

To Estimate or #NoEstimates, that is the Question

Todd E. Little, *Member, IEEE*, Chris Verhoef, *#NoMember, IEEE*

Abstract—We analyzed project data from 55 projects claiming to use agile methods to investigate the predictive value of story point estimation and velocity for project forecasts. The data came from nine organizations ranging from startups to large multinational enterprises. We found that projections based on throughput (story counts) were essentially identical to that of using velocity (story points). Neither velocity nor throughput were great predictors as the P90/P10 ratios of the projections was about 3.5. Through the use of a simulation model we replicated our findings which aid in understanding the boundary conditions for when story point estimates may be better predictors.

Index Terms—Software Estimation, Agile Development, Project Management

I. INTRODUCTION

TO Estimate, or #NoEstimates, that is the question:
*Whether 'tis nobler in the mind to suffer
 The Slings and Arrows of outrageous errors,
 Or to take Arms against a Sea of troubles,
 And by opposing, end them? #NoEstimates, so easy;
 No more; and by #NoEstimates, focus on value and end
 The headaches and the thousands of natural unknowns
 That estimation is prone to, 'tis a consumption
 Devouring all resources. #NoEstimates; so easy;
 So easy; perchance it seems: ay, there's the rub;
 For in #NoEstimates what surprises may come,
 When we develop in earnest and begin our toil.
 Must give us pause. There's the respect
 That makes calamity of software's life;*

Since the early days of software development, teams have struggled with estimation challenges. Many have searched for processes or tools to make us better estimators, to limited avail. Recently, the #NoEstimates movement in the twitterverse has challenged some of these fundamental tenets. As defined by originator Woody Zuill: “#NoEstimates is a hashtag for the topic of exploring alternatives to estimates [of time, effort, cost] for making decisions in software development. That is, ways to make decisions with ‘No Estimates’.”[1]

We analyzed data [2] from 55 projects claiming to use agile methods to explore those decisions and what value estimates may add. The data came from 9 different organizations ranging from startups to large multinationals. To the best of

our knowledge, none of the data came from organizations that were studied by the authors in [4,5]. 37 projects came from one organization, but analysis using only the remaining projects provided similar results.

We built a simulation model of the agile estimation and delivery process in which we could replicate the findings of the data and thus explore some of the parameters that might impact the usefulness of the estimates.

II. CORE FINDINGS

A standard approach for agile/Scrum teams is to use story points and velocity [3] to track progress. Story points are an estimate of the relative effort required for a given user story and are only meaningful within a particular team. Velocity is defined as story points completed per time unit or iteration. Just as with a car we have instantaneous velocity of the iteration, and the average velocity across multiple iterations. Assuming that average velocity converges to a constant, the team can extrapolate the time required to complete the remaining stories in the backlog, or to determine the approximate number of story points that can be completed in a fixed time. One approach often suggested by #NoEstimates advocates is to discontinue estimating story points and instead simply count the number of stories completed per iteration, also called throughput.

Our analysis of the data showed that for the purpose of tracking progress and projecting into the future, there was no significant advantage to using story point estimates and velocity over tracking throughput. We also replicated this in simulations.

By altering the simulation conditions, we explored when story point velocity is a better predictor than throughput. The key parameters we altered were:

- Distribution of story size
- The team's estimation accuracy
- The bucketing approach used (e.g none, modified Fibonacci, power of 2, etc)

It turned out that story point estimation adds some value when there is large variation in story size. Bucketing had very little impact although larger buckets such as powers of 4 slightly eroded the added value of velocity.

Neither velocity nor throughput correctly accounted for what we believe to be a hardening period. Approximately 50% of the projects reported a 2-12% timespan at the end of the project with no stories delivered. In five projects we observed zero velocity from 30%-50% of the overall schedule. These teams were likely #NotAgile or following some form of WaterScrumFall and were excluded from further analysis.

The variability across all the projects and all the iterations of the instantaneous velocity provides a measure of estimation accuracy. The P90/P10 ratio is commonly used to report the range of distributions, where P90 is the 90th percentile and P10 is the 10th percentile. The data had a P90/P10 ratio of 4.5 and exhibited characteristics of a failure distribution (such as Weibull, gamma, lognormal, etc). This large of a distribution ratio indicates that velocity is not very consistent from iteration to iteration. We also discovered that this range did not improve over time, consistent with other findings about the cone of uncertainty [4,5,6].

The distribution of normalized story points had a P90/P10 ratio of 2.9 which again had characteristics of a failure distribution. Although this was the distribution from the data, if a team wanted to narrow their distribution they could easily use story splitting techniques to obtain similar story sizes.

While our findings support that project tracking is not significantly improved by using story point estimates, there may still be times where estimates are valuable or necessary. We looked at many of the business decisions that are encountered in software projects to examine the value of estimation. Practitioners should understand what decisions they are making and the implications that estimating or not estimating might have on those decisions. Our data and the simulations provide a valuable context for those discussions.

III. ESTIMATION IN AGILE PROJECTS

Estimation in software development can mean many things. Teams estimate cost, effort, schedule or value at varying degrees of detail and time horizons. Agile teams generally work with some form of user stories and/or tasks. When teams use estimates, they frequently work with estimates at the levels of granularity listed in Table 1.

Release	A release is a collection of features/stories that completes a delivery of interest to the customer. Estimates are typically in the form of effort, time and cost.
Feature or Epic	A high level story covering a broad area. It will generally be broken up into smaller stories. Estimates are typically representative of effort, and as story points, T-shirt sizes, or other.
Story	A unit of deliverable value to the customer. Stories will generally adhere to the INVEST [7] principles (Independent, Negotiable, Valuable, Estimable, Small, Testable). Typically estimates are in story points.

Task	Specific activities that the delivery team will do in order to complete a story. Typically estimated in hours.
------	--

Table 1

Our data are at the story level (so stories and story points), likewise our simulations were at that level for comparison. In Section VII we suggest what our findings at the story level may imply regarding the value of estimates at other levels based on the decisions that the team is trying to address.

IV. DECISIONS

One of the primary reasons for estimating is to improve decision-making. We list some of the key decisions that project teams, sponsors and customers typically make during a software project.

- Decisions at project sanction
 - Is it worth doing?
 - What are the priorities?
 - When is the target time to ship?
 - What is the critical scope?
 - Do we have the right investment?
- Decisions to steer towards the release
 - Is it worth continuing?
 - Have priorities changed?
 - Are we on target to meet our commitments?
 - Do we have data to support our ability to meet our commitments?
 - What are the optimal scope/schedule tradeoffs?
 - Can we do anything to accelerate delivery?
 - What is the cost of delay?
- Decisions to help with managing iterations
 - Can we make our iteration commitment?
 - What is our capacity?

We will revisit the context of these decisions based on the findings from the data and the simulations.

V. ANALYSIS OF THE DATA

Each project reported a tabulation of story points and number of stories delivered for each (timed) iteration. We initially focused on the decision area of project tracking, specifically comparing the predictive power of velocity with that of throughput.

A. Project Burnup Charts

In order to compare projects we normalized time based on the overall time and likewise normalized story points and number of stories based on the totals for the release. With this normalization the burnup chart for each project starts at the origin and ends at the upper-right corner of the unit square. In theory, a perfect agile project would have a straight line from (0,0) to (1,1).

Figure 1 shows a plot of normalized story points and normalized stories versus normalized time for three of the projects and the aggregate curve of all the projects. This shows the trends for story points and stories to be nearly identical. As well we observe that many projects that trend towards (1,1) required additional iterations at zero velocity to finally be “complete” as depicted in the Aggregate curve.

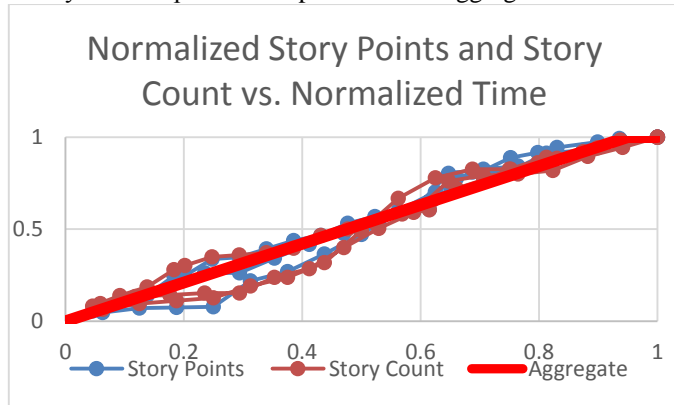


Figure 1

B. Velocity and Uncertainty Range over Time

For all projects, for each iteration we also calculated the normalized velocity. Figure 2 shows the P90 and P10 bands of velocity over time and Table 2 shows the numerical values.

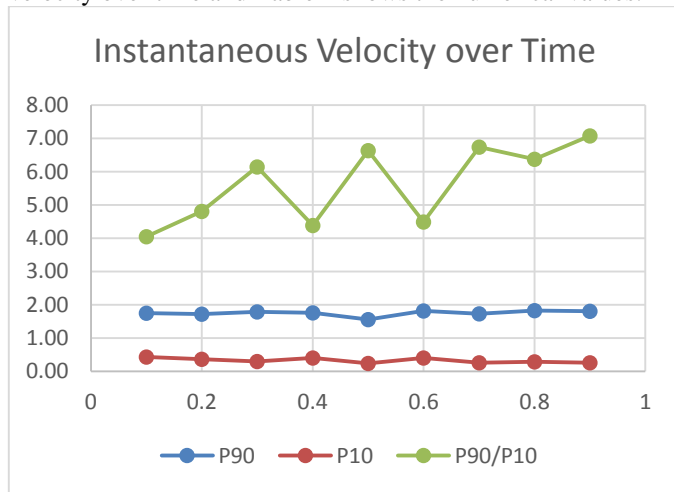


Figure 2

Normalized Time	P90	P10	P90/P10
0.1	1.75	0.43	4.06
0.2	1.72	0.36	4.81
0.3	1.79	0.29	6.14
0.4	1.76	0.40	4.39
0.5	1.56	0.24	6.62
0.6	1.82	0.41	4.49
0.7	1.73	0.26	6.73
0.8	1.83	0.29	6.38
0.9	1.81	0.26	7.07

Table 2

The range of the P90/P10 is between 4.06 and 7.07 with an overall P90/P10 of 5.1. We observe that the range is

moderately consistent and does not reduce over time, consistent with previous studies [4,5,6].

C. Project Projections Using Velocity

To determine the predictive power of using velocity versus using throughput, at each iteration we made projections based on the remaining work using the average velocity and throughput. We compare these projections against the known actuals to determine the relative errors. Figure 3 shows the results of the range of errors in the projections made as a function of normalized time.

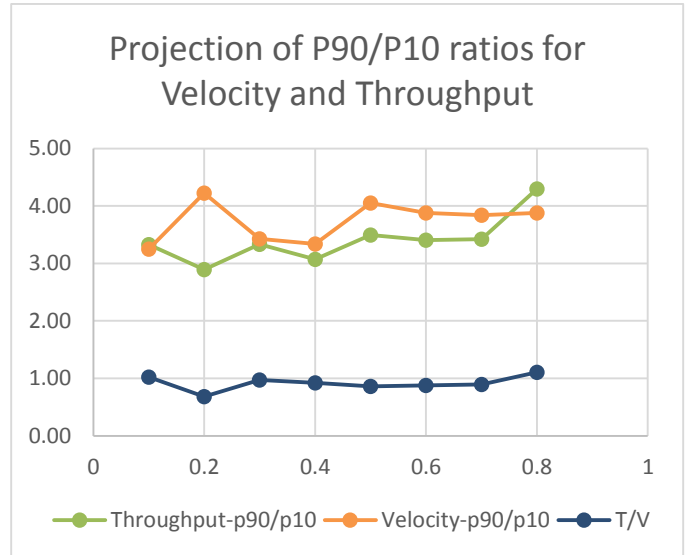


Figure 3

The ratio of the throughput to velocity P90/P10 ratios averages 0.94 indicating throughput to be a better predictor by 6%, but there is really no significant difference in the accuracy of the forecast projections. While both velocity and throughput projections are similar in their degree of accuracy, neither is a phenomenal predictor with a P90/P10 ratio of about 3.5 in both situations.

In practical terms, whether using velocity or throughput, if a team forecasts that they have about 6 months remaining, the P10 to P90 bands for 80% confidence are roughly 3.2 to 11.2 months. This does not bode well for teams or stakeholders that are expecting estimates that are commitments or even within 25% accurate as suggested by [8]. This implies a mismatch between the degree of accuracy expected and the reality of the range of uncertainty that is generally encountered. The uncertainty range from this data is consistent with other research [4,5,6].

D. Story Point Distributions

We did not have complete information for each story, but using the story points and number of stories delivered each iteration, we had an average number of story points per story per iteration which we used to establish a distribution. For the purpose of simulating we started with Weibull and lognormal distributions with a P90/P10 ratio of 3.0. Since we only had the average per iteration, the actual distribution range for individual stories might be larger. We used the range as a sensitivity parameter in the simulations.

VI. ANALYSIS USING SIMULATIONS

In order to better understand the findings from the data we simulated the results using a Monte Carlo technique.

A. Simulation Approach

For each scenario, we simulated 1000 projects each with 50 stories. For each story we sampled the story point distribution to establish its nominal story points. If a bucketing approach was used then we determined how it would be “bucketed.” This was used for tabulating story points. Using the original (pre-bucketed) story points, the actual time was derived from the estimation accuracy distribution. As with the analyzed data, we converted to normalized values for consistency across projects.

B. Simulation Parameters

The parameters that we explored are fourfold:

1) Story Point Distribution

We started with the empirical distribution from the data and then also explored failure distributions like Weibull, lognormal curve fits and variations on these distributions to see the impact that it would have on the simulations. Story point distribution is one area where individual teams have significant control over how they define and split their stories. Some teams may have large epics interspersed with smaller stories, while others may take the effort to split stories into relatively similar sizes.

2) Estimation Accuracy

Again, we started our simulations with the empirical distribution provided by the data itself. We also used curve fits for lognormal and Weibull. In further scenarios we varied the P90/P10 ratio to explore its sensitivity. The data also showed a correlation between Story Point Distribution and Estimation Accuracy which we used in our simulations.

3) Bucketing Approach

Agile teams often use some form of bucketing for story points. Planning Poker [9] is a popular approach using a modified Fibonacci series of {1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100}. Other approaches include powers of 2 or 4. We varied the bucketing to find the impact they might have on the overall predictive power.

4) Hardening Effort

Roughly 50% of the projects required 2-12% of the total time at the end with zero velocity. We included this in the simulation as well.

C. Analysis

Table 3 shows the analyses of the different story point distributions, accuracy distributions, and impact of bucketing. For each scenario we show the P90/P10 ratio of the projections using velocity and throughput respectively. We also show the percentage improvement in range reduction by using velocity. A negative improvement would indicate that throughput was better than velocity.

#	Story Point Distribution	p10/p90	Estimation Accuracy	p10/p90	Bucketing	p90/p10 T-put	p90/p10 Velocity	Velocity %Impr
1	Empirical	2.9	Empirical	4.5	None	4.04	3.63	11%
2	Lognorm	3.0	Lognorm	5.7	None	2.60	2.55	2%

3	Weibull	3.0	Weibull	5.2	None	3.34	3.33	0%
4	Lognorm	3.0	Weibull	5.2	None	3.37	3.29	2%
5	Lognorm	3.0	Weibull	2.5	None	1.86	1.86	0%
6	Lognorm	3.0	Weibull	6.0	None	3.87	3.76	3%
7	Lognorm	2.0	Weibull	5.2	None	3.18	3.22	-1%
8	Lognorm	6.0	Weibull	5.2	None	3.95	3.52	12%
9	Lognorm	3.0	Weibull	5.2	None	3.37	3.29	2%
10	Lognorm	3.0	Weibull	5.2	Fibonacci	3.37	3.38	0%
11	Lognorm	3.0	Weibull	5.2	2X	3.37	3.50	-4%
12	Lognorm	3.0	Weibull	5.2	3x	3.37	3.63	-7%
13	Lognorm	3.0	Weibull	5.2	4x	3.37	3.61	-7%

Table 3

The first three scenarios show the results of using various story point and accuracy estimation distributions to see how well we could match the data from this study. We started with the empirical curves and also tried lognormal and Weibull curve fits. We believe the curve fits smooth some of the noise in the empirical data resulting in a better match. Scenario 4 using a lognormal fit for story point distribution and a Weibull fit for estimation accuracy gave a good match with our observed data. The plot of the P90/P10 projection ratios are all near 3.5 as shown in Figure 4.

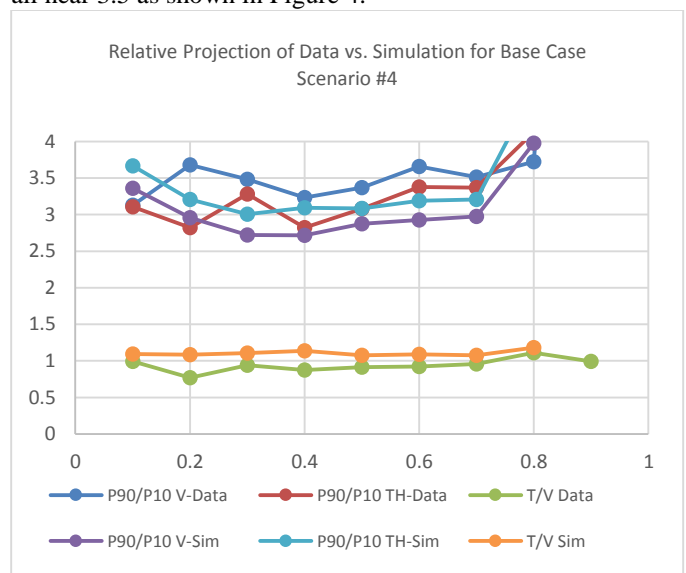


Figure 4

The simulations showed projections using velocity to be nearly identical to using throughput, similar to what we found in the data. This gave us confidence to proceed with some sensitivity analysis.

For scenarios #5 and #6 we modified the estimation accuracy distribution to P90/P10=2.5 and 6.0 respectively. A surprising finding is that improved estimation accuracy helps both velocity and throughput projections equally. If estimation accuracy range is narrower, then the corresponding distribution of actual story points is also narrower.

We then explored the range of the story point distributions varying the baseline lognormal distribution in scenarios #7 and #8. There is a clear relationship: the broad P90/P10 ratio of 6.0 gives much more benefit to velocity (11%), while the very narrow P90/P10 of 2.0 shows throughput to be slightly better than velocity (-1%). The narrower story distribution shows improved predictability over the broader distribution, due to the story distribution constraining the uncertainty.

Lastly scenarios 9-13 investigated the impact of bucketing: comparing no bucketing with Fibonacci, power of 2, power of 3 and power of 4 bucketing. The buckets do not impact throughput, but they do degrade the predictive power of velocity as buckets get large. The impact is not significant as the velocity advantage starts at 2% for no buckets and inverts to favor throughput (-7%) for large buckets.

D. Summary of Simulation Results

We were able to match the data quite well with our simulation which gave us confidence to explore scenarios. The advantage of using velocity is almost negligible except when story point distribution is large. This is good news as the team has a lot to say about how they split their stories so as to reduce the range of the story distribution.

VII. IMPLICATIONS ON DECISIONS

A key reason given for estimation is to make decisions. What do our findings suggest regarding some key decisions?

Decision	Role of Estimation
Decisions to steer towards the release	We observed from both the data and the simulations that story point estimates provide minimal improvement in forecasting compared to using throughput. The simulations showed that estimates may help when there is a large range of story distribution, although an alternative approach would be to split large stories so that the overall distribution is not large. When estimating a container of mixed nuts, we don't really care too much whether we have smaller peanuts or larger brazil nuts, but we do want to spot any coconuts!
Decisions to help with managing iterations	Many teams use detailed task estimation to help them manage their iterations. We did not have access to task estimations for this study, however the findings with story points should be very enlightening. Task estimation can often be a very time consuming activity. Teams should look at how much value they are getting from these estimations.
Decisions at project sanction	Some level of macro-estimation of costs and benefits is likely necessary for business decisions. If the benefits are so overwhelming that it should be done at any cost, then it could be wasteful to spend

	time on estimating something that does not impact the decision. In general it is waste to spend more time on cost estimation than on benefits. In fact, a study of a number of projects at a major organization found that value generated was negatively correlated to cost forecast accuracy [10]. Too much emphasis on cost or on reduction of uncertainty can destroy forecasting accuracy of value predictions.
--	--

Table 4

VIII. PRACTITIONER'S GUIDE

A. Velocity vs. Throughput

With the typically observed story point distribution range, our results show there is minimal added value to using velocity over using throughput for estimating purposes. When story size distribution is very large, then there is some improvement gained from using velocity.

B. Hardening

In about half the projects there was a period at the end of the project of between 2-12% of the overall timeline with zero velocity, most likely for release and hardening activities. Unless teams have reasons to believe that they will not require such activities, we recommend either allocating a corresponding time buffer or adding stories (and story points if used) for such activities.

C. Estimation Accuracy

This study provides additional confirmation that the range of uncertainty with software estimation accuracy is significant and we can confidently say that this range of uncertainty is much larger than many decision makers realize. An interesting finding was that improvements in estimation accuracy helped throughput projections just as much as velocity projections. So while improving estimation accuracy may be a noble goal it is not a reason to favor velocity over throughput.

D. Bucketing of Estimates

While there was some degradation of the predictive power of velocity as buckets get very large, the overall impact is still very small. Since bucketing approaches are used for expediting estimation processes this finding suggests that teams may continue to use them should they find value in estimating at all. However, we have seen situations where religious adherence to bucketing approaches slowed down the estimation process and in those circumstances teams may be better suited with simpler approaches. Bucketing or #NoBucketing? You decide.

E. Uncertainty over Time

Perhaps a bit more bad news for teams and decision makers is that it doesn't get better over time. The range of relative

uncertainty of the work left to be done is large and stays large over time, which is consistent with other findings [4,5,6].

F. Decisions

Decisions are being made at multiple levels. For some decisions there may be value to estimates of stories or story points. But those estimates most likely have very large uncertainty ranges. The important question for the team is to understand the decisions they care about, and to comprehend the range of uncertainty to make the appropriate decisions. Decision makers would be wise to learn more about making decisions under uncertainty. There is significant research in many other industries (e.g oil and gas exploration, financial institutions, actuaries, etc.)

G. Estimates or #NoEstimates

To paraphrase Polonius' advice to Laertes, "Neither an Estimator nor a #NoEstimation bigot be, for estimation oft implies a false sense of both accuracy and certainty, while NO estimates may make suboptimal decisions. To thine own self (and team) be true."

REFERENCES

- [1] <http://zuill.us/WoodyZuill/2013/05/17/the-noestimates-hashtag/>
- [2] <https://bitly.com/NoEstimatesProjectsDB>
- [3] <http://guide.agilealliance.org/guide/velocity.html>
- [4] T. E. Little, "Schedule Estimation and Uncertainty Surrounding the Cone of Uncertainty", IEEE Software, vol.23, no. 3, pp. 48-54, May/June 2006
- [5] J. L. Eveleens, C. Verhoef, "Quantifying IT forecast quality," *Science of Computer Programming*, vol. 74, no. 11-12, pp.934-988, Sept, 2009.
- [6] L. Cao, "Estimating Agile Software Project Effort: An Empirical Study", (2008). *AMCIS 2008 Proceedings*. Paper 401.
- [7] W. Wake, "INVEST in Good Stories, and SMART Tasks", <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
- [8] S.D. Conte, H.E. Dunsmore, V.Y. Shen, *Software Engineering Metrics and Models*, Menlo Park, CA: Benjamin/Cummings Pub. Co., 1986, pp
- [9] https://en.wikipedia.org/wiki/Planning_poker
- [10] J.L. Eveleens, M. van der Pas, C. Verhoef, "Quantifying forecast quality of IT business value", *Science of Computer Programming*, vol. 77, no. 3, pp. 314-354, Mar. 2012