Global Software Development: A challenge for Project Management

Marcos Eliud Guevara Camacho

a00611038@itesm.mx

May, 2004

Contents

Abstract

Today's global environment is possible because of technology. Every day, more organizations are implementing global information systems and developing software in distributed locations. Software project managers take the challenge of making the global environment the best way to build software.

This paper explains the relation between globalization and technology, and the need for globally distributed software projects. A number of issues related to project management had risen in the past years with the advent of global software development projects. Communication, development and cultural issues are explained in several cases. Also, important approaches are discussed to success in a global software project.

1 Introduction

Today's global environment is possible because of technology. Every day, more organizations are implementing global information systems and developing software in distributed locations. Software project managers take the challenge of making the global environment the best way to build software; however, important issues emerge when these projects progress.

This paper explains the relation between globalization and technology, and the need for globally distributed software projects. A number of issues related to project management had risen in the past years with the advent of global software development projects. Global software development projects exceed the compliance with classical software project management called 'issues'. Communication, development and cultural issues are explained in several cases through this research. Also, important approaches are discussed to success in global software projects.

2 Technology and globalization

Globalization is changing how we live in our societies. Most of the change is enabled by technology. In economic, social, political and legal terms, technology gives big boosts of development and automation. Today, globalization is also affecting the development of technology, specifically, Software Engineering. Many issues, such as communication patterns, global governance, security systems must be addressed by computer professionals 1. The Global Software Development is an specific issue that involves important considerations to the conventional methodologies of software project management.

Issues	**Technology enables globalization**	**Globalization drives technology**
Social	- Communication patterns - Information revolution - Gender differences	- Localization - Cyber communities
Economic	- Free trade - Telecommunications - Digital divide	- Global software development - Global business applications
Political	- Global governance - Policy initiatives	- Security systems and encryption
Legal	- Privacy and civil liberties - Intellectual property laws	- Security systems and encryption

Table 1: Principal concerns between Technology and Globalization [Gab03].

1 Introduction

Today's global environment is possible because of technology. Every day, more organizations are implementing global information systems and developing software in distributed locations. Software project managers take the challenge of making the global environment the best way to build software; however, important issues emerge when these projects progress.

This paper explains the relation between globalization and technology, and the need for globally distributed software projects. A number of issues related to project management had risen in the past years with the advent of global software development projects. Global software development projects exceed the compliance with classical software project management called 'issues'. Communication, development and cultural issues are explained in several cases through this research. Also, important approaches are discussed to success in global software projects.

2 Technology and globalization

Globalization is changing how we live in our societies. Most of the change is enabled by technology. In economic, social, political and legal terms, technology gives big boosts of development and automation. Today, globalization is also affecting the development of technology, specifically, Software Engineering. Many issues, such as communication patterns, global governance, security systems must be addressed by computer professionals 1. The Global Software Development is an specific issue that involves important considerations to the conventional methodologies of software project management.

Issues	**Technology enables globalization**	**Globalization drives technology**
Social	- Communication patterns - Information revolution - Gender differences	- Localization - Cyber communities
Economic	- Free trade - Telecommunications - Digital divide	- Global software development - Global business applications
Political	- Global governance - Policy initiatives	- Security systems and encryption
Legal	- Privacy and civil liberties - Intellectual property laws	- Security systems and encryption

Table 1: Principal concerns between Technology and Globalization [Gab03].

2.1 Global Information Systems

The development of Global Information Systems is related to Global Software Project Management. International companies need to share data, information and knowledge around the globe [Tan03]. Global information systems research is involved in areas such as development and management of e-commerce, customer-relationship management, enterprise resource planning, datawarehouse and decision-support systems. More often globalized enterprises need the power of IT to manage global operations, take global-impact decisions and obtain international indicators of performance. The transfer of knowledge is transforming these organizations into collaborative networks that help achieve corporate goals [Hus04].

3 Classical Software Project Management

The Software Project Management is a discipline widely spread. Over decades, the process of creating software has evolved from ornamental building to structured and then as a measured practice.

3.1 Project planning

Project planning is the set of actions and decisions for the future of the project. Thayer [Tha97] and Cori [Cor85] establish six steps for project planning. IEEE has published standards for project planning and are useful as concept background [Com98].

1. Define the project objectives.
2. Break down the work to be accomplished.
3. Sequence the project activities.
4. Estimate the durations and costs.
5. Reconciling the schedule with project time and resource constraints.
6. Review the schedule periodically.

Project planning is a very important task for project management because it provides a '‘start' milestone of the project and it defines the paths to walk in it. Some useful tools in project planning are Gantt [Cor85], Pert [Scu03] and WBS (Work breakdown structure) charts [Tha97].

2.1 Global Information Systems

The development of Global Information Systems is related to Global Software Project Management. International companies need to share data, information and knowledge around the globe [Tan03]. Global information systems research is involved in areas such as development and management of e-commerce, customer-relationship management, enterprise resource planning, datawarehouse and decision-support systems. More often globalized enterprises need the power of IT to manage global operations, take global-impact decisions and obtain international indicators of performance. The transfer of knowledge is transforming these organizations into collaborative networks that help achieve corporate goals [Hus04].

3 Classical Software Project Management

The Software Project Management is a discipline widely spread. Over decades, the process of creating software has evolved from ornamental building to structured and then as a measured practice.

3.1 Project planning

Project planning is the set of actions and decisions for the future of the project. Thayer [Tha97] and Cori [Cor85] establish six steps for project planning. IEEE has published standards for project planning and are useful as concept background [Com98].

1. Define the project objectives.
2. Break down the work to be accomplished.
3. Sequence the project activities.
4. Estimate the durations and costs.
5. Reconciling the schedule with project time and resource constraints.
6. Review the schedule periodically.

Project planning is a very important task for project management because it provides a '‘start' milestone of the project and it defines the paths to walk in it. Some useful tools in project planning are Gantt [Cor85], Pert [Scu03] and WBS (Work breakdown structure) charts [Tha97].

3.2 Risk management

Risk management is the set of activities that will assure -in managerial terms- that the project will succeed. Independently of the development model to use, a risk management must be checked to avoid problems in terms of budget, time and technology. Thayer [Tha97] explains potential risk in projects.

- Schedule risks
- Cost risks
- Requirements risks
- Quality risks
- Operational risks

Any of these types of risk must be addressed in some way. For example, projects could avoid, assume or prevent an specific cost risk; others could transfer the risk or get external knowledge (consulting) to overcome it.

3.3 Software Development Lifecycles

Project Management is tied to the development style of the project itself. So as a framework for this paper, we include some of the most important approaches: the waterfall method, spiral method and the object-oriented method.

Waterfall method This method was presented by Winston Royce in 1970 [Roy87]. Also called the "conventional" software process, it divides the whole process into subprocess and before to continue to the next phase (see figure 1), it is needed to fully complete the actual phase. In practice, this process overlaps and shares most of the information. These methods are still used in some projects when the risks are most controlled [Som01]. The most important subprocess are:

1. *System plan and requirements.* The services, constraints and objectives are defined through intensive consulting with the client.
2. *Detailed design.* Developing of the whole architecture according to the requirements.
3. *Code/Program building.* Most of the programming is done at this subprocess.

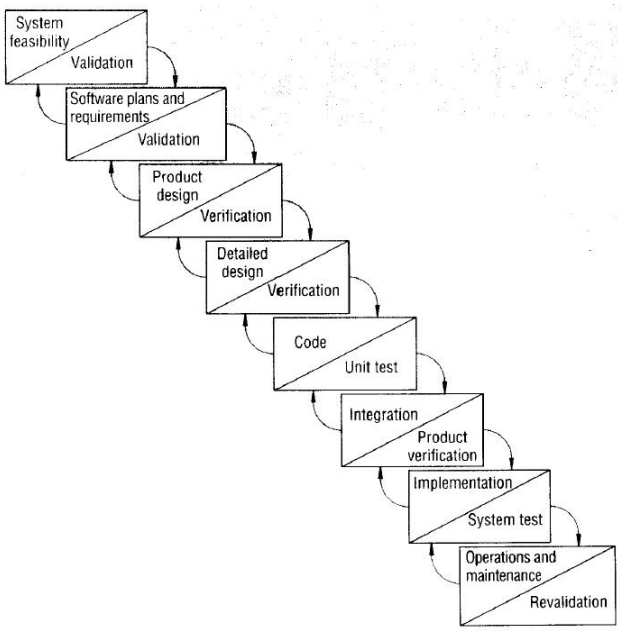

Figure 1: The Waterfall method [Boe88].

4. *Integration/Validation.* Modules of software are joined together and a set of tests are done to certify its compliance with the specification.

5. *Operations and maintenance.* Further changes in the final release are done at this subprocess.

Spiral method The spiral method was proposed by Boehm in 1988 [Boe88] and its widely known. This approach represents the succession of activities with a retrospective from one activity to another (see figure 2). With this, each cycle could have a deliverable protoype [Som01]. The quadrants refers to:

1. *Determine objectives.* In this phase, the constraints of the product are identified. Also a detailed plan is established.

2. *Risk evaluation and reduction.* Several solutions are developed to counteract the risks of software development.

3. *Development and validation.* Another method is used to develop and validate the software. This phase occurs after the risk evaluation and reduction phase.

4. *Planning.* At this stage the team takes a decision wether to continue to the next cycle or to finish the project.

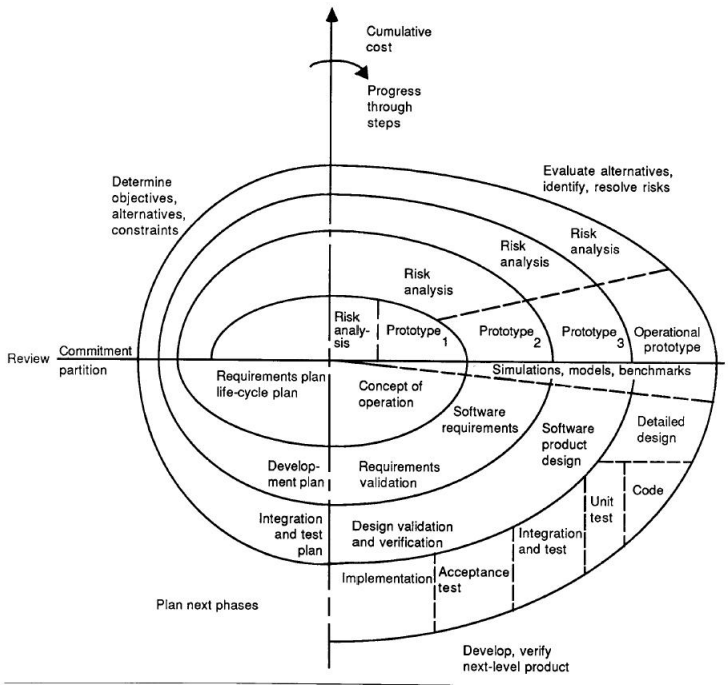

Figure 2: The Spiral method [Boe88].

Object-Oriented Method Also called "The Rational Unified Process", the object-oriented method surpass a programming methodology and takes the benefits from older methods (see figure 3). It allows sufficient control to project management and how people work[Can98]. The phases are based in problem-solving theories:

1. *Inception.* Achieving initial understanding and agreement of the deliverables.
2. *Elaboration.* Achieving initial understanding and agreement of the detailed design.
3. *Construction.* Creating the first fully functional product build.
4. *Transition.* Delivery of the product that meet the initial goals.

4 Global Software Project Management

Global Software Project Management goes to a new paradigm for building software. The main idea is to develop software (in any of the approaches before analyzed) in a distributed environment. Several points shape a global software project:

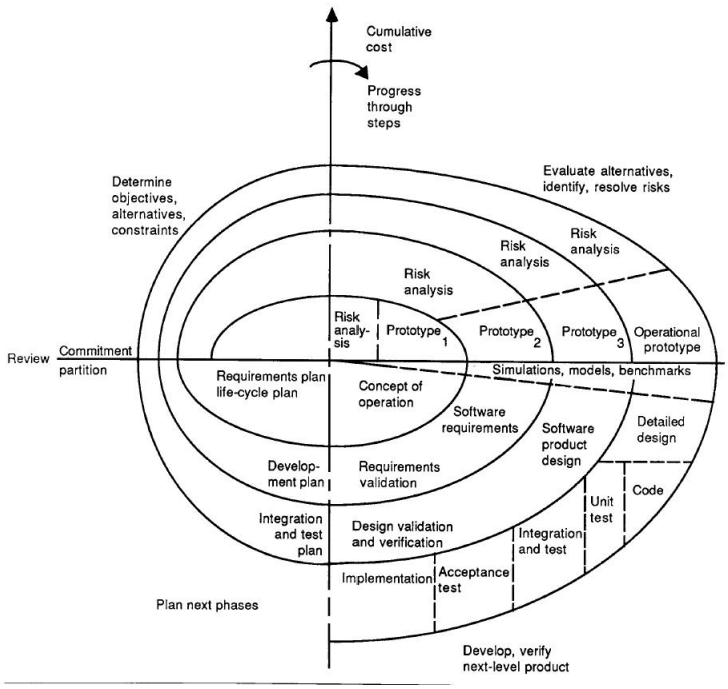

Figure 2: The Spiral method [Boe88].

Object-Oriented Method Also called "The Rational Unified Process", the object-oriented method surpass a programming methodology and takes the benefits from older methods (see figure 3). It allows sufficient control to project management and how people work[Can98]. The phases are based in problem-solving theories:

1. *Inception.* Achieving initial understanding and agreement of the deliverables.
2. *Elaboration.* Achieving initial understanding and agreement of the detailed design.
3. *Construction.* Creating the first fully functional product build.
4. *Transition.* Delivery of the product that meet the initial goals.

4 Global Software Project Management

Global Software Project Management goes to a new paradigm for building software. The main idea is to develop software (in any of the approaches before analyzed) in a distributed environment. Several points shape a global software project:

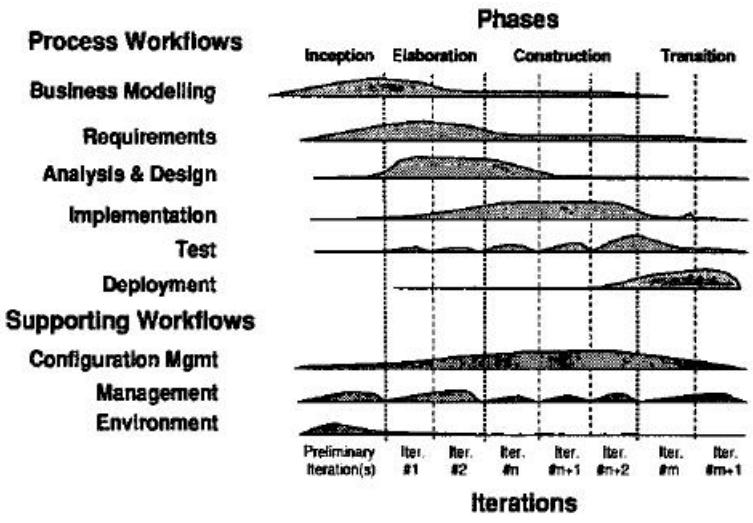

Figure 3: the object-oriented method [Kru99].

- There are teams of people in different places in the world that will participate actively in the project.
- These teams could be part of a large corporation or a consulting service.
- The project has an specified budget, deadline and a final release. Even tough these concepts may vary from approach to approach, they must be clearly established.

4.1 The need for Global software projects

At this time, many projects are developed in a global distributed environment. In 2000, at least 185 Forutne companies outsourced software development to other countries [MH01]. The most important reasons that motivates global software project are [Cam99]:

1. *Limited pool of trained workforce.* Sometimes, local team is not sufficient to the project's complexity and is needed to incorporate a foreign team (of the same company or a consulting service).
2. *The need to localize products.* 'Localize' means to specialize them to suite a local (geographical) zone.
3. *Differences in development cost.* The use of distributed teams will short budget costs.

4. *Promise of round-the-clock development.* The use of different time zones could assure all-day progress in the project.

5 Issues in global software project management

There are several cases of success in global project management. Motorola and Alcatel had built software in a distributed work environment. Motorola developed a third generation cellular system (3G) [BCKS01] using developer teams from different software centers located at United States, China, Japan, Singapore, India and Australia. Alcatel is a globally acting telecommunication supplier with multitude of development projects. Alcatel developers work conjunctly in 15 centers in Europe, Asia, US and Australia. Alcatel worked on a Java/CORBA middleware for its S12 proprietary switch system [EHSK01]. The results of these two cases were the principal source for the development of the issues mentioned above. Other helpful resource was the report from Oppenheimer [Opp02] at Bell Laboratories.

5.1 Communication issues

The communication issues in global software development reside in loss of communication richness. Physical distances, different time zones, a domain expertise are critical for the well coordination of the activities planned. A bad internal communication strategy could lead to misunderstood in any part of the software lifecycle, and increased deadlines and budgets.

5.1.1 Issue: physical distances and time zones

The time zones affect the live communication of the teams, it's difficult to make a videoconference call with teams at different places. An interdependency between teams makes communication a critical issue to cover. A travel solution of all the teams, would be impossible in cost and time terms. Motorola and Alcatel established a special strategy for this kind of projects. At the stage of analysis and design, they gather the leaders of each team at a specific place. Motorola [BCKS01] called them 'liaisons' and Alcatel 'Coaches' [EHSK01]. These leaders defined the project milestones, get all the information they needed and settle a common vocabulary of the way each one work. They later returned back to their places and shared the vision and the work they will do.

4. *Promise of round-the-clock development.* The use of different time zones could assure all-day progress in the project.

5 Issues in global software project management

There are several cases of success in global project management. Motorola and Alcatel had built software in a distributed work environment. Motorola developed a third generation cellular system (3G) [BCKS01] using developer teams from different software centers located at United States, China, Japan, Singapore, India and Australia. Alcatel is a globally acting telecommunication supplier with multitude of development projects. Alcatel developers work conjunctly in 15 centers in Europe, Asia, US and Australia. Alcatel worked on a Java/CORBA middleware for its S12 proprietary switch system [EHSK01]. The results of these two cases were the principal source for the development of the issues mentioned above. Other helpful resource was the report from Oppenheimer [Opp02] at Bell Laboratories.

5.1 Communication issues

The communication issues in global software development reside in loss of communication richness. Physical distances, different time zones, a domain expertise are critical for the well coordination of the activities planned. A bad internal communication strategy could lead to misunderstood in any part of the software lifecycle, and increased deadlines and budgets.

5.1.1 Issue: physical distances and time zones

The time zones affect the live communication of the teams, it's difficult to make a videoconference call with teams at different places. An interdependency between teams makes communication a critical issue to cover. A travel solution of all the teams, would be impossible in cost and time terms. Motorola and Alcatel established a special strategy for this kind of projects. At the stage of analysis and design, they gather the leaders of each team at a specific place. Motorola [BCKS01] called them 'liaisons' and Alcatel 'Coaches' [EHSK01]. These leaders defined the project milestones, get all the information they needed and settle a common vocabulary of the way each one work. They later returned back to their places and shared the vision and the work they will do.

5.1.2 Issue: Domain expertise

Domain expertise refers to the technical expertise in which the system will be involved. For example, local teams could have excellent IT expertise, but the systems will analyze financial operations. In classical software projects, it is common to have a domain expertise technician. In global software development is necessary to provide all distributed teams the domain expertise to avoid context problems. In the case of Motorla [BCKS01], they transmitted part of the domain expertise to the liasions.

5.2 Coordination and Development issues

The coordination issues are related to the architecture and integration of the system. This critical milestone gives the trust so the 'building does not fall out'. The success of a project depends also in a good plan and organization of activities. The architecture and development must follow team agreements. There must be a solid global strategy for the software validation.

5.2.1 Issue: Architecture and Development

Motorola based its architecture with strong principles like well defined interfaces, correct use of semantics and low coupling between elements. Alcatel used a continuous design validation process to obtain the best result in the architecture. An important strategy is to publish the efforts at the development stage with good documentation. This will help to fast the accordance with the interfaces of the software pieces.

5.2.2 Issue: Software validation

Both cases confirmed that the best method is the integration testing. Even tough they started by teams to make unit testing, the best way they assured the quality of the product was joining all the pieces of software to one single environment. They recommended that if the architecture is by layers, is better to start with an up-down approach.

5.3 Cultural and geographical issues

The global software development underlies on the intercultural communication and strategy to success the geographical issues. The vendor support is one of the most hiding issues in this topic as also the government restrictions. Most of the success of

the other techniques could be useless if the management team (leaders and liaisons) are not aware of this specific issues.

5.3.1 Issue: Global team integration

Global team integration refers to the level confidence and friendship between all the local teams. Due to cultural differences, teams tend to manage an "us vs. they" attitude, lack of support and trust [Opp02] and [MH01]. Managers without cultural sensibility usually provoke several misunderstandings between teams [MY01].

5.3.2 Issue: Vendor support and Government restrictions

Is better to count on commercial platforms than in open source technologies for the international technical support without the same coverage. Also is important to check the boundaries between countries and times to make any procedure. It could get as far as 4 months to obtain a visa and it would retard the overall process of the project.

6 Approaches in global software project management

6.1 Intercultural analysis

Yohama [MY01] proposes a framework to overcome intercultural issues. This framework studies cultural behavior, manages communication awareness and includes international considerations into the risk analysis. Figure 4 illustrates the primordial considerations in an intercultural analysis.

Cultural behavioral study. At the process of staffing, its important to make a cultural behavior matrix including these four concepts vs. the types of cultures. This would give an overview of potential problems when these cultures will relate in future phases of the project. Yohama suggest four concepts to consider in a study but more concepts could be included [MY01]. Table 2, an example is suggested to identify differences between cultures.

- *Power-Distance* is how long does the power reach. Cultures with high power-distance level leave decisions to higher rank executives. Low power-distance cultures have more empowerment environment.

the other techniques could be useless if the management team (leaders and liaisons) are not aware of this specific issues.

5.3.1 Issue: Global team integration

Global team integration refers to the level confidence and friendship between all the local teams. Due to cultural differences, teams tend to manage an "us vs. they" attitude, lack of support and trust [Opp02] and [MH01]. Managers without cultural sensibility usually provoke several misunderstandings between teams [MY01].

5.3.2 Issue: Vendor support and Government restrictions

Is better to count on commercial platforms than in open source technologies for the international technical support without the same coverage. Also is important to check the boundaries between countries and times to make any procedure. It could get as far as 4 months to obtain a visa and it would retard the overall process of the project.

6 Approaches in global software project management

6.1 Intercultural analysis

Yohama [MY01] proposes a framework to overcome intercultural issues. This framework studies cultural behavior, manages communication awareness and includes international considerations into the risk analysis. Figure 4 illustrates the primordial considerations in an intercultural analysis.

Cultural behavioral study. At the process of staffing, its important to make a cultural behavior matrix including these four concepts vs. the types of cultures. This would give an overview of potential problems when these cultures will relate in future phases of the project. Yohama suggest four concepts to consider in a study but more concepts could be included [MY01]. Table 2, an example is suggested to identify differences between cultures.

- *Power-Distance* is how long does the power reach. Cultures with high power-distance level leave decisions to higher rank executives. Low power-distance cultures have more empowerment environment.

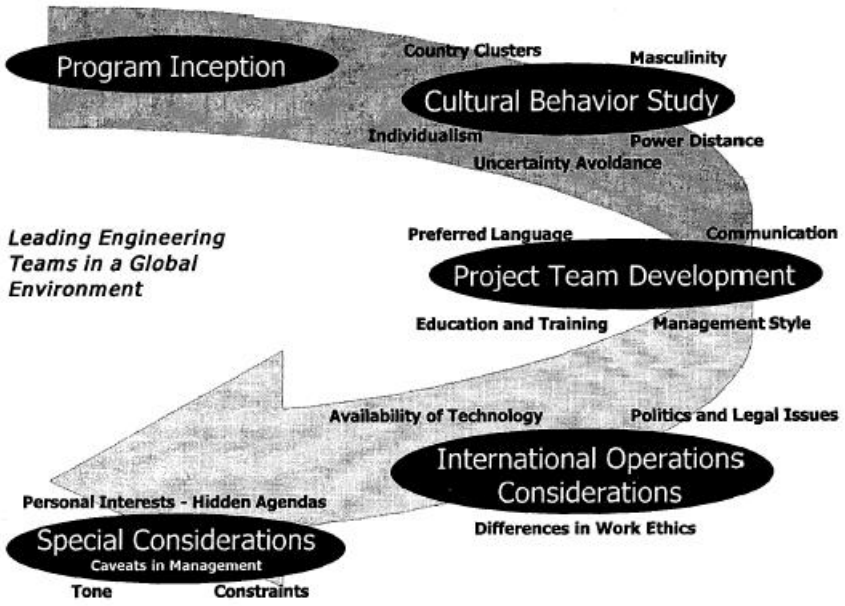

Figure 4: Framework for intercultural analysis [MY01]

- *Uncertainty avoidance* is the character that cultures held to ambiguous situations. Cultures with high uncertainty avoidance will prefer structured and high detailed approach.
- *Individualism* is "the society's acceptance of individual performance over group performance".
- *Masculinity* is a the level of importance of material values to a culture. A feminine culture will place more emphasis to society contributions.

Culture	**Power Distance**	**Uncertainty avoidance**	**Individualism**	**Masculinity**
Mexican culture	*On high executives*	*Low*	*Low*	*Medium*
German culture	*Distributed*	*Very High*	*Medium*	*High*

Table 2: Example of cultural behavior matrix

Global project team development This step does not refer to the software development but to the development of the persons involved in it. Specific considerations must be analyze.

- Preferred language. This policy must be established in project planning. Factors to take this decision could be cost of translation-interpretation, language of the end-customer and capability of most of the members to speak the same language.
- Communication methods. A special policy that defines the reporting levels. Yohama suggest to keep communication simple and easy to follow to avoid high costs and increase quality (Fig. 5). Some tools available are [HMFG01]:
  - Instant Messaging tools. Rather than commercial IM tools, there exists other communication tools specialized for group communication. This tools offers live status of the persons working on a project.
  - Videoconference meetings. This tool is one of the most used in GSP. Motorola [BCKS01] used it widely to avoid high costs of travel and offers a good face-to-face communication. The only constraint that this technology imply is the variety of time zones and could make a team not available to a conference.
  - Groupware and Centralized CM tools. These tools are commonly off-the-shelf and could be integrated to IM and videoconference tools. The advantages of centralized communication tools is that an update is casted automatically to all teams.
  - Travel. Travel is an expensive and tedious technique but achieves the best global team integration. A wise use could be at requirements and integration phases of a project and will include only to local team leaders. If this technique is used, it must be clearly stated in the budget: quantities and destinations.
- Education and training. Consider any overbalances in technical knowledge of teams as a project risk. This is one of the blurry problems for project management. Is not good to that assume a engineering degree in one country is equivalent to another country.

International operations considerations. Three main areas must be considered in operating a project management. The availability of technology in a country could stop partially the progress in an specific team (i.e. constant phone, electricity or internet services breakdown). In addition, politic and legal issues must be addressed in risk management (i.e. visa issuing, legal constraints of development and/or implementation)

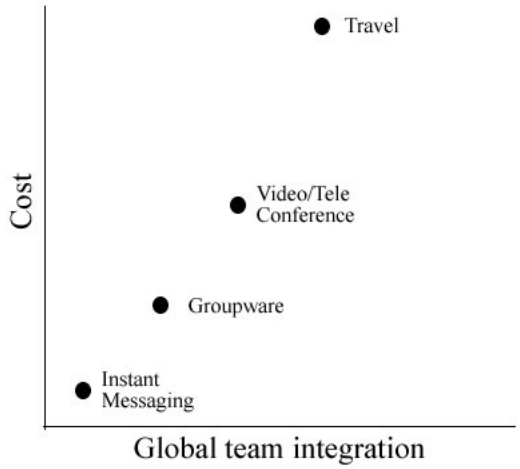

Figure 5: Global team integration vs. cost

6.2 Management strategies

Kim and Peterson [KP03] consider some strategies that will help the successful finalization of a software project. These managerial strategies focus on overcoming communication and development issues discussed earlier. The following presents a justification of each of the strategies selected in the current report.

Top management support. A research from Kweku Ewusi-Mensah [EM97] shows that abandoned information systems development is caused by the lack of support from top management. The active participation of senior management making decisions of critical junctures is essential, especially in global software development. When important decisions are delegated to a local team it may not have the whole picture of the project and most of the times leads to never-end project expansion in cost, and time terms.

Proper project scope. Scope defines the boundary of the project or what aspect of the system will be included in the project [KP03]. Liebowitz [Lie99] identified that a very big scope projects tend to have built-in difficulties and levels of complexity that may frustrate to all the global team. However, a small project scope that do not satisfy requirements may require more corrections or adaptations to comply them.

Project's leader monitoring/control. In global software projects is indispensable a measurement control. In The benefits of measure each local team performance gives a complete idea of global performance and helps to identify new risks. For teams that are part of a global or international company, and standard of measure should be established.

Adequate training for local teams. After a cultural analysis for the project, training programs should be implemented over specific local teams. These training programs are about IT and not about the domain of the system. Improving IT expertise is well received in most cultures and they will be capable to perform the same tasks. This training could be about groupware, products in support of development or IT concepts. Training cost must be considered as a cost risk because of the complexity of training logistics [Opp02].

7 Conclusion

Most of the issues and approaches for global software project management are similar from classical project management; however the global complexities make them a difficult task that could counteract the primarily reason of a global software project. Managers must develop special strategies when they administer global software projects. Only a good planning, correct risk management and a specialized intercultural sensibility will bring the success of global projects.

Project's leader monitoring/control. In global software projects is indispensable a measurement control. In The benefits of measure each local team performance gives a complete idea of global performance and helps to identify new risks. For teams that are part of a global or international company, and standard of measure should be established.

Adequate training for local teams. After a cultural analysis for the project, training programs should be implemented over specific local teams. These training programs are about IT and not about the domain of the system. Improving IT expertise is well received in most cultures and they will be capable to perform the same tasks. This training could be about groupware, products in support of development or IT concepts. Training cost must be considered as a cost risk because of the complexity of training logistics [Opp02].

7 Conclusion

Most of the issues and approaches for global software project management are similar from classical project management; however the global complexities make them a difficult task that could counteract the primarily reason of a global software project. Managers must develop special strategies when they administer global software projects. Only a good planning, correct risk management and a specialized intercultural sensibility will bring the success of global projects.

References

[BCKS01] Battin, Crocker, Kreidler, and Subramanian. Leveraging resources in global software development. *IEEE Software*, 18(2):70–77, April 2001. IEEE Xplore 914750.

[Boe88] B. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, May 1988. IEEE Xplore.

[Cam99] E. Camel. *Global Software Teams*. Prentince-Hall, Saddle River, NJ, 1999.

[Can98] M. Cantor. *Object-oriented project management with UML*. Wiley Computer Publishing, New York, NY, 1998.

[Com98] The Software Technical Comittee. *IEEE Std 1058.1-1998. IEEE standard for software project management plans*. Computer Society of the IEEE, USA, December 1998. IEEE Xplore 741937.

[Cor85] K. Cori. Fundamentals of master scheduling for the project manager. *Project Management Journal*, 16(2), June 1985. Proquest Direct DID 1272892.

[EHSK01] Ebert, Hernández, Suttles, and Kolarczyk. Improving validation activities in a global software development. In *Proceedings of the 23rd International Conference on Software Engineering*, pages 545–554. IEEE, June 2001. IEEE Xplore 919129.

[EM97] K. Ewusi-Mensah. Critical issues in abandoned information systems development projects. *Communications of the ACM*, 40(9):74–80, 1997. ACM Digital Library DOI 2600775.

[Gab03] P. Gabbert. Globalization and the computing curriculum. *SIGCSE Bull.*, 35(2):61–65, 2003. ACM Digital Library DOI 782982.

[HMFG01] Herbsleb, Mockus, Finholt, and Grinter. An empirical study of global software development: distance and speed. In *Proceedings of the 23rd international conference on Software engineering*, pages 81–90. IEEE Computer Society, 2001. ACM Digital Library DOI 381481.

[Hus04] E. Hustad. Knowledge networking in global organizations: the transfer of knowledge. In *Proceedings of the 2004 conference on Computer personnel research*, pages 55–64. ACM Press, 2004. ACM Digital Library DOI 982384.

[KP03] Kim and Peterson. Critical strategies for information systems development projects: Perceptions of developers from the united states and japan. In F. Tan, editor, *Advanced Topics in Global Information Managment*, volume 2, chapter 13, pages 251–267. Idea Group Publising, Hershey, PA, 2003. T 58.5.A39 2003 v. 2.

[Kru99] P. Krutchen. *The Rational Unified Proccess. An Introduction.* Addison Wesley, Reading, MA, 1999. QA 76.76.D47 K79 2000.

[Lie99] J. Liebowitz. Information sytems: Success or failure? *Journal of Computer Information Systems*, 40(1):17, Fall 1999. Proquest Direct DID 45482580.

[MH01] Mockus and Herbsleb. Challenges of global software development. *Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International*, (2001):182–184, April 2001. IEEE Xplore 915526.

[MY01] M. Mar-Yohana. Leading engineering teams in a global environment. *Frontiers in Education Conference, 2001. 31st Annual*, 3:S2D – 6–S2D–12, October 2001. IEEE Xplore 964025.

[Opp02] H. Oppenheimer. Project management issues in globally distributed development. In Daniela Damian, Frank Maurer, and Nigamanth Sridhar, editors, *International Workshop on Global Software Development ICSE*, pages 47–50, Orlando, FL, May 2002. University of Technology, Sydney. <http://www.cse.ohio-state.edu/ nsridhar/ICSE02/GSD>, last visited May 7th, 2004.

[Roy87] W. W. Royce. Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering*, pages 328–338. IEEE Computer Society Press, 1987. ACM Digital Library DOI 41801.

[Scu03] P. Sculte. *Complex IT project management : 16 steps to success.* CRC Press, Boca Raton, FL, 2003. T 58.64.S38 2003.

[Som01] I. Sommerville. *Software engineering.* Addison Wesley, London, 6 edition, 2001. QA76.758 .S657 2001.

[Tan03] F. Tan. *Advanced Topics in Global Information Managment*, volume 2. Idea Group Publising, Hershey, PA, 2003. T 58.5.A39 2003 v.2.

[Tha97] R. Thayer. *Software Engineering Project Management.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 1997. QA 76.758 .T87 1997.