren *Produkt* und *Sprint* in einer einfacheren LeSS-Organisation beschäftigen.