Tracking Progress through Earned Value

Hakan Erdogmus

Considered a gross misnomer by some, earned value for progress tracking in software development projects is still little known in the software industry. But the concept shouldn’t be so unfamiliar: tracking progress by earned value is similar to tracking progress through burnout charts (http://xprogramming.com/articles/bigvisiblecharts)—the ubiquitous, simple and powerful visuals that are so popular in agile software development. The two techniques, although developed independently in very different contexts, are similar in terms of their information content. This isn’t a new discovery: Alistair Cockburn talked about the relationship between earned value and burnout charts as early as 2004 in Chapter 3 of his book Crystal Clear (Addison-Wesley, 2004). Section 7.3 of Grigori Melnik’s and Gerard Meszaros’ volume 1 of Acceptance Test Engineering Guide (http://testingguidance.codeplex.com) also discusses the same relationship.

Burnout charts track progress using a “countdown to zero” approach. Remaining work scheduled to be in a future release or project is represented as the sum of features that are weighted using an effort estimate, such as story points or ideal programming days. Earned value achieves the same effect by adopting a “count up to 100 percent complete functionality” approach. Cockburn argues that while burnout charts are emotionally more powerful, they treat planned work as fixed rather than expandable. This limitation makes them less than ideal when the scope changes midstream within the tracking period.

Why Consider Earned Value?

Earned value differs from the burnout approach in two ways. First, earned value expresses the amount of work in monetary terms by weighting a feature’s percent completeness by the feature’s estimated cost. Detractors of earned value see one major problem with this representation: by referring to the effort as “value,” the monetary equivalent of a feature is seemingly equated to the feature’s business value. They are right. Earned value is essentially an expression of the feature’s completeness in cost, or effort, terms.

The second distinction is the emphasis placed by earned value on tracking progress against a plan, an emphasis that is absent in the burnout approach. The presence of a point of reference against which to compare progress introduces a number of additional metrics that burnout charts do not incorporate. Tracking progress through earned value and related metrics give rise to earned value management. Let’s elaborate on this larger concept a bit.

What’s Earned Value Management?

Earned value management (EVM) is a budget and schedule tracking technique developed by the Project Management Institute (Practice and Standard for Earned Value Management, 2006). EVM connects tightly with other PMI standards and concepts, such as work breakdown structures. The sidebar (“Earned Value Management and Agile Projects”) I contributed to volume 1 of Accep-
Software Boards

I welcome Ipek Ozkaya to Software’s Advisory Board. She is a senior member of the technical staff at the Research, Technology, and System Solutions Program of the Carnegie Mellon University’s Software Engineering Institute. Her work focuses on development and application of techniques for improving software architecture practices. Ipek, a real architect by education, also teaches software architects, developers, and managers in SEI’s Software Architecture Certificate Program. Ipek received her PhD and MSc in computational design from Carnegie Mellon University. She was very instrumental in Software’s successful collaboration with the 2010 SATURN conference.

Cost and Schedule Variance

In EVM, actual costs are also tracked. If at any point in the project, the actual accumulated cost exceeds the earned value of the project at that time, the project is considered over budget, with the shortfall representing a budget overrun. In EVM terms, this shortfall, or slack if actual costs are below earned value, is called cost variance.

If the accumulated earned value at a point in time is below the estimated total expenditures according to the planned budget, then the project is considered over schedule, with the difference representing the monetary expression of the project’s schedule shortfall at that point. In EVM the name suggests, earned value is essentially a relative effort tracking metric.

Figure 1. Earned value management metrics (adapted from http://testingguide.codeplex.com, sidebar “Earned Value Management and Agile Projects”). As of July 2010, this project is 120 percent over budget, 50 percent over schedule, and three months behind schedule.
Software Experts Summit

After another intense and productive annual board meeting in Munich, many members of IEEE Software’s editorial and advisory boards presented talks on 9 June at the Software Experts Summit hosted by Siemens AG. The summit exclusively featured IEEE Software speakers; I was in exceptionally good company with Ayse Bener, Jan Bosch, Christof Ebert, John Favaro, Kevin Henney, Frank Maurer, Maurizio Morisio, Frances Paulisch, Linda Rising, Helen Sharp, Forrest Shull, Michiel van Genuchten, Laurence Tratt, and Markus Voelter. The speakers tackled a wide range of topics: domain-specific languages, effectiveness of test-driven approaches, organizational change, business models, motivational factors, holistic product development, compositional software engineering, use of predictive models, impact of software in non-ICT fields, and usefulness of static analysis tools.

The summit was a great success, thanks to organizers Frances Paulisch, Christine Kuznik-Lohr, and Christian Lescher. IEEE Software sponsored students Frank Denninger, Markus Luckey, and Stefan Schießl to attend the summit (see Figure A).

Check the IEEE Computer Society’s Computing Now portal at http://computingnow.computer.org/sw/ses10 for a synopsis of the summit, slide presentations, and interviews with the speakers.

Figure A. Markus Luckey, a University of Paderborn student, and Frank Denninger and Stefan Schießl, students at the University of Erlangen, were chosen to be IEEE Software’s sponsored guests at the Summit.

For more information, visit www.ieee.org/web/aboutus/whatis/policies/p9-26.html.

Figure: Markus Luckey, a University of Paderborn student, and Frank Denninger and Stefan Schießl, students at the University of Erlangen, were chosen to be IEEE Software’s sponsored guests at the Summit.

terms, the schedule shortfall or slack is called schedule variance. Schedule variance can also be expressed temporally, in calendar time.

Figure 1 illustrates schedule variance (in both monetary and temporal terms) and cost variance for a project with an estimated total budget of $5 million and an estimated schedule of one year. In July 2010, at an earned value of $2.5 million, the project has a total earned value that’s only 50 percent of its planned total cost, but it should have accumulated 75 percent of that total cost according to the project plan. Therefore it’s over schedule by $1.25 million, still 50 percent, in monetary terms. Since the project should have

accumulated that much earned value back in April 2010, it’s three months behind in calendar time terms. This is the schedule variance. The project is also 110 percent over budget on July 2010 because it has accumulated an actual cost of $5.25 million at that date compared to the $2.5 million it has earned.

EVM in a Time-Boxed Incremental Context

In “Earned value and Agile Reporting” (Proceedings of AGILE ’06, IEEE Computer Society Press, 2006, pp. 17–22), Mike Griffiths and Anthony Cabri advocated applying EVM incrementally, on an iteration-by-iteration or release-
IEEE Software Supports Community Conferences and Promising Work

IEEE Software supports a number of conferences in order to spread the word about useful community activities and to recognize outstanding work in the software profession. Thanks to Ipek Ozkaya and Hakan Erdogmus, here are highlights from two recent conferences.

SATURN: Best Presentation Awards

IEEE Software sponsored two outstanding presenter awards during the Sixth Software Engineering Institute (SEI) Architecture Technology User Network (SATURN) Conference, in May 2010. The awards were created to honor presenters for their contributions to architecture-centric practices. The awards, the first ever at a SATURN conference, were conferred based directly on conference attendees’ votes.

A joint committee consisting of IEEE Software board and SATURN technical committee members created the awards with the goal of contributing to the maturation of architecture-centric engineering practice. The Architecture in Practice Presentation Award recognizes excellent description of architecture-centric practices in real-world projects, whereas the New Directions Award recognizes excellent presentation of emerging ideas about how architecture-centric practices can catalyze change and innovation in today’s practices to deliver better systems faster.

The conference attendees awarded the Architecture in Practice Award to Anthony Tsakiris, a vehicle-controls technical specialist from Ford Motor Company, for his presentation titled “Managing Software Interfaces of On-Board Automotive Controllers.” Tsakiris summarized an effort at Ford to standardize the control-system software interfaces across a set of in-vehicle, software-intensive controllers in an automotive setting. His presentation described the re-architecting of a system from a conventional power train to a hybrid electric power train within organizational and legacy-system constraints. Tsakiris emphasized the importance of having an architecture roadmap. He also suggested that when organizational habits are deeply rooted, evolving existing products and processes can be more effective than trying to revolutionize them. Tsakiris succeeded by slowly introducing architecture-centric practices, working incrementally from implementation to architecture, and persistently allocating time to educate his team about the architecture.

Managing system evolution was a strong thread among this year’s presentations. Tsakiris’s point that an architecture roadmap can significantly increase the quality of the end result was unanimously advanced by presenters from ABB and Credit Suisse.

The New Directions Presentation Award went to Olaf Zimmermann, senior IT architect and currently a research staff member at IBM Zurich Research Lab, for his presentation titled “An Architectural Decision Modeling Framework for SOA and Cloud Design.” Zimmermann talked about how reusable architectural decision models can guide the architect through architectural decisions in SOA/cloud design. The work Zimmermann presented aimed to influence the way architectural decisions are captured and used throughout the software development life cycle. Through this work, architectural decision rationale could become another essential viewpoint of architecture. Zimmermann also asserted that this important aspect of software architecture should be viewed not as capturing decisions retrospectively but as using them proactively to improve system design.

Tsakiris’s and Zimmermann’s presentations fit into the larger theme of the conference, architecting for change. Several presentations emphasized the need for organizations to meet their business goals in rapidly and continuously changing environments by reconciling agile software development and software architecture. Jim Highsmith in his keynote stated that architects can accelerate agility by being not only architects of structure but also of time and transition; by creating agile design rules; and by developing technical-debt prevention and reduction strategies. Philippe Kruchten offered a zip metaphor for combining functional features and architectural tasks in iteration planning.

The consistent message, echoed in the March/April 2010 Agility and Architecture issue of IEEE Software, was that agile development and software architecture are not oppositional, but that people in all systems development roles must do their part in building high-quality software that meets its business and stakeholder goals.

The award-winning presentations, as well as the keynote and other presentations from the SATURN 2010 Conference, are available for download at www.sei.cmu.edu/saturn. —Ipek Ozkaya

XP 2010: Best Paper Awards

This year IEEE Software gave two best paper awards at the 11th International Conference on Agile Software Development (XP 2010) in June. I announced the recipients at the conference’s opening reception, where the delighted authors were presented with certificates recognizing their contributions. We gave awards for two accepted submissions that were deemed most relevant or having the best potential to impact software practice. The winners also received a one-year complimentary subscription to IEEE Software.

The Experience Reports and Research Papers track chairs shortlisted five articles, and a 13-member committee selected the final recipients. Many thanks to the committee members who served with me: Jutta Eckstein, Xiaofeng Wang, Alberto Sillitti, Elizabeth Whitworth, Kieran Conboy, Frances Paulisch, Lars Arne Skar, Steve Fraser, Rachel Davies, Nils Brede Moe, and Angela Martin.

So who were the two winners?

continued on p. 6…
by-release basis. In some projects, a grand plan for the whole project may not exist. However, a local plan in terms of work scheduled for the current scope (features or stories) together with estimated costs (resources allocated to the current scope) could be available and reliable enough to serve as a reference point on a per-iteration or per-release basis. Budget and schedule overruns can then be calculated within each such increment, whether for a single iteration or release, using EVM. Effort or a proxy for effort such as story points can be substituted for cost, just as done in burndown charts.

Even if mini, local plans aren’t available and scope changes over the course of the project, EVM can still be applied. Cockburn describes how to handle scope changes using an idea suggested by Phil Goodwin and Russ Rufer (http://alistair.cockburn.us/Earned-value+and+burn+charts).

The IEEE Software XP 2010 Best Experience Report Award went to Jørn Ola Birkeland for his report entitled “From a Timebox Tangle to a More Flexible Flow” (Figure B). Birkeland works for the consulting firm Beck Consulting AS in Oslo, Norway. His report recounts a team’s experience and discusses lessons learned in moving from a timeboxed process to a flow-based process that uses a continuously updated, prioritized backlog with a single work-in-progress limit.

Here is how the members of the selection committee characterized Birkeland’s contribution:

- “It hit arguably the hottest topic of the moment: lean, and agile vs. lean.”

The IEEE Software Best Research Paper Award went to Tor Erlend Fægri for his work entitled “Adoption of Team Estimation in a Specialist Organizational Environment” (Figure C). Fægri works for the Norwegian research institute SINTEF’s Information and Communication Technology group in Trondheim. In his article, Fægri describes how intervention by an outside party (in this case, his intervention as an action researcher) produces significant change in a team operating under a vicious cycle. The team in Fægri’s article faces intense work pressure, which impedes group learning, and the poor group learning in turn further amplifies work pressure.

Members of the selection committee praised Fægri’s work as follows:

- “Fægri covers a topic of importance, that is, specialization vs. generalization, which also is under some scrutiny and debate in the agile community. So research on this topic, as well as guidance in how to deal with it, is highly appreciated.”
- “The practical relevance is absolutely high. There are teams who can benefit from this report when undergoing the change to agility.”
- “The article demonstrates the ability of a researcher to effect positive change in a real project by taking a balanced stake using action research.”

EVM in a Kanban-Style Incremental Context

If a Kanban-based, continuous-workflow approach is adopted instead of timeboxes, applying EVM becomes problematic. Because typically budget plans or estimates are absent altogether in Kanban systems, variance-tracking metrics cease to exist, but tracking actual throughput is still feasible. Tracking project progress reduces to tracking flow. In Chapter 12 of his book Kanban: Successful Evolutionary Change for Your Technology Business (Blue Hole Press, 2010), David J. Anderson explains the metrics and reporting mechanism applicable in a Kanban context. Anderson recommends the use of cumulative flow diagrams that plot total number of work items as a function of time. These diagrams can be converted to actual cost curves of the kind depicted in Figure 1 by weighting each work item with the item’s actual cost. Earned value can then be defined to coincide with cumulative actual cost rather than cumulative relative cost. Effectively, the plan equals reality: there’s no need to reason about schedule or cost variance.

Substituting Business Value for Earned Value

When customer-assessed value, or business value, is more critical to project funders than textbook earned value as defined in EVM, a substitution of concepts may be warranted. An EVM-like model can track the business value provided that each work unit has a tangible financial value or an intangible utility assigned by the product owner or project funder.

Business value can be expressed in absolute, monetary terms (in currency) or in abstract, utility terms (for example, in utils). Before the fact, the business value is an estimate based on the product owner’s or project funder’s assessment. After the fact, it represents a reassessment in an accepted and delivered product, thus proxying actual value. Unlike actual cost in classical EVM, actual business value is thus often subjective rather than observed or measured. Although actual business value is hard to measure, especially at the feature level, it can still be informed and backed up by meaningful economic analysis.

When business value replaces earned value, adjustments might be required. For example, only fully deployed features may earn their respective business value. Similarly, if a feature is useful in the field only when delivered together with another feature, it shouldn’t earn its respective business value until all the features on which it depends are also deployed. Percentage-based adjustments proportional to a feature’s status of completeness might not make much sense either for tracking business value. Partial or percentage business value earned is only sensible at the whole project level.

What’s the verdict on EVM? EVM might not be adaptable in an out-of-the-box form for our industry, but it does offer a few extras that better-known alternatives don’t mention. The EVM approach deserves more attention from the software development community in environments where planning and estimation is part of the culture.

CALL FOR ARTICLES

Software Components beyond Programming—From Routines to Services

Final submissions due: 1 November 2010
Publication date: May/June 2011

Components have been known in software engineering since 1968 and have had several revivals, including component-based software engineering and service-oriented architectures. These approaches brought new expectations and advanced the current state of art and practice, but also left unfulfilled promises.

This special issue aims to provide an overview of what component software has contributed to software engineering, what challenges have been addressed, which have been solved and which not, and what is the state of the art and current industrial practice. We plan to show successes, experiences, and possibly failures in component-based approaches in practice. Our aim is also to bring about a discussion of different aspects of component software used in different application domains—such as information systems, or embedded systems—and also in different software engineering domains—such as SOA, product line architectures, and so on. We especially welcome case studies, lessons learned, and success and failure stories in component software. The issue will summarize current results and expectations for the near future.

For more information about the focus, contact the Guest Editors:
- Ivica Crnkovic, Mälardalen University, Sweden, ivica.crnkovic@mdh.se
- Judith Stafford, Tufts University, US, jas@cs.tufts.edu
- Clemens Szyperski, Microsoft, clemens.szyperski@microsoft.com

For complete CFP: www.computer.org/software/cfp3
For author guidelines: www.computer.org/software/author
For submission details: https://mc.manuscriptcentral.com/sw-cs