CHRISTIAN EMINENT COLLEGE, INDORE

(Academy of Management, Professional Education and Research)

An Autonomous Institution Accredited with 'A' Grade by NAAC

Department of Computer Science & Electronics

Subject: Software Engineering

Class: M.Sc.(Comp.Sc.) II Semester

Prepared By: Prof. N. Jaiswal

Software Engineering Problem

Software engineering is the systematic approach to the development, operation, maintenance, and retirement of software. Following are the fundamental problems that software engineering faces.

The Problem of scale: A fundamental problem of software engineering is the problem of scale; development of a very large system requires a very different set of methods compared to developing a small system. In other words, the methods that are used for developing small systems generally do not scale up to large systems. A different set of methods has to be used for developing large software. Any large project involves the use of technology and project management.

For software projects, by technology we mean the methods, procedures, and tools that are used. In small projects, informal methods for development and management can be used. However, for large projects, both have to be much more formal.

While dealing with a small software project, the technology requirement is low and the project management requirement is also low. However, when the scale changes to large systems, to solve such problems properly, it is essential that we move in both directions-the methods used for development need to be more formal, and the project management for the development project also needs to be more formal.

Cost, schedule and quality: The cost of developing a system is the cost of the resources used for the system, which, in the case of software, are the manpower, hardware, software, and the other support resources. Generally, the manpower component is predominant, as software development is largely labor-intensive and the cost of the computing systems is now quite low.

Hence, the cost of software project is measured in terms of person-months, i.e. the cost is considered to be the total number of person-months spent in the project. Schedule is an important factor in many projects. Business trends are dictating that the time to market of a product should be reduced; that is, the cycle time from concept to delivery should be small. Any business with such a requirement will also require that the cycle time for building software needed by the business be small. One of the major factors driving any production discipline is quality. We can view quality of a software product as having three dimensions:

- Product Operation
- Product Transition
- Product Revision

The Problem of consistency: Though high quality, low cost and small cycle time are the primary objectives of any project, for an organization there is another goal: consistency. An organization involved in software development does not just want low cost and high quality for a project, but it wants these consistently.

Software Characteristics

Software is often the single largest cost item in a computer-based application. Though software is a product, it is different from other physical products.

- Software costs are concentrated in engineering (analysis and design) and not in production.
- Cost of software is not dependent on volume of production.
- Software does not wear out (in the physical sense).
- Software has no replacement (spare) parts.
- Software maintenance is a difficult problem and is very different from hardware (physical product) maintenance.
- Most softwares are custom-built.
- Many legal issues are involved (e.g. intellectual property rights, liability).

As stated earlier, commercial software are generally different from computer programs written for academic or research purposes. The characteristics of real-world software are:

- It is generally developed by software firms for their clients under formal business contracts.
- Like any product, software is designed based on some software specification.
- It is usually developed in teams and not by individuals.
- It generally includes clear and detailed documentation (i.e. design manual and users' manual).
- It is meant for users who need not have good knowledge of computers.

- It generally has user-friendly interfaces so that users having limited expertise in computers can operate the system.
- Generally. Software is designed to be run on different platforms.
- It has a lifetime in years, after which it becomes obsolete. Hence, software is designed keeping its intended life and cost in view.
- It generally requires some modification from time to time to accommodate changes taking place in the organization and the environment.
- It is developed under formalized product reviews (quality assurance) and formalized testing procedures.
- The cost of software failure may amount to an economic catastrophe. Hence, software is designed for utmost reliability.
- A computer system is prone to misuse or sabotage by persons having ulterior motives from within or from outside the organization. Hence, software is designed to be tamper-proof and protected from misuse or damage.
- Ethical issues (like protecting privacy) are also taken into consideration in designing software.

Development of software is also different in some way from building a physical object such as a bridge, house or factory. Problems solved with software solutions are complex. Finding solutions requires some ingenuity and good planning. The progress made in a software project is not visible to the eyes as in the case of other projects. Due to the uncertainty and uniqueness of each project, it is difficult to estimate project duration and cost accurately. During the course of a project, there can be changes in the requirements and environment. Since software is not a physical product and its development is not visible to the eyes, customers often feel no compunction to put pressure for incorporating some changes at the last minute. There is no universal method for software development that can be followed in all situations. The method used in a well-managed software project may not be suitable for a similar project when the project size is different. However, keeping in view the above characteristics of software, some systematic approaches, engineering principles, tools and techniques have been devised to produce quality software within time and resource constraints.

Software Applications

Software applications as per different task can be categorized as follows:

Word Processing software - To create worksheets, type letters, type papers, etc.

MS Word, WordPerfect, MS Works, AppleWorks

Desktop Publishing software - To make signs, banners, greeting cards, illustrative worksheets, newsletters, etc.

Adobe PageMaker, MS Word, MS Publisher, AppleWorks, MS Works, Quark Express

Spreadsheet software – To compute number-intensive problems such as budgeting, forecasting, etc. A spreadsheet will plot nice graphs very easily.

MS Excel, Quattro Pro, Lotus 1-2-3, MS Works, AppleWorks

Database software - To store data such as address, membership and other text information. A database can be used to easily sort and organize records.

MS Access, Filemaker Pro, AppleWorks, MS Works

Presentation software - To create multimedia stacks of cards/screens that can effectively present a lesson or a sales pitch. The user often clicks on buttons to advance to the next screen in a sequence.

MS PowerPoint, AppleWorks (slideshows), HyperStudio, Flash, Director, HyperCard, Digital Chisel, SuperCard, Corel Envoy,...

Internet Browsers - To surf the Web. Read email and Create Web pages.

Netscape Navigator (or Netscape Communicator), MS Internet Explorer, AOL Browser

Email programs - These programs send and receive email.

Netscape Messenger (part of Netscape Communicator), MS Outlook Express, MS Outlook, Eudora, AOL browser (has email built in)

Graphics Programs (pixel-based) - This software allows one to touch up photographs and create graphics from scratch.

Adobe Photoshop, Paint Shop Pro, AppleWorks, MS Works, MS Paint (comes free on Windows PC's), Painter

Graphics Programs (vector-based) - This software creates graphics that are similar to illustrations or cartoon drawings.

Adobe Illustrator, Corel Draw, AppleWorks, MS Works, MS Word

Communications Software - This software allows two computers with modems to communicate through audio, video, and/or chat-based means.

MS NetMeeting, AOL Instant Messenger, IRC, ICQ, CU-SeeMe

The Project Life Cycle

The project life cycle consists of four phases, initiation, planning, execution (including monitoring and controlling) and evaluation. The MPMM Project Management Methodology is an excellent resource for this part of the Unit. The Initiation phase begins by defining the scope, purpose, objectives, resources, deliverables, timescales and structure of the project. The next step is to develop a Business Case, including several possible solutions and a cost/benefit analysis for each. A Feasibility Study should then be carried out to ensure that the chosen solution is feasible and has an acceptable level of risk. The next step is to define the Terms of Reference, followed by the appointment of the project team. The final step is to carry out Phase Review before seeking approval to proceed. The first step of the Planning phase is the creation of a detailed Project Plan which the project manager will refer throughout the project to monitor and control time, cost and quality. The project manager will then create the following plans:

- Resource Plan: to identify the staffing, equipment and materials needed
- Financial Plan: to quantify the financial expenditure required
- Quality Plan: to set quality targets and specify Quality Control methods
- Risk Plan: to identify risks and plan actions needed to minimize them
- Acceptance Plan: to specify criteria for accepting deliverables

Finally, a Phase Review is carried out to assess the deliverables produced to date and approve the start of the Project Execution phase. During the Project Execution phase the project team produces the deliverables while the project manager monitors and controls the project delivery by undertaking:

- Time Management: tracking and recording time spent on tasks against the Project Plan
- Cost Management: identifying and recording costs against the project budget
- Quality Management: reviewing the quality of the deliverables and management processes
- Change Management: reviewing and implementing requests for changes to the project
- Risk Management: assessing the level of project risk and taking action to minimize it
- Issue Management: identifying and resolving project issues
- Acceptance Management: identifying the completion of deliverables and gaining the customers acceptance
- Communications Management: keeping stakeholders informed of project progress, risks and issues

Once the customer has accepted the deliverables and a Phase Review has been carried out to determine whether the project objectives have been achieved, the project is ready for Closure. A Project Closure Report should list all of the actions required. When this has been approved, the listed actions are completed to release project resources, hand over deliverables, and inform all stakeholders that the project is now closed. Shortly after the project has been closed, an Evaluation (also known as a Post-Implementation Review) should be carried out to determine the project's overall success and find out whether the benefits stated in the original Business Case were actually realized. Any lessons learned should be documented for future projects.