



Dan Rawsthorne, Ph.D. is a Certified Scrum Trainer and one of Danube Technologies' transformation coaches, helping organizations transition to Agile practices. Rawsthorne is a 27-year veteran of the software industry, having worked in a wide range of capacities, from coder to project manager, for companies large and small. In addition to possessing deep knowledge of many software processes, procedures, and techniques, he holds a Ph.D. in mathematics.

## What Do Story Points Represent?

Posted by Dan Rawsthorne on March 5, 2008 at 10:32 a.m. and updated May 21, 2008, based on comments

Since we are astute enough that we're not expecting a story's size (in Story Points) to be an *accurate* reflection of the effort it takes to complete the story – even though we expect a correlation on average – what is our expectation of what a story's size reflects? This is a very good question, and causes us to think about the four factors that influence estimates: intrinsic difficulty, robustness of the solution, viscosity of the code we're working with, and team ability.

Here's what I think.

I think that we're trying to size the intrinsic difficulty of the story. That is, we're trying to get a relative estimate of how hard solving the problem will be in an *ideal* world, not how much work it will *actually* take to provide a solution. There are two major reasons I believe it should be done this way and here they are:

- This type of size can be estimated with the information we have. When we are sizing the story, we are discussing the story without thorough, or detailed, knowledge of the codebase or who will be actually working on the story. We can ask a question like “how hard is this story, given a reasonable definition of ‘done,’ clean code to work with, and an ideal team?” Since we are looking for a fairly ambiguous answer, we can give it. If we can't understand it well enough to agree if it's XS, S, M, L, or XL, then we just say it's an epic – it's too difficult to size right now... and we wind up with an analysis story/spike to figure it out.
- I know that the other three factors can't help but creep into the estimators' minds, but if we know that all we're looking for is intrinsic difficulty, the others can't dominate the conversation – they're technically out of scope for the discussion. And, if they do creep in, the sizes just get better (more accurate, more consistent, etc.)... which is a good thing.

Anyway, what we're trying to do is to estimate the intrinsic difficulty of the story. We like to ask questions like “how big is the pile of rocks?” or “how many miles is it?” I like this type of metaphor for a story's size, because it clearly separates the story's intrinsic difficulty from the other variables.

**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

If we are running a mile, for example, it's always a mile. However, some miles are harder to run than others. Sometimes we're in a park, on the grass, the sun is shining, and there's a gentle breeze. Sometimes we're in the woods, going up and down hills, crashing through the underbrush, it's snowing, it's dark, and the monster is after us. Sometimes the runner is an Olympic-class runner. Other times it's a fat old man like me. But it's always a mile...

In order to make sure a mile is always a mile and that there's not some extra stuff hiding in there, I have another basic rule that applies to "coding stories." I like to think of a story as being a single scenario, a single thread, a single interaction, a single case or state, or as having a single "positive" test. This notion limits the scope of a story within a feature, and prevents scope creep within the story. It also focuses the "sizing" conversation about a story on aspects of the story itself, rather than on things we can't be expected to know right now.

Because a mile is always a mile, we have some notion about how fast we can run it, under reasonable circumstances. Basically, we would expect a mile to be run in a matter of minutes, not hours or days. This type of expectation is also true for teams and this is why I sometimes think about story sizes in terms of effort, the 2-3 people for 1-4 days concept. I have these expectations based on teams I've worked with; however, your mileage may vary (pun intended).

I know that these two notions of size are inconsistent with each other: intrinsic difficulty of problem versus amount of effort. Sorry 'bout that... but one of the strengths of human beings is that we're able to keep two perfectly good, though incompatible, ideas in our head at one time.

This is one of those times.

We have a team working in iterations, so we need the team to work on stories that are the right size (in effort) so that they get them "done" in a single iteration. We must have faith that stories that are understandable (intrinsic difficulty of XS thru XL) can be done in a single iteration. If they can't, the team must adapt: lengthen the iteration (usually the wrong choice), have more people swarm on the story (not a bad choice), or start estimating with different thresholds for difficulty estimates (what usually happens automatically).

**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

I have never found this issue to be a problem. A fear, yes. A risk, yes. A good multi-beer conversation, yes. A problem, no. People adapt. Agility happens. And it just works out...

Thanks, Dan ;-)  
[dan@danube.com](mailto:dan@danube.com)

## When a Mile Isn't a Mile

Comment Submitted by Michael James on May 20, 2008 at 4:53 p.m.

Dan and I had some follow up discussion that's worth posting here.

For me there's nothing analogous to a "mile" in the problem space, only in the solution space. And the solution space is more than one-dimensional because different teams will invent different solutions paths to the same problems. This is independent of how fast they can run down those paths (velocity). A team that's generally slower than another might know a shortcut for a particular story. This is part of why Dan and I insist on teams estimating their own work.

One team might implement the solution using third party libraries, another with some regexp magic, someone else by coding the whole thing from scratch. The "intrinsic difficulty" of a problem is a nonexistent thing which (in my opinion) isn't as useful to consider without a gut feel guess about the solution.

However, this doesn't mean we need a detailed plan for the solution to give our gut feel effort estimate. In fact there's some evidence that overanalyzing leads to *\*worse\** guesses than under-analyzing.

To make matters worse, most of the work we do creating products isn't the creative design and programming we enjoy. A lot of it is:

- talking to customers to tease out what they really need,
- getting the development tools and libraries to do what they're supposed to do,
- finding workarounds for limitations of other people's stuff we can't change (e.g. database servers, Web browsers, hardware),
- chasing down phantom regression failures from our imperfect test harnesses,
- etc.



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

In my experience the variation of these things from story to story comprises the bulk of the differences in estimates! A “complex” story that can be implemented mostly on the server side could take less time and effort than a “simple” story that entails getting a page to render nicely on multiple buggy versions of Internet Explorer. Some work exacts a higher effort cost from the team by virtue of being more annoying to do.

In practice, Dan and I agree it comes down to gut feel. Your team’s intuition may not be enough, but it’s all you’ve got.

--mj

## You’ve Changed My Mind

Comment Submitted by Dan Rawsthorne on May 20, 2008 at 7:42 p.m.

MJ, you’ve made me change my mind. I still think that “a mile is a mile,” but now agree that the mile we’re talking about is a mile in the idealized problem space. That is, we ask “how far is it, given a ‘normal’ definition of done, a good codebase to work, and your team working at its best?”

Maybe we’ve finally reached agreement... I hope so.

Dan ;-)

Dan Rawsthorne, PhD, CST  
Transformation Coach  
<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