

A Case Study in Implementing Agile

Everyone is looking for a competitive edge. It's not surprising, then, that solutions promoting decreased time to market and increased productivity would be appealing. As more organizations begin to implement agile into their software development practices, it is logical to wonder whether agile development methodologies truly differ from traditional waterfall methods and what quantifiable advantages may be realized by adopting agile.

To tackle these questions with some objective numbers and data, software estimation company Quantitative Software Management Inc. conducted a case study for a large technical business organization (that wishes to remain anonymous). Initially a waterfall shop, this company attempted to adopt agile on a small scale in 2010. The results were less than optimal, primarily because the company lacked the necessary infrastructure and organizational mind shift necessary to truly embrace the principles of agile in its environment.

In 2011 it made a second attempt, this time using a more integrated approach. To start, it had organizational support and buy-in from senior management and key stakeholders. The process consisted of conducting an initial baseline assessment of development systems using an integrated set of tools and methods that would support agile and help manage the backlog, as well as a training component.

What QSM found from this instance (and others) was that successfully adopting agile may take upfront investment of both time and resources in order to realize optimal results.

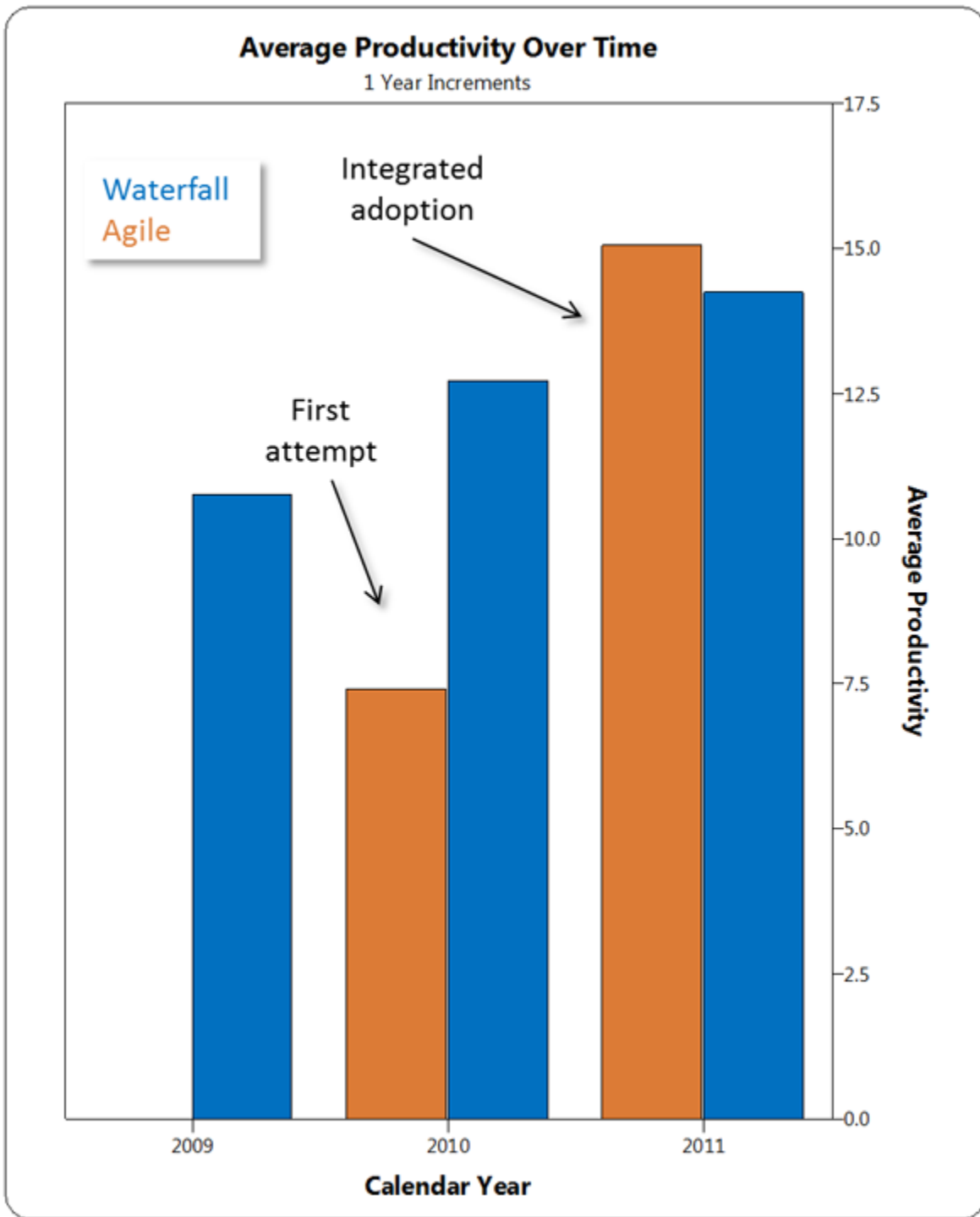


Figure 1: Average Software Development Productivity for Agile and Waterfall Methods over Time

Figure 1 shows a comparison over time between the average productivity of agile and waterfall projects. Productivity was measured in index points, a calculated proprietary QSM unit that ranges from 0.1 to 40. They allow for meaningful comparisons to be made between projects and account for variable software development factors such as management influence, development methods, tools, experience levels, and application type complexity. The projects developed using waterfall methods increased their average productivity ratings between 1.5 and 2 index points per year, which is fairly typical of this organization’s industry. As organizations improve their software development techniques and become more efficient, they also tend to improve their

productivity over time.

You can see that the projects developed using agile methods did not have the highest productivity ratings when first adopted in 2010. However, by 2011, productivity not only increased dramatically—by 7.5 index points—but also surpassed the average productivity of the projects using waterfall methods.

To put some additional numbers around that finding, we modeled the software development lifecycles of typical agile and waterfall projects of the same functional size and staffing to better understand the differences between the two methodologies over time. One of the first, glaring observations was that agile methods utilize a much higher degree of overlap between high-level design and construction phases than waterfall methods: 97 percent versus 30 percent, respectively, as shown in figure 2. This makes sense, as agile methods leverage an iterative approach to release planning and delivery.

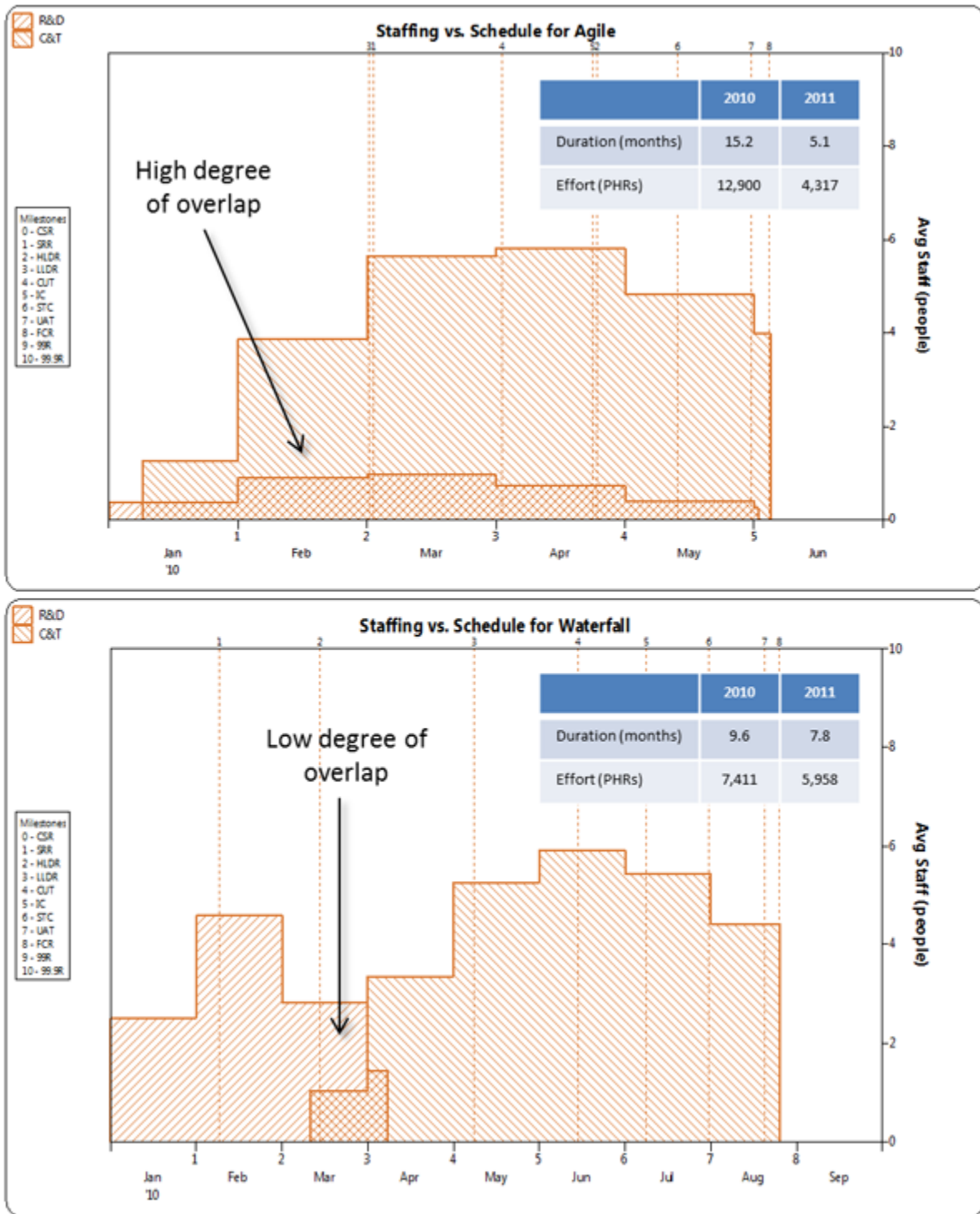


Figure 2: Staffing Profile over Time for Waterfall and Agile Methods

The second observation was the slope of the learning curve. When adopting agile, development teams often have to learn a whole new methodology of defining requirements, writing code, and concurrent testing. The effects of this can be seen in the schedule duration and effort expended.

In 2010, when agile methods were initially adopted, the projects using waterfall methods delivered 58 percent faster and used 74 percent less effort. This equates to about \$550,000 in upfront costs for adopting agile when using a normalized labor rate of one hundred dollars per person hour.

However, 2011 saw a shift after the integrated agile adoption. This time, agile methods achieved 34 percent faster deliveries and utilized 27 percent less effort than waterfall methods, resulting in a cost savings of \$160,000 per project.

From this we can see that using an integrated management approach to adopting agile yielded far more beneficial results than simply changing the technical development process by itself. However, realization of these benefits required an initial time investment in order to obtain organizational buy-in and allow adequate time for the necessary learning curve.

Knowing that adopting agile can be a lengthy process, it may be valuable to examine whether it is a worthwhile endeavor for an organization. Agile methods thrive when developing smaller, less complex systems, but now it seems that people want to push the limits of what agile can do. Can agile scale to larger organizations and more complex systems? Is there a “sweet spot” in which agile can realize the greatest benefits?

To answer these questions empirically, we compared the average trendlines of the agile and waterfall projects in a variety of areas, including duration, effort expended, staffing, and productivity. Because the projects used different sizing methods, we normalized the project sizes into a common standard called implementation units (IU) to allow us to make an apples-to-apples comparison. What we found was that there was a scaling “sweet spot” for this organization, whereby projects larger than 12,000 IU realized greater benefits from using agile methods in terms of time to market, cost, and productivity. Projects that were smaller than 12,000 IU achieved better results using waterfall.

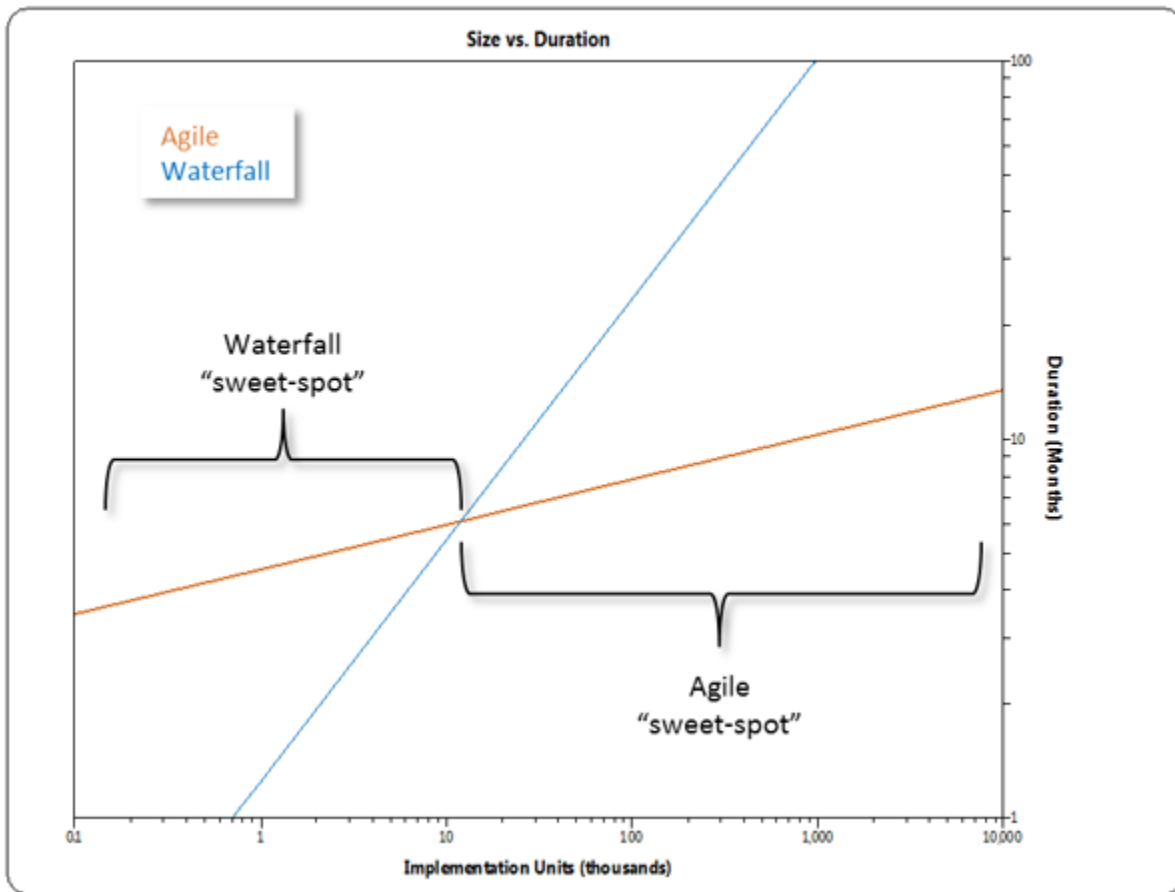


Figure 3: Average Durations for Agile and Waterfall Projects

This finding, though entirely based on one organization’s environment, demonstrates that agile has the potential to scale to larger project sizes and enterprises. While this may be counter to commonly held beliefs that agile methods should only be used for developing small software systems, we’ve seen similar findings with other large enterprise organizations as well, suggesting that this case study is not merely an outlier. That said, this finding is not generalizable to the software industry at large. We have seen other organizations achieve greater benefits from using agile methods for smaller sized projects, indicating that the “sweet spot” varies by organization.

Agile is not a silver bullet, and the decision to use agile methods should be entirely situational. Within this organization there were situations in which using agile resulted in the greatest benefits and others in which using waterfall was still the best option, as shown in figure 3, suggesting that waterfall is still a relevant development method and should not be abandoned. When deciding between adopting agile or sticking with more traditional methods, my suggestion would be to use the development method best suited for the project’s intended environment.

This case study serves as an example of how adopting agile can be extremely beneficial to an organization, as long as situational factors are considered. If adopted impulsively without the appropriate cultural modifications and organizational support, agile—or any new development methodology, for that matter—has the potential to negatively impact organizational productivity.

Adopting a new development method is a strategic, long-term investment rather than a quick fix. Before making drastic changes such as those needed to properly implement a similar change in development practices, invest time at all levels of the organization to fully understand the task at hand and assess how long it will take to reap

the benefits and achieve the expected return on investment. As in this case study, making deliberate, fully formed decisions will ultimately lead to better outcomes.