Software engineering case studies: Solutions

W. H. Bell

January 26, 2024

1 Case study: Data acquisition software

- The hardware device drivers can be written and tested first. For example, software libraries can be written that connect to each device and read it or write to it. These can be tested with different input values and validated to perform correctly. Once these are complete, they should not be modified.
- The software should be developed with a bottom-up approach, going from the device drivers to the data format. The internal data format can be chosen to allow the storage of all data that are read from or written to the hardware devices.
- A user interface design should be sketched and discussed with the users, quickly determining how the software will be used.
- Additional user acceptance tests should be created to verify the functionality.

2 Case study: Embedded software

- The major risk is that the hardware selected might not have enough memory, connections or processing power to support the final software. If the software development commences with the wrong type of hardware, it will be time consuming to move the software to another microcontroller. This issue can be present due to the low-level functionality of the hardware being different between different microcontrollers. Low-level device drivers for one microcontroller are not the same as for another. Some abstraction could be implemented using interfaces, allowing functions that are not associated with hardware to be ported to another microcontroller. However, with memory limitations, this abstraction may be limited in its complexity.
- The development team should buy a microcontroller that has more connections than needed and more memory. Microcontrollers are sometimes available in a product family, such that a less powerful microcontroller could be used for the final product.
- Rather than purchase a new microcontroller at the start of the project, it might be possible to use an existing microcontroller development board that the company have and construct some

prototype code. This would allow scaling of the resources needed to the scale of final product and associated features. In particular, the amount of memory needed to buffer input signals or output communications should be considered.

3 Case study: Long-term development

Discussions with a customer are normally conducted by a project manager who is in or is associated with the development team.

- The major risk is that the customer has unrealistic expectations, concerning the timescale and cost.
- The project manager should schedule a meeting with the customer as quickly as possible to discuss issues found in the software.
- If the customer is unable to raise additional funds for the work, then it may be necessary to look for ways to leave the contract without continuing work for the customer.
- Before developing the software, the development team should establish how the existing software behaves. The existing software should be archived and copied into a software repository, such that any changes to it can be carefully tracked. It might be possible to use documentation of the calculations to verify that these are correctly implemented in the existing software.
- The project manager should provide the customer with a rough overview of the issues found and a cost estimate to understand the software before changes can be made to it. As part of this work, the development team could produce a developer guide for future developers to more quickly understand the software. It might be possible to generate some low-level design documentation from the software, depending on the computer programming language used.
- If the customer agrees to the work, the development team will have to carefully debug the application to verify that they understand its functionality. The development team should add code comments as needed and document their understanding of the software.
- Once the development team understands enough of the software, they should be able to go ahead and fix software defects, verifying that changes have not broken existing functionality.

4 Case study: agile development

Discussions with a customer are normally conducted by a project manager who is in or is associated with the development team.

- The development team could use all the money available for the project without delivering a working version of the software. The customer might not be aware of this direction until it is too late. While the customer may be responsible for driving the project in this direction, the software company will have failed to deliver. This may lead to a major loss of reputation and large settlement costs.

- An agile approach implies that it is harder to estimate the total cost of the project, since the requirements may change within the project. It would be easier to use the V-lifecycle, in which the user requirements are more constrained.
- Sometimes it is necessary to have hard conversations with customers or refuse to work for them. This will lead to a better outcome than a costly failure. The project manager should document the risks to the project and make these clear to the client. The project manager should provide working increments and keep the client up to date with spending.
- Communication throughout a project is key to its success. This includes responding to request for quotation (RFQ) documents with carefully written acceptance letters or alternative plans. Minutes of all meetings with the client should be recorded and sent to the client, such that they are made aware of risk as they emerge.