

Der dornige Weg zum agilen Team, Teil 6: Metriken – Ballast oder Notwendigkeit?

Schlachtbericht

■ VON JIRI LUNDAK

Messen liegt in der Natur des Menschen. Er will vergleichen, analysieren und informiert sein. Deshalb hat der Mensch auch ein ambivalentes Verhältnis zum Messen und „Gemessenwerden“. Im Projektmanagement werden Metriken – also das Messen und Vergleichen – von Werten als zentrale Aktivität betrachtet, sozusagen als Notwendigkeit. Dies, um z.B. den Projektfortschritt belegen zu können oder um die Güte eines Teams oder von geleisteter Arbeit abschätzen zu können. Metriken werden deshalb oft vom Management gefordert. Vom Softwareentwickler hingegen werden sie oft geschmäht. Welches Lager hat Recht?

Wie so oft liegt die Wahrheit irgendwo in der Mitte. Um beide Parteien zu verstehen, ist es sinnvoll, sich zuerst einmal mit den Hintergründen zu beschäftigen, warum es hier überhaupt ein Zerwürfnis gibt. Die zunehmende Bedeutung von Metriken kann man leicht zurückverfolgen bis in die Zeit von Frederick Taylors „Scientific Management“ [1]. Taylor verstand es meisterlich, alles Mögliche zu messen und anschließend ganze Produktionsprozesse

so zu optimieren, dass die Metriken befriedigt wurden, allerdings mit sehr zweifelhaftem Erfolg, besonders was Qualität anbelangt.

Trotzdem liegt es in der Natur der Sache, dass das Management, z.B. in einem Projekt, gerne weiß, wo das Projekt steht und ob der ursprüngliche Plan eingehalten werden kann oder nicht. Leider ist aber das Management oft sehr weit entfernt vom eigentlichen Projektverlauf (man könnte manchmal auch sagen vom „Schlachtfeld“). Deshalb wird oft auf irgendwelche Kennzahlen (lies Metriken) vertraut, die den Herren in den oberen Etagen irgendwie bestätigen sollen, dass alles in bester Ordnung ist.

Ganz anders diejenigen, die sich mitten in der „Schlacht“ befinden. Sie kämpfen jeden Tag mit dem Compiler oder der Entwicklungsumgebung. Der Build läuft wieder einmal nicht und der Application Server führt ein Eigenleben. Wie sollen Metriken da helfen? Warum soll da ein Entwickler noch irgendwelche bedeutungslosen Zahlen liefern?

Vom Sinn und Unsinn

Da ein agiles Team in einem konstanten Inspektions- und Adaptionszyklus lebt, ist es darum bemüht, sich selbst und seinen Fortschritt zu messen. Allerdings müssen die verwendeten Metriken die agile Vor-

gehensweise fördern und nicht nur dem Selbstzweck dienen. Die wichtigsten Kriterien für Metriken, die im agilen Umfeld wirksam sein sollen, sind:

1. Förderung von agilem Verhalten
2. Messen von „wertschöpfender“ Arbeit und nicht von Arbeitsumfang
3. Aufzeigen von Trends
4. Offenlegung des Kontextes der Metrik (was beeinflusst sie etc.)
5. Förderung der Kommunikation im Team und teamübergreifend

Ein agiles Team wird normalerweise von einer zentralen Metrik angetrieben: dem geschaffenen Wert (des fertig gestellten Features). Warum ist diese Metrik von solch großer Bedeutung? Weil sie das Team in die richtige Richtung führt, nämlich kontinuierlich neue Features (die für den Kunden wichtigsten zuerst) vollständig und möglichst reif für den Einsatz auszuliefern. Diese Metrik führt im Normalfall dazu, dass nur fertige Software [2] geliefert wird. Zudem beeinflusst sie den Entwicklungsprozess dahingehend, dass das Team in jeder Iteration versuchen wird, dieses Ziel zu erreichen.

Der zweite Nutzen dieser wichtigsten Metrik ist die Transparenz für den Kunden. Der Kunde weiß, dass er nur vollständig implementierte, getestete und seinen

Der dornige Weg zum agilen Team – die Serie

- Warum agile Teams so schwer zu schaffen und am Leben zu erhalten sind
- Was ist kein agiles Team?
- Warum müssen Builds immer brechen?
- Selbstorganisation – Chaos oder Wunderwaffe?
- Was soll heißen: „Ich bin fertig“?
- **Metriken: Ballast oder Notwendigkeit?**
- Abnahmen: Verhinderer des Projekterfolges?
- Brauchen agile Teams agile Kunden?
- Gedeihen agile Teams nur in agilen Unternehmen?
- Agiles Vorgehen als rotes Tuch
- Nur die Besten sind für das agile Team gut genug
- Cross-funktionale Teams – eine Illusion?
- Wo bleibt der Manager im agilen Team?
- Agile Teams skalieren nicht!
- Wie erhalte ich ein agiles Team am Leben?

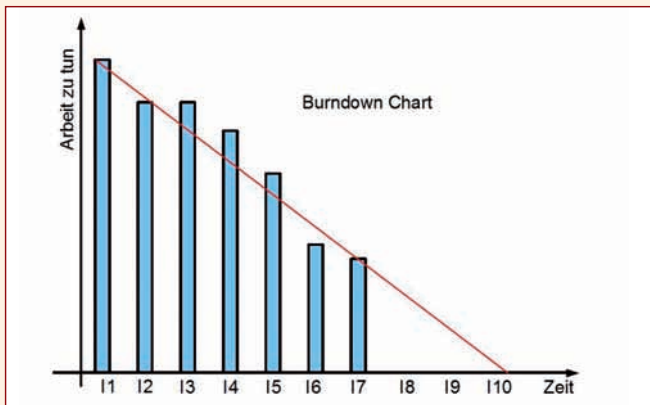


Abb. 1: Burndown Chart

Wünschen entsprechende Software erhält, und kann deshalb auch besser abschätzen, in welchem Zustand sich das Projekt befindet bzw. wie lange das Projekt noch zur Fertigstellung braucht.

Die Abbildung 1 zeigt eine für Scrum typische, so genannte Burndown Chart. Darin wird auf der Zeitachse (pro Iteration) der noch offene Umfang an Produkte-Features aufgereiht. Im Idealfall nehmen die Balken kontinuierlich ab. Dadurch kann leicht extrapoliert ein geschätztes Fertigstellungsdatum abgeleitet werden.

Aber auch Trends können (besonders bei der Kommunikation mit dem Kunden) sehr wichtig sein. Abbildung 2 zeigt eine Trendlinie, die in einem Projekt mit fix definiertem Preis (Ressourcen) und einem ebenfalls fix definierten Fertigstellungsdatum und Umfang eingesetzt wurde. Das Diagramm stellt pro Iteration den voraussichtlichen Fertigstellungstermin dar, abhängig vom jeweils noch offenen Produkte-Backlog (Umfang der noch offenen User-Stories, Anforder-

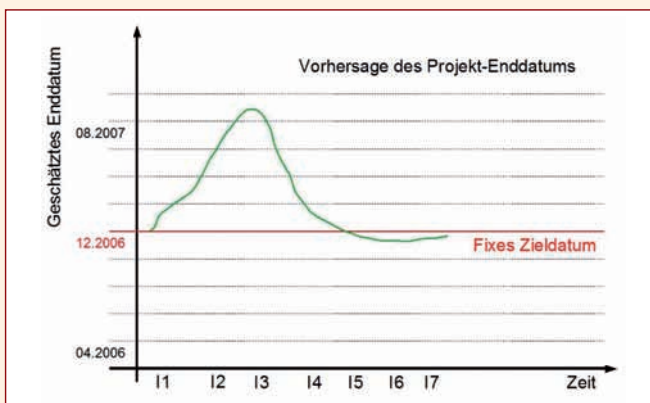


Abb. 2: Fertigstellungs-Trendlinie

rungen etc.) und der Geschwindigkeit des Gesamtteams. Es ist leicht vorzustellen, dass der Kunde, als er bei Iteration 2 und 3 die schlechten Prognosen sah, sich viel eher darum bemühte, unwichtige Dinge nach hinten zu priorisieren, und deshalb erheblich dazu beitrug, die Trendlinie wieder auf das Zieldatum hin zu drücken.

Neben dieser zentralen Metrik benötigt ein Team oft noch diverse diagnostische Metriken, die helfen, den gelebten Prozess immer wieder zu überprüfen und das eigene Verhalten anzupassen [3]. Solche Metriken sind in der Regel kurzlebiger, d.h., sie können abgeschafft werden, sobald sie nicht mehr gebraucht werden, entweder weil sich das Team in dem Bereich, der von der Metrik beleuchtet wird, massiv verbessert hat oder weil kein Fortschritt (mehr) zu erwarten ist.

Ungeeignet für agile Teams sind Metriken, die die reine Arbeitsleistung messen. Warum? Da diese Metriken keine Aussage darüber zulassen, ob am Richtigen gearbeitet wurde (was für den Kunden am meisten Wert hat). Sie fördern auch nicht das sehr wichtige Priorisieren der noch offenen Arbeiten seitens des Kunden. Außerdem kann im Team die Tendenz aufkommen, einfach möglichst viel Arbeit zu leisten, nur um die Metrik

zu befriedigen, statt auf den Wert der geleisteten Arbeit zu achten.

Es war einmal eine sinnvolle Metrik ...

Wie aber kann man erkennen, wann eine Metrik sinnvoll ist und wann man besser nichts misst? Die Liste im Kasten „Eigenschaften sinnvoller Metriken“ soll hierzu ein paar Anhaltspunkte liefern [3] [4].

Es stellt sich in der Praxis heraus, dass wir nicht alles messen sollten, das messbar ist. Dies aus dem einfachen Grund, dass das Messen nicht Selbstzweck sein sollte. Sobald man eine Vollzeitstelle braucht, um Metriken und Statistiken auszuwerten und zu verarbeiten, dann stimmt etwas nicht mit der Verwendung von Metriken. Das Projekt profitiert nicht von besserem Informationsgehalt, sondern leidet in der Regel nur unter mehr Datenfülle – und es wird schwieriger, den wahren Fortschritt im Projekt sichtbar zu machen.

Messen wir lieber weniger und sind dabei sicher, dass die Metriken von allen verstanden und auch verwendet werden. Denn Metriken sollten immer nur ein Mittel zum Zweck darstellen. Unsere Maxime sollte sein, was Ken Schwaber einmal auf der Scrum-Mailing-Liste [5] etwas überspritzt, aber durchaus treffend formulierte: „Don’t measure, do!“

Anzeige

Eigenschaften sinnvoller agiler Metriken

(Eine nicht erschöpfende Liste:)

- Transparenz: Teams, Management, Kunde, alle verwenden gleiche Datenbasis, eventuell mit einfacher Verdichtung.
- Kennzahlen sind von allen einfach zu verstehen.
- Messen fertig gestellte Software, gewichtet nach Wertschöpfungskriterien (kein Papier, keine Konzepte) und nicht reine Arbeitsleistung.
- Dienen im Team als Feedback und beeinflussen es zu positivem Verhalten.
- Legen Risiken offen (z.B. Qualitätsprobleme).
- Fördern „genügende“ Qualität.
- Möglichst automatisiert erhoben (für geringen administrativen Aufwand).
- Trends werden visualisiert, nicht absolute Zahlen.
- Kriterien, wann eine Metrik nicht mehr verwendet werden soll, sind definiert.



Jiri Lundak ist mit mehr als 20 Jahren Erfahrung ein Urgestein der Softwareentwicklung. Er arbeitet gegenwärtig als Entwicklungsleiter für Löwenfels Partner (www.loewenfels.ch) in der Schweiz. Er entwickelt als Architekt immer noch aktiv Software, hat aber erkannt, dass die meisten Probleme in Entwicklungsprojekten nicht technischer, sondern vielmehr menschlich-sozialer Natur sind. Als praktizierender Certified ScrumMaster lebt er die Rolle eines Vorkämpfers für agiles Verhalten innerhalb der Firmenorganisation. Kontakt: jiri.lundak@loewenfels.ch.

■ Links & Literatur

- [1] Frederick Taylor: The Principles of Scientific Management (1911): melbecon.unimelb.edu.au/het/taylor/sciman.htm
- [2] Jiri Lundak: Was soll heißen: „Ich bin fertig.“?, in *Java Magazin* 11.2006: www.bersteigconsulting.com/AppropriateAgileMeasurement.pdf
- [3] Deborah Hartmann, Robin Dymond: Appropriate Agile Measurement: Using Metrics and Diagnostics to Deliver Business Value: www.bersteigconsulting.com/AppropriateAgileMeasurement.pdf
- [4] Anna S. W. Allison: Meaningful Metrics: Conveying the Context of Your Numbers, STQE Magazine, Januar/Februar 2002
- [5] Ken Schwaber: Diskussion (11.08.2006): „Key Metrics for Agile Methods ...“ – in Forum: scrumdevelopment@yahoo.com