Scrum Investigation Analysis for Android Application

S. Karuppasamy, Dr. G. Singaravel and P. Kaveen

Abstract— Agile is one of the software development methodology used in the current scenario of information technology. This methodology is mainly concentrated with how the cost and the time of industrial people are utilized effectively. Agile is an iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams with just enough ceremony that produces high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders. Scrum approach has been developed for managing the systems development process. It is an empirical approach applying the ideas of industrial process control theory to software development resulting in an approach that reintroduces the ideas of flexibility, adoptability & productivity. Scrum concentrates on how the team members should function in order to produce the system flexibly in a constantly changing environment.

Keywords— Agile Methodology, Scrum Process, Sprint, Android Studio.

I. AGILE METHODOLOGIES

Agile Methodologies are the conventions that are adopted for a project and it can vary from one project to another. Some of the well-known agile software development methods are described below

1.1. Agile Requirement Modelling

At the beginning of a project this is the stage to envision the high-level requirements and to understand the scope of the release. In this stage emphasis is to understand the higher level of requirements as to what the system should do. Typically initial requirement model can start as a usage model exploring the user view of the system. This could be Use-cases for Unified process project, a collection of feature for a feature Driven Development Project or collection of user stories for Extreme Programming Project. Then the domain information needs to be added containing the main domain entities, their major attributes, and the relationship between these entities.

1.2. Agile Modelling & Agile Model Driven Development

Agile Modeling (AM) is based on a collection of principles, such as the importance of assuming simplicity during modeling and embracing change because requirements will change over time. Ability to recognize that incremental change of the system over time enables agility and it is the rapid feedback which ensures accuracy. Furthermore, multiple model are effective. Agile modellers believe that content is more important than representation, that there are many way you can model the same concept yet still get it right. The fundamental practices of Agile Model Driven Development include:

- Creating multiple models in parallel.
- Applying the right artifacts for the situation.
- Modelling is the small increments.
- Collective ownership of project artifacts.
- Active end user’s/customers participation.
- Applying modelling standards.

1.3. Extreme Programming

Extreme Programming (XP) is a deliverable and disciplined approach to agile software development. XP is successful because it stresses customer satisfaction and allows the software developers to confidently respond to changing software requirements even late in the lifecycle. The business culture affecting the development unit is another focal issue in XP. Dynamic market situation test, constant working in pairs and vigilance are used to make design simple and aggressive refactoring.

1.4. Scrum

Scrum approach has been developed for managing the systems development process. It is an empirical approach applying the ideas of industrial process control theory to software development resulting in an approach that reintroduces the ideas of flexibility, adoptability & productivity. Scrum concentrates on how the team members should function in order to produce the system flexibly in a constantly changing environment.

1.5. Pair Programming

Pair Programming is the technique where two programmer works together on programming of one module. While one is actually keying in the information another one works as reviewer and they frequently change roles. The benefits of pair programming are:

- Improved design quality due to shorter programs, better design, fewer bugs, more readable, more maintainable designs, as well as catch design defects very early.
- Reduced cost of development due to large reduction in defect rate.
- Automatic learning and training by Knowledge passing, sharing and learning new programming techniques.
- Overcoming difficult problems due to working together.
• Improved morale.
• Decreased management risk as knowledge of the system is shared among programmers.
• Increased discipline and better time management.

1.6. Test Driven Development

Test Driven Development (TDD) is a software development technique that uses short development iterations based on prewritten test cases that define desired improvement or new functions. Each iteration produces code necessary to pass that iteration’s test. Finally, the programmer or team refactors the code to accommodate changes. Test Driven Development is a software design method, not merely a method of testing. With this method debuggers are very rarely used.

1.7. Feature Driven Development

Feature Driven Development (FDD) is an iterative and incremental software development process. It is one of a number of agile methods for developing software; FDD blends a number of industry recognized best practices into a cohesive whole. Its main purpose is to deliver tangible, working software repeatedly in a timely manner.

Five phases of FDD
• Develop a overall model
• Build a features list
• Plan by feature
• Design by feature
• Build by feature

II. SCRUM

The origin of Scrum goes back to 1986. The need for speed and flexibility in product development resulted in a different approach compared to the traditional sequential way. The old method went from phase to phase and the functions were specialized and segmented. With the ‘rugby’ approach the product development is done by a multidisciplinary self-organizing team whose members work together from start to finish. This resulted in the Scrum-framework, which is named after the term ‘scrum’ in rugby.

Scrum is the most popular agile method and is based on the Agile Manifestos:
• Individuals and interactions over processes and tools
• Working software over comprehensive documentation
• Customer collaboration over contract negotiation
• Responding to change over following a plan

The values on the left are more important than on the right. These values compensate for the shortcomings of the waterfall model. “The Agile Software Development Methods have the potential to provide higher customer satisfaction, lower bug rates, shorter development cycles, and quicker adaptation to rapidly changing business requirements.” Scrum is a framework to organize people and deliver a quality product on time. “Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques.”

Figure 2.1: Scrum Framework

Scrum approach has been developed for managing the systems development process. It’s an empirical approach applying the ideas of industrial process control theory to systems development resulting in an approach that reintroduces the idea of flexibility, adaptability and productivity. Scrum concentrate on how the team members should function in order to produce the system flexibly in constantly changing environment.

The main idea of scrum is that system variables involve several technical and environmental variables (e.g. Requirements, time frame, and resources) that is likely to change during a process. Scrum is a set of guidelines that govern the development process of a product, from its design stage to its completion. Scrum is implemented with the help of Sprint.

2.1. Team

The Scrum team consists of a Product Owner, Scrum Master and a Development Team. Scrum teams are self-organizing and cross-functional, they are not directed by others outside the team and have all the competencies to accomplish their work.

2.2. The Product Owner

It is responsible for maximizing the value of the product and the work of the Development Team. He is the only person responsible for managing the Product Backlog. This includes prioritizing items and making sure that the back-log is clear to all team members of the team.

The Product Owner interfaces with management and both internal and external stakeholders, who want to influence to the Product Backlog. The Product Owner has to be convinced, if a priority of an item needs to be changed, or a new item added. No one else but the Product Owner is allowed to change the priorities on items the Development Team is working.

2.3. The Scrum Master

It ensures that the theory, practices and rules of Scrum are understood and enacted. For the Development Team he serves as a coach, removes impediments and facilitates Scrum events. For the Product Owner he helps to manage the product backlog and ensures that backlog items are clear. For the
2.4. The Development Team

It is responsible for delivering a potentially releasable product at the end of each Sprint. These self-organizing teams must be empowered by the organization. They are cross-functional, all the skills combined together must be sufficient to deliver a quality product. The whole team together is responsible for the product. The maximum size of a Scrum team is 9 members.

2.5. Events

All events in Scrum are time-boxed; they have a maximum duration. The length of a Sprint is fixed; other events may be shorter when the purpose of the event is achieved. The Scrum meetings are a formal opportunity to inspect and adopt something, they are essential to get a high level of transparency.

The Sprint

It is a time-box of four weeks or less during which a potentially releasable (working) product is created. Each Sprint has a specific goal on what is going to be built. A new sprint starts immediately after the previous Sprint. Only in exceptional cases the Product Owner can cancel the Sprint, but this is rarely done because of the complications that can occur and the short nature of Sprints. The Sprint consists of the Sprint Planning, Daily Scrums, development work, Sprint Review and Retrospective.

The Sprint Planning

It is the first meeting in a Sprint. In this meeting the work to be performed in the Sprint is planned. This plan is created by the team and is agreed upon. The Scrum Master ensures that the planning is realistic. The first topic is what has to be done in this Sprint. Objectives are discussed and the Product Backlog items that should be completed to achieve the Sprint goal.

The Development Team must decide, based on their capacity, how many Backlog Items they can accomplish over the upcoming Sprint. The second topic is the way in which the chosen work is going to be done to turn this functionality into a potentially releasable product. The Backlog Items are decomposed in work items of a day or less and added to the Sprint Backlog. These work items should be fully understood by the team members, possibly with the help of the Product Owner.

The Daily Scrum

It is a meeting of 15 minutes every day. Its purpose is to synchronize activities and create a plan for the next 24 hours. The Scrum Master inspects the work done since the last meeting and the work that has to be done before the next one. These meetings are held every day at the same time and place. Every team member explains:

- What work they have done since the last meeting;
- What work they are going to do today;
- If they see any impediment that prevents an individual or the team from meeting the Sprint goal.

The purpose of Daily Scrum is to check progress towards the goal and completing the work in the Sprint Backlog. This optimizes the probability that the goal is met.

“Daily Scrums improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team’s level of knowledge. This is a key inspect and adapt meeting.”

The Sprint Review

It is held at the end of each Sprint to inspect the product that is implemented during the Sprint. This also includes feedback to the Product Backlog and decisions on what needs to be done in the next Sprint. It is an informal meeting and is intended to elicit feedback and foster collaboration. Items with higher priority that are on the top of the list are usually clearer and more detailed than those with lower priority.

The Sprint Retrospective

It is intended to reflect on the results of the Sprint and create a plan for improvement. This plan makes the next Sprint more effective and increases quality. Improvements may be implemented at any time, but the Sprint Retrospective provides a formal opportunity to focus on inspection and adaption.

2.6. Artifacts

“Scrum’s artifacts represent work or value to provide transparency and opportunities for inspection and adaptation. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifacts.”

The Product Backlog

It is an ordered list of functionality that might be needed in the product; it is the only source for requirements. The Product Manager manages the Product Backlog. In the beginning of development it only represents the first known and best-understood requirements, later on it evolves as the environment where the product will be used evolves. Thus, the Product Backlog is never complete and always represents work in progress. It lists all features, functions, requirements, enhancements and fixes that are needed for future release.

The Product Owner and Development Team constantly add detail or estimates to the Backlog, this is an on-going process called Backlog refinement. Items with higher priority that are on the top of the list are usually clearer and more detailed than those with lower priority. High priority items need to be refined enough to complete them in the next Sprint. If so, they are considered ready for planning in the next Sprint. The progress toward completing specified work is tracked by the Product Owner and made visible to all stakeholders. This can be visualized in various ways like burn-downs, burn-ups or cumulative flows. These have proven useful.

The Sprint Backlog

It is a subset of the Product Backlog items selected for the Sprint. It represents the functionality that will be added in the
next increment and the work to be done to reach the goal of this Sprint. The Development Team can update the Sprint Backlog during the Sprint. Tasks are added and if tasks are completed, the remaining work is updated. Unnecessary items are removed. Only the Development Team can change the Sprint Backlog during a sprint. It gives a clear overview of the work that is performed by the Development Team during the Sprint.

The Product Increment

It is the sum of all items completed during the Sprint and the previous Sprints. The new increment must be in useable condition and must meet the definition of done.

The Definition of Done

It is a description of when a Backlog Item can be considered done and is different per Development Team. They have a shared understanding of when work is complete to ensure transparency.

According to case-study research in five companies the conclusion was that “SCRUM works in any environment and can scale into programming in the large”. A case study proves that Scrum can result in a high productivity increase in comparison with traditional methods.

Another advantage: Scrum cuts through project complexity and brings order from chaos by enabling a team to organize itself, which allows a particularly productive order to emerge. However, there are some pitfalls in Scrum implementation which are stated below.

2.7. Pitfalls of Scrum

A few pitfalls of Scrum:

- Scrum does not provide adequate design documentation necessary for future development. It may not work well with projects that require high level of innovation because its focus is on bringing order to the development process.
- Quality assurance and testing need to be done in a different way compared to Waterfall, there are some challenges in implementing a new testing methods.
- Teamwork and dedication of team members is very important.
- Resistance to change is a major factor in the adoption of Scrum.

2.8. Android Studio Overview

Android provides a rich application framework that allows you to build innovative apps and games for mobile devices in a Java language environment. The documents listed in the left navigation provide details about how to build apps using Android's various APIs.

Android Studio is the official IDE for Android application development, based on IntelliJ IDEA. On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Flexible Gradle-based build system
- Build variants and multiple apk file generation
- Code templates to help you build common app features
- Rich layout editor with support for drag and drop theme editing
- lint tools to catch performance, usability, version compatibility, and other problems
- ProGuard and app-signing capabilities
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine and much more.
- The flexibility of the Android build system enables you to achieve all of this without modifying your app's core source files

The Android build system is the toolkit use to build, test, run and package of the apps. The build system can run as an integrated tool from the Android Studio menu and independently from the command line. App can use the features of the build system to:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app with different features using the same project and modules.
- Reuse code and resources across source sets.

Take advantage of the latest Google technologies through a single set of APIs (application programming interface), delivered across Android devices worldwide as part of Google Play services. Start by setting up the Google Play services library, then build with the APIs you need.

III. Conclusion

Agile Methodologies are the conventions that are adopted for a project and it can vary from one project to another. The need for speed and flexibility in product development resulted in a different approach compared to the traditional sequential way. The old method went from phase to phase and the functions were specialized and segmented. The Android build system is the toolkit use to build, test, run and package your apps.

REFERENCES


