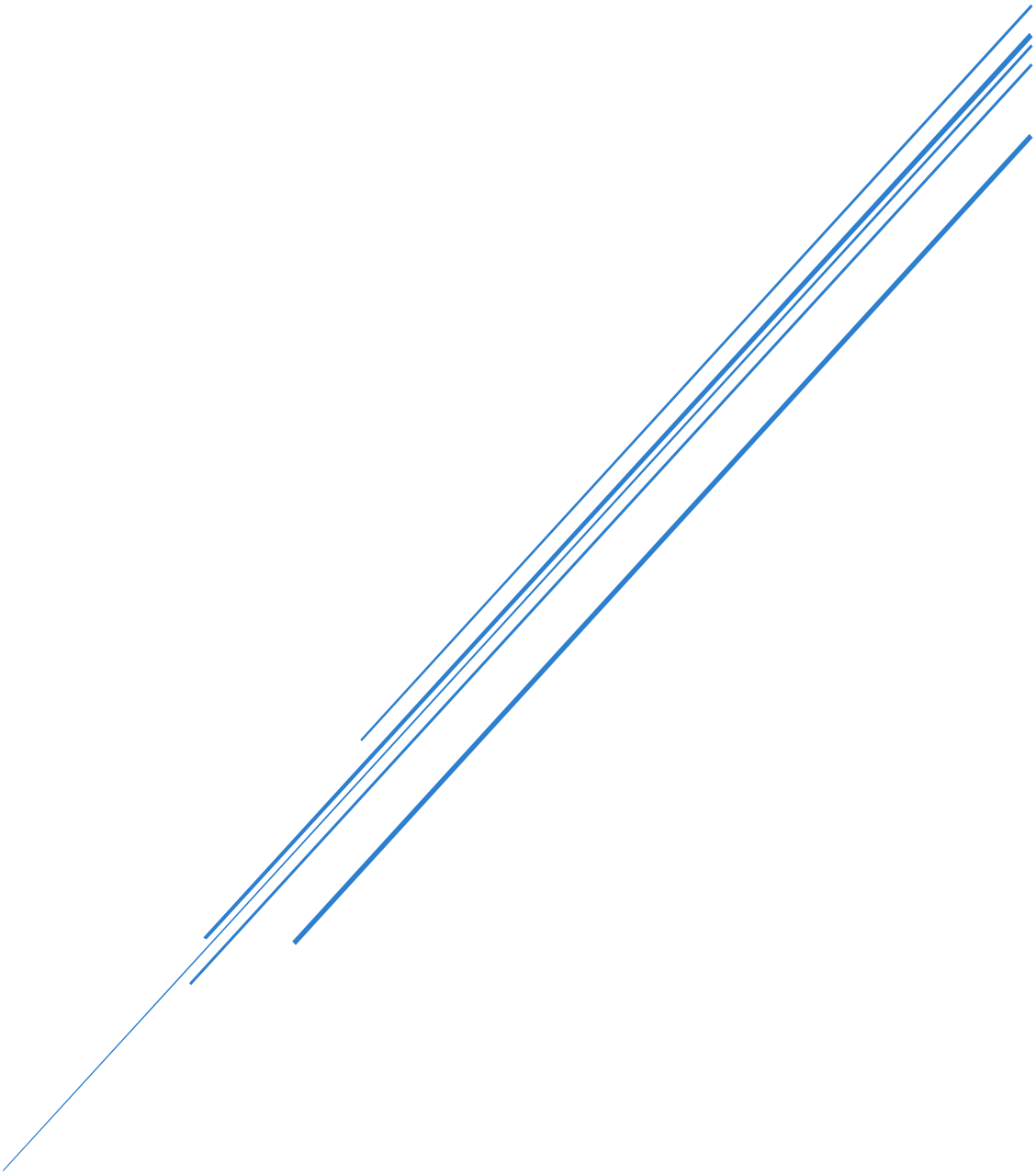
SYNPUTER

Team Project Report



Tim Brayshaw, Todd Edge, Karl Jackson, Ben Waters

Table of Contents

Development Methodology

This project involves the development of a complex system in which many different hardware and software components must align, increasing the risk of problems arising. This is a key consideration in the decision to use Scrum - an agile methodology - for the projects software development lifecycle (SDLC).

Whereas insufficient testing is widely regarded as a key reason for project failure (Lehtinen et al., 2014; Goatham, 2025), a Scrum methodology would allow for a testing stage during each iteration - or sprint - wherein each sprint goal will aim to add a feature or fix a problem (Heath, 2024). Such regular testing cycles allow developers to respond and make amendments in the next sprint where such changes are easier to make (Heath, 2024).

Scrum facilitates regular interactions with Will Burns (WB), allowing feedback and updates on progress. This would suit WB's fastidious nature and help to reassure him of both the project's alignments with requirements and its chance of success.

Requirements Analysis

Requirements Table

ID	Requirement	Comment	Priority
1	CPU with forward compatibility	CS: "Well my new Synputer – that's the working name", (Smiles), "will use a CPU from the Motorola 68k series." WB: "Ahh, the 68 thousand, very good. We have used the Motorola series in some of our Unix workstations and departmental servers." CS: "Yep – they also have forward compatibility – so any code written for them now will still work with subsequent chips." WB: "Yep – that's definitely a feature that we aim for – compatibility – sounds like your design could work for us…" WB: "Really? That sounds fascinating – but I always think that having industry standard compatibility is very important – especially in business where people need to expand on a regular basis."	SHOULD
2	Option for standard drive	CS: "Well, we could also add a standard drive – or make it a main expansion option?" WB: "Oh yes…"	SHOULD
3	An expandable machine	CS: "On the previous two Syn Computing machines we had an expansion option that allowed users to add both native and third party expansion packs. There is no reason not to continue that approach with the new machine." WB: "Oh yes, I agree. An expandable machine is essential – and your approach seems excellent. I mean, the technology changes so quickly nowadays and you do not want to lock users into an obsolete system, do you?"	MUST
4	A business suite of programs	WB: "So, what about the software? Any applications bundled? What about an OS and programming options?" CS: "Well… I have already started conversations with one of our established suppliers – they have agreed to create a business suite that we can bundle with the system. It will have all the basic tools required – word processor, spreadsheet, database, graphics – a complete office suite." WB: "That sounds excellent – in our experience many users would need little more than the suite you just described."	SHOULD
5	Industry compatible OS	WB: "That sounds marvellous, like the 3rd party Unix that is. As I said industry compatibility is key."	MUST
6	External keyboard port	WB: "The external keyboard is important in a business machine – many European countries have different keyboard layouts and it is much easier to just change an external keyboard than it is to have to modify an integrated keyboard…"	SHOULD
7	Network ports	WB: "Also, our future systems are destined to be integrated as part of a networked product line – so some form of networking would be essential."	MUST
8	Multiple serial ports	WB: "Multiple serial ports would be ideal – one could be dedicated for networking and the other used for additional connections."	SHOULD
9	Backwards compatible gaming	WB: "Excellent – yes, emulators for backward compatibility are something we use too. And being able to run them alongside business applications – first rate!"	SHOULD

Assumptions

Although discussed by Colin Syn (CS), the following specifications were not included as requirements by WB. The decision regarding their inclusion can be seen as somewhat flexible:

- RAM
- Form-factor
- Built-in screen

The following specifications were not mentioned by either CS or WB. This suggests that there is also flexibility regarding their inclusion:

- ROM size
- Sound chip
- User interface
- Motherboard
- Glue chips

Specification

Category	Actual Specification	Satisfies Requirement ID	Fails to Satisfy Requirement ID	Justification
CPU	Motorola 68k8	1	N/A	The 68008 has an 8-bit external bus, simplifying board design and lowering cost, while maintaining 68k compatibility.
RAM	512KB (4×128KB)	N/A	N/A	Reduced cost, simpler address decoding, and sufficient for command-line, basic GUI, and productivity software.
ROM	64KB (2 × 32KB)	N/A	N/A	Enough for core OS and BASIC. Modular ROM banks simplify updates and testing.
Storage	3.5" floppy only (1×720KB)	N/A	N/A	Flash storage adds cost and complexity. Using common floppy format simplifies boot/distribution and lowers cost.
Form Factor	Desktop with integrated keyboard	N/A	N/A	Portable would require advanced battery, display, and miniaturisation, going desktop makes development and assembly more manageable.
Glue chips	G1–G4	N/A	N/A	Hardware acceleration is expensive and complex. Using onboard graphics, G3.
Sound	i8042	N/A	N/A	Basic mono sound chip, simpler solution for cost/simplicity.
Expansion	16550 UART: 1 channel serial port Expansion board	2, 3, 7	8	Expansion board gives option for future upgrade to SCSI port
Ports	1 × SC150: 2 channel Joystick/Mouse/ Keyboard connector	6	N/A	Versatile port, allowing for multiple external devices, as well as 2 player gaming from one port (cable required).
OS	HB/OS (custom lightweight OS)	N/A	5	Unix-like OS is complex to implement on 68k with limited RAM. HB/OS is more manageable, tailored to hardware and BASIC.
UI	GUI likely minimal or CLI	N/A	N/A	A full GUI with WIMP requires hi-res display and more RAM. Reduced to command shell.
Programming	HyperBasic (extensible structured BASIC)	N/A	N/A	Retained, still central to the platform's identity. A structured, extensible BASIC makes the system approachable and programmable.
Applications	EZ-SUITE (licensed)	4	N/A	Still meets original goal. Licensing simplifies development.
Legacy Support	Not included	N/A	9	Emulator support adds significant complexity and expense. Dropped in favour of focusing on HB/OS and native software.
File System	Floppy-based, simple FS	N/A	N/A	Cross-platform FS adds complexity. Using standard floppy format is easier and practical.

Gherkin statements

Operating System and Boot

```
@usability
Scenario: The user wishes to use the Synputer's OS software
  Given that the user has powered on the Synputer
  When the OS attempts to boot
  Then BASIC will be loaded from the ROM
  And the user will see the CLI prompt on screen
```

```
@usability
Scenario: The user has read the included warning document and wishes to avoid a system
crash
  Given that the user has selected the boot-up option to run the HWCFG application
  When the user changes the run-time behaviour so that warning logs are written to RAM
disks
  Then the system avoids crashing, despite the lack of a drive B
```

Business Software

```
@usability
Scenario: The user wishes to see which business applications are available from the EZ-
SUITE software
  Given that the user has inserted the floppy disk containing the EZ-SUITE software
  When the user inputs LOAD EZ-SUITE
  Then the following menu <string> is available to the user
    | software     | word processor |
    | software     | spreadsheet    |
    | software     | database       |
    | software     | graphics       |
```

```
@usability
Scenario: The user wishes to use one of the business applications
  Given that the user has loaded the EZ-Suite software from the floppy disk
  When the user types LOAD <string>
  Then the <string> program loads
    | word processor |
    | spreadsheet    |
    | database       |
    | graphics       |
```

```
@usability
Scenario: The user wishes to run 2 EZ-Suite office applications simultaneously
  Given that the user is already running an EZ-Suite office suite application and wants
to open another
  When the user enters LOAD <string> for an application that he/she is not currently
running
  Then the second application loads without slowdown
    | word processor |
    | spreadsheet    |
    | database       |
    | graphics       |
```

Gaming

```
@usability
Scenario: The user wishes to play a Synputer game
  Given that the user has inserted a Synputer-compatible game floppy disk
  When the user types LOAD GAME
  Then the game loads
```

```
@usability
Scenario: The user wants to use external devices via the serial port
  Given that I am a user of the Synputer
  When I connect a <string> to the machine via the 16550 UART port connection
  Then it becomes fully operational
    | device | printer        |
    | device | monitor        |
    | device | network adapter |
    | device | hard disk drive |
```

Others

```
@usability
Scenario: The user wishes to use a program they have written on their current Synputer
on a new machine
  Given that the new machine has a CPU from the Motorola 68K series
  When the user loads their program onto the new machine
  Then the program will launch and run without additional errors
```

```
@usability
Scenario: The European user wishes to connect a standard or non-standard external
keyboard
  Given that the user has a compatible standard or non-standard external keyboard
  When the user plugs their standard or non-standard keyboard into the SC150 port
  And they insert a compatible floppy disk containing the keyboard's driver
  Then they are able to see an option to install the driver so that the keyboard
becomes fully operational
```

```
@usability
Scenario: The user wishes to use a standard or non-standard external keyboard
  Given that the user has installed the correct driver from a floppy disk
  When the user presses keys on the connected keyboard
  Then the corresponding characters are displayed on the screen
```

```
@usability
Scenario: Two users wish to use a joystick to be able to play a loaded multiplayer game
on the machine
  Given that a user has a compatible cable and two joysticks
  When the cable is plugged into the SC150 port
  And the joysticks are plugged into the two cable ports
  Then the OS recognises both joysticks and both become operational
```

```
@usability
Scenario: The user wishes to use two external devices
  Given that a user has a compatible cable and any combination of two <string>
  When the cable is plugged into the SC150 port
  And any combination of two <string> are plugged into the two cable ports
  Then the OS recognises both devices and both become operational
    | device | keyboard |
    | device | mouse    |
```

Costing

Hardware Specification

Component	Design Staff	Design Cost	Model	Spec	Unit Price £ (qty thousand)	Quantity per board	Component Design Cost (person weeks)
CPU2			68k8	5Mhz, 8/32, 1MB Max Ram	5.5	1	-
ULA1	HA	£5,000.00	G1	glue IOP-CPU	5	1	4
ULA2	HA	£5,000.00	G2	glue RAM-CPU	5	1	4
ULA3	HA	£5,000.00	G3	glue DISP-CPU	5	1	4
ULA4	HA	£5,000.00	G4	glue SYSTEM	5	1	4
ROM3	HA	£5,000.00	32K	32 KB ROM chip	4	2	4
RAM2			128Kb	8/16 bit, 100ns	2.5	4	-
IOP-J2			SC150	2ch Joy/mse/keybd connector	15	1	
IOP-S1			16550 UART	1 ch serial port	5	1	-
BOARD-SLDR	HA	£10,000.00	A83	CPU, IOP, G1-4, XXKb RAM SERPORT, INTSND	15	1	8
Storage1			disk	3.5" floppy	7.5	1	
CASE1	HA	£12,500.00	DESKTOP	int keyboard, 3 ext ports (+ exp)	25	1	10
KEYB1			int	int keyboard for case	5	1	
Pro Expansion			ProEx	CPU-Glue-SCSI - 4xRAM	15	1	
INTSND1			i8042	mono snd, 2 8-bit ports	1.5	1	-

Software Specification

Component	Design Staff	Design Cost	Component2	Producer	Design Cost (person weeks)
S1	SA	£3,000.00	Boot ldr & HWcfg	In House	2
S2	SA	£12,000.00	Sys: Kernel	In House/ HB/OS	8
S3			SYS: Libraries	In House/ HB/OS	
S4			SYS: Drivers	In House/ HB/OS	
S9	SA	£12,000.00	BAS: Kernel	In House/ HB OS	8
S10			BAS: core lib&I/O	In House/ HB OS	
S11	SA	£3,000.00	BAS: fs libs	In House/ HB OS	2
S38	SA	£6,000.00	CPM+BIOS	3rd party	4
S39			Libs & CLI	3rd party	
S40	SA	£6,000.00	68kBASIC	3rd party	4

Licenses

A BSD copy license is required at £500 per version of the design. Each machine is bundled with EZ-SUITE. A licence is required at a cost of £25 per machine.

Design Cost

The specification outlined above indicates a design cost of:

- Hardware - 38 weeks
- Software - 28 weeks

The company has one Hardware Architect (HA) (£250 per day) and one Software Architect (SA) (£300 per day). The time in the design phase can be reduced, using agency staff (HA - £400 per day; SA - £450 per day). Some of this time may be offset against the Project Manger's (PM) time, as this could reduce the overall timeline.

Production Cost

The production cost can be considered in 4 elements, per machine:

- Hardware components - £132.50
- Case production - £8.75*
- Board production - £7.00*
- 2 Software disks - £1.00
- Sale price - £349.99

*The case and board production can be conducted simultaneously as the company has two in-house Hardware Engineers (HE) (£175 per day). The cost per machine is dependent on a maximum build capacity of 20 cases and 25 boards per day. While the production cost can be reduced through agency staff offset against the Project Manager (PM) time, there can be issues with quality control, so agency staff are not used here.

Testing Cost

Hardware and software testing is conducted by the HE and Software Engineer (SE) respectively at a cost of £175 and £195 respectively when using internal staff.

Project Management Cost

The PM has a daily cost of £275 per day and will be employed daily throughout the project. While an additional PM can be employed from the agency, there is no financial benefit to do so.

Overall Costing

This model uses agency staff during the development phase to speed up hardware development and reduce the overall timeline.

Phase	Role	Units	Fixed Cost	Wks	Days	Agency Staff (Wks)	Agency Staff (Days)	Agency Cost	Internal Staff (Wks)	Internal Staff (Days)	Internal Cost	Total Cost
Hardware Design	HA	1		38	190	9	45	£18,000.00	29.0	145	£36,250.00	£54,250.00
Software Design	SA	1		28	140	0	0	£0.00	28.0	140	£42,000.00	£42,000.00
BSD copy license		1	£500.00									£500.00
Case Build	HE1	1000		0.01	0.05	0	0	£0.00	0.0	0.05	£8.75	£8,750.00
Case Build	HE2	1000		0.01	0.05	0	0	£0.00	0.0	0.05	£8.75	£8,750.00
Board Production	HE1	1000		0.008	0.04		0	£0.00	0.0	0.04	£7.00	£7,000.00
Board Production	HE2	1000		0.008	0.04	0	0	£0.00	0.0	0.04	£7.00	£7,000.00
Hardware Components		2000	£132.50									£265,000.00
EZ-Suite License		2000	£25.00									£50,000.00
Disk		2000	£1.00									£2,000.00
Testing - HW	HE1	1		1.0	5	0	0	£0.00	1.0	5	£875.00	£875.00
Testing - SW	SE1	1		1.0	5	0	0	£0.00	1.0	5	£975.00	£975.00
Project Management	PM	1		38.0	190	0	0	£0.00	38.0	190	£52,250.00	£52,250.00

The cost of delivering the project is £499,350.00, leaving a contingency of £650. This places the project at a financial risk, as shown in the OWASP risk register below (Williams, no date).

Financial Risk Register

Risk	Likelihood (L) 0-9	Rating	Impact (I) 0 - 9	Rating	Overall Risk OWASP Methodology
Software Errors	3	Low	6	Medium	Low
Hardware Errors (internal staff)	4	Medium	8	High	High
Hardware Errors (agency staff)	8	High	8	High	Critical
Project overrun	2	Low	8	High	Medium
Labour cost – increased agency due to staff absence or illness	6	Medium	9	High	High
Increase in component cost	3	Low	9	High	Medium
Import cost due to variation in exchange rate	1		4		4

The costing for the machine has involved changing the specification to meet the budget constraint while still delivering most of the requirements. The changes made have impacted the form factor of the machine and a lack of UNIX licence, however these could be included in future production with a revised price for the point of sale.

While the project is in budget, the small contingency means that any issues identified within the sprint tests or final testing would place the project at risk of being over budget. Similarly, the project would be at risk for any fluctuations in either component or labour costs. Finally, the project would be at risk regarding international exchange rates, specifically the GBP to US dollar, for components purchased outside of the UK.

Timings

Nascimento et al. (2022) analysed different sprint lengths in Agile software development. The research found that two-week sprints balance rapidity and risk management. The 38-week plan uses two-week sprints to manage complexity, reduce risk and align hardware/software development.

Sequential Phases: Design phases precede build phases. There are clear dependencies between integration of case, board and software.

Sprint Length: Two-week sprints give enough time for progress while allowing regular review and adjustment.

Testing Structure: The plan includes dedicated testing phases, distributed across the design and build lifecycle. It includes unit, integration, system and final user acceptance testing. These stages are scheduled to avoid last-minute failure and to allow early-issue detection.

Estimation Methods Used:

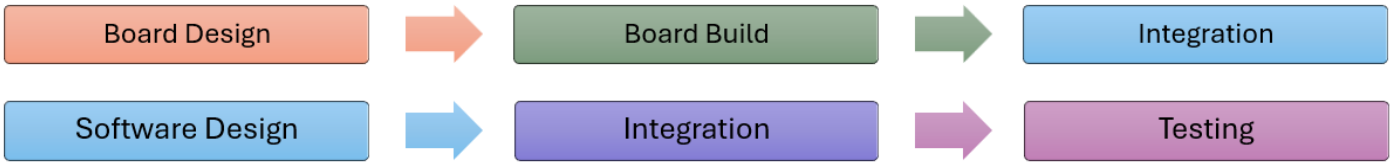
Expert Judgement: Applied to sprint durations, production timelines and batch estimates. Informed by standard engineering practice and available guidance.

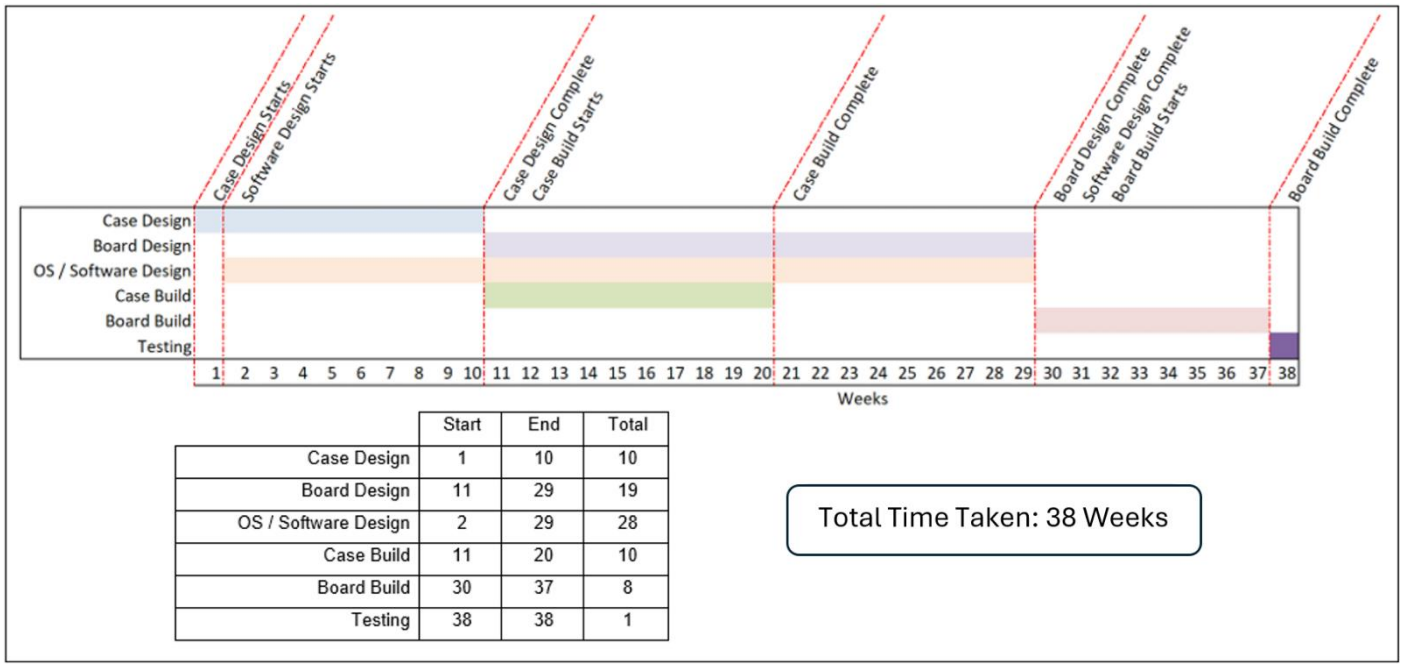
Bottom-Up Estimation: The case, board and software phases were divided into sprints with defined tasks. Duration estimates were made per task. Rivera et al. (2024) state that estimating total effort, by summing task efforts, is common and effective in Agile projects.

Production Timelines: Case and board production use batch estimates based on output per engineer.

Risk and Documentation: High-risk areas are addressed early. Separate documentation sprints ensure proper handover and traceability.

Gantt Chart





	Start	End	Total
Case Design	1	10	10
Board Design	11	29	19
OS / Software Design	2	29	28
Case Build	11	20	10
Board Build	30	37	8
Testing	38	38	1

Total Time Taken: 38 Weeks

Sprint Breakdown

Case Design - 5 Sprints			
Sprint	**Tasks**	**Description**	**Duration**
1	Requirements	Design constraints, dimensions, ports	2
2	Initial CAD and Prototype	3D modelling, component fit	2
3	Iteration & Review	Feedback from fit testing, structural changes	2
4	Finalisation	Lock design, prepare files for production	2
5	Testing and Documentation	Prepare spec docs, production review	2
		Total Time (in weeks)	**10**

Board Design - 10 Sprints			
Sprint	**Tasks**	**Description**	**Duration**
1	ROM & Board Layout Start	ROM chip integration, board planning	2
2-3	Glue Logic Integration	G1 - G4 chip layout and signal routing	4
4	SLDR A83 Board Design	Component mapping, logic checks	2
5-6	Electrical Testing	Prototype build, power testing	4
7	Interface Finalisation	CPU, RAM, I/O mapping	2
8-9	Layout Optimisation	EMI reduction, trace clean-up	4
10	Testing and Documentation	Full board validation, signal check	1
		Total Time (in weeks)	**19**

Software Design – 15 Sprints			
Sprint	**Tasks**	**Description**	**Duration**
1	Bootstrapping	Boot loader, HW config	2
2-5	OS Core Development	Sys kernel development	8
6-9	BASIC Environment	BAS kernel	8
10	File System Integration	File system libraries	2
11-12	Compatibility & BIOS	CPM support, BIOS integration	4
13	68k BASIC Development	Specific interpreter	2
14	Testing	OS and app integration, edge case tests	2
15	Documentation	API notes, build instructions	2
		Total Time (in weeks)	**28**

Case Production			
Phase	**Tasks**	**Description**	Duration
1	Case Production	Manufacture case unit in batches of 100 per week, per engineer.	10
		Total Time (in weeks)	**10**

Board Production			
Phase	**Task**	**Description**	**Duration**
1	Board Production	Board manufacturing and assembly in batches of 125 per week, per engineer.	8
		Total Time (in weeks)	**8**

Testing			
Phase	**Task**	**Description**	**Duration**
1	Testing	System integration and final validation testing	1
		Total Time (in weeks)	**1**

Project Summary

In summary, the challenge of planning the project to meet customer expectations has been challenging to navigate. The breakdown of requirements from the initial study has presented questions and a further need to define the requirements with more clarity. Embracing the Agile mindset has allowed the team to adopt the principles of the Agile Manifesto to combat this challenge.

The balance of project costs and timeframes has presented discussions on use of in-house staff or agency, again trying to align project deliverables and cost to meet the budget set to the team. One of the identifiable pitfalls of project failure has always been quality and in turn the testing process. However, again highlighting how numerous projects can lead to failure with the aim of meeting strict timeframes project delivery, it became clear that this phase was also seen as something that may need to be shortened to meet such deliverables.

References

Goatham, R. (2025) *Why Projects Fail*. Available at: https://calleam.com/WTPF/?page_id=2213 [Accessed 20 May 2025].

Heath, F. (2024) *The Scrum Master Guide: A Practical Guide to Successfully Practicing Scrum and Achieving Scrum Master Certifications*. 2nd ed. Birmingham: Packt Publishing, Limited. Available at: https://learning.oreilly.com/library/view/the-scrum-master/9781835885024/?sso_link=yes&sso_link_from=university-of-essex [Accessed 24 May 2025].

Lehtinen, T.O.A., Mäntylä, M.V., Vanhanen, J., Itkonen, J. and Lassenius, C. (2014) 'Perceived causes of software project failures – An analysis of their relationships', *Information and Software Technology*, 56(6), pp. 623–643. Available at: 10.1016/j.infsof.2014.01.015.

Nascimento, N., Santos, A., Sales, A. and Chanin, R. (2022) 'Is There an Optimal Sprint Length on Agile Software Development Projects?', in *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) – Volume 2*, pp. 98–105. SCITEPRESS – Science and Technology Publications. Available at: 10.5220/0011037700003179.

Rivera Ibarra, J.G., Borrego, G. and Palacio, R.R. (2024) 'Early estimation in agile software development projects: A systematic mapping study', *Informatics*, 11(4), p.81. Available at: 10.3390/informatics11040081.

Williams, J. (no date) *OWASP Risk Rating Methodology*. OWASP Foundation. Available at: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology [Accessed 8 June 2025].