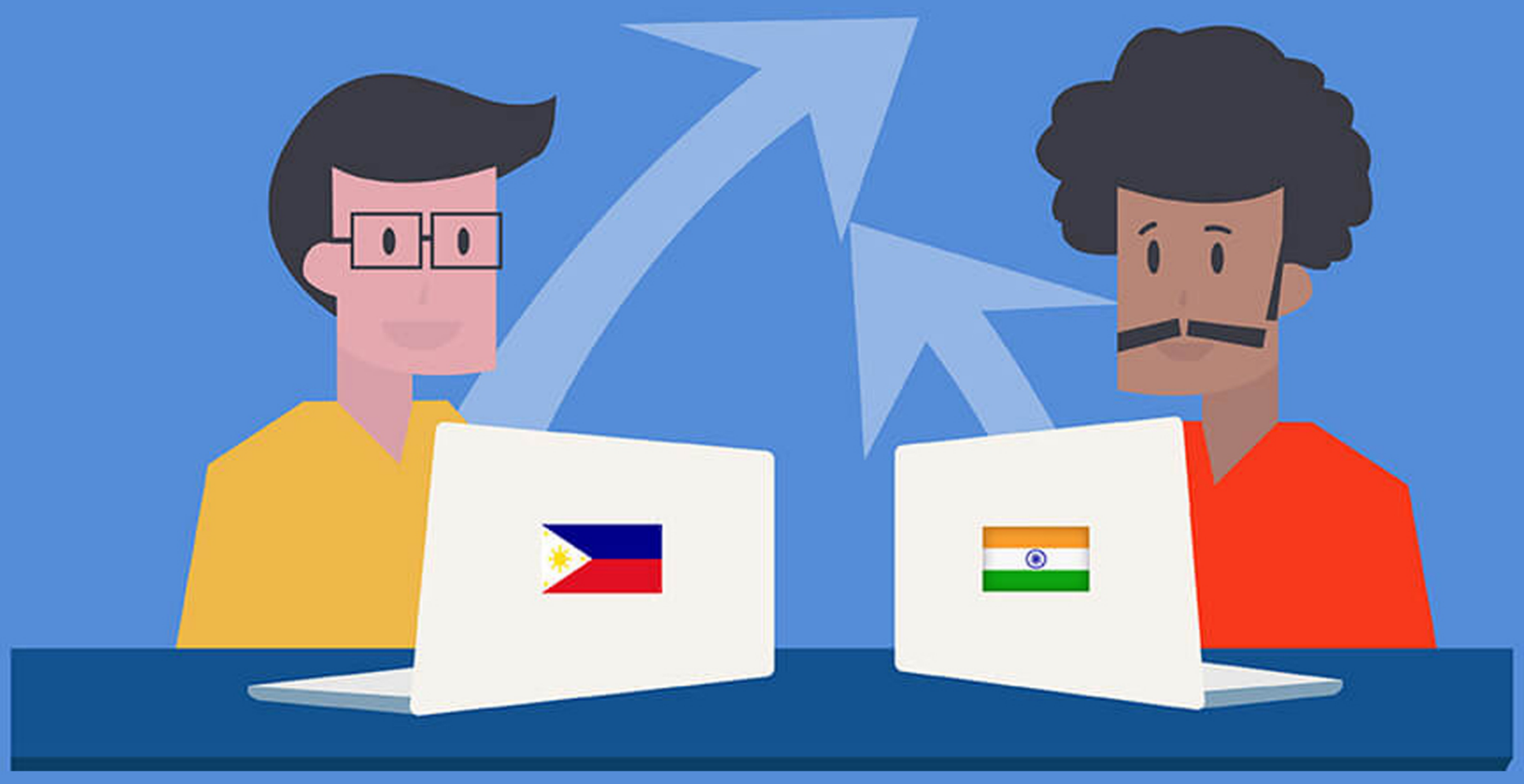
Foreignerds

# Software Development in an Outsourcing Environment



# Contents

# List of Figures

# Chapter 1

# Outsourcing

Outsourcing is the term that has replaced the traditional term for subcontracting. Proceeding further with this, Van Mieghem defines "sub-contracting as the acquisition of an item which the firm is capable of developing, whereas outsourcing involves the acquisition of an item which the firm is not capable of producing internally to a satisfactory level". This concept has become so popular in the field of Information technology that many of the companies have started to outsource many of their functions like Human Resource, Accounting, Business-Processes, IT Applications, Customer Services and many more. There are many benefits of Outsourcing apart from the cost reduction. According to Gildron, Rueda and McIvor ," the motivations to outsource are adopting an increasingly tactical profile as a search for the short term achievement of results in particular in terms of cost reduction, together with a more strategic perspective aiming to attain and maintain competitive advantage" [39].

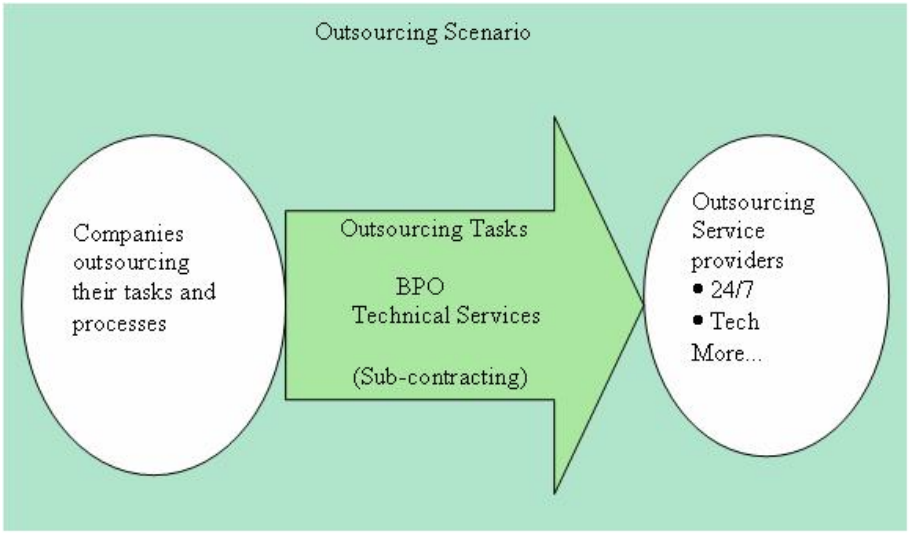



Figure 1.1: Outsourcing Scenario

Outsourcing [1] is the delegation of tasks for jobs from internal production to an external entity.

Outsourcing [14] is defined as the management and day to day execution of business functions by a third party service provider.

The transfer of components or large segments of an organization's internal IT infrastructure, staff, processes or applications to an external resource such as an application service provider. [4].

Although many definitions exist for the term outsourcing we would like to define it as "The internal work being assigned to an external service provider to reduce cost and also to get the work done more effectively and efficiently" The reason for defining this Outsourcing in most simple terms is because most of the companies in the current arena are trying to find a third-party service provider for performing their work easily and also cost effectively. The other reason for opting this is mainly because work done is more efficient; since the companies have the state of art technologies by which they can perform the work more fastly than the work done in-house.

Many synonyms exist for the term outsourcing like near shoring (i.e, Near Outsourcing) and off shoring (i.e, offshore outsourcing) . Near shoring refers to outsourcing within nearby territories and off shoring refers to outsourcing across borders or overseas in a separate country.

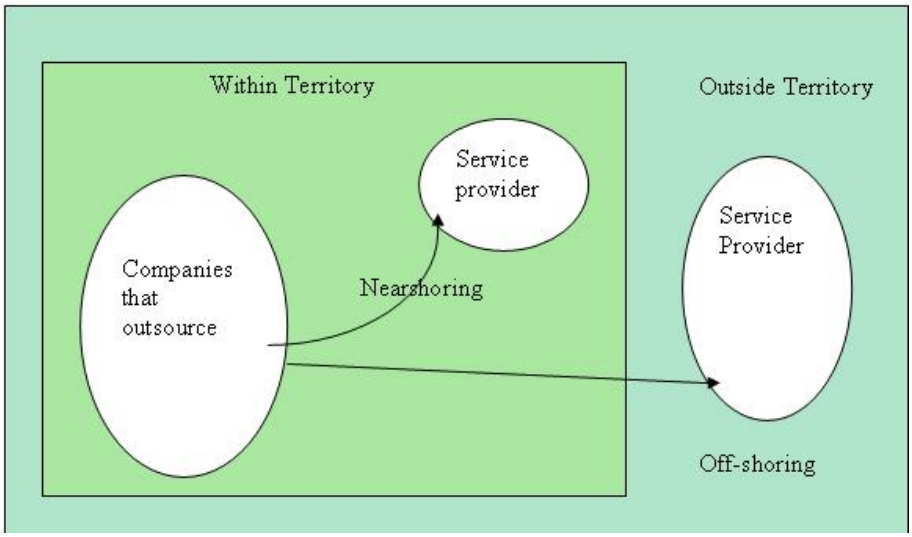



Figure 1.2: Near-shoring and Off-shoring

## 1.1 Types of Outsourcing

Outsourcing of services can be mainly divided into technological services outsourcing and business process outsourcing [18]. Technology services outsourcing relate to technical services while Business process outsourcing relate to front office management functions.

**Technical Services:** These services are like, application development and maintenance, web hosting, E-Commerce and more. Here, the sort of outsourcing is like the client ask's the service provider to develop a software application or hosting a website and provide maintenance and support for the developed product. The total software phases or part of them are performed by the service provider. This is mainly because

the companies by doing so can concentrate on their core activities.

**Business Process Outsourcing:** These services are like, back office operations, customer relationship management, call centers and telemarketing, payroll maintenance, and more. In this type of outsourcing, the work which was done by the enterprise itself earlier will be outsourced to a third party service provider who has the expertise in the related areas. The best example for this BPO are call center services.

This thesis mainly concentrates on software development in an outsourcing environment. Software outsourcing which falls in the group of technology services outsourcing, involves shifting of software development activities by a company to third party service vendors or software firms that are/may be located in offshore destinations. This software development being outsourced will be developed according to the models available with the outsourcing service provider, i.e., onsite model, offsite model or the global delivery model.

## 1.2 Benefits of Outsourcing

Outsourcing was started as a way to cut costs during 1990's, but now it has been adopted to meet managerial functions of the company like payroll maintenance of the employees, front office management, etc. Some of the benefits that seem to have more impact on outsourcing [36, 21, 43] are as follows:

- **Reduced costs:** It has been observed that, an Outsourcing firm's coder, be it an Asian or European takes almost less than one-third the wage of an American coder. So, management considers outsourcing as a primary option to cut the staff and capital costs [22]

- **Service:** Providing best service to the clients is the other function of organizations considering Outsourcing. For example, Outsourcing help desk functions to a third party service provider who has the expertise in that can help the company to concentrate more on their core activities that is providing the best service to the customer.

- **Access to Technology and Skills:** It's been observed that companies look to outsource when they doesn't meet the industry standard technology and skills in-house. By outsourcing the company has higher access to technology and skills, by which the company can concentrate on other aspects without thinking about the process that requires more skills to be used.

- **Reduced Risk:** By outsourcing both the client and service provider will share the risk, with both taking some responsibility and accountability. For example if a particular service is outsourced and requirements are not clearly mentioned, both the client and service provider will share the risk.

- **24/7 Production Process:** The services being provided by the outsourcing service provider are available round the clock. As, they work all the time providing the necessary service, to meet the industry standards and also to overcome their competitors. This is mainly because of different time zones.

## 1.3 Risks of Outsourcing

When the firms are outsourcing their internal functions to a third party service provide there are always risks associated with it. These risks highlight the clients to act smartly in deciding what needs to be outsourced and what doesn't. Some of the risks associated with outsourcing [36, 21, 43] are as follows:

- Clients are expected to loose the control on the project not as a whole but on the part that has been outsourced. So, project managers are always at risk and this has been the major threat for them when outsourcing the work. This is mainly because; the Outsourcing service provider's try to overlook the company after the project is outsourced to them. So, in order to minimize the risk, the project manager should carefully study the background of that particular service provider.
- Many of the Outsourcing Service Provider's just pop theirs advertisements on the internet about their success and achievements in that related industry. And many of the company's trust on such type of advertisements and end up in problems. It's hard for the clients to simply trust on such type of advertisements and give them the work.
- It is a fact that companies loose most of their profits in incurring lots of amounts just for traveling to attend the meetings organized by the Outsourcing Service Provider's. It is often necessary for the clients to have minimum direct meetings with the service provider's in order to minimize this risk.
- Security related risks are also higher with respect to the outsourcing environment, since the service provider's doesn't only deal with one project but with multiple projects that too with multiple vendors. So, there is always a risk that many of the functions could leak from a company that could be very advantageous for competitors.

# Chapter 2

# Software Development Outsourcing

## 2.1 Definition

Software Development Outsourcing refers to the development of software offshore. The offshore provider not only offers the development of the software but also offers support services like maintenance, help desk, and documentation for reference. Some of the software developments outsourced [17] are as follows.

- **Application Development:** From the Off-shoring of software development perspective the application development can be done in many ways. Some of the ways are, to develop the entirely new application to the client, system architecture design, low level coding, quality assurance testing, and to implement some additional features to the existing application. Well, the first case is often an easy procedure since the application has to be developed according to the clients requirements and also it is a new application. But this is not the case with the modification of the existing application since it has to be modified according the business needs of the clients and moreover each and every thing has to be done according to the client's specification and his needs. For such type of development it is not enough for the OSP to just deliver the developed software but also to give them the support and maintenance services. So, it is often necessary for both the client as well as the service provider to detail the pricing model, the coding rules and also the copyright acts very well in advance before the actual development phase begins.

- **Systems Integration:** Integration refers to the summing up of some activities in order to achieve the required goal. Systems Integration refers to the connecting of existing applications or components in a common architecture so as to achieve the intended functionality. In order to implement this Service provider must know the needs of the clients, and then with the help of his industry knowledge and technical expertise; designs the solution for the client and implements it. This is done in a more matured ways so as to keep the place in this highly competitive market. For achieving this service provider's now have different off-shoring models through which they achieve the solution for systems integration.

– **Database management:** For any organization or a company data plays a major role in the success. For this each and every company stores its database with utmost care at a different location as it is needed for some more critical tasks. The offshore service providers simplify this task by implementing a database management system at the client's location. Many packages that are being outsourced not only give the benefits of storing and retrieving the data easily but also with some other services like data migration and data cleansing.

## 2.2 Benefits of Software Outsourcing

During the recent years software development outsourcing has been adopted by more and more companies. The majority of them state that this is only to reduce costs and some say that this is due to the technical expertise. According to a survey , the cost reduction is the major factor which is influencing a lot, for this it can be observed that the average annual salary for an engineer in USA is $70,000 during the year 2004, compared with $13,580 for an engineer in India [21]. Well is cost only the major benefit to the companies or are there any more benefits? Let take a look into this,

– Many of the US companies are trying to cut cost just by off-shoring their software development since they believe that it saves them money because of low labor costs in some of the countries like India, China and also the eastern Europe countries. One of the biggest elevator company OTIS spent about $420,000 on costs as 15 people were working offshore but after one year the company recovered its investment and now it is saving around $1.4 million a year on application development and maintenance [21, 36].

– The other option why majority of the companies opt for off-shoring is because the company can save money by paying less taxes; by outsourcing their work to an offshore destination [21].

– The company can concentrate on its core activities by outsourcing their work to an offshore location. This has been the major benefit for many of the companies nowadays.

– It is the policy for many of the companies to find a third party service provider if they cannot find that particular skills in-house, in that case it is beneficiary to outsource their work. So, majority of the companies look for highly skilled persons who can develop their application with low labour costs when they are off-shoring their work [36].

– Quality is another factor for the company's opting for off-shoring. Many of the companies say that the quality of the work done in the development of the application is very good, when compared with the work done in house.

– Many of the companies state that they can collaborate with the offshore development companies and can perform the software development 24 hours round the clock due to different time zones. For instance, a company which has the existence in multiple continents, for example in Asia and America due the difference in the time, when the development team in Asia stops the work, the development team in America continues to start the development work and likely the total development goes on round the clock.

– **Job Market:** There has been a consistent increase in job's for both offshore destination's and USA. This can be due due to the companies prospering from low labour costs and creating new jobs in US. [43]

## 2.3 Risks of Software Outsourcing

The risks in off-shore software development have not gained much importance since many of the projects developed offshore have not been reviewed after their completion. Even though many benefits are there we should not overlook on the risks related to this. So, let's take a look on them:

– **Requirements related:** This can be due to many factors, according to an author at project management institute, states that "poor deliverables due to poor requirements" [23]. Another author says that requirements related risks are because of language. Differences in language can sometimes lead to miscommunication, either due to language styles or incorrect vocabulary use. Miscommunications can lead to many problems since the fair communication between the user and developer is the most important factor in getting the exact requirements for the software being developed offshore.

– **Intellectual property related:** There is a high possibility for the theft of the developed product since the security is very less in some offshore development locations where the programmer's consistency of working is very low [23].

– **Cost and Time:** By selecting a wrong off-shore vendor the costs incurred as well as the time being spent on that lacks the necessary details for executing the offshore project [16].

– **Cultural mismatch with the outsourcing partner.**

– **Data Privacy.**

– **Legal and contractual issues:** Inability to take the necessary action or to recover damages if problems occur. "For example, A Programmer at Geometric software solutions ltd who was fired from his job stole the source code for a solid works plus 3-D CAD package and offered it to the company's competitors for a high price. Under the Indian intellectual property (IP) laws he might never be convicted of a crime because India has no laws against trade theft [23].

– **Sub Sub-contracting:** Majority of the firms that are into outsourcing have begun to outsource their part of the work to some other countries where they can get it more and more cheaper which increases the risk for the actual owner of the project. For example, some of the outsourcing companies which take the projects from US have started to outsource the coding part to China where the price per line of code is even cheaper [32].

– **Technology related:** It is always important for the client to check out the hardware and software development platforms used in developing the end product are compatible. Otherwise, later they might create some problems.

– There are many other factors such as coding styles, documentation patterns, and many can be problematic [21].

# Chapter 3

# Offshore Software Development

## 3.1 Overview

In today's rapidly changing environment, where it is not possible for a company to meet the industry standards in providing better quality services, especially in the case of software development, it goes for offshore service provider for a part of its work and then it can concentrate more on its activities in a much better way. There are a number of ways by which a company can go for the offshoring. There are mainly six delivery models for the delivery of software development being performed in an outsourcing environment and many of the offshore outsourcing service providers have been dependent on these delivery models that have evolved since the last one decade. They are:

- **Onsite model:** In this model the service provider finishes the outsourced work at the client's location.
- **Offsite model:** In this model the service provider will have its office nearby the client's premises.
- **Offshore model:** In the model the service provider is located in a different country from that of the client.
- **Hybrid model:** Also called as the onsite/offshore model wherein the client will have their people working at the client place as well as the offshore team working in a different country.
- **Offsite/Offshore model:** In this model, the client's task is done by the people working at the offsite near the client's premises as well as the team working abroad.
- **Global delivery model:** In this model, the client's task will be completed by a team working at the client's premises as well as the group of offshore teams working from different parts of the world.

The most important factors that are considered in choosing the software delivery model are: work being outsourced for completion, interaction required and mostly suitability

to the client. We will look at these models in a more detailed view in the preceding chapter.

As said by Pressman [35], a software process is a defined by a set of activities, methods, practices and technologies that people and companies use to develop and to keep related software and products. The interest in the software process is based on the following premises, the software quality is strongly dependent on the quality of the process used in its preparation; the software process can be defined, managed, measured and improved. So, to solve the actual problems in an industry setting, a software engineer or a team of engineers must incorporate a development strategy that encompasses the process, methods and tools and the generic phases like the design, analysis, construction, verification and management. This process is often referred to as a process model or a software engineering paradigm. All the software developments can be categorized as a problem solving loop in which four distinct stages are encountered: status quo, problem definition, technical development and solution integration see figure 3.1.

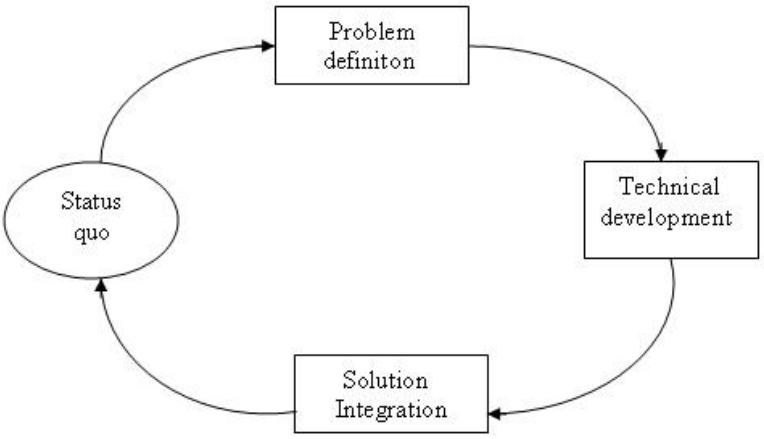



Figure 3.1: Phases of Problem Solving Loop

However, it is not a simple task to develop software using a well-defined development process. Such process has become increasingly more complex, where the demands from the companies increase according their business requirements and operations. As part of the globalization efforts currently pervading society, software project teams have also become geographically distributed on a world wide scale. This characterizes the global software development. Many tools and technological developments such as automated document elaboration, processes and other non-interactive communication channels have favored the teams working in the distributed environments. Moreover, Grinter, Herbsleb and Prikladnicki [30] point out that Global Software Development is one of the biggest business-oriented challenges that the current environment presents under the software development process point of view

## 3.2 Typical Phases in Software Development

Irrespective of the product being developed, there are a series of typical phases or steps that will be carried out during the development. Software development isn't all about the code. In fact coding is a part of the overall project lifecycle. Most companies have between four and six phases see figure 3.2 for their projects. The typical software phases [23] would be as follows:

1. Project initiation
2. Systems engineering.
3. Requirements analysis and definition.
4. Design and Development.
5. Testing and Integration.
6. Deployment and Maintenance

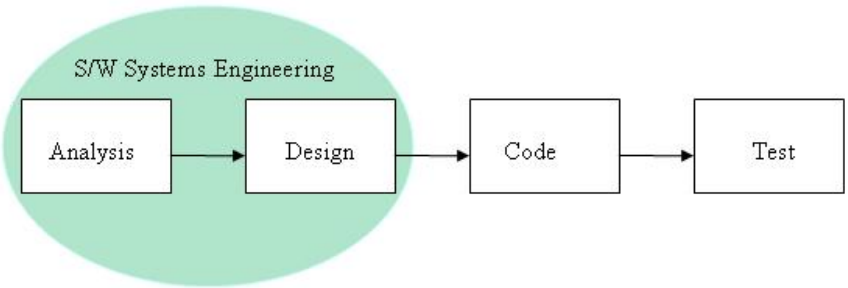



Figure 3.2: Typical Phases in Software Development

**Project initiation:** The process of approving an idea for the project, writing the details of the project, defining the project manager, and in general getting it started. Later on the effort of the project manager and other responsible persons for the project; define the overall schedule and project related needs.

**Systems engineering:** Here the work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. This system view is essential when software must interface with other elements such as hardware, people and databases. Systems engineering encompasses requirements gathering at the system level with a small amount of top level design and analysis.

**Requirements analysis and definition:** This phase begins by analyzing what exactly has to be done. The requirements gathering process is intensified and focused specifically on software. To understand the nature of the programs to be built, the software engineer must understand the information domain for the software, as well as required function, behavior, performance and interfacing.

**Design and Development:** This is the phase where the architecture of the product is developed and the design decomposed into more details until the software modules can be specified and the coding process begun. In terms of time this is often the

longest phase of the project. In terms of cost this is almost always the most expensive because the majority of the project's resources are working on this portion.

**Testing and Integration:** Once coding has been done, program testing begins. The testing process focuses on the logical internals of the software, assuring that all statements have been tested, and on the functional externals; that is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with required results. After the testing is done then the product is integrated into its operating environment. This phase typically ends when the user or clients signs off on the user acceptance results.

**Deployment and Maintenance:** This is the installation of the software product into the production environment. The project is officially over at this point and the product enters in to maintenance period. Software will undergo changes after it is delivered to the customer. So the software support and maintenance re-applies each of the preceding phases to an existing program rather than a new one.

## 3.3 Drivers for Offshore Software Development

The main drivers for offshore software development are the increased facilities of telecommunications leading to globalization and also the global software development. The other drivers include the low cost, access to expertise and access to latest technologies play the next role [31].

Globalization, with the advent of many communication technologies, the world is becoming a place where the companies are not limited to work in their local areas, but they can look for expertise throughout the world. Unlike material goods, software components if required can be transferred to the place where the required expertise is available and also the cost is less. Moreover the trend is the same in the global software development.

# Chapter 4

# Models for Software Development in Outsourcing Environment

## 4.1 Onsite Model

In this model a team of skilled software professionals are placed on the client's center, who works in continuous interaction with the client's team for the whole period of the project, starting from gathering the information to its implementation till the maintenance and support.

In this model the onsite team is located at the client's premises and they work together with the client's employees. Mostly in this model the client has direct interaction with the service provider and also has the chance to implement the required changes as needed according to the project execution. In other words, in this model the whole set of the project is implemented at the client's premises i.e., from information gathering to the maintenance and support. Accordingly, based on the needs of the project and the requirements of the client, the design, development and test teams are sent to the client's location for a short period of time [12, 34].

This model is regarded as the best model when the scope of the project is repetitive and open ended, as is the case with process re-engineering related services. This model best suits, when the client is not clear of the end results and also when there some unsteady requirements during the project. In some cases when the client wants to upgrade the existing system and where the client's participation and interaction is desired after each and every step involved in the project execution. Not only this, but when the client wants to upgrade the exiting system and wants to implement the latest technology then he chooses the onsite model. Onsite model is best for short term projects and also when the project concerned is a highly confidential task and the discretion has to be maintained.

The advantages of the Onsite model:

1. **Direct and continuous interaction with client:** The service provider will have a face to face interaction with client at each and every step of the project so that the service provider will have a better understanding of client's expectations and also the end results desired by the client.

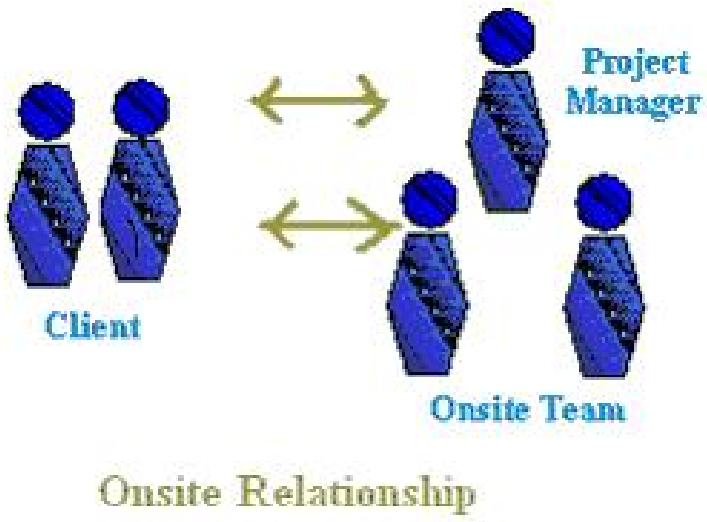



Figure 4.1: Relationship in Onsite model [15]

2. **On-hand information:** As the service provider is working along with the client in the same premises, the service provider can obtain the first hand information by understanding the scenario at the client's place.

3. Major advantage of this is no chance of communication gap since both of them will interact regularly. This helps in fewer changes to the system later when the end-product is developed.

4. When the client wants to upgrade the existing system by implementing some new business processes functionality with the help of the latest technology, this model is best suited.

5. Since the project is considered as the highly confidential task the client has the authorities to look up at the security issues very thoroughly.

## 4.2 Offsite Model

In this delivery model, the service provider will have their office located in the client's place so that the service provider will have a better understanding of the client's requirements. Offsite model helps the client and service provider to have a face to face interaction on regular basis which will benefit both of them. In the case of the IT outsourcing, the Offsite model has not only the benefit of being close to the client, but also this model supports the onsite team and also the offshore development activities at the offshore center [10].


Figure 4.2: Relationship in Offsite Model [15]

Offsite model is usually preferred in cases when the client's requirements are not properly defined and are expected to change during the project execution. So the offsite service provider who is located near the client's center with help them to better understand and accommodate the changes in a faster and better way about the project. This model is also used when the client is not in a position so as to expand its facilities all off a sudden to accommodate the service provider's team and simultaneously if the client want to outsource their project to someone who are located near to them then offsite model is the best which caters to both these needs. By choosing this model the client can also have a significant and sufficient control on the development process [34].

Another most important reason why the client chooses the offsite model is that, during the project, if the client wants some additional functionality to be incorporated to the existing process so that the project can meet the additional requirements and the modifications that are done to the project should be compatible with the client's existing setup. In this case the client goes for the offsite model because the service provider can come over to the client's place and study the exiting system and in detail and then work on the modifications. In the case, if the client has a problem after the implementation of the modified software, the offsite team can help them out.

The advantages of the offsite are:

1. **Quick response:** Whenever the client needs some modifications to be done to the existing system the response from the service provider will be very fast.

2. **Complete Understanding:** Due to the fact that the service provider is located near the client's premises that helps them to understand the client's needs in a better way.

3. **Good Synchronization:** In this model the client and service provider and the client will have a good coordination between them

## 4.3 Offshore Model

In the offshore model, the project development takes place completely at the service provider's premises which is located in a different country. The client will interact directly with the offshore center where the development is ongoing.


Figure 4.3: Relationship in Offshore Model [15]

The client and the service provider will not have any face-to-face interaction, when once the project requirements are captured. Later the client will have the interaction with the offshore team via, fax, email and phone if any changes in the requirements or to clear some of the doubts that may arise during the development at the offshore premises [7, 34].

This model is preferred when the project is well planned to meet the end product's requirements given by the customer. It is the responsibility of the client to give the requirements to the service provider as what they are expecting in the end product being developed. In this model the requirements level is very low since the service provider will not be in contact all the time. This is best suited when the scope of the project is long term [8].

Advantages of Offshore Model:

– **Quality of Service:** The quality of service being provided in this model is very high due to many of the offshore companies coders are waiting for chances and also they work for low labour costs. Which eventually reflects in the lower costs incurred in the project.

– **Time:** Clients can get the service round the clock (24/7) due to different time zones.

– **Skills:** Exposure to high-tech skills

Risks in Offshore Model:

– **Communication:** There is always of risk of communication gap in this model since both the service provider and the client will not have direct contact with each other.

– **Requirements Related:** Clients requirements may not be met correctly resulting in more and more modifications taking place during the development of the project.

## 4.4 Onsite/Offshore Model

The onsite/offshore delivery model is the fusion of both offsite and onsite models, in which the outsourcing work is distributed between onsite and offshore locations.

In this model, the outsourcing work that has to be distributed between the onsite and offshore project depends on the type of the project. The Onsite team is in continuous interaction with the client from the beginning of the project till the maintenance and support and working with the client at the client's center with the other employees of the client [13].

The tasks assigned at the onsite center in this model are, to gather the initial information about the project through direct interaction with the client, to understand the requirements or specifications clearly, to plan and do initial design as to how the project has to go about, to interact directly with clients in order to adapt new changes and minimize or eliminate last minute changes, to ensure that client is fully satisfied with the execution, implementation and end results [9]


Figure 4.4: Relationship in Onsite/Offshore model [15]

The tasks that are undertaken at the offshore development are, to understand the specifications and requirements of the project so that they will be have an detailed overview, responsible for project progress and also checking that the outcome matches to the specifications given by client and also giving the essential and uninterrupted support to the onsite center [34].

Generally 20-30% of work is performed onsite whereas 70-80% is outsourced offshore depending upon the criticality of the project

Onsite/Offshore model is usually considered when the project is complicated, knowledge transfer is more and duration of the project is long. This model is regarded as the most popular among the various outsourcing models because it gives the client an opportunity to directly contact with the service provider through the onsite center and at the same time gives the client the chance of enjoying the benefits of offshore outsourcing.

The advantages of the Onsite/Offshore model are:

- **No communication gap:** In this model the client can interact directly with the onsite center and also the service provider will get a good idea of client's requirements.
- **Proper use of Resources:** As this model is a combination of both Onsite and Offshore models, so the client can benefit from skilled manpower, best technologies and also the equipment that is available from offshore development center.
- **Cost Savings:** Since most of the work is outsourced and developed at the offshore development center the client benefits from this, by getting the work done at cheaper costs.
- **24/7 productivity :** For example when a project is been outsourced from United States to India clearly there will be a time zone difference between two countries and thus the work can be performed round the clock.

## 4.5 Offsite/offshore Model

Offsite/Offshore model is the combination of offsite and offshore models and is also the most successful and popular outsourcing model. In this model the service provider will have their offsite office close to the client's location and the offshore development at the service provider's location

In this model the offsite center of the service provider that is located near the client's premises serves as mediator between the client and the offshore development center. Both the offsite and the offshore development centers are well connected with highly secured network facilities which enable the seamless transition and execution of projects and with continuous communication with the client without compromising on the quality. This kind of secured connectivity ensures failure-free and secure access for the client to monitor the progress of development and support activity at both the offsite and offshore outsourcing locations [11, 34].

Generally, in this model the offsite development team handles 20-30% of the total work and the offshore team takes care of the rest.

What are the tasks that are accomplished at the onsite and offshore centers? Firstly the tasks that are performed at the Offsite outsourcing are: To start with, the offsite center first starts collecting the initial requirements from client and then the analysis


Figure 4.5: Relationship in Offsite/Offshore model [15]

of the requirement specifications of the project. Then the team starts the planning the initial design, after they are done with the initial design they start communicating with the offshore center, not only this but the offsite team also involves in testing the software at client's place before submitting to client. The offsite team also interacts with clients regularly and then taking care of the deliverables and also helps the client in maintenance and implementation of the project.

The work that is done at the offshore center is detailed design of the project and the developing the project according to the design specified and after developing the project, it thoroughly tested before handling to the offsite team and also providing technical and operational support to the offsite center. So these are the tasks that are performed in the Offsite/Offshore model [9].

This model is preferred in cases where the client can outsource the project to the service provider, who is located near to the client's premises, so that the client can have the control over the development process and also at the same time can avail the benefits resulting from offshore outsourcing. One more reason why the client chooses this model is if there are any changes in the client's requirements they can be better communicated to the service provider.

One of the most vital things is why the client chooses this model is that, this model has an efficient back up facility for the regular project backups and disaster recovery. The advantages of the Offsite/Offshore model are:

- As this model is a combination of offsite and offshore the client gets all the advantages of the offshore model.
- If there are any changes in the client's requirements there will be an immediate response to that from the offsite center.
- Since the offsite center is located near to client's office the client can have an eye on the development process to some extent.

## 4.6 Global Delivery Model

Global delivery model is fusion of onsite and offshore models but unlike onsite/offshore model where as in the offshore model the service provider is located at only one place, but in the case of global delivery model the service provider has the offshore development centers all over the globe. The service provider may not have a offshore development center of their own but can use the resources of their partners located all over the globe [3, 34].

According to McCarthy, "Global delivery model means having multiple locations that coordinate to deliver low-cost solutions to customers using consistent processes. While there may have been some degree of sophistication for some of the customers, it must be remembered that the market is maturing. The vendors are maturing and so are the customers." However, it may be noted that this transition is still at a nascent stage and the companies will require at least two to three years before they attain a 'fully developed sophisticated global delivery model', he said.

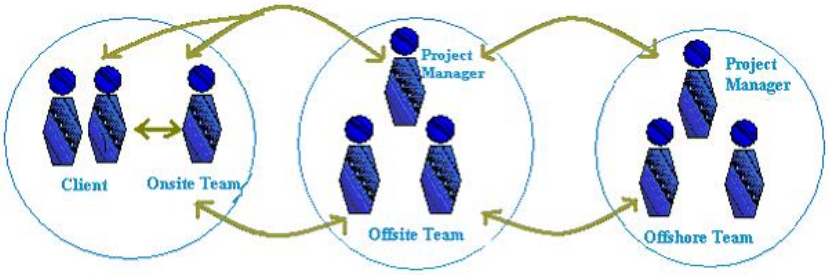



Figure 4.6: Relationship in Global Delivery Model [15]

As the Global delivery model is a combination of both the onsite and offshore delivery models, the work of the onsite center is that , firstly the onsite have to understand the client's requirements and then to interact with the client regularly in order to get a better understanding of the client's requirements and the changes made if there are any. The onsite center acts as a mediator between the client and the offshore center and also plan the initial design and then to distribute the work to the available resources. Finally testing the outcome of the project at the client's premises [6].

Offshore outsourcing is the most important part in the Global Delivery model, because in this firstly the work is divided into logical components and to distribute this work to the offshore centers that are geo-locationally located in order to perform the work where it creates a maximum value. The offshore center tests the project before handling it to the onsite team and also by providing continuous technical support. Thus this model provides the client with a large pool of team with different qualities and expertise in multiple fields.

This model is usually considered when the client wants the project to be done very quickly, which is possible by making the different offshore teams working together in order to finish the task. This model is also considered when is project is very big where varied expertise are required to finish the project.

The advantages of the global delivery model are:

- **Minimum Risks:** As the offshore center is spread all over the globe, if there is an emergency at one of the centers, the work can be distributed to the other offshore centers in order to reduce the risk.
- **Geo-locationally located:** Round the clock productivity will be possible because of the time-zone difference.
- **Quick completion of the project:** Since the work is distributed logically to the different offshore centers the project can be completed very soon.
- **Cost benefits:** The client can have an appreciable cost saving benefits.
- **Continuous interaction:** The onsite team will be interact with the client regularly and to see if the client needs any additional changes to be done to the given requirements and can be in regular contact with client in order to better understand the client.

Table 4.1: Various Delivery Models

| | | | |
|---|---|---|---|
| Onsite Model | -Onsite team is located at the client's premises<br>- Project is implemented at the client's premises i.e., from req. engg. to Maintenance and support<br>-Direct interaction with the client which helps in implementing the required changes as and when required during project execution | -Continuous Interaction with Client<br>-Instant Availability of information<br><br>-No communication gap | -Suitable for Short term projects<br>-Suitable for projects with constant change in requirements |
| Offsite Model | - Both the Client and Service Provider have Face to face interaction on regular basis | -Quick response<br><br>-Understanding of client's requirements clearly<br>-Good synchronization | -Not suitable for projects if the requirements are not stated before hand.<br>-Not suitable for long term projects. |

Table 4.1: (continued)

| | | | |
|---|---|---|---|
| Offshore Model | -Client and Service provider will not have any face to face contact after the requirements are captured<br>-The development of the application is performed offshore | -Quality of service is high<br>-Work Progresses 24/7<br>- Availability of high sills | -Risk of communication gap<br>- Risks related to requirements |
| Hybrid Model | -Combination of both offsite and onsite model<br>-The development team is divided into two parts.<br>-Onsite team is in contact with the client from the beginning till the maintenance<br>- Offsite team concentrates on the development process according to the onsite team specifications. | -No Communication gap<br>- Proper use of resources<br>- Cost savings<br>-24/7 productivity | -The project completely get ruined if the knowledge transfer is not good. |
| Offsite/ offshore Model | -Combination of offsite and offshore model<br>-Offsite office close to the clients location and the offshore development at the service providers location. | -Clients have all the advantages of offshore model<br>-Quick response<br>-Continuous Monitoring of the work progress | -If client's requirements change, and if the communication with the client is not proper then the project fails |
| Global Delivery Model | -Fusion of onsite and offshore models<br>-Initial phases (Req. Engg., High level design) of SDLC are performed onsite<br>-Later phases (Low-Level Design, coding, initial testing) are performed offshore | -Minimum risks compared with all the models<br>-Quick completion of project<br>-Cost benefits<br>- Constant interaction | -Could be the worst model if the projects are small<br>- Cultural differences. |

# Chapter 5

# Phases Elaborated

This Chapter gives a brief introduction to each and every phase in the software life cycle and how these phases are implemented at the offshore development center. Each phase starts with the general process and ends with implementation of that phase in an outsourcing environment.

## 5.1 Systems/Requirements Engineering in Outsourcing Environment

Requirements engineering plays a major role in any software development. This phase is the initial phase for any software project and is regarded as the most crucial phase in the total development process. This phase actually begins by determining what has to be done and implemented.

Requirement is defined as "1. A capability needed by a user to solve a problem or achieve an objective. 2. A capability that must be met or possessed by a system or system component to satisfy a contract, standard, or specification or other formally imposed document" [37]. The requirements engineering process was totally technical but this is not the case any longer with the projects being sent overseas. Many other requirements like, performance, security, reliability, usability and many others are needed [23].

Generally, it is the duty of the requirement analyst to take care of the requirements in any software development project. Based on the different levels and types of the requirements the Requirements analyst prepares the specification document both onsite and offshore before and after the project transfer. Figure 5.1 Shows the different levels and types of requirements [42], according to this we have three levels of requirements, they are Business level, User level and Product level. The business level requirements, define the business process and the problems to be solved by the software product. User level requirements look at the functions to be performed by the software product from the user's point of view. The product level requirements define the products functionality to be built into the product to enable users to accomplish their tasks, and there by satisfying the business requirements.

The different types of requirements are Business Requirements, User requirements Business rules, quality attributes, Functional requirements, non functional requirements, interface requirements, data requirements, constraints, etc.

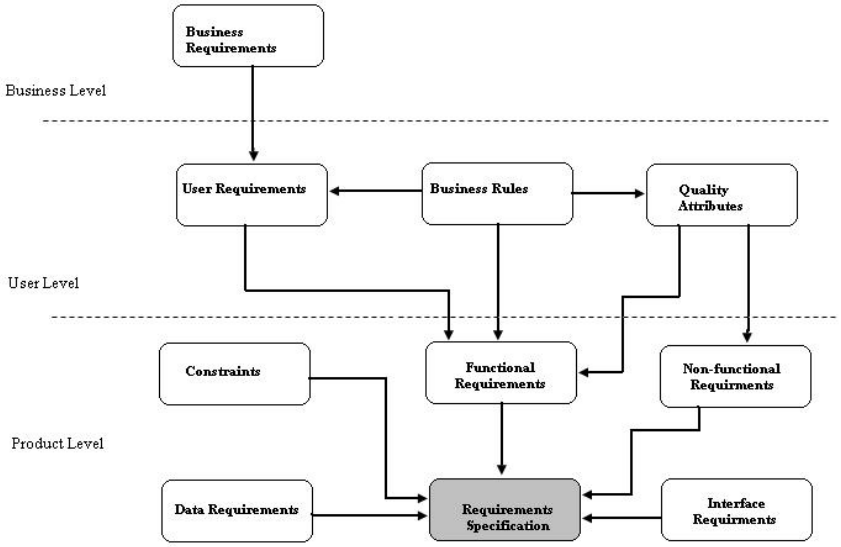



Figure 5.1: Levels and Types of Requirements [42]

- **Functional:** These requirements specify how the system should perform, how the data is to be manipulated.
- **Non-Functional:** These requirements are the functions that must and should support the functional requirements. Quality requirements are the non functional characteristics which only define the software products quality.
- **User Requirements:** These requirements define what the software or the system has to do in order for the users to perform their tasks or objectives. Sometimes multiple user level requirements are needed to satisfy one business requirement.
- **Interface:** These requirements define the flow of data or information to the external devices like printer, fax, shared resources and other software applications that are connected outside the software product.
- **Performance:** requirements that deal with the performance of the system. How the system should act in certain instances and how should it respond to the users actions, etc,
- **Data:** These requirements deal with data manipulation, where to be stored etc.
- **Security requirements:** These requirements are the most important when the project is being sent overseas since the security risks are very high. These requirements deal with all the security issues in the project.

**Requirements Engineering Process:** Many research theories and literature's states that many of the software projects fail because of poor requirements engineering [28, 24]. Even though the requirements analyst tries to gather all the requirements from the users and the customer, the final product doesn't meet all. This is because, firstly, the requirements gathering is not a one step process, secondly, the requirements are not easy to be described in words, thirdly, there are different requirements at different levels and finally, requirements change (requirements volatility) all the time during the project life cycle.

As stated earlier, requirements engineering process is a disciplined task to identify, define, document and maintain the software requirements throughout the software development. This generally encompasses of two different tasks, namely, requirements development and requirements management see figure 5.2 [41]. The requirements development consists of Elicitation, analysis, Specification and validation.

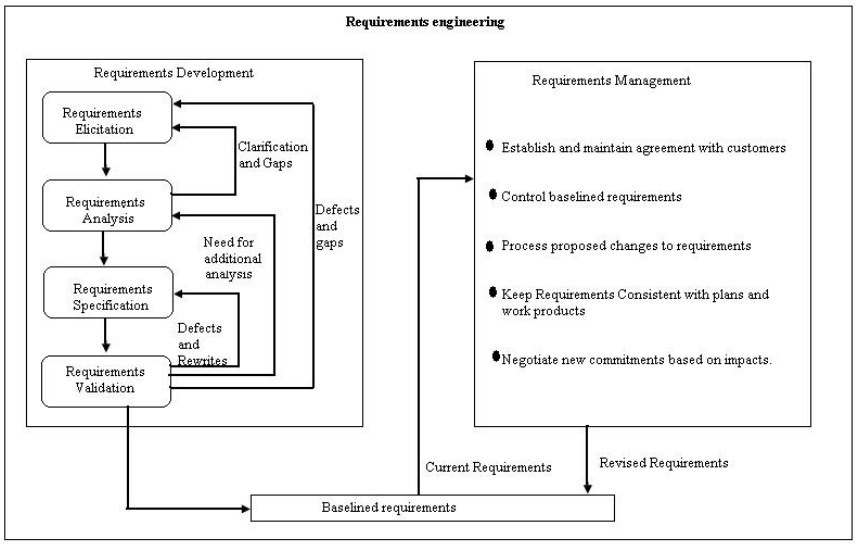



Figure 5.2: Requirements Engineering Process [42]

## Requirements Development:

1. **Requirements Elicitation:** Requirements elicitation is the process of gathering the information in the development process. This phase includes various activities that are involved in identifying the requirements from each and every stakeholder who is involved in using the proposed system. Different techniques like interviews, workshops, observations on the current business process, group tasks are used in gathering the requirements. Once the gathering activity comes to an end, a work document in produced which includes all the things like, a statement of need, scope for the proposed system, systems technical environment, domain constraints, and usage scenarios.
2. **Requirements analysis:** The requirements analysis phase categorizes the identified requirements into the related subsets, explores each relationship to the

others, examines the consistency, ambiguity and prioritizes them based on the needs of the users of the proposed system. Later, this step includes representing the requirements in various forms including prototypes and models, looking for gaps in the requirements.. Priorities are set for each and every requirement after discussing with the users. Risks are identified and analyzed. This step may iterate the requirements elicitation process to remove/add more requirements.

3. **Requirements Specification:** In this step, the requirements are formally documented so that it can serve as the foundation for the hardware engineering, software engineering and database engineering. This defines the functionality and the performance of the proposed system, and also the information flow. A standard template is used for the specification purpose. This specification document may vary from single software requirements specification (SRS) to multiple documents. One good process of documenting SRS is to use up a predefined template that can ensure that key items are not overlooked as the requirements are documented.

4. **Requirements validation:** This is the last step in the requirements development process which validates the identified requirements to ensure that all of them are correctly written, unambiguous, and complete and satisfy the customer needs. This step may lead to iterate the other steps in the development process due to the defects, gaps, additional information, or other issues. Many validation techniques are used in the requirements validation process. One of the validation mechanisms is to conduct peer evaluation of the requirements specification document. Empirical evidences show that the peer evaluation of the requirements documentation has the highest priority of defect detection activity. Another major tool used for validation is to start writing the test cases for the functional (black box) testing of the software.

**Requirements Management:** Requirements management is the process of tracking the requirements from the start to the deployment of the project. Requirements management is a set of activities that help the project team to identify, control, and track requirements and changes to requirements at any time as the project proceeds [35]. This involves an analysis of the requested changes, approving or disapproving those changes, and implementing those changes. Requirements management also takes care of keeping the project plans consistent and tracking the status of the requirements in the ongoing software development process. Requirements management begins with the identification activity. Each requirement is assigned a unique identifier that has the form

<Requirement type> <Requirement # >

Where requirement type takes on values such as: F = functional, D = Data, B = Behavioral, I = interface and P = output. It's a common practice to develop the traceability tables once the requirements have been identified. Some of the traceability tables are source traceability table, dependency traceability table, subsystem traceability table, interface traceability table. These tables are maintained in the requirements database so that they can be quickly searched to understand how a change in one requirement will affect different aspects of the system.

Many research studies have been suggesting that gathering the requirements in geographically distributed locations is a challenging task and a process model should be proposed for the requirements engineering in global software development (GSD) [28]. Mainly 4 types of categories have been identified in the analysis made by many of the

researchers and they are communication, culture, knowledge management, and technical aspects see figure 5.3. Each category has several factors and the relationship between them is so close, that it is very hard to define the limits.

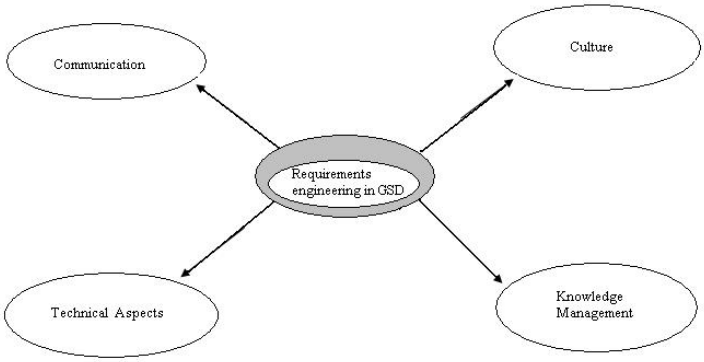



Figure 5.3: Categories Related to Requirements Engineering in Global Software Development

- **Communication:** Requirements engineering process in any software development mainly depends on the communication. When the project is being sent overseas communication between the customer and the service provider becomes the most important activity. Clear communication is a pre-requisite to avoid misunderstandings and conflicts during the requirements gathering process. The major important factors considered for communication are time zone, language and communication media. Language barrier should be removed so that the requirements are gathered correctly. The use of state of art communication media may help them in removing the barrier.
- **Culture:** The culture of the teams also affects the requirements gathering process. Since, the teams are geographically distributed, culture plays an important role, both the organizational cultures and national cultures will affect the requirements engineering. The main factors related to culture are context, attitude and values.
- **Knowledge management:** There is large information flow in the requirements engineering process. Making available this information flow or the knowledge is also a tedious task. There must be some sort of knowledge management that can collect, process, and store and make available and also unify the organizational vision. The main factors identified are expectations, awareness and management of cultural information.
- **Technical aspects:** Several technical aspects affect the RE process in distributed environments. The main factors found are patterns, process and configuration

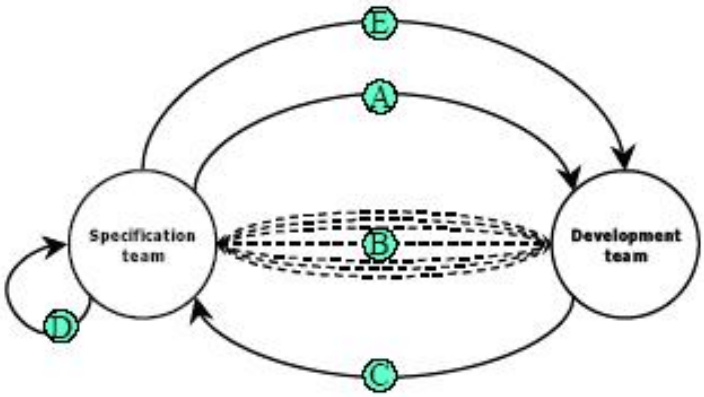



Figure 5.4: Proposed Process

management. Tools, products that are used to gather the requirements do not fulfill the total aspects in getting the total requirements in distributed environments.

A Process model is needed for the requirements engineering process in an outsourcing environment irrespective of the outsourcing models being used. Where the interaction between the customer and the developer increases thereby reducing the four factors stated above. A process model was proposed by a team of members at the school of computer science, Brazil [28] which intends to reduce distribution difficulties and understanding the SRS in development team. According to their findings, offshore outsourcing has several roles related to software requirements engineering and these roles are performed by one person and sometimes by groups. Therefore two teams are identified with tasks in requirements engineering in offshore environments, business analyst team also called as specification team and application analyst team also called as development team. This process model is composed of five steps as shown in figure 5.4. According to this process it has to be considered that the specification team is in United States while the development team could be in any other part of the world (here India, Brazil, or China) . The specification team is responsible for conducting elicitation, analysis, specification and validation of requirements and the development team is responsible for developing the software based on the specification document with the help of coding and testing.

1. **SRS First version is concluded and sent to development team.** The specification team after conducting the elicitation, negotiation writes the SRS initial document. After concluding these tasks, the document is sent to development team offshore where the coding could begin. However difficulties due to distribution could reduce the understanding of the SRS document.

2. **SRS analysis and adaptation by development team.** This would be the first contact by the specification team with the development team. After receiving the

document form the specification team, the major part of the development team is try to understand the total document deeply. Factors like culture and language could cause the lack of clearness and ambiguity could arise. Under the worst cases the total SRS is rewritten. Several questions could arise and they are cleared by the specification team.

3. **SRS adaptation is concluded. SRS is sent to specification team for approval.** When the SRS document is adapted with the new changes in it by the development team it is sent to the specification team for verification.

4. **SRS Validation and approval by specification team.** Once the specification team receives the document from the development team it reviews it to assure that after the adaptation it still reflects the requirements needed by the client.

5. **SRS Final version is defined.** After the specification team approves the document, the final version of the SRS is defined as approved SRS. After this , the development team could concentrate on the modeling, coding and testing activities depending on the final version of the SRS document.

The process model stated above tries to reduce the four factors considered during the global software development. But, according to our findings, communication is the major important factor and hence it would be a very good approach to include, a person from the specification team to the development team during the development process and a person from the development team to the specification team during the requirements elicitation, negotiation and specification process. By doing so, it would help both the client and service provider to maintain a balance between the requirements and development process.

## 5.2 Design phase in Outsourcing Environment

Design is the technical kernel of software engineering. This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture. This phase represents the "how "phase. Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces and many other engineering details are established. During the design phase, Software Design Traceability Analysis, S/W Design Evaluation, S/W Design Interface Analysis and Test Plan Evaluation are carried out [35].

Depending on the design method being implemented the design model comprises of a data design, an architectural design, an interface design and a component-level design see figure 5.5

**Data Design:** This step translates the data objects defined in the analysis model into data structures that reside within the software. The data objects and relationships defined in the entity relationship diagram and the detailed data content depicted in the data dictionary provide basis for the data design activity. At a higher level of abstraction, data design may lead to the definition of architecture for a database or a data warehouse.

In many software applications, the architecture of the data will have a profound influence on the architecture of the software that must process it. At the program component level, the design of the data structures and the associated algorithms required

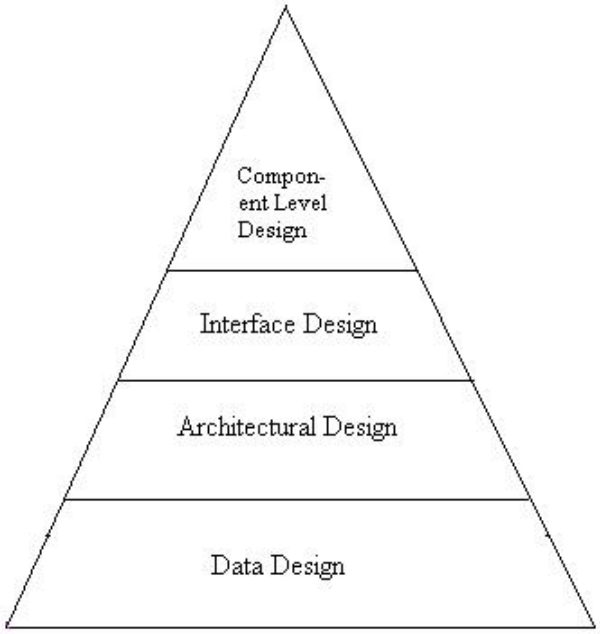



Figure 5.5: Design Model

to manipulate them is essential to the creation of the high- level applications. More detailed data design occurs as each software component is designed.

**Architectural Design:** Architectural design encompasses the initial set of design activities that lead to a complete design model of the software. The architectural design defines the relationship between major structural elements of the software, the "design patterns for the system, and the constraints that affect the way in which architectural design patterns can be applied. The architectural design representation-the framework of a computer-based system-can be derived from the system specification, the analysis model, and the interaction of the subsystems defined within the analysis model.

**Interface Design:** User interface design begins with the identification of user, task and environmental requirements. It also describes how the software communicates within itself, with systems that interoperate with it, and with users. An interface implies a flow of information and a particular type behavior. Therefore data control flow diagrams provide much of the information required for interface design.

**Component-level design:** At this level the structural elements of the software architecture is transformed into a procedural description of the software components. At the component level, the software engineer must represent data structures, interfaces, and algorithms in sufficient detail to guide in the generation of programming language source code.

In the case of outsourcing, activities that require more concentration like the requirements analysis, high level design are performed onsite. When the analysis model, design

specification are captured correctly the development activity is moved offsite. At the offsite, the low-level design, development part, initial testing parts are performed. This approach is mostly seen in service provider's global delivery model [2].

**High Level Design:** The purpose of this design phase is to analyze the problem domain, establish a sound architectural foundation, complete the project plan, and eliminate the highest risk elements of the project. The overall functioning of the system and establish the functional and physical rules and design guidelines [5].

**Low Level Design:** During this phase, the detailed design of the software components is carried out and a detailed specification is written for the various software components based on high-level design. In the low level design phase, the 'System Design Document' (SDD) consisting of database structure, test cases and detailed program specifications is drawn [5].

The architecture of the software product represents the high level view of its internal structure. The architecture defines how the design patterns, modules and the technical components of the product are related. During the architecture design part the outsourcer has to consider, how the software would be developed, structure of the team that is going to work on the development of software and finally what is the impact of the part of the product development that is outsourced will have on the product.

There are three ways to architect the software product to take the advantage of outsourcing [19]. Firstly, partition of the architecture into pieces that can be developed independently. By doing so independent modules can be developed with outsourcing and the others can be developed by the client's team (internal team) . The integration of the product is usually handled by the client's team before the product release. The modules will perform well if they are loosely coupled.

Secondly, making the internal and outsourced development teams work in a truly collaborative environment. This approach works well when both theteams are involved with architectural decisions and thus the technical strengths of the development teams can also be involved.

The last case is to outsource the development of the entire product. In this case, the architecture should be defined with input from the developers and is an important tool for communicating the design of the software to new members of the team.

## 5.3 Coding and Security issues

In most of the software development projects, coding phase is mostly overlooked since this is more about logical parts of the project, but this isn't the case with outsourcing. In fact this is the most crucial phase in any of the projects that are developed overseas. This phase is the longest phase in the software development and incurs a lot of investment. The outsourcing service provider will give a number of options to customer for providing the coding services. Some of them are given below [40].

**Outsourced coding services:** Here in this model the customer will completely outsource the coding part and the work will be completed by the team of members in a stipulated time for a specific cost. This type is the most common in projects that are sent overseas.

**In-sourced Coding services:** In this model, the service provider will send some of the best personnel to the customer's place where a development team is already working in the environment.

**Co-sourced Coding services:** Here the customer can hire the development team that will work as the extension to the existing team at the customer's premises.

**Dedicated Development group:** In this a dedicated team is completed hired by the customer and this team will work at the customers premises and they will be paid by the customer itself regardless of time and requirements.

**Build Operate Transfer:** In this model the customer first tests the team whether they can work with them or not. So, the team first works exclusively for the customer on some project at the customer's premises and after some time they become part of the customer's team.

It has been estimated that 80% [Gartner Group] of the US companies are considering outsourcing critical services to foreign based companies [25]. Mostly these countries are India, china and Russia along with some ex-soviet nations. But, all these three countries pose security threats like cyber terrorism and data theft. So, companies that consider outsourcing their coding parts to these countries should consider verification and auditing services at the service provider's premises like checking their file servers and source-code servers and also to conduct some background investigations of software coders working overseas. For example, a coder in India who has been fired from his job stole the code from a CAD application and sold it to the company's competitor and the company could do nothing as there are no laws written against the trade thefts in India.

Privacy is the other major concern for companies outsourcing their business services-Unauthorized access to information by unauthorized personnel [38]. Privacy is not a big issue when we talk of the firms in America and Europe since both of them have very high standards for maintaining the security of the employee's information. There are several Federal laws which are made against them to prevent the theft of data; like Act of 1996 (protecting health and medical information) , Act of 1999 (protecting financial information) . Even the European Union laws are more stringent. But when companies outsourcing their Business process to countries like India, china where there are no laws against such type of thefts, preventive measures should be taken by the companies to prevent them. One way of ensuring security in the offshore destinations is to go on regular checks at the client's place and the other way is to put the need for security issue in the contract document and say the outsourcing provider that the company will pay dividends for taking care of such issues [29].

## 5.4 Testing in an Outsourcing Environment

For any software development projects, the development starts with the identification of the requirements and ends with the formal verification of the developed software against those requirements. Testing phase starts with creating some test cases to demolish the built system. Mostly testing is a one step software process that could be viewed as destructive rather than constructive [35]. Software testing has been defined as the process of executing a software system to determine the compliance with specification [44]. Testing has been identified as the basic form of defect identification for majority of software projects.

Software testing strategy can be viewed in the context of a spiral as shown in figure 5.6 Unit testing begins at the vortex of the spiral, concentrating on each unit or component of the software. Then the spiral progresses outwards to integration testing where the focus is on design and software architecture. Moving more outwards we arrive at the validation testing which concentrates on the validation of the requirements against the software that is being constructed. Finally we arrive at the system testing where the software built is tested with the other components like hardware peripherals.

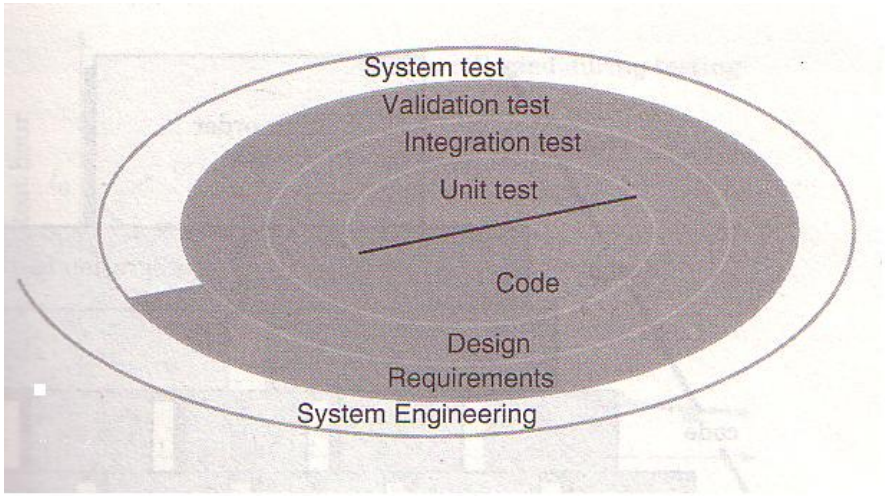



Figure 5.6: Software Testing Strategy

Testing is basically divided into three broad categories namely, General testing also called as the white box testing, specialized testing also called load testing and business testing also called as black box testing that involves the users themselves. Any software engineering project can be tested in two ways, firstly, testing whether each function is fully operational, while at the same time searching for errors in each function (i.e. black box testing) , secondly, testing the internal operations of the functions (white box testing) . There may be 18 stages of testing any software product, Jyothi and Bhatia [27] for example describe a testing process to be consisting of 18 stages, out of which 7 are normally black box test methods, 7 are white box test methods, and 4 are mixed test methods. But these may not be common for every product being tested.

Considering the Outsourcing environment, it is a good approach to send the testing of the developed application to an other service provider. By doing so, the testing phase can be done more effectively. The testing service provider provides a number of testing strategies for testing depending on the type of application viz., application testing, Automation testing and package testing are some. The below giving testing lifecycle is valid for testing, standard and custom build application using different technologies [20].There are totally 7 phases see figure 5.7 to test the application.

1. Test Requirements
2. Test Planning
3. Test Environment setup
4. Test Design
5. Test Automation
6. Test Execution and Defect Tracking
7. Test Reports and Acceptance

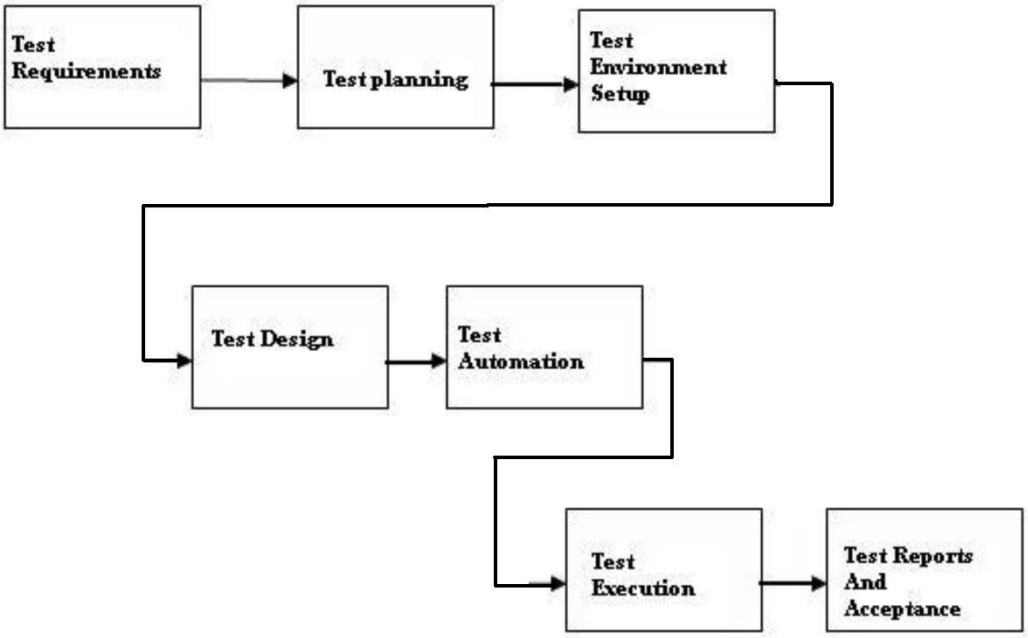



Figure 5.7: Testing Life Cycle

**Test Requirements:** This phase focuses on documents like requirements specification document, functional specification document, design specification documents, and use case documents to arrive at a decision that all the requirements are met and are in accordance with the system being built. It is always stated that if the software is built on inaccurate requirements, then despite the well written code, the software will be unsatisfactory. Defining the quality measure for each requirement, could help in arriving at the solutions which is acceptable. For example if we have a requirement like the system should respond to the user's request very promptly. If we consider like the system should not take more than 5 minutes to give the response to the user then 5min will become the quality measure for the system's response.

A number of questions are asked here in this phase to test whether the built system is according to the requirements specification document [33] and all the requirements are satisfied. Some of the questions which are used in this test are as follows:

*Is every requirement relevant to the system being built?*

*Are there any requirements that are internally connected to the other parts of the system and if so, changes to them, can you identify which parts of the system are affected?*

*Are requirements uniquely identifiable?*

*Does the requirements specification document contain any solutions as a requirement?*

The software requirements specification document should contain everything of the system being built and also the conditions under which it must perform. Thus this

requirements testing helps in assuring the quality of the requirements which helps in reworks due to the requirements defects.

**Test Planning:** This phase actually provides the practical way to manage and control the testing activities. Test plan also talks about the different test phases and test methodologies to be implemented. This planning helps the testing manager to control the testing process from a top level with confidence. Project performance, defect management configuration management, risk management are some of the activities that come under this phase.

The test plan presents the following, test objective, test scope, test approach, test assumption, test dependencies, and test risks [26]. The test objective is a clear statement stating why the testing effort is being implements and what will be the scope of the test. Test scope is all about the in's/out's of the system to be tested. The testing approach will talk about what are the testing activities "to do" that will be applied against the application for the current testing phase. Assumptions are the facts or the statements that the testing team thinks are true. If any assumptions prove to be not true, then it will have an effect on the testing activity. So the tests assumptions are documented that are critical to the current situation. Test dependencies are the events that are to be completed in order to proceed in the testing activity. Risks are the factors that stop the testing activity to proceed further. All the expected risks are documented in the list so that they are not repeated.

**Test Environment Setup:** This phase talks about like setting up the testing environment required for testing the system being built. This involves like test bed installation and configuration, network's connectivity, all the tools installation and configuration. All this is done in coordination with the vendors and others.

**Test design:** In this phase a total test design is created on how to proceed with the total testing process on the built application. Testing scenarios are identified and test cases are prepared. Also the test cases are reviewed and are approved. A common idea in this phase is to implement a test traceability matrix and also use some tools to design a test coverage model to implement the tests. A test traceability matrix is a matrix which maps the test requirements, with test methods and test cases so that all of he tests are done by the testers according to the user requirements.

**Test Automation:** In this phase the need for the software test automation is identified. Automation requirements and tools are evaluated and a frame work is designed. To put clearly, test automation is an art of converting manual test cases to machine readable code i.e., automating the current manual testing process. This process consists of detailed test cases along with the expected results built with the help of the specification document, and a test database with all the test cases included in that so that those cases are repeated when there are modifications made to the application.

**Test Execution:** In this phase all the test cases are executed and the test scripts are tested so that all the tests are completed as per the test design and test plan. The testers develop a document with the errors found in the application and are sent to the development team for verification for its closure.

**Test reports and acceptance:** The results from the previous phase are noted, test metrics are generated and a final report is created by the testers made on their test efforts stating whether the software or application tested is ready for release or not.

After the total application passes through all the testing phases then the outsourcing team develops a release version of the software or application and is sent to the client for acceptance.

## 5.5 Deployment and Maintenance

When the total application is developed and tested by the offsite team; the application is deployed into a package and is sent to the client's location for final testing. Field testing is done by the client's personnel and the feedback is obtained by the service providers personal from the client for processing. After the total application is deployed at the clients premises, the application enters into the maintenance period where any additional requests by the client and defects fixing are done during the warranty period.

# Chapter 6

# Conclusion

Although Outsourcing is regarded as a company's way to reduce Costs there are many other factors which are to be considered like, security issues, Cultural mismatch, requirements related etc. Utmost care should be taken by the companies when going for outsourcing. This paper on outsourcing software development tries to give a brief picture of how software is developed in an outsourcing environment. Firstly, it states whether it is a good way to think for outsourcing or not. Secondly, what happens when the software is developed offshore with the available[various types]software lifecycle?

Requirements engineering is a task where the total development of the project depends. But, when it comes to outsourcing, there are many other issues that came into picture and it was strongly assumed that a process model is needed for the total requirements engineering in an outsourcing environment. It is always a good approach to think for requirements engineering outsourcing when the company can follow a step-by-step procedure to ensure that no communication gap and cultural differences are encountered during the total requirements gathering process. Since, communication and culture are the important factors that play a major role for interacting with both the client and service provider. The less the gaps between them the more efficient and effective the work progress.

The Design phase is the task where the actual prototype is built depending on the requirements document delivered by the requirements analyst. While the requirements engineering and High-Level Design is done onsite and low level design is done offsite. The onsite team is responsible for both the requirements gathering as well as the high level design. They are responsible for The integration of the product is usually handled by the client's team located onsite.

The Coding phase is the most crucial phase in the development where the total requirements document and the design document play a major role in making software development success. There are a number of issues like security and privacy which are to be taken into consideration while outsourcing coding to a third party. It is always important to make several checks at the service provider's center by the customer before signing the contract.

The Testing phase usually concentrates on the final corrections to the developed application (or software) . It is always a good approach by the company to look for a different service provider when outsourcing testing services. This not only helps the company in taking up the cost advantage but also in ensuring higher quality.

Finally, Outsourcing is a good approach if it is organised in a meaningful manner. A

company has to first concentrate on its core competencies and then let the other things done by a third party. Since outsourcing is a risky approach it is always recommended for the company to make through checks at the service provider center.

# CONTACT US :-

Foreignerds

+1 (201) 381 - 5152

contact@foreignerds.com

251 Little Falls Drive, Wilmington
DE,USA 19808

*Thankyou*