1 Team Organisation

1.1 Introduction

The team name is Team LEAD where LEAD stands for Linux Embedded Automotive Dashboard. The team consists of three members:

- Amal Kakaiya
- Simon Jouet
- Esiri Igbako

An original fourth member, left the team and course due to other project commitments.

1.2 Communication

Weekly meetings of the team, and meeting with project supervisors, will track the project progress against the team Gantt chart. These meetings will be used to assess deadlines, revise requirements, delegate tasks, and organise meetings with UGRacing (the client) should any clarification of requirements be needed.

1.3 Development and Tracking

Outside of formal meeting times, team members are expected to perform delegated tasks to deadlines and track progress through a ticketing system, Trac. This, along with the Subversion (SVN) versioning repository will ensure that the team has a well documented and transparent development process. Weekly time-sheets will also be filled in to match Trac tickets for further documentation. All team work, including code, documents, schematics, diagrams, data-sheets and sources are required to be added to the SVN repository throughout the development process. Not only is this a safe and secure means of keeping the data (providing the repository is backed up), it also allows all team members to work on up-to-date versions of all documents in the development process. Which allows concurrent development on code and documents.

2 Risk Management Plan

As part of the development process of the proposed system, the team has calculated the following potential risks, and probabilities for the risks becoming real, along with the forecasted consequences.

2.1 Risk 1: Component Compatibility Issues

Concerns have been raised about the fact that even though the components we have selected may conform to the same standards (for CAN), they might not work together as expected. Namely the Microchip and Freescale components.

Probability: Medium - High
Impact: Medium

Contingency actions: Select different components, and work on finding/deveoping Linux drivers for them.

2.2 Risk 2: Component failure

This is mostly concerning the main COM Gumstix component. If this fails, then the entire project is compromised as the device is expensive and takes a while to deliver.

Probability: Medium
Impact: High
Contingency actions: Multiple Gumstix (2 max) could be ordered just in case. However, this is expensive, so it will not be done. Other components can easily be ordered as they are fairly inexpensive and quick to obtain.

2.3 Risk 3: Team integration

Poor inter-team communication (with teams developing other components) means that we would be may be to interpret data generated by other teams. Since we are collecting data from all teams, for display on our dashboard, standards must be agreed and adhered to in order for the project to succeed.

Probability: Medium
Impact: High
Contingency actions: Agree standards with other teams and adapt code accordingly.

2.4 Risk 4: Personnel loss

In the event that a team member leaves the team and/or course, we must be able to prepare for the remaining workload.

Probability: Low
Impact: High
Contingency actions: Re-distribution of workload. Potential case for extension of deadlines. Revised Gantt chart required.

2.5 Risk 5: Misunderstood requirements

Our understanding of UGRacings requirements for the dashboard, may be different from their understanding.

Probability: Medium
Impact: Varying
To ensure that both sides have equal understanding, an iterative requirements gathering process can be used until an agreement is met.

3 Conclusion

The team has put a framework in place to allow for smooth development and delivery to the client. Risks have appropriate contingency plans, and the team should be able to deal with these forecasted risks should they become real. Unexpected risks will require further risk assessment and plan revision, depending on the severity of their impact.