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

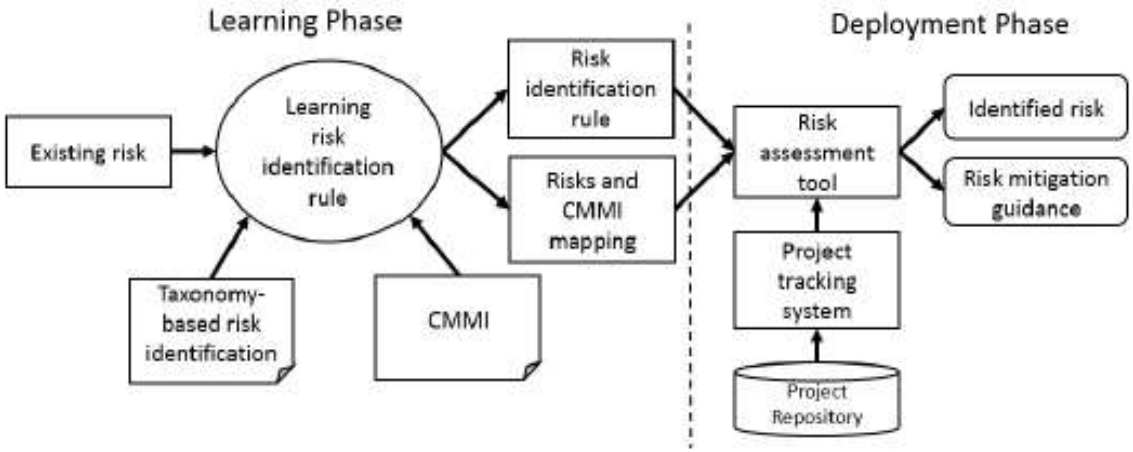
Aims

- 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

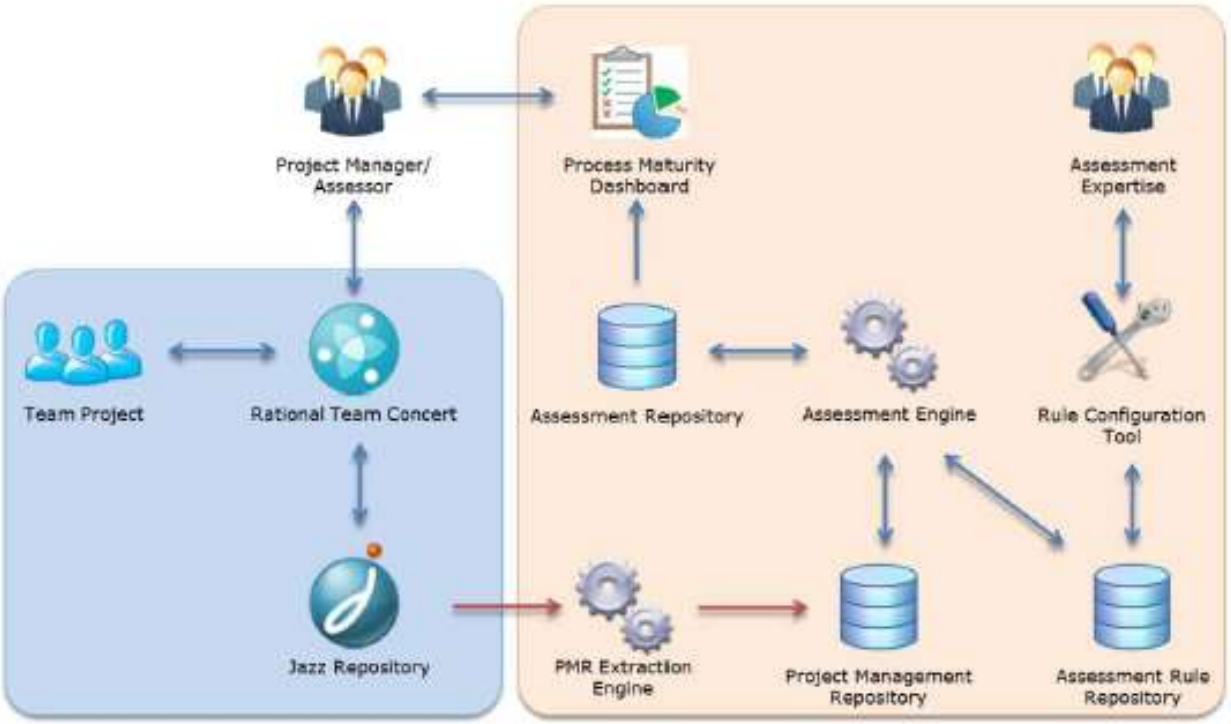
The Learning Phase

- 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

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

Case study 1

Project Phase	Check Point	QOP (%)	No. of Risks	Risk Description	PA
Initial	Checkpint1	74.32	1	- Software development methodology is inadequate	IPM, OPD, SAM
	Checkpint2	96.38	0		
Planning	Checkpint1	88.41	0		
Execution	Checkpint1	80.43	0		
Monitoring	Checkpint1	65.01	1	- Software development control process is inadequate	PMC
Closing	Checkpint1	85.67	0		

Case study 2

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