Computer Aided Software Engineering (CASE) Tools /lec5

Marrwa alabajee

Contents

1 The taxonomy of CASE-Systems by their function

Software Metrics

What is Software Metrics?

A software metric is a measurable or countable measure of software characteristics. Software engineering metrics can be used for a range of things, such as evaluating software performance, planning work items, estimating productivity.

Software metrics are related to the four functions of management: Planning, Organization, Control, or Improvement.

How can Software Metrics be Used?

Software development metrics can be used by project owners, project managers, developers, and quality assurance teams.

Software Metrics

Categories of Software Metrics

- **Product Metrics** - Size, complexity, performance, and quality level of the product are all factors to consider.

- **Process Metrics** - To optimize software development and maintenance, process metrics can be used. Eliminating the faults rate during development, and the time it takes for a fixed operation to complete, They are used to measure the characteristics of methods, techniques, and tools that are used for developing software.

- **Project Metrics** - The characteristics and execution of a project are described by project metrics. The number of software developers, cost, scheduling, and productivity are all examples of factors to consider.

Metrics and management tools

Software metrics improve a manager's ability to control and coordinate the software engineering process and a practitioner's ability to improve the quality of the software that is produced. Today's metrics or measurement tools focus on process and product characteristics.

Technically oriented tools determine technical metrics that provide greater insight into the quality of design or code.

Metrics and management tools

object oriented Software metrics provides measurements at two levels: class-level and system-level.

Class-level metrics measure the complexity of individual classes contained in the system. These metrics can be very useful for predicting defect rates and estimating cost and schedule of future development projects. for Example class coupling.

System-level metrics deal with a collection of classes that comprise an object-oriented system for example number of class hierarchies .

Metrics and management tools

Management-oriented tools capture project specific metrics such as Size-oriented metrics or Function-oriented metrics.

Size-oriented metrics focus on the size of the software and are usually expressed as kilo lines of code (KLOC).Some examples include:

Errors per KLOC, Defects per KLOC, Cost per KLOC.

Function-oriented metrics focus on how much functionality software offers. But functionality cannot be measured directly. So function-oriented software metrics rely on calculating the function point (FP),it is a unit of measurement that quantifies the business functionality provided by the product.

Errors per FP or Defects per FP.

Documentation tools

software Documentation is any written document that explains how a piece of software works, why it was built, and how it is intended to be used. Depending on the complexity of your software, your documentation can contain information on the general use of the product and in-depth dives into functions and features.

The deliverable documents should be organized graphically and should be able to incorporate text and diagrams from the central repository. This helps in producing up-to-date documentation.

Documentation tools

Document production tools support nearly every aspect of software engineering and represent a substantial opportunity for all software developers.

Most software development organizations spend a substantial amount of time developing documents, and in many cases the documentation process itself is quite inefficient.

It is not unusual for a software development organization to spend as much as 20 or 30 percent of all software development effort on documentation.
For this reason, documentation tools provide an important opportunity to improve productivity.

Documentation tools

Best Software Documentation Tools

Document360, Nuclino, GitHub, MarkdownPad

Nuclino: is a good way to organize information within teams into workspaces. You can use Nuclino to create beautiful software documentation for your employees or your customers. You can bring your content to life with text, images, videos, files, tasks, code blocks, and more.
You can collaborate in real-time so you can see the changes your team members are making as they type, which means there's no risk of version conflicts. You can type @ inside an item to link to another page.

System software tools

System software refers to the low-level software that manages and controls a computer's hardware and provides basic services to higher-level software, example Communication software, .

with System software tools the CASE environment must accommodate high-quality network system software, object management services, distributed component support, electronic mail, and other communication capabilities.

Quality assurance tools

quality define as “a characteristic or attribute of something.” As an attribute of an item, quality refers to measurable characteristics—things we are able to compare to known standards such as length, color, electrical properties.

Nevertheless, measures of a program’s characteristics do exist. These properties include cyclomatic complexity, cohesion, number of function points, lines of code.

A project isn’t successful just because it’s been completed on time and within budget. There’s another project constraint that is critical to success: quality. Stakeholders aren’t going to be happy if the quality of the product or service doesn’t meet their expectations. So, how can project managers control quality? Quality assurance

Quality assurance tools

Quality assurance (QA) is a quality management process that consists of establishing standards, guidelines and procedures to prevent quality issues and maintain the integrity of the product or service throughout its development.

The majority of CASE tools that claim to focus on quality assurance are actually metrics tools that audit source code to determine compliance with language standards.
Other tools extract technical metrics in an effort to project the quality of the software that is being built.

Database management tools

Database management tools

Database management software serves as a foundation for the establishment of a CASE database (repository) that we have called the project database.

Given the emphasis on configuration objects, database management tools for CASE are evolving from relational database management systems to object-oriented database management systems.

Database management tools

RDBMS stands for Relational Database Management System. It is a database management system based on the relational model i.e. the data and relationships are represented by a collection of inter-related tables. It is a DBMS that enables the user to create, update, administer and interact with a relational database. Its like MS SQL Server, Oracle, MySQL, and Microsoft Access.

OODBMS stands for Object-Oriented Database Management System. It is a DBMS where data is represented in the form of objects, as used in object-oriented programming. OODB implements object-oriented concepts such as classes of objects, object identity, encapsulation, and inheritance. An object-oriented database stores complex data as compared to relational database. Some examples of OODBMS are Objectivity/DB.

Software Configuration Management

Software Configuration Management (SCM) is extremely important from the view of deployment of software applications.

SCM controls deployment of new software versions. Software configuration management can be integrated with an automated solution that manages distributed deployment.

This helps companies to bring out new releases much more efficiently and effectively. It also reduces cost, risk and accelerates time.

Software Configuration Management

A current IT department of an organization has complex applications to manage .
These applications may be deployed on many locations and are critical systems.
Thus, these systems must be maintained with very high efficiency and low cost and time.

The problem for IT organizations has increased tremendously as some of the organizations may need to rebuild and redeploy new software versions a week over multi-platform global networks.

Software Configuration Management

In such a situation, if rebuilding of application versions is built or deployed manually using a spreadsheet, then it requires copying of rebuilt software to many software locations, which will be very time consuming and error prone.

What about an approach where newer software versions are automatically built and deployed into several distributed locations through a centralized control.

Thus, an automatic deployment tool will be of great use under the control of SCM.

Software Configuration Management tools

Software Configuration Management (SCM) tools

Software configuration management lies at the kernel of every CASE environment. Tools can assist in all five major SCM tasks:

Identification.
Version control.
Change control.
Auditing.
Status accounting.

Software Configuration Management tools

Software Configuration Management (SCM) tools

The CASE database provides a mechanism for identifying each configuration item and relating it to other items;

the change control process can be implemented with the aid of specialized tools; easy access to individual configuration items facilitates the auditing process; and CASE communication tools can greatly improve status accounting (reporting information about changes to all who need to know).

Analysis and design tools

at the analysis level: The models contain a representation of data, function, and behavior .

Analysis Tools: These tools help to gather requirements, automatically check for any inconsistency, inaccuracy in the diagrams, data redundancies or erroneous omissions.

at the design level: characterizations of data, architectural, component-level, and interface design.

Design Tools: These tools help software designers to design the block structure of the software, which may further be broken in smaller modules using refinement techniques and provides
detailing of each module and interconnections among modules.

Analysis and design tools

Several diagramming techniques are used for structured analysis and structured design, A CASE tool should support one or more of the structured analysis and design technique.

The CASE tool should support effortlessly drawing analysis and design diagrams. The CASE tool should support drawing fairly complex diagrams and preferably through a hierarchy of levels.

It should provide easy navigation through different levels and through design and analysis.

The tool must support completeness and consistency checking across the design and analysis and through all levels of analysis hierarchy.

Prototyping and Simulation (PRO/SIM) tools

We have already seen that prototyping is useful to understand the requirements of complex software products, to demonstrate a concept, to market new ideas, and so on. The prototyping CASE tool's requirements are as follows:

Define user interaction.
Define the system control flow.
Store and retrieve data required by the system.
Incorporate some processing logic

Prototyping and Simulation (PRO/SIM) tools

There are several stand alone prototyping tools. But a tool that integrates with the data dictionary can make use of the entries in the data dictionary, help in populating the data dictionary and ensure the consistency between the design data and the prototype.

A good prototyping tool should support the following features:

- Since one of the main uses of a prototyping CASE tool is graphical user interface (GUI) development, a prototyping CASE tool should support the user to create a GUI using a graphics editor. The user should be allowed to define all data entry forms and menus.

- It should integrate with the data dictionary of a CASE environment.

Prototyping and Simulation (PRO/SIM) tools

- If possible, it should be able to integrate with external user defined modules written in C or some popular high level programming languages.

- The run time system of prototype should support mock up run of the actual system and management of the input and output data.

Example is Figma , Invision , Proto.io
Figma is a collaborative user interface design tool that lets users work together to create vibrant and interactive prototypes. Since its release in 2016, Figma has become a popular tool both in the web design industry and in online communities. Users can collaborate and share templates and designs with millions of users across the globe.

THANKS