2014

A CMMI-based automated risk assessment framework

Morakot Choetkiertikul  
University of Wollongong, mc650@uowmail.edu.au

Hoa K. Dam  
University of Wollongong, hoa@uow.edu.au

Aditya K. Ghose  
University of Wollongong, aditya@uow.edu.au

Thanwadee T. Sunetnanta  
Mahidol University

Publication Details
A CMMI-based automated risk assessment framework

Abstract
Risk assessment is crucial to the increase of software development project success. Current risk assessment approaches provide only a rough guide. Risk assessment experts and domain experts are required in conducting risk assessments in software projects. Therefore, traditional risk assessment approaches require extra activities besides development tasks, and possibly leading to extra costs. We believe that an effective risk assessment approach should be transparently embedded in software development process. This paper aims to present an automated risk assessment framework using CMMI and risk taxonomy as a guidance to develop a risk assessment model. A pragmatic approach will be applied as a basis in building this suggested risk prediction model and the case studies of our practice. These studies are considered as our proof of concept.

Publication Details

This conference paper is available at Research Online: http://ro.uow.edu.au/eispapers/5448
A CMMI-based automated risk assessment framework

Authors: Morakot Choetkiertikul,
Hoa Khanh Dam,
Aditya Ghose
University of Wollongong, Australia

Thanwadee T. Sunetnanta
Mahidol University, Thailand

Presenter: Lei Tan
University of Newcastle, Australia
Introduction

• Risk Assessment – crucial to success
• Success rate was below 30% (1996-2004)
• Low quality of risk assessment leads to project failure
• Traditional risk assessment process
  – Risk identification
  – Risk analysis
  – Risk prioritization

• Current risk assessment
  – Only provides a rough guide
  – A lot of effort from experts
  – Involves extra activities which lead to extra costs
Aims

• An effective and practical risk assessment framework
• An automated assessment tool to avoid subjective judgments
• Using Capability Maturity Model Integration (CMMI) approach
Proposed Framework

• Assumptions
  – Risk is a probability of loss
  – Risk is related to the quality of software development process
  – Cost and effort could be minimized

• Step-by-step procedure

• Using data collected from a project tracking system
Conceptual Framework

- Two main phases
  - The learning phase
  - The deployment phase
The Learning Phase

- Step 1: Study existing risks
- Step 2: Identify risk taxonomy
- Step 3: Identify the best practice
- Step 4: Derive the risk identification rule
The Deployment Phase

• Step 5: Deploy the risk identification rule
• Step 6: Assess risks in current projects
Architecture
## Evaluation

### Case study 1

<table>
<thead>
<tr>
<th>Project Phase</th>
<th>Check Point</th>
<th>QOP (%)</th>
<th>No. of Risks</th>
<th>Risk Description</th>
<th>PA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>Checkpoint1</td>
<td>66</td>
<td>1</td>
<td>- Communication and coordination are inadequate.</td>
<td>IPM</td>
</tr>
<tr>
<td></td>
<td>Checkpoint2</td>
<td>77.3</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planning</td>
<td>Checkpoint1</td>
<td>18.81</td>
<td>3</td>
<td>- Lack of configuration management.</td>
<td>CM, RM, RD, PMC, PP, SAM</td>
</tr>
<tr>
<td></td>
<td>Checkpoint2</td>
<td>65.69</td>
<td>1</td>
<td>- System requirements are not clearly articulated to the team.</td>
<td>RM, RD</td>
</tr>
<tr>
<td></td>
<td>Checkpoint3</td>
<td>76.03</td>
<td>0</td>
<td>- Software development control process is inadequate.</td>
<td></td>
</tr>
<tr>
<td>Execution</td>
<td>Checkpoint1</td>
<td>61.79</td>
<td>1</td>
<td>- System requirements are not clearly articulated to the team.</td>
<td>RM, RD</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Checkpoint2</td>
<td>41.29</td>
<td>3</td>
<td>- System requirements are not clearly articulated to the team.</td>
<td>CM, RM, RD, IPM</td>
</tr>
<tr>
<td></td>
<td>Checkpoint3</td>
<td>87.83</td>
<td>0</td>
<td>- Communication and coordination are inadequate.</td>
<td></td>
</tr>
<tr>
<td>Closing</td>
<td>Checkpoint1</td>
<td>98.87</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Case study 2

<table>
<thead>
<tr>
<th>Project Phase</th>
<th>Check Point</th>
<th>QOP (%)</th>
<th>No. of Risks</th>
<th>Risk Description</th>
<th>PA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>Checkpoint1</td>
<td>74.32</td>
<td>1</td>
<td>- Software development methodology is inadequate</td>
<td>IPM, OPD, SAM</td>
</tr>
<tr>
<td></td>
<td>Checkpoint2</td>
<td>96.38</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planning</td>
<td>Checkpoint1</td>
<td>88.41</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execution</td>
<td>Checkpoint1</td>
<td>80.43</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitoring</td>
<td>Checkpoint1</td>
<td>65.01</td>
<td>1</td>
<td>- Software development control process is inadequate</td>
<td>PMC</td>
</tr>
<tr>
<td>Closing</td>
<td>Checkpoint1</td>
<td>85.67</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• Risk assessment process – within software development activities
• Better solution and contributions to project success
• Guidance for the development of auto-risk assessment
• Future works – open source projects