Software Engineering Risk Assessment

(STILL UNDER DEVELOPMENT: Last updated Aut/3/Thu by AW)

Changelog:

Starts on next page

Risk Details			Risk Score			Mitigation and Contingency			Action	Raised
Category	Name	No.	Likeli-hood	Impact	Score	Mitigation	Contingency	Score after		
Software Requirement Risks	Requirements being impossible to fulfil	1.1	1	3	3	Contact project sponsor and explain the circumstances	Attempt to fulfil requirements as much as possible.	3		15-Oct
	Change of requirements	1.2	3	3	9	Keep work and code well documented, making it easy to modify		7		15-Oct
	Requirements being poorly defined	1.3	2	2	4	Contact project client and request clarification.	Interpret requirements as a team as best possible.	2		15-Oct
	Not inspecting requirements correctly	1.4	2	3	6	When reading / receiving requirements, discuss and interpret as a team. Check understanding.	N/A	4		15-Oct
Risks with Implementation	Inadequate knowledge about tools	2.1	2	3	6	Team to work together to learn how to use the necessary tools	Use available tools to the best knowledge available / change tools	3		15-Oct
	Lack of skill	2.2	1	3	3	Ask project sponsor for help / team to collaborate	Complete project to brief to best standard possible.	2		15-Oct

Risks with Implementation	Difficulty of implementation	2.3	3	3	9	Team to collaboratively discuss problems with implementation and their solutions, seeking help where necessary.	Seek help from project sponsor and other resources	6		15-Oct
	Lack of tools	2.4	1	3	3	Speak to project sponsor / technical support to acquire necessary tools.	Build software using the only available tools as a last resort	2		15-Oct
	Tools failure	2.5	2	3	6	Use sponsor / technical support where possible	Use alternative tools if possible	6		15-Oct
Risks with Team Members	Human Error	3.1	2	3	6	Work completed by team must be checked by other members to avoid mistakes	Attempt to fix or isolate any errors made	4		15-Oct
	Disagreement between team members	3.2	3	2	6	Disagreements should be discussed and decisions made democratically.	Team chair to make the final decision to avoid arguments.	4		15-Oct
	Short-term loss of team member (illness)	3.3	3	3	9	All work to be shared with all team members, no one person should have exclusive knowledge / access to a resource	Redistribute work amongst other team members	7		15-Oct
	Long-term loss of team member (dropout etc)	3.4	2	4	8		Redistribute work, explain situation to project sponsor	6		15-Oct

Risks with Software Quality	Documentation is inadequate	4.1	1	3	3	Update documentation regularly to ensure software is easy to use and understand	Create as much documentation as possible when software is ready	4		15-Oct
	Lack of design documentation	4.2	2	4	8	Ensure documentation is kept updated and referred to throughout the project		6		15-Oct
	Product has errors due to a lack of testing	4.3	2	3	6	Throughout the project, create and document tests of all features, of the program on all relevant platforms	Perform as much testing as possible on project completion	5		15-Oct
	Lack of interest for final product	4.4	2	3	6	Ensure the “public face” of the product is kept updated and attractive to the end user.Maintain good software quality.	Gather any negative feedback and attempt to change the “public face” of the program	5		15-Oct