

# Reflecting on Measurements in an Agile World

by Joe Schofield

Measurements have been a part of software development and application delivery for decades. Software providers—often project managers—are eager to predict how soon they can deliver, at what cost, and how richly featured the product will be. Different providers are quick to “outbid” each other on schedule, value, and of course, cost. Estimators are pressured to trim their numbers to make them more attractive. Ultimately, teams are given a “frozen” deadline and budget, and a constantly changing, often expanding, scope of work. This pattern has persisted as long as IT itself, and often continues unabated in our agile world. Here are the primary reasons why estimation and measurements continue to haunt estimators, developers, and IT consumers.

**In general, most of us just don’t like to be measured and compared to others.** *Measurements pervasively surround us:* unrelenting statistics in sports, vehicle mileage, total cost of ownership, steps on your wearable device, interest rates, tax rates, and queue lengths. *Measurements provide opportunity to invasively offend us:* calories burned, weight, scores in school, and rankings in the work place. Competition often suggests that one of us wins, while the rest of us lose (finish a close second?). Many of us are suspect of measurement motives, accuracy, analysis, confidentiality, and consequences.

**Measurement wariness translates to our teams and organizations.**

Organizations, teams, and individuals often unknowingly measure the wrong behaviors hoping for favorable outcomes. A small set of examples include:

Measuring individual productivity can reveal some startling and

unexpected results. **Individuals are notoriously optimistic when estimating.**<sup>3</sup> Under pressure to perform or compete, estimates have been demonstrated to get worse.<sup>4,5</sup> Biases and over-confidence cast further doubts on estimating skills in general.<sup>6</sup>

**Individual performance measures are not well understood** but that doesn’t discourage experts from predicting outcomes with some certainty. In one study, no single developer outperformed his/her peers on each of nine assigned programs written in C. Of the eight programmers, five had both a longest and shortest program across the nine programs in a relatively controlled environment.<sup>7</sup> The data for this study was originally peer reviewed and published in 2005 and it is still cited today by the National Institute on Standards and Technology (NIST)<sup>8</sup> as a reference to coding variation. Often selected team members are treated with a halo effect when it comes to meeting an urgent time-critical need. Data suggests that those considered experts for a given task may not be. Individual performance measures are based on individuals that may exhibit a wide variation in performance—even for apparently similar tasks.

Another factor that impacts individual performance is the number of projects to which a team is “committed” concurrently—the fragmented time challenge that eludes few organizations. Organizations often remain stuck in specialist and silo roles with little bench strength. Ongoing frequent

crises trigger—what I typically describe as the “911 methodology”—and much like a black hole, key resources just can’t escape the distractions. The gravitational pull of these black holes skew measurement and, likely, benchmark data.

Intent / Measure	Unintended Compromises
Increasing value delivery	Inflating the value of stories; increase in defects
Increasing velocity	Exaggerating the number of story points for stories <sup>9</sup> ; unexpected team burnout and churn over
Reducing the accumulation of technical debt	Increasing refactoring (lean disciples recognize refactoring as waste) thereby shrinking value delivery; as an alternative CAST suggests starting architecture early <sup>1</sup>
Reducing the time between releases	Releasing product increments with known but less significant defects
Improving quality by increasing defect removal efficiency (DRE)	Remediating defects that are easily found (while leaving the difficult defects latent); keep in mind that developing products with fewer defects may drive DRE in the wrong direction but improve the product’s reliability. A preferred objective is to prevent defects, not get better at finding them!
Lowering (or meeting) cost by minimizing labor	Neglecting to record time and overtime <sup>2</sup>

**Most benchmark data employed for estimating is irrelevant to current development practices, outdated for current platform development, and unrelated to team development practices.** Parametric models often rely on similar data that doesn't closely match the features or context of the work being estimated. Organizations that measure are likely to discover that the best data (I'm not suggesting it's good, it may be as horrendous as the data you buy) is your own recent data because it reflects your development environment, team capability, and culture. As such, your own data, afforded integrity in the collection and measurement process, may provide the most realistic data for both traditional and agile-based development.

(if you measure) the best data is often your own recent data

**Notable variation in agile practices makes agile measurement that much more challenging.** *"Please don't tell me you're doing 'agile'"* starts [Keep the Baby](#).<sup>9</sup> Available today are at least twelve so-deemed agile approaches. Limited conformance to even core scrum practices elicit the use of unproven and unmeasured hybrids, and scrumfall and water-gile-based projects. Being agile might very well encourage this behavior; however, 85 percent of surveyed organizations admitted to some failures with agile.<sup>10</sup> The old quality saying, "You can't manage what you can't measure" might be revised to include "and you can't measure what you can't define!"

**Widespread reports of teams "doing agile" without a culture of "being agile" predominate.** Culture is often the most cited impediment to agile success. Agile team measures are likely not reflected in classic quarterly and annual reports using ROI, KPIs, and PMO performance. The interests of top IT leadership are more likely directed at preventing data breaches, scaling in the cloud, aging infrastructure, and responding to the Board. Such worthy initiatives leave few cycles for ongoing involvement in the enterprise's transition to an agile mindset, or beyond with DevOps and Product Model initiatives. Velocity and burndown charts get filtered and recast into more traditional representations. The successes attributable to agile are lost in the translations as history, culture, and conformity in reporting to financial committees supplant the transparency and simplicity of agile measures. Agile successes can be measured; 23 such characteristics are reported in one survey alone.<sup>11</sup>

**Project Management Offices (PMOs) often prevail** with a steadfast control in traditional organizations; but they can provoke conflict with many agile teams. PMOs often influence estimates (when the least is known about the work), budgeting (and then financial management), resourcing (long before teams are assembled), scheduling, milestones, and development approaches. Almost every one of the twelve agile principles are violated with this front-end loaded approach—usually

because organizations don't know how else to manage. PMOs may also feel at risk when the organization promotes entrusting self-organizing teams to work in collaboration with a product owner. As examples:

- "working software as the primary measure of progress" isn't the same as meeting phased milestones,
- "sustainable development" interferes with imposed overtime and the heroics often needed to meet inherited deadlines, and
- "accepting changes to requirements, even late in the development," may introduce difficult-to-manage volatility with traditional requirements management practices.

While many organizations tout the PMP certification, they remain less committed to the Agile Certified Practitioner also sponsored by the Project Management Institute (PMI-ACP®).

With a variation in team sizes, practices, product owners, and backlog volatility, **teams are justifiably concerned with comparisons to other teams for productivity and quality.** Avoid measuring between teams; agile practitioners describe this as *meaningless* due to the unique nature of individuals and the tasks at hand. Avoid extrapolating velocity and quality data across teams. Instead, share proven practices among teams. Encourage ongoing improvements with teams. (Should be natural if teams are using retrospectives as intended.) Lastly, strive for consistency in practice—not always uniformity in performance.

### Some useful foundational agile measurements

- Use story points to estimate relative degree of difficulty with stories—the understanding that results from the discussion is at least as valuable the number itself.
- Use time to estimate tasks associated with stories that are being considered in a sprint.
- Use the sum of available (the work "week" does not equate to available time) team time as a threshold for stories committed in a sprint. Velocity, as derived from story points completed, can be used as an orthogonal approach and a reasonable alternative to validate the stories committed.
- Use velocity to estimate the lead time of the remaining estimated backlog. (Backlog items that have not been estimated with something like story points, are not well enough understood to be included.) Recall the estimates are valid as long as:

1. the product backlog remains unchanged, but recall also that we are using agile because we welcome changes to the product backlog rendering it to be more dynamic than static, and

*continued on page 16*

(Reflecting on Measurements, continued from page 15)

2. as long as the environment and team composition remain stable.

- Advocate agility by measuring value delivered by release, though some might also find this useful at a sprint level.
- Measure team wellness using techniques like niko-niko-like charts when in need.
- Avoid any schedule or budget related measures that can be misconstrued as a commitment. Be certain to include abundant footnotes to indicate the intentionally evolving nature of the backlog, priorities, and deliverables.

**Size and function points.** A long tradition of valuing software on the capabilities delivered using Function Point Analysis is tilted with agile development. The compelling factor is no longer the most capability for the investment; rather, the driver is the most valued delivery in rapid increments based on constantly changing priorities. “Yes” one can still ascribe function points to stories delivered—accurately and consistently if stories are “decomposed” to the transactional level. And why not decompose capabilities needed by the business in business language, such as:

As a customer, I want to make {Create} a reservation, so that . . .

As a customer, I want to look up {Read} a reservation, so that . . .

As a customer, I want to change {Update} a reservation, so that . . .

As a customer, I want to cancel {Delete} a reservation, so that . . .

CRUD is powerful for initially analyzing the customer needs—in the language of the business, with minimal translation needed. “Logical files,” a variable in Function Point Analysis aren’t changed by value delivery, but they too may be impacted with the ongoing delivery of product increments. Just because counting function points for agile projects can be done, don’t expect agile teams to display exuberance around another measure unless a value can be demonstrated to the agile team. Consider project-level size measurements to be ancillary at best to value delivery and velocity. While it may seem intuitive, even tempting, avoid any comparison of story points to function points.

**Very little scrutinized (credible) data exists across the agile community to compare teams and practices.**

First, because very few organizations use an agile framework like scrum consistently. The ScrumButs far outnumber any *standard* Scrum process or practices, in part, because there is no industry standard.<sup>12, 13</sup> Aspects of certain agile and scrum practices meet some criteria for elements of various ISO standards such as requirements and project management; nevertheless, a standard for an end-to-end agile process is non-existent, perhaps even intentional.

So, while the **challenges of measuring in the agile world abound, so too do the opportunities.** The intent to deliver value often does not conflict with having meaningful measures

that further help us to understand our capabilities. Empirical process control can be supplemented with data to verify our observations. Opinions matter. Data matters more—if you have it. The goal of both is to improve. Team-inspired improvement seems preferred to imposed improvement or a mandated edict to conform to classic cultural norms. Agile teams are said to thrive on self-organization and accountability for results. Ongoing reflection is the twelfth and last agile principle. Selected measures can bolster improvements and increase team cred. Hmmmmmmmmmmmm? How might we measure that?

**Special thanks.** *I want to thank the following colleagues, active agile practitioners, for their review and insightful comments that enhanced this article and sharpened my thoughts.*

- Brian Strickland, *Passionate Scrum Evangelist*
- Jennifer Turgeon; *Principal Member of the Technical Staff, Systems Research and Analysis; Sandia National Laboratories*
- Perry Waldner; *Vice President of Software Engineering, Walker Digital Table Systems* ■

## References:

- 0 *Inflate Gate: Mastering Overestimation for Agile Software Projects*; Computer Aid’s Accelerating IT Success, featured article; 8/3/2015; Schofield
- 1 *The CRASH Report 2014 – 2015; Executive Summary*; CAST
- 2 Actually disclosed during discussion to bring a military contract in “under cost”; Schofield, circa 2004
- 3 ORGANIZATIONAL BEHAVIOR AND HUMAN DECISION PROCESSES 48, 100 -130 (1991); Influences on the Appropriateness of Confidence in Judgment: Practice, Effort, Information, and Decision-Making; PAUL W. PAESE, University of Missouri-St. Louis; JANET A. SNIEZEK
- 4 *Measurements, Biases, Judgments: Understanding Variations for Reliable Estimates*; Keynote Address - Sao Paulo, Brazil; Brazilian Software Measurement & Analysis Conference; November 11, 2011; Joe Schofield
- 5 Underestimation in the “When It Gets Worse Before it Gets Better” Phenomenon in Process Improvement; Advanced Concurrent Engineering, 2011, Part 1, 3-10, DOI: 10.1007/978-0-85729-799-0\_1; Ricardo Valerdi and Braulio Fernandes
- 6 The Oz Principle; Connors, Smith, Hickman
- 7 *The Statistical Case Against the Case for using Lines of Code in Software Estimation*; 4th World Congress on Software Quality; Bethesda, MD.; September 17, 2008; slide 13
- 8 NIST citation
- 9 *Keep the Baby*; IFPUG *MetricViews*, January, 2015; Schofield

- 10 VERSIONONE; 8th Annual State of Agile™ Survey; 2014
- 11 VERSIONONE; 11th Annual State of Agile™ Survey; 2017; slide 13 –  
*How Success is Measured with Agile Projects*
- 12 Standards and Agile Software Development; W.H. MORKEL  
THEUNISSEN, DERRICK G. KOURIE AND BRUCE W. WATSON;  
ESPRESSO Research Group, Department of Computer Science,  
University of Pretoria, 2003; <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.5853&rep=rep1&type=pdf>
- 13 Conference on ENTERprise Information Systems / International  
Conference on Project MANAGEMENT / Conference on Health and  
Social Care Information Systems and Technologies, CENTERIS /  
ProjMAN / HCist 2015 October 7-9, 2015; A Compliance Analysis of  
Agile Methodologies with the ISO/IEC 29110 Project Management  
Process; Sergio Galvana\*, Manuel Morab, Rory V. O’Connorc,  
Francisco Acostad, Francisco Alvareze

**Additional resources:**

*Wanted: A GPS to Track My Agile Project*; ISMA<sup>11</sup>; Sao Paulo, Brazil;  
November 18, 2015

*Implications and Opportunities Relevant to “Agile” Measurement*; ISMA<sup>10</sup>;  
Charlotte, N.C., April 30, 2015

**About the author:**

**Joe Schofield** is a Past President of the International Function Point Users Group. He retired from Sandia National Laboratories as a Distinguished Member of the Technical Staff after 31-years. During twelve of those years he served as the SEPG Chair for an organization of about 400 personnel which was awarded a SW-CMM® Level 3 in 2005. He continued as the migration lead to CMMI® Level 4 until his departure. Life long learning: Joe holds six agile-related certifications: SA, SCT™, SMC™, SDC™, SPOC™, and SAMC™. He is also a Certified Software Quality Analyst and a Certified Software Measurement Specialist. Joe was a CMMI Institute certified Instructor for the Introduction to the CMMI®, a Certified Function Point Counting Specialist, and a Lockheed Martin certified Lean Six Sigma Black Belt. He completed his Master’s degree in MIS at the University of Arizona in 1980. Visit <http://joejr.com/bio.htm> for the full bio.



**CURRENT CONTACT  
INFORMATION?**

To ensure you won’t miss out

on any IFPUG communications,

please log in to your profile

on the IFPUG Members Services

Area and update your information.

Go to [www.ifpug.org](http://www.ifpug.org)

Send emails to [ifpug@ifpug.org](mailto:ifpug@ifpug.org), call

609-799-4900 or fax 609-799-7032

Write to:

IFPUG, 191 Clarksville Road,

Princeton Junction, NJ 08550