Software Project Management

Durga Prasad Mohapatra
Professor
CSE Deptt.
NIT Rourkela

Contents

- Change control
- Software configuration management

Change control

- When a document such as the user requirements is being developed, there may be many different versions of the document as it undergoes cycles of development and review.
- Any change control process at this point would be very informal and flexible.
- At some point of time, the final version will be created. This is *baselined, effectively frozen* .

Change control cont...

- Baselined products are the foundation for the development of further products
  - for instance, interface design documents may be developed from baselined user requirements.

Change control cont ...

- Any changes to the baselined document could have knock-on effects on other parts of the project.
- For this reason subsequent changes to baselined documents need to be stringently controlled.

Steps in Change control procedure

- One or more users might perceive a need for a modification to a system and ask for a change request to be passed to the development staff.
- The user management would consider the change request and, if they approve it, pass it to the development management.
  - There is a single authorized channel for *requests for change* (RFCs) between the client community and the management of the developers.
  - There would be some filtering within the client community to ensure that the proposed change does genuinely provide a benefit before the RFC is generated.

Steps in Change control procedure cont ...

- There would be one person within the development area who would receive and process RFCs.
  - They would delegate a member to look at the request and to report on the practicality and cost of carrying out the change.
  - The developer would assess the products that would be affected by the change.
- The development representative would report back to the user management on the findings and the user management would decide whether, in view of the cost quoted, they wish to go ahead or not.

Steps in Change control procedure cont ...

- There would be some individual or group (may be Project Board) who represented the major stakeholders, both users and developers and also the project sponsor, who would have the authority to prioritize the RFCs for action.
- A further step is to give the project managers allowances that would allow them accept minor changes (as long as they are documented with an RFC, etc.) as long as they do not exceed planned cost and delivery targets.

Steps in Change control procedure cont ...

- A very large number of seemingly small changes could have a serious accumulative effect on project progress which may call for the attention of higher management.
- A very large set of changes might trigger the project manager to produce an exception report.

Steps in Change control procedure cont ...

- Once an RFC has been approved for action, one or more developers are authorized to take copies of the master products that are to be modified (not the master products).
  - This would need to be done through the configuration librarian.
- The copies are modified.
  - In the case of software components, this would involve modifying the code and recompiling and testing it.

Steps in Change control procedure cont ...

- When the development of new versions of the product has been completed
  - the user management will be notified and copies of the software will be released for user acceptance testing.
- When the user is satisfied that the products are adequate, they will authorize their operational release.
  - The master copies of configuration items will be replaced.

Duties of configuration librarian, configuration manager or project librarian

- Identification of all items that need to be subject to change control
- Establishment and management of a central repository of the master copies of all project documentation and software products
- Setting up and running of a formal set of procedures to deal with changes
- Maintenance of records of who has access to which library items and the status of each library item (e.g. whether under development, under test or released)

Typical change control process

1. One or more users might perceive the need for a change
2. User management decide that the change is valid and worthwhile and pass it to development management
3. A developer is assigned to assess the practicality and cost of making the change
4. Development management reports back to user management on the cost of the change; user management decides whether to go ahead based on the cost of change.

Change control process cont...

5. One or more developers are authorized to make copies of components to be modified
6. Copies are modified (not the original version). After initial testing, a test version might be released to users for acceptance testing
7. When users are satisfied, then operational release is authorized – master configuration items are updated

Software Configuration Management (SCM)

- Changes in a project can take place in any of the work products and may be due to many reasons such as bug fix, changes on account of work simplification, efficiency considerations, etc.
- Change management can be done manually by a designated configuration librarian.
- However, the manual change management process gets overwhelmed
  - when we consider changes taking place on all work products and
  - when there are multiple variants of the product.

Software Configuration Management (SCM)

- In this situation, a systematic software configuration management (SCM) process with appropriate tool support needs to be deployed.
- SCM is concerned with tracking and controlling changes to a software.
- In any systematic development and maintenance environment:
  - Various work products (such as code, design document, code, etc.) associated with the software continually change during the development as well as the maintenance phase.

Software Configuration Management (SCM)

- In a team development environment, each member of the development or maintenance team would be assigned to handle some modification requests.
- Therefore every work product would have to be accessed and modified by several members.
- In such a situation, unless a proper configuration management system is deployed, several problems can appear.

What is configuration management?

- The set of activities through which the configuration items are managed and maintained
  - as the product undergoes its life cycle phases.

Context in which Configuration Management is necessary

- During the development phase, the work products get modified as development activities are carried out.
- During the maintenance phase, the work products change due to various types of enhancements and adaptations that are carried out including bug fixes.
- Thus, the state of the work products continually change both during the development as well as maintenance phase.

Context in which Configuration Management is necessary cont …

- The state of all work products at any point of time is called the configuration of the software product.
- Software configuration management deals with effectively tracking and controlling the configuration of a software product during its entire life cycle.

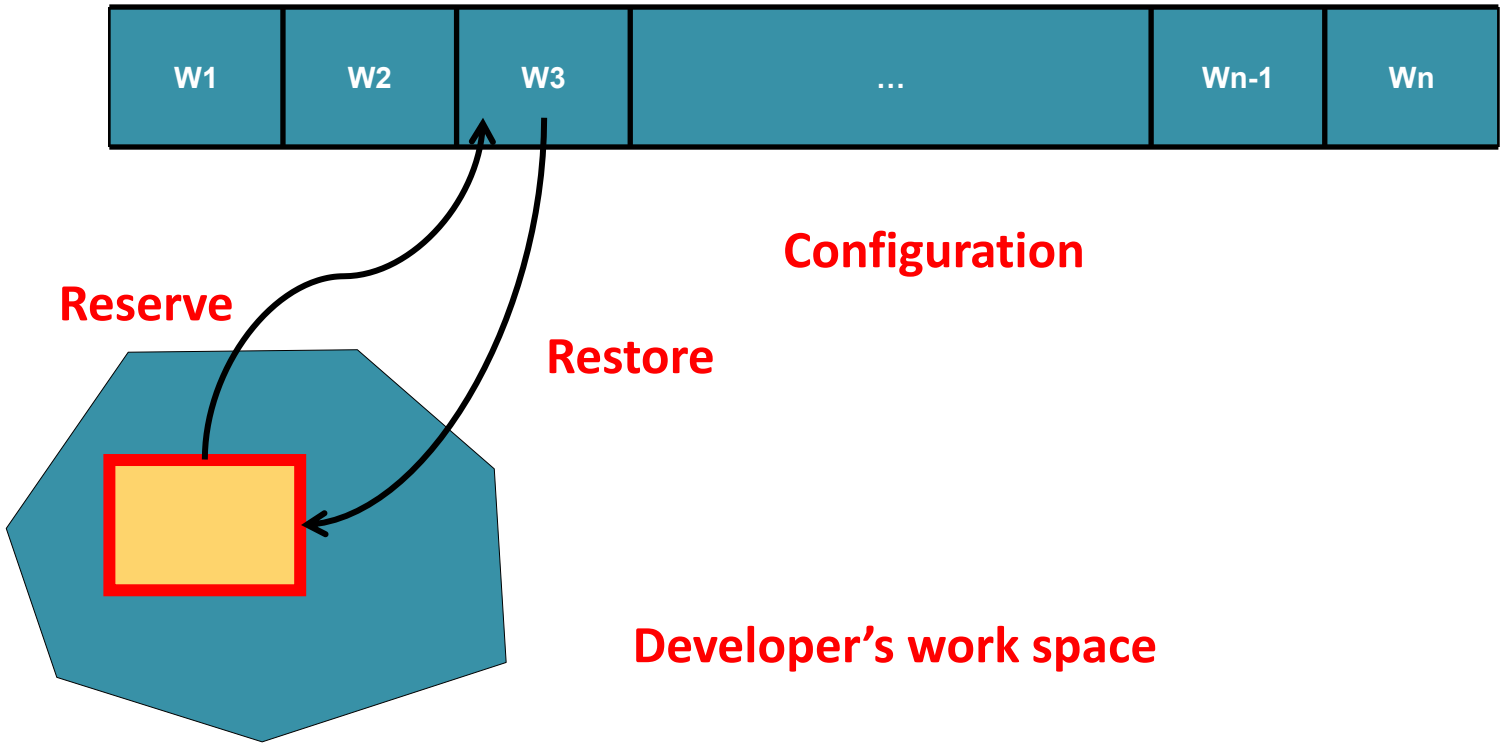
Context in which Configuration Management is necessary

- For effective configuration management, it is necessary to deploy a configuration management tool.
- There are many configuration management tools available, some are open software, i.e. free of any licensing fees, and others are commercial tools.
- Configuration management practices include version control and the establishment of baselines.

Configuration

- The configuration of software is the state of various work products that are under configuration control.
- The work products that are under configuration control are usually referred to as the configuration items.
- It is convenient to think of a configuration as a set of files representing various work products.
- For example, the configuration of a sample software product shown in next figure consists of the configuration items (work products) *W1*, *W2*, …,*Wn*.

Work product modifications under configuration management



Version

- As development and maintenance activities are carried out on a software product, its configuration (that is, one or more configuration items) keeps changing.
- It often becomes necessary to refer to the configuration that existed at certain point of time.
- For example, we can say that refer to the last week's configuration of the software.

Version cont …

- Therefore, a version is a configuration that existed at certain point in time.
- More technically, versioning is a numbering scheme that helps us identify a specific configuration at a certain point in time.
- This is achieved by a configuration tool by tagging the files resenting the configuration items with the version name.

Version cont …

- As a software is released and used by the customers:
  - errors are discovered,
  - enhancements to the functionalities might be needed.
- A new release of the software is an improved system intended to replace an old one.
- Usually a product is described as version m and release n (or as version m.n)

Revision

- A revision system is a numbering scheme that is used to identify the state of a configuration item at any time.
- Each time a work product is updated, its state changes.
- Thus, we can think of a work product going through a series of updates till it reaches a desired state.
- The successive states of a work product are its successive revisions.
- Thus each time a configuration item is updated, a new revision gets formed.
- It becomes possible to refer to a specific state of a work product by using its revision number.

Baseline

- A baseline is a software configuration that has been formally reviewed and agreed upon, and serves as a basis for further development.

Variant

- Variants are versions that are intended to coexist.
- Different variants may be needed to run the software on different operating systems or on different hardware platforms.
- For example, one variant of a mathematical computation package might run on Unix-based machines, another on Microsoft Windows machines, and another on Solaris.

Variant cont …

- Variants may also be required to be created when the software is intended to be used with different levels of sophistication of the functionalities (e.g., novice version, enterprise version, professional version, etc.).
- Variants are often created during the operation phase, during the development phase, and as and when software products with overlapping functionalities are required.
- Even the initial delivery of software might consists of several versions and more variants may be created later.

Purpose of SCM

- Problems associated with concurrent access
- Undoing Changes
- System accounting
- Handling variants and helping fix bugs in them
- Accurate determination of project status
- Preventing unauthorized access to the work products

Problems associated with concurrent access

- Possibly, the most important reason for configuration management is to control the access to the different deliverable objects.
- Unless strict discipline is enforced regarding update and storage of different work product, several problems can appear.
- Let us assume that only a single copy of a program module is maintained, and several developers are working on it.
- Two developers may simultaneously carry out changes to the different functions of the same work product, and while saving overwrite each other.

Undoing Changes

- It becomes easy to undo some part of a revision or even rollback development to a certain version.
- Unless proper configuration management system is in place, it becomes very difficult to undo a change.

System accounting

- System accounting denotes keeping track of who made a particular change to a configuration item, what change was exactly made, and when the change was made.
- Knowing the what, who, and when of changes will help in understanding why changes were made and whether some changes are redundant or for comparing the performance of particular versions.
- It may at times be required to rollback to a previous baseline if a change is not justified or is improper.

System accounting cont …

- Users may wish to compare today's version of some software with yesterday's version or last year's version.
- Since a configuration management system keeps track of every version and revision, this becomes a simple task.

Handling variants

- It often becomes necessary to create variants.
- Without a configuration management system, keeping track of all variants, their versions and revisions is a nontrivial task.
- Further, existence of variants of a software product causes some peculiar problems.
- Suppose you have several variants of the same module, and find that a bug exists in one of them. Then it has to be fixed in all versions and revisions.

Handling variants

- To do it efficiently, you should not have to fix it in each and every version and revision of the software separately.
- Making a change to one program should be reflected in all relevant versions and revisions.

Accurate determination of project status

- Normally, a project manager performs the configuration management activity by using a configuration management tool.
- In addition, a configuration management tool helps to keep track of various deliverable objects so that the project manager can quickly and unambiguously determine the current state of the project.
- The configuration management tool enables the developer to change the various components in a controlled manner.

Preventing unauthorized access to the work products

- Configuration management helps implement a controlled change process.
- It therefore becomes possible to prevent unauthorized changes to the work products.

Summary

- Discussed the need of change control.
- Also discussed the change control procedure.
- Explained the concept of software configuration management, the context in which it is necessary and it's purpose.

References :

1. B. Hughes, M. Cotterell, R. Mall, *Software Project Management*, Sixth Edition, McGraw Hill Education (India) Pvt. Ltd., 2018.
2. R. Mall, *Fundamentals of Software Engineering*, Fifth Edition, PHI Learning Pvt. Ltd., 2018.

Thank you