DCG
david consulting group
TRUSTED ADVISOR

# "Story Points or Function Points or both?"

## July 2015

## Scope of this Report

Story Points and Function Points are both methods for 'sizing' software. This Trusted Advisor report will establish why sizing is important and present an overview of the two sizing methods followed by a discussion on the merits of both Story Points and Function Points by answering some very common questions:

*Can I use function points on an agile project?*
*"Story points are much easier and faster than function points."*
*Is there a relationship between story points and function points?*

## Importance of Sizing

When managing the delivery of your software product you need to know how big or small it is so you can properly plan, estimate and manage the delivery of that software. Sizing software requires a size measure that is ideally meaningful to both the development team as well as to the end user. And it should be a measure that can be applied consistently across all projects and all applications.

The development team will want to have an accurate measure of size so they can properly estimate the level of effort and duration. They will also use the size measure to monitor changing requirements (scope creep) as features or functions or stories are added to the original requirements document or product backlog. An end user or Product Owner may want a measure of size so they can understand the relative business value of what is being delivered; what features and functions the user is actually getting.

## Story Points

According to https://tcagley.wordpress.com, story points are a relative measure based on the team's perception of the size of the work. The determination of size is based on level of understanding, how complex and how much work is required compared to other units of work. Story points are expressed according to a numerical range, which is usually constrained to a limited set of numbers such as an adaptation of a Fibonacci sequence (e.g. 1, 2, 3, 5, 8 etc.).

Story points are a relative measure used by agile teams typically during a sprint session. Each story is assigned a story point value based on everyone's best understanding as to the "level of difficulty" of that particular story. Of course, “level of difficulty” can include different things such as complexity, size, duration, effort and so on. Regardless of the scale being used, in a process called planning poker, the values assigned are assessed independently by each individual, compared by the team and then discussed to reach a consensus. There is no consistent definition of what the values represent other than to use it as a comparative value of one story being larger/harder or smaller/easier than another within the one team. Over a number of iterations (sprints) an agile team can develop a consistent velocity (number of story points delivered per sprint) which can serve to estimate future amounts of work/effort in future sprints. Of course, even if that one team is achieving exactly the same volume/complexity of work as another team, their story points will not necessarily be the same.

## Function Points

“Function points measures software by quantifying the functionality requested by and provided to the customer, based primarily on logical design”; as defined by the International Function Point Users Group (www.ifpug.org) . Function points measure “software size” or, more precisely, the size of the requirements/design specified to which the resulting software provides a “no more, no less” solution. The size of a defined business requirement is a necessary piece of information if you want to estimate how long it will take and how much effort it will take to develop that piece of software. Unlike story points, function points are a defined, reproducible unit of measure. They can be measure consistently regardless of who is measuring them. Function points can be used on both agile and non-agile projects. For example, agile user stories, for the most part, describe the features and functions requested by the product owner.

The function point methodology calls for the identification of 5 key elements including inputs, outputs, inquiries, interfaces and internal stores of data. Naturally there needs to be some description of these elements; e.g., requirements documentation or stories, in order for a function point sizing to be accomplished. Once a function point size is determined it can be used to estimate level of effort or on the backend, the size information can be used to calculate productivity (fp/effort hours) and quality (defects/fp) levels of performance.

## Some Answers

***Can I use Function Points on an agile project?***

Yes, function points can be used on an agile project. In fact, both story points and function points can be used on agile projects and serve to effectively manage the project and measure performance.

We already know that story points are used to size the user stories for a given sprint/iteration. Stories can also be sized using function points. However, you don’t need to use function point size to estimate how long a collection of stories in a sprint are going to take because you have already set up a 2, 3, or 4 week cadence for your sprints.

Function Points are most useful and frequently used at the beginning of an agile project and upon delivery of a release or some significant delivery of functionality. In the beginning of an agile project, you may use function points to size the entire backlog and use that size information along with additional historical data points to estimate a total project cost and a predicted delivery date. At the backend of the project you may capture total function points delivered to look at performance levels and compare agile project performance levels to performance levels of other methodologies currently in use.

***"Story Points are much easier and faster than Function Points"***

This is a true statement; story points are quicker and easier than function points. The question really becomes, which method is more appropriate for the task at hand. Sitting down with the agile team and assigning story points to selected stories for a sprint backlog is an excellent exercise in approximating the complexity and required effort of selected stories. This is a collaborative approach that involves the team and provides a group understanding of each work element (story) and what may be involved. Even if story points were not assigned, the discussion alone would be of significant value in driving team efficiency.

Function points require a more detailed examination of the information (stories) available and achieving reproducible counts requires expertise and practice. There are specific guidelines to be applied and calculations to be made. It may be unrealistic to expect every team member of an agile team to have this skill set. As a result, the use of function points throughout an organization is usually performed by a central specialist team thus allowing for comparisons among the various agile teams and portfolios. Function points are also a size measure that serves both the developer and the end user. For the developer, they are used to manage the project outcomes. For the end user (product owner) Function points can be a useful vehicle for setting expectations with regard to identifying (and agreeing) what features and functions are being developed and deployed. However, the direct involvement of the agile team members in sizing the tasks they are going to work on has motivational benefits over the seemingly imposed sizing of the Central FP counting team.

Easier and faster are nice, but that is not the issue. The issue should be about which metric or set of metrics will provide you with the information you need to best manage the software deliverable, to make decisions and to manage expectations.

Of course, the real issue with the speed and ease of story points is that they are hard to scale across many agile teams. For the agile teams themselves this is not an issue but for the organization which needs to build product road maps, annual budgets, resource plans and so on, the loss of coherence is a significant one.

***Is there a relationship between story points and function points?***

The narrative below references the following example:

Iteration 1 – the team completed 10 stories (in a two week sprint) that were assigned a total of 50 story points. The function point size for those 10 stories was 100. The stories were focused on simple transaction I/O processing.

Iteration 2 – the team completed 5 stories in their second two week sprint. The stories were assigned 55 story points in total. The function point size for those 5 stories was 25.

Question: Assuming the team has achieved a fairly consistent velocity (50) why isn't there a correlation between SPs and FPs?

Observations:

Story points are often assigned with some consideration of required level of effort. In the first iteration the stories involved fairly simple processing and therefore were assigned an average of 5 story points each. In the second iteration, the stories represented more complex processing and were assigned an average of 10 story points.

Function Point analysis does not consider level of effort. It is accounting for the features and functions being delivered. The stories in iteration 1 were about processing inputs and outputs and accounted for a high number of function points. In the second iteration the stories required a greater degree of processing logic, but the features and functions being delivered were fewer.

| Iteration | # of stories | Story Points | Function Points | Complexity |
|---|---|---|---|---|
| 1 | 10 | 50 | 100 | Simple |
| 2 | 5 | 50 | 25 | Complex |

Story Points are a relative measure whereas function points are a well-defined consistent method of sizing.

Does this mean that function points cannot be used to estimate at a sprint level? Sprints are time boxed, usually as two week iterations. The desired state is to achieve a steady flow of work from sprint to sprint (velocity). For agile teams, this is adequately measured using story points. Function points are more appropriately applied to measure the overall project outcome. This can be done upon delivery of a release and/or function points can be applied when the product backlog is first developed as a means to estimate the total level of effort that may be required across all sprints.

## Conclusion

Story Points vs Function Points; so do we settle on one or the other or both? The answer is, yes.

Both these measures are useful and serve the intended purpose to more effectively manage a software deliverable.

Function Points are good for measuring the overall product deliverable at the beginning and at the end. The FP size information at the beginning of a project can be used to estimate overall schedules and costs. Also, the size information upon delivery can be used to measure performance.
Story Points are an effective method for managing the flow of work in an agile project. They too serve a purpose of estimating the amount of work that can be accomplished by the team in a defined period of time (sprint/iteration).

Clearly, sometimes the best use of these two methods overlaps and so it is important to make strategic decisions about when and how they will be used rather than local, tactical decisions.