SQMS THEORY

1)Software Development Lifecycle Cycle

Life and its Cycles

Anything that exists over a period of time is said to have a life span, be it a living creature or an event; For each of these life spans, there is a definite process of beginning and end, be it for a butterfly or a human being

Events and their life cycles

Even having a meal at a restaurant has a life cycle. Similarly, the software process also has its life cycle.

SW Engg phases

Just as human life starts from infancy and goes through childhood, youth, middle age, old age and death, the software development life cycle has the following stages: Requirements, Design, Coding, Testing, Production Rollout, and Maintenance. These stages are formalized in different ways to form different Software Life Cycle models.

Waterfall model

In the mid 1960s, while working for the United States Department of Defense, A.Enthoven and Henry Rowan developed a linear list of stages for the software development. Winston Royce introduced the first S D L C model in 1970. This became known as the Waterfall model. However, real projects seldom follow a sequential flow of phases, and changes in between cause concern. A lot of patience is needed from customers and accuracy in stating the requirements correctly at one go is of prime importance. Customers are generally very very short on accuracy and patience.

Here is an example that explains the concept of a waterfall model. The waterfall model is like ordering a platter in a restaurant. But it has its own risks, for example, requirement changes. The waterfall works best when requirements are clear and unchanging.

In the early 1980s, the more flexible Incremental Model was introduced. It is also called the Staged Delivery Model. This model performs the waterfall in overlapping sections, attempting to compensate for the length of projects by producing usable functionality earlier, in increments. All the requirements are collected at one shot. The technical architecture is finalized upfront. The objectives are divided into several increments or builds. The first build is developed and delivered. This is followed by the next portion until all the objectives have been met. It is easier to build and design than a whole project. However, it has its own drawbacks.

Incremental MODEL

In the early 1980s, the more flexible Incremental Model was introduced. It is also called the Staged Delivery Model. This model performs the waterfall in overlapping sections, attempting to compensate for the length of projects by producing usable functionality earlier, in increments. All the requirements are collected at one shot. The technical architecture is finalized upfront. The objectives are divided into several increments or builds. The first build is developed and delivered. This is followed by the next portion until all the objectives have been met. It is easier to build and design than a whole project. However, it has its own drawbacks.

The Incremental Model is analogous to being served course by course in a restaurant after ordering the whole meal upfront. In this case, incorporating change is not that difficult. Delivering in phases builds the confidence of the customer as he sees the deliverables in increments.

Iterative model

The ideal waterfall model where nothing goes wrong is the Project Manager's dream, with defined requirements and timelines. But problems creep in with changes to requirement. The project becomes risky. Often, the requirements of the ultimate software system are not very well defined whereas the software solution is required to address a business requirement. To counter this, in the early 1990s Barry Boehm developed the spiral model- a waterfall that flowed back to the source. Each phase starts with a design goal and ends with the client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the business goal of the project.

For a typical application, the spiral model means that you have a rough-cut of user elements as an operable application, add features in phases, and, at some point, add the final graphics. Each phase starts with a design goal and ends with the client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the business goal of the project. The spiral model is used most often in large projects and needs constant review to stay on target. However, if we use the spiral in the reverse, that is adding the peripheral features and then developing the core, we more often than not end up going nowhere.

Other models

There are several other Life Cycle models: Rational's Iterative Development Model, Adaptive Model, Rapid Application Development, Evolutionary Prototyping, and V Model. You are not required to memorize the models now. The intention is to give you an idea of what the software life cycle phases are and how they are combined into different structures to form life cycle models.

Maintanence models

For projects that require us to maintain software, Cognizant has its own Application Value Management process model. This involves initial knowledge transition, followed by steady state, where enhancements, production support, and bug fixes take place

Model selection

The selection of the appropriate model depends on a number of factors. It depends upon the project type, client requirements and priority, nature of customer, nature of requirements, technology to be used, budget, and various other factors.

For example, stable requirements, well understood technology, small software solution for complex and well understood problems are criteria that can seem apt for the waterfall model.

2. SW QUALITY BASICS

WHAT IS QUALITY

What is quality all about?
Here is an illustration that shows what one means by a quality product.

- I need a shirt that will fit my specifications.
- My shirt needs to be defect free.
- It should also be cost effective.
- This shirt has two pockets, which is a unique feature.
- I will take this shirt because it is also branded as a quality shirt.

Quality can be defined in various ways depending upon the customer's point of view or from the point of view of the product or from a value angle or even according to the manufacturing specifications. So many definitions of quality leave us confused. In reality, all of them combine to define the quality of a product.
Quality distinguishes a product and becomes the deciding factor in competition. Whatever work product you produce during day-to-day activities contribute to the quality of the product.

THE PROCESS OF ENSURING QUALITY

The process of ensuring quality in a product comprises the following components:

- Quality assurance ensures proper process for production of the end product.
- Quality control ensures that the end product meets standards.

The entire process for ensuring quality makes up the quality management of an organization.

RESPONSIBILITY OF QUALITY

In this page you, will learn the responsibilities involved in ensuring the quality of a product. The responsibility of ensuring proper quality of every deliverable lies with the organization. This implies that it is the responsibility of each and every employee of the organization. The quality of the process and product of the organization needs to conform to certain set standards. These standards need to be adhered to by the members and checked for conformance by certain specialists of the company.

QUALITY MODELS AND STANDARDS

Now you will learn about the various Quality Models and Standards.

Frameworks to ensure proper quality processes can be developed by the company aided by the standards and the models, which are internationally accepted in the industry. Adhering to internationally accepted quality standards and models may lead to certification. The various standard models that are accepted worldwide are:

- I S O
- C M M i
- P C M M
- BS 7 7 9 9
- Six Sigma

These models have been developed by established academic and professional institutions and have been tested over time.

NECESSITY OF CERTIFICATION

Apart from the obvious benefits of producing quality products by following processes that conform to internationally recognized standards and models, certifications also have the brand value, which helps us in obtaining the confidence of the customers.
Quality and associated certifications are the differentiating factors that allow companies to beat their competitors and win clients.

QUALITY AND SOFTWARE

In Software Engineering, the quality components map to similar activities as in any other industry:
The quality control activities verify whether the deliverables are of acceptable quality and that they are complete and correct. Examples of quality control activities in software involve testing and reviews
Quality assurance verifies the process used to create the deliverables. This can be performed by a manager, client, or a third-party reviewer. Examples of quality assurance include the software quality assurance audits, software configuration management and defect prevention.

The quality management system includes all the procedures, templates, and guidelines associated with software engineering in the organization. In Cognizant, the quality management system is the QView.

Even though there are policies, frameworks, review, testing, audits and conformance checks in place, the ownership and responsibility of producing a quality product rests with the individual.
Quality is a reflection of your ability.
Whatever you produce needs to conform to a specified quality, be it code, documents, manuals or even a printout.

3. SOFTWARE QUALITY IN COGNIZANT

QUALITY THE JOURNEY

Quality is not a goal but a journey. There are multiple paths leading to proper quality.

LEARNING OBJECTIVES

At the end of this session, you will be able to:
Learn the basics of the major quality models
Study Cognizant's quality journey and method of implementation
Describe process management through Q View
Explain Cognizant's quality culture

COGNIZANT QUALITY'S CERTIFICATIONS

To ensure the highest possible quality of software development and maintenance,

Cognizant lays stress on a Quality Management Process that integrates Cognizant's quality approach throughout the software development life cycle, thereby ensuring that quality is built in as development progresses. Here you will learn about the major quality models that Cognizant employs for ensuring the highest quality for its product. To facilitate the on-time, on-budget completion of projects, integrating on-site and offshore teams, Cognizant utilizes ISO 9001 certified methodology and SEI or CMMi Level 5 processes to define and implement projects, and to allow frequent deliverables as well as feedback from customers. Cognizant has also become the first software company to be assessed at People-CMM Level 5, across all of its development centers in India. Along with this, Cognizant ensures security of information by obtaining and adhering to the requirements of BS 7799 certification. Finally, on the journey of quality toward continuous improvement, Cognizant has focused management direction toward the Six Sigma method of minimizing defects in the delivered products.

Every activity leading to production of deliverables needs to be developed using the process and practices of the Standards and Guidelines. In the following pages, you will learn in detail about these quality models.

ISO 9001:2000

ISO 9001 is a model for Quality Management System promoted by the International Organization for Standards. It ensures uniformity across development units through standard procedures. It is heralded as the first step to achieving high quality.

ELEMENTS OF ISO

ISO 9001 consists of five main sections which covers the standard requirements.
Click the highlighted section to learn more about these standard requirements.
Quality management system encompasses quality manual documentation and control of documents and records.
Management responsibility deals with management's commitment to the quality management systems and explains that management must be dedicated to the organization's products and customers and to the planning and review processes.
Resource management provides the criteria needed to perform a job competently and in a safe environment. Human resources, infrastructure planning, and work environment are discussed in this section.
Product realisation defines the steps in product development. These steps include everything from the initial design phase to the final delivery phase. Measurement, analysis, and improvement focuses on measuring, analyzing, and improving the quality management system by performing periodical internal audits, monitoring customer satisfaction, controlling nonconforming products, analyzing data and taking corrective and preventive actions. It is aimed at achieving customer satisfaction by preventing nonconformity at all stages from Design to Servicing.

CAPABILITY MATURITY MODELS

While ISO is a generic model applicable to any industry, Capability Maturity Models deal with processes typical to the software industry. The Capability Maturity Model was developed by the Software Engineering Institute in Carnegie Mellon University in 1987. The US Department of Defense funded SEI to define a model for evaluating the capability and maturity level of the vendors supplying software. The results of the work include various Capability Maturity Models, such as The Capability Maturity Model for Software, The Systems Engineering Capability Model, and The Integrated Product Development Capability Maturity Model. Although these models have proved useful to many organizations, the use of multiple models has been problematic. Further, applying multiple models that are not integrated within and across an organization are costly in terms of training, appraisals, and improvement activities. The CMM Integration project was formed to sort out the problem of using multiple CMMs. The result was the Capability Maturity Model Integrated or CMMi.

ELEMENTS OF CMMI

You learned that, CMMi stands for Capability Maturity Model Integrated.
CMMi consists of guidelines for managing the:
Process Management Processes
Project Management Processes
Engineering Processes
Support Processes
Linking the four, CMMi forms guidelines to manage and execute software projects.

BENEFITS OF CMMI

The different benefits from CMMi are in the areas of improved process, budget control, delivery capability, increased productivity, and reduction of defects.

THE MATURITY LEVELS OF CMMI – STAGED REPRESENTATION

CMMi level 1 organization denotes that processes are unpredictable, poorly controlled and reactive.
When an organization is at level two, process is characterized for projects and often reactive.
At level three, the process is characterized for the organization and is proactive.
At level four, the process is measured and controlled.
At the final level, the focus is on process improvement.
One has to progress maturity level by maturity level and reach the pinnacle of excellence.

PEOPLE CMM(PCMM)

The People Capability Maturity Model measures how effectively an organization manages its professional force. People are the most important resources in the software industry. People CMM stresses on people-centric processes, which ensure retention of workforce and growth of the organization. Organizations that achieve Level five certification are not only implementing and benchmarking their execution of enterprise-wide, workforce management practices, but they are in continuous improvement mode, always seeking opportunities for improvement. A high PCMM level is characterized by effective professional training and mentoring programs and emphasis on continuous improvement.

BS7799

Information is one of the most valuable assets owned by Cognizant. Securing information should be one of the most important responsibilities of every associate. Loss of information could result in a loss of man hours spent creating information as well as several more man-hours trying to recover lost information. Information lost outside the corporate environment could allow competitors to gain undue competitive edge. Information is also trusted to Cognizant by the customers. Information received from the customers need to be protected against loss and unauthorized access and use. BS 7799 is

a Standard for developing and maintaining an Information Security Management System. The ten domains that come under BS 7799 are Security Policy, Security Organization, Access Classification and Control, Personnel Security, Physical and Environmental Security, Communications and Operations Management, Systems Development and Maintenance, Access Control, Business Continuity, and Compliance. The following page gives you the details on Cognizant's security arrangements.

COGNIZANT'S SECURITY CLASSIFICATION

The Security classification in Cognizant is divided into:
C1 or, Highly Critical - This involves firewall and finance related data.
C2 or, Critical, which involves customer supplied item and personal folders.
C3 or, Protected - Includes any project document.
C4 or, General - Consists of General information, like cognizant.com.
The Security classification of the data implies the following:
Storage of Information
Transmission of Information
Access to Information
Destruction of Information

SIX SIGMA – REDUCTION IN VARIATION

Just as security of information, reduced variability is also very critical for customer satisfaction. Six Sigma is a statistical methodology that significantly improves customer satisfaction and shareholder value by reducing variability in every aspect of our business. "Six Sigma is not something else that you do...it is what you do."

Six Sigma is using a standard methodology to use statistical solutions for reducing variability of a product and thereby reducing defects. The ultimate aim is to produce products with a variability of 3.4 defects per million opportunities. From a statistical point of view, it means to work toward achieving a normal distribution of the produced products with very low variance around the target.

COGNIZANT'S QUALITY JOURNEY

Take a look at how Cognizant's quality journey has evolved over time and what it implies. All the certifications of Cognizant have been achieved enterprise-wide across all centers. The certifications have been achieved in an incremental approach with Cognizant's growth as a company.

QVIEW 5.24

Cognizant's Quality policy is outlined in a web-based application known as Qview. The Quality Management System is available through Intranet
http://cognizantonline/qview/
You will view a demo on Qview in the next page.

QUALITY IMPLEMENTATION IN COGNIZANT

The responsibility for monitoring all the process and quality related activities lie with the individual employees of Cognizant. The group responsible for ensuring quality processes and continuous improvement is the process and quality group. The diagram shows the structure of the process and quality group. SEPG is a virtual group consisting of quality champions and representative practitioners from all locations and vertical or, horizontal or, support groups. It is responsible for process definition, maintenance, and improvement of the software processes used by the organization. The SQAG is responsible for guiding projects through facilitations and monitoring the process and quality of the Projects through auditing and reporting, periodic status reporting, and timely escalations. The SQAG is also responsible for conducting the internal quality audits that serve as periodic assessments of project and process health and compliance in the organization.

SOFTWARE QUALITY ASSURANCE PROCESS

This illustration shows the types of audits that take place in a project and their distribution throughout the project life cycle. All the audit results, which include, the non-conformances and the observations, are logged in the Q Smart online tool. Like any other process, the quality process also strives for continuous improvement in Cognizant. This improvement is enhanced by the feedback about the quality process received from the associates. Q Smart can also be used to provide feedback on the quality process. Any associate can provide feedback in this manner.

QUALITY ENCOURAGAMENT IN COGNIZNANT

Apart from the certifications to be adhered to and the quality champions and quality reviewers, the management also encourages quality and innovation by awarding the project of the year, associate of the year, and the best practice awards. The best ideas and innovations are recognized, rewarded, and re-used to ensure continuous improvement and high quality in process, product, and practice.

4. SOFTWARE ENGINEERING IN COGNIZANT:

LEARNING OBJECTIVES

At the end of this session, you will be able to:

Learn the types of projects in Cognizant and their adaptation of SDLC

Study the software engineering concept through the QView
Know how to address the customer requirements in designing delivery

TYPES OF PROJECT

The types of projects that are dealt with by Cognizant can be classified into Development Projects and Application Value Management or Maintenance Projects. In the following pages, you will learn in detail about application development projects.

APPLICATION DEVELOPMENT PROJECTS

he application development projects handled by Cognizant are of the following types: New development - A project developed from scratch for a customer based on given requirement, Re-engineering - Converting an existing software system into another system having the same functionalities, but with different environment, platform, programming language, operating system, etc., and Product development - A software product developed not for a single customer, but based on some market demand for a customer base for that type of product.

APPLICATION DEVELOPMENT CLASSIFICATIONS

Development projects are classified in two different ways:
Classification based on life-cycle models like Waterfall, Incremental, and Iterative.
Classification based on size like Large, Medium, Small, and Very Small.

The reason why the projects are categorized into large, medium, small, and very small is that they need to follow different processes for development. A large project and a very small project cannot follow the same steps while executing deliveries. To keep a large project on track, one needs to have substantial processes in place, whereas having few processes leads to the risk of unmanaged processes, which may lead to failure. Whereas, for a small project, having a huge process makes it a tedious and unnecessary overhead.

PROJECT SIZE CLASSIFICATIONS

Decisions regarding a project's size, whether it is a large, medium, small or very small project, are arrived at depending upon the duration and the effort associated with the project. The total project duration includes requirements through the system testing phase in calendar months. The total project effort includes on-site as well as off-shore effort and requirements through the system testing phase. If a project is of less than one-month duration or the effort required is less than three person months, it is considered a very small project. If a project is of between one and three months or the effort required is between three and sixteen person months, it is considered a small project. If a project is of between three and six months or the effort required is between sixteen and forty-eight

person months, it is considered a medium project. If a project is greater than 6 months or requires more than forty-eight person months, it is considered a large project.

OPERATION LIFE CYCLE MODELS – RECOMMENDED

The table shows the recommended life-cycle models for the different types of development projects in Cognizant. A large project can follow the waterfall, incremental, or iterative model. A medium project follows the waterfall or iterative model. Small and very small projects follow the waterfall model. Most of these recommendations are because of understandable reasons. A project of small duration could run into problems if there were too many cycles of delivery as in iterative. Also, unless a project is of a large duration, breaking it up into increments does not make sense and might lead to complications from multiple deliveries. The following pages will give you details on these three types of life-cycle models.

1)WATERFALL MODEL ACTIVITIES

Take a look at the different activities of a waterfall model project.

WATERFALL IS RECOMMENDED WHEN

A waterfall model is recommended when:
The requirements are clear
The project duration is smaller
The design or technology is proven or mature
The proposed application is very similar to an existing system
The customer does not have any intermediate release requirements

2)INCREMENTAL MODEL ACTIVITIES

Here are the different activities that take place in an incremental model project.

INCREMENTAL MODEL IS RECOMMENDED WHEN

An incremental model is recommended when most of the requirements are well-understood with the exception of some T B Ds, and the total project duration is greater than 3 months. In this model, the initial increment may focus on validating architecture and key technical risks. Functionality is built over increments, such that the initial increment may focus on high-priority functionality, and the last increment may focus on nice-to-have features.

3)ITERATIVE OR FEATURE – DRIVEN DEVELOPMENT ACTIVITIES

The activities and flow of an Iterative Development Life Cycle are illustrated here.

ITERATIVE MODEL IS RECOMMENDED WHEN….

The iterative model is recommended when:
The requirements are not clear.
The initial planning broadly scopes the iterations and arrives at a roadmap
The initial iterations may involve architecture, UI frameworks, etc.
Subsequently, every iteration focuses on the set of requirements that need to be built into that iteration
The iterations may be individually delivered to the customer
A release to production can combine one or more iterations

This model is mainly suited for time and material-based engagement.

MODEL SELECTION

The selection of the appropriate model depends on a number of factors. It depends upon the project type, client requirements and priority, nature of customer, nature of requirements, technology to be used, budget and many different factors. For example, stable requirements, well understood technology, small software solutions for well **understood problems are criteria that can seem apt for the waterfall model.**

PROCESS MODEL FINALISATION

The processes recommended according to the Cognizant quality system are combined with the processes required by the client to arrive at the project's process model. This process is then tailored to suit the different demands of the project. Tailoring to suit your project requirements is a very essential defined process because, the project needs to be aligned with this. Tailoring of the recommended process may be required due to customer requirements or project-specific requirements. This procedure is followed for all types of projects.

PROJECT SW HANDBOOK – APPLICATION DEVELOPMENT PROJECTS

Here is a schematic description of the process model definition.
The steps include:
Determining the size of the project based on duration and effort, contractual requirement, and project scope
Selecting an adequate project life-cycle model to arrive at an operational model

Merging with the customer's process and project-specific requirement to form the project's software process
Tailoring for project needs to arrive at the process model

PROCESS MODEL CONFIGURATION

Process model configuration is the single most important activity to finalize the process model before the start of the project. Process Model is the basis for some processes like, Project planning and tracking, Application build, integration, testing, release, branching or merging strategies may be derived based on this, Distribution of work within teams and across teams, Planning for Rollouts, etc.

PROCESS – OVERVIEW

All the development life-cycle models follow the parallel processes of development and delivery management. While the development process deals with the project development stages and associated activities, delivery management deals with the project management activities relevant throughout the project.

APPLICATION VALUE MANAGEMENT PROJECTS

The types of application value management projects that are handled by Cognizant are:
Maintenance: This means taking ownership of the software system and ensuring that it meets current and future needs of all users at prescribed service levels.
Mass Change: These are projects involving change of attributes of a system from one to the other.

Examples are Mass Conversions, such as field expansion, Internationalization, Decimalization, Euro Conversion, Rehosting, Upgrade Projects, Database Conversions,
Testing are Projects whose scope is limited to testing of a system.

APPLICATION VALUE MANAGEMENT (AVM)

Any AVM project consists of taking ownership of the software system and ensuring that it meets current and future needs of all users. A maintenance project typically involves providing the applicable activities from the following:
Production support
Bug fix
Minor enhancements
Major enhancements
Testing
Documentation
Re-engineering, etc.

The process of any application value management process consists of planning, knowledge transition from the current team to the maintenance team, guided support during which the software is maintained by the maintenance team with the guidance of the current team, followed by transition to full-fledged support, and finally the steady state during which the maintenance team supports the application fully.
The picture shows the process for one typical maintenance project.

PROCESS MODEL FINALISATION – AVM

Similar to development projects, in AVM projects also, the processes recommended according to the Cognizant quality system are combined with the processes required by the client to arrive at the project's process. The process is again tailored to suit the different demands of the project.

PROJECT SOFTWARE PROCESS HANDBOOK FOR AVM

The schematic description of the process model definition for an AVM project is shown here. The steps include:

Selecting adequate project life-cycle based on the type of AVM project, such as maintenance, mass change or testing, and project scope to arrive at an operational model

Merging with the customer's process to form the project's software process, and

Tailoring for project-specific needs to arrive at the process model

AVM PROCESSES

The AVM life-cycle also follows the parallel processes of maintenance and delivery management. Take a look at the process models for maintenance, mass change, testing.

CRITICAL TO AVM PROCESSES

Some of the critical success factors of an application value management project are:
Proper Knowledge Transition - Ensures that knowledge of the system is properly transitioned to the team that will ultimately maintain the system.
Well-defined service-level agreement - Is an agreed upon timeframe within which a service request like bug fix, production support request, etc., is supposed to be responded to and ultimately resolved.
Well-defined tracking mechanism - The performed tasks need to be logged and tracked as part of documented records. Tracking is also required to ensure that the service-level agreements are met.
Generally, AVM projects use the eTracker tool to log the activities and to track adherence to SLAs. Maintained Documentation details the process for request categories, such as production support, major bug fix, minor bug fix, etc.
Proper channels of escalation in case of unresolved problems.

POINTS TO REMEMEBER FOR DELIVERY

While designing for delivery for development as well as AVM projects, it is imperative that all customer requirements are addressed. Apart from the stated functional requirements of the system, there are typically other requirements which have to be derived from the client. They may be explicitly stated by the client or may have to be extracted from them. In most cases, one has to extrapolate the complete requirements inclusive of must have, nice-to-have features from the requirements directly available from the client.

ADDRESSING CUSTOMER REQ –SPECIFIC CONSIDERATIONS

The requirements typically include:

Non functional requirements inclusive of speed, response time, and scalability
Adhering to service-level agreements
Prioritization of modules, tasks, etc. due to business needs
Aligning deliveries to meet client constraints

5. PROJECT MANAGEMENT CONCEPTS

WHAT IS PROJECT MANAGEMENT

Imagine organizing a college fest. Behind the face of a day or two of enjoyment and celebrations, what are the challenges involved for the one who is in charge of organizing it? One runs up against budget issues, time and schedule issues, resource problems, risks, constraints, and managing expectations. Arranging and executing a college fest is a project. And all the challenges faced are a part of any project in the industry.

LEARNING OBJECTIVES

At the end of this session you will be able to:
Study the concepts of a project and project management
Know who the stakeholders in a project are
Learn the structure of the project team and the role of team members in the project
Explain the basics of the important project management tools in Cognizant

WHAT IS A PROJECT

in the previous example, organizing the college fest was a temporary endeavor and it was unique-that is to say that every college fest is different. Similarly, any project is a temporary endeavor undertaken to create a unique product or service and it progresses through a number of life cycle phases. Temporary implies that every project has a

definite beginning and a definite end. Unique implies that the product produced by the project is different in some way from all similar entities

TYPES OF PROJECT IN COGNIZANT

Some of the types of projects executed in Cognizant are Development of new applications, Maintenance of existing applications, Re-engineering an existing application, and Mass Change or migration like Euro, Internationalization.

WHAT DOES PROJECT MGT INVOLVE

Whatever be the type, a project always has certain characteristics and challenges. Just like the college fest that you saw previously, any project in the industry involves challenges with respect to cost, risks, scope, quality, resources, expectations, time, etc. Therein lies the need of project management.
Project management involves balancing competing demands among: scope, time, cost, resource, and quality; stakeholders with differing needs and expectations, identified requirements or needs and unidentified requirements or expectations. Developing software products involves more than just combining programming instructions together and getting them to run on a computer.
Effective project management processes guide you toward disciplined, superior software engineering.

RE-LOOK AT PROJ MGT

Project management involves application of knowledge, skills, techniques, and tools to project activities in order to meet or exceed stakeholder needs and expectations from the project. You will learn about the stakeholders in a project on the next page.

STAKEHOLDERS IN PROJECT

What are stakeholders and who are the stakeholders in a project? A stakeholder is anyone who has an interest in the project. They may be at the client's end, internal to the project, or external. The external are Sub-contractors, Suppliers, External consultants, and Service providers. The internal stakeholders are Senior management, Project manager, Human resources, Business development, Finance, Administration, Training, Internal systems, Quality assurance, and SEPG. The client stakeholders are Sponsor, Contact person, End-user, and Outsourced party.

STAKEHOLDER EXPECTATIONS

A project manager's job consists of balancing numerous stakeholder expectations. What are the stakeholder expectations? Examples of stakeholders and their expectations could

be: The customer who asks: When can I get the product? What are the features that exist? What are the additional features required? How much would I have to pay? The employee who asks: What is my role and the job or task specifications in this project? What learning opportunity or new skill will I get from this project? The shareholders who ask: What is the company doing to minimize overheads and maximize profits? What are the efforts and direction for new avenues of business this year? How does this company compare with and stay ahead of the competition? And finally the project manager himself who asks: When can I get the product out? What is the quantum of work involved? How many people and what skill will this project need? What would be their utilization and availability on this project? How much would the cost or investment for resources be? Managing the expectations of stakeholders can be demanding and it is the primary job of the project manager.

PROJECT ORGANISATION STRUCTURE

The project organization structure of a typical offshore-onsite delivery model in Cognizant is shown in the illustration. As evident from this illustration, every individual has a role to play. While the people at the top of the picture are there to provide guidance and to ensure communication with the end customer and proper execution of the project, the team members are essential for performing the different activities that go on to satisfy the requirements of the project. All the while, the support groups including NSS, Admin, HR, and quality assurance provide all the necessary support for the project to meet the various infrastructural, financial, resource, and quality requirements. Big or small, everyone has a role to play in the project.

INDIVIDUAL RESPONSIBITY

It is a basic rule of project execution that every individual in the project has a role to play. Every task assigned to the individual is important and any task, however insignificant it seems to be, is crucial to the project.

It is very important that individual responsibilities are addressed according to the plan and in coordination with other team members for proper alignment with each other. Otherwise, there are chances of making individually perfect pieces, which will not work in unison. This concept of teamwork is especially important since in Cognizant we follow the onsite-offshore model with the members of the project team working in geographically different locations. Each task needs to be aligned to the project objective and goal. Otherwise, we may end up with something like .. this

IMPORTANT PROJ MGT TASKS

The most important project management tasks during the execution of the project involve:
Planning for the project's cost, effort, and schedule at the beginning

Tracking the execution of the project according to the plan

TIME SHEET

To track the progress of the project, one needs to monitor whether the effort, cost, and schedule are following the plan or are deviating. In case of deviations, one needs to take corrective action, re-modify the plan, etc. To track the effort and the cost of the project, it is of extreme importance to log the time spent by each individual in the project against the activities performed by them. For this purpose, each associate is expected to log his activities in the peoplesoft tool. Peoplesoft Demo follows this.

6. DELIVERY MGT CONCEPTS

DELIVERY MGT

Obtaining a degree is more than studying and taking the test. It additionally involves arranging for hostel accommodation, arranging for commuting services, setting up a time table, selecting a study environment, identifying resources and study material, having periodic meetings with the teachers, receiving the grades, and finally celebrating the graduation.

Similarly, a software project does not begin and end with the software development life cycle consisting of requirements, analysis and design, coding, testing, and delivery stages. Additionally, it involves activities throughout the project life cycle such as business understanding, configuration set up, infrastructure set up, execution, progress analysis, change management, and finally closure. All these activities fall under the delivery management framework. Delivery management is a new framework to address all project management practices across the entire life cycle of a project. This framework describes the various stages in project management, such as the entry and exit criteria, tasks, input and output, tools and techniques to perform the tasks, responsibilities, verification and validation criteria.

DIFFERENT PROCESS OF DELIVERY MGT

The different processes of delivery management are Proposal, Formalization, Startup, Execution, and Closure. The following pages will give you details on these processes.

PROPOSAL PHASE

The proposal phase involves understanding the business, making high-level estimates and doing the initial risk assessment.

FORMALISATION PHASE

The formalization phase deals with the formalization of the project. A signed letter of intent or contract or statement of work is obtained during this phase. The work order is raised in Peoplesoft.

START UP PHASE

The project startup phase involves: Project plan, which includes the details of scope, resources, communication plan, team details, schedule and milestones, infrastructure, training, knowledge management details, metrics plan, quality assurance plan, etc.

Project configuration set up involves setting up the project details and methodology particulars in Prolite, setting up the different tools required by the project, and preparing the project-specific process handbook. Detailed estimation, and finally, defining the business continuity plan. This plan is the contingency plan designed to ensure that the business operations of Cognizant are not affected by the occurrence of a disaster or in an emergency situation.

EXECUTION PHASE

The execution phase involves delivery of product, tracking activities against the plan, performing defect prevention, communicating with the client, sending status reports, managing change, managing risks, having peer reviews, etc.

CLOSURE PHASE

The closure phase consists of obtaining the project sign off. Finally, project retrospection is carried out covering best practices, tools or re-usable components developed, lessons learned, project performance against the quality goals, and associated learning.

DELIVERY MGT AND SDLC

The delivery management phases run parallel to the SDLC phases in a project. The two diagrams on the page show how the different delivery management phases are aligned to the SDLC phases of a standard development project and a maintenance project.

7. METRICS

SOFTWARE METRICS

Measurements act as indicators of progress and are understood by everyone irrespective of experience level and technology background.

LEARNING OBJECTIVES

At the end of this session, you will be able to:
Explain what software metrics are
Describe goal setting and actual analysis
Learn about the importance of software metrics

MEASUREMENTS

One cannot chase a target without knowing it. Nor can one know about success, and achievements, without defining success in measurable terms. And on the other hand, one cannot also accurately gauge the degree of danger one is in without appropriate figures. Measurement is an essential part of life, and we cannot control what we cannot measure.

NEED FOR MEASUREMENTS

The performance of the project has to be measured so that it can be kept under control. A project works toward a goal the measurements along the way compare the actual results of the project against the projections to determine whether the project is on track or deviating from the goal.

WHAT DO WE MEASURE

In the manufacturing industry, we know we have to measure the length, breadth, height, or other tangible specifications of the product. But when the generated products are formless as in the case of software, what does one measure? The process involved in software development is measured along with the final software product.

CONCEPT OF SW METRICS

The software is measured using software metrics - "Quantitative measures of specific desirable attributes of software". To arrive at metrics, some basic measures such as schedule and effort are collected and combined to produce meaningful indicators of the progress of software development. For example, to find out how productive a team is, the size of the software project and effort required by the team to do the project are considered as the base measures. Next, we divide size by effort to find the productivity metric for the team.

BASE MEASURES

Some examples of base measures that are considered for measuring the software process and product are schedule, effort, size, and defect. These measures are combined in different ways to form the different software metrics. The following page will give you some examples on process metrics for application development projects.

EXAMPLES OF METRICS –PROCESS METRICS

The metric schedule variance is got from the measure schedule, by comparing the actual with the planned. It is a measure of whether or not the project is meeting the planned dates for the start and finish of modules. Similarly, the metrics effort variance is got from the measure effort - This is a measure of whether the planned effort is matching the actual effort or not. We can also know about the load factor, which is a measure of whether the people in the project are adequately, lightly, or heavily loaded. All these are examples of process metrics. They indicate whether the process of producing software products is adequate or not. Some of the other process metrics are review efficiency and requirement stability. In the next page, you will see some examples of product metrics in application development projects.

EXAMPLES OF METRICS – PRODUCT METRICS

The metric defect density is got by putting defects and size together. This gives an indication of how robust the product is. This is an example of a product metric. Product metrics indicate whether the quality of the produced software product is adequate or not. Other product metrics are maintainability, reliability, and availability.

EXAMPLES OF METRICS – AVM PROJECTS

Similarly, in AVM projects, there are different measurements that evolve into metrics.
For Example:
Percentage of requests meeting service-level agreements
Percentage of fixes without escalation

HOW DO WE SET GOALS

You have seen measurements and metrics. However, metrics and measurements are meaningful only if we have a goal against which we compare the metrics to find out whether or not the project is on the right path. How are goals set? The business objectives of the organization are laid down. One example of a business objective may be to meet delivery commitments on time. These business objectives are then translated to one or many process objectives. Following the previous example, the process objective aligned to meeting delivery commitments on time may be translated to having less than 5.86 percent variation from the schedule during the process of coding and unit testing. These process objectives are mapped to sub processes, which make up the process. In our case, the process of coding and unit testing is made up of coding, code review, and unit testing.

After the sub processes are identified, metrics are attached to the sub processes, which can be used to measure the performance in the sub processes. In this example, coding is mapped to the coding schedule variance, code review to the code review schedule variance, and unit testing to the unit testing schedule variance. Finally, a goal is set for each of these metrics to ensure that the process objective is met. Here, for example, the goal for the coding schedule variance is 1.65 percent, the code review schedule variance is 0.33 percent, and the unit testing schedule variance is 1.32 percent.

SOME OF OUR BUSINESS OBJECTIVES

Here are some of Cognizant's business objectives:
In terms of finance - Improve profitability or reduce cost overruns
Through process - Improve productivity, Meet on-time delivery commitment, Reduce development cycle time, Improve product quality, Reduce cost of quality, Reduce rework
With respect to customer - Improve customer satisfaction index, and
With regard to people - Improving training effectiveness, Improving internal satisfaction of the associates

Business objectives are at the organization level and will span across projects and support groups.

HOW DO WE SET GOALS

Along with the business goals, a set of metrics derived from the data collected from past projects as well as data from industry figures are set as organizational baselines. These baselines are used as a basis to arrive at project-specific quality goals. They give an indication of the capability of the organization. The organizational baseline values for the metrics are documented in the OLBM or organizational-level benchmarks, which contain the mandatory and optional metrics for each type of project, the goals associated with them, and the upper and lower control limits for each metric. Each project sets its goals by either following the goals laid down in the OLBM or setting their own goals as appropriate for the specific project characteristics.

Along with goal setting, the projects have to formalize the data-collection tools to collect the actual measures for the project for the metrics to be computed and compared to the goals. The tools that are generally used for data collection are Prolite, e-Tracker, Time sheet, Defect logs, and Project plan.

USE OF METRICS – PROJECTS

How are the metrics used? Periodically, metrics are collected and analyzed. As the metrics are analyzed, one can get a quantitative idea of whether the project is proceeding along the right track or not. If a metric deviates significantly from the goal so as to come

close to or cross the control limits, it indicates something is wrong in the process. This necessitates corrective action. For example, too much schedule or effort variance may mean that our initial estimates were wrong and may force us to revise our estimates for the project.

USE OF METRICS – ORGANISATION

Apart from the projects, the metrics collected across the organization are analyzed by a group called the metrics council and the organization's baselines are modified periodically based on the analysis. This revision of baselines indicates a change in the performance capability of the organization.