

A Process Maturity Model for Geographically Dispersed Software Sustenance Operations

Rajeev Sharma¹, and S. Krishna²

¹XLRI School of Management, Jamshedpur- 831 00, Jharkhand, India

²Indian Institute of Management Bangalore, Bannerghatta Road, Bangalore - 560 076, Karnataka, India.

Abstract. An increase in demand for software services has led to development of software from different dispersed locations. This has brought in complexities to managing software projects. This research work focuses on the development of a process maturity model that balances different perspectives in one organization that is carrying out software sustenance work from geographically dispersed locations.

1 Introduction

The development of tools and processes using information technology for coordinating the activities of geographically dispersed team members is not the only challenge that is being faced by many software organizations. The biggest challenge is to create an organization that can increase the benefits for all its stakeholders. This necessitates the improvement in processes and tools while at the same time retaining a balance among competing demands of diverse stakeholders.

Over the last few years, there has been an increase in studies that try to understand process maturity in software organizations. The Software Engineering Institute (SEI) has developed a comprehensive model predicated on a set of software engineering capabilities that should be present as organizations reach different levels of process maturity. To determine an organization's current state of process maturity, a five point grading scheme is used. The grading scheme determines compliance with Capability Maturity Model (CMM), which defines key activities required at different levels of process maturity. This model deals with the process maturity for a specific location, but in the current environment, when the work has become

geographically dispersed, we also need to look into how organizations are managing this transition. Therefore, we have built a process maturity model for geographically dispersed projects. This model has four different stages: Initial, Localized, Sensitized, and Synchronized. Transition from one stage to the next takes place over a period of time.

We also see that to successfully synchronizing software sustenance activities from dispersed locations, organizations need to strike a balance among different competing requirements of its stakeholders. We propose a framework for developing a better understanding of this process to achieve a balance among these competing needs.

2 Research Design

To build a theory about process maturity in geographically dispersed software sustenance activities, we decided to adapt grounded research design [1]. The following paragraphs justify the choice of the qualitative approach and describe the methods adopted. There are three main reasons for adopting a grounded research based qualitative research methodology [2] for this study:

- At present there is no theory that explains the process maturity that happens over a period of time in a geographically dispersed software sustenance organization. Hence, we wanted to explore this phenomenon in its natural setting to gain fresh insights into this type of work arrangement.
- At the same time research literature on co-located software and non-software development teams shows there are a number of factors that have an influence on how the work gets done. Though there are a large number of theoretical and empirical studies, there is no clear understanding about the phenomenon of process maturity in a geographically dispersed software sustenance operation.
- As this is a first attempt at process maturity in an era marked by the development of various web-enabled tools and processes, we adopted grounded theory based methods to describe the phenomenon in greater detail.

The above-mentioned conditions presented us with a unique opportunity for conducting a grounded theory, approach-based study in its natural setting [1].

We collected data in the form of interviews, visits to the worksites, documents, etc. The primary source of data for this study is transcripts of interviews with the employees of different software sustenance organizations. For conducting the interviews, a brief interview guide was prepared to facilitate the interviewing process as well as facilitate the comparison of the data collected from different sources.

The questions mainly dealt with the business, technology, and operations aspects of the software sustenance activities. At the senior level the questions covered the business and operation aspect of the development; at the junior level the questions were restricted to operations and technical aspects of development. Through these questions we also tried to find out the difference between what really works and the

standard operating procedures in different situations faced by the developers. We also tried to check the consistency of the answers given by the senior and junior employees by cross checking their responses.

Beyond transcripts of actual interviews, the study is also informed by interview notes, notes on site visits, company documents, and project files (when they were made available). When possible we tried using these data sources to cross check our deductions about the operations of an organization.

All the interviews were conducted at the project sites, and on average, each interview lasted for more than an hour. Going to the development sites also helped the researcher understand the work environment at these organizations. Some of the interviewees also showed some electronic documents, which would not have been possible otherwise. To maintain the confidentiality of the organizations and the individuals who supported our research, names have been changed to protect their identity.

To analyze the data using a constant comparative method, each interview was reviewed and an initial analysis attempted as soon as possible after the data was collected. After completing the data collection, we again went through the transcripts, coded the data, and tried to make sense of the phenomenon of our interest. After completing the coding we wrote case studies [3] and conducted analysis for generating some generic results from this study.

In the following section we will present a description of the operations of “Orion,” which is maintaining proprietary software from three different locations in the world.

3 Case Study on Orion Limited

Since its inception in 1982, a singular vision—The Network is The Computer—has propelled Orion Ltd. to its position as a leading provider of industrial-strength hardware, software, and services that make the “Net” work. Orion entered India in 1987, through Wipro its sole distributor. Orion India was established in 1995. Soon after, the company established a 100 percent subsidiary, the first such subsidiary of any MNC in the country. Orion's India Engineering Centre (IEC), in Bangalore, commenced operations towards the end of 1998 and was formally launched in May 1999.

IEC has now become a key site for technology development for the company worldwide. Each of Orion's six divisions (Enterprise Services, OrionOne, Network Service Provider, Network Storage, Systems Products, and Software Systems Group) has teams working at IEC. Products and technologies at IEC include a mission-critical Stern Operating Environment and the OrionOne product line: E-commerce infrastructure and application products, security products, and advanced network storage solutions. The India engineering center is slated to grow to a core-engineering site with 3,000-5,000 employees.

The IEC follows a business model of ‘virtual teams’ where all teams at India IEC are part of worldwide team that operates across geographies. The final decision on

work assigned to a particular location is made after giving due consideration to the competence of the team members from different geographically dispersed locations.

The data presented in this case study has been collected from the Stern Software Sustenance team in Bangalore, India, and from some of their counterparts in the US. The sustenance activities are being carried out from different geographical locations to provide year round 24-hour support. At the same time, software development activities are being carried out from different locations to take advantage of local competencies and new business opportunities.

3.1 Stern Sustenance Group

Stern is a high-end proprietary operating system developed by the organization. This software is installed mainly on the servers and high-end workstations. In most of the cases Stern includes the hardware, which is also provided by Orion. This places Orion in a good position to provide support services to its clientele as it provides both the hardware and software. Orion's clientele includes most big companies, in various industries, that use computers extensively in their operations. Its competitors include various other variants of Unix, Windows and Open Source Operating systems.

At present the support for Stern software is being provided from three locations around the world: the US, the UK, and India. Earlier, support was being provided from the US and UK only. The India support center was established in the year 2000. These teams of engineers are providing 24X7 support to customers. Different sub-teams have been constituted to handle different jobs. For example, one team is working in the area of kernel; one team is working only in the networking related areas; one team is working only in the area of interface and naming services; one team is working in the area of commands and libraries, etc. Each sub-team can have three to four engineers in it. There are also sub-teams in the UK and US who are involved in Stern Sustenance. These teams are sandwiched between the core developers who have built and are working on newer versions of the Stern Operating System and the engineering service group that services customer needs.

There are approximately 65-70 engineers in the US, 32-33 engineers in the UK, and 32 engineers in India who are involved in Stern Sustenance. These teams comprise a Senior Staff member, one or two Technical leads, and members of technical staff (MTS). Members of the technical staff are at four levels: MTS1, MTS2, MTS3, and MTS4. Members of the technical staff are the engineers at various levels in the hierarchy.

3.1.1 Setting Up a Sustenance Center in India

This group was started in California to support the Stern operating system. Over time the demand for their services increased and problems arose. In response the group was expanded globally to better provide service to customers located in different geographical regions. It first expanded to the UK, then Europe (France and

Germany), and then a group was started in India to complete the circuit. Together these centers now provide 24X7 support to their customers.

The director of the Stern Sustenance group acted as the sponsor for the team in India. It was her decision to locate the team in India, and she was very clear on how she wanted to go about it, what she wanted to do, and how to spread a nearly uniform culture at all of the locations. The Indian team was able to blend with the existing teams without any major hiccups. As the team head stated:

Few things she did was get some people from there (US) to come here and work with the team here (India) in the initial stages . . . so that rapport is built and stuff like that . . . and initially we had people traveling either way quite frequently to get to know each other and things like that . . . so those things really helped . . . otherwise if it was only a team sitting here and working over emails and phone calls it wouldn't have come to this level . . . this is what I feel.

The Indian sustenance team is using some of the processes that were put in place by the US and UK teams, and some new processes have been developed to increase productivity. The Indian group started with a disadvantage as most of the people who joined the team had very little work experience and required training. This opportunity was used to put processes in place that were not there in the UK or US groups.

The Indian team was able to develop a well-documented process of doing the 24X7 work, which was later adopted by different teams. According to the team head, “what we have done is we have generated what we call a Process Bible, which basically is a reference material which anyone can go and have a look in case you don't know what to do next.” One of the key team member involved in the development of the “Process Bible” said:

We realized that we need something . . . a big process and bunch of documents to guide the young graduates . . . we used to call it then and we still call it the “process bible” for our group . . . so we have a thick web based book called as bible . . . an ever evolving book . . . which defines all the processes that our engineers need to know to work here . . . so we picked up lot of guys from the team here . . . seven or eight guys . . . once it was done . . . we took it to the global team and said we did that as part of ramping up the team here and obviously all these processes are applicable to the whole global team who are working on this . . . and it was totally accepted in the global team . . . so now that book is sort of sustained by voluntary effort from all over the globe and not just the team here. . . we have cut down the team maintaining that bible to two people here and we have two or three people in Europe and two or three people in US and they all together sustain the book . . . some time we get new processes and some time we remove old processes . . . so we have to keep the Process Bible correct . . . the team does that.

The top management in the organization emphasized processes and the development of team spirit. According to the Bangalore team head,

Working with team is a significant thing . . . about 25% weightage in performance review is given to the soft skills . . . things like that. When it started from the beginning this was encouraged and since every one was new and everyone was interested in becoming part of the team . . . when we started we were around five-six people and so when we started it was a small team and there was lot of opportunity for people to share and so that bonding developed . . . and once that foundation was there then it was easy to build on top of it.

The development of the Process Bible and team spirit played a major role in successfully setting up the Stern Sustenance Center in Bangalore and after almost three years of operations the team has emerged as the best team in the organization.

3.1.2 Operations

The customer support service model is comprised of four different layers: direct interaction with the customer, field support, back line engineering also known as the enterprise service group, and the fourth layer that solves the problems that cannot be solved through normal routine maintenance activities. Engineers in back line engineering separate problems into different categories depending on whether they are related to the application software, the operating system, or the hardware, etc.

The sustenance team in Bangalore takes care of service provided at the fourth level for two categories of customers: standard support customers and premium support customers. Standard support entitles customers to a response within four hours for urgent issues during extended business hours, and premium support entitles customers to engineering support twenty-four hours a day, seven days a week.

3.1.3 Technology and Quality Control

The whole service cycle is part of the Process Bible. The organization has guidelines for resolving reported bugs from the logging of the problem to its final resolution. A web-enabled tool called “Scopus” is used to recording the software bugs and all the steps taken to resolve them. Every engineer who works on a bug is expected to periodically file a report on the progress. The periodicity of the reports filed varies depending on the criticality of the bug. Reports are filed on highly critical bugs every three hours. This tool helps keep customers and other stakeholders informed about the progress made in resolving the bug.

The engineers follow a process known as Analytical Troubleshooting (ATS), which is a proprietary problem-solving framework developed by a company called Kepner-Tregoe. The organization has named it as Orion Global Resolution (OGR) process, training of which is imparted to every sustenance engineer in Orion. This framework provides a process for solving problems and doing a root cause analysis, which helps reduce the complexity involved in solving problems. It also facilitates the transfer of activities from one geographical location to another.

Each bug is also recorded in a web-enabled tool known as “Bug-Track.” When a bug is reported a Bug-Track work item is created. The item contains the details of the bug as well as a record of who reported the bug, who is responsible for resolving it, and who are the other stakeholders in this process. This tool also helps maintain a

historical record of all the bugs that have been reported and who was involved in resolving them. It also facilitates scheduling of various internal reviews and code walkthroughs.

The organization has built various tools, like Scopus and Bug-Track, which incorporate the processes mentioned and collect various metrics. The engineers working at the Bangalore development center have developed some of these tools. One senior engineer told us about one such tool:

There is something called time stamp in Scopus. Scopus currently gives 'x' amount of states, what we actually wanted to do was we wanted to reduce the amount of time it takes for us to fix a problem . . . so as to do that we needed to do some fine grained measurement of all the phases involved . . . so we came up with this event based mechanism . . . so now every time we update Scopus we kind of put some code in there which corresponds to one of the event which has been predefined with help from management team. Now there are 34 states instead of just 8 states that used to be there in the earlier version of Scopus . . . so now anytime we update Scopus we also mention which event has happened . . . so it now allows us to do a great amount of post mortem analysis . . . in terms of what are the areas that need improvement.

The quality of the service provided to customers is controlled through various such metrics and the quality of the fix is ensured through a comprehensive review process and testing. Fixes generated by this team are reviewed by an internal team code review process, the product group, a change review board, and, if necessary, by the business group before they are released to customers. This ensures that the quality of the fix, generated by this team of dispersed engineers, is not compromised in any way by the manner in which the work is being carried out.

3.1.4 Managing Work Across Space and Time Boundaries

The work managed by the group is transferred from one country to another to facilitate speedy resolution of problems. According to the team head in Bangalore:

So any time a problem comes . . . if it is hot customer problem . . . and they demand a 24x7 attention . . . then each time zone handles the problem during their time zone and passes it on to the next time zone when their daytime ends. For example if a problem comes up and we pick it up then we start working on it and if it is a hot issue then we shift it to UK at the end of the day . . . UK continues to work on it conducts the analyses . . . and then they pass it on to US. So this goes on till the problem is resolved.

Well-documented processes and the technological infrastructure put in place to facilitate 24X7 has played an important role in bridging the time-space gulf among the geographically dispersed teams.

3.1.5 Best of Stern Sustenance (BOSS)

Various other management initiatives have also played a critical role, together with processes and technological infrastructure, in managing the geographically dispersed work with ease. As the team head said:

Once we had the foundation of processes and technology that then we said now it is time for us to work toward our specific vision and mission . . . so last year our vision was to be the BOSS . . . it is "Best of Stern Sustaining" . . . so basically among the three teams we wanted to be the best . . . so we identified what are the things that we want to do best . . . so we participated a lot in global technical discussions and do quality work . . . having one single team goal . . . as a team we needed to achieve this . . . so that year we became BOSS team.

The use of internal websites for sharing information about the project statistics, team performance on various parameters, mission, vision, and specific goals has helped disseminate information across the group and also created a shared space for the group. The team regularly conducts mission and vision exercises to facilitate build-up of shared meaning. In these exercises management tries to link the vision of top management with the personal objectives of the engineers. According to the senior technical lead:

We have started a website for the team so this is all published there and then on a quarterly basis then we have our staff meeting . . . we discuss these things and tell them ok these are the objectives . . . then we sit with the individuals and chalk out what they are supposed to do to achieve their objectives . . . that will be tied to the main objectives . . . so we have specific objectives tied to teams objective and . . . and in the end of the year we go back and see what all we have done in each of these areas . . . and map it across and share it with the team.

3.1.6 Managing the People

Technical brilliance . . . the speed and accuracy with which engineers identify problems and solve them with the help of high quality fixes are the cases where we have outperformed . . . these have been the things that have been touted by outside people about our team . . . like the product group many a time have come back and said . . . Oh this is an excellent fix . . . these are seasoned engineers in the product group . . . they have been able to recognize the quality of the fixes and all that . . . so I think it is their innate ability and skills that they have developed and the demonstration of that . . .

In any software company, managing people is one of the most important concerns of the project leaders. This becomes even more important when you have to retain good people. People management becomes even more important because software professionals are well educated and they know their rights and duties. As one of the senior manager said,

We have done all these processes and tools etc . . . that is the nitty-gritty part of it . . . at the macro level you have to get the buy in of the engineers to do all these things . . . and the only way to get buy in from the engineers is possible when they say it themselves . . . basically software engineers are not like factory workers where you can force them to do things.

As we have mentioned before, management designed an objective setting exercise that links the goals of the organization to the goals of individual engineers. This exercise also helps in obtaining buy-in from engineers for achieving various objectives of the organization without compromising individual needs and objectives.

Management also facilitates healthy peer-to-peer competition while at the same time promoting teamwork. Engineers are motivated to excel in their domain of work and, at the same time, they constantly compare their skills with other members in the team. This peer-to-peer competition motivates the engineers to learn new technologies and soft skills in their spare time.

Orion itself has a performance review process where lot of weightage is given to working in team . . . in fact . . . you have heard of performance rating scheme of GE . . . that relative ranking . . . and we have adopted that recently . . . so the initial fears at least in the management was that it will lead to lot of back biting and stuff like that but it didn't turnout that way . . . people are still working the way they used to do earlier in a team.

3.1.7 Improvement in Performance Over Years

When the Indian team began operation its reputation was not worth talking about.

Two and half years back when we came online . . . you know . . . even if you goof up once it would be a big issue . . . oh you guys cannot do this . . . what we did was we focused . . . our focus was two pronged . . . one was focus on process so we virtually eliminated process breakdowns secondly we focused a lot on technical capability and maturity of engineers.

The development of processes and expertise in the team has been a self-sustaining virtuous cycle. Improvements in both arenas have led to increases in productivity for the Indian team.

Basically both of them have helped each other . . . it is an ongoing process . . . you develop expertise and then you say . . . this process is not needed you knock it off or you would say this is where we are going wrong we need a new process for that or something like that. Whatever work we are doing there are two aspects to it . . . one is the process that we follow and second is the expertise in that domain.

A senior engineer commented on the rise in productivity:

We had a graph where we charted out “time to report” to “time to resolve” metric . . . so we had a linear line which came down for seven to eight consecutive quarters . . . so that was a huge amount . . . and right now our work load is the highest . . . we are pretty much one-third the strength of the global team but we are handling more than forty percent of global defects . . . so we are delivering more.

Indian management has also come up with a new mission statement—onward and upward—for facilitating the upward trend in the improvement of productivity. According to the team head, the mission statement reflects the status of the Indian team, as other teams in the group now have them as a benchmark and before they only had to benchmark themselves against other teams in the group.

4 Findings

4.1 Increasing Role of “Shared” Space in Process Maturity

Organizations have implemented new processes to facilitate better coordination efforts of the geographically dispersed work teams. Organizations also try to increase the transparency of their geographically dispersed operations by using tools that facilitate the sharing of information and artifacts. These tools help reduce problems related to teleconferencing and emails when coordinating various activities among the geographically dispersed team members. These processes and tools have evolved over a period of time.

The Software Engineering Institute’s Capability Maturity Model (CMM) deals with process maturity for a specific location, but in the current environment, when the work has become geographically dispersed, we also need to look at how the organization is managing this transition. Therefore, we have built a process maturity model for geographically dispersed projects. The process maturity stages are:

Stage 1: Initial – Different locations do not follow any process and the software process is characterized as ad hoc and occasionally even chaotic. Few processes are defined, and success depends on individual effort.

This can be seen from the case description and the summary given in Table 1. Orion started its India operation with almost no standardized processes in place so they had to start from scratch to document the processes and then incorporate them in tools (refer to section 3.1.1).

Stage 2: Localized – Different locations have different software processes for managing activities. These are documented, standardized, and integrated into a location-wide software process. Process data is still not being collected for controlling activities within a location.

As described in section 3.1.1 of this paper, the process of development and tools was highly localized in the initial stages. Sustainance processes that were put in place

in the UK, France, and Germany were very different from those which were being put in place by the India office.

Stage 3: Sensitized – All the locations are aware of the process data and requirements for coordinating development activities, but the tools that are being put in place are still evolving as different locations fine-tune their processes to get synchronized.

As described in section 3.1.3 of this paper, Orion has put in place various web-enabled tools to coordinate the activities of its various sustenance centers. These web-enabled tools were placed only after the Process Bible was accepted by all the centers. Hence we can say that they were aware of each other’s processes at that point in time.

Stage 4: Synchronized – Organizations use tools inscribed with a uniform process, collect similar metrics, and facilitate continuous process improvement from quantitative feedback from the tools and processes in the shared domain. Control and coordination of activities are facilitated with the help of automated tools.

Orion uses, Scopus, to synchronize activities. This tool has evolved over time to capture more stages in the process so that locations are in synchronization with each other when work is being handed over from one location to another (Refer to section 3.1.3). Therefore, we can see that organizations put in place various tools to increase synchronization among different locations.

We have summarized the development of various tools and processes in the form of a table that is given below. These tools (shared repositories, online project management tools, and intranet web sites, etc.) can be considered as ‘Abstract Systems’ as defined by Giddens [4]. Over a period of time these abstract systems facilitate faceless commitments, which in-turn lead to the re-embedding of face-work commitments among the team members who are separated in time and space [4].

We discuss the impact of these abstract systems in three different domains: local, shared, and global domains. The “local” is one in which people work in their respective individual locales. The “global domain” implies individuals working from different locations towards a common project deliverable. The “shared” electronic spaces enable developers to share messages, data, or software programs with each other [5].

As we can see from the table below, documented processes and tools have been put in place to negotiate the place-space duality. These organizations started geographically dispersed operations with almost no abstract systems in place. Therefore, there were slippages in performance of the team members as they were not able to communicate and coordinate their activities.

Table 1. Time Space Configuration (at the time of starting the organization in India)

	Place: Orion India	Shared Space	Place: Orion US
Local	Independent process to handle problems	Emails, Teleconferencing	Independent process to handle problems
Global	Designated team members	Coordination team	Designated team members

Orion has put in place processes that reduce the need for telecommunication and emails, and increased reliance on processes and tools to support collaboration. Tools like Scopus, Bug Tracker, Test Director, Concurrent-Version System, etc. (refer to section 3.1.3) have reduced the need of teleconferencing to a great extent and have helped to increase trust in the systems used to coordinate activities. From the table, we can also see that there has been an increase in formalization of the operation in all of these organizations since the time they started their operations in India. These processes and tools that facilitate use of these processes help reduce the need for frequent routine communication in projects. Most of these tools and processes were implemented after studying the actual processes in use and were modified to take into consideration any change in the way things are being done (refer to section 3.1.1).

Table 2. Final Time Space Configuration of the organization

	Place: Orion India	Shared Space	Place: Orion US
Local	Similar process for Bug Root Cause Analysis, Solution Design, Coding, Unit Testing,	Documented Process Bible for doing various sustenance activities incorporated in various tools like Scopus, Bug Tracker, Secured websites, CVS, VSS,	Similar process for Bug Root Cause Analysis, Solution Design, Coding, Unit Testing,
Global	Designated team members, and Reviews at various stages.	Coordination team, Formation of Build, Change Control Board.	Designated team members, and Reviews at various stages.

These two tables show that the organization has transitioned from a state where there were no abstract systems in place to another state where these systems are facilitating most of the activities of the organization. In Orion, usage of tools like Scopus and Bug Tracker has increased from the time of its beginning of operations to the present time. These tools are now being used to take care of the processes that have been put in place to decrease the coordination related problems among the team members located in India and the US. This movement from one to state to another state can be described using the process maturity model described below.

We can see that Orion has put in more and more structures as their processes matured. In the later stages (Stage 3 and Stage 4) they started collecting quantitative data that facilitates control and coordination of the development activities. The results are tabulated below:

Table 3. Changes in Process Maturity Stages of Orion

	Commencement of the projects (1999)	At the time of the study (Now)
Orion	Stage 1 (Initial): Orion's India office had to start from scratch in designing its processes.	Achieving Stage 4 maturity (Synchronized): It has been able to put in place various tools and processes in the shared domain to control and coordinate the activities

When it started its operations in India, the processes were of a localized nature but over a period of time they have transitioned to a stage where the processes and tools from different locations are synchronized with each other (shown in table above). We can also see, from the Orion case analyses, that the organization was attempting to create a balance by emphasizing processes (section 3.1.1), people (section 3.1.6), results (section 3.1.5), and customers (section 3.1.7).

4.2 Balancing Different Perspectives

Orion's India operation was able to succeed because of the balance created among the different competing goals of a software sustenance operation. To successfully synchronize different software sustenance activities from dispersed locations, organizations need to balance four dimensions for the long-term viability of their operations:

Learning – Developers working on these projects are also interested in developing their technical and managerial skills. Organizations have to create processes that help fulfill this need for the developers. Tools put in place for coordinating the activities should also incorporate this need.

In section 3.1.6, we can see that managers are focusing on the needs of developers to write better code. Developers themselves set the targets, the number of errors, etc., and these targets are then linked with organizational requirements. This process has helped the sustenance center deliver better code fixes. The potential of developers is recognized by the best in the organization as it is transparent system and everyone is linked through different tools.

Operation – Processes that are to be followed across different geographically spread locations need to be synchronized with the help of information technology intensive tools. These tools should be able to demonstrate any improvements that need to be made for the success of any change in existing process.

In the section 3.1.3, we saw that Orion introduced a new system called Scopus, which was developed by in-house programmers. At later stages of process maturity, more stages were incorporated into this tool to manage global interactions more effectively. This shows that not only did the tool become important for operations but also changes were incorporated into it on an ongoing basis to reflect the growing needs of the teams working from different locations.

Benefits – Managers should be able to use these tools and processes for evaluation and allocation of different team members to different projects depending on their past performance.

In the section 3.1.6, we saw that some of the programmers who designed the original Stern operating system appreciated the code fix which was being developed by some of the new programmers joining Orion. This process of recognition was transparent as each code fix was allocated to a person and the senior programmers were following it very closely. This helped the unit in India to not only improve the capabilities of the programmers but it also resulted in a better allocation of work to different programmers according to their capabilities.

Relationship – These tools and processes should be able to increase transparency of the operations so as to facilitate better customer service. Customers should be able to see the benefits accruing to them and should start trusting this mode of delivery.

As is mentioned in the section 3.1.7, Orion was able to dramatically reduce the time taken to resolve customers' problems. The metrics relating to customer satisfaction were captured by the organization online and were visible to all the different dispersed offices of Orion and its customers. This helped the company develop better relationships with its customers.

We can deduce that these four dimensions are linked to each other. Better learning opportunities for developers lead to improvement in processes, which in turn lead to increases in financial as well as non-financial benefits for the developers. All of this results in better service to the customer and hence strengthens the relationship over a period of time. Better relationships with the customers lead to more projects in the future, and so on. Therefore, this cycle can continue moving forward in a virtuous fashion. The model presented in this paper is not very different from the balanced scorecard proposed by Kaplan and Norton in 1992 [6]. However, this model is a modification as it emphasizes the role of shared space in balancing these four perspectives.

We saw this virtuous cycle in the case of Orion's sustenance operation. Customer satisfaction from dispersed operations increased over time as the processes and tools, put in place by the organization attained maturity. These tools and processes were co-created by developers and managers working from different locations and were used across all the dispersed locations of the organization. As these processes and the tools inscribed with these processes were put in the shared space, the balancing of these four different perspectives assumed greater importance for sustainability of the operations over a longer period of time.

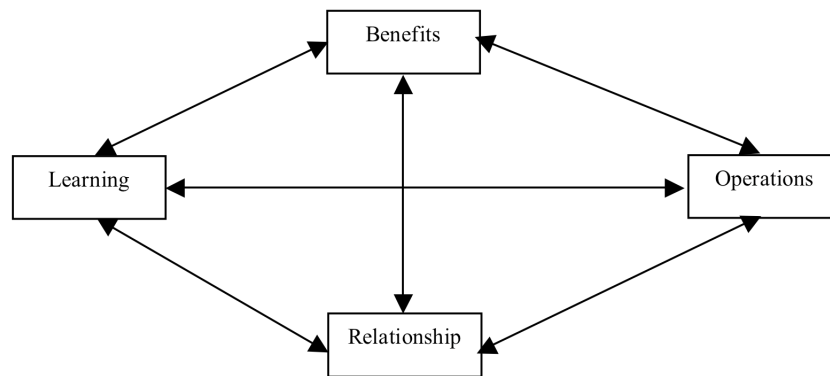


Figure 1: Balancing of different perspectives

5 Implications

Many researchers [7-10] have written about different factors that can have an impact on globally distributed work, but they have not proposed a model that can help practitioners. This study tries to build a case for a balanced model taking into consideration the different factors that other researchers have also discussed.

We have shown that “shared space” is a result of negotiations among different teams involved in globally distributed work. Initially, organizations start their operations with very little shared understanding, but over a period of time their understanding of each other gets synchronized. This was shown with the help of the process maturity model. Once created, this shared space can start a virtuous cycle, leading to a mutually satisfactory situation for all stakeholders.

Creation of this shared space leads to a balancing of the different perspectives of the stakeholders. Customers are interested in developing better relationships, benefits, and operations; developers are interested in learning, operations and benefits; and an organization is interested in all of the four dimensions of this model. Hence, balancing these four dimensions assumes great importance in globally distributed work.

The findings reported in this paper are the result of only one case. The risk of forming conclusions from this small sample may lead to generalizations that may not hold true for all cases. Additional studies are required so that we can understand how globally distributed software work evolves over a period of time. It is expected that the proposed model will facilitate further research in gaining deeper and balanced

understanding about learning, operations, benefits and relationships in globally distributed work.

References

1. J.W. Creswell, *Qualitative Inquiry and Research Design – Choosing Among Five Traditions* (Sage Publications, New Delhi, 1998).
2. A. Strauss and J. Corbin, *Basics of Qualitative Research – Grounded Theory Procedures and Techniques* (Sage Publication, London, 1990).
3. R.K. Yin, *Case Study Research: Design and Methods*, (2nd Ed.) (Sage Publications, California, 1994).
4. A. Giddens, *The Consequences of Modernity* (Stanford University Press, Stanford, California, 1990) chapter 3.
5. R. Heeks, S. Krishna, B. Nicholson, and S. Sahay, Synching or Sinking: Trajectories and Strategies in Global Software Outsourcing Relationships, *IEEE Software* April- May 2000.
6. R. S. Kaplan and D.P. Norton, The Balanced Scorecard: Measures that Drive Performance, *Harvard Business Review* 70: 1, pp. 71-79, 1992.
7. Aman and B. Nicholson, The Process of Offshore Software Development: Preliminary Studies of UK Companies in Malaysia, in: *Information Systems Perspectives and Challenges in the Context of Globalization*, edited by M. Korpela, R. Montealegre and A. Poullymenakou vol. 254 (Kluwer, 2003).
8. E. Carmel, *Global Software Teams: Collaborating Across Borders and Time Zones*, (Prentice Hall PTR, Upper Saddle River, New Jersey, 1999).
9. S. Sahay, B. Nicholson, and S. Krishna, *Global IT Outsourcing – Software Development across Borders*, (1st ed.), (Cambridge University Press, United Kingdom, 2003).
10. G. Walsham, Cross-cultural Software Production and Use: A Structural Analysis, *MIS Quarterly* 26: 4, pp. 359–380, December 2002.

About the Authors

Dr. Rajeev Sharma is an assistant professor at XLRI School of Management (India). He is also a fellow of Indian Institute of Management. His areas of interest include information system strategy of organizations, implementation of information systems, and financial valuation of benefits accruing from information system implementation.

S. Krishna is a Professor in Information Systems at the Indian Institute of Management Bangalore (IIMB). His teaching and research interests focus on managerial issues in global software & IT industry and information systems in developing countries. His recent books (authored with collaborators) are *Global IT Outsourcing: Software Development across borders* (Cambridge University Press, Cambridge) and *The Digital Challenge: Information Technology in the Development Context* (Ashgate, London).