

# **Flow Metrics for Scrum Teams**

**German Edition**

# Warum dieses Buch

Dieses Buch entstand aus Slack-Diskussionen zwischen Dan und Will über viele der Dinge, die Dan an Scrum nicht mochte. Er ist immer noch nicht der größte Fan, aber er wird es immer mehr. Wir wollten einen nützlichen Guide für eine datengesteuerte Arbeitsverwaltung erstellen und hoffentlich einigen Teams dabei helfen, vom Story Point Theater wegzukommen. Vor allem aber hoffen wir, dass diese Tools Ihrem Team helfen, weniger Zeit mit der Planung und Diskussion von Plänen zu verbringen und mehr Zeit mit der Entwicklung großartiger Dinge.

## Dank

Die Autoren bedanken sich bei James Scrimshire für die Gestaltung des Covers. Ein weiterer Dank geht an unsere Korrektoren Colleen Johnson, Stephan Vlieland, Stas Pavlov, Prateek Singh und Frank Steeneveld für ihr Feedback. Wir haben versucht, alle Witze über Jira zu entfernen. Wir mögen es trotzdem nicht.

## Übersetzung

Dieser Guide wurde aus der englischen Originalversion übersetzt, die von der oben genannten Entwicklergruppe zur Verfügung gestellt wurde. An der Übersetzung haben mitgewirkt:

2023: Christoph Dibbern (Erstübersetzung)

# Kapitel 1 - Fangen wir an

Bevor Sie überhaupt an die Anwendung von Arbeitsflussmetriken in Ihrer Scrum Implementierung denken können, müssen Sie ein paar Dinge vorbereiten. Leider schweigt sich der Scrum Guide über diese wichtigen Punkte weitgehend aus, so dass wir sie hier etwas ausführlicher erläutern müssen. Um die Sache ein wenig interessanter zu machen, wollen wir den Rest dieses Kapitels in ein Trinkspiel verwandeln. Jedes Mal, wenn wir von nun an "Flow" oder "Workflow" sagen, nehmen Sie einen Schluck. Holen Sie sich also Ihr Lieblingsgetränk und lassen Sie uns fortfahren.

## Was ist Flow?

Der Grund für die Existenz Ihres Scrum Teams ist die Schaffung von Wert für Ihre Kunden/ Stakeholder. Wert entsteht jedoch nicht einfach auf magische Weise. Es muss ständig daran gearbeitet werden, potenzielle Produktverbesserungen in greifbaren Kundenwert zu verwandeln. Die Schritte, die erforderlich sind, um eine Produktverbesserungsidee in etwas Konkretes zu verwandeln, das unsere Kunden als wertvoll empfinden, werden als Prozess bezeichnet.

Sie haben sich für Scrum als Framework/Rahmenwerk entschieden, auf dem Sie Ihren Prozess für die Wertschöpfung aufbauen können. Es ist ein häufiges Missverständnis, dass Scrum selbst ein Prozess ist. Ist es aber nicht. Es ist ein Framework, innerhalb dessen Sie einen Wertschöpfungsprozess erstellen und kontinuierlich verbessern. Ob Sie es wissen oder nicht, Sie und Ihr Team haben einen Wertschöpfungsprozess entwickelt, der weit über Scrum hinausgeht. Dieser Prozess kann explizit oder implizit sein, aber er existiert.

Wir können die Bedeutung dieses Konzepts gar nicht hoch genug einschätzen, denn das Verständnis des Prozesses ist die Grundlage für das Verständnis von Flow. Wenn Ihr Prozess etabliert ist, dann ist Flow einfach definiert als die Bewegung von potenziellem Wert durch diesen Prozess.

**Flow:** die Bewegung des potenziellen Wertes durch einen bestimmten Prozess.

Vielleicht haben Sie schon einmal von der anderen Bezeichnung für einen Prozess gehört, die als Workflow bekannt ist. Es gibt einen Grund, warum er WorkFLOW genannt wird. Denn was bei jedem Prozess wirklich zählt, ist der Fluss der Arbeit.

**Hinweis:** Ich werde die Begriffe "Prozess", "Workflow" und "System" oft synonym verwenden. Ich werde mein Bestes tun, um einen Unterschied zwischen diesen Begriffen aufzuzeigen, wenn ein Unterschied gerechtfertigt ist. Für die meisten Zusammenhänge in diesem Buch ist der Unterschied zwischen diesen Begriffen jedoch vernachlässigbar, so dass sie problemlos synonym verwendet werden können. -Dan

Der Grund, warum Sie sich für Flow interessieren sollten, liegt darin, dass Ihre Fähigkeit, Flow in Ihrem Prozess zu erreichen, darüber entscheidet, wie effektiv, effizient und vorhersehbar Sie als Team sind, wenn es darum geht, einen Mehrwert für den Kunden zu schaffen - was, wie wir zu Beginn festgestellt haben, der eigentliche Grund ist, warum Sie hier sind.

# Sich einrichten / bereit machen, um Flow zu messen

Die Teams, die wir sehen, die mit Scrum kämpfen, kämpfen nicht wirklich mit Scrum. Sie kämpfen mit Ownership oder sie kämpfen mit Flow. Bei Ersterem können wir Ihnen nicht helfen, aber bei Letzterem schon. Zum Glück verfügt Flow über eine Reihe grundlegender Metriken, die uns einen hervorragenden Einblick in den Zustand und die Leistung unseres Prozesses geben. Bevor wir jedoch darüber sprechen können, welche Metriken wir verwenden sollten, müssen wir zuerst darüber sprechen, was wir brauchen, um diese Metriken zu berechnen.

Alle Metriken sind Messungen, und alle Messungen haben dasselbe gemeinsam: einen Start- und einen Endpunkt. Bei Flow ist es nicht anders. Um Flow zu messen, müssen wir daher wissen, was es bedeutet, wenn die Arbeit in unserem Prozess begonnen wurde, und was es bedeutet, wenn die Arbeit in unserem Prozess beendet wurde.

## Kapitel 1 - Fangen wir an

Der Einstieg in die Flow Metriken / Arbeitsflussmetriken bedeutet zunächst, dass wir einen klar definierten Punkt haben, an dem die Arbeit begonnen wird, und einen klar definierten Punkt, an dem die Arbeit beendet ist.

Die Entscheidung über Anfang und Ende mag trivial erscheinen, aber wir können Ihnen versichern, dass sie es nicht ist. Denn hier werden die Dinge in Scrum sehr viel nuancierter (wenn nicht geradezu kompliziert), und der Scrum Guide bietet als Guide für ein Framework und nicht für einen Prozess wenig bis gar keine wirkliche Anleitung.

An dieser Stelle denken Sie vielleicht, dass Sie in Ihrem Kontext bereits einen etablierten Arbeitsablauf wie To Do -> Doing -> Done haben. Sie könnten weiter annehmen, dass "begonnen" einfach bedeutet, dass ein Element von "Zu erledigen" zu "Erledigen" übergeht, und dass "abgeschlossen" bedeutet, dass ein Element von "Erledigen" zu "Fertig" übergeht. Das ist zwar technisch nicht falsch, aber es gibt ein paar Probleme mit dieser Denkweise.

Erstens handelt es sich bei den Objekten / Work Items, die Teams durch einen To Do -> Doing -> Done-Prozess verfolgen, in der Regel um Aufgaben. Mit anderen Worten: Die meisten Teams nehmen ein Product Backlog Item (PBI) und unterteilen es in Aufgaben wie "Analysieren, Erstellen, Testen, Prüfen, Beheben". Sie verfolgen dann den Fortschritt dieser Aufgaben während des Sprints. Dies ist potenziell problematisch, denn - wie bereits mehrfach erwähnt - geht es bei Flow speziell um die Bereitstellung von Wert. In Scrum wird der Wert jedoch auf der PBI-Ebene und nicht auf der Aufgabenebene definiert. Das heißt, um Flow Metriken in unserem Scrum Kontext zu verwenden, müssen wir zunächst definieren, was "begonnen" und "beendet" für ein PBI bedeutet.

**Anmerkung:** Die Aufteilung von Elementen in Aufgaben ist ein guter Ansatz und kann für ein brandneues Team sicherlich hilfreich sein, um die Art von Arbeit besser zu verstehen, die für die Erledigung einer Aufgabe erforderlich sein wird. In der Tat lassen sich die meisten Arbeitsabläufe daraus ableiten, welche Aufgaben bei jedem PBI wiederholt werden. Aber irgendwann wurde dies von einer guten Idee für einige zu dem weit verbreiteten Mythos, dass dies ein obligatorischer Teil von "Scrum" ist. Das ist es aber nicht. Das ist es, was uns die Möglichkeit gibt, die Kanban Strategie innerhalb von Scrum zu nutzen. -Will

Oberflächlich betrachtet mag es so aussehen, als ob die letztere Entscheidung - das Setzen eines Fertigstellungspunktes - in Scrum ziemlich einfach wäre: Sie müssen nur Ihren Fertigstellungspunkt dem zuordnen, was Ihre Definition of Done (DoD) erfordert. Mit anderen Worten: In den meisten Fällen ist der Endpunkt in Ihrem Scrum Workflow immer dann

## Kapitel 1 - Fangen wir an

erreicht, wenn Sie ein oder mehrere PBIs zu Ihrem Inkrement hinzugefügt haben und dieses Inkrement Ihre Definition of Done (DoD) erfüllt hat.

Was aber, wenn Ihre anfängliche DoD nicht beinhaltet, dass (a) das Inkrement ausgeliefert wurde, (b) der einmal ausgelieferte Wert validiert wurde oder (c) das Wert-Feedback in die künftige Entscheidungsfindung einbezogen wird? Und was ist, wenn unser Kunde in jedem dieser Fälle Antworten auf die Frage "wann wird es fertig sein" haben möchte? Die Scrum Lösung wäre, die DoD Ihres Scrum Teams so zu ändern, dass sie diese zusätzlichen Kriterien enthält (denken Sie daran, dass Ihr Scrum Team die in der DoD der Organisation festgelegte Qualität jederzeit anheben kann - Sie können sie nur nicht unter den organisatorischen Standard senken). Organisatorische Einschränkungen können Sie jedoch daran hindern, die DoD Ihres Teams so zu ändern, dass sie einen viel breiteren Umfang hat (z. B. darf Ihr Team keinen Code in die Produktion überführen). Das Schöne an Scrum ist, dass es in diesem Fall nicht restriktiv, sondern ziemlich flexibel ist.

Example Workflow				
Options	Discovery	Building	Validating	Done

Abbildung 1.1 - Beispiel-Workflow

Bei der Verwendung von Scrum hindert Sie nichts daran, einen Arbeitsablauf zu definieren, bei dem die Arbeit beendet ist, nachdem die vorgegebene DoD erfüllt wurde. (An dieser Stelle kommen wir zur Diskussion über "Done" vs. "Done Done" vs. "Done Done Done" usw.) Nehmen wir zum Beispiel an, dass das Scrum Team im obigen Arbeitsablauf (Optionen -> Entdeckung/Discovery -> Aufbau -> Validierung -> Done) nur am Schritt "Aufbau" beteiligt ist und dass die DoD nur Inkremente betrifft, die aus der Aufbauphase hervorgehen (und wir sagen nicht, dass das eine gute Idee ist). Wenn das der Fall wäre, dann steht nirgendwo in Scrum, dass nur gemessen werden soll, was in der Bauphase passiert. Nichts hindert Sie daran, bis zum Done oder bis zur Validierung oder einfach nur bis zum Building zu messen. Für diejenigen, die mit dem Kanban Konzept der Definition des Workflows (DoW) vertraut sind, wollen wir damit sagen, dass Ihre DoW Ihre DoD abdecken sollte, aber darüber hinausgehen kann und wahrscheinlich auch wird.

Das Gleiche gilt für die Wahl des Startpunkts. Nichts hindert Sie daran, einen Startpunkt zu wählen, an dem die Arbeit beginnt, bevor das Scrum Team tatsächlich ins Spiel kommt. Das sollte eine sehr gute Nachricht für Sie sein, denn wann beginnt ein Scrum Team eigentlich mit der Arbeit? Beginnt die Arbeit erst, wenn die Entwicklung innerhalb des Sprints beginnt? Beginnt die Arbeit, wenn ein PBI für das Sprint Backlog ausgewählt wurde? Beginnt die

## Kapitel 1 - Fangen wir an

Arbeit, wenn die Verfeinerung/Refinement eines PBIs beginnt? Beginnt die Arbeit vor der Verfeinerung? Wir sind der Meinung, dass dies alles großartige Fragen sind, auf die es in verschiedenen Kontexten sehr unterschiedliche Antworten geben kann, die nicht unbedingt richtig oder falsch sind. Das Schöne an Scrum ist, dass es keine Einschränkungen bei der Wahl des Startpunktes gibt.

Es wird noch ein wenig komplizierter, denn in Flow ist es durchaus erlaubt, mehr als einen Startpunkt und mehr als einen Endpunkt innerhalb eines bestimmten Workflows zu haben. Im obigen Beispiel möchten wir vielleicht sowohl den Startpunkt "Discovery" als auch "Building" und den Endpunkt "Validating" und "Done" messen. Warum wir das tun sollten, hängt davon ab, welche Fragen Sie auf der Grundlage der gesammelten Metriken beantworten wollen. Vielleicht wollen wir zum Beispiel wissen, wie lange es dauert, bis ein Element fertiggestellt ist, nachdem wir mit der Arbeit daran begonnen haben (Entdeckung bis Fertigstellung), und wie lange unser Erstellungsschritt dauert (Erstellung bis Validierung). Alle diese Permutationen sind zulässig.

Es sind nicht nur verschiedene Permutationen erlaubt, sondern es wird auch empfohlen, dass Sie mit verschiedenen Start- und Endpunkten in Ihrem Prozess experimentieren, um Ihren Kontext besser zu verstehen. Sie werden schnell feststellen, dass Sie durch die Änderung der Definition von "begonnen" und "beendet" sehr unterschiedliche Fragen zum Flow in Ihrem Prozess beantworten können. Wenn alles gut geht, wird die Erweiterung der Start- und Endpunkte Sie auf den Weg zu echter geschäftlicher Agilität / Business Agilität bringen.

## Fazit

Ihre Aufgabe als Scrum Team ist es, Ihren Kunden einen Mehrwert zu liefern. Mit Scrum haben Sie sich für ein sehr spezifisches Framework entschieden, auf dessen Grundlage Sie einen Prozess für die Wertlieferung implementieren.

Wenn Sie das noch nicht getan haben, müssen Sie sich mit Ihrem Team zusammensetzen und für Ihren Prozess entscheiden, was es bedeutet, dass ein PBI begonnen hat und was es bedeutet, dass ein PBI beendet ist. Alle anderen Gespräche zum Arbeitsfluss / Flow hängen von Ihrer Entscheidung über die Grenzen ab.

Die Bewegung des potenziellen Wertes zwischen den definierten Start- und Endpunkten wird als Flow bezeichnet. Das Konzept der Bewegung ist von entscheidender Bedeutung, denn das

## Kapitel 1 - Fangen wir an

Letzte, was wir als Team wollen, ist, eine ganze Reihe von PBIs zu beginnen, sie aber nie abzuschließen. Das ist die Antithese zur Wertschaffung. Hinzu kommt, dass unsere Kunden uns bei unserer Arbeit ständig Fragen stellen werden (ob es uns gefällt oder nicht), wie z. B. "wie lange" oder "wie viele" - Fragen, für deren Beantwortung unser Verständnis von Bewegung erforderlich ist.

Hier kommen Flow Metriken / Arbeitsflussmetriken ins Spiel.



# Kapitel 2 - Die grundlegenden Flow Metriken / Arbeitsflussmetriken

Die vier für den Flow zu verfolgenden Messgrößen sind:

**WIP:** Die Anzahl der begonnenen, aber nicht abgeschlossenen PBIs.

**Durchlaufzeit:** Die Zeit, die zwischen dem Start eines Work Items und dem Abschluss eines Work Items verstreicht.

**Work Item-Alter:** Die Zeit, die zwischen dem Start eines Work Items und dem aktuellen Zeitpunkt verstreicht.

**Durchsatz:** Die Anzahl der abgeschlossenen PBIs pro Zeiteinheit. Beachten Sie, dass die Messung des Durchsatzes die genaue Anzahl der PBIs ist.

Im letzten Kapitel haben wir viel Zeit damit verbracht, über die Bedeutung eines genau definierten Zeitpunkts zu sprechen, an dem die Arbeit begonnen wird, und eines genau definierten Zeitpunkts, an dem die Arbeit beendet wird. Der Grund für diese Bedeutung ist, dass alle grundlegenden Metriken des Arbeitsflusses in Bezug auf diese Start- und Endpunkte definiert sind. Auch hier gilt, dass Scrum selbst nicht verlangt, dass diese Start- und Endpunkte über die Sprint Grenzen hinaus formell identifiziert werden, aber wenn Sie Arbeitsflussmetriken in einem Scrum Kontext auf einer granulareren Ebene verwenden wollen, müssen Sie sich die Mühe machen, diese Punkte in Ihrem Prozess zu definieren. Wir gehen davon aus, dass Ihr Team diese Punkte für unsere weitere Diskussion bereits definiert hat.

Ein letztes Wort der Vorsicht: Diese vier Messgrößen stellen nur das Minimum dar, das erforderlich ist, um Erkenntnisse über den Ablauf in Scrum zu gewinnen. Teams und

Organisationen können und sollten oft zusätzliche kontextspezifische Messgrößen verwenden, die dabei helfen, datengestützte Entscheidungen zu treffen<sup>1</sup>. Fühlen Sie sich frei, diese zusätzlichen Maße in Ihrem Kontext zu verwenden, solange es sich nicht um Story Points handelt. Ja, das war (hauptsächlich) Sarkasmus.

# Laufende Arbeiten

Die allgemeine Definition für **WIP** in einem bestimmten Flusskontext lautet: Alle diskreten Einheiten mit potenziellem Kundenwert, die in einen bestimmten Prozess eingetreten sind, ihn aber noch nicht verlassen haben. In einem Scrum Kontext werden die diskreten Arbeitseinheiten als Product Backlog Items (PBIs) bezeichnet; um den WIP zu berechnen, zählen Sie also einfach die PBIs innerhalb Ihrer Prozessgrenzen, wie oben definiert. Das war's: Zählen Sie einfach die PBIs und Sie haben den WIP berechnet.

Ihre erste Frage könnte lauten: "Wie passt die Komplexität in die WIP Berechnung?" Die kurze Antwort ist, dass sie es nicht tut. Dies ist wahrscheinlich das am schwersten zu begreifende Konzept für Menschen, denen beigebracht wurde, dass die Kapazität eine Funktion der PBI Komplexität ist. Das ist sie nicht. Die Arbeitsflussprinzipien verlangen in keiner Weise, dass Sie die relative Komplexität der Elemente, die Ihren Prozess durchlaufen, verstehen. Wir werden im nächsten Kapitel erklären, warum das so ist, aber für den Moment bitten wir Sie, Ihre Ungläubigkeit zu unterdrücken und dies einfach als wahr zu akzeptieren. Die sehr gute Nachricht ist, dass Sie, wenn Sie die Schätzung in Story Points hassen, diese Praktik mit der Einführung von Flow Metriken sofort aufgeben können. Aber dazu etwas später mehr.

Ihr nächster Einwand könnte lauten: "Wenn die Komplexität keine Rolle spielt, dann spielt die Größe der PBIs sicherlich eine Rolle" Schließlich sind die PBIs, die Ihren Prozess durchlaufen, von sehr unterschiedlicher Größe. Wie können Sie all diese Unterschiede berücksichtigen und ein vorhersehbares System entwickeln, indem Sie nur die PBIs zählen? Das ist zwar eine berechtigte Frage, aber man sollte sich nicht daran aufhängen. Wie bei der Komplexität gibt es auch beim Flow keine Notwendigkeit, die Größe im Voraus abzuschätzen, abgesehen von einem kurzen Gespräch über die Service Level Expectation / Service Level Erwartung (erklärt in Kapitel 6), wenn Sie eine Arbeit / ein Work Item ziehen. Aber dazu später mehr, wenn wir über die Sprint Planung sprechen.

Wenn Sie Kanban bereits im Scrum Kontext verwenden, sollten Sie auch beachten, dass es einen Unterschied zwischen WIP und WIP-Limits gibt. Sie können den WIP nicht einfach

berechnen, indem Sie alle WIP-Limits auf Ihrem Board zusammenzählen. So sollte es zwar funktionieren, aber in Wirklichkeit ist es nicht so. Dieses Ergebnis sollte offensichtlich sein, da die meisten Kanban Boards nicht immer Spalten oder Boards haben, deren WIP-Limit voll ausgeschöpft ist. Häufiger kommt es vor, dass eine Kanban Board Verletzungen des WIP-Limits in mehreren Spalten - oder auf dem gesamten Board - aufweist. In beiden Fällen führt die einfache Addition der WIP-Grenzwerte nicht zu einer genauen WIP-Berechnung. Die traurige Wahrheit ist, dass man nicht darum herumkommt, die physische Anzahl der in Arbeit befindlichen Arbeiten / Work Items zu zählen, um den Gesamt-WIP zu ermitteln.

Fazit: Wenn Sie den Arbeitsfluss optimieren wollen, aber den WIP noch nicht erfassen, sollten Sie damit anfangen. Früher ist besser als später.

# Durchlaufzeit

Im vorigen Abschnitt haben wir festgestellt, dass ein Prozess bestimmte Ankunfts- und Abgangsgrenzen hat und dass jede Arbeit / Work Item mit Kundenwert zwischen diesen beiden Grenzen als WIP gezählt werden kann. Sobald Ihr Team die Abgrenzungspunkte bestimmt hat, die den Arbeitsfortschritt definieren, wird die Definition der Durchlaufzeit sehr einfach:

**Durchlaufzeit:** Die verstrichene Zeit, die eine Arbeit / Work Item als Work in Progress verbringt.

Diese Definition basiert auf derjenigen von Hopp und Spearman in ihrem Buch *Factory Physics<sup>2</sup>* und stimmt genau mit der zu Beginn dieses Kapitels gegebenen Definition überein. Die Definition der Durchlaufzeit anhand des WIP beseitigt einen Großteil - wenn nicht alle - der Willkürlichkeit / Beliebigkeit einiger anderer Erklärungen der Durchlaufzeit, die Sie vielleicht gesehen haben (und die Sie verwirrt haben), und gibt uns eine genauere Definition für die Messung dieser Kennzahl. Die Moral von der Geschichte ist: Sie haben im Wesentlichen die Kontrolle darüber, wann etwas in Ihrem Prozess als laufende Arbeit gezählt wird. Nehmen Sie sich etwas Zeit, um die Richtlinien / Policies zu definieren, die festlegen, was ein Element in Ihrem System als "in Arbeit befindlich" bedeutet, und starten und stoppen Sie Ihre Durchlaufzeituhr entsprechend.

Sie sollten auch die Bedeutung der "verstrichenen Zeit" nicht übersehen. Die Verwendung der verstrichenen Zeit unterscheidet sich wahrscheinlich stark von den Anleitungen, die Sie bisher

## Kapitel 2 - Die grundlegenden Flow Metriken / Arbeitsflussmetriken

erhalten haben. Die meisten anderen Methoden fordern Sie auf, nur die Zeit zu messen, die Sie tatsächlich für die aktive Arbeit an einer bestimmten Aufgabe aufwenden (wenn Sie überhaupt aufgefordert werden, die Zeit zu messen). Dies wird manchmal als Touch Time bezeichnet. Wir sind der Meinung, dass diese Anleitung falsch ist. Und wir haben einige Gründe dafür.

Erstens, und das ist der wichtigste Grund, denken Ihre Kunden wahrscheinlich in Form von verstrichener Zeit. Nehmen wir zum Beispiel an, Sie teilen Ihren Kunden am 1. März mit, dass etwas in 30 Tagen erledigt sein wird. Wir gehen davon aus, dass Ihre Kunden erwarten, dass sie ihre Arbeit / Work Item am oder vor dem 31. März erhalten. Wenn Sie jedoch 30 "Geschäftstage" gemeint haben, dann erwarten Sie, dass der Kunde die Ware irgendwann Mitte April erhält. Sie können sich sicher vorstellen, dass dieser Unterschied in den Erwartungen ein Problem darstellen könnte.

Zweitens: Wenn Sie nur die aktive Zeit messen, lassen Sie einen großen Teil Ihres Arbeitsflussproblems außer Acht. In der Zeit, in der eine Arbeit wartet oder sich verzögert (d. h. nicht aktiv bearbeitet wird), liegt in der Regel der größte Teil Ihrer Unvorhersagbarkeit. Und genau in diesem Bereich wollen wir die Vorhersagbarkeit am stärksten verbessern. Denken Sie daran: Verzögerung ist der Feind des Arbeitsflusses!

Es gibt noch einen weiteren wichtigen Grund, die Durchlaufzeit zu verstehen. Die Durchlaufzeit ist auch die Zeit, die benötigt wird, um eine Rückmeldung zu dem zu erhalten, was Sie in Ihrem Arbeitsablauf liefern. Im Idealfall entspricht die Cycle Time der Zeit, die Sie benötigen, um Kundenfeedback zu erhalten, wenn "Fertig" in Ihrem Arbeitsablauf "Freigegeben" bedeutet. Kundenfeedback ist in unserer wissensbasierten Arbeitswelt von entscheidender Bedeutung. Der Wert selbst wird letztendlich vom Kunden bestimmt, was bedeutet, dass Ihr Team sicherstellen möchte, dass es dieses Feedback so schnell wie möglich erhält. Das Letzte, was Sie wollen, ist etwas zu entwickeln, das der Kunde nicht braucht - vor allem, wenn Sie dafür ewig brauchen. Wenn "Fertig" in Ihrem Arbeitsablauf nicht "Freigegeben" bedeutet, können Sie immer noch ein Feedback vom nächsten Schritt erhalten, der nachgelagert ist. In beiden Fällen verkürzt eine Verkürzung der Durchlaufzeit die (im Idealfall kundenbezogene) Feedbackschleife. Und um die Durchlaufzeit zu verkürzen, müssen Sie sie zunächst messen.

# Alter der Arbeitsaufgabe

Das Alter ist bei weitem die wichtigste aller zu verfolgenden Arbeitsflussmetriken. Aus diesem Grund wird es in Kapitel 4 dieses Buches sehr ausführlich behandelt (was vielleicht der Grund ist, warum Sie überhaupt angefangen haben, dieses Buch zu lesen). Die Definition von Work Item Age lautet:

**Work Item Age:** die Gesamtzeit, die verstrichen ist, seit ein Work Item / eine Arbeit in einen Workflow eingetreten ist.

In der Literatur wird das Work Item Age manchmal auch als "WIP Age" bezeichnet. Wir werden versuchen, in diesem Buch konsistent zu sein und nur den Begriff "Work Item Age" zu verwenden, aber Sie sollten wissen, dass beide Bezeichnungen austauschbar sind.

Da das Alter von Arbeitsaufgaben ein Maß für die aktuelle Zeit in Bearbeitung für alle Ihre laufenden Arbeiten ist, gilt es - per Definition - nur für Aufgaben, die in den Workflow eingetreten sind, ihn aber nicht verlassen haben. Sobald ein Element den Workflow verlässt, wird das gesamte bis dahin angesammelte Alter sofort in Durchlaufzeit umgewandelt.

## Durchsatz

Die am einfachsten zu definierende Metrik kommt zum Schluss. Einfach ausgedrückt, ist der Durchsatz definiert als:

**Durchsatz:** die Menge an WIP (Anzahl der PBIs), die pro Zeiteinheit abgeschlossen wird.

Etwas anders ausgedrückt: Der Durchsatz ist ein Maß dafür, wie schnell die Elemente einen Prozess verlassen. Die Zeiteinheit, die Ihr Team für die Messung des Durchsatzes wählt, liegt ganz bei Ihnen. Ihr Team kann die Anzahl der Elemente messen, die es pro Tag, pro Woche, pro Sprint usw. erledigt. Sie könnten den Durchsatz Ihres Systems zum Beispiel mit "drei Elementen pro Tag" (für einen bestimmten Tag) oder "fünf PBIs pro Monat" (für einen bestimmten Monat) angeben.

Ein weiterer wichtiger Punkt zum Thema Durchsatz ist, dass viele agile Coaches und Berater die Begriffe "Velocity" und "Throughput" synonym verwenden. Velocity kann zwar mit den

gleichen Begriffen wie Throughput definiert werden, aber wenn ein Coach "Velocity" sagt, meint er in der Regel "Story Points pro Sprint". Wenn sie in Form von Story Points definiert werden, sollten Sie wissen, dass Durchsatz und Velocity alles andere als synonym sind.

Wenn der Durchsatz angibt, wie schnell Elemente einen Prozess verlassen, dann gibt die Ankunftsrate an, wie schnell Elemente in einem Prozess ankommen. Wir erwähnen diese Tatsache hier, weil die Ankunftsrate je nach Sichtweise als Analogon zum Durchsatz betrachtet werden kann. Nehmen wir zum Beispiel an, dass der Schritt "Entwicklung" und der Schritt "Test" in Ihrem Arbeitsablauf nebeneinander liegen. Dann könnte man den Durchsatz aus dem Schritt "Entwicklung" auch als die Ankunftsrate im Schritt "Test" betrachten.

Noch wichtiger ist jedoch, dass der Vergleich der Ankunftsrate eines Schritts in Ihrem Prozess mit dem Durchsatz eines anderen Schritts, Ihnen einige dringend benötigte Einblicke in Probleme der Vorhersagbarkeit geben kann. Auf diesen Vergleich werden wir in den nächsten Kapiteln noch genauer eingehen. Unser unmittelbarer Grund für die Erörterung der Ankunftsrate ist jedoch, darauf hinzuweisen, dass es genauso wichtig sein kann, wie schnell die Dinge in Ihrem Prozess ankommen und wie schnell sie ihn verlassen.

Die Durchsatzmetrik beantwortet die sehr wichtige Frage: "Wie viele PBIs werde ich in der nächsten Version erhalten?" Irgendwann werden Sie diese Frage beantworten müssen, also verfolgen Sie den Durchsatz und seien Sie vorbereitet.

# Datenerfassung und -berechnung

Metrikdefinitionen sind schön und gut, aber sie sind bedeutungslos, wenn Sie nicht wissen, welche Daten Sie sammeln oder wie Sie die einzelnen Metriken aus diesen Daten berechnen sollen.

Was die Datenerfassung betrifft, so zahlt es sich endlich aus, wenn wir Sie auffordern, Anfangs- und Endpunkte zu definieren. Erfassen Sie einen Zeitstempel, wenn ein PBI Ihren Startpunkt überquert, und einen weiteren Zeitstempel, wenn dasselbe PBI Ihren Zielpunkt überquert. Führen Sie dies für jedes PBI durch, das Ihren Prozess durchläuft, wie in Abbildung 2.1 dargestellt (verzeihen Sie die amerikanischen Datumsangaben):

## Kapitel 2 - Die grundlegenden Flow Metriken / Arbeitsflussmetriken

Abbildung 2.1: Zeitstempel-Date

<b>PBI ID</b>	<b>Started</b>	<b>Finished</b>
1	01/01/2022	01/03/2022
2	02/02/2022	03/03/2022
3	01/02/2022	03/04/2022
4	01/03/2022	
5	01/04/2022	

Das war's. Für die Berechnung aller grundlegenden Arbeitsflussmetriken sind dies die einzigen Daten, die Sie benötigen. Noch besser ist es, wenn Sie ein PBI-Tracking-Tool zur Unterstützung Ihres Teams verwenden, dann sammelt Ihr Tool höchstwahrscheinlich bereits alle diese Daten für Sie.

Der Nachteil bei der Verwendung eines Tracking-Tools ist jedoch, dass Sie sich nicht auf die Berechnungen verlassen können, die das Tool für Sie bereithält. Es ist eines der großen Geheimnisse des Universums, warum viele Agile Tools die Arbeitsflussmetriken nicht richtig berechnen können, aber in den meisten Fällen können sie es nicht. Um die einzelnen Metriken korrekt aus den Daten zu berechnen, gehen Sie wie folgt vor:

### **WIP**

WIP ist die Anzahl aller PBIs, die einen Startzeitstempel / Startzeitmarke aber keinen Endzeitstempel für einen bestimmten Zeitraum haben. Der letzte Teil ist für die meisten Menschen etwas schwer zu begreifen. Obwohl WIP technisch gesehen eine Momentanmetrik / unmittelbare Metrik ist - d.h. Sie könnten jederzeit alle PBIs in Ihrem Prozess zählen, um den WIP zu berechnen - ist es in der Regel hilfreicher, über den WIP in einer bestimmten Zeiteinheit zu sprechen: Tage, Wochen, Sprints usw. Unsere dringende Empfehlung - und das wird unsere dringende Empfehlung für alle diese Metriken sein - ist, dass Sie den WIP pro Tag verfolgen. Wenn wir also wissen wollen, wie hoch unser WIP an einem bestimmten Tag war, würden wir einfach alle PBIs zählen, die an diesem Tag begonnen, aber noch nicht abgeschlossen waren. In Abbildung 2.1 beträgt unser WIP am 5. Januar 3 (die PBIs 3, 4 und 5 wurden alle vor dem 5. Januar begonnen, aber an diesem Tag noch nicht abgeschlossen).

### **Durchlaufzeit**

## Kapitel 2 - Die grundlegenden Flow Metriken / Arbeitsflussmetriken

Die Durchlaufzeit ist gleich dem Fertigstellungsdatum minus dem Startdatum plus eins ( $CT = FD - SD + 1$ ).

Wenn Sie sich fragen, woher das "+ 1" in der Berechnung kommt, liegt das daran, dass wir jeden Tag, an dem der Posten bearbeitet wird, als Teil der Gesamtsumme zählen. Wenn zum Beispiel ein PBI am selben Tag beginnt und endet, würden wir niemals sagen, dass es null Zeit in Anspruch genommen hat. Wir addieren also einen Tag hinzu und runden so den Teiltag auf einen vollen Tag auf. Was ist mit Aufgaben, die nicht am selben Tag beginnen und enden? Nehmen wir zum Beispiel an, ein Vorgang beginnt am 1. Januar und endet am 2. Januar. Die obige Definition der Durchlaufzeit würde eine Antwort von zwei Tagen ergeben ( $2 - 1 + 1 = 2$ ). Wir halten dies für ein vernünftiges, realistisches Ergebnis. Aus der Sicht des Kunden bedeutet die Angabe einer Durchlaufzeit von einem Tag, dass er realistischerweise davon ausgehen kann, dass er sein fertiges Work Item noch am selben Tag erhält. Wenn wir den Kunden zwei Tage angeben, haben sie die realistische Erwartung, dass sie das fertige Work Item am nächsten Tag erhalten usw.

Sie werden vielleicht befürchten, dass die obige Berechnung der Durchlaufzeit zu sehr auf die Messung der Durchlaufzeit in Tagen ausgerichtet ist. In Wirklichkeit können Sie jeden beliebigen Begriff von "Zeit" verwenden, der für Ihren Kontext relevant ist (deshalb haben wir bis jetzt immer gesagt, dass wir einen "Zeitstempel" und kein "Datum" verfolgen). Vielleicht sind Wochen für Ihre spezielle Situation relevanter. Oder Stunden. Oder sogar Sprints. Wenn Sie für Ihr Scrum Team die Durchlaufzeit in Form von Sprints messen wollten, dann wäre die Berechnung einfach Beendeter Sprint - Start Sprint + 1 (unter der Annahme, dass PBIs in Ihrem Kontext die Sprintgrenzen überschreiten). Der Punkt hier ist, dass diese Berechnung in allen Kontexten anwendbar ist. Wie beim WIP empfehlen wir jedoch nachdrücklich, die Durchlaufzeit in Tagen zu berechnen. Die Gründe dafür sind zu zahlreich, um sie hier zu erörtern. Berechnen Sie daher die Durchlaufzeit zu Beginn in Tagen und experimentieren Sie später mit anderen Zeiteinheiten, falls Sie sie benötigen (wir gehen davon aus, dass Sie das nicht tun werden).

### **Work Item Alter / Alter einer Arbeit**

Das Work Item Age ist gleich dem aktuellen Datum minus dem Startdatum plus eins ( $Age = CD - SD + 1$ ).



## Kapitel 2 - Die grundlegenden Flow Metriken / Arbeitsflussmetriken

Das Argument "plus eins" ist dasselbe wie bei der Durchlaufzeit oben. Wir bitten um Entschuldigung, aber Sie werden nie ein PBI mit einem Alter von null Tagen haben. Auch hier empfehlen wir dringend, das Alter in Tagen zu messen.

### Durchsatz

Werfen wir einen Blick auf einen anderen Datensatz, um unser Beispiel für die Durchsatzberechnung ein wenig zu verdeutlichen:

Abbildung 2.2 - Beispielhafte Prozessdaten

Work Item ID	Arrived	Departed
1	01/01/2022	03/01/2022
2	01/02/2022	03/03/2022
3	02/02/2022	03/03/2022
4	01/02/2022	03/04/2022
5	03/02/2022	03/04/2022

Um den Durchsatz zu berechnen, notieren Sie zunächst das früheste Datum, an dem ein Element abgeschlossen wurde, und das späteste Datum, an dem ein Element abgeschlossen wurde. Dann werden diese Daten aufgezählt. In unserem Beispiel lauten diese Daten der Reihe nach:

Abbildung 2.3 - Aufeinanderfolgende Kalendertage zwischen dem ersten und dem letzten abgeschlossenen Element

Completed Date
03/01/2022
03/02/2022
03/03/2022
03/04/2022

## Kapitel 2 - Die grundlegenden Flow Metriken / Arbeitsflussmetriken

Zählen Sie nun für jedes aufgezählte Datum einfach die Anzahl der Arbeiten / Work Items, die genau an diesem Datum abgeschlossen wurden. Für unsere Daten sehen diese Zählungen wie folgt aus:

Abbildung 2.4 - Berechneter Durchsatz

Completed Date	Throughput
03/01/2022	1
03/02/2022	0
03/03/2022	2
03/04/2022	2

Aus Abbildung 2.4 geht hervor, dass wir am 01.03.2016 einen Durchsatz von 1 Work Item / Arbeit, am nächsten Tag 0 Arbeiten, am dritten Tag 2 Arbeiten und am letzten Tag 2 Arbeiten hatten. Beachten Sie den Durchsatz von Null am 02.03.2016 - an diesem Tag wurde nichts erledigt.

Wie bereits erwähnt, können Sie für die Berechnung des Durchsatzes beliebige Zeiteinheiten wählen. In Scrum könnte Ihre erste Neigung darin bestehen, den Durchsatz pro Sprint zu berechnen: "Wir haben im letzten Sprint 14 PBIs erledigt". Wir raten Ihnen dringend davon ab und raten Ihnen dringend, den Durchsatz in Tagen zu messen. Auch hier wäre es ein Buch für sich, um zu erklären, warum, aber lassen Sie uns nur zwei kurze Begründungen anführen: (1) die Verwendung von Tagen bietet Ihnen eine viel bessere Flexibilität und Granularität, wenn wir mit Dingen wie Monte-Carlo-Simulationen für Planungsaktivitäten beginnen (ausführlich erklärt in Kapitel 5); und (2) die Verwendung einheitlicher Einheiten für alle Ihre Metriken erspart Ihnen eine Menge Kopfschmerzen. Wenn Sie also WIP, Durchlaufzeit und Alter in Tagen erfassen, wird Ihr Leben sehr viel einfacher, wenn Sie auch den Durchsatz in Tagen erfassen. Aus diesen Daten können Sie ganz einfach den Durchsatz pro Sprint ableiten, falls dies für Sie noch von Bedeutung ist.

## Zufälligkeit

Den schwierigsten Teil haben wir uns für den Schluss aufgehoben. Sie wissen jetzt, wie Sie die vier grundlegenden Arbeitsflussmetriken auf der Ebene der einzelnen PBIs berechnen können.

Außerdem wissen wir jetzt, dass alle diese Berechnungen deterministisch sind. Das heißt, wenn wir ein PBI am Montag beginnen und sie einige Tage später am Donnerstag abschließen, dann wissen wir, dass die PBI eine Durchlaufzeit von genau vier Tagen hatte.

Aber was ist, wenn uns jemand fragt, wie lang unsere Gesamtprozessdurchlaufzeit ist? Was ist, wenn uns jemand fragt, wie hoch der Durchsatz unseres Scrum Teams ist? Wie sollen wir diese Fragen beantworten?

Wir vermuten, dass Sie das Problem hier sofort erkennen. Wenn wir uns beispielsweise die Durchlaufzeit unseres Teams für die letzten sechs Sprints ansehen, werden wir feststellen, dass wir PBIs in einer großen Bandbreite von Zeiten abschließen. Manche in einem Tag, manche in fünf Tagen, manche in mehr als 14 Tagen, usw. Kurz gesagt, es gibt keine einzige deterministische Antwort auf die Frage "Was ist unsere Prozessdurchlaufzeit?". Anders ausgedrückt: Die Durchlaufzeit Ihres Prozesses ist keine eindeutige Zahl, sondern eine Verteilung möglicher Werte. Das liegt daran, dass die Durchlaufzeit Ihres Prozesses eine so genannte Zufallsvariable ist. (Übrigens haben wir in diesem Abschnitt nur zur Veranschaulichung über die Durchlaufzeit gesprochen, aber alle grundlegenden Arbeitsflussmetriken (WIP, Durchlaufzeit, Alter, Durchsatz) sind allesamt Zufallsvariablen).

Was Zufallsvariablen sind und warum Sie sich dafür interessieren sollten, gehört zu den Themen, die den Rahmen dieses Buches sprengen würden. Was Sie jedoch wissen müssen, ist, dass Ihr Prozess von Ungewissheit und Risiko geprägt ist, was bedeutet, dass alle Arbeitsflussmetriken, die Sie verfolgen, diese Ungewissheit und dieses Risiko widerspiegeln, und dass diese Ungewissheit und dieses Risiko in allen Ihren Flussmetrikberechnungen als Zufallsvariablen auftauchen werden. Im weiteren Sinne bedeutet dies, dass Sie den Determinismus aus dem Fenster werfen können, sobald der Zufall auftaucht. Sobald Sie wissen, dass Sie es mit einem Zufallsprozess zu tun haben, müssen Sie einen probabilistischen Ansatz wählen. Zum Glück für uns ist probabilistisches Denken das Thema des nächsten Kapitels.

# Schlussfolgerung

Was wir hier gezeigt haben, sind nur die grundlegenden Metriken des Arbeitsflusses, um Ihnen den Einstieg zu erleichtern: WIP, Durchlaufzeit, Work Item Age und Durchsatz. Es gibt sicherlich noch weitere Kennzahlen, die Sie in Ihrer eigenen Umgebung verfolgen möchten, aber diese Kennzahlen sind allen Arbeitsflussimplementierungen gemeinsam. Wenn Ihre Ziele

Verbesserung und Vorhersagbarkeit sind, dann sind dies die Kennzahlen, die Sie verfolgen wollen.

# Quellen

1. Coleman, John and Vacanti, Daniel S. "The Kanban Guide" <https://kanbanguides.org>, 2020.
2. Hopp, Wallace J., and Mark L. Spearman. Factory Physics. Irwin/McGraw-Hill, 2007.

# Kapitel 3 - Probabilistisches Denken

Diejenigen unter Ihnen, die Das Buch "When Will It Be Done?"<sup>1</sup> gelesen haben, können dieses Kapitel getrost überspringen, ohne dass die Kontinuität verloren geht. Für alle anderen gilt: Bitte lesen Sie weiter, denn die folgenden Konzepte sind grundlegend, um mit Hilfe von Arbeitsflussmetriken das Beste aus Scrum herauszuholen.

## Das Ende des Determinismus

Für diejenigen unter Ihnen, die in der Welt nach der Pandemie noch pendeln, stellt sich die Frage, wie lange Sie morgen früh brauchen werden, um zur Arbeit zu kommen. Das scheint eine ganz einfache Frage zu sein. Vielleicht denken Sie sofort an einen einzigen Wert, sagen wir 25 Minuten. Aber denken Sie einen Moment über Ihre Antwort nach. Werden Sie wirklich genau 25 Minuten brauchen? Wir vermuten, dass Ihre wahre Antwort "es kommt darauf an" lautet.

Wenn Sie mit dem Auto zur Arbeit fahren, welche Faktoren könnten Ihre Fahrzeit beeinflussen? Verkehr, Straßenarbeiten, Unfälle, Abfahrtszeiten - all das kann einen Unterschied machen. Wenn Sie mit dem Zug zur Arbeit fahren, kann Ihre Fahrzeit davon abhängen, wie stark die Züge ausgelastet sind, ob auf den Strecken gearbeitet wird oder ob gelegentlich ein Arbeitskampf stattfindet.

Nehmen wir an, wir würden damit beginnen, Ihren täglichen Weg zur Arbeit zu messen. Wenn wir den Beginn Ihres Arbeitsweges als den Moment definieren, in dem Sie aus Ihrer Wohnungstür treten, und das Ende Ihres Arbeitsweges als den Moment, in dem Sie durch die Tür Ihres Arbeitsplatzes treten (wo haben wir schon einmal genau definierte Start-/Endpunkte gesehen?), dann müssen wir einfach die Zeitstempel an diesen beiden Punkten sammeln und eine Tabelle wie die unten Abgebildete erstellen:

## Kapitel 3 - Probabilistisches Denken

Abbildung 3.1 - Beispielhafte Pendlerdaten

Day	Start Time	End Time
1	07:17	07:43
2	07:35	08:03
3	07:22	07:44
4	07:44	08:58
5	07:12	07:37

Wie lange würde diese Person angesichts der Beispieldaten in Abbildung 2.1 morgen für den Weg zur Arbeit brauchen? Nehmen Sie sich einen Moment Zeit, um Ihre Antwort zu formulieren. Wenn Sie sie haben, lassen Sie mich raten, wie Sie zu Ihrer Antwort gekommen sind. Haben Sie die Endzeit genommen und die Startzeit abgezogen, um die Gesamtzeit für jeden Tag zu ermitteln? Haben Sie zum Beispiel an Tag 1 die Zeit von 07:43 minus 07:17 genommen, um auf eine Fahrzeit von 26 Minuten zu kommen? Das ist gut. Haben Sie dann aus allen verstrichenen Zeiten den Durchschnitt gebildet, um Ihre tägliche Fahrzeit zu ermitteln? Nicht so gut. Wenn Sie mathematisch versiert sind, haben Sie vielleicht sogar die Standardabweichung berechnet, um ein Confidence Level / Eintrittswahrscheinlichkeit zu ermitteln? Das ist noch weniger gut.

Eine Vorhersage auf eine einzige Zahl zu reduzieren, ist fragwürdig, und wenn diese einzige Zahl die Form eines Durchschnitts hat, wird die Sache in der Regel noch schlimmer (mehr zu Durchschnittswerten weiter unten).

Und warum?

Die Frage "Wie lange brauche ich morgen für den Weg zur Arbeit?" ist im Grunde genommen eine Aufforderung zu einer Prognose. Mit anderen Worten: Sie werden gebeten, die Zukunft vorherzusagen. Aber die Zukunft ist voller Ungewissheit. Wie wir gerade besprochen haben, wissen wir nicht, ob unser Arbeitsweg durch den Verkehr beeinträchtigt oder durch einen Unfall unterbrochen wird oder ob nichts Außergewöhnliches passiert. Sollte einer oder mehrere dieser Faktoren eintreten, wird Ihre tatsächliche Fahrzeit stark beeinflusst werden. Eine andere Möglichkeit, darüber nachzudenken, besteht darin, sich vor Augen zu führen, dass es, bevor Sie morgens vor die Tür treten, viele mögliche Zukunftsszenarien in Bezug auf Ihren Arbeitsweg gibt - jedes mit seiner eigenen Eintrittswahrscheinlichkeit.

Die gute Nachricht ist, dass es einen Wissenschaftszweig gibt, der sich speziell mit Ungewissheit beschäftigt (Situationen, in denen es mehrere mögliche zukünftige Ergebnisse gibt). Diese Wissenschaft nennt sich Wahrscheinlichkeitsrechnung. Wenn es um viele mögliche Ergebnisse geht, muss man einen probabilistischen Ansatz wählen - im Gegensatz zu einem deterministischen. Für mich lässt sich das Wesen des probabilistischen Denkens in einer einzigen, prägnanten Aussage zusammenfassen:

"Probabilistisch zu denken bedeutet, anzuerkennen, dass es mehr als ein mögliches zukünftiges Ergebnis gibt."

Eine einfache Möglichkeit, diese Art des Denkens zu erklären, ist das Würfeln mit zwei fairen sechsseitigen Würfeln. Es ist unmöglich, das Ergebnis eines beliebigen Wurfs mit 100-prozentiger Sicherheit vorherzusagen, bevor die Würfel gefallen sind. Das bedeutet jedoch nicht, dass das Ergebnis eines Wurfs mit zwei Würfeln völlig unvorhersehbar ist. Tatsächlich gibt es einige Dinge, die wir über jeden Wurf wissen:

- Die Wahrscheinlichkeit, eine 1 oder 13 zu würfeln, liegt bei 0 %.
- Die möglichen Ergebnisse sind jede ganze Zahl zwischen 2 und 12 (einschließlich)
- Das wahrscheinlichste Ergebnis eines jeden Wurfs ist 7
- Es gibt eine bestimmte Wahrscheinlichkeit, genau 7 zu würfeln. Die Chance, genau 7 zu erhalten, ist - nun, wissen Sie, wie hoch die Wahrscheinlichkeit dafür ist? Die Antwort wird weiter unten gegeben.

Die Erkenntnis, dass die Zukunft mit ähnlichen Begriffen beschrieben werden muss, ist genau das, was wir tun müssen, um praktischere Vorhersagen machen zu können.

# Der Fehler bei Durchschnittswerten

Vielleicht lautete Ihre Antwort auf meine Frage nach dem Arbeitsweg eher: "Im Durchschnitt brauche ich 25 Minuten für den Weg zur Arbeit". Wie oben erwähnt, haben Sie vielleicht einen Durchschnittswert für die Pendlerdaten berechnet, um die Frage zu beantworten, wie lange es dauern wird. Jedes Mal, wenn Sie eine Vorhersage in Form eines Durchschnittswerts

## Kapitel 3 - Probabilistisches Denken

kommunizieren, sind Sie dem Fehler der Durchschnittswerte (FoA, Flaw of Averages) zum Opfer gefallen.

Der Flaw of Averages (FoA)<sup>2</sup> ist ein Konzept, das in dem gleichnamigen Buch von Dr. Sam Savage beschrieben wird und sehr eng mit dem probabilistischen Denken verbunden ist. Einfach ausgedrückt, kann der Fehler des Durchschnitts als "Pläne, die auf dem Durchschnitt basieren, scheitern im Durchschnitt" bezeichnet werden.

Um FoA zu erklären, möchten wir das gleiche Beispiel verwenden, das Dr. Savage in vielen seiner Vorträge benutzt. Nehmen wir an, es findet um 9:00 Uhr ein Geschäftstreffen / Business Meeting statt, zu dem 10 Personen eingeladen sind und bei dem alle Teilnehmer anwesend sein müssen, bevor das Treffen beginnen kann. Nehmen wir weiter an, dass alle Teilnehmer im Durchschnitt pünktlich zu den Besprechungen erscheinen (für dieses Beispiel nehmen wir an, dass die durchschnittliche Pünktlichkeit bei 50 % liegt). Wie stehen die Chancen, dass die Besprechung pünktlich beginnt?

Auch hier könnte man meinen, die Antwort auf diese Frage sei einfach. Wenn im Durchschnitt alle Teilnehmer pünktlich erschienen sind, kann man davon ausgehen, dass die Besprechung mit einer durchschnittlichen Wahrscheinlichkeit pünktlich beginnen wird. Leider ist auch diese Antwort falsch. Wenn jeder Teilnehmer die gleiche Chance hat, pünktlich zu erscheinen, wie wenn er zu spät kommt, dann besteht nur eine Wahrscheinlichkeit von 0,1 %, dass die Besprechung pünktlich beginnt. Stellen Sie sich das so vor: Da jeder Eingeladene eine 50 %ige Chance hat, pünktlich zu erscheinen, könnten Sie mit dem Werfen einer Münze modellieren, ob ein bestimmter Teilnehmer pünktlich erscheinen wird - Kopf: er/sie wird pünktlich erscheinen, Zahl: er/sie wird nicht pünktlich erscheinen. Denken Sie daran, dass die Besprechung nur beginnen kann, wenn alle Teilnehmer eintreffen. Der Fall, dass die Besprechung pünktlich beginnt, ist also gleichbedeutend mit dem Umdrehen von 10 Köpfen hintereinander - nur eine Zahl zu werfen bedeutet, dass der Teilnehmer zu spät kommt und die Besprechung selbst verspätet beginnt. Die Wahrscheinlichkeit, 10 Mal hintereinander Kopf zu werfen, beträgt 0,1 % ( $1/2^{10}$ ), also etwa 1 zu 1.000. Die Chance, dass die Besprechung pünktlich beginnt, ist praktisch gleich null, was deutlich schlechter als der Durchschnitt ist.

Dieser Punkt kann durch einen Witz veranschaulicht werden, den Sie wahrscheinlich schon einmal gehört haben: "Wenn Bill Gates oder Jeff Bezos in eine Bar (Kneipe) gehen, dann ist im Durchschnitt jeder in der Bar ein Milliardär". Durchschnittswerte haben außerhalb einiger sehr spezifischer Anwendungsfälle keine große Bedeutung. Prognosen gehören nicht dazu.



### Kapitel 3 - Probabilistisches Denken

Die Lehre daraus ist, dass Sie jedes Mal hellhörig werden sollten, wenn jemand "im Durchschnitt..." sagt, denn alles, was nach der Aussage "im Durchschnitt..." folgt, hat wenig bis gar keinen Informationswert. Ein Beispiel: Dan lebt derzeit in Südflorida, und wie Sie vielleicht wissen, ist Florida ziemlich anfällig für Hurrikane. Vor jeder Hurrikansaison versuchen die Meteorologen, die Schwere der bevorstehenden Hurrikansaison vorherzusagen. Sie werden Aussagen hören wie: "Die Hurrikansaison 2022 wird aktiver sein als der Durchschnitt". Es sollte Ihnen jetzt sofort klar sein, dass diese Aussage ein klassischer Fall von FoA ist und keinerlei Informationswert enthält. Vergleicht man einen einzelnen Wert (die Hurrikansaison 2022) mit einem Durchschnittswert (dem Durchschnitt aller Hurrikansaisons in der Vergangenheit), dann würde man erwarten, dass die kommende Saison in etwa 50 % der Fälle aktiver als der Durchschnitt sein wird und in etwa 50 % der Fälle weniger aktiv als der Durchschnitt sein wird. Die Aussage, dass die Saison 2022 aktiver als der Durchschnitt sein wird, sagt uns also nicht wirklich etwas, da die Wahrscheinlichkeit, dass diese Vorhersage richtig ist, genauso groß ist wie die, dass sie falsch ist.

Ein letzter Gedanke zur FoA, bevor wir uns dringlicheren Dingen zuwenden.

Englischsprachige Menschen sind sehr faul und verwenden oft das Wort "Durchschnitt", wenn sie eigentlich "typisch" meinen. Wenn in unserem Beispiel jemand sagt: "Ich brauche im Durchschnitt 25 Minuten zur Arbeit", dann meint er wahrscheinlich wirklich: "Ich brauche normalerweise 25 Minuten zur Arbeit". Das Problem mit dieser Faulheit ist, dass der Durchschnitt bei der Wahrscheinlichkeitsrechnung (und der FoA) meist sehr weit vom typischen Ergebnis entfernt ist - und in vielen Fällen ist der Durchschnitt nicht einmal ein mögliches Ergebnis.

Zur Veranschaulichung: Wenn Sie einen einzelnen, fairen sechsseitigen Würfel werfen, ist das erwartete Ergebnis (oder, grob gesagt, das durchschnittliche Ergebnis) tatsächlich 3,5. Wir wissen aber, dass es unmöglich ist, eine 3,5 zu würfeln, so dass auch hier die Aussage "im Durchschnitt werden wir eine 3,5 würfeln" keinen Informationswert hat.

Selbst wenn der Durchschnitt im Bereich des Möglichen liegt, ist das durchschnittliche Ergebnis normalerweise nicht sehr wahrscheinlich. Nehmen wir an, wir wollen zwei faire sechsseitige Würfel werfen. Das durchschnittliche Ergebnis in diesem Szenario beträgt (mussten Sie das nachschlagen?) etwa 16,7 %. Dies ist eine Eigenart der Durchschnittswerte, mit der die meisten Menschen nur schwer zurechtkommen. Auch wenn das Würfeln einer Sieben das durchschnittliche Ergebnis ist und auch wenn in diesem Fall das Würfeln einer Sieben das wahrscheinlichste Ergebnis ist, sind die Chancen, tatsächlich eine Sieben zu würfeln, tatsächlich ziemlich gering. Um Dans Freund und Kollegen Frank Vega zu zitieren: "Das wahrscheinlichste Ergebnis ist nicht sehr wahrscheinlich". Mit anderen Worten: Würden Sie

auf etwas wetten, wenn Sie **im Durchschnitt** nur eine Chance von 16,7 % hätten, erfolgreich zu sein? Leider ist es genau das, was viele Scrum Teams tun.

# Fazit

Wie wir in den kommenden Kapiteln sehen werden, liegt die wahre Stärke von Arbeitsflussmetriken für Scrum Teams in der Anerkennung ihrer probabilistischen Natur. Nicht jedes PBI, das Ihren Prozess durchläuft, wird genau dieselbe Zeit in Anspruch nehmen; nicht in jedem Sprint werden Sie genau dieselbe Anzahl von PBIs zu Ende bringen, usw.

Wenn Sie sich probabilistisches Denken zu eigen machen, können Sie genauer planen und sind letztlich für Ihre Kunden/Stakeholder berechenbarer. Aber worüber denken wir probabilistisch und was hat probabilistisches Denken mit Arbeitsflussmetriken zu tun? Witzig, dass Sie das fragen...

# Quellen

1. Vacanti, Daniel S. "When Will It Be Done?" ActionableAgile Press, 2017.
2. Savage, Sam L. "The Flaw of Averages". John Wiley & Sons, Inc., 2009.

# Kapitel 4 - Zwei Diagramme

Bevor wir weitermachen, müssen wir herausfinden, wie zwei der vier Arbeitsflussmetriken, die Durchlaufzeit und das Work Item Age, visualisiert werden können. Die Durchlaufzeit wird in einem Diagramm dargestellt, das als Streudiagramm bekannt ist, und das Alter der Arbeitsaufgaben wird in einem (Überraschung) Diagramm über das Alter der Arbeitsaufgaben / Work Item Age Chart angezeigt. Diese beiden Diagramme werden uns nützlich sein, wenn wir im nächsten Abschnitt die Verwendung von Arbeitsflussmetriken und Scrum Events untersuchen. Wenn Sie mit den Durchlaufzeit-Streudiagrammen und dem Work Item Aging Chart vertraut sind, können Sie dieses Kapitel getrost überspringen, ohne die Kontinuität zu verlieren. Wenn Sie noch nie von diesen Diagrammen gehört haben oder Ihr Wissen auffrischen wollen, lesen Sie bitte weiter.

## Das Durchlaufzeit-Streudiagramm

Wenn Sie noch nie ein Durchlaufzeit-Streudiagramm gesehen haben, dann haben Sie jetzt die Gelegenheit dazu:

## Kapitel 4 - Zwei Diagramme

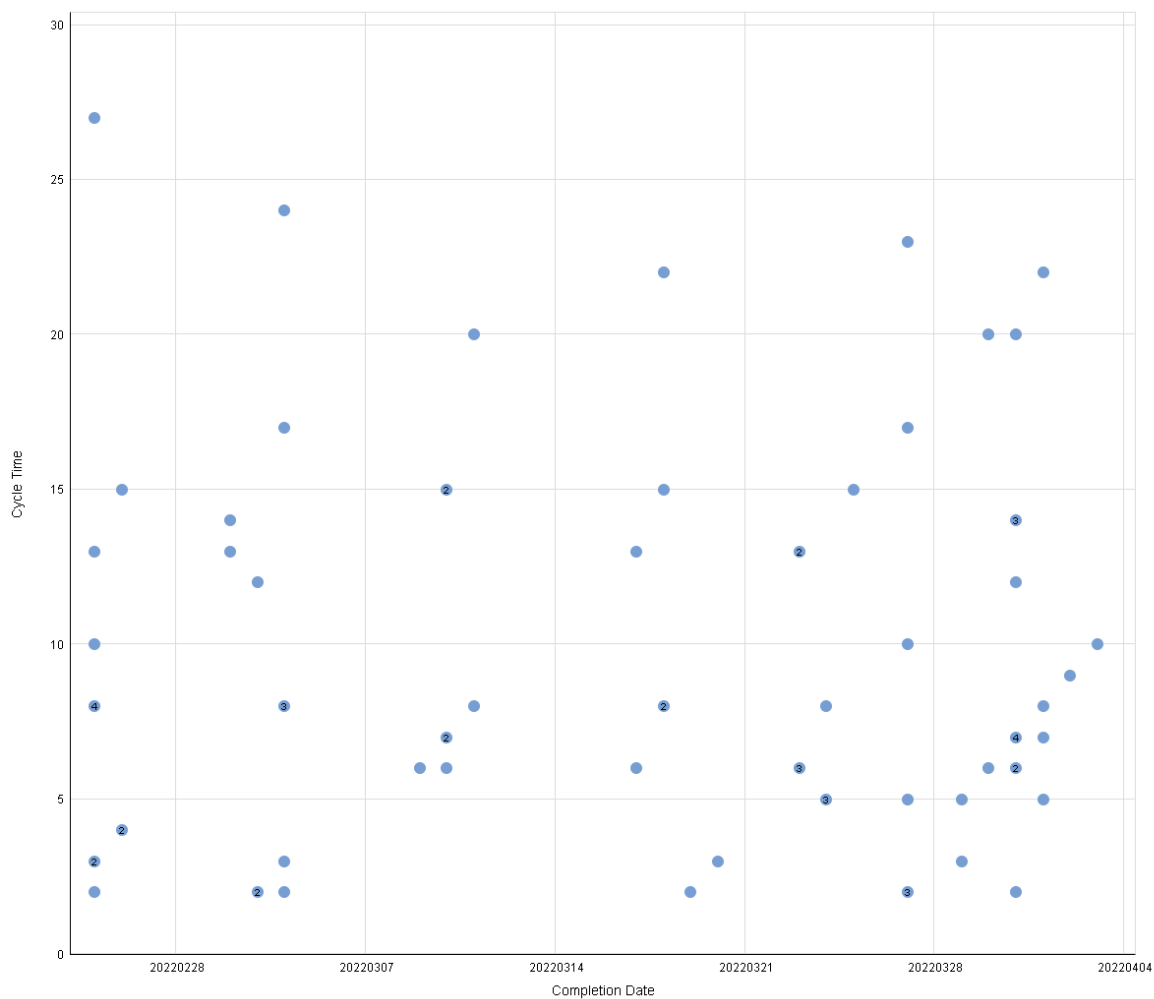


Abbildung 4.1: Ein grundlegendes Durchlaufzeit-Streudiagramm

Wie Sie in Abbildung 4.1 sehen können, befindet sich auf der unteren Seite (der X-Achse) eine Darstellung des Zeitverlaufs. Die X-Achse stellt eine Zeitachse für Ihren Prozess dar. Sie werden feststellen, dass in Abbildung 4.1 der zeitliche Verlauf von links nach rechts dargestellt ist. Dies ist keine Vorschrift, sondern nur eine Vorliebe. Alle Beispiele für Durchlaufzeit-Streudiagramme in diesem Buch zeigen jedoch einen Zeitverlauf von links nach rechts.

Auf der Y-Achse Ihres Diagramms wird eine Darstellung der Durchlaufzeit zu finden sein. Auch hier können Sie die Einheiten der Durchlaufzeit für diese Achse frei wählen: Tage, Wochen, Monate, usw.

Um ein Streudiagramm zu erstellen, suchen Sie jedes Mal, wenn ein PBI abgeschlossen ist, das Datum, an dem das PBI abgeschlossen wurde, am unteren Rand und zeichnen einen Punkt auf dem Diagramm in einer Höhe, die der Durchlaufzeit entspricht. Nehmen wir zum Beispiel an, ein Work Item hat sieben Tage gedauert und wurde am 1. Januar 2022 abgeschlossen. Im

## Kapitel 4 - Zwei Diagramme

Streudiagramm würden Sie unten den 1. Januar 2022 suchen und dann nach oben gehen und einen Punkt bei sieben Tagen setzen.

Beachten Sie, dass Sie mehrere Posten haben können, die am selben Tag mit derselben Durchlaufzeit enden. In diesem Fall würden Sie einfach die verschiedenen Punkte übereinander legen. Ich hoffe, dass das Tool, das Sie für die Erstellung des Streudiagramms verwenden, mit diesem Fall umgehen kann und Sie darüber hinaus auf die Fälle aufmerksam macht, in denen mehrere Punkte übereinander liegen.

Im Laufe der Zeit, wenn Sie mehr und mehr PBIs vollständig aufzeichnen, wird sich eine zufällige Reihe von Punkten in Ihrem Diagramm herausbilden. Ausgehend von der Diskussion im letzten Kapitel ist diese Zufälligkeit angesichts der Ungewissheit, der unsere Teams bei der Arbeit begegnen, zu erwarten. Die Zufälligkeit der Punkte ist einfach ein Ausdruck dafür, dass unterschiedliche Risiken zu unterschiedlichen Zeitpunkten für unterschiedliche PBIs realisiert werden. In Anbetracht dessen können wir das Diagramm unterteilen, um ein besseres Verständnis für das mit bestimmten Ergebnissen verbundene Risiko zu gewinnen.

**Hinweis:** Es gibt eine unterhaltsame Retrospektivenübung, die Sie mit Ihrem Team durchführen können, wenn Sie derzeit auch mit Story Points arbeiten. Versuchen Sie, eine Beziehung zwischen den geschätzten Story Points und der tatsächlichen Durchlaufzeit von PBIs herzustellen. Ihr Team könnte sich dabei sehr schwer tun. Beginnen Sie die Diskussion.  
-Will

### **Perzentil-Linien / Quantil-Linien in einem Streudiagramm**

Eine Möglichkeit, das Risiko zu segmentieren, besteht darin, Perzentil-Linien in unser Diagramm zu zeichnen. Eine Perzentil-Linie ist eine horizontale Linie, bei der ein bestimmter Prozentsatz der Punkte auf oder unter diese Linie fällt. Diese Berechnung ist in Abbildung 4.2 unten dargestellt.

## Kapitel 4 - Zwei Diagramme

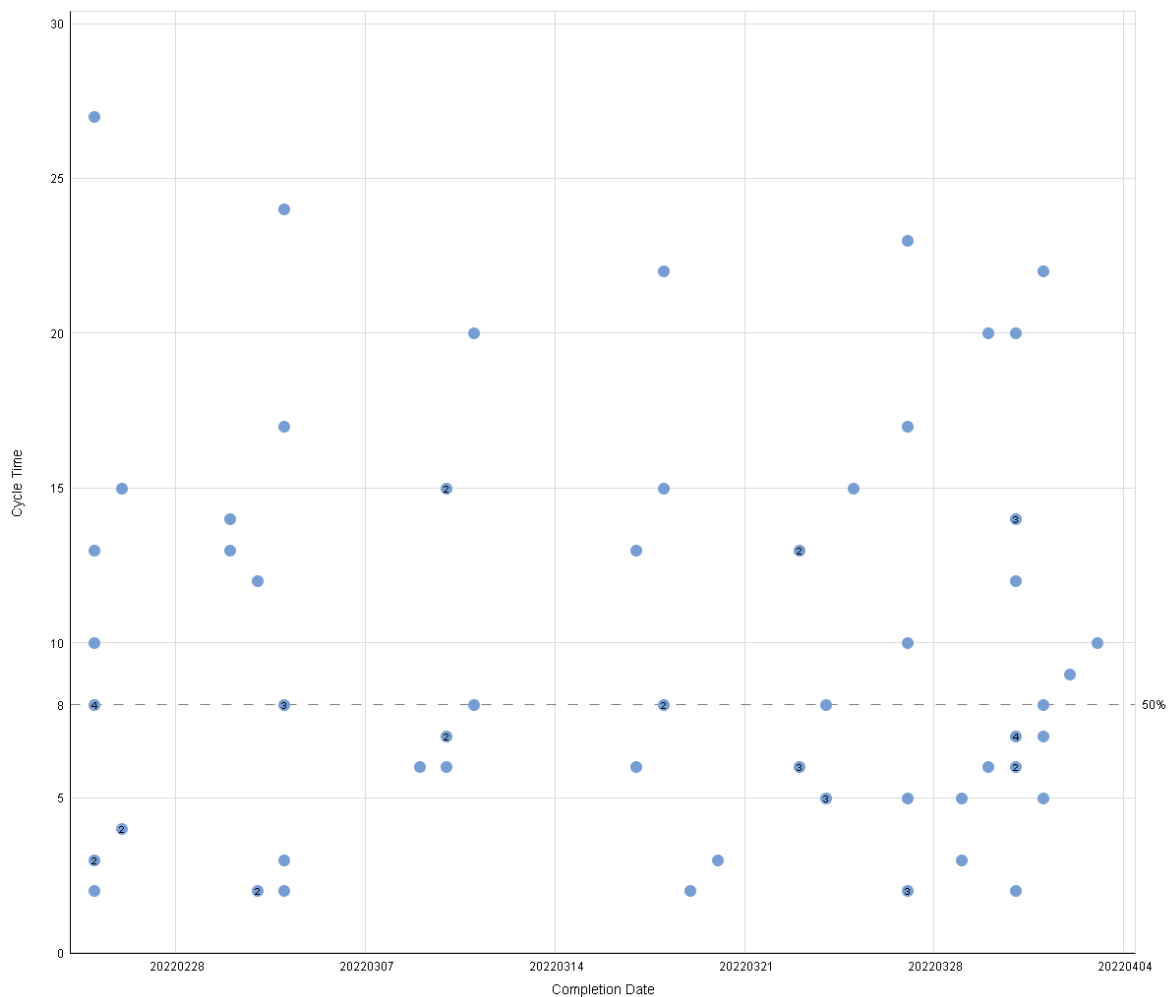


Abbildung 4.2 - Die 50<sup>te</sup> Perzentil-Linie in einem Streudiagramm

In Abbildung 4.2 liegt die 50<sup>te</sup>-Perzentillinie bei acht Tagen. Das bedeutet, dass 50 % der PBIs, die unseren Prozess durchlaufen haben, in acht Tagen oder weniger abgeschlossen wurden. Wir können also sagen, dass ein PBI, die unseren Prozess durchläuft, eine 50%ige Chance hat, in acht Tagen oder weniger abgeschlossen zu werden. Und das, ohne eine Schätzung vorzunehmen! (Mehr zu diesem Konzept in einem späteren Kapitel).

Mit dem gleichen Ansatz können wir jedes Perzentil berechnen. Ein häufig verwendetes Perzentil ist das 85<sup>te</sup>. Auch diese Linie stellt die Zeit dar, die 85 % unserer Arbeitsprozesse benötigen haben, um fertig zu werden. In Abbildung 4.3 unten sehen Sie, dass die 85<sup>te</sup>-Perzentil-Linie bei 15 Tagen liegt. Das bedeutet, dass 85 % der Punkte in unserem Diagramm auf oder unter dieser Linie liegen, und 15 % der Punkte in unserem Diagramm liegen über dieser Linie. Diese Perzentil-Linie sagt uns, dass ein Work Item, wenn es in unseren Prozess

## Kapitel 4 - Zwei Diagramme

eintritt, eine 85%ige Chance hat, in 15 Tagen oder weniger fertig zu werden, wiederum ohne Schätzung.

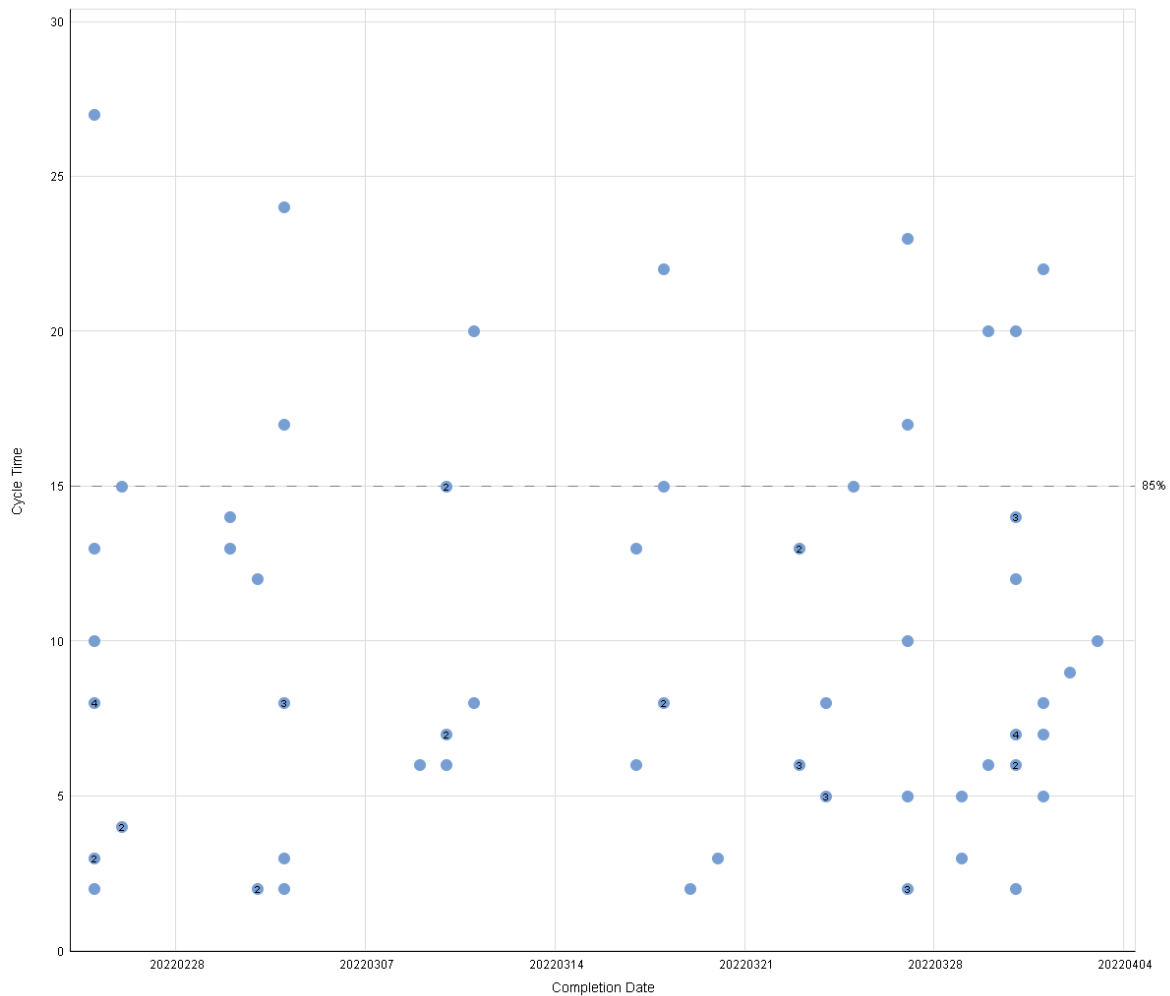


Abbildung 4.3 - Die 85<sup>th</sup>-Perzentillinie in einem Streudiagramm

Die 50<sup>th</sup>, 85<sup>th</sup> und 95<sup>th</sup> Perzentile sind wahrscheinlich die gebräuchlichsten "Standard"-Perzentile zum Einzeichnen. Da Sie aber auch andere Perzentile sehen können, haben wir in Abbildung 4.4 einige weitere hinzugefügt.

## Kapitel 4 - Zwei Diagramme

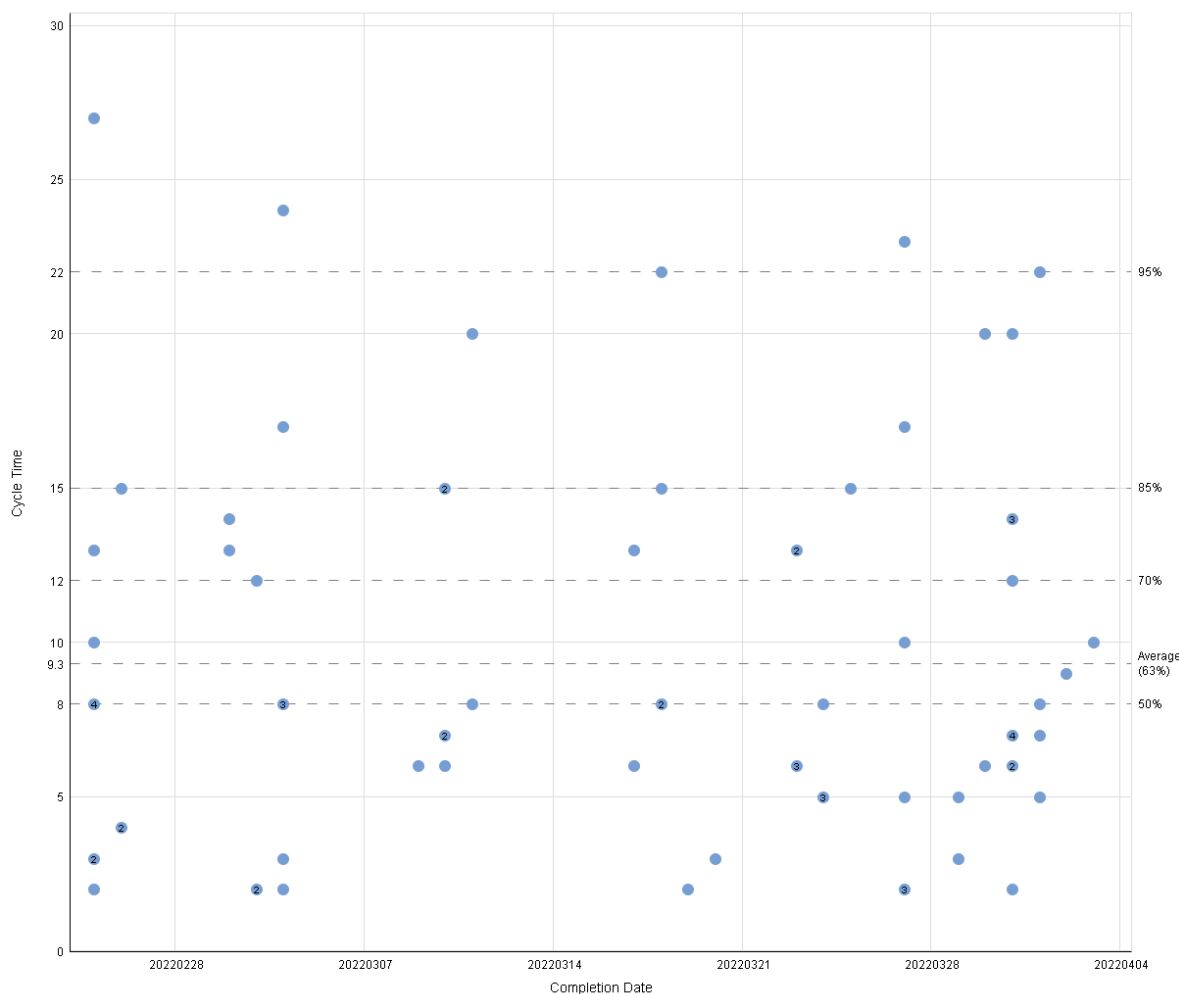


Abbildung 4.4 - 30<sup>th</sup>, 50<sup>th</sup>, 70<sup>th</sup>, 85<sup>th</sup>, 95<sup>th</sup> und mittlere Perzentil-Linien

Auch auf die Gefahr hin, dass wir uns wiederholen, dienen diese Perzentile als eine Möglichkeit, die Unsicherheit in unserem Prozess zu segmentieren. Diese Segmentierung wird benötigt, wenn wir darüber sprechen, wie wir eine Service Level Expectation / Erwartung (SLE) in unseren Scrum Events nutzen können - aber diese Diskussion muss auf den nächsten Abschnitt warten.

Achten Sie hier auf die Sprache. Ein SLE ist kein SLA: Eine Erwartung ist keine Vereinbarung und sollte daher auch nicht als solche behandelt werden. Erklären Sie unbedingt den Unterschied, wenn jemand die beiden verwechselt. Außerdem lassen viele Teams den Teil "oder weniger" weg, wenn sie über die Prozentzahlen sprechen. Dies kann zu Missverständnissen innerhalb des Teams und bei den Beteiligten führen. Sie müssen in dieser Hinsicht ein wenig pedantisch sein, bis es allgemein bekannt ist.



### **Warum Perzentile bevorzugt werden**

Es gibt mindestens drei Gründe, warum wir den Ansatz der Perzentil-Linien für die Segmentierung von Streudiagramm-Daten bevorzugen. Erstens haben wir bei der Beschreibung des Zeichnens der Standard-Perzentil-Linien in einem Streudiagramm nie erwähnt, wie die zugrunde liegenden Durchlaufzeitdaten verteilt sein könnten. Und genau das ist das Schöne daran. Um diese Linien zu zeichnen, müssen Sie nicht wissen, wie Ihre Daten verteilt sind. Die Berechnung der Perzentil-Linien funktioniert unabhängig von der zugrunde liegenden Verteilung.

Zweitens sind Perzentile extrem einfach zu berechnen. Sie zählen einfach alle Punkte zusammen und multiplizieren sie mit den Prozentsätzen. Es ist kein fortgeschrittener Abschluss in Statistik erforderlich!

Drittens: Perzentile werden nicht durch Ausreißer verzerrt. Einer der großen Nachteile des Ansatzes von Mittelwert und Standardabweichung (abgesehen von der falschen Annahme normal verteilter Daten) besteht darin, dass diese beiden Statistiken stark von Ausreißern beeinflusst werden. Wie wir bereits sagten: "Wenn Bill Gates eine Bar betritt, dann ist im Durchschnitt jeder in der Bar ein Milliardär". In diesem Fall ist der Durchschnitt natürlich keine nützliche Statistik mehr. Das gleiche Phänomen der extremen Ausreißer gibt es auch in unserer Welt. Wenn es jedoch zu diesen extremen Ausreißern bei der Durchlaufzeit kommt, werden sich die Perzentil-Linien nicht sonderlich verändern. Diese Robustheit gegenüber Ausreißern ist ein großer Vorteil der Perzentil-Linien in der Durchlaufzeitanalyse.

## **Das wichtigste Diagramm, von dem Sie noch nie etwas gehört haben (und wie Sie es verwenden können)**

Das Problem bei einem Durchlaufzeit-Streudiagramm - wenn es denn eins gibt - ist, dass ein Punkt per Definition erst dann auftaucht, wenn ein PBI abgeschlossen ist. Wenn jedoch ein PBI zu lange braucht, um fertig zu werden, ist es zu spät, zu warten, bis es fertig ist, um etwas dagegen zu tun. Was wir also brauchen, ist ein Einblick in unsere Prozessdaten, während PBIs

## Kapitel 4 - Zwei Diagramme

noch in Arbeit sind. Wir nennen diese Ansicht ein Aging Work In Progress Diagramm (oder kurz "Aging Chart").

Um ein Aging-Diagramm zu verstehen, betrachten wir einen Prozess-Workflow, der wie folgt aussieht: Analyse aktiv -> Analyse erledigt -> Entwicklung aktiv -> Entwicklung erledigt -> Testen -> Erledigt, wobei "Analyse aktiv" unser Startpunkt und "Erledigt" unser "Endpunkt" ist. Wir sagen nicht, dass dies ein guter Arbeitsablauf ist, wir verwenden dies nur zur Veranschaulichung. Wir hoffen, dass Sie bei der Gestaltung Ihres Arbeitsablaufs viel kreativer sind.

Ein Beispiel eines Diagramms für die Alterung des Arbeitsfortschritts für diesen speziellen Arbeitsablauf könnte wie in Abbildung 4.5 aussehen:

## Kapitel 4 - Zwei Diagramme

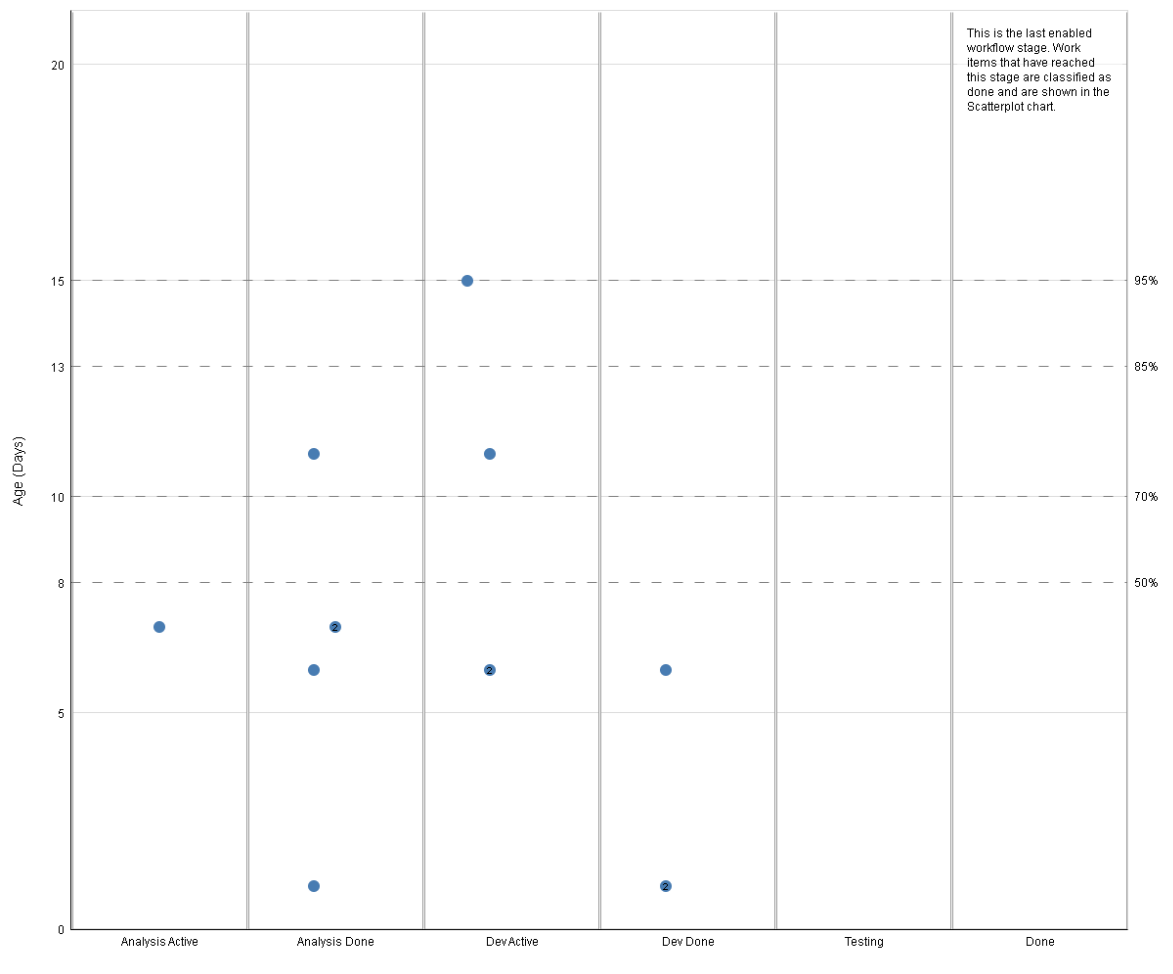


Abbildung 4.5 - WIP Aging Chart / Diagramm zur Alterung der aktuell in Bearbeitung befindlichen Arbeiten für einen Beispiel-Workflow

Bevor wir uns mit der Verwendung dieses Diagramms befassen, sollten wir kurz die Anatomie des Diagramms erläutern, damit Sie wissen, was Sie vor sich haben. Im Gegensatz zum Streudiagramm ist das gesamte Diagramm in Spalten unterteilt, die dem Workflow Ihres Prozesses entsprechen. An der oberen Seite des Diagramms befindet sich das Alter der PBIs.

Das Alter unterscheidet sich in diesem Zusammenhang von der Durchlaufzeit im Streudiagramm. Was wir in diesem Diagramm (Abbildung 4.5) darstellen, ist die gesamte verstrichene Zeit, die ein Element begonnen, aber nicht abgeschlossen wurde (Alter), im Gegensatz zur gesamten verstrichenen Zeit, die ein Element zum Abschluss benötigt hat (Durchlaufzeit). Genau wie bei der Durchlaufzeit können Sie für das Alterungsdiagramm jedoch jede beliebige Zeiteinheit verwenden: Tage, Wochen, Monate, Sprints usw. Jeder Punkt im Diagramm steht also für ein PBI, das in den Prozess eingetreten ist, ihn aber noch nicht verlassen hat. Um einen Punkt darzustellen, suchen Sie einfach den Workflow-Stadium / Schritt, in dem er sich gerade befindet, und subtrahieren dann das heutige Datum vom

## Kapitel 4 - Zwei Diagramme

Startdatum des Elements (denken Sie daran, dass Sie den Zeitstempel für den Eintritt des Elements in Ihren Prozess verfolgen sollten!)

Sie werden auch Perzentil-Linien auf dem Altersdiagramm sehen. Diese Perzentil-Linien sind genau dieselben Linien, die wir für unser Streudiagramm berechnet haben. Sie überlagern die Perzentil-Linien im Altersdiagramm, so dass Sie sehen können, wie sich die aktuellen PBIs im Vergleich zur Gesamtzeit, die frühere Elemente zur Fertigstellung benötigten, entwickeln.

Die Möglichkeit, das aktuelle Alter der Elemente mit der Leistung in der Vergangenheit zu vergleichen, wird uns bei unserem Daily Scrum enorm helfen. Da dies das zweite Mal ist, dass wir in diesem Kapitel Scrum Events erwähnen, ist es vielleicht an der Zeit, zu einer tieferen Diskussion dieser Ereignisse überzugehen.

## Fazit

Die Verfolgung von Metriken allein ist nur von geringem Nutzen, wenn wir keine effektive Möglichkeit haben, diese Daten darzustellen. Das Durchlaufzeit-Streudiagramm und das WIP Alterungsdiagramm / WIP Aging Chart sind zwei von unzähligen analytischen Diagrammen, die wir für datengestützte Entscheidungen verwenden können. Diese beiden sind am hilfreichsten, wenn wir über die Einführung von Arbeitsflussmetriken in die Scrum Events sprechen. Eine ausführlichere Diskussion dieser beiden Diagramme sowie eine ausführlichere Diskussion anderer nützlicher Arbeitsflussdiagramme finden Sie in den beiden Büchern von Dan, "Actionable Agile Metrics for Predictability"<sup>1</sup> und "When Will It Be Done?"<sup>2</sup>.

## Quellen

1. Vacanti, Daniel S. "Actionable Agile Metrics for Predictability". ActionableAgile Press. 2014.
2. Vacanti, Daniel S. "When Will It Be Done?" ActionableAgile Press, 2017.

# Kapitel 5 - Sprint Planung, Teil I

In der Version 2020 des Scrum Guides<sup>1</sup> sind drei Themen aufgeführt, die in der Sprint Planung behandelt werden müssen:

- Warum ist dieser Sprint wertvoll?
- Was kann in diesem Sprint getan werden?
- Wie wird die ausgewählte Arbeit erledigt?

Arbeitsflussmetriken können bei der Beantwortung einer dieser Fragen helfen, aber wir werden Ihnen nicht sagen, bei welcher.

Übrigens ist dies ein sehr umfangreiches Kapitel, daher sollten Sie sich eine Tasse Tee machen, bevor Sie sich einarbeiten.

## Was kann in diesem Sprint getan werden?

"Die Auswahl, wie viel in einem Sprint erledigt werden kann, kann eine Herausforderung sein. Je mehr die Entwickler jedoch über ihre bisherige Leistung, ihre künftige Kapazität und ihre Definition von "fertig" wissen, desto sicherer werden sie in ihren Sprint Prognosen sein."<sup>1</sup>

Da Scrum nur ein Framework ist, bleibt der Guide absichtlich vage in Bezug darauf, wie die Sprint Planung tatsächlich durchgeführt werden sollte. Und wie die meisten, die dieses Manuskript lesen, wissen, ist viel über die richtigen Techniken zur Planung eines Sprints veröffentlicht worden. Darüber hinaus kennen viele von Ihnen auch Ihre Gedanken über die etablierte Orthodoxie. Anstatt gegen einen Haufen schlechter Praktiken zu wettern, die ohnehin nicht Teil von Scrum sind, möchten wir uns hier auf eine ausführliche Diskussion darüber konzentrieren, wie Arbeitsflussmetriken Ihnen helfen können, einen Sprint genau zu planen.

### Ein kurzes Gedankenexperiment, bevor wir beginnen

Nehmen wir an, Sie haben gerade ein Projekt begonnen und in zwei Wochen zehn Elemente fertiggestellt (zu diesem Zeitpunkt spielt es keine Rolle, ob Sie Scrum anwenden oder nicht). Nehmen wir weiter an, dass noch fünfzig Punkte im Projekt verbleiben, von denen einige in Arbeit sind. Wenn Ihr Kunde fragt: "Wann wird das Projekt fertig sein?", wie würden Sie dann berechnen, wie lange es dauern wird, diese fünfzig Punkte abzuschließen?

"Ganz einfach", sagen Sie, "da wir zehn Aufgaben in zwei Wochen erledigt haben, bedeutet das, dass wir fünf Aufgaben pro Woche erledigen können. Bei einer Rate von fünf Aufgaben pro Woche brauchen wir zehn Wochen, um die restlichen 50 Aufgaben zu erledigen. Unsere Antwort lautet also zehn Wochen."

Bei der Überprüfung Ihrer Antwort möchten wir darauf hinweisen, dass Ihre Antwort in einem Punkt richtig und in einem Punkt falsch war. Sehr falsch. Der richtige Teil Ihrer Antwort war die Metrik, die Sie zur Berechnung Ihrer Prognose verwendet haben. Sie haben die Rate berechnet, mit der Sie die Aufgaben erledigt haben - fünf pro Woche. Sie haben die Durchsatzkennzahl verwendet, die wir in Kapitel 2 erläutert haben.

Wie wir dort erläutert haben, ist der Durchsatz ein Maß dafür, wie schnell Elemente einen Prozess verlassen. Etwas anders ausgedrückt: Der Durchsatz ist die Anzahl der Arbeitsposten, die pro Zeiteinheit erledigt werden. Die Zeiteinheit kann eine beliebige Einheit sein, die für Sie nützlich ist - Sie können die erledigten Aufgaben pro Tag, pro Woche, pro Sprint usw. messen.

Was Sie bei Ihrer Berechnung jedoch falsch gemacht haben, war, die verbleibende Arbeit (fünfzig Aufgaben) einfach durch den Durchsatz (fünf Aufgaben pro Woche) zu teilen, um Ihre Prognose (10 Wochen) zu erhalten. Diese Art des Denkens ist ein klassisches Beispiel für den Fehler der Durchschnittswerte<sup>1</sup> (der Flaw of Averages, FoA schleicht sich einfach an, nicht wahr?).

Die Verwendung einer durchschnittlichen Fertigstellungsrate (Durchsatz) zur Projektion des Fertigstellungstermins mehrerer Positionen in einem Backlog leidet unter demselben fatalen Fehler wie das Beispiel der durchschnittlichen Pendelzeit aus dem letzten Kapitel. Dieser fatale Fehler besteht in dem Versuch, die Unsicherheit auf eine einzige Zahl zu reduzieren. Bei jeder Planungsaktivität gibt es mehr als ein mögliches zukünftiges Ergebnis, so dass wir probabilistisch denken müssen.

Wie können wir also einen probabilistischen Ansatz für die Sprint Planung wählen?

### **Monte-Carlo-Simulation**

Wie hoch ist die Wahrscheinlichkeit, dass Sie eine Münze werfen und sie Zahl zeigt? Wie können Sie das wissen? Nehmen wir an, Sie sind wie Will sehr schlecht in Mathematik und haben keine Ahnung, wie Sie dieses Problem mathematisch modellieren können. Was würden Sie dann tun? Eine Möglichkeit wäre, die Münze immer und immer wieder zu werfen und die Ergebnisse zu verfolgen. Die Anzahl der Münzwürfe, bei denen "Zahl" herauskam, geteilt durch die Gesamtzahl der Münzwürfe, ergibt die Wahrscheinlichkeit. Um eine genaue Antwort zu erhalten, müssen Sie die Münze vielleicht Hunderte, wenn nicht Tausende Male werfen, aber mit der Zeit, wenn Sie genügend Würfe verfolgen, werden sich Ihre Ergebnisse der richtigen Antwort annähern.

Glückwunsch! Indem Sie eine Münze immer und immer wieder werfen, haben Sie gerade eine Monte-Carlo-Simulation (MCS) durchgeführt.

Die Monte-Carlo-Methode (und speziell MCS) ist ein statistisches Stichprobenverfahren, bei dem eine Simulation verwendet wird, um die Wahrscheinlichkeit verschiedener Ergebnisse angesichts der Eingabe von Zufallsvariablen vorherzusagen. Diese Definition ist sehr langatmig und sagt nicht wirklich viel aus, daher werden wir im nächsten Abschnitt tiefer eintauchen. Für den Moment sollten Sie einfach wissen, dass jedes Mal, wenn Sie mit einem Problem konfrontiert werden, das nicht mathematisch modelliert werden kann oder das zu schwer zu berechnen wäre, selbst wenn Sie es korrekt modellieren könnten, oder das zu lange dauern würde, selbst wenn es eine Antwort gäbe, ein gutes Kandidatenproblem haben, das mit MCS gelöst werden kann. Spoiler-Alarm: Sprint Planung ist ein guter Kandidat für MCS.

## **MCS in der Sprint Planung**

Aus dem Scrum Guide zur Sprint Planung: "...die Entwickler wählen Elemente aus dem Product Backlog aus, die in den aktuellen Sprint aufgenommen werden sollen".<sup>1</sup> Wenn sich das Scrum Team auf ein (vorläufiges) Sprint Ziel geeinigt hat, kann es sich anschauen, welche PBIs helfen würden, dieses Ziel zu erreichen. Dies ist ein iterativer Prozess, da sie vielleicht feststellen, dass sie mehr Kapazität haben (was bedeutet, dass sie ein ehrgeizigeres Sprint Ziel haben könnten), oder das Ziel ist zu ehrgeizig für die Menge der PBIs, die sie tatsächlich ziehen / pullen können. Mit anderen Worten: Sobald ein mögliches Sprint Ziel gefunden ist, werden

## Kapitel 5 - Sprint Planung, Teil I

die Entwickler gebeten, zu prognostizieren, wie viele PBIs sie in diesem Sprint fertigstellen werden. In Anbetracht all der möglichen Variablen, die mit einer solchen Vorhersage verbunden sind (Abhängigkeiten, Unterbrechungen, Pandemien usw.), gibt es keine Möglichkeit, eine Gleichung (wie  $E=mc^2$ ) aufzustellen, die eine Lösung für dieses Problem angemessen modellieren würde.

**Hinweis:** Lassen Sie sich nicht dabei erwischen, wie Sie versuchen, Ihren Sprint zu füllen, ohne ein Sprint Ziel festzulegen. Das Ziel hat immer Vorrang vor dem Umfang, den Sie gezogen haben, damit Sie die nötige Flexibilität haben, um mit der Komplexität umzugehen. Probabilistische Prognosen heben dies nicht auf. -Will

Aber was wäre, wenn wir uns ansehen könnten, wie viele Aufgaben wir in der Vergangenheit erledigt haben, und uns daran orientieren könnten, wie viele Aufgaben wir in der Zukunft erledigen können? Und was wäre, wenn dieses Modell das mit diesen zukünftigen Ergebnissen verbundene Risiko berücksichtigen könnte? Hier kommt unser Freund, die Monte-Carlo-Simulation ins Spiel.

Monte-Carlo-Methoden sind unterschiedlich, aber im Allgemeinen folgen sie einem typischen Muster:

1. Definieren einer Wahrscheinlichkeitsverteilung der möglichen Eingaben
2. Einer zufälligen Auswahl von Werten aus der Eingabeverteilung und Durchführung einer Berechnung mit den ausgewählten Eingaben
3. Aggregieren der Ergebnisse (in der Regel unter Verwendung eines Ergebnishistogramms)
4. Wiederholen der Schritte 2-3 beliebig oft, bis Sie ein klares Bild davon haben, wie die Ergebnismenge aussieht

In der Sprint Planung sieht dies folgendermaßen aus:

### **Schritt 1: Definieren Sie eine Wahrscheinlichkeitsverteilung der möglichen**

**Inputs.** Wann immer Sie (jedenfalls in einem agilen Kontext) eine Frage hören, die wie folgt formuliert ist: "Wie viele Elemente...", sollten Sie sofort an "Durchsatz" denken. Wie bei dem früheren Gedankenexperiment für das Projekt benötigen wir auch für dieses Problem eine Arbeitsflussmetrik für die Vorhersage, und diese Arbeitsflussmetrik ist der Durchsatz. Die Wahrscheinlichkeitsverteilung ist einfach die Menge der historischen täglichen Fertigstellungsraten von PBIs. Erinnern Sie sich an Kapitel 2, dass der Durchsatz eine



Zufallsvariable ist und seine Verteilung durch die historischen Fertigstellungsraten von PBIs für unser Scrum Team gegeben ist.

Heißt das, wir nehmen alle historischen Durchsatzdaten, die wir haben, und speisen den gesamten Datensatz in unser MCS ein? Nun, möglicherweise, aber nicht ganz. Das schmutzige kleine Geheimnis von MCS ist, dass wir davon ausgehen, dass die Zukunft, die wir modellieren, in etwa so aussieht wie die Vergangenheit, für die wir Daten haben. Das Schöne an der Verwendung von professionellem Scrum ist, dass dies in den meisten Kontexten eine sichere Annahme ist. Aber es bedeutet trotzdem, dass wir eine fundierte Vermutung darüber anstellen müssen, welche Teilmenge historischer Daten uns bei unserer Planung helfen wird. Diese Praxis ist definitiv mehr Kunst als Wissenschaft, aber wir haben eine Reihe von Heuristiken, die uns dabei helfen:

1. Neuere Daten sind in der Regel besser als weniger aktuelle (ältere) Daten. Möglicherweise haben Sie schon einmal gehört, dass "mehr Daten besser sind als weniger Daten". Diese Aussage ist aufgrund des Aktualitätsprinzips nicht unbedingt zutreffend. Nehmen wir an, Sie hätten Durchsatzdaten über Ihr Scrum Team, die 10 Jahre zurückreichen. Wenn mehr Daten besser sind als weniger Daten, bedeutet das, dass Sie den gesamten Durchsatz der letzten 10 Jahre für Ihre Simulation verwenden sollten? Offensichtlich nicht. Aber wie viele aktuelle Daten brauche ich, um eine erfolgreiche Simulation durchzuführen? Es gibt keine genaue Antwort auf diese Frage, aber die allgemeine Faustregel lautet: mindestens 10-12 Datenpunkte und idealerweise 20 Punkte als komfortables Minimum. Bevor Sie sich zu sehr erschrecken: Das bedeutet nicht, dass Sie 20 Sprints durchführen müssen, bevor Sie genügend Daten haben, um eine MCS durchzuführen. Erinnern Sie sich an Kapitel 2, in dem wir empfohlen haben, Tage als Zeiteinheit für den Durchsatz zu verwenden. Das heißt, wenn Sie zweiwöchige Sprints durchführen, dann sollten Sie in zwei Sprints mehr als genug Daten haben, um loszulegen. (Zwei Sprints von jeweils zwei Wochen würden 28 Tage an Durchsatzdaten ergeben. Ja, Wochenenden zählen - googeln Sie es!)
2. Aber auch neuere Daten sind kein Allheilmittel. Nehmen wir an, Sie sind ein westeuropäisches Scrum Team, das einen Sprint hat, der am 7. Januar beginnt. Wäre es sinnvoll, die Daten des letzten Sprints zu verwenden, um den kommenden Sprint zu planen? Auch hier gilt, dass dies aufgrund der Weihnachts- und Neujahrsfeiertage nicht möglich ist. In Amerika wird dieses Problem noch schwieriger, weil man wegen des Thanksgiving-Feiertags auch nicht unbedingt die Daten vom November verwenden kann. Es liegt also an Ihnen zu entscheiden, welche Teilmenge historischer

Daten die Zukunft am besten widerspiegelt, für die Sie planen. Der Vorteil von MCS ist, dass es ein so schnelles und einfaches Tool ist, dass Sie realistischerweise mit mehreren Eingaben herumspielen und sehen können, was die Ergebnisse sind (aber dazu später mehr).

**Schritt 2: Wählen Sie nach dem Zufallsprinzip Werte aus der Eingabeverteilung aus und führen Sie eine Berechnung mit den ausgewählten Eingaben durch.** Dies ist der Punkt, an dem das „MCS-Gummi“ auf die Straße trifft. Um zu prognostizieren, wie viele PBIs fertiggestellt werden, "simulieren" wir, wie viele Elemente an jedem aufeinanderfolgenden Tag des bevorstehenden Sprints erledigt werden. Das funktioniert so, dass wir unsere Simulation zum Beispiel am 1. Januar beginnen. Für den 1. Januar wählen wir dann eine zufällige Durchsatzzahl aus unseren historischen (Eingabe-)Durchsatzdaten aus. Nehmen wir an, wir wählen zufällig eine 3. Für die Zwecke der Simulation bedeutet dies, dass unser Team am 1. Januar 3 PBIs durchgeführt hat. Dann wiederholen wir die gleiche Berechnung für den nächsten Tag, den 2. Januar. Nehmen wir an, wir wählen zufällig eine 5 für den 2. Januar. Das bedeutet, dass wir in diesem Sprint bisher insgesamt acht PBIs durchgeführt haben. Machen Sie das für die Gesamtzahl der Tage im Sprint, addieren Sie die Ergebnisse und das ist **ein** mögliches Ergebnis für die Gesamtzahl der im nächsten Sprint abzuschließenden PBIs (dies wird gemeinhin als ein "Lauf" der Simulation bezeichnet).

**Hinweis:** Der Algorithmus, den wir hier verwendet haben, bei dem wir einfach einen historischen Durchsatz-Input zufällig auswählen, ist nur eine solche Option für unsere Simulation. Es gibt noch viele andere. Weitere Informationen finden Sie in Dans Buch "When Will It Be Done?"<sup>3</sup>

**Schritt 3: Aggregieren Sie die Ergebnisse.** Für jeden "Lauf" der Simulation werden wir die Ergebnisse in einem so genannten Ergebnishistogramm aufzeichnen. Was das Ergebnishistogramm ist und wie es zu interpretieren ist, wird in Kürze erläutert.

**Schritt 4: Wiederholen Sie die Schritte 1-3 beliebig oft, bis Sie ein klares Bild davon haben, wie die Ergebnismenge aussieht.** Mit jedem Simulationsdurchlauf wird die Wahrscheinlichkeitsverteilung, wie sie das Ergebnishistogramm beschreibt, immer deutlicher. Denken Sie zum Beispiel an das Beispiel mit dem Münzwurf. Meinen Sie, Sie hätten eine bessere Vorstellung von der Wahrscheinlichkeit, Zahl zu erhalten, wenn Sie zweimal oder zweitausendmal geworfen hätten? Trotzdem wird MCS ziemlich schnell zu einem Ergebnis konvergieren, so dass weitere Durchläufe lediglich zu einem besser aussehenden Ergebnishistogramm führen. Für die meisten Sprint Planungen haben wir festgestellt, dass 1.000-2.000 Durchläufe ziemlich gute Ergebnisse liefern. Mehr als 10.000

Durchläufe führen in der Regel nicht zu einer wesentlich besseren Qualität. Probieren Sie dies selbst aus, um es in Aktion zu sehen.

### **Interpretieren des Ergebnishistogramms**

Wie bereits erwähnt, ist das Ergebnis Ihrer Sprint Plannungs - MCS ein Ergebnishistogramm. Dieses Histogramm stellt die Form Ihres Risikos in Bezug auf die verschiedenen möglichen zukünftigen Ergebnisse dar, d. h. die unterschiedlichen Chancen, dass Sie eine unterschiedliche Anzahl von PBIs bis zur Fertigstellung erreichen. Ein interessanter Nebenaspekt ist, dass das Ergebnishistogramm eines MCS in der Regel einer Normalverteilung ähnelt. Dies ist auf das so genannte zentrale Grenzwertsyndrom (CLT) zurückzuführen. Sie brauchen sich nicht mit den Einzelheiten des CLT zu befassen, Sie müssen nur wissen, dass das Ergebnishistogramm im Allgemeinen einer Glockenkurve ähnelt.

Wie verwandeln wir also unser Ergebnishistogramm in eine Antwort auf die Frage "Was kann in diesem Sprint getan werden?"

Da das Ergebnishistogramm die Form Ihres Risikos in Verbindung mit möglichen Ergebnissen darstellt, müssen wir dieses Risiko zunächst in akzeptable und nicht akzeptable Ergebnisse unterteilen. Dies können wir tun, indem wir sogenannte Perzentil-Linien in das Histogramm einzeichnen.

Bevor wir beginnen, wollen wir zunächst auf ein wichtiges Prinzip des soeben erstellten Histogramms hinweisen. Wenn Sie die Höhen aller Balken im Diagramm zusammenzählen, erhalten Sie die Anzahl der Durchläufe, aus denen die Simulation bestand. Wenn wir z. B. 10000 Durchläufe in unserer Simulation durchgeführt haben, ergibt die Summe der Höhen der Balken im Diagramm den Wert 10000. Die Wahrscheinlichkeit eines bestimmten Ergebnisses ist also die Höhe eines bestimmten Balkens geteilt durch die Gesamtzahl der Durchläufe.

Zur Veranschaulichung sehen Sie sich das Ergebnishistogramm in Abbildung 5.1 unten an.

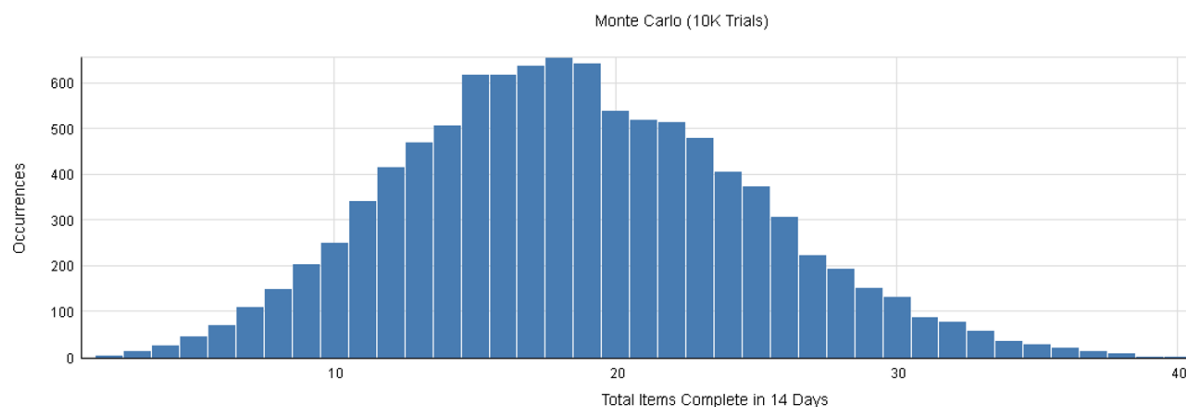


Abbildung 5.1: Histogramm der Sprint Planungsergebnisse

Abbildung 5.1 ist das Ergebnis eines MCS-Laufs zur Prognose eines zweiwöchigen (14-tägigen) Sprints. In diesem Beispiel wurde das MCS 10.000 Mal ausgeführt, um die hier gezeigten Ergebnisse zu erhalten. Richten Sie Ihre Aufmerksamkeit auf den höchsten Balken in der Mitte des Diagramms. Da dies der höchste Balken des Histogramms ist, bedeutet dies, dass dieses Ergebnis von allen Ergebnissen der Simulation am häufigsten auftrat. Mit anderen Worten, dies ist unser wahrscheinlichstes Ergebnis. Es ist auf dem Bild etwas schwer zu erkennen, aber dieser Balken kommt bei 18 Elementen vor. Die Höhe dieses speziellen Balkens beträgt 656. Das bedeutet, dass dieses Scrum Team eine Chance von 6,56 % ( $656/10.000$ ) hat, im nächsten Sprint genau 18 PBIs zu erledigen. Das ist nicht sehr gut, oder? (Erinnern Sie sich, dass das wahrscheinlichste Ergebnis nicht sehr wahrscheinlich ist?)

Aber es gibt noch eine andere Möglichkeit, diese Ergebnisse zu interpretieren. Nehmen wir einmal an, wir hätten geplant, in diesem Sprint 18 Aufgaben zu erledigen. Wenn wir genau 18 Aufgaben erledigt haben, dann wäre das ein gutes Ergebnis. Aber was ist, wenn wir 19 Aufgaben erledigt haben? Aus dem Histogramm können Sie ersehen, dass die Chance besteht, dass wir tatsächlich 19 Elemente fertigstellen können. Wenn wir unserem Product Owner sagen würden, dass wir 18 Elemente fertigstellen konnten, aber tatsächlich 19 Elemente fertiggestellt haben, was glauben Sie, wie er reagieren würde? Ich vermute, dass er das als gutes Ergebnis ansehen würde. Das Gleiche würde gelten, wenn wir 20 Elemente fertigstellen würden, 21, 22 und so weiter... Tatsächlich würde jedes Ergebnis rechts von 18 (einschließlich) als akzeptabel angesehen werden, wie in dem grün schattierten Bereich in Abbildung 5.2 unten dargestellt:

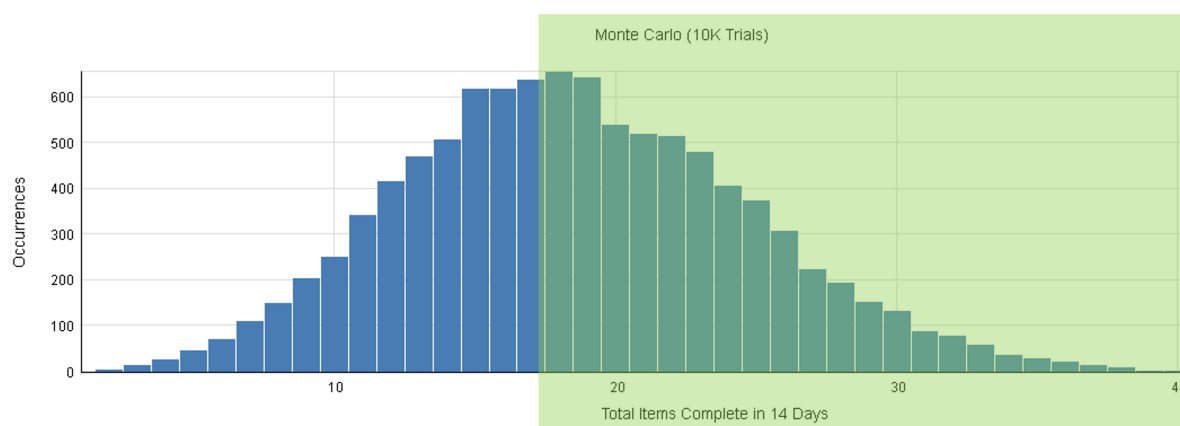


Abbildung 5.2: 18 oder mehr Elemente abgeschlossen

Um die Wahrscheinlichkeit der Erledigung von 18 oder mehr Aufgaben zu ermitteln, addieren Sie einfach die Höhen aller Balken im grün schattierten Bereich und teilen Sie durch 10.000. In diesem Beispiel summieren sich die Höhen dieser Balken auf 5.000, so dass die Wahrscheinlichkeit, dass unser Team in diesem Sprint 18 oder mehr Aufgaben erledigt, 50 % beträgt (Sie haben übrigens gerade das 50. Perzentil für Ihre Ergebnisverteilung berechnet).

50% ist nicht schlecht, aber auch nicht großartig. Das Tolle daran ist, dass wir mit dieser Technik die Wahrscheinlichkeit eines beliebigen Bereichs von Ergebnissen berechnen können. Die typische Herangehensweise an die Planung besteht daher darin, zu ermitteln, wie viel Risiko unser Team bereit ist, in Kauf zu nehmen, und dies in die Planung einzubeziehen. Wenn es also zu viel Risiko ist, in 50 % der Fälle falsch zu liegen, wie viel Risiko ist dann tolerierbar? Nehmen wir einmal an, unser Team möchte so planen, dass es in 85 % der Fälle richtig liegt. Ganz einfach. Wir fangen einfach rechts im Diagramm an und addieren die Höhen, bis wir 8500 erreichen ( $8500 / 10.000 = 85\%$ ). Dies ist in Abbildung 5.3 unten dargestellt:

## Kapitel 5 - Sprint Planung, Teil I

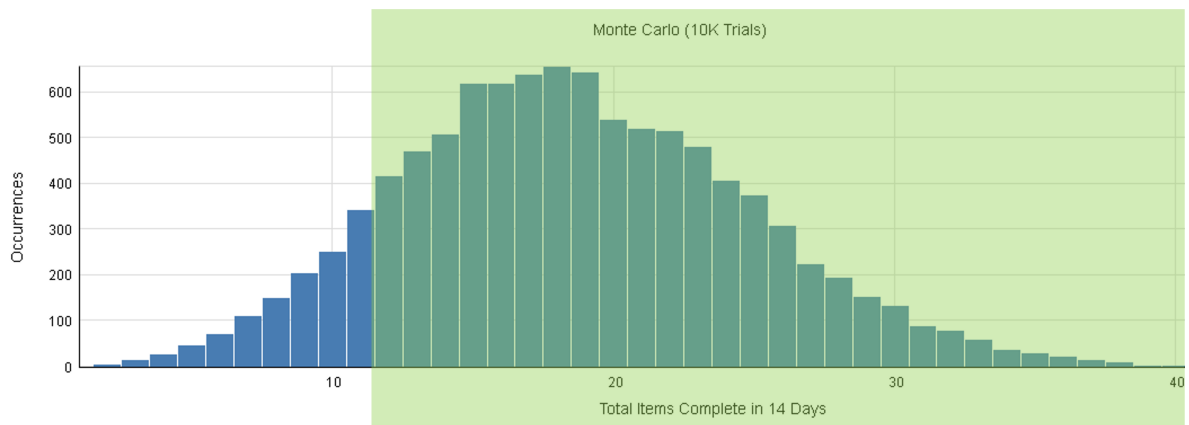


Abbildung 5.3: 18 oder mehr Elemente abgeschlossen

Für diese Ergebnisse liegt das 85. Perzentil bei 12, was bedeutet, dass wir eine 85%ige Chance haben, in diesem Sprint 12 oder mehr Aufgaben zu erledigen. Sie werden feststellen, dass wir von einer 50-prozentigen Chance, richtig zu liegen, zu einer 85-prozentigen Chance, richtig zu liegen, übergegangen sind, indem wir von der Vorhersage, dass 18 oder mehr Elemente abgeschlossen werden, zur Vorhersage, dass 12 oder mehr Elemente abgeschlossen werden, übergegangen sind. Dieses Ergebnis sollte intuitiv einleuchtend sein: Um die Wahrscheinlichkeit zu erhöhen, dass wir richtig liegen, prognostizieren wir, dass wir weniger Aufgaben abschließen können.

Mit dem gleichen Ansatz können wir jede beliebige Wahrscheinlichkeit berechnen: 70 %, 95 %, 30 %, was auch immer. Abbildung 5.4 zeigt einige Standard-Perzentile, die über ein Ergebnishistogramm gelegt sind:

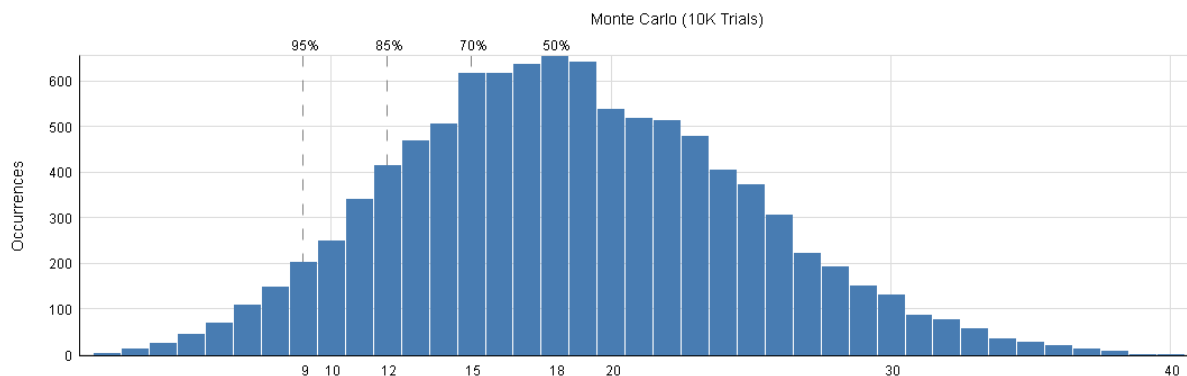


Abbildung 5.4: Standard-Perzentile

Nachdem Sie dies gesehen haben, könnte Ihre erste Neigung (und sicherlich auch die Ihrer Kunden) darin bestehen, zu fragen: Wie hoch ist die Wahrscheinlichkeit, zu 100 % richtig zu liegen? Wenn Sie das denken, bedeutet das, dass Sie Kapitel 3 nicht gelesen haben. In diesem

Kapitel haben wir erklärt, dass es so etwas wie 100 % bei Prognosen nicht gibt. Ihre zweite Neigung könnte darin bestehen, sich zu fragen, wie hoch Ihre Chance ist, zu 99 % richtig zu liegen. Das ist zwar eine berechtigte Frage, aber sie ist nicht sehr nützlich. Auch wenn es nicht im Diagramm in Abbildung 5.4 eingezeichnet ist, liegt das 99. Perzentil irgendwo bei 1. Um also zu 99 % sicher zu sein, dass dieser Sprint stattfindet, werden Sie prognostizieren, dass Sie 1 oder mehr PBIs zustande bringen können. Sagen Sie das Ihrem PO ruhig, ohne zu lügen.

Es geht darum, ein Risikoprofil zu wählen, das für Ihren Kontext angemessen ist. In den meisten Situationen ist eine Planung von 99 % absurd. Überlegen Sie stattdessen, was eine vernünftige Erfolgsquote ist, die Ihnen eine gewisse Planungssicherheit gibt, aber auch einräumt, dass wir uns von Zeit zu Zeit irren können. In der Anfangsphase finden viele Teams, dass das 85. Perzentil dieses richtige Gleichgewicht darstellt - obwohl es keine feste Regel gibt. Wenn Sie Prototypen erstellen oder auf der grünen Wiese entwickeln, dann sind vielleicht 70 % (oder sogar 50 %) akzeptabel. Wenn Sie behördliche Vorschriften umsetzen oder bei Verspätungen mit hohen Geldstrafen rechnen müssen, ist vielleicht ein Wert von 95 % besser. Das Tolle an MCS ist, dass Sie all diese Optionen in einem einzigen Ergebnisdiagramm sehen und entsprechend planen können.

# Die Sprache der Prognosen

Wir haben in der vorangegangenen Diskussion nicht explizit darauf hingewiesen, aber Sie werden feststellen, dass alle Prognoseaussagen demselben Muster folgen. Das heißt, jede Prognose enthielt einen Bereich möglicher Ergebnisse und die Wahrscheinlichkeit des Eintretens eines Ergebnisses in diesem Bereich. Wir haben zum Beispiel gesagt: "Wir haben eine 50-prozentige Chance, in diesem Sprint 18 oder mehr PBIs zu erreichen". Ich empfehle Ihnen, Ihre Prognosen immer nach diesem Muster zu formulieren.

Streng genommen ist die Vorhersage der Anzahl der PBIs, die in einem Sprint fertiggestellt werden sollen, zwar speziell die Aufgabe der Entwickler, aber der Scrum Guide sagt ganz klar, dass die Sprint Planung in enger Zusammenarbeit mit dem Product Owner des Scrum Teams erfolgen sollte. Stellen Sie sich vor, dass die Entwickler in der Sprint Planung eine MCS wie oben durchgeführt haben und dann ihrem PO mitgeteilt haben, dass sie 18 Punkte erledigen können. Was denken Sie, was Ihr PO in diesem Moment denkt? Natürlich denkt er, dass das Team eine 100-prozentige Chance hat, genau 18 Aufgaben zu erledigen (während wir in Wirklichkeit wissen, dass diese Chance eher bei 6 % liegt). Wir sind der Meinung, dass dies ein sehr wichtiger Punkt ist, denn oft ist es der PO, der das Geschäftsrisiko, das mit einem

bestimmten Sprint Ziel verbunden ist, besser einschätzen kann. Wenn Sie dem PO nur "18" sagen, dann verschweigen Sie wichtige Informationen, die den PO daran hindern, ein brauchbarer Partner für die Zusammenarbeit zu sein (ganz abgesehen davon, dass die bloße Angabe von "18" nicht sehr transparent ist). Wir sind der Meinung, dass sowohl die Entwickler als auch der PO einen gleichwertigen Beitrag dazu leisten sollten, bis zu welchem Prozentsatz jeder Sprint geplant werden sollte. Dies gilt insbesondere, wenn man bedenkt, dass das Scrum Team als Ganzes für das Erreichen des Sprint Ziels seit dem Update des Scrum Guide 2020 verantwortlich ist.

# Aber was ist mit der Komplexität?

In der bisherigen Diskussion über MCS haben wir nirgends erwähnt, dass wir die relative Komplexität der PBIs, die wir prognostizieren, schätzen. Das liegt daran, dass bei einem MCS-Ansatz eine Schätzung nicht erforderlich ist. Es ist bedauerlich, dass der agilen Gemeinschaft das Schlangenöl der Story Points verkauft wurde, aber die Wahrheit ist, dass die geschätzte relative Komplexität ein extrem schlechter Prädiktor für die Kapazität ist. Das Geniale an der Monte-Carlo-Methode ist, dass bei richtiger Auswahl der Inputs die meisten Variablen, die sich auf die zukünftigen Ergebnisse auswirken könnten, in den historischen Arbeitsflussdaten enthalten sind und keine Schätzung mehr erforderlich ist.

Denken Sie einen Moment darüber nach. Ihr historischer Durchsatz (ganz zu schweigen von den anderen Arbeitsflussmetriken) wird Daten für PBIs enthalten, die, während Sie an ihnen arbeiteten:

- ein breites Spektrum an Komplexität aufwiesen
- anderen PBIs, an denen Sie in der Vergangenheit gearbeitet haben, ähnlich waren
- anderen PBIs, an denen Sie in der Vergangenheit gearbeitet haben, überhaupt nicht ähnlich waren
- aufgrund interner Abhängigkeiten blockiert waren
- aufgrund externer Abhängigkeiten blockiert waren
- durch andere Arbeiten unterbrochen wurden
- durch Feiertage und Urlaube unterbrochen wurden



## Kapitel 5 - Sprint Planung, Teil I

- und so weiter..

Das Einzige, was Ihre historischen Daten nicht enthalten, sind Dinge, die nicht planbar sind (wie Pandemien, die die Welt für mehrere Monate lahm legen). Aber wenn sie eintreten, würden diese Ereignisse jede Prognosetechnik untergraben.

Ihr Scrum Team ist bei seiner Arbeit jeden Tag mit Risiken und Unsicherheiten konfrontiert. Dieses Risiko und diese Ungewissheit werden in den historischen Aufzeichnungen in Bezug auf Ihre Fähigkeit, PBIs zu erledigen, festgehalten - nämlich in Ihren Durchsatzdaten. Wählen Sie Ihre historischen Eingabedaten richtig aus, und die Chancen stehen gut, dass Sie den Großteil der Faktoren abgedeckt haben, die Ihren Sprint Plan beeinflussen könnten. Und obendrein haben Sie noch die zusätzliche Flexibilität eines Sprint Ziels, das (bis zu einem gewissen Grad) Änderungen an den von Ihnen gezogenen Daten zulässt.

Und das ist noch nicht einmal das Beste daran (obwohl es das wirklich ist). Das Beste daran ist, dass ein MCS mit dem richtigen Tool innerhalb von Minuten durchgeführt werden kann, während die Schätzung der Komplexität Stunden dauern kann.

Apropos Komplexität und Sizing: Es gibt noch einen Aspekt der Sprint Planung, den wir behandeln müssen. Da dieses Kapitel nun aber lang genug ist, sollten wir eine Pause einlegen und uns neu gruppieren, bevor wir über das kritische Thema Right Sizing sprechen.

## Quellen

1. Schwaber, Ken and Sutherland, Jeff "The Scrum Guide" <https://scrumguides.org>, 2020.
2. Savage, Sam L. "The Flaw of Averages". John Wiley & Sons, Inc., 2009.
3. Vacanti, Daniel S. "When Will It Be Done?" ActionableAgile Press, 2017.

# Kapitel 6 - Sprint Planung, Teil II

Wir haben unseren historischen Durchsatz genommen, ihn in ein MCS eingespeist, eine Antwort auf der Grundlage unseres tolerierbaren Risikos erhalten und damit ist die Planung abgeschlossen, oder? Nicht ganz.

Nehmen wir an, unser MCS hat einen Bereich möglicher Ergebnisse mit einem bestimmten akzeptablen Prozentsatz an Unsicherheit ausgespuckt, z. B. "8 oder mehr Elemente mit 85 % Erfolgswahrscheinlichkeit". Das bedeutet, dass wir zu Planungszwecken 8 PBIs in diesen Sprint einbeziehen sollten. Um das klarzustellen: Das MCS sagt Ihnen nicht, welche Elemente Sie einbeziehen sollten, sondern nur, dass Sie Kapazitäten für bis zu 8 haben. Es liegt daher in der Verantwortung der Entwickler in unserem Scrum Team, (in Zusammenarbeit mit dem Product Owner) zu entscheiden, welche 8 PBIs in den Sprint aufgenommen werden sollen. Der Punkt hier ist, dass der MCS Ihnen nicht sagt, welche Aufgaben Sie im nächsten Sprint erledigen können, er gibt Ihnen nur die Mindestkapazität an, die Sie einplanen müssen. Dies kann sicherlich hilfreich sein, um zu erörtern, welches Sprint Ziel Sie sich vornehmen wollen.

ABER! Es ist ein wenig differenzierter als das. Die Ergebnisse des MCS deuten nicht darauf hin, dass Sie einfach irgendein altes PBI in den Sprint einbringen können. Ob Sie sich dessen bewusst sind oder nicht, die MCS-Ergebnisse sagen uns wirklich, wie viele PBIs in der richtigen Größe vernünftigerweise durchgeführt werden können.

## Richtiges Sizing / Größenschnitt der Arbeit

"Das Scrum Team kann diese Elemente während dieses Prozesses verfeinern, was das Verständnis und das Vertrauen erhöht."^1^ Die Frage der meisten Scrum Teams ist "wie viel Refinement / Verfeinerung ist genug Verfeinerung"? Aus der Flow-Perspektive lautet die Antwort auf solche Fragen fast immer "weniger als Sie denken".

## Kapitel 6 - Sprint Planung, Teil II

Es gibt einen weit verbreiteten Mythos über den Ablauf, dass alle Elemente, die Ihren Prozess durchlaufen, gleich groß sein müssen. Da Sie die Komplexität nicht im Voraus abschätzen, ist die einzige Möglichkeit, die Arbeit zu verstehen, die gleiche Größe für alles, oder? Falsch. In der Arbeitsflusstheorie gibt es nichts, was verlangt, dass alle Elemente im Flusssystem genau gleich groß sein müssen. Tatsächlich gibt es eine ganze Theorie der Variation, die anerkennt, dass die Elemente nicht nur nicht genau die gleiche Größe haben müssen, sondern dass es auch nichts gibt, was man jemals tun könnte, um sie alle genau gleich groß zu machen - selbst wenn man es wollte. Das heißt, es wird immer Unterschiede in der Größe der Work Items / Arbeiten geben.

Daher müssen wir ein System entwerfen, das in der Lage ist, die unterschiedlichen Größen der Work Items / Arbeiten, die letztendlich in unser System gelangen, zu bewältigen. Aber es gibt Grenzen für die Menge an Variation, die wir bewältigen können. Zur Veranschaulichung dieses Gedankens verwendet Frank Vega gerne das Beispiel eines Holzhackers (jeder, der den Film Fargo gesehen hat, weiß genau, wovon wir sprechen). Stellen Sie sich vor, was passiert, wenn Sie versuchen, einen zu großen Ast in die Holzhackmaschine zu schieben (a la Fargo). Zumindest wird der Ast stecken bleiben. Schlimmstenfalls wird der Ast Ihren Häcksler zerbrechen. Und was passiert, wenn Sie einen Haufen Sägespäne nehmen und diese in den Häcksler werfen? Auch das würde das Gerät verstopfen. Aber das sind Extremfälle. Der Holzhäcksler wäre in der Lage, jede Größe zwischen Sägemehl und einem kleinen Baumstamm zu verarbeiten. Jede Astgröße, die der Holzhäcksler ohne Probleme verarbeiten kann, gilt als richtig dimensioniert.

Es geht also darum, den Grad der Variation zu verstehen, den unser Arbeitsflusssystem / Ablaufsystem bewältigen kann. Für ein Scrum Team, das zweiwöchige Sprints durchführt, könnte diese Variation in der PBI-Größe zum Beispiel zwischen 1 und 14 Tagen liegen, in der Realität sollte sie aber wahrscheinlich etwas geringer sein. Um eine Vorstellung davon zu bekommen, was "richtig dimensioniert" in unserem Kontext bedeutet, müssen wir uns unsere historischen Durchlaufzeitdaten ansehen und etwas berechnen, das als Service Level Expectation / Erwartung oder SLE bekannt ist.

Was eine SLE genau ist und wie man sie berechnet, wird in einem späteren Kapitel behandelt, aber schon jetzt sollten Sie Folgendes wissen: Wenn Sie durch die Sprint Planung gehen und auf der Grundlage der Ergebnisse Ihrer MCS PBIs auswählen, sollten Sie und Ihr Team kurz darüber sprechen, ob Ihr Element die richtige Größe hat oder nicht. Ihr SLE könnte zum Beispiel lauten: "85 % der PBIs werden in 7 Tagen oder weniger abgeschlossen" (kommt Ihnen diese Formulierung bekannt vor?). Für jedes PBI, das Sie auswählen, könnte Ihre Konversation also etwa so ablaufen: "Basierend auf dem, was wir im Moment über dieses PBI wissen - und

das ist nicht viel - glauben wir, dass wir eine 85%ige Chance haben, es in 7 Tagen oder weniger zu erledigen?" Wenn die Antwort "Ja" lautet, ist das Gespräch beendet und Sie ziehen das PBI in den Sprint ein. Wenn die Antwort nein lautet, ist möglicherweise eine weitere Verfeinerung erforderlich. Denken Sie daran, dass die Entwickler immer die Möglichkeit haben, entweder (a) das PBI so zu übernehmen, wie es ist, und dieses Risiko einzugehen, oder (b) sich dafür zu entscheiden, in diesem Sprint nicht an diesem PBI zu arbeiten.

Anmerkung: Ich persönlich bin in diesem Punkt voreingenommen und glaube, dass, wenn das PBI das nächste aus dem Product Backlog ist, man sich um eine Refinement / Verfeinerung bemühen sollte (Verfeinerung bedeutet für mich: das PBI in mehrere andere PBIs aufteilen - nicht in Aufgaben - oder die Akzeptanzkriterien ändern, oder was auch immer). Wie Sie jedoch wissen, liegt die letztendliche Verantwortung für die Auswahl von PBIs bei den Entwicklern, so dass es an ihnen liegt, wie sie vorgehen (dies ist einer von vielen Punkten, bei denen der Scrum Guide und ich nicht einer Meinung sind). -Dan

Eine letzte Sache zum Thema Right Sizing ist, dass der Scrum Guide noch weiter geht und sagt, dass die Entwickler, sobald die PBIs ausgewählt sind, noch weiter gehen können. Aus dem Scrum-Guide<sup>1</sup>: "Dies geschieht oft durch die Zerlegung von Product Backlog Items in kleinere Arbeitselemente von einem Tag oder weniger". Oft bedeutet nicht immer, und Sie werden feststellen, dass Sie diesen Schritt bequem überspringen können, wenn Sie Arbeitsflussmetriken verwenden. Wenn Ihre Elemente die richtige Größe haben, dann sollte - zumindest aus der Arbeitsflussperspektive - keine weitere Aufschlüsselung erforderlich sein.

# Werkzeugausstattung

Es gibt viele Tools, die Ihnen bei der Verwendung von MCS für die Sprint Planung helfen, und wenn Sie mit Dans Arbeit vertraut sind, wissen Sie, dass er eine starke Vorliebe für ein Tool im Besonderen hat.

Eine weitere Empfehlung, die wir Ihnen geben können, wenn Sie mit MCS beginnen, ist, einige Modelle selbst manuell zu erstellen. Unter Verwendung des oben beschriebenen vierstufigen Algorithmus ist es eine einfache Übung, eine Tabellenkalkulationsvorlage zu erstellen, um eine Zufallsauswahl zu treffen und ein Histogramm der Ergebnisse zu erstellen. Wir sprechen diese Empfehlung aus zwei Gründen aus:

1. Die manuelle Durchführung einer MCS hilft Ihnen nicht nur, die Technik gründlicher zu erlernen, sondern gibt Ihnen auch Vertrauen in den Ansatz als gültige Prognosetechnik.
2. Was vielleicht noch wichtiger ist: Wenn Sie in der Lage sind, MCS-Modelle selbst zu erstellen, können Sie, wenn Sie schließlich zur Verwendung eines Tools übergehen, die Ergebnisse der Simulationen des Tools mit Ihren eigenen vergleichen. Diese Validierung ist etwas, was wir normalerweise tun, wenn wir ein neues Tool ausprobieren, da man nie wirklich weiß, wie die Tools die Dinge hinter den Kulissen berechnen (hust ;), Jira).

# Drastische Reduzierung der Planungszeit

Wir wissen, dass sich dieses Kapitel anhört, als würde Dan in einem großen Scrum Guide Wortschwall schwärmen, und dafür entschuldigt er sich natürlich. Er lernt immer noch. Aber es gibt einen letzten Punkt in Bezug auf den Scrum Guide, den wir ansprechen müssen, wenn es um die Verwendung von MCS in der Sprint Planung geht.

Der Scrum Guide erlaubt es Teams, bis zu 8 Stunden für die Sprint Planung zu budgetieren, als Timebox. Das ist das größte Einzelbudget für ein Ereignis im Guide. Für denselben Sprint stehen für Sprint Reviews nur vier Stunden zur Verfügung, für Sprint Retrospektiven nur 3 Stunden, und wenn man alle Daily Scrums über einen einmonatigen Sprint zusammenzählen würde, käme man nur auf 5 bis 6 Stunden. 8 Stunden sind eine Menge Zeit, und die Autoren haben selten gesehen, dass sie voll genutzt wurde. Will kann die Anzahl der Fälle an einer Hand abzählen. Allerdings haben viele Teams, mit denen wir zusammengearbeitet haben, die Zeit auf etwa 2 bis 3 Stunden reduziert. Wenn Sie Arbeitsflussmetriken richtig einsetzen, werden Sie feststellen, dass Sie weit weniger Zeit benötigen als das.

Wenn Sie das richtige MCS-Tool wählen, können Sie den "Was?"-Teil der Sprint Planung buchstäblich auf ein paar Minuten reduzieren. Monte-Carlo-Simulationen sind in sekundenschnelle durchgeführt, und Sie brauchen ein oder zwei Minuten, um die Ergebnisse zu interpretieren. Nehmen Sie sich die Zeit, die Sie brauchen, um den "Warum"-Teil Ihres Sprints herauszufinden und Ihr Sprint Ziel festzulegen. Nehmen Sie sich danach ein paar Minuten Zeit, um die zu erledigenden PBIs auszuwählen, und lassen Sie den "Wie?"-Teil beiseite (da Ihr Prozess gemäß Kapitel 1 bereits etabliert sein sollte), und Ihr Event zur Sprint

Planung sollte in etwa einer Stunde erledigt sein. Vielleicht sogar 30 Minuten, wenn Sie ein klares Produktziel, ein Sprint Ziel und ein gut abgestimmtes Team haben. Und das alles unabhängig von der Länge des Sprints. Keine verschwendete Zeit mehr für Schätzungen. Keine nutzlosen Streitereien mehr darüber, ob es sich um ein 2-Punkt- oder ein 3-Punkt-Projekt handelt. Führen Sie einfach Ihr MCS aus und legen Sie los. Das ist wirklich agil. Und warum? Weil man die Unsicherheit nicht durch die Planung der Arbeit beseitigt, sondern nur durch die Ausführung der Arbeit.

# Schlussfolgerung

Wir wollen wissen, welche (wie viele) PBIs im nächsten Sprint erledigt werden können. Die Lösung für dieses Problem ist einfach: Geben Sie historische Durchsatzdaten in ein MCS ein und interpretieren Sie die Ergebnisse, um probabilistische Prognosen darüber zu erstellen, was erledigt werden kann.

Die Grundannahme bei der Verwendung von MCS für die Sprint Planung ist, dass die Zukunft, die wir vorhersagen wollen, in etwa so aussieht wie die Vergangenheit, für die wir Daten haben. Einer der großen Vorteile von Scrum besteht darin, dass Sie, wenn Sie sich professionell an das Framework halten, im Allgemeinen sicher sind (Wortspiel beabsichtigt), Vergangenheitsdaten für Zukunftsprognosen zu verwenden.

In sehr kurzer Zeit können Sie eine sehr genaue Vorhersage darüber treffen, was in einem Sprint erledigt werden kann, so dass Sie sich schnell auf das konzentrieren können, was wichtig ist - die eigentliche Arbeit. Und das ist genau das Thema, das wir als nächstes behandeln werden.

# Quellen

1. Schwaber, Ken and Sutherland, Jeff "The Scrum Guide" <https://scrumguides.org>, 2020.

# Kapitel 7 - Das Daily Scrum

Wenn Sie der Meinung sind, dass die Verwaltung des Workflows Ihnen die besten Chancen bietet, PBIs zu erledigen, dann sollte der Aktionsplan, der aus Ihrem Daily Scrum hervorgeht, vor allem darauf ausgerichtet sein, Probleme mit dem Arbeitsfluss anzugehen. Die beste Metrik zur Anzeige von Flow-Problemen ist das Work Item Age / Diagramm zur Alterung der Arbeit.

Um zu verstehen, warum das so ist, sollten wir einen Schritt zurückgehen und uns die Flow-Metrik Cycle Time (CT) / Durchlaufzeit noch einmal ansehen. Die meisten Leute denken, dass der Grund, warum Flow die CT so sehr betont, darin liegt, dass wir agile Teams unter Druck setzen können, mehr Dinge schneller zu erledigen. Nichts könnte weiter von der Wahrheit entfernt sein. Der Grund, warum Flow sich um die CT kümmert, ist, dass die CT die Zeit bis zum Kundenfeedback darstellt.

Solange ein PBI nicht in den Händen des Kunden ist, stellt dieses Element nur einen hypothetischen Wert dar. Der Wert kann nur von den Kunden selbst bestimmt werden, und diese Bestimmung kann erst erfolgen, nachdem die PBI durchgeführt wurde. Somit ist CT eigentlich ein Maß für die "Zeit bis zum validierten Feedback" (wir gehen hier natürlich davon aus, dass Ihr DoD irgendeine Art von validiertem Feedback beinhaltet - siehe Kapitel 1 für weitere Informationen).

Die CT selbst kann jedoch nur zu oder nach dem Zeitpunkt berechnet werden, an dem die PBI tatsächlich durchgeführt wird. Davor ist die einzige Messgröße, die wir haben, das Alter der Arbeit. Dieser Alterungsprozess beginnt sofort, wenn die Arbeit beginnt. Und wir wissen, dass der Alterungsprozess erst aufhört, wenn die PBI fertiggestellt (an den Kunden geliefert) ist. Je mehr Arbeiten also altern, desto länger verzögern wir die wertvolle Rückmeldung des Kunden.

Dieses verzögerte Feedback erhöht die Wahrscheinlichkeit, dass bei der Lieferung etwas schief geht. Vielleicht ändert sich das Geschäftsumfeld, vielleicht ändern sich die Kundenanforderungen, vielleicht bricht eine weltweite Pandemie aus - es ist unmöglich zu wissen, was passieren könnte, um die Bedürfnisse eines Kunden zu ändern (vielleicht haben Sie irgendwo gelesen, dass es bei Agile darum geht, auf Veränderungen zu reagieren...). Was wir aber wissen, ist, dass ein höheres Alter das Risiko erhöht, dass ein oder mehrere schlimme

Dinge passieren könnten. Und das größte Risiko besteht darin, dass wir lange Zeit an etwas arbeiten, das am Ende nicht wertvoll ist. Wie Dans Freund und Kollege Prateek Singh zu sagen pflegt: "Es geht darum, so schnell wie möglich herauszufinden, wie falsch man liegt." Wenn Sie Arbeiten unnötig altern lassen, sabotieren Sie nicht nur Ihre Lieferfähigkeit, sondern auch Ihre Fähigkeit, das zu liefern, was Ihre Kunden wirklich wollen.

Eine Arbeit / Work Item, die an und für sich altert, ist jedoch nicht unbedingt etwas Schlechtes. In Wirklichkeit müssen alle Arbeiten bis zu einem gewissen Grad altern, bevor man sie als erledigt betrachten kann. Die Frage, die wir uns daher stellen müssen, lautet: Wie viel Alter ist zu viel Alter?

Die Antwort ist die Service Level Expectation / Erwartung oder SLE. SLEs sind ein weiteres Thema, von dem Sie wahrscheinlich noch nicht viel gehört haben.

# Die SLE

Sowohl im Histogramm der MCS-Ergebnisse als auch im Streudiagramm der Durchlaufzeit haben wir Perzentile verwendet, um die Daten entlang von Unsicherheitslinien zu segmentieren. Bei MCS halfen uns diese Linien, das Risiko zu verstehen, das mit der Erledigung mehrerer PBIs verbunden ist. Das Durchlaufzeit-Streudiagramm ist insofern etwas anders, als dass jeder Punkt im Diagramm angibt, wie lange es dauerte, bis ein einzelnes Element fertiggestellt war. Die Perzentil-Linien auf dem Streudiagramm stellen dann das Risiko dar, das mit der Erledigung eines einzelnen PBIs verbunden ist.

Das Verständnis der Unsicherheit auf der Ebene der einzelnen PBI ist deshalb so wichtig, weil auf der Ebene der einzelnen PBI der tatsächliche Fluss stattfindet. Während sich die einzelnen Elemente durch unseren Prozess bewegen, müssen wir wissen, ob sie so fließen, wie wir es erwarten. Wie bei MCS basiert diese Erwartung darauf, wie viel Unsicherheit wir bereit sind, in Kauf zu nehmen, wenn es darum geht, wie lange ein einzelnes PBI braucht, um in unserem Prozess vom Anfang bis zum Ende zu fließen.

Um diesen Punkt zu veranschaulichen, stellen Sie sich das einmal so vor: Glauben Sie, dass in einem zweiwöchigen Sprint 100 % der PBIs immer innerhalb von 14 Tagen abgeschlossen sein werden? Wir wissen, dass dies unmöglich ist. Wie viel Prozent der Zeit sind Sie also bereit, sich zu irren? 85%? 70%? Oder etwas anderes? An dieser Stelle kommen die Perzentile auf dem



## Kapitel 7 - Das Daily Scrum

Streudiagramm der Durchlaufzeit ins Spiel, denn diese Perzentile geben uns die prozentuale Chance an, ein PBI in einem bestimmten Zeitbereich zu beenden.

Schauen Sie sich noch einmal das Beispiel mit den Perzentilen aus Kapitel 4 an:

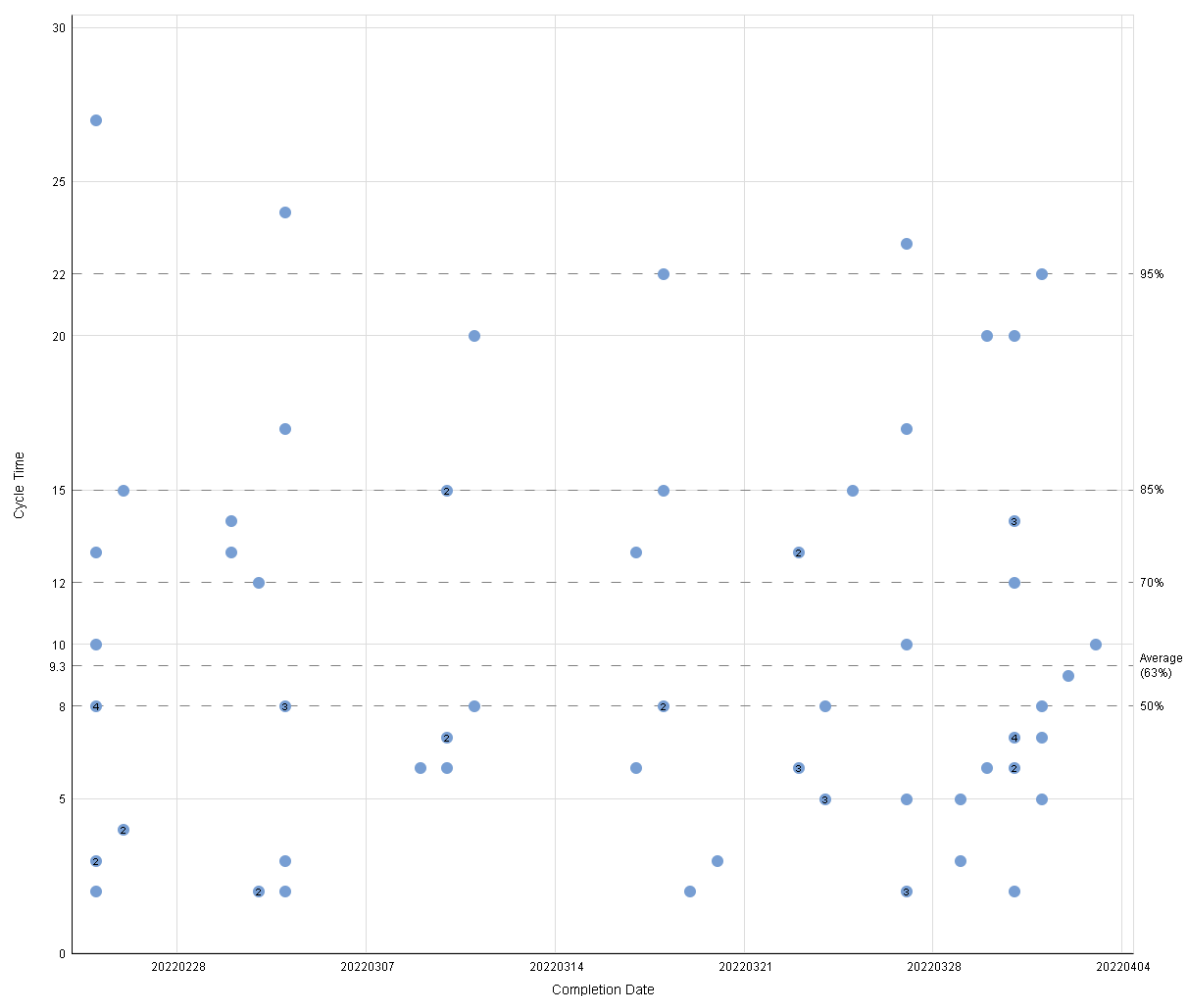


Abbildung - 30<sup>th</sup>, 50<sup>th</sup>, 70<sup>th</sup>, 85<sup>th</sup>, 95<sup>th</sup> und mittlere Perzentil-Linien

Diese Daten sagen uns etwas sehr Interessantes. Sie sagen uns, dass 50 % der PBIs in 8 Tagen oder weniger abgeschlossen sein sollten, dass 85 % der PBIs in 15 Tagen oder weniger abgeschlossen sein sollten und dass 95 % der PBIs in 22 Tagen oder weniger abgeschlossen sein sollten. Basierend auf der Art und Weise, wie wir unseren Prozess derzeit durchführen, sollten alle diese Ergebnisse als Routine angesehen werden. (Nebenbei bemerkt: Wenn Ihnen nicht gefällt, was Ihre Daten Ihnen sagen, dann müssen Sie Ihren Prozess ändern. Scrum hat ein wunderbares Event für genau das, das wir natürlich in einem späteren Kapitel behandeln werden].

Wählen Sie aus diesen Prozentsätzen das Ergebnis aus, mit dem Sie am besten leben können, und diese Spanne und Wahrscheinlichkeit wird dann als Ihre Service-Level-Erwartung (SLE) bezeichnet. Um es in der Sprache von Kapitel 5 auszudrücken: Die SLE ist Ihre Vorhersage, wie lange ein einzelnes Element brauchen sollte, um Ihren Prozess vom Anfang bis zum Ende zu durchlaufen.

Wie können wir also unseren neu gebildeten SLE zur Unterstützung des Daily Scrum anwenden?

# Perzentile als Auslöser für Interventionen

Erinnern Sie sich an die in Kapitel 4 getroffene Aussage:

"Sie werden auch Perzentil-Linien auf dem Aging-Diagramm / Diagramm zur Alterung der Arbeit sehen. Diese Perzentil-Linien sind genau die gleichen Linien, die wir für unser Streudiagramm berechnet haben. Sie überlagern die Perzentil-Linien im Aging-Diagramm, damit Sie sehen können, wie die aktuell in Bearbeitung befindlichen Elemente im Vergleich zur Gesamtzeit, die frühere Elemente zur Fertigstellung benötigten, vorankommen."

Je älter die Aufgaben werden, desto mehr Informationen erhalten wir über sie. Die Prozentsätze auf unserer Alterungstabelle sind perfekte Kontrollpunkte, um unsere neu gewonnenen Informationen zu überprüfen. Wir werden diese Kontrollpunkte nutzen, um während des Daily Scrum unseren Aktionsplan zu erstellen.

Wie funktioniert das? Lassen Sie uns zuerst über das 50. Perzentil sprechen. Und nehmen wir für diese Diskussion an, dass unser Team einen 85%igen SLE verwendet. Wenn ein Element bis zu einem Punkt in Arbeit ist, an dem sein Alter der Durchlaufzeit der 50-Perzentil-Linie entspricht, können wir einige Dinge sagen. Erstens können wir sagen, dass dieses Element nun per Definition mehr Zeit benötigt als die Hälfte der PBIs, die wir zuvor gesehen haben. Das könnte ein Grund sein, innezuhalten. Was haben wir über diesen Punkt herausgefunden, das uns veranlassen könnte, Maßnahmen zu ergreifen? Müssen wir ausschwärmen? Müssen wir es auflösen? Müssen wir die Beseitigung eines Blockers eskalieren? Die Dringlichkeit dieser Fragen ergibt sich aus der zweiten Sache, die wir sagen können, wenn das Alter einer Arbeit das 50stige Perzentil erreicht hat. Als wir das PBI zum ersten Mal in unseren Prozess

aufgenommen haben, bestand eine 15 %ige Chance, dass es gegen sein SLE verstößt (das ist die eigentliche Definition der Verwendung des 85sten Perzentils als SLE). Jetzt, da das Work Item / die Arbeit das 50. Perzentil erreicht hat, hat sich die Wahrscheinlichkeit, dass sie gegen sein SLE verstößt, von 15 % auf 30 % verdoppelt. Auch wenn dies kein Grund zur Besorgnis ist, so sollte es doch zumindest Anlass zu Gesprächen sein. Genau darum geht es bei der Aktionsplanung im Daily Scrum.

Wenn eine Arbeit bis zur 70. Perzentil-Linie gealtert ist, wissen wir, dass sie älter ist als zwei Drittel der anderen Arbeiten, die wir zuvor gesehen haben. Und jetzt ist die Wahrscheinlichkeit, dass sie ihre SLE verpasst, auf 50 % gestiegen. Werfen Sie eine Münze. Die Gespräche, die wir zuvor geführt haben (z. B. Pairing, Swarming, Refinement / Aufteilung der Arbeit), sollten jetzt umso dringlicher werden.

Und sie sollten noch dringlicher werden, wenn sich das Alter des PBI immer mehr dem 85. Perzentil annähert. Das Letzte, was wir wollen, ist, dass diese Arbeit ihre SLE verletzt - auch wenn wir in diesem Beispiel wissen, dass dies in 15 % der Fälle passieren wird. Wir wollen sicherstellen, dass wir alles in unserer Macht stehende getan haben, um einen Verstoß zu verhindern. Der Grund dafür ist, dass ein Verstoß gegen die SLE nicht bedeutet, dass wir plötzlich den Fuß vom Gas nehmen müssen. Wir müssen die Arbeit noch zu Ende bringen. Irgendwo wartet ein Kunde darauf, dass sein Wert geliefert wird.

# Was ist, wenn PBIs nicht wie erwartet fließen?

## Pairing, Swarming, und Mobbing

PBIs offenbaren oft eine bisher unbekannte Komplexität, wenn sie sich durch den Workflow / Arbeitsablauf bewegen. Diese Komplexität kann dazu führen, dass sie mehr altern als andere Elemente. Ein Element, das so sehr gealtert ist, dass es im Kontext des Arbeitsablaufs des Teams auffällt, verdient besondere Aufmerksamkeit. Dies kann in der Form geschehen, dass mehrere Teammitglieder einspringen, um bei diesem Punkt zu helfen (Achtung: Brooks' Gesetz: "Wenn man einem verzögerten Softwareprojekt Arbeitskräfte hinzufügt, wird es sich noch mehr verzögern" ^1^). Das bedeutet oft, dass der WIP reduziert werden muss, damit das

überfällige Projekt, Fortschritte macht. Entwickler, die das nächste Projekt abschließen, sollten gebeten werden, bei dem veralteten PBI mitzuhelfen, anstatt neue Aufgaben zu übernehmen. Die Verringerung des WIP auf eine geringere Anzahl von Teammitgliedern ist unter verschiedenen Namen bekannt - Pairing, Swarming und Mobbing, um nur einige zu nennen. Der Einfachheit halber bezeichnen wir all dies als "Ensemble"-/ Zusammenarbeit.

Es gibt drei wesentliche Möglichkeiten, wie Zusammenarbeit zur Kontrolle des Alters beitragen kann.

- Frühere Erledigung nachgelagerter Aufgaben - Wenn ein Element in einer früheren Phase altert, können wir einen schnelleren Fluss durch die späteren Phasen ermöglichen. Wir können Schritte in den späteren Phasen früher ausführen, damit die Aufgabe nicht unnötig weiter altert, sobald sie die aktuelle Phase hinter sich gelassen hat.
- Aufteilung der PBI-Aufgaben unter den Teammitgliedern - Wenn die Aufgabe selbst nicht in lieferbare Teile zerlegt werden kann, ist es möglich, Teilaufgaben für die Aufgabe zu bestimmen. Verschiedene Teammitglieder können die verschiedenen Teilaufgaben parallel übernehmen, um die Aufgabe voranzubringen.
- Beseitigung von Stolpersteinen - Oft hilft es, neue Perspektiven auf ein Problem zu bekommen, mit dem ein einzelnes Teammitglied konfrontiert war, um kreative Lösungen zu finden. Ganz gleich, ob es sich dabei nur um ein "Rubber Ducking" oder um eine funktionsübergreifende Zusammenarbeit handelt, um neue Perspektiven zu erhalten, sie helfen einer veralteten Aufgabe, Fortschritte zu erzielen.

## **Blockierte Arbeit freigeben**

Jede Arbeit, die blockiert oder zurückgehalten wird, ist per Definition nicht im Fluss. Diese PBIs altern in der Regel aufgrund von internen oder externen Abhängigkeiten. Wenn die Ursache eine interne Abhängigkeit ist, müssen wir unsere Prozessrichtlinien / Policies überprüfen und nach Verbesserungen suchen. Liegt die Ursache in einer externen Abhängigkeit, müssen wir herausfinden, wie wir die Wahrscheinlichkeit dieser externen Abhängigkeit für künftige Arbeiten verringern oder die Auswirkungen dieser Abhängigkeit auf das Alter reduzieren können. Mit anderen Worten: Wie können wir die Abhängigkeit beseitigen oder die Lösungszeit unbedeutend machen? Externes Fachwissen ins Haus zu holen, die Beziehungen zu Partnern/Lieferanten zu verbessern oder die Abhängigkeit vollständig zu beseitigen sind alles Optionen, die wir hier nutzen können. Unabhängig davon, ob es sich um

## Kapitel 7 - Das Daily Scrum

eine interne oder externe Abhängigkeit handelt, müssen wir einige Richtlinien für den Umgang mit blockierter Arbeit aufstellen. Es gibt mindestens drei Stufen von Blockern, die festgelegt werden müssen:

- Wann wird eine Aufgabe als blockiert gekennzeichnet - Wie viel Zeit muss vergehen, bevor wir eine Aufgabe, deren Fortschritt gestoppt wurde, als blockiert kennzeichnen? Handelt es sich dabei um Stunden, Tage oder Wochen?
- Blockierte Aufgaben und WIP-Grenzen - Wie lange sollte eine blockierte Aufgabe auf unsere WIP-Grenzen angerechnet werden und uns daran hindern, andere Arbeiten in Angriff zu nehmen? Führt die Aufnahme in den WIP dazu, dass wir uns stärker auf die Lösung des Problems konzentrieren?
- Entfernen von blockierten Arbeiten aus dem System - An welchem Punkt sollten wir sagen, dass das Work Item / die Arbeit so lange blockiert ist, dass sie nicht mehr relevant ist, sie zu verfolgen? Sollten wir den Posten stornieren oder zurück in das Backlog verschieben?

## Richtiges Sizing

Wenn Sie das Buch "Principles of Product Development Flow" <sup>2</sup> von Don Reinertsen lesen, werden Sie schnell feststellen, dass einer der größten Nachteile für den Arbeitsfluss darin besteht, an zu großen Aufgaben zu arbeiten. In Bezug auf den Flow bedeutet dies die Kontrolle der Losgröße / Badge Size. Wir haben bereits gesehen, dass ein PBI in der Regel deshalb im Prozess stecken bleibt, weil es zu groß ist - es wurde nicht richtig dimensioniert.

Die richtige Dimensionierung ist die Kunst, PBIs in kleinen Wertchargen fließen zu lassen. Das bedeutet, die Dinge in kleine, überschaubare, aber immer noch wertvolle Teile zu zerlegen.

Wir haben im letzten Kapitel über die Praxis des Right Sizing gesprochen, aber als kurze Zusammenfassung hat Prateek Singh einem seiner Teams diese Anleitung gegeben (er verwendete User Stories für PBIs):

"Das Cycle Time Streudiagramm für unser Team zeigt, dass 85% der Stories, an denen wir arbeiten, in 11 Tagen oder weniger fertig werden. Dies ist ein Guide für die richtige Dimensionierung. Jedes Mal, wenn das Team die nächste Story in Angriff nimmt, sollte es sich die Frage stellen: "Ist dies das kleinste Stück von Wert und kann es in 11 Tagen oder weniger erledigt werden? Wenn die Antwort auf diese Fragen ja lautet, großartig, dann ist keine weitere

Verfeinerung / Refinement erforderlich, und die Arbeit kann beginnen. Wenn die Antwort nein lautet, sollten wir versuchen, diese Story zu refinieren. Das ist der Kern des Right Sizing. Jedes Team wird anhand seiner eigenen Daten herausfinden, welche Stories die richtige Größe haben."

Das ist ja alles schön und gut, werden Sie sich fragen, aber wie gehen wir vor, wenn wir erkannt haben, dass die Größenordnung nicht stimmt? Wir sind froh, dass Sie fragen.

### **Aufschlüsselung von PBIs**

Wenn Sie feststellen, dass ein PBI für Ihren Prozess zu groß ist, sollte Ihre erste Hypothese sein, dass dieses große PBI wahrscheinlich aus kleineren, einzeln lieferbaren Wertstücken besteht. Die Zusammenfassung mehrerer Elemente zu einem einzigen PBI kann viele der Vorteile zunichte machen, die das Management für den Arbeitsfluss bietet. Wenn wir z. B. mit einer WIP-Grenze von 1 arbeiten, diese eine Arbeit aber potenziell aus 5 separaten Arbeiten bestehen könnte, ist unser WIP in Wirklichkeit fünfmal größer, als er sichtbar ist. WIP versteckt sich oft in großen PBIs. Um unseren tatsächlichen WIP aufzudecken, sollten wir die Arbeit frühzeitig und häufig in einzelne "rückmeldbare" Teile zerlegen.

Lassen Sie uns nun einige einfache Strategien durchgehen, mit denen wir PBIs aufschlüsseln können. Es handelt sich um eine Liste, die Dan, Will und andere in der Vergangenheit erfolgreich zur Untergliederung von Arbeit verwendet haben. Mehrere dieser Strategien können auch auf ein und dieselbe Aufgabe angewendet werden. Ziel ist es, kleinere Arbeitseinheiten zu schaffen, die uns helfen, schneller Feedback von unseren Kunden zu erhalten.

### **Verwendung von Akzeptanzkriterien**

Die Praxis des Hinzufügens von Benutzerakzeptanzkriterien (AC) hilft uns zu verstehen, welchen Nutzen ein Kunde von der PBI erwarten würde. Wenn das Team an Initiativen arbeitet, die sich in große PBIs aufgliedern, die wiederum aus (künftigen) PBIs in der richtigen Größe bestehen, sollte jede dieser Ebenen Akzeptanzkriterien haben. Auf jeder Ebene sollten wir in der Lage sein, die Arbeit herunterzubrechen, indem wir uns einem 1:1-Verhältnis von Akzeptanzkriterien zum PBI annähern. Das bedeutet nicht, dass jeder Fehler, jede Story, jedes Experiment, jedes Feature oder jede Initiative nur ein Akzeptanzkriterium haben sollte.

## Kapitel 7 - Das Daily Scrum

Stattdessen sollten wir darauf achten, dass jedes PBI eine Mindestanzahl von ACs hat, die uns helfen, Feedback zu bekommen.

Betrachten Sie zum Beispiel das folgende PBI.

"Als Gutachter möchte ich die relevanten Abschnitte einer eingereichten Arbeit aufgeschlüsselt sehen, damit ich sie leicht beurteilen kann.

- AC 1 - Gutachter sollten in der Lage sein, den Titel der Arbeit getrennt vom Hauptteil zu sehen.
- AC 2 - Gutachter sollten die Anzahl der Wörter in der Beschreibung sehen können
- AC 3 - Gutachter können jeden Abschnitt separat bewerten
- AC 4 - Gutachter können die Haupthypothese in einem separaten Abschnitt anzeigen
- AC 5 - Gutachter können optional Details zu den Experimenten herausfiltern, um sie separat zu bewerten.

In diesem Fall kann jedes dieser ACs eine separate PBI sein. Sie können alle unabhängig voneinander an Kunden (intern oder extern) übermittelt werden, damit wir ein Feedback erhalten. Jedes dieser ACs beginnt mit der Lösung eines Kundenproblems und liefert einen Mehrwert, ohne dass die anderen bis zur Fertigstellung aufgehalten werden.

## **Kunden und Personas verwenden**

Viele Teams, die wir im Laufe der Jahre kennengelernt haben, bedienen mit ihren Produkten einen sehr großen Kundenstamm, egal ob es sich um Softwareprodukte, Roboter oder Personalrichtlinien / Policies handelt. Unabhängig vom Produkt sprechen viele dieser Teams über "den Kunden" in einer sehr generischen, allumfassenden Weise. Manchmal sogar mit einem Hauch von Stolz: "Unsere Kunden sind alle Büroangestellten weltweit". Das hört sich zwar für die Presse und künftige Investoren gut an, lässt aber keine große Konzentration in den Teams zu. Viel wichtiger ist, dass es den Umfang der Arbeit vergrößert. Wir können aber auch unser Wissen über unsere Kunden nutzen, um die Arbeit wieder aufzuteilen.

Betrachten Sie das folgende PBI:

"Sichere Anmeldung für Mitarbeiter aktivieren".

Wir könnten dies auf verschiedene Arten für verschiedene Kunden aufteilen:

- Mitarbeiter, die von zu Hause aus arbeiten
- Mitarbeiter, die vom Büro aus arbeiten
- Mitarbeiter in einem bestimmten Land oder einer bestimmten Region
- Mitarbeiter in einer bestimmten Abteilung des Unternehmens
- Mitarbeiter in einer bestimmten Rolle

Die Liste lässt sich beliebig fortsetzen. Wenn wir unsere Kunden genauer kennen, können wir ihnen schneller etwas liefern und erhalten schneller ein Feedback. Wir haben festgestellt, dass dies sowohl für interne als auch für externe Kunden funktioniert. Nicht alle Kunden sind gleich, und das können wir zu unserem Vorteil nutzen.

### **Annahmen**

Ok, jetzt kommen wir zu einem Thema, mit dem Dan weniger gut zurechtkommt: Ergebnisse. Aber wie Jeff Gothelf und Josh Seiden in ihrem Buch *Lean UX*<sup>3</sup> beschreiben, ist jedes Design ein Lösungsvorschlag - eine Hypothese. Ihr Ziel ist es, die vorgeschlagene Lösung so effizient wie möglich durch Kundenfeedback zu validieren." Mit anderen Worten, wie wir bereits in diesem Buch gesagt haben, sind alle PBIs eine Wertannahme, bis sie vom Kunden validiert werden.

Wenn wir also ein PBI aufschlüsseln wollen, können wir unter anderem fragen: "Welche Annahmen machen wir über den Wert dieser Sache?". Daraufhin können wir die Arbeit in einzelne Annahmen aufteilen und diese separat liefern. Nehmen wir das Beispiel von vorhin, so könnten wir den Punkt in einzelne Annahmen aufteilen, wie z. B.:

- Die aktuelle Anmeldung ist nicht sicher
- Alle Mitarbeiter benötigen ähnliche Sicherheitspraktiken
- Alle Länder benötigen ähnliche Sicherheitspraktiken
- Das von uns entwickelte sichere Login ist sicher
- Die Sicherheit lässt sich am besten durch eine technische Lösung durchsetzen.

Ein weiterer Ansatz, den wir verwenden könnten, möglicherweise in Kombination, ist Giff Constables *Wahrheitskurve*<sup>4</sup>. Die Arbeit ist eine gute Lektüre, aber im Wesentlichen geht es darum, Folgendes herauszufinden: Was ist der geringste Aufwand, den wir betreiben sollten,



um herauszufinden, ob es sich lohnt, mehr Zeit/Budget in diese PBI zu investieren und die Beweise für ihren Wert zu erweitern? In unserem Fall sollte dieser Aufwand bequem in Ihre SLE passen. Um bei dem Beispiel der Sicherheit zu bleiben, würden wir wahrscheinlich damit beginnen:

- Einige Gespräche, um zu prüfen, warum dieses Element überhaupt existiert: Ist es eine gesetzliche Anforderung? Ein Problem der Einhaltung von Vorschriften? Funktioniert das derzeitige System nicht?
- Wenn diese PBIs auf einen Änderungsbedarf hindeuten, führen wir eine Beobachtungsstudie der derzeitigen Benutzer durch: Was passiert jetzt? Warum sind die derzeitigen Richtlinien / Policies unzureichend?
- Führen Sie eine Prototyping-Sitzung auf Papier durch, um die neue sichere Anmeldefunktion zu testen. Ist sie einfach? Sicher? Narrensicher?
- Führen Sie ein "Wizard of Oz"-Experiment durch, um herauszufinden, ob die Benutzer sie nutzen, ohne zu wissen, dass sie beobachtet werden.
- Bauen Sie einen Prototyp, um die Technologie zu validieren.

Die Liste ließe sich beliebig fortsetzen, aber ich hoffe, Sie verstehen, worum es geht. Alle diese PBIs können viel schneller und mit schnellerem Feedback umgesetzt werden als der große Ausgangspunkt "Sichere Anmeldung für Mitarbeiter ermöglichen".

# Ein tägliches Scrum-Beispiel

Nehmen wir an, die Entwickler eines Scrum Teams kommen zu ihrem Daily Scrum und erstellen ein „WIP-Aging-Diagramm“ / Diagramm zur Alterung der Arbeit, das wie folgt aussieht:

## Kapitel 7 - Das Daily Scrum

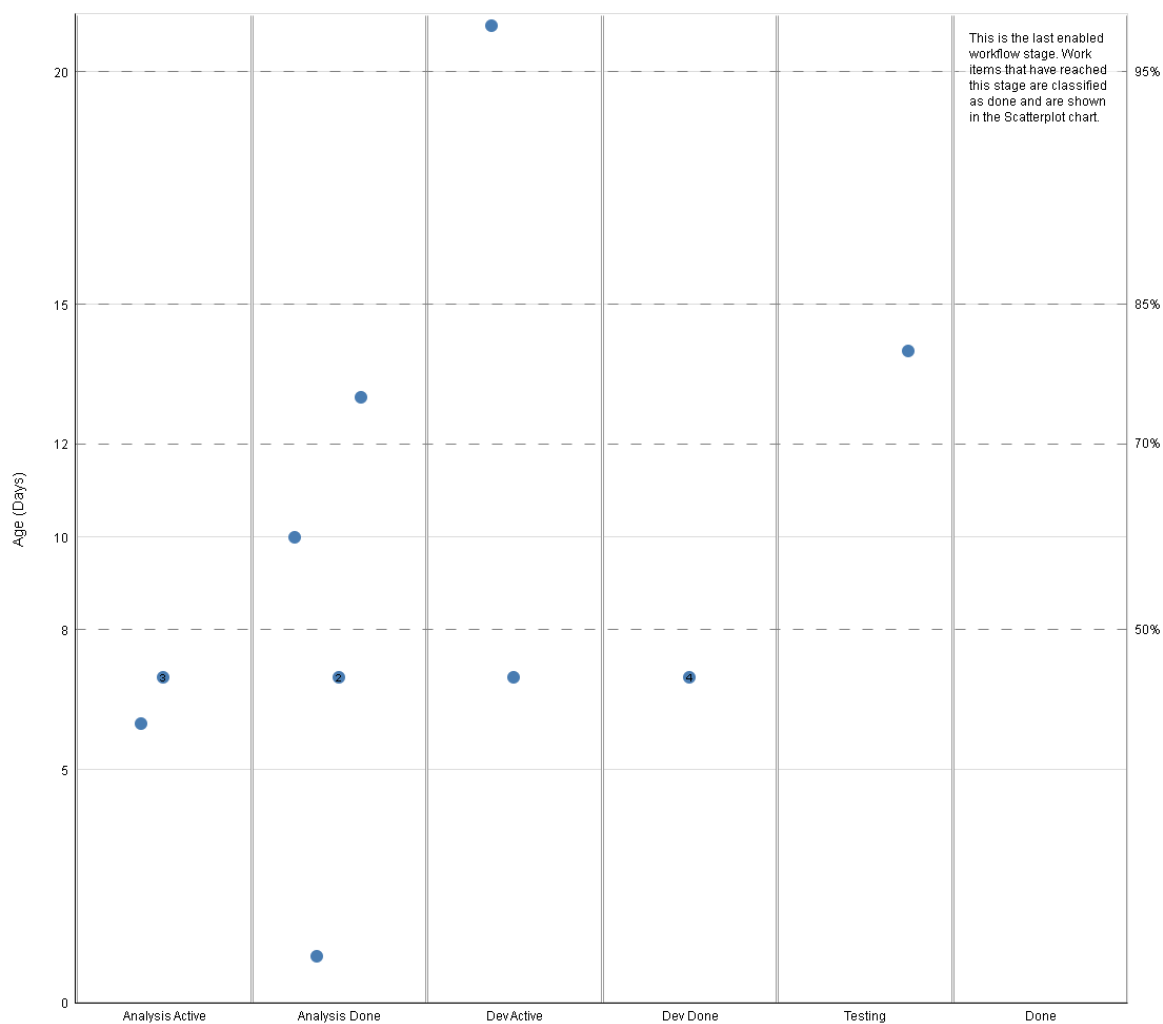


Abbildung 7.1: Oh nein!

Was sollten die Entwickler auf der Grundlage dieser Informationen tun?

Zunächst sollten wir darauf hinweisen, dass alle Daten und Metriken kontextspezifisch sind, so dass dieses Beispiel nur zur Veranschaulichung dient. Die Entscheidung, die Ihr Team auf der Grundlage von Kontextinformationen trifft, kann sich von diesen unterscheiden, und das ist auch in Ordnung so - es ist sogar empfehlenswert. Im Folgenden wird jedoch eine Strategie vorgestellt, die in den meisten Fällen funktionieren sollte.

- **Betrachten Sie zuerst die ältesten Arbeiten / Work Items:** Welches ist das älteste Element auf dem Board und wird es gerade bearbeitet? Was müssen wir tun, um bei diesem Punkt / Work Item heute Fortschritte zu machen? Muss er abgebrochen werden? Müssen wir dafür swarmen/mobben/paaren? Ist er noch

wertvoll? Wiederholen Sie den Vorgang für den nächstältesten Punkt, bis das Team seinen Plan für den Tag hat.

- **Notieren Sie überschüssiges WIP:** Es gibt 4 Dinge in "Erledigt", was passiert dort? Wir müssten natürlich einen Blick auf das Kanban Board werfen, um zu wissen, ob 4 Arbeiten / Work Items überflüssiges WIP sind oder nicht.
- **Diskutieren Sie Arbeiten, bei denen die Gefahr besteht, dass sie die SLE verletzen:** Es gibt Arbeit „in Analyse erledigt“, die bereits 13 Tage alt ist. Die Chance, sie noch innerhalb der SLE abzuschließen, ist zu diesem Zeitpunkt ziemlich gering. Sollten wir swarmen? Oder schauen, ob wir sie in den nächsten zwei Tagen noch fertigstellen können (unter der Annahme, dass das 85<sup>te</sup> Perzentil Ihr SLE ist)?

Die einfache Heuristik für Daily Scrums ist - unter sonst gleichen Bedingungen - dass Sie Ihre Aufmerksamkeit zuerst auf die ältesten PBIs richten sollten. Um den Arbeitsfluss zu optimieren, wollen wir das Alter minimieren, was bedeutet, dass die ältesten PBIs in Ihrer Alterstabelle ein guter Indikator dafür sind, dass etwas nicht stimmt. Das Letzte, was Sie tun sollten, ist, alte Work Items / Arbeiten zu ignorieren, denn das führt nur dazu, dass diese PBIs noch älter werden.

# Schlussfolgerung

Das Daily Scrum ist vielleicht das Ereignis, auf das sich die Verwendung von Arbeitsflussmetriken am stärksten auswirkt. Während die Sprint Planung mit diesem Tool dramatisch kürzer ist, ändert sich die Natur dieses Events nicht allzu sehr. Man legt immer noch ein Sprint Ziel fest und versucht zu prognostizieren, welche Arbeit geleistet werden kann, um dieses Ziel zu erreichen. Beim Daily Scrum hingegen gibt es eine Änderung. Das Ziel des Daily Scrum bleibt dasselbe: den Fortschritt in Richtung des Sprint Ziels zu überprüfen. Aber mit dem Einsatz von Arbeitsflussmetriken ändert sich die Konversation von "wie arbeiten wir heute zusammen und helfen uns gegenseitig, das Sprint Ziel zu erreichen" (personenorientiert) zu "wie alt ist das Zeug und wie können wir es vorantreiben" (arbeitsorientiert). Die Verwendung von Arbeitsflussmetriken in Ihrem Daily Scrum fördert tatsächlich die Teamarbeit, indem sie sich auf die zu erledigende Arbeit konzentriert. Denn wenn sich die Arbeit bewegt und erledigt wird, erhalten Sie das Feedback, das Sie brauchen, um Ihren Fortschritt in Richtung des Sprint Ziels zu überprüfen, und nicht nur die Dinge, von denen Sie glauben, dass Sie sie liefern müssen.

# Quellen

1. Brooks, F. (1995). *Mythical Man-Month, The: Essays on Software Engineering, Anniversary Edition (Anniversary ed.)*. Addison-Wesley Professional.
2. Reinertsen, Donald G. "The Principles of Product Development Flow". Celeritas Publishing, 2009.
3. Gothelf, J., & Seiden, J. (2021). "Lean UX: Designing Great Products with Agile Teams (3rd ed.)". O'Reilly Media.
4. Constable, G. (2021, April 8). "The Truth Curve and the Build Curve". <https://giffconstable.com>.

# Kapitel 8 - Das Sprint Review

Die Verwendung von Arbeitsflussmetriken in einem Sprint Review kann leicht dazu führen, dass das Gespräch in eine Diskussion über den Umfang abgleitet. Andererseits kann der kluge Einsatz von Arbeitsflussmetriken die Diskussion mit Ihren Stakeholdern über Zeitplanung, Budgets und Ergebnisse eröffnen. Unabhängig davon können Sie mit Arbeitsflussmetriken die Diskussion auf die Zukunft ausrichten. Viele Teams tun sich immer noch schwer damit, "Demos" durchzuführen, da alle Anwesenden (zu Recht) das Gefühl haben, dass dies Zeitverschwendung ist. Wenn dies nicht einfach auf mangelndes Wissen darüber zurückzuführen ist, worum es bei der Veranstaltung gehen sollte, liegt es vor allem daran, dass sie nicht wirklich etwas über die Zukunft sagen können. Nun, jetzt nicht mehr. Denn jetzt haben Sie Kennzahlen, Diagramme und Antworten auf die dringendsten Fragen Ihrer Stakeholder.

Lassen Sie uns darüber sprechen, wie das alles aussehen könnte.

## Was haben wir im letzten Sprint gelernt?

Jedes gute Review (und Will ist bei diesem Thema sehr eigenwillig) beginnt damit, dass alle Anwesenden sich wieder auf das Produktziel und die Vision, die Sie zu erreichen versuchen, und das Sprint Ziel, das Sie für diesen Sprint hatten, alignen / fokussieren. Nachdem Sie alle daran erinnert haben, warum Sie Zeit (und Geld!) in die Entwicklung von etwas investiert haben, können Sie darüber sprechen, was Sie in diesem Sprint gelernt haben. Aus dem Scrum Guide<sup>1</sup>: "Der Zweck des Sprint Review ist es, das Ergebnis des Sprints zu überprüfen und zukünftige Anpassungen zu bestimmen." Das mag jetzt umstritten sein, aber wir glauben nicht, dass dies unbedingt bedeutet, dass Sie Ihr Produkt vorführen müssen (obwohl es aus der Sicht von "show, don't tell" hilfreich sein könnte). Vielmehr sollten Sie darüber sprechen, was beim Sprint herausgekommen ist / ob Sie Ihren Kunden einen Mehrwert bieten, indem Sie den

aktuellen Stand des Produkts nutzen, um die Punkte des erhaltenen Feedbacks zu veranschaulichen.

Im Anschluss daran können Sie transparent machen, was Sie aus Ihrem Lieferprozess gelernt haben. Hier können Sie anfangen, Arbeitsflussmetriken zu verwenden. Sind Sie auf irgendwelche Verzögerungen gestoßen? Vielleicht aufgrund von Abhängigkeiten außerhalb des Teams oder aufgrund von Arbeit, die von nachgelagerten Stellen zurückgeschickt wurde (wenn Sie nicht an den Endkunden liefern)? Wenn Sie hier Ihr Durchlaufzeit-Streudiagramm anzeigen und in den letzten Sprint hineinzoomen, können Sie die Auswirkungen von Problemen erkennen, bei denen Ihre Stakeholder Ihnen helfen könnten. Vielleicht zeigt es auch nur, warum Sie von Ihrem ursprünglichen Umfang abweichen mussten, um das Sprint Ziel zu erreichen.

Dies ist auch ein guter Zeitpunkt für Stakeholder, um über Markt- oder Geschäftsentwicklungen zu berichten, die sich auf Ihren Arbeitsablauf auswirken könnten. Vielleicht treten sie in ausländische Märkte ein, die Änderungen in Ihrem Validierungsprozess erfordern. Oder Sie erweitern endlich die Arbeitsabläufe Ihrer Teams weiter nach unten, denn Übergaben sind schlecht (da sie zu Verzögerungen und Wissensverlusten führen können). Unabhängig davon sollte der erste Teil Ihres Sprint Reviews eine Überprüfung des aktuellen Stands in Bezug auf Wert und Prozess sein.

# Wann ist er abgeschlossen?

Dan hat ein Buch über dieses Thema geschrieben. Das war ein ziemlich kluger Schachzug, wenn man bedenkt, wie weit verbreitet diese Frage in modernen Organisationen noch immer ist. Die Frage, ob "es" jemals "fertig" sein wird oder nicht, ist ein großes Fass ohne Boden, das wir hier nicht öffnen wollen. Wenn Sie sich für solche Dinge interessieren, empfehlen wir Ihnen, andere Quellen zu Rate zu ziehen. Manchmal ist es eine berechtigte Frage. Dinge wie gesetzliche Anforderungen oder vertragliche Vereinbarungen zur Funktionalität haben sicherlich einen klar definierten Umfang, der erfüllt werden muss. In den meisten Fällen ist dies jedoch nicht der Fall, da die Komplexität der Arbeit bedeutet, dass man nicht weiß, ob Funktion A zu Ergebnis A führen wird, bis man sie gebaut hat.

All das ist gesagt: Wenn man diese Frage nicht beantworten kann, unabhängig von ihrer Gültigkeit, werden die Leute sehr nervös. Glücklicherweise können wir diese Frage mit unserer bewährten Monte-Carlo-Technik einigermaßen beantworten. Das Verfahren für eine *Monte-*

## Kapitel 8 - Das Sprint Review

*Carlo-Prognose: Wann ist eine Wahrscheinlichkeitsprognose etwas anders als die Monte-Carlo-Prognose: Wie viel Arbeit wird fertig / How Much.?*, deshalb gehen wir es kurz durch.

Nehmen Sie eine bestimmte Anzahl von Aufgaben, die Sie erledigen möchten. Dabei kann es sich um eine beliebige Anzahl handeln (10, 50, 100 usw.) oder um die Anzahl, die in Ihrem Product Backlog steht. Nehmen wir für dieses Beispiel **10**. Schauen Sie sich nun Ihre historischen täglichen Durchsatzzahlen an, die wir für die Prognose verwenden werden. Nehmen wir an, wir haben in den letzten 6 Tagen **2, 1, 2, 2, 0, 1** Arbeiten / Work Items pro Tag ausgeliefert. Für die Zukunft können wir diese Werte als möglichen Durchsatz verwenden und sehen, wie viel Zeit wir dafür benötigen würden, etwa so:

Abbildung 8.1: Beispiel für eine Durchsatzprognose

<b>Day</b>	<b>Throughput</b>	<b>Remaining</b>
1	2	8
2	0	8
3	2	6
4	1	5
5	2	3
6	2	1
7	1	0

In diesem "Lauf" würde es also 7 Tage dauern, diese 10 Sendungen zuzustellen. Dies ist jedoch noch keine probabilistische Prognose. Wenn Sie sich an die vorangegangenen Kapitel erinnern, sollten Sie diese Simulation noch ein paar Mal durchführen (etwa 1000-10.000 Mal). Dabei werden wir sehen, wie unsere vertraute Verteilung der Möglichkeiten aussieht, und wir können wieder zählen, wie oft sich bestimmte Gelegenheiten ergeben. Hier sehen wir jedoch, dass die Wahrscheinlichkeit, mindestens 10 Arbeiten zu liefern, mit zunehmender Zeitspanne ansteigt.

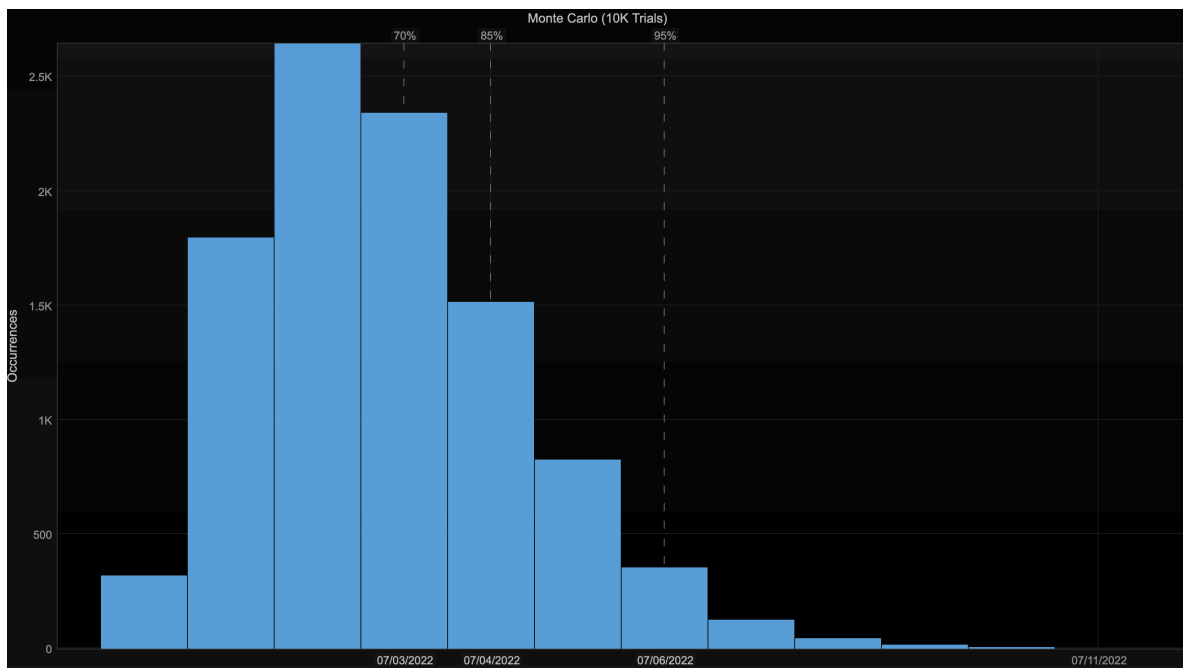


Abbildung 8.2: Monte Carlo: Wann werden wie viele Arbeiten fertig / When?

Wenn Sie diese Technik mit Ihren eigenen Daten anwenden, können Sie eine probabilistische Antwort auf die Frage "Wann wird es fertig sein?" bekommen. Wir können allerdings nicht garantieren, dass Ihre Stakeholder die Antwort mögen werden. Um die Nerven zu beruhigen, können Sie dann darüber sprechen, ob alle diese Punkte notwendig sind, welches Risiko die Beteiligten in Kauf nehmen würden und ob genug Budget oder Zeit übrig ist, um die geplanten Termine zu erreichen.

## Was kann getan werden?

Natürlich hat nicht jedes Team mit einem festen Umfang zu tun (zum Glück) oder sogar mit einem bekannten Umfang. Manchmal findet man sich auch in der umgekehrten Situation wieder: Ihnen steht nur eine bestimmte Zeit und/oder ein bestimmtes Budget zur Verfügung, und Sie versuchen, in dieser Zeit Ihr Bestes zu geben. In einem solchen Fall würden Sie eine *Monte-Carlo-Prognose* verwenden. Dieselbe, die Sie bereits in Ihrer Sprint Planung verwenden, allerdings mit einem Unterschied: Anstelle einer Prognose bis zum Ende des Sprints machen Sie eine Prognose bis zu dem Datum, an dem Ihre Zeit und/oder Ihr Budget abläuft. So erhalten Sie eine Reihe von Elementen und die zugehörige Wahrscheinlichkeit, die dann als Grundlage für Ihre Gespräche über die zu erwartenden Arbeiten / Work Items dienen kann.



## Kapitel 8 - Das Sprint Review

Es könnte aber auch sein, dass es viel weniger ist, als Ihre Stakeholder sich erhofft haben, und Sie können sich auf eine nette Budgetdiskussion gefasst machen. Oder ein Wortgefecht.

Hier ist ein Beispiel dafür, wie eine *How Many-Projektion* in einem Sprint Review aussehen könnte, wenn Sie von 60 Tagen ausgehen (weil dann das Geld ausgeht). Beachten Sie, dass hier ein vollständigerer Datensatz als im vorherigen Beispiel verwendet wird.

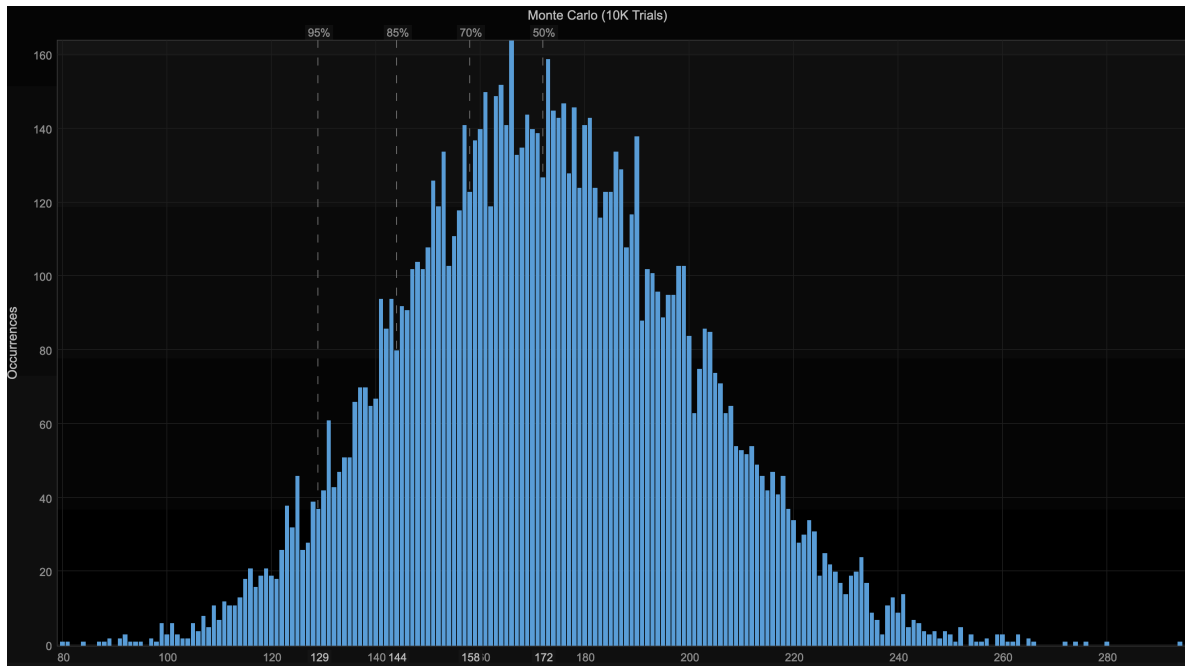


Abbildung 8.3: Monte Carlo: Wie viele

Wenn Sie es mit einem vertraglich vereinbarten Umfang von 120 Arbeiten zu tun haben, sieht es hier ziemlich gut aus. Es besteht eine 95%ige Chance, MINDESTENS 129 Elemente zu liefern. Die Stakeholder werden zufrieden sein, und Sie sind bereit für eine Wertdiskussion / wie wertvoll ist es?

Wenn der Umfang größer ist als dieser, wird es natürlich immer unwahrscheinlicher, dass Sie alles liefern können. Wenn Ihr Umfang beispielsweise 170 Elemente umfasst, ist es fast ein Münzwurf, wenn die 50 %-Linie bei 172 liegt. Bitte beachten Sie (auch für Ihre Stakeholder), dass dies bedeutet, dass Sie in 50 % der Projektionen NACH 172 Punkten geliefert haben. Es handelt sich nicht um eine 50/50-Chance, genau 172 Arbeiten zu liefern. In jedem Fall könnte dies der Moment sein, in dem Sie die Idee aufbringen, eine Veröffentlichung / Release zu verschieben, Prioritäten zu setzen (De-Scoping), das Team weiter zu befähigen oder

Verzögerungen zu beseitigen, oder im schlimmsten Fall: mehr Ressourcen einzusetzen. Letzteres funktioniert (fast) nie.

# Was nun?

Natürlich ist alles, was bis hierher geschah, auf den Output bezogen. Und wenn es nur auf den Output ankäme, wäre dies das Ende der Diskussion. Aber Scrum Teams zielen darauf ab, Wert zu liefern. Ihr Sprint Review sollte sich darauf konzentrieren, die Erkenntnisse aus Ihrem Sprint, das Feedback zu Ihrem Produkt, die Wertmaßstäbe und die Überprüfung des Marktes zu nutzen, um Ihren Plan (Ihr Product Backlog) zu aktualisieren. Arbeitsflussmetriken sollen Ihnen bei diesem Prozess helfen, ihn nicht ersetzen. Die Wahrscheinlichkeitsprognose kann Ihnen Aufschluss darüber geben, wie viele Punkte erledigt werden könnten, aber Ihr Sprint Review ist dazu da, herauszufinden, welche Punkte das sein könnten.

Vergessen Sie aber nicht, Ihr Team und die Stakeholder immer an die Prozentsätze zu erinnern, die mit Ihren Prognosen verbunden sind. In einem komplexen Umfeld gibt es keine Garantien, so dass sich Ihre Prognosen im Laufe der Zeit ändern können (und wahrscheinlich auch werden). Machen Sie also diese Gespräche und Maßnahmen zu einem Standardbestandteil Ihrer Sprint Reviews.

Erforschen Sie also mit Arbeitsflussmetriken als Grundlage diesen Wert!

# Schlussfolgerung

Bis jetzt haben wir nur über die Verwendung von Arbeitsflussmetriken als Scrum Team zur Steuerung Ihres Workflows gesprochen. In diesem Kapitel haben wir die Verwendung von Arbeitsflussmetriken über Ihr Team hinaus und als Teil des Produktmanagements untersucht. Und wir haben Sie hoffentlich von dem enormen Wert überzeugt, den sie für die Verbesserung Ihrer Beziehungen zu Ihren Stakeholdern bieten können. Von der datenbasierten Antwort auf die Frage "Wann wird es fertig sein?" bis hin zur Vorhersage, wie viele Aufgaben in einer bestimmten Zeitspanne erledigt werden können: Mit Hilfe Ihrer Daten können Sie die Konversation ankurbeln, indem Sie die ergebnisbezogenen Fragen überspringen und schneller zur Wertschöpfung übergehen.

# Quellen

1. Schwaber, Ken and Sutherland, Jeff "The Scrum Guide" Scrumguides.org, 2020

# Kapitel 9 - Die Sprint Retrospektive

Dan's Freund und Kollege Frank Vega weist gerne darauf hin, dass "Ihre Richtlinien / Policies Ihre Daten formen und Ihre Daten Ihre Richtlinien formen". Mit anderen Worten: Ihre Daten erzählen genau die Geschichte des Prozesses, den Sie entworfen haben. Nicht der, den Sie sich gewünscht haben, nicht der, den Sie erwartet haben, sondern der, den Sie implementiert haben und jeden Tag ausführen. Diese Geschichte wird im Durchlaufzeit-Streudiagramm festgehalten, und einer der besten Zeitpunkte, um diese Geschichte zu lesen, ist während der Sprint Retrospektive.

Das Durchlaufzeit Streudiagramm erfasst nicht nur die Geschichte Ihres aktuellen Sprints, sondern auch die aller vergangenen Sprints, für die Sie Flussdaten erfasst haben. Der Vorteil dieser historischen Aufzeichnung ist, dass Sie bestimmte Muster und Cluster erkennen können, die sich im Laufe der Zeit in Ihrem Streudiagramm abzeichnen. Es sind diese Formen, die uns den größten Anhaltspunkt dafür geben, worauf wir unsere Bemühungen zur Prozessverbesserung im nächsten Sprint und darüber hinaus konzentrieren sollten.

Im Folgenden werden einige dieser Muster besprochen und wie sie im Hinblick auf Verbesserungen interpretiert werden können. Viele dieser Informationen sind eine Wiederholung dessen, was Dan zuerst in seinem Buch "Actionable Agile Metrics for Predictability" veröffentlicht hat<sup>1</sup>. Noch einmal: Wenn Sie mit diesem Buch vertraut sind, können Sie dieses Kapitel getrost überspringen, ohne die Kontinuität zu verlieren. Da das ursprüngliche Buch jedoch mit der Absicht veröffentlicht wurde, prozessunabhängig zu sein, könnte es sinnvoll sein, einige der folgenden Details mit einem Scrum Blick zu betrachten.

## Das Dreieck

Ein dreieckiges Muster, wie in Abbildung 9.1 dargestellt, tritt in jeder Situation auf, in der die Durchlaufzeit mit der Zeit zunimmt.

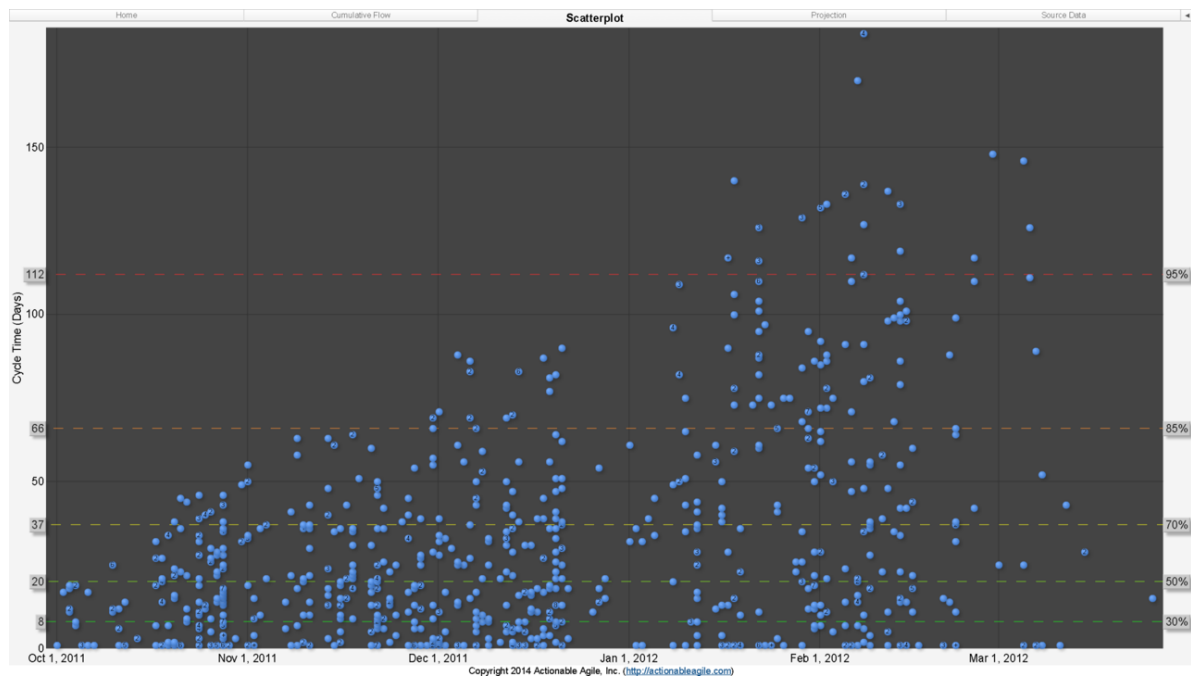


Abbildung 9.1: Ein dreieckiges Muster in einem Streudiagramm

Beachten Sie, dass die Punkte im obigen Streudiagramm ein Muster bilden, das einem Dreieck ähnelt. Um dieses Phänomen zu erklären, müssen wir uns die grundlegende Eigenschaft von Streudiagrammen vergegenwärtigen: Die Punkte erscheinen erst dann, wenn ein Work Item / eine Arbeit beendet ist. Die Elemente mit längeren Durchlaufzeiten benötigen einen längeren Zeitraum, bevor sie im Diagramm erscheinen. Das heißt, je länger die Durchlaufzeit (die Y-Komponente des Punktes), desto länger die Zeit, die wir warten müssen (die X-Komponente des Punktes), um diesen Datenpunkt zu sehen.

Das Dreiecksmuster tritt immer dann auf, wenn Sie mit einem Prozess mit Null WIP beginnen. Das liegt daran, dass es Zeit braucht, um "die Pumpe anzufahren" und die Elemente zu erledigen. Es ist klar, dass in diesen frühen Phasen die Arbeit schneller reinkommt als sie weggeht - auch wenn wir den WIP begrenzen. Wir brauchen Zeit, damit sich jeder Arbeitsablaufschritt bis zu seiner Kapazität füllt und ein vorhersehbarer Fluss in Gang kommt. Dieses Muster wird noch verschärft, wenn Teams das Gefühl haben, dass sie den Prozess in regelmäßigen Abständen leeren müssen. In diesem Fall werden Sie im Laufe der Zeit ein sich wiederholendes Dreiecksmuster sehen, das auf ein Problem mit dem Arbeitsfluss hinweist. Ein Beispiel dafür ist in Abbildung 9.2 zu sehen:

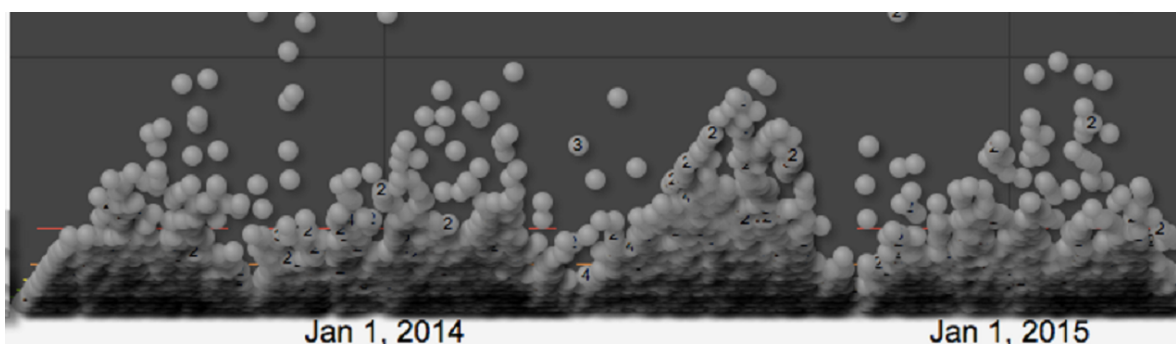


Abbildung 9.2: Ein Dreiecksmuster auf einem Streudiagramm

Obwohl die Zeiträume der beiden Abbildungen 9.1 und 9.2 viel länger sind als ein Sprint, kann man dennoch erkennen, wie dieses Muster im Laufe der Zeit in Scrum entstehen könnte, wenn man bedenkt, was wir gerade oben erklärt haben. Aus der Flussperspektive ist Abbildung 9.2 schrecklich: Das Team hat mit allem ungefähr zur gleichen Zeit begonnen, was einen massiven WIP zeigt. Aus der Scrum Perspektive bedeutet es lediglich, dass dieses Muster in der Sprint Retrospektive besprochen werden sollte. Als Übung für den Leser: Welche Fragen könnten Sie stellen, wenn Sie das Muster in Abbildung 9.2 in jedem Sprint wiedererkennen?

## Cluster von Punkten

Die zweite Art von Muster, die auftauchen könnte, ist eine offensichtliche Anhäufung von Punkten in Ihrem Streudiagramm. Betrachten Sie z. B. das folgende Diagramm in Abbildung 9.3:

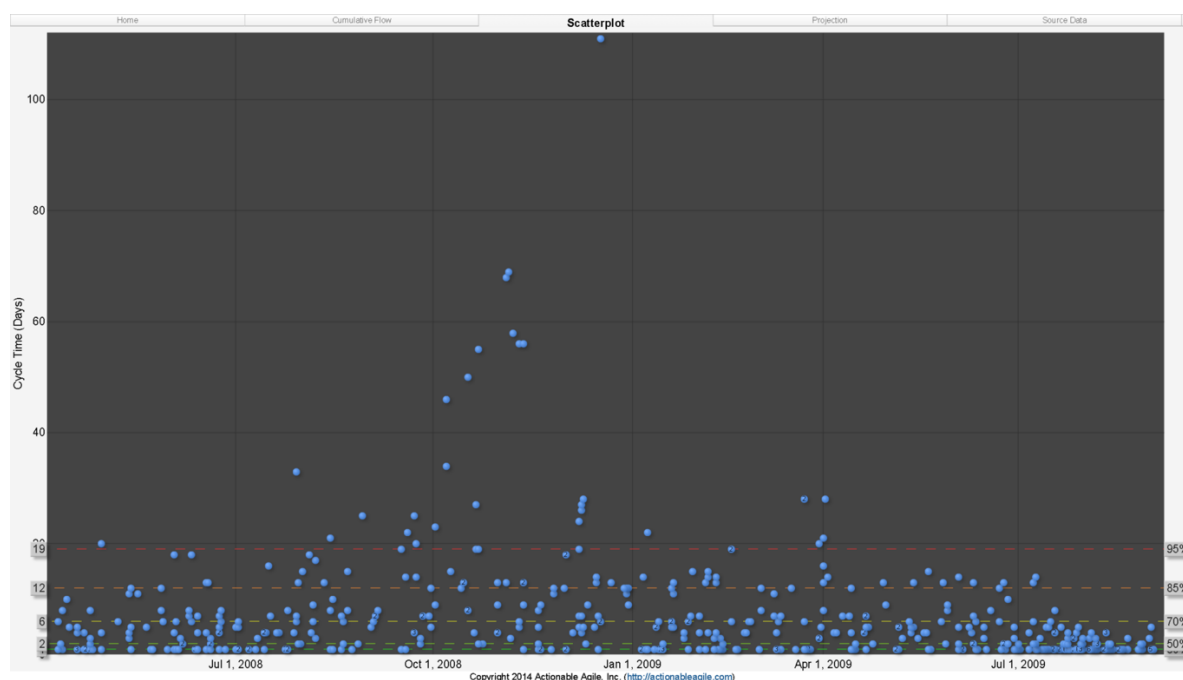


Abbildung 9.3: Cluster in einem Streudiagramm

Beachten Sie die Häufung von Punkten Anfang Oktober 2008 (etwa in der Mitte von Abbildung 9.3) und Ende Juli 2009 (unten rechts in Abbildung 9.3). Wie bei all diesen Analysen geht es auch hier darum, die richtigen Fragen früher zu stellen. Wenn wir also eine Ansammlung von Punkten wie in Abbildung 8.11 sehen, sollten wir uns zumindest fragen: "Was ist hier los?" Darauf sollte wahrscheinlich schnell die Frage folgen: "Ist das eine gute oder eine schlechte Sache?" Wenn es schlecht ist, was können wir dagegen tun?

Übrigens sind nicht alle Häufungen von sehr niedrigen Durchlaufzeiten gut. Sehen Sie sich noch einmal die Punktgruppe für Juli 2009 in Abbildung 8.11 an. Was könnte Ihrer Meinung nach der Grund dafür sein, dass sich unsere Durchlaufzeiten so radikal verringert haben? Fallen Ihnen nur gute Gründe ein? Was könnten schlechte Gründe sein, die dies bewirken könnten? Ein unheilvoller Grund, den wir nur allzu oft sehen, sind obligatorische Überstunden. Wenn Ihre normalen Daten auf 8-Stunden-Tagen und 5-Tage-Wochen basieren, liegt es auf der Hand, dass die Umstellung auf 12-Stunden-Tage und 7-Tage-Wochen Ihre Durchlaufzeit wahrscheinlich besser aussehen lässt (vorausgesetzt natürlich, dass Sie den WIP weiterhin begrenzen!) Aber ist das eine gute Sache? Einige Manager würden das bejahen. Wir würden das Gegenteil behaupten. Und aus Sicht der Vorhersagbarkeit ist es schrecklich. Nicht nur, dass lange Perioden / Zeiträume mit obligatorischen Überstunden nicht tragbar sind, sie verfälschen auch unsere Daten. Wollen Sie wirklich eine Service Level Erwartung / SLE anbieten oder eine Prognose erstellen, in der obligatorische Überstunden als eine der

Annahmen von vornherein fest eingeplant sind? Wenn Ihre Antwort auf diese Frage "ja" lautet, dann ist dieses Buch nichts für Sie.

# Lücken

Lücken in den Punkten auf Ihrem Streudiagramm bedeuten, dass in diesem bestimmten Zeitintervall keine Arbeitspunkte abgeschlossen wurden:

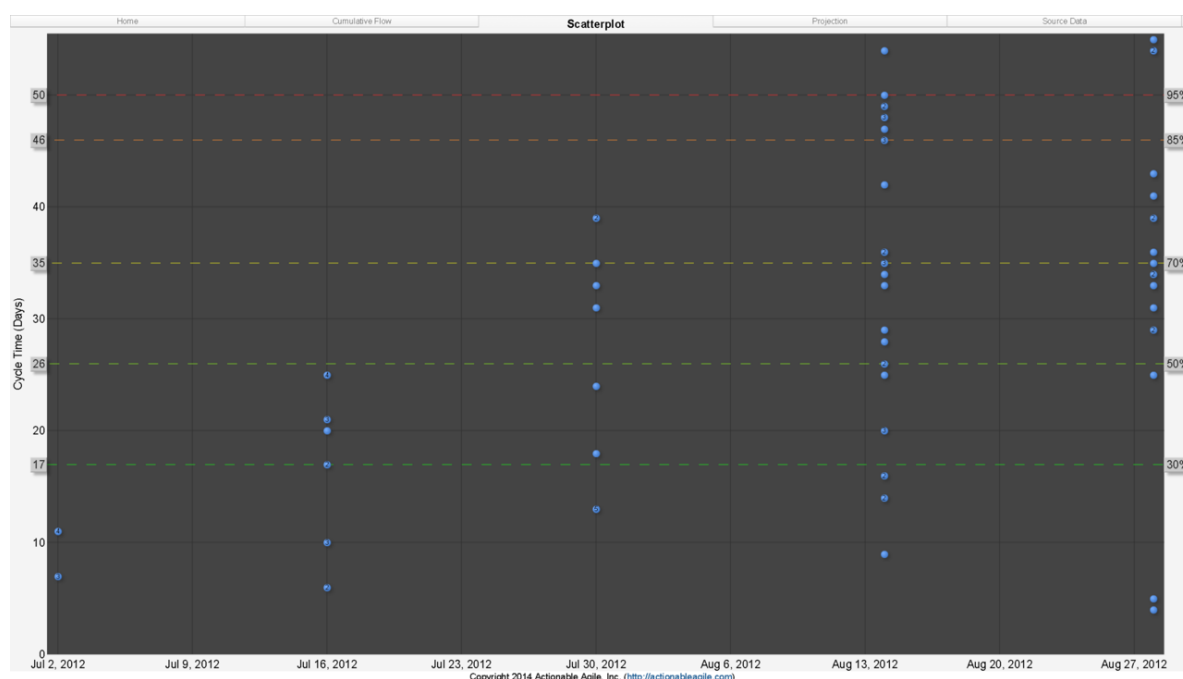


Abbildung 9.4: Lücken in einem Streudiagramm

Kurz gesagt, die Linien, die Sie in Abbildung 9.4 sehen, deuten auf eine Stapelübertragung / Übertragung der Arbeiten auf einen anderen Sprint hin. Es ist nicht ungewöhnlich, dass ein Scrum Team ein Streudiagramm erstellt, das wie Abbildung 9.4 aussieht. In diesem Beispiel ist es ziemlich offensichtlich, dass die Punktstapel, die Sie sehen, an den Sprint Grenzen liegen - wahrscheinlich, wenn ein wahnsinniger Ansturm auf die Fertigstellung von PBIs vor dem Ende des Sprints herrscht. Aber sehen Sie sich an, wie sich die Daten zwischen diesen Stapeln ausdünnen. Ist das eine gute oder eine schlechte Sache? Soll Scrum überhaupt so funktionieren? Wie auch immer, welche Auswirkungen hat dies auf unsere Vorhersagbarkeit? Wenn Sie der Meinung sind, dass es schlecht ist, was könnten Sie tun, um das zu ändern?



# Interne und externe Variabilität

Der Grund dafür, dass Streudiagramme im Laufe der Zeit wie eine zufällige Ansammlung von Punkten aussehen, liegt in der Variation, die in Ihrem Prozess existiert. Das erste, was man über Schwankungen wissen muss, ist, dass sie immer existieren werden. Unter dem Gesichtspunkt der Vorhersagbarkeit geht es nicht darum, die Schwankungen zu beseitigen, sondern vielmehr darum, die Ursachen dieser Schwankungen zu verstehen, um Ihren Prozess vorhersagbarer zu machen.

Schauen Sie sich zum Beispiel Abbildung 9.5 an:

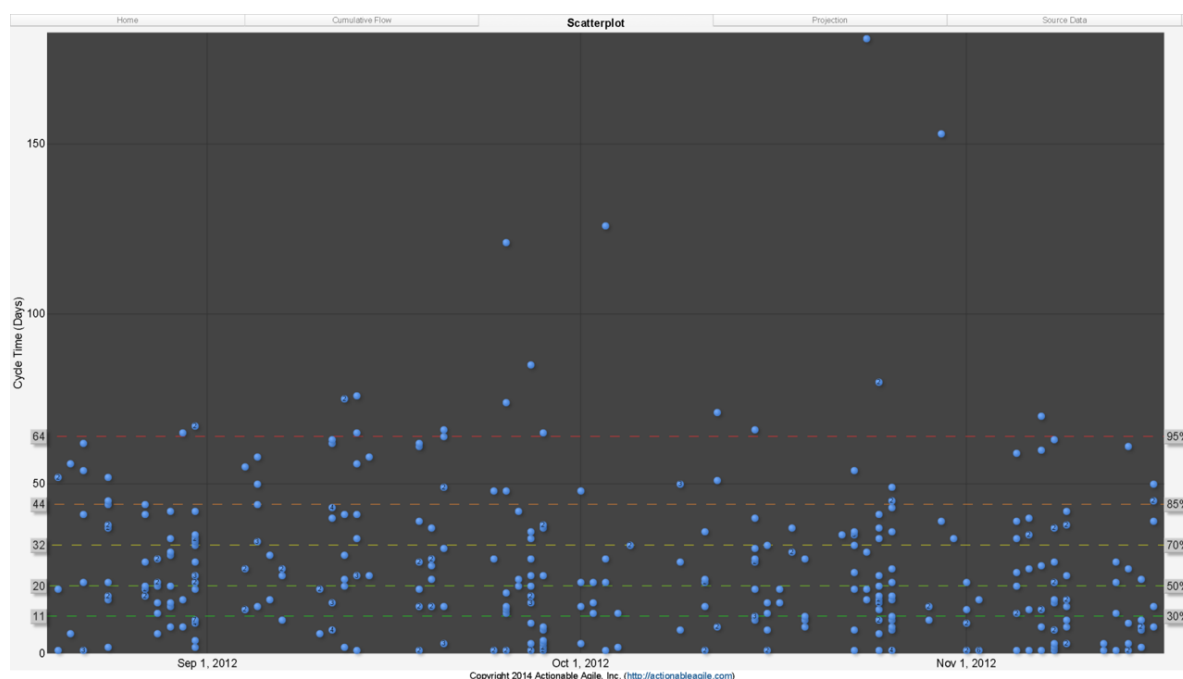


Abbildung 9.5: Lücken in einem Streudiagramm

Auf den ersten Blick sind Sie vielleicht geneigt, diese Punkte oben im Streudiagramm als Ausreißer abzutun. Sie könnten sich fragen, ob es sinnvoll ist, sie einzubeziehen, da es sich eindeutig um Ausreißer handelt. Sie könnten sogar (wenn Sie sich selbst nicht sehr mögen) eine weitere quantitative Analyse durchführen, um zu beweisen, dass diese Punkte statistisch nicht signifikant sind. Und wissen Sie was, wenn Sie diese Behauptungen aufstellen würden, würden wir Ihnen wahrscheinlich nicht allzu heftig widersprechen. Wir würden jedoch sagen, dass es sich bei diesen Punkten zwar um Ausreißer handelt, dass sie aber offensichtlich aufgetreten sind und wahrscheinlich eine eingehendere Untersuchung rechtfertigen. Wir würden auch sagen,

## Kapitel 9 - Die Sprint Retrospektive

dass es, auch wenn sie möglicherweise statistisch unbedeutend sind, einige gute kontextuelle oder qualitative Gründe geben könnte, sie aus einer Analyseperspektive beizubehalten.

Um diesen Punkt zu veranschaulichen, betrachten wir, was uns das Diagramm in Abbildung 9.5 mitteilt. Das 50. Perzentil der Durchlaufzeit liegt bei 20 Tagen und das 85. Perzentil bei 44 Tagen. Sie können aber sehen, dass ein Work Item / eine Arbeit in diesem Diagramm 181 Tage benötigt hat! Fallen Ihnen Gründe ein, die dazu geführt haben könnten, dass dieses bestimmte Work Item so lange gebraucht hat? Vielleicht war das Team bei der Entwicklung von einem externen Anbieter oder von einem anderen internen Entwicklungsteam abhängig. Vielleicht hatte das Team nicht sofort eine Testumgebung zur Verfügung. Vielleicht war der Kunde nicht sofort zur Freigabe verfügbar. Der gemeinsame Nenner all dieser Gründe ist, dass die Fertigstellung dieser Aufgaben aus Gründen, die außerhalb der Kontrolle des Teams lagen, so lange dauerte. Und das ist in der Regel das, was Sie finden werden, wenn Sie den Stapel der Punkte auf einem Streudiagramm nach oben verschieben. In den meisten Fällen sind diese Ausreißer auf Umstände zurückzuführen, die sich der Kontrolle des Teams entziehen.

Das Gegenteil ist in der Regel auch der Fall. Wenn Sie den Stapel nach unten verschieben, haben die Aufgaben, die weniger Zeit in Anspruch genommen haben, in der Regel Gründe, die vollständig unter der Kontrolle des Teams lagen. Denken Sie zum Beispiel an die eben erwähnte Aufgabe, deren Erledigung 181 Tage dauerte. Glauben Sie wirklich, dass diese Aufgabe 181 Tage gedauert hätte, wenn das Team, das daran gearbeitet hat, die volle Kontrolle darüber gehabt hätte? Vielleicht, aber wahrscheinlich nicht. Sehen Sie sich außerdem die Punkte an, die nur knapp die 85%-Linie verfehlt haben. Glauben Sie, dass es Dinge gab, die das Team hätte tun können, um sicherzustellen, dass es nicht zu dieser Überschreitung gekommen wäre? Wahrscheinlich. Swarmen oder das Work Item / die Arbeit zerteilen sind zwei Ideen, die mir sofort in den Sinn kommen.

Wir hoffen, dass Sie ein Gefühl für die Art der Variabilitätsanalyse bekommen, die wir mit diesen Streudiagrammen von Ihnen verlangen. Sind alle Ausreißer auf externe Ursachen zurückzuführen? Sicherlich nicht. Vielleicht hat das Team ein Element in den Prozess aufgenommen, das am Ende zu groß war. Vielleicht hat das Team ein Element ignoriert, nachdem es herausgezogen worden war. Gibt es auch externe Probleme, die sich in den kürzeren Durchlaufzeiten verstecken? Mit ziemlicher Sicherheit. Aber zumindest haben wir Ihnen gezeigt, wie Sie ein Streudiagramm mit Perzentil-Linien verwenden können, um die Diskussion darüber zu beginnen, wie diese Probleme angegangen werden können. Und je mehr Sie sich an die Annahmen des Little'schen Gesetzes halten, desto sicherer können wir

sein, dass die Punkte "oben im Stapel" auf Ausreißer und die Punkte "unten im Stapel" auf die Teamrichtlinien zurückzuführen sind.

# Schlussfolgerung

Daten ohne Maßnahmen sind bedeutungslos. Der Zweck Ihrer Sprint Retrospektive besteht darin, Ihren Gesamtprozess im Laufe der Zeit zu verbessern. Metriken sind nur ein, wenn auch wichtiger Teil davon.

Denken Sie bei Ihrer nächsten Sprint Retrospektive daran, dass Ihre Richtlinien Ihre Daten formen und Ihre Daten Ihre Richtlinien formen. Ihre Daten erzählen Ihnen die Geschichte Ihres Prozesses. Hören Sie zu?

# Quellen

1. Vacanti, Daniel S. "Actionable Agile Metrics for Predictability". ActionableAgile Press. 2014.

# Kapitel 10 - Werkzeuge

Jetzt geht es los, das Kapitel, das uns im Internet zum Gespött machen wird. Werkzeuge sind ein unglaublich umstrittenes Thema. Wir werden Ihnen aber keine expliziten Empfehlungen geben, was Sie verwenden sollten. Stattdessen geben wir Ihnen einige Tipps, worauf Sie bei einem Tool im Hinblick auf seine Funktionen achten sollten. Diese sind praktischerweise auch nach Priorität geordnet.

Wir werden auch einige Warnsignale nennen. Wenn Sie sich in einer Situation befinden, in der Sie gezwungen sind, eines dieser Tools zu verwenden, werden Sie wissen, dass Sie diese umgehen müssen. Oder besser noch, vermeiden Sie sie ganz.

**Anmerkung:** Ich weiß zwar, dass die meisten Teams dem verpflichtet sind, was ihre Organisation auswählt, aber meiner Meinung nach bedeutet die Zeile im Scrum Guide<sup>1</sup> im Abschnitt über Scrum Teams, die lautet: "Sie sind von der Organisation strukturiert und ermächtigt, ihre eigene Arbeit zu verwalten", dass Scrum Teams frei sein sollten, das Tool auszuwählen, von dem sie glauben, dass es ihnen am besten dient. Selbst wenn dies bedeutet, dass jedes Team sein eigenes Workflow-/Backlog-Management-Tool hat. Die meisten Argumente gegen diese Ansicht / Auslegung kommen entweder von einem "Management muss die Dinge kombiniert sehen" oder einer Art von Kostensenkungshintergrund. Das ist völliger Blödsinn. Menschen außerhalb des Teams sollten auf die Ergebnisse schauen. Nicht auf den Output. Und nichts bremst eine Organisation schneller aus als die Standardisierung der falschen Dinge. -Will

**Weitere Anmerkung:** Gelegentlich wird die Frage der Verfolgung von Abhängigkeiten zwischen den Teams mit Hilfe eines Tools aufgeworfen. Obwohl man Abhängigkeiten idealerweise wie jedes andere Hindernis behandeln und versuchen sollte, sie zu beseitigen, bin ich mir der Tatsache / dem Fakt bewusst, dass sie für viele Teams immer noch eine Tatsache sind. Wenn man sie unbedingt verfolgen muss, ist das eine Frage der Daten, nicht der Werkzeuge. Selbst in einem solchen Fall, in dem Sie die Daten teamübergreifend verfolgen wollen, würde ich dazu raten, die Daten direkt über das Backend zu kombinieren, anstatt dem Team die Wahl des Tools abzunehmen. Bedeutet dies, dass Sie Tools wählen sollten, die sich an gemeinsame Standards mit APIs für den Datenzugriff halten? Offensichtlich. - Will

# Worauf Sie bei einem Tool achten sollten

So geht's! Dies ist Ihre Checkliste für die Auswahl des richtigen Tools. Verwenden Sie sie entweder selbst oder schicken Sie sie an den Einkauf. In letzterem Fall sollten Sie den Begriff "nicht verhandelbare Anforderung" verwenden. Denken Sie daran, dass Pragmatismus der Ort ist, an dem gute Dinge sterben. Wir werden unsere Tipps in der Reihenfolge ihrer Wichtigkeit auflisten.

## Daten

Zuallererst müssen Ihre Werkzeuge gute Daten erzeugen. Die meisten anderen Dinge können Sie mit Disziplin erledigen, aber wenn Sie die Zeitangaben zu den einzelnen Elementen manuell erfassen müssen, wird ein großer Teil Ihres Tages verloren gehen. Wir meinen es hier ernst: Will hat einmal ein Team gecoacht, das mit einem Ticketingsystem arbeiten musste, das nur den Zeitpunkt der Erstellung (nicht des Pulls / des Ziehens), des Schließens und der letzten Verschiebung eines Vorgangs erfasste, warum auch immer. Das bedeutete, dass jemand aus dem Team einmal am Tag manuell protokollieren musste, was sich in welchem Schritt des Workflows befand, um die Durchlaufzeit zu ermitteln. Das dauerte etwa eine Stunde. Jeden Tag. Also ja, suchen Sie sich ein Tool, das die Zeit genau festhält. Und was meinen wir damit?

Es sollte zumindest:

- Die ID des Work Items / der Arbeit protokollieren.
- Das Datum (und möglicherweise die Uhrzeit) protokollieren, an dem jedes Element in einen Schritt Ihres Arbeitsablaufs eintritt.
- Korrektes Verwerfen von Zeiten, wenn Sie etwas zurückschieben müssen (weil es nicht korrekt fortgesetzt wurde).
- Exportieren Sie diese Daten in einem einfach zu verwendenden Format (CSV, Tabellenkalkulation oder API), damit Sie sie überprüfen (und bei Bedarf in andere Tools importieren) können.

Es ist großartig, wenn es auch:

## Kapitel 10 - Werkzeuge

- Den Namen des Work Items / der Arbeit protokolliert (so dass Sie keine Querverweise vornehmen müssen).
- Die Zeit protokolliert, in der das Work Item / die Arbeit blockiert war.
- Den Work Item Type / Arbeitstyp als Attribut protokolliert (so dass Sie filtern können).

Hier ein Beispiel für die Art von Daten, die Sie von einem Work Item / einer Arbeit benötigen, unter Verwendung der Notation Jahr-Monat-Tag-Stunde-Minute und der Sperrzeit in Minuten:

Example output									
Item ID	Item Name	Product Backlog	Sprint Backlog	Design	Creation	Tweaking	Done	Type	Time Blocked
1	Wobbles	2022 0701 1437	2022 0708 0932	2022 0708 1045	2022 0708 1557	2022 0710 1102	2022 0710 1649	Feature	137

Abbildung 10.1: Beispiel für ein Datenformat

Dies ist bereits ein umfangreicher Datensatz. Er enthält alles, was wir brauchen, um die zuvor erwähnten Diagramme zu erstellen. Wir können Durchlaufzeit-Streudiagramme erstellen, indem wir die Daten "Sprint Backlog" und "Done" verwenden, denn wenn es kein Datum in "Done" gibt, ist es nicht "Done". Wir können ein Diagramm zur Fälligkeit von Arbeitsaufgaben erstellen, indem wir die Daten aus jedem Schritt des Workflows verwenden. Wir können den Durchsatz messen, indem wir uns die Anzahl der Aufgaben ansehen, die innerhalb eines Datumsbereichs erledigt wurden, und wir können den historischen WIP messen, indem wir die Aufgaben in jedem Schritt innerhalb eines Datumsbereichs zählen. Und wir können dies nach Arbeitsart / Typ des Work Items tun. Als Sahnehäubchen können wir uns sogar die Zeit ansehen, die eine Arbeit aus irgendeinem Grund blockiert war. Dies ist die wichtigste Funktion für Sie. Wenn es das nicht kann, sollten Sie sich ein anderes Tool suchen.

## Einfaches Workflow-Design

Produkte und die dazugehörigen Arbeitsabläufe werden sich im Laufe der Zeit aus allen Gründen ändern, die die Arbeit komplex machen. Manchmal geschieht dies im Voraus, meistens jedoch aufgrund plötzlicher Erkenntnisse und Einflüsse. Das bedeutet, dass Ihr Tool flexibel genug sein sollte, um eine schnelle und einfache (Neu-)Gestaltung von Arbeitsabläufen

zu unterstützen. Dies beginnt natürlich damit, dass Ihr Team in der Lage ist, seinen eigenen Arbeitsablauf in dem von ihm verwendeten Tool zu ändern.

Anmerkung: Ehrlich gesagt ist die Tatsache, dass es bis heute Teams gibt, die eine Art "Admin" bitten müssen, ihren Workflow anzupassen, oder gezwungen sind, einen von jemand anderem entworfenen Workflow zu verwenden, einfach nur ärgerlich. Diese respektlose Infantilisierung von Teams macht mich wirklich wütend. -Will

Denken Sie daran, dass es bei der Gestaltung Ihres Arbeitsablaufs nicht nur um die Erstellung von Spalten geht. Es bedeutet auch, dass Sie WIP-Limits, Ausstiegskriterien / Kriterien zum Verlassen der Spalten bzw. Schritte, Work Item Types / Arbeitstypen, Pull-Richtlinien, die Anzeige blockierter Arbeiten usw. definieren und visualisieren müssen. Spalten allein sind kein guter Anfang. Wenn Sie jedoch in der Klemme stecken, können Sie den Rest mit Disziplin oder der Erfassung von Dingen an anderer Stelle abdecken. Stellen Sie sicher, dass Ihre Daten auch mit dem neuen Workflow zugänglich bleiben. Ihr Tool sollte dies problemlos bewältigen können. Wenn Sie beschließen, einen Schritt in Ihrem Arbeitsablauf zu streichen (z. B. den Schritt "Optimieren" in unserem früheren Beispiel), sollte Ihr Tool den Schritt "Optimieren" nicht mehr exportieren, auch nicht für Elemente, die dort Daten enthalten. Wenn Sie diesen Schritt jedoch wieder einführen wollen, sollen auch keine Daten verloren gehen.

Was wir damit sagen wollen, ist, dass Ihr Werkzeug die Entwicklung Ihres Workflows / Arbeitsablaufs nicht einschränken oder verzögern sollte.

## WIP-Ansichten und Warnungen

Ok, Sie haben Ihre Daten und Ihren Arbeitsablauf geordnet. Als Nächstes steht etwas auf der Liste, das man unbedingt haben sollte: WIP-Ansichten und -Warnungen. Der Grund dafür, dass wir dies als "nice to have" bezeichnen, ist, dass Sie WIP in erster Linie durch Disziplin verwalten können (und sollten). Der andere Grund ist, dass Ihr WIP, wenn er anfängt, zweistellige Spaltenzahlen zu erreichen, wahrscheinlich sowieso schon zu hoch ist.

Aber Spaß beiseite, der Grund, warum wir darüber sprechen, ist, dass es auf dem Markt ein weit verbreitetes Missverständnis gibt, dass es in einem Arbeitsflusssystem (oder Kanban-System) WIP-Grenzen nur auf einer spaltenweisen Basis gibt. Im Kanban Guide<sup>2</sup> heißt es dagegen ganz klar: "Ein WIP-Limit kann sich auf Arbeitsaufgaben in einer einzelnen Spalte, mehreren gruppierten Spalten/Ebenen/Bereichen oder eine ganzen Board beziehen (ist aber

nicht darauf beschränkt)." Das bedeutet, dass Ihr Tooling es Ihnen ermöglichen sollte, WIP-Limits auf jede beliebige Art (oder Kombination von Arten) zu definieren. Beispiele hierfür sind:

- Pro Spalte
- Pro Arbeitsart / Typ des Work Items
- Pro Zeile / Swim Lane
- Pro Person
- Pro Freigabe
- Insgesamt
- (Eine beliebige Kombination aus) allen oben genannten Punkten

Darüber hinaus möchten Sie, dass Ihr Tool Sie in irgendeiner Weise warnt, wenn Sie die Vorgaben überschreiten, oder dies sogar verhindert. Viele Werkzeuge lassen Sie fröhlich und stillschweigend weiterarbeiten, wobei die einzige Rückmeldung ein winziges "(!)23/7" am oberen Rand der Spalte ist. Das ist nicht gerade eine Warnung, oder? Wenn es Ihrem Team mit dem Arbeitsfluss ernst ist, sollten Sie das Überschreiten von Grenzwerten unterbinden oder für jedes Work Item / Arbeit, das Sie über das Limit hinaus ziehen wollen, zusätzlichen Aufwand kalkulieren. Vielleicht sogar mit einer Warnung in einem internen Slack oder MS Teams Kanal.

Auch hier gilt, dass dies kein Ersatz für Teamdisziplin ist. Aber die Unterstützung durch ein Tool kann die Aufrechterhaltung dieser Disziplin sehr viel einfacher machen.

## **Ausstiegskriterien / Kriterien zum Verlassen der Spalten bzw. Schritte**

An diesem Punkt sind wir aus Sicht der Arbeitsflussmetriken auf einem absoluten Luxusniveau angelangt (obwohl es sich immer noch um grundlegende Kanban-Hygiene handelt). Aber wenn wir einen Workflow visualisieren, sollten wir mehr tun, als nur die Schritte aufzulisten: Wir sollten visualisieren, wann jeder Schritt abgeschlossen ist. Mit anderen Worten: Was sind die Ausstiegskriterien für jede Spalte? Auch hier könnte ein Team diese Kriterien an anderer Stelle dokumentieren, oder sogar in einer allgemeinen Definition of Done, die sich nicht auf eine bestimmte Spalte bezieht. Im Idealfall sollte Ihre Definition of Done jedoch über den gesamten Arbeitsablauf verteilt und in einem leicht zu erfassenden Format sichtbar sein. Nicht



viele Tools unterstützen dies, aber diejenigen, die dies tun, machen das Leben der Entwickler sehr viel einfacher.

# Rote Flaggen

Nachdem wir uns die Funktionen angesehen haben, nach denen Sie suchen, wollen wir uns nun einige Warnhinweise ansehen, nach denen Sie Ausschau halten sollten. Diese reichen von sofortiger Ablehnung bis hin zu "Sie können das Tool verwenden, aber vermeiden Sie diese Funktion". Uns ist wiederum klar, dass nicht jedes Team den Luxus hat, sich sein eigenes Tool auszusuchen *grummel - Will*, aber selbst in diesem Fall sollte diese Liste für Gesprächsstoff mit den Verantwortlichen sorgen.

## Obligatorische Kostenvoranschläge

Right Sizing bedeutet, dass Sie keine Schätzungen in das Tool eingeben müssen, weil Sie keine herkömmlichen Schätzungen vornehmen. Beachten Sie, dass Schätzungen hier alles von Story Points über T-Shirt Sizing bis hin zu tatsächlichen Stunden bedeuten können. Wir verwenden nichts davon. Sie könnten argumentieren, dass ein "Ja" oder "Nein" zu einem Element, das in den SLE passt, eine Schätzung ist, aber das ändert nichts an der Tatsache, dass Ihr Tool dies nicht verfolgen muss. Dieses Feld wird also nicht benötigt. "Aber die Autoren", hören wir Sie sagen, "warum können wir nicht einfach irgendetwas in das Feld für den Kostenvoranschlag eintragen?". Weil ein Tool mit obligatorischen Schätzungen versuchen wird, "hilfreich" zu sein, und das wird bestenfalls als Ablenkung während Ihrer Sprints dienen und Ihnen schlimmstenfalls Schrottdaten liefern, die Ihr Team verankern oder Ihre Stakeholder auf unangenehme Weise beeinflussen werden. Einfach ausgedrückt: Wenn Sie Arbeitsflussmetriken verwenden, müssen Sie keine Schätzungen in Ihr Tooling eingeben. Ein Pflichtfeld für Schätzungen ist ein deutliches Warnzeichen dafür, dass das Tool keine guten Arbeitsflussmetriken unterstützt. Es sei denn, ein Administrator (Admin) hat dies eingerichtet. In diesem Fall sollte Ihr Scrum Master Ihrem Team Admin-Rechte für seine Tools verschaffen. Oder führen Sie ein gutes Gespräch mit dem Administrator.

# Verpflichtende Vereinbarungen

"Das gesamte Scrum Team ist dafür verantwortlich, in jedem Sprint ein wertvolles, nützliches Inkrement zu erstellen"<sup>1</sup>. Klingt ziemlich klar, oder? Oder? Warum sollte ein Tool Ihnen dann erlauben, Backlog-Elemente bestimmten Personen zuzuweisen? Das ist schon ziemlich furchtbar. Wissen Sie, was noch schlimmer wäre? Das Tool zwingt Sie dazu, bestimmten Personen Arbeit zuzuweisen. Dadurch wird sichergestellt, dass der Rest des Teams versteht, dass er sich nicht um diesen Punkt kümmern muss. Okay, genug mit dem Sarkasmus. Sie wollen, dass Ihr Tool die Teamarbeit fördert und dass sich jeder um das Sprint Backlog kümmert und wenn nötig swarmed, um Dinge voranzutreiben, die sonst gegen die SLE verstoßen würden. Individuelle Zuweisungen signalisieren das genaue Gegenteil davon. Wenn also ein Tool dies vorschreibt, schalten Sie die Einstellung aus oder verwenden Sie das Tool nicht. Wenn Sie das Tool verwenden müssen, erstellen Sie ein Dummy-Teammitglied mit dem Namen "Scrum Team" und weisen Sie ihm die gesamte Arbeit zu.

# Durchschnittswertbildung in Diagrammen

An diesem Punkt der Lektüre dieses Buches werden Sie bereits eine ganze Reihe von Diagrammen gesehen haben. Diagramme sind schön, wenn sie die richtigen Dinge visualisieren. Diagramme, die Durchschnittswerte zeigen, tun das nicht. Das Verständnis probabilistischer Prognosen ist für das Team, das täglich damit arbeitet, schon schwer genug, da brauchen Sie keine zusätzliche Ablenkung durch sinnlose Durchschnittswerte. Schlimmer noch: Ihre Stakeholder könnten das sehen und falsche Erwartungen entwickeln. Alles in allem ist es besser, ein Tool zu haben, das keine Diagramme erstellt (da Sie dafür andere Tools verwenden können, einschließlich Excel), als ein Tool, das schlechte Diagramme anzeigt.

# Fazit

Werkzeuge können eine große Hilfe oder ein großes Hindernis sein. Wir hoffen, dass Sie in diesem Kapitel einige gute Kriterien für die Auswahl oder Bewertung eines Tools gefunden haben. Ein letzter Tipp, den wir Ihnen geben möchten, ist, einfach anzufangen. Verlassen Sie sich nicht auf ein Tool, um mangelndes Wissen oder mangelnde Disziplin im Team zu

kompensieren. Erreichen Sie einen Punkt, an dem Ihr Team einen reibungslosen Ablauf erreicht hat, und finden Sie ein Tool, das diesen Ablauf unterstützt.

# Quellen

1. Schwaber, Ken and Sutherland, Jeff "The Scrum Guide" <https://scrumguides.org>, 2020.
2. Coleman, John and Vacanti, Daniel S. "The Kanban Guide" <https://kanbanguides.org>, 2020.

# Kapitel 11 - Erste Schritte

Wenn Sie dieses Buch lesen, haben Sie wahrscheinlich schon eine Vorstellung davon, wie die Verwendung von Arbeitsflussmetriken in Ihrem Kontext aussehen könnte. Der schwierigste Teil ist oft, einfach anzufangen. Hier beschreiben wir einen Ansatz, der Ihnen den Einstieg erleichtert. Es gibt mehrere Möglichkeiten, mit Arbeitsflussmetriken zu beginnen, von denen die meisten im Wesentlichen auf die Implementierung einer (minimalen) Kanban Strategie hinauslaufen. Die im Folgenden beschriebene Strategie hat sich aus unserer Sicht als die erfolgreichste erwiesen. Elemente dieses Ansatzes wurden bereits in früheren Kapiteln beschrieben (insbesondere in Kapitel 4). Wir haben die Kapitel, in denen das Konzept vertieft wird, neben jedem Schritt angegeben. Bitte lesen Sie in diesen Kapiteln nach, wenn Sie sich eingehender mit diesen Schritten befassen möchten.

## Erste Schritte

### **Schritt 0: Überwachen Sie das Alter des Work Items in Ihrem Daily Scrum**

Die wichtigste Erkenntnis aus diesem Buch ist, dass Sie, wenn Sie Arbeitsflussmetriken in Ihren Scrum Teams einführen wollen, mit dem Work Item Age beginnen sollten. Und der Zeitpunkt, an dem Sie sich das Alter ansehen sollten, ist während Ihres Daily Scrum. Wenn Sie das Alter richtig hinbekommen, wird sich alles andere von selbst ergeben. Es ist jedoch etwas unaufrichtig zu sagen, man solle mit dem Alter beginnen, denn wenn Sie eine traditionellere Scrum Implementierung durchgeführt haben, ist die Wahrscheinlichkeit groß, dass Sie noch nicht einmal die Grundlagen für die Überwachung der Alterung geschaffen haben. Im Folgenden finden Sie daher einige Schritte, die Sie möglicherweise zuerst durchführen müssen, um die Alterung und andere Arbeitsflussmetriken in Ihren Scrum-basierten Prozess einzubauen.

### **Definieren Sie Start- und Endpunkte**

Wir müssen festlegen, wann wir PBIs als begonnen und wann wir sie als beendet betrachten. Diese bilden die Grenzen, innerhalb derer wir alle Arbeitsflussmetriken messen und verwalten

werden. Diese Grenzen werden und sollten sich im Laufe der Zeit ändern, aber wir müssen sie immer im Auge behalten, damit wir das System effektiv beobachten und verbessern können.

### **Bestimmen Sie alle Aktivitäten, die zur Wertschöpfung beitragen**

Während sich PBI vom Startpunkt eines Systems zum Endpunkt eines Systems bewegt, führen wir Aktivitäten durch, um es in einen lieferbaren Wert zu verwandeln. Wir müssen bestimmen, was diese Aktivitäten sind. Bei einer Software könnten dies sein: Problemverständnis, technischer Entwurf, Codeerstellung, Code-Review, Unit-Tests, Integrationstests, Erstellung und Bereitstellung, Kundenvalidierung. In Systemen gibt es eine unterschiedliche Anzahl von Aktivitäten, die dazu beitragen, dass aus einem Work Item / einer Arbeit ein Wertbeitrag wird. Wenn wir sie auflisten, erhalten wir eine Vorstellung vom Ablauf des Systems. Die Alterung kommt erst richtig zur Geltung, wenn wir den Kontext des Wertstroms verstehen, durch den sich ein PBI bewegt.

### **Auswahl einer anfänglichen Service-Level-Erwartung**

Als Nächstes müssen wir mit Hilfe eines SLE ein gemeinsames Verständnis für die Frage "Wie lange brauchen wir, um die Arbeit zu erledigen?" schaffen. Dies kann durch eine Analyse der Durchlaufzeitverteilung der bereits erledigten Aufgaben geschehen. Auf der Grundlage dieser Verteilung können wir eine SLE auswählen, die die Zeit angibt, die die Mehrheit (70 %, 85 %, 95 % usw.) für die Erledigung der Aufgaben benötigt. Diese SLE wird zur Messlatte für die im System aktiven Aufgaben. Beachten Sie, dass Sie am Anfang wahrscheinlich nicht über die Daten verfügen, die Sie zur Berechnung Ihrer SLE benötigen. Das ist in Ordnung. Wir empfehlen, dass Sie mangels Daten einfach eine vernünftige SLE für den Anfang wählen. Wenn Sie z. B. einen 14-Tage-Sprint absolvieren, könnte eine vernünftige SLE "7 Tage oder weniger bei 85 %" lauten. Es könnte aber auch ein noch kürzerer Wert sein. Der Punkt ist, dass Sie einfach etwas auswählen sollten, damit Sie einen Maßstab haben, mit dem Sie das Alter während Ihres Daily Scrum vergleichen können. Im Laufe der Zeit, wenn Sie eine gute Menge an repräsentativen Daten haben, sollten Sie die SLE nach oben oder unten anpassen, je nachdem, was Ihre Daten Ihnen sagen. Der Zeitpunkt für dieses Gespräch? Sie haben es erraten: die Sprint Retrospektive.

# Schlussfolgerung

Es gibt viele Möglichkeiten, mit Arbeitsflussmetriken in Scrum zu beginnen. Dieses Kapitel bietet einen Entwurf des Ansatzes, der sich unserer Erfahrung nach bewährt hat. Der größte Teil der Arbeit für den Einstieg besteht darin, Ihre aktuellen Aktivitäten zu verstehen und Ihre Arbeitsabläufe und Richtlinien zu definieren. Sobald wir ein gemeinsames Verständnis davon haben, können wir damit beginnen, die Gründe zu beobachten, warum Elemente in unserem System altern, und die entsprechenden Anpassungen vornehmen.

Von diesem Punkt an liegt es an Ihnen. Nutzen Sie Ihre Daten, überprüfen Sie Ihre Arbeitsabläufe und passen Sie sie bei Bedarf an, um großartige Dinge zu schaffen. Viel Erfolg!

# Bibliography

Bertsimas, D., D. Nakazato. The distributional Little's Law and its applications. *Operations Research*. 43(2) 298–310, 1995.

Brooks, F. (1995). *Mythical Man-Month, The: Essays on Software Engineering, Anniversary Edition* (Anniversary ed.). Addison-Wesley Professional.

Brumelle, S. On the relation between customer and time averages in queues. *J. Appl. Probab.* 8 508–520, 1971.

Coleman, John and Vacanti, Daniel S. "The Kanban Guide" <https://kanbanguides.org>, 2020.

Constable, G. (2021, April 8). The Truth Curve and the Build Curve | [giffconstable.com](https://giffconstable.com). Giffconstable.Com. <https://giffconstable.com/2021/04/the-truth-curve-and-the-build-curve/>

Deming, W. Edwards. *The New Economics*. 2nd Ed. The MIT Press, 1994.

Deming, W. Edwards. *Out of the Crisis*. The MIT Press, 2000.

Glynn, P. W., W. Whitt. Extensions of the queuing relations  $L = \lambda W$  and  $H = \lambda G$ .

*Operations Research*. 37(4) 634–644, 1989.

Goldratt, Eliyahu M., and Jeff Cox. *The Goal*.

2nd Rev. Ed. North River Press, 1992.

Gothelf, J., & Seiden, J. (2021). *Lean UX: Designing Great Products with Agile Teams* (3rd ed.). O'Reilly Media.

Heyman, D. P., S. Stidham Jr. The relation between customer and time averages in queues. *Oper. Res.* 28(4) 983–994, 1980.

Hopp, Wallace J., and Mark L. Spearman. *Factory Physics*. Irwin/McGraw-Hill, 2007.

## Bibliography

Little, J. D. C. A proof for the queuing formula:  $L = \lambda W$ . *Operations Research*. 9(3) 383–387, 1961.

Little, J. D. C., and S. C. Graves. “Little’s Law.” D. Chhajed, T. J. Lowe, eds. *Building Intuition: Insights from Basic Operations Management Models and Principles*. Springer Science + Business Media LLC, New York, 2008.

Ripley, Ryan and Miller, Todd. *Fixing Your Scrum*. The Pragmatic Programmers LLC, 2020.

Reid, Steve and Singh, Prateek and Vacanti, Daniel "Ultimate Kanban: Scaling Agile without Frameworks at Ultimate Software" <https://www.infoq.com/articles/kanban-scaling-agile-ultimate/>, 2016.

Reinertsen, Donald G. *Managing the Design Factory*. Free Press, 1997.

Reinertsen, Donald G. *The Principles of Product Development Flow*. Celeritas Publishing, 2009.

Savage, Sam L. *The Flaw of Averages*. John Wiley & Sons, Inc., 2009.

Schwaber, Ken and Sutherland, Jeff "The Scrum Guide" <https://scrumguides.org>, 2020.

Shewhart, W. A. *Economic Control of Quality of Manufactured Product*, 1931.

Shewhart, W. A. *Statistical Method from the Viewpoint of Quality Control*, 1939.

Stidham, S., Jr.  $L = \lambda W$ : A discounted analogue and a new proof. *Operations Research*. 20(6) 1115–1126, 1972.

Stidham, S., Jr. A last word on  $L = \lambda W$ . *Operations Research*. 22(2) 417–421, 1974.

Vacanti, Daniel S. *Actionable Agile Metrics for Predictability*. ActionableAgile Press. 2014.

Vacanti, Daniel S. *When Will It Be Done?* ActionableAgile Press, 2017.



## Bibliography

Vacanti, Daniel S. and Bennet Vallet. “Actionable Metrics at Siemens Health Services”. <https://www.agilealliance.org/resources/experience-reports/actionable-metrics-siemens-health-services/>. 1 Aug 2014.

Vallet, Bennet. “Kanban at Scale: A Siemens Success Story.” <https://www.infoq.com/articles/kanban-siemens-health-services/>. 28 Feb 2014.

Vega, Frank. “Are You Just an Average CFD User?” <https://www.vissinc.com/2014/02/21/are-you-just-an-average-cfd-user/>. 21 Feb 2014.

Vega, Frank. “The Basics of Reading Cumulative Flow Diagrams”. <https://www.vissinc.com/2011/09/29/basics-of-reading-cumulative-flow-diagrams/>. 29 Sep 2011.

Wheeler, Donald J., and David S. Chambers. *Understanding Statistical Process Control*. 2nd Ed. SPC Press, 1992.

Wikipedia “Monte Carlo method.” <https://en.wikipedia.org/wiki/MonteCarlomethod#>: