Software Engineering Tools Environment for Outsourcing Teams Collaboration

Extend Abstract

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

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

Flavio Beck
Samsung R&D Center Brazil
Campinas, SP
flavio.beck@samsung.com

ABSTRACT

Samsung R&D Center Brazil (SRBR) is one of the Samsung 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.

Considering the complexity of the scenario (different locations and different levels of access to internal tools environment and different maturity level of teams) it has been a challenge to produce high quality software. This report describes DMZ ("Demilitarized Zone"), a remote software engineering environment built by SRBR to minimize differences and distances among teams that work on the same software project. DMZ development and deployment include the definition of SRBR outsourcing process, improvements to the SRBR software process and setup/deployment of a set of integrated tools to support projects.

CCS CONCEPTS

• **Software and its engineering** → *Agile software development*;

KEYWORDS

Technical Outsourcing Process,Agile, SW Development Process, SW Development Tools

ACM Reference Format:

Marcia Cristina de C. Costa, Gizelle S. Lemos, and Flavio Beck. 2018. Software Engineering Tools Environment for Outsourcing Teams Collaboration: Extend Abstract. In *Proceedings of ICGSE '18: 13th IEEE/ACM International Conference on Global Software Engineering (ICGSE '18).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3196369.3196393

1 INTRODUCTION

Samsung R&D Institute Brazil (SRBR) is one of several Research Centers of Samsung. It is located in Brazil and geographically distributed in five different sites: 3 sites located in southeast (SRBR-C and strategic partner-1 and strategic partner-2) and sites located in north (SRBR-M and the strategic partner-2)

All research and development (R&D) projects developed by SRBR have been demanded by Samsung headquarter (HQ) in South Korea or by the local Latin America team to aggregate value to Samsung products with focus on the Latin America Market and to support worldwide business. In both types of projects not only the internal SRBR development team has participated, but also external partners, such us: research and development institutes, universities and outsourced development organizations located in different regions of Brazil. See Figure 1.

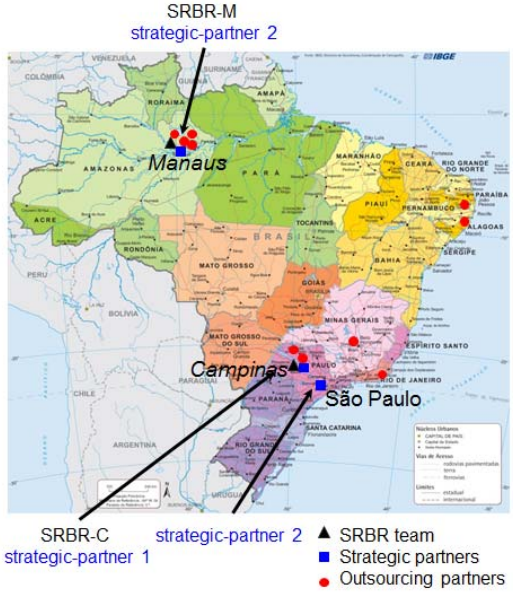

Figure 1: Samsung R&D Institute Brazil (SRBR) and outsourcing partner

Currently, the total number of software engineers (project managers, project leaders, software developers, software testers) involved in SRBR SW Development is around 1200 (considering SRBR internal' plus strategic partner's teams). In addition, there are close to 500 members of outsourcing partners working in our DMZ environment.

This business model that groups internal and external development teams on R&D projects has been heavily adopted by SRBR due to Brazil's incentive laws [4, 5, 8] that concede tax exemptions to the company since it works with Brazilian research institutes and Universities. Samsung must assure compliance with the requirements of the law, it means that all evidence, results and deliverables of all projects must be promptly available in case of government

ICGSE '18, May 27–29, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5717-3/18/05.
https://doi.org/10.1145/3196369.3196393

audit. In addition, SRBR project's deliverables must be adherent to Samsung's quality and security requirements.

The majority of SRBR projects are about research with specific characteristics. Due to this nature, the environment and process adopted on each project is totally customized based on:

- Number of partners involved;
- Type of partners involved;
- Size of team;
- Applicable phases of development lifecycle;
- Type of project: mix of research and development, application and solution development, UI & UX projects;
- Maturity of organization: some organizations have a software process defined and established following knowledge SW maturity models or using Agile methodologies, others have no defined process and have adopted few software engineering practices.

Considering the complexity of SRBR project's scenarios, the efforts and difficulties in managing outsourcing partners with different levels of maturity and the absence of a secure environment to exchange project information, SRBR decided to establish a DMZ with the following elements:

- SRBR SW Development Process as a reference source for phases, activities and mandatory deliverables is the standard to be followed by outsourcing partners working on projects that have adopted Agile methodologies;
- SRBR SW Development Tools Framework supports SRBR SW Development Process with collaborative and development tools.

2 SRBR DMZ ENVIRONMENT

The DMZ environment comprises four components: SW development process, SW development tools framework, infrastructure and management, and project's environment. These components are described below.

2.1 SRBR SW Development Process

SRBR SW Development Process was defined and established considering phases, activities and deliverables (mandatory and non-mandatory) to be followed by SRBR SW Development projects using an Agile methodology or Waterfall life-cycle as reference.
SRBR SW Development Process documentation was structured as a set of regulations, guidelines, templates, checklists and training material.
SRBR SW Development Process was tailored from Samsung Engineering Process (SEP) defined by HQ and references ISO/IEC 12207 [3]. It has been applied together with organizational and supporting processes as shown in Figure 2. Below is presented an overview of the process SRBR have defined:

The Primary Processes

- **SW Development Process:** Define the main SW Engineering processes to develop a software product according to an Agile or Waterfall life-cycle.

Supporting Processes

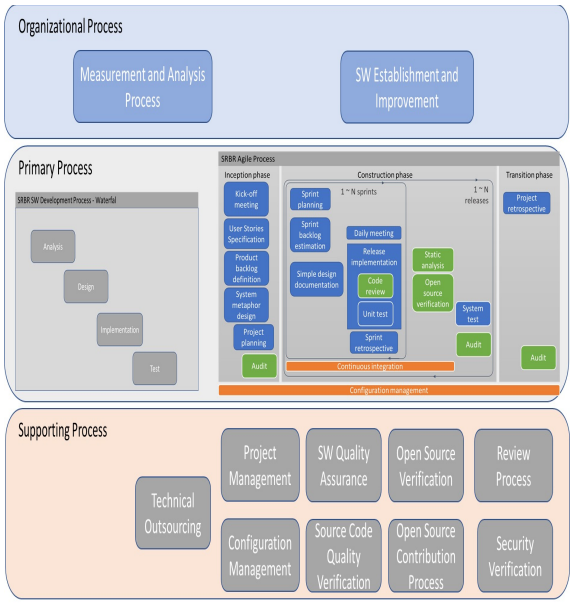

Figure 2: SRBR SW Development Process Overview

- **Technical Outsourcing Process:** Define the set of activities for managing outsourcing teams: identification of opportunities for technology outsourcing, selection of partners and establishing a formal contract, controlling the contract and assuring the contract requirements were achieved, closure of the contract and evaluation of partners.
- **Project Management:** Define a set of SW project management activities that should be performed throughout SW development plan, SW development progress and control, and SW development completion. SW project management is performed in order to obtain maximum project performance, efficient resource management, and high quality of deliverables while the project is in development following the SW development process.
- **Configuration Management Process:** assure the identification and definition of configuration items and baselines, control the changes in items assuring the integrity of the software produced.
- **SW Quality Assurance:** conduct periodical SW audits to verify the adherence to process.
- **Source Code Quality Verification:** perform static analysis on source code to verify code quality.
- **Review Process:** review project assets in order to guarantee that all stakeholders agree about them and that there are no quality problems.
- **Open Source Verification:** verify if only allowed open source software has been incorporated into Samsung software projects and guarantee that all open source obligations have been followed.
- **Open Source Contribution Process:** guarantee that all source code submitted as a contribution in open source

projects have been analyzed, avoiding inclusion of defects in open source repositories.
- **Security Verification:** perform activities during all development life-cycle that identify possibilities of vulnerabilities (lack of security).

Organizational Processes

- **Measurement and Analysis Process:** define activities for measuring and analyzing the process, define key process indicators - KPI(s) and define goals for each KPI
- **SW Establishment and Improvement Process:** define the activities for supporting continuous improvement of SRBR Process and Tools.

SRBR SW Development Process is maintained by SRBR SE Team through SRBR SW Establishment and Improvement Process (see section 2.3 for more details). Nowadays, at least two baselines of SRBR SW Development Process are released per year and they implement requests from SRBR team, and Strategic and Outsourcing Partners.

2.2 SRBR SW Development Tools Framework

SRBR SW Development Tools Framework supports the SRBR SW Development Process. It was created based on the following premises:

- Include mandatory software development tools required by HQ, such us Samsung Static Analysis Tool and Open source verification Tool.
- Incorporate open source and commercial tools that already have been consolidated in development communities or the market and permit integration into the environment.

See in Table 1 the tools that compose SRBR SW Development Tools Framework.

Tool	Purpose
Jira™[1]	Plan and track Project using issue types such us: Stories, Tasks, Risks, Bugs, Improvements, Audits, Audit Issues, Open source analysis
Confluence™[1]	Document collaboration, Project Space Documentation
Bitbucket™[1]	Git code management
Jenkins©[6]	Continuous integration
Testlink©[9]	Test Management
Samsung Static Analysis Tool	Static Analysis, code quality verification
Protex™[2]	Verification of the usage of open source components
Spark©[7]	Instant messenger

Table 1: SRBR SW Tools

SRBR SW Development Tools Framework has been used by each project based on project strategies and characteristics. All tools are integrated and customized to SRBR SW Development Process needs. If SRBR SW Development Process changes, we need to analyze the impacts and, if necessary, provide changes in SRBR SW Development tools (improvements, new features).

2.3 SRBR DMZ Infrastructure and Management

SRBR DMZ Infrastructure

The SRBR DMZ Infrastructure was designed in order to attend to Samsung's security requirements. The solution and maintenance service (data center, servers) are provided by Samsung SDS organization. Inside DMZ all project data is protected and accessible only to permitted users. Below is presented an overview of this infrastructure, see Figure 3.

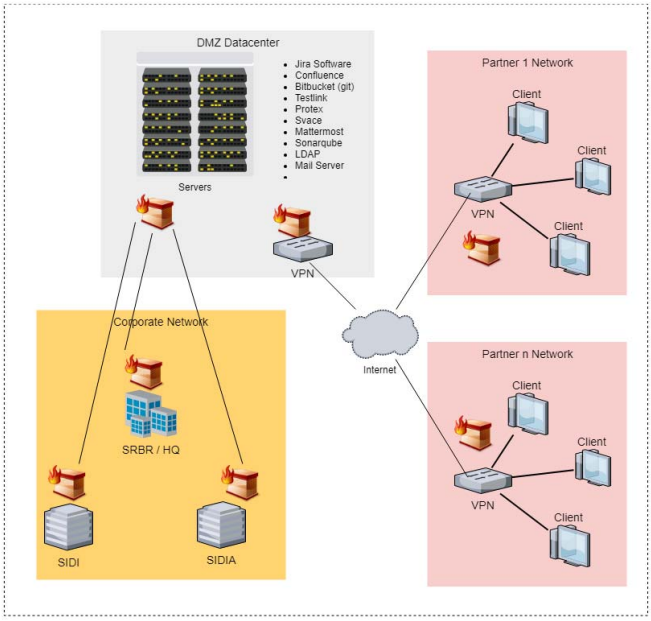

Figure 3: DMZ Infrastructure Overview

SRBR DMZ Environment Management

The SRBR DMZ Environment is managed and maintained by SRBR SE and strategic-partner-1 SE team located at Campinas and at Manaus, except the infrastructure that is maintained by SAMSUNG SDS team. DMZ environment team has five non-dedicated team members and they are responsible for:

- Exploring potential tools to be included in framework
- Selecting and evaluating potential tools
- Making tools acquisition
- Performing impact analysis for adopting new/replacement/upgrade tools
- Installing and configuring new/replacement/upgrade tools
- Integrating and/or developing new tools
- Creating and maintaining tools administration and user manuals
- Providing support and training to users
- Maintaining backups
- Managing users account
- Creating/configuring project environment

SRBR SW Development Process and Tools Baseline
Any user of SRBR SW Development Process and SRBR SW Development Tools Framework from SRBR team, strategic partners or outsourcing partners can report a bug or request an improvement or new tool or process asset in SRBR Jira Service Desk. All requests are analyzed by SRBR SE team and Strategic partners' SE team. If approved, a request is planned to be incorporated into a new SRBR SW Process or Tools baseline. When it is implemented it is released when the new baseline is published.
This permits all organizations in SRBR DMZ to contribute on environment, share lessons learned across projects and organizations and incorporate it in the environment following a formal process.

SRBR DMZ Project Environment
The project environment is configured based on project needs. When a new project starts, the SW Engineering team participates in a Kickoff meeting. During this meeting, the software process is tailored and is defined the set of software tools to be used. SRBR SE team configures the environment and provides the access to project team. The standard project environment configuration is composed by the following tools:

- Confluence Project Space: project documentation organized following the SW process templates
- Bitbucket/Git Repository
- Jira project with stories, tasks, bugs, risks, audit workflows

The other tools, such as Jenkins, Static Analysis, Testlink are configured depending on project characteristics.

3 RESULTS AND DISCUSSION

Since 2014 when DMZ was created, SRBR has been working with Strategic partners to create the SRBR SW Development Process and Tools Framework, selecting, evaluating and deploying new practices and tools in environment to improve the quality of results provided by partners, and the visibility about the progress of the projects. After three years, we have a consolidated process, a set of tools to support it and we are in continuous improvement. The partners have recognized the value of the DMZ environment and they are engaged in delivering projects results in adherence with Samsung's requirements. The benefits of SRBR DMZ establishment include:

- All projects independent of organization are following the same process and the results and deliverables are using the SRBR standard templates.
- The project' documentation and project' status can be accessed any time by any team member from Samsung, including SRBR project sponsors located in Korea.
- Best practices of one project, or one outsourced organization, can be shared and incorporated in the process easily.
- Efforts in activities that do not add value, such as: meetings, time spent to transfer files, replication of SW development environment, outsourcing management was reduced.
- Usage of open source or commercial tools facilitated the adoption by partners teams.

During the establishment of the DMZ environment, we have faced some challenges:

- We found a kind of resistance from our partners' side to move from their own internal environment to the Samsung DMZ, where the use of this environment was not explicit in terms of contract (statement of work). Also, sometimes, partners used to work with different tools, and DMZ tools were not used due to constant follow-up of project tasks details.
- Effort and time to deploy VPN solution via hardware to partners. We have adopted a HW VPN solution as suggested by the SDS team to be adherent to security requirements, but it demanded a lot of effort to manage HW transportation and logistics, to install that equipment inside partnersâĂŹ datacenters.

3.1 What we could have done better?

We could have discussed our design more deeply, and insisted on adopting a SW solution for VPN instead of a HW solution, so it could have relieved a lot of effort and time related to routers, logistics and physical configurations inside our partners' networks.
To deal with our partner's resistance, we could have improved the contract terms to consider using the development environment as defined by SRBR, and also we could improve the relationship with our partners so they could be more involved with the DMZ environment as a contributor, helping to take decisions.

3.2 Next Steps

Our next step is including new tools such as SonarQube, Artifactory to improve productivity, security and enhance self-verification activities. Our challenge is to promote agile methodology and a quality culture and increase the maturity of developers to check and assure that they can provide better code and better project documentation. Also, we have plans to replace TestLink and Spark Instant Messenger tool. We expect to have a Test Management tool integrated to Jira and improve test case revision workflow, Spark will be replaced by Mattermost a good web interface instead of desktop installed client.

REFERENCES

[1] Atlassian. 2017. Atlassian Products. (2017). Retrieved March, 2018 from https://www.atlassian.com/
[2] Black Duck Company. 2018. Protex tool. (2018). Retrieved March, 2018 from https://www.blackducksoftware.com/
[3] IEEE/IEC. 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
[4] MCTIC. 1991. Brazil Informatic Law - Lei 8.248, from October 23, 1991. (Oct. 1991). Retrieved March, 2018 from http://www.mctic.gov.br/mctic/opencms/tecnologia/incentivo_desenvolvimento/lei_informatica/_informacoes/a_lei_informatica.html
[5] MCTIC. 2005. Lei do Bem - Lei 11.196. (nov 2005). Retrieved March, 2018 from http://www.mctic.gov.br/mctic/opencms/tecnologia/incentivo_desenvolvimento/lei_bem/_bem/Lei_do_Bem.html
[6] Jenkins Project. 2011. Jenkins. (2011). Retrieved March, 2018 from https://jenkins.io/
[7] Ignite Realtime. 2017. Spark Instant Messenger. (2017). Retrieved March, 2018 from https://igniterealtime.org/projects/spark/index.jsp
[8] Suframa. 1991. SUFRAMA âĂŞ Superintendencia da Zona Franca de Manaus âĂŞ Research and Development (R&D) Amazon Incentive Law. (1991). Retrieved March, 2018 from http://site.suframa.gov.br/assuntos/pesquisa-e-desenvolvimento
[9] Testlink Dev Team. 2013. Testlink. (2013). Retrieved March, 2018 from http://testlink.org/