„Holiday Out" Hotel project

Configuration Management

Configuration management involves all the documents of a software project(code, documentation, test cases and so on..) as well as changes that occur throughout the whole process (for example changes in the documentation). Change is a basic , yet unpleasant phenomenon. This however is a neccesary thing in any software project. Change cannot be avoided, because in its core it is a response of the project team to problems (errors, adjusting), which occur in a project.

We have developed these procedures for our configuration management:

- ***ways of storing document history have been created.***
- ***document name and place has been defined.***
- ***access rights have been defined (who has access to which particular file and what are his privilages).***
- ***ways of defining, storing and resolving problems have been created.***

They will be described in detail on the following pages.

1. Storing document history

Due to the fact that our project is being made in a small and compact enviroment, we have decided to create a very easy yet efficient way of marking and storing documents.
We have decided on the following information to be stored :

- The author of the document / the person responsible for changes
- Has the document been altered
- What was the reason for the change
- What did the change involve
- Date the event took place

It has been decided that this information will be presented in a form of a simple template that will be stored either at the end of the document it concerns or as a separate file
with the following format <document_name>_history.*.*
(so for example configuration-management_history.odt).

Here is the form of our template :

document:
current version:
version after change:
action description:
reason for change (blank if no change occurs):
date:
author:

2. Document names and storage.

In a serious software developement company the documents themselves should be stored on a secure server inside the company. We however are not a real company:) thus we have decided to store the documents on some private FTP server that has some form of revision control system that would help us keep track of what is going on. The easyest way of aquiring such a server would be through PJIIT. Such a server would be perfect, as it would allow easy access to the files that would be both fast and secure, also a server inside our school would be very easy to maintain.

Of course this is not the only solution. We could also store the documents on our team leaders personal Z drive as well as CD's.. but we have decided that the first solution is a lot better.

Below the list of documents with their destination path are presented :

1. *Requirements definition*
   name: requirements_definition.pdf
   place: root\project\

2. *Task planning*
   name: task_planning.pdf
   place: root\project\

3. *Rist management plan*
   name: risk_management.pdf
   place: root\project\

4. *Quality assurance plan*
   name: quality_assurance.pdf
   place: root\project\

5. *Software testing plan*
   name: software_testing.pdf
   place: root\project\

6. *Project complexity analisys*
   name: complexity_analisys.pdf
   place: root\project\

7. *Team communication plan*
   name: communication.pdf
   place: root\project\

8. *Configuration management*

name: configuration_management.pdf
place: root\project\

3. Access rights for users

This section covers access control. The purpouse of course is to create certain „areas” for users to move in, make adjustments, changes etc. The reason for this action is very simple. We as a team want to protect the consistensy of this project.

In our case due to our team and project being rather small, access has been devided very simply yet restrictivly. Eeach person has total access to the area he/she is responsible for, and each person has access to the work of others but on (read only) rights. This simple decision enables the protection of every document and also the inspection of everyones work.

Here is a list of access rights for users :

document: requirements_definition.pdf
read/write access: Fok Damian
read access: the rest of the team

document: task_planning.pdf
read/write access: LeNgoc Minh
read access: the rest of the team

document: risk_management.pdf
read/write access: Szymerski Michał
read access: the rest of the team

document: quality_assurance.pdf
read/write access: Mirosław Dąbrowski
read access: the rest of the team

document: software_testing.pdf
read/write access: Mendyk Krzysztof
read access: the rest of the team

document: complexity_analisys.pdf
read/write access: Bondarowicz Adam
read access: the rest of the team

document: communication.pdf
read/write access: Kobayashi Ewa
read access: the rest of the team

document: configuration_management.pdf
read/write access: Staszewski Rafał
read access: the rest of the team

4. Procedures for problem management

The following standards of dealing with problems in our project have been developed(change management). The „problems registry" with the following characteristics has been created:

1. **identification part** (*includes information describing the problem as well as its current state*)
2. **notifcation part** (*includes information on the configuration element that the problem applies to , also who, when and with what actions has identified the problem and its symptoms*)
3. **planning part** (*this part defines who, how (in general) and how fast should resolve and check the problem*)
4. **solution&check part** (*this part contains information about the necesarry actions that were taken in order to resolve the problem as well as where have the changes been made and what has been checked and how much time did this take*)
5. **prevention part** (t*his part contains the classification of the problem (using general error classes*))

The use of a problems registry has the following advantages :

- it helps the overall communication in the team.
- It enables defining change boundies in the configuration elements.
- It helps the management of software projects by means of delegating tasks as well as planning their starting times.
- It helps to find the overall work consumption (time)
- It enables seeing the current state that the project is in (the number of problems recorded, the difference between problems solved and problems recorded)
- It enables better quality due to the possibility of analizing a greater number of problems as well as defining mechanisms that should prevent similar problems from occuring in the future.

Configuration management is a process that has software to help it. For our project we have chosen a comerciall tool StarTeam, because it has a built in problems registry , such a feature is greatly compatible and apriciated by our software developement team.