2015 IEEE 10th International Conference on Global Software Engineering

Continuous software testing in a globally distributed project

Nils Brede Moe, Daniela Cruzes, Tore Dybå
SINTEF
Norway
nilsm@sintef.no, danielac@sintef.no, tored@sintef.no

Edda Mikkelsen
DNV GL Software
Norway
Edda.Mikkelsen@dnvgl.com

***Abstract*—In globally distributed software projects the testing expertise may be scattered across multiple locations. We describe and discuss a globally distributed agile project at DNV GL Software, a multinational provider of software for a safer, smarter and greener future in the energy, process and maritime industries. DNV GL Software is headquartered in Norway. The project is distributed across two locations with 12 team members in Norway and three testers in China. In a distributed agile team with little overlap in working hours the challenge is to coordinate tasks and test activities in a way that makes coordination and communication efficient. DNV GL Software believes in including the remote testers as part of the agile team, enabling self-managing, cross-functional virtual teams that are capable of taking the full responsibility for implementing and verifying one entire feature. To support the communication between testers in China and the rest of the team in Norway, the team needs a shared understanding of the goal of a release and how to collaborate.**

We conducted interviews with the team and representatives from different roles in the organization, and we performed retrospectives with the team. In this article we describe how continuous testing based on continuous and frequent feedback ensures knowledge sharing and safeguarding the quality of the system under development. We found the following enablers for a successful virtual agile team: coordination by mutual adjustment, dedicated testers and low turnover, shifting working hours, and self-management and autonomy. Non-technical factors, such as socio-technical and organizational factors, have a significant influence on the way software testing is performed in an agile virtual team. To be successful the organization needs to invest in bringing the remote testers closer to the rest of the team, as part of the virtual team.

Keywords—testing, test centers, agile, virtual teams, distributed agile

I. INTRODUCTION

Many industry reports, research papers, and other resources discuss the advantages and disadvantages of adopting global software development practices, such as outsourcing and offshoring [1, 4, 5]. The majority of empirical research present problem-oriented reports with few reports focusing on solutions and, particularly empirically evaluated solutions [7]. There is also a lack of studies related to processes, methods and tools for testing and quality assurance in global software development [6, 7]. At the same time, companies need new knowledge on how to implement their testing processes in a global context.

The experience reported in this paper is based on a virtual project team at DNV GL Software. DNV GL Software is headquartered in Norway, and is a multinational provider of software for a safer, smarter and greener future in the energy, process and maritime industries. DNV GL Software has software developers and testers in several countries, including Norway, USA, Malaysia, UK, Germany and Poland, and a testing center in China.

The main motivation for having the testers in a test center in China is that DNV GL Software can get access to highly skilled and low cost resources. In addition, because the testers are in a different time zone, testing can happen when the developers are sleeping, maximizing the flow in the project. However, even though the test center has a skilled and stable workforce, there are challenges, such as an extensive communication and documentation overhead caused by the remote testing.

DNV GL Software has decided to adopt continuous testing, which involves immediately integrating changes into the main system, continuously testing all changes and updating test cases to be able to run a regression test at any time to verify that changes have not broken existing functionality. It is in general infeasible to implement this kind of approach in many contexts because it requires more interactions between the developers and the testers, which is a challenge for virtual teams. Continuous testing increases a need for improved communication and coordination between testers and developers.

In this paper we describe DNV GL Software's approach for continuous software testing in a globally distributed virtual team and discuss the enablers, challenges and main lessons learned. The virtual team consists of developers and testers in Norway and testers in China, working across locational, temporal, and relational boundaries to fulfill the team's goal.

In Section 2 we introduce the background on agile testing in distributed environments. In Section 3 the project under study is described in more detail. Section 4 presents our findings, while Section 5 concludes the paper.

978-1-4799-8409-1/15 $31.00 © 2015 IEEE
DOI 10.1109/ICGSE.2015.24

II. AGILE TESTING/DISTRIBUTED TESTING

DNV GL relies on agile software development and one goal is to implement agile testing. Agile testing does not just mean testing on agile projects, but testing an application with a plan to learn about it and let the product information and customer feedback guide the testing in line with the agile values of working software and responding to change.

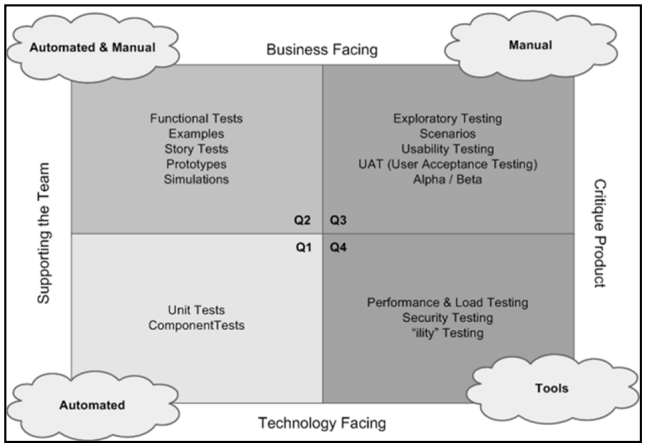

Figure 1 Agile test quadrants [2, 3]

Different tests have different purposes [2, 3]. Each quadrant in figure 1 reflects different reasons to test. Traditional software testing focuses almost exclusively on the right hand side (Q3 and Q4), criticizing the product, but not playing a productive part in supporting the creation and guidance of the product (Q1 and Q2). Moreover, traditional software testing is involved late in the development process to detect failures but not to prevent them. In agile testing, the testers are not only involved in identifying, but also in preventing failures by regular interaction with developers and customers. Automation is an important enabler for agile testing. Automation of the tests in Q1 is usually easiest to implement for a team, and at the same time makes a big impact on the process effectiveness. Automated tests in Q2 are more challenging and costly to maintain than automated tests in Q1. And tests in Q3 are usually done manually. In this context, as in other phases of global software development, geographical and temporal distribution makes regular interaction challenging [1].

Nowadays, rather than deciding whether or not to get involved in global testing, many companies are facing decisions about whether or not to apply agile testing methods in their distributed projects. Yet, few papers have addressed the solutions and challenges in global software testing. Collins et al. [2] describe an industrial experience on the application of distributed testing in a software development environment following Scrum. The authors report on the adaptation in a testing process to make feasible the distributed tests in an agile environment, the solutions implemented to reduce the impact of communication issues and the difficulties observed in the allocation of tasks in the software project. Collins et al. identified four main challenges and key lessons learned for minimizing the impact of the geographical distance between the testing teams:

a) *communication and coordination are essential factors for the success of distributed testing;*
b) *the project information should be available with details to all members;*
c) *automation reduces the needs of physical presence in the testing process;*
d) *supporting tools are always important, but testing team organization is more.*

Shah et al. [6] conducted a study of three vendor-side testing teams. Their findings show how the participant test engineers perceive global software testing (GST) and deadline pressures, the challenges that they encounter, and the strategies that they use for coping with the challenges. The authors highlight the need for (1) appreciating test engineers' efforts, (2) investigating the team structure's influence on pressure and the GST practice, (3) understanding culture's influence on other aspects of GST, and (4) identifying and addressing quality-dilemma situations.

III. CONTEXT SETTING

The experience reported in this paper is based on interviews, a retrospective and observations of a team consisting of 15 members (12 in Norway) including 4 testers (3 in China, one in Norway) and one test manager (in Norway). Some members are working part-time in the team. Both the Norwegian team-members and the Chinese team-members are sitting together making it easy for them to communicate and coordinate work locally. The team follows a Scrum-based process, where a release is 2-4 months long. The objectives of this process are to be a common platform for communication and understanding of the software testing processes; clarify responsibilities, milestones/tollgates and expected outputs; form the basis for project governance; and, be a high level checklist for projects.

The continuous software testing process at DNV GL Software is described in Figure 2. In the Planning phase, the main objectives for the testing are to develop a high level test strategy for the product, and a test plan for the releases, and to decide on how to follow up the project (test environment, resource, schedule/milestones). In the Development and Testing phase the main objectives are to design and refine test cases for the features (user stories / enhancements) within the scope and to execute and report the test according to the test plan. This phase is typically accomplished in several sprints. In the Stabilizing Phase, the main objectives are to perform a complete system functional test, system integration test, and relevant non-functional tests in a test environment similar to the production environment. Typically, several runs of system / non-functional test are accomplished during the stabilizing phase. Each run of the system test is done in a new release candidate. When the acceptance criteria are matched according to the test plan, the project is released and deployed. After the release, some of the team members start a new release while some do support and others do bug fixing.

The tester in Norway is a newly hired person fully dedicated to the team. At the time of the study, he is still trying to learn the system under development. In China, two testers

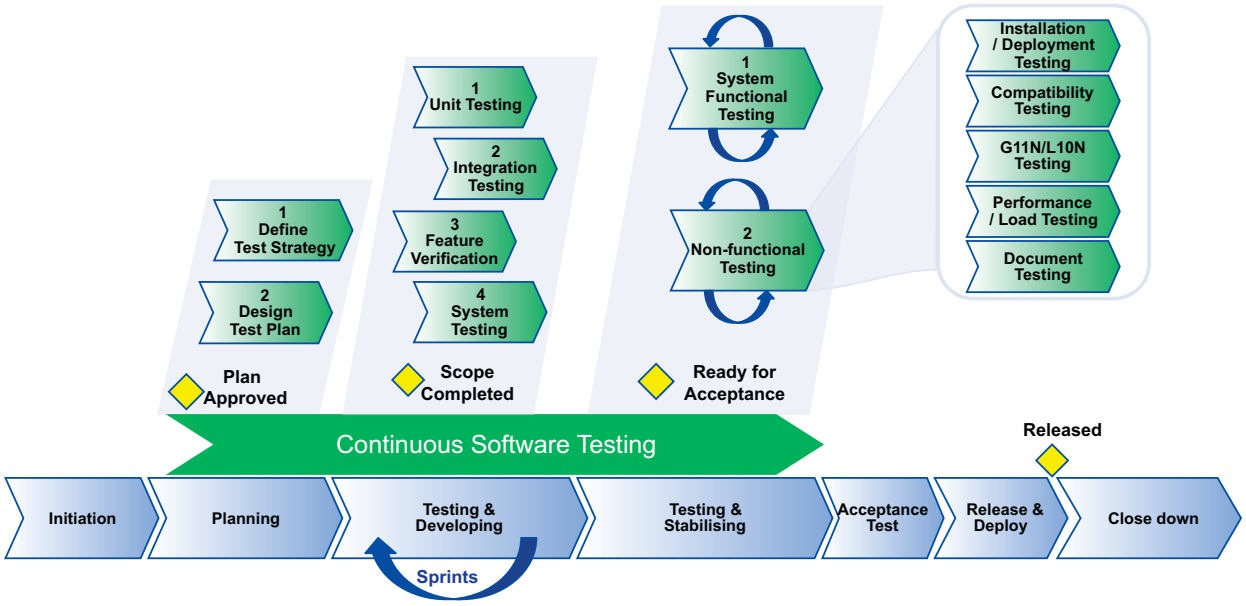

Figure 2: the development and testing process

with four and one year of experience on the product work 100% on the team and one tester with six years of experience on the product is working 50%.In Norway, the test manager is the one responsible for planning, communicating and organizing the test work between the team members in Norway and the testers in China. Working with China takes 40-50% of her time. The test manager is also working with the support department, and she acts as buffer between testers and developers. She is also engaged in the user acceptance testing with the customer.

Most of the testing in China is performed manually by following test scripts developed and maintained by the Chinese testers. When new functionality is to be implemented in the software, the team uses some time to ensure that everyone understands what is being developed, so that the test scripts can be updated. While test automation is seen as important in agile testing, it is not widely implemented in the testing activities at DNV GL Software. So far, the automatic user interface tests have a very short "life". The user interface is changed frequent and then the tests are broken. According to the testers, it takes too long to update the automatic tests. In addition, the ability to automate tests depends on the knowledge of the tester. The tester responsible for implementing automatic testing is currently on maternity leave, which is one reason for why the level of automation is low at the time of investigation.

Furthermore the Chinese testers are also responsible for integration testing and performance testing. The system integrates with several external systems (e.g. SAP), and when functionality is added or changed there is a need to run integration tests. Performance is important; however, performance testing is not needed for every release. When important modules, components or the database are changed, performance testing is scheduled. The team-lead and one architect take the decision regarding performance testing. The testers in Shanghai use designated tools when they are doing performance testing.

IV. Findings

The management perceives the team as highly effective, and the team is known for delivering on time in a complex and turbulent environment. When we asked the team why they are successful, they answered (in a retrospective):

"*The main reason is that we are a highly skilled team that is flexible and working agile. We have a focus on improving collaboration and our working process, and we are good at testing.*"

When investigating the key enablers for creating a good team with remote testers, we found the following:

- Coordination by mutual adjustment;
- Dedicated testers and low turnover;
- Shifting working hours; and
- Self-management and autonomy.

A. Coordination by mutual adjustment in the virtual team

Teams with remote testers often coordinate work by writing detailed documentation. This team encouraged mutual adjustment over documentation, which enables coordination by informal communication.

Testing is performed continuously during development and stabilizing. In the development phase a developer picks tasks in Jira. The task is implemented and the developers run unit tests. When the task is completed, Jira is updated. Each morning the testers in China check out changes in Jira to understand what they should test. Some developers write a very detailed Jira

description while others write only a very short text, which makes it challenging for the testers to understand how to run the test. However, the varying levels of details in the specifications are not seen as a problem by the team management. The team leader explained:

"*If you write very little then the remote testers need to ask. And then the team-members start communicating more often. Frequent communication is needed to achieve success. If you write too much it is more likely that they [the testers] do not ask, even if they do not understand it well enough.*"

The test manager also commented on this matter:

"*Some [developers] write understandable descriptions in Jira while others write only a few words, which are then difficult to understand. You should not try to standardize the way of writing in Jira, you should continuously focus on reminding people to ask questions. It is better to have a culture where people ask questions, than a culture where people use a lot of time trying to write good documentation."*

Because the team management encouraged talking over writing detailed documents, the testers and developers communicated frequently (across temporal, cultural and geographical distance), which resulted in a culture based on mutual adjustment. While basing the handover between developers and testers on frequent communication was found to be beneficial, the strategy also introduced some challenges. When a tester is missing important information about the new feature to test and can´t get immediate feedback from a developer because he or she is busy, or because of times zone issues, there is a delay in the process. Then a tester might end up wasting time trying to find documentation for that feature (that might be incomplete or missing), or waiting to the next day before continuing on testing the new feature.

B. Dedicated testers and low turnover in the virtual team

Because the remote testers are working mostly on one development project, they are usually always available and can immediately start testing a new feature. Because of the time difference, tests usually are run in China while the Norwegians are sleeping. The Chinese testers follow the test scripts and tests all combinations of how the user might interact with the system. Bugs are then continuously reported. Each morning, the team leads and/or the test manager in Norway check new bugs that have been reported to understand if they are 'real' bugs and what to prioritize.

When a bug is identified and prioritized, the bug is handed back to the developers to fix. After the developer has solved the issues, the bug is handed over to the testers for verification. This frequent feedback enables a quick verification, fast flow and a constant control of the quality. In addition, when everyone involved got a clear understanding of the tasks, the waiting time is reduced because team members are in different time zones.

While the time zone is a benefit, it is also challenge. There is only a short time with overlapping working hours, and when the Norwegians go home, the Chinese testers cannot get help or answers to their questions. In theory, there is an overlap between 07am and 11am Norwegian time, but in practice, the Chinese work with the Norwegians for only one hour (between 9am and 10am).

Encouraging each other to ask questions is important when relying on mutual adjustment and frequent feedback. The Chinese testers are perceived as good in asking their Norwegian teammates questions, and this skill has been developed over years. The main enabler is the stabile workforce. While the team has a relatively low turnover there have still been some changes.

When a new person joins the testing team, there is a need for putting more effort into coordination of tasks and communication between developers and testers. The developers notice a little drop in the communication, because new people joining in China seem to communicate less frequent and ask fewer questions than the experienced ones.

Building trust to enable remote team members to ask questions takes time. When new members join the team they might be a bit shy, or they might strive to give a good impression resulting in less questions about the things they do not understand. A tester might not ask for help because he or she does not want to look incompetent. Having a high level of trust is necessary for achieving a frequent and open communication. Therefore, in every meeting and in all communication the test manager reminds the testers about the importance of asking questions.

C. Shifting working hours

The stabilizing phase is the most hectic period for the testers and the test managers.. When entering the stabilizing phase the goal is to run the manual regression tests as many times as possible. Usually there will be 3 rounds, each round taking 3-4 days. In this phase new bugs are found, bugs in code that has not been changed or tested during the sprint. Testing in this phase also covers testing for different devices (e.g. iPad and tablets), browsers and offline use. To be able to meet the deadline in this hectic period it is important that the testers are not delayed if they run into problems or got questions.

The nature of the stabilizing phase makes this phase very intense, and delays in this phase will cause a delay in the delivery. Therefore, the communication within the virtual team needs to be more frequent, and 1-2 overlapping hours are too little. In the stabilizing phase the developers in Norway may work long days and then it is important that the test environments are updated with the latest build before the Chinese testers come into office the next day. Information about this is thus sent late evening Norwegian time. As a consequence the test manager in Norway usually writes and answers e-mail from China every evening. This is necessary to reduce the delay and maximize the performance of the team when the release day is approaching.

D. Self-management and autonomy

A successful virtual team and its members need to have a high level of autonomy. The testers in China need to be able to make decisions while the team members in Norway are sleeping. The key enabler for high autonomy among testers in China is high domain knowledge (understanding of solutions to

domain problems) and ownership of decisions on the test scripts. Ownership also strengthens team commitment. High domain knowledge makes it easier for the Chinese testers to understand the various user situations, even though they are not part of user acceptance testing or do not have a close cooperation with the customers

When testers find bugs or when they identify questions they need to ask their remote colleagues in Norway, they first discuss among themselves to understand if it is a real bud and if they can solve the issue. Even though the testers and developers have been on the product for a long time, the testers sometimes find bugs that the Norwegian developers do not perceive as a bug. Then the Norwegian team-lead (who knows the technology and the technical system), and the test manager (who knows what happens in support and production), look at the test-report and discuss with the Chinese. The test manager explained:

"Sometimes we tell them [testers] that this is something the users will never do. Sometimes they protest and argue that the finding is very relevant. Then we have to look at it. It is good that they are able to give feedback and to speak up if they do not agree."

If the Chinese testers and the developers agree that it is not a bug, then the Chinese update the test scripts accordingly. Having such discussions and aiming at achieving a shared understanding enable shared mental models in the team and strengthens the autonomy of the Chinese testers.

V. Discussion and conclusion

From this study we see that socio-technical and organizational factors have a significant influence on how continuous testing in a global virtual team is performed. The main success factor for a continuous testing process is that the team has invested a lot of time and effort into bringing the developers and testers close to each other. Building one team. Our lessons learned are thus summarized in the following:

- Encourage feedback and questions over writing detailed specifications in the virtual team even though there are few overlapping working hours.
- Testers fully allocated on the team enable continuous testing and maximum flow.
- Autonomy and self-management are premises for a virtual agile team. To achieve autonomy and self-management the team needs full control over its testing resources.
- It takes time to build effective virtual agile teams, therefore the team needs to be stable.

The studied team experienced some time zone benefits and challenges. It is beneficial for the team that testing happens while developers are sleeping, because it enables continuous testing and integrates testing activities as closely as possible with coding. Firstly, errors can be fixed quickly while the context is fresh in the developers' minds and before these errors lead to knock-on effects. Secondly, the underlying root causes that led to the problems may be more easily identified and eliminated. In addition, it is beneficial to the team in a longer term, that testers do not get too close to the developers which make them report what they perceive as bugs, not what the developers want them to report.

The challenges related to the time zone differences are that when problems or questions emerge, the process might be delayed one day or longer because of few overlapping working hours. In hectic periods there is a need to extend and shift working hours to be more flexible which is a cost for some of the team members. Being in a different time zone makes it difficult for the testers to be part of the daily activities of the teams. Sometimes the testers miss knowledge on how the system is supposed to work and on what is expected when it comes to quality. The testers are also far from the customer, which makes it more difficult to understand the customer needs and to build domain knowledge.

To continue improving the testing and development activities, the team plans to invest more in the initial phase of each release. The goal is to make sure that everyone in the virtual team has a shared mental model of what will be delivered for the next release. The testers need to be more involved in the planning phase to get a better understanding of the goal of the iteration. Organizing early demos for the testers might be one option. Also the whole team need discuss more on how to improve the working process.

Acknowledgments. This work was supported by the Smiglo project, partly funded by the Research Council of Norway under the grant 235359/O30.

References

[1] N.-D. Anh, D.S. Cruzes, and R. Conradi, "The impact of global dispersion on coordination, team performance and software quality - A systematic literature review," Information and Software Technology, 2015, pp. 277-294.

[2] E. Collins, G. Macedo, N. Maia, and A. Dias-Neto, "An Industrial Experience on the Application of Distributed Testing in an Agile Software Development Environment," Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on, 2012, pp. 190-194.

[3] L. Crispin and J. Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Addison-Wesley Professional, 2009.

[4] N.B. Moe, D. Smite, G.K. Hanssen, and H. Barney, "From offshore outsourcing to insourcing and partnerships: four failed outsourcing attempts," *Empirical Software Engineering*, 2013, pp. 1-34.

[5] E. Ó Conchúir, P. Ågerfalk, H.H. Olsson, and B. Fitzgerald, "Global software development: where are the benefits?," Commun. ACM, 2009, pp. 127-131.

[6] H. Shah, M.J. Harrold, and S. Sinha, "Global software testing under deadline pressure: Vendor-side experiences," Information and Software Technology, 2014, pp. 6-19.

[7] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," Empirical Software Engineering, 2010, pp. 91-118.