

On the Identification of Best Practices for Improving the Efficiency of Testing Activities in Distributed Software Projects

Preliminary Findings from an Empirical Study

Cristina Camacho
Software Tester
Porto Alegre, Brazil
cristinacamacho@gmail.com

Sabrina Marczak
PUCRS University
Porto Alegre, Brazil
sabrina.marczak@puers.br

Tayana Conte
Universidade Federal do Amazonas
Manaus, Brazil
tayana@icomp.ufam.edu.br

Abstract—Testing aims at identifying whether the software product attends its specification and the customer needs. It is a collaborative task since it requires coordination and communication of activities with those responsible for defining the project scope and for developing the source-code in order to clarify requirements, to stay up-to-date about changes, and to coordinate working schedule. It is a challenging but necessary activity in any software development life cycle. Distribution exacerbates the difficulties faced by testing team members. This paper presents empirical exploratory preliminary findings on challenges faced by testing teams in distributed projects and discusses best practices adopted by such teams to facilitate coordination and, as a consequence, increase the efficiency of their work. Our findings suggest corroboration of preliminary literature that focuses mainly in requirements engineering and in development activities. We aim to replicate our investigation in a larger scale. Meanwhile, our work provides initial contributions to practitioners and inspires our future research.

Keywords— software testing; challenges; best practices; distributed software teams; empirical study

I. INTRODUCTION

Testing is performed to evaluate and improve product quality by identifying defects. Software testing consists of the dynamic verification of a program's behavior on a finite set of test cases suitably selected from the usually infinity executions domain, against the expected behavior [1]. It can be performed at different levels—unit, integration and system, along the development life cycle and can focus on different aspects such as purpose, use, behavior, or structure of the system.

Software testing requires an understanding of the product behavior through the comprehension of its requirements. It also involves negotiation and planning of what has to be tested and how. Because of changes to scope, it requires a systematic change management approach and a continuous refinement of testing specifications. Once test cases are executed, defects should be reported and assigned to those responsible for fixing them. Tracking procedures need to be in place to manage defects until they are resolved. Therefore, testing is a

collaborative activity in which team members performing different roles in a project are expected to provide information for or resolve requests from the test team. Constant communication and coordination are necessary to successfully accomplishing the activities and ensuring the product quality.

Because of its collaborative nature, testing is a challenging activity. Murugesan [2] named some challenges faced by testing teams such as lack of user involvement, lack of management support, improper staffing, aggressive schedules (e.g., pressure for a sign-off next to the product launch deadline to production), reduction in testing time due to delays from designing and coding, and poor supporting documentation.

Distribution is known to increase difficulties in communication and coordination throughout the life cycle [3] and testing activities are not an exception. Time zone, culture, language, and working practices are some of the factors associated to the challenges commonly faced by distributed software teams.

We sought to empirically explore what are the challenges faced by test teams working in distributed settings and what are the best practices they adopt to minimize the impact of such challenges increasing the efficiency of testing activities. Therefore, this paper presents early findings of an empirical investigation of a large IT multinational company that develops software in-house to support its businesses processes. We expect our preliminary results to motivate researchers and practitioners to better look into the overall working practices of distributed development teams to help testing professionals to more effectively perform their work in distributed settings.

II. COMPANY BACKGROUND

We investigated a large IT multinational company with offices spread out worldwide. Development teams are distributed in three locations: in the headquarters' office in the US, in Brazil and in India. The IT department follows a matrix organizational structure based on business areas (e.g., sales, manufacturing) and IT functions (e.g., requirements analysts, developers, testers). Therefore, each software project has a

team allocated to serve the project during its development and the team is dismantled when the project is over. Team members are either employees or contractors.

Projects vary from the development of new products to the maintenance of legacy systems, including the integration of applications and the customization of commercial packages. Iterative and incremental is the most followed development model. Processes can vary from formal (following CMMI Level 3 practices) to informal (defined by project according to the identified needs), and by business area.

Project scope is defined based on prioritized business needs identified with the help of business representatives and requirements analysts, and project schedule is negotiated among business representatives, project managers, and functional managers. Once the project goal and overall schedule is agreed, requirements analysts move on specifying the software requirements. Development team leaders and architects work together to define the technical solution and the developers to code it. The test team contributes by defining the testing strategy and carrying out the tests themselves. Development and testing environments are provided by the infrastructure team who serves the entire IT organization (their work in hours is not accounted to the project). Database infrastructure follows the same solution. Tool adoption is an organization-wide decision.

More specifically, the test team is composed of test managers, test leader, test analysts, and testers. Test managers are responsible for defining a detailed schedule for the testing activities and coordinating it with the project manager. They are also responsible for requesting the test environment to the infrastructure team and assuring it is available in a timely manner. Test leaders are accountable for understanding and analyzing the requirements and for defining the testing strategy and deliverables. Approval with stakeholders and changes to testing documentation when applicable are also of this role's responsibility. Test analysts are responsible for designing, developing, and automating test cases. Testers are mainly responsible for running the test cases followed by reporting, assigning, and tracking resolution to identified defects.

When a change takes place in the project, it is propagated and communicated to others in various ways. For instance, documentation should be updated and shared via e-mail or via the project repository, in a weekly project meeting, or through synchronous communication via conference call or IM.

III. METHOD

Our empirical study consisted of a series of interviews conducted both on-site and remotely over the phone. Interview subjects were recruited by convenience as well as word of mouth (snowball) by the author who works at the investigated company as a tester. She announced the invitation at the monthly meeting of the testing community of practice. This group meets periodically to discuss new testing technologies, working processes, and best practices aiming to quickly spread out knowledge among testing professionals. Participants voluntarily agreed to participate in our research. Some participants suggested colleagues at the end of their interviews. Some of these indications resulted in additional participation.

Each interview lasted for an average of forty-five minutes. We asked the participant to provide personal information and experience background. We also asked the participant to respond to our questions based on the projects she has been involved in the past three years. The participant was then asked to describe the most relevant challenges faced to perform her activities and which are the best practices that have been adopted to overcome them. Last, we asked the participant to provide overall comments about her perception of the role that distance played in defining the challenges and on putting the best practices in place in the projects she worked on.

A total of sixteen interviews were conducted however three were discarded due to incompleteness. All remaining thirteen study participants were based in Brazil. Eight are company employees and five contractors for two distinct provider companies. The participant pool consisted of 9 male and 4 female respondents. Participants had an average of 5 years' experience working in distributed teams and 3.3 years' experience in the organization. Overall, participants had an average of 10 years of overall work experience, 7 years of software engineering experience, and 3.3 years' experience working as a testing professional. The participants' roles in the testing team are: 3 test leaders, 6 test analysts, and 8 testers. Most of the participants (11 of them) work with the testing of new products and of legacy systems.

All interviews were transcribed, and transcriptions were prepared for analysis in the ATLAS.ti qualitative data analysis software. Our subsequent analysis was guided by Grounded Theory procedures [4]. We coded the interviews, identifying the main concepts and their properties and dimensions, until we could define a set of main challenges reported by our respondents. Best practices were then grouped per challenge as per their original citations.

IV. PRELIMINARY FINDINGS

In this section we provide a brief outline of our preliminary findings listing the main challenges and associated best practices reported by our respondents.

- (1) *Long defect reports lacking focus on their description.* Participants claimed that most of the defect reports are long and lack focus on the most important item of the report, the issues found. In their opinion this makes asynchronous communication between testers and other project members difficult. Some testers also ask their colleagues to write in a sharper and a more concise way to allow for fast comprehension and response from the remote colleagues. They also suggest the standardization of defect reports using guidelines to write them. For instance, they propose a brief introductory sentence to state the defect found followed by a brief summary. Technical details should be included last allowing developers to replicate the steps to identify the defect themselves. In addition, they suggest the inclusion of the printout of where the defect was found as supplementary information. This allows for developers to quickly grasp what the defect is about and to provide a more fast response to it.

- (2) *Difficulty communicating with remote colleagues due to the lack of a common ground and shared vocabulary.* Some participants mentioned that it requires additional effort to communicate with colleagues playing distinct roles from the test team due to the lack of a common ground or shared vocabulary. They suggest explicitly defining terms and discussing vocabulary as early as possible in the project and sharing such definitions with everyone involved in the project despite team or role. This includes involving the customer and business analysts too. They mentioned it facilitates their understanding of the requirements and as a consequence allows the test team to do a better work defining test cases and testing the application.
- (3) *Lack of availability of remote colleagues to discuss the identified defects.* Respondents complained of not being able to synchronously reach a remote colleague because of the long hours of separation to discuss a certain defect causing it to be, sometimes, poorly reported. They also said that defects are frequently not fixed properly for the same reason. Detailed reports and fixes is the immediate suggested action for the participants to minimize the time differences. They experienced a more independent work from the test team perspective when defects were better written.
- (4) *Difficulty gathering data to test applications from remote colleagues.* Most of the data needed to test applications is prepared by Easter Indian colleagues, and because of the over 9 hours of separation often Brazilians need additional data or clarifications when the colleagues are already done for the day. To avoid a one-day delay or long asynchronous discussions about the datasets, some testers reported that they negotiate shifting working hours allowing for synchronous conversations. Another practice adopted is to kindly and clearly request colleagues to read the text messages thoroughly increasing the understanding of what has been requested. They mentioned that this has helped them to reduce the number of poorly provided datasets.
- (5) *High requirements volatility causing documentation to be obsolete.* Respondents reported that requirements change constantly despite approved scope agreements. Documentation is not updated in a timely manner letting the testing team working based on obsolete documentation. Therefore, test cases are often written based on out-of-date specifications. They suggest that requirements analysts should make a more thorough work with business representatives in earlier phases to avoid constant changes. Also, requirements analysts should try to adopt requirements management tools to track changes and automatically propagate notifications to those involved. Downstream artifacts can be then timely updated. In addition, the adoption of issues tracking tools is also relevant to help testers to communicate with developers and quickly manage changes to documentation.
- (6) *Poorly written documentation to support testing activities.* Requirements documentation is often written in a very high level description offering a poor understanding of the application definition and expected behavior. Such poor documentation leads the testing and development teams to disagree in the understanding of certain functionalities reaching deadlock situations in which they need to contact business representatives to clarify the requirements. Such situations waste time and effort from those involved. To avoid poor definitions, participants request requirements analysts to better detail the requirements even when information is not available from the business representatives. This avoids disagreements later on and a more accurate understanding of the application behavior.
- (7) *Difficulty identifying non-functional testing requirements.* Participants reported that they often have difficulties gathering testing requirements, mainly for performance tests, from business representatives due to their lack of technical knowledge. Requirements analysts are responsible for gathering such requirements. Therefore, participants suggest that test analysts could interact earlier on in the process with requirements analysts and help them collect such requirements from the business representatives facilitating the definition of test requirements and, as a consequence, allowing for a more precise development of the testing activities and ensuring better results. In addition, respondents also reported they have difficulty making the development team understands what testing requirements for performance tests are causing confusion and delays in closing defects related to such requirements. They suggest that development members should be trained on how to understand testing needs to provide better information to the testing team.
- (8) *Poorly prepared datasets for performing tests.* Some respondents described situations in which available datasets not always attend the testing requirements for a certain application (e.g., data does not comply with application format or data cannot be loaded due to technology incompatibility) and that the creation of new datasets by the development team is time-consuming and hard to be achieved in a short-time spam window. Participants suggest having a dedicated team responsible for preparing data for testing purposes. This team should work closely with the business representatives and be responsible for checking whether the applications are running and data is accurate.
- (9) *Difficulty gathering information required for preparing test environment.* Respondents reported that time is wasted verifying and fixing the test environment, which has to be compliant with the production environment, due to lack of appropriated information and resources for the infrastructure team to prepare such infrastructure. They propose the organization could allocate dedicated resources to manage testing environments per project instead of having a shared serviced across business areas and projects. This would suffice to promote efficient discussions and would allow for actions to be quickly taken avoiding idle time from the test team.

V. DISCUSSION

Testing activities are undeniable necessary and of value to any software project. Despite its recognizable need, software teams still suffer from mismanagement, lack of appropriated resources, and misunderstandings of project scope, among others. Communication and coordination issues are the consequent implications of such challenges. The now so common distance among team members only exacerbates the challenges faced by testing teams. This work aimed at empirically investigating what are the challenges and best practices adopted by testing professionals of a large IT company. We interviewed 13 participants who play diverse roles in the company and learned from their experience.

Our preliminary findings highlight challenges faced by several sorts of reasons. For instance, there are challenges related to *communication* difficulties: long defect reports lacking focus and lack of common ground and shared vocabulary (challenges 1 and 2, respectively). We know that distance introduces barriers to informal and face-to-face communication (e.g., [3]) and increases the inherently difficult information exchange among members playing different roles in a software development project. However, practices like standardization of defect reports and a formal definition of a project vocabulary have been helping testing members to better receive from and share information with project colleagues.

Two of the challenges are related to *time zone* differences: lack of synchronous availability of remote colleagues and difficulty gathering data to perform tests (challenges 3 and 4, respectively). We also know that time zone differences can cause difficulties (e.g. [5]). Non-overlapping working hours can cause distributed team members to never having the chance to discuss matters in a synchronous fashion increasing misunderstandings and difficulties in coordination. It can also cause delays [5] jeopardizing the conclusion of tasks on time and with the expected quality. Despite the imposed challenges, testing members reported that better detailing the defect reports and negotiating shifting working hours has been helping them to couple with the faced challenges of availability due to time differences. Both communication and time zone issues are classic reported challenges in global software engineering [3].

Requirements volatility, obsolete documentation, and poor requirements specification are well-known software engineering challenges faced by software teams for decades [6]. Such findings suggest that despite the current advancing stage of tool support—integrating developing, communication, and management environments, testing teams are still somehow disconnected from the development teams and suffering of the lack of efficient solutions for such difficulties.

We know that knowledge about the organization and its defined processes, about the business and product domain as well as awareness about colleagues and their preferences are important for achieving the project's goals [3]. Therefore, we can say that challenges 8 to 9 can be related to knowledge management. Testing datasets and testing environments are required but poorly provided. Imposed organization structure that does not allocate dedicated resources for supporting the needed preparation for testing activities might explain such difficulty. Despite the root cause, testing members have been

overcoming these challenges by helping the requirements analysts to work more thoroughly and closely to the business representatives, by adopting management tools that help automate notifications, by training other members on how to understand testing requirements, and by requesting the organization to change the allocation of certain roles to better support the testing teams. These initiatives have been assisting them to minimize their weaknesses and helping increasing the efficiency of the activities that have to be performed.

VI. FINAL CONSIDERATIONS

This paper presents preliminary results of an empirical study aiming to investigate the challenges faced and best practices adopted by testing teams working in distributed environments aiming to identifying what helps them overcome the difficulties increasing the efficiency of testing activities. We identified that the challenges are mostly related to communication, time zone differences, classic software engineering challenges, and knowledge management, well known factors from literature that influence the performance of distributed software teams. This reinforces the need to understanding testing as a continuous activity throughout the life cycle and that it needs constant collaboration and support from all those involved in developing software.

We expect these preliminary findings to motivate other researchers and practitioners to take a better look in software engineering practices and to help testing teams to better define and execute their responsibilities in the project. We understand that our preliminary findings are limited (e.g., low number of participants, single company) and need to be confirmed. We plan to conduct additional interviews with practitioners from other companies and with distinct characteristics to minimize the impact of certain variables such as experience on the job, experience working with distributed teams, role played, and language expertise on the responses provided. We also aim to survey a larger population to confirm the results identified in our second round of interviews.

ACKNOWLEDGMENT

We would like to thank the PDTI Program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91) and the participants for their collaboration and time.

REFERENCES

- [1] A. Bertolino, "SWEBOK: Guide to the Software Engineering Body of Knowledge, Guide to the Knowledge Area of Software Testing," 2004 version, chapter 5, IEEE CS, 2004.
- [2] S. Murugesan, "Attitude towards testing: A key contributor to software quality," in Proc. of the International Conference on Software Testing, Reliability and Quality Assurance, 1994, pp.111 -115.
- [3] J. Herbsleb, D. Moitra, "Guest editors' introduction: Global software development," IEEE Software, vol. 18, no. 2, 2001.
- [4] J. Corbin and A. Strauss, The basics of qualitative research: Techniques and procedures for developing grounded theory. Sage Publications: Los Angeles, 3rd ed., 2007. 400 p.
- [5] E. Carmel, I'm working while you are sleeping: Time zone separation challenges and solutions, Nedder Stream Press: New York, 2012.
- [6] D. Damian, D. Zowghi, "RE challenges in multi-site software development organisations", Req. Eng., vol. 8, no. 3, pp. 149-160, 2003.