

Association for Information Systems AIS Electronic Library (AISeL)

International Research Workshop on IT Project
Management 2006

International Research Workshop on IT Project
Management (IRWITPM)

December 2006

Using the Troubled Project Recovery Framework: Problem Recognition and Decision to REcover

Douglas Havelka
Miami University

T.M. Rajkumar
Miami University

Follow this and additional works at: <http://aisel.aisnet.org/irwitpm2006>

Recommended Citation

Havelka, Douglas and Rajkumar, T.M., "Using the Troubled Project Recovery Framework: Problem Recognition and Decision to REcover" (2006). *International Research Workshop on IT Project Management 2006*. 3.
<http://aisel.aisnet.org/irwitpm2006/3>

This material is brought to you by the International Research Workshop on IT Project Management (IRWITPM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in International Research Workshop on IT Project Management 2006 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Using the Troubled Project Recovery Framework: Problem Recognition and Decision to Recover¹

Douglas Havelka

Dept. of Decision Sciences and MIS
Richard T. Farmer School of Business
Miami University
Oxford, OH-45056
havelkdj@muohio.edu

T.M. Rajkumar

Dept. of Decision Sciences and MIS
Richard T. Farmer School of Business
Miami University
Oxford, OH-45056
rajkumtm@muohio.edu

ABSTRACT

Information system development projects continue to be delivered significantly behind schedule, drastically over budget and without meeting specifications. To improve the likelihood of success for these projects a staged framework for recovery and rehabilitation has been suggested. This framework is composed of four stages that include twelve steps. The four stages are recognition, immediate recovery, sustained recovery and maturity. A study was performed to validate the first stage of the framework, the Recognition Stage. The study results indicate support for the steps of the Recognition Stage and provide detailed “symptoms” that could indicate trouble during an IS development project. These symptoms were logically categorized into 11 areas of interest to project managers: Client/Stakeholder, Goal, Meeting, Team, Task, Project, Project Management, Communication, Management, Project, and Process.

Keywords

Projects, Project Management, Troubled Projects, Information Systems Development.

INTRODUCTION

More than \$275 billion is spent every year in the US on information systems development projects (Standish Group, 2000). However, most information technology (IT) software development projects are not successful in at least one of the key measures: over budget, over schedule, or do not meet specifications. Some research has shown that only 28% of software development meets the three success criteria and as many as 72% of software development projects are “troubled.”

In one year nearly half of all projects were reported to be challenged, i.e. they were over budget, behind schedule, and the end product did not meet user requirements. Projects that are canceled prior to completion, i.e. impaired projects, make up nearly a quarter of all projects. Prior researchers have used the following measures as indicative of trouble: (1) being over schedule by 30%, (2) being over budget by more than 30%, and (3) having an end product that does not meet user requirements (Whittaker, B., 1999). Glass (1998) uses a more restrictive definition and states that a runaway project is one where the schedule, cost, or functionality is twice as bad as what was sought. Regardless of the label used for troubled projects or the rule for classifying a project as in trouble, it is clear that schedule, cost, and functionality goals are not being adequately achieved by IT project managers.

In addition, it appears that businesses do not seem to learn from past mistakes in software development. They repeat the same mistakes over and over again and are habitually in trouble (Tarek, and Madnick 1990). A reason for this may be the fact that a majority of software development improvement is done at the project management rather than at the organizational level (McConnell, 2002a). In a study that measured project management maturity across different industries including engineering construction, information management and movement (telecommunication), information systems development, and high tech products and services; information systems (IS) companies ranked lowest in maturity, achieving only a mean of 3.07 out of a scale of 5 (Ibbs and Kwak, 2000). The Software Engineering Institute has developed a model called the capability maturity model which measures the software development capabilities of companies (Jalote, 1999). Many US companies rank at Level 1 (initial level) or 2 (repeatable level), which are the lowest levels. This is indicative of the lack of standard, consistent, and predictable project management processes.

¹ A preliminary version of this paper was presented at the 2004 Americas Conference for Information Systems.

To be considered a “recovery” a project headed towards some kind of disaster or cancellation is at least able to overcome the initial problem situation and achieve at least average results (Capers Jones, 1996). Recovery may not always be possible. Early discovery of the problem should lead to improved likelihood of success or recovery. It is proposed that if trouble in projects is not discovered or if discovered and left unattended, the project will have a higher likelihood of failure. We contend that in the software industry problems are seldom detected early enough. Based on this belief we attempt to address the following research question:

How can organizations identify trouble in an IT development project (sooner)?

To address this question, a field study was conducted using a nominal group process to identify symptom or symptoms of trouble in IT development projects. The rationale for the field study is based on the Trouble Project Recovery Framework proposed by Aiyer et al. (2005). Different perspectives of troubled projects and recovery processes from software project management, de-escalation of project commitment, and crisis management exist. All of these areas contribute to our understanding of how to improve troubled projects.

It has been proposed that both long-term and short-term recovery methods are important (Connell, 2000) for IT project management success. The capability maturity model and ISO 9001 that help all projects across the organization would be examples of long-term methods. These approaches have been found to be effective, but may be inappropriate or difficult to implement for an on-going troubled project. For a specific project in trouble, short-term recovery methods are needed. Connell suggests: an initial assessment is used to identify the areas of trouble in the project and a plan is then developed, the purpose and final objectives and goals of the project should be reviewed and agreed upon by all major stakeholders, modest, stable, and achievable goals for imminent milestones should be established, realistic schedules are set, the project is tracked with source control, automatic builds are in place, and efficient processes are specified to track and fix bugs. Connell’s method seems to place an emphasis on the programming as opposed to the overall life cycle of the project.

The decision to add additional resources to a failing or troubled project is referred to as the escalation of commitment. Through de-escalation, troubled projects may be successfully turned around or sensibly abandoned (Keil 1999). Keil (2000) looks at the some recovery processes from the de-escalation perspective. Keil (2000) identifies the following steps as necessary: problem recognition, reexamination, selecting alternative courses of action, and implementing an exit strategy. By following these steps, organizations should do a better job of supporting projects more likely to succeed and ending projects that have a strong chance of failing. Alternate scenarios to kill projects have been presented (Kapur, 2001). Kapur’s approach includes: check for vital signs of the troubled project, meet with decision makers so that key stakeholder and senior management buy-in can be obtained before a decision to kill or recover the project is taken; if the decision is to recover, then the manager must put together both a recovery plan that outlines the steps needed to bring the vital signs back to acceptable levels and a mechanism to track the project recovery; if the decision is to kill the project, the aim then is to put together a cancellation plan, obtain stakeholder approval to the plan, and kill the project while causing the least amount of harm; finally, salvage usable project components and conduct a post mortem analysis that includes team members and stakeholders to glean lessons from the project so that the mistakes are not repeated again. In general the de-escalation literature focuses on how to cancel a project when appropriate and cut losses to the organization. Recovering and rehabilitating the project is only a secondary focus and is not as well developed.

Another area of related research is crisis management: the process of assessment, response, mitigation, and relief of crisis situations (Seeger and Ulmer, 2001). It is defined as the immediate response to such events, recovery efforts, and mitigation and preparedness efforts to reduce the impact of future crises (National Research Council, 1998). Immediate response is dedicated to the immediate protection of life and property. It requires urgent action and the use of coordinated resources and facilities not available to routine problems. Recovery efforts encompass both short-term activity intended to return vital support systems of the project to operation and longer-term activities designed to return infrastructure systems to pre-disaster conditions. These activities are directly analogous to the recognition and recovery stages of the Troubled Project Recovery Framework (Aiyer et al., 2005). In troubled software projects, immediate response might involve recognizing the problem and communicating to the stakeholders the nature and magnitude of the problem. Recovery efforts include assessing the problems and taking quick corrective action to restore the health of the project, such as changing team composition when one member of the team creates conflicts within the team (Brown et. al. 1998, p. 235). Recovery also includes long-term plans such as reorganizing and implementing a new base plan.

Mitigation includes steps such as documenting the problems, preparing measures for corrective actions, and preparing specific actions to address each problem. Preparedness is the process of risk management: identifying, analyzing, and elaborating the symptoms and creating contingency plans to mitigate the effects of the risk. For IT projects, documenting the problems identified and the success or failure of corrective actions can provide lessons learned for these processes and can be used to proactively prepare risk management procedures to be used in the current and future projects.

Some research has directly focused on crises in IT projects (Boundy and Diamond, 1998). Again, a set of steps are suggested for dealing with the crisis: recognizing and acknowledging the crisis including discussion of the crisis and consideration of solutions to the problem, identifying the magnitude of the problem and communicating it to the relevant stakeholders, ensuring that the problem is being handled by the right people within the team including escalating the problem to higher levels and to apply additional resources (such as a consultant) if needed and appropriate, continuing work on other parts of the project for which people can be spared while keeping the resolution of the crisis as priority number one, and ensuring that management and stakeholders are updated daily with regard to the crisis.

Some confusion in the troubled projects area is also attributable to the varying definitions of project failure or success (Belassi and Tukel, 1996, Linberg 1999, Klein and Jiang, 2001). Different parties to the project might view success and failure differently (Pintro and Slevin, 1988). IS professionals have deemed systems as successful even when a system is 417% over cost and 193 % behind schedule (Linberg, 1999), because their notion of success was based upon the knowledge gained by the innovative nature of the solution and the relationships built within the team. Such views, while deviant from normal perceptions of project management success, indicate how different groups view success criteria differently. Other factors include the fact that IT professionals might view success from a narrow, information technology perspective and miss the overall business and organizational environment within which the software solution is being deployed (Day, 2000).

As can be seen throughout the literature, a logical, structured approach with specified activities and tasks is recommended to help recover (or kill) troubled projects. The Troubled Project Recovery Framework proposed by Aiyer et al. (2005) provides just such an approach for use in IT project management.

THE TROUBLED PROJECT RECOVERY FRAMEWORK (TPRF)

The Troubled Project Recovery Framework is composed of four main stages: (1) problem recognition, (2) immediate recovery, (3) sustained recovery, and (4) maturity. This framework was built with prior research as a foundation and is similar to the crisis management approach which also has four main stages (immediate response, recovery, mitigation and preparedness) and the four stages of de-escalation that Keil (2000) proposed (recognition, reexamination, selecting course of action, and exit strategy). The conceptualization of the four-stage 12-step recovery framework is shown in Figure 1. The four main stages are each composed of two to four steps. A short discussion of the four stages and the twelve steps follows.

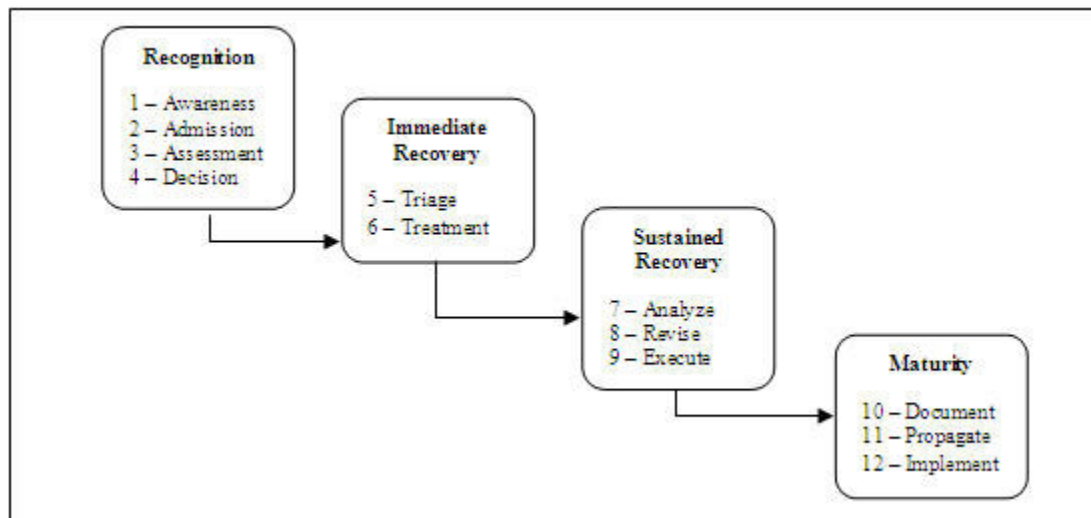


Figure 1. Framework for Troubled Project Recovery

Stage 1: Problem Recognition and Decision to Recover

Before any project can be recovered, the fact that the project is in trouble must be recognized. A decision to recover is then made and the recovery process begins. The steps of the first stage include: 1) problem awareness, 2) admission of trouble and seeking help, 3) quick assessment and review, and 4) decision to recover.

Step 1 is problem awareness. When projects are in trouble it is expected that they would display symptoms of the trouble, similar to a patient displaying symptoms of a disease. Many times some type of trigger or other significant event occurs that

demands a response or action. This may be in the form of an event, i.e. a subcontractor quitting the project, or in the form of a state (red flag) that needs attention, i.e. missed milestones or deliverable, that indicates things are not operating as they should. In addition, many organizations do not use good estimation tools or techniques and lack any kind of historical measures or metrics to make comparisons during project management (Capers Jones 1995). Therefore, projects may get into serious trouble without knowledge of the project manager or other stakeholders until the projects are very late in development. These symptoms may be identifiable and monitored to help project managers recognize and acknowledge trouble earlier in a project's life and thereby increase the likelihood of successful recovery.

Step 2 is admitting that trouble exists and to get help. According to a survey by the center for project management (Kapur, 2001), many project managers are afraid of being labeled as quitters or failures and only 20% have a process for identifying troubled projects. This suggests that project managers may resist or ignore symptoms until it is too late for recovery. Denial is an unconscious coping mechanism to block out and not deal with major change that may have some pain associated with it (Kiechel, 1994). Project managers recognize trouble, but deny that they are in trouble (Brady and Demarco, 1994), keep silent and do not report the bad news (Smith et al. 2001). Project managers tend to hold on to a problem too long and do not ask for help, thinking it makes them look bad (Boundy and Diamond, 1998). Keil and Robey (2001) also identify a deaf effect; i.e. even when the trouble is reported, the manager may fail to hear it. By ignoring the trouble report, managers may insulate themselves from dealing with the problem.

Step 3 is to conduct a quick and honest assessment and review of the project status. This assessment must be a quick review of the project execution and its true status. The aim is to create as complete and accurate an assessment of the project as possible in a short time frame. A check for the vital signs of the project must be conducted (Kapur, 2001). This includes a check of planned vs. actual achievement on the schedule, resources used, milestones met, and deliverables met. Also a comparison of estimated vs. actual cost of the project to date must be made.

Step 4 is the decision to recover the project or to kill it. Once an assessment is made, a decision as to whether the project should be recovered or canceled must then be made. Once the decision to recover is made in consultation with upper management and stakeholders, an internal commitment to recover must be made. Not only must a decision to recover be made, but one must truly believe that project recovery is possible. Buy-in to the commitment must come from both stakeholders and managers.

To partially validate the TPRF, a field study composed of focus groups using a nominal group technique was performed to identify a set of symptoms or symptom for trouble during IT project development. The study is discussed in greater detail following a quick discussion of the remaining stages in the TPRF.

Stage 2: Immediate Recovery Stage

In this stage, steps are taken to nurture the troubled project back to stability. The aim in immediate recovery is to identify and isolate the critical problems and take immediate corrective or mitigating action. The aim is not to revamp the complete plan. The two steps in this stage are triage and treatment. These steps may occur in sequence or concurrently and iteratively within the recovery process. The triage step, similar to triage in major accidents or natural disasters, the most critical problems are identified, isolated, and prioritized. The major problem may be identified in Stage 1 and additional problems or issues may have been identified in the assessment step of Stage 1. For example, a trigger may have been the loss of the project manager and the assessment reveals several missed milestone deliverables and cost overruns. The issues should be prioritized so that the treatments in the next step occur appropriately. Closely linked to the triage step is the treatment step. As the critical problems are identified and prioritized, a treatment action is determined for each critical problem. For example, the sponsor may decide to forgo an imminent milestone until further analysis can be performed. The aim is to stabilize the project and bring the project to a stage from which a full recovery can be made.

Stage 3: Sustained Recovery

The long-term or sustained recovery process can be put in motion once the project is in a stable stage. The steps include analyzing the project status and creating an issue list with resolutions for addressing the issues, creating a revised baseline plan and obtaining management approval, followed by executing and monitoring the revised plan. Building on the prior steps there should be an initial inventory of issues that need to be analyzed and appropriate action planned. By incorporating resolutions into the original project plan and making further adjustments as necessary, a new baseline plan (revised master plan for the project) is now created. This might involve a reduction in scope, reprioritization of goals and objectives, and clarification of the objectives and expectations. A re-estimation of tasks will be performed, a new schedule will be drawn up, and resources allocated to the project accordingly. In addition, the mix of personnel and skills inventory present should be compared to the project requirements and appropriate actions planned, such as training in new technologies must be provided

to team members if needed. The risks involved in the project should be re-evaluated and plans for mitigating the risk must be established and evaluated. In essence, a complete new baseline plan is drawn up. This revised plan and milestones becomes the metric against which success for the project is now measured. The revised baseline must be put through the normal business value approval process. Management and customer approval for the new baseline must be obtained before proceeding further, i.e. a go-no go decision must be made. After the revised baseline is established and approved, the plan should be implemented and the status of the project must be continually monitored. Implementing the plan may include adding or redeploying resources. This may be human resources that may involve re-assigning, refocusing, or removing personnel on the team. Or other project resources such as software development tools or equipment may need to be purchased or changed in some manner.

Stage 4: Maturity Stage

The final stage is the maturity stage. This stage focuses on the on-going management of projects in general and is critical for increasing the return on investment in recovered projects. Post implementation reviews of projects to learn from successful and fruitless actions are not standard practice (Brady and DeMarco, 1994). Research indicates that failures continue to recur in organizations for various reasons (Flowers 1996). Organizational learning should help the individual project manager accomplish three goals: deliver a successful project, deliver a series of successful projects, and build project management capabilities (Kotnour, 1999). The detailed steps (or activities) in this stage are documenting the lessons learned from each project, propagating project management knowledge throughout the organization, and finally to implement this knowledge in other projects.

A FIELD STUDY TO IDENTIFY SYMPTOM OF TROUBLE IN IS DEVELOPMENT PROJECTS

In order to assist project managers in applying the TPRF a field study was performed to identify common symptoms or symptom that might cause or contribute to trouble during information systems development (ISD) projects. The purpose of the study is to identify and categorize these symptoms to allow managers to achieve the first stage of the proposed framework, i.e. recognize and acknowledge that trouble exists. This, in turn, should lead to the later stages where the set of symptoms could be used to improve, measure, or take corrective action on projects that are in trouble or may be heading for trouble. These results should be of interest to business managers and ISD project managers dealing with the project management process as well as to IS researchers attempting to better understand the ISD process.

Research Method

The field study consisted of a series of focus groups undergoing a nominal group process. The method used to identify the symptoms of troubled projects is a nominal group technique (Adam et al., 1986; Delbecq et al., 1975, 1982; Sutton, 1993; Sutton and Lampe, 1989, 1991; Van de Ven, 1974; Van de Ven and Delbecq, 1971, 1974). An underlying assumption of the method is that individuals who perform a task can provide valuable insight into the important factors influencing their ability to achieve a high level of productivity when performing the task (Adam et al., 1986). This method has been used successfully in several domains (Adam et al., 1986; Aubrey and Eldridge, 1981; Sutton and Lampe, 1991; Theby, 1981) including systems development (Havelka, 2002; Havelka et al, 2001, 1998; Hershauer et al., 1984).

In the current study, this method was used to identify symptoms of trouble on IS development projects. The nominal group technique used in the focus groups is a modification of procedures developed by Delbecq et al. (1982) and Sutton and Lampe (1991). The technique consisted of five steps. All five steps were conducted during a single meeting (of each group) that took approximately two hours. The technique is discussed in the next paragraphs and is summarized in Table 1. Detailed instructions for conducting the nominal group technique are available upon request.

The first step in the nominal group technique was a general discussion of the subject being addressed by the group. In this case, the subject chosen for study was the identification of symptoms that might indicate trouble on an IS development project. The objective of the first step was to provide a common understanding of the subject and to determine the scope of the topic under study. A general definition of IS development process, the tasks performed during the process, and the outputs of the process were given. The participants were then given the opportunity to clarify their understanding of the subject. After any questions were raised and discussed, group consensus was used to define the process and the boundaries of the problem to be addressed in the subsequent steps of the group process.

The second step in the nominal group technique was a period of silent generation of ideas. This was achieved by presenting a nominal question to be addressed by the group and asking them to record their thoughts individually and silently on worksheets. The objective of this step was to generate as many ideas as possible for later discussion. The participants were

encouraged to be creative and to interpret the question based on their own experiences and knowledge. The nominal question posed to the groups in this step was:

"What symptoms can you identify that might indicate that an information system development project is in trouble, i.e. it is over-budget, behind schedule, or not meeting specifications?"

The benefits of this silent, independent idea generation include: adequate time for thinking and reflection, social facilitation (i.e., constructive tension by observing other participants working), avoidance of interruptions, avoidance of prematurely focusing on particular ideas, sufficient time for search and recall, avoidance of competition, avoidance of status pressures, avoidance of conformity pressures, and avoidance of choosing between ideas prematurely (Delbecq et al., 1982). The nominal question being asked should be general enough that the participants interpret it based on their personal perspectives. To this end, the leader of the meeting should take special care to avoid answering the question for the group, thereby focusing the group on a particular viewpoint toward the question.

After the silent generation of ideas is complete, a round-robin recording of the ideas generated is performed. The leader of the group prepares an aggregate list of the participants' ideas by asking for one idea from each person on each round. The aggregate list was prepared in written format and was presented so that all the participants could view it. During this step, participants were encouraged to add new ideas to their lists. The benefits of the round-robin recording in step 3 include: equal participation in the presentation of ideas; increase in problem-mindedness; depersonalization of ideas from personalities; increase in the ability to deal with a large number of ideas; tolerance of conflicting ideas; encourages "hitchhiking" of ideas; and provides a written record and guide of the ideas presented. It is thought that by sharing ideas and equalizing participation the creativity of the group is increased (Delbecq et al., 1982). The round-robin format of eliciting the ideas also establishes an important group behavior. After two or three rounds, each member in the group is an achieved participant. This sets the precedent for further active participation.

After recording of the ideas is complete, each idea was discussed for clarification and definition so that the group came to a consensus as to the item's meaning. Discussion of each idea was performed to share the logic behind the idea, the relative importance attached to the idea, and to air agreements and disagreements. The benefits of discussing each idea include: opportunity for clarification and elimination of misunderstanding; opportunity to provide logic for arguments and disagreements; and recording differences of opinion without undue argumentation (Delbecq et al., 1982). To accommodate these benefits the leader should make it clear that disagreements regarding the relative importance of each idea is acceptable and expected. A final note regarding this step, it should be emphasized that the individual responsible for suggesting an idea is not directly responsible for clarifying the idea. It should be stressed that the clarification is a group task, and not the unilateral responsibility of the author of the idea (Delbecq et al., 1982).

The final step in the nominal group technique was to gather demographic data from the subjects and an informal de-briefing. Information regarding each subject's project management experience, training, IS development experience, education level, etc. were gathered using a questionnaire. The informal de-briefing usually consists of a comparison of the subjects inputs and evaluations of the group results and the process overall.

There were four focus groups; the first was used as a "pre-test" of the data gathering method and to establish some baseline constructs. Each group met for approximately two hours. Group I consisted of five subjects. All the subjects were at the time of the focus group employed as instructors for a large mid-western university and had significant (> 5 years each) experience as IT consultants for information services organizations. Group II consisted of five subjects all of whom worked in the IT department of a large financial services firm and have acted as the project manager for IT projects for a minimum of six years (average = 8.4 years). Group III consisted of six subjects from the same IT department as Group II. These subjects had at least 3 years of IT project management experience and averaged 10.1 years of IT project management experience. Group IV consisted of four subjects from two separate organizations, one a large information services firm and the other a major telecommunications firm. These subjects had an average of 4 years of IT project management experience. The output of these focus groups included a list of symptoms from each participant and a comprehensive, master list of symptoms for each group.

Results

The results of the field study will be presented as follows: a general discussion of the focus group outputs is given and then a discussion of the symptoms identified is given within a logical categorization of the symptoms. The four focus groups yielded a total of 157 separate potential symptoms; the groups identified 39, 51, 43, and 24 symptoms respectively. After review, 49 of the symptoms were considered duplicates leaving 108 unique symptoms. Of the 108 unique symptoms

identified, only 7 of these were identified by all four of the groups, 9 were identified by three of the four groups, 22 were identified by two of the four groups, and 70 were identified by only one of the four groups.

Using prior research as a guide, a set of potential logical categories for the symptoms was identified. This set of symptom categories was determined by having independent judges (the researchers and colleagues) assign each of the symptoms identified to a category. The researchers then finalized these categories collaboratively and allowed a specific symptom to be assigned to multiple categories as appropriate. There were 11 categories identified in the final model. This set included: 1) client or stakeholder-related symptoms, 2) symptoms related to the project’s goal, 3) symptoms related to meetings, 4) team-oriented symptoms, 5) task-oriented symptoms, 6) symptoms related to the project’s characteristics, 7) symptoms related to the project management of the project, 8) symptoms related to communication issues, 9) management-oriented symptoms, 10) project-portfolio symptoms, and 11) process-related symptoms. Each of these categories is discussed below and the associated symptoms are presented.

The first category of symptoms identified contains symptoms that are all related to the client or other stakeholders of the project. These symptoms are presented in Table 1. These symptoms are in one way or another dependent upon or driven by the project’s client(s) or other stakeholders (excluding members of the IS department or project team members). As can be seen by the symptoms identified, the root cause of many of these symptoms appears to be related to the client’s commitment, buy-in, and support of the project. Also, management of client’s expectations and a clear specification of the project’s goal would appear to be related to the client-related symptoms. This leads directly to the second category: goal-related symptoms.

| | I | II | III | IV |
|---|---|----|-----|----|
| Loss or lack of project sponsorship. | | ✓ | ✓ | |
| Sponsor demands end date before scope and schedule are set. | ✓ | ✓ | ✓ | |
| Efforts to redefine project. | ✓ | ✓ | | |
| Poorly defined requirements – sponsor continually adding requirements (scope creep). | | ✓ | | |
| Different user groups communicate different points of view – adding to the complexity of the project. | ✓ | ✓ | | |
| Lack of subject matter expert or user involvement. | | ✓ | | |
| Lack of commitment by major stakeholders. | ✓ | ✓ | | |
| Client informs that project is in trouble. | ✓ | | ✓ | |
| Client does not understand their needs. | | | ✓ | |
| Sponsor does not want to meet with you (team members). | | | ✓ | |
| Unrealistic stakeholder expectations. | | | ✓ | |
| Lack of willingness to compromise | | | ✓ | |

Table 1. Client/Stakeholder Symptom

The focus group subjects identified 10 symptoms that are included as Goal-Related Symptoms (presented in Table 2). These symptoms are all characteristics or attributes of the project’s objectives and desirable outcomes. It appears that these symptoms become relevant when there is a lack of clearly defined purpose, scope, and agreed upon metrics for success for the project. These symptoms may become more important when there are multiple clients with multiple expectations from the project.

| | I | II | III | IV |
|---|---|----|-----|----|
| Lack of clear scope definition. | ✓ | ✓ | ✓ | ✓ |
| Efforts to redefine project. | | ✓ | ✓ | |
| High amount of change requests (II) or fluctuations of the project (IV) | ✓ | ✓ | | ✓ |

| | | | | |
|---|---|---|---|--|
| Project does not meet business strategy. | | ✓ | ✓ | |
| Reoccurring questions about the value of the project. | | ✓ | | |
| Lack of success criteria. | | ✓ | | |
| Lack of definition of quality. | | | ✓ | |
| Client does not understand their needs. | | | ✓ | |
| Lack of clearly articulated goals. | ✓ | | | |
| Conflicting objectives of participants. | ✓ | | | |

Table 2. Goal Symptom

A handful of symptoms were identified that centered on meetings. These symptoms appear to indicate trouble in two ways: 1) evidence that team members have “given-up” by no longer attending or participating in meetings and 2) an increase in meeting activity with clients and management that indicates something out of the ordinary is happening. Regardless, changes in the quality, quantity, and scheduling of meeting may be useful metrics to gauge a project’s health.

| | I | II | III | IV |
|--|---|----|-----|----|
| Lack of team participation at meetings. | | ✓ | | |
| Reduced attendance at project meetings. | ✓ | ✓ | | |
| Too many meetings. | | ✓ | | |
| Increased meetings with clients to clarify problems. | ✓ | | | |
| Senior management involvement at project meetings. | ✓ | | | |
| Change in meeting frequency. | ✓ | | | |

Table 3. Meeting Symptom

The Team-Related Symptoms represent characteristics of the team or team behaviors that may indicate trouble on the project. These symptoms included items such as the skill-set of the team members as well as team morale and stress level. These symptoms emphasize the importance of creating and managing a well-blended, high-performance team. As Table 4 shows, there are four symptoms in this category that were identified by all four groups: high team turnover, high stress levels, depletion of resources, and morale. Clearly, these symptoms could be useful to IT project managers as indicators of trouble or risks that may need special attention.

| | I | II | III | IV |
|---|---|----|-----|----|
| Team fighting. | | ✓ | ✓ | ✓ |
| High team turnover. | ✓ | ✓ | ✓ | ✓ |
| Lack of communication on team. | | ✓ | ✓ | |
| Resources dragging their feet on tasks – hoping they will go away. | | ✓ | ✓ | ✓ |
| Lack of critical skills. | ✓ | | ✓ | ✓ |
| Lack of team buy-in. | | | | ✓ |
| Different levels of expertise within the team – communication problems. | | | | ✓ |
| Stress. | ✓ | ✓ | ✓ | ✓ |
| Adding new resources. | | | | ✓ |
| Depletion of resources (allocation to other projects). | ✓ | ✓ | ✓ | ✓ |

| | | | | |
|--|---|---|---|---|
| Morale. | ✓ | ✓ | ✓ | ✓ |
| Team does not understand requirements. | | | ✓ | |
| Lack of willingness to compromise | | | ✓ | |

Table 4. Team Symptom

Several symptoms were identified that could be considered task-related. These symptoms were associated with specific tasks and activities that need to be performed during software development or the assignment, documentation, or completion of these tasks during the project. In general, it appears that most of these symptoms point to a lack of good project management. Issues with scoping and task assignment are regularly the responsibility of the project manager. This suggests that perhaps these symptoms should be monitored by the project’s client sponsor.

| | I | II | III | IV |
|--|---|----|-----|----|
| Limited, or lack of, documentation – not documenting tasks completed or technical specifics. | | ✓ | | |
| Consistent levels of overtime. | ✓ | ✓ | ✓ | ✓ |
| Design problems discovered during code review. | | | | ✓ |
| WBS (work breakdown structure) not assigned | ✓ | | | |
| Major defects identified during the inspection – changes in the requirements or solution. | | | ✓ | ✓ |
| High amount of rework or defects. | | ✓ | ✓ | |
| Training deferred or postponed. | ✓ | | | |

Table 5. Task Symptom

The next category is made up of symptoms that are related to the project itself, that is, these are symptoms that are particular or unique to the specific project. These symptoms include characteristics of the system being developed, i.e. size, value, as well as symptoms related to the project itself, i.e. adequate budget and resources. Some of these symptoms may also be considered project management symptoms, i.e. use of a sound methodology.

| | I | II | III | IV |
|---|---|----|-----|----|
| Over allocation of resources, more work than a person can do. | | ✓ | | |
| Constantly changing scope. | | ✓ | | |
| Lack of budget. People, time, and materials for the project. | ✓ | ✓ | | ✓ |
| Failing to use any methodology – team is not following the method or sponsor is not. | ✓ | ✓ | | |
| New technology unfamiliar. | ✓ | ✓ | | |
| Risk mitigation – delayed until end of the project (all major symptoms appear at the end of the project). | | ✓ | | |
| Scope is too large, indicator of size of project. | | ✓ | ✓ | |
| High CPI (cost/performance index). The costs higher than benefits. | | | ✓ | |
| Lack of budget for training | | | ✓ | |
| Team feels lack of ownership. | | | ✓ | |
| Lack of project value. | | | ✓ | |
| Budget unknown. | | | ✓ | |

Table 6. Project Symptom

A total of 21 symptoms were identified that were assigned to the category labeled project management. These symptoms were directly associated with how the project is being managed or the project management practices followed or established by the organization (or lack thereof). Similar to the task-related symptoms, this category may include primarily items that client or project manager's supervisor should monitor. One of these symptoms, project status unknown, was identified by all four groups. The fact that the project's status is not known, i.e. budget, schedule, or meeting specs, would be a clear indicator that trouble is present.

| | I | II | III | IV |
|--|----------|-----------|------------|-----------|
| Project status not known – poor status report meetings. | ✓ | ✓ | ✓ | ✓ |
| Schedule slippages – critical path. | | ✓ | | ✓ |
| Poor change management. | ✓ | | | |
| Schedule of work not defined – poorly defined schedule. | | ✓ | ✓ | |
| Revisiting issues that you thought were closed (if requirement not addressed or if addressed and keeps recurring DH) | | ✓ | | |
| Milestones are not met. | ✓ | | | ✓ |
| Difficulty in assessing the impact of change requests. | | | | ✓ |
| Variance of time estimates (high or low). | | | | ✓ |
| Constantly behind schedule. | ✓ | | ✓ | ✓ |
| Deliverables are not met on schedule. | | | | ✓ |
| Symptoms not identified. | | ✓ | ✓ | |
| Poor estimates. | ✓ | ✓ | | |
| No one to take ownership of issues. | | ✓ | | |
| Lack of leadership initiatives. | ✓ | | | |
| Late PM engagements. | | ✓ | | |
| Key risk issues not addressed. | | ✓ | ✓ | |
| Not able to define team resources up front. | | | ✓ | |
| Poor management of outsourced resources. | | | ✓ | |
| Wrong product or deliverable | | | ✓ | |
| Expenditures are overbudget. | ✓ | | | |
| Multitasking of resources. | | ✓ | | |

Table 7. Project Management Symptom

The next category was made up of symptoms that were all related to communication. This included communication among the team members, between the team and user groups, and among the team, user groups and management. Communication during systems development has been shown by previous research to be critical for establishing correct system requirements and to maintain good relationships between developers and users. From the symptoms listed it appears that communication would lead to errors and possibly re-work or incorrect features.

| | I | II | III | IV |
|--------------------------------|----------|-----------|------------|-----------|
| Lack of communication on team. | ✓ | ✓ | ✓ | |

| | | | | |
|---|---|--|---|---|
| Different levels of expertise within the team – communication prob. | | | | ✓ |
| Miscommunication between user and team. | | | ✓ | ✓ |
| Lack of “say” or “input.” | | | ✓ | |
| Escalation increases. | ✓ | | | |

Table 8. Communication Symptom

The next category includes symptoms that were all related to the organization’s general management, including support of upper management and management issues such as clearly defined roles and responsibilities. The importance of support and involvement of management in systems development projects has also received attention in prior research. Management can impact IT project success in many ways including control of resources to setting the tone that a project is important (or not) to the organization.

| | I | II | III | IV |
|--|---|----|-----|----|
| Weak project sponsorship – lack of commitment. | | ✓ | | |
| Pressure from upper level management – causes stress on team. | | | | ✓ |
| Reduction in staff – project still on schedule (original milestones to be met). | | | | ✓ |
| Micromanagement by upper management. | | | | ✓ |
| Lack of confidence in management. | | ✓ | | |
| Unclear roles and responsibilities. | | ✓ | | |
| Inability to commit resources. | | | ✓ | |
| Multicultural offshore teams – communications, time zone, cultural habits, language differences. | | | | ✓ |

Table 9. Management Symptom

The next category included was for project portfolio symptoms. These symptoms are all related to how the organization manages its portfolio of projects and where the project under consideration relates to other projects in the portfolio. These symptoms impact a project in several ways including projects doing redundant work or competing for resources. In addition, the value each project may add toward organizational goals may change over time and affect the priority of one project vis-à-vis another.

| | I | II | III | IV |
|--|---|----|-----|----|
| Cross-organizational blend between teams or management not in sync with each other – teams from different areas do not identify the same problem in the same manner. | | | ✓ | ✓ |
| Management and project not in sync. | | | | ✓ |
| Project interdependency. | | | ✓ | ✓ |
| Project has a low priority. | | ✓ | ✓ | ✓ |
| No portfolio management – two projects doing similar things. | | ✓ | | |
| Inability to get project prioritized. | | ✓ | | |
| Putting a project on hold while management works it out. | | ✓ | | |

Table 10. Project Portfolio Symptom

The last category of symptoms identified includes process-related items. These symptoms were primarily related to the IT development or implementation process being used. These symptoms could be considered the project manager's responsibility also; however, most of the items do not appear to be directly within the project manager's control, i.e. no quality assurance process in place.

| | I | II | III | IV |
|--|---|----|-----|----|
| Testing is rushed, limited, or eliminated. | | ✓ | | |
| Automating process without analysis | | ✓ | | |
| No lessons learned from a previous failure. | | ✓ | | |
| No QA process or evaluative process. | | ✓ | | |
| Lack of definition of quality. | | | ✓ | |
| Lack of documentation – results, planning, design. | | | ✓ | |
| No R&D prior to planning. | ✓ | | | |

Table 11. Process Symptom

In summary, the results of the field study yielded a total of 108 unique symptoms. Of these symptoms the vast majority were identified by only one of the groups and only seven were identified by all four. The researchers suggest 11 distinct categories for classifying the symptoms. Potential implications of these results are discussed below.

DISCUSSION

Overall, the results of the focus groups suggest that there are different types of “symptoms” that may present themselves during a project that could cause or indicate trouble. Some of the symptoms identified by the groups appear to be symptoms or indicators of trouble, while others are clearly the cause of trouble, e.g. lack of management commitment. Further study should be performed to follow-up on these symptoms to identify potential patterns that could apply across projects and organizations for these symptoms. Clearly, further research is needed to determine which of the symptoms might be causes of trouble and which are effects of trouble.

Additionally, the fact that only seven symptoms were identified by all four of the groups suggests that there may be a small set of "critical" symptoms that project managers and sponsors could track as early indicators of trouble. The seven symptoms are:

1. Lack of clear scope definition.
2. High team turnover.
3. Stress.
4. Depletion of resources (allocation to other projects).
5. Morale.
6. Consistent levels of overtime.
7. Project status not known – poor status report meetings.

CONCLUSION

Many information system projects run into trouble. Many can be recovered and rehabilitated to success. The TPRF has been outlined in this paper and a field study was performed to validate and provide guidance for managers using the TPRF. The four stages include recognition, immediate recovery, sustained recovery, and maturity. Without recognition, no recovery can take place. The field study identified a broad set of symptoms that could present themselves during a project that project managers can use to recognize trouble. The diversity of symptoms identified may indicate that there exist industry specific symptoms or that some symptoms may be more prevalent or more important in certain industries or application domains. These questions require additional research into the details of the causes and effects of trouble in IT development projects.

Additionally, more research is needed to identify successful solutions to common problems, perhaps building on the design pattern movement in software design.

REFERENCES

1. Aiyer, J., Rajkumar, T.M., and Havelka, D., "A Staged Framework for the Recovery and Rehabilitation of Troubled IS Development Projects," *Project Management Journal*, Research Quarterly, Vol. 36, No. 4, December 2005, p32-43.
2. Belassi, W, Tukel, O I "A New Framework for Finding Critical Success/Failure Factors in Software Development", *International Journal of Project Management*, Vol. 14, No. 3, pp. 141-151, 1996.
3. Boehm, B, Port, D, Al-Said, M "Avoiding the Software Model-Clash Spider Web", *IEEE Computer*, Nov. 2000, 121-123.
4. Boundy, G. and Diamond M, "Keeping Crisis out of Project Management" *Computing Canada*, Nov. 9, 1998, pp. 29,32.
5. Brady, S. and DeMarco, T. "Management Aided Software Engineering", *IEEE Software*, 1994, Nov, pp. 25-32.
6. Brown, W.J., Malveau, R.C., McCormick III, H.W. and Mowbray, T.J. "AntiPatterns: Refactoring Software, Architectures and Projects in Crisis", John wiley & Sons, New York, 1998.
7. Brownfield, C. "Software Glitch has DBCC Thinking about Going to Court". *Daytona Beach News Journal*, Oct. 10, 1998, Page 05C
8. Brownfield, C. "Software Firm Faces DBCC Suit". *Daytona Beach News Journal*, Dec. 11, 1998, Page 07C
9. Brownfield, C. "DBCC Sues Firm for Not Delivering Software". *Daytona Beach News Journal*, Jan. 6, 1999, Page 05C
10. Brownfield, C. "Computer Firm, DBCC Settle Software Lawsuit". *Daytona Beach News Journal*, Jan. 22, 2000, Page 07C
11. Caper Jones, D. "Patterns of Large Software Systems: Failures and Success" *IEEE Computer*, March 1995, 86-87.
12. Caper Jones, D. "Patterns of Software System Failure and Success", International Thompson Computer Press, 1995, Boston. USA.
13. Connell, C. "Healing Sick Software", <http://www.chc-3.com/>
14. Day, J. "Software Development as Organizational Conversation: Analogy as a Systems Intervention", *Systems Research and Behavioral Science*, 2000, V. 17, pp. 349-358.
15. Delbecq, A.L., Van de Ven, A.H., and Gustafson, D.H. Guidelines for Conducting NGT Meetings, In Hampton, D.R., Summer, C.E., and Webber, R.A. Organizational Behavior and the Practice of Management, Fourth Ed., 1982, Scott, Foresman, and Company, Glenview, IL, 279-298.
16. Drummond, H. "Are we Closer to the End: Escalation and the Case of Taurus" *International Journal of Project Management*, Vol. 17, No.1, pp11-16, 1999.
17. Fowler, A and Walsh M. "Conflicting Perceptions of Success and Failure in an Information Systems Project" *International Journal of Project Management*, 1999, Vol. 17, No. 1, pp. 1-10.
18. Glass, R.L "Failure is looking more like success these days" *IEEE Software*, Jan-Feb 2002, pp. 103-104.
19. Glass, R.L. "Software Runaways" 1998: Prentice Hall, Upper Saddle River, NJ-07458
20. Havelka, D. "Requirements determination: An information systems specialist perspective of process quality," *Requirements Engineering Journal*, v6, n4, 2002, 220-236.
21. Havelka, D., S.G. Sutton, and V. Arnold. "Information Systems Quality Assurance: The Effect of Users' Experiences on Quality Factor Perceptions" *Review of Business Information Systems*, v5, n2, Spring 2001, 49-62
22. Havelka, D., Arnold, V., and Sutton, S. "A Methodology for Developing Measurement Criteria for Assurance Services: An Application in Information Systems Assurance," *Auditing: A Journal of Practice and Theory*, v17 (Supplement), 1998, 73-92.
23. Ibbs, C.W. and Kwak, Y.H. "Assessing Project Management Maturity" *Project Management Journal*, March 2000, pp. 32-43.
24. Jalote, P. "CMM in Practice", Addison-Wesley, 1999.
25. Jiang, J.J, Klein, G., Balloun, J.L., Crampton, S.M. "System Analysts Orientations and Perceptions of System Failure", *Information and Software Technology*, Vol. 41, 1999, pp. 101-106.

26. Kapur, G. "How to Kill a Troubled Project", CIO Insight, 2001. http://www.cioinsight.com/print_article/0,3668,a=1932500.asp
27. Kiechel, W. "Facing up to Denial" Fortune, 1993, Vol. 123 (9), p. 163.
28. Keil, M. and Montealegre, R. "Cutting Your Losses: Extricating Your Organization Losses When a Big Project Goes Awry", Sloan Management Review, Spring 2000, pp. 55-68.
29. Keil, M. and Robey, D. "Blowing the Whistle on Troubles Software Development Projects". Communications of the ACM, April 2001, Vol 44, No. 7, pp. 87-93.
30. Keil, M. and Robey, D. "Turning Around Troubled Software Projects: An Exploratory Study of the De-escalation of Commitment to Failing Courses of Action". Journal of MIS, Spr. 1999, Vol. 15, No. 4, pp. 63-87.
31. Klein G. and Jiang J.J. "Seeking Consonance in Information Systems", Journal of Systems and Software, 2001, Vol. 56, pp. 195-202.
32. Kotnour, T. "A Learning Framework for Project Management", Project Management Journal, June 1999, pp. 32-38.
33. Linberg, K. R. "Software Developer Perceptions about Software Project Failure; A Case Study", The Journal of Systems and Software, 1999, Vol. 49, pp. 177-192.
34. McConnell, S. "The Business of Software Improvement" IEEE Software, July-August 2002, pp. 5-8.
35. NRC (National Research Council): Summary of a Workshop on Information Technology Research for Crisis Management, National academic press, available on line at (http://books.nap.edu/html/itr_crisis_mgmt/index.htm)
36. Pinto, J.K., and Slevin, D.P. "Project Success: Definition and Measurement Techniques", Project Management Journal, 1988, Vol. 19, 1, pp. 68-72.
37. Seeger and Ulmer, Virtuos Responses to Organizational Crisis, Journal of Business Ethics, 31:369-376, 2001.
38. Smith, J., Keil M, and Depledge, G. "Keeping Mum as the Project Goes Under", Journal of MIS, Fall 2001, Vol. 18, No.2, pp. 189-227.
39. Srinivasan, R. Anti-Patterns a Corrective Strategy in Software Development, <http://www.expressitpeople.com/20011210/management1.shtml>
40. Standish Group, Inc. "CHAOS: A recipe for Success", Res. Report 1999, <http://www.standishgroup.com/>
41. Tarek, A.M and Madnick, S. "The Elusive Silver Lining: How we Fail to Learn from Project Management Failures" Sloan Management Review, Fall 1990, pp. 39-49.
42. Whittaker, B. "What went Wrong? Unsuccessful Information Technology Projects" Information Management and Computer Security, Vol. 7(1), 1999, pp.23-29.
43. Wilson, S. "Failed IT Projects: The Human Factor" available at <http://faculty.ed.umuc.edu/~meinkej/inss690/wilson.htm>
44. Yeo, K.T. "Critical Failure Factors in Information System Projects" International Journal of Project Management, Vol. 20, 2002, pp. 241-246.