





Home

Home > My courses > System Integration And Architecture 2 > 08 Software Implementation > Lesson Proper for Week 8

Lesson Proper for Week 8



CONFIGURATION MANAGEMENT

The configuration management software team or organization ensures that configuration management practices are applied consistently throughout the software life cycle for work products that are developed and maintained by programs and projects. The team focuses on identifying and managing changes and maintaining software configuration and documentation visibility. The configuration management concept is a cross-functional process applied over the life cycle of a software work product and provides visibility and control over functional and physical attributes. The processes that are used during all phases of software design/development provide the necessary disciplines that identify applicable products, establish and control baselines, and document and track changes to those baselines. Also, configuration management processes control the storage, access, changes, archive, and release of the software work products.

This team develops operating procedures that describe implementation of processes required to satisfy the requirements and direction provided under associated and documented plans.

Ø Build Requests

When software engineering builds are requested, electronic files or hard copy paperwork is written to provide build checklists to assemble, compile, link source code, build archive copies, and provide listings for use in software design/development, test, and work product deliveries to customers.

Automatic generation of build deployments ensures customer confidence in the releases. For the program and projects to be successful, processes used by build engineers include the capability to package builds and documentation together. Creating an approach to meet build and installation processes requires coordination



between internal and external teams to become efficient and available when supporting scheduled tests or configuration checkouts.

The build engineer with the direction or authorization for a requested engineering build has a defined role to perform tasks related to software construction and configuration control, including the following:

- Creates build folders to store documentation of software building
- · Provides source code changes and control of the source code
- · Maintains and controls records during program and project development



CONFIGURATION MANAGEMENT TOOLS

The management and use of effective configuration management software tools provide version control and change management concepts. The tools (i.e., ClearCase and ClearQuest) may be used to provide the capabilities for adding new files to a software design/development environment and provide version control to applicable directories and files. File sharing, parallel software design/development, multiple team support, and software reuse are essential for meeting integration test activities demanded by the schedule. The configuration management software team administrates or manages software tools. Table 6.1 provides an example and overview of tools included in a software and systems integration environment.

TABLE 6.1: Configuration Management Tools

| Tool or Vendor | Software Activity Support | Host System | Purpose of SCM Tool |
|---|---|-------------|--|
| ClearCase (IBM) | Design, code, and unit test, software builds/installation, integration, and test | UNIX/PC | Tools for documentation and source code, support multiple developments, and release baselines |
| ClearQuest (IBM) | Code and unit test, integration, and integration testing | PC | Software problem reporting, logs, tracking, and software debugging and fixes |
| FORTE (SUN) APEX Rational Clearmake (IBM) | Software engineering builds | UNIX | Compile and build released software executable products |
| Microsoft Office | Software engineering activities | PC | Support documentation, software design/ development, e-mail communication, and data analysis |

Ø IBM Rational ClearCase

The configuration management tool for software design/development that I have experience with and used in military and aerospace programs and projects has been IBM Rational ClearCase. This software tool is an object-oriented database utility provided to establish software product archiving, automation, identification,



version/change control, engineering building, product releases, and status accounting, and auditing activities. The ClearCase software tool provides an open architecture to implement configuration management and control solutions. Web site content for computer software companies, the military, and aerospace industries employs many different development environments.

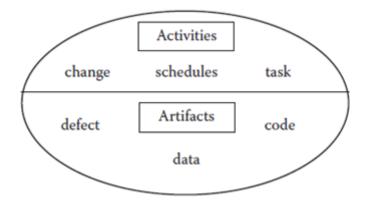


FIGURE 6.1: Unified change management definition.

The concept of unified change management (UCM) combines activities and artifacts as shown in Figure 6.1.

Many programs and projects use different names for ClearCase roles and assignments, but configuration management can use the following as an example:

- · Configuration manager: Familiar with change and control processes
- Lead: Responsible for schedules and assignments
- Software design engineer: Makes changes to files under configuration control
- · Build engineer: Utilizes software build concepts and tools

A functional overview of a ClearCase concept is the repository named version object base (VOB). This is a data repository where files, directories, and data are stored. All files and directories are managed inside the VOB and can expand from hundreds of files and directories to thousands. Table 6.2 defines the ClearCase—UCM roles, responsibilities, and main objectives. The files and directories can be moved or transferred to other VOBs when the repository becomes too large. They can also be split and work together. The ClearCase architecture and the VOB database ensure the checkout of files and support data recovery if needed.

TABLE 6.2: ClearCase—UCM Roles and Responsibilities



| Role | Main Objectives | | |
|---------------------------|---|--|--|
| Architect | Define models (architecture) | | |
| Configuration manager | Set up configuration management environment (i.e., repositories, importing files, etc.) | | |
| SCM lead | Assign and schedule work activities and define written software configuration management policies | | |
| Software design/developer | Make changes to files/directories and deliver software to build engineers | | |
| Build engineer | Builds components for established baselines ready for test | | |

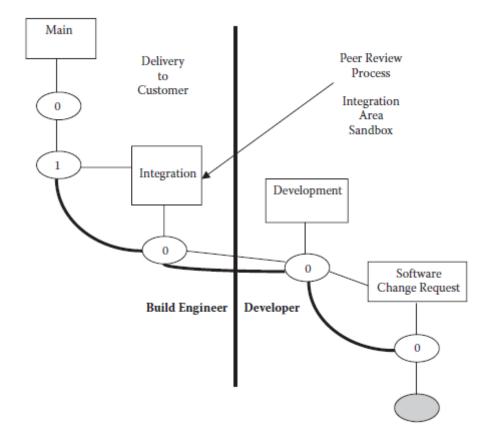


FIGURE 6.2: ClearCase VOB architecture.

The ClearCase VOB structure example is shown in Figure 6.2.

Ø IBM Rational ClearQuest

The change request management process is critical when reporting any requests from team members that are needed to change or update software and systems integration work products. IBM Rational ClearCase comes into play with another software configuration management (SCM) tool, IBM Rational ClearQuest.

This software tool provides support for change request management processes and is a complementary tool for ClearCase. The database utility is used for recording, tracking, and reporting and provides internal access control mechanisms for permitting the restriction of work product updates at various stages of software design/development, integration and test, and production processes.

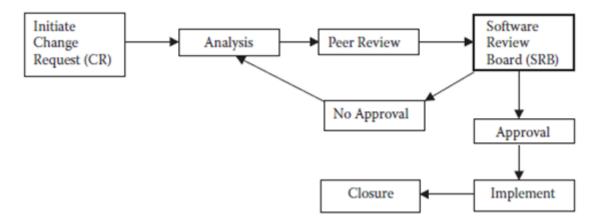


FIGURE 6.3: Change request process.

The change request is a request from team members to change an artifact or process. Documented in this type of request is the information for problems occurring during software design/development and impacts that could occur. An example of the change request process flow from initiation to closure is shown in Figure 6.3.

The change request administrator maintains a tracking system to control software code and software documentation change status. The administrator manages the database, coordinates, and provides software inputs to program and project changes to establish traceability to a higher-level change authority with an impact on software. Software plans and process procedures provide the information to coordinate the required review boards to maintain records, including status of change requests, reports, and documented releases.

This review board is established for software teams to review and disposition changes that affect controlled software and related documentation. All software changes are documented, approved, and implemented per a change request. The review board meetings are scheduled and coordinated by the program or project manager, leader, or designated representative serving as the review board chairperson.

The review board members include, as a minimum, those from the following areas:

- The affected software teams
- · Configuration management
- · Test
- System engineering
- Quality
- · Security
- Change sponsor



The major activities performed by the review board are evaluations and dispositions of change requests, assignment of priorities, review of action items, change dispositions from prior meetings, and the evaluation of deviations that occurred for discussion.

There are many configuration management software tools used in military and aerospace programs and projects that can be discussed. The software tools selected are required to fit the environment used by the teams and for software and systems integration activities. Do not take my word that the Rational ClearCase and ClearQuest are the only usable software tools. Other tools will work and support the program and projects, so they will be okay.



SOFTWARE MEDIA AND DATA

The physical software media (i.e., disk units, CD, DVD, hard drives, etc.) identification and media labels must also be in accordance with the program and project documented media requirements in affected plans.

Marking information could be displayed electronically on the exterior of the physical media containing the software or provided within the media through a file in each piece of software data or a written set of electronically submitted files (identified as .doc, .txt, executable files, etc.). These media files reside in a computer media library (CML) for engineering use. Copies of the software media generated are verified and validated by a quality team.

An example of what could be documented on a media label follows:

Date: Day/month/year format

Title: Document the title of the software being produced

o Derived from: Program and project

o Special handling: Distribution requirements

Contract number: Document contract number

o Part number: Document software identifier

o Software version: Media version



FUTURE TRENDS

There are major improvements in software technology and future trends for effective use of software tools. With the technology, there will be time to address and resolve issues and improvements required for:

- · Software design/development
- · Software process definition and enhancements
- · Reuse of software program and project artifacts



- · Ongoing support of past tool artifacts
- · Training for software design engineers
- · Software tool disciplines

The software design/development emphasizes SEE (software engineering environment) technology to allow detailed definitions of the required roles and responsibilities for users and if organizations related to the program and projects are ready. Acquiring software tools should not be a solution to show tools that are out of control for software design/development but ensures processes are defined for the management for software development activities.

Having effective software tools in place will improve software design/ development and quality produced and increase the productivity for software and test engineers. The insertion of SEE technology in the program and projects is successful when implementation plans are well defined.

The main reason for software tools and how they are adapted to requirements should be based on how these tools can approach implementation of the design/development. In the future, many tools for software design/development will support the life-cycle work products and the processes defined by the user. The major obstacle will be the balance and control for a stable software implementation plan and for adapting to changes that occur.

Ø Tool Support

When a program or project is ready for software tools that will be effective during design/development, the key is selecting the right vendor products to match engineering needs. Questions are asked, and the primary steps for organizational needs are as follows:

- · Become effective for designing and developing work products
- · Establish the resources for use of software tools
- · Conduct software implementation with no problems
- Conduct training



CONCLUSION

The major building block of software design/development improvement is to make sure the automation of software tools is understood. Costs are significant for short- and long-term use. In programs and projects, it is critical that the organizations enhance software implementation for productivity and quality.



Jump to...

Analysis, Application, and Exploration for Week 8 ▶



Navigation

Home



Dashboard

Site pages

My courses

Capstone Project 1

Network Attacks: Detection, Analysis & Counter...

Ojt/Practicum 1

Social And Professional Issues

System Integration And Architecture 2

Participants

General

06 - Preliminary Examination

07 Software Requirements

08 Software Implementation

Preliminary Activity for Week 8

📄 Lesson Proper for Week 8

🗸 Analysis, Application, and Exploration for Week 8

Generalization for Week 8

Evaluation for Week 8

Assignment for Week 8

Courses



Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission.

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.



COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.



Activities



Assignments



肩 Forums



Quizzes



Resources

Bestlink College of the Philippines College Department

Powered byeLearning Commons

