

ScrumWorks® Pro Tutorial: Two-level Estimation System

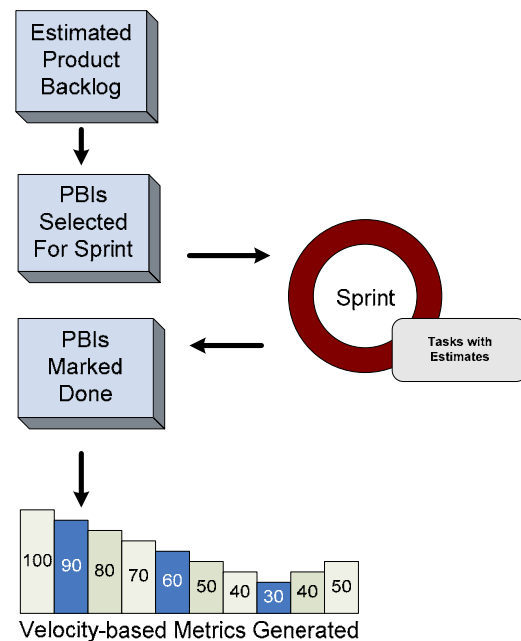
Question:

“Why does ScrumWorks Pro have one estimate for Product Backlog Items (PBIs) and a separate estimate for Tasks? Shouldn’t task decomposition result in task estimates rolling up to the PBI level?”

This tutorial addresses the topic by introducing the concept of Velocity-based metrics and why two levels of estimation are appropriate on Scrum/agile projects.

The Big Picture: Why Two Levels of Estimation?

At its core, Scrum is an “empirical” process: It relies on frequent inspect-and-adapt cycles to update the plan according to the latest information. In other words, Scrum is not a predictive, speculative process in which management attempts to anticipate all possible contingencies and plan for them in advance. In Scrum, current Sprint work is decomposed to granular detail (called “Tasks”), while more distant, uncommitted work on the Product Backlog (“Product Backlog Items”) is left at an intentionally high level. As such, Scrum provides for **two levels** of estimation; one for work (Tasks) to be completed within the current Sprint and one for more distant Product Backlog Items (PBIs).



What Purpose Do Product Backlog Items Serve?

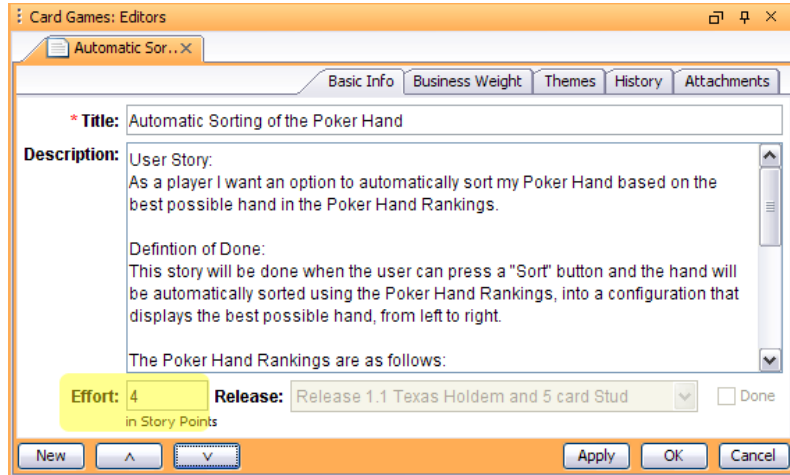
Scrum discourages Teams from engaging in a detailed, upfront requirements analysis phase. Because the goal of Scrum is to build a product incrementally, feedback generated during each Sprint will help shape future requirements and refinements throughout the development lifecycle. In Scrum, PBIs usually express a product goal or feature. PBIs are prioritized by the Product Owner (using drag-and-drop in ScrumWorks Pro). Typically, the lower the PBI’s priority, the less precisely it is detailed. PBIs are therefore used to capture feature-oriented requirements and specified in detail on-demand without spending much time in upfront requirements analysis.

Releases are scheduled and tracked at the PBI-level because PBIs represent the goals of the product or Release cycle. Teams commit to PBIs in Sprint Planning without Task-

level decomposition.

What Purpose Do Product Backlog Item Estimates Serve?

Because Releases are scheduled and tracked at the PBI level, Scrum provides for an estimation system that enables teams to estimate distant, uncommitted work with a low degree of precision. This quick-and-dirty approach avoids “analysis paralysis” pitfalls and accurately reflects the general uncertainty surrounding the work in question.



By providing a way to quantify the size or effort outstanding (however imprecise), Scrum provides for valuable long-range metrics such as:

- **Velocity:** The rate at which the team completes PBI effort on a Sprint-by-Sprint basis. For example, if a Team finishes 25 PBI effort points during a Sprint, the Team’s Velocity for that Sprint is 25.
- **Rate of Change:** The rate at which things are changing on the Product Backlog due to new work being introduced or the re-estimation of existing items.

These two metrics can be used to forecast a Release completion date (see the “Tracking and Reporting Progress” tutorial). Notice, however, that they do not rely on granular Task-level analysis.

What Purpose Do Task Estimates Serve?

While PBIs describe “what” is being built, Tasks serve to detail “how” the PBI will be achieved. In other words, tasks describe the Team’s implementation plan for the PBI. A separate estimation scheme is appropriate because Tasks describe granular work (usually on the order of a few hours). In fact, Scrum recommends using “ideal engineering hours” for Task-level estimates.

The screenshot shows a software window titled 'Card Games: Editors' with a sub-window 'Automate I.'. The form contains the following fields:

- Title:** Automate Integration Test
- Description:** Automate an integration test to verify that this is fixed.
- Point Person:** (unspecified)
- Estimated Hours:** 4
- Status:** In Progress
- Backlog Item:** The A in all Ace cards is Upside Down

At the bottom of the window are buttons for 'New', 'Apply', 'OK', and 'Cancel'.

Shouldn't the Sum of Task Estimates Replace Earlier PBI Estimates?

The short answer is no. While it may be tempting to replace earlier PBI estimates with the sum of Task estimates, this would prevent the calculation of the Velocity metric at the end of each Sprint. Velocity is the measure of a Team's ability to complete PBI effort in a Sprint. Therefore, a historical record of a Team's Sprints and corresponding Velocities is extremely valuable for planning future Sprints and forecasting Release dates.

For example, if a Team takes on 50 PBI effort points worth of work for Sprint A and completes only 40 points during that Sprint, the Team may be inclined to take on less than 50 points in future Sprints.

What Units Are Appropriate for PBI Estimates?

There are two main schools of thought on this topic. The original Scrum books recommend using "Ideal Team Days" as a unit of estimation. While this unit of estimation is appropriately "high level" and reflects a degree of imprecision, it is still a chronological unit and many organizations have found it tempting to convert back and forth between hours (e.g., 5 hours per Ideal Team Day), leading to analysis paralysis during PBI estimation sessions. Today, most organizations use an amorphous, relative estimation scheme like "Story Points" to further disassociate Task estimates from PBI estimates. Please refer to Mike Cohn's book [Agile Estimation and Planning](#) for more on PBI estimation techniques.