		Owner				Date			Risk Rating						
Ref No.	Keywords	MSc	MS	BL	Work Package	Open	Due	Closed	Likelihood	Time	Performance	Overall Rating	Description	Risk Level	Impact
R01		All							3	3	2	9	Team members designing the software are inexperienced.	Medium	Development cycles are extended.
R02		All							3	5	3	15	Weak team communication.	High	Project milestones missed and overall project delay.
R03		Stefan							3	4	5	15	Poor communication with the stakeholders.	High	Failure to capture user needs. Project failure. Stakeholder's dissatisfaction.
R04		Nektaria							3	4	4	12	Ineffective project plan - underestimate the workload of tasks.	High	Project delay which can evolve to project failure.
R05		All							4	5	5	20	Inability to complete critical milestones.	High	Project delay which can evolve to project failure.
R06		All							3	4	3	12	Incomplete milestone dependancies.	Medium	Failure to meet the agreed requirements. Missed deadlines and subsequently milestones delayed.
R07		All							5	5	5	25	Poor capture of user requirements - over ambitious, non-specific requirements	High	Failure to capture all the requirements. Project failure.
R08				James, Adam					3	4	3	12	Client changes requirements.	Medium	Rescheduling overall project workflow.
R09			All						3	3	3	9	Limitations of third party libraries, frameworks or tools.	Medium	Increased workload overhead. Project delay.
R10		Stelios							4	4	4	16	Weak implementation of testing plan.	High	Poor software quality which can lead to undesired and unexpected outputs.
R11		All							1	3	2	3	Ineffective code collaboration.	Medium	Incosistent code. Code duplication. Integration failures. Integration leaks. Code smell.
R12		Wendy							2	3	3	6	The users (mostly non-technical) find the prototype hard to use, reuse and learn.	Low	Dissatisfication of stakeholders.
R13		All	All	All					1	3	3	3	Unexpected events.	Low	Project delay which can evolve to project failure.

Mitigation	Contingency Plan	Notes & Updates
• Training. • Read resources, materials, relevant to the project scope and domain. • Train on tools and frameworks that are vital for the project development cycle.	• Read additional resources, materials, relevant to the project scope and domain. • Schedule urgent additional sessions on tools and frameworks that are vital for the project development cycle. • Get advice from domain experts and supervisor.	9/06/2014 Everyone agreed that it is best, we agreed to review…. 13/06/14 The work was carried out……..
• Effective meeting scheduling. • Exchange communication channels (e.g. e-mail, telephone etc.) and use accordingly. • Team bonding.	• Schedule team administration. • Get advice from supervisor. • Team bonding. • Team leader discusses the difficult person's problem one to one. • Clarify that all problems should be addressed to the team leader as a first point of contact.	
• Well structured preparation for meetings. • Schedule regular meetings with stakeholders. • Use visual aids.	• Escalate issue to project supervisors and project sponsor.	
• Thoroughly investigate similar project plans. • Review initial project plan with project supervisors.	• Review and amend the project plan accordingly. • Review the new project plan with project supervisors.	
• Effective project scheduling.	• Agreement on task prioritisation. • Commit more staff time. • Change in strategy: work may be divided into two groups in order to complete a key task. • Increase resources, give more time to ensure critical tasks are delivered on time.	
• Effective project scheduling. • Manage the critical path.	• Reschedule the project plan. • Redistribute the project resources.	
• Focus on capture of user requirements, at start of project. • Ensure user requirements are properly assessed. • The user requirements are fully investigated and agreed before specification. • Do not sign off on ambigous, ambitious and incorrect requirements.	• Immediately review the requirements with stakeholder. • Review the requirements with project supervisors and project sponsor.	
• Early prototype to identify reasonable expectations. • Use an iterative and incremental software development methodology.	• Escalate issue to project supervisors and project sponsor. • Reprioritise the tasks. • Reschedule the project plan.	
• Read resources and materials about tools, libraries and frameworks. • Guidance from technical experts.	• Support from technical experts.	
• Schedule a comprehensive testing plan. • Background research in testing methodologies. • Use automated testing tools. • Schedule testing plan for every iteration of the project cycle.	• Review the testing plan: schedule a comprehensive testing plan. • Read additional research materials on testing methodologies.	
• Use software configuration management tools. • Code convention.	• Use software configuration management tools. • Use high cohesion low coupling.	
• Use an iterative and incremental software development methodology. • Once a feature is fully functional, a prototype of the product is given out to the clients in order to evaluate it. • In the case that the clients are satisfied the feature is integrated into the existing system. • Otherwise, the feature is revised.	• Rebuild the system. • Improve some features of the system.	
• Use software configuration management tools. • Implement scheduled backups. • Code convention and consistency. • Assign milestones to both a primary and a secondary owner.	• Use software configuration management strategies. • In the case that a person is sick, redistribute project resources. • Communication with the team and supervisors in the case of an unexpected event. • Implement scheduled backups.	