Validation of Outsourcing Teams Work on Agile Projects of Samsung R&D Institute Brazil

Extended Abstract

Gizelle S. Lemos
Samsung R&D Institute Brazil
Campinas, SP, Brazil
g.lemos@samsung.com

Marcia Cristina de C. Costa
Samsung R&D Institute Brazil
Campinas, SP, Brazil
m.costa@samsung.com

Tatiana D. Borghi
Samsung R&D Institute Brazil
Manaus, AM, Brazil
t.borghi@samsung.com

Paula G. Povoas
Samsung R&D Institute Brazil
Manaus, AM, Brazil
paula.povoas@samsung.com

ABSTRACT

Samsung R&D Institute Brazil (SRBR) is one of Samsung's research centers in the world in which there is research focused on software areas. SRBR teams have worked in collaboration with Samsung headquarter and outsourcing partners for producing software that aggregates value on Samsung products. SRBR has different partners with different levels of skill, maturity and that work in different contexts. For this reason, managing software development projects and guaranteeing the quality of resulting products have been challenging for SRBR. This experience report describes the process created to improve partners' management and proposes methods for tracking and improving the methods applied for validating the work delivered by outsourcing partners in order to guarantee that it meets Samsung quality requirements.

CCS CONCEPTS

• **Software and its engineering** → *Acceptance testing*;

KEYWORDS

Outsourcing management process, acceptance testing, outsourcing development

1 INTRODUCTION

SRBR is one of the several research centers of Samsung. It is located in Brazil and geographically distributed in five different sites: three

in southeast and two in the north of the country. Many research projects have been developed by SRBR demanded by Samsung Headquarter (HQ) in South Korea or by the local Latin America team. All software products produced by the SRBR teams must follows policies and processes defined by HQ and tailored to SRBR needs. Its teams develop software projects on different research areas such as artificial intelligence, machine learning, image&video processing and security. Besides research projects, there are SRBR teams working on applications for local market and projects for contributing on platforms, such us Android [5] and Tizen [6]. In both types of projects, not only the internal SRBR team has participated but also outsourcing partners, such us universities, research and development institutes and outsourced development organizations located in different regions of Brazil, Figure 1.

Figure 1: SRBR and Partner's Locations

This business model that groups internal development teams and external partners on research projects has been heavily adopted by SRBR due to Brazilian incentive laws that concede tax exemptions to the company since it works with Brazilian research institutes and universities. Since 2015 SRBR has worked with about thirty partner companies. Difficulties present when working with outsourcing are related with the following aspects:

- Outsourcing companies have different levels of skill and maturity when considering software development process;
- Different teams from the same outsourcing company have different levels of knowledge about SRBR SW Development Process; [3]
- Teams from different types of outsourcing companies or companies from different country regions have different capabilities;
- Outsourcing companies are located in different places, regions of Brazil.

In order to standardize the acceptance of work produced by SRBR' outsourcing partners, SRBR Software Engineering team (SRBR SE) defined on *SRBR Technical Outsourcing Process* (Section 3) some clear criteria for acceptance tests and on *SRBR Software Development Process* a set of activities applicable during acceptance that have been applied on the majority of ongoing projects with participation of outsourcing partners.

Whenever a *release* is delivered by a partner, SRBR team performs acceptance activities which results on a verdict about the *release*' quality. All releases should be formally accepted by SRBR before being incorporated into the product.

The purpose of this report is to present the experience of SRBR, in special of SRBR SE team in working with outsourcing partners and how SRBR are accepting the deliverables. We will present some results of projects developed in 2017 and projects in developing in 2018.

On next section we describe *SRBR Software Development Process* and the processes that support the outsourcing activities (Section 2). On Section 3 we detail how SRBR have accepted deliverables from outsourcing organizations, some results obtained and what we need to improve. On section 4 we present the conclusions of our experience.

2 SRBR SOFTWARE DEVELOPMENT PROCESS

SRBR software projects have been developed following a SRBR SW Development Process. The SRBR SW Development Process was tailored from Samsung Engineering Process (SEP) defined by HQ and references ISO/IEC 12207 [1], Figure 2.
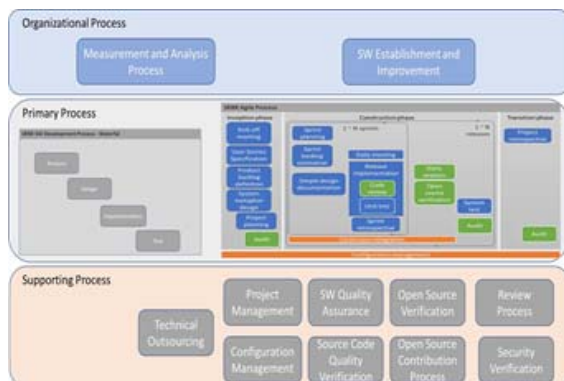

Figure 2: SRBR SW Development Process Overview

Sections 2.1 and 2.2 describe the processes used during validation of outsourcing teams. An overview of entire SRBR SW Development Process is available on [3].

2.1 SRBR Technical Outsourcing Process

The SRBR Technical Outsourcing Process comprises four major phases:

- *Plan outsourcing project*: define the scope and strategy for outsourcing;
- *Prepare outsourcing project*: prepare Request for Proposal (RFP), selected outsourcing organization, negotiate and sign off contract and statement of work;
- *Track outsourcing project*: monitor the progress of development of the project, verify and validate deliverables according statement of work, conduct acceptance tests;
- *Close outsourcing project*: project transfer (transfer the technology developed by outsourcing organization to SRBR internal development team when it is applicable), conduct lessons learned meeting, evaluate outsourcing organization and close contract.

2.2 SRBR Agile Process

SRBR SW Development Process was defined and established considering mandatory and non-mandatory phases, activities and deliverables that software development projects follow. It describes either Waterfall life-cycle and Agile methodology as reference. From Agile methodologies, *SRBR SW Development Process* considers:

- From Xtreme programming [8]: User stories, system metaphor, simple design, release planning, continuous integration;
- From Scrum [7]: Sprints, estimation, daily meetings, retrospective meetings, burn-down/burn-up charts;
- From Lean software development [4]: Kanban board.

Besides activities from Agile methodologies, *SRBR SW Development Process* defines a set of activities for software quality assurance:

- *Static analysis*: automated detection of potential errors through static code analysis without running the target program. Defects on code and build without preparation of test cases or program execution. It supports a wide range of languages and checkers for versatility to deal with functions needed;
- *Open source verification*: source code analysis to verify if open source components included into the software project are allowed considering rules for commercial software;
- *System test*: test performed by an independent team that may include conformance, stress, integration, and usability tests of software product releases;
- *Code review*: technical evaluation by developers contributing for improvement and fixing of source code before integrating it on a product.
- *SW quality assurance audits*: periodical SW audits performed to verify the adherence of projects with respect to their pre-specified SW Development Process;
- *Configuration management*: identification of configuration items (CI) and baselines definition, control changes in project's CIs assuring the integrity of the software produced.

Quality assurance activities are mandatory to SW projects and most of them have specific criteria that projects must achieve before development is finished.

Agile concepts and quality assurance activities have been applied on phases of the software development life cycle on mentioned below and summarized on Figure 3:

- *Inception phase*: beginning of project in which occurs definition of plans, initial backlog and initial idea about Software architecture;
- *Construction phase*: execution of N sprints in order to generate releases according prioritization defined by client (product owner);
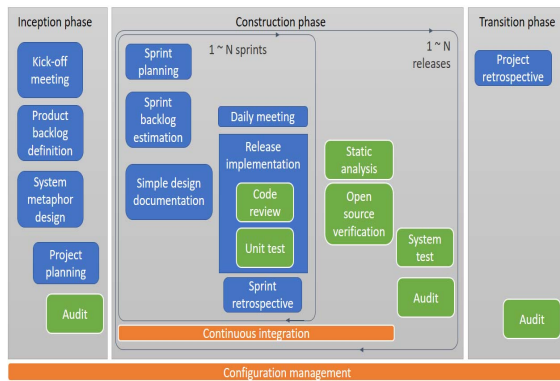- *Transition phase*: technology transference, lessons learned and completion of project.

Figure 3: Phases of Agile methodology defined in SRBR SW Development Process

Inception Phase

The focus of this phase is plan main project milestones and document the first version of the project's planning deliverables. All team members participate on a kick-off meeting and define the tailoring needs for the project considering the *SRBR SW Development Process*.

Project team documents initial product backlog and system metaphor (high level software architecture) to be able to plan the projects' releases schedule.

After definition of project plans, an audit is performed by SRBR SE team in order to verify if all required activities and deliverables have been generated. The summary of activities and deliverables of Inception Phase is on Table 1:

Construction Phase

During this phase, the product is developed and released according with project strategy and planning defined in inception phase. One release can be comprise by one or more sprints. During the sprints, project team performs for each sprint: sprint' planning, backlog estimation, Daily meetings, software source code generation, unit tests, Code review, and sprint retrospective.

Whenever, after the end of sprint, a release has been planned, project team performs some activities in order to guarantee its quality: static analysis, open source verification, system test and release demonstration (to product owner).

Table 1: Summary of Inception Phase

Activities	Mandatory Deliverables
Kickoff Meeting	Kickoff Meeting Kick-off Meeting notes, Project Tailoring Document
User Stories Specification	Initial Product Backlog
System Metaphor Definition	System Metaphor
Project Planning	Agile Release Plan, SW Configuration Management Plan, SW Quality Assurance Plan
Audit	Inception Phase Audit Report

Automatic builds, unit tests and static analysis run in the continuous integration system increasing the time available for issues resolution.

Before the delivery of each release, SRBR SE team performs an audit in order to verify if all software quality assurance activities were performed and criteria achieved. The summary of activities and deliverables of Construction Phase is on Tables 2 and 3.

Table 2: Summary of Construction Phase - Sprint activities

Activities	Mandatory Deliverables
Sprint Planning	Sprint planning report (sprint goal, sprint backlog, product backlog, sprint needs), sprint backlog estimation
Open source verification	Open Source Pre-review Report
Daily Meeting	Burndown/Burnup charts updated
Source Code Implementation	Source code reviewed on repository, Unit Tests Reports
Sprint Retrospective	Sprint Retrospective Report

Table 3: Summary of Construction Phase - Release activities

Activities	Mandatory Deliverables
Static Analysis	SW Code Quality Verification Report
Open source verification	Open Source Verification Report
System Test	System Test Plan, System Test Cases, System Test Report
Audit	Construction Phase Audit Report

Transition Phase

The last phase of *SRBR SW Development Process* focus on project closure. All team members participate on the lessons learned meeting and the final report of the project is documented by project leader comparing planning and achievements for the project. Final product release is delivered and audit verifies all product deliverables. Below table (Table 4) lists activities and mandatory deliverables of this phase.

Table 4: Summary of Transition Phase

Activities	Mandatory Deliverables
Project retrospective	Lessons Learned Meeting Notes
Project completion	Project Completion Report
Audit	Transition Phase Audit Report

3   ACCEPTANCE ACTIVITIES

Due to problems presented in Section 1 regarding outsourcing management and result's evaluation, SRBR SE Team, responsible for defining and improving *SRBR SW Development Process* and for supporting development teams on process deployment in projects, decided to define complimentary activities and make some improvements on SRBR Technical Outsourcing Process for projects that involve development teams from outsourcing partners. The improved process has been applied in projects since 2017.

Below are presented changes on SRBR Technical Outsourcing Process and on SRBR Software Development Process.

The first activity included on projects is *Review and Detail of Statement of Work*. In summary, this document describes the work scope, project deliverables, schedule of releases and acceptance criteria for the partners' work. However, there before process improvement there were no template or minimum mandatory set of information for it.

SRBR SE team defined a document named SW Annex that summarizes all activities and deliverables expected by software development phase. Nowadays, it is part of Request of Proposal (RFP) sent to outsourcing candidates before partner's choice.

When an outsourcing is selected, SW Annex is included in Statement of Work and/or in Partnership' Agreement. SW Annex details:

- The type of outsourcing project, based on the scope of development: e.g. requirements to system test, design to system test, implementation to system test, requirements to implementation;
- Mandatory activities;
- Mandatory deliverables and minimum content;
- Roles and responsibilities during development;
- Releases acceptance criteria. Specific acceptance criteria have been defined for some activities performed during the project. The focus is on construction phase - release activities:
  - Static analysis:
    * All potential defects should be classified as defect or false-positive;
    * All potential defects classified as false-positives must be justified and audited;
    * All defects of critical severity must be fixed;
    * A maximum of one defect may be left open in one million lines of code;
  - Open source verification: only allowed open source components must be adopted by partners' development teams and their licenses obligations must be abide;
  - System test: performed and reported by outsourcing team before each release delivery:

* All defects found by partner's test team must be classified according severity: A, B, C in witch A is more severe and C is less severe;
* Al defects with severities A and B must be resolved before release delivery to SRBR;
- Acceptance test: on test performed by SRBR after all releases delivery by partner are expected no defects of severity A and B.

Based on the Statement of Work, SRBR SE team performs quality assurance activities as defined on SW Quality Assurance Plan and acceptance test activities for each release. In parallel, SRBR project team defines what portion of the product backlog will be under partner's responsibility. The outsourcing scope is the target for validation from the point of view of the product owner and of SW Annex criteria. In case of projects with participation of outsourcing partners, a member of SRBR team represents the product owner on release acceptance.

The Construction Phase starts after the agreement between SRBR and partner about plans defined on Inception Phase. On SRBR software process each release is a result of one or more sprints produced according to scope and schedule of Statement of Work.

For each release delivered by outsourcing team, SRBR performs the set of activities for quality assurance: static analysis, open source verification and acceptance test. Only if the release meets all acceptance criteria it is considered *Accepted* by SRBR. In addition, all deliverables that compose the release are verified during the Construction Phase Audit.

Below are presented some results of eleven closed and ongoing projects in the period of 2017/2018 that have followed this improved process.

Static Analysis

In projects in which static analysis is applicable (only C, C++, C# and Java source code), SRBR tool (proprietary tool developed by HQ) identified more than 350 Critical and Major issues that escaped in code review (Figure 4).
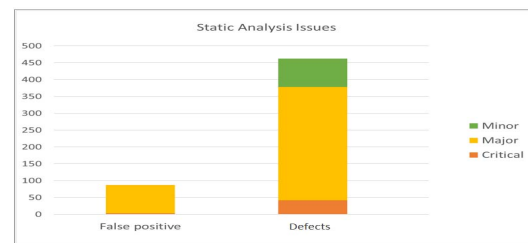
Figure 4: Static Analysis results

Open Source Verification

During verification of open source components incorporated by partner's team we have used BlackDuck Protex™tool[2], we identified the usage of 23 open source components and we could guarantee that those projects are compliant with Samsung's open source policies (Figure 5).

System Tests

SRBR SE team verified all system test reports delivered by outsourcing teams. System Test Reports are compared with SRBR
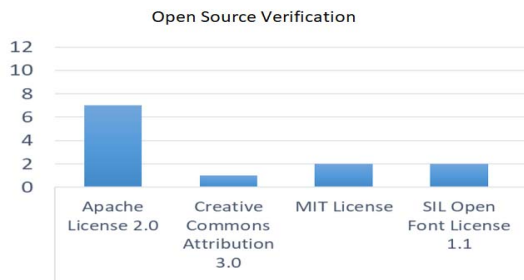
Figure 5: Open source verification results

Acceptance Test Reports in order to verify what defects escaped during partner's tests.

Acceptance Tests

SRBR SE team performs acceptance test activity after all releases delivery. For verified releases were identified defects of severity A and B that escaped during system test cycles (around 38% of defects found). In this case the release delivered was rejected by SRBR and new version was generated after defects fixing by partners.

During the acceptance tests, SRBR SE realized that was difficult to classify the severity of some defects found based on actual severity definitions when testing some types of projects, e.g. games.

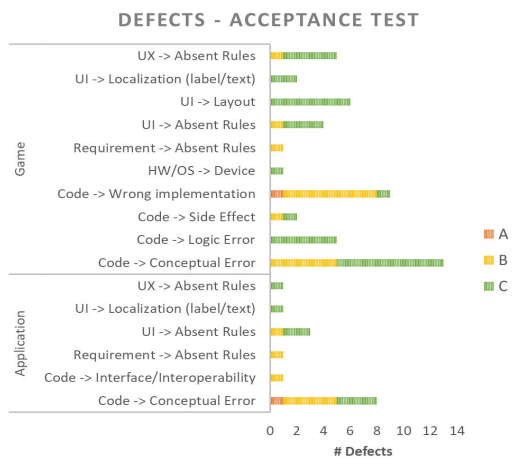Figure 6 illustrates defects found during acceptance test divided by defect category and project type.

Figure 6: Acceptance Test defects by category

Audits

On audits performed by SRBR SE team a set of issues was found. All issues were classified as non-conformities or improvement opportunities. Most of issues identified on outsourcing teams are related with configuration management and project management processes (Figure 7). It demonstrate that SRBR and partners should work to improve the level of maturity of these organizations in these process areas.
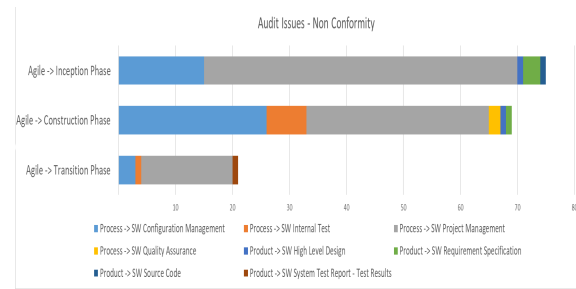

Figure 7: Audit Issues

4 CONCLUSIONS

SRBR has obtained some gains performing the formal acceptance of the outsourcing work:

- Independently of partner's company or SRBR team that manages the software project under development the criteria for releases acceptance are the same;
- Acceptance test has avoided that product releases with escaped defects (not found by partner's test teams) were delivered to SRBR;
- Open source verification has guaranteed that only allowed open source components have been adopted by partners' teams and also their licenses obligations were abide;
- Audits have helped on standardize deliverables produced by partners facilitating the understanding and analysis of by SRBR teams;
- Audits also have helped teams to find weakness points on agile practices understanding and execution by partners;
- Different improvement opportunities were identified such as:
  - Possibility of Use the results of each project to evaluate and select outsourcing companies;
  - Feedback about the quality of process and deliverables was uses to establish an action plan to improve the capability of outsourcing organizations;
  - Definition of severity of defects has been reviewed and specialized by project type, such us games and applications in order to facilitate classification of defects.

REFERENCES

[1] 2008. *ISO/IEC 12207:08 - Systems and Software Engineering - Software Life Cycle Processes.* Technical Report. c1–138 pages. https://doi.org/10.1109/ieeestd.2008.4475826
[2] Black Duck Company. 2018. Protex tool. (2018). Retrieved March, 2018 from https://www.blackducksoftware.com/
[3] Marcia C. C. Costa, Gizelle S.Lemos, and FlavioBeck. 2018. Software Engineering Tools Environment for Outsourcing Teams Collaboration. In *Proceedings of 13th IEEE/ACM International Conference on Global Software Engineering.* https://doi.org/10.1145/3196369.3196393
[4] Mary Poppendieck and Tom Poppendieck (Eds.). 2003. *Lean Software Development: an Agile Toolkit.* 203 pages.
[5] Android Open Source Project. 2018. Android. (2018). Retrieved March 10, 2018 from https://www.android.com/
[6] Linux Foundation Project. 2018. Tizen An open source standards-based software platform. (2018). Retrieved March 10, 2018 from https://www.tizen.org/
[7] Scrum.org. 2018. Homepage - Scrum.org. (2018). Retrieved March 10, 2018 from https://www.scrum.org/
[8] Don Wells. 2013. Extreme programming - a gentle introduction. (2013). Retrieved March 10, 2018 from http://www.extremeprogramming.org/