IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 8, Issue 6 (Jan. - Feb. 2013), PP 20-35
www.iosrjournals.org

Towards Large Scale Software Project Development and Management

Tanmaya Kumar Das[1], Dillip Kumar Mahapatra[2], Gopa Krishna Pradhan[3]
[1](Department of CS&E, CVRCE, Bhubaneswar, Odisha)
[2](Department of IT, Krupajal Engineering College, Bhubaneswar, Odisha)
[3](Ex-Professor, Department of CS&E, SOA University, Bhubaneswar, Odisha)

ABSTRACT: *These days, a lot of companies all over the world have interest in large scale Software Development in a distributive manner. The question is how does distributed development works? There are stakeholders from different national and organizational cultures and time zones are involved in developing software. Today, it is not possible to meet members of software development teams regular in a face-to- face interaction, especially stakeholders from different national and organizational cultures. People work together in development projects to accomplish project goals from different geographical locations. It must be overcome various barriers. Beginning with the geographical dispersion of the globally distributed teams, the different time zones and the various cultures and languages, there often communication problems can occur. This paper discusses some key principles for managing large scale software project development.*
Keywords *- Globalization, Project Management, LSDLC, Risk Certification, Quality Control*

I. INTRODUCTION

As globalization is driving organizations to become more and more distributed, multi-site development is becoming a norm. With the increasing globalization in this industry, it is necessary to better prepare software development projects to manage work in distributed environments(Ref. 07).

Several difficulties exist in preparing and initiating a globally distributed project. As most distributed projects underestimate the required time and resources needed for a successful project ramp-up, the projects tend to be initiated with a lack of proper planning and preparation. However it is necessary to address how the early-life of globally distributed software development projects should be managed, with an aim of finding successful practices for preparing and initiating distributed projects.

Through globalization, it has not come only to a worldwide internationalization in trading, finance, business, etc., but software (engineering) is also increasingly developed around the world. The reason is that software plays a vital role for business success. This trend, of a Large Scale Software Development (LSSD) has been speeded up by several different aspects and factors:

● computers in general and software (development) in particular have become more and more important in the last several decades
● today, organizations are faced with a global competition, so they have the need to develop software 24 hours a day ("round-the-clock") independent of time zones in order to improve time-to-market
● the need to access on qualified personal and a higher flexibility with the deployment of the personal used in the project in order to react on the prevailing skilled worker shortage.
● the business advantages of the presence on regional markets and the proximity to the customers in order to handle their individual characteristics (e.g. cultural aspects).
● Furthermore there is a high potential of cost advantages when companies use globally distributed software developer teams (e.g. shifting of software development activities in low-wage countries).
● efficiency enhancement in order to better survive in competition and fast building of virtual teams to exploit market opportunities.

II. CRITICAL SUCCESS FACTORS FOR LARGE SCALE SOFTWARE DEVELOPMENT

What can a company do to meet these challenges and realize the benefits of global development and delivery? As highlighted in Figure 3, we suggest five critical factors are needed for success:

Figure 1: LSSD success factors

2.1 Ensure coordination and oversight for all sites:

It's important to "think global" from the start by planning how the organization will support the delivery model we want. Establish how roles and responsibilities will be distributed across the sites, including where the management and decision-making will reside. Identify the tools the teams at each site will use and determine the infrastructure required to enable their use and any integration with tools at other sites. Visibility into distant teams, especially outsourced teams, can be a particular challenge -- how can we be sure of the current actual status, with a holistic view across the entire project and deliverables? How can we reliably measure and compare between components or projects? Compare apples to apples, using consistent measures (as much as possible) for every team. Identify up front what we will measure to track progress and status, and work with the teams to ensure they can produce the needed metrics. Select discrete units we can measure objectively. For example, avoid measuring artifacts as "50 percent complete"; rather, indicate they are either complete or incomplete. Where possible, collect data directly from development tools, so we can eliminate human error and subjective bias. (In some cultures, team members may be more reluctant to voice bad news.)

2.2 Establish Well-Defined And Consistent Workflows:

To avoid misunderstandings about how work should be done, define, document, and educate the team on consistent processes that will be followed across all sites. It is probably unrealistic to expect all teams will follow the same process; different tools and technologies, different business needs, and inherited process and culture from acquisitions lead to diverse working styles and methodologies. An organization may have teams following waterfall, iterative, and agile methods, depending on their circumstances. Ideally, all teams in a single project would follow the same process, but in a global supply chain scenario, even those teams may be decoupled, contributing components with some level of autonomy.

From the perspective of the overall software organization, it becomes most important to define the bounds or parameters that all processes must meet so that consistent touch points and deliverables across and between teams will be established. Define the artifacts to be produced, the metrics to be tracked, the measures of success, and the mechanisms for providing oversight and governance (Ref. 03).

Within a given team, define, document, and educate all members on the processes that will be followed. If exceptions are necessary (e.g., due to access restrictions or tool differences), clearly document where they apply.

Clearly define authority and responsibilities: Who does what, what is delivered to whom, and what standards are to be met. We may want to use the concept of a "**RACI**" matrix to determine who is *Responsible, Accountable, Consulted, or Informed* for each deliverable or process step.

Handoffs between teams or sites often pose problems. Using terms as concrete as possible, processes should clearly specify the deliverables required at each transition and the expected inputs and outputs. For example, help ensure clarity by providing templates for required artifacts and defining control gates with objective criteria for entrance and exit. Clearly specifying handoff deliverables and content can define a common ground and ensure both parties get what is needed regardless of steps in between. This can be especially useful when working with outsourcing providers who may follow their own internal processes.

Consider how to ensure change tracking, traceability, and accountability across the lifecycle, so we can validate if and when a particular change was implemented and what impact it had, or will have, on other tasks or deliverables.

Where possible, automate the processes and artifact workflows and enforce them in the tools themselves. Tools should make it easier to follow the process rather than to deviate from it, acting as the proverbial "carrot" to encourage compliance, rather than as the "stick." By reducing manual steps, we both

reduce the opportunity for errors and save time and effort for the team members. Automation can also help overcome delays due to time zone differences. For example, having an automated build-and-build verification process can enable a team in India to complete and test a defect fix, without waiting for the build team in the United States to come back to work. Using the tools to enact process can also ease tool and process adoption and help ensure process improvement as teams learn how to work more effectively both within their team and across teams.

2.3 Manage The Inventory And Information:

Manage the assets and artifacts. Determine who needs access to which artifacts and ensure they have the required access. This includes considerations for network access and availability of the assets over WAN or Web, security permissions to perform the appropriate actions, and the simple ability to locate the artifacts (and the correct version of the artifacts).

Provide traceability between related artifacts where possible, so those further "downstream" in the process are able to understand the context of the artifacts they are using. For example, enabling traceability between defects and test cases to requirements and to use case models can help those executing the tests understand what results they should be seeing and why, and whether tests adequately cover the desired functionality(Ref. 01).

Use a versioning mechanism for the artifacts, tracking and tracing changes, and ensuring traceability between the correct versions of different artifacts. Take a snapshot or "baseline" of the assets at defined intervals, such as the end of each iteration or milestone. Each baseline includes a complete set of associated versioned artifacts that can be treated as a unit or single asset, shared between teams, and used as a reference point. Consider an asset management tool as a way to store and manage these aggregations of related and versioned artifacts and make them available for reuse.

2.4 Clear And Accessible Communication:

Clear and effective communication could be the most important of the success factors; it is certainly key to building any solid team. Successful projects encourage collaboration and promote team building and social networking. In a collocated team, collaboration can happen informally in the hallway or "at the water cooler," as well as in face-to-face meetings. With distributed teams, we must work harder to keep everyone in the loop.

Consider a communications plan that defines how we'll keep all team members informed and working toward common, shared goals. Communicate frequently and openly and work to build trust between team members.

Take advantage of any and all available collaboration mechanisms. Many of the standard mechanisms are well-known and widely used: e-mail, teleconferences, and various notes or documentation facilities built into development tools. Most of us are also now familiar with chat and instant messaging, Web conferences, and screen sharing. Wikis and team blogs are other mechanisms whose adoption has been spreading recently. Establish a "virtual water cooler" by way of instant messaging, team rooms, or other collaboration mechanisms (Ref. 09).

Trust can be hard to establish in a distributed team, where interpersonal contact may be limited. Create online profiles of team members in a central location, including pictures and even audio clips, so others can connect faces and other personal information to the name and voice. If budget allows, periodic face-to-face meetings provide opportunity to establish relationships with the benefit of nonverbal communication (although note that cultural differences can also affect these nonverbal modes of communication).

Fig. 2: Challenging Scenario of LSSD project

It goes without saying that we should schedule team meetings with consideration for the different time zones of the teams' locations. In cases where no time will conveniently work for all teams, schedule meetings at

alternating times, allowing teams to take turns attending at convenient or inconvenient hours. When scheduling meetings and milestones, we must also consider the holiday schedules that vary across the different countries and cultures. Scheduling a major milestone when one region has a month of vacation or a significant political or religious occasion is likely to cause schedule delays and potential team conflict.

Where possible, collaborate in the context of the object or artifact under discussion. Use URL references to share content via e-mail or chat. Annotate or make comments directly on the artifact, and preserve comments with the artifact for future reference. Keep clear records of communications and decisions, so those who were not involved know what was decided and can understand the rationale. For team members in other time zones, this can also save a lot of time in round-trip queries, where an entire workday may elapse before a question on a particular design decision or rationale can be answered by someone on the other side of the globe.

Textual records can also help to address language barriers. Those team members, who are challenged by spoken language can read (and reread if necessary) the written documents in order to better understand the decisions and context. Similarly, it may be helpful to provide text backups to vocal communications; for example, shared charts, text captioning, and chat "back-channel" in a teleconference. (Note that such backups also help those with hearing difficulties.)

Be aware of the diversity across the organization and the differences in culture and work habits across sites. For example, team members in some cultures may hesitate to question superiors; others may be more confrontational and challenge authority. Understand these differences and ensure all team members understand them as well, so we can adapt and mesh working styles between teams. Some companies offer resources in diversity or cultural training; in other cases, we may need to learn from experience or ask others who have worked with similar teams.

2.5 Establish A Flexible, Adaptable IT Infrastructure And Architecture

The IT infrastructure, including networks and development tools, should enable us to support a variety of distributed organization scenarios. Ensure the tools and platforms we select provide the capability and flexibility we need to support the topologies we require, whether they are replicated, hub-and-spoke, centralized, or some combination. Specific considerations include (Ref. 03):

Client access to servers via LAN, WAN, or Web, Security and the ability to restrict access to artifacts as needed, Ability to interact and integrate with other tools and platforms in a heterogeneous operating environment, Support for team collaboration, Ability to support necessary languages and code pages etc.

This paper consists of a set of successful practices and recommendations for preparing and initiating global software development projects and the principles towards efficient management of large scale software project (LSSPD).

III. PROJECT MANAGEMENT

Company organizations are changing to reflect these new approaches to product delivery. Many global organizations initially assigned product or project ownership by locality, making each location or branch responsible for delivering its own application or project, and sites worked independently (Ref. 14).

As companies expand their presence around the world, multiple sites become a "team of teams" contributing to a global delivery chain. Each team may own a module or component they deliver upstream for integration with components from other locations or companies, culminating in a final application or product. Teams may belong to the same organization, division, or company, or to different ones. In global organizational distribution, where multiple sites collaborate on a single component to be delivered in the chain. Sites may be very large or, as a result of workforce mobility, as small as a single individual.

In many cases, distribution is due in large part to mergers and acquisitions. In industries such as telecommunications, technology, and financial services, companies acquire market share, technologies, or market entry by acquisition, rather than by internal development. The diverse components of the new, merged business must then integrate into an effective organization and supply chain.

Project management is, in fact, shorthand for project, program and portfolio management. And more companies are clearly seeing the payoff from investing time, money and resources to build organizational project management expertise: lower costs, greater efficiencies, improved customer and stakeholder satisfaction, and greater competitive advantage.

An Economist Intelligence report showed that 80 percent of global executives believed having project management as a core competency helped them remain competitive during the recession.

“The delivery of business outcomes is realized through the success of projects, and in essence that is the way that project management strategies drive organizational success,”

It has also been surveyed that nearly 60 percent of senior executives said building a strong project management discipline is a top-three priority for their companies as they look to the future.

3.1 Why Project Management Matters?

Leading organizations across sectors and geographic borders have been steadily embracing project management as a way to control spending and improve project results. When the recession began, this practice became even more important. Executives discovered that adhering to project management methods and strategies reduced risks, cut costs and improved success rates—all vital to surviving the economic crisis. More than half of the executives in the Economist Intelligence Unit report said following a project management practices became more important since the recession began.

Companies are also discovering that as their project management strategy matures, the business value derived from it also increases. To increase that value and ensure strategic alignment across the project portfolio, executives at many global organizations are creating formal project management offices (PMOs).

3.2 Project Management and the Competitive Advantage

Implementing project management across the organization helps create a strategic value chain that gives companies an edge on their competitors, particularly in high-risk sectors and markets. Being able to deliver projects on time and within budget often determines whether a company will get the next job or whether its new product hits the market. Ninety percent of global senior executives ranked project management methods as either critical or somewhat important to their ability to deliver successful projects and remain competitive, according to the Economist Intelligence Unit survey.

"A strong project management discipline brings exceptional value to the business, because when there is a demand for a product, we are the ones who deliver it,"

IV. LARGE SCALE SOFTWARE DEVELOPMENT LIFE CYCLE

The System Development Life Cycle framework provides a sequence of activities for system designers and developers to follow. It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.

A Systems Development Life Cycle (SDLC) adheres to important phases that are essential for developers, such as planning, analysis, design, and implementation, and are explained in the section below.A number of system development life cycle (SDLC) models have been created: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, and synchronize and stabilize. The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next. These stages can be characterized and divided up in different ways, including the following (Ref. 07).

- Project planning, feasibility study: Establishes a high-level view of the intended project and determines its goals.
- Systems analysis, requirements definition: Refines project goals into defined functions and operation of the intended application. Analyzes end-user information needs.
- Systems design: Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.
- Implementation: The real code is written here.
- Integration and testing: Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
- Acceptance, installation, deployment: The final stage of initial development, where the software is put into production and runs actual business.
- Maintenance: What happens during the rest of the software's life: changes, correction, additions, and moves to a different computing platform and more? This, the least glamorous and perhaps most important step of all, goes on seemingly forever.

In the following example (see picture) these stage of the Systems Development Life Cycle are divided in ten steps from definition to creation and modification of IT work products:

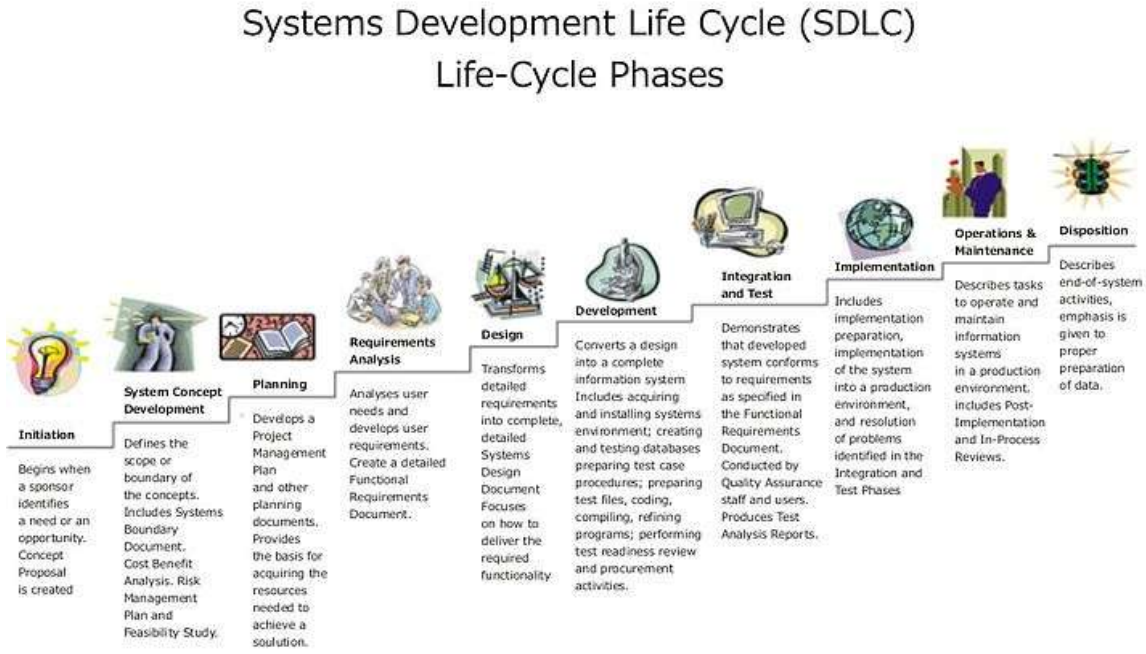

Fig. 3 Large Scale Software Development Life Cycle Phases

The tenth phase occurs when the system is disposed of and the task performed is either eliminated or transferred to other systems. The tasks and work products for each phase are described in subsequent chapters. Not every project will require that the phases be sequentially executed. However, the phases are interdependent. Depending upon the size and complexity of the project, phases may be combined or may overlap.

System analysis:

The goal of system analysis is to determine where the problem is in an attempt to fix the system. This step involves breaking down the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

Requirements analysis sometimes requires individuals/teams from client as well as service provider sides to get detailed and accurate requirements; often there has to be a lot of communication to and from to understand these requirements. Requirement gathering is the most crucial aspect as many times communication gaps arise in this phase and this leads to validation errors and bugs in the software program.

Design:

In systems design the design functions and operations are described in detail, including screen layouts, business rules, process diagrams and other documentation. The output of this stage will describe the new system as a collection of modules or subsystems.

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts.

Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input design.

Implementation:

Modular and subsystem programming code will be accomplished during this stage. Unit testing and module testing are done in this stage by the developers. This stage is intermingled with the next in that individual modules will need testing before integration to the main project.

Testing:

The code is tested at various levels in software testing. Unit, system and user acceptance testings are often performed. This is a grey area as many different opinions exist as to what the stages of testing are and how much if any iteration occurs. Iteration is not generally part of the waterfall model, but usually some occur at this stage. In the testing the whole system is test one by one

Following are the types of testing:

- Defect testing
- Path testing
- Data set testing
- Unit testing
- System testing
- Integration testing
- Black box testing
- White box testing
- Regression testing
- Automation testing
- User acceptance testing
- Performance testing

Operations and maintenance:

The deployment of the system includes changes and enhancements before the decommissioning or sunset of the system. Maintaining the system is an important aspect of SDLC. As key personnel change positions in the organization, new changes will be implemented, which will require system updates.

Management and control:

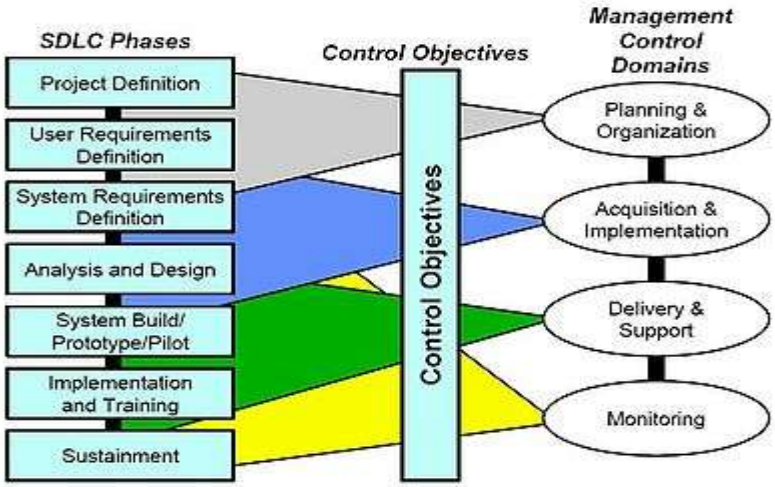

Fig. 4 . LSDLC Phases Related to Management Controls.

The Systems Development Life Cycle (SDLC) phases serve as a programmatic guide to project activity and provide a flexible but consistent way to conduct projects to a depth matching the scope of the project. Each of the SDLC phase objectives are described in this section with key deliverables, a description of recommended tasks, and a summary of related control objectives for effective management. It is critical for the project manager to establish and monitor control objectives during each SDLC phase while executing projects. Control objectives help to provide a clear statement of the desired result or purpose and should be used throughout the entire SDLC process. Control objectives can be grouped into major categories (Domains), and relate to the SDLC phases as shown in the figure.

To manage and control any SDLC initiative, each project will be required to establish some degree of a Work Breakdown Structure (WBS) to capture and schedule the work necessary to complete the project. The WBS and all programmatic material should be kept in the "Project Description" section of the project notebook. The WBS format is mostly left to the project manager to establish in a way that best describes the project work. There are some key areas that must be defined in the WBS as part of the SDLC policy. The following diagram describes three key areas that will be addressed in the WBS in a manner established by the project manager.

Work breakdown structured organization

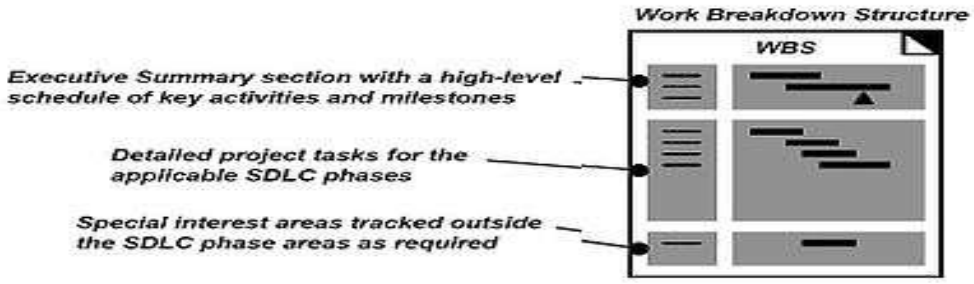

Fig 5. . Work Breakdown Structure.

` The upper section of the Work Breakdown Structure (WBS) should identify the major phases and milestones of the project in a summary fashion. In addition, the upper section should provide an overview of the full scope and timeline of the project and will be part of the initial project description effort leading to project approval. The middle section of the WBS is based on the seven Systems Development Life Cycle (SDLC) phases as a guide for WBS task development. The WBS elements should consist of milestones and "tasks" as opposed to "activities" and have a definitive period (usually two weeks or more). Each task must have a measurable output (ex. document, decision, or analysis). A WBS task may rely on one or more activities (e.g. software engineering, systems engineering) and may require close coordination with other tasks, either internal or external to the project. Any part of the project needing support from contractors should have a Statement of work (SOW) written to include the appropriate tasks from the SDLC phases. The development of a SOW does not occur during a specific phase of SDLC but is developed to include the work from the SDLC process that may be conducted by external resources such as contractors.

Baselines in the LSDLC:
Baselines are an important part of the Systems Development Life Cycle (SDLC). These baselines are established after four of the five phases of the SDLC and are critical to the iterative nature of the model. Each baseline is considered as a milestone in the SDLC.

- Functional Baseline: established after the conceptual design phase.
- Allocated Baseline: established after the preliminary design phase.
- Product Baseline: established after the detail design and development phase.
- Updated Product Baseline: established after the production construction phase.

Complementary to LSDLC:
Complementary Software development methods to Systems Development Life Cycle (SDLC) are: Software Prototyping, Joint Applications Design (JAD), Rapid Application Development (RAD), Extreme Programming (XP); extension of earlier work in Prototyping and RAD, Open Source Development, End-user development, Object Oriented Programming etc.

V. PHASES OF PROJECT MANAGEMENT

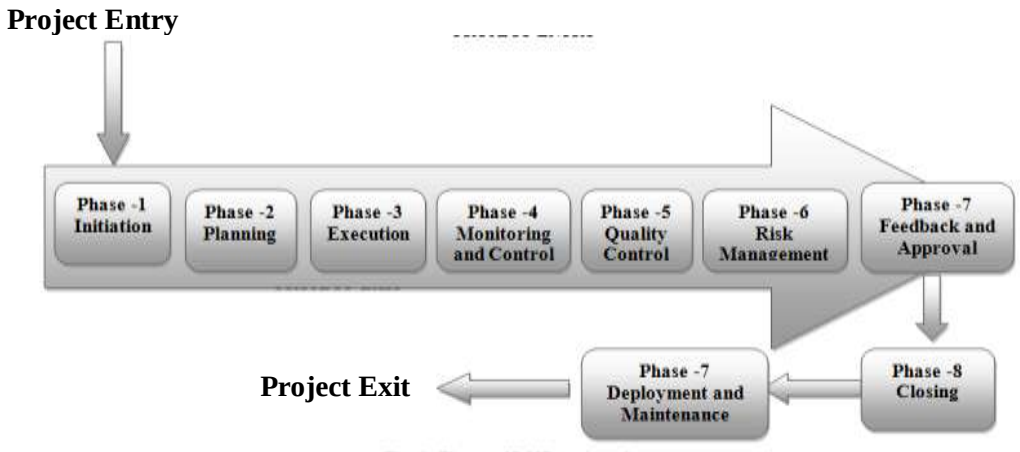

Fig. 6 : Phases of LSSD project phases

5.1 The Initiation Stage in Project Management

The first phase of project management is the Initiation phase. It's during this initial time that the project goal is established. During Phase 1, if a project manager has been assigned, this person works with the involved parties, otherwise known as the project stakeholders, to fully determine how to measure the success of the project once all work is complete (Ref. 10).

This allows the project manager and project stakeholders *(these are the people with a vested interest in the project, and often the ones shelling out the money to make it happen)* to agree on the project scope. The project scope will include project goals, budget, timelines and any other variables that can be used for success measurement once we reach the final phase, Closing.

There aren't a whole lot of software programs that can help we during the Initiation phase, aside from a word processor to create the Project Charter. During the Initiation phase we are not making a list of the things that need to happen to accomplish the total project goal, but rather a list of end-results. For example, "Digitizing two hours of video" is a task, but "offer streaming videos of lectures to my class" is a goal.

Some project managers will disagree with this approach arguing that the Initiation phase is exactly where we'll define the project's tasks and milestones. However, in my experience, the Initiation phase is about clearly defining the target so we'll know when the objectives have been met. We can use the next phase, planning, to address the project details.

Before anything else, it's important to be clear on a project's main goals and expectations. Get started on the right foot with a proper initiation.

5.2 Project Planning Phase

The *Project Planning Phase* is the second phase in the *project life cycle*. It involves creating of a set of plans to guide the team through the execution and closure phases of the project.

The plans created during this phase will help we to manage time, cost, quality, change, risk and issues. They will also help we manage staff and external suppliers, to ensure that we deliver the project on time and within budget.

There are 10 Project Planning steps we need to take to complete the Project *Planning Phase* efficiently. These steps and the templates needed to perform them are discussed below.

Create a Project Plan: A Project Plan sets out the phases, activities and tasks needed to deliver a project. The timeframes required delivering the project, along with the resources and milestones are also shown in the Project Plan. Using this *Project Plan* Template, we can quickly and easily create a comprehensive Project Management Plan for our project, as it already lists the commonly used tasks needed to complete projects from start to finish.

Project Plan Template**:** This will help us to quickly and easily create a Project Plan for the project. We can use it to create the own customized project management plan for delivering the project on time and under budget. If we want to create a Project Plan for our project within a few easy steps, then this project plan template will tell how to do it.

By using this Project Plan template, we can:

- Identify all of the phases, activities and tasks
- Sum up the effort needed to complete those tasks
- Document all of the project inter-dependencies
- List the planning assumptions and constraints
- Create a detailed project planning schedule

This *Project Plan Template* will also help us to:

- Define the project scope & milestones
- Identify the Work Breakdown Structure
- Set and agree the target delivery dates
- Monitor and control the allocation of resource
- Report on the progress of the project, to the sponsor

The Project Plan is the most important document in the project, as it provides the Project Manager with a roadmap ahead, and it tells them during the journey whether they are on-track. Using this Project Plan template, we can create a comprehensive project management plan for the project today.

Create a Resource Plan**:** A Resource Plan summarizes the level of resources needed to complete a project. A properly documented Resource Plan will specify the exact quantities of labor, equipment and materials needed to complete the project. This Resource Planning template also helps we gain approval from the Sponsor, ensuring their buy-in.

Project Resource Management Plan :It helps us to identify all of the resources required to complete the project successfully. Using this *Resource Plan*, we will be able to identify the quantity of labor, equipment and materials needed to deliver the project. We will then create a resource schedule, which enables us to plan the consumption of each type of resource, so that we know that we will have enough resources to complete the project.

This Resource Plan template will help us identify the:

- Types of labor required for the project
- Roles and key responsibilities for each labor type
- Number of people required to fill each role
- Items of equipment to be used and their purposes
- Types and quantities of equipment needed
- Total amount of materials needed
- Plan the dates for using or consuming these resources
- Identify the amount of resource required per project activity
- Create a detailed resource utilization schedule

By purchasing this resource planning template, we can schedule the resources needed to complete the project successfully.

Create a Financial Plan: A Financial Plan identifies the Project Finance (i.e. money) needed to meet specific objectives. The Financial Plan defines all of the various types of expenses that a project will incur (labor, equipment, materials and administration costs) along with an estimation of the value of each expense. The Financial Plan also summarizes the total expense to be incurred across the project and this total expense becomes the project budget. As part of the Financial Planning exercise, a schedule is provided which states the amount of money needed during each stage of the project.

Financial Planning Template: This will help us to quickly and easily create a Financial Plan for the project. A *Financial Plan* enables we to set a "budget", against which we measure the expenditure. To deliver we project "within budget", we need to produce the project deliverables at a total cost which does not exceed that stated in the budget. Using this financial plan template, we can create a detailed budget against which to measure the success of the project.

This Financial Plan template will help we to identify the:

- Types of labor costs to be incurred during the project
- Items of equipment needed to deliver the project
- Various materials needed by the project
- Unit costs for labor, equipment and materials
- Other costs types such as administration
- Amount of contingency needed

We can then use the Financial Plan template to create a budget by:

- Calculating the total cost involved in completing the project
- Identifying the total cost of each project activity
- Creating a schedule of expenses

Creating a project budget is an extremely important part in any project, as it gives we a *goal post* to aim for. This Financial Plan will help we meet that goal post, by giving we a clear process and template for creating a budget for the project.

Create a Quality Plan: A Quality Plan helps we schedule all of the tasks needed to make sure that the project meets the needs of the customer. It comprises two parts; the *Quality Assurance Plan* lists the independent reviews needed and the *Quality Control Plan* lists the internal reviews needed to meet the quality targets. By using Quality Assurance and Quality Control techniques, we can create a comprehensive Quality Management Plan for the project.

It will help us to set quality targets for the project to ensure that the deliverables produced, meet the needs of the customer. We can then use it to schedule *quality control* and *quality assurance* activities, to assure the customer that the quality targets will be met.

We can use this Quality Plan to set quality targets by:

- Identifying the customers requirements
- Listing the project deliverables to be produced
- Setting quality criteria for these deliverables
- Defining quality standards for the deliverables

- Gaining the customers agreement with the targets set

We can then use this Quality Plan to monitor and control quality by:

- Identifying the quality control tasks needed to control quality
- Creating a *Quality Control Plan*, by scheduling the control activities
- Listing the quality assurance activities required to assure quality
- Building a *Quality Assurance Plan*, by creating an activity schedule

Quality Planning is a critical part of any project. It enables we to agree a set of quality targets with the customer. It then helps we to monitor and control the level of quality produced by the project, to ensure that we meet the quality targets set. By using this quality plan template, we can set quality targets and ensure that the project produces deliverables which meet the customers needs, thereby ensuring the success.

Create a Risk Plan: A Risk Plan helps we to foresee risks, identify actions to prevent them from occurring and reduce their impact should they eventuate. The Risk Management Plan is created as part of the Risk Planning process. It lists of all foreseeable risks, their ranking and priority, the preventative and contingent actions, along with a process for tracking them. This Risk Plan template will help we perform these steps quickly and easily.
Risk Management Plan helps to identify risks and implement a plan to reduce them. It provides a complete *risk management plan*, showing *how to take action to reduce risk in the project.* Using this risk plan, we can monitor and control risks effectively, increasing the chances of achieving success.

This Risk Planning template will help to

- Identify risks within the project
- Categorize and prioritize each risk
- Determine the likelihood of the risks occurring
- Identify the impact on the project if risk does occur
- Identify preventative actions to prevent the risk from occurring
- List contingent actions to reduce the impact, should the risk occur
- Schedule these actions within an acceptable timeframe
- Monitor the status of each risk throughout the project

Creating a Risk Management Plan is a critical step in any project, as it helps we to reduce the likelihood of risk from occurring. Regardless of the type of risk, we will be able to use this template to put in place processes and procedures for reducing the likelihood of risk occurring, thereby helping we to deliver the project successfully.

Create Acceptance Plan: An Acceptance Plan (also known as an "Acceptance Test Plan") is a schedule of tasks that are required to gain the customers acceptance that what we have produced is satisfactory. It is more than just a task list though. An Acceptance Plan is in fact an agreement between we and the customer, stating the acceptance tasks that will be undertaken at the end of the project to get their final approval. The Acceptance Plan includes a list of the deliverables, the acceptance test activities, the criteria and standards to be met, and the plan for their completion.
Acceptance Plan helps us to gain the customers acceptance for the deliverables produced by the project. Creating an Acceptance Plan (or 'Acceptance Test Plan') is an important part of any project, as it allows the customer to accept the deliverables we have produced for them. By using this acceptance plan template, we can gain customer acceptance for the deliverables, quickly and efficiently.
This Acceptance Plan template will help we gain acceptance, by:

- Creating a full list of all project deliverables
- Listing the criteria for gaining customer acceptance
- Putting in place, acceptance standards to be met
- Identifying the acceptance test methods
- Allocating acceptance test resources
- Scheduling acceptance reviews with the customer
- Gaining the final acceptance of the deliverables

By creating an Acceptance Plan for the projects, we'll boost the chances of success - as we will constantly produce deliverables which meet the customer's requirements. The Acceptance Plan template helps us to schedule customer acceptance tests to ensure that the deliverables meet the customers needs, every time.

Create a Communication Plan**:** A Communication Plan (or *Communications Plan*) describes how we intend to communicate the right messages to the right people at the right time. Within a Communication Plan, the communication goals, stakeholders and strategies, activities and timeframes are described. A Communication Plan helps we keep everyone informed so that we can communicate a consistent message to the target audience.

This Communication Plan template will help us to communicate the right information, to the right people, at the right time. It will also help we create a schedule of communications events to ensure that the stakeholders are always kept properly informed, ensuring their continued buy-in and support.
The template helps us to build Communication Plans *by:*

- Listing the communications stakeholders
- Defining each stakeholders communication needs
- Identifying the required communications events
- Determining the method and frequency of each event
- Allocating resource to communications events
- Building a communication event schedule

We can then use this Communication Plan template for:

- Monitoring the communications events completed
- Gaining feedback on communications events
- Improving communications processes

Communication Planning is an important part of any business. Using this template we can create a comprehensive Communications Plan for the project or team, helping keep the stakeholders properly informed at all times.

Create a Procurement Plan**:** A Procurement Plan defines the products and services that we will obtain from external suppliers. A good Procurement Plan will go one step further by describing the process we will go through to appoint those suppliers contractually. Whether we are embarking on a project procurement or organizational procurement planning exercise, the steps will be the same. First, define the items we need to procure. Next, define the process for acquiring those items. And finally, schedule the timeframes for delivery.
Procurement Plan helps we procure products and services from external suppliers. It provides we with a complete project procurement *plan template, to help we to quickly and easily create a Procurement Plan for the business.*
By planning the procurement carefully, we can ensure that we buy the right products for the business, at the right price.

This Procurement Plan helps to:

- Define the procurement requirements
- Identify all of the items we need to procure
- Create a sound financial justification for procuring them
- List all of the tasks involved in procuring the products
- Schedule those tasks by allocating timeframes and resources
- Create a robust project procurement process for the business

Procurement Planning is critical if we want to get the most out of the supplier relationships. By using this Procurement Plan template, we can quickly and easily define the procurement requirements, the method of procurement and the timeframes for delivery.

5.3 Perform a Phase Review**:** A Project Phase Review is completed at the end of each *project phase*. During this project management review, the reviewer completes a Phase Review Form describing the progress of the project to date and recommending whether or not it should continue to the next project phase. If approved, the next project phase can be commenced.
Project Phase *Review Form is completed at the end of the* Project Planning *phase to tell the sponsor whether the project has achieved its objectives to date.*
A *project management review* is completed and the results are documented on this Project Review Form. This form helps we conduct a Project Phase review quickly and easily.

The Project Phase Review Form states whether the*:*

- Project is under schedule and within budget
- Deliverables have been produced and approved
- Risks have been controlled and mitigated
- Issues have been resolved
- Project is on track
- Document the results of the Project Reviews
- Clearly communicate the progress of the project to the sponsor
- List any risks or issues which have impacted the project

- Show the sponsor the deliverables produced to date
- Seek approval to proceed to the next project phase

By implementing Project Phase Reviews, we are putting in place the necessary "check-points" to monitor and control the project, thereby ensuring its success.
The Project Planning Phase is often the most challenging phase for a Project Manager, as we need to make an educated guess of the staff, resources and equipment needed to complete the project. We may also need to plan the communications and procurement activities, as well as contract any 3rd party suppliers.
In short, we need to create a comprehensive suite of project plans which set out a clear *project roadmap* ahead. The Project Planning Template suite will help we to do this, by giving we a comprehensive collection of Project Planning templates. By using these project planning templates, we can quickly and easily plan the project. Learn more about the Project Planning Template suite now.

5.4 Project Execution:
During the Execution phase, the best-laid plan from Phase 2 - Planning is put to work. This is also a great time to use the project management tracking software to it's fullest extent. Project management software is a must during this time. It doesn't have to be fancy, or even expensive, but it does need to keep us on top of everything we thought would happen during this project and whether or not it is actually happening.
While the Planning phase can take a considerable amount of time, depending upon the project deliverables, the Execution phase can take as long as or longer than the Planning phase. This is also the time when we'll spend the bulk of the money and keep the resources busy "executing" the project plan.
During the Execution phase, the project manager spends a considerable amount of time in communication making sure the resources *(or people, equipment and materials)* are available to do their work and know what work needs to be completed.
There's quite a bit to this phase as a project manager as we work to juggle many aspects of the project. During this phase, we'll use all of the management skills to implement and manage cost and quality, risks and change, and several other factors.
This is also a great time to work to keep the project stakeholders informed of the project's progress.

5.5 Project Monitor and Control:
Project controls are, "the process of tracking, reviewing, and regulating the process to meet the performance objectives defined in the project management plan." That statement can mean implementing many different controls to reach the project goal.
Project monitoring entails supervision over the project execution phase to ensure that the project keeps to its time schedule and meets other parameters. The project development team resolves any issues or contingencies that come up, which may disrupt the project schedule.
Project controlling closely relates to the monitoring, and ensures that the project execution takes place with the agreed standards. In software projects, this includes validation, or ensuring that the codes developed work properly, and meet the accepted standards.
In order to set good project controls, we need to start at the beginning or at the project initiation phase. Our monitoring and controls should be based upon:

- Creating the Project Scope – Defining every aspect of the project at hand.
- Team & Task Structure – Deciding upon which teams will do certain tasks while developing an effective facilitation plan to ensure progress.
- Associated Risk & Risk Management Plan – Identify both acceptable and harmful risks, and prioritize and deal with risks in a risk management plan.
- Change Control Plan – Change is inevitable so what will the change control process be? How will we deal with the human element in the change control plan?
- Status Monitoring – Whether it's daily, weekly, in-person meetings or written status reports, we must have a way to grasp project progression.
- Effective Communication Plans set up –lines of communication that are efficient. We also need to include stakeholder or client communication.
- Budgets & Deadlines – Keep track of associated costs and deadlines.
- Evaluation & Analysis – Is every element completed correctly as described in the project scope?
- Corrective Plans – If through analysis, correction is needed, how will we plan for adjustments?
- Presentation Plan – Who will present the project and what sources, (both internal and external) will be required?

Each one of these controls implemented into the projects must be monitored. No monitoring really means no control. The need for project control in every task or challenge given to we grows ever important, especially in the competitive world of project management.

Project Change Control Management:

Project change control management is the careful commitment to ensuring that any change occurring within a project is a) monitored and b) kept from becoming a disaster. When we work with projects, quality improvement efforts, or even change management, knowing and understanding the core concepts related to project change control management is key. The thrust behind project change control management is the careful planning out of the project and the treating of the project as if it is a change management project. Change control is vital in ensuring that the project doesn't run away beyond what the project team can handle.
One of the core concepts related to project change control management understands the types of change that can occur during the project management process. Changes during the project may occur due to a variety of factors. These factors include:

- The client or customer requests a change
- The project manager requests a change
- In-house stakeholders request a change
- A risk comes to realization and requires a change
- Unforeseen circumstances force a change
- The project team creates a change
- The end users find that the deliverable or service does not meet their current needs

Quality Control Management:

One important sub-function of project control is quality control - or ensuring that the project adheres to the quality standards agreed in the project initiation phase. This role is crucial to catch errors, and to bring about an early resolution to these errors, before they escalate into major obstacles to smooth project execution.
The nature of quality control adopted by the project development team depends on the quality approach such as Six Sigma, Lean, 5S, Total Quality Management (TQM), or other methodologies adopted for the project. The benefits of Total Quality Management in particular, should not be dismissed.

Risk Management:

When considering project development team activities, the characteristics of good project risk management must be looked at. Risk management entails handling uncertainties, and relates closely to monitoring and control.

Process: Successful software process improvement requires the synergistic interaction of several elements. Begin with smart, trained, creative software engineers and managers, who can work together effectively. Consistent management leadership and expectations help grow a culture that shares a focus on quality, with honest appraisal of problem areas, clear improvement goals, and the use of metrics to track progress. Time must be provided for the team members to identify, pilot, and implement improved processes, with every team member becoming involved in the improvement effort over time. Finally, realize that most of software process improvement is based on thoughtful common sense, combined with a commitment to improve the way software engineering is performed in the organization.

Product: Product risk is the risk associated with the software or system, the possibility that software or system may fail to satisfy end user/customers expectations is known as product risk. There may be the possibility that the software or system does not have the functionality specified by the customer or the stakeholders which leads to unsatisfactory software. Unsatisfactory software has risk and has the possibility of failure which can cause major functional damage. Low quality software can have many problems like functionality, reliability, usability or performance.

Project: As we know that testing is an activity and so it is subject to risk which may endanger the project, so we can say that the risks associated with the testing activity which can endanger the test project cycle is known as project risk. In order to deal with project risks we need to apply concepts like identifying, prioritizing and managing the project risks.
Some of the risks associated with project are:

- Delay in the test build to test team and unavailability of test environment.
- Delay in fixing test environment due to lack of system administration.
- Delay in fixing defects by development team.
- Organizational problems which can be like shortage of staff. Required skills etc.
- Major changes in the SRS which invalidates the test cases and requires changes in the test case.

Identification of risks takes place early in the project, during the initiation phase, and developing strategies on how to manage such risks, takes place during the planning phase. Risk management during the time of project execution entails approaching the anticipated risks when they arise. Possible risks include the supply chain breaking down, shortage of human resources, or any other related factor.
The role of the project development team is in initiating the planned response to such risks. Another important role is maintaining a risk log that details the risks encountered and the response to it.

5.6 Evaluation and Certification:

When product development results in a candidate product release, the acquirer must determine, whether the product as built is acceptable for deployment into operation. Product acceptance includes three levels of product evaluation:
Verification: determining that the product is a correct solution to the problem as understood and expressed in customer acceptance criteria.
Validation: determining that the product properly addresses the customer's actual current and future needs.
Certification: determining that the product satisfies all organizational and statutory requirements concerning safety, security, and legal regulation applicable to its operational environment.

The goal of product evaluation is to encourage practices, regardless of method, that distinguish adequately between essential characteristics of a needed product and those that are only included for sufficient descriptive completeness to enable shared understanding. The best choice may be a mix of a more formal notation that supports precise specifications and an associated, less formal notation that is more anecdotal and overly detailed as an aid to understanding, with the formal notation being authoritative.

Pre-implementation: Pre-implementation is one of the essential phases for enabling successful implementation of every project in organizations. This becomes more vital when for large-scale projects which bring into play the whole efficiency of an organization.

Post- implementation: The product should be evaluated in terms of whether its observable behavior—augmented by documented evidence of its proper construction—satisfies identified customer needs. At this point, the acceptance criteria should be a clear representation of the customer's needs. This high level of confidence is due to the growing experience with the off-the-shelf product or to product evaluation efforts during product development. To establish acceptability of a product, product evaluation is performed progressively to independently verify and validate the product:

Validation and certification mainly concentrate on the Case of use, Level of utilization, Suitability of information format and Goal of the organization.

Taking feedback from a client at the end of each deliverable or agreed upon milestone, is another of the important project development team activities. The purpose behind this feedback is to ensure that the client remains satisfied with the progress, and to incorporate changes suggested by the client. One major issue that many project development teams face at this stage is scope creep, or requests from clients outside the original project charter, and project scope originally specified.
The extent of feedback depends on the nature of the project and project architecture. Extreme programming architecture for instance, caters to a constant involvement of the client, whereas traditional linear programming architecture, takes client feedback at the end of the execution phase. Satisfactory feedback leads to client approval, which denotes the end of the execution phase.

5.7 Termination:

Project termination takes place at the end of the project, on completion of the execution phase. The role of the project development team in the closing stage is integration of the various modules, tying up of the loose ends, improving on the user-friendly features of the project, and adding to the overall appearance, to make the project appear attractive to the customer.

5.8 Deployment and Maintenance:

Project deployment entails handing over the project to the customer, and providing training to ensure the customer can use the project seamlessly. A related activity is maintaining the project at the client's site for a year or so, to remove any bugs that may come up during real time running of software product.

VI. CONCLUSION

Large Scale Software Development (LSSD) are mostly adopted by many organizations with the intent of capitalizing on the many potential benefits like ;Ability to gain market share, technologies, or a foothold in a new area of business or geographical market via merger with or acquisition of another company. Greater flexibility and a variable staffing model, afforded both by outsourcing practices (that can be increased or decreased as needed) and by differing labor rates that allow we to find the right skills at the right price. Lower-cost labor that can offset budget cuts and cost reductions, as well as supplement existing teams so they can expand their focus. LSSD provides access to a broader set of skilled workers and ability to leverage outsourcing providers with specialized skill and experience to address new requirements or innovations in specific areas. In general, the ability to gain a competitive edge -- to increase speed and decrease costs -- by leveraging skilled but lower-cost resources, experienced outsourcers, and overlapping time zones. And, as more companies embrace global delivery, not doing so becomes a competitive disadvantage.

How ever, large scale software project development can be fraught with a number of issues like; misunderstood processes or mismatched processes between teams can lead to mistakes in work transfer, increased rework, and decreased productivity. By some estimates, productivity in a LSS project can drop up to 50 percent, with rework two to five times greater than for a collocated project, Communication issues can lead to misunderstandings, omissions, errors, and rework, Cultural issues, such as language barriers and differences in work customs or communication styles, can cause delays and affect working relationships, coordinating work across multiple sites and time zones is more time-consuming and costly than for a collocated project. visibility into and control of the development activities at all sites can be challenging, especially when collaborating with other companies or with teams in different time zones., Project metrics may be inconsistent or difficult to obtain from heterogeneous infrastructures, different processes, or company security boundaries, making it difficult to measure success, Political issues both within the company (organizations that fear losing work or resent the overhead of remote sites) and externally in the country or region, can lead to hidden agendas and conflicting goals, organizations may not share the same objectives, especially when reporting through different management chains or different companies, Concerns with security and IP protection, especially in outsourcing situations in countries where IP laws are more lax, can restrict infrastructure and organizational decisions, Infrastructures and development tools may vary widely due to mergers, acquisitions, and outsourcing. Even internally, many smaller teams are adopting lightweight tools, frequently from the open source domain and often to support new processes, such as agile development.

To address these limitations it is strongly required to propose an interoperable and service oriented environment for the efficient management of large scale software projects.

References

[1]. Rushby, J. Software Verification and System Assurance. Proc. 7th IEEE Int. Conf. on Software Engineering and Formal Methods. Hanoi, Vietnam. 2009.

[2] Sommerville, I. 'Designing for Recovery: New Challenges for Large-scale Complex IT Systems'.Keynote address, 8th IEEE Conference on Composition-based Software Systems. Madrid. 2008.

[3] http://sites.google.com/site/iansommerville/keynote-talks/DesigningForRecovery.pdf Sillitto, H. 'Design Principles for Ultra-Large Scale (ULS) Systems'. Proc. 20th INCOSE International Symposium, Chicago, July 2010.

[4] Cohen, S. A Software System Development Life Cycle Model for Improved Stakeholders' Communication andCollaboration. Int. J. of Computers, Communications & Control. Vol. V, No. 1, pp. 20-4. 2010.

[5] p. Sage and James D. Palmer, Software Systems Engineering (New York: John Wiley & Sons, 1990).

[6] B. Curtis, H. Krasner, and N. Iscoe. A field study of the software design process for large systems. Communications of the ACM, 31(11):1268–1287, 1988.

[7] Software Engineering Advice from Building Large-Scale Distributed Systems Jeff Dean, http://labs.google.com/people/jeff

[8] Project management,Ian Sommerville 2000 Software Engineering, 6th edition.

[9] Managing the Software Process, Watts S. Humphrey, SEI Series in Software Engineering, Addison-Wesley, August 1990.

[10]. Reference Model for Frameworks of Software Engineering Environments, European Computer Manufacturer's Association, TR/55, 1991.

[11] Applications of Metrics in Industry Handbook, a quantitative approach to software management, Centre for Systems and Software Engineering, South Bank University, London, 1992.

[12] www.ebookdust.com/.../software-project-management-bob-hughes.hml...

[13] Barry Boehm, USC, COCOMO/SCM Forum #17 Tutorial "Software Risk Management Overview and Recent Developments "October 22,2002 (boehm@sunset.usc.edu) (http://sunset.usc.edu)

[14] Project Management Institute, inc. PMBOK: A Guide to the Project Management Body of Knowledge, fourth edition.

[15] Boehm et al., "Spiral Development of Software-Intensive Systems of Systems". Proc. of International Conference on Software Engineering, USA, 2005

[16] R. Priklandnicki, J. Audy Nicolas, R. Evarito, "A Reference Model for Global Software Development: Findings from a Case Study.", IEEE International Conference on Global Software Engineering (ICGSE '06), 2006.

[17] Maier, M.W., 'Architecting Principles for System of Systems', Systems Engineering, 1(4), 1998,pp. 267-284.

[18] Northrop, L. et al. Ultra-Large-Scale Systems: The Software Challenge of the Future. Technical Report. Carnegie Mellon University Software Engineering Institute. 2006.