Requirements Management :
from technical to managerial aspects

by: Ghislain Levesque
Michel-Ange Zamor
Department of Computer Science
University of Quebec at Montreal
P.O. box 8888, Station Centre-Ville
Montréal, Canada, H3C 3P8
Tel.: 514-987-3000 (3864)
Fax: 514-987-8477
Email: Levesque.Ghislain@UQAM.CA

and: Barry Shostak
CGI Group, Inc.
1800 McGill College, 15th floor
Montréal, Canada, H3A 3J6
Tel: 514-281-6729
Fax: 514-281-1973
Email: Barry.Shostak@CGI.CA

ABSTRACT

Following a CMM appraisal which identified the Requirements Management Process (RMP) as a target for improvement, CGI published a new requirements management policy. An action research study was then undertaken to find out the root-causes of some requirements-related problems and to design improvements to the process. Although development projects had been managed according to an internal development methodology for many years, the research concluded that the initial requirements for projects were too often incomplete, not validated by their end-users and that they were not really intelligible to the users. The Requirements Engineering Process and particularly the elicitation process was at the source of the problems. To improve the elicitation process, requirements checklists and a user scenario template were designed and tested in a pilot project. These were then included in the formal methodology. The weak parts of the methodology having been resolved, the real challenge of enforcing the Requirements Management policy now becomes a responsibility of project managers with the support of the high-level management for its institutionalisation.

Key words : requirements management, requirements engineering, requirements elicitation, software process improvement, outsourcing.

INTRODUCTION

CGI Group Inc[1] is an IT outsourcing and management company, headquartered in Montreal, Quebec, Canada. Their activities include the operation, development and maintenance of Management Information Systems on behalf of clients. CGI's Telecom Division has an unusual relationship with a large telecommunication company wherein the client retains responsibility for developing business and system requirements (with CGI acting as consultant on request), and CGI is responsible for the entire development process on a commercial preferential basis with the support of the client's information system and technology group.

CGI's Telecom Division conducted a CMM appraisal of their development and maintenance processes in late winter,1999, using the Software Engineering Institute's CBA-IPI[*] methodology. Various activities were then undertaken to address appraisal findings, among them the announcement of a policy to establish the management of project requirements and changes to them. An action-research project (CMM-RM) was initiated as well in response to specific requirements management-related findings from the CMM appraisal. This project is the subject of this paper.

This paper starts by describing the research method. Then the main findings which emerged are presented along with some of the tools that were designed to put the requirements management process under control after their test in a pilot project. A discussion of the results and their possible impacts follows.

RESEARCH METHOD

The purpose of the project was to look at the weaknesses identified in the RMP in order to get a better understanding of the root-causes and to propose process improvements via methods and tools to be included in CGI's formal methodology. The analytical framework of the research study took into account the current internal methodology and the IEEE Software Engineering Standards (1999) as an external reference for comparison and particularly the Concept of Operation document (IEEE Std 1362-1998),

---

[*] CBA-IPI stands for CMM Based Appraisal for Internal Process Improvement.

the Guide for Developing System Requirements Specifications (IEEE Std 1223-1998) and the Recommended Practice for Software Requirements Specification (IEEE Std 830-1998).

The project began by surveying a cross-section of projects to determine how they managed requirements. A data protocol was designed to identify the major aspects of the survey and set the limits of the data gathering process. Five application projects and one technology infrastructure project were surveyed**.** They were not the same as the projects chosen for the CMM appraisal; however, they nevertheless represented a cross-section of the technology and type of projects undertaken over the last two years: small, medium and large enhancements, conversion and infrastructure projects.

This data gathering process was initiated by interviews with the selected project managers in order to understand the context and the major steps by which their projects had evolved. The researchers then examined each projects' documentation in order to understand the contents of the requirements documents provided by the customer or prepared by the development team according to the methodology. A report of observations done was then prepared for each project and submitted to the project manager for his or her validation.

The global results of this investigation were then submitted to an internal peer review by development people not involved in the projects and by members of the process engineering staff. The research team was then invited to submit alternatives for software process improvements. Among the alternatives suggested, the general preference was for checklists as tools to introduce checkpoints into the requirements engineering process. The set of proposed tools was then tested in a pilot project and adjusted afterward before their institutionalisation.

RESEARCH FINDINGS

The initial investigations indicated that the main obstacle to requirements control was the initial system specification produced at the feasibility phase of development. The quality of the initial requirements in this specification was insufficient to allow for the management of small changes to requirements later on during the projects. This observation was based on the five application projects only; the infrastructure project was not considered due to inconsistencies observed in its requirements documentation. In the following sections, we present the main findings of the study, the root-causes of the problem and their consequences:

The main findings were:

- Most projects are time-boxed and the requirements provided by the customer are often incomplete, unclear and leave questions unanswered;
- The required functionality is presented on a detailed event basis. Functional requirements pertaining to each business transaction or from the overall user's point of view are missing;
- The underlying business processes are poorly described and, consequently, are frequently not well understood by development teams;
- The users do not validate the requirements produced, leaving major defects in the requirements that could have been found right at the beginning;
- The risks inherent to poor requirements are not fully assessed and addressed.

The root-causes for these problems were:

- The internal methodology does not prescribe a specific requirements elicitation process[*] nor does it promote user interface prototyping to clarify unclear requirements;
- The requirements documents prepared by the customer are full of technical details (solution instead of problem-oriented description) but fail to adequately describe the changes done to business processes and the users' point of view;
- No quality criteria are provided to review and assess the quality of documents produced by either the customer or the supplier;

The consequences of these problems were:

- At the design phase, the requirements are often reworked instead of refined;
- System and validation testing generates a high percentage of enhancement requests;
- Target budgets and schedules for complete customer satisfaction are not always met.

Those results were consistent with the findings of the CMM appraisal and were, as already mentioned, agreed upon by the peer review committee as being accurate. The type of solutions preferred by CGI had to be simple, applicable in the short term, and transferable to development teams with a minimum of training, hence, the preference for checklists.

SOLUTIONS TO BRING REQUIREMENTS UNDER CONTROL

The IEEE standards proved to be very helpful here since they prescribe the form and the content of good requirements documents. A requirements checklist was derived from the IEEE standards mentioned above. The main topics covered in the checklist related to the description of the current system, the justification for changes and the description of the proposed system. The checklist consisted of 52 "yes"/"no" questions. Another checklist was then prepared to help determine whether it was appropriate to use prototyping to elicit the requirements. Finally, to comply with the requirement of IEEE standard 1362-1998 to describe each operation scenario, a template was designed to describe business transactions from a user point of view [Levesque, 1998], following the recommendations of Jacobson (1999) for use cases. This has been identified as the "User scenario".

These tools were then presented in a training session to a development team that was initiating a project and preparing their system specification. This was our pilot team. The team members attended a four-hour training session where the tools were presented,

---

* A process by which requirements are clarified and refined.

explained and discussed before their use. The requirements checklist was presented to be used in the following ways:

1. As a reminder of the information that has to be delivered by the customer;
2. As a tool to review the requirements with the customer;
3. As a tool to progress in the collection of information and appropriate requirements;
4. As a tool to ensure requirements are complete.

The participants were also asked, as an exercise, to write the user scenario for the transaction: "A customer withdraws money from her bank account at an automated teller machine", a task familiar to them. Using the characteristics of a well-written requirements specification that was introduced to them, they were also asked to inspect a solution we provided to them for the same transaction. This inspection had to be done according to the criteria that requirements were complete, unambiguous, consistent and verifiable.

The development team had a month to work with the tools during the pilot. The research team remained available for any consultation when necessary. During that period, the project manager requested clarification on questions related to the requirements checklist on several occasions. At the end of the month, the tools were evaluated with the system analyst and the project manager. The following observations and conclusions were drawn :

1. The requirements checklist (52 questions) is too long. It would be easier to use if it was segmented into three checklists : one to assess the requirements of the current system*, another one for the justifications for changes, and finally a third one for the description of the new system. In this last one, it was also recommended to add a few questions at the end to make sure that the traceability of the system specification to the business requirements is also assessed.

* A sample of this checklist, after segmentation, is provided in the appendix.

Each of three checklists was completed based on the users' feedback and then linked to a corresponding step in the methodology. These steps were "Review current situation", "Analyse System requirements", and "System specification".

2. The most surprising observation related to the "User scenario". Since the functional requirements had been typically presented on a detailed event basis, the methodology did not support visualising a global view of the requirements for a business transaction. We questioned that in the training session; the system analyst responded : "We do it in our head". The rest of the team agreed. It quickly became apparent that the tool was as an essential part of the process, perhaps more important than the requirements checklists themselves. It was then decided that if the customer requirements do not provide the Users' scenarios, they should be written by the system analyst during the requirements analysis and then the client representatives should validate them. In the pilot project, the client representative also acknowledged that his job to prepare the validation tests would be facilitated by starting from the user scenarios provided.

3. The prototyping checklist used in the pilot test seemed to be more oriented toward user interface prototyping than architectural prototyping. This checklist was also revised to ask the appropriate questions about both types of prototyping, improving its consistency in support of establishing the need and justifications for prototyping. It had also been linked to the "System Specification" phase of the methodology.

From the quality assurance perspective, a list of ten quality criteria for a well-written system requirements specifications document was introduced according to the recommendations of IEEE 830-1998 and IEEE 1233-1998. The use of this list was recommended for internal walkthroughs during the analysis phase and a subset of it limited to *completeness, consistency, traceability and verifiability* was recommended as a guide for a formal inspection at the end of this process.

DISCUSSION OF THE RESULTS

The CMM appraisal found that the Requirements Management Process was a prime candidate for improvement. The action research discovered that the requirements engineering process was the root-cause of the requirements-related problem and that tools were needed to assure a better support to the elicitation process.

As a result, three requirements checklists and a prototyping checklist were prepared and tested. A user scenario template was introduced to support the development of a high-level view of the user transactions. Finally, a list of quality criteria was proposed to assess the quality of the system requirements specification and be used for inspection. These instruments address the technical point of view of this research. The literature and our experience both tell us, however, that the technical part is only half of the story when Requirements Management is concerned.

As mentioned in the introduction, the CGI Telecom division do not usually develop business and system requirements nor have direct access to the end-users in this specific case. Therefore, project requirements management activities are conducted with customer representatives acting on behalf of business requirements developers and end-users. That is precisely where the whole question falls: on the ground of the customer-supplier relationship. On the one hand, for example, CGI project managers can use the checklists and user scenarios to ask for more information and dig into the requirements until a satisfactory level of clarity has been achieved. On the other hand, the customer representatives are unaccustomed to the flow of questions and justifications arising from the use of these new instruments and may have difficulties to find the appropriate answers.

It becomes clear that the expected changes will not happen overnight simply by introducing a few new tools into a methodology. For the requirements management problems described, a culture change has to take place. Our expectation is that this will only take place over time, with training, monitoring and management support playing key roles in the implementation process. Information sharing on the level of requirements change requests agreed-to and their costs will provide key information to assess the success or failure of the intervention.

CONCLUSION

This case demonstrates again that when there are Requirements Management problems, it is important to look at both sides of the problems, the technical and the managerial aspects. Doing root-cause analysis on the technical aspects is not difficult when there is a will to change the situation. The root-causes can be brought to light and process improvements identified. But the institutionalisation of the technical changes, when they mean changing the organisation culture of two companies, cannot be taken for granted. Considerable effort will be needed in order to get the business to business relationship adapted to the new context and these changes can not be left to the project managers alone. They should get the support of the entire management team and the evolution of the process should be monitored regularly.

REFERENCES

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specification, 1999 Edition, Volume 4, Resource and Technique Standards.

IEEE Std 1362-1998, IEEE Guide for Information technology – System Definition - Concept of Operations (ConOps) Document, 1999 Edition, Volume 1, Customer and Terminology Standards.

IEEE Std 1233-1998, IEEE Guide for Developing System Requirements Specifications, 1999 Edition, Volume 1, Customer and Terminology Standards.

Jacobson Y, G. Booch and J. Rumbaugh, 1999, The Unified Software Development Process, Reading Mass., Addison-Wesley, 463 p.

Lévesque G., Analyse de système orientée-objet et génie logiciel, concepts, méthodes et application, Editions Chenelière/McGraw-Hill, Montréal, février 1998, 458 p.

APPENDIX: SAMPLE CHECKLIST

Review current situation checklist

Business Unit	
Document Reference	

	Verification **YES : we already have this information** **NO : we do not have this information for the moment**	***Y***	***N***	***Note***
	DESCRIPTION OF THE CURRENT SYSTEM (if any)			
1	Mission, objectives, and scope of the current system are generally described.			
2	The current operational policies (pre-determined management decisions) are identified.			
3	The current operational constraints (limitations placed on the operations of the current system) are identified.			
4	The current operational environment and its characteristics are described.			
5	The major system components and the interconnection among components are provided.			
6	Interfaces to external systems are identified.			
7	Capabilities, functions and features of the actual system are identified and generally described.			
8	Charts and accompanying descriptions depicting inputs, outputs, data flows, manual and automated processes sufficient to understand the current system or situation from the user's point of view are provided.			
9	The operational risk factors of the actual system are described.			
10	The performance characteristics such as speed, throughput, volume, frequency are known.			
11	The provisions for safety, security, privacy, integrity, and continuity of operations in emergencies are known.			
12	Modes of operations (ex. Internet, screenphone, IVR,…) that apply to all classes of users are provided.			
13	User classes profiles (common responsibilities, skill levels, work activities, modes of interaction with the system, particular operational scenarios, …) are provided.			
14	Other involved personnel relevant to the operation of the existing system are identified with their roles.			
15	The support environment of the actual system is described.			

ENDNOTES

1. CGI is the largest independent information technology consulting firm in Canada and the fifth largest in North America, based on its revenue run rate of $1.6 billion. Its order backlog totals approximately 7.2 billion by the end of year 1999, and the company currently has proposals outstanding for an additional $4 billion in potential large contracts. CGI has 10,000 professionals and provides end-to-end IT services and business solutions to 2,500 clients in Canada, the United States and more than 23 countries around the world.