

The QSM Agile Round Table

How can agile practitioners find the right “fit” for parametric estimation?

As new ways of thinking and doing are introduced, businesses must adapt and grow. To remain relevant, technology companies in particular face the challenge of adopting the latest and greatest process methods and tools quickly, without “breaking” the methods and processes that have worked in the past. Implementing shiny new methods is often high on executives’ priority lists, because they are keen to realize the espoused productivity and quality benefits.

For well over a decade, agile software development methods have been adopted by software organizations across the globe. Whether in commercial software companies, systems integrators, IT departments of large financial companies, or branches of state and federal government, agile methods and communities are well established and continue to grow. QSM has worked with these types of software organizations for more than 35 years to establish data-driven, [defensible estimation](#) and lifecycle management practices as the foundation of quality software projects and products. Does the adoption of agile practices mean that formal estimation practices are obsolete?

As time goes by, software technologies have come and gone, each offering solutions to a myriad of problems encountered across the industry. But over these same 35 years, none of these turned out to be the proverbial silver bullet that destroyed all obstacles. I remember when personal computers first became available to the general public. Visions of how home computers would revolutionize our lives were widely exaggerated. Of course, many ideas such as video chat have been around for some time, and the quality of life has improved. But home computers have yet to fully replace any aspect of daily living.

Agile development is different from previous software process innovations, like object-oriented programming, because it is a holistic approach to improving software development embodied in these four values, expanded to twelve principles in the Agile Manifesto:

- **Individuals and interactions:** self-organization and motivation are important, as are interactions like co-location and paired programming
- **Working software:** working software is more useful and welcome than just presenting documents to clients in meetings
- **Customer collaboration:** requirements cannot be fully collected at the beginning of the software development cycle; therefore, continuous customer or stakeholder involvement is very important
- **Responding to change:** agile methods are focused on quick responses to change.

It is tempting to believe that new methods will so revolutionize software development that tried and true practices like project management and estimation are no longer needed. Agile methods challenge established practices and precipitate the need for change across the entire project lifecycle. The toughest challenge in process improvement is change management. Not only is change uncomfortable, but the nature and magnitude of the change can be downright frightening. The big question organizations struggle to answer is “*Which established methods and practices are still relevant with agile?*” We at QSM, along with many of our customers, have grappled with the relevance of [top-down, scope-based estimation](#). Our clients have realized the value of parametric estimation and lifecycle management: project portfolio estimation for annual budgeting and

resource demand planning, project and program estimation for responding to proposal requests, estimate validation for vendor management, [performance benchmarking](#) and process improvement. But, they struggle to communicate with agile teams who have their own way of estimating, and rarely take a “project” view of the product development process. While agile development teams share common goals with SLIM users, because they [measure size](#) in epics, features, user stories, and story points, calculate velocity, and track delivered product value, customers have found some challenges communicating what is common and what needs to be adapted.

The **QSM Agile Roundtable** was formed to provide a platform to brainstorm the role of estimation in agile environments, and chart a path toward better understanding for all stakeholders. A mixture of long-standing and newer customers shared their questions, challenges, and experiences to answer the big question, and effectively communicate the relevance and benefits of scope-based estimation. From December 2016 through June 2016, the QSM Agile Round Table met one or two times each month. Consistent with the legend of King Arthur, there was no head. Participating companies shared their questions and concerns. The group as a whole set the agenda for each gathering. QSM facilitated the discussions, and provided technical insights about the SLIM model. The talking stick was available to all.

So, what is the answer to the big question? Is scope-base estimation relevant with agile? We agree that it is, and want to share both the strategies and methods that work for our clients, and the insights gleaned from QSM Agile Round Table discussions. This writing is the first of the QSM Agile Round Table series of publications that will present specific concepts and practices that connect SLIM and agile, creating common ground for the benefit of all. Here is a sample of the questions to be answered:

- **What are we estimating?** – SLIM estimates projects, so what is a “project?” Should we estimate at the team, release, or portfolio level? Teams may not need parametric estimation, but companies do. Agile development practices do not eliminate the need for year-to-year corporate planning, budgeting, and resource management. What is the right fix, and how does the language or terminology need to change to communicate the common ground?
- **How is size measured?** – Measuring scope is common with SLIM and agile, but what are the appropriate size units? Not only for estimation, but in performance benchmarking as well, we need to map agile metrics to traditional project level core metrics. How can we convert between measures to take advantage of the metrics we have? Many clients use Function Point sizing. Can we still use Function Points?
- **What metrics should be collected?** – How do we use team velocity in conjunction with SLIM Productivity Index (PI)? Since Story Point counting rules vary across teams, how can we use this and the associated velocity measures? Do units of measure change as we progress from project concept to completion, e.g., can we estimate in Function Points and count actuals in Story Points? Our team does not report effort by task, so how will I properly account for actual effort spend?
- **How is SLIM configured for agile methods?** – What amount of time and effort are attributed to story writing and architecture, versus development and test? Should all sprints be contained in the main build? Organizations adopt agile methods with varying levels of both rigor of practice and consistency across the organization. What project environment settings are used to model pure agile and hybrid approaches? Are nonlinear relationships and staffing buildup curves still applicable? What resources are included?

A handful of papers will be forthcoming that address these topics. The next issue will address the question “What are we estimating?” You are invited to join in the discussion. It is our hope that this series will answer some of your questions, and that you will share your thoughts.