ESCUELA SUPERIOR DE INFORMÁTICA
UNIVERSIDAD DE CASTILLA-LA MANCHA

SOFTWARE ENGINEERING II

Configuration Management Plan
Curso 2015/16

AUTHORS:
Miriam Castillo Torroba
Jaime Cepeda Villamayor
Jose Antonio Martinez López

Contents

1 Introduction

In the next document the CM plan would be described following the IEEE 828:2012 to guarantee the whole project quality and also the final product obtained at the end of the development process. The project called Animalia will be carried out by JuManJi Software Solutions.

The Software processes evolve throughout their life cycle, when it progresses through the different parts of the development stages, you end milestones and you get artefacts. This artefacts shapes the structure and takes the appearance of what is conceived when the product was devised.

When the product is committed, new changes can be applied, new capabilities, changes in previous requirements or maybe bug fixes. Because of that, the software must be maintained till it is removed.

The CM is responsible for investigate what elements will be modified throughout the project and outline how these elements should be in the different versions of the product.

1.1 Plan aim

The aim of the plan is to establish and define the responsibilities, functions and activities of each member and also set up the work plan to achieve successfully the project objectives, in this way, you can ensure the integrity in the continuous evolution of the software product. The changes are produced in the life cycle of the software so they must appear in the documentation and in the software.

To guarantee the integrity certain objectives should be met, This objectives are associated with the requirements fulfilment setted up in the SRS. To achieve that procedure, we will identify the characteristics of any product that could be changed and we will prioritize these changes to develop them later.

The project objectives are developed following the UDP (Unified Development Process).

1.2 Plan scope

- Determine the iterations of the different phases that defines the life cycle of the project.
- Making a report to document the iterations and later give them to the client.
- We need to be able to go back during the project, to adequate the iterations to what the client needs, so we need:

  - Identify the configuration items.
  - Managing the repositories.
  - Development record maintenance.
  - Configuration status documentation.
- When an iteration is finished it should be verified.

1.3 Relationship with the organization and other projects

The project has been requested by the client directly.

At the time of the elaboration of this document, the company doesn't have any other project that could interfere with this one. If any new project comes up, this must appear in the documentation of the project, showing which resources would be reallocated.

1.4 Key terms

- **Baseline**: specification of components that together allow the operation of a project, with the particularity that establish the minimum basis on which this is built.
- **CM**: Configuration management
- **SRS**: Software requirements specification
- **Version**: Product status at a given time, in this case refers to a publication of an item or more of a software configuration.

1.5 References

To elaborate this document we have used:

⋆ IEE Standard for Software Configuration Management Plans byIEE Std. 828-2012.

⋆ Software Engineering II subject notes.

2 Criteria for the identification of configuration items to which the CM will apply

The CM will be apply to several elements using certain criteria:

- Elements that are modified regularly.
- Elements performed collaboratively in groups.
- Elements with an special follow up.
- Elements that needs backups.

The configuration elements of the project that are in the website or in the repository are:

- SRS
- Iterations planning and use case priorities.
- Estimated project costs
- Company quality standard.
- Company quality evaluation.
- Processes improvement.
- Software analysis model.
- Software design model.
- CM Plan.
- Visual Paradigm files.
- Implemented project source code files.

3 Limitations and assumptions that affect the plan

Those assumptions and limitations is determined by three possible factors:

- **Client factor**: along the project duration the client could suggest or demand different changes in the product development(maybe to change any requirement, maybe to change the GUI for example).Those changes will affect the delivery cost and time. In that case we should go back and change whatever we need. Because of that we need suitable feedback and regression tools.

- **Time factor**: the delivery time is determined by the development time of our organization and by the client changes. A planning error could be decisive in the delay of the project, an unexpected event could appear so the planning must have margin
- **Money factor**: The CM plan will be limited by the costs of the product development, the amount of resources assigned to a task should be planned and decree with the client.

4 Responsibilities and authorities of the Plan

- **Project manager**: He will manage the project in terms of:
  - **Baseline creation authorization**, this means that he has responsibility over the new baseline work teams.
  - **Availability assessment and resource use convenience** to carry out the suggested changes, meet deadlines and meet costs.
  - **Evaluate** the result of **the changes** made in the configuration.
- **Configuration Manager**: Manages the planning, identification, control, monitoring and auditing of configuration items, promoting the effective use of configuration management in the organization. He must be prepared to recognize that, while not having reached the end of what at first was considered as a complete iteration, it has reached a new baseline.
- **Change manager**: The function of this role is to assess change requests, determining the impact it may cause on all areas of the project (changing requirements, new costs, new deadlines) to determine whether this change is acceptable or if it is not.
- **Configuration management coordinator**: He will decide if the changes will be carried out and also will manage the processes involved on the configuration.

5 Project organization

The Project has been developed using the UDP.The work has been divided in cycles and each cycle has 4 phases detailed below:

1. **Inception**

A description of the final product is developed coming from a good specification. In this phase are identified the most important risks, the elaboration phase is completely planned and the cost is estimated.

2. **Elaboration**

   The product use cases are specified and the system architecture is designed.. The most critical use cases detected in the initial phase are developed here. At the end of this phase, you can estimate the needed resources and plan the activities.

3. **Construction**.

   The product is created, completing the baseline of the architecture to have the full system, where the product already contains all the use cases that management and customer have agreed to develop version.

4. **Transition**

   It covers the period during which the product is converted to the beta. From this version the defects will be covered, and developers will work to correct bugs and incorporate improvements suggested in the final version. This phase includes activities such as manufacturing or customer training.

6 CM responsibilities

Each member of JuManJi S.S. is committed to this conditions:

- At the end of each iteration we will get a module correctly documented and satisfying all the requirements.
- The testing process will be incremental, performing tests at the end of each iteration to test new modules created and finally, tests will be made to the full system with all integrated modules.
- The whole process should be appropriately documented and found in repositories available to the client. The documentation must follow temporary management criteria to be linked with a versioning system.

7 Applicable policies, directives and procedures

Documentation and code generation policies:

- The iterations will be properly documented.
- The possible changes and modifications have to be documented.
- The source code of the application will be divided by iterations using Maven.

Repository policies:

- The project documents will be in a repository, this repository has to be accessible by the client.
- The project changes and modifications must be documented.
- The source code will be stored in the repository, allowing everybody see the different generated versions.

Management team and meeting policies:

- The client will be able to meet the team in any moment if this meeting is notified 24 hours before.
- Once an iteration is done, the client could meet the team to see the progress.

7.1 Version Control

All project changes must follow the incoming directives:

- Regarding the general project:

  All changes must follow the incoming versioning: *CHANGE_MAYOR* . *CHANGE_MINOR* . *NO_OF_BUGS*, where *CHANGE_MAYOR* is the iteration number, *CHANGE_MINOR* is the number of functionalities added (e.g Class created or function added), and *NO_OF_BUGS* is the number of errors that have been fixed.

8 Planned activities, schedule and resources

Every activity carried out during the project is properly specified, planned and documented in project planning.

9 CM Plan maintenance

Maintaining the plan will be the task of Change Manager and the Configuration management coordinator. They will determine if the documentation fits with the regulations required for the configuration changes requests.