International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 21, 2017 ISSN: 2348-6600
Reference ID: IJCS-256 PAGE NO: 1635-1640
Alagappa University, Karaikudi, India 15th -16th February 2017
IT Skills Show & International Conference on Advancements in Computing Resources (SSICACR-2017)
http://aisdau.in/ssicacr ssicacr2017@gmail.com

DEVELOPMENT OF A COMPUTER AIDED SOFTWARE ENGINEERING-A LITERATURE SURVEY

Dr.M.P.Indra Gandhi #1,
Department of Computer Science
Mother Teresa Women's University
Kodaikanal-624101.
1jarishsmpig@gmail.com

B.Thilagavathi*2
Department of Computer Science
Mother Teresa Women's University
Kodaikanal-624101.
2thilabala7@gmail.com

Abstract - Software Engineering is an area with a wide range of concepts and knowledge. Such diversity of topics requires the application of different teaching and learning techniques for an effective education. Computer-Aided Software Engineering(CASE). CASE tools are used to develop software that is high-quality, defect-free, and maintainable. CASE software is often associated with methods for the development of information systems together with automated tools that can be used in the software development process. Software development environments are the computer-based tools that are intended to assist the software development process. Tools allow repetitive, well-defined actions to be automated, thus reducing the cognitive load on the software engineer. The engineer is then free to concentrate on the creative aspects of the process. Tools are often designed to support particular methods, reducing any administrative load associated with applying the method manually. This tool will make the process automatically to finish the work within the time duration.

Keywords: CASE, Development Process, Application, Cognitive load, Designing.

I INTRODUCTION

Software Engineering is a field of engineering, for designing and developing a program using computers or other electronic devices. A software engineer, or programmer, writes a program using selectedsoftwareand compiles the program using methods that make it better quality. Software engineering, on the other hand, focuses on technical and managerial leadership for large and complex systems. Software engineering is the branch of systems engineering concerned with the

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 21, 2017 ISSN: 2348-6600
Reference ID: IJCS-256 PAGE NO: 1635-1640
Alagappa University, Karaikudi, India 15th -16th February 2017
IT Skills Show & International Conference on Advancements in Computing Resources (SSICACR-2017)
http://aisdau.in/ssicacr ssicacr2017@gmail.com

development of large.. It is also concerned with the processes, methods and tools for the development of software intensive systems in an economic and timely manner. Computer science focuses on foundations of computer including algorithms, programming languages, theories of computing, artificial intelligence and hardware design.

II PROPOSED WORK

ADefinitions of Software Engineering

Research, design, develop, and test operating systems-level software, complier and network distribution software for medical, industrial, military, communications, aerospace, business, scientific, and general computing applications.[1]

The systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software. [4]

Software engineering is often done as part of a team.

- Steps in creating software
- Tools used in creating software

B. Steps in creating software

Software engineering can broadly be split into the following steps:

- Requirements say what the software should do.
- Software design is usually done on paper. It says that the different parts of the software are, and how they link to each other.
- After the design phase is done, each component of the software is coded. Code is what tells the computer exactly what to do at each step.
- Testing is done to see if the components meet the requirements and that the system as a whole meets the requirements.
- Part or all of this process can repeat if bugs are found or new requirements are needed.

C. Tools used in creating software

[9]Software engineers use many tools and practices for making software. Some of the most common graphical representations are:

- Flowchart
- UML Diagram
- Debugging
- Compiler
- Text Editor

1. Flowchart:

Flowchart is a sequence of movements or actions of people or things to involve in a complex system or activity.

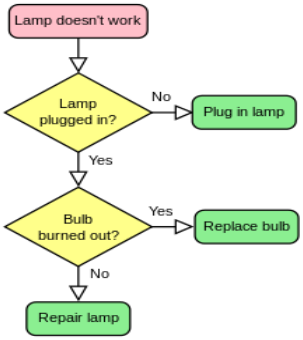


Fig 1: A simple flowchart

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 21, 2017 ISSN: 2348-6600
Reference ID: IJCS-256 PAGE NO: 1635-1640
Alagappa University, Karaikudi, India 15th -16th February 2017
IT Skills Show & International Conference on Advancements in Computing Resources (SSICACR-2017)
http://aisdau.in/ssicacr ssicacr2017@gmail.com

A flow chart is a type of diagram, made of boxes and arrows. It can be used to show:

- An algorithm, a step-by-step list of directions that are needed to be followed to solve a problem
- A process, a series of stages in time where the last stage is the product, result or goal.
- The planned stages of a project.

2.UML Diagram:

UML stands for Unified Modeling Language which is used in object oriented software engineering. Although typically used in software engineering it is a rich language that can be used to model an application structures, behavior and even business processes.

They can be divided into two main categories;

- **Structure diagram** shows the things in a system being modeled. In more technical term, they show different objects in a system.
- **Behavioral diagrams** shows what should happen in a system. They describe how the objects interact with each other to create a functioning system.

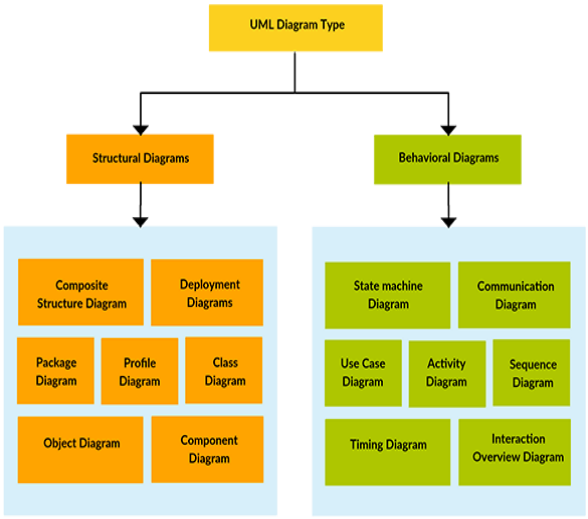


Fig 2: UML Diagram Type

3. Debugging:

Debugging is the process of finding errors in a computer program. If we think that the program may have an error, we must edit the program to do the bugs.

4. Compiler:

A compiler is a computer program that translates instruction text into different language. The first language is called the source language, which means the instruction text is called source code. The second language is called the target,it gives instruction to the computer to follow. In that case, the instructions become machine code.[10]

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 21, 2017 ISSN: 2348-6600
Reference ID: IJCS-256 PAGE NO: 1635-1640
Alagappa University, Karaikudi, India 15th -16th February 2017
IT Skills Show & International Conference on Advancements in Computing Resources (SSICACR-2017)
http://aisdau.in/ssicacr ssicacr2017@gmail.com

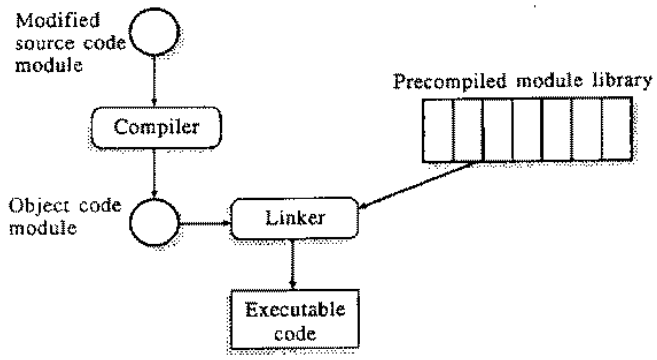


Fig 3: Compiler Process

5. TextEditor:
A text editor is a program that is run on a computer that can be creating change the text. The text can be saved into a file that is called a text file. Text can be used to do lot of things. Some text editors can edit rich text. Rich text allows the person to edit the text to have bold text, italic text, and other things.

D. CASE Tools
Computer Aided Software Engineering technologies are tools that provide automated assistance for software development. The goal of introducing CASE tools is the reduction of the time and cost software development and the enhancement of the quality of the systems developed. The interest in CASE tools and environments is based on exceptions about increasing, improving the product quality, facilitating maintenance and making software engineers task less odious and more enjoyable.[5]

E. Role of CASE Tools

CASE tools play a major role in the following activities:

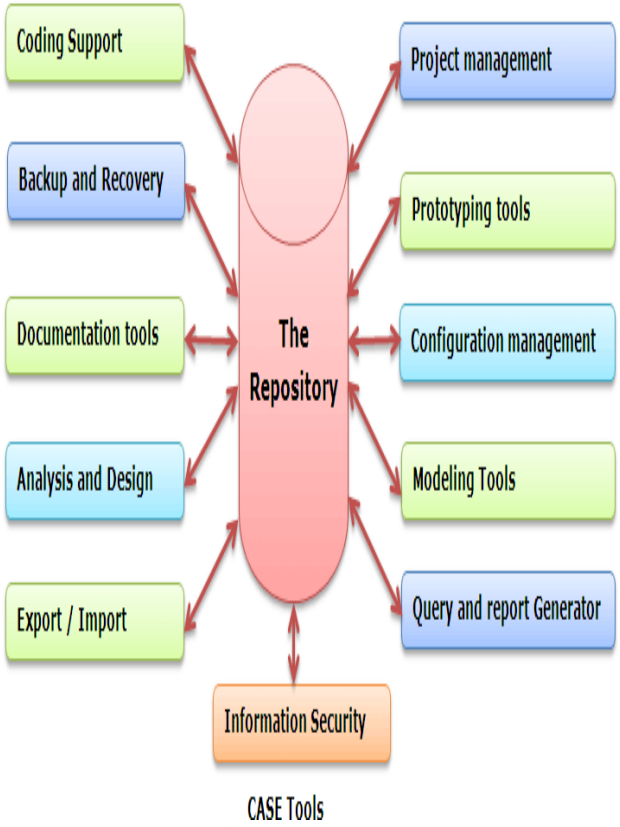


Fig 4: CASE Tools

1. Needs of CASE Tools:
The software development process is expensive and as the projects become more complex in nature, the project implementations become more demanding and expensive. CASE tools provide the integrated homogenous environment for the development of complex projects. These tools provide computerized setting to software

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 21, 2017 ISSN: 2348-6600
Reference ID: IJCS-256 PAGE NO: 1635-1640
Alagappa University, Karaikudi, India 15th -16th February 2017
IT Skills Show & International Conference on Advancements in Computing Resources (SSICACR-2017)
http://aisdau.in/ssicacr ssicacr2017@gmail.com

developers to analyze a problem and then design its system model.

2. CASE Tools are used for:

Reduce the cost as they automate many repetitive manual tasks. Develop better quality complex projects as they provide greater consistency and coordination. Create good quality documentation.

3. Categories of CASE Tools:

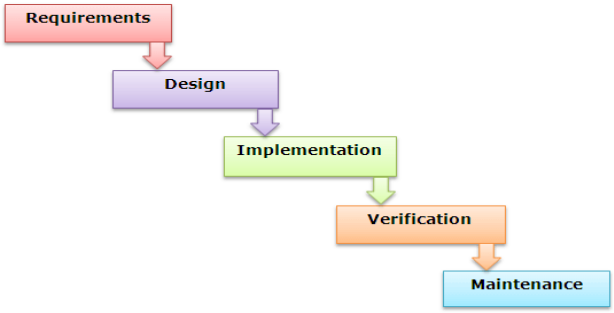
[3]Sometime CASE Tools are classified into following categories due to their activities:

(a) UPPER CASE Tools: They support the analysis and the design phase.

(b) LOWER CASE Tools: They support the coding phase, configuration management, etc...

(c) INTEGREATED CASE Tools: It is known as I-CASE and also support analysis, design and coding phase.

The phase is supported by the LOWER and UPPER CASE Tools are shown through the waterfall model.



4. Automated Diagram Tools:

CASE Tools offer an excellent array of features that support the development and business community through its Automated Diagram Support Feature. The various popular features are listed below:

(i) Checks for syntactic correctness
(ii) Data dictionary support
(iii) Checks for consistency and completeness
(iv) Navigation to linked diagrams
(v) Layering
(vi) Requirements traceability
(vii)Automatic report generation
(ix) System simulation
(x) Performance analysis

5. CASE Tools and its scope:

CASE technology is the automation of step by step methodologies for software and system development. CASE tools are characterized in the form of stage by stage software development life cycle on which they focus. Since different tools are covering different stages that share common information, it is required that they integrate through some central repository system (data dictionary) to have a consistent view of such information. Case Tools are used in many ways in our organizations. Case tools can be broadly classed into these broader areas:

(i) Requirement Analysis Tool
(ii) Structure Analysis Tool
(iii) Software Design Tool
(iv) Code Generation Tool
(v) Test Case Generation Tool
(vi) Document Production Tool
(vii) Reverse Engineering Tool

International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
http://www.ijcsjournal.com Volume 5, Issue 1, No 21, 2017 ISSN: 2348-6600
Reference ID: IJCS-256 PAGE NO: 1635-1640
Alagappa University, Karaikudi, India 15th -16th February 2017
IT Skills Show & International Conference on Advancements in Computing Resources (SSICACR-2017)
http://aisdau.in/ssicacr ssicacr2017@gmail.com

[6]While many organizations still use the SDLC methodology, it is often supplemented with other methods. Many system developers use the CASE tools in various stages of the Software Development Life Cycle. It is mainly usedfor developing the following methodologies:

(i) Life Cycle
(ii) Object-Oriented Approach
(iii) Rapid Applications Development (RAD)
(iv) Prototyping
(v) Joint Applications Development (JAD)

6. Advantages of CASE Tools:

(i) Produce system with a longer effective operational life.
(ii) Produced system with excellent documentation.
(iii) Produce more flexible system.

7. Disadvantages of CASE Tools:

(i) Produce initial system that is more expensive to build and maintain.
(ii) May be difficult to customize.
(ii) May be difficult to use with existing systems. [7]

III CONCLUSION

This paper takes a positive view of current progress and future challenges in software engineering. We believe that the discipline has delivered and is well set to continue to deliver both practical and support to the software developers and the theoretical frameworks. Despite this lag current software engineering practice is being radically reshaped by object-oriented design methods, CASE tools with powerful code generation, testing and analysis environments, development patterns, incremental delivery based life-cycles, component models and document management environments. All of these have been formed through software engineering research.

REFERENCES

[1] Pressman, R.S. Software Engineering : A Practitioners, 4th edition (August 1996), McGraw Hill College Div.
[2] Redwine S.T. & Riddle, W.E. Software Technology Maturation, Proceedings of the 8th International Conference on Software Engineering, 1985, pp 189- 200, IEEE Computer Society.
[3]Sommerville, I. Software Engineering (International Computer Science Series), 5th edition (November 1995) Addison-Wesley Pub Co.
[4]Edward V. Berard. Essays on Object-Oriented Software Engineering.Prentice-Hall, 1993.
[5]Alan W. Brown et al. Principles of CASE Tool Integration.Oxford University Press, 1994.
[6]http://en.wikipedia.org/wiki/Computer_aided_software_engineering
[7]http://www.scirp.org/journal/jsea/
[8]http://ieeexplore.ieee.org/Xplore/login.jsp?url
[9]https://simple.wikipedia.org/wiki/Software_engineering
[10]www.mhhe.com/engcs/compsci/pressman/olc_linkedcontent/casetools.html