Managing Software Requirements Risks with Software Development Impact Statements

Don Gotterbarn
School of Computer & Information Sciences,
Auckland University of Technology
Private Bag 92006 Auckland, New Zealand,
Donald.gotterbarn@aut.ac.nz
(Visiting from East Tennessee State University
gotterba@etsu.edu)

Tony Clear

Choon-Tuck Kwan
School of Computer & Information Sciences,
Auckland University of Technology

Defining software requirements is a difficult and complex process. It is best characterised as a joint learning process in which shared understandings evolve through dynamic interactions between clients and developers. The inherent difficulties in this complex process are exacerbated in outsourced software development projects. In this paper we review the use of Software Development Impact Statements (SoDIS) to improve the quality of software in two outsourced software development projects. Our progressive refinement of the SoDIS process is discussed and we introduce the notion of a SoDIS inspection. The paper demonstrates the value contributed by the SoDIS process through highlighting critical issues, and the resulting modifications to specified requirements, project definitions and contractual documents. We conclude by recommending the wider application of the SoDIS process to all software development projects, given its demonstrated contribution in the extreme context of outsourced development projects.

Keywords

Software Development Impact Statements, Outsourcing, Global Software Outsourcing, Requirements Engineering, Project Management, Risk Management, Software Engineering Ethics Research, Software Inspections

1. INTRODUCTION

“Defining requirements is a complex and difficult process, and defects in the process often lead to costly project failures. There is no complete and well defined set of requirements ready to be discovered in system development (Boehm *et al.*, 2001).

Software development begins with an attempt to understand the nature of the desired product. Difficulties in getting good requirements have long been recognized and have led to a variety of attempted solutions: rapid prototyping, formal methods, special requirements notation, etc. Boehm and colleagues (2001) describe a requirements negotiation process “that builds mutual understanding and a shared vision in successive steps”. But this is not always possible when development is out-sourced. Difficulties in clearly stating, and comprehending requirements before formalising them for third party developers to design and implement, are at the heart of this risk in outsourced software development projects.

But in addition to these requirements risks, software development has been argued to suffer from “a tension between four opposing forces:

- A force for change built upon an initial and evolving vision, which drives the software process
- A commercial force for certainty of cost and outcomes
- A project management force for certainty of delivery against targets
- A professional force for delivering quality software (Clear, 2003)”

Outsourcing of software development arises largely from the second of these points. Commercial imperatives drive a quest for certainty of cost and outcomes, yet are inconsistent with Boehm’s notion of an evolving and jointly negotiated vision of software, and thus may lead to the very opposite in practice. The inherent difficulties in achieving a jointly negotiated vision are exacerbated by outsourcing. Therefore, the need for better techniques and processes to manage the requirements process, to improve software quality and reduce project risks, is even more critical in outsourced software development projects. Using software development impact statements (SoDIS) can improve the quality of software by ensuring that the needs of all stakeholders involved in a project have been properly taken into account. The resulting initial set of requirements thus reflects a more comprehensive vision. By applying the SoDIS process a project pre-audit can be undertaken, the results of which feed through into a refined set of requirements expressed in the Request for Proposal (RFP) documents common in an outsourced software development model. Development contracts may also be modified to stipulate further SoDIS project audits or inspections at defined points in the life of the project. Thus promising mechanisms to address the inherent weaknesses in an outsourced model of software development are now available. Application of the SoDIS process in projects to date has been shown to improve the quality of the project scoping, requirements analysis, project management and risk assessment processes.

We show how the SoDIS process was developed into an inspection model based on work with the UK government and then demonstrate how the SoDIS inspection process has been successfully applied to an outsourced software development project through a specific case described below.

The original SoDIS concept as developed by Gotterbarn and Rogerson (Rogerson 1998) was based on research that had been done on multiple software development projects. The first set of data indicated that the two primary causes of software project failure were poor risk analysis and that the major contributor to the poor risk analysis was too narrow an identification of stakeholders. This narrow consideration of stakeholders contributed to a limited view of project scope. The second set of data they used was the information from the large number of software development projects, which had had significant negative impacts on society and its citizens. From these two sets of data they formulated a hypothesis about a way to mitigate social and ethical software disasters. The hypothesis proposed that a preliminary analysis of software development plans would alert the developer to a broader range of stakeholders and expand the range of risks considered for these stakeholders. In turn this would have a positive impact on the development of the software and thereby reduce the negative impact of the software developed. They developed methods to test these propositions and developed the Software Development Impact Statement (SoDIS) process to do the preliminary project auditing. They tested this process on real software projects in industry and academe. The application of the SoDIS process in a blind parallel test with a US company in 1998 led to significant modifications to the analysis process and to the development of a prototype tool (called the SoDIS Project Auditor, SPA) to apply the SoDIS process to software project plans.

They applied the results of other data, namely the evidence that it costs less and is more effective to identify and fix a problem early in the development cycle. The SoDIS analysis process was again modified and applied to projects in Australia, New Zealand, UK and the USA. Later, in 2002 Rogerson contracted with the UK government to continue this action research on the UK government's plan to implement electronic voting in the UK by 2005. This application of the SoDIS process led to various modifications to the SoDIS process and the discovery of some difficulties with the process. Simon Rogerson, a co-developer of the SoDIS process, and Ben Fairweather, both from the Centre for Computing and Social Responsibility, were contracted by the UK government to analyze the UK government's requirements to establish electronic voting by 2005. Changes were made to the SoDIS and tested during the UK analysis process. The application of the process and its results in this action research project with the UK government are described below.

2. THE SODIS AND UK ELECTRONIC VOTING REQUIREMENTS

The study commenced with the identification of the technical and social issues related to the project. Meetings were held between the SoDIS Team and the Development team responsible for outsourcing the project. These meetings served as a way for the SoDIS Team to gain a high-level understanding of the nature of the application and the key social and technical issues.

At this level, all the available technical options were formulated, after evaluating such technical requirements as Location of Polling, the Means of Authentication, the User Interface, the Network Communication Interface and the Collection and

Table 1: Stakeholders in UK Electronic Voting Project

Stakeholders Role	Name
Customer	Central Government
	Local Government
	Those seeking election
Community	Minority groups:
	those overseas,
	those with disabilities,
	those with linguistic constraints,
	those from minority ethnic groups,
	those belonging to fringe political parties,
	those with restricted movement for example on remand or in hospital long term,
	those living in rural areas
User	Citizens as voters
Vendor	Suppliers of technological elements
Developer	Systems developer

Processing Infrastructure. (Fairweather, B. & Rogerson, S. 2002)

At this initial stage, the potential stakeholders of the system were identified as well as the generic requirements of the system. For the UK Election Project, 10 generic requirements and 13 stakeholders falling under five different stakeholder roles were identified. The list of stakeholders was as follows:-

As the focus was to satisfy the needs and obligations of the public, the stakeholders within the Community and User roles were singled out for detailed analysis using the (SPA). The SoDIS Team evaluated each requirement against each stakeholder by asking a series of 32 ethically-related questions to uncover any potential negative impact the requirement might have on the stakeholder. Where such problems are identified, the team examined the possible modifications to the project or the technical option that will minimize the adverse impact on the stakeholders.

Even with the shortlisted stakeholder groups, the number of iterations needed to cover all the requirements, the ethical questions and the stakeholders was significant. The rigorous exercise of going through the entire set of questions, and answering each one with a negative or non-applicable option, became an intellectually numbing experience. It was also noted that working alone using the tool was less than effective. The approach of using a pair or team of analysts was later found to be more effective in sustaining the attention of the analysts.

At the end of the rigorous and iterative process, the Software Development Impact Statement was generated by the tool. This was a register of concerns and the actions for their potential resolution.

In all, a total of 103 concerns with the high level requirements were identified. To facilitate managing such a long list common threads running through the list were identified and labeled. This grouping helped in communications for all involved. In the case of the UK project, this taxonomy was used for communications with governmental officials and politicians.

The team also used these outputs to guide a re-application of the SPA to the requirements list. This iterative process was conducted until no new concerns were identified.

The concerns identified through the SoDIS Process as well as the potential solutions became the basis for formulating a RFP for outsourcing the design and development of the system. Some of the key concerns surfaced through the SoDIS process and included in the RFP had to do with designing the system to ensure voter secrecy and safety, equity of access, system performance and data integrity and security. The concerns of usability for minority groups and those with disabilities were also highlighted. There was also the concern that the means of authentication adopted should not be cost-prohibitive or result in an unacceptable violation of privacy.

The application of the SoDIS process to a high level set of requirements supported the belief that the process is useful in identifying potential problems with requirements. In principle the initial hypothesis was supported. The application of a project pre-audit will reduce the number and degree of the problems related to a software project. This pre-audit is useful even for outsourced requirements.

The lessons learned from this original analysis led to further modification of the SoDIS process and research on the modified process supported by SEPIA.

Table 2: Results from Applying the SoDIS Process to the UK Electronic Voting Project

No.	Key Result
1.	Tasks could be described at such a high level that they fail to lead to useful results.
2.	The analysis needs to be conducted on a developer (technical) level and a stakeholder level.
3.	The number of totally negative questions generated dulls the senses causing analysts to occassionally change focus and think of positive improvements for the project.
4.	The results of a SoDIS analysis, a large number of specific low level issues, resembles a kind of project specific grounded research. The data points, potential problems identified, can be grouped to help developers focus on categories of problems related to their system development.
5.	The process was iterative at every phase.

3. SEPIA RESEARCH PROCESS

Research into the use of SoDIS in outsourced software projects has been conducted under the umbrella of the Software Practice Improvement Alliance [SEPIA] (Clear *et al*., 2003). The adopted research approach is a form of "practical action research" (Carr & Kemmis, 1983), aimed at improving software engineering practices. This research was based upon principles learned from work done in the UK.

The process used in the UK analysis showed that simply using the SoDIS Project Auditor on a list of tasks or requirements was inadequate. This led to several adjustments during the analysis process. Lessons learned from the UK process are shown in table two below.

These results led to the next hypothesis that the SoDIS process could be improved if it were grounded in a formalized context called a 'SoDIS Inspection'. The SoDIS Inspection it is believed could resolve or mitigate the problems identified in the Introduction about requirements Since the concept of a SoDIS inspection was grounded in the analysis of requirements, the initial SEPIA action research project was related to a commercial partner's high level requirements. First we present the SoDIS inspection model and then our research applying it with a commercial partner.

3.1 THE INSPECTION MODEL PROPOSAL (HYPOTHESIS)

The SoDIS inspection model is directly grounded in the 5 results from table two above. The inspection process we tested has 5+ stages: 1) Context scoping; 2) SoDIS audit; 3) concerns clustering; 4) cluster guided SoDIS review; and, 5) analysis summary, 5+). Although presented linearly, these steps are iterative.

3.1.1 Phase One

Result 1 indicated a need to get an initial understanding of the project. **Phase One** –*Context scoping*- involves understanding critical elements in the context including the identification of tasks and overlooked stakeholders.

This phase commences with meetings of project team members and meetings with the client and analysts. These meetings help the team to understand the client's perspective on the project and to identify the project's and extended stakeholders and project issues which are beyond the typical risk analysis considerations. This provides an initial project '*context of concern'* or an identification of concern from perspectives of developing the product and the developed product. It is believed that this will help the developer begin to look beyond the purely technical side of development.

This process helps resolve result 1 of the UK project. Context scoping provides an organization's context and some preliminary directions about where to focus the initial SoDIS audit. Bringing the user and stakeholder perspective into this phase also starts to address the second UK research result.

3.1.2 Phase Two

In **Phase two** - the SoDIS Audit- the 'Context scoping' structures the SoDIS analysts selection of tasks to start their analysis approach and determines the number and types of questions produced by the SoDIS Project Auditor (SPA). This phase searches in a structured way for potential issues related to the project's development, delivery, or use.

Working through the significant number of SPA generated questions and assuming multiple roles in answering them is very wearing on the analysts. The negative nature of the questions also makes the process very repetitive. These problems have reduced the effective consideration of the SPA questions.

Research results from pair programming (Cockburn, n.d.) have shown that there is a synergy that produces better software design, continual review of each other's work leading to more effective defect removal, and enhanced problem solving ability.

During the SoDIS Audit process the SPA forces the analysts to first identify potential stakeholders for this project. The SPA provides a partial list of stakeholder types that have been associated with that type of project. Once the stakeholders have been identified, the analysts answer questions, seeking to identify and note potential negative consequences for the identified stakeholders or for the project and, where possible suggest solutions for the identified items.

This audit is repeated because 1) during the audit, new stakeholders are identified generating new questions, 2) answering the questions generates a new and more complete picture of the project which helps clarify issues analysts addressed earlier and 3) the suggested solutions to earlier concerns may in fact introduce new concerns.

The resulting Software Development Impact Statement indicates all concerns for the project and project impacts on citizens and organizations. To utilize the occassional change of focus to positive values we modified the process. During this phase, a Positive Modification Form (PMF) is also maintained. The analysts are provided the opportunity to simply record any ideas they have for improving the value of the project. This captures positive creative thought and reduces the sense of negativism identified in the UK election study.

The Audit results in a list of concerns and potential solutions and a list of positive impacts. These lists are the input into the next phase.

3.1.3 Phase Three

In **Phase three** - Concerns Clustering- the SPA reports are used to identify trends in the analysis data. The goal of this phase is to provide high level abstractions used to further SoDIS analysis and to provide high level risk categories that developers can use in reviewing their projects. Cluster analysis is a technique that enables the project to be rebuilt along a set of new perspectives. The phase meets needs identified in the fourth result of the electronic vote analysis.

These clusters can convey clear meaning to all levels of client and the clusters provide the platform on which to undertake practical action to address the projects identified ethical risks. The cluster list is also used as a guide for subsequent analyses.

Clusters can be given priority based upon the priority of individual issues within the cluster together with an overall view of the cluster's relative importance. This allows them to evaluate criticality and prioritise clusters. Cluster analysis is a technique that enables the project to be rebuilt along a set of new perspectives; thus addressing result 4 from the UK election research.

The cluster analysis also facilitates communicating concerns with key stakeholders who stand outside of the project development. The Clusters also serve as a filter to determine the completeness of the analysis in Phase 4. The Cluster Analysis document (CAD) is used as input to the next phase.

3.1.4 Phase Four

In **Phase Four -** *Cluster guided SoDIS review* – using the developed cluster breakdown structure, the analysts identify missed issues and stakeholders by comparing the task list and the stakeholder list to the Cluster list. Access to the PMF is also maintained in this stage. This process is repeated until no new clusters or no unidentified issues are identified. This is the phase gate to the development of the SoDIS Inspection Analysis Summary document in Phase 5.

3.1.5 Phase Five

When the cluster analysis review is completed a *SoDIS Analysis Summary* is produced. This is an overview of the results of the inspection showing the cluster structure and indicating the priority of the cluster issues. This document is then used to determine what needs to be addressed for the project to continue in a positive direction or determine if it is to be terminated. This document becomes part of the project library that can be used on similar projects.

During this phase the PMF is revisited to tidy up the positive suggestions before turning the document over to management for further review.

3.1.6 Phase Five +

Work at any phase of the inspection is likely to uncover new information relevant to another phase. That phase should be revisited to analyze the impact of the new information on the project and the project's stakeholders.

4. TEST PROCESS ON OUTSOURCED REQUIREMENTS

The SoDIS inspection process can be used at most phases of software development. Some form of requirements is common to all software development models. The normal risks of requirements gathering are exacerbated by outsourcing. SEPIA supported research of a commercial project in which customer developed requirements were outsourced to diverse developers.

4.1 Applying the New SoDIS Inspection Model

4.1.1 Background

Working with a company in New Zealand an action research study was undertaken using students and staff as SoDIS analysts.

There was a high potential for requirements problems because they were developed in-house and their implementation was outsourced. If the SoDIS could help mitigate potential problems in this worst case then we can have a higher degree of confidence in our hypothesis.

The Company " NZ*" gathers information from various sources and locations in New Zealand and makes statistical summaries of that information available to its members and to requesting government agencies. Anyone who works in the NZ* business sector must also be a member of NZ*. The members use the information made available by NZ* in their business. Their member organizations are a variety of types and sizes.

NZ*'s paper intensive system has inputs in a variety of formats. Because of the data entry formats, the wide variety of IT systems employed by members, and the types of member organizations submitting information, the statistical data is sometimes significantly out of date. There is no way to verify the data in the current paper system before it is used to generate the statistical information for its members. NZ* is responsible for recording and tracking the complaints against its members.

NZ* is planning an automated replacement for the current system. The database design and security will be outsourced to one company and web development and security to another company. The Content Management System to process additions, alterations of the data, and member input and retrieval formats will be a purchased off the shelf system. The hardware technology is leased from a US company.

4.1.2 The Inspection and NZ*

4.1.2.1 Phase One

In the initial meetings the team learned about a brand new domain to them and NZ* learned about the inspection process. The enormity of NZ* replacing its entire manual system was being treated with great care. NZ* described their business and characterized its primary stakeholders as its members. Because the NZ* representative presented a technical view, the inspection team represented the user community. The high level of the NZ* plan presupposed familiarity with the system. Most of the stakeholders identified came out in the explanations (scenarios) of how the system would be used. The team then developed Context scoping documents listing initial concerns and potential stakeholders impacted by the system. The skills of the NZ* project manager had an intimidating effect on the analysis team.

4.1.2.2 Phase Two

Scoping documents were used to isolate some tasks for analysis. The choice of tasks was also partially influenced by the order in which NZ* was going to approach the project. This influence diminished during the analysis as the team came to believe that some tasks scheduled for the later stages of the project had to be addressed early on.

4.1.2.3 Phase Three

The concerns identified in Phase Two were reviewed and clustered together. A sample of some clusters identified is listed below:

1) Modeling the existing system-

a) There was an acceptance of inevitable inaccuracy or out of date data in the manual system. The initial design of the new system reflected that acceptance in the manual system. We felt this unsatisfactory because the new system would provide for greater distribution of this data. The PMF also suggested a potential way to mitigate this difficulty by using the collected data and sending the entered data back to the person who entered it for verification.

b) In the manual system complaints were handled by a data entry person who entered the complaint that had been faxed or phoned in. There was no verification of the accuracy of the data entry or the tone of the complaint. The review process identified the need for a new set of tasks facilitating complainant review and yet maintaining consistency with existing privacy laws.

2) Project Management- Postponing decisions- A generic issue emerged related to the overall development strategy. NZ* had planned to first develop the underlying hardware system to house and manage the types of data they had always managed in the manual system. Once the software and database were functioning they were going to then address issues of security and access. The SoDIS inspection revealed sufficient necessary and desired differences from the manual system that this generic approach had to be abandoned.

Observations of the Phase Three process

As expected some large clusters, e.g., modeling the existing system, were identified. This is too abstract to be useful as a filter in Phase Four so they were broken into sub-clusters: privacy-complaints, Data verification, multiple user interfaces, and database integrity.

One of the common problems in the cluster identification and analysis phase is the tendency to collapse similar clusters into a single cluster. Sometimes this causes the loss of significant information. Most of the cluster issues were agreed to, but the seriousness of some clusters was trivialized with the claim that they would be addressed at a later stage of the project or that “the outsource agency would surely take care of that.”

4.1.2.4 Phase Four

In this phase the analysts reviewed the NZ* task list comparing it to the cluster analysis document asking if there were any unidentified concerns which fit into the defined clusters or possible new clusters. The review indicated some additional subclusters. For example, the data accuracy cluster had missed the possibility that the system would not recognize that information was missing. This became a subcluster of the data accuracy cluster. Several places where reviews were planned did not allow any time for corrections which might be identified. A sub-cluster ‘review time’ was added to the project management cluster. After the project task review, a stakeholder perspective guides further review of the clusters. It was discovered that two significant stakeholder groups had been left out of the development plan and if they were not considered early the primary database would have to be redesigned or an arbitrary table added to the database which would significantly diminish its design quality.

4.1.2.5 Phase Five

The results of phases 2-4 are the foundation of the SoDIS Analysis Summary report, which provides the project manager with an early list ordered by criticality of potential issues identified in the inspection. Among these were a number of critical issues identified for the development process related to the database design and some negative impacts for the user community.

5. CONCLUSION

The work reported here reflects an ongoing programme of research into developing and refining the SoDIS process. Its application in different field settings has led to deeper understanding of the process and supporting case tool, together with their strengths and weaknesses, and has resulted in the new concept of a SoDIS Inspection. Work continues on refining the SoDIS process and the prototype CASE tool, with further development planned to incorporate positive improvement suggestions in addition to the negative process of noting concerns. Guidelines for conducting SoDIS Inspections are being developed, with further work required to better determine how far to extend the analysis and when to terminate the process.

Results from trials with commercial partners have shown significant improvement in project planning and requirements identification activities. In the UK electronic voting case, the original goal of e-voting by 2005 has been tempered by the large number of issues identified through SoDIS analysis, and a more measured programme of pilots to be conducted at local body election level has now been adopted (ODPM, 2003). The process has also identified specific issues for responding vendors to address in their e-voting proposals. In the NZ* case, several critical issues have been identified which have broadened the scope of the specified requirements to include the needs of additional stakeholders, and have also suggested some significant changes in the outsourcing strategy. Given the higher requirements risk inherent in the case of outsourced software development projects, the value demonstrated by the SoDIS process in these cases demonstrates its efficacy for more general application to software development projects of all kinds.

REFERENCES

Boehm, B., Grunbacher, P., & Briggs, R. (2001). Developing Groupware for Requirements Negotiation: Lessons Learned. *IEEE Software*(May/June), 46-55.

Carr, W., & Kemmis, S. (1983). *Becoming Critical: Knowing Through Action Research*. Melbourne: Deakin University press.

Clear, T. (2003). The Waterfall is Dead - Long Live the Waterfall!! *SIGCSE Bulletin, 35,* 13-14.

Clear, T., McHaney, R., & Gotterbarn, D. (2003, 6-9 Jul). *SoDIS SEPIA -Collaborative Partnerships in Software Engineering Research.* Paper presented at the NACCQ Conference, Palmerston North.

Cockburn, A and Williams, Laurie "The Costs and Benefits of Pair Programming" accessed http://collaboration.csc.ncsu.edu/laurie/Papers/XPSardinia.PDF

Fairweather, B., & Rogerson, S. (2002). *Technical Options Report [Online available: http://www.odpm.gov.uk/stellent/groups/odpm_localgov/documents/page/odpm_locgov_605190-03.hcsp, accessed 10/03/2004]*. London: Office of the Deputy Prime Minister.

Heeks, R., Krishna, S., Nicholson, B., & Sahay, S. (2001). Synching or Sinking: Global Software Outsourcing Relationships. *IEEE Software, 18*(2), 54-60.

ODPM. (2003). *The Government's Response to the Electoral Commission's Report: The Shape of Elections to Come - A Strategic Evaluation of the 2003 Electoral Pilot Schemes.[Online available: http://www.odpm.gov.uk/stellent/groups/odpm_localgov/documents/page/odpm_locgov_024001.hcsp. accessed 10/03/2004]* (No. Cm 5975). London: Office of the Deputy Prime Minister.

Rogerson, S., & Gotterbarn, D. (1998). The Ethics of Software Project Management. In G. Collste (Ed.), *Ethics and Information Technology*. Delhi: New Academic Publisher.