

What is the optimal batch size?

Some jobs are too expensive to perform for every individual item. There are costs incurred by delaying delivery of items. What is the optimal number of items to perform as a batch, or what is the right amount of time for regular delivery?

The basic concept – balance per item cost versus delayed delivery cost

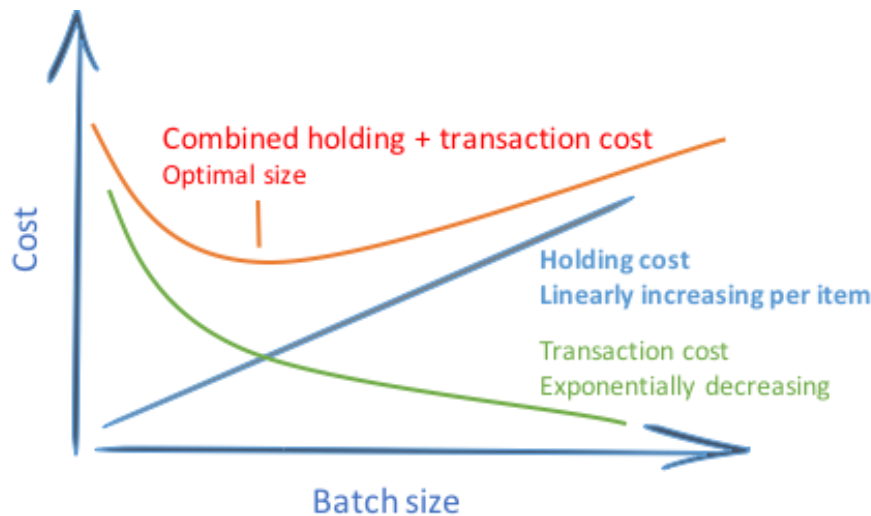
There are two types of cost to balance -

1. The costs of setup and execution of the process (per unit) = Per Item Transaction costs
2. The costs of holding back items from delivery e.g. delayed feedback = Per Item Holding costs

Optimal batch size is the number of items that minimizes the total combined costs. There are various types of costs that need to be considered. Delayed feedback, storage cost, decay in value.

If setup and execution costs of a task is high, it gets exponentially more efficient (as a per item cost) for higher batch sizes. Holding costs are often more linear (it is assumed all items deliver equal value, and that any item could give valuable feedback). Brainstorm all the transaction costs and holding costs. Think about how they respond to different batch sizes.

Most importantly: Just DON'T be near either end. That is where total cost escalates. It is almost always safe to halve the batch size putting you in the flatter section of the total cost U-curve. You will know when batch size gets too small, costs and effort escalate enormously.



References and more information

Google for “Economic Batch Size” or “Economic Order Size” in production/inventory management.
Read “Principles of Product Development Flow” by Donald Reinertson, Chapter 5 – Batch Size
Read “An optimal batch size for a JIT manufacturing system” Lutfar R. Khana, Ruhul A. Sark

Example: Performance Throughput Testing – How often?

Some projects require performance testing. This type of testing can be labor intensive to setup and time consuming to execute and analyze. Some teams only perform this late in a project causing late feedback on poor performance architecture and code. The right batch size in time and amount of features to test in one batch is important to consider. Balance the costs -

Transaction Costs

Setup time
Blocked testing environment

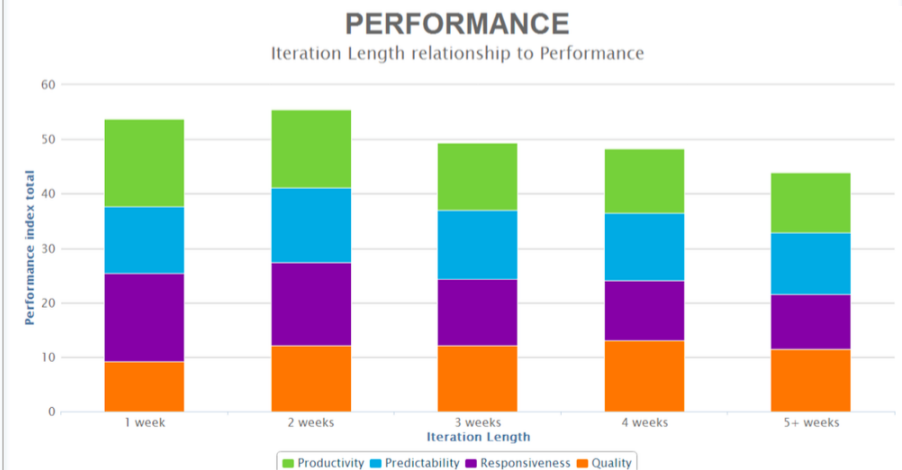
Holding Costs

Delayed feedback on architecture decisions
Harder to find the culprit of any regression
Late knowledge release will be delayed

Tip: Do twice as often as you do now. Once you achieve that rate, halve that again. Stop when the disruption of performing these tests becomes intolerable

Study: Sprint Length – How many weeks are optimal?

Two weeks is the most commonly used sprint cadence, but is it optimal? Turns out, “yes, mostly.” In a study, Larry Maccherone then with Rally Software performed looking at approximately 10,000 projects. Measured with a balanced set of metrics across four different categories (quality, predictability, performance and responsiveness), two week sprints performed best overall. However, teams with one week sprints had better responsiveness scores, and teams using three-four weeks sprints had higher quality scores. The optimal length might depend on the trade-off between quality and responsiveness in your context. Tip: Start at two weeks and experiment over time. Go lower if responsiveness is important, but keep an eye on quality due to moving too fast.



<http://maccherone.com/publications/KEYNOTE-MetricsAndInsights-Maccherone-0604-1245PM.pdf>