

**Project Report:**

**System Design Proposal**

**Authors:**

Anna León, Aleksandr Vygodchikov, Ola Durowoju,

Jordel Davidson-Swann, Ojabo John Heart,

Emmanuel Koranteng, Ales Tekavcic 18.12.2023

**Table of Contents**

[1. Development Methodology 3](#_Toc153805371)

[2. Requirements 3](#_Toc153805372)

[3. Gherkin Language 4](#_Toc153805373)

[4. Project Costs and Estimates 4](#_Toc153805374)

[5. Project Timeframes Justification 5](#_Toc153805375)

[Appendix 1: System Requirements 8](#_Toc153805376)

[Appendix 2: Gantt Chart 10](#_Toc153805377)

[Appendix 3: Project Costs and Estimates 11](#_Toc153805378)

[Appendix 4: Gherkin Language 16](#_Toc153805379)

**List of Tables**

[Table 1: System Requirements 9](#_Toc153805492)

[Table 2: Project costs and estimates 14](#_Toc153805493)

[Table 3: Costs per role 15](#_Toc153805494)

[Table 4: Budget Cost 15](#_Toc153805495)

**List of Figures**

[Figure 1: Gantt Chart 10](#_Toc153805505)

# Development Methodology

SEPM has chosen to adopt a hybrid approach, applying the waterfall method to the production of hardware and the agile methodology to the software development process. The waterfall model ensures that there is a completed product before moving onto the next phase-gate, in essence, the waterfall methodology is an extremely linear process, requiring a much more defined structure to meet deadlines. However, the agile methodology is iterative, flexible, allowing for changes to be made at any time due to the nature of the planning and feedback cycle, this gives stakeholders insight into the development process.

Hardware production is time consuming and particularly expensive in relation to change, the waterfall method addresses these challenges along with, allowing for teams to work collaboratively and concurrently by following an explicit plan that is unlikely to change.

The software development process can be adapted at any time issues occur. The agile methodology directly addresses problems as testing is performed during the development process.

# Requirements

System requirements are a fundamental and initial part of planning a final product. Requirements are specifications of features that the final product should have, along with the goals to be achieved through its offered functionalities (Technology Research, n.d.). To meet the business needs within the desired timeframe and cost, multiple stakeholders are included, consequently enhancing the possibility of putting the project at risk. To avoid the letter, the client (EDC) and vendor (Synful Computing) agreed upon the requirements outlined in Appendix 1 (SEPM, 2023).

# Gherkin Language

Gherkin's structured format—Given, When, Then—provides clear, testable scenarios, ensuring that each feature meets specific user needs and technical requirements. Each Gherkin statement for this project is outlined in Appendix 4 and offers a comprehensive view of the system's capabilities. This approach aligns with best practices in software development for creating user-centred, functional specifications (Wynne et al., 2017).

# Project Costs and Estimates

The 13-month project plan (found in Appendix 3) outlines important stages like project initiation, hardware and software design, prototype, production, quality assurance, and sales launch. The expenditures are broken down, including agency fees for specialist outside services in addition to internal charges for positions such as project manager, hardware architect, and software architect. The proposal guarantees income creation through the sale of 2,000 units, and the budget of £500,000 has been determined. The program follows a stringent schedule, with the product being available for purchase by January 1984. All things considered, the strategy is effectively organized, adhering to financial limitations, and effectively allocating resources to ensure a prosperous product launch. From the calculations, it would cost Synful Computing:

* £229,150.00 in labour to produce the machines.
* £220,850.00 in components to produce the machines.
* £50,000.00 in any overhead cost that may exist.

The proceeds from the sale of 2,000 units, assuming a £300 per unit sale price, would total £600,000.

# Project Timeframes Justification

Gantt diagram for the project timeline can be found in Appendix 1 (Sharon and Dori, 2017). The project begins by outlining tasks and assigning responsibilities. The hardware development phase commences, defining specifications, designing architecture, creating CPU components, and finalizing the circuit board layout, concluding by April 1983.

The Hardware Section follows the Initiation phase, ensuring a smooth process. Specifications set the base for specific features, and architecture and design development occur concurrently to speed up the process. The phase concludes with layout and component selection, aligning with the established hardware plan. The longer CPU design duration is crucial for a strong foundation. This mix of sequential and parallel steps ensures organised and efficient progression (Wilson, 2015).

Software development starts with kernel development, establishing the core operating system. This allows parallel development of OS Shell, Utilities, and File System. With system software in place, Business Suite development begins, including word processors and graphics applications. Continuous Integration and testing operate in parallel, ensuring ongoing code quality. A phase of iterative refinement allows further enhancement, ensuring a robust and error-free software system (Berkling et al., 2009).

Prototyping follows, creating early versions for hardware functionality testing. Iterative refinement based on testing outcomes leads to a final round of prototype testing.

Quality assurance covers unit testing, integration testing, system testing, and user acceptance testing (UAT). The unit and integration testing stages can occur in parallel since unit testing focuses on individual components, while integration testing ensures they work harmoniously together. Once these individual and combined assessments are completed, system testing follows, examining the overall system functionality and interactions.

User acceptance testing (UAT) comes after these stages and is granted a more extended timeframe to carefully check everything, make possible changes, and fix any issues found. Taking more time recognizes how crucial it is to make sure the system meets what users expect. This way, it's a thorough process focused on users, and any mistakes or ways to make things better discovered during UAT can be fixed. It ensures the final product is improved and approved by users (Pandit and Tahiliani, 2015).

The manufacturing phase aligns strategically after quality assurance, preparing for mass production. Concurrently, the launch phase synchronizes with final production stages, releasing the product for sale and marking the project's conclusion. This parallel approach minimizes delays, optimizing the overall project timeline from testing to production and market release (Swift and Booker, 2013).

**References**

* AND Technology Research. (N.D.) *The importance of software requirements*. Available from: <https://andtr.com/the-importance-of-software-requirements> [Accessed 26 November 2023]
* Berkling, K. et al. (2009). ‘Timeline Prediction Framework for Iterative Software Engineering Projects with Changes’, in *Software Engineering Approaches for Offshore and Outsourced Development*. Springer Berlin Heidelberg, pp. 15–32.
* Pandit, P. & Tahiliani, S. (2015) ‘AgileUAT: A framework for user acceptance testing based on user stories and acceptance criteria’, *International Journal of Computers & Applications*, 120(10), pp. 16–21.
* SEPM. (2023) Project Management Case Study (based on real world events) [Lecturecast]. Software Engineering Project Management December 2023. University of Essex Online.
* Sharon, A. & Dori, D. (2017) ‘Model‐based project‐Product Lifecycle Management and Gantt chart models: A comparative study’, *Systems Engineering and Electronics*, 20(5), pp. 447–466.
* Swift, K.G. & Booker, J.D. (2013) *Manufacturing Process Selection Handbook*. Butterworth-Heinemann.
* Wilson, R. (2015) *Mastering Project Management Strategy and Processes: Proven Methods to Meet Organizational Goals*. Pearson Education.
* Wynne, M., Hellesoy, A. & Tooke, S. (2017) The Cucumber Book: Behaviour-Driven Development for Testers and Developers. 2nd ed. Raleigh: Pragmatic Bookshelf.

# Appendix 1: System Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Category** | **Requirement Details** | **Potential Missing Requirements** | **Assumptions** |
| **Hardware Specifications** | * Portable form factor * Weight not exceeding 2 kg * 4” low power screen * CPU: Motorola 68k series * RAM: At least 512Kb * Storage: solid-state medium-high capacity with expansion slots. * Battery life: At least 2 hours | * Specific dimensions for portability. * Detailed specifications for the 4” low-power screen. | * Standard industry dimensions for portability and reasonable specifications for the low-power screen. |
| **Software Components** | * In-house development of a Unix-like OS. * Development of HyperBasic (HB) programming language. * Business suite including word processor, spreadsheet, database, graphics. * Emulator | * Detailed functionality and features of the Unix-like OS. * Specific features of HyperBasic (HB) compared to TeleBasic (TB). | * In-house development is cost-effective and meets performance requirements. * External consultancy is not prioritised. |
| **Industry Standards and Compatibility** | * Compatibility with industry-standard peripheral ports and connectors. * Consideration of Networking Standards. * Compatibility with existing software, especially TeleBasic (TB). * Overall, the system has to be backward & forward compatible | * Detailed specifications for industry-standard ports. * Confirmation of compatibility with popular networking standards. * Compatibility testing with existing software. | * Industry-standard ports and networking standards are well-defined. |
| **Collaboration and Funding** | * EDC to purchase 2000 machines at a cost price of £250 each to augment Syn Computing’s capital challenge. | * Detailed terms of the collaboration. * Potential additional support from EDC. | * The agreed-upon collaboration terms are sufficient for Syn Computing's capital needs. * The estimated unit prices provided are accurate and will not change during the project. |
| **Risks and Contingency** | * Identification of potential risks such as market changes, technological advancements, or delays in production. | * Specific risks related to the electronics industry and the Synputer project. | * The identified risks are representative of potential challenges. |
| **Resource Allocation** | * Assessment of resource requirements for hardware and software development. | * Specific details on the expertise and resources EDC can provide. | * EDC can provide additional resources as needed for the project. * Synful Computing has the capacity to build the required components. |

Table 1: System Requirements

# Appendix 2: Gantt Chart

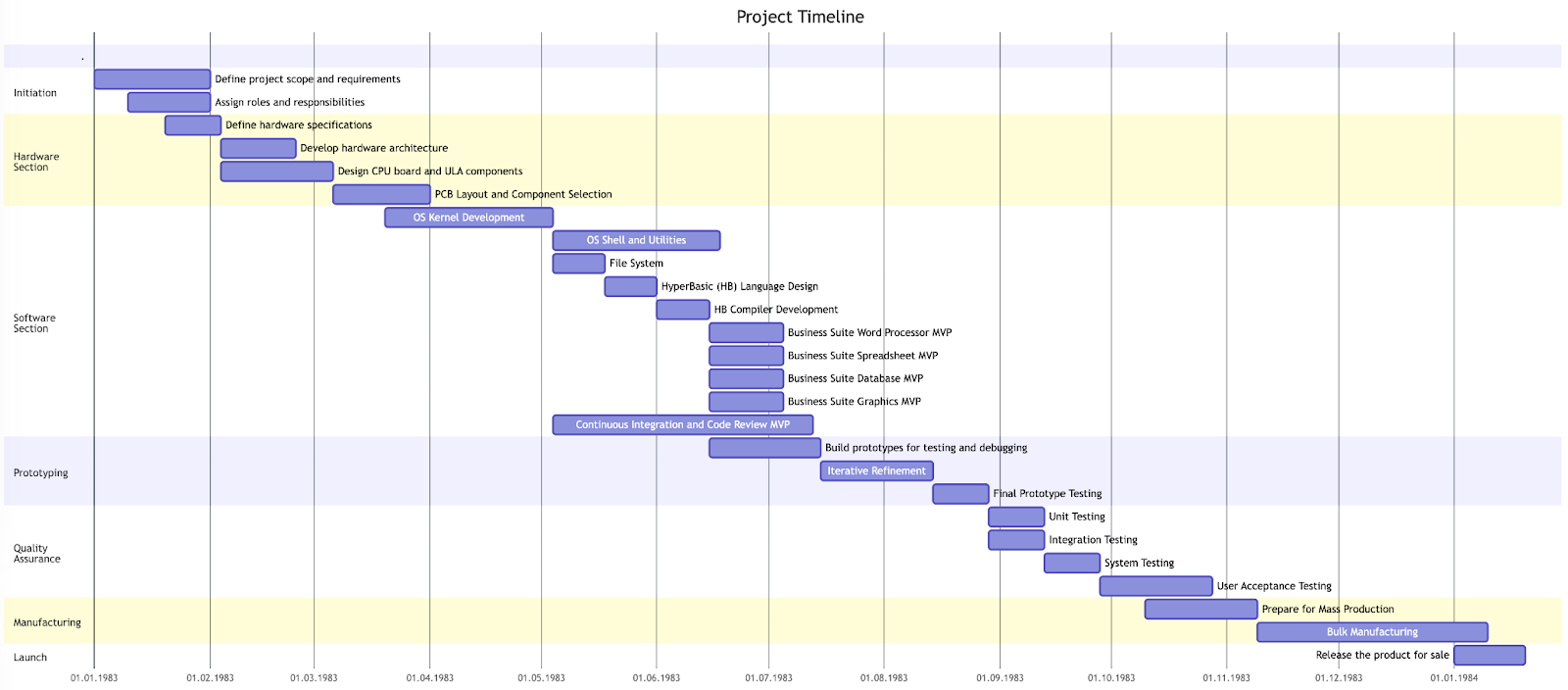


Figure 1: Gantt Chart

# Appendix 3: Project Costs and Estimates

1. Project Initiation (Month 1):
2. Define project scope and requirements.
3. Assign roles and responsibilities.
4. Internal Cost: £275 per day for Project Manager.
5. Time: 10 days.
6. Cost: £2,750.

1. Hardware Design (Months 1-3):
   1. Develop hardware architecture.
   2. Design CPU board and ULA components.
   3. Internal Cost: £250 per day for Hardware Architect and £175 per day for 2 HW Engineers.
   4. Agency Cost: £400 per day for Hardware Architect.
   5. Time: 60 days.
   6. Cost: £36,000 (Internal) + £24,000 (Agency) = £60,000.

1. Software Design and Development (Months 4-6):
2. Develop system software and applications.
3. Internal Cost: £300 per day for Software Architect and £195 per day for 2 SW Engineers.
4. Agency Cost: £450 per day for Software Architect.
5. Time: 60 days.
6. Cost: £41,400 (Internal) + £27,000 (Agency) = £68,400.

1. Prototyping and Testing (Months 7-9):
2. Build prototypes for testing and debugging.
3. Internal Cost: £175 per day for 2 HW Engineers and £195 per day for 2 SW Engineers.
4. Time: 45 days.
5. Cost: £33,300 (Internal).

1. Manufacturing and Assembly (Months 10-12):
2. Begin manufacturing of 2,000 units.
3. Assemble components.
4. Internal Cost: £175 per day for 2 HW Engineers and £195 per day for 2 SW Engineers.
5. Time: 60 days.
6. Cost: £44,400 (Internal).

1. Quality Assurance and Testing (Months 13):
2. Conduct thorough testing of manufactured units.
3. Internal Cost: £175 per day for 2 HW Engineers and £195 per day for 2 SW Engineers.
4. Time: 20 days.
5. Cost: £14,800 (Internal).
6. Launch and Initial Sales (Month 13):
7. Release the product for sale in January 1984.
8. Internal Cost: £275 per day for Project Manager.
9. Time: 20 days.
10. Cost: £5,500 (Internal).

**Tabular Display of project costs and estimates:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Phase** | **Role** | **Qty** | **Time(days)** | **Cost (£)** | **Total (£)** |
| Project Initiation (Month 1): | Project Manager | 1 | 10 | 275 | 2750 |
| Hardware Design (Months 1-3) | Hardware Architect | 1 | 60 | 250 | 15000 |
|  | HW Engineer (inhouse) | 2 | 60 | 175 | 21000 |
|  | Hardware Architect (Agency) | 1 | 60 | 400 | 24000 |
|  |  |  |  |  | 60000 |
| Software Design and Development (Months 4-6) | Software Architect | 1 | 60 | 300 | 18000 |
|  | SW Engineer | 2 | 60 | 195 | 23400 |
|  | Software Architect (Agency) | 1 | 60 | 450 | 27000 |
|  |  |  |  |  | 68400 |
| Prototyping and Testing (Months 7-9) | HW Engineer | 2 | 45 | 175 | 15750 |
|  | SW Engineer | 2 | 45 | 195 | 17550 |
|  |  |  |  |  | 33300 |
|  |  |  |  |  | 0 |
|  |  |  |  |  | 0 |
| Manufacturing and Assembly (Months 10-12) | HW Engineer | 2 | 60 | 175 | 21000 |
|  | SW Engineer | 2 | 60 | 195 | 23400 |
|  |  |  |  |  | 44400 |
| Quality Assurance and Testing (Months 13) | HW Engineer | 2 | 20 | 175 | 7000 |
|  | SW Engineer | 2 | 20 | 195 | 7800 |
|  |  |  |  |  | 14800 |
| Launch and Initial Sales (Month 13) | Project Manager | 1 | 20 | 275 | 5500 |
| TOTAL | | | | | 229, 150 |

Table 2: Project costs and estimates

Summary of costs per role:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **# of Employee** | **Cost/Day** | **Total Days** | **Total Cost** |
| Project Manager | 1 | £275.00 | 30 | £8,250.00 |
| Hardware Architect (Inhouse) | 1 | £250.00 | 60 | £15,000.00 |
| Hardware Architect (Agency) | 1 | £400.00 | 60 | £24,000.00 |
| HW Engineer (inhouse) | 2 | £175.00 | 185 | £64,750.00 |
| Software Architect | 1 | £300.00 | 60 | £18,000.00 |
| SW Engineer | 2 | £195.00 | 185 | £72,150.00 |
| Software Architect (Agency) | 1 | £450.00 | 60 | £27,000.00 |
| TOTAL | | | | £229,150.00 |

Table 3: Costs per role

Total Budgeted Cost:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Amount £ | Amount £ |
| Payment Received |  |  | 500,000.00 |
| Expenses: | Labour | 229,150.00 |  |
| Materials | 220,850.00 |  |
| Overhead | 50,000.00 |  |
|  | Total Expenses | 500,000.00 | (500,000.00) |
|  |  |  | 0.00 |

Table 4: Budget Cost

# Appendix 4: Gherkin Language

1. **Portable Form Factor**

Given a user requires a portable computer,

When they use the Synputer,

Then it should not exceed a weight of 2kg and should be easily transportable.

1. **Display Requirements**

Given a user needs a computer for basic computing,

When they operate the Synputer,

Then it should display content on a 4” low power screen.

1. **CPU Specification**

Given the need for efficient computing,

When the Synputer is operating,

Then it should utilise a Motorola 68k series CPU for optimal performance.

1. **RAM Capacity**

Given the requirement for multitasking capabilities,

When the Synputer is used for various applications,

Then it should have at least 512Kb of RAM.

1. **Storage Medium**

Given the need for storing data and programs,

When a user saves or accesses data on the Synputer,

Then it should use a high-capacity, solid-state storage medium with expansion slots.

1. **Battery Life**

Given the need for portability,

When a user operates the Synputer away from a power source,

Then it should run on battery for at least 2 hours.

1. **Unix-like OS Development**

Given the requirement for a robust operating system,

When the Synputer boots up,

Then it should load an in-house developed Unix-like OS.

1. **HyperBasic Language Development**

Given the need for a versatile programming language,

When a developer writes software for the Synputer,

Then they should use the HyperBasic programming language.

1. **Business Suite Software**

Given the necessity for basic office functions,

When a user operates the Synputer for business tasks,

Then it should provide a suite of applications including a word processor, spreadsheet, database, and graphics tools.

1. **Compatibility with TeleBasic**

Given the existing use of TeleBasic in the market,

When a user runs a TeleBasic program on the Synputer,

Then it should either support it natively or provide a seamless conversion to HyperBasic.