

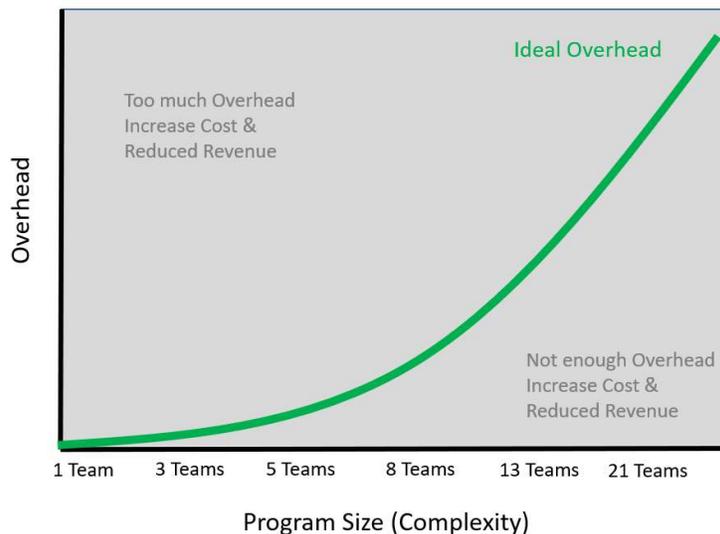
Overhead and Scaled Agile Development

Think twice before adding new processes

Scaling agile development is a long road rife with opportunities for failure, and many experienced agile coaches will recommend programs avoid scaling unless absolutely necessary. The hesitation to scale is valid because the addition of more development teams results in a subsequent addition of more overhead. This extra overhead comes in the form of non-direct value activities associated with the coordination and management of the group. Status meetings, progress charts and layers of approval are examples of non-direct value activities, as opposed to the direct value-adding work of creating working software.

This overhead comes at a cost to the program, but it doesn't necessarily mean the cost isn't worth it. Striking the right balance of overhead, especially for a growing program, is key to optimizing the program's ability to deliver value to their customers. When additional teams are added without focusing on coordination and alignment, your program will struggle and subsequently drive up cost. On the other hand, if a program spends too much time on overhead activities, teams will be overburdened with non-development work, which will also drive up the cost of development.

The below chart is a simple visual we use to explain the relationship between scaling and overhead. The green line depicts the ideal overhead for a program. Since each additional team has to coordinate with all the other teams, the line exponentially increases as the number of teams in an organization increases.



The intent of this chart is to visually help programs consider the unintended consequences of scaling their development program. One could point to the left side of the chart to show that adding layers of process on top of a few teams is likely not worth the cost. It

can also illustrate to stakeholders that adding more teams may cost more in overhead than will be gained in productivity.

The other benefit of this visual is to help determine if the value gained in new processes is greater than the cost. Answering this question is the responsibility of the program manager, or the Release Train Engineer in a SAFe Framework. Program managers should continually think about the unintended consequences of changing or adding new processes.

Programs should first determine if they are above the line (too much overhead) or below the line (not enough overhead). Understanding where a program sits on the chart can usually be seen through leading indicators. A decrease in productivity, low morale, and over-booked calendars are clues overhead may be too high. Communication failures, missed dependencies, and a trend of reporting inaccurate program status are indicators overhead may be too low.

The next step is to cross-check this information with the number of teams on a program. A program with three teams has a much smaller need for overhead when compared to a program with eight teams. This information, combined with an initial assessment of where the program currently sits on the overhead chart, helps build situational awareness and can tailor a response to new process proposals. If a program is small, and the overhead is high, then a program manager should look to reduce overhead. The same is true for low overhead on a large program where more process can increase productivity.

The key takeaway from this visual is the importance of understanding the program makeup and processes in an integrated way. Adding new processes in response to issues is a common practice to reduce risk. However, program managers should also understand that new processes come with a cost. The next time your program is faced with the decision to add additional overhead, take the time to evaluate your current overhead state. This will reduce the chance of making problems worse and can substantially increase the effectiveness of the teams doing the direct value-add work. Hope it helps!