

Estimating Business Value

Ken Pugh
kenpugh.com

Creating software is about delivering business value. Without some measure of business value, it's hard to determine whether the software has any. For several years, I've presented a session on estimating business value to local user groups and national conferences. My new book, *Lean-Agile Acceptance Test-Driven Development: Better Software through Collaboration*, includes a section on estimating business value. Here are some ideas from that book and others accumulated over the years.

Business Value

Every feature in a system should have business value. Business value can come from numerous areas, such as:

- Increased revenue (sales, royalties, fees)
- Decreased expenses
- Using less resources
- More efficient use of resources
- Customer satisfaction
- Product promoters / satisfiers/ detractors
- Staying in business
- Avoiding risk

Business value in some areas is easy to quantify. It is straight dollars. However in other areas, quantification is more difficult. But without some way to compare these "apples and oranges" it is hard to determine which stories should have higher priority. For example, should a story that saves \$10,000 be prioritized over a story that gives some customers more satisfaction?

One way to compare is to assign a relative business value to every story. The customer unit has the responsibility to assign business value. The business value does not have to be an exact measure. It's just relative. If the \$10,000 savings seems to be the same as giving customers more satisfaction, then the stories have the same business value.

Relative determination may be made by putting the stories up on the wall one at a time. If a story has more business value than another story, place it above that story. If less value, then put the new story below, or if about the same, on one side. If a story fits between two stories, move the stories and make space for the new one in-between. For example, in Figure 1, Story Two has the highest value, Story One and Story Three are about the same, Story Four less than those, and Story Five seems a lot less.

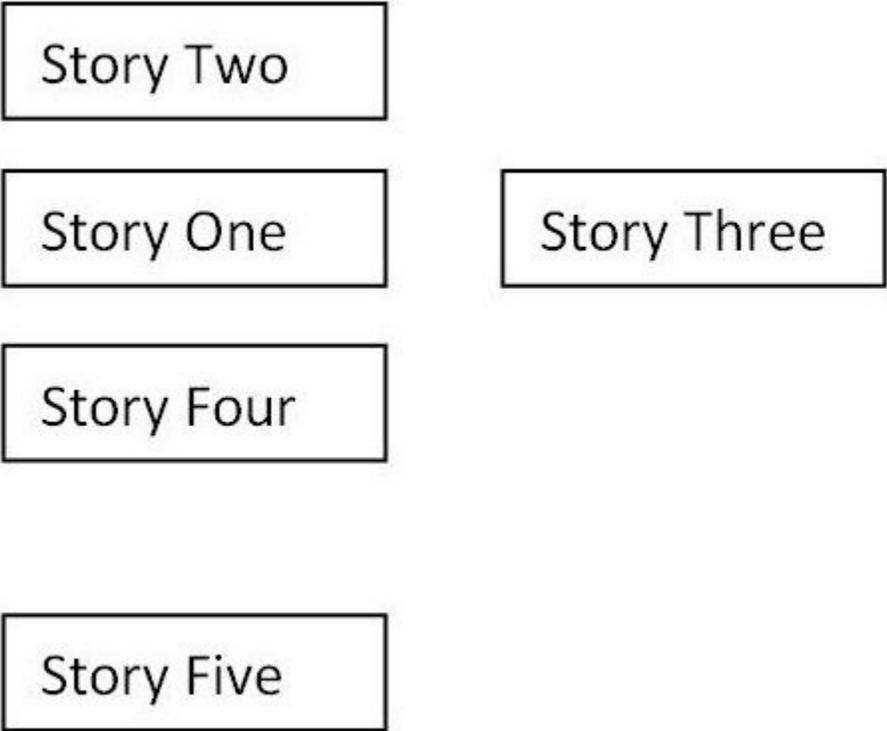


Figure 1 Relative Story Placement

You can use a Fibonacci series (1, 2, 3, 5, 8, 13 ...) or a power-of-two (1,2,4,8,16,32, ...) series to assign stories in each row a value. (Figure 2). Where you start with the numbering is not important, as long as you assign the same numeric value to equivalent stories in the future.

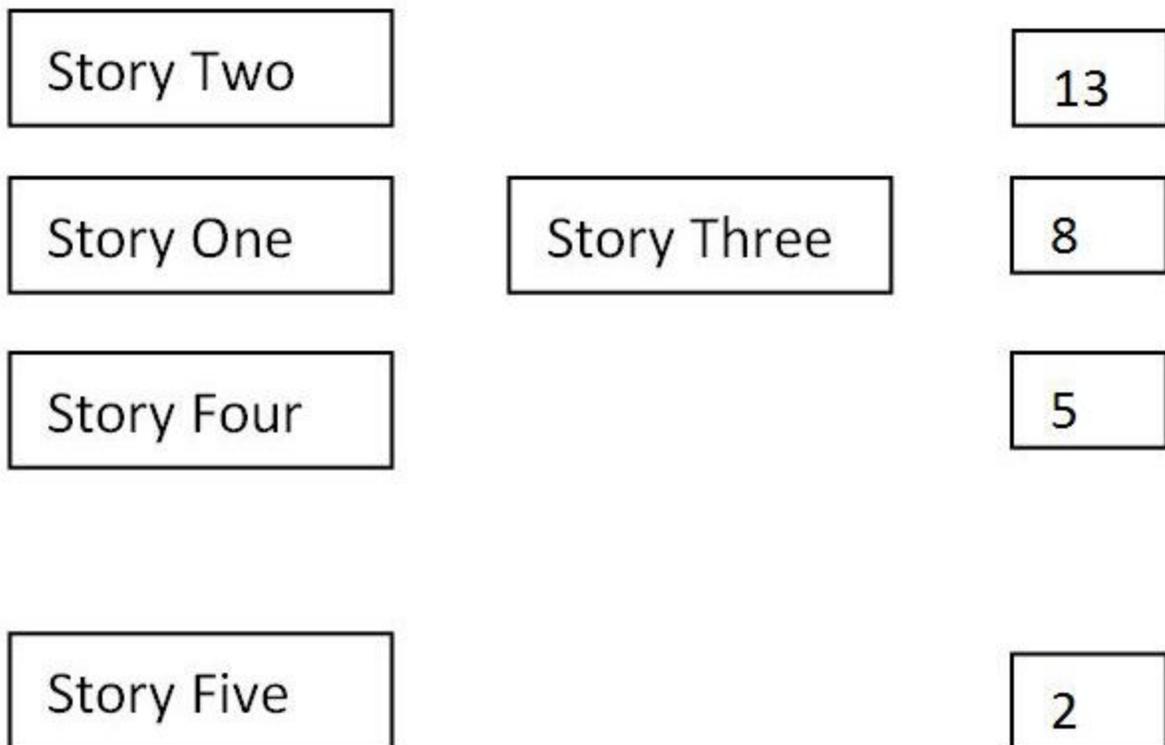


Figure 2 Relative Story Values

Now periodically, you can measure the cumulative business value of the delivered stories (Figure 3). Since the key in agility is to deliver business value quickly, such a chart provides feedback for the developer and tester units to see how well they are doing. Presenting a business value chart prior to or in lieu of a burn-down/burn-up chart can focus attention on this critical measure of software delivery. Since the business values are determined by the customer and are not accumulated until the corresponding story is accepted, this chart can present a more accurate picture of the value the software is currently providing to the company.

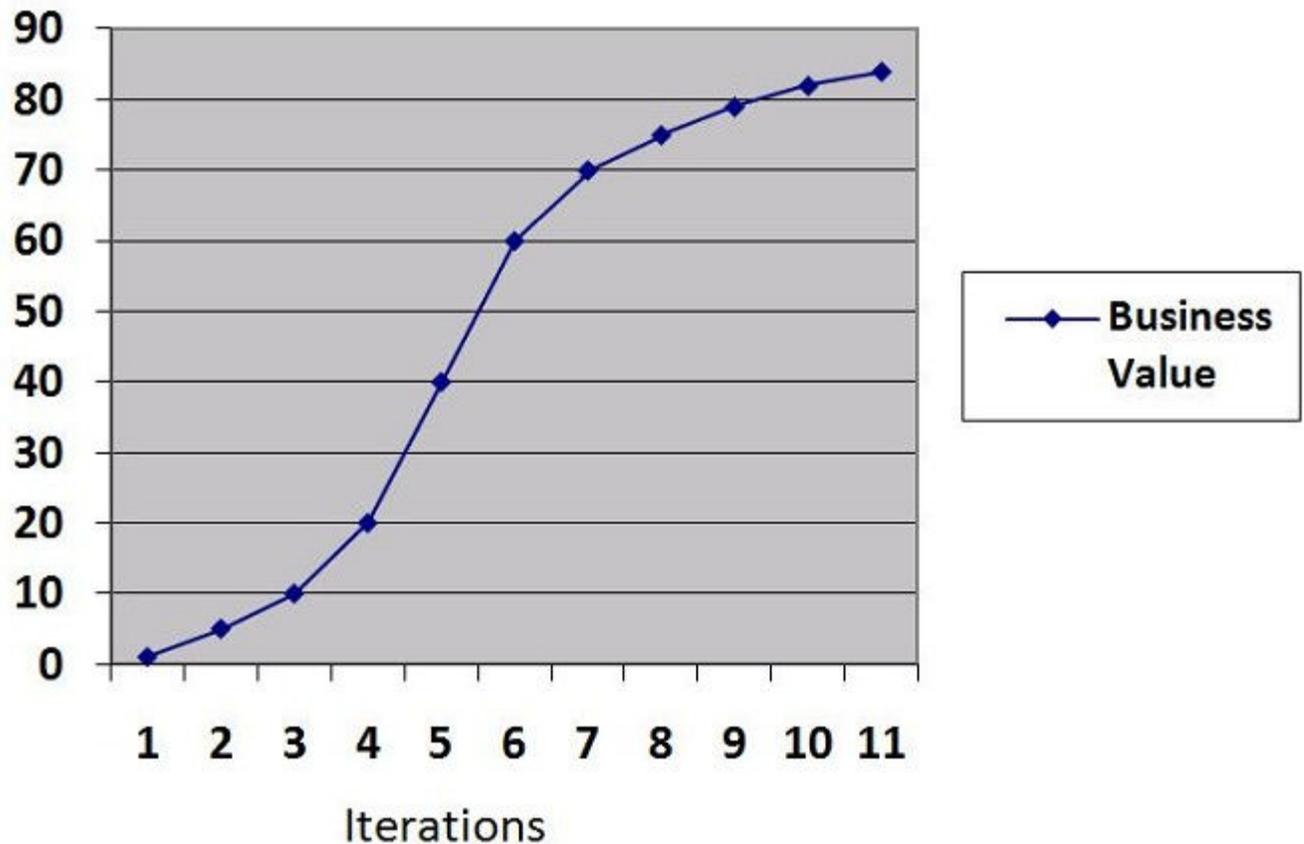


Figure 3 Business Value Chart

The developer and tester unit can estimate story effort in story points using the same technique as for business value as an alternative to story point poker.

A story's business value estimate divided by the estimated story effort yields "Rough Return on Investment" or "Bang for the Buck" (BB). It is a rough guide to the relative return on investment. It can provide another input when the customer unit makes decisions as to what stories to develop.

If stories are developed in a BB sequence (high BB valued stories completed first), then the return per effort spent will decrease over time (as shown in the later iterations in Figure 3). At some point, this return will decrease below that of another project. The current project should be terminated. Not because the project is a failure, but because it no longer provides the best use of resources.

For financial-oriented individuals, the total return on a project (dollars saved or revenue earned) can be divided by the total business value for all stories. The result is the dollars per business point.

Developer Stories

If a story is estimated to take a long time to implement, it can be broken into smaller stories. However, the customer unit must be able to realize business value from one of the smaller stories. If they cannot, then the smaller stories are developer stories. They exist to make it easier to track units of work. There may be technical value in completing a business story, such as having a new display component or a new service. When the technical value story is incorporated into a business story, then the business value is credited.

A developer should create an acceptance test for every developer story, such as cross-team requests for user interface components or web services. Depending on the story, it may not be possible to create a specific test until there is collaboration with the implementer, but you should have at least some criteria to be able to know when the story is done.

Is It a Technical Project?

There are customer stories that require a lot of framework development. This translates into a lot of developer stories. If you have this situation, you have an embedded technical project. There should be a "technical product owner" who can break a customer story into "technical customer stories", each with acceptance criteria. And if the framework is vague enough, so these developer stories cannot be created, then you should break this framework out into its own technical project with the customer project being a "test bed" for it, rather than vice-versa.

Summary

- Estimate business value for stories
- Track cumulative business value for delivered stories
- If necessary, break stories into developer stories for tracking
- Business value for a story is not achieved until all associated developer stories are completed