

WSJF: A LEAN-AGILE MODEL FOR PRIORITIZING A PRODUCT PORTFOLIO

Getting more bang for the buck.

Leaders in every enterprise must face the reality that there is never enough money, time, or people to change their products to meet the demands of customers, regulators and many other stakeholders. They must choose. But many lack a formal or structured way to make these choices. Those having a formal method usually do so at a relatively late stage in the decision-making process, requiring that someone develop a business case or cost-benefit analysis prior to making a go/no-go decision. But having developed a fair number of business cases, I can attest that the effort to create one is not trivial. It's not easy; especially when the list gets very large. For example, I read that executives at Target Corporation recently prioritized a list of nearly 800 projects.

How can we, as leaders, evaluate and prioritize a large list of change requests, enhancements, or project requests without investing undue amounts of effort but still feel confident that we made the right decisions?

The answer is Weighted Shortest Job First (WSJF).

Leaders and their enterprise can realize immediate benefits from adopting the WSJF model to evaluate and score work items in its portfolio of potential investments in products and services. Some of these benefits include, but are not limited to:

- A structured way to engage a broader number of representative stakeholders in the process for deciding what gets done and what doesn't
- Greater transparency into the criteria used to evaluate and decide on a request
- Ability to measure the relative holding costs and benefits of the work that is in our portfolio
- Technology leaders get visibility into the work that is being prioritized and can act upon resource levels much sooner than before
- Getting to a decision earlier so that efforts can be focused on the most valuable things first



I helped the Human Resources area of one of the largest global banks implement the WSJF method to evaluate and prioritize their portfolio of change requests, enhancements and project requests. The adoption of the method was straightforward and the benefits were readily evident.

Don Reinertsen first described the WSJF model in 2009 in *Principles of Product Development Flow: Second Generation Lean Product Development*. He described it as the cost of delay divided by job duration. Cost of delay is the cost to the enterprise of waiting, or not doing the specific change or request. Job duration is the amount of time it takes to complete the job. It is a measure of “bang for the buck.” However, Reinertsen placed greater importance on the numerator – cost of delay. In *Principles of Product Development Flow*, he said, “If you quantify only one thing, quantify the cost of delay.”

Dean Leffingwell refined Reinertsen's model in 2012 in the Scaled Agile Framework (or SAFe), which is now the most widely-adopted method for running large, multi-team Agile efforts with cross-dependencies. In SAFe, the cost of delay is made up of three factors:

- User-Business Value – the relative value of an item to the business or customer
- Time Criticality – Does the cost of this problem increase over time if we do not act? Is there a deadline? Are there penalties for non-compliance that take effect? Are volumes increasing or remaining steady?
- Risk Reduction-Opportunity Enablement – Does this reduce risk, or allow us to take advantage of opportunities not available to us previously?

Also in SAFe, since we cannot easily estimate a job's duration so early in the decision-making process, we use job size as a proxy. These factors are scored using the same modified Fibonacci sequence (1,2,3,5,8... 40, 100) used in Scrum's planning poker. So, WSJF in SAFe is:

$$WSJF = \frac{\text{Business Value} + \text{Time Value} + \text{Risk Reduction Opportunity Enablement}}{\text{Job Size}}$$

A work item with a higher WSJF gets more priority than those with a lower one because the ratio of cost of delay to its size is greater.

At the bank I mentioned, I facilitated sessions in which a group of six to ten senior leaders in the HR organization scored each of the requests for the three factors in the numerator of WSJF. A technology manager facilitated sessions with technical teams to score those requests for relative job sizing, the denominator. In a matter of hours, we evaluated and prioritized over 250 items, consisting of things that were very large and very small.

One caveat: we did not rank order and prioritize projects and requests blindly per WSJF. Sometimes an item with a low WSJF ratio would get a higher priority than one with a high WSJF ratio; we considered many other reasons. But WSJF gave us a starting point and a frame of reference.

The senior leaders in HR appreciated that they were involved in making rational decisions that were previously just hallway conversations. Their participation and enthusiasm in the sessions were high. The technology managers now had a prioritized work list that remained stable. They could start at the top of the list and work down until they reached their capacity for a release cycle. And the business now had better information to evaluate the contents of a release, and better justification for additional funding. Developers and architects appreciated the opportunity to give a relative sizing score to items in the portfolio (which had been ranked by cost of delay beforehand). The feedback they gave me was that now they would work on things that the business deemed truly important, and not someone's pet project.

All of this was possible before anyone had to create a work breakdown structure, estimate tasks, or prepare a business case. In the end, if a business is grappling with how to prioritize its abundance of work requests, a great place to start is with the WSJF model.



This TIP was written by David O'Brien, who specializes in Financial Services and Agile Frameworks. David welcomes comments and discussion on this topic and can be reached at david.obrien@trexin.com.