SOFTWARE ENGINEERING IN GLOBALLY DISTRIBUTED TEAMS

Manasés Jesús Galindo Bello

Department of Applied Computer Science, Hochschule Fulda, Germany

ABSTRACT

Software engineering principles are brought into practice by Information Technology companies all around the world. Software can be developed by local teams which members have different cultural backgrounds, as well as by teams distributed across countries. To save costs and be close to markets and customers, companies offshore or outsource the personnel. Although developing software in distributed teams offers multiple benefits, there are also stiff challenges that engineers and managers have to deal with, e.g. communication and collaboration may get affected because of geographic distance, different time zones and distinct cultural backgrounds among team members. If not addressed on time and effectively, these challenges generate misunderstanding and conflict among the team which eventually may impact the projects deadlines and quality of the software. This paper presents the most common software engineering practices, challenges and tools in global teams, as well as practical cases in the industrial and academic realms.

KEYWORDS

Collaboration, Management, Culture, Conflict, Challenges

1. INTRODUCTION

The Information Technology (IT) world is in constant change. Day after day more and more big companies as well as start-ups are relying on a geographically dispersed workforce to be able to quickly bring to the market innovative and enhanced products. Being agile to respond to the customer needs is key to succeed in the global economy nowadays. Therefore, companies build their teams in such a way that gathers the most skilled and qualified people from around the world and relying on the benefits of international diversity by bringing together high qualified personnel from different countries with diverse work experiences and perspectives.

Software development methodologies have their own well-known strengths and weaknesses [1], and over the years most of those methodologies have been improved. In software engineering, a software methodology refers [1] to the framework that is used to structure, plan and divide the software development work into distinct phases to improve design, product management and project management; and it is also known as software development life cycle. Each methodology has a different specification and it may include the creation of specific deliverables and artifacts to document, develop or maintain a system. One methodology that works for a project or a company, it is not necessarily suitable for another. The methodology shall be chosen based on different factors such as the nature of the project and technical, organizational and team considerations.

Natarajan Meghanathan et al. (Eds) : SOFE, ADCOM, ITCSS - 2018
pp. 01–16, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81301

In the latest years, companies have been evolving [11] from using traditional plan-driven software development to start experimenting with one or more agile frameworks, as well as opting to outsource or offshore the resources and collaborating with geographically dispersed teams. Diverse and numerous are the effects of physical distance [2] between software development teams. Proximity, in contrast to distance, promotes constant and face-to-face communication and the development of closer and more positive relations. In the same way, the continuous presence of colleagues improves feelings of familiarity and liking towards them, but on the contrary, physical distance avoids or significantly decreases affinity and fondness, both of which are oppositely connected to conflict [3]. The challenges by project managers leading globally distributed teams can turn to be complex and not easy to overcome. Collaborating successfully is not a simple task when every team member is local and even share the same working space; and when people come from different countries and cultural backgrounds, or when the team members are located in different time zones, communication can be inefficient and generate misunderstanding, cooperation can deteriorate, and at the end these factors create conflicts within the team members [4] and the project.

To communicate over physical distance, globally distributed teams have to rely heavily on computer and telecommunications technologies to mediate their interactions. Notwithstanding that such technologies enable remote team members to interact, it is practically impossible to transmit non-verbal language. Nevertheless, companies have migrated from using telephone conference bridges to video conferencing, which has earned its place amongst new online collaborative tools and it was recognized in 2016 among the top five collaborative trends [5].

One of the biggest challenges in globally distributed teams can be the communication between the team members. Their personal diversity of race, culture and backgrounds lead to some of these challenges that project managers encounter, and such diversity may create barriers that could not allow a sharp understanding of the objectives and tasks. English is commonly known as one of today's global and most dominant languages, and it is primarily spoken by those for whom it is not a first language. In globally distributed teams, English is used as a communicative tool to interact within and across international borders by those who do not share a common first language [6]. Even though the primary communication language is the same, pronunciation, regionalisms, idiomatic expressions and/or lack of practice and fluency make it difficult to communicate clear ideas to other team members.

Conflict in work groups is unavoidable and may happen in all hierarchy levels, between directives and/or team colleagues. Work groups can opt to see the conflict as a negative factor and act to solve it once it is present, or as a positive factor and use it in favor [7] to achieve beneficial changes to the team and the company. Therefore, conflict becomes a part of life and it can occur in almost any area, and it is of high importance to learn to manage it efficiently and effectively to reduce or avoid the stress levels. In every work group of every organization, each team member has different personalities, education, background, goals and opinions. Because of this, conflict in the workplace becomes inevitable. It is a normal, and even healthy, part of business relationships; after all, two individuals cannot be expected to agree on everything at all times. When conflict is not addressed and properly managed, it can deteriorate the team members' communication or a business relationship, but when it is addressed in an intelligent, respectful and positive way, it creates an opportunity for growth and ultimately strengthening the relation between two parts. Learning how to manage conflict efficiently is an important skill for anyone in management and the key to preventing it from hindering people's professional growth; and needless to say, it is extremely important to handle it efficiently to ensure the success of a global project and the well-being of the globally distributed team members.

The rest of this paper is organized as follows: Section 2 presents the related work on global software engineering. Section 3 briefly describes what global software engineering is and the state-of-the-art methodologies and collaboration tools. Section 4 mentions the major challenges of global software engineering followed by the author's empirical knowledge (Section 5) and the challenges faced in the industrial and academic realms. The conclusion and further research is presented in Section 6.

2. RELATED WORK

Researching about software engineering and software development in a global context is not new, but it has gained importance over the last decade due to the capitalization of the talent pool and resource usage wherever needed [8], the advantages of the different agile development methodologies that have emerged [9-10] and many companies opting to adopt such agile frameworks [11-12] and relaying in a geographically dispersed workforce to quickly kickoff projects and save costs by outsourcing or offshoring [13, 16]. Numerous articles and books have been written based on empirical investigation to demonstrate the advantages and disadvantages of global software development and the challenges that globally distributed teams may face [14-17]. In many occasions, researches have explored the particular challenges associated with global software engineering bringing into practice the agile methodologies to the industry [17-18] as well as to the academic realm [19-20].

In his book, Carmel [14] provides original "prescriptions" for dealing with cultural differences and collaborative teams. Another approach that works for projects of any size is presented by Ebert and it is based on his first-hand experience and expertise [16]. The book offers a proven and balanced framework for planning global development and provides best practices for managing a variety of software projects, coordinating the activities of several locations across the globe while accounting for cultural differences.

Holmstrom et al. [18] identified through their empirical investigation the particular challenges of global software development at three different USA based companies. They presented the problems associated to geographical and socio-cultural distance and their proposals to overcome or at least reduce distance problems. They discovered that companies work hard to stimulate knowledge sharing between teams' members, but in the end, it comes down to the capacity and interest of one understanding each other. In a more recent research [11] the authors presented how companies have been evolving from traditional waterfall development to agile development, and from there making a transition towards continuous integration and continuous deployment in what they call the "stairway to heaven". Based on interviews at different companies with globally distributed development teams, they presented the challenges in the transition of development methodologies as well as the actions needed to overcome them.

Oktaba et al. [21] investigated how Latin American software companies have tried to improve their software processes' capability as a fundamental step towards increasing the quality of their products. They found out that many of these companies implement good practices, models and standards proposed by the Software Engineering Institute (SEI), the Capability Maturity Model Institute (CMMI) or the International Organization for Standardization (ISO). Nevertheless, these companies have faced troublesome situations when trying to collaborate with geographically dispersed teams. It is mentioned in their research that cultural differences play an important role in the success of software processes improvement, and that an organization would reject a process if it does not match its culture, in a similar way as the human body would reject a mismatched transplanted organ.

In the academic realm, since Fall 2004 the Fulda University of Applied Sciences (HSF) and the San Francisco State University (SFSU) have been teaching a distributed and collaborative software engineering class [19-20] for the students to get acquainted with local and global software engineering. After five years of analyzing the outcomes of the classes, it was found out that both, local and global teams, were able to produce similar high-quality work. However, the biggest challenges that both teams faced were cultural differences and lack of development experience, and additionally time zone difference and physical distance for the global teams.

3. GLOBAL SOFTWARE ENGINEERING

Companies involved in large-scale software development tend to engineer and release systems used by a high number of users with different needs. The role of product management is to document these needs and to combine, merge and prioritize them to be able to prepare a roadmap with a set of requirements for the next release of the system. Since the adoption of agile methodologies, each release of the system is usually done within two to four weeks; and when more requirements arise or new projects are being negotiated, it may become hectic and hard to keep the agile development pace. Finding qualified workforce locally, in the same country or within the same time zone, it is indeed a difficult task and –in most of the cases– expensive. As an example, a software development company located in the USA would pay annually about $94,000 USD to a middle level software developer located also in the USA, whereas hiring a remote software developer in India would cost about $14,000 USD annually [22]. Besides salaries, companies also save costs by outsourcing the resources as they do not have to provide neither medical insurance nor an ergonomic work space to remote employees (also known as vendors or contractors).

Global software development is [18] when the distribution of the members of a distributed software development team exceeds the frontiers of a country. It is characterized by having distributed teams involved in IT projects and consisting of stakeholders from different cultures, nationalities, mother languages, geographic locations and probably different time zones too. Companies leverage skilled workforces in lower-cost economies thanks to high-speed Internet-based communication. India and China are well known [23] for offering workforces at greatly reduced cost compared with employment markets in the USA and western Europe. Other countries that have become skilled in software development in the last decade [24] and have come to compete in the market are Brazil, Ukraine, Belarus, Russia, Pakistan, Malaysia and Vietnam.

In the last decade, companies have opted to migrate from plan-driven software development approaches to agile methods because they are more flexible [25] when it comes to taking into consideration changes in the requirements in all phases of software development. Besides the methodologies, collaboration tools have been improved significantly and they help global teams in a daily basis. Version control, instant messaging and project management are some of the most common collaboration tools of global software development.

3.1. Common Methodologies

The waterfall methodology [1, 26] was the first established modern approach for building a system. It was originally defined by Winston W. Royce in 1970 and it quickly gained support from project managers because everything flows logically from the beginning through the end of a project. Since then, engineers, developers and researches improved and redefined the methodology to define the different plan-driven or traditional software engineering methods –which follow a linear framework type– and have been widely used in both large and small software engineering projects [26-27]. Notwithstanding the success of the waterfall methods with large and complex systems, it has disadvantages such as inflexibility in the face of changing requirements, punctilious processes irrespectively of the nature and size of the project, and the creation of a big amount of documentation [27] which could take more time than coding the system itself.

The waterfall approach provides a framework with five distinct stages [26]: requirements analysis and definition, system and software design, implementation and unit testing, integration and system testing, operation and maintenance. As it is a linear framework, a stage shall not start until the previous one has finished and the results have been validated and approved. Agile methodologies [10, 25] were developed to address the drawbacks presented in the traditional and linear frameworks. Agile methodologies deal with unstable and volatile requirements by using simple planning, short iterations, earlier releases and frequent customer feedback. These characteristics enable agile methods to deliver product releases in a shorter period of time compared to the waterfall approach. In contrast to traditional and linear frameworks, agile methodologies provide frameworks to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system (retrospective). Figure 1 presents a comparison of the stages of waterfall and agile life cycles.

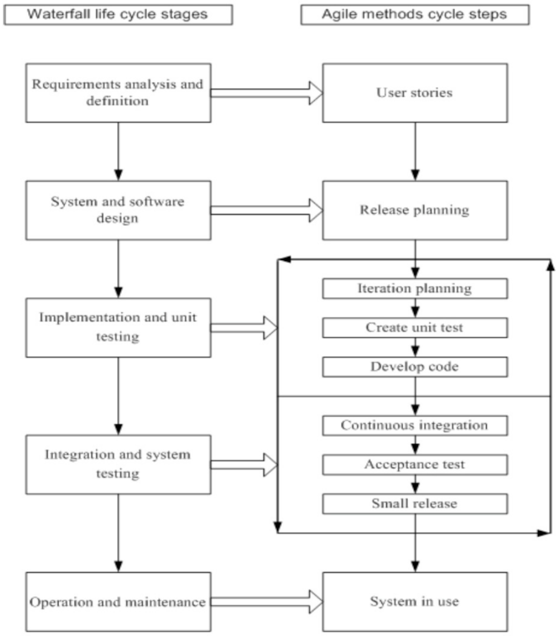
Waterfall life cycle stages
Agile methods cycle steps
Requirements analysis and definition
User stories
System and software design
Release planning
Implementation and unit testing
Iteration planning
Create unit test
Develop code
Integration and system testing
Continuous integration
Acceptance test
Small release
Operation and maintenance
System in use

Figure 1. Waterfall methodology vs Agile methodologies [25]

The most common agile methodologies are Scrum and Kanban [10]. They are widely used for well-known international companies all over the world, being the USA the country with the highest number of projects using Scrum [28]. Nevertheless, in the past five years Kanban has become more popular for the extra benefits and flexibility offered over Scrum [29-30].

Another framework that has become popular and quickly spread all the way through the technical community is DevOps, this because it understands the value of collaboration between development and operations staff through all stages of the software life cycle when creating and operating a service. One definition of DevOps is [31] the practice of development (Dev) and operations (Ops) engineers participating together in the entire service life cycle, from design through the development process to production support. DevOps has strong affinities with agile methodologies. The old view of operations tended towards the Dev-side being the “makers” and the Ops-side being the “people that deal with the creation after its birth” [31]. In this way, DevOps can be interpreted as an outgrowth of agile software development, which prescribes close collaboration of customers, product management and developers to rapidly iterate towards a better product. Furthermore, DevOps states that service delivery and how the application and systems interact are a fundamental part of the value proposition to the client too, therefore, the product team needs to include those concerns as a top-level item. From this perspective, DevOps [31] extends agile principles beyond the boundaries of "the code" to the entire delivered service.

3.2. Common Collaboration Tools

In order to succeed in a globally distributed software development project, teams need to have effective collaboration, which is one of the most common challenges of global software engineering and it is described in the next section. To achieve this collaboration, companies need to rely with full trust on different tools, and needless to say, on a high-speed internet connection. These collaboration tools can be grouped in six categories mentioned below, as well as the major service providers of each one [5, 31-33]. It is out of the scope of this research to mention the technical characteristics and benefits of each tool; therefore, each one has been just discretionary hyperlinked.

- Communication. Screen-sharing and instant messaging applications (chats), private social networks and wikis (internal to the companies), telephone and video conferencing, email applications and virtual boards are the tools that comprise this category, and the most common are Yammer, Rocket.Chat, IBM Sametime, Slack, Skype, Google Hangouts, GoToMeeting, WebEx, Zoom, IBM Verse, Outlook.

- Design. When team members need to prototype, brainstorm, create mockups and use virtual boards with sticky notes, the tools that can help on these tasks are Invision, Mural, Zeplin and MindMeister.

- Documentation. Tools designed to allow teams editing files at the same time and save their changes automatically. In this group Quip, Google Docs, Office Online and Zoho Projects are well known.

- File Sharing. When it comes to safe storing and sharing any type of files, Google Drive, Dropbox, Box, and OneDrive are the leading tools.

- Project Management. One of the most important tools while it comes to collaboration, and they can be used for all the members of a team. These tools have the ability to assign and prioritize tasks and help to identify what is critical versus what can wait. These tools keep teams up-to-date by informing the group of milestones and due dates for individual and team goals. Besides that, good management tools have the ability to provide quick feedback and score or measure performance. The leading tools are Asana, Jira Software, Trello, Basecamp, Nimble and Zoho Projects.

- Version Control. A source code repository is where developers can host their projects (either publicly or privately) to collaborate with other developers. At the time of this paper, GitHub is the most popular software repository on the web [34] followed by Bitbucket, SourceForge and GitLab.

Cloud-based applications have become popular and widely adopted because they excel in real-time team collaboration. However, when it comes to privacy and data security, a company may think twice before adopting a new technology or opt to implement it within its own infrastructure. In such cases, some service providers offer an enterprise version of their software for the companies to deploy and manage it in their own secure environment in order to keep their data private. Some of these providers are GitHub Enterprise, Trello Enterprise, Atlassian Enterprise, Slack Enterprise Grid and Box Enterprise. Often companies make public their paths on adopting a technology, e.g. IBM adopting Slack [35] even though the big company has its own cloud-based communication tool (IBM Same time). Sharing the tech stack that a company uses is beneficial to stay up-to-date with the top technologies in the industry and to discover new tools and services. With this information, developers can get to know which skills companies are looking for when searching for a job offer.

4. MAJOR CHALLENGES OF GLOBAL SOFTWARE ENGINEERING

Real life global software engineering projects have been widely analyzed and it has been pointed out the main problems that affect such environments, especially related to communication. Global collaboration creates significant coordination costs, e.g. incompatible schedules can lead to project delays, intercultural or linguistic misunderstandings can create non-billable rework, and technology failures can cause missed deadlines [14]. Working at a distance is complicated because it affects both the way one feels and thinks. It can lead to group the work colleagues into categories rather than see them as individuals. It also has bad consequences because one tends to view those with the same or similar culture/nation more positively, and those from a different culture/nation more negatively. Having an English-based video conference with a colleague in a different country, who speaks with an accent and exhibits different culture, makes that person seem a world apart [3]. If that was not bad enough, working at a distance also limits the amount of information we acquire about our colleagues and this can have negative effects for an effective collaboration. Besides this, other major challenges are overcoming the cultural differences, project management and the process of negotiation and decision-making.

4.1. Cultural Difference and Effective Collaboration

Cultural attributes can always be noticed in groups of people that share the same learned values, habits and behaviors; and these play a vital role in how one person performs different tasks at work based on individual patterns of thinking, feeling and acting [38]. Attributes that can be easily noticed by the naked eye are clothing, religious rituals, traditions, architecture or sports; while invisible attributes comprise of orientations to environment, time, communication, personal space, power, individualism, competitiveness, structure and mental frame. Cultural dimensions [38] are identified factors of great influence on the expected success of IT teams. These aspects must be contemplated by companies outsourcing or offshoring projects, as well as for local projects developed by teams with people coming from different countries, social backgrounds and perspectives.

When working in the same office with the colleagues it is possible to notice, interpret, incorporate and leverage an important amount of information as trying to make sense of one's daily experiences. Information that might not seem relevant (e.g. personal lives, moods or even the weather) also plays an important role. Having a bad mood due to weather conditions or being sleep-deprived, does affect the way one faces the daily activities. Unfortunately, barriers in the form of distance, time, culture, language and technology, all stand in the way of communicating such information, creating what Cramton [36] defined as The Mutual Knowledge Problem. A large percentage of the information is unknown when interacting with distant colleagues, and the less information is known about them, the more they are seen as "them". These outcomes are the exact opposite of a positive teamwork, but they happen so naturally that one may not usually identify such problems at the moment, especially when the colleagues are focused exclusively on results without regard for interpersonal relations. The problem with "us and them" thinking is focusing on the differences over the similarities among the colleagues, but one way to overcome this is to reverse the focus. Highlighting the things one has in common with the distant colleagues is the best way to reduce the problem [3].

Effective global collaborations have processes, procedures and good practices designed in a way that enforces information sharing across sites. Some of these are [37] scheduling regular meetings to share task-related information, taking time to share personal updates, equipping the team with the right resources and giving and taking a virtual tour to provide context. The more distant teammates know about each other's environment, the better to be able to understand one another's behavior.

4.2. Project Management

Functional, cross-functional and self-managing teams are the three different types of teams that can be found within a company [39], each one has its own specific goals and objectives, and they are created for both long term and short term interaction. This makes project management a more challenging discipline when it comes to initiate, plan, control and close the work of a globally distributed team. Project managers make project goals their own and use their skills and expertise to inspire a sense of shared purpose within the project team.

The Project Management Body of Knowledge [40] describes the core areas involved in the right management of a project and clearly identifies communication, human resources and stakeholders' management as core knowledge areas, but the dimension of intercultural teams plays an important role on the potential failure or success of project management for global teams. Cultural difference in project management is a stiff challenge also for local intercultural teams. These teams often face the problem of coping with local habits, cultural dimension conflicts, cross-cultural communications challenges, cultural shock and human resources issues. An intercultural project manager who works abroad, with people located in other countries and with local colleagues coming from different countries, has to take these cultural differences in religion, language, traditions, ethics, background, etc., into consideration and avoid adopting an ethnocentric perspective in order to succeed on the role as leader of the team.

4.3. Negotiation and Decision-Making

Since negotiation is strategic in nature, it necessarily represents a set of overlapping goals. For professional business negotiators [41], the job is to secure a settlement within their bargaining range and to trade interests wisely. Breakdowns in negotiations, when parties are from different cultures, many times are attributed to cultural differences. Furthermore, if culture has an effect on negotiation, the mental models of negotiators from one culture may not map on to the mental models of negotiators from another culture [41], making the specification of a single mental model problematic.

Project managers of global engineering projects many times have to assume the role of negotiators when making business and/or establishing relationships with other project managers in different companies and countries. Nevertheless, just because project managers can be from different cultures, it does not necessarily mean that their negotiation strategies will clash and their agreements will be suboptimal. Distinct social groups may have similar cultural values, and members may find the intercultural negotiation process trouble-free. In addition, not all members of a cultural group with a different profile believe and act consistently and strictly within the cultural norms [41]. When parties are motivated to reach an agreement, much can go on during the course of a negotiation to overcome individual, contextual and cultural differences. Three key factors leading to successful agreements that are affected by culture are [41] value for information sharing, means of searching for information and motivation to search for information. For long term positive outcomes, the goal should be to encourage a win-win situation in order to bring a set of agreements that are clearly acceptable and beneficial to all concerned parties. It is worth mentioning, the process of negotiation is difficult per se when it takes place among people of similar socioeconomic and cultural backgrounds, and it becomes even more complex when it takes place across countries and different cultures. Before starting any negotiation or during the decision-making process, it is very important to have in mind the assumption that individuals on the other side of the road do not necessarily think, act and feel the same way one does. Project managers who have been successful in the past and have worked with people from different cultures and nations, have a much better sense of the often hidden significance of the agenda based on the cultural values of their counterparts.

5. A POSTERIORI KNOWLEDGE

It is discussed in this section what the author has seen and experienced about software engineering by working for international companies, as well as during his academic career, and being involved in global software engineering projects.

5.1. Software Engineering vs. Software Development

Although the terms might seem similar and interrelated, they are not quite the same thing. The IEEE defines[1] Software Engineering as the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing and documentation of software. Based on this complete definition, it is possible to infer that software engineers are experienced computer scientists involved in the whole software development life cycle. Nevertheless, programmers, software developers and testers are not necessarily engineers, rather experts whom possess a particular set of skills in programming languages, frameworks or technologies, and they are highly reliable during the implementation and testing phases of a software development project. In the other hand, software engineers and software architects have broader skills that allow them to orchestrate the whole software engineering process.

Nowadays it is quite common to see in the news how software companies generate millions of revenues, and in particular those companies developing mobile applications and games. For these and other reasons, many people have decided to pursue a career as software developers; and there are some interesting cases of switching career paths in arts and music or aeronautics to become developers. There are plenty of options for those who want to pursue a career as programmers or software developers. Many courses are offered online on in-site by recognized universities and educational institutions related to programming languages, databases, apps development et al. In Mexico and other countries in Latin America, it is offered a technological high school program in Informatics with the focus to highly prepare the schoolers as programmers being able to work as soon as they graduate, but the salaries for such technicians in informatics are quite low due to the fact that companies value more experience and educational level. However, not all people are fond of studying and, therefore, they choose the technical career path knowing that their skills as well as salaries may improve over the years. In the other hand, to become a software engineer, the first step is to study engineering in a bachelor's or master's level. During the studies, one must necessarily be involved in software development projects; for that reason, in many university programs it is mandatory to make a professional residence or internship in a company, and it is precisely while being hands-on in a real project when students get the necessary practical knowledge and skills for what they are studying for.

5.2. Experience in the Industrial Realm

I have had the opportunity to work in several companies and being part of globally distributed and intercultural teams in Mexico. After graduating from B.Sc. in Computer Engineering, I got

[1] The Bureau of Labor Statistics - Systems and software engineering - Vocabulary, ISO/IEC/IEEE std 24765:2010(E), 2010.

the opportunity to start an intense training organized by two international companies: T-Systems and Volkswagen. The aim was to train 20 software engineers and take the certification exams from Solaris and Oracle. It was my first time in an international environment and I got to see how projects can be started in a short notice as well as getting cancelled. It was precisely what happened there, as we were notified that the project we had been trained and certified for was cancelled, but there was the opportunity to enroll to other training for a new project.

After that, I decided to check other options and I found opened positions with a Brazilian consultancy firm for a huge project within HSBC Retail Banking and Wealth Management. My professional experience was only the one I had from my internships and the training at T-Systems and Volkswagen. After personal, English language and technical interviews, this firm hired me and sold my services to the bank as a Senior specialist. There I got to meet many other people from other firms as it is a common practice for the bank to outsource the services at the beginning of a project. During this time, I faced the conflicts and challenges with cultural distance and effective collaboration as the teams were multicultural, locally and globally. I collaborated with people from China, Brazil, Argentina, Paraguay, Chile, USA, Canada, France, India and England. One big issue was time difference because often meetings were hold at 03:00 (Central Standard Time) at the offices of HSBC Mexico in order to coordinate with the teammates in China and India, and it was necessary to come back to a normal working day again at 09:00 in the morning.

A couple of years later, another consultancy came into the picture and offered a project within IBM in the USA. Time difference was one hour, so I decided to take the offer. I was the only one in Mexico joining that project whose managers and technical leaders were located in the USA, and it was allowed to all teammates to work from home in a regular basis. During this time, I got to collaborate in many projects as well as with many people located in China, India, Brazil, USA, Canada, Philippines, Denmark, Egypt, Slovakia, England and France. Something positive from IBM is that all the time new methodologies and technologies to improve the collaboration are implemented, and the career path is quite wide. As in my case, I worked in backend and frontend development before becoming the product owner and application architect of a project. Working by IBM helped me to get more familiar with intercultural teams. Once a month there was a team meeting where two team members randomly chosen would take five to ten minutes to share something about their countries and cultures. It was a very bold action from the managers as it helped the teammates to get to know each other better and reduce the cultural distance between them.

Both companies have well defined coding standards, good development practices and defined project management methodologies are followed. They are an example of the companies that have migrated from traditional to agile methodologies. Some of the tools used by HSBC [42] and IBM [43] for effective collaboration are precisely those mentioned in Section 3.2.

5.3. Experience in the Academic Realm

During my studies of Computer Engineering I had several courses of programming languages, databases, software development and software engineering. Most of the theory, methodologies and best practices were learnt there, but it was until the time working in the industry when everything was put fully into practice. Unfortunately, at the time of this paper, universities in Mexico do not offer collaborative courses with other universities in a bachelor’s level.

The first time I faced global collaboration in an academic field was while studying a M.Sc. in Global Software Development at the Fulda University of Applied Sciences in Germany. As briefly mentioned in Section 2, since Fall 2004 the HSF and the SFSU have been teaching a distributed and collaborative software engineering class for the students to get acquainted with local and global software development. The goal of the class is not only to polish the students’ hard skills, i.e. programming and architecting a software, but also to cover the soft skills such as leadership and effective teamwork which are key to succeed in software development projects.

At the beginning of the Winter semester, Professor Todtenhoefer (HSF) informed during the first class that everybody would work in teams for a Software Engineering project and collaborating remotely on the same project with other team from the SFSU, but only one group would work in a local project. The teams were made randomly by the professor based on the skills and expertise of each student. I was selected to work in a global project with other three students from Pakistan and they chose me as the team leader. Both teams, HSF and SFSU, had four members and each one with different development responsibilities, i.e. front-end or back-end, besides the roles of team lead and tech lead. The team from SFSU was more multicultural as the members were from Germany (team lead), Czech Republic (tech lead), China and the USA. I was very confident at the beginning and thought that we would not face or practically avoid all issues due to everybody in my team had more than three years of experience in the industry and also in global teams. We joined the SFSU team about two months after the project was started, this because in the USA the study programs have different schedules. We discovered that there was no organization whatsoever and the collaboration tools were not set up. As professionals, we asked to have a kick-off meeting but the SFSU team refused by saying that it was not necessary and email communication was enough. After two weeks, the goals and tasks for HSF team were not clear. Our professor played the role of Project Manager and Negotiator. We did bring all the details to him (escalating the issue), so he could talk with the other professor playing the role of Project Manager of SFSU team. It was until this point when SFSU team started to cooperate and we had the first video call and we communicated only with the two leads using Skype. We were very limited by the professors on the tools we could use, so everything was kept in minutes of the meetings using emails and all the tasks were specified on the minutes. A very rudimentary process, but we put a lot of effort for bringing to both teams harmony and organization. The teams had weekly meetings with their respective Project Managers to check progress and the planned tasks for next week, and also to talk about strategies and actions in order to improve and have a more harmonic, happy and effective collaboration. During the last month of developing a web application using software engineering principles and effective collaboration, we managed to create a peaceful and harmonic working environment, even though the project management techniques were rudimentary. Both professors graded very high our developed software and team collaboration, so the outcomes turned to be copacetic.

5.4. Conflicts and Challenges

Conflict is like water: too much causes damage to people and buildings; too little creates a dry, barren landscape denuded of life and color. It is a fact that one needs water to survive as well as an appropriate level of conflict to develop and grow one's character [4]. The way to manage natural resources of water through dams and reservoirs is key to achieve the balance necessary for life, and likewise with conflict management; a balance must be stablished between opposing forces and competing interests. In order to well manage conflict, members of globally distributed teams must develop and polish a broad set of soft skills, i.e. leadership, poise, tolerance, proactivity, intercultural knowledge, organization, customer relationship and management. In particular, project managers have to be able to use this set of skills to create an environment in which the project reaches its immediate goals and also functions successfully in terms of meeting the expectations, feelings and needs of its team members.

It is known that sleep deprivation affects people's mood, quality of life and level of happiness. Project managers know –or at least they should– that happy employees are more productive [7]. Base on these facts, companies and managers should be more flexible when working in distributed teams and implementing practices that would benefit everybody in the team, e.g. scheduling meeting in reasonable times, letting the employees work from home if they had to be working overnight and provide good equipment for remote work and state-of-the-art collaboration tools. Something extremely useful to overcome the challenges and cope with a multicultural team is a training in intercultural communication. Unfortunately, this is something that is not practiced in small software development companies and only some big international corporations have considered giving such trainings to their employees. Being the world so into globalization and software engineering projects being outsourced and offshored, it is of high importance for the companies to consider giving the employees interpersonal and intercultural trainings for everyone to be ready to face, understand and successfully collaborate with other people with different backgrounds and way of thinking.

5.5. The Dark Side of Global Software Engineering

Working remotely creates distance with other teammates that could probably never be reduced. Project managers may become more task-focused than people-focused and caring more about weekly or monthly results and performance rather than the well-being of the team. As already mentioned, happy employees are more productive, but this is something that global managers do not consider in many occasions and specially in organizations that are shaped like a pyramid.

Another negative side of global and intercultural collaboration is the constant stereotyping. A team member can be stereotyped or be less accepted based on the country he or she is coming from. This creates a distance even in local teams. Team members may search for those who share the same or a similar culture when needing help related to the project or just to have someone to talk to, but when this is not found, they may also become task-oriented employees avoiding talking with other team members and eventually having a not happy working environment.

And last but not least, the bad practices that consultancy firms still do while hiring with low salaries recently graduated engineers or developers with less than one year of experience, and selling their services in a Senior level to big corporations abroad; and as the projects are being outsourced, such corporations primarily care about results. When it comes to offshoring, also big corporations have been pointed out for their deliberately malicious business practices. Such is the case of IBM [44, 45] making millions of dollars by selling services to governments in different countries and delivering software that does not work because it was offshored in India with low skilled specialists [45].

6. CONCLUSION AND FUTURE WORK

Bruce Lee once said *"Empty your mind. Be formless, shapeless, like water. When you put water into a cup, it becomes the cup. You put it into a bottle, it becomes the bottle. You put it into a teapot, it becomes the teapot. Water can flow or it can crash. Be water, my friend"*. Running water is never steady, it continues flowing; when it reaches an obstacle, it continues pushing until it can flow again. In a similar way, one will always face conflicts and challenges in both personal and professional life, and it is important to face them and move on by being flexible and open-minded to be able to understand that every head is a different world and people will not always think in the same way one does. There are no easy answers or a standard and unique way on how a manager from Canada, Mexico, Brazil, Germany or Australia should manage conflicts that are inevitable while working in global teams or in negotiations across cultures. Understanding the cultural differences and gathering the maximum information about the culture of the countries, helps to understand what type of people are in the team, how to deal with them and what are their expected reactions or behaviors. Collaboration in globally distributed teams is and always will be a difficult thing to do, and it is important to be aware that issues and conflicts may arise. There are a wide range of useful skills for handling conflicts, and one very important is assertiveness. An individual needs to be able to express his/hers views clearly and firmly, but without aggression. It is important to emphasize that dealing with conflict in an early stage is generally easier because positions are not so entrenched and others are less likely to have started to take sides. The best way to address a conflict in its early stages is through negotiation between the participants.

Bachelors and Masters programs in universities may prepare good software engineers, but still it is until one gets involved in real life projects when those important soft skills are well developed. Learning about Conflict Resolution and Emotional Intelligence helps to deal in a more professional way with the multiple issues that one may face while working in an intercultural and/or globally distributed team.

The information presented in this paper is based mainly on experience and informal interviews carried out with former colleagues as well as students and professors of software engineering classes at universities in Mexico, Germany and the USA. It was observed that students as well as professionals have faced similar conflicts when collaborating with distributed teams and they did not have a proper preparation in social and cultural aspects that could have been useful to cope with those kind of conflicts. It is therefore, as plan of the future work, to conduct a case study adopting a qualitative research approach to analyze how software engineers deal with such conflicts when collaborating with global teams and which techniques or soft skills would be useful in their daily work. Furthermore, it is the plan to invite more universities offering software engineering programs to organize and collaborate with other universities, and especially those located in non-English speaking countries and not highly developed like Germany or the USA.

REFERENCES

[1] Paul Fisher, James McDaniel and Peter Hughes. 2008. System Development Life Cycle Models and Methodologies. Canadian Society for International Health Certificate Course in Health Information Systems, Module 3: System Analysis & Database Development, Part 3: Life Cycle Models and Methodologies.

[2] Sara Kiesler and Jonathon N. Cummings. 2002. What do we know about proximity and distance in work groups? In P. J. Hinds & S. Kiesler (Eds.). Distributed work. Cambridge. MA: MIT Press. 77-81.

[3] Mark Mortensen and Pamela J. Hinds, (2001). Conflict And Shared Identity In Geographically Distributed Teams. International Journal of Conflict Management. Vol. 12, Issue 3 (2001), 212-238. DOI: https://doi.org/10.1108/eb022856

[4] Craig E. Runde and Tim A. Flanagan. 2008. Building Conflict Competent Teams (1st. ed.). John Wiley & Sons, San Francisco, CA, USA.

[5] Rob Marvin. 2015. 5 Collaboration Trends to Expect in 2016. PC Mag Middle East. (October 2015). Retrieved March 7, 2018 from http://me.pcmag.com/your-business/4249/feature/5-collaboration-trends-to-expect-in-2016

[6] Alan Firth. 1995. The Discourse of Negotiation: Studies of Language in the Workplace (1st. ed.). Pergamon, UK.

[7] Cathy A. Costantino and Christina S. Merchant. 1995. Designing Conflict Management Systems: A Guide to Creating Productive Healthy Organizations (1st. ed.). Jossey-Bass, San Francisco, CA, USA.

[8] Andrew Begel and Nachiappan Nagappan. 2008. Global Software Development: Who Does It? In Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on Global Software Engineering. DOI: https://doi.org/10.1109/ICGSE.2008.17

[9] Samireh Jalali and Claes Wohlin. 2011. Global software engineering and agile practices: a systematic review. Journal of Software: Evolution and Process. Vol. 24, Issue 6 (August 2011), 643-659. DOI: https://doi.org/10.1002/smr.561

[10] Smartsheet. 2018. What's the Difference? Agile vs Scrum vs Waterfall vs Kanban. (February 2018). Retrieved March 8, 2018 from https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban

[11] Helena Holmstrom, Hiva Alahyari and Jan Bosch. 2012. Climbing the "Stairway to Heaven"—a Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. Proc. 38th EUROMICRO Conf. Software Eng. and Advanced Applications (SEAA 12). 2012, 392–399. DOI: https://doi.org/10.1109/SEAA.2012.54

[12] Jan Bosch. 2016. Speed, Data and Ecosystems: The Future of Software Engineering. IEEE Software, Vol. 33, Issue 1 (2016), 82–88.

[13] Hannah Whittenly. 2016. Ways Your Business Can Save Money by Outsourcing. (December 2016). Retrieved March 10, 2018 from http://www.leadershipgirl.com/5-ways-business-save-money-outsourcing

[14] Erran Carmel. 2011. Global Software Teams: Colloborating Across Borders and Time Zones (1st. ed.). Prentice Hall.

[15] Jutta Eckstein. 2010. Agile Software Development with Distributed Teams: Staying Agile in a Global World (1st. ed.). Dorset House Publishing.

[16] Christof Ebert. 2011. Global Software and IT: A Guide to Distributed Development, Projects, and Outsourcing (1st. ed.). Wiley-IEEE Computer Society Press, Hoboken, New Jersey, USA.

[17] Helena Holmstrom, Anna Sandberg, Jan Bosch and Hiva Alahyari. 2014. Scale and Responsiveness in Large-Scale Software Development. IEEE Software. Vol. 31, Issue 5 (October 2014), 87-93. DOI: https://doi.org/10.1109/MS.2013.139

[18] Helena Holmstrom, Eoin O. Conchuir, Par J. Agerfalk and Brian Fitzgerald. 2006. Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance. International Conference on Global Software Engineering (ICGSE2006), Costão do Santinho, Florianópolis, Brazil, 2006. DOI: https://doi.org/10.1109/ICGSE.2006.261210

[19] Shihong Huang, Dragutin Petkovic, Gary D. Thompson and Rainer Todtenhoefer. 2010. Teaching and Assessment for Global and Collaborative Software Engineering Course.

[20] Gary D. Thompson, Dragutin Petkovic and Shihong Huang. 2009. Teaching Distributed Collaborative Development Techniques in a Software Engineering Class Setting.

[21] Hanna Oktaba, Félix García, Mario Piattini, Francisco Ruiz, Francisco J. Pino and Claudia Alquicira. 2007. Software Process Improvement: The Competisoft Project. Computer, IEEE Computer Society. Vol. 40, Issue 10 (October 2007), 21-28. DOI: https://doi.org/10.1109/MC.2007.361

[22] Visually, Inc. 2012. Salaries of web developers in India, the Philippines, USA and around the world. Retrieved March 10, 2018 from https://visual.ly/community/infographic/business/salaries-web-developers-india-philippines-usa-and-around-world

[23] Human Engineers. 2015. China and India: Comparative HR Advantages. Retrieved March 10, 2018 from http://www.humanengineers.com/hr_library/general/china-and-india-comparative-hr-advantages

[24] Pär J. Ågerfalk, Brian Fitzgerald, Helena Holmström and Eoin Ó Conchúir. 2008. Benefits of Global Software Development: The Known and Unknown. In Q. Wang, D. Pfahl, and D.M. Raffo (Eds.): Making Globally Distributed Software a Success Story, ICSP 2008, LNCS 5007, 1-9. Springer-Verlag Berlin Heidelberg.

[25] Ming Huo, June Verner, Liming Zhu, Muhammad Ali Babar. 2004. Software Quality and Agile Methods. Proceedings of the 28th Annual International Computer Software and Applications Conference. IEEE Xplore. (October 2004). DOI: https://doi.org/10.1109/CMPSAC.2004.1342889

[26] Ian Sommerville. 2015. Software Engineering (10th. ed.). Pearson, Harlow, England.

[27] Mukesh Jain. 2008. Delivering Successful Projects With TSP and Six Sigma: A Practical Guide to Implementing Team Software Process (1st. ed.). Auerbach Publications, Boca Raton, FL, USA.

[28] Scrum Alliance. 2013. The State of Scrum: Benchmarks and Guidelines. Retrieved March 17, 2018 from https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2013-State-of-Scrum-Report_062713_final.pdf

[29] Kanbanize. 2016. Kanban VS Scrum. Retrieved March 17, 2018 from https://kanbanize.com/blog/kanban-vs-scrum-infographic

[30] Kanbanize. 2018. Top Reasons Why Companies Consider Using Kanban. Retrieved March 17, 2018 from https://kanbanize.com/blog/why-use-kanban-infographic

[31] Jennifer Davis and Katherine Daniels. 2016. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale (1st. ed.). O'Reilly, UK.

[32] Nikoletta Bika. 2017. 14 collaboration tools for productive teams. Retrieved March 17, 2018 from https://resources.workable.com/tutorial/collaboration-tools

[33] Jagina McIntyre. 2017. Agile Collaboration: Tools, Techniques & Games. Study.com. Retrieved March 17, 2018 from https://study.com/academy/lesson/agile-collaboration-tools-techniques-games.html

[34] Wikipedia. 2018. Comparison of source code hosting facilities. Retrieved March 17, 2018 from https://en.wikipedia.org/wiki/Comparison_of_source_code_hosting_facilities

[35] Bill Higgins. 2017. Listen to the wild ducks: How IBM adopted Slack. Retrieved March 18, 2018 from https://medium.com/design-ibm/listen-to-the-wild-ducks-how-ibm-adopted-slack-2bcfd3732680

[36] Catherine D. Cramton. 2001. The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration. Organization Science. Vol. 12, Issue 3 (June 2001), 346 - 371. DOI: https://doi.org/10.1287/orsc.12.3.346.10098

[37] Heidi K. Gardner. 2017. Smart Collaboration: How Professionals and Their Firms Succeed by Breaking Down Silos (1st. ed.). Harvard Business Review Press, USA.

[38] Germinal Isern. 2014. Intercultural Communication and Management Factors and Their Impact to the Process of Global Software Development for Virtual and Non-Virtual Teams. Journal of Intercultural Management. Vol. 6, Issue 1 (January 2014), 5–16. DOI: https://doi.org/10.2478/joim-2014-0001.

[39] Susan M. Heathfield. 2016. How to Build Powerfully Successful Work Teams. Retrieved March 18, 2018 from https://www.thebalance.com/how-to-build-powerfully-successful-work-teams-1918510

[40] Project Management Institute. 2013. A guide to the Project Management Body of Knowledge (PMBOK guide) (5th. ed.). Project Management Institute.

[41] Michele J. Gelfand and Jeanne M. Brett. 2004. The Handbook of Negotiation and Culture (1st. ed.). Stanford Business Books, Stanford, USA.

[42] FeaturedCustomers. Business Software used by HSBC Bank. Retrieved March 19, 2018 from https://www.featuredcustomers.com/customer/hsbc-bank/reviews

[43] IBM. 2018. IBM Whitewater tools and services. Retrieved March 19, 2018 from https://status.whitewater.ibm.com

[44] Tristan Yates. 2005. How IBM Conned My Execs Out Of Millions. Retrieved August 04, 2018 from http://atdt.freeshell.org/k5/story_2005_9_27_95759_4240.html

[45] Alec Kinnear. 2018. A billion reasons never to buy IBM services. Retrieved March 22, 2018 from https://foliovision.com/2018/03/why-not-buy-ibm

AUTHOR

Manasés Jesús Galindo Bello completed a M.Sc. in Global Software Development (Germany) and a Master in Mobile Business (Spain) in addition to a B.Sc. in Computer Engineering (Mexico). He has broad experience in software engineering, has led IT projects at HSBC, HP and IBM and trained software engineers in different areas and companies.