

"No Estimates" - Problematiken von Schätzungen

Vom Fragen wird die Straße länger.

~ Bernhard Schloß, 2013

Als "Gegenentwurf" zur klassischen Pflicht zur Schätzung von Aufwänden gilt die #noEstimates-Bewegung. Im folgenden sind einige Argumente, Alternativen und Hintergründe kurz angerissen.

Im Vordergrund der Kritik an Schätzungen steht dabei insbesondere, dass der Grund für Schätzungen in der Hauptsache externe Kräfte (Management, Kunde,...) sind, und diese meist geliefert werden, weil es "üblich ist", und nicht weil das Projektteam darin einen intrinsischen Sinn sieht. Der Gegenentwurf "NoEstimates" sieht eher eine Ableitung/Extrapolation von vorhandenen Informationen (z.B. bekannte Durchlaufzeit in einem Kanban-Fluß) vor, als immer bessere Methoden zu entwickeln eine nicht vorhersehbare Zukunft vorauszusagen. Darüber hinaus stellt sich ein solches System deutlich besser auf Veränderungen im Team oder der Organisation ein, da es nicht möglich ist, diese in Voraussagen zu berücksichtigen.

- Vortrag zu NoEstimates von Robert Weißgraeber: [Why estimates are bad for your software & team & management](#)

Machen Schätzungen überhaupt Sinn?

These: Schätzungen in einem Projekt sind unsinnig, da ein Projekt (oft oder immer?) die Erstellung von etwas "Neuem" beinhaltet. Das Neue ist nicht mit bereits existierenden Erfahrungen ausreichend vergleichbar/erfassbar. Daher wird jede Schätzung falsch sein. In IT-Projekten entstehen in eher kurzen Zyklen neue technische Möglichkeiten, um Projekte umzusetzen. Die hier fehlenden Erfahrungen führen zwangsläufig zu falschen Schätzungen.

Machen Schätzungen dann überhaupt noch Sinn? Wenn ja, dann vielleicht nur in grober/abstrahierter Form?

- "die neue Newsletter Applikation wird in weniger als 1 Jahr fertiggestellt sein"
- "die Integration des Payment Moduls wird ca. 3 Monate dauern"

Ansatzpunkte für die Kritik an Schätzungen

Schätzung von Dauer

z.B. Personentage: schon Scrum verzichtet auf Zeit-Schätzungen, da eine Dauer einer Aufgabe von der Produktivität, und damit mehr von der umsetzenden Person oder deren Zustand zum Umsetzungszeitpunkt oder externen Einflußgrößen abhängt. (Literatur: <http://scrum.jeffsutherland.com/2013/04/why-hours-dont-work.html>)

"Parkinson'sche Gesetz"

"Work expands so as to fill the time available for its completion."

„Arbeit dehnt sich in genau dem Maß aus, wie Zeit für ihre Erledigung zur Verfügung steht.“

Abgeleitet heißt das: Schätzen wir eine Aufgabe auf eine Dauer X, wird sie nie in weniger als dieser Zeit fertiggestellt.

fehlende Wertschöpfung (Muda)

Die Schätzung an sich, als Tätigkeit, gibt keinen Beitrag zur Wertschöpfung des Produktes, oder auch provokativ formuliert: "Durch Schätzung wird die Arbeit nicht früher fertig."

Systemische-Problematiken

Probleme ergeben sich auch aus der Interaktion zwischen Schätzenden und Auftraggebern. Diskussionen über die Schätzung ("das ist zu viel") führen zur Einführung von verdeckten Puffern in die Schätzung. (Literatur: <http://www.estherderby.com/2012/03/estimating-is-often-helpful-estimates-are-often-not.html>)

Wissenschaftliche Studien

Studienergebnis zur Produktivitätsmessung ohne Schätzung: (Quelle siehe: http://books.google.de/books?id=TVQUAAAAQBAJ&pg=PT19&lpg=PT19&dq=%22table+5-3.+Productivity+by+Estimation+Approach%22&source=bl&ots=OYO9fGteYU&sig=_kX02jh7bgAFmj1TQu2B9knSF6c&hl=de&sa=X&ei=IMQ9UuqsCYSXtAbPpIGwDg&ved=0CDIQ6AEwAA#v=onepage&q=%22table%205-3.%20Productivity%20by%20Estimation%20Approach%22&f=false)

The most surprising part of the 1985 Jeffery-Lawrence study appeared at the very end, when they investigated the productivity of 24 projects for which no estimates were prepared at all. These projects far outperformed all the others (see Table 5-3).

Effort Estimate Prepared by	Average Productivity	Number of Projects
Programmer alone	8.0	19
Supervisor alone	6.6	23
Programmer & supervisor	7.8	16
Systems analyst	9.5	21
(No estimate)	12.0	24

Table 5-3. Productivity by Estimation Approach (Full Result)

Projects on which the boss applied no schedule pressure whatsoever ("Just wake me up when you're done.") had the highest productivity of all. Of course, none of this proves that Parkinson's Law doesn't apply to development workers. But doesn't it make you wonder?

Zwischenlösungen

Einige Frameworks, besonders im Agilen Bereich, beantworten diese Defizite mit abweichenden Schätzungs-Methoden.

Relative Schätzung (z.B. in Gummipunkten)

Die Ermittlung von relativen Größen (oft verwendet z.B. Story Points, T-Shirt-Größen) der einzelnen Aufgaben innerhalb eines Projektes wird dazu verwendet, zumindest fiktive Zahlen an die Aufgaben anhängen zu können. Über Auswertungen der erreichten Punkte werden Größen wie eine relative Projektgeschwindigkeit (Velocity) oder projizierte Endtermine mittels Burndown Chart ermittelt.

Planning Poker

Beim Planning Poker steht der Prozeß der Schätzung im Vordergrund. Durch Spiegelung der unterschiedlichen extremen Meinungen im Expertenteam und der daraus folgenden Diskussion wird ein gezielter Austausch über die Aufgabe bzw. deren Ansätze zur Umsetzung gefördert. Insbesondere getroffenen Annahmen können dabei aufgedeckt werden.

Function Points

Eine mittlerweile schon "alte" Variante zur Aufwandsermittlung in der Softwareentwicklung: siehe Wikipedia: <http://de.wikipedia.org/wiki/Function-Point-Verfahren>

Expertenschätzung

Ähnlich wie beim Planning Poker tauschen sich die Experten auf dem Gebiet aus und geben jeweils ihre Schätzung ab, beruhend auf Erfahrungen aus früheren Projekten.

Funktioniert allerdings nur dann gut, wenn die Experten genügend Erfahrung haben und über eine Datenbasis mit den Aufwendungen für vergleichbare, in der Vergangenheit abgewickelte Projekte verfügen.

Gegenvorschläge

Nichtsdestotrotz sollte man in der Lage sein, den berechtigten Fragestellungen vom Typ "Bis wann ist XY fertig" qualifiziert zu beantworten. Dazu wird zur Messung von Fortschritt, etc. u.a. Folgendes vorgeschlagen:

- Messung & Plot von aktueller Durchlaufzeit
- Messung von Durchsatz (Anzahl der gelieferten Tickets, Stories etc. pro Zeitraum)
- -> Bei Nutzung von WIP-Limit lässt sich mittels Warteschlangentheorie (z.B. Littles Gesetz) eine Voraussage für ganze Teams treffen

Der grundsätzliche Ansatz seitens des Managements/Kunde ist einer von Vertrauen & Beobachtung mit Regulierungsmöglichkeit, statt der einer Kontrolle von fiktiven Zahlen.

Stabile Teams ermöglichen verlässliche Schätzungen

Teams, die über einen längeren Zeitraum mehrere Projekte gemeinsam bearbeiten sind in der Lage neue Aufgaben gemeinsam zu bewerten. Durch ein funktionierendes Team werden eventuelle Fehleinschätzungen auf mehreren Wegen ausgeglichen. Ein stabiles (Kern-)Team ist somit eine Voraussetzung für eine möglichst geringe Falscheinschätzung, unabhängig von der gewählten Methodik.

Informationen zu #noEstimates

- <http://xprogramming.com/articles/the-noestimates-movement/>
- <http://mobprogramming.org/how-does-the-mob-get-away-with-no-estimates/>
- <http://neilkillick.com/2013/02/12/noestimates-part-3-the-palm-off/>
- <http://softwaredevelopmenttoday.blogspot.de/2013/09/the-noestimates-questions-part-i.html>
- <http://softwaredevelopmenttoday.blogspot.de/2013/09/the-noestimates-questions-part-ii.html>
- <http://plan.io/blog/post/145902333978/noestimates-6-software-experts-give-their-view>

Entstanden aus dieser Twitter-Diskussion: <https://twitter.com/fogbird/statuses/326636695946661888>