Meeting 2 Summary:

- Brad's sketched a schedule for the project-plan, based on 6 week cycles each composed of three 2 week sprints drawing from Main course reading. Michael and Brad drew on academic and professional observations to validate it as suitable. We believe that 6 weeks between full progress reviews is frequent enough to be adaptive but not too frequent to be detrimental to progress. Localised progress reviews will happen every 2 weeks.
- We discussed scrum team compositions, and how many teams would be needed, arriving at the conclusion that it depends upon budget. Brad will create and analyse a systems workers model, to identify associated risks, and help gauge good budget options.
- Michael's created a list of Gherkin statements for the requirements. We noted the difficulty of identifying behaviour-driven specifications to describe hardware specifications. We need to add a table of requirements to the plan. We noted that the plan will look less well tailored if the table is too big, but large tables help to get around word limits, so some care is needed.
- A blank document will be created to add the unit 6 submission material too, guided by Tarek's template. Brad will do that.
- We discussed the importance of drawing from references where possible for justification. We note many academic disagreements, but it's better to have some model, than none, in order to gauge effectiveness of decisions.
- Michael will look at hardware/software configurations that would meet the requirements, to identify associated risks, and help gauge good budget options. We discussed the possibility of creating a lite version of the system that costs £250, and other better versions. Hardware takes some priority as software can be rolled out later (by floppy disc). The inhouse memory solution might be worth prioritising to cheapen costs.
- It would be helpful to identify all the specific tasks that will make up the project lifecycle, so we can identify task dependencies and task parallelism opportunities
- We discussed risk mitigations such as phased deployments and saw them as advantageous. It would be helpful to identify requirements that have multiple solutions differing by ease, for a versioning plan
- We discussed TOGAF as an enterprise architecture that could be compared to our plans structure
- We discussed the possibility of creating more than 2000 machines, so the project makes profit for building more machines