The Quest for system testing in Scrum

by Paul Quik

Where does system testing belong in agile system development? Is it in the development sprints or do we need special arrangements? The Scrum theory teaches us that after each sprint "a potentially shippable product/system" needs to be delivered to the business. Personal experience has shown that this is scarcely possible or even not possible at all in the case of newly built products or systems, especially in projects which include hardware. Usually the hardware is not available at the start of the software project and changes in the hardware are not realized within a single sprint. Adjusting hardware usually takes longer than one sprint because it requires, for example, trial runs and engineering samples before the final hardware becomes available.

In the past I have participated in several large and medium-sized projects. These were in the administrative sector (e.g. banking) as well as in technical automation. Within both sectors, I was looking for the optimal solution to position the system test.

During the previous projects I participated in, I have seen the following approaches to the positioning of system testing within a project or program:

1. Integration and system test is executed within the product development teams in the same sprint in which the product is developed.
2. System test is executed in the same sprint but is executed by a separate team.
3. A separate team is created for the system test. This team will lag half a sprint.
4. A separate team is created for the system test. This team will lag a full sprint.
5. System test starts when the final products are delivered.

The main advantages and disadvantages are described below.

Within variations 2 to 4, it is also possible to vary in the position of **the system integration**. It can either be executed by the product development teams or it can be positioned in the system test team. I have worked with both options, but you always need the expertise of the members of the product development teams since integration is development work, not testing work.

I will use the following system as a reference for illustrating the main advantages and disadvantages.

The system consists of a physical device controlled by a mobile app and contains:

1. Mobile App: the front end user interface for controlling the physical device.
2. Control unit: this translates the signals from the app to the protocol of the physical device.
3. Physical device: the device which is controlled. This can be, for example, a thermostat, lamp, or security camera.

All the products are developed by separate product development teams that work via the Scrum framework. The integration of the hardware with the software from the control unit and the physical device is done within the product development teams. So the product development teams deliver an integrated product at the system level.

Prior to starting the development, we agreed on the duration of the sprints and points in time when the products would be delivered at the system level. An integration diagram for the complete project was created and this stated which feature/user story should be delivered during each sprint. This plan was not fixed, but gave us a clear insight of the impact of any changes on the planning of the project. Also a "Scrum of scrums" was organized.

All the products are developed by separate product development teams working via the Scrum framework. The integration of the hardware with the software from the control unit and the physical device is done within the product development teams. So the product development teams deliver an integrated product at the system level

The main advantage and disadvantages for each of the five approaches mentioned are described below:

1. System test is executed in the product development teams (see Figure 1)

Within this approach, the system test is included in the product teams. This means that all testing activities are done at the end of the sprint. The system test activities are an item in the Definition of Done. This option will be found in a mature Scrum environment

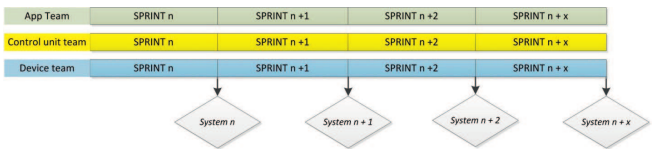

Figure 1. System test within the product development teams

Advantages of this approach are:

- Potentially shippable products will be delivered to the business each sprint.

- Integration into one system is done immediately. The knowledge of the team members is still fresh in the memory.
- The teams are multi-disciplinary, because the expectation is that all team members need to know how to develop and to test.

Disadvantages of this approach are:

- Velocity of building new features is relatively low, due to the integration and test activities.
- Developers are not only creating unit tests, but also are doing the system test. In my experience this is not what the developers like to do most.
- Availability of mature/multi-disciplinary team members is scarce.

2. Separate system test team, in flow with the product development teams (see Figure 2)

The separate system test team (integrates and) tests the newly developed End-2-End features immediately they are delivered to the system test team during the sprint. Within this approach, the sprint backlogs are aligned in detail (create E2E features before product features). This approach is also seen in mature Scrum environments

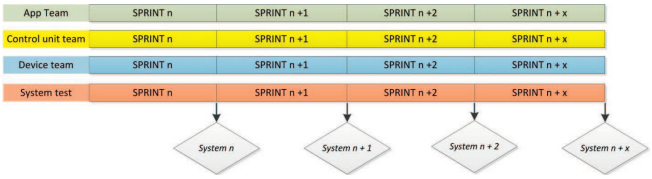

Figure 2. System test separate teams (in flow)

Advantages are:

- Expertise of system testing is bundled in a single team.
- Velocity of the product development team increases, due to the focus they have on their own expertise.
- All teams are working on the same features. All the knowledge is fresh in everyone's memory.

Disadvantages are:

- Knowledge of the software and interfaces is within the product development teams. Time needs to be reserved for helping the system test team.
- Setbacks within the sprint execution can have great consequences on the system test team and can cause the failure of the system test sprint.
- Product development teams need to support the system test. This decreases the velocity and focus of the product development teams.

3. Separate system test team lagging half a sprint (see Figure 3)

Within this approach, the sprint of the system test starts in the middle of the product development team sprints. The purpose of this shift is that the preparation and specification of the features will be done before the product development teams deliver their (integrated) product to the system test team.

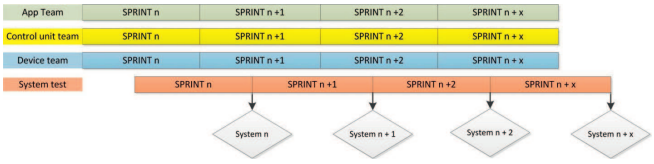

Figure 3. System test separate teams (lagging half a sprint)

This approach also has advantages:

- Test execution can start immediately when the products are delivered.
- At a product level, the E2E features are worked out in sufficient detail that the test specification at the system level is more stable.
- During preparation/specification, the focus of the product development team is on the features that will be tested.

As well as advantages there are disadvantages:

- Product development teams are no longer focused on the delivered features during the execution of the tests.
- If defects are found, no (or limited) time is available for the product development teams to deal with them.
- No potentially shippable product will be released after the product development sprint is finished.

4. Separate system test team lagging a full sprint (see Figure 4)

Lagging a full sprint behind the product development teams gives the system test team more certainty about what is delivered. This way, the system test team only prepares and executes the tests for the delivered features.

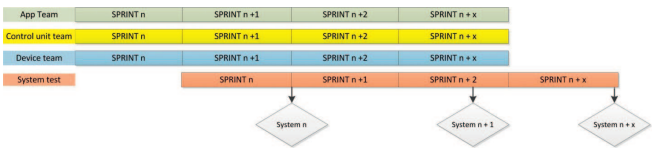

Figure 4. System test separate teams (lagging full sprint)

Advantages are:

- It is known which features are delivered and no effort is wasted on features that were not produced during the product development team sprints.

- Less dependency for product development teams to have setbacks during the sprint. They are able to make up lost time right to the end of the sprint.

Disadvantages:

- Focus of the product teams is not on the features at the system level.
- No fast feedback to the product development teams.
- The available time to deal with the issues found is limited. Only the blocking issues will be solved. The other issues need to be planned for a future product development team sprint.

5. System test after final sprint (see Figure 5)

This approach is the least Scrum-like approach. The products are developed in the Scrum approach, but system testing is only executed at the end of the development phase. The phasing of the project is still the traditional waterfall model.

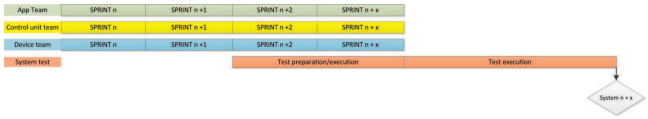

Figure 5. System test after final products have been delivered

The advantages are:

- Support of the product development teams is maximal. All issues found can be dealt with as agreed.
- The integration into one system only needs to be done once.
- No major interface changes are expected because the products are feature-complete.
- Requirements are mostly clear and written down.

The disadvantages in this case are:

- A big bang system integration usually results in big challenges
- No quick feedback loop for the product development teams. Usually when the development of the products is finished, the team is minimized and some of the key team members are relocated
- From experience, we know that development of the products is not finalized on the planned delivery date. The release date is fixed so the system test time will be shortened

Conclusion

Which approach to choose depends on the maturity of the organization. It cannot be said that one of these approaches is the best. It is also possible to change the approach during the project. If you see that one approach does not work as expected or you want to optimize the process, please feel free to adjust your approach to the wishes of the organization.

My experience is that **communication and alignments before and during the sprints is a key success factor** for creating a complete system. Definitions of Done and Definitions of Ready also need to be created, so the quality of the deliveries can be predicted. ■

> about the author

Paul Quik

As test consultant at Improve Quality services, a company that offers high-end services in the area of testing and quality management, I have worked for more than 13 years in several projects at several small and big companies. This was as well in the technical automation (e.g. climate control systems and lighting products) as in administrative organizations (website and application testing). In the last 5 years I mainly worked in agile projects as system tester, test lead or test manager. During these experiences I have seen several implementations of system testing and experienced the good aspects, but also the pitfalls. I have learned that each company has a certain level of experience in Agile and you need adjust the system testing to this level of experience.