Comprehensive Project Plan

Author: William Brian Richmond

Institution: Champlain College

Course: SDEV-590-857: Software Engineering and Project Management Capstone

Instructor: Nathan Braun

Date Due: 23 November 2025

Date Submitted: 22 November 2025

Introduction and Problem Statement

The growing complexity of modern computing systems highlights the limitations of traditional byte-oriented architectures when handling AI-driven workloads, high concurrency demands, and advanced optimization needs. Conventional CPU/GPU models process data at scale, but their underlying abstractions constrain how efficiently individual operations can be coordinated, scheduled, and reasoned about at fine granularity. As emerging computing environments require more deterministic, secure, and adaptive behavior, there is a need to explore alternative models that deliver bit-level transparency, predictability, and control.

This project proposes the development of a simulated secure operating system, BitGrid OS, built around a conceptual bit-level computing model. Rather than building hardware, the project focuses on designing a detailed architectural blueprint and a simulation environment that demonstrates how BitGrid OS could manage data, scheduling, optimization, and security at the bit level. The goal is to show that a bit-level architecture—supported by an intelligent supervisory layer (BitIQ) and a structured control framework (CSC)—could improve determinism, efficiency, and transparency within modern computing systems. This project plan outlines the work required to design, model, and document this conceptual system over the course of the seven-sprint Capstone timeline.

Project Scope and Objectives

Scope

The scope of this project centers on the design, simulation, and documentation of BitGrid OS as a conceptual operating system for bit-level computing. The project includes:

- Designing architecture diagrams representing BBPU-BitIQ-CSC interactions
- Developing a simulation model to demonstrate task flow and system behavior
- Creating a risk register and management plan
- Producing technical documentation, including user guides and workflow narratives
- Preparing presentation materials for stakeholders
- Integrating project management and software engineering principles throughout development

Exclusions:

To maintain feasibility within the Capstone timeframe, the project will not include hardware prototyping, driver development, fabrication processes, real-time performance benchmarking, or proprietary bit-level algorithms. All materials will remain conceptual and non-enabling.

SMART Objectives

The project will operate under the following objectives:

1. **Specific:** Develop a fully documented conceptual model of BitGrid OS, including architecture diagrams, simulation workflows, risk documentation, and presentation materials.

2. **Measurable:** Produce four primary deliverables—architecture documentation, project plan, risk register, and a final presentation package—that align with Capstone learning outcomes.
3. **Achievable:** Work will be organized into seven one-week sprints, ensuring a manageable scope.
4. **Relevant:** The project applies software engineering principles such as modular design, Agile planning, and DevSecOps practices.
5. **Time-bound:** All project work will be completed within the six-week Capstone timeline, with defined milestones for each sprint.

Methodology and Approach

The project will use a hybrid methodology that integrates Agile principles with software engineering best practices. Development will proceed through seven structured sprints, each focused on a specific component of the system. A Kanban workflow in Trello and Jira will support backlog refinement, sprint planning, WIP limits, and iterative updates.

Software Engineering Approach:

- **Modularity:** System components (BBPU, BitIQ, CSC) will be modeled independently to ensure clarity and ease of integration.
- **Architectural Modeling:** Lucidchart will be used to create layered diagrams representing system flow, trust boundaries, scheduling pathways, and AI interactions.

- **Documentation Standards:** All technical artifacts will adhere to IEEE and ISO documentation patterns to ensure traceability.

Project Management Approach:

- Agile–Kanban workflow
- Weekly sprints mapped to key milestones
- Use of Jira/Trello for task management and visualization
- Application of DevSecOps principles, including Zero-Trust Architecture
- Continuous integration of risk monitoring using the risk register

This combined approach ensures that the conceptual system is developed in a structured, iterative manner reflecting industry best practices.

Expected Outcomes

Upon completion of the Capstone project, the following outcomes are anticipated:

- **System Architecture Package:** A complete set of diagrams illustrating the conceptual structure of BitGrid OS and its bit-level workflow.
- **Simulation Model:** A functioning conceptual simulation of bit-level task scheduling and AI-assisted optimization.
- **Project Documentation:** User guides, logic descriptions, and a system-flow narrative supporting the conceptual implementation.
- **Risk Register:** A detailed, continuously updated risk management record aligned with the project's lifecycle.

- **Final Presentation Materials:** A slide deck and supporting narrative intended for stakeholders to understand the project's goals, methodology, and conceptual significance.

These expected outcomes will demonstrate the application of software engineering methodologies, project management techniques, and theoretical architectural modeling.

Project Timeline

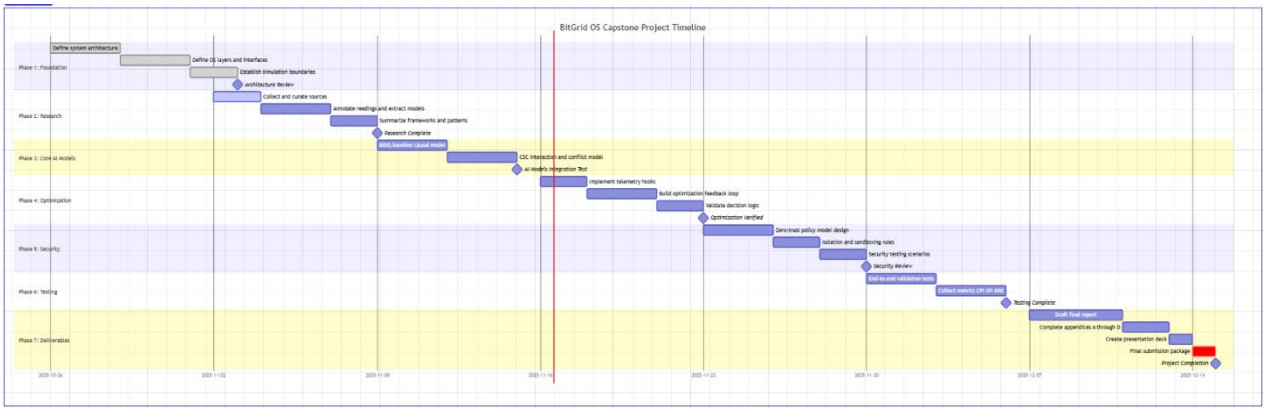
The project timeline follows the seven-sprint plan established in Week 1 and represented in the Gantt chart. Each sprint includes specific deliverables and milestones aligned with the project's objectives and scope.

Table 1. Project Phases and Timeline

Phase / Sprint	Dates	Key Activities	Primary Deliverables
Sprint 1 – Architecture & Scope	Oct 26 – Nov 1	Define architecture, OS layers, and simulation boundaries	Architecture outline
Sprint 2 – Research Foundation	Nov 2 – Nov 8	Collect and annotate research, summarize frameworks	Research synthesis
Sprint 3 – BitIQ & CSC Models	Nov 9 – Nov 15	Develop BitIQ logic and CSC interaction models	Logic models
Sprint 4 – Optimization Workflows	Nov 16 – Nov 22	Add telemetry and develop an optimization loop	Workflow prototype
Sprint 5 – DevSecOps Security Layer	Nov 23 – Nov 29	Apply Zero-Trust Architecture, develop test scenarios	Security model

Phase / Sprint	Dates	Key Activities	Primary Deliverables
Sprint 6 – Validation & Metrics	Nov 30 – Dec 6	Conduct validation tests and collect key metrics	Test results
Sprint 7 – Final Documentation	Dec 7 – Dec 13	Complete final report, appendices, and presentation	Final deliverables

Figure 1. BitGrid OS Project Timeline (Gantt Chart)



Resource Allocation Plan

Human Resources

- **Project Lead / Architect:** Responsible for system design, simulation development, documentation, and overall project coordination.
- **Project Analyst:** Performs research synthesis, risk analysis, and artifact preparation.
- **Technical Writer:** Prepares documentation, user guides, and presentation materials.

  (Note: All roles performed by a single developer for this project.)

Technical Resources

- **Tools:** Jira, Trello, MS 365 Business, GitHub, Lucidchart
- **Languages/Frameworks:** Python (simulation), SimPy, NumPy
- **Documentation:** APA style, IEEE/ISO documentation structures

External Resources

- Champlain College library databases
- Instructor feedback and peer review
- Online documentation for tool integration

Expected Outcomes

At the conclusion of the Capstone project, the following outcomes are expected:

1. **System Architecture Package**
    - Lucidchart diagrams showing subsystem interactions
    - Layered architecture views
    - BitIQ logic and system flow depictions
2. **Simulation Outline**
    - Conceptual bit-level scheduler
    - BBPU interaction patterns
    - Telemetry and optimization model
3. **Documentation Set**
    - Architecture descriptions
    - System flow narrative
    - BitIQ conceptual reasoning overview

- APA-formatted Capstone report

4. **Project Management Artifacts**
    - Completed risk register
    - Full timeline (Gantt or sprint chart)
    - Task-to-requirement trace matrix
5. **Stakeholder Presentation Materials**
    - 12–15 slide deck covering problem, architecture, and methodology

Success will be measured through:

- Completion of all six primary deliverables
- Alignment between requirements, tasks, and artifacts
- Instructor evaluation
- Coherence and clarity of architectural documentation

Challenges and Risks

Risk Category	Likelihood / Impact	Mitigation Strategy
Technical Complexity	High / Critical	Modular modeling, staged integration, simulation testing
Security Modeling	Medium / Critical	Zero-Trust Architecture, consistent reviews
Scope Overextension	Medium / Moderate	Strict scope boundaries, MoSCoW prioritization
Operational Constraints	High / Moderate	WIP limits, schedule buffers, and load balancing
Timeline Risk	Medium / Moderate	Time-boxed sprints, automated reporting

Risk monitoring uses Jira dashboards, severity scoring, and weekly retrospectives.

References

Atlassian. (2024). *Jira Software documentation.* https://www.atlassian.com/software/jira

Champlain College Online. (2025). *SDEV-590 Capstone materials.*

Comer, D. E. (2025). *Operating system design: The Xinu approach* (3rd ed.). Chapman and Hall/CRC.

Deloitte. (2023). *2023 Semiconductor transformation study.* Global Semiconductor Alliance.

IEEE Electronics Packaging Society. (2023). *Heterogeneous Integration Roadmap (HIR).* IEEE, SEMI, and ASME.

Microsoft. (2024). *Microsoft 365 Business tools overview.* Microsoft Corp.

National Institute of Standards and Technology. (2023). *NIST SP 800-53: Security and privacy controls for information systems and organizations.* U.S. Department of Commerce.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.