PUCRS

Faculdade de Informática PUCRS

Software Configuration Management over a Global Software Development Environment: Lessons Learned from a Case Study

Leonardo Pilatti
(School of Computer Science – PUCRS – Porto Alegre (RS) – Brazil)

Jorge Luis Nicolas Audy
(School of Computer Science – PUCRS – Porto Alegre (RS) – Brazil)

Rafael Prikladnicki
(School of Computer Science – PUCRS – Porto Alegre (RS) – Brazil)

CDPe Centro de Desenvolvimento e Pesquisa DELL/PUCRS

MUNDDOS

Agenda

- Context and Motivation
- Research Question
- Software Configuration Management
- Case Study and Results
- Final Considerations

Where are we?

Boa Vista
Roraima
Amapá
Macapá
Belém
São Luis
Maranhão
Fortaleza
Rio Grande do Norte
Natal
Ceará
Teresina
Paraíba
João Pessoa
Piauí
Pernambuco
Recife
Alagoas
Maceió
Sergipe
Aracaju
Bahia
Salvador
N
Pará
Manaus
Amazonas
Acre
Rio Branco
Porto Velho
Rondônia
Palmas
Tocantins
Mato Grosso
Cuiabá
MW
Brasília DF
Goiânia
Goiás
Minas Gerais
Belo Horizonte
Espírito Santo
Vitória
Mato Grosso do Sul
Campo Grande
São Paulo
SE
Rio de Janeiro
Paraná
Curitiba
Santa Catarina
Florianópolis
S
Rio Grande do Sul
Porto Alegre

PUCRS University

PUCRS University

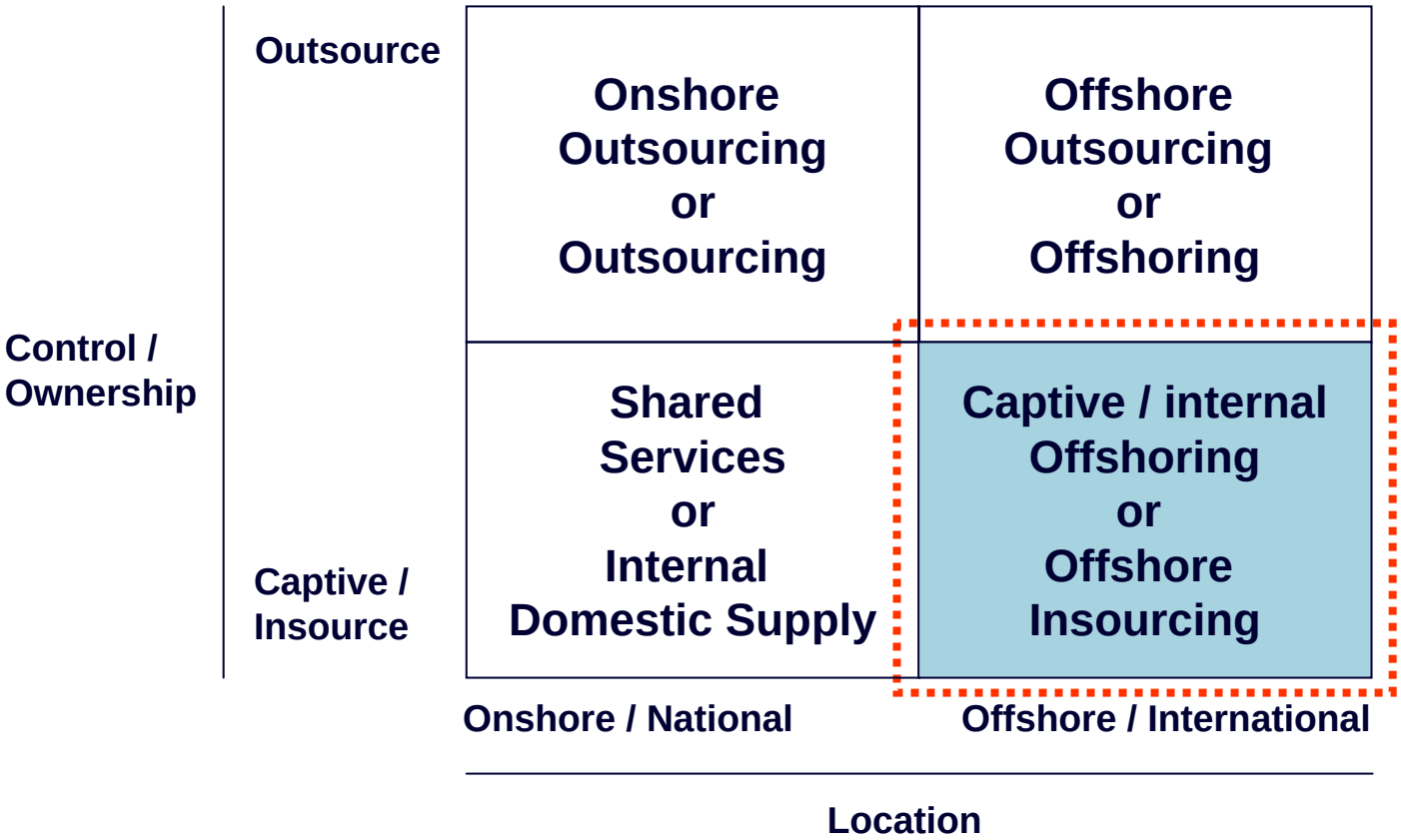
The Tecnopuc

The Tecnopuc

Motivation

- The crescent globalization in business environments has affected the software development market
- Several organizations decided to distribute their software development processes inside or outside their countries (onshore, nearshore, offshore)
- Software configuration management (SCM) is also influenced by the team distribution

Motivation



Source: Robinson & Kalakota (2004), and Carmel & Tija (2005)

Research Question

- What kind of problems project teams have faced when working with SCM process in a global software development (GSD) environment, and how these problems have been addressed?

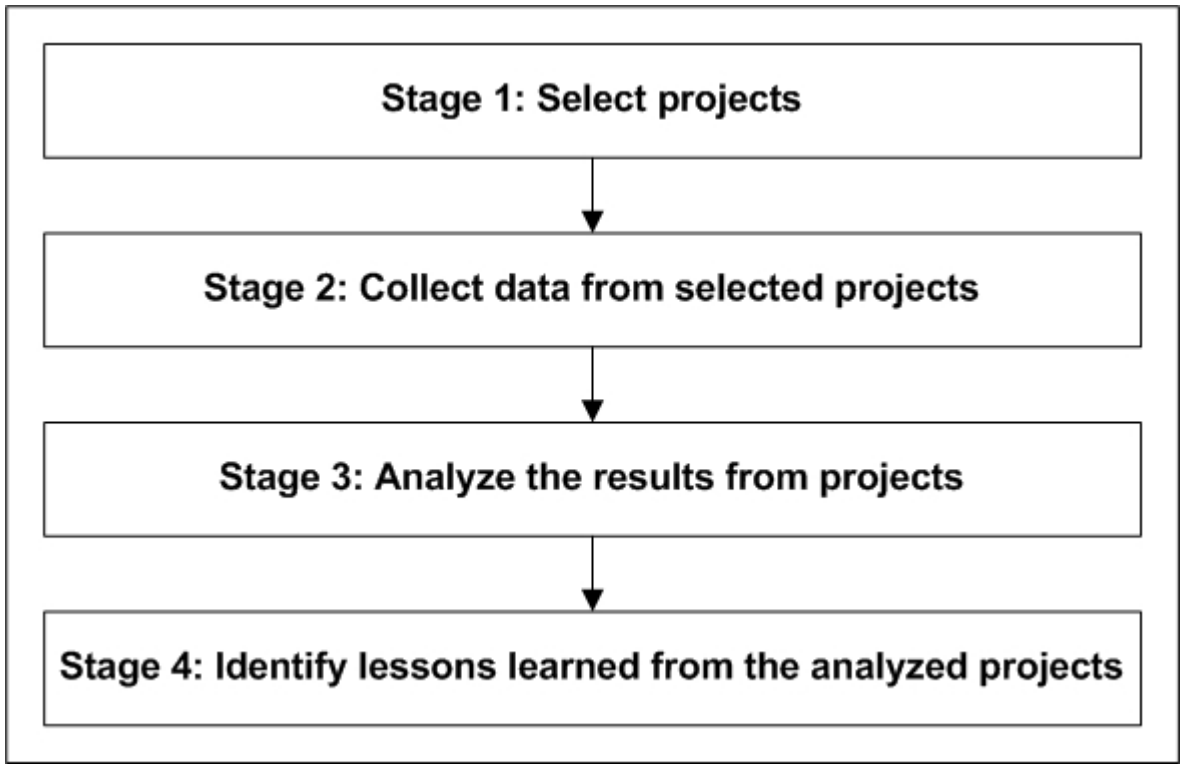
Software Configuration Management

- To establish and maintain the integrity of software products throughout the development life cycle (IEEE Standards, Berczuk et. al.)
- Involves identifying the configuration of the software (i.e., selected software work products and their descriptions) at given points in time
  - Systematically controlling the changes
  - Keeping the integrity and traceability of the configuration throughout the software development life cycle

Case Study

- Exploratory study
- The case study was conducted in a multinational organization that has global software development centers around the world
- The study was conducted from the point of view of the center located in Brazil
  - It is recognized as CMM level 2 since 2002
- Case study goal
  - To analyze 4 distributed projects and to identify problems with the SCM in a globally distributed context (more than 40 developers in 3 countries)
  - Lessons learned were collected at the end of each project

The Research Plan



Projects Selected

Characteristics of projects				
Project	Countries involved	# of SCM focal points	# of developers	Use of Distributed Environment
A	BR, US	1	12	NO
B	BR, IN, US	1	17	YES
C	BR, US	2	4	YES
D	BR, US	2	8	YES

Project A

- First to be developed simultaneously in different centers, one in the US, and the other in Brazil
  - Motivation: application size and the aggressive delivery date
  - 5 developers in Brazil, and 7 developers in the US
- Brazilian team
  - 70% of the use cases
- American team
  - All data access, implemented through stored procedures
  - Around 30% of the use cases
- Two SCM environments (CMM project in Brazil, but not in the US)
  - Simple version control in the US
  - More complex SCM process in Brazil
  - One person to synchronize both environments (SCM controller)
    - Informal communication
    - Unclear tasks, and deadlines

Project B

- Was developed simultaneously in 3 different centers (US, Brazil, and India)
  - Try to execute a follow-the-sun development
  - 12 developers in Brazil, 2 developers in the US and 3 in India
- Only one main repository and SCM environment
- Brazil still the only one with SCM process based on the CMM model
- Role of the SCM Coordinator
  - Plan the SCM activities, synchronize with SCM controllers (team member)
  - Code synchronization, and integration

Project C

- Maintenance project
- 2 developers in Brazil and 2 developers in the US
- Scope divided in modules and separated between the two sites
- The teams agreed to use the CMM level 2 processes
- Single SCM repository
- 2 experienced SCM Controllers (one at each site)
  - Validate the artifacts to be uploaded into the SCM tool

Project D

- 5 developers in Brazil, 3 developers in the US
- Scope divided in modules and separated between the two sites
- Similar strategy to the Project C
  - CMM level 2 processes
- 2 SCM coordinators, one at each site
- 2 SCM controllers
- Single SCM repository

Results

- Distinct SCM environments
  - The need for synchronization on the SCM environments
    - The SCM tools were different, and this had to be performed manually
    - Took from the configuration controller 3 to 4 hours each week.
    - Only one SCM coordinator in some projects
- Some configuration items were updated by the two teams
  - Manually file merges
- It was decided to perform two builds every day in one project – project size and complexity (Build coordinator)
- Lack of baseline planning
  - Delays in some planned deliveries

Results

- Baselines were not enough to rebuild an application environment
  - Other resources that were not under configuration management were used (database backups, for example)
- Weak engagement on the SCM processes
  - Misunderstandings and high number of non-compliances found (SQA)
- Dependency between the modules developed by the distributed teams
  - Modules dependency caused some rework and idle time for the developers
  - Scope changes
    - The importance of updating the plan due to scope changes

Lessons Learned

N.	Lesson Learned	Projects
#1	The work breakdown in distributed projects should minimize dependencies between geographically distributed teams.	A

- If possible, each team should work in their modules without any dependency. No matter how integrated the distributed teams are, communication will always be expensive.

Lessons Learned

N.	Lesson Learned	Projects
#2	Distributed development projects should work with only one instance of SCM environment.	A

- Both teams should agree in a common management process. Distinct SCM environments caused overhead on CM work and activities.

Lessons Learned

N.	Lesson Learned	Projects
#3	Put all configuration items required for a build under configuration management is a good approach.	B, C

- All files related to build (source codes and build files) should be stored in a global configuration environment. Even training and end-user documents can be stored in the same environment.

Lessons Learned

N.	Lesson Learned	Projects
#4	Distributed development projects with centralized SCM environments should define one build coordinator.	B, C

- The configuration manager should have great experience on the application and technology in place, so, the build will be more efficient. This was because the high number of required builds throughout the project development.

Lessons Learned

N.	Lesson Learned	Projects
#5	Establish and clarify all main concepts on SCM discipline, before actually starting the development, is a good approach.	B

- Some concepts are essential for the understanding of what needs to be under configuration management, and its contents. It was noted that the fundamental concepts in SCM weren't completely understood by project team members.

Lessons Learned

N.	Lesson Learned	Projects
#6	Even with experienced teams in distributed development, the SCM engagement in the beginning should be prioritized.	C

- Focus on the set of SCM processes, responsibilities of each team member and communication. This can avoid the weak engagement on SCM at the beginning of the project.

Lessons Learned

N.	Lesson Learned	Projects
#7	Always plan baselines and document them in the project's SCM plan, as soon as possible.	C, D

- Avoid requesting baselines on demand; otherwise the deliveries can be delayed. The lack of planned baselines is a problem root cause.

Lessons Learned

N.	Lesson Learned	Projects
#8	The re-planned activities due to scope floating across teams should take in place.	D

- The analysis should include dependency verification on the module being floated against the other modules. This can predict the problems with configuration management activities.

General Observations

- SCM in a GSD context can become an arduous task if the process is not well defined and if the teams are not prepared to perform SCM activities
- Many of the efforts spent in the CMM Model level 2 project contributed to minimize problems in terms of SCM process definition
- Some lessons identified in prior projects were applied in other projects (SCM environment , focal points)
- Some lessons were forgotten, such as the dependency on tasks executed by the American and Brazilian teams (Projects A and D)
- Tendency to relax on engagements, as times goes by

Final Considerations

- SCM has a critical role in software development process
  - Most of the difficulties are increased when software development teams are distributed
- Few studies about the impact on the SCM process and tasks
- This study enables a better understanding of the relationship between distributed project teams related to the SCM.
- Limitation: the small number of projects studied, in only one organization
  - The study was conducted in 18 months, and the projects were developed in sequence
- Next steps: to run this study again to collect more empirical data, bringing more accuracy to the results.
  - To explore other aspects of GSD such as culture, communication, coordination, trust and cooperation, related to the SCM process.

PUCRS

Faculdade de Informática PUCRS

Software Configuration Management over a Global Software Development Environment: Lessons Learned from a Case Study

Leonardo Pilatti
(School of Computer Science – PUCRS – Porto Alegre (RS) – Brazil)

Jorge Luis Nicolas Audy
(School of Computer Science – PUCRS – Porto Alegre (RS) – Brazil)

Rafael Prikladnicki
(School of Computer Science – PUCRS – Porto Alegre (RS) – Brazil)

CDPe Centro de Desenvolvimento e Pesquisa DELL/PUCRS

MUNDDOS