Not All Time Matters

Be Sure To Count What Does

Timothy A. Chick, SEI
Lana Cagle, Naval Oceanographic Office
Gene Miluk, SEI

Abstract. “Time” seems like a simple measure to use when planning and tracking projects. Most accounting systems track employees’ work hours. The resulting time measure is in essence a proxy for cost: Labor hours translate into dollars. However, if you have used time that has been tracked in that way to plan and estimate projects that depend on developers, testers, engineers, or other knowledge workers, you have probably found that your estimates aren’t as close to the actuals as you would like. If you have difficulty bringing projects in on schedule with the promised functionality, you might want to rethink the “time” you are using.

For this article, we are going to call the hours employees spend working “ordinary time.” Ordinary time is very useful for accounting and payroll purposes, but several major issues make it nearly impossible to use for creating accurate estimates or precisely tracking projects, especially in system development.

- Unrelated activities are included. The time being measured for payroll is often a mixture of activities that contribute to costs, but that are not directly related to project deliverables. Ordinary time includes tasks such as attending meetings or training classes, reading email, creating reports, and working on a myriad of administrative requirements.
- Detailed estimating is difficult to accomplish using ordinary time. Accounting systems do not generally track time at the level necessary for bottom-up, detailed estimating. Their data is instead well suited to top-down estimations for payroll and budget purposes.
- Precise project tracking is not supported by the data. The time data captured supports gross project tracking, but does not support the precise project tracking that system development projects need to be repeatedly successful. Fred Brooks best states the importance of adequate tracking in his book The Mythical Man-Month when he says, “How does a project get to be a year late? One day at a time.”

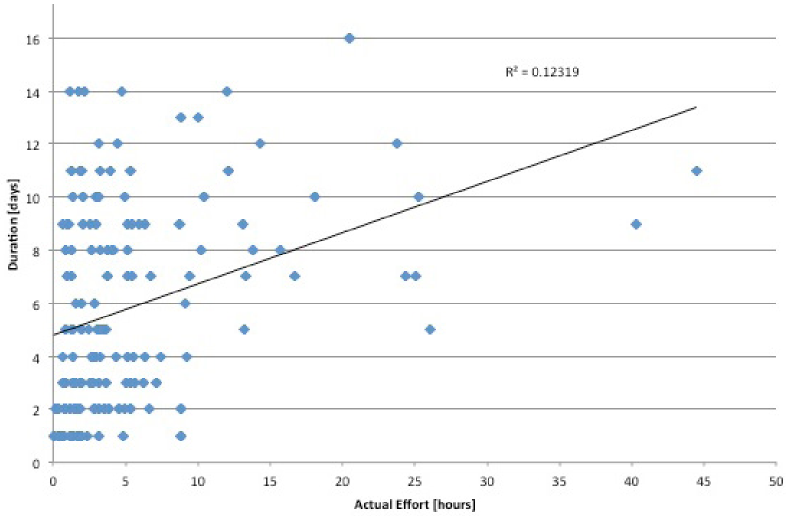

Figure 1: Effort And Duration, Work Packages < 20 Days Of Development

In short, ordinary time is not what a team of knowledge workers needs to successfully deliver a product or service on schedule and within budget. Using research done at the Software Engineering Institute (SEI) and experience gained working with the Naval Oceanographic Office, we hope to help others avoid the time trap by measuring the time that matters.

The Time Trap

The assumption of a relationship, or correlation, between effort and duration is fundamental to traditional project planning and tracking techniques, but is simply not a valid assumption for knowledge-based efforts. The problem is evident when we examine project planning approaches and tools for system development, which fit into two general categories: traditional and agile-based.

The traditional approach follows the PMBOK [1], which focuses on creating a work breakdown structure (WBS) in order to capture the project scope. The WBS is then used to define the activities and their sequence and to estimate activity resources and durations to facilitate the schedule. Given this, cost can be estimated and a budget generated. In order to generate cost estimates for labor, most financial systems require ordinary time, thus employee’s time is planned and tracked for every hour worked. This in turn is used in an earned value management system. The PMBOK uses costs to determine earned value and schedule variance. While these techniques have proven effective in industry, they have been shown to be less effective in software development [2,3,4] and on other knowledge-based projects.

One reason for the ineffectiveness can be seen in Figure 1, which shows that there is no correlation between effort and duration. To illustrate the difficulty in using ordinary time in this traditional approach we took data from 89 software development and knowledge-based projects in which the duration of the work package was less than 20 days, from open to close. The data shows that there is no correlation between the duration of a work package (i.e., ordinary time—in days— spent during development) and the actual effort required to complete it (i.e., only the time spent working directly on project deliverables).

Agile-based approaches and tools usually plan and track progress using story points. First, an ordered list of requirements called a product backlog [5] is generated, then the backlog is prioritized and each item in the backlog is estimated using story points. A story point is an arbitrary measure that represents the effort required to implement a story. The point system is based on the Fibonacci sequence [6]. All effort is performed using an iterative, time-boxed approach, usually called a sprint. Past performance is used to predict how many story points can be implemented per sprint. Most agile teams do not collect effort in terms of hours because they consider it to be a waste of effort [7]. However, there are several problems with relying solely on story points as a project’s only source of planning and tracking data.

- Story points and velocity are subjective measures that are calibrated by each team based on previous team performance [8].
- Estimates are biased [9].
- The use of story points is not objective and thus cannot be used to define a standard practice for the estimation of software size [10].

Time That Does Matter

A much more useful measure of effort for estimating, planning, and tracking systems development projects is "task time." Task time is defined as the actual time spent working on a specific task in the plan. To determine task time, each individual working on the team is responsible for tracking the time spent working on each specific task they are assigned in the plan.

Figure 2 shows that there is a very strong correlation between the planned and actual task time required to implement an individual work package. This high correlation enables very effective bottom-up estimating. Figure 3 shows the use of task time in creating accurate estimates. It plots actual task time against planned task time, in hours, for completed projects. This demonstrates that there is a strong correlation between the bottom-up estimates for a project and the actual task time spent on the project.

Task time can be used to overcome the deficiencies of both traditional and agile based approaches. Traditional planning and tracking methods can use task time to overcome the shortcomings of using ordinary time or financial information alone to determine when a project will be completed. Once you can determine when a project will be completed using a given set of resources, you can also determine how much the project will cost. Task time can also be used in place of agile's subjective story points approach, by providing an objective measurement which can be used as a standard estimating practice across teams and organizations, while conforming to the Agile Manifesto and Principles.

What Makes Task Time So special?

The reason task time is more effective for project planning and tracking is because it represents the project's value chain and ignores the other activities, which skew a predictability. The value chain is the set of tasks directly associated with a work package that is carried out to create value for the customer or end user. While effort reports record both ordinary time and time spent on primary tasks such as coding and testing, task time applies only to the primary tasks of a project's value chain. While the cost of project members can be predictable, the amount of task time individuals are able to commit to a project's value chain is highly variable, as seen in Figure 4. Understanding the variability at the individual and team level is key in producing accurate estimates and for precise status tracking.

Task time is measured in hours or minutes. Interrupt time, or off-task time, is not included in the time measure for a task; if there is an interruption during the work, that time is subtracted from the time measurement. Task time only measures the time spent working on a specific work package. In general, off-task time is not measured or tracked since it does not contribute to meeting the stated project goals.

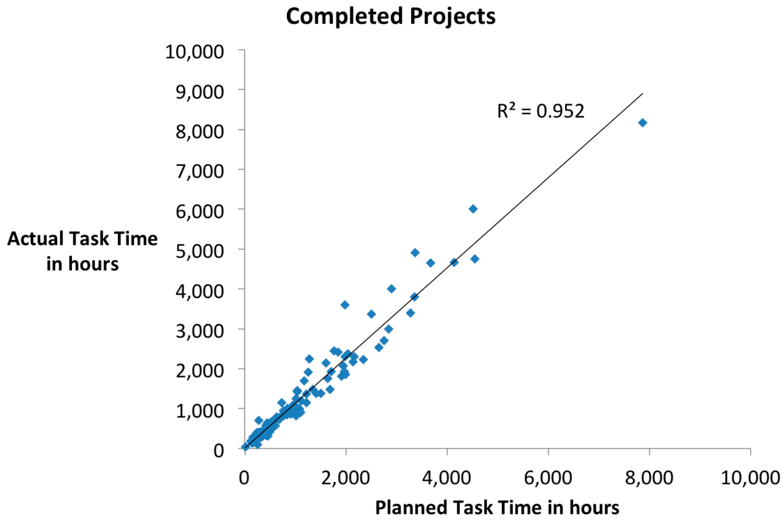

Figure 2: Actual Versus Planned Task Time For Completed Projects

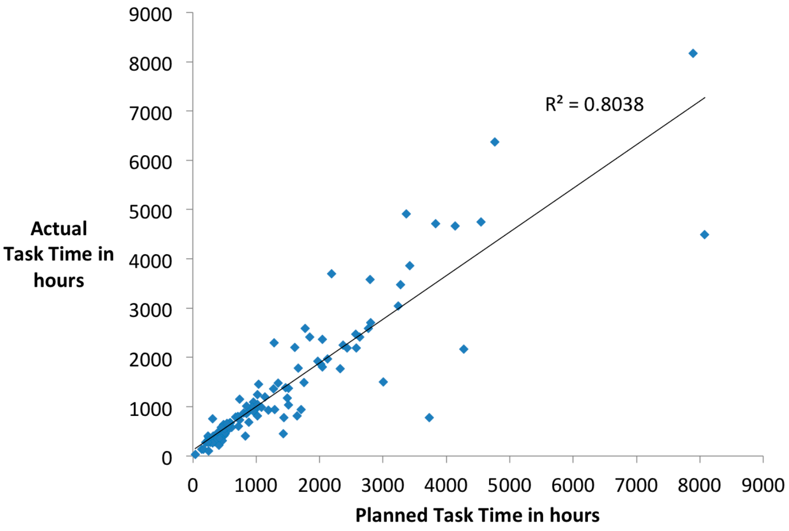

Figure 3: Actual Task Time Versus Plan Task Time

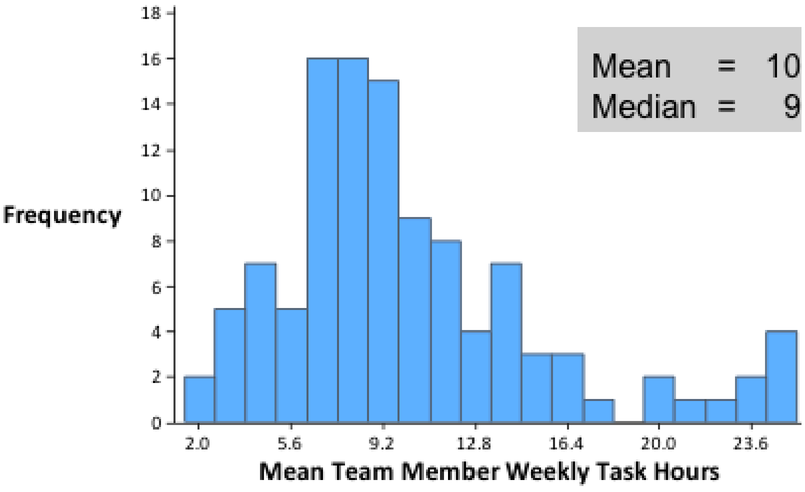

Figure 4: Average Weekly Task Time, In Hours, Per Individual

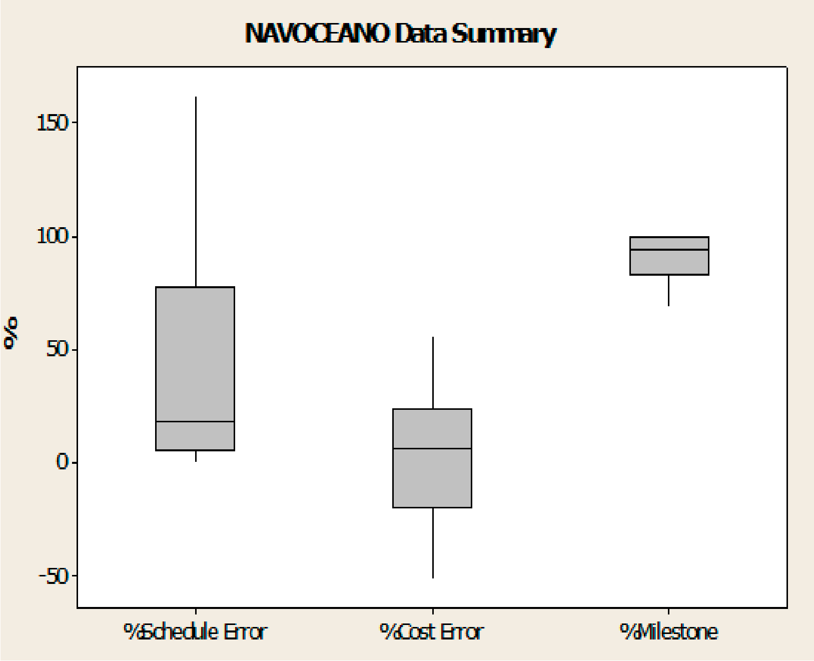

Figure 5: NAVOCEANO Summery Results

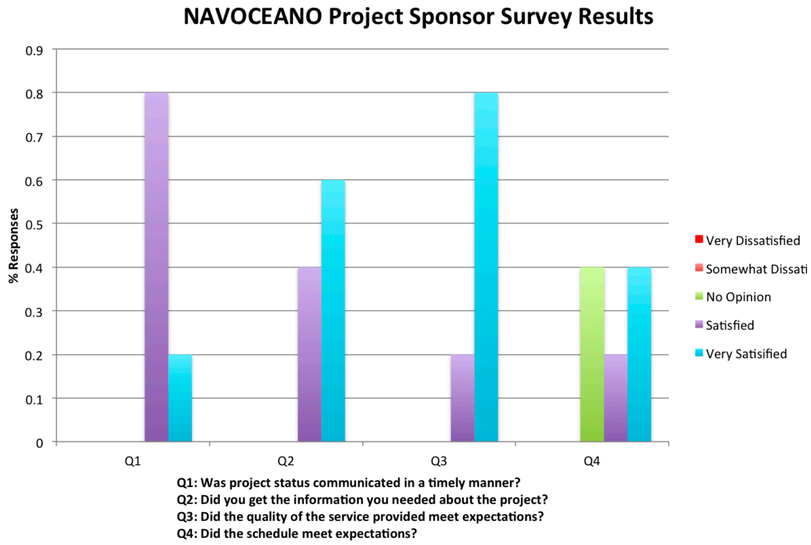

Figure 6: NAVOCEANO Project Sponsor Survey Results

The Naval Oceanographic Office Experience Using Task Time

In 2010 the Naval Oceanographic Office (NAVOCEANO) asked us to provide some support for their measurement program. NAVOCEANO is responsible for providing oceanographic products and services to all elements of the Department of Defense. As such they are operationally focused, and they focus less on development and engineering than many of the other commands. Their primary mission is to collect data through production and analysis to provide warfighters with the best available knowledge of the maritime battle space [11]. Thus, most of their staff consists of specialized knowledge workers [12].

Two of the key objectives they hoped to achieve by improving their measurement program were to provide more accurate estimates and to implement more precise status tracking. Given these objectives, we examined the mechanisms available for planning and tracking product development work and began to define the measures necessary to meet their requirements. We found that understanding and accurately tracking their "time" measure was vital to meeting their goals.

NAVOCEANO established a broader process improvement initiative in 2010, and one of their goals was to improve the organization's ability to plan, monitor, and control projects within the organization, specifically ones that crossed multiple organizational boundaries. As one of their managers put it, "We do a pretty good job of meeting our commitments within our individual silos, but when we try to work across silos we spend a lot of time and don't always deliver." Realizing that traditional command and control techniques were not optimal for certain types of work and based on past experience using the SEI's Team Software Process (TSP) for software engineering development [13], they decided to work with the SEI to more broadly apply the task-time-based, team-focused planning and tracking techniques to non-software teams within NAVOCEANO. Their approach included developing training specific to general knowledge workers, which was taught as just-in-time training for projects using the task-time approach. They also established a process working group to identify projects and support the transition.

In addition to learning what task time was, the knowledge workers were taught a systematic planning and tracking framework. The framework involved defining a project's scope or requirements, and then breaking down the requirements into manageable parts to create a conceptual design. The conceptual design was then organized in a WBS.

Once the work packages were defined, tasks could be determined. The tasks were based on the processes selected or created to produce the work packages. The work was then estimated. Historical data from past projects, contained in an effort database, and resource availability for the current project were used in the planning framework to develop the effort estimates and schedule.

During a three-year period NAVOCEANO established 12 different projects, which in total consisted of 23 "cycles." A cycle is the project's detailed planning horizons, which ran between 3 and 23 weeks and averaged 16 weeks in duration. All of the projects consisted of full-time employees working on the assigned project on a part-time basis. Every cycle included deliverables and milestones. Most projects consisted of members from across multiple organizational silos and ranged in size from 6 to 18 members, with an average of 9 members.

Each project was planned and executed incrementally. During initial planning, near-term tasks were planned in detail and distant tasks were planned at a higher level. These detailed plans were used to guide the work of individuals and allow them to precisely track their progress. As the measurement framework was implemented, managers for cross-silo projects found they had a new ability: to obtain accurate data about the status of their projects.

Figure 5 provides a summary of their results. It shows that they had a tendency to deliver late about 19% of the time, which was mostly due to the part-time nature of the work and the changing demands from the operational environment. A big difference from previous efforts was that managers now had the data to be able to communicate the state of the project in a quantitative, accurate, and defensible way. This allowed management to be more proactive in negotiating resources across silos and setting expectations with customers.

Had they been able to spend the time on the project they had planned, they would have probably delivered early. This is due to the fact that the teams had a tendency to underspend, which is shown in the 7% Cost Error. That said, 95% of the projects objectives were met, which was a huge improvement from past experiences.

At the end of each project the project's sponsors were surveyed. Figure 6 shows the sponsors' evaluations of the projects. In general the survey shows that the projects were able to communicate their status in a timely manner, meet the sponsor's information needs and schedule expectations, and still deliver quality services.

Conclusion

Understanding the dynamics of your system development process in a quantifiable way increases the accuracy of estimates, the reliability of status reporting, and perceived customer quality and satisfaction. The key to any fundamental knowledge of system development dynamics and economics is an accurate, reliable, and repeatable measure of time. Using experiential and quantitative data, this paper demonstrates that the most common means of capturing time for system development projects is often inaccurate for the intended purposes. The use of task time with a well-defined process value chain improves accuracy and reliability while providing what customers desire most: quality products on schedule and within budget.

REFERENCES

1. A Guide to the Project Management Body of Knowledge – Fourth Edition, 2008. Project Management Institute, An American National Standard, ANSI/PMI 99-001-2008.
2. Cabri, Anthony, and Mike Griffiths. "Earned Value and Agile Reporting." Web. 30 May 2014.
3. Corovic, M. By Radenko. Why EVM Is Not Good for Schedule Performance Analyses (and How It Could Be...). Web. 30 May 30.
4. Solomon, Paul. "Perfomance Based Earned Value." CrossTalk, 2005: http://www.crosstalkonline.org/storage/issue-archives/2005/200508/200508-Solomon.pdf
5. "Scrum (software Development)." Wikipedia. Wikimedia Foundation, 30 May 2014. Web. 30 May 2014.
6. "What Is a Story Point ?" Web log post. AgileFaq. Web. 30 May 2014.
7. Berteig, Mishkin. "Agile Advice." Agile Advice. N.p., 29 July 2013. Web. 30 May 2014.
8. Moser, R., Pedrycz, W., Succi , G. 2007. Incremental effort prediction models in Agile Development using Radial Basis Functions. Proc. 19th International Conf. on Software Engineering & Knowledge Engineering (Boston, MA, USA, July 9-11, 2007), SEKE'07, pp. 519-522.
9. N.C. Haugen, "An empirical study of using planning poker for user story estimation," Minneapolis, MN, Agile 2006 Conference Proceedings
10. Javdani, T., Zulzalil, H., Ghani, A., Sultan A., Parizi, R. 2012. "On the Current Measurement Practices in Agile Software Development," IJCSI, Vol. 9, Issue 4,No 3, July 2012
11. Naval Oceanographic Office. US Navy, n.d. Web. 30 May 2014. http://www.public.navy.mil/fltfor/cnmoc/Pages/navo_home1.aspx
12. Rouse, Margaret. "Knowledge Worker." Web blog post. What Is ? Web. 30 May 2014.
13. Battle, Ed. Leading & Learning – Using TSP at the MSG Level, Sept. 2009. Web.
14. "Partners - Find or Become an SEI Partner or CMMI Institute Partner." Partners Clearmodel. N.p., n.d. Web. 30 May 2014.
15. Kasunic, Mark. SEI Interactive Session: Empirical Study of Software Engineering Results, September 18, 2013. Web.

Acknowledgements

Many people have participated in the work that led to this article, but we would like to give special thanks to Mark Kasunic, William Nichols, and James Over. Without their hard work analyzing all the data, these finding would not have been as compelling or insightful. We would also like to thank Erin Harper for all her technical editorial improvements, which enhanced the overall readability.

The data reported in figures 1, 2, 3, and 4 comes from 113 different software development projects, which reported their results to the SEI through its Partner Network [14]. The projects started between July 2000 and July 2012 and had an average project duration of 119 calendar days, with a team size of less than 20 people. All the projects in this data set used the same definition of task time and a planning framework [15]. ◈

ABOUT THE AUTHORS

Timothy A. Chick is a Senior Member of the Technical Staff at Carnegie Mellon University's Software Engineering Institute (SEI). He is a certified PMP, and Scrum Master. He also holds a Lean Six Sigma Black Belt certification and was a certified CMMI-DEV/SVC instructor. Prior to the SEI, he worked for Naval Air Systems Command, as the Software Acquisition Lead for the Fire Scout and former software project manager for the E-2C Hawkeye Program.

Lana Cagle is the Quality Advisor for the Systems Integration Division of the Naval Oceanographic Office, Stennis Space Center, Mississippi. She has worked as a process improvement change agent for 20 years, serving in a lead capacity since 1997. For the last four years she has been assisting in transitioning process improvement methods to the larger organization. She is a certified Team Software Process Mentor Coach and a Lean Six Sigma Green Belt.

Dr. Gene Miluk is currently a Senior Member of the Technical Staff at the Software Engineering Institute (SEI), Carnegie Mellon University. For the past 20 years Gene had been working with SEI client organizations undertaking software process improvement, software acquisition improvement and technology transition. He is a TSP instructor and SEI Certified Team Software Process Mentor Coach . Gene is also a Six Sigma Black Belt, a certified SCRUM Master and certified Project Management Professional.