ENTAGO

Schätzungen in agilen Projekten der Software-Entwicklung

Der vorliegende Artikel beschäftigt sich mit Aufwandsabschätzungen bei agilen Projekten der Software-Entwicklung. Es wird gezeigt, wie traditionell Schätzungen auf zeitlicher Basis durchgeführt wurden und welche abstrakte Alternative es dafür bei agilen Projekten gibt. Daraufhin wird eruiert, wie absolute und abstrakte Schätzungen korrelieren und wie Aufwand und Zeithorizont bei agilen Projekten für eine Planung berechnet werden können.

Zeitbasierte Schätzung

Die nachfolgend beschriebene Herausforderung mit zeitbasierten Schätzungen wird jeder kennen, der schon mal ein Projekt in der IT oder Software-Entwicklung durchgeführt hat:

Wenn beispielsweise für ein Arbeitspaket drei Tage geschätzt wurde und die Entwicklung innerhalb der drei Tage abgeschlossen wurde, ist damit eine grundlegende Erwartung erfüllt worden. Falls die Entwicklung aber länger dauert, kann es mit der Projektleitung zu einer Konfrontation kommen.

Als Entwickler kommt man in eine Situation, in welcher man sich rechtfertigen muss – auch wenn die Organisation versucht, das «positiv» zu handhaben. Wiederholt sich diese unangenehme Erfahrung, schätzt man beim nächsten Mal einfach «etwas mehr». Im besten Fall ist man schneller, im Fall von unvorhergesehenen Ereignissen ist man immer noch im Zeitplan, da direkt mehr geschätzt wurde.

Falls es zu solch einer Situation kommt, kann man keinem der Involvierten wirklich einen Vorwurf machen. Die Projektleitung will den versprochenen Zeitplan für das Projekt einhalten, der Entwickler will die versprochene Schätzung einhalten.

Dabei muss man sich allerdings fragen, wie gut und hilfreich solche verzerrten Schätzungen sind. In der beschriebenen Konstellation hat es für einen Entwickler keinen echten Nutzen, eine Schätzung abzugeben. Läuft es gut, wird die Schätzung eingehalten, läuft es schlecht, wird die Schätzung gegen ihn verwendet.

Diminishing returns

Vom Gefühl her könnte man sagen, dass Schätzungen umso besser werden, desto mehr man sich inhaltlich mit der Schätzung befasst. Allerdings stimmt diese Hypothese nur bis zu einem bestimmten Punkt.

Bei Schätzungen in der Software-Entwicklung wird oft das sogenannte «Law of diminishing returns» genannt. Umso mehr Zeit in die Schätzung fliesst, desto langsamer wird die Schätzung genauer. Darüber hinaus wird eine Schätzung selten 100% genau sein – eine Schätzung bleibt eine Schätzung und involviert per Definition einen bestimmten Grad an Ungewissheit. Ab einem bestimmten Punkt kann sogar zu viel Zeit in Schätzungen gesteckt werden, was zu weniger akkuraten Ergebnissen führen kann.

Abbildung 1 zeigt in Anlehnung an [1] diesen Zusammenhang. Erfolgreiche agile Teams befinden sich eher auf der linken Seite der Kurve. Kleine Aufwände bei Schätzungen liefern verhältnismässig gute Ergebnisse.

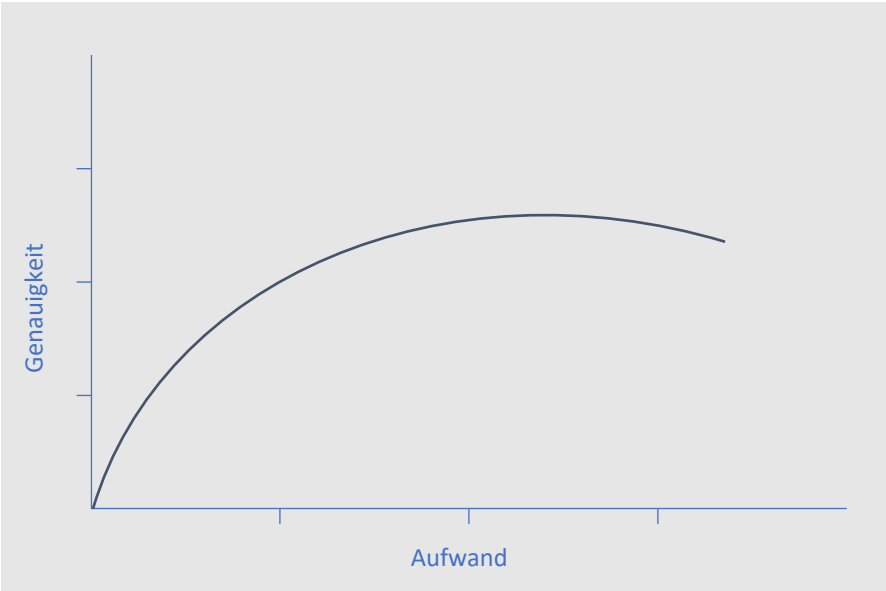

Abbildung 1: Effizienz Aufwand/Genauigkeit [1]

Vorgehensweise Schätzungen

Studien zeigen, dass Menschen gut darin sind, in Grössenordnungen zu schätzen [2] [3]. Stellen Sie sich beispielsweise vor, verschiedene Gegenstände auf Ihrem Tisch vom Gewicht her untereinander vergleichbar zu machen. Wir sind dazu in der Lage, relativ genau zu sagen, dass die Tastatur doppelt so schwer ist wie die Maus und die Tastatur ungefähr so schwer wie das volle Glas Wasser ist.

Geht es allerdings darum, das genaue Gewicht der Gegenstände in Gramm zu erraten, sind diese Schätzungen deutlich ungenauer. Agile Vorgehensweise macht sich die Stärke des relativen Vergleichs zunutze. Statt Arbeitspakete absolut zu schätzen, werden diese stattdessen untereinander verglichen.

Etabliert in der Software-Entwicklung haben sich sogenannte «T-Shirt-Sizes» (S, M, L, XL oder auch 1, 2, 4, 8) oder die Fibonacci-Zahlenreihe (1, 2, 3, 5, 8), um Arbeitspakete miteinander vergleichen zu können.

Abstrakte Schätzungen

Mit der beschriebenen Vorgehensweise können Teams Arbeitspakete untereinander vergleichbar machen. Die im vorangehenden Absatz genannten Zahlen haben allerdings noch keine Einheit. Möglichkeiten für eine Einheit wäre ein Zeitmass oder ein abstraktes Mass wie die sogenannten «Story Points».

Wenn Zeiteinheiten verwendet werden, hätte man zwar eine Vergleichbarkeit zwischen den Arbeitspaketen, die eingangs beschriebene Problematik besteht aber weiterhin.

Bei Schätzungen auf Zeiteinheit gibt es einen weiteren Nachteil: Ein Tag Arbeit für eine Person ist nicht unbedingt der gleiche Tag Arbeit für eine andere Person. Das könnte man mit zwei Freunden vergleichen, die gerne wandern gehen, von denen einer deutlich schneller als der andere ist: Wanderer A wird sagen «Das ist eine 3-Stunden Tour», während Wanderer B sagt «Das ist ganz sicher eine 5-Stunden Tour».

Worauf sich die beiden aber trotz unterschiedlicher Fertigkeiten eher einigen können, ist der relative Umfang: «Wanderung A ist doppelt so lang wie Wanderung B».

Genauso funktionieren Schätzungen mit abstrakter Einheit wie Story Points: Ein Team wird sich darauf einigen, dass Arbeitspaket B den gleichen Umfang wie Arbeitspaket A hat, welches mit 5 Story Points bewertet wurde.

Um eine «richtige» Basis zu haben, etablieren erfolgreiche Teams sogenannte «Muster-Arbeitspakete», welche 1, 2, 3 usw. Story Points umfassen. Somit gibt es immer einen Orientierungspunkt für die Vergleichbarkeit.

Kopplung an Zeiteinheit?

Angenommen, die Arbeitspakete werden nun in Story Points geschätzt und das Entwicklungsteam führt so die Arbeit durch. Für ein Management könnte nun die Ambition bestehen, einen direkten Zusammenhang zwischen Stunden und Story Points herzustellen.

Allerdings sollte von solch einer Vorgehensweise abgesehen werden. Durchschnittswerte lassen sich auf jeden Fall berechnen, doch wird die Varianz höher sein als initial gedacht.

In Abbildung 2 wird das dargestellt. Die Mittelwerte zeigen, dass ein Arbeitspaket, welches mit zwei Story Points bewertet wurde, doppelt so lange dauert wie eines, das mit einem Story Point bewertet wurde. Allerdings ist es auch möglich, dass ein Arbeitspaket mit einem Story Point länger dauert als eins mit zwei.

Für einzelne Arbeitspakete ist eine Kopplung von Story Points zu Zeiteinheit sehr ungenau. Diese Kopplung wird im Schnitt für mehrere Arbeitspakete besser und kann somit ein Werkzeug zur Planung und Kostenschätzung sein. Voraussetzung dafür ist, dass Zeitbuchungen auf Arbeitspakete konsequent stattfinden und diese Daten retrospektiv ausgewertet werden.

Fazit

Ein Wandel weg von zeitbasierten, absoluten Schätzungen hin zu abstrakten, relativen Schätzungen kann in der Softwareentwicklung einen Mehrwert bieten und nutzt unsere Stärke aus, Vergleichbares gut schätzen zu können.

Ein Nachteil, der an dieser Stelle genannt werden sollte: Erfahrungswerte müssen zunächst aufgebaut werden. Falls abstrakte, relative Schätzungen verwendet werden, ist es zu Beginn eines Projekts kaum möglich, Aussagen über die Zukunft zu treffen.

Fragen, wie «Wie viele Story Points kann das Entwicklungsteam pro Iteration liefern» kann erst nach Durchführung einiger Iterationen beantwortet werden. Dadurch ergibt sich aber auch der grosse Vorteil: Diese Werte werden mit Abschluss einer jeden neuen Iteration immer besser.

Dadurch lassen sich dann ebenfalls genaue Planwerte und Kosten bestimmen: Angenommen, ein Entwicklungsteam hat bereits einige Iterationen durchgeführt, wobei eine Iteration zwei Wochen umfasst. Pro Iteration schafft das Team durchschnittlich 20 Story Points, die Varianz wird bei den letzten Iterationen immer kleiner, das Team ist also immer bei 20 Story Points oder relativ nahe dran.

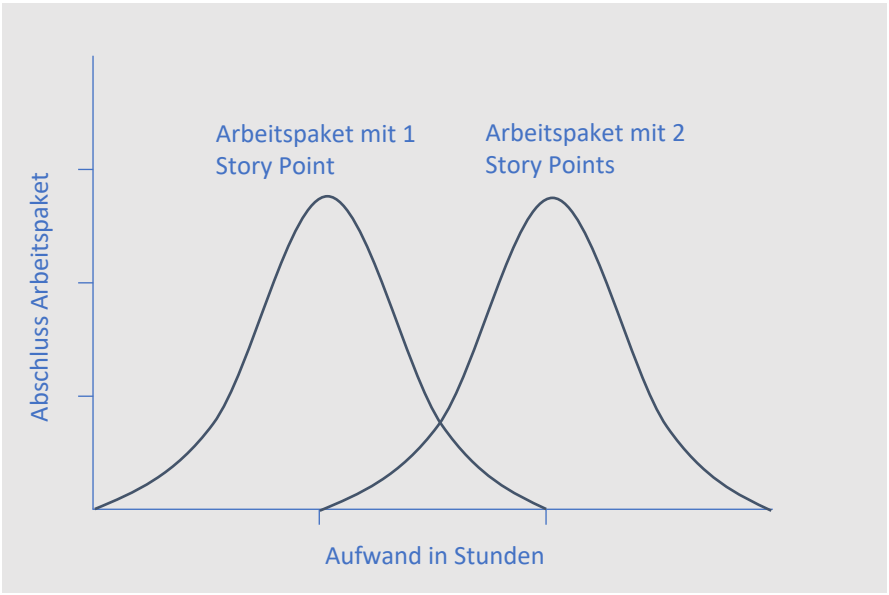

Abbildung 2: Distribution Aufwand für Arbeitspakete [1]

Nun soll eine neue Funktionalität entwickelt werden, welche insgesamt 100 Story Points umfasst, aufgeteilt in mehrere Arbeitspakete (ein Arbeitspaket hat maximal acht Story Points).

Da bekannt ist, wie viel Story Points das Team pro Iteration schafft, kann darauf basierend auch genau der Zeithorizont für die neue Funktionalität berechnet werden.

Einzelne Arbeitspakete können nur schlecht zeitlich abgeschätzt werden, bei Zusammenfassungen über mehrere Arbeitspakete wird diese Schätzung durch die Durchschnittswerte aber immer genauer (Abbildung 2).

Falls Sie auch gerade bei der Einführung von agilen Vorgehensweisen in der Software-Entwicklung sind oder mehr über die Transition von zeitbasierten, absoluten Schätzungen hin zu abstrakten, relativen Schätzungen erfahren möchten, beraten wir Sie gerne. Kontaktieren Sie uns unverbindlich.

Literatur

[1] Cohn, Mike: «Agile Estimating and Planning»; Prentice Hall, 2006

[2] Miranda, Eduardo: «Improving Subjective Estimates Using Paired Comparions»; IEEE Software, 2001

[3] Saaty, Thomas: «Multicriteria Decision Making: The Analytic Hierarchy Process»; RWS Publications, 1990

Moritz Eberhard
Senior Software Engineer

Entago AG
Buckhauserstrasse 34
CH-8048 Zurich

info@entago.ch

www.entago.ch