Description	Likelihood (1-9)	Impact (1-9)	Importance (B*C)	Preventive actions	Remedial actions	comments
End of the world	0.000001	10	0.00001	Sacrifice goats	None, it won't happen before december, and we'll be done in may	to be removed later. ("The hell it is!" ~Stig Tore)
Lack of Room. No work space for us.	6	5	30	Book rooms in time and possibly work from home.	Find an alternative room on showing up at Gløs. The backup plan is to be at Drivhuset in the red room.	This has been a bit of a problem as you can't book a room for five weeks at a time Monday through Thursday between 1000 and 1600.
Illness	5	4	20	Eat healthy, sleep enough, be well clothed.	Stay home and sleep a lot to get better.	Probably not going to become a problem but, you never know.
Miscommunication with the customer	6	5	30	Confirm any and all changes or decissions about the project explicitly with the customer	Costly and time consuming changes to the project	
Conflicts within the group	3	5	15	Beer	More beer	
Broken codebase	1	8	8	Use Git (distributed as opposed to sentralised SVN). Keep one or more testing branches, which are merged with the Master branch only after having passed a full and rigourous test suite	Don't panic	Probably not going to occur, as we use Git and will at all times do development against a testing branch
Hardware failure	2	5	10	Maintain and take care of equipment, keep backups of important stuff	Acquire and set up new, or fix old, equipment as soon as possible	proper use of Git should minimize loss of work
Faulty planning	3	6	18	Make sure we properly research things before we decide anything	Improvise or bang head into the wall	
Failure to implement chosen technologies	3	9	27	Proper research into the technologies we are using and a proper understanding of those	Acquire knowledge and ask questions	This could be very bad if it should occur, but with proper research and a good understanding it should not be a problem
Optimistic scheduling			0			
incomplete schedule			0			
project to large in required effort or code size			0			
missing deadlines			0			
inability to work under pressure			0			
cascading delays			0			
Unfamiliarity with the coretech of the design			0			
Inefficient team structure			0			
Disruptive facilities			0			
Too coarse-grained requirements			0			
Additional requirements turn up			0			
Reimplementation due to faulty design			0			
Integration with external libraries more complicated than expected			0			
Unfamiliar software or hardware environments			0			
Team unfamiliar with the type of project			0			
Reliance on unfinished software			0			
Design is too simplistic			0			
Design is too complicated			0			
External libraries not suited for project			0			
Poor code-quality in external libraries			0			
Paperwork overhead is too big			0			
Schedule slips without being discovered			0			