



**Romel Cabiling** ▾



Home

Home > My courses > System Integration And Architecture 2 > 14 Software And Systems Integration (Cont.) > Lesson Proper for Week 14

# Lesson Proper for Week 14



## SYSTEMS/SOFTWARE DESIGN

The software design definition is developed and controlled by plans for development or design. The constraints for the software are identified during the start of the software design/development life cycle. There are objectives that are required to meet schedules from the start to the end of software and systems integration.

The software design engineer identifies risks and software restraints that could occur during development and hinder schedules. The software requirements are analyzed for software and systems integration to ensure the software and systems or systems design work together and are viewed to make continual improvements.



## INTEGRATION

Before performing software and systems integration, lab operations implements a readiness review to ensure the lab environment is ready for design and testing. What I mean by a readiness review is to create a high-performance work team (HPWT). The following trained personnel should be in this team:

- Systems designer
- Systems engineer
- Software designer
- Configuration management
- Quality personnel
- Hardware designer



- Subcontractor (if required)

The HPWT will perform a software engineering review and audit of each discipline in the software life cycle to ensure processes are being followed per defined plans, procedures, and expected requirements from the customer. By performing this audit and review, results are reviewed and documented by the HPWT and presented to senior management for discussion with affected program and project managers. It is hoped that before programs and projects are ready to integrate work products into the integration environment, processes are in place and compliant. The program and project managers do not want to hear the following:

- "I have technical issues and concerns in terms that are understood by other team members."
- "My software does not work when delivered for software and systems integration."
- "Processes? We don't have any processes defined and implemented."

### Ø Team Coordination

When conducting informal or formal peer reviews, ensure guidelines are understood by the team members from the start to the finish. Team assignment responsibilities define the data collected for each peer review and which tools are used to establish, collect, and store the required data.

Maintain the schedule or plans for new and revised work products to include peer reviews at the completion of the entire work product. The scheduled plan for a peer review should be divided and show incremental peer reviews immediately after completion of each section and should not delay these until the end of the phase when there is limited time for rework. The schedule includes dates for team training.

### Ø Plans and Procedures

Before software and systems integration testing is started, test plans, documents, and procedures are required to be released through a documentation release system. All software and systems integration tests are performed based on definitions of elements that are documented and identified during integration.



## EXECUTION

The software and system integration recommendations are to show execution of test-built systems for integration activities and to ensure the builds provided for execution are not broken. Build and test times should be reviewed to minimize problems that occur during the software and system working together for integration activities. Acceptance tests are performed along with the customer as witness depending on the program and project requirements.



## Ø Acceptance Test

The acceptance test approach or methods will provide answers to questions asked such as whether the code will do the right thing during software and systems integration. There are many considerations that apply, such as:

- Understanding of the specification
- Efficient integration of design and test
- Improvement of processes
- Decreased regression tests and costs

Requirements define acceptance testing to validate implementation of software and systems integration. The features must have acceptance to involve automation methods, which is my first choice over manual steps. The software and systems or units that are working together are accepted when completion occurs. I feel acceptance testing shows that the code provided by the designer is working as expected and has been peer reviewed and tested. Remember that the software and system are not ready for release to the customer, so perform a readiness review before production.



## CONTINUOUS INTEGRATION

Continuous integration is the automation of build and test processes, starting first with the software code being checked into the computer media library (CML) repository. Teams can assure that the code quality is under configuration control. The automation of the build and testing approach should be implemented to support the following:

- Source code capabilities
- Confidence in builds and testing
- Restoration of previous configurations
- Interaction of compilers and systems design personnel
- Building entire systems from scratch
- Team awareness of builds and test failures

## Ø Automation

The automation and generation of software and systems packaging ensure confidence in personnel requests by the integration labs or the customer. Staging builds and tests together supports requirements that could include the package to facilitate the integration process.



Dedicated systems or a hardware platform should run tests continuously. The software designers could hinder and slow testing if this discipline is not applied. There are always new types of tests to be conducted. Test artifacts created by the design team provide a start to build automation.



## **CONFIGURATION MANAGEMENT**

The definition of CM is a discipline applying an administrative process and direction for work products developed during the software or hardware life cycle. The importance of CM is to identify and document both functional (as-designed) and physical (as-built) work products.

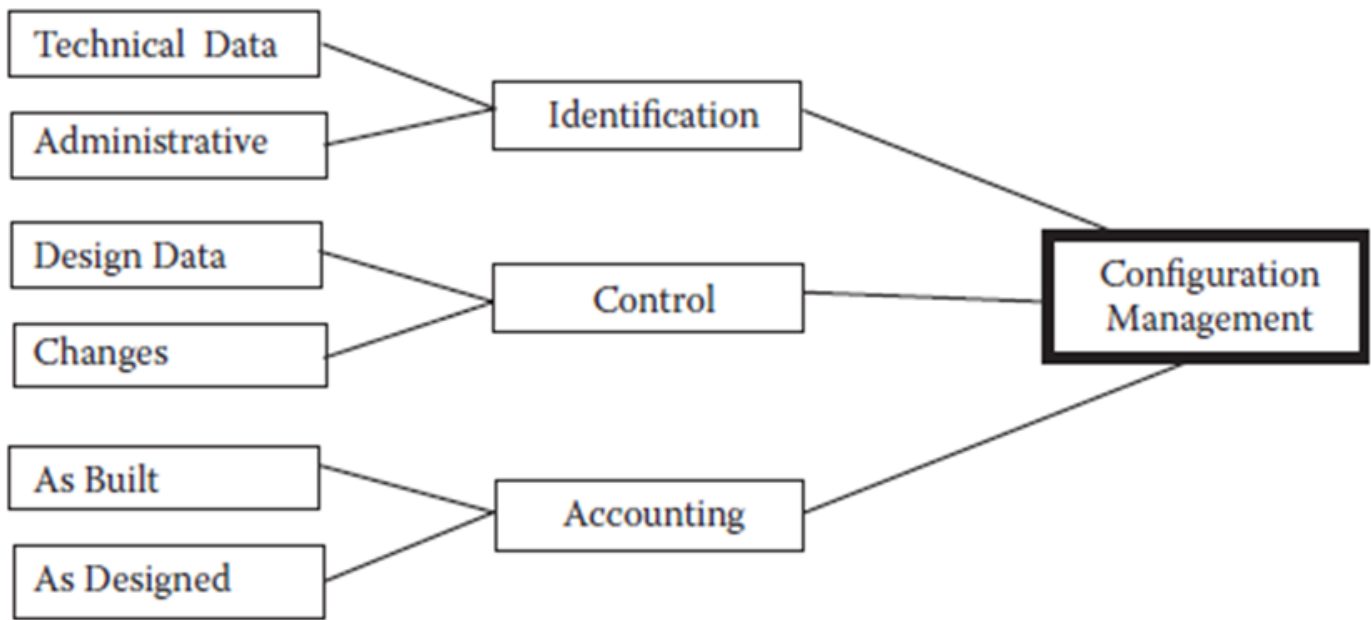
The control of changes to work product plans and procedures during software and systems integration provides and records the information required to manage work products more effectively. Authorization is an important factor; with it comes performing the methods for integration activities in lab environments.

Many programs and projects are required to document a configuration management plan (CMP) to define how to implement sound and effective CM policies. The configuration status accounting (CSA) method for recording and reporting information needed to control work products includes:

- Technical data
- Administrative data
- Design data
- Changes
- As-built products
- As-designed products

By looking at CM, you can see the programs and projects have a better understanding of how each part constitutes the functions such as identification, configuration control, and status accounting and the importance of CM. The CM method is shown in Figure 8.4.





**FIGURE 8.4:** Method of configuration management.



## QUALITY

The essential practice that teams should follow is quality. The more quality teams adopt, the more successful they will become. At times within the integration activities for both software and systems, teams will see higher-quality software and systems efficiently produced and ready for delivery to customers.

As mentioned in software design, peer reviews and testing can be implemented early in the software and systems integration activities. The concept of automation in building and testing software to ensure systems work together provides the framework for how software design teams write software.

Work toward perfection and show continuous improvement early in testing and evaluations. Teams always think about the best way to improve the processes for testing. Making mistakes or wrong decisions early provides opportunities to learn from mistakes and have them addressed correctly.

Quality attributes ensure teams build and test the right work products. These quality attributes are:

- Build and integrate the correct work product
- Build and integrate the work product to be correct

The relationship between program and projects makes it much easier to understand the requirements early in the software life cycle. At an early point in the life cycle, customers have opportunities to be involved and provide comments and recommendations.



The major goal of effective building and testing for integration is to prevent problems or defects early during the life cycle. Quality is the responsibility of everyone on the team. In software, quality involves communication with other members and discussions of issues and concerns. In testing, teams are required to pursue perfection. To be a success, perform quality disciplines as early in the process and show improvement in all those quality aspects that involve:

- Code quality
- Peer reviews
- Builds
- Testing

**TABLE 8.2:** Code Quality

Quality	Description	Ensure
Code and unit test	Tested software interfaces	Test coverage on systems
Rules applied	Prevent redundancies	Roles and responsibilities
Team work	Communication	Clarity and understanding
Desktop integration	Code design and test process	Early detection deployment

Code quality and peer reviews are the key elements to ensure software and systems integration are working together. Consistent patterns for using quality solutions solve problems that consistently appear. Experienced software designers incorporate better solutions, and terminology is understood from the start of code development. Make sure conversations are held with team members so there is consistent understanding about code quality. The software architecture allows code to be more manageable and for changes to exhibit good-quality attributes for code development as shown in Table 8.2.

Ø **Peer Review Assurance**

I am a strong advocate for peer review assurance and will always be. Allowing software engineers to build software often and provide software that is ready for the test team to minimize the number of problems or changes at one time will reduce the risk of defects and errors in the software.

The CMMI process model provides an understanding of integration processes for software and systems integration. The integration process requires a continual emphasis on “repeatable processes.” Conducting peer and code reviews defines verification and validation when it is time to audit the software processes performed in the software and systems labs.



Software designers work to requirements, write effective code, and take pride in being considered excellent or exceptional at their profession. The peer and code review process, which is a repeatable process, ensures that other software designers review their code with proper software tools or pair programming compliance to coding standards. Implementation of high coding standards enforces software processes that are implemented with defects or issues are resolved early in software design/development stages, before software and systems integration starts.

## Ø **Software and Systems Assurance**

A fundamental objective of software and systems assurance is continuous improvement in the quality of work products and processes during integration. The assurance and process improvements are achieved by defining, documenting, measuring, analyzing, and improving the development and integration processes to reduce error rates and flow time. The requirements for assurance are documented in plans for software design/development and a software quality plan (SQP). Results are reviewed by integration teams.

## Ø **Additional Quality Concepts**

There are additional quality concepts to review and understand when conducting software and systems integration. If we are unable to explain or say what quality is, then how does anyone know quality exists? Quality does exist. What mechanism or tools helps us understand quality and its meaning and definition? I have heard many engineers and test teams' question what quality is and why it is needed.

In software design/development, the quality concepts focus on the degree to which software disciplines are implemented. I have said this repeatedly; if senior, program, and project managers involve all teams in planning to understanding schedules, and contribute and support presented schedules, then the software and test team will have confidence from the start to succeed. The quality concepts will be made easier for the quality teams to support the software and test teams better.

## Ø **Improving Quality and Productivity**

Improving quality and productivity for software basically indicates how well quality meets the requirements and expectations for supporting software and systems integration tasks. This assurance provides adequate, reliable, and efficient software design/development lifecycles. The growth in computer use for software and systems integrations places demands on increasing high-level use and complexity.

The use of effective technology is a means to improve quality and productivity for programs and projects. Military and aerospace companies have looked at software technology as a means to improve quality and better predict costs and schedules required to develop and maintain very complex software systems. Current and future technologies support software design/development processes throughout automation of software engineering



practices. The quality infrastructure is a means to integrate the disciplines that assist systems and software designers, CM, tests, program and project managers, and so on. The communication between team members promotes other team members for improving quality and productivity during software and systems integration.



## **CUSTOMER SATISFACTION**

Customer satisfaction is the concept of assuring the customer that effective methods for software and systems integration have been compliant and do meet concrete requirement expectations. Many mistakes are made when programs and projects work to poorly defined requirements from the start. Poor execution to system design and software requirements compromises the quality of integration methods to deliver value or causes outright failure. Managers need to ensure customer expectations are understood; unexpected surprises could wreak havoc on program and project schedules.

When poor software and systems integration methods are not effective, program and project schedules lead to major problems with customers.

Everyone wins when there is more focus on the success of the program and project to meet budgets, schedules, and the customer's satisfaction. Strong management, effective team support, and the understanding of what is required to be successful will lead to software and systems that are in alliance with business needs as well as user expectations.

## **Ø TAKING THE INITIATIVE FOR CHANGE**

How many times have we heard someone say, "That's not my job," and "I don't want to change the way I do things"? It does happen often in the field of software development. We know that things will become better when we resolve issues and concerns, but at times we do not take the initiative to improve or there is not enough time. The creation of software and systems integration problems has generated change that affects programs and projects. Numerous changes are good, but taking time to get organizations working together can cause problems. When someone states, "We can make integration of work products more effective," we are not perfect. There is an integration environment established to find problems and fix them. People who are working toward improvement shun others, do not listen, and then walk away. The software companies need to fix this in their organizations, even down to the teams supporting each other during integration activities.

When you see that problems occur during activities (i.e., software design/ development, testing, CM, quality testing) that are supporting integration, you need to become an expert and take the initiative to change. Improving processes to better fit your work environment will have everyone on the same page. No more walking by and acting like you do not want to help solve problems. That is your job. Change the way you do things now.








## 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

12 - Midterm Examination

13 Software And Systems Integration (Cont.)


14 Software And Systems Integration (Cont.)

 Preliminary Activity for Week 14

 **Lesson Proper for Week 14**

 Analysis, Application, and Exploration for Week 14

 Generalization for Week 14

 Evaluation for Week 14

 Assignment for Week 14

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.

## Graduation Announcement



**BESTLINK COLLEGE  
OF THE PHILIPPINES**

# ANNOUNCEMENT

Due to the insistent demand of BCP graduates and alumni and the IATF pronouncement of the low Alert Level Status, and in coordination with the DepEd and CHED, the BCP Administration is happy to announce that face-to-face graduation rites will proceed as scheduled.

<u>Level</u>	<u>Date of Graduation</u>	<u>Venue</u>	<u>Graduation Fee</u>	<u>Downpayment</u>
SHS	July 16, 2022	MV Campus	P 1,000.00	P 200.00
College	July 10, 2022	PICC	P 4,000.00	P 500.00

Balance must be paid two (2) weeks before the date of graduation.





**BESTLINK COLLEGE  
OF THE PHILIPPINES**

# **SCHEDULE GRADUATION PHOTOSHOOT**

**College Department Batch 2021-2022  
(Main Campus)**

**May. 23 & 30 - CRIM**

**May. 24 & 31 - EDUC**

**May. 25 & Jun. 01 - BSBA/BSOA/BSAIS/ENTREP**

**May. 26 & Jun. 02 - BSIT/BLIS/BSP/BSCpE**

**May. 27 & Jun. 03 - BSHM/BSTM**

**"Be trained to be the best, be linked to success"**



More information call us (028)4420-8601 | (028)8518-8050







**BESTLINK COLLEGE  
OF THE PHILIPPINES**

# **SCHEDULE GRADUATION PHOTOSHOOT**

**Bulacan Branch Batch 2021-2022**

**May 28**

**8am - 12nn: Senior High School**

**12nn - 5pm: College Department**

**"Be trained to be the best, be linked to success"**



More information call us (028)4420-8601 | (028)8518-8050



## **Activities**



Assignments



Forums



Quizzes



Resources



