An Agent-Based Approach to the Automation of Risk Management in IT Projects

Kimberley Jackson, Dr Goran Soldar

School of Computing, Engineering and Mathematics,
University of Brighton, Brighton, UK.
Kimberley.jackson@me.com
G.Soldar@brighton.ac.uk

Abstract

The failure of projects in the field of Information Technology (IT) is a common problem. The aim of this work was to analyse and suggest a programmatic solution to improving the success of project outcomes. This was achieved through analysis of project variables to identify and quantitatively evaluate the risks surrounding a project and suggest possible mitigations to the project manager. The solution makes use of agent-based systems to monitor project risks and recommend mitigations in response to changing project variables. A knowledge base of common risks and related mitigations was collected through research and a selection of industry professionals in the field of IT with a wide range of experience in various areas of IT. This was used to provide a set of real world test data of risk/mitigation mappings to allow the suggestion and ranking of project risks based on a set of quantitative measures of risk likelihood. An impact for the risk is set by the project manager which, when combined with the calculated likelihood, can allow the calculation of a risk priority. The solution implementation takes two forms; a management tool for the project manager and a monitoring tool, which makes the project statuses visible to all in the project to aid in communication among the project team and stakeholder management, which are two common factors of failure.

The research results were positive and showed that the use of real world data in combination with an identified set of measures can identify risks and propose mitigations as well as facilitating communication improvement among project team members through the use of the website monitoring system. This encourages a clear and open approach to project, stakeholder and team management thereby increasing the likelihood of project success.

Keywords

Risk Management, Process Control, Monitoring Systems

Acknowledgements

With thanks to the research questionnaire participants for sharing their IT industry experiences.

1. Introduction

All IT projects run the risk of failure and in most cases this could be avoided if the risks had been addressed at an early stage and increased risk highlighted when it occurred. Success factors and reasons for failure are rarely analysed appropriately due to lack of time and resources, therefore a way to analyse and address these highly influential variables is needed to assist in the monitoring of IT projects and attempt to improve the success rate. The research aim is to address aspects of IT projects that can contribute to their failure, in particular, early identification and mitigation of risk as an aid to project management.

2. Project Success and Failure

Identifying the reasons for the success or failure of any project can often be a difficult task. Many factors can contribute to the outcome of a project from both human and non-human sources. Belassi et al. (1996) propose three criteria by which to define a project outcome:

- *Successful* – the project is completed on time, within budget and functionally meets the requirements
- *Challenged* – the project is completed and in operation but over budget, exceeded the time estimate and only part of the requirements have been met.

- *Failed* – the project is cancelled before it is completed.

The outcome of an IT project within a company can have a very large impact on future implementations, which can have a very large impact on whether the company remains competitive and effective. It is important to understand where a project has gone wrong in order to learn from previous mistakes and ensure that future undertakings are as successful as possible. Nelson (2005) highlights the importance of project retrospection in evaluating project success and failure with an aim to learn from the shortcomings of previous project implementations. They propose that a project retrospection allows for:

- Organisational learning from all perspectives of the project team
- Continuous improvement
- Better estimation and scheduling
- Team building and improvement of recognition and reflection to acknowledge accomplishments before tackling the next task.

A project can be looked at in terms of failed successes (process success + outcome failure) and successful failures (process failure + outcome success) (Nelson, 2005). It is also suggested that as part of the retrospection, evaluating project success should involve both process and outcome criteria. These consist of:

Process-related criteria:

- *Time* – did it meet the deadline?
- *Cost* – was it produced within budget?
- *Product* – were the requirements met?

Outcome-related criteria:

- *Use* – was the solution actually used?
- *Learning* – did the solution assist the organisation in preparing for the future?
- *Value* – did the solution improve efficiency or effectiveness of the client organisation?

With these criteria, a high-level success measure may be defined that indicates the areas that could have gone better and require improvement in the project and the areas that were successful. This is useful when attempting to identify potential issues a future project may encounter under similar circumstances and try to mitigate the risk of problems occurring at an early stage. This in turn can contribute significantly to a project being delivered successfully. Petter, Delone and McLean (2008) propose six key attributes of a successful information system:

- *System Quality* (ease of use, flexibility, reliability, ease of learning)
- *Information Quality* (system outputs e.g. relevance, understand-ability, accuracy, completeness and usability of information
- *Service Quality* (responsiveness, accuracy and technical competence from support staff)
- *System Use* (amount, frequency, extent and purpose of use)
- *User Satisfaction*
- *Net Benefits* (e.g. improved decision making, productivity and efficiency)

These attributes can also aid in identification of areas with potential problems from an organisational perspective, such as; information transfer, product functionality, requirements etc., which narrow down the list of required improvements to specific areas of the project lifecycle.

3. Intelligent Systems

Kapetanakis (2012) uses a case-based reasoning approach for the intelligent monitoring of business processes. Similarity measures, temporal logic and case-based reasoning were used with input from past data collected over several years, which was compiled into knowledge repositories that were used to evaluate the proposed approach as to its effectiveness in the monitoring and diagnosis of business processes. This yielded results that showed that case-based reasoning could be used effectively to monitor and diagnose business processes.

Athanasiadis (2004) proposes an agent-based intelligent environmental monitoring system to provide surveillance and on-line decision making for rapid environmental changes. This is combined with data mining techniques to define customised intelligence within the agents. The software agents were assigned a goal and made decisions based on inductive decision models created by data mining techniques and data-driven knowledge in order to highlight interesting patterns from data volumes. These patterns are then used to generate a predictive model made up of two parts; the predictor and the response. The predictive model is made up of decision trees, neural networks or association rules and can be translated into logical rules with the form: “If assumption then consequence” (Athanasiadis, 2004), which can be easily implemented and executed through a rule engine.

4. Risk Monitoring and Assessment

Several tasks had to be taken into consideration to give insight into how to solve the problem of automatically measuring the level of risk.

4.1 Data Gathering

In addition to the research done and knowledge obtained from common industry sources such as the PMBOK (Duncan, 1996), real world data was obtained through a questionnaire, which was sent out to collect responses from individuals in the IT industry. The questions on the questionnaire asked participants to:

1) Indicate the length of time they have worked in IT
2) Indicate the areas in IT they have worked in
3) Order risks in terms of impact on an IT project should they materialise
4) Give 3 risks they have encountered throughout their career
5) Suggest a mitigation for each risk
6) Indicate whether they agree with a suggested mitigation for a risk. If not then suggest a mitigation for the risk.
7) Indicate roughly how long the questionnaire took to complete
8) Any comments that the participant may have about the study or questionnaire

The responses to these questions facilitated the use of real-world data to input into the software solution and give a real-world perspective from a range of companies and years of experience in the IT industry.

4.2 Identifying Risk Indicators

Initial risk indicators for each distinct risk and mitigation proposed in the results from questions four and five of the questionnaire were identified. A risk indicator is a specific occurrence or lack of a project factor that can indicate that there may be increased risk for the project. Examples of the potential indicators are given in Table 1 below.

Table 1 - Examples of risk indicators

	Risk	Potential Risk Indicators
1	Risk that vague requirements are received	-Absence or length of requirements description -Rate of change of requirements description
2	Risk that the proposed design is not fit for purpose	-Number of defects raised -Project not signed off -Additional tasks added to project throughout -Tasks over running
3	Risk that a resource becomes unavailable	-Resource becomes unavailable -Resource is running at more than 100% capacity
4	Risk that the project team is not communicating effectively	-Tasks not regularly updated
5	Risk that stakeholder expectations are too great	-Additional tasks added to project throughout
6	Risk that additional functionality will be requested for the project part-way through	-Changes in estimates -Additional tasks added to project throughout -Tasks overrunning
7	Risk that testing may be blocked by the environment becoming unavailable	-Environment Resource becomes unavailable -Environment Resource is running at more than 100% capacity
8	Risk that a particular task will take longer than	-Rate of change of estimates of tasks over time

	Risk	Potential Risk Indicators
	expected to complete	-Tasks overrunning
9	Risk that a shared component may cause unnecessary delays to the project	-Environmental tasks overrunning -Task and resource dependencies introduced

4.3 Measures of Risk Likelihood

The indicators were then used to formulate a series of empirical measures to evaluate the potential risk at stages in the project, which were grouped into categories for ease of assessment.

a) *Template Rules* – These are based on the category that a task has, which outlines the stage of the project lifecycle the task fulfils. An example of a category would be testing, development, support etc. This type of rule can indicate that an area of the project lifecycle has been missed increasing risks of unclear requirements, changes of scope at a late stage or insufficient testing, support and training etc. An example of a template rule would be that a project must contain at least one analysis task to ensure that the project has been properly scoped out.

b) *Content Rules* – These are based on whether certain fields have content such as success criteria and description. This type of rule can indicate that requirements may be insufficient or that the scope is unclear for the task as no content indicates that this information has not yet been clarified. This may lead to scope creep or misunderstanding of requirements and ultimately the task may overrun. An example of content rule would be that a task must have success criteria to ensure that the expected outcome of the task is clear.

c) *Defect Count Rules* – These are based on how many defects are raised at various times in the project and also the type of defect. This type of rule can indicate things like; missed requirements, insufficient testing or a problem with a given environment etc. An example of a defect count rule would be the number of defects raised as a whole.

d) *Rate of Change Rules* – These are based on how often a given item is updated. This type of rule can indicate areas of uncertainty, such as vague requirements or scope creep on tasks. These rules are only evaluated if the actual start date of the project is in the past as a project manager may make many changes in the run up to starting the project as scope and requirements are defined. An example of a rate of change rule would be the rate of change in estimates for tasks, which can indicate that scope creep has occurred or other factors are causing delays to the task.

e) *Project Proportion Rules* – These are based on the number of development and testing tasks in comparison to the number of training tasks. This type of rule can indicate that end users may not receive appropriate training time to fully cover the functionality of the application. An example of a project proportion rule would be the proportion of development and testing tasks to the number of training tasks, which can lead to misunderstanding of the application and defects raised in error after go live.

f) *Resource Capacity Rules* – These are based on the status and the capacity of the resource for a given project. This type of rule can indicate whether there will be increased risk of resource allocation problems in the current state or at a later date. An example of a resource capacity rule would be to determine if a resource is over allocated on their capacity i.e. at more than 100%.

g) *Status Rules* – These are based on the status of the project and tasks as a whole. This type of rule can indicate that there is scope creep or vague requirements if a given task is not signed off in testing or a project is not signed off for go live due to issues. An example of a status rule would be if a project has not had signoff by the estimated end date.

h) *Timescale Rules* - These are based on the type of task and the estimated end dates. This type of rule can indicate that a given environment or task is having issues potentially due to other factors. This may also then cause delays on other aspects of the project. An example of a timescale rule would be tasks exceeding their estimated end dates.

i) *Priority Rules* - These are based on whether projects and tasks within a project have conflicting priorities. This type of rule can indicate that a dependent task or project may cause delays to other projects should it overrun. An example of a priority rule would be to assess contradicting project or task priorities.

j) *Dependency Rules* – These are based on the start and end dates of parent and child tasks that have a dependency on one finishing before the other can be started. This type of rule can indicate when delays to the dependant task may occur and highlight the need for the parent task to be managed. An example of a dependency rule would be identifying dependent tasks that are overrunning.

k) *Usage Rules* - These are based on how often users log in to the system. This type of rule can indicate that team members are not communicating effectively by updating or viewing tasks or that stakeholders are not buying in to the project. An example of a usage rule could be the login count for a user

Each of these identified rules is evaluated by a Rule Engine and if the rule is not met, then a risk measure is added to a list to be reviewed by the Risk Assessor, which alters the likelihood of risks based on whether a given combination of measures is present in the risk measures list.

4.4 Measures of Risk Impact

Defining the impact of a project risk project can be very difficult to quantify as it very much depends on the outside factors and contextual information of the company and the project as well as the people working on the project. In most cases the key impact is non-delivery of the project, and the low level impact of a risk is based mostly on 'gut-feel' from the project manager. Therefore, for the purposes of this research, it is anticipated that the Project Manger would update this manually.

The recommendations for the mitigations are mainly derived through analysis of the project ontology, which, as previously discussed, is based on the data collected in the questionnaire. This is analysed and the most likely mitigation is displayed first depending on the measure that was identified by the monitoring agent as the indicator for the increased risk.

4.5 The Monitoring Agent

The monitoring agent is responsible for identifying when an element has changed, reacting to the change by evaluating risk likelihood and informing the project manager of any increased risks. The following pseudo-code demonstrates the high level process the monitoring agent uses to achieve these tasks:

```
While true
    Monitor()
    If update then
        // Reassess risk
        React()
        foreach project in projects
            Assess()
            Update()
            Inform()
        End
    End if
End while
```

4.6 Software Solution

Many companies who develop software internally tend to have some form of team tracking that is readily available and on show. This tends to be through a website or the presence of TV screens that show the status of a project or company servers etc. The monitoring tool is intended to be displayed on a screen or viewed by individuals and consists of one main page that gives the status of all projects.

The status of the projects is displayed as both text and also as a three-point scale 'traffic light' system; Many Issues: RED, Some Issues: ORANGE, Completed, New or On Track: GREEN. This can be seen from a distance to highlight potential issues. The status is set based on how many of the total risks are above 50% likelihood that have a high impact or the number of risks with medium or low impact above 50% likelihood. This takes into account the fact that many low impact or medium impact items at increased risk can mean that the impact is actually higher if combined. Those projects show as completed are classed as in production but are being monitored for the number of live defects that are raised so still appear in the list of projects.

4.7 Test Cases

These measures were assessed both through the user interface and through structured unit tests that setup the project to mirror a specified scenario. The scenarios used were taken from part 6 of the questionnaire (section 4.1).

5. Results

The results table below shows the expected risks and the actual risks that were identified from the evaluation of the measures discussed in Section 4.3 against specific test scenarios and the intended coverage of the test.

Table 2 - The results of the test cases

	Most Common High Level Risks	Measures Used (Section 4.3)	Low level Risk(s) Identified by the Application	Pass (Y/N)
1	Risk that vague requirements are received	Template rules, Content rules	-Vague requirements are received -Unclear scope -Poor estimates -Unclear definition of success criteria	Y
2	Risk that the proposed design is not fit for purpose	Template rules, Timescale rules, Status rules	-Lack of advertising and training for users of the product -Inappropriate architecture and product design	Y
3	Risk that a resource becomes unavailable	Resource capacity rules	-Insufficient resource	Y
4	Risk that the project team is not communicating effectively	Rate of change rules, Project proportion rules	-Team is not communicating effectively	Y
5	Risk that stakeholder expectations are too great	Rate of change rules	-Insufficient or inadequate tools -Incomplete or vague requirements -Lack of resource training and support -Poor estimates -Poor documentation from industry bodies	N

	Most Common High Level Risks	Measures Used (Section 4.3)	Low level Risk(s) Identified by the Application	Pass (Y/N)
6	Risk that additional functionality will be requested for the project part-way through	Rate of change rules,	-Insufficient or inadequate tools -Incomplete or vague requirements -Unclear scope -Lack of resource training and support -Stakeholder expectations are too great -Poor product selection and implementation -Poor understanding of incumbent business processes and systems -Lack of forethought regarding application support documentation -Poor documentation from industry bodies -Live issues which crop up which need fixing before a planned release can go live -Untried methodology for product delivery	Y
7	Risk that testing may be blocked by the environment becoming unavailable	Resource capacity rules	-Insufficient resource	Y
8	Risk that a particular task will take longer than expected to complete	Task dependency rules, Rate of change rules	-Task dependencies are causing delays on the project -Differing project and task priorities between teams -Live issues which crop up which need fixing before a planned release can go live	Y
9	Risk that a shared component may cause unnecessary delays to the project	Defect count rules, Rate of change rules	-Insufficient or inadequate tools -Incomplete or vague requirements -Lack of resource training and support -Poor estimates -Poor documentation from industry bodies -Untried methodology for project delivery	Y
				8/9

6. Evaluation

The evaluation of the research was undertaken in two parts to assess both the test scenarios themselves and the measures of risk likelihood.

6.1 Evaluation of the Test Scenarios

The test cases present a series of standard scenarios that are common to IT projects. This allows the testing of the measures to be done from a real world perspective with real world data.

The majority of the test cases successfully identified the correct risk with 89% accuracy. However, several of the measures also identified other risks as potentially high likelihood. This is likely due to the initial identification of potential risk indicators as many of the indicators were replicated across multiple risks such as; additional tasks added, rate of change in estimates etc. This indicates that some of the risks have common causes and the measures can be seen as both successful and unsuccessful. With the measures successfully identifying the intended risk as well as other linked risks, this can be seen as an early indicator that potential risks may be of greater likelihood. However, this can also indicate that the evaluation of the measures are too general and need to be more complex.

With regard to the test that failed to identify the correct risk, this scenario is generally a very vague one that is difficult to identify even for a project manager. It is more a measure of opinion rather than one that can be quantified.

6.2 Evaluation of the Measures

The template rules allow the project manager to determine where the gaps may be in the plan with regard to the project lifecycle and the transfer of information as well as highlighting tasks that need to be done after go live, which are often forgotten or insufficient time is allocated to achieve them. Measuring for the lack of these tasks is a sound way of determining whether these stages have been considered and also whether these have been factored into the timescales and budget for the project.

The content rules focus on areas that may hinder the completion of a given task such as lack of success criteria, description and estimate. The lack of any of these may result in the task requirements being vague or non-existent, which can lead to scope creep on the task or failure for the users to sign off the project due to misunderstanding of requirements. These rules check for the presence of this information for each task in the project, which is again a sound measure of whether the resource would be able to achieve the task without these.

The defect count rules are also a good indication of potential project issues as these actively show areas of potential misunderstanding of requirements or lack of code quality. This criteria is typically used as a measure of quality of both the product and the requirements for real world projects when conducting a project reflection.

The rate of change rules are a new concept and can be assessed from the audit table in the database, which records every database action the user takes e.g. insert or update of a task. Many updates or few updates can be equally negative when conducting projects as this can indicate lack of communication among the team or tasks being constantly updated with no real structure. Many changes in estimates can indicate that there may be underlying factors that are invalidating the original estimates or that the task has unclear requirements and scope. Few changes can indicate that work is not being completed or that items are not being kept up to date.

The project proportion rules again indicate to the project manager areas that may have been assigned an insufficient amount of time. In this case, this may be very specific to the needs of the company or the project depending on what the end solution is intended to do and how much the end users of the solution already know about the processes and the application. This measure is therefore very much dependant on the specifics of the intended usage of the project.

The resource capacity rules are currently more effective due to the simple nature of evaluating the resource status but this can be seen as a very high-level viewpoint as it is often more complicated due to the need to efficiently allocate resource. This may take into account the level of the resource e.g. senior or junior developer or tester as well as their availability.

The status rules are a good comparator for determining whether the project is overrunning or whether a given element is meeting expectations. The timescale rules and dependency rules were simple and were too general to provide detailed information due to lack of stored information.

The measures were, in general a good indication of the level of risk of a given problem occurring, with the potential to be too generic as they have generally identified more potential risks than was intended. The measures could be enhanced to include more specific rules on particular risks, which may help to narrow down those risks that are of a higher likelihood. However, this could also be seen as a good way of identifying risks at an early stage but may also slow down the process of assessing the most relevant risks due to many risks having the same likelihood.

7. Conclusions

As a whole the research demonstrates that both commercial and academic concepts can be combined to aid in assessing and managing common problems across both areas. The combination of real world data and the use of industry standard tools allows the solution to be assessed from an industry perspective as well as from a research perspective with regard to the use of software agents and heuristic measures.

The software applications implemented form a useable prototype that could be adapted to use alongside existing project monitoring tools such as Microsoft Project, Team Foundation Server (TFS) or Jira, each of which has a different core function for different aspects of the project lifecycle within the field of IT and therefore demonstrates the diversity of the application and it's potential uses. It may have been more appropriate to use the TFS plugin from the start of the project but the data given from this may not have been structured in a very useable way as the databases used are known to be very complicated. Also the required information may not be easily interpreted into useable information for the measures to be effective.

The collection of real-world data has greatly enhanced the potential of the application and has highlighted the difficulty in developing algorithms and heuristics to solve everyday project problems. Although the application has performed well with simple scenarios, there is still the need to have human input alongside the application to provide a contextual analysis that is very difficult to assess through a software solution. As is evidenced by the diversity of answers to questions in the questionnaire, a great deal of variation exists across projects and processes in different companies and areas of IT. This variation highlights the need for human input and final decision making to be maintained in order to fully comprehend the scale of issues that may occur in a project. This is especially the case when assessing the impact of a risk, as this is highly dependent on outside variables such as, company process, environment setup, organisation of employees etc. Because of this, the application would be used as an aid to project management only, to facilitate making decisions faster and more efficiently by tracking the highlighted risks in an automated way and displaying the facts at a glance, giving clarity, openness and visibility to the whole project team.

Communication was identified by the industry professionals completing the questionnaire, to be a common problem with the potential to have a very high impact on the outcome of projects. This application can help to alleviate the issues of bad communication within the project team through the use of the website that can be accessed anywhere or displayed for everyone to clearly see. This also has the potential to aid in stakeholder management by applying the 'no surprises' approach to project management.

In conclusion, the research was a success and aided in addressing many of the common causes of project failure in IT. The use of such a generic approach to the general principles of project management means that there are many areas that could be enhanced and extended to improve both the measures of likelihood and also the functionality of the application from a commercial perspective. This concept could also be applied to non-IT projects or other areas where a tracking process would be advantageous.

8. Bibliography

Athanasiadis, IN., and P.A. Mitkas (2004). "An agent-based intelligent environmental monitoring system." *Management of Environmental Quality: An International Journal* 15.3: 238-249.

Belassi, W., and O.I. Tukel, (1996). "A new framework for determining critical success/failure factors in projects." *International journal of project management* 14.3: 141-151.

Duncan, William R. (1996)."A guide to the project management body of knowledge. (PMBOK)".

Kapetanakis, S., (2012). Kapetanakis, S. *Intelligent monitoring of business processes using case-based reasoning*. Diss. University of Greenwich, 2012.

Nelson, R. R. (2005). "Project retrospectives: Evaluating project success, failure, and everything in between." *MIS Quarterly Executive* 4.3 (2005): 361-372.

Petter, Delone, McLean (2008). "Measuring information systems success: models, dimensions, measures, and interrelationships." *European journal of information systems* 17.3 (2008): 236-263.