VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

Digital Assignment - I, AUGUST 2019

B.Tech, Fall-2019-2020

NAME	PRIYAL BHARDWAJ
REG. NO.	18BIT0272
COURSE CODE	ITE1005
COURSE NAME	Software Engineering Principles & Practices
SLOT	C2+TC2
FACULTY	Prof. SANTHOSH KUMAR S V N

PERT tools (Planning tool)

The **program** (or **project**) **evaluation and review technique (PERT)** is a statistical tool used in project management, which was designed to analyse and represent the tasks involved in completing a given project.

What is PERT in project management?

- PERT is a project management planning tool used to calculate the amount of time it will take to realistically finish a project.

- PERT stands for Program Evaluation Review Technique.

- PERT offers a management tool, which relies "on arrow and node diagrams of *activities* and *events*: arrows represent the *activities* or work necessary to reach the *events* or nodes that indicate each completed phase of the total project."

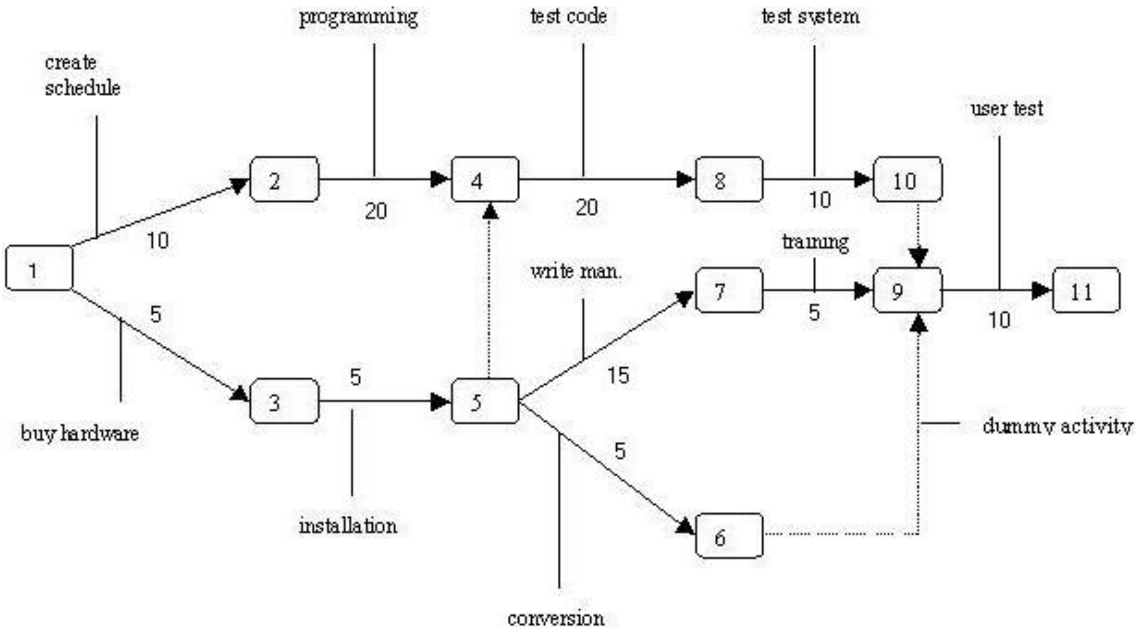
- PERT charts are tools used to plan tasks within a project - making it easier to schedule and coordinate team members accomplishing the work.

- PERT charts were created in the 1950s to help manage the creation of weapons and defence projects for the US Navy. While PERT was being introduced in the Navy, the private sector simultaneously gave rise to a similar method called Critical Path.

- With PERT, you create three different time estimates for the project: you estimate the shortest possible amount time each task will take, the most probable amount of time, and the longest amount of time tasks might take if things don't go as planned.

- PERT is calculated backward from a fixed end date since contractor deadlines typically cannot be moved.

What is PERT chart?



- PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered *nodes* (either circles or rectangles) representing events, or milestones in the project linked by labelled *vectors* (directional lines) representing tasks in the project.
- The direction of the arrows on the lines indicates the sequence of tasks.
- In the diagram, for example, the tasks between nodes 1, 2, 4, 8, and 10 must be completed in sequence. These are called dependent or serial tasks
- The tasks between nodes 1 and 2, and nodes 1 and 3 are not dependent on the completion of one to start the other and can be undertaken simultaneously. These tasks are called parallel or concurrent tasks.
- Tasks that must be completed in sequence but that don't require resources or completion time are considered to have *event dependency*. These are represented by dotted lines with arrows and are called *dummy activities*.
- For example, the dashed arrow linking nodes 6 and 9 indicates that the system files must be converted before the user test can take place, but that the resources and time required to prepare for the user test (writing the user manual and user training) are on another path.
- Numbers on the opposite sides of the vectors indicate the time allotted for the task.

Word Processors (Editing tool)

What is a word processor?

A word processor is software or a device that allows users to create, edit, and print documents. It enables you to write text, store it electronically, display it on a screen, modify it by entering commands and characters from the keyboard, and print it.

Standard Features of Word Processors:

- *insert text:* Allows you to insert text anywhere in the document.
- *delete text:* Allows you to erase characters, words, lines, or pages.
- *cut and paste:* Allows you to remove (cut) a section of text from one place in a document and insert (paste) it somewhere else.
- *copy:* Allows you to duplicate a section of text.
- *page size and margins*: Allows you to define various page sizes and margins, and the word processor will automatically readjust the text so that it fits.
- *search and replace:* Allows you to direct the word processor to search for a particular word or phrase. You can also direct the word processor to replace one group of characters with another everywhere that the first group appears.
- *word wrap:* Automatically moves to the next line when you have filled one line with text, and it will readjust text if you change the margins.
- *print:* Allows you to send a document to a printer to get hard copy.
- *File management:* Provides file management capabilities that allow you to create, delete, move, and search for files.
- *font specifications:* Allows you to change fonts within a document. For example, you can specify bold, italics, and underlining. Most word processors also let you change the font size and even the typeface.
- *windows:* Allows you to edit two or more documents at the same time. Each document appears in a separate window. This is particularly valuable when working on a large project that consists of several different files.
- *spell checking:* Identifies words that don't appear in a standard dictionary.

Full-Featured Word Processors:

- *grammar checking*: Identifies sentences, paragraphs, and punctuation that doesn't appear to meet commonly recognized rules of grammar.

- *footnotes and cross-references*: Automates the numbering and placement of footnotes and enables you to easily cross-reference other sections of the document.

- *automated lists*: Automatically creates bulleted or numbered lists, including multi-level outlines.

- *graphics*: Allows you to embed illustrations, graphs, and possibly even videos into a document. Some word processors let you create the illustrations within the word processor; others let you insert an illustration produced by a different program.

- *headers, footers, and page numbering*: Allows you to specify customized headers and footers that the word processor will put at the top and bottom of every page. The word processor automatically keeps track of page numbers so that the correct number appears on each page.

- *layout*: Allows you to specify different margins within a single document and to specify various methods for indenting paragraphs.

- *macros*: Enables users to define and run macros, a character or word that represents a series of keystrokes. The keystrokes can represent text or commands. The ability to define macros allows you to save yourself a lot of time by replacing common combinations of keystrokes.

- *merge*: Allows you to merge text from one file into another file. This is particularly useful for generating many files that have the same format but different data. Generating mailing labels is the classic example of using merges.

- *tables of contents and indexes*: Allows you to automatically create a table of contents and index based on special codes that you insert in the document.

- *thesaurus*: Allows you to search for synonyms without leaving the word processor.

- *collaboration*: Allows users to track changes to the document when more than one person is editing. Some cloud-based word processors also allow multiple users to edit the same document at the same time.

- *Internet features*: Allows users to embed Web links into their documents and format their documents for the Web. Some also link to Web services that can help users create their documents.

- *translation and speech*: As artificial intelligence capabilities become more commonplace, some word processors have gained the ability to read text aloud, to accept voice commands, and to translate text from one language to another.

Change Control Systems (Change Management tool)

- Change control system is any system that has been implemented that serves the essential purpose of assuring that the process of making changes is not done arbitrarily and without thought but rather is carefully considered and ultimately signed off on by a responsible party.
- The change control system typically encompasses not only the specific elements involved in ultimately making the decision to approve, reject, or postpone any changes, but also all of the processes that should be utilized in the process.
- For example, some of the elements involved in a properly functioning change control process can include, but are not necessarily limited to, previously documented Change Control Policy, an established Change Control Board, an established core of Change Management Tools, and in some cases a Quality Assurance team and an Asset Management Team.

- The Change Control System, in addition to providing as established policy for the routine process of making changes, should also have in place provisions for emergency changes which may arise.

Steps in a Structured Change Control System:

1. Initial request
2. Pre-approvals
3. Change execution
4. Follow-up approvals
5. Implementation
6. Storing change history for audit

- A change control solution should be able to handle every type of change request, including batch, documentation, EH&S, equipment, hardware/network, materials, packaging/labels, process, regulations, specifications, supplier, validation.

- The change control system should also enhance transparency, improve regulatory reporting, and optimize management of change requests throughout the supply chain.

- Finally, change control systems should provide a checks and balances system to ensure the right changes are being made.
  At each step in a change control process, there may be personnel from different departments that need to get involved.

- The change may require regulatory filing and approvals and quality processes that must be addressed before the change can take place.

- This is where a change control solution from Sparta Systems can help project managers organize all tasks needed to complete the change as well as provide transparency needed across all of the departments that are affected by the change.

Interactive Debugging System (Debugging tool)

- Set of unit test functions that can be specified by the programmer.
- It deals with execution sequencing, which is the observation and control of the flow of program execution.
- The programmer may define breakpoints which cause execution to be suspended when a specified point in the program is reached.
- After execution is suspended, other debugging commands can be used to analyse the progress of the program and to diagnose errors detected. Execution of the program can then be resumed.
- The programmer can define conditional expressions that are continually evaluated during the debugging session. Program execution is suspended when any of these conditions becomes true. Given a good graphical representation of program progress, it may ever be useful to run the program at various speeds called gaits.
- Tracing can be used to track the flow of execution logic and data modifications. The control flow can be traced at different levels of detail procedure, branch, individual instruction and so on.
- Traceback can show the path by which the current statement was reached.
- It can also show which statements have modified a given variable or parameters.
- Debugging system to have good program display capabilities. It must be possible to display the program being debugged, complete with statement numbers.

- Single debugging tool that is applicable to such multilingual situations.
- Debugger commands that initiates actions and collect data about a program's execution should be common across languages.
- The debugger must be able to determine the language in which the program is written and set its context accordingly.
- The debugger should be able to switch its context when a program written in one language calls a program written in a different language.
- Debugging system to be able to deal with optimized code. Application code used in production environments is usually optimized. Many optimizations involve the rearrangement of segments of code in the program.

Program Restructuring System (Re-engineering tool)

- Maintenance tends to degrade the structure of software, ultimately making maintenance more costly.
- At times, then, it is worthwhile to manipulate the structure of a system to make changes easier.
- However, it is shown that manual restructuring is an error-prone and expensive activity.
- By separating structural manipulations from other maintenance activities, the semantics of a system can be held constant by a tool, assuring that no errors are introduced by restructuring.
- To allow the maintenance team to focus on the aspects of restructuring and maintenance requiring human judgment, a transformation-based tool can be provided--based on a model that exploits preserving data flow-dependence and control flow-dependence--to automate the repetitive, error-prone, and computationally demanding aspects of restructuring.

- A set of automatable transformations is introduced; their impact on structure is described, and their usefulness is demonstrated in examples.

- A model to aid building meaning-preserving restructuring transformations is described, and its realization in a functioning prototype tool for restructuring scheme programs is discussed.

- Program restructuring is a key method for improving the quality of ill-structured programs, thereby increasing the understandability and reducing the maintenance cost. It is a challenging task and a great deal of research is still ongoing.

- The objective of program restructuring is to improve the structure or internal strength of a function.

- The program restructuring approach proposed in this paper is supported by a set of tools. The approach is based on clustering analysis, with cohesion as the main criterion.

- The existing structure of a program with quantitative measure is shown in a tree after a clustering analysis has been performed.

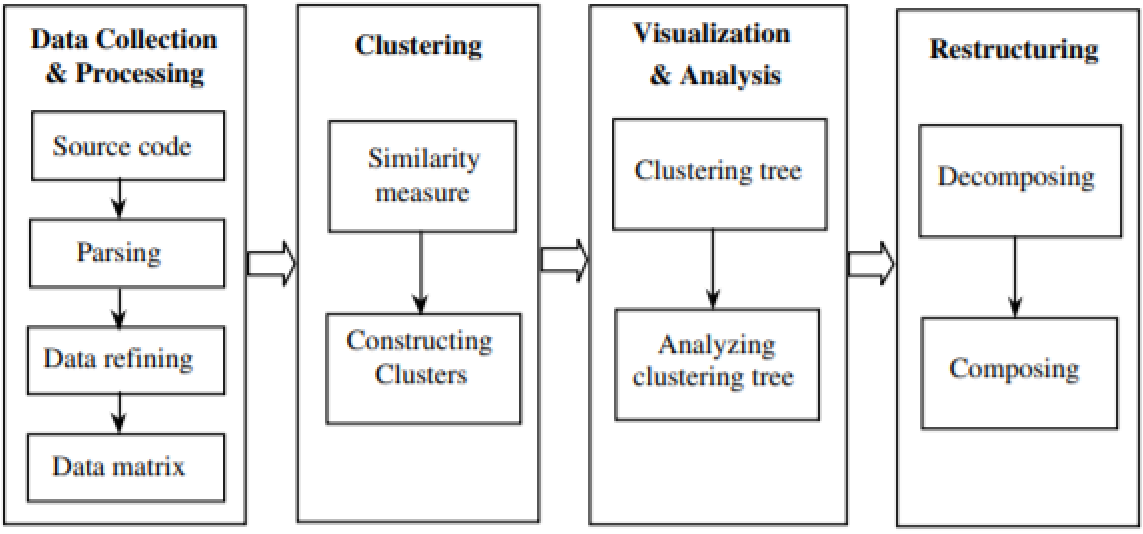
- The approach provides information about the existing structure of a function, the quantitative structure measure, and a heuristic guideline for improving the existing code.

- It can be used to help software designers make a decision – why and how to restructure an existing program.

- The approach also deals with some fundamental and challenging issues of clustering for functions.

- Those issues include: definition of entities and attributes inside of a function, an algorithm to calculate resemblance coefficients, and selection of the best clustering method.

- Currently, the study is conducted for C programs; however, the technique can be applied to other languages as well.



- An approach to program restructuring inside of a function is based on clustering techniques with cohesion as the major concern.

- Clustering has been widely used to group related entities together. The approach focuses on automated support for identifying ill-structured or low-cohesive functions and providing heuristic advice in both the development and evolution phases.

- A new similarity measure is defined and studied intensively specifically from the function perspective. A comparative study on three different hierarchical agglomerative clustering algorithms is also conducted.

- The best algorithm is applied to restructuring of functions of a real industrial system. The empirical observations show that the heuristic advice provided by the approach can help software designers make better decision of why and how to restructure a program.

- Specific source code level software metrics are presented to demonstrate the value of the approach.

*****