Summative Assessment 1: Software Engineering Project Management July 2024

TEAM 2: Andrius Busillas, Thomas Gray, Seyun Hur, Zukiswa Tuso

# Project Report

## Executive Summary

This proposal presents the design and specifications of the Synputer, a desktop computer developed to meet the needs of English Digital Computers (EDC). It outlines a comprehensive approach to the Synputer's development, including design methodology, core technical requirements, and a detailed analysis of projected costs and anticipated revenue, ensuring a robust solution for modern computing.

## 1. Design Approach/Methodology

### Selected Methodology: Agile Development

Choosing the right methodology is crucial for software and hardware development (Sommerville, 2016; Pressman, 2015). Three approaches were assessed: Agile, Waterfall, and Spiral.

| Methodology | Analysis |
|---|---|
| **Agile** | Flexible, iterative, and adaptable to change. |
| **Waterfall** | Structured, linear, lacks flexibility. |
| **Spiral** | Iterative with risk management, suitable for large projects. |

For the Synputer project, Agile was chosen for its flexibility, allowing continuous feedback incorporation and rapid adjustments. A comprehensive analysis of each methodology is provided in Appendix A.

## 2. Requirements Gathering

The requirements were meticulously gathered through an in-depth analysis, as provided in Appendix B and C, focusing on critical factors such as battery life, expandability, overall performance, etc., ensuring they meet diverse user needs and expectations.

### 2.1. Key Requirements

| Requirements | Details | Assumptions |
|---|---|---|
| CPU | Motorola 68k | Meets future performance needs |
| RAM | At least 512Kb | Sufficient for all applications |
| Portability | Portable/Luggable with a screen, under 2kg with 2 hours of runtime | Increasing demand for portability |
| Storage | Floppy disk or Cartridge | Aligned with demands for portability |
| Expandability | Upgradable system with ports | Increasing demand for expandability |
| Software | Unix-Like OS with HyperBasic support | Industry-standard OS is preferred |

## 3. Gherkin Specifications

Gherkin specifications listed in Appendix D demonstrate several parts of the Synputer case design, ensuring they fulfil the user needs and expectations and specify system behaviour clearly and structured as part of Behaviour-Driven Development (BDD)(Smart, 2014).

## 4. Costed Project Plan

### 4.1. Overview

The project is funded by EDC with an advance payment of £500,000 for 2,000 units of the Synputer. With strict constraints, this proposal will provide summaries of cost breakdowns, the key milestones, and the purposes of planned tests. Comprehensive information, including project plan (Gantt chart) and labour cost plan are provided in Appendix E.

### 4.2. Cost Breakdown

| Items | Cost (£) | Notes |
|---|---|---|
| Hardware Components | 200,000 | Includes core components, CPU, RAM, ROM, etc. |
| Software Components | 100,000 | Development of OS and applications |
| Labour Costs | 150,000 | Internal and Agency Labour Costs |
| Marketing and Sales | 30,000 | Advertising and promotional materials |
| Miscellaneous Expenses | 20,000 | Contingency for unforeseen costs |

### 4.3 Key Milestones

| Milestones | Target Date |
|---|---|
| Design Completion | November 1982 |
| Unit Test Completion | February 1983 |
| Integration Test Completion | April 1983 |
| System Test Completion | May 1983 |
| User Acceptance Test Completion | June 1983 |
| Prototype Ready | July 1983 |

### 4.4 Test Details

| Categories | Weeks | Notes |
|---|---|---|
| Unit Testing | 2 | To ensure functionality of parts of hardware and software |
| Integration Testing | 3 | To verify components are working seamlessly |
| System Testing | 3 | To validate the complete system against requirements |
| User Acceptance Testing | 4 | To gather feedback from end-users and to apply adjustments |

## 5. Risk Analysis

While it is not anticipated that there will be any major risks relating to the project, there is potential scope for these to arise. A number of factors relating to the project, such as human resource availability, financial changes and timelines can all be subject to delays, changes and potential unavailability.

As the project is being managed using Agile methodology, there is scope for reviews and amendments to be made at previous stages of the project to rectify these risks and changes. Any major changes to the financial impacts of the project or the timeline on deliverables would be completed in conjunction with the key partners and stakeholders of the project in a timely manner. This is a core aspect of effective software engineering project management, and ensures that there are no unexpected or unanticipated problems that could cause the working relationship or the end product to fracture or suffer.

## Conclusion

The Synputer project offers a well-rounded solution tailored to the request of English Digital Computers (EDC). Utilising Agile methodology ensures flexibility and continuous improvement throughout development. The design integrates key technical requirements like the Motorola 68k CPU and a Unix-like OS, providing performance and expandability. With a clear cost structure and defined milestones, the project shall meet its budget constraints while ensuring quality through testing. Ultimately, the Synputer will be positioned to deliver a modern, portable, and scalable computing solution.

**Reference list**


Biswas, T. et al. (2024) ‘ScrumSpiral: An improved hybrid software development model’, International Journal of Information Technology and Computer Science, 16(2), pp. 57–65. doi:10.5815/ijitcs.2024.02.05.

Cohn, M. (2004) User Stories Applied: For Agile Software Development. Boston: Addison-Wesley Professional.

Diansyah, A.F., Rahman, R.M., Handayani, R., Nur Cahyo, D.D. & Utami. E. (2023) ‘Comparative analysis of software development lifecycle methods in software development: A systematic literature review’, International Journal of Advances in Data and Information Systems, 4(2), pp. 97–106. doi:10.25008/ijadis.v4i2.1295.

Foley, J., McEwan, C. & Foster, T. (2024) What is Agile Software Development? Agile Alliance. Available at: https://www.agilealliance.org/agile101/ [Accessed: 25 August 2024].

GeeksforGeeks (2024) Comparison between agile model and other models in software engineering, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/software-engineering-comparison-between-agile-model-and-other-models/ [Accessed: 28 August 2024].

Guerrero-Ulloa, G., Rodríguez-Domínguez, C. & Hornos, M.J. (2023) ‘Agile methodologies applied to the development of internet of things (iot)-based systems: A Review’, Sensors, 23(2), p. 790. doi:10.3390/s23020790.

Ismail, R., Othmani, I.M., Said, I. & Dan, W. (2023) ‘The effective project management approach towards minimising environmental impacts in the construction industry’, Journal of Technology and Operations Management, 18(2), pp. 60–72. doi:10.32890/jtom2023.18.2.5.

Leong, J., May Yee, K., Baitsegi, O., Palanisamy, L. & Ramasamy, R. (2023) ‘Hybrid project management between Traditional Software Development Lifecycle and agile based product development for future sustainability’, Sustainability, 15(2), p. 1121. doi:10.3390/su15021121.

Malikov, M., Aloraini, F.A., Kavak, H., Kennedy, W.G. & Crooks, A. (2023) ‘Developing A Large-Scale Agent-Based Model Using The Spiral Software Development Process‘, 2023 Annual Modeling and Simulation Conference (ANNSIM), Hamilton, ON, Canada, pp. 282-293.

Mishra, A. & Alzoubi, Y.I. (2023) ‘Structured software development versus agile software development: A comparative analysis’, International Journal of System Assurance Engineering and Management, 14(4), pp. 1504–1522. doi:10.1007/s13198-023-01958-5.

Pressman, R.S. (2015) Software engineering: A practitioner’s approach. New York, NY: McGraw-Hill Education.

Rahman, A., Cysneiros, L.M. and Berry, D.M. (2024) ‘An empirical study of the impact of waterfall and agile methods on numbers of requirements-related defects’, Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing [Preprint]. doi:10.1145/3605098.3635901.

Saravanos, A. & Curinga, M.X. (2023) ‘Simulating the software development lifecycle: The waterfall model’, Applied System Innovation, 6(6), p. 108. doi:10.3390/asi6060108.

Smart, J.F. (2014) BDD in Action: Behaviour-Driven Development for the Whole Software Lifecycle. Shelter Island: Manning Publications.

Sommerville, I. (2016) Software engineering. Boston: Pearson.

Yu Stepanov, D. (2021) ‘Using waterfall, iterative and spiral models in ERP-system implementation projects under uncertainty’, Journal of Physics: Conference Series, 2142(1), p. 012016. doi:10.1088/1742-6596/2142/1/01201.

# Appendix A

## Analysis of Development Methodology

| Criteria | Agile | Waterfall | Spiral |
|---|---|---|---|
| **Definition** | An iterative and incremental approach that emphasizes flexibility and customer collaboration (Foley et al., 2024; GeeksforGeeks, 2024). | A linear and sequential approach where each phase must be completed before the next begins (Yu Stepanov, 2021). | A risk-driven model that combines iterative development with the systematic aspects of the waterfall model (Biswas et al., 2024; Malikov et al., 2023). |
| **Flexibility** | Highly flexible; changes can be made at any stage based on feedback (Foley et al., 2024). | Rigid; changes are difficult to implement once a phase is completed (Biswas et al., 2024; Rahman et al., 2024). | Moderate flexibility; allows for changes at the end of each iteration but is more structured than Agile (Biswas et al., 2024; Malikov et al., 2023). |
| **Customer Involvement** | Continuous customer involvement throughout the project lifecycle (Diansyah et al., 2023). | Limited customer involvement; feedback is typically gathered at the end of the project (Saravanos & Curinga, 2023). | Customer involvement is essential at the end of each iteration to assess risks and gather feedback (Yu Stepanov, 2021). |
| **Risk Management** | Risks are managed through regular iterations and feedback loops (Guerrero-Ulloa et al., 2023). | Risks are identified initially but only revisited at the end, which can lead to unforeseen issues (Rahman et al., 2024). | Emphasises risk assessment and management at each iteration, allowing for proactive adjustments (Biswas et al., 2024). |

| | | | |
|---|---|---|---|
| **Project Size Suitability** | Best suited for small to medium-sized projects with evolving requirements ((Diansyah et al., 2023; Leong et al., 2023). | Ideal for small projects with well-defined requirements and scope (Diansyah et al., 2023). | Suitable for large, complex projects where risk management is critical (Yu Stepanov, 2021). |
| **Documentation** | Minimal documentation; focuses on working software and collaboration (Mishra & Alzoubi, 2023). | Extensive documentation; each phase requires detailed documentation before moving on (Mishra & Alzoubi, 2023; (Yu Stepanov, 2021). | Balanced documentation; documentation is created as needed, focusing on risk management (GeeksforGeeks, 2024). |
| **Time to Market** | Faster time to market due to iterative releases and continuous feedback (Pressman, 2015). | Longer time to market as all phases must be completed sequentially (Saravanos & Curinga, 2023). | Moderate time to market; iterations allow for partial releases but require thorough risk assessments (GeeksforGeeks, 2024). |
| **Team Collaboration** | High collaboration among team members; encourages cross-functional teams (GeeksforGeeks, 2024). | Limited collaboration; team members work in silos based on their phase responsibilities (Saravanos & Curinga, 2023). | Encourages collaboration, especially during risk assessment phases, but can still be siloed in execution (Yu Stepanov, 2021). |

# Appendix B

## Requirements Analysis

| Category | Requirement | Description | Critique | Priority Level | Priority Justification |
|---|---|---|---|---|---|
| **Hardware Requirements** | **R001** Industry Standard Operating System: Support for an industry-standard OS. | The system must be compatible with widely used operating systems to ensure interoperability with existing software and hardware, facilitating user adoption. | **Assumption:** It assumes that users will prefer an industry-standard OS without considering niche markets that may require specialized systems.<br>**Gap:** Hyperbasic is a preferred option. | High | Essential for compatibility with existing software and systems, ensuring user acceptance and marketability. |
| | **R002** External Keyboard/Connect: It can connect an external keyboard. | The design should include ports or connectors that allow users to attach an external keyboard, enhancing usability and flexibility for various user preferences. | **Assumption:** It assumes all users will want an external keyboard.<br>**Gap:** The final device will then need to include an IOP component. | High | Critical for user experience and functionality; many users expect an external keyboard for productivity. |
| | **R003** Memory: Minimum of 512KB of RAM. | The system should have at least 512KB of RAM to support multitasking and run modern applications | **Assumption:** It assumes that 512KB is sufficient for all applications. | High | Necessary for running modern applications and multitasking; aligns with user |

| | | | | | |
|---|---|---|---|---|---|
| | | efficiently, ensuring a smooth user experience. | **Gap:** The final product still needs to consider upgrades, especially demands for larger RAM. | | expectations for performance. |
| | **R004** Removable Drive: At least one industry-standard removable drive. | Including a removable drive is essential for data transfer, backup, and software installation, adhering to industry standards. | **Assumption:** It assumes that users will require a removable drive.<br>**Gap:** While Floppy Disks are the industry standard in this era, Cartridge is expected to take over the market. | High | Important for data transfer and storage flexibility; users require the ability to easily manage data. |
| | **R005** Small Computer System Interface (SCSI) Expansion Capability: Support for SCSI for future expansion options. | The system should have SCSI ports to allow for the connection of additional storage devices and peripherals, providing scalability and flexibility for future upgrades. | **Assumption:** It assumes that SCSI will be a preferred standard.<br>**Gap:** Inclusion of SCSI will increase the cost of hardware units. | Medium | Provides future-proofing and scalability; while not immediately necessary, it enhances the system's longevity. |
| | **R006** CPU: Minimum of a 68000 CPU, with upgrade options. | The system must be powered by at least a 68000 CPU to ensure adequate processing power, with the potential for future CPU upgrades to enhance performance. | **Assumption:** It assumes that the 68000 CPU will meet future performance needs.<br>**Gap:** To allow CPU upgrade, a board with a socket should be considered as a primary option (BOARD-SCKT). | Medium | Important for performance; an upgradable CPU allows users to extend the life of their investment. |

| | | | | | |
|---|---|---|---|---|---|
| | **R007** Serial Ports: At least two serial ports supporting RS 422/485 standards. | The inclusion of two serial ports that comply with RS 422/485 standards is necessary for connecting various peripherals, such as printers and modems, ensuring compatibility. | **Assumption:** It assumes that users will need serial ports.<br>**Gap:** By taking SCSI as a component option, the choice of IOPs can be disregarded. | Medium | Relevant for business applications that require multiple connections; enhances the system's utility in professional environments. |
| | **R008** GUI Support: The Board is ready to support a GUI system and mouse if required. | The hardware should be designed to accommodate a graphical user interface (GUI) and support mouse input, aligning with user expectations for modern computing experiences. | **Assumption:** It assumes that all users will want a GUI.<br>**Gap:** No details on the specific GUI capabilities or requirements, it could lead to confusion about what is supported. | Medium | Increasingly expected by users; a GUI enhances usability and aligns with modern computing trends. |
| | **R009** Battery Life: Minimum of 2 hours of continuous operation under typical usage conditions, including moderate application use, screen brightness settings, and | The system should provide at least 2 hours of battery time to ensure portability and usability. | **Assumption:** The requirement presumes that users will engage in typical office tasks, which may not reflect the full range of potential use cases for the device.<br>**Gap:** Battery for luggagable cases may limit the level of performance of the device. | High | Essential for user satisfaction, especially for portable systems; users expect at least 2 hours of battery life for practical use. |

| | | | | | |
|---|---|---|---|---|---|
| | wireless connectivity. | | | | |
| **Software Requirements** | **R010** Bundled Applications: Complete office suite including word processor, spreadsheet, database, and graphics tools. | The system should come pre-installed with essential productivity applications to provide users with a comprehensive software environment for various tasks. | **Assumption:** It assumes that users will need a complete office suite.<br>**Gap:** Any mentions of the specific applications or their compatibility with existing file formats, could limit usability. | High | Critical for immediate usability; providing a complete office suite enhances the system's value and appeal to users. |
| | **R011** Multi-tasking OS: Development of a multi-tasking, Unix-like operating system. | The operating system should support multi-tasking capabilities, allowing users to run multiple applications simultaneously, enhancing productivity and user experience. | **Assumption:** It assumes that a Unix-like OS will be suitable for all users.<br>**Gap:** Since HyperBasic, the self-developed OS, must be included in Synputer, it may collide with consumers' needs. | High | Important for modern computing needs; it allows users to run multiple applications simultaneously, improving productivity. |
| | **R012** Programming Language: Inclusion of a structured, modular superset of BASIC (HyperBasic). | The system should support HyperBasic, a more advanced version of BASIC, enabling users to develop applications easily and efficiently, | **Assumption:** It assumes that users will want to program in HyperBasic.<br>**Gap:** No mention of support for other programming languages, which could limit the appeal to a broader audience. | Medium | While important for developers, it is less critical for end-users; a robust programming language can enhance the system's capabilities. |

| | | | | | |
|---|---|---|---|---|---|
| | | catering to beginners and experienced programmers. | | | |
| **Expansion and Compatibility Requirements** | **R013** Expandable System: Allow for the addition of native and third-party expansion packs. | The system architecture should be designed to support both native and third-party expansion packs, enabling users to enhance their systems with additional features and capabilities. | **Assumption:** It assumes that users will want to expand their systems.<br>**Gap:** No details on the types of expansion packs or how they will be integrated, which could lead to confusion about compatibility. | Medium | Provides future-proofing and adaptability; users appreciate the ability to upgrade and expand their systems as technology evolves. |
| | **R014** Documentation and Support: Comprehensive documentation covering basic operation and HyperBasic. | Detailed user manuals and technical documentation should be provided to assist users in understanding system operations, programming in HyperBasic, and troubleshooting issues. | **Assumption:** It assumes that users will read and utilize the documentation.<br>**Gap:** No mention of support channels (e.g., forums, customer service) for ongoing assistance, which could impact user satisfaction. | Medium | Necessary for user onboarding and troubleshooting; good documentation and support can significantly enhance user experience. |
| **Compliance and Performance Requirements** | **R015** Performance Specifications: Meet performance needs of >80% of user requirements. | The system must be designed to fulfil the majority of user requirements (over 80%) as identified in market | **Assumption:** It assumes that market research accurately reflects user needs.<br>**Gap:** No specific metrics or criteria for measuring performance against user | High | Directly impacts user experience and satisfaction; meeting performance expectations is crucial for |

| | | | | | |
|---|---|---|---|---|---|
| | | research, ensuring it meets the expectations of the target audience. | requirements, it could lead to subjective evaluations. | | the system's success in the market. |
| | **R016** Response to Complaints: Official response detailing how requirements will be addressed. | An official communication must be prepared to address any complaints or concerns raised by stakeholders, outlining how the system will be modified to meet their requirements and the timeline for implementation. | **Assumption:** It assumes that all complaints can be addressed satisfactorily.<br>**Gap:** No clear plan for prioritizing and addressing complaints, it could lead to unresolved issues and dissatisfaction among users. | High | Essential for maintaining customer relationships and trust; addressing complaints promptly can prevent potential legal issues and enhance reputation. |

# Appendix C

## Missing Requirements

| Category | Requirement | Description | Critique | Priority Level | Priority Justification |
|---|---|---|---|---|---|
| **User Interface** | **MR001** Detailed GUI Specifications | Define specific design elements, layout, and user interaction guidelines. | **Assumption:** It assumes that the development team understands user needs without explicit guidelines.<br>**Gap:** in clarity. | High | A well-defined GUI is critical for user satisfaction and usability, impacting overall product success. |
| **Performance** | **MR002** Specific Performance Benchmarks | Establish benchmarks for processing speed, memory usage, and response times. | **Assumption:** It assumes that performance will meet user expectations without defined metrics.<br>**Gap:** gap in measurable. | High | Clear performance benchmarks are essential to ensure the system meets user demands and industry standards. |
| **Security** | **MR003** Data Protection Features | Outline required security measures, including encryption and authentication. | **Assumption:** It assumes that basic security will be sufficient.<br>**Gap:** the gap in addressing potential vulnerabilities. | High | Security is paramount, especially for business applications; a lack of measures could lead to data breaches. |
| **Connectivity** | **RM004** Connectivity Options | Specify required ports and wireless capabilities for peripheral compatibility. | **Assumption:** It assumes standard connectivity will suffice. | Medium | Connectivity is important for user experience; however, it |

| | | | | | |
|---|---|---|---|---|---|
| | | | **Gap:** the gap in addressing specific user needs for connectivity. | | may not be as critical as performance or security. |
| **Customization** | **MR005** User Customization Options | Allow users to adjust settings for performance, display, and interface. | **Assumption:** It assumes users will be satisfied with default settings.<br>**Gap: the** gap in personalization needs. | Medium | Customization enhances user experience and satisfaction, but may not be essential for all users. |
| **Environmental Impact** | **MR006** Energy Efficiency Requirements | Define energy consumption limits and eco-friendly materials. | **Assumption:** It assumes users do not prioritize sustainability.<br>**Gap: the** gap in addressing growing environmental concerns. | Low | While important, it may not be a top priority compared to functionality and performance in the short term |
| **Documentation** | **MR007** Comprehensive User Documentation | Provide detailed user manuals and technical documentation. | **Assumption:** It assumes users will figure out the system without guidance.<br>**Gap:** the gap in support for less tech-savvy users. | Medium | Comprehensive documentation is important for user support and training, enhancing overall usability. |
| **Testing** | **MR008** Comprehensive Testing Strategy | Outline a strategy for unit, integration, and user acceptance testing. | **Assumption:** It assumes that existing testing processes will suffice.<br>**Gap:** the gap in ensuring thorough validation of the system. | High | A robust testing strategy is crucial to identify and resolve issues before launch, ensuring product quality. |

# Appendix D

## Gherkin Statements

```
Scenario: Operating System

    Given the system is powered on

    When the user checks the operating system

    Then it should display "Industry Standard OS"
```

```
Scenario: RAM Capacity

    Given the system is assembled

    When the user checks the RAM

    Then it should be at least 512KB
```

```
Scenario: CPU Upgradeability

    Given the system is powered on

    When the user checks the CPU model

    Then it should be a 68000 CPU and should be upgradeable
```

```
Scenario: Successful login with valid credentials

    Given I am on the login page

    When I enter a valid username and password

    And I click the "Login" button

    Then I should be redirected to my dashboard

    And I should see a welcome message
```

```
Scenario: Port accessibility
    Given the case is designed for user interaction
    When the user looks for ports
    Then they should find ports for a mouse and two joysticks
    easily accessible
```

```
Scenario: Cooling system
    Given the internal components generate heat
    When the case is designed
    Then it should include ventilation to prevent overheating
```

```
Scenario: Aesthetic appeal
    Given the market research indicates a preference for
    modern
    designs
    When the case is completed
    Then it should have a visually appealing design suitable
    for business and home users
```

```
Scenario: Documentation inclusion
    Given the case is finalised
    When the user receives their Synputer
    Then it should include a printed manual covering basic
   operation and an introduction to Hyperbasic
```

```
Scenario: External Keyboard Connection

    Given the system is on

    And an external keyboard is connected

    When the user types

    Then the system responds

    And the keyboard layout is configurable
```

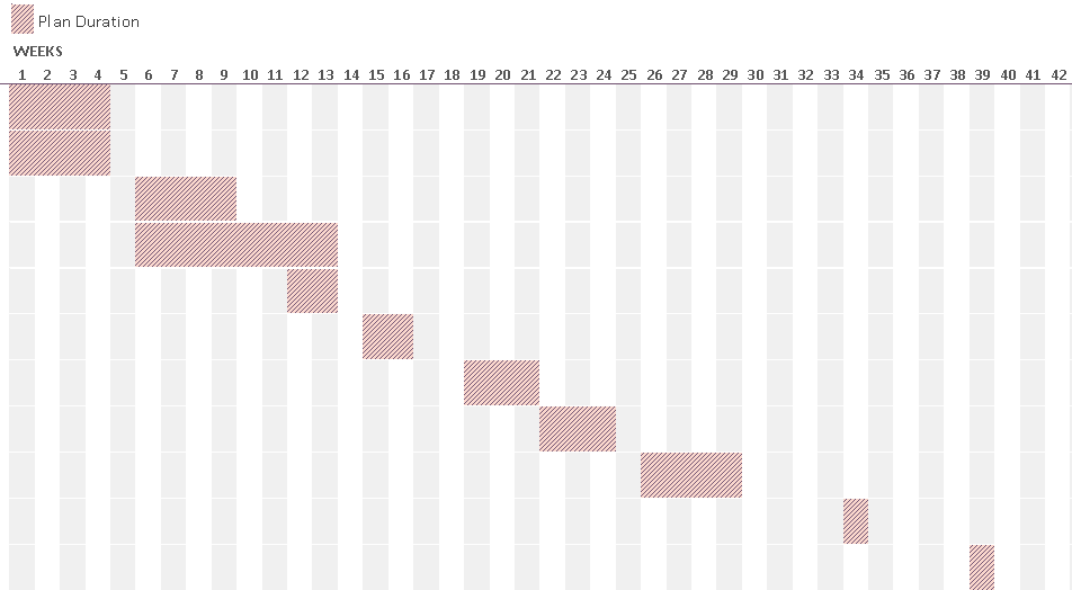
```
Scenario: Storage Compatibility

    Given the system is on

    When a removable drive is inserted

    Then it is recognized

    And allows read/write access
```

# Appendix E

## E1. Project Plan

# The Synputer Project

| ACTIVITY | START WEEK | DURATION (WEEKS) | START DATE | END DATE |
|---|---|---|---|---|
| Hardware Design | 1 | 4 | 1982-11-01 | 1982-11-28 |
| Software Design | 1 | 4 | 1982-11-01 | 1982-11-28 |
| Hardware Assembly | 6 | 4 | 1982-12-13 | 1983-01-09 |
| Software Development | 6 | 8 | 1982-12-13 | 1983-02-06 |
| Unit Testing (Hardware) | 12 | 2 | 1983-01-24 | 1983-02-06 |
| Unit Testing (Software) | 15 | 2 | 1983-02-14 | 1983-02-27 |
| Integration Testing | 19 | 3 | 1983-03-14 | 1983-04-03 |
| System Testing | 22 | 3 | 1983-04-04 | 1983-04-24 |
| User Acceptance Testing | 26 | 4 | 1983-05-02 | 1983-05-29 |
| Prototype Ready | 34 | 1 | 1983-06-27 | 1983-07-03 |
| Sales Ready | 39 | 1 | 1983-08-01 | 1983-08-07 |



## E2. Labour Cost Plan

| Role | Internal Cost (£) | Hired | Agency Cost (£) | Hired | Days | Total Cost (£) |
|---|---|---|---|---|---|---|
| Hardware Architect | £250.00 | 1 | £400.00 | 1 | 56 | £36,400.00 |
| Software Architect | £300.00 | 1 | £450.00 | 1 | 84 | £63,000.00 |
| Hardware Engineer | £175.00 | 2 | £275.00 | 1 | 35 | £21,875.00 |
| Software Engineer | £195.00 | 1 | £295.00 | 1 | 63 | £30,870.00 |
| Total Cost | | | | | | £152,145.00 |