Software Project Management – a practitioner's Perspective

Murali Chemuturi

Dt. 30th October 2008

32-78/3 Sainik Nagar Rama Krishna Puram, Secunderabad - 500056
+91-40-2722 0771 - www.chemuturi.com - murali@chemuturi.com

Prologue

Before we talk of Software Project Management, I believe that we need to understand the terms “Project” and “Management”. Once we understand these two terms the rest is applying these two to the field of software development. In the subsequent sections I will strive to define and explain these two terms before I proceed to the subject of Software Project Management.

Software Projects are initiated and executed at two places – one – within the same organization that is going to use the end product and – two – in an organization that has specialized in developing software that would be used in other organizations. I wish to name them as “Internal Projects” and “External Projects”. Internal Projects are those whose end product is used within the same organization and External Projects are those whose end product is used by another organization. I will also expand on these terms in the subsequent sections.

I am focusing this book on the management of software projects that are tightly planned, monitored, controlled and executed. These could be either Internal or External Projects.

Table of Contents

CHAPTER 1 – PROJECT BASICS 4

1.1 INTRODUCTION 4

Chapter 1 – Project Basics

1.1 Introduction

Human Endeavor was originally carried out in agricultural farms. Then there was no organization – only the owner. He had employed laborers and Overseers.

Industrial Revolution moved some of the endeavor into factories. Factory signified production of goods for consumption. Rules were promulgated for various aspects of running factories. The art and science of managing production was labeled “Production Management”.

The Production Management has divided organizations, depending on the manner in which goods are produced into –

1. Mass Production organizations – produce continuously the same products
2. Batch Production Organizations – produce goods in batches – each batch is similar but not identical
3. Job Order Production organizations – goods are tailor-made – that is they are produced only when an order is received
4. Flow Process Production organizations – industries like chemicals, pharmaceuticals, electricity generation, fertilizer production etc are categorized under this head

Mass Production, Batch Production and Flow Process systems are also called “Made to Stores” – that is goods are produced and stored in the warehouses for distribution. The significant feature here is that the “rate of production” exceeded the “rate of consumption / demand” for the product.

Job Order Production is also referred to as “Made to Order” – that is something is produced only after an order is received.

This left out organizations that construct buildings, highways and other infrastructure facilities. These organizations are certainly not production organizations even though they are creating wealth and employing people. These were classified as Projects. Some knowledge was gathered and released under the titles of Project Management.

Job Order Production system organizations latched on to this concept and they became Project-based production systems.

At that time those providing services were either under Government or individuals. As the demand for services grew, they employed more and more people. Also, more and more services came out of Governmental fold into private hands. Service Organizations also became larger and comparable to production organizations in terms of number of people employed and revenues generated.

So much so, presently the organizations are basically of two types – Production organizations and Service Organizations.

The art and science of managing these organizations metamorphosed from Production Management to Operations Management.

Similarly by the nature of operations – organizations are classified into –

1. Continuous Operations and
2. Project Operations

Organizations who have fixed facilities and carry out similar operations day after day continuously and produce to warehouses are grouped under Continuous Operations organizations.

Organizations who have fixed but flexible facilities carry out dissimilar operations from day to day and produce against a customer order are grouped under Project Operations Organizations.

More and more organizations are moving towards Project Operations owing the market forces, which are putting emphasis on individual preferences than on reducing costs.

Gone are the days of the famous saying of Henry Ford Sr. “You can have the car of any color as long as it is black”.

Having put the Project Operations in its historical perspective, let us contrast the continuous operations with project operations.

Table 1.1 Comparison of Continuous Operations with Project Operations

Item No.	Aspect	Continuous Operations	Project Operations
1	Product Design	Designed once – updated as needed – dictated by market forces	Designed against every order received
2	Trigger for commencement	Marketing asks for it	A customer’s order triggers commencement
3	Planning	Periodic – annual, quarterly, monthly, weekly etc	Order-wise as well as periodic
4	Workstation design	Low cost – designed to produce one type of components	High cost – versatile workstations to produce a wide variety of components
5	Workmen education levels	Low – need to understand instruction and can be easily trained – flatter training curve adequate	High – need to be able to interpret drawings/instructions – may need longer training and steeper learning curve
6	Products	Batches of identical products	Products are similar but never identical

The Operations Management has divided organizations, depending on the type of operations

What are the unique and specific attributes of software development projects?

a. **The output is not physical** – in the sense that there are no physical components that are delivered – everything is inside the computer
b. **Process inspection doesn’t facilitate progress assessment** – we need to wait till the program is finished and tested to get a feel of progress. In a manufacturing organization one can see semi-finished goods and the proof of working is in the noise made. In a software development organization, visual inspection is not enough to ensure that people are performing – one needs to walk-thru the code being developed to ensure that the person is working.

c. While there is significant progress in the software engineering tools, they are no where near the precision of engineering drawings to get the predictability that we get in other engineering disciplines
d. The professional associations in software development still are far from other professional associations in the matter of defining standards or bringing discipline in the process of developing software. True IEEE (Institute of Electrical and Electronic Engineers) is defining a number of standards but these standards are far from the other engineering standards in terms of level of granularity.
e. While there is significant improvement in software development methodology, it is still largely dependent on human beings for productivity and quality. While tools are available to help development or testing they still need to go far to meet the standard of tools used in fabrication / inspection / testing in other engineering disciplines. In other engineering disciplines tools are available that shift the onus for productivity / quality from human beings to tools – true – still the human being can screw-up both quality and productivity, but an average skilled person can achieve higher productivity / quality that can come only from a super-skilled person – this is still a far cry in software development arena – an average programmer cannot achieve higher standard of productivity / quality with the assistance of tools – as yet.

Therefore, the rigor of planning becomes all the more important in software development than in other engineering projects. In other engineering projects, a simple schedule based on PERT/CPM would suffice, where as, in software development projects, increased rigor is required and more plans are required. The plans that are commonly required are described in subsequent sections.

Software project – here the focus is on software development projects. That is –

a. The project has a definite beginning and a definite end
b. The project deliverable is software and related artifacts
c. The activities that may be included under this head are user & software requirements, software design, software construction, software testing, acceptance testing, delivery of software, deployment and hand-over.
d. The activities of project selection / acquisition and post–handover activities are not included under this head.

Introduction
Nobody plans to fail – they just fail to plan – Anonymous

If I were given six hours to fell a tree, I would spend the first four hours sharpening the axe – Abraham Lincoln

All articles and books on achieving success (in any endeavor) start with the necessity to plan and - to plan well. Success may be possible without planning in some cases, but Planning reduces the risk of failure and increases the chances of success in all cases. Better still, coupled with control, planning brings predictability to the probable ending of the endeavor.

While so, the question most often asked is - should planning be on paper?

Not necessarily – for small short duration endeavor, mental planning can be adequate. In fact, none of us fail to plan, we do plan – however, we may not put it on the paper, all the time.

By putting the plan on paper, the advantages are -

a. We can **get it reviewed** by someone knowledgeable or review it ourselves and see if we missed any important aspect and thereby improve the plan.
b. A documented plan acts as **point of reference** for all the persons concerned / involved in the endeavor
c. Facilitates **control and performance evaluation** during execution of the endeavor as well as validate the efficacy of the norms by contrasting the plan with the actual values obtained in execution

The granularity (required detail) of planning depends on –

a. The **duration** of the endeavor
b. The **number of resources** employed
c. The **complexity** involved
d. The **relationship** between the above three aspects

Now consider the sequel to the above statements –

a. Longer the duration, greater the necessity for increased rigor
b. If the duration available to complete the endeavor were adequate, the rigor of planning needed would be less. However, if the duration were less than adequate, then the rigor would increase.
c. The rigor of planning is proportional to the number of resources employed. Higher the number of resources employed, the more is the rigor required in planning.
d. If the complexity is higher – limitations are imposed or the expertise of the domain in which the endeavor is being attempted is lacking – then the rigor of planning would need to be higher

Consequently, the rigor of planning for execution of software projects also depends on the above three aspects, namely duration, number of resources employed and the complexity.

Before we proceed further, let us define Planning.

Definition of Planning

Planning is defined as the intelligent anticipation of resources required to perform a predefined endeavor successfully at a future date in a defined environment.

The key terms are

a. **Anticipation** - this indicates that the Planning precedes performance as well as using the best guess – anticipation is basically estimation of resources
b. **Resources** – the 4 M's – Men (persons), Materials, Methods, and Machines (equipment) plus the time (duration).
c. **At a future date** – the dates are decided or are to be decided during the course of planning
d. **In a defined environment** – the environment where the work is going to be performed is known and is defined or is to be defined during the planning exercise. Any variation in the environment would have an effect on the plan. Environment refers to the working conditions including work & workstation design, processes and methods of management, prevailing morale at the workplace etc.
e. **Predefined endeavor** – the endeavor is defined – the scope of work is known

Now we can move forward.

Planning for software development projects is also planning "per se" and is tailored to suit the specific attributes of software development.

Types of plans prepared by the project management

There is a misunderstanding among software development fraternity that a schedule based on PERT/CPM constitutes software project planning in its entirety. It is not so. Software project Planning needs to go beyond PERT/CPM based schedule. The following are the typical plans prepared for a large software development project

PMP – Project Management Plan – in other engineering projects this is covered in the operating procedures and policies of the organization (or the production facility, which more or less remains consistent) – it is understood how a project is managed. In software development projects, as the development environment changes in every project, we need to plan how to manage the project and we record the following in this plan -

- Project info
- The software estimate (software size, effort, cost and schedule
- The milestones, delivery schedules
- Acceptance Criteria for deliveries effected
- Human resources required along with the dates of their requirements
- Management methods – work allocation, information and code management, quality control, communication management etc
- Tools used in the project
- **IEEE standard 1058** gives various aspects of preparing PMP as well as the suggested template.

CMP – Configuration Management Plan – we record the following info in this plan –

- Naming conventions to be followed for naming project artifacts including documents, code units, variables and constants used in programming, table names, field names and so on
- Procedure for change management
- Organization of project information management so that it would be easy for project team when a reference is needed
- Reference to organizational standards and processes for use in the project

- Code and code library organization along with check-in and check-out criteria and authorizations and procedure for state-change (from development to review / testing, to integration and delivery etc.) of programming artifacts
- **IEEE standard 828** gives details of preparing CMP including the suggested template for preparing CMP.

QAP – Quality Assurance Plan – we record the following in this plan –

- Standards (coding guidelines, design guidelines, testing guidelines etc) slated to be used in the project
- Quality control activities proposed for the project – such as code walk thru, review of requirements and design, proposed tests like unit testing, integration testing, functional testing, negative testing, end-to-end testing, system testing, acceptance testing and so on
- Software metrics to be collected for the project
- Procedure and triggers for causal analysis – for failures / defects / successes
- Audits proposed for the project
- **IEEE standard 730** gives details of preparing QAP including the suggested template for preparing QAP.

Schedule – this contains the work breakdown structure with start and end dates for the activities. This document is used to monitor progress of the project. Network analysis techniques, namely, PERT (Program Evaluation and Review Technique) and CPM (Critical Path Method) are very useful techniques in this activity. There are plenty of resources on these techniques on the World Wide Web and if you are not knowledgeable in them, my suggestion is to study these techniques. PERT handles uncertainty inherent to planning thru use of probability theory. CPM assists in locating the activities that are critical to the meeting of the over all schedule and assists in resource allocation in such a manner that the delayed activities do not affect the overall completion date of the project. Both the techniques are referred together as PERT/CPM now-a-days. Knowledge of PERT/CPM is necessary to arrive at a credible schedule for a project.

Induction Training Plan – this contains the list of training requirements to be undertaken by a new team member joining the project to bring him on par with other team members. Also, these courses are imparted to all the team members at the start of the project. Normally, the courses would include training on project plans, which specify how the project would be executed and controlled, quality assurance activities, communication, issue resolution and project specific tool / development platform training etc.

Build Plan – this would contain the number of builds, when they would be built, how they would be tested etc.

Deployment Plan – This document would contain the plan of deploying the software at the target location including deployment of hardware, system software, middleware, pilot runs etc.

User Training Plan – this document would contain the course outline for the user training, duration and schedule of user training etc.

Hand-over Plan – this document would contain the details of handing over the system, timelines, person to whom the handover would take place, what artifacts would be handed over and the signoffs thereof etc.

Software Maintenance Plan – this document would contain the mechanisms for raising maintenance work requests, service levels, turnaround times etc.

All these plans may not be required to be made as separate documents in all cases. For smaller projects, all the above could be included in the PMP itself. For medium sized projects, it is normal to prepare PMP, CMP, QMP and Schedule and include all others in PMP itself.

Project Management Planning

This is the top-level plan. This takes info (that is relevant to the project) from the Purchase Order received from the client and places it in the plan. Additionally, the methods of managing the project and tools used, project milestones, communication protocols, escalation mechanisms, issue resolution mechanisms etc also would be part of this plan.

Planning for resources

This forms part of the PMP. To achieve this we need to carry out effort estimation. We may carry out effort estimation using any of the popular techniques. For a detailed discussion on effort estimation, see "**Software Estimation – Practitioner's Perspective**" by the same author. Then, work out a schedule for the execution of the project.

We have to estimate human resources with respect to

a. The **skill sets** required for the project – this can be derived from the technical specs of the project
b. The size of software that is to be developed using any accepted software size measure such as Function Points or Software Size Units etc.
c. The amount of **effort** that has to be spent on the project – this can be arrived at by converting software size into effort required using the productivity norms of the organization
d. The amount of **duration** that has to be spent on the project – this can be arrived at by allocating resources and assigning calendar days to the activities that have to be performed for executing the project. This is also called as **Scheduling** and is covered in greater detail in the book **Software Estimation – a Practitioner's Perspective** referred to above.
e. The **likely dates** on which the resources would be needed by the project – this can be derived thru the schedule

Types of skill sets needed in a software project

We may need different skill sets to play the following roles for the project team. The following is a typical list of roles – in addition to Project Leader / Project Manager - needed for a large project –

a. **Programmers** - for each of the languages used in the project – to write the necessary programs
b. **Team Leaders** – to lead and manage teams of programmers
c. A **DBA** (Database Administrator) or a database specialist – for data modeling and to design the database initially and then to offer assistance to programmers in the efficient development of data handling routines
d. **Software Testers** / **testing specialists** – to prepare test plans and test cases and guide the team to ensure that testing is properly carried out and all defects are uncovered and rectified
e. **Language Smiths** – expert programmers for the languages used in the project to assist in troubleshooting programming issues
f. **Software (Solution) Architect** – for process modeling, to develop application architecture and to integrate the developed solution
g. **Business (Systems) Analysts** – who interact with the customer to understand client requirements and translate those needs/requirements into documents, which then, can be

understood and used by software developers to produce the solution that meets the client needs. Business Analysts also act as the customer-representative within the project team.

h. **Configuration Controller** – who would control the artifacts – information and software – such that right info is available to various team members and ensure that the deliveries to customer contain right versions
i. **Process Coordinator** – to ensure that organization's processes are implemented and the process related info is made available on time to concerned functionaries of the organization

Often, some of these roles are handled on a part time or as-needed basis. For example Project Leader or Project Manager often takes on the roles of Configuration Controller, Process Coordinator and Software Architect. Programmers may take on the role of testers too.

Once the required skill sets and the duration for which they are required, resource requests can be placed on the department that allocates person-power to the projects.

Planning for Types of machines

This would be a part of the PMP. The project team needs various hardware items for the project execution, depending on the nature of the project. The following are the typical hardware and system software requirement for a project.

a. Special computers based on the project need
b. PCs, if necessary, with appropriate terminal emulation software to connect to the development machine/server
c. Networking hardware
d. Connectivity to the customer machines, if the project is to be executed from a remote location
e. Bandwidth – if communication with remote customer or testing of a web application is involved
f. Special software – like databases, programming languages, testing tools, configuration management tools, documentation tools and so on

Methods for managing the project

These form part of PMP. The following are the typical methods that are included in the PMP –

a. **Work allocation methods** – work allocation can be carried out using an Excel sheet or a scheduling tool like MS-Project or Primavera or a WBS (Work Breakdown Structure) collaboration tool like PMPal. Intimation of work allocation may be thru formal emails or collaboration tools like PMPal or over phone or in person. All these are used across software development organizations. Similarly reporting of completion by team members may be thru email or in person or over phone.
b. **Progress measurement & review methods** – Some of the popular techniques for progress measurement and review are Earned Value technique and Line Of Balance technique. Weekly Status Report is the most common vehicle for reporting progress and for using as a base for progress review and deciding action points.
c. **Communication methods** – meetings, emails and phone calls are the most frequently used mechanisms to achieve communication within and without the project team. The means of communication is stated in PMP for the following scenarios –
    a. Communication within the project team
    b. Communicating work allocation and completion
    c. Progress Reporting
    d. Communication with client

e. Communication with project support group

Environment – tools, techniques, hardware, system software, database, IDE, testing tools, CM tools, Folder structures for artifacts in various states

Issue Resolution Mechanisms – whenever there is ambiguity or clarifications are needed, an issue arises. The mechanism used to record all issues of the project, communicating the issue to the concerned person and track the issue to resolution that is implemented for each of the issue are under the Issue Resolution Mechanism.

Escalation Mechanism – sometimes it becomes necessary to raise the issue to the next higher level. The levels which are above the normal level of communication and when to escalate an issue and the concerned executive for the issue – all these are planned and recorded under this head.

Configuration Management Planning

In a software development project, there are different configurations used –

a. **Development Configuration** – the arrangement of hardware (development machines – PCs), and the software (the development platform including the programming language, the database, IDE, third-party software used in the project etc) for use by the programmers for developing the software. The data in the database undergoes frequent change and may contain junk data. The software does change very frequently during development. Programmers use this configuration. This would have two distinct portions –
    a. **Code** – this portion would have the source code being developed
    b. **Information** / **documentation** – this portion would have all the info received from client, or developed for use in the project (like requirement specs, design, test plans, test cases etc), and project generated info (like test logs, review logs, etc) and change requests received etc.
b. **Review/Testing Configuration** – the arrangement of hardware and software for use by the reviewers/testers. Software does not change in this configuration. Programs enter this configuration and after testing are sent to either development configuration for rectification or to integration configuration for integration with other code units. Software is transient here. Data here is test data that is designed to unearth defects in the software. Data in this configuration stays consistent and does not undergo frequent change except as modified by the software being tested.
c. **Integration (Build) Configuration** – the arrangement of hardware and software for use by product integrators to receive software components and integrate them to build the product. Software components enter this configuration only when they are reviewed and tested and all unearthed defects are rectified satisfactorily.
d. **Delivery configuration** – the arrangement of software components for delivery to the client. Typically this configuration would contain –
    a. The Software Build
    b. The source code components
    c. Third-party software, if any.
    d. Software Libraries, if any
    e. Artifacts received from client, if any
    f. Images, if any
    g. User Documentation and training materials
    h. Installation guide, Operations Guide and Troubleshooting Manual
e. **Deployment (Target/Production) configuration** – this is the arrangement of hardware and software components on the target system on which the developed software would be deployed and used.

In a software development project, the daily configuration management activities revolve around promoting software components from one configuration to another. And finally, ensuring that the "right" software components are assembled in the delivery configuration for delivery to the customer.

Another important aspect of Configuration management planning is about how to manage the obsolete artifacts and the artifacts that are being improved upon. To ensure that the concerned persons refer only the current set, we have to provide unambiguous reference to the current set. We do not destroy obsolete artifacts until the project is completed. So we must make provision to store the obsolete artifacts in such a way that it is not easily accessible but available when needed using a system of appropriate authorizations. Similarly, when an artifact is being modified or being improved upon, it is to be ensured that it is stored in a designated folder so that others would not be able to access it. We can achieve this by creating three folders, namely, **Current** – this folder would contain all artifacts that are to be referred when needed, **Archive** – this folder would contain previous version of artifacts and **In-Process** – this folder would contain artifacts that are being developed / revised.

Naming Conventions

This forms part of Configuration Management Plan. Earlier days, there was restriction on the number of characters that can be used for a name. Now, we can have long meaningful names. Then why do we need naming conventions? We need naming conventions –

1. Prevent duplicate names or bring clarity when duplicate names are used
2. Easily recognize the contents of the artifact
3. Easily identify a group of artifacts, such as all artifacts of a specific module
4. Achieve uniformity in naming artifacts
5. In programming, naming conventions allow us to distinguish the one type of variable from the another type of variable – such as programming variables from table field names

Naming conventions normally use pre-fixes to distinguish between various categories. The prefixes can be one or more. Naming conventions are defined for the following –

a. Document Names
b. Program / Subprogram Names
c. Screens, Reports
d. Numeric Variable
e. Alphanumeric Variables
f. Flags
g. Counters
h. Database Table Names
i. Database table field names
j. Error Messages
k. Information messages
l. Window
m. Controls like Combo Box, Text Box, Command Buttons etc

Change Management

This forms part of configuration Management Plan. Change Management in software projects includes –

a. Receiving Change Requests (CRs) – this involves designating a single point to receive change requests from all sources, consolidate them and maintain a Change Register
b. Analyzing the CRs received – this involves specifying agencies responsible for analyzing the impact CRs received on schedule, effort, and cost and providing this data for taking a decision on implementing the CRS
c. Implementing the CRs – this involves the mechanism for implementing the CRs, namely,
    a. Accept or reject decision
    b. If accepted, decision on when to implement the CR – as and when received or to retrofit all CRs at the end
    c. If accepted, the decision on how to absorb the impact or to pass it on to the customer
    d. Obtain / accord approval for CR implementation
    e. Implement the CR
    f. Quality control of the CR implementation
    g. Closure of the CR
d. Progress Reporting of CRs this involves the mechanism to track all CRs received, and tracking all CRs to resolution – communicate the status of all CRs to all the concerned agencies on a periodic basis
e. Closing the CRs

Quality Assurance Planning - QAP

This plan basically focuses on achieving the specified level of quality on the artifacts to be produced by the development team. This plan normally contains the following details –

a. Standards to be used in the project – such as –
    a. Coding Standards
    b. Database Design Standards
    c. GUI Design Standards
    d. Test Case Design Standards
    e. Testing Standards
    f. Review Standards
    g. Organizational Process reference
    h. Etc.
b. The specifications of quality levels for the project. These may be –
    a. Defect Injection rate
    b. Defect Density
    c. Productivity for various artifacts of the project
    d. Schedule variances
    e. Etc.
c. Quality Control activities to be implemented in the project – such as –
    a. Code Walk-thru
    b. Peer Review
    c. Formal Review
    d. Various types of Tests that would be carried out during the project execution. At a minimum these are –
        i. Unit Testing
        ii. Integration Testing
        iii. System Testing
        iv. Acceptance Testing
d. Measurement for the defined quality levels, periodicity and reporting
e. Causal Analysis for the variances – both positive and negative – and its schedule
f. Schedules for various audits proposed for the projects –
    a. Periodic Conformance Audits

   b. Phase-end Audits
   c. Criteria for Investigative Audits
   d. Delivery Audits
g. Process Improvement activities, if any
h. Progress reporting to all concerned agencies of the status of quality assurance activities implemented in the project

Schedule

This is best achieved using a scheduling-software such as MS-Project or Primavera. All the activities that are need to be performed to execute the project are enumerated; their predecessor relationships defined; resources allocated and dates are set for the activities.

Induction Training Plan

This plan contains the topics on which the project personnel need to undergo training – examples are –

a. Project Plans
b. Team Communication methods
c. Quality control activities
d. Issue resolution mechanisms
e. Escalation procedures
f. Development platform
g. The methods of training
h. Self study material availability
i. Etc.

Build Plan

Build Plan contains the following info –

a. Approach for integration – top down or bottoms up
b. Configuration of integration environment
c. Quality assurance activities before accepting components into build environment
d. Quality assurance activities after integrating each component, each module and at the completion of the build
e. Etc.

Deployment Plan

Deployment Plan consists of the following info –

a. A schematic diagram of the deployment components including hard ware, software, networking etc
b. Floor Plans, if necessary for deployment of hardware and networking
c. A BOM (Bill of Material) listing all components being deployed along with technical specs of each of the components
d. Quality assurance activities planned for deployment
e. Technical methods, if necessary for deploying the configuration
f. Etc.

User Training Plan

User Training plan consists of the following info –

a. Details of courses on which users have to be trained, user-class-wise
b. Course material for each course including training slides, teaching notes, lesson plans, session breakdown, participant handouts etc
c. Schedule of courses to be conducted
d. Details of facilities needed for conducting the training
e. Etc.

Handover Plan

The handover plan consists of the following info –

a. A BOM (Bill of Material) of all components to be handed over
b. The mode of handover / takeover
c. Required sign-offs details
d. Schedule of handover
e. Etc.

Software Maintenance Plan

Software Maintenance Plan could cover only the software maintenance during warranty period or even later depending on the contractual requirements. This plan consists of the following info –

a. Process of raising requests for software maintenance
b. Formats, and templates for raising software maintenance requests
c. Service Levels agreed – turnaround times for software maintenance requests
d. Procedure for classifying software maintenance requests and prioritizing the same
e. Issue Resolution mechanisms and escalation mechanisms
f. Environment for software maintenance
g. Etc.

Documenting the plan

Documenting the plan differs from normal writing. A Project Plan is a document that is used by many persons as a reference and is used in guiding human effort and causes expense. This document ought to be like an engineering drawing. Hallmarks of an engineering drawing are –

1. Unambiguous representation – the same inference is drawn irrespective of the person interpreting it
2. One fact is presented at one and only one location – never repeated. (Presentation of information at multiple locations may cause conflict – especially during revisions it may be updated at one place and ignored at another)
3. Free-flowing language is not made use of – it is subject to guidelines

Taking a clue from engineering drawings, the project plan be better –

1. Adhere to documentation guidelines of the organization in the matter of presenting information
2. Avoid duplication of information at multiple locations
3. Avoid ambiguity

Uniformity in plan documents can be achieved in an organization, keeping in mind that multiple persons are likely to prepare documents, by using templates. Each organization needs to define its own templates. One best source for guidance on templates would be the IEEE standards on software engineering.

Roles in Planning

Achieving effective project planning needs collaboration between various agencies in the organization. Basically there are two entities in the organization that impact project planning, namely, the organization that needs and infrastructure for project planning and the individual who carries out the project planning. Their roles are discussed below.

Organizational Role

Organization needs project planning and provides infrastructure that facilitates and enables effective project planning. These include –

1. Develop, establish, implement and continuously improve Project Planning Process in the organization – including procedures, templates, formats, quality assurance for plans
2. Various guidelines and standards including documentation guidelines, checklists for preparation and review of plans, estimation guidelines and productivity figures for various technologies used
3. Establish a nodal agency that takes charge of all project plans and assists concerned individuals preparing project plans in preparing project plans as well as receive feedback and ensures that it is analyzed and dovetailed back into the process, standards and guidelines
4. Arrange for review of the plans vis-à-vis actual execution at the end of every project to capture best and worst practices and the efficacy of the project plans
5. A Knowledge Repository on the subject of Project Planning
6. A Corporate Memory (a Repository) of past estimates and project plans for reference by persons planning the projects
7. Structured Training for individuals planning the projects
8. Recognizing that Project Planning is a specialist activity and subjecting it to the rigors of Process Improvement as well as recognizing and rewarding individuals who excel at it

Individual Role

Individuals can make or mar the planning. Making best use of the available infrastructure for project planning, individuals vested with the responsibility of planning projects ought to excel at project planning. Individuals can –

1. Assist the organization in the developing, establishing, implementing and continuously improving the project planning process
2. Adhere to the organizational process, standards and guidelines in letter and spirit
3. Give feedback to concerned agencies
4. Participate in the process improvement activities wholeheartedly
5. Carry out the project planning activity to the best of one's ability as diligently as possible
6. Recognizing that Project Planning is not preparing documents but that it is planning of the project and documenting plans is only for the purpose of review and reference by other concerned agencies

==========

About the Author:

Murali Chemuturi is a Fellow of Indian Institution of Industrial Engineering and a Senior Member of Computer society of India. He is a veteran of software development industry and is presently leading Chemuturi Consultants, which provides consultancy in software process quality and training. He can be reached at murali@chemuturi.com