Towards Distributed Development: MORPHOS Process

Robson Ytallo Silva de Oliveira[1], Luciana de Queiroz Leal[1], Ryan Ribeiro de Azevedo[1], Jalmaratan Luís de Melo Macêdo[2], Silvio Lemos Meira[1,3]

[1]CIn – Informatics Center, UFPE - Federal University of Pernambuco, Recife - PE, Brazil

[2]M.P.U – Public Ministry of Government, Brazil

[3]C.E.S.A.R – Recife Center for Advanced Studies and Systems, Recife – PE, Brazil

{ryso, lql, rra2}@cin.ufpe.br, jalmaratan@gmail.com, silvio.meira@cesar.org.br

Abstract. *The distributed development has emerged as one of most effective strategy in software industry. It aims at providing software products with quality assurance, low cost and in a quick way. Nevertheless, many challenges threaten the good results, like distance and availability. It is necessary to have a well-defined process to support the dispersed development. This work describes a real experience: JOSE factory, and suggests improvements for its process: MORPHOS, in order to support other factories in collaboration.*

1. Introduction

Since 70's, supported by software engineering, the industry has been carving solutions in order to have a systematic and controlled development of systems. In the last years, the distributed development has been a trend of the software factories which are organizations to provide services of software with quality, low cost and speed [Herbsleb and Grinter 1999]. The distance is the main challenge of the distributed approach. On the other hand, it permits flexible schedules as well as counting on different experiences and knowledge. In addition, it is possible to be served by community with quickly, cheap technical growth.

According to [Wikipedia 2007], a software factory requires considerable organizational, process and project discipline. In distributed projects, a well-defined process is fundamental to coordinate the dispersed teams and their demands. A software process is a set of policies, technologies and assets to conceive, develop and maintain a software. The goal of this work is to expose a real experience of success with distributed development: JOSE Factory and the GVS Project. In an attempt to contribute with others distributed factories, this work aims to detailing the MORPHOS process, developed into JOSE Factory, to provide insights in proposal of a well-defined process.

Besides this introduction, the reminder of this paper is organized as follows. The Section 2 presents the base processes and methodologies for MORPHOS Process. The next section describes the factory and its project. The process details are presented in Section 4 and analyzed (problems and improvements) on practice by Section 5. Finally, the Section 6 presents the concluding remarks and directions for future work.

2. Distributed Process and Methodologies

The processes and methodologies of software development have been studying in order to identify their potential to create or tailor new processes according to the software

industry necessities. With different focus and levels of details, the software engineering community presents solutions for distributed development which has arisen as a market trend during last years. This section presents the most relevant proposals for this work.

2.1 CSCW

The CSCW term (Computer Supported Cooperative Work) was created in 80's by Paul Cashman and Irene Grief [Grudin 1994] and can be defined as the research field to study techniques and methodologies of work in group and the modes which the technological ways can support that work [Grudin 1994]. The CSCW has as focus the communication techniques used to give support to cooperation and coordination. The computational support engaged is denominated groupware, term created by Peter and Trudy Johnson-Lenz [Johnson-Lenz and Johnson-Lenz 1982], which assign the computational applications projected to give support to collaborative work, supplying an interface for shared environments and standing up for typical cooperative tasks, like: collective write, shared schedules, access to databases and electronic meetings.

Coordination, communication and cooperation are the main factors of an activity in group [Duitshof 1995]. For that reason, the CSCW aims at four collaboration aspects: communication, coordination, group memory and perception. The CSCW approaches collaborative process joining individuals in order to work in group to reach common goals, been physically or not at same place, doing with synchronous or asynchronous way. In systems which take advantages of CSCW techniques, a group of users is coordinated to achieve a task in an environment of collaborative work, needing to access shared data. According to [Xudong and Qiuqi 2004], the focused research in CSCW is based in the group of users, in the appearing of certain roles whose necessity of cooperation makes to emerge, otherwise will result in difficulties, inefficiency and eventual solution failed. The coordination strategy is exactly those roles incorporation.

2.2 RUP

Rational Unified Process, RUP, is a proprietary process of software development created by IMB Rational Software Corporation. The RUP is a well-structured process to develop software with quality in quick and previsible way [Kruchten 2003] [Pimentel 2006]. The RUP is, declared, elaborated to be extended too, as RUP for J2EE and RUP for eXtreme Programming (XP) illustrate. It can also be extended in function of 3 dimensions as seen in [Pimentel 2006]: Collaboration = Communication + Coordination + Cooperation, entitled 3C of Collaboration Model [Ellis et al. 1991]. Collaboration, assigns the action to work in group, the execution of a common work by two or more people [Ferreira 1986]. The collaboration has been realizing from 3C Model which puts emphasis on that a group, to collaborate, needs to establish appropriate communication, coordination and cooperation.

In a group, there are people with different points of view which can generate the complement of individual perceptions [Fuks et al. 2002] [Pimentel 2006]. In such case, communication is the action to become common, to exchange messages objectifying the mutual comprehension, to chat, to dialog. In the collaboration, normally the members of group communicate themselves to the action: negotiate, take decisions and fix commitments [Winograd 1989] [Pimentel 2006].

Objectifying to avoid that communication and cooperation efforts are squandered [Raposo et al. 2004] [Pimentel 2006], the coordination is the action to

dispose according to some charge and method, to organize, to arrange. The coordination of a collaborative work organizes the group members in order to resultant commitments of the negotiations are performed in the foreseen charge and time accomplishing their goals and restrictions. The cooperation is the action to operate in conjunction - the group members actuate in range to achieve the defined and organized tasks during the coordination. When cooperate, the individuals need to communicate themselves to renegotiate and take decisions about unexpected situations.

2.3 DXP

The DXP (Distributed eXtreme Programming) was proposed by Michael Kircher [Kircher et al. 2001] and is an adaptation of XP (eXtreme Programming) for distributed software projects, open-source and large projects. DXP applies the XP principles in environments with teams of mobil and distributed development. The relevant principles and values from XP to DXP are planning game, pair programming, continuous integration and on-site customer. In accordance with [Kircher et al. 2001], DXP offers several challenges like communication, coordination, infrastructure, availability and management. However, each one can be addressed and in most cases overcome.

3. JOSE, an Open Source Factory

Some academics, geographically remote, had made part of an open source software factory, by Master Degree Program at Federal University of Pernambuco. In order to take advantage of real client and scenarios, J.O.S.E (Joint Open Source Environment) Software Factory was created to develop an open source project called GVS (Grid Versioning Service). The Factory counted on 11 members and external collaboration during the project development. A software development process, named MORPHOS, was defined. It was based on interactive, agile and incremental approach directed to distributed development environments.

3.1 GVS Project

The GVS project was idealized to be a versioning control with decentralized administration and functions respecting the p2p (peer to peer) concepts like replication and synchronization. The system is composed by two components: a service and a client. Some troubles like synchronism, authentication and domains problems are able to show up when several different repositories are used. Thus the main goal of GVS is to minimize those threats, through a distributed versioning service able to abstract the used version control system (CVS, SubVersion, ClearCase, and so on) and performing remote operations in transparent way for the developers. Although tools like IBM Rational ClearCase MultiSite are able to create and manage distributed versioning service, they have a high cost. On the other hand, the GVS Project is an open source project respecting the BSD (Berkeley Software Distribution) license.

4. MORPHOS Process

MORPHOS Process joins the dispersed development concepts of CSCW, DXP and RUP. The key point is the CSCW whose actions are adapted, once that the collaboration is for distributed software factories. Several processes and good practices of software engineering give subsidies to carve the MORPHOS. It has a well-defined control and coordination of the activities as well as perception of the distributed tasks and their

dependencies. In addition, it shares information among the members, involving collaboration and communication.

4.1 MORPHOS Coordination

To reach success with the proposed deliverables in time, the MORPHOS Process splits the factory members in management cells (MC) based on CSCW concepts. Although CSCW pre-defines some cells like requirements, implementation and tests groups, the MORPHOS permits to classify the cells according to demands. Through meetings, the members responsible for coordinating the process - coordination team (CT) - allocate participants into the cells and define their roles in order to carry out parallel tasks (See Figure 1). The CT is composed for all participants of the factory, according to their availability. The cell manager (CM) is the main role, responsible to support solving the cell demand and report the current activities and the task status.

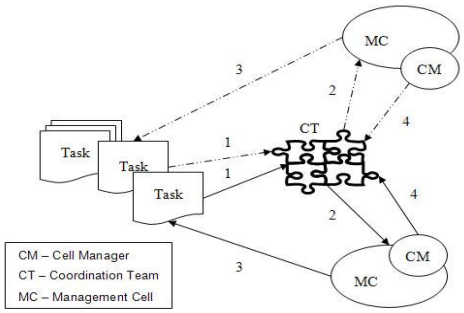

Figure 1 - MORPHOS Coordination

In general, according to Figure 1, the CT analyses the demands (1) and mount MC's (2). The MC's supported by their CM's try to solve the problems in time with high quality (3). The CM is responsible to follow all activities of members in the MC, in consequence, prepare a brief description to report to the factory (4). For that, it is a good practice of the members to report, daily, a summary of their activities and results.

4.2 MORPHOS Perception

In accordance with MORPHOS Coordination, the members of the Factory are divided in cells to perform tasks. Each cell member knows all current activities of the project developed and mainly their own ones. Nevertheless, distributed tasks generate dependencies and can provide retardation or idleness. Thus, the CM's also are responsible to be a contact point, having information to help the cells members. To know the whole project is an opportunity of having harmonica teams and good communication. The Cell Manager knows who is responsible to solve each problem or clear doubts up. To improve perception, the MORPHOS Process suggests using time trackers which are software able to track time spent on projects. The times are tracked individually according to activity performed.

4.3 MORPHOS Collaboration

Groupware tools are used to improving decision taking, generating and manipulating information. In dispersed development, the polemic decisions must be voted. For that, synchronous and asynchronous medias, depending on the exigency, can be used to execute elections and to discuss subjects among members, client and external collaborators. In addition, the MORPHOS Collaboration includes the bug tracker usage

to support the problems of development process and bugs found during the test steps. Bug Trackers are systems designed to track of reported bugs in a software development.

4.3.1 MORPHOS External Collaboration

In open source projects, the motivation to learn is a characteristic which stimulates volunteers' participation. The external collaborator is who participates in the project as member of open source community. External collaboration is a challenge for software factories. It is necessary to facilitate the code maintainability for the purpose of propitiating external collaborators integration. For that, MORPHOS Process stimulates to use a default template with important information to support to generate code.

4.4 MORPHOS Communication

In a distributed project, the most important factor of success is the communication. Each participant needs to know completely the factory, the process and the project. MORPHOS Communication suggests synchronous and non-synchronous ways to share information and maintain contact of members. The most productive remote meetings are done by video-conference. As those kinds of meetings are not accessible for all dispersed factories, another alternative is to use chats or sound conferences. Advices and discussions can be accomplished by e-mail. Inclusively, it is a good practice to exchange off-topics messages by list mail, the end groupware is not just technology, it is also social [Coleman 2007].

4.5 MORPHOS Phases

The MORPHOS Process uses the concepts of Management Cells which are responsible to perform the activities of iteration, from a project plan defined in high level. When the activities are finished by a cell, the new iteration is planned and executed immediately without waiting that the end of the last one happens in the others MC's. It is important that the independent work of cells does not obstruct the communication among their members; it must be constant during whole project. The MORPHOS Process is defined in three phases: Project Start, Iteration and Final Product, as is presented in the Figure 2.

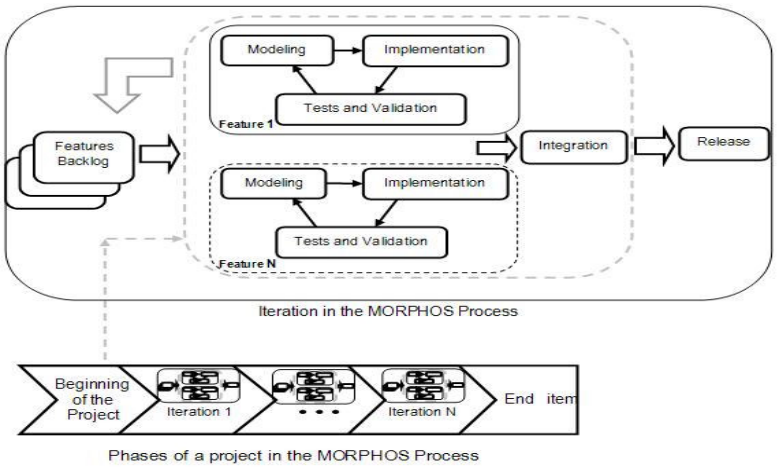

Figure 2 - MORPHOS Phases

Project Start - Elaboration part of a project plan in high level which specifies the expected functionalities for the project, with a bit of details. The team develops a draft of the system architecture.

Iterations - Characterized by short time of duration, they must be two or four weeks. For each iteration, depending on team or quantity of features to be implemented, work fronts are defined to develop functionalities in parallel. The functionalities are prioritized in a Feature Backlog. Each feature is extracted from Feature Backlog and circulates by the Modeling, Implementation and Tests and Validation steps.

- *Modeling* - Is the elaboration of the architecture for better comprehension of the functionalities to be implemented.
- *Implementation* - After having a done model, the feature is codified. One or more features can be implemented during same iteration. In the end of iteration, the features developed are integrated generating a release.
- *Tests and Validation* - After creating each class, unit tests are codified to ensure that the requirements are implemented correctly and do not have failed code. That kind of test is implemented and executed by developer of the class. When a release is generated, integration tests are performed to certify that no errors were introduced in the integration. After those tests, the client receives each release and verifies if is in accordance to the necessities and definitions. Thereon its feedback, other iteration starts. Some functionalities which were not implemented according to client whishes can return to the Feature Backlog and be implemented again in a future iteration.

Final Product - Thereafter the last iteration, the final release is delivered.

5. MORPHOS on Practice

In contrast to RUP recommendations, MORPHOS Process is not well-defined and documented to be used; there is only one document on the factory site (www.josefactory.org). The creation, implementation and first calibrations of MORPHOS were accomplished into JOSE environment according to its necessities. The maturity and capability of factory members should be decisive factors to real success.

5.1 Coordination

In JOSE scenario, the MC's were created without some important criteria, like availability, skills, capabilities and wishes. On practice, sometimes, the MC's were composed only for members whose times of dedication for factory were short, while other ones were formed for participants full time dedicated. Usually, all times were used to study for the tasks because the cells members had not skills and capabilities to solve the activities attributed to them. Moreover, other members did not wish to make part of their cells due to some person or activity. In the beginning of experiment, each member informed the individual commitments and the quantity of hours dedicated per week to the JOSE Factory. Those information must be considered before the CT splits the MC's.

5.2 Perception and Collaboration

Following the CSCW concepts adapted for MORPHOS, the JOSE Factory integrants adopted time tracker (Allnetic Working Time Tracking - www.allnetic.com) to report their activities. The reports are committed on the CVS. Internally, some members did not use the Time Tracker due to they were not familiar with or their Operational Systems did not support the free version of tool. JOSE Factory still counted on bug tracker (mantis - www.mantisbt.org). To be as transparent as possible, the client should

receive frequently status reports describing the development status of the GVS Project. Nevertheless, on practice, it was not a reality for JOSE Factory. The big problem in the MORPHOS Perception was the tasks dependencies. The only members who knew the activities of each one were the CM's. In addition, when a problem was solved, several members spent their times synchronizing the activities or just understanding the solution. Though remember that the late members were not on purpose.

5.2.1 External Collaboration

During GVS project, there was participation of external collaborators who were geographically distributed. They weren't effective members of the JOSE Factory. The external collaboration happened by four ways:

Viewer - followed the factory course through available resources, like list mail;

Collaborator of Ideas - commented and gave suggestions about project;

External Programmer - worked in the system codification;

External Tester - performed code inspections and tests, and reported bugs through Mantis (www.josefactory.org/mantis).

5.3 Communication

Apparently, MORPHOS Communication was well-developed. The list mail were used to vote decisions, share knowledge, discuss organizational subjects and interact with the client whose role was represented by a technical person – very important in challenge like GVS Project. The meetings were done by MSN Messenger, Google Talk or Skype using conferences (chat or audio) scheduled by GoogleCalendar (www.google.com/calendar). The Factory site counts on news and it maintains all member contacts. The members didn't have a synchronous communication, and they didn't look like to work in groups sometimes. As the communication process is completely linked with collaboration, and the first one was usually used in a passive way the both areas were compromised and sometimes was necessary to carry out personal meetings. Nevertheless, the factory had enough maturity to support a workshop virtually for members absent – including one travelling abroad - in the event.

6. Conclusions and Future Remarks

This work has presented an analyze of an academic project using real clients and scenarios in order to identify points to improve a software development process for open source factories using some approaches found in the literature and the practical experience through JOSE Factory which was a successfully project implementing the Grid Versioning Service.

MORPHOS process was not well-implemented by JOSE Factory once that almost every time dedicated to the factory was wasting in studies and sometimes was necessary heroic efforts, like in CMM-1, to obtain good results. To solve some problems, it will be necessary, mainly, to adjust MORPHOS coordination, giving criteria to split the Management Cells. Other improvement is to create a threat analyses phase to be embodied by MORPHOS once that the process does not predict risks.

Finishing, we have identified some problems like: knowledge problems and user stories necessity. One of our first activities was to create a flow gram of the process

(showed in Section 4.5). For future, there are many works to be evaluated to document the MORPHOS. One of the main motivations for us is due to the fact that JOSE Factory reached, successfully, its goal into an academic environment with real client.

References

Coleman, D. (2007) "Groupware - The Changing Environment", http://www.collaborate.com/publication/publications_resources_groupware_book_toc.htm, March.

Duitshof, M. (1995) "Workflow Automation in Three Administrative Organizations", Masther's Thesis. Departament of Computer Science – Section Information Systems - University of Twente - The Netherlands.

Ellis, C., Gibbs, S. and Rein, G. (1991) "Groupware - Some Issues and Experiences", In: Communications of the ACM, v. 34, n. 1, p. 38-58.

Ferreira, A. (1986) In: Novo Dicionário da Língua Portuguesa. 2. ed. Rio de Janeiro: Nova Fronteira.

Fuks, H., Gerosa, M. and Lucena, C. (2002) "The Development and Application of Distance Learning on the Internet", Open Learning - The Journal of Open and Distance Learning, v. 17, n. 1, p. 23-38.

Grudin, J. (1994) "Computer-Supported Cooperative Work: History and Focus", IEEE Computer, v. 27, n. 5, p. 19-26. http://www.ics.uci.edu/~grudin/Papers/IEEE94/IEEEComplastsub.html, June.

Herbsleb, J. D. and Grinter, R. E. (1999) "Splitting the Organization and Integrating the Code: Conway's Law Revisited", Proceedings of ICSE. US.

Johnson-Lenz, P. and Johnson-Lenz, T. J. (1982) "Groupware: The Process and Impacts of Design Choices", Communication Systems, Status and Evaluation. Academic Press.

Kircher. M., Jain, P., Corsaro. A. and Levine. D. (2001) "Distributed eXtreme Programming", XP2001, Italy, May 21-23.

Kruchten, P. (2003) "Introdução ao RUP – Rational Unified Process", Rio de Janeiro: Ciência Moderna.

Pimentel, M. (2006) "RUP-3C-Groupware: um processo de desenvolvimento de groupware baseado no Modelo 3C de Colaboração", Phd's Thesis**.** PUC-Rio. 162 pages.

Raposo, A., Pimentel, M., Gerosa, M., Fuks, H. and Lucena, C. (2004) "Prescribing e-Learning Activities Using Workflow Technologies", In: 1ST International Workshop on Computer Supported Activity Coordination – CSAC. p. 71-80, Porto, Portugal.

Wikipedia – The Free Encyclopedia (2007) "Software Factory", http://en.wikipedia.org/wiki/Software_factory. March.

Winograd, T. (1989) "Groupware: The next wave or another advertising slogan?" In: COMPCON 89, IEEE Computer Society Press, p. 198-200.

Xudong, L. and Qiuqi, R. (2004) "Research of Access Control and coordination strategy in CSCW". In: International Conference on Signal Processing, Beijing.