



Michael James is a software process mentor and Certified Scrum Trainer, focusing on the engineering practices that enable Agile project management. Having worked in the software industry for more than 20 years as a software developer (formerly “architect”), he has experience in automated testing that predates the Extreme Programming movement; formal, phased, high-ceremony processes based on DOD-STD-2167A; chaotic non-processes of the dot-com era; and Agile processes including Scrum and XP.

## Reasons to Disable Your Tool’s Estimates vs. Actuals, Ideal Line, and Individual Burndown Charts

Posted by Michael James on June 6, 2008 at 12:08 a.m.

Organizations that are looking for an automated tool to help manage Scrum fall along a continuum.

- At one end are the Scrum purists, who feel that resistance to doing real Scrum is an *organizational impediment* to be exposed and rooted out rather than accommodated as business as usual. These folks believe the highest performance and ingenuity comes from intense collaboration, small cross-functional teams lacking defined roles, self organization, face to face communication, and even a certain amount of chaos during Sprint execution. They typically prefer lightweight tools that make the Scrum artifacts visible without impeding team self organization.
- At the other end are those who want some version of the Scrum/Agile practices, but also feel “traditional” practices (such as Taylorism, waterfall, or parts of the PMBOK) would meet their current needs. They may be dealing with impediments to self organization, lack of co-location, suboptimal team composition, large departments, entrenched practices, regulatory requirements, etc. Or they may not be in a position to wait for teams to go through the “forming, storming, norming, performing” growth stages. These folks tend to request features that make the purists cringe.
- Some are in organizations that are transitioning from traditional practices to Scrum. Some skeptics I’ve known became full-fledged advocates after seeing results from pilot teams attempting uncompromised Scrum with good coaching.

As someone who’s seen the power of uncompromised Scrum firsthand, I feel our job is to give you the tools you need today while nudging you to use them in a way that unlocks what your team is really capable of. If your existing practices are working fine, read no further. If not, consider this article a nudge.

**danube**

Danube Technologies, Inc.  
520 SW 6th Ave, Suite 940  
Portland, OR 97204 USA  
1.888.5.danube  
T: +1.503.248.0800  
F: +1.503.248.0801

## Why not compare task estimates to actuals?

In Frederick Taylor's world of Scientific Management<sup>1</sup>, people engage in repetitious and/or procedural work that requires their bodies more than their imaginations. In a repetitious world we can make accurate predictions down to the daily or hourly task level, and reasonably conclude something is wrong when the actual doesn't line up with the estimate.

Scrum/Agile approaches are intended for complex work that is *inherently unpredictable* down at the hourly task level. If our hourly activities are inherently unpredictable and we're being judged by how well our actuals match our estimates, we'll feel pressure to:

1. pad our estimates,
2. decline to give estimates without extensive analysis,
3. lie (perhaps even to ourselves) about our actuals,
4. compromise the definition of "done" when allotted time runs out,
5. decline to take on difficult, risky tasks, or
6. blame others.

Can micromasurement really invite micromanagement? A famous Scrum pioneer recently reported a large web sales company losing 50% of total production from their hourly reporting system.

You'll have to decide for yourself whether your work fits more into the space where reality follows plans, or the one where plans follow reality. Most projects have a blend of both kinds of work. New product development, a kind of inventing requiring human imagination and team ingenuity, probably fits into the second category more than the first.

## Why not draw an "ideal line" on the burndown?

The "ideal line" some well-intentioned people like to draw for the Sprint Burndown Chart is another subtle force against the transparency required for collaboration. As Kane Mar has written<sup>2</sup>, a high-performing team can expect the Sprint Burndown to go up before it

---

<sup>1</sup> <http://www.eldritchpress.org/fwt/t1.html>

<sup>2</sup> <http://kanemar.com/2006/11/07/seven-common-sprint-burndown-graph-signatures/>

goes down<sup>3</sup> because (in the complex realm) it's impossible to predict all contingencies before beginning the actual work. The task discovery, including detailed requirements analysis, is part of the Sprint's work. In [The Enterprise and Scrum](http://www.amazon.com/Enterprise-Scrum-Ken-Schwaber/dp/0735623376)<sup>4</sup>, Ken Schwaber writes, "Many are developed during the Sprint Planning Meeting, but up to 40 percent might emerge during the Sprint."

The "ideal line" (which Kane Mar refers to as "The Fakey-Fakey") is not ideal at all, and may actually suggest a combination of padding and self deception. If you're using a tool that draws this line, consider disabling that feature to reduce its anchoring<sup>5</sup> effect.

### Why show the weekend on the burndown?

Have you ever struggled with a difficult problem all evening, then solved it easily the next morning? Your subconscious mind works weekends too, and you may find it useful to see breakthroughs that occur after the team has rested. Also see "ideal line" discussion above - why does seeing a bumpy line make you uncomfortable?

### Why not focus on "individual burndown charts"?

Another flavor of misapplied Taylorism is the "individual burndown chart" -- a perversion of Ken Schwaber's intended burndown chart that shows lines for each individual rather than the whole team together. A high performing Scrum team is characterized by people collaborating (or "swarming"<sup>6</sup>) on tasks. Tasks have point people as primary champions to ensure they get done, but they're owned by the entire team. As a senior developer, sometimes the best thing I can do for my team is mentor less experienced people with their tasks. An "individual burndown chart" would certainly discourage me from doing this. The last person I talked to using this feature described a "race" between developers to finish their tasks, then no one taking responsibility for testing and overall quality. If this is happening to you, consider disabling individual burndowns. Redirect focus to the whole team's work, particularly the story acceptance criteria.

---

<sup>3</sup> <http://danube.com/system/files/SprintBurndown.png>

<sup>4</sup> <http://www.amazon.com/Enterprise-Scrum-Ken-Schwaber/dp/0735623376>

<sup>5</sup> <http://en.wikipedia.org/wiki/Anchoring>

<sup>6</sup> [http://danube.com/blog/victorszalvay/famous\\_last\\_words](http://danube.com/blog/victorszalvay/famous_last_words)



## Is there any good news?

It may be painful to accept the inherent unpredictability of complex work at the micro level. But remember, your competitors are facing the same challenges. Letting go of attachment, unsticking your compass needle before they do, can give you an edge.

*It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.*

--Charles Darwin

Here's more good news: At the *macro* level, some predictability does emerge. I describe a proven method to nail delivery dates by looking at the big picture here <http://danube.com/whitepaper/macro-measurements>.

--mj

Michael James  
Software Process Mentor  
<http://www.danube.com>

**danube**

Danube Technologies, Inc.  
520 SW 6th Ave, Suite 940  
Portland, OR 97204 USA  
1.888.5.danube  
T: +1.503.248.0800  
F: +1.503.248.0801