8. Conclusion

8.1 Summary

Software is becoming an important issue in our life because it is embedded in many systems, but it is inherently risky. Risks make project facing problems of over-running its schedule and cost or failure to satisfy the required quality. Software failures and risks are considered as failures and risks to the other systems, which involve the software. There is no way to absolutely avoid those risks, but rather to manage them. Software risks can be reduced or prevented if there is a proper risk management. Early risk management costs the project some extra budget, but it pays off at the end and avoids the project many crises.

Software risk management consists of effective techniques that identify, estimate, assess, monitor, and control software risks. Computer support for software risk management has been suggested as means of further increasing its efficiency and effectiveness. This has resulted in the development of many tools. Unfortunately, these tools suffer from some crucial shortcomings such as they do not cover well some aspects like risk documentation and monitoring, or they are developed for specific type of projects.

In this thesis, risk management support systems was investigated, and reviewed. Three major weaknesses were found:

- The first concerns risk documentation. Virtually all-existing tools do not support the risk documentation sufficiently.
- The second concerns the use of historical and statistical risk data which help in estimating current risks, predicting future risks, and determining the efficient
controlling techniques. It must be noted that the existing tools do not pay enough attention to these important aspects.

- The third concerns the ability of these tools. The existing tools emphasize only on certain aspects of software risk management (especially risk analysis). Other aspects are either ignored or not covered well (e.g. risk monitoring). Most of them are designed to fit specific type or size of projects. Furthermore, due to the higher cost of these tools they are purchased to handle only large-scale projects.

Proper risk management requirements have been identified in this research. Then a tool prototype has been built in order to fulfill those requirements.

8.2 Contributions

8.2.1 Research Contributions

Two major contributions to computer supported software risk management have been made. The first contribution is the proposition of less complexity, easy-to-use, efficient, and less time consuming technique. The fact that the proposed technique involves, many others, the "Risk Documentation Step" has considerably increased the efficiency of software risk management techniques. The second is the development of SoftRisk prototype tool around the aforementioned technique as a software risk management tool. SoftRisk has been developed in order to tackle the weaknesses of the existing software risk management tools. It may be the first prototype of its genre that handles software risk management steps by emphasizing on risk documentation.
Other contributions are listed below:

- Due to the difficulties in the risk estimation, a mechanism to estimate the risk probability and magnitude has been presented.

- Additional top seven software risk items together with their management techniques have been identified. Furthermore, some new formulas have been proposed in order to improve the sensitivity of risk assessment.

- This thesis presents a result of the first survey that has been conducted on software risk management practicing in Malaysia. Survey's result reflects many reasons and difficulties that the software firms are facing in the implementation of the software risk management. For instance, the cost of running risk management program lacks in required information and experience, and lacks in available tools. The result has been used in specifying the requirements of SoftRisk.

8.2.2 The Features of SoftRisk

The prototype has been developed in order to tackle some deficiencies that have been addressed by this research. All the components of SoftRisk have been tested and validated. According to the result of the validation operation which is carried by 15 Master, and PhD students, the SoftRisk has achieved and satisfied all proposed goals and requirements. The following are some of prototype features:

- SoftRisk has been designed and developed to be applicable to any type, any lifecycle phase, and any size of software development projects.
- **SoftRisk** prototype supports risk documentation step.

- **SoftRisk** uses a simple technique to manage software risks. It pays more attention to the developer's time and effort by adopting the Risk Exposure (RE) and top risk items techniques.

- **SoftRisk** includes a graphic builder to ease the risk-monitoring task.

- **SoftRisk** assists software developers to estimate the probabilities and magnitudes of software risks.

- **SoftRisk** uses quantitative data for the internal side (calculations) to get much accuracy and qualitative data for external side (users) to make it easy to handle.

- **SoftRisk** introduces risk reduction advice in terms of risk mitigation plan, contingency plan, or crisis plan. Moreover, it documents reduction actions, controlling results and risk situation after controlling any risk.

- **SoftRisk** prototype provides Risks' Database which is dotted with all important data and estimation checklists for the existing top ten risks and the proposed additional top seven risks, in order to help the developers in identifying and assessing those risks. Furthermore, any available resources (past projects, software surveys, developer's experience, current projects, publications, and Internet) can be used to feed this database.

- **SoftRisk** is a platform independent. It was developed using the JAVA language. However, Table 8.1 compares **SoftRisk** prototype with previous software risk management tools that have been compared in chapter 2. As can be seen in the table the **SoftRisk** prototype provides a comprehensive set of features to manage software risks.
Table 8.1: Comparison with Previous Tools

<table>
<thead>
<tr>
<th>Comparison Criteria</th>
<th>STRA</th>
<th>Risk-Database</th>
<th>METRIX</th>
<th>RRMOR</th>
<th>SoftRisk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to identify risks</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Ability to estimate risks</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Ability to assess risks</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Ability to monitor risks</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Ability to control risks</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Provides graphics support</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Provides Prob. / Mag. estimation assistant</td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provided with risks database</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Provided with risk advisor</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Not limited to specific projects, etc.</td>
<td>*</td>
<td>Limited</td>
<td>Limited</td>
<td></td>
<td>Preparation Only</td>
</tr>
</tbody>
</table>

8.3 Limitation and Future Perspective

Due to the scope of this research further effort must be carried out in order to improve the existing software risk management approaches. The limitation and future works will be discussed in the following sections.

8.3.1 Risk Statistics

Statistical analysis is very important to predict and avoid future risks. SoftRisk prepares the risks data to become ready for statistical operation. So far there is no efficient use of statistics in software risk management approaches. Another approach is recommended to use and invoke this system’s data in real statistical analysis.
8.3.2 Specialization

*Softrisk* has been supported with risks' database. Extending this prototype and making it as an expert system or knowledge-based is advisable to make risk management more flexible and simple. Each topic could have its risks and management strategies. As an example, specific parts cover software reuse risks, software pattern design risks, multimedia risks, coding risks, etc.

8.3.3 Integration

Software risk management topic should be embedded in all project management tools as an essential component. So software project managers, developers and decision-makers can use it concurrently with the project management tools. For instance, this prototype can be embedded in one of the existing software project management tools.

8.3.4 Risk Cost Estimation

How much does the risk cost the project? It is a very important question and needs to be addressed. The risk cost of risk management does not mean the magnitude of the risk itself. Amount of losses, and the management cost must be considered. This part has not been touched well and it is recommended as future work.