

# Framework für Agiles Projektcontrolling



## How to get Scrum under control...



Scrum ist eine dynamische und flexible, menschenorientierte Entwicklungs- und Projektmanagement Methode. Controlling hingegen basiert auf langfristiger Planung. Wie kann man also ein Projekt steuern, wenn sich Anforderungen ständig ändern? Der Irrtum, dass Controlling und Scrum sich nicht vereinen lassen, ist grundlegend falsch. Dieser Guide zeigt euch den Scrum Controlling Prozess, welche Parameter gemessen werden sollten und wie diese festgehalten werden können.

Dieses Framework basiert auf den Erkenntnissen aktueller Literatur in Kombination mit Erfahrungswerten aus der Industrie, um euch beim Controlling von agilen Projekten, wie Scrum, zu unterstützen und gängige Methoden aufzuzeigen.

---

### Inhalt

---

Einführung	2
Controlling Prozess	5
Key Performance Indicators	8
Iteration Scorecard	14
Anmerkungen	18

---

# Grundlage für diesen Guide

Dieser Guide ist das Resultat der Untersuchung in einer Masterarbeit, die in Kooperation mit blu Professionals über das Thema „Agile Project Controlling“ geschrieben wurde. Er basiert auf den Erkenntnissen aktueller Literatur und wird unterstützt, sowie ergänzt durch Experteninterviews. Es wurden 23 Interviews mit Experten aus der Industrie aus verschiedenen Branchen durchgeführt. Die ausgewählten Interviewpartner haben unterschiedliche Positionen in ihren Unternehmen und reichen von Projektleitern, -controllern, und -managern bis hin zu Scrum Mastern, Product Ownern und Agile Coaches. Durch ihre praktische Projekterfahrung und theoretisches Wissen konnten sie die Ergebnisse der Literaturrecherche sinnvoll ergänzen und auf praktische Anwendung prüfen.

Für die Interviews wurden Unternehmen folgender Branchen befragt:

- (Projekt-) Management und IT-Beratung
- Automobil
- Versicherungen und Asset Management
- Haushaltsgeräte
- Software-, IT- und Webentwicklung
- Baufinanzierung

Der Guide geht von einem Grundverständnis von Controlling und Scrum aus und beabsichtigt diese zwei Themen hier zu verbinden. Hintergrundwissen und mehr Informationen über die Interviews und deren Auswertung können der Masterarbeit „Agile Project Controlling – Development of a Framework to Enable Controlling in Scrum“ von Franziska Prost entnommen werden. Diese liegt in der blu Bibliothek und auf dem Laufwerk.

Zusätzlich sollte beachtet werden, dass sich das Controlling, wie es hier untersucht wurde, nicht nur auf das Kostencontrolling bezieht, sondern Projekte in ihrer Gesamtheit steuert. Daher ist die Projektbeauftragung nicht Teil der Untersuchung. Es geht vielmehr darum, wie man den Projektfortschritt und die Entwicklung steuern kann, sodass man die Projektziele einhält.

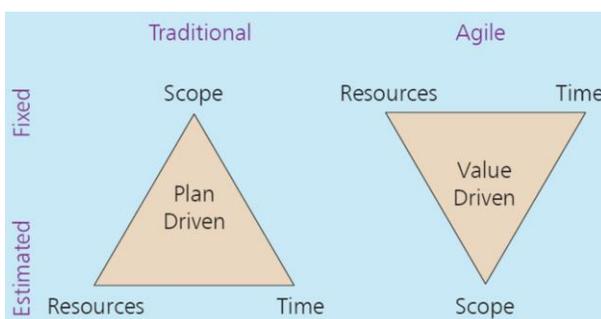
# Wo liegt eigentlich das Problem?



Um den agilen Controlling Prozess zu verstehen, müssen wir erst klären, wo das eigentliche Problem im Vereinen von Controlling und Scrum liegt. In Bezug auf Projekte hat Controlling die Aufgabe, diese zu steuern und zu überwachen. Dies beinhaltet das Erstellen eines Projektplans für die gesamte Projektdauer, das Messen von Abweichungen dieses Plans und das Vorschlagen von Korrekturmaßnahmen, um die Projektziele einzuhalten. Jedes Projekt hat vordefinierte Ziele, die von drei Komponenten umschlossen werden. Diese sind Zeit, Kosten und Projektumfang.

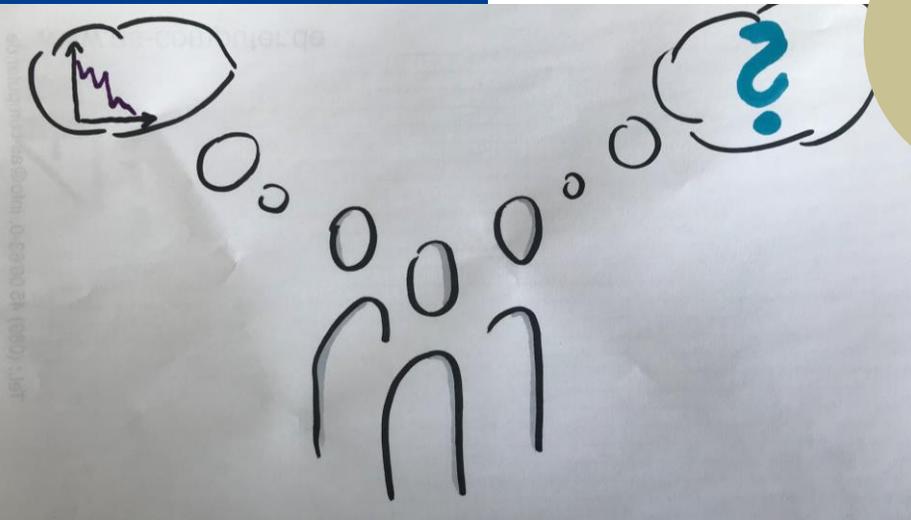
Scrum ist ein iteratives Modell zur Entwicklung von (Software) Produkten und basiert auf häufigen Feedbackschleifen. Zu Beginn gibt es keinen vordefinierten Projektumfang, da sich die Anforderungen erst im Laufe der Zeit entwickeln und sich aufgrund des regelmäßigen Kundenfeedbacks ändern können. Dies hat die Konsequenz, dass kein langfristiger Plan entwickelt werden kann. Der Sinn von Scrum ist, dass eben genau dadurch bessere und schnellere Ergebnisse erzielt werden können, da neue Kundenwünsche direkt umgesetzt werden. Die untere Graphik stellt diesen Zusammenhang dar. Im Vergleich zu traditionellen Projekten ist der Projektumfang nur geschätzt und das Projektmanagement werteorientiert, anstatt plangesteuert. Es bestehen fixe Ressourcen und ein Zeitplan, während in traditionellen Projekten der Umfang festgelegt ist und Ressourcen und Zeit geschätzt sind.

Das Grundproblem liegt in der unterschiedlichen Erwartungshaltung von Kunde und Dienstleister. Der Kunde stellt ein bestimmtes Budget zur Verfügung und möchte die Vorteile der agilen Entwicklung nutzen. Gleichzeitig hat er noch keine genaueren Vorstellungen von dem, was eigentlich dabei entstehen soll. Der Dienstleister führt das Projekt nach Kundenwünschen agil durch und hält sich dabei ganz an die Regeln des Scrum Guides. Nun entsteht aber das Problem, dass der Kunde dennoch über den aktuellen Stand des Projektes informiert werden will. Er fordert die gleiche langfristige Plan- und Prognostizierbarkeit, die er in traditionell durchgeführten Projekten erwartet. Für ihn ist dies eine Selbstverständlichkeit, da er es so aus früheren Wasserfallprojekten kennt. Fakt ist, er versteht Scrum in seiner Tiefe nicht. Wie kann man nun Kunden- und Controllerwünsche abdecken, ohne dabei weder die Essenz von Scrum zu verlieren, noch aufs Controlling zu verzichten?



Quelle: Carroll, J. (2017). Earned Value Management. In easy steps limited.

## EINFÜHRUNG



# Controlling agiler Projekte – eine Notwendigkeit!

Diese Antwort ist relativ einfach. Jedes Projekt muss gesteuert werden, auch agile Projekte mit Scrum. Am Ende des Tages steht hinter jedem Projekt ein Kunde. Dieser möchte über den aktuellen Stand seines Projektes und dessen Verwendung von Ressourcen informiert werden. Es muss sichergestellt werden, dass genügend Budget zur Verfügung steht, um das Projekt erfolgreich durchzuführen und dieses Budget muss überwacht werden. Ebenso muss der zeitliche Verlauf bei Projekten beachtet werden, da es immer Stichtage gibt, zu denen etwas fertig sein muss. Ebenso wollen andere Stakeholder über den Fortschritt des Projektes informiert werden. Zudem ist ein funktionierendes Controlling Grundlage für Entscheidungsfindung, wenn es darum geht, das Budget zu erweitern oder das Projekt fortzusetzen. Controlling bietet zusätzlich die Möglichkeit des Benchmarkings mit anderen Projekten.

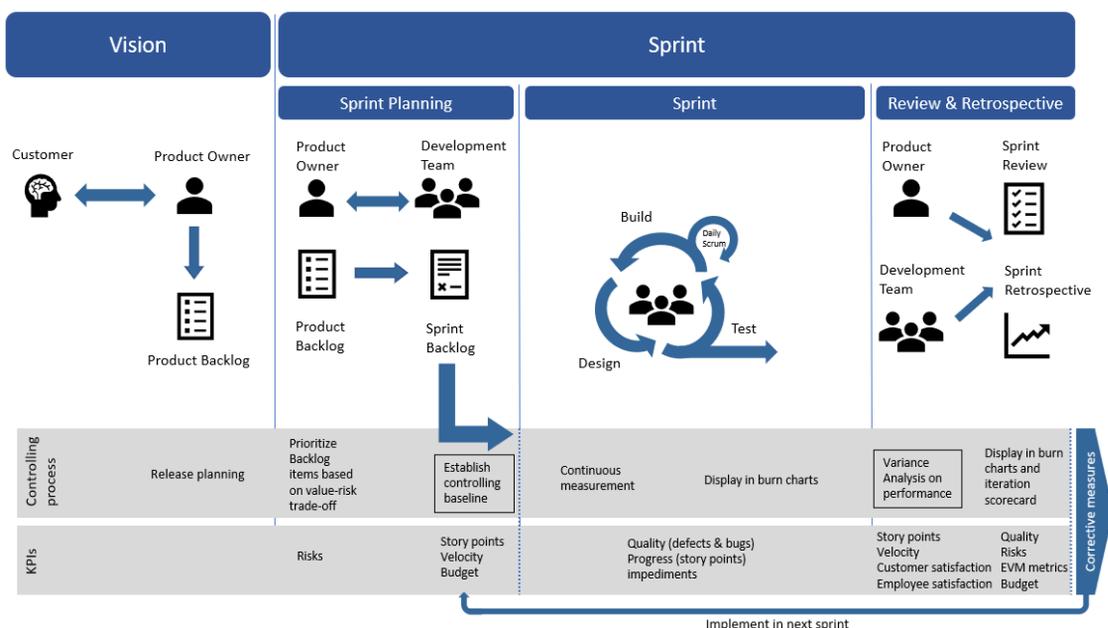
Es ist also klar, dass auch agile Projekte ein Controlling benötigen.

## CONTROLLING PROZESS

# Aber wie sieht das Controlling nun aus?

In diesem Teil des Frameworks wird nun der agile Controlling Prozess vorgestellt. Die untere Grafik stellt diesen Prozess und relevante KPIs dar. Im Vergleich zu traditionellen Projekten bezieht sich der agile Controlling Prozess auf einzelne Sprints. Langfristige Planung lässt sich mit Scrum nur schwer vereinen, weswegen die Planung hier auch auf Sprint Basis durchgeführt wird. Das bedeutet wiederum, dass der Plan in der Sprint Planung in jedem Sprint aufs Neue erstellt und angepasst wird. Diese Planung bildet unseren Controlling Ausgangswert und ist enorm wichtig. Wir planen nun nicht mehr das gesamte Projekt bzw. die gesamte Entwicklung voraus, sondern nähern uns dem Ziel mit kurzfristigen Planwerten. Dennoch gibt es einen groben Releaseplan, der uns sagt, wann welche Anforderungen live gehen müssen. Nach diesem richtet sich auch die Sprint Planung. In der Sprint Planung werden die Anforderungen mit höchster Priorisierung im Product Backlog zuerst ausgewählt. Die Priorisierung im Product Backlog beachtet dabei aber nicht nur den Kundennutzen/Wert, sondern ebenso das Risiko. Basierend auf dem Nutzen/Risiko Ausgleich werden die User Stories priorisiert und in der Sprint Planung für die Umsetzung ausgewählt.

Das Sprint Backlog in der Sprint Planung ist entscheidend für den Start des Controlling Prozesses. Hier werden die Ausgangsmesswerte festgelegt, von denen später in der Product Review (und während des Sprints) die Abweichungswerte gemessen werden. Das Sprint Backlog ist sozusagen ein Auszug aus dem Product Backlog und enthält all die Anforderungen, die basierend auf Kundennutzen und Risiko als nächstes umgesetzt werden sollen. Wichtig zu beachten ist hierbei, dass eine Bewertung dieser Anforderungen in Story Points stattfindet. Diese Story Points bewerten die Komplexität einer User Story. Story Points werden zweckmäßig für die Schätzung und Planung herangezogen und können später aber in andere Werte umgerechnet werden. Für den Anfang bilden sie jedoch unsere Basis für die Planung und das Controlling.



Quelle: Eigene Darstellung

## CONTROLLING PROZESS

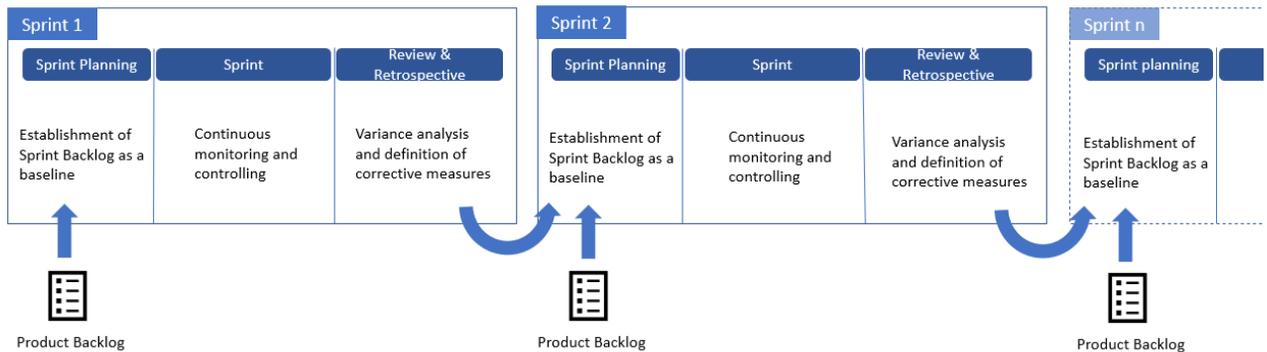
Ein weiterer wichtiger Ausgangswert ist die Velocity (auch basierend auf Story Points). Diese stellt die durchschnittliche Entwicklungsgeschwindigkeit des Entwicklungsteams pro Sprint dar. Zu Beginn ist die Velocity meist ein Erfahrungswert und ergibt sich aus historischen Werten, nachdem aber ein paar Sprints vollendet sind, kann der Durchschnitt der Geschwindigkeit gemessen werden. Die Velocity ist für uns unheimlich wichtig, denn sie hilft uns bei der Planung, wie viele Sprints insgesamt nötig sind, um das Projekt zu beenden. Durch die Abschätzung des Product Backlogs wissen wir nun, wie viele Story Points das Product Backlog enthält und die (gemessene oder anfangs festgelegte) Velocity sagt uns wieviel wir davon in einer Iteration schaffen. Dadurch ergibt sich dann die Anzahl der geplanten Sprints und wir können eine grobe Zeitabschätzung für das Gesamtprojekt treffen. Natürlich ist zu beachten, dass sich das Product Backlog im Laufe der Zeit anpasst und die anfangs noch sehr vagen Schätzungen verfeinert und geändert werden. Jedoch werden diese Anpassungen in jedem Sprint und somit auch in der Planung neu mit aufgenommen. Dadurch ergibt sich eine zunehmend detailliertere und genauere Planung, die immer auf aktuellem Stand ist. Die Anzahl der benötigten Sprints, um das Product Backlog umzusetzen, wird sich demnach auch ändern. Der Vorteil ist hier jedoch, dass das neue geplante Fertigstellungsdatum pro Sprint errechnet werden kann und immer aktuell ist. Da das Entwicklungsteam in der Schätzung immer besser und präziser wird, verbessert sich auch die Planung und das Controlling, im Gegensatz zu traditionellen Projekten.

Story Points helfen auch für die Budgetierung des Projektes. Sie lassen sich letztendlich in Zeit umrechnen, und somit auch in Kosten. Das Projektbudget kann man durch die Anzahl der Sprints teilen, um ein Budget pro Sprint zu erhalten. Dies ist leicht umsetzbar, da die Länge der Sprints in Scrum immer gleichbleibend ist. Besonders für die Software Entwicklung lässt sich dies leicht umsetzen. Während oder spätestens am Ende jedes Sprints sieht man dann in der Abweichungsanalyse, ob das Budget pro Sprint eingehalten wurde. Der große Vorteil des agilen Controlling Prozesses ist nun, dass man diese Abweichungen direkt im nächsten Sprint wieder gut machen kann.

Während der Sprint Phase können Burn Charts zur Darstellung des Entwicklungsfortschrittes und der Leistung herangezogen werden. Das hilft dem Entwicklungsteam direkt im Sprint Verbesserungen vorzunehmen. Die Sprint Review und Retrospective sind ein weiterer wichtiger Teil vom agilen Controlling Prozess. Hier kann eine Abweichungsanalyse durchgeführt werden. Die in der Sprint Planung geplanten Story Points können nun mit den tatsächlich umgesetzten Story Points verglichen werden. Diese können auch hier wieder in Kosten umgerechnet werden, um das Budget zu verfolgen. Die User Stories mit nicht umgesetzten Story Points können direkt in den nächsten Sprint geschoben werden, ohne kosten- und zeitintensive Change Requests anfordern zu müssen. Für die Umsetzung von Änderungen muss nicht in vorherige Phasen zurückgegangen werden, wie in traditionell durchgeführten Projekten. Die geänderten Anforderungen können zusammen mit den neu ausgewählten Anforderungen im neuen Sprint umgesetzt werden. Die Grafik auf der nächsten Seite visualisiert diesen Prozess.

## CONTROLLING PROZESS

Die neu ausgewählten User Stories aus dem Product Backlog ergeben zusammen mit den noch umzusetzenden User Stories aus dem vorherigen Sprint den neuen Ausgangs- und Planwert des neuen Sprints. Der Controlling Prozess beginnt also hier erneut.



Quelle: Eigene Darstellung

Die Velocity passt sich bei nicht umgesetzten Story Points entsprechend an und das Entwicklungsteam lernt, dass es eventuell weniger einplanen sollte, um sein Ziel zu erreichen. Dies kann direkt im neuen Sprint berücksichtigt werden. So wird auf Dauer die Planung immer genauer und somit natürlich auch das Controlling.

Für Kunden, die eine längerfristige Planung benötigen, können zusätzlich fixe Meilensteine gesetzt werden, um ihnen Sicherheit und eine etwas längerfristige Planung zu geben. Es muss jedoch beachtet werden, dass das die Entwicklung des Teams beeinflussen kann.

Das Controlling, wie es hier beschrieben ist, kann als sogenanntes Lean Controlling beschrieben werden. Es reduziert Planungsaufwand und Change Requests. Die Planung in Scrum ist eher eine rollende Planung, die sich ständig pro Sprint anpasst und dadurch genauer und aktueller ist. Dadurch, dass Änderungen und Kundenwünsche direkt um nächsten Sprint umgesetzt werden können, wird die Planung verbessert und effizienter gestaltet. Das Controlling wird somit auch effektiver, da man Abweichungen von einem realistischen Plan misst anstatt von einem Plan, der auf alten Werten basiert.

## KEY PERFORMANCE INDICATORS

# Was soll ich eigentlich messen?



Da der agile Controlling Prozess nun klar ist, werden euch hier Kennzahlen vorgestellt, die ihr in Scrum messen könnt. Die Auswahl dieser hier vorgeschlagenen KPIs basiert auf den Literatur- und Interviewergebnissen. Da jedes Projekt einzigartig ist, kann es sein, dass eine unterschiedliche Auswahl an Kennzahlen erforderlich ist, um das Projekt zu steuern. Deswegen sollten die Kennzahlen aus diesen Vorschlägen individuell ausgesucht werden. Generell gilt: Es sollten nur die Kennzahlen berechnet werden, die einen Mehrwert für das Projekt bringen. Es ist wichtig, die richtigen Kennzahlen zu berechnen, anstatt einer Vielzahl für das Projekt bedeutungsloser Parameter, die nur Aufwand erzeugen. Wenige, aber alle nötigen Kennzahlen sollten häufig und regelmäßig gemessen werden. Es zählt: few, but necessary!

Die Tabelle der nächsten Seite zeigt euch alle KPIs, ihre Basis und in welchem Teil des Controlling Prozesses man sie messen kann. Die darauffolgenden Seiten geben eine noch genauere Erklärung ab.

## KEY PERFORMANCE INDICATORS

Die nachfolgende Tabelle zeigt welche KPIs im agilen Controlling Prozess gemessen werden können und liefert Informationen zur Berechnung und zum Verständnis.

Controlling Prozess	KPIs	Informationen	Gemessen anhand
Sprint Planning	Risiken	Priorisiertes Product Backlog anhand Nutzen/Risiko Ausgleich: Anforderungen mit höchstem Nutzen und höchstem Risiko werden zuerst umgesetzt; detaillierere Risikomessung kann hier ebenso stattfinden	Product Backlog
	Story Points	Plan aller umzusetzenden Story Points im Sprint	Sprint Backlog
	Velocity	Festlegung der Summe der Story Points, die umgesetzt werden sollen, basierend auf der Velocity	Sprint Backlog
	Budget	Budgetaufteilung auf alle Sprints: das Budget pro Sprint kann ermittelt werden, indem die Summe aller zu realisierenden Story Points durch die Velocity geteilt wird. Dann erhält man die Anzahl der benötigten Sprints und kann das Budget daran aufteilen. Voraussetzung ist, dass man mit einem konstanten Team arbeitet und die Länge der Sprints immer gleich ist.	Sprint Backlog
Während des Sprints	Qualität	Gemessen in "defects and bugs" (Fehlern) in der Software; sollte durch automatisierte Tests durchgeführt und vereinfacht werden	Sprint Backlog
	Fortschritt	Vergleich des Sprint Ziels (basierend auf Story Points) mit der erbrachten Leistung des jeweiligen Tages	Sprint Backlog
	Impediments	Impediments Tracking im Daily Scrum	Sprint Backlog
Sprint Review & Retrospective	Story Points	Ausführung einer Abweichungsanalyse (Sprint Ziel vs. erbrachte Leistung)	Sprint Backlog
	Velocity	Ausführung einer Abweichungsanalyse (Sprint Ziel vs. erbrachte Leistung)	Sprint Backlog
	Kundenzufriedenheit	Messung der Kundenzufriedenheit, um Änderungen im nächsten Sprint umzusetzen, Erfüllung der Akzeptanzkriterien	Sprint Backlog
	Mitarbeiterzufriedenheit	Messung der Mitarbeiterzufriedenheit pro Woche oder Sprint und Diskussion in der Retrospektive	Sprint Backlog
	Qualität	Gemessen in "defects and bugs" (Fehlern) in der Software; sollte durch automatisierte Tests durchgeführt und vereinfacht werden	Sprint Backlog
	Risiken	Identifikation überbliebener Risiken und Maßnahmenumsetzung im nächsten Sprint	Sprint Backlog
	EVM Metriken	Planned Value (in Story Points oder Stunden), Actual Cost (in Story Points oder Stunden), Earned Value (in Story Points oder Stunden), Budget at Completion, Schedule Performance (Index), Cost Performance (Index), Estimate at Completion, Estimate to Complete, Planned Percent Complete, Actual Percent Complete	Sprint Backlog
Budget	Abweichungsanalyse des geplanten und tatsächlichen Budgets; kann erst auf Basis der Story Points durchgeführt und später umgerechnet werden. Nicht umgesetzte Story Points werden in den nächsten Sprint verschoben, was den Entwicklungsprozess verlängern könnte. Somit müsste auch das Budget neu angepasst werden.	Sprint Backlog	

Quelle: Eigene Darstellung

## KEY PERFORMANCE INDICATORS

### Risiken

Risiken werden schon auf Product Backlog Level festgehalten. Jede definierte User Story sollte bezüglich ihres Nutzen und Risikos bewertet werden. Die Priorisierung erfolgt dann auf der Basis, dass wichtigste und zugleich auch risikoreichste Anforderungen zuerst umgesetzt werden sollen und somit auch höher priorisiert werden. Risikocontrolling ist also schon automatisch Teil des Scrum Prozesses. Der Product Owner sollte diese Risiken dem Entwicklungsteam in der Sprint Planung auch kommunizieren. Eine detailliertere Risikoanalyse kann zudem noch extra durchgeführt werden.

### Kosten

Wie in dem Controlling Prozess schon vorher beschrieben, dienen Story Points im agilen Controlling als *der* Ausgangswert. Auf der Basis von geplanten Story Points, kann eine Ziel Velocity definiert werden, welche das Entwicklungsteam erreichen sollte. Anhand dessen kann das Budget auf die Sprints aufgeteilt werden. Da man die Summe der Story Points des Product Backlogs kennt und die anfängliche Velocity oder eine Ziel Velocity definiert wurde, kann berechnet werden, wie viele Sprints benötigt werden, um alle Anforderungen umzusetzen. Wenn wir also die Zeit kennen, die das Team braucht, um eine gewisse Anzahl von Anforderungen umzusetzen, dann können wir diese in Kosten umrechnen. Dies funktioniert besonders in Softwareentwicklungsprojekten, da Zeit den größten Kostenfaktor darstellt. Also bezieht sich auch Kostencontrolling in erste Linie auf Story Points, bevor diese über die Zeiteinheit in Kosten umgerechnet werden. Die Voraussetzung dafür ist, dass mit einem konstanten Entwicklungsteam gearbeitet werden soll und die Länge der Sprints immer gleich ist.

### Qualität

Generell ist eine Qualitätsüberprüfung inhärent in Scrum. Durch die häufigen Kundenfeedbacks wird die Qualität dauernd überprüft. Auch die Definition der Akzeptanzkriterien zur Abnahme des Inkrements in der Product Review unterstützt diesen Prozess. Zusätzlich wird das Inkrement aber auf Fehler (defects und bugs) im gesamten Entwicklungszeitraum getestet. Hier können automatisierte Tests und ein Vier-Augen-Prinzip durch Pair Programming helfen.

### Fortschritt

Der Projekt- /Entwicklungsfortschritt kann anhand von Burn Charts auf der Basis von Story Points dargestellt werden. Für die Darstellung helfen auch Tools wie Jira oder Confluence, mit denen solche Diagramme automatisch dargestellt werden. Schon während des Sprints kann der Fortschritt pro Tag ermittelt werden. Das gibt dem Entwicklungsteam Hilfestellung und visualisiert die bisher erbrachte Leistung. Auch am Ende des Sprints kann der Fortschritt gemessen werden. Durch eine Abweichungsanalyse von geplanten und tatsächlich erreichten Story Points kann man ein Fazit über die erbrachte Leistung ziehen. Der Fortschritt des gesamten Projektes kann ebenso ermittelt werden, in dem man alle bisher erbrachten Leistungen mit dem Inhalt des Product Backlogs vergleicht.

## KEY PERFORMANCE INDICATORS

### Kunden- und Mitarbeiterzufriedenheit, Impediments

Kunden- und Mitarbeiterzufriedenheit sind genau so wichtige Kennzahlen wie Qualität und Kosten. Denn beide wirken sich indirekt auf die Qualität und somit auch auf die Kosten aus. Der Scrum Master hat die Aufgabe die Mitarbeiterzufriedenheit sicherzustellen und sammelt Kommentare und Verbesserungsvorschläge für die Zusammenarbeit. Diese werden in der Sprint Retrospective analysiert. Grundvoraussetzung für eine erfolgreiche Entwicklung mit Scrum ist eine gute Zusammenarbeit des Teams, daher stellt die Mitarbeiterzufriedenheit einen wichtigen Bestandteil der Messung dar. Eine Möglichkeit der Mitarbeiterzufriedenheitsmessung kann die Sammlung von Sticky Notes oder Kommentaren auf einem White Board darstellen. Hier kann das Team in Form von Noten oder Smileys die eigene Zufriedenheit im Projekt bewerten.

Impediments können während des Sprints direkt im Daily Scrum diskutiert werden. Der Scrum Master hat die Aufgabe diese zu sammeln und zu beseitigen. Die Zufriedenheit der Mitarbeiter kann dabei auch ein Impediment darstellen und ist somit auch Teil des Impediments Tracking.

Kundenzufriedenheit wird direkt in der Sprint Review überprüft, indem der Kunde das Inkrement durch seine Anwesenheit überprüft. Die vorher definierten Akzeptanzkriterien dienen dazu, sicherzustellen, dass alle Bedingungen der jeweiligen Anforderung erfüllt wurden. Der Kunde kann zudem weitere Änderungswünsche äußern, die direkt im nächsten Sprint implementiert werden.

### Earned Value Management Metriken

Am Ende jedes Sprints wird eine Abweichungsanalyse durchgeführt, um zu sehen, ob das Sprint Ziel erreicht wurde. Hier kommt Earned Value Management (EVM) ins Spiel. Eine kurze Einführung findet ihr hier:

„**Earned Value Management (EVM)** ist eine Controlling Methode, mit der auf der Basis von Plan- und Ist-Daten Kennzahlen berechnet werden, die den Projektfortschritt darstellen und das Projekt hinsichtlich Kosten, Zeit und Leistung bewerten. Im Gegensatz zu anderen Methoden, wie eine Kosten-Trend-Analyse oder ein Meilensteinplan, integriert EVM alle drei Dimensionen (Kosten, Zeit und Leistung). EVM liefert wichtige Informationen wie ein voraussichtliches Projektenddatum und Kosten. Hauptgrundlage ist die Berechnung des sogenannten Earned Value (Leistungswert).

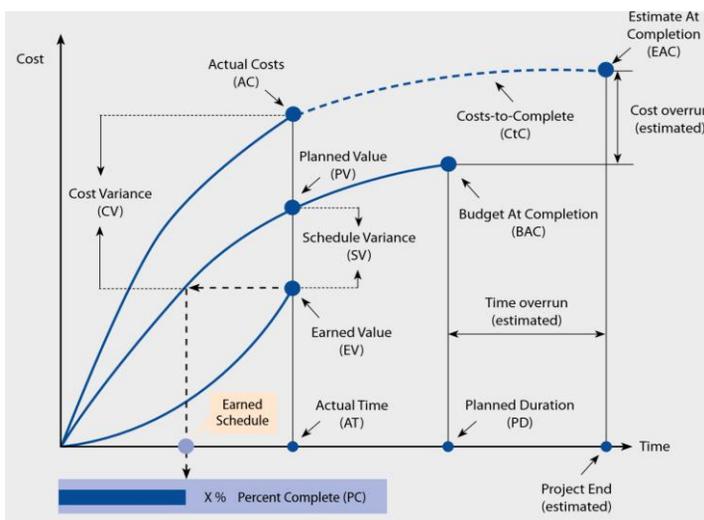
Weitere Informationen zu EVM können hier entnommen werden:  
<https://www.projektmagazin.de/methoden/earned-value-management>

In der Literatur wird Earned Value Management in agilen Projekten oft auch gern als Agile EVM bezeichnet. Die Literaturrecherche, sowie die Interviewergebnisse zeigen, dass Earned Value Management auch in agilen Projekten eine gängige Controlling Methode ist, da sie Scrum zusätzliche Informationen liefert. Auf der folgenden Seite werden die EVM Metriken, wie sie in traditionellen und agilen Projekten angewandt werden, kurz beschrieben.

## KEY PERFORMANCE INDICATORS

Earned Value Management kombiniert alle relevanten Erfolgsfaktoren, anstatt nur einen davon zu messen. Im Vergleich zu anderen Controlling Methoden berechnet EVM die erbrachte Leistung, also den eigentlichen Wert des Projektes unter Berücksichtigung von Zeit und Kosten. Hier scheitern viele andere Berechnungsmethoden. Earned Value Management setzt die wichtigsten Größen wie Zeit, Kosten und Fortschritt in Relation zueinander. Die anfänglichen Messgrößen sind Actual Costs (AC – Ist-Kosten), Planned Value (PV - Planwert) und Earned Value (EV - Leistungswert). Der Leistungswert (EV) umfasst alle erbrachten Arbeitspakete im Vergleich zum Planwert (PV). Sind die Arbeitspakete so erbracht, wie geplant, so ist  $EV=PV$ .

Die unteren Grafiken zeigen die wichtigsten Messgrößen und ihre Darstellung:



Quelle: Menze, S. (2013). Projekt-Controlling mit Earned Value und Earned Schedule. Controlling and Management Review.

Abkürzung	Begriff	Berechnung	Definition
EV	Earned Value		Soll-Kosten zum Stichtag
PV	Planned Value		Plan-Kosten zum Stichtag
AC	Actual Cost		Aktuelle Ist-Kosten zum Stichtag
BAC	Budget at Completion		Geplante Gesamtkosten
SPI	Schedule Performance Index	$SPI=EV/PV$	Kennzahl über den Zeitplan
CV	Cost Variance	$CV=EV-AC$	Kostenabweichung (Soll-Ist Vergleich)
CPI	Cost Performance Index	$CPI=EV/AC$	Wirtschaftlichkeitsfaktor
SV	Schedule Variance	$SV=EV-PV$	Planabweichung (Soll-Plan Vergleich)
EAC	Estimate at Completion	$EAC=AC+((BAC-EV)/CPI)$	Geschätzte Gesamtkosten bei Fertigstellung

Quelle: Wanner, R. (2013). Earned Value Management - die wichtigsten Methoden und Werkzeuge für ein wirkungsvolles Projektcontrolling. CreateSpace Independent Publishing Platform.

## KEY PERFORMANCE INDICATORS

Die Kennzahlen des Agile EVM in Scrum Projekten sind gleich, jedoch basieren die Werte auf Story Points, bzw. auf Stunden, bevor sie in Kosten umgerechnet werden. Das muss bei der Anwendung des Agile EVMs beachtet werden. Zusätzlich wird EVM auch pro Sprint berechnet, da die Plangrößen erst im Sprint genau festgelegt werden. So werden hohe Abweichungen aufgrund sich ändernder Anforderungen vermieden. Earned Value Management ist ein nützliches Tool für das Controlling, es berücksichtigt aber nicht alle wichtigen Kennzahlen, die gemessen werden sollten. Qualität, Risiken, Kunden- und Mitarbeiterzufriedenheit missachtet Earned Value Management in seiner Kennzahlenberechnung und sollten daher zusätzlich gemessen werden.

Wie genau so ein Agile EVM in agilen Projekte integriert werden kann, zeigt euch das Kapitel der Iteration Scorecard.

## ITERATION SCORECARD



# Wie soll ich kontrollieren?

Um den Projektstatus und –fortschritt gut zu überwachen und darzustellen, wurde eine sogenannte Iteration Scorecard auf Excel Basis entwickelt. Diese kann in agilen Projekten zum Einsatz kommen, um relevante KPIs in jedem Sprint zu verfolgen. Die Iteration Scorecard muss an das jeweilige Projekt angepasst werden und durch entsprechende zusätzliche Parameter ergänzt werden, falls diese für die Steuerung nötig sind. Neben Tools, wie Jira und Confluence, kann die Iteration Scorecard Metriken des Earned Value Managements erfassen. In jedem Sprint wird der vorher festgelegte Planned Value (PV) auf Stundenbasis mit dem Leistungswert (Earned Value – EV) pro Sprint verglichen. Die Abweichung wird entsprechend notiert und es kommt zu einer Verzögerung des Auslieferungsdatums des Produktes. Die folgende Grafik stellt ein Beispiel mit fiktiven Werten solch einer Iteration Scorecard dar. Die Vorlage, die dementsprechend anzupassen ist, befindet sich auf dem Laufwerk. Eine nähere Erklärung der einzelnen Positionen findet ihr auf den darauffolgenden Seiten.

Agile EVM Iteration Scorecard								
Sum of story points in PB	150							
Assumed story points per sprint	30							
	Start	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Total
PV (cum hours)		1600	3200	4800	6400	8000	8213	
PV (iteration hours)		1600	1600	1600	1600	1600	213	
Planned user stories		4	5	4	4	5	1	
Planned story points		30	26	30	30	32	4	
Average velocity		30	30	28	28,67	28,5	29,2	
number of iterations until completed (adjusted)		5	5	5,36	5,23	5,26		
Completed user stories		4	5	4	3	5	1	22
Completed story points		30	26	30	28	32	4	150
Remaining story points to be completed	150	120	94	64	36	4	0	
User stories deferred		0	0	0	1	0	0	1
Replanned user story count					2			
Replanned user story value					106,67			
EV (iteration)		1600	1600	1600	1493,33	1600	213	8106
EV (cum)		1600	3200	4800	6293,33	7893,33	8106	
SV (it)		0	0	0	-106,67	0	0	-106,67
SV (cum)		0	0	0	-106,67	-106,67	-106,67	-106,67
Delay in days		0	0	0	2	0	0	2
Sprint duration (wks)		4	4	4	4	4	4	
FTE resources		10	10	10	10	10	10	
EAC for original scope		8000	8000	8572	8373	8422		

Quelle: Eigene Darstellung

## ITERATION SCORECARD

Die Werte, die gemessen werden, setzen sich zusammen aus Metriken des Earned Value Managements zusammen mit Story Points. In der Literaturrecherche, sowie in den Interviews wurde herausgefunden, dass Earned Value Management eine gängige Methode des Controllings ist, auch in agilen Projekten. Deswegen wurden EVM Metriken in die Iteration Scorecard integriert.

### **Legende der Iteration Scorecard:**

PV (cum hours) – kumulierte geplante Stunden

PV (it hours) – geplante Stunden pro Iteration

Planned user stories – Anzahl der geplanten User Stories pro Iteration

Planned story points – Anzahl der geplanten Story Points pro Iteration

Average velocity – Durchschnittsgeschwindigkeit des Entwicklungsteams (gemessen in tatsächlich realisierten Story Points der letzten Sprints)

Number of iterations until completed (adjusted) – Anzahl der geplanten Sprints, um das Product Backlog umzusetzen (angepasst an die derzeitige Velocity)

Completed user stories – Anzahl tatsächlich realisierter User Stories

Completed story points – Anzahl tatsächlich realisierter Story Points

User stories deferred – Anzahl der nicht planmäßig realisierbaren User Stories

Replanned user story count – Anzahl der umdisponierten User Stories

Replanned user story value – Stundenwert der umdisponierten User Stories

EV (it) – Earned Value = Leistungswert pro Iteration

EV (cum) – kumulierter Leistungswert

SV (it) – Schedule Variance = Terminabweichung pro Iteration

SV (cum) – kumulierte Terminabweichung

Delay in days – Verzug gemessen in Tagen aufgrund umdisponierter User Stories

Sprint duration (wks) – Sprintlänge in Wochen

FTE resources – Full time employee = Vollzeitmitarbeiter

EAC for original scope – Estimate at Completion = prognostizierter Stundenaufwand zu Projektende bei ursprünglichem Umfang

## ITERATION SCORECARD

Das auf Seite 14 in der Grafik beschriebene Beispiel soll hier nun weiter erläutert werden, um die Rechnung zu verstehen.

### Grundannahme:

Das Beispiel geht von einem Product Backlog mit insgesamt 150 Story points aus. Basierend auf Erfahrungswerten wird die Velocity mit 30 Story Points bemessen, bis diese nach den ersten Sprints gemessen werden kann. Das bedeutet, dass es insgesamt 5 Sprints benötigt, um das gesamte Product Backlog mit den 150 Story Points umzusetzen. Es wird davon ausgegangen, dass das Entwicklungsteam aus 10 Personen besteht, die jeweils 8 Stunden pro Tag arbeiten. Da jeder Sprint 4 Wochen dauert, ergibt sich ein Planwert (PV) von 1600 Stunden, die das gesamte Entwicklungsteam zur Verfügung hat, um die geplanten 30 Story Points umzusetzen. Daraus würde sich ein prognostizierter Stundenaufwand zu Projektende von 8000 Stunden ergeben, bei einer Annahme von 5 Sprints.

### Erste Iteration:

Für die erste Iteration ist die Umsetzung von 4 User Stories geplant, welche 30 Story Points entsprechen. Wir sehen in der Review, dass diese 30 Story Points wie geplant umgesetzt werden konnten. Daraus ergibt sich ein Earned Value (Leistungswert) von 1600 Stunden, der auch unserem Planwert von 1600 Stunden entspricht. Folglich haben wir keine Terminabweichung und unser Estimate at Completion ist weiterhin 8000 Stunden, da wir unsere Velocity einhalten konnten.

### Zweite Iteration:

In der zweiten Iteration planen wir nur 26 Story Points anstatt 30. In der Review sehen wir, dass unser Leistungswert immer noch unserem Planwert entspricht. Wir haben nur 26 Story Points umgesetzt, aber wir haben auch nur 26 geplant. Daher ergibt sich keine Terminabweichung. Jedoch passt sich im nächsten Sprint die Velocity an, da wir ursprünglich von 30 Story Points pro Iteration ausgegangen sind, um unser mögliches Projektende zu prognostizieren.

### Dritte Iteration:

Hier können wir nun diese Veränderung beobachten. Unsere Velocity sinkt auf 28 statt 30 Story Points und somit verändert sich auch unser Estimate at Completion, also die Stunden, die bis zum Projektende prognostiziert werden. Weil wir nun mit einer geringeren Velocity rechnen, aber dennoch von insgesamt 150 Story Points ausgehen, kommen wir auf 8572 benötigte Stunden, um unser Backlog umzusetzen. In diesem Sprint planen wir wieder 30 Story Points und setzen diese auch um. Das bedeutet, dass unsere Velocity im vierten Sprint wieder steigt. Unser Leistungswert entspricht auch hier unserem Planwert, somit haben wir keine Abweichung von der Planung.

## ITERATION SCORECARD

### **Vierte Iteration:**

Hier können wir das erste mal eine Terminabweichung feststellen. Wir planen 30 Story Points und setzen davon aber nur 28 um. Eine User Story, die mit 2 Story Points bewertet wurde, konnte nicht planmäßig umgesetzt werden. Deswegen haben wir ein Replanned User Story Value von 106,67 Stunden. Das ist der Stundenwert, den wir nicht umsetzen konnten. Unser Leistungswert ist aufgrund dessen auch nicht 1600, sondern nur 1493,33 Stunden. Für diese Rechnung ziehen wir die 106,67 Stunden von unserem PV = 1600 Stunden ab. Die 106,67 Stunden ergeben sich daraus, dass unser Entwicklungsteam, bestehend aus 10 Personen, 53,33 Stunden benötigt, um einen Story Point umzusetzen. Diese 106,67 stehen für unsere Terminabweichung und müssen im nächsten Sprint nachgeholt werden. In diesem Sprint sinkt unser Estimate at Completion trotzdem, da wir wieder von einer erhöhten Velocity im Gegensatz zu Sprint 3 ausgegangen sind.

### **Fünfte Iteration:**

Diese Iteration war ursprünglich die geplant letzte, da wir von einem Estimate at Completion von 8000 Stunden ausgegangen sind. Da wir aber eine Terminabweichung hatten, wird sich unser Entwicklungsprozess verzögern und wir benötigen einen Sprint mehr. Im fünften Sprint konnten wieder alle geplanten 30 Story Points umgesetzt werden und unser Leistungswert entspricht dem Planwert. Die in Iteration 4 überbliebenen Story Points konnten hier umgesetzt werden. Am Ende der Iteration 5 verbleiben nun noch 4 Story Points. Diese ergeben sich daraus, dass wir im zweiten Sprint nur 26 Story Points geplant und umgesetzt haben, aber eine Velocity von 30 angenommen wurde.

### **Sechste Iteration:**

Somit ist eine sechste Iteration notwendig, um die letzten Story Points mit einem Stundenwert von 213 umzusetzen. Am Ende der Iteration können wir sehen, dass unser Planwert erreicht wurde. Insgesamt haben wir eine Terminabweichung (SV) von nur 106,67 Stunden, da nur die Stunden als Abweichung zählen, die wir tatsächlich nicht wie geplant umsetzen konnten (also nur die aus Sprint 4).

Anhand dieses Beispiels kann man sehen, wie Earned Value Management in Scrum zum Einsatz kommen kann und zusätzliche Informationen liefert. Es können natürlich noch andere Kennzahlen berechnet werden, falls diese benötigt werden. Eine Umrechnung der Stunden in Kosten kann auf der Basis dieser Vorlage auch erfolgen.

## ANMERKUNGEN

# Was muss ich beachten?

Bei den Forschungsergebnissen, die in diesem Guide gesammelt werden, sollten einige Punkte beachtet werden. Die Interviews wurden in verschiedensten Branchen und Unternehmen durchgeführt. Da der Guide eine Sammlung dieser Interviewergebnisse der Unternehmen ist, sollte beachtet werden, dass Controlling und Agilität in jedem Unternehmen unterschiedlich gelebt wird. Es wurden kleine, mittelständische wie große Unternehmen befragt und deren Umsetzung von Agilität ist teils sehr individuell. So verhält sich auch das Controlling. Es gibt keine vorgefertigte Lösung, denn das Controlling muss in Scrum in jedem Projekt angepasst werden. Das Framework ist als eine Art Empfehlung zu behandeln.

Der Unterschied im Controlling wird besonders bei kleinen und großen Unternehmen deutlich. Kleinere Firmen können viel agiler vorgehen als Großkonzerne und benötigen ein weniger striktes Controlling aufgrund weniger vorhandener Strukturen und festgefahrener Prozesse. Großkonzerne müssen eventuell mehrere Sprints vorausplanen und Meilensteine einführen, weil interne Strukturen das so vorgeben. Deswegen muss jedes Unternehmen für sich entscheiden, inwiefern es Agilität leben kann und das beeinflusst wiederum das Controlling. Der Kunde spielt dabei eine wesentliche Rolle. Erwartet dieser mehr Planungs- und Prognosesicherheit, so muss ihm dies trotz Scrum Regelwerk gegeben werden. Für manche Unternehmen eignet sich daher eine hybride Vorgehensweise, indem sich der Kunde mit den Scrum Prozessen und dem Regelwerk vertraut machen kann. Die Veränderungen, die durch Scrum entstehen, stellen für viele Firmen eine Herausforderung dar. Mittels hybridem Projektmanagement können sich Unternehmen an Agilität herantasten und so kann nach und nach eine Transformation zu einem rein agilen Projekt entstehen. Die Praxis hat gezeigt, dass dies in manchen Projekten gut funktioniert, da der Kunde sich mit den neuen Anforderungen vertraut machen kann. Er muss verstehen, wie sich Scrum auf das Controlling auswirkt und gewisse Grenzen akzeptieren.

Generell sollte beachtet werden:

- Controlling muss als Mehrwert verstanden und behandelt werden
- Transparenz ist eine Grundvoraussetzung für agiles Controlling; Jeder muss den Sinn der Kennzahlen verstehen
- Controlling bedeutet kein Micromanagement, es ist eine unterstützende Servicefunktion
- Etwas Planungs- und Prognosesicherheit gehen verloren
- Agiles Projektcontrolling fordert ein Umdenken aller Beteiligten

Zusätzlich sollte man eine klare Definition von KPIs haben. Auch das Entwicklungsteam muss den Hintergrund dieser Kennzahlen verstehen. Es geht im Controlling nicht darum, das Entwicklungsteam einzuschränken, sondern um die Schaffung von Transparenz, sodass alle einen Mehrwert in den gelieferten Zahlen sehen. Die Zusammenarbeit kann dahingehend verbessert werden, was sich positiv auf den Entwicklungsprozess und die Kosten auswirkt. Grundsätzlich erfordert agiles Projektcontrolling ein Umdenken von Controllern und Kunden. Agilität muss akzeptiert und verstanden werden, um sicherzustellen, dass Controlling möglich ist, ohne dabei den Mehrwert der agilen Entwicklung zu verlieren.

## Was ist nun anders?

Generell können alle Aufgaben und Verantwortlichkeiten des Controllings in agilen Projekten wahrgenommen werden. Ebenso können alle nötigen Kennzahlen berechnet werden. Es ändert sich lediglich der Zeithorizont des Controllings. Es werden Pläne für kürzere Zeiträume erstellt und die Ausgangswerte werden in jedem Sprint neu definiert (neue und nicht fertiggestellte Anforderungen). Die Basis der Kennzahlenberechnung ändert sich und nimmt Story Points als Grundlage. Dennoch können gängige Controlling Methoden wie Earned Value Management angewandt werden. Das Controlling ergänzt Scrum in vielerlei Hinsicht und ist aber teils gleichzeitig ein Teil davon. Zusätzlich liegt der Fokus in Scrum nicht nur auf dem Kostencontrolling, sondern vielmehr auf der eigentlichen Wertschöpfung.

Ein traditionelles Controlling kann man Scrum nur schwer aufzwingen. Die langfristige Planbarkeit und die genauere Formulierung des Projektumfangs würde sich gegen die Dynamik und Flexibilität von Scrum richten. Es ist aber nicht nötig, in Scrum ein traditionelles Controlling erzwingen, da es auch anders geht. Wenn man das Controlling an die jeweiligen Projektanforderungen anpasst, dann kann man Kennzahlen mindestens genau so gut messen und das Projekt steuern.

Aber vor allem: Agiles Projektcontrolling ist effektiver und effizienter als traditionelles Projektcontrolling. Pläne werden ständig aktualisiert und sind somit präziser. Dies wirkt sich auch auf das Controlling aus. Änderungen können zusätzlich direkt im nächsten Sprint übernommen und Maßnahmen in die neuen Planwerte mit eingebaut werden. Das vereinfacht die Maßnahmenumsetzung.