



# Managing Agile at Scale

A briefing for Software Executives  
and Chief Information Officers

**July 2017**

## Executive Summary

Agile methods have undoubtedly brought major benefits to business of faster delivery of software that better meets evolving customer needs. However, the freedom given by Agile processes to individual teams to manage their own affairs has made it more difficult for senior management to set budgets and allocate resources optimally, and to track progress against budgets.

Controlling value-for-money and understanding performance in contracts with external suppliers of Agile services is particularly difficult and increases with the scale of the organization.

In this paper we offer simple but effective and long-established international standard solutions to enable Software Executives and CIO's to manage Agile delivery at scale, without risk of losing the speed and flexibility benefits of Agile processes.

## The benefits of using Agile Processes

The Agile Manifesto has caused a revolution in our understanding of how software should be delivered.

This revolution is clearly bringing great benefits to software customers in the form of:

- earlier delivery of working software that more closely meets customer needs, hence earlier delivery of business value;
- faster response to changing business needs.

There is also evidence that Agile adoption is reducing the cost and incidence of complete project failure. Agile processes help ensure that if a development is going to fail, it will 'fail-early'. This is a big step forward in overcoming what has been an endemic software industry problem of the frequent occurrence and high cost of failed projects.

## So what are the challenges of using Agile methods?

Whilst it's great to get the business benefits of Agile speed of delivery and flexibility, senior management needs to be able to control what's going on, as in any business activity, even if it is only 'light-touch', supportive control, as advocated by Agile teaching.

With the introduction of Agile, some of the long-established 'top-down' software project governance methods such as phase-reviews, project-planning for resource allocation, and use of past performance measurements for new project estimating have become weaker or have disappeared altogether. So managers need to find a new balance between the value of allowing individual Agile teams the freedom to deliver quickly and flexibly, and the value of control. This 'new balance' will vary with the type of business.

- Companies in markets for very fast-moving goods and services will consider their Agile capability to deliver business value via many software updates per day as much more important than the value of accurate estimating and performance controls.
- At the other extreme, public-sector bodies, contracted to outsourced software suppliers, will be as much concerned to control the scope and budget of their Agile activities and to demonstrate value-for-taxpayers' money, as to deliver functionality at the highest possible speed.
- All organizations, however, have an obligation to control costs regardless of domain or size.

From organizations using Agile processes, we hear more and more reports that the initial enthusiasm of software customers for the hype of 'pure' Agile processes, is now entering the phase of 'disillusionment'. The challenge for CIO's is how to resolve this freedom-versus-control balance for their organization.

**If this is of interest, read on.**

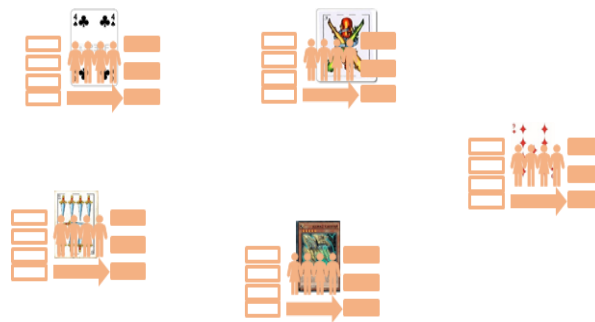
## Agile performance measurement and estimating processes

To achieve flexibility, Agile processes encourage development teams to be self-organizing, i.e. to agree what to do next with their customers and to measure, plan, estimate and control their own activities. Customers of a well-managed single team really appreciate the benefits of this approach.

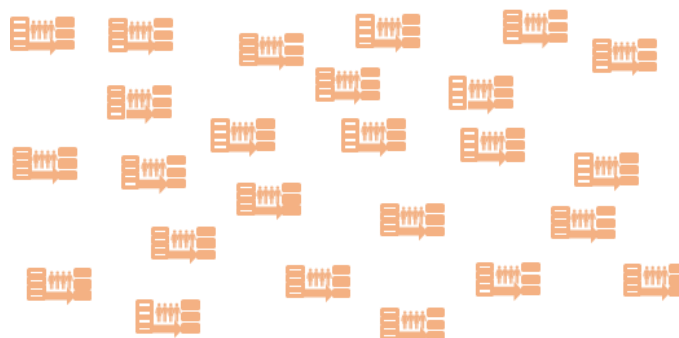
However, when each team is independently managing its own affairs, using its own non-standard size measure (Story Points) for estimating and performance control, it's very difficult for a program manager or a CIO to either support or control these activities in a consistent way. Story Points are great for use at the team level but cannot be used reliably for higher-level management tasks.

What criteria can you use to plan the number and size of teams? How can you gather data that is consistent across projects that can be used to track performance, estimate costs for future projects and ensure organizational learning?

The problem grows with scale. A small group, say five teams with 20 staff, may be manageable with informal reporting and organizational learning. See the Figure below, where each icon represents a single team.



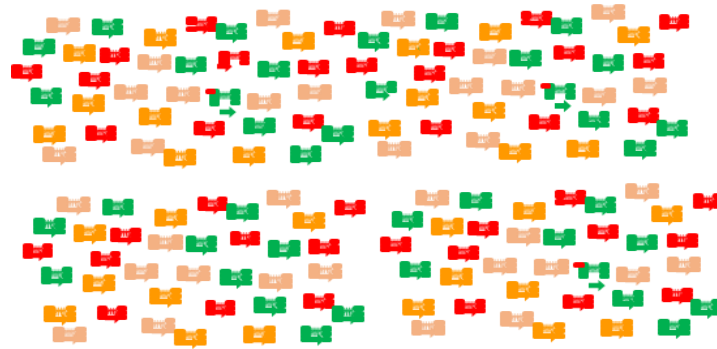
But what happens when an Agile development group has a big budget and comprises 20 self-organizing teams working in parallel on multiple projects? See the next Figure.



In these circumstances, how does a CIO or Program Manager

- know what progress is being made against budget?
- know which teams are under-performing and need help, and whether overall performance is improving, stagnating or declining?

Answering these questions gets even more difficult as the scale of activities increases further and external software suppliers become involved. See the Figure below.



In these circumstances, how will a CIO know which suppliers are delivering the promised cost savings and giving the best value for money?

In general, how can a CIO be held accountable for the performance of his/her resources, when performance is only known to each independent Agile team, in terms only they understand?

## The solution to these challenges: use 'FSM' methods

Technically, the solution to the problem outlined above is straightforward. The key is for all teams to measure the size (or 'amount') of software they produce in a standard way. Estimates of the effort and duration to deliver a new release or a whole new system may be obtained from an early estimate of the size of software product to be delivered, combined with performance data from comparable previously-delivered releases or systems. With such estimates, a business case can be made, a budget can be set, resources allocated and a basis established to control the software scope, all in a consistent way.

Progress may be controlled by measuring the size and number of backlog items delivered in a certain period. Each team's performance may be measured, if required, by measuring the size delivered for a given amount of effort (the team 'velocity').

The best way of measuring an amount of software product is to measure a 'functional size' of its requirements using one of three leading ISO-standard Functional Size Measurement (FSM) methods. These are the COSMIC [1], IFPUG [2] and Nesma [3] methods. Each method claims its own advantages.

Being based only on requirements, functional sizes are independent of the technology, processes or teams used to develop the software.

## Introducing an FSM method

Agile culture tends to expect external support, not control. So if a CIO wishes to introduce a FSM method into an existing Agile development group, the Agile culture must be carefully taken into account so as not to disrupt teams and risk losing the benefits of Agile.

The most important questions are who will do the measurements and when? The answer to the 'who' question depends on the scale of the organization and the relationship (contractual or informal) between the software customer and the Agile development teams.

Ideally, teams should do their own measurements and see FSM as an integral aid to Agile processes, not as an overhead. Functional sizes can be easily measured on delivery of a release or system, without any effects on Agile processes. At the level of individual sprints or iterations, each organization should decide for itself whether it is preferable to introduce FSM methods or to continue to use Agile methods for short-term estimating.

However, at the level of planning releases or whole deliveries in a large organization, and/or in a customer-supplier contract situation, and/or in the early stages of introducing a standard FSM method, it will probably be beneficial to use either:

- an internal expert group of measurement experts, e.g. in a Project Management Office,
- or an external specialist supplier of functional size measurement services.

The advantages of using an expert group of measurement specialists will be their objectivity, speed and accuracy of the measurements.

Whichever organizational solution is adopted it is vital that the measurement data is collected centrally for shared use and as a basis for organizational learning.

## Conclusions

Setting targets and budgets, and measuring performance against these goals is absolutely standard practice in every type of business, private or public, large or small. Agile software development activities should be no exception.

FSM methods can help middle and senior management achieve the support and control of Agile activities that is essential for them to do their job effectively.

Introducing a FSM method into an established Agile development group must be handled carefully as a typical change management project. The aim must be to convince the organization of the collective benefits of using objective support and control methods without upsetting the Agile culture and losing the benefits of the speed and flexibility of Agile processes in delivering software.

Mauricio Aguiar, [IFPUG](#) (*The International Function Point Users Group*)

Charles Symons, [COSMIC](#) (*The Common Software Measurement International Consortium*)

Eric van der Vliet, [Nesma](#) (*International Software Measurement and Metrics Association*)

COSMIC, IFPUG and Nesma are international organizations that maintain the ISO/IEC standards for sizing software requirements. These standards are used in industry for estimation, budgeting, contract and project management, supplier performance measurement, benchmarking and other management activities.

## Acknowledgements

The authors wish to acknowledge the support of the COSMIC, IFPUG and Nesma Boards. We thank Rafael de la Fuente of LEDA, Spain for good ideas and for permission to use the three Figures.

## References

- [1] 'Introduction to the COSMIC method of measuring software, v4.0.1', January 2016, <http://cosmic-sizing.org/publications/introduction-to-the-cosmic-method-of-measuring-software-2/>
- [2] See [www.ifpug.org](http://www.ifpug.org) In particular, see 'Function Point Counting Practices Manual', Release 4.3.1', 2010.
- [3] See [www.nesma.org](http://www.nesma.org) In particular, see 'Nesma FPA standard', Release 2.1, 2004, soon to be replaced by Release 2.3, 2017.
- [4] 'The value of using Functional Size Measurement methods in Agile activities', obtainable from the COSMIC, IFPUG and Nesma web-sites (in preparation).