As organizations increasingly become invested in Agile, we’re repeatedly hearing the same question, “Should I use story points, function points or both?” It doesn’t end there! The following, more specific questions are just as common:

1. Can I use function points on an Agile project?
2. Story points are much easier and faster than function points … right?
3. Is there a relationship between story points and function points?

When it comes to Agile development, most of us are not sure how to proceed. To clear up that confusion, let’s start by putting the two measures into perspective.

**Story Points – The Background**

Story points are often used by Agile teams during a sprint session. Each user story (software feature) is assigned a story point value based on the level of difficulty of that particular story. Story points are a relative measure 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.). They are selected based on the team’s perception of the size of the work to be completed. The determination of size is based on a level of understanding about the project’s degree of complexity, the team’s capabilities and how much work is likely required compared to other units of work.

In a process called Planning Poker, each team member shares what they believe the story point value of that user story should be – this is done anonymously. The values are then collectively assessed and discussed as a team in order to reach a consensus.

There is no consistent definition of what the values represent; they are simply a comparative tool. 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. 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 – The Background**

According to the International Function Points Users Group (IFPUG), “Function points measure software by quantifying the functionality requested by and provided to the customer, based primarily on logical design.” That is to say, function points measure “software size” or, more precisely, the size of the requirements/design specified. The size of a defined business requirement/design is a necessary piece of information in order to estimate how long and how much effort it will take to develop that piece of software.

The function point methodology calls for the identification of five key elements: 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 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.

Unlike story points, function points are a defined, reproducible unit of measure. They are consistent, no matter the user or requirement. Also unlike story points, function points can be used on both Agile and non-Agile projects.

**Some Answers**

Now that we have some background on these measures, let’s get back to those common questions.
1. Can I use function points on an Agile project?

Yes, as stated above, function points can be used on an Agile project. Both story points and function points can be used on Agile projects, effectively managing a project and measuring performance. However, you don’t need to use function point size to estimate how long a collection of stories in a sprint will take because you have already set up a 2-week (or similar) cadence for your sprints.

Function points are frequently used at the beginning of an Agile project and upon delivery of a release (or some other significant delivery of functionality). In the beginning of an Agile project, you may use function points to size the entire backlog. Using that information, along with additional historical data points, you can estimate the total cost of the project 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.

2. Story points are much easier and faster than function points … right?

This is a true statement; story points are quicker and easier than function points. But the real question is 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 are 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 skillset. As a result, the use of function points throughout an organization is usually performed by a centralized specialist team, thus allowing for comparisons among the various Agile teams and portfolios.

In addition, unlike story points, function points serve both the developer and the end user. For the developer, function points help to manage the project outcome. For the end user (product owner), function points can be a useful vehicle for setting expectations with regard to identifying (and agreeing on) the features and functions to be 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 centralized FP counting team.

Of course, the real challenge 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 that needs to build product road maps, set annual budgets, manage resource plans and so on, the loss of coherence is a significant one.

So, easier and faster are nice characteristics, but the key is to ask yourself 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 set expectations.

3. Is there a relationship between story points and function points?

The narrative below references the following example:

<table>
<thead>
<tr>
<th>Iteration</th>
<th># of Stories</th>
<th>Story Points</th>
<th>Function Points</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>50</td>
<td>100</td>
<td>Simple</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>50</td>
<td>25</td>
<td>Complex</td>
</tr>
</tbody>
</table>
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 five stories in their second two-week sprint. The stories were assigned 55 story points in total. The function point size for those five stories was 25.

Question: Assuming the team has achieved a fairly consistent velocity (50), why isn’t there a correlation between story points and function points?

Story points are 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 five 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 solely accounts for the features and functions being delivered. The stories in Iteration 1 were related to processing inputs and outputs and they 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.

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? Not necessarily, but sprints are time-boxed, usually as two-week iterations. The goal 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. As mentioned, this can be done upon delivery of a release and/or 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

So, what do we do? Do we use story points, function points or both? The answer is yes. Both of these measures are useful and help us to more effectively manage a software deliverable.

Function points are useful for measuring the overall product deliverable at the beginning and at the end. The function point size information at the beginning of a project can be used to estimate overall schedules and costs; the size information upon delivery can be used to measure performance.

Story points are effective 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.

\[\text{https://tcagley.wordpress.com}\]
\[\text{www.ifpug.org}\]