ETM
ENGINEERING & TECHNOLOGY MANAGEMENT

Title: Monitoring and Controlling Contractors in Software Development

Course:
Year: 1994
Author(s): J. Alvarez, i. Kan, M. Posovich, F. Rivera, A. Uslu, J. Weisensee and W. Yuwana

Report No: P94022

Abstract: This project is to investigate the pros and cons of using contractors for software development project; to elaborate the potential problems of using outside contractors in this particular case. The areas explored are: contractor logistics, cultural differences, incentive difficulty, scheduling problems, quality assurance issues, project termination, software risk management,. For each of the potential problem we suggest a set of counter measures and early warning signals. It is integrated that the counter measures with other common risk factors and it is developed that an integrated methodology for software project risk management.

MONITORING AND CONTROLLING CONTRACTORS IN SOFTWARE DEVEPLOMENT

J. Alvarez, L. Kan, M. Posovich, F. Rivera
A. Uslu, J. Weisensee and W. Yuwana

EMP - 9422

EMGT 545/645

PROJECT MANAGEMENT IN ENGINEERING & TECHNOLOGY

SPRING 94

Monitoring and Controlling Contractors in Software Development

BY

JOHN ALVAREZ

LING KAN

MIKE POSOVICH

FERNANDO RIVERA

AKIN USLU

JOHN WEISENSEE

WIRA YUWANA

FOR

Dr. DUNDAR F. KOCAOGLU

OBJECTIVES 4
PROCEDURE 5
1 INTRODUCTION 6
1.1 ADVANTAGES AND DISADVANTAGES OF USING OUTSIDE [SUB-]CONTRACTORS 7
1.1.1 Advantages 7
1.1.2 Disadvantages 7
1.2 GENERAL PROBLEMS 8
1.3 SOFTWARE LIFE CYCLE 10
2 SOFTWARE CONTRACTOR LOGISTICS 14
2.1 GENERAL POTENTIAL PROBLEMS 16
2.2 SPECIFIC PROBLEMS CREATED 16
2.3 POTENTIAL SOLUTIONS 17
2.4 CONCLUSIONS 20
3 CULTURAL DIFFERENCES 21
3.1 WHAT IS CULTURE DIFFERENCE 23
3.2 CULTURE INFLUENCE IN BUSINESS AND PROJECT 25
3.3 CONCLUSION 27
4 INCENTIVE DIFFICULTY 28
4.1 PRODUCT DEFINITION 29
4.2 INTELLECTUAL PROPERTY PROTECTION 30
4.3 PAYMENT STRUCTURE 31
4.4 CONCLUSIONS 31
5 SCHEDULING SOFTWARE PROJECTS 34
5.1 IMPLEMENTATION 35
5.2 THE USE OF A SINGLE SCHEDULER 38
5.3 SCHEDULING AND COMPUTERS 39
5.4 SCHEDULING AND AI 40
5.5 SCHEDULING AND COMPRESSION 41
5.6 SCHEDULING TIME AND STATISTICAL DISTRIBUTION 42
5.7 SOFTWARE COST ESTIMATIONS 43
5.8 COST-BENEFITS ELEMENTS TO BE CONSIDERED IN SOFTWARE ESTIMATIONS 50
5.9 CONCLUSION 52
6 QUALITY FOR CONTRACTORS 55
6.1 QUALITY MEASURES 55
6.2 LEVEL ASSESSMENT OF THE SUBCONTRACTOR 57
6.3 MANAGEMENT STYLES ACCORDING TO LEVELS 59
LEVEL 1 59
LEVEL 2 59
LEVEL 3 60
6.4 PROPOSED SQA PROGRAM 60
DEFINITION 60
6.4.1 Goals 61
6.4.2 Role 61
6.4.3 Responsibilities 61
6.4.4 Relation of SQA with subcontractor 62

Objectives

- Investigate the pros and cons of using contractors for Software Development Project
- Elaborate the potential problems of using outside contractors in this particular case.

  The areas that we explored were:

  a) Contractor logistics

  b) Cultural differences

  c) Incentive difficulty

  d) Scheduling problems

  e) Quality assurance issues

  f) Project termination

  g) Software risk management

1 Introduction

Contracting means acquiring an external resources to accomplish specific task or tasks for a price in a finite amount of time. There are various reasons for hiring contractors. In this report, we concentrate on the cases of hiring contractors to (help to) develop software packages.

Roughly there are two reasons to hire a software development contractor. The first reason is to temporarily fill a position because it is a short term task, or because it is cheaper than hiring a permanent employee, or because of heavier than expected work load or because of other kinds of short term resource problems. The second reason to hire a contractor is the contractor have expertise that is not cost-effective for the hiring company to re-develop. Notice, because of the high skills required to be a software contractor, even when a contractor was hired for the first reason, he or she usually is a highly skilled person. The distinguish here is that in the second case, in addition to the general software development skills, the contractor have specialized knowledge that is not found in the hiring company.

In this report, we emphasis our discussion on the second kind of contracting. As was just pointed out, the reasons of not to develop the skill internally is cost-effectiveness and time-to-market. Often it is much cheaper to acquire the technology through

- If the subs are far away, then the 'soft' but essential intimacy and learning that come from day-to-day contact will be absent.

- The danger of 'sub' your soul away. If not careful, a firm could sub away its core competency.

- There is less 'loyalty' in subs than in employees.

1.2 General Problems

Software contracting can be viewed in terms of trust. Lacking a fundamental trust to a contractor, the contracting party will attempt to do the following:

- Start with a very precise statement of what was wanted.

- Insist on rigid standards and detailed documentation of every process step.

- Require that each step be completed and approved before the next step was initiated.

- Demand a firm commitment at the outset.

The project is in accordance with the Product Plan. The project then proceeds in two steps. Firstly, the system architecture is established: this determines the hardware and software trade-offs. Then, each subsystem is specified to satisfy the Functional Specifications. As a result, the subsystem specifications are produced which allow the Product Plan to be redefined. The software cost, time and performance estimates play an important role here as it can influence the hardware and software trade-offs.

3. Program Construction

During the construction phase, the software modules are coded and tested. The construction costs, time and performance of a system will vary widely, depending upon such factors as the type of applications, the product complexity, language level, personnel attributes, use of software tools, etc.

4. Documentation

Software projects by nature generate an enormous quantity and diversity of paperwork, documents, diagrams and tables. The cost associated to the time devoted to this activity might represent a significant percentage of the entire cost of the development.

5. Program Test

The tested modules during construction phase are integrated into sub functions, which are then tested; subsystems are integrated and tested in their target processors and then the system is integrated and tested. Is in this step that last opportunity to check the performance of the software before to deliver to the users.

6. Maintenance

During this phase the product is update and adapted to match changing organizational need. The maintenance costs of a system will vary widely, depending upon such factors as the applications, the complexity of the system, and the needs for periodic update.

2 Software Contractor Logistics

When a contracting software company is developing software at a site other than at the contractor's site, "Logistics" are the tools or means used by a contracting company and a contractor company to communicate and provide access as needed to satisfy the terms of the contract.

Logistics can be an important factor to consider if:

- The contracting company will accomplish the contract from a remote site. (i.e.: The software will be developed at a different site than the contractor's.)
- The contractor desires to have reports regarding progress and status of a project.
- The project design consists of installing software on a mainframe.

If the factors above are important, the logistics to be considerations may be different for a local company versus a company that is not local. Contracting a local company, a company in the US, can be fairly simple. The time zones are not that extreme. However, many companies are going abroad to satisfy their software needs. India software companies have had a noticeable effect on the world software market. The World Bank last year surveyed 150 prominent US and European hardware and software manufacturers that ranked India's programmers as No. 1 out of eight countries for both on-site and offshore software development, ahead of Ireland, Israel, Mexico, and Singapore[2]. Only the United States and possibly Russia have more software engineers than India[3]. Dealing with a software company in a remote area such as India can be a concern to a company. Motorola was attracted to India by the talent and the relative low cost of software development.

"But despite having access to some of India's most talented software engineers, Motorola's management remained concerned about its ability to control an operation 8,855 miles from corporate headquarters, particularly in a country with poor communications and difficult transportation links. 'We were prepared for it to fail,' Mr. Braun (Motorola Senior Vice President) said. 'The key concern at the outset was quality. The key concern was how do you manage something so far away.'"[4]

We will consider both the logistics for dealing with a local company and the extreme of dealing with an Indian software company for this study since the contracting of an Indian company encompasses all of the potential logistical problems associated with contracting.

---

[2] Bharagava, Sunita Wadekar, "Software from India? Yes it's for Real", Business Week, January 18, 1993.
[3] Gargan, Edward A., "India Booming as a leader in Software for Computers", New York Times, December 29, 1993.
[4] Ibid.

2.1 General Potential Problems

The general problems of logistics that may be considered when dealing with a software contractor are:

- If the contracting company has a mainframe computer, is the contractor company required to have access?

- What are the contracting company's requirements for status reporting? Is it daily, weekly, or monthly?

2.2 Specific Problems Created

1. If status reports are required, what are the modes of communication required?

If the requirement is to have a live person to interface, the distance can be a problem. India is 8500 miles flying distance away.

If the communications are required via phone, some foreign countries, especially India, are known for poor communications links.[5]

---

[5] Ibid.

There can be a time zone difference. There is a 10 to 11 hours difference between India and the United States.

If letters are sufficient for monthly reports, for India the mail system is unreliable.

Telex, fax, E-mail all may require phone lines which can be a problem in India as discussed above.

2. If access to a main frame is required, how is networking the computer systems done between contractor and contracting company.

Time Zone and geographic distances can be a problem for networking. How do you network then there is a 10-11 hour time zone difference and 8500 miles between India and the mid-continental US?

2.3 Potential Solutions

1. The contracting company can position a permanent on-site person to facilitate communications. This will preclude the requirements for people to do numerous flying back and forth.

2. Remote daily and weekly communications can be handled by new technologies of video-conferencing with desktop applications have became a reality in 1993 through the efforts of PictureTel Corp. and several startup companies.[6] If the companies are to network, this new technology will facilitate communications between the two companies' software engineers.

3. Remote daily and weekly communications can be handled by E-mail, fax or phone if the phone lines are reliable.

4. Remote daily and weekly communications can be handled by interactive video-conferencing.

5. If phone lines are unreliable, some companies are finding other solutions. An example is The Deere and Company. They have arranged their own methods. The contract is for a multi-long-term project developing and delivering code from Satyam Computer Services of India to The Deere & Company in Moline, Illinois. The contract required hooking into Deere's mainframe computer. To do so required high speed data communications links that was bought from India's national telephone carrier and from AT&T stateside. A 486 bridge PC in Madras, India ties data from the gateway into the Indian telephone network, which is routed by microwave transmission to an earth station in Bombay for an uplink to an Intellsat satellite. From there, information is bounced to an earth station near London

---

[6] Vizard, Michael, "Video-conferencing viability on way", Computerworld, May 17, 1993, p. 37.

and routed via AT&T phone lines to Deere's Moline data center.[7] Altogether, the logistics included the following:

- 50 development PCs
- a 486 gateway PC
- a 486 bridge PC
- VSNL (India telephone) phone link to earth station
- Intellsat satellite
- London earth station
- AT&T cable routing to Moline data center

6. An Internet dial-in system can be used to link the contracting and contractor PC and or mainframe.

7. The time difference problem can sometimes be beneficial. For Deere, since India is 10 and 1/2 hours ahead, the daytime development in India accesses Deere's mainframes in the dead of night, when the computers are relatively untaxed and far more accessible than they are during business hours at Deere.[8] Time zone differences in the US are not as severe and can be managed.

---

[7] Halper, Mark, "Deere's faraway IS solution", Computerworld, February 15, 1993.
[8] Ibid.

2.4 Conclusions

The company must resolve the issues of logistical communications and it should be written into the contract what the communication requirements are. Contracting companies require a status report on a results oriented contract at least once a month. The software contractor should have the ability to be able to communicate with the contracting company on a live basis. Companies use telephones, in person contact and video-conferencing to achieve the logistics of monitoring software development. In addition to the phone system, another form of communication should be set up to provide "real time" discussions while troubleshooting problems. Faxing sketches or drawings is a basic necessity in today's business environment and are moving beyond to video conferencing and interactive desktop video conferencing. Networking a contractor's system with the contracting company's system is a method to provide access for remote software engineers. This may be necessary if the contracting company is providing the mainframe capabilities.

3 Cultural Differences

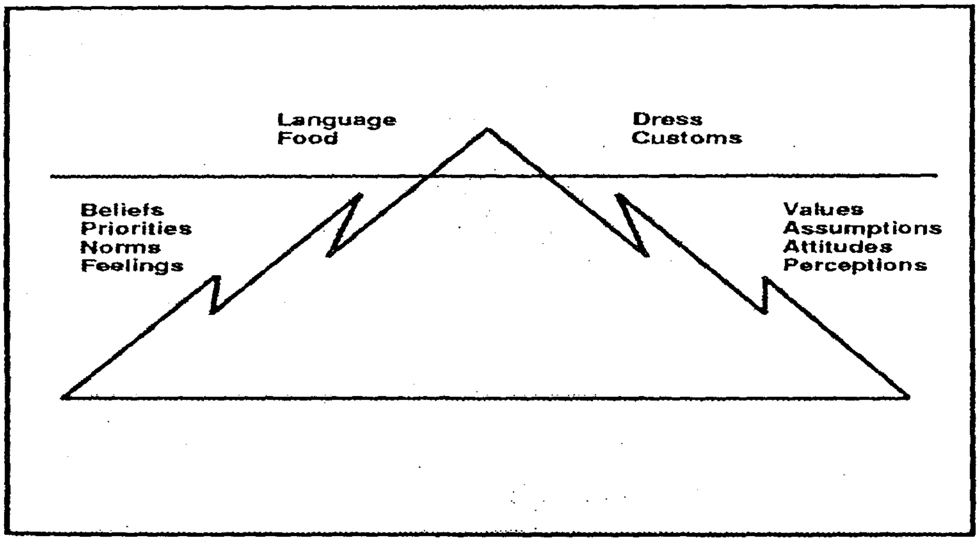
Business is exchange between people. Person in their role as employees are bound each other in exchange relationships. Both companies and persons can be considered as actors in relationship. In almost every project and business we must deal with people. Sometimes we must deal with people from other sites, organizations, and countries. In this part we want to discuss about cultural influence in the project when the organization need to deal with outside contractors. These outside contractors may be in different site, different level of industry, and different country.

In general culture can be defined as the way of life, the sum total of one's philosophy, belief, norms, values, morals, habits, customs, art, and literature. These variables are interlocked each other and ultimately influence the behavior of individuals and groups.

picture 1

(from Aviel, 'Cultural Barriers to International Transactions', Journal of General Management, vol 15 no 4 Summer 1990, p 6)

Culture can be manifested in many ways: dress, language, food, gestures, etc. Most of the cultural components are less visible. Harris and Moran compare culture to an iceberg, only the tip is visible while the bulk is concealed.



picture 2
(from Aviel, 'Cultural Barriers to International Transactions', Journal of General Management, vol 15 no 4 Summer 1990, p 7)

Different organization has different culture. This difference become more significant when the organizations are located in other countries. Culture is often considered to be a factor causing difficulties in international business.

3.1 What is Culture Difference

Culture can be defined on a national, business, organizational, and personal level.[9] National culture may be the first to come to mind when the culture is mentioned. This level include language, social structure, manifestation of power, religions and philosophies that permeate the attitudes of the society.

Business level is a subculture of national culture in the sense that it involves a set of ideas that are generated by values in the national culture but are different in certain respect.[10] This culture is more like values and norms connected with the type of activities in the business. The group of business cultures in a country have common characteristics in some aspects from national culture. The power distance between the hierarchical levels, willingness to take responsibility, business behavior, the value of agreement, the degree of formalization, the extent to which business is socially bound, etc. are largely determined by national culture. Business culture may influence particular business in several countries.

The culture of the organization can be used to explain the pattern of basic assumptions developed by the organization as a group.[11] The tolerance and expectation of the exercise of the power and prestige within the organization and between higher and lower levels in the hierarchy is the part of organization culture

---

[9] Forsgren-Johanson, Managing Networks in International Business, Philadelphia: Gordon and Breach, 1992, p. 49.
[10] Ibid. p. 50.
[11] Ibid. p. 50.

Some managers are reluctant to think about cultural difference problem. They think that it is discrimination to talk about culture difference. But culture difference is an exist fact and that will generate problem if we do not consider it in doing business, especially when deal with organization from other country. Culture has a profound impact on the attitudes, priorities and behavior of individual and To deal with people from different organization or countries, a manager should understand their values, norms, and priorities.

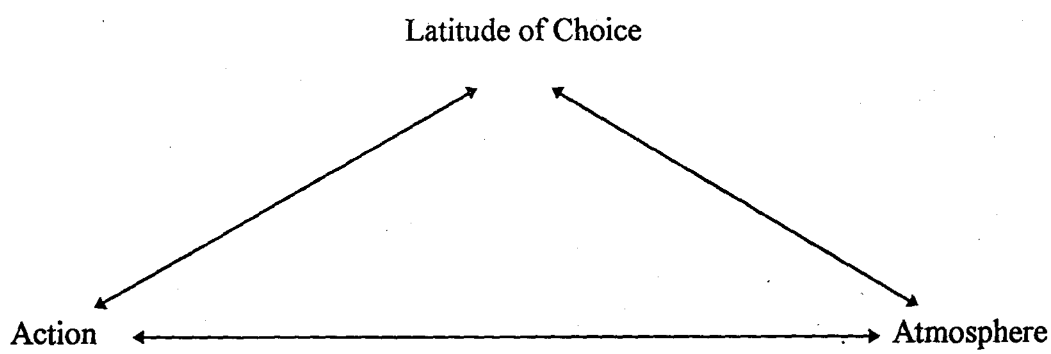
3.2 Culture Influence in Business and Project

A business relationship must have mutual orientation. From this mutual orientation two or several companies establish a mutual intention to continue an exchange and thus become bound to each other. This condition will develop a emotional setting for the people who interact in this environment.

Culture influences the atmosphere, the emotional setting in which the interaction take place. Atmosphere in a relationship indicates why the interest of both parties is to reduce factors that could damage the relationship.

Atmosphere is an important, intangible aspect of relationship. the atmosphere differs from feeling. Feelings are the interpretations made by a party to relationship on the basis what he knows or assumes about the activities performed by other party in the relationship. In other hand, the atmosphere is produced by the feeling of both parties.

The atmosphere has interrelation with latitude of choice and action profile. Latitude is what can or must be done. Action profile is what actually is done. Picture 3 shows the interrelation among those elements.



Picture 3

([1] Forsgren-Johanson, Managing Networks in International Business, Philadelphia: Gordon and Breach, 1992, p. 55)

To reduce cultural distance, all parties in the relationship must develop common rules to guide the relationship and the activities. When the rules are broken, the activities impinge the atmosphere and change it from its status quo and influence the relationship.

A common mistake when discussing cultural difference and their influence on business relationship is to mix this element with different level of technical factors. Some manager identify cultural difference as shortage of resources and this make the problem greater.

Culture become important to relationship if the rules are broken or the relationship shifts in some other way. Cultural differences blossom as the atmosphere alters from one to the other as this cancel out the rules.

3.3 Conclusion

A manager that handle projects in different sites or countries must aware that there are cultural differences and distance among people that involve in those projects. To reduce those cultural distance a manager must develop an atmosphere that support that purpose. A manager must distinct between cultural differences and cultural distance. Cultural differences is something that have been exist. We can not do much to eliminate it but we can develop a situation to reduce the cultural distance.

4 Incentive Difficulty

Contracting with an international software developer can create incentive difficulties as well. There is a great deal of uncertainty involved in software development. This uncertainty adds to the difficulty of offering adequate incentive to all parties in a joint project. The contractor generally knows more about the costs of developing the software, while the user knows more about the benefits of the project. Each parties' main incentive is to maximize their own return on the project and they do not care as much if the other parties' objectives are met. The large number of software development projects that fail with huge cost overruns and long delays point out these difficulties.

Incentive difficulties can be traced to the contract that the user and developer sign. There are three main areas to consider in the contract that are directly related to incentive. They are product definition, intellectual property protection and the payment structure[13]. These areas must be considered in detail to define incentives in joint projects.

[13] Whang, Seungjin, "Contracting for Software Development", Management Science, March, 1992, p. 307.

4.1 Product Definition

Product definition is inherently difficult with software development. Prospective users cannot evaluate a system before they actually use the system, therefore the final product is difficult to define in the early stages. Still, as many resources as possible should be used before a contract is signed to attempt to adequately define the product. The tasks, input/output of the software and the operating environment must be considered.

Along with product definition goes a detailed definition of services that will be performed by the developer and user. Examples of services that must be considered include requirements analysis, design, programming, testing, implementation, documentation and training. The expectations of both parties in these areas must be clearly delineated in the contract negotiation phase. A potential problem in the US is whether software is considered a *good* or a *service*. If it is considered a good, implied and express warranties apply while categorizing it as a service provides no such protection to the user[14]. The contract should clearly state whether the product is a good or a service to avoid confusion later in the project.

Lastly, product definition should specify delivery conditions. Delivery conditions will be closely linked to incentives that are offered. Timetable for the development cycles, installation dates and acceptance tests are key elements of delivery conditions. The contract should discuss each of these points. The contract should specify the measures that

---

[14] Ibid. p. 309.

can be taken in the event of project delays. Measures can include a penalty of a certain dollar amount per day of delay or days of free labor per delay. The contract may also specify after how much delay a user may terminate the agreement. Conversely, a contract can specify incentives for early delivery. Performance criteria may also be included in delivery conditions. Criteria such as average response time as measured under certain working conditions are an example of performance criteria that should be stated in the contract.

4.2 Intellectual Property Protection

Intellectual property protection is another part of the contract that can cause incentive difficulty later in the project life cycle. In the course of a joint software development project, trade secrets of the user may be revealed to the developer who may have contacts with the user's competitors. The user may learn unique programming techniques from the developer that can be used to create a product that is in direct competition with the developer's products. Also, when the time comes to upgrade the product, ownership of a license to modify becomes critical. These contingencies should be addressed in the contract. Areas to address include; title or license to the software, patent protection, ownership of the copyright to the source code and supporting documentation, rights to enhance the software, restrictions to entering the other party's business as a competitor, restrictions to hiring key employees of the other party and protection of trade secrets and confidentiality.

4.3 Payment Structure

Payment structure is the third area to address in contracts to clearly define incentives. Payment schedules are usually tied to the development cycles. A portion of the total fee is paid upon completion of a cycle or module. The overall payment structure of software development contracts is usually either a "fixed price" or "cost plus percentage" contract. There are problems with both. A fixed price contract shifts the majority of the risk of the project to the developer. With the uncertainty of software development being so great chances are high that the developer will lose money with this type of payment structure. The developer has the incentive to finish at cost even at the price of a lower quality final product. Developers charge a premium for this type of contract since they bear so much of the risk. Cost plus percentage contracts, where the developer is paid for the actual development costs plus a percentage as the contract fee are also problematic. The developer has the incentive to overstate cost and the user cannot easily audit these costs. If the cost is based on the number of delivered source code, the developer has an incentive to increase the size of the program.

4.4 Conclusions

Clearly, it is important that a well defined contract that addresses all of these contingencies be written so that incentive questions do not arise after the project starts. This is more difficult then it sounds. Entering into a partnership with a lengthy contract can create mistrust and animosity from the beginning. Personal relationships are extremely important in joint projects, but people working on the project change and problems may arise. As John Petrush, a manager of a joint systems project between Digital and Mcdonnell Douglas Corp. puts it, "The results of our project were very good, but there were times we came close to threatening to take action on commitments in the contract. It helps to have something solid to fall back on."[15] Without a contract that addresses specific issues there is nothing "solid" to fall back on and the user has limited ability to offer incentive to the developer to finish a project. Often the only recourse is to terminate the agreement and start over.

In the particular case we studied, many of these general problems were prevalent. Interviews indicated difficulty in defining the scope of the project in the initial stages as was expected (see Appendix I). The major portions of the project could be defined, but specifics were more difficult to address. Specifics were worked out as the project progressed. This lack of early definition also made it difficult to establish performance criteria that could be linked to incentives. More resources should be used to define what the final product will do under what conditions and tie these criteria to incentives offered.

---

[15] Moad, Jeff, "Contracting With Integrators", Datamation, May 15, 1989, p. 22.

No effort was made to establish penalties for delay. The contract did state that the user could terminate the agreement in the event of long delays. An intermediate measure that falls short of termination that the user could use in the event of delay on the part of the developer would be helpful. Also, an incentive for early completion of work could be used.

In our case, the user did not have as many problems with protection of intellectual property. Source code is normally placed in an escrow account as a protection method. The company in our case also normally requests an exclusive license in its market as a safeguard.

The company in our case normally negotiates fixed price contracts. There is not normally any renegotiation of price after the contract is signed. As discussed earlier, this shifts a great deal of the risk on to the developer. A more appropriate type of contract might be a hybrid of price fixed and cost plus percentage. Initially, a price fixed contract would be signed with a clause that states the price could be renegotiated upon completion of major portions of the project. This would make the developer feel more a part of the team and less like an outside player that was bearing the majority of the risk. At the same time our company would have the opportunity to contain costs by module and have a higher probability of a high quality finished product.

5 Scheduling Software Projects[16]

Coordinating and controlling the project direction and status by integrating the schedules (level scheduling), costs (cash flow forecasts), and relevant information (document tracking) of each project and the overall project program is a critical issue in multicontract software projects.

The ideas that are involved in the use of a master schedule are the need of developing clear and strong project management system to successfully plan, organize, create internal and external communication channels, and control of complex projects. Then, it is important to integrate the scheduling activities and costs of the external contractors with the house company that comprise a project management project into cash flow system. This cash flow system vary in complexity according to the size of the project from cash flow forecast for the individual projects (individual contracts) to cumulative or master cash flow forecast for the project program (entire project). Implementing document tracking systems, through the use of the computer, to keep track of project information (correspondence, analysis, designs, changes specification, etc.), is critical either.

A solid organization structure must be created within the project program to coordinate the different projects, track the project's status and analyze the impact of change in

[16] Adapted from "Major Multicontract Project Scheduling", PMI Seminars/Symposium Proceedings, 1984, pp. 4-7.

specifications on the individual contract schedule and the master schedule. A scheduler or scheduler team must be created in the house company and software engineer could be assigned to each external contractor to help coordination and communication. To manage the impact of the change in specification special forms or uniform standards must be created to be used among different contractors with interface points among each other.

5.1 Implementation

1.- Each party is responsible for scheduling his own work and the integration of the varied schedules, using a standard format.

2.- The different levels of information is organized by a scheduler into a combined schedule to assist the contractor in organizing their work.

3.- Six levels of schedules have been developed to monitor the individual contractor schedules and to maintain a master schedule that would allow the successful completion of the project. This levels of schedules are:

1. Short-term Weekly Schedules
2. Detailed Development Schedule
3. Summarized Schedule

4. Master Schedule

5. Summary Schedule

6. Element Schedule

Each party develops one Short-term Schedule by each week, that includes the work completed the previous week. They present this information in a standard form. The Short-term Schedule is a subpart of the Detailed Development Schedule. The objective to have a discussing tool for potential interfaces.

A second Short-term Weekly Schedule, created by each party and the software engineer, integrates a summarization of each party's detailed schedule. This schedule could be used as a discussing tool for project review meeting.

Each party develops his own Detailed Development Schedule. This schedule is used to monitor the development progress and is the base line for the contract. Each month an updated network diagram is submitted to the scheduler and software engineer for review to determine the current status of meeting contract milestones and interface dates.

Each party also provides a time-scaled summarization of the detailed network diagram in the Summarized Schedule. This report is distributed to interfacing parties to apprise them of other party's' status for meeting milestones.

The Master Schedule is an integration of a time-scale summarization of each party's detailed network diagram and other selected party schedule activities and dependencies.

The Summary Schedule, is a time-scale report based on the Master Schedule, developed by the parties that show the individual contract duration. The Summary Schedule should be distributed to all parties. The objective is to assist coordination among the parties.

The Element Schedule is a report that shows all parties. The Element Schedule could be issue every month and the objective is to present an executive report to the general manager of the house company.

4.- The cost and schedule for each contract are integrated into a cash flow forecast of each party. The forecast is further integrated into a master cash flow forecast for the entire project (final cumulative construction cash flow).

An evident advantage of the centralization of the information in a master schedule where all parties can retrieve information of the total project without loosing the general scope of the entire project. Without the master schedule, the interface points between the parties are potential points of delays and conflicts. However, it is a time consuming task and special personnel must be allocated (scheduler), increasing the cost of the project.

This scheduling model may not be applicable in projects that do not involve parallel projects by independent developers that have interface points between them.

5.2 The use of a single scheduler

This scheduling model must be coordinated by a scheduling team that should have experience in the type of project being developed. Otherwise this scheduling model may render useless causing more problems than benefits in the overall project conclusion.

If a single scheduler is not used to integrate the independent project schedules, the information provided, may be affected due to manipulation by different sources (multiple schedulers) that may have a different understanding and interpretation of the project activities and their contribution to the overall project goals. Therefore, the project will have an inconsistent and/or less exact master schedule and divided direction.

The scheduler has to maintain a close relation with each party, understands the methodology and symbology that every one is using (if it is not the same) and must understand their projects' schedules. Therefore, the scheduler has to have knowledge in the overall project work breakdown structure presented in the different schedules' levels. The scheduler must also have a close relationship with the project manager and keep him informed about the progress of each party and the status of the overall project. Finally, the scheduler should keep in touch with the software engineer, that is the interface with each contractor.

The scheduler should not assume the responsibility for the completion of the individual contracts (this is the responsibility of the party) and must only provide the party the required information.

5.3 Scheduling and Computers

The use of the computer in multicontract project, contributes to accelerate the scheduling process (development, progress control, monitoring, report generation), speed up information integration and improve information quality. The computer also was used to track correspondence, analysis, design, change specifications, request for information, plan clarification, and approvals. In general, the computer could be used to integrate and summarize large amounts of information into reports. The computer is an expediting tool.

The computer could have been used to facilitate the task of the scheduler. A standard software -or at least compatible- could be required to each external contractor by the house company to handle the schedule information. In this way the integration would have been easier.

Today with the improvement in the communication area, a better alternative would be implementing a network system. This network would allow each party to make on-line actualization's and retrieve information using a single package for scheduling. Other facilities provided by networking systems such as e-mail could also be used. In this way, the communication would be facilitated and meeting would only have to be organized if necessary. The changes would be communicated instantaneously without waiting to the next weekly or monthly meeting.

5.4 Scheduling and AI

Managers have difficulty understanding and visualizing software systems. Also, existing project planning tools have many inherent limitations, such as the need for manual input of data, constant training in planning skills and the lack of means to accumulate expertise. An intelligent construction scheduling advisor could be conceived and developed to take advantage of state-of-the-art expert system techniques, popular relational database management system, and an established project management system. In that way the use of expert system and database technologies could overcome such limitations[17] and assists managers in scheduling. The system could operates in a distributed environment where the manager has access to the software developer workstations through a computer network. The system could helps managers anticipate problems that allows early corrective action. As a result, software projects can be developed on time, within budget, and to customer satisfaction.[18]

---

[17] Soh, Chee-Kiong; Phang, Kok-Wai; Wong, Woon-Ping, "Integrated intelligent system for scheduling construction projects", Proceedings of the Institution of Civil Engineers, Civil Engineering, v. 97 n. 4 Nov. 1993, pp. 156-162.

[18] Simmons, Dick B.; Ellis, Newton C.; Escamilla, Terry D., "Manager associate", IEEE Transactions on Knowledge and Data Engineering, v. 5 n. 3 Jun. 1993, pp. 426-430.

5.5 Scheduling and Compression

Schedule compression using the Direct Stiffness Method for structural analysis[19], is a method for critical path (CPM) scheduling that optimizes project duration in order to minimize the project total cost in contractual projects. In addition, the method could be used to produce constrained schedules that accommodate contractual completion dates of projects and their milestones. The method establishes a complete analogy between the structural analysis problem with imposed support settlement and that of project scheduling with imposed target completion date. The project CPM network is replaced by an equivalent structure. The equivalence conditions are established such that when the equivalent structure is compressed by an imposed displacement equal to schedule compression, the sum of all members' forces represents the additional cost required to achieve such compression. In addition, the proposed method has some interesting features: i) it is flexible, providing a trade-off between required accuracy and computational effort, ii) it is capable of providing solutions to CPM networks where dynamic programming may not be directly applicable, and iii) it could be extended to treat other problems including the impact of delays and disruptions on schedule and budget of contractor projects.

[19] Moselhi, Osama, "Schedule compression using the direct stiffness method", Canadian Journal of Civil Engineering, v. 20 n. 1 Feb. 1993, pp. 65-72.

5.6 Scheduling Time and Statistical Distribution

Scheduling of external contractors' projects with an uncertainty content requires that the scheduler's subjective knowledge of various factors that might influence the duration of the activities comprising the project is incorporated. Depending on the participating factors and their significance, a different duration outcome is often observed for each activity. To include this uncertainty in the schedule, a statistical distribution is frequently used. But part of the information available, when duration distribution is being estimated, exists in a subjective form. Abourizk and Simaan[20] present an automated system that requires the scheduler to specify the activity's minimum land maximum times and a set of linguistic descriptors of the external factors that are suspect of influencing the duration. The lower and upper end point estimates are often available from familiarity with the technology used[21], physical and logical constraints, or a combination of these situations. The subjective information collected by the scheduler is modeled as fuzzy parameters and is quantified using fuzzy set theory. The result of the fuzzy set analysis is a sample of activity duration from the underlying distribution, which is then used to characterize the first two moments of that distribution. Since earlier research has shown that the beta distribution provides an adequate representation for project duration[22], the end points specified by the user and the two moments resulting from the fuzzy set analysis are used to

---

[20] Abourizk, Simaan M.; Sawhney, Anil, "Subjective and interactive duration estimation", Canadian Journal of Civil Engineering, v. 20 n. 3 Jun. 1993, pp. 457-470.
[21] Software technologies in general include broad categories such as operating systems, databases, communication protocols, software specification, design and testing techniques, programming languages, man/machine interfaces, software quality, documentation and so on.
[22] Nicholas, J., Managing Business & Engineering Projects, Prentice-Hall, NJ, 1990.

fit a beta distribution. The system then allows the user to visually assess the quality of the fit and modify the shape of the beta density using the visual interactive beta estimation system.

5.7 Software Cost Estimations

The software estimating process requires a basic understanding of the fact that software development is not a 'mechanic' process, like that which occurs in building constructions. There, tasks are concrete, visible, measurable by simple means and of finite quantity. By contrast, software development is a probabilistic process, consisting of a large number of task of undetermined complexity[23].

Success in software engineering depends primarily on managerial considerations, which are in turn contingent on an accurate and reliable cost-estimation's procedure. Software size estimates provide a basis for software cost estimation during software development. Hence, it is important to measure the system size reliably as early as possible, i.e., during the requirements' specification. One of the first challenges lies in finding an estimating method to insure that everything that needs to be included in the estimate of software development costs is identified.

---

[23] Londeix, Bernard, Cost Estimation for Software Development, Addison-Wesley, 1987.

Among the many software cost-estimation procedures available today, COCOMO[24] stand out by its concern for accuracy and the thoroughness of its procedures.

Other best known specification level metrics are Albrecht's Function Points and DeMarco's Function Bang. One problem in using these metrics has been that there are only few tools that can calculate them during the specification phase. Another problem has been that no research data is available how these metrics correlate with one another.

To get a reliable software cost estimate, we need to do much more that just put numbers into formulas and accept the results. A seven-step process for software cost estimation is provide by Boehm[25], which show that a software cost-estimation activity is a mini project and should be planned, reviewed and followed up. The seven steps are:

1. Establish Objectives
2. Plan for Required Data and Resources
3. Pin Down Software Requirements
4. Work Out as Much Detail as Feasible
5. Use Several Independents Techniques and Sources
6. Compare and Iterate Estimates
7. Follow-up

[24] Boehm, Barry W., Software Engineering Economics, Prentice-Hall NJ, 1981.
[25] Ibid.

In a software cost esfimate, the more detail we provide as input to a software cost-estimate, the more accurate our resulting estimate is likely to be. A tentative cost driver attributes list is provided as input to a software cost estimate.

1. Product Attributes

    - Required software reliability
    - Data base size
    - Product complexity
    - Type of application

2. Computer Attributes

    - Execution time constraint
    - Main storage constraint
    - Computer turnaround time
    - Hardware configuration
    - Language level

3. Personnel Attribute

    - Analyst capability
    - Applications experience
    - Programmer capability
    - Programming language experience

- Personnel continuity

- Personnel morale

- Management quality

4. Project Attribute

  - Modern programming practices

  - Use of software tools

  - Required development schedule

5. User Attribute

  - User interface quality

  - Amount of documentation

  - Requirement volatility

Estimating size is already a difficult task in organizations where a software metric's activity exists, but it is made particularly hard when there is no measurement baseline of previous project.

An equation for estimating the number of man-months (MM) required to develop the most common type of software product, in terms of the numbers of thousands of delivered source instructions (KSs) in the software product, according Boehm is:

$$MM = 2.4(KSs)$$

He also presents an equation for estimating the development schedule (DS) in month:

$$DS = 2.5(MM)$$

A metric assistance to size estimating was proposed by Albrecht[26]. He analyzed the statement of requirements of 24 data processing programs in term of the following:

- Inputs
- Outputs
- Inquiries
- Files
- Interfaces

Albrecht then adjusted these numbers according to three levels of complexity -simple, average and complex - to within a range of -25% - +25%. Next, he carried out a statistical analysis on these numbers in relation to the size of the programs and the effort they required for development. During the course of this work, he discovered a high degree of correlation between the size of data processing type of programs his company (IBM) and

[26] Albrecht, A. and Gaffeney, J. "Software Functions, Source Lines of Code, and Development Effort Prediction: A Software Science Validation", IEEE Transaction on Software Engineering, SE-9, 6, pp. 639-647, 1983.

a measure called the Function Points. This measure is defined as the linear combination of the five adjusted terms such that:

Fp = a.inputs + b.outputs + c.inquires + d.files = e.interfaces

where a,b,c,d and e are constant as determined by his statistics.

Albrecht also determined a series of simple equations, using the Function Points, that enable the calculation of the estimated size and cost for some languages used in his organization. As an example, here is the formula used to estimate the size of a COBOL program when its Function Points is known:

S = 118.7Fp - 6490

In practice, once a software requirement is known, a top-level analysis can provide the Function Points count. Then, by using the appropriate formula - the size formula in this case - the size estimate can be obtained.

To convert the estimate of size S into estimate of time, effort and cost, organization must look at past project.

Another metric assistance to size estimating was proposed by Putnam[27]. For each module, the size is given by a set of three values: So (optimistic size), Sm (most likely size) and Sp (pessimistic size). From these values, the expected module size, Smd, can be deduced by the formula:

$$\mathbf{Smd = (So + 4Sm + Sp)/6}$$

To convert the estimate of size into estimate of time, effort and cost, Putnam considers that there is a fundamental relationship in software development between the number of source statements in the system and the effort, development time and the state of the technology being applied to the project. The equation that describes this relationship is:

$$\mathbf{Ss = Ck\ \ K\ \ t}\text{, where}$$

Ss	is the numbers of end product source lines of code delivered
K	is the life cycle effort in man-years
t	is the development time
Ck	is a state of technology constant

[27] Putnam, Lawrence H., "Tutorial 'Software Cost Estimating and Life-Cycle Control: Getting the Software Numbers'", COMPSAC80 The IEEE Computer Society's Fourth International Computer Software & Applications Conference, 1980.

Ck is determined by the use of modern programming practices, the language used, and the development environment among, other factors. While Ck is difficult to determine from its individual components, this value can be calibrated for an organization by looking at past project.

5.8 Cost-Benefits elements to be considered in software estimations

1 Acquirement

- Hardware
- Software
- Testing
- Planning & Analysis
- Initial Training
- Documentation
- Installation

2 Operation

- Operating
- System Management
- Maintenance (hardware & software)
- Programming

- Ongoing Training
- Facilities
- Security Provision

3 Conversion

- Program Transfer
- Data Transfer
- Retrofit

4 System Performance

- Computer system
- Application Software
- System Users

5 Computer Misuse

- Errors
- Omissions
- Fraud
- Denial of service

5.9 Conclusion

Project scheduling is a vital in controlling and monitoring the duration time of a project.

A single scheduler to integrate the independent project schedules, must be used in complex projects. In that way a unique understanding and interpretation of the project may be achieved.

The use of the computer, contributes to accelerate the scheduling process, speed up information integration and improve information quality. The computer can also be used to track correspondence, shop drawings, change specifications, request for information, e-mail, etc.

The use of special forms should be developed to assure uniformity and to measure the impact caused by changes of scheduled activities in the overall project completion.

The complexity of the project scheduling model is linked to the scope of the project. The use of multilevel scheduling helps organize and coordinate the interface points among the parties involved in the project. Simpler projects may require different and less complex scheduling models.

An intelligent construction scheduling could be developed to take advantage of expert system techniques, relational database management system, and project management system. The system could operates in a distributed environment where the manager has access to the software developer workstations through a computer network. The system could helps managers anticipate problems that allows early corrective action.

Schedule compression using the Direct Stiffness Method could be used to produce constrained schedules that accommodate contractual completion dates of projects and their milestones. In addition, it is capable of providing solutions to CPM networks where dynamic programming may not be directly applicable, and it could be extended to treat other problems including the impact of delays and disruptions on schedule and budget of contractor projects.

Scheduling with an uncertainty content requires that the scheduler's subjective knowledge of various factors that might influence the duration of the activities. Part of the information exists in a subjective form. If the subjective information is modeled as fuzzy parameters and is quantified using fuzzy set theory, the result of the fuzzy set analysis is a sample of activity duration from the underlying distribution, which is then used to characterize the first two moments of that distribution. The end points specified by the user and the two moments resulting from the fuzzy set analysis are used to fit a beta distribution.

The software estimating process requires a basic understanding of the fact that software development is not a mechanic process. Software development is a probabilistic process, consisting of a large number of task of undetermined complexity. Among the many software cost-estimation procedures available today, we have COCOMO, Albrecht's Function Points and DeMarco's Function Bang. The problems in using these metrics has been that there are only few tools that can calculate them during the specification phase and that it is made particularly hard to use when there is no measurement baseline of previous project.

measures are valuable, however, so we must find some and use them," states Watts S. Humphrey[28].

Quality measures fall into the following general classes[29]

- Development: Defects found, change activity
- Product : Defects found, software structure, information (documentation) structure, controlled tests
- Acceptance: Problems, effort to install, effort to use
- Usage: Problems, availability, effort to install, effort to use, user opinions

These measures are characterized as follows

- Objective: Can the measure be repeatable produced by different people?
- Timely: Is it available in time o affect the development or maintenance process?
- Available: How hard is it to obtain?
- Representative: To what degree does it represent the customer's view of "goodness?"
- Controllable: To what extent can its value be changed by development or maintenance actions?

[28] Watts S. Humphrey, Managing the Software Process, Addison-Wesley Publishing Company, Massachusetts, 1989, p. 28.
[29] Ibid., p. 338.

In the light o these measures and their characters, each company should use a different set of measurements for each subcontractor, depending to the conditions and availability of the data.

6.2 Level Assessment of the Subcontractor

First step before assessing the level of the subcontractor is to check the relative competence of the parties. The four basic cases are:

1. Both vendor and buyer are technically competent: a truly cooperative partnership is possible.
2. The vendor is technically competent and the buyer is not: trust become more important between parties.
3. The buyer is technically competent and the vendor is not: buyer should look for another vendor, if it is possible.
4. Neither is technically competent: finding professional help is the best choice for buyer.

To promote the delivery of software that is on time, within budget, contractually compliant, technically adequate, and of high quality, companies have to pursue activities with the following objectives in order to assess the level of maturity of the subcontractor:

a) Start a wide initiative that emphasizes a total quality management teamwork approach; by bringing together personnel from various disciplines within company, buying commands, and the contractor to focus on improving deliverable products.

b) Assessment of technical adequacy

- Evaluate contractor's process/plans/procedures to include system decomposition; software engineering, development, configuration management, testing; system integration, risk management, and corrective action.
- Flow, proof, and audit the contractor's processes to be used in software product development
- Evaluate selected products.

c) Assessment of software quality

- Review the contractor's plans and procedures for adherence to contract quality and format requirements.
- Verify the contractor's processes to find if the actual processes are implemented according to approved plans and procedures.
- Audit the contractor's products for compliance with the contract's requirements and adherence to the quality plan.

6.3 Management Styles According to Levels

Level 1

Subcontractor has no established planning or management methods, it is best to find another development group. If this cannot be done, it is essential to impose discipline through contract or management edict. The critical steps are:

- Install management review system
- Insist on comprehensive development plan
- Set up a software configuration management (SCM) function
- Ensure that software quality assurance (SQA) organization is established and sufficiently staffed to review a reasonable sample of the work products.

Until Level 1 organizations demonstrate their ability to perform the necessary quality procedures, the traditional management system should be retained. Management essentially pushes them toward Level 2

Level 2

At the Level 2 maturity stage, the management process motivates the organization to improve its process maturity. Following steps are helpful to increase the expected output quality:

- Review SQA staffing to see that it is adequate to handle the anticipated review workload

- Ensure SCM function is used, not only to monitor contract but also for the ongoing development work
- Establish some key process and product standards, such as design, coding test and inspection

Level 3

At Level 3 and beyond, the organization is effectively managing its own process, in order to meet quality standards. The prime area for focus should be:

Establish a comprehensive program of process measurement and analysis. This data is retained in the process database and used by SQA, and project managers to monitor product and task quality.

Ensure that suitably aggressive quantitative quality plans are established and tracked.

Initiate a comprehensive defect prevention program.

Establish a strategy and plan for process environmental support.

6.4 Proposed SQA Program

Definition

The purpose of SQA is to provide management with appropriate visibility into the process being used by the software project and of the products being built.

6.4.1 Goals

- To improve software quality by appropriately monitoring both software and the development process that produces it.
- To ensure full compliance with the established standards and procedures for the software and the software process.
- To ensure that any inadequacies in the product, the process or the standards are brought to management's attention so these inadequacies can be fixed.

6.4.2 Role

- It is a mistake to assume that the SQA people themselves can do anything about quality.
- The existence of an SQA function does not ensure that the standards and procedures are followed.
- Unless management periodically demonstrates its support for SQA by following their recommendations, SQA will be ineffective.
- Unless line management requires that SQA try to resolve their issues with project management before escalation, SQA and development will not work together effectively.

6.4.3 Responsibilities

- Review all development and quality plans for completeness.
- Participate as inspection moderators in design and code inspections.
- Review all test plans for adherence to standards.
- Review a significant sample of all test results to determine adherence to plans.

6.4.4 Relation of SQA with subcontractor

- Review subcontractor QA system and policy
- Write SQA requirements for subcontractors
- Review subcontractor specs for requirements traceability
- Approve subcontractor for SQA plan
- Audit subcontractor SQA program with respect to design standards
- Monitor subcontractor SQA plan implementation
- Audit subcontractor compliance with design standards
- Audit subcontractor compliance with coding standards
- Monitor subcontractor test
- Witness final tests and acceptance
- Review/approve subcontractor changes
- Responsibilities and resources are established to track subcontractor performance against plan.
- Subcontractor standards and procedures for estimating, planning, and change management are reviewed.
- Subcontractor SQA resources, procedures, and standards are periodically reviewed to ensure they are adequate to monitor subcontractor performance.
- Standard subcontractor quality metrics are established and tracked.
- Subcontractor SQA performance is tracked and reviewed.

The SQA group works with the software project during its early stages to establish plans, standards, and procedures that will add value to the software project and satisfy the constraints of the project and the organization's policies. By participating in establishing the plans, standards, and procedures, the software quality assurance group helps ensure they fit the project's needs and verifies that they will be usable for performing reviews and audits throughout the software life cycle. The SQA group reviews project activities and audits software work products throughout the life cycle and provides management with visibility as to whether the software project is adhering to its established plans, standards, and procedures.

6.5 Procedures of SQA Implementation for Subcontractors

The prime contractor's software quality assurance group monitors the subcontractor's software quality assurance activities according to a documented procedure.
This procedure typically specifies that:

1. The subcontractor's plans, resources, procedures, and standards for software quality assurance are periodically reviewed to ensure they are adequate to monitor the subcontractor's performance.
2. Regular reviews of the subcontractor are conducted to ensure the approved procedures and standards are being followed.
    - The prime contractor's software quality assurance group spot checks the subcontractor's software engineering activities and products

- The prime contractor's software quality assurance group audits the subcontractor's software quality assurance records, as appropriate

3. The subcontractor's records of its software quality assurance activities are periodically audited to assess how well the software quality assurance plans, standards, and procedures are being followed.

6.6 Conclusion

Quality assurance is a necessary activity for software products and product quality is one of the best measures of a software development project, specially when the companies deal with the subcontractors. But before implementing any quality procedures, necessary quality measures have to be defined depending to the subcontractors' skill level.

Quality Measures establish a clear record of progress, represent a powerful vehicle for setting objectives and provide a graphic framework for current situation.

The purpose of the quality plan is to motivate action, not to evaluate people. The idea is to create an environment where vendor and buyer are working together to accomplish the objectives.

international assistance organizations found the following to be some common causes of difficulty upon project completion.

"1. Failure to expeditiously terminate "unsuccessful " projects due to inadequate monitoring and control by central government authorities and international assistance agencies, political defensiveness and unwillingness to admit failure, or resistance on the part of constituency groups.

2. Restriction of benefits and outputs to a smaller group of recipients than intended by project design; demonstration and spread effects of the project are limited except were special efforts are made to amplify them.

3. Failure to prepare and submit project completion reports, complicating project evaluation and delaying formal close-out of loans and grants.

4. Inadequate or inappropriate utilization of outputs of completed projects by beneficiaries; failure of the government and international assistance agencies to plan for user training.

5. Difficulties of the donor agencies in terminating their contribution to a successful project prior to completion without jeopardizing continuation of the project."[31]

---

[31] Louis J. Goodman, Ralph N. Love, Project Planning and Management: An Integrated Approach, Pergamon Policy Studies, 1980, p. 205.

These authors state that to avoid this problems the project manager should define a carefully and thoroughly analyzed completion procedure .

According to Goodman and Love the completion procedure(termination) for all projects must include a completion report that includes a phase-in strategy for the permanent or temporary organization that will be responsible for continuing operational activities. The completion procedure should contain the following steps:[32]

1. A project completion schedule should be prepared.

2. A special reporting and management information should be established.

3. All the contracts must be finalized, loan facilities terminated, and institutions such as local banks, insurance organizations, and other authorities notified of the proposed changes in administration.

The project completion schedule must include financial aspects of the project, the risk assumed by the subcontractor and by the organization, the available resources, and a notification of any change of the control status made to those involved with the project.

---

[32] Ibid.

The special reporting and management system must make full information of the project available to the contractor, funding agencies and organization. Finally, step 3 implies that unexpired contracts will have to be negotiated and some activities like maintenance and insurance may need to be carried out.

7.2 Project Summary

The organization should expect the subcontractor to provide a detailed summary evaluation after project termination. This summary is defined as the Post-completion summary by John M. Nicholas[33].

The post-completion summary includes the following steps that have been modified to apply to the subcontractor:

- Review of the initial objectives in terms of performance, schedule, and cost.

- Review of the evolution of the objectives and an analysis of the project team and teams performance regarding these objectives.

---

[33] Nicholas, J. M., Managing Business & Engineering Projects, Prentice-Hall, 1990, p. 456.

- Review of project teams activities and relationships through the project life cycle. his includes a review of interfaces, performance, project management effectiveness, cause and process of termination, customer reaction and satisfaction.

- Identification of good performance, profitability, organizational benefits provided, marketable innovations, etc.

- Identification of mistakes, problems, oversights, areas of poor performance and its causes.

7.3 Software Acceptance

Software acceptance is a process of approving or rejecting software system during development, maintenance or purchase, according to how well the software satisfies pre-defined criteria. The final acceptance decision occurs with verification that the delivered documentation is adequate and consistent with the executable system and that the complete software system meets all buyer or user requirements. This decision is usually based on software acceptance testing [16]. It consists of tests to determine whether the developed or package system meets predetermined functionally, performance, quality, and interface criteria.

Software acceptance is specified in a formal plan. The software acceptance plan identifies products for acceptance, the specific acceptance criteria[34], acceptance reviews, and acceptance testing.

Examples of information that should be included in a software acceptance plan are:

- Project Descriptions: Type of system, major task system must satisfy; external interfaces; expected normal usage; standards.

- Management Responsibilities: Responsibilities for acceptance activities; resources and schedule requirements; standards.

- Administrative Procedures: Anomaly reports; record keeping; communications.

- Acceptance Testing: Test plan and acceptance criteria; test cases and procedures; test results and analyses.

[34] According to the Webster's New World Dictionary, a criteria is a standard, rule, or test by which something can be judged.

To avoid problems during the project termination phase a completion procedure should be developed containing a clear project completion schedule, and an established special reporting management information system.

The user should expect the subcontractor to provide a detailed summary evaluation after the project evaluation after the project termination including a review of performance, schedule and cost. This summary report should also include a review of objectives met, team performance, user benefits provided through the system and identification of potential problems.

The software acceptance during project termination should be done by verifying that the delivered documentation is complete, adequate and consistent with the buyers needs by meeting the performance, quality and interface criteria defined by the customer.

The manager should assure that the system meets users needs and that contractual agreements with the subcontractor are clearly defined and met.

8 Software Risk Management

Software risk management is an emerging discipline whose objectives are to identify, address and eliminate software risk items before they become either threats to successful software operation or major sources of software rework.

The primary goal of (any) risk management is to identify and confront risk factors with enough lead time to avoid a crisis.

The seven steps of risk-management process:

1. Identify risk factors.

A risk is a potential problem; a problem is a risk that has materialized.

2. Assess risk probabilities and effects on the project.

Because risk implies a potential loss, you must estimate two elements of a risk: the probability that the risk will become a problem and the effect the problem would have on the project's desired outcome.

3. Develop strategies to mitigate identified risks.

Two essential parts of risk management are setting thresholds, beyond which some corrective action is required, and determining ahead of time what the corrective action will be.

Without such planning "How does a project get to be a year late?" "One day at a time."

Action planning addresses risks that can be mitigated by immediate response.

Contingency planning addresses risks that require monitoring for some future response should the need arise. Contingency plan should have a duration to avoid contingent actions of interminable duration.

4. Monitor risk factors

5. Invoke a contingency plan

6. Manage the crisis

When the contingency plan failed, you enter crisis management (usually means drastic corrective actions.)

1. Announce and generally publicize the problem.

2. Assign responsibilities and authorities.

3. Update status frequently.

4. Relax resource constraints.

5. Have project personnel operate in burnout mode.

6. Establish a drop-dead date.

7. Clear out unessential personnel.

7. Recover from a crisis

1. Conduct a crisis postmortem.

2. Calculate cost to complete the project.

3. Update plans, schedule and work assignments.

4. Compensate workers for extraordinary efforts.

5. Formally recognize outstanding performers and their families.

Risk management procedures for project managers:

1. Identify the project's top ten risk items.

2. Present a plan for resolving each risk item.

3. Update list of top risk items, plan and result monthly.

4. Highlight risk-item status in monthly project reviews and compare with previous month's ranking and status.

5. Initiate appropriate corrective actions.

One critical aspect of the customer-contractor relationship is effectively handling incentive difficulties. Incentive difficulties arise from the uncertainty involved in software development. The uncertainty of the software project reflects adds to the difficulty to meet objectives traced by both parties involved. In essence the incentive difficulties can be traced down to the contract that the user and the developer sign and may be classified into three main areas which include product definition, intellectual property protection and payment structure. These categories must be considered in detail in the contract agreements set with the contractor to assure the success of the project.

Subcontracting software projects should clearly address scheduling problems. The complexity of project scheduling is linked to the scope of the project. In general, "soft" projects of ambitious scope should include a multilevel scheduling model, computer networks to take advantage of system techniques, relational database management system techniques, and project management systems. The scheduling model should be directed ideally by a single scheduler that must integrate the independent project schedules. The project manager must understand that the software development process is a probabilistic process full of a large number of tasks of undetermined complexity and be informed of software cost-estimation procedures available like COCOMO, Albrecht's Function points and DeMarco's Function Bang as well as schedule compression Direct Stiffness Method to be used in the accommodation of project contractual completion dates and milestones.

Another important aspect of project software development by a contractor involves a clear definition of the software product quality level. To assure success of the project quality measures must be defined to establish a clear record of progress within the frame of a quality plan. The final selection of the subcontractor must be linked to his relative skills level which affects the quality of the product finally implemented.

The project termination of any software development is defined in this project to occur when the subcontractor operates the system and evaluates its performance in terms of the needs it should meet. The project termination occurs during the operation phase of the systems development cycle. To assure successful project termination the user should develop a clear project completion schedule and a clear information reporting system to monitor subcontractor. The organization should also request the subcontractor to submit a detailed summary evaluation after the project termination including a review of objectives met, team performance, user benefits provided by final software product and potential problems encountered and solved. The final acceptance decision and termination of the project should occur when the organization verifies that the delivered documentation is completed, consistent with the executable system and the completed software system meets all the specified requirements.

Finally, to deal with the risk associated with any software project development the organization should consider the application of software risk management theory. Software management is an emerging discipline whose objectives are to identify, address and eliminate software risk items before they become either threats to successful software operation or major sources of software rework.

Appendix I

Interview questions and answers I

This is the first interview results. The person interviewed was a functional manager, he also was a software contractor for many years before he joined the Company.

Scheduling Problems

1) What type of scheduling do you use when you work with contractors:

Every one has its Independent schedules or a Master schedule?

Describe?

For time-and-material (TM) type of contract, the usual scheduling tools are used e.g., CA Super Project, Microsoft Project etc.

For result-oriented (RO) type of contract, schedule is usually made up of deliverable milestones.

2) Who is responsible for maintain the schedule updated? Is there any person especially assigned to this purpose?

For TM, the manager who is directly supervising the contractors. A lot of time, RO contracts are not watched as closely.