
PixelCrafters

Jobify
Software Development Plan (Small Project)
Version <1.6>

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

Revision History

Date	Version	Description	Author
<03/06/2025>	Draft	Initial draft from template	Nguyễn Lê Hồ Anh Khoa
<05/06/2025>	<1.0>	Add 4.1 Project Estimates. 4.3 Project Monitoring and Control.	Nguyễn Lê Hồ Anh Khoa
<07/06/2025>	<1.1>	Add 4.2 Project Plan.	Bùi Minh Duy
<07/06/2025>	<1.2>	Add 2. Project Overview. 3. Project Organization.	Nguyễn Thị Như Quỳnh
<08/06/2025>	<1.3>	Add 1. Introduction	Hình Diễm Xuân
<09/06/2025>	<1.4>	Update 4.2 Project plan (planning detail on Jira and correct date)	Nguyễn Lê Hồ Anh Khoa, Bùi Minh Duy
<10/06/2025>	<1.5>	Update 3. Project Organization.	Nguyễn Thị Như Quỳnh
<13/06/2025>	<1.6>	Read again and correct errors	All members of Group02_PixelCrafters

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

Table of Contents

1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Overview	4
2. Project Overview	4
2.1 Project Purpose, Scope, and Objectives	4
2.2 Assumptions and Constraints	5
2.3 Project Deliverables	6
3. Project Organization	7
3.1 Organizational Structure	7
3.2 Roles and Responsibilities	7
4. Management Process	8
4.1 Project Estimates	8
4.1.1 Estimated Cost	8
4.1.2 Estimated Schedule	8
4.1.3 Basis for Estimates	8
4.2 Project Plan	9
4.2.1 Phase Plan	9
4.2.2 Iteration Objectives	9
4.2.3 Releases	10
4.2.4 Project Schedule	10
4.2.5 Project Resourcing	12
4.3 Project Monitoring and Control	13
4.3.1 Requirements Management	13
4.3.2 Reporting and Measurement	13
4.3.3 Risk Management	13
4.3.4 Configuration Management	14

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

Software Development Plan (Small Project)

1. Introduction

1.1 Purpose

The purpose of the *Software Development Plan* is to outline the approach, processes, and resources that will be used throughout the development of the software project. The plan serves to define the project's scope, development methodology, key milestones, roles and responsibilities, risk management strategies, and supporting tools to ensure the project is delivered according to its objectives. It also assists project managers in effectively coordinating and monitoring the software development process.

The *Software Development Plan* is used by the following people:

- **Project manager** uses the plan to establish the project schedule, allocate resources, and check how the project is progressing.
- **Project team members** use it to understand their tasks, deadlines, and dependencies on other activities within the project.
- **Customers and Stakeholders:** refer to the plan to understand the project scope, key milestones, deliverables, and progress updates to ensure the project meets their requirements and expectations.

1.2 Scope

This *Software Development Plan* outlines a comprehensive plan for the Jobify project, including the features to be developed, both functional and non-functional requirements to be delivered. The document aims to ensure that all stakeholders have a clear understanding of the project's objectives and deliverables.

The planning and approach described in this document are guided by the product requirements defined in the *Vision Document*.

1.3 Overview

This *Software Development Plan* includes the following information:

Project Overview: describes the project's purpose, scope and objectives, assumptions and constraints, project deliverables.

Project Organization: Explains the organizational structure, roles and responsibilities of the project team.

Management Process: Covers project estimates, planning, monitoring, and control activities.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

Project Purpose: Jobify aims to enhance job matching and recruitment processes by providing a more streamlined and efficient platform. It builds upon traditional job board functionalities, focusing on improving interconnection between job seekers and employers through AI-powered recommendations and user-friendly interfaces.

Project Scope: The project scope is focused on the development of a comprehensive job matching platform by implementing standard recruitment functionalities including job posting, candidate management, application tracking, and content moderation systems.

Project Objectives: Develop a web-based platform with multi-role functionality (Candidates, HR, Moderators, Admins) and AI-powered job matching capabilities within the planned development timeline.

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

2.2 Assumptions and Constraints

Actor	Feature	Time	Expense
All actor	User authentication (login/logout)	8 hrs	200,000đ
	Password management (reset/forgot)	6 hrs	150,000đ
	Profile management	5 hrs	120,000đ
	Navigation menu	4 hrs	100,000đ
	Responsive design implementation	12 hrs	300,000đ
	Search functionality	6 hrs	150,000đ
	Notification system	8 hrs	200,000đ
Candidate	User registration & profile creation	8 hrs	200,000đ
	Browse job listings with filters	10 hrs	250,000đ
	Job application submission	8 hrs	200,000đ
	Save & bookmark jobs	4 hrs	100,000đ
	Application tracking dashboard	12 hrs	300,000đ
	Resume upload & management	6 hrs	150,000đ
	AI-powered job recommendations	15 hrs	400,00đ
	Job match suggestions interface	6 hrs	150,000đ
HR/Recruiter	Company registration	6 hrs	150,000đ
	HR dashboard	10 hrs	250,000đ
	Create/edit/manage job postings	12 hrs	300,000đ
	Review candidate applications	8 hrs	200,000đ
	Applicant status management	6 hrs	150,000đ
	Job performance analytics	8 hrs	200,000đ
	Email notification system	5 hrs	125,000đ
	Coin initialization on registration	4 hrs	100,000đ
	Coin deduction when posting job	4 hrs	100,000đ

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

	Show coin balance in the dashboard	4 hrs	100,000đ
Moderator	Content review interface	6 hrs	150,000đ
	Company account approval system	8 hrs	200,000đ
	Job posting moderation	5 hrs	125,000đ
	Flag inappropriate content	4 hrs	10,000đ
	User support tools	6 hrs	150,000đ
	Moderation dashboard	8 hrs	200,000đ
Admin	System oversight dashboard	10 hrs	250,000đ
	User role management	6 hrs	150,000đ
	Analytics dashboard	10 hrs	250,000đ
	Platform settings management	5 hrs	125,000đ
	Backup & recovery system	8 hrs	200,000đ
	Security & permissions management	10 hrs	250,000đ
	Database management tools	6 hrs	150,000đ
Total time/expense:		242 hrs	6,050,000đ

2.3 Project Deliverables

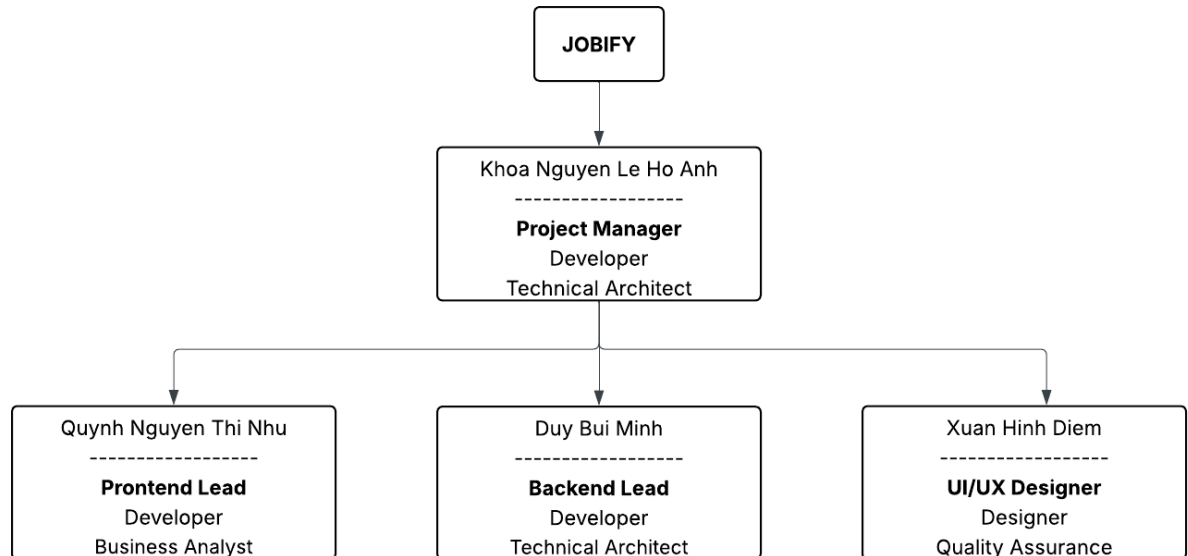
Deliverables of this project include a multi-page and multi-functional website and attached documents.

Details of the artifacts will be provided in section 4.2.1.

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

3. Project Organization

3.1 Organizational Structure



3.2 Roles and Responsibilities

Person	Role
Khoa Nguyen Le Ho Anh <ul style="list-style-type: none"> - Project Manager - Developer (Full-stack) - Technical Architect 	<ul style="list-style-type: none"> - Project Manager: build a project plan including things to do, technical issues, human resources, work schedule, risks,.. - Full-stack Developer: build and maintain both the front-end and the back-end of the website - Technical Architect: select tools and technical solutions before developing a product.
Quynh Nguyen Thi Nhu <ul style="list-style-type: none"> - Frontend Lead - Developer (Front-end) - Business Analyst 	<ul style="list-style-type: none"> - Frontend Lead: lead the frontend development team and coordinate with the backend team - Developer (Front-end): develop user interface components, dashboards, job listing pages using React.js and Next.js - Business Analyst: elicit, analyze, and document business requirements, translating them into clear specifications for technical teams.
Duy Bui Minh <ul style="list-style-type: none"> - Backend Lead - Developer (Back-end) - Technical Architect 	<ul style="list-style-type: none"> - Backend Lead: lead backend development and database design - Developer (Back-end): develop and test server components, APIs, and database operations using Node.js and Express.js - Technical Architect: design system architecture, database schema, and API structure

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

Xuan Hinh Diem <ul style="list-style-type: none"> - UI/UX Designer - Developer (Front-end) - Quality Assurance 	<ul style="list-style-type: none"> - UI/UX Designer: design user interface, create wireframes and prototypes using Figma - (Front-end): implement responsive designs and optimize user experience - Quality Assurance: execute testing and bug reporting, ensure design consistency and usability standards
--	---

4. Management Process

4.1 Project Estimates

4.1.1 Estimated Cost

The estimated cost for the Jobify project, focusing on a Minimum Viable Product (MVP) to be delivered within approximately 3.5 months (14 weeks), is approximately 6,000,000VNĐ - 10,000,000 VNĐ. This reduced estimate reflects a highly focused scope for the initial release, a lean team, and includes personnel costs, essential software licenses, cloud infrastructure, and operational expenses.

Cost Breakdown (Preliminary - Focused MVP):

- Phase 1: Inception (PA0 + PA1): 1,000,000 VNĐ - 2,000,000 VNĐ
- Phase 2: Elaboration (PA2 + PA3): 2,000,000 VNĐ - 3,000,000 VNĐ
- Phase 3: Construction (PA4 + PA5): 3,000,000 VNĐ - 5,000,000 VNĐ
- Contingency (10-15%): Included in the upper range of each phase.

4.1.2 Estimated Schedule

The estimated schedule for the Jobify project from inception to initial production release (MVP) is approximately 3.5 months (14 weeks).

Schedule Breakdown:

- Inception (PA0 + PA1): 4 weeks
- Elaboration (PA2 + PA3): 8 weeks
- Construction (PA4 + PA5): 4 weeks
- Total: 14 weeks

4.1.3 Basis for Estimates

These revised estimates are based on the following assumptions:

Team Size & Velocity: A very lean and efficient team of 4 core members focused on rapid development for an MVP.

Technology Stack: Continued use of modern web technologies (NextJS with Tailwindcss for frontend, Nodejs with Express for backend, PostgreSQL database) but with an emphasis on readily available libraries and minimal custom development.

Scope: A significantly reduced scope focusing only on the absolute core features necessary for a Minimum Viable Product (e.g., Candidate Registration & Profile, Basic Job Browsing, Simple Apply for Jobs, and essential HR posting/review for a limited set of users). Advanced features like AI-powered recommendations, comprehensive moderator/admin dashboards, and extensive tracking will be deferred to later phases.

Complexity: Assumes low-to-medium complexity for the initial MVP, focusing on essential user flows.

Historical Data: Leveraging industry benchmarks for rapid MVP development projects. This is a preliminary estimate and will be refined.

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

4.2 Project Plan

This section outlines the project's structured approach to scheduling, resource allocation, iteration objectives, software releases, and major milestones for the Jobify MVP development effort. The plan ensures that the project is delivered on time, within budget, and according to agreed scope and quality standards.

4.2.1 Phase Plan

The Jobify MVP will be delivered in three primary phases following the Rational Unified Process (RUP) for Small Projects:

Phase	Duration	Major Milestones	Achievement Criteria
Inception (PA0 + PA1)	4 weeks	Vision document finalized, initial risk assessment completed, MVP scope signed off	All requirements and risks documented and approved
Elaboration (PA2 + PA3)	6 weeks	Architecture baseline established, first working prototypes delivered	Core infrastructure, APIs, and preliminary UIs completed
Construction (PA4 + PA5)	4 weeks	MVP feature-complete, all acceptance tests passed	All core features implemented and tested; MVP ready for release

Important Release Points and Demos:

- End of Elaboration Phase (Week 10):
Demo of system skeleton and core job browsing workflow.
- End of Construction Phase (Week 14):
Final MVP demo for stakeholders.

4.2.2 Iteration Objectives

The Jobify MVP development will be structured into **six iterations** across the three RUP phases. Each iteration has clear objectives to guide incremental development and risk mitigation.

Iteration	Phase	Duration	Objectives
Iteration 1	Inception (PA0)	2 week	Define project vision, identify high-level requirements, initial stakeholder alignment
Iteration 2	Inception (PA1)	2 week	Complete detailed requirement gathering, conduct initial risk assessment, finalize MVP scope
Iteration 3	Elaboration (PA2)	3 weeks	Establish system architecture, implement core infrastructure and APIs, deliver first prototypes
Iteration 4	Elaboration (PA3)	3 weeks	Develop preliminary UI components, integrate job browsing and search features, conduct first demo
Iteration 5	Construction (PA4)	2 weeks	Implement remaining MVP features (user authentication, job posting, application tracking), conduct integration testing
Iteration 6	Construction (PA5)	2 weeks	Complete system testing, fix outstanding issues, prepare for final demo and MVP release

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

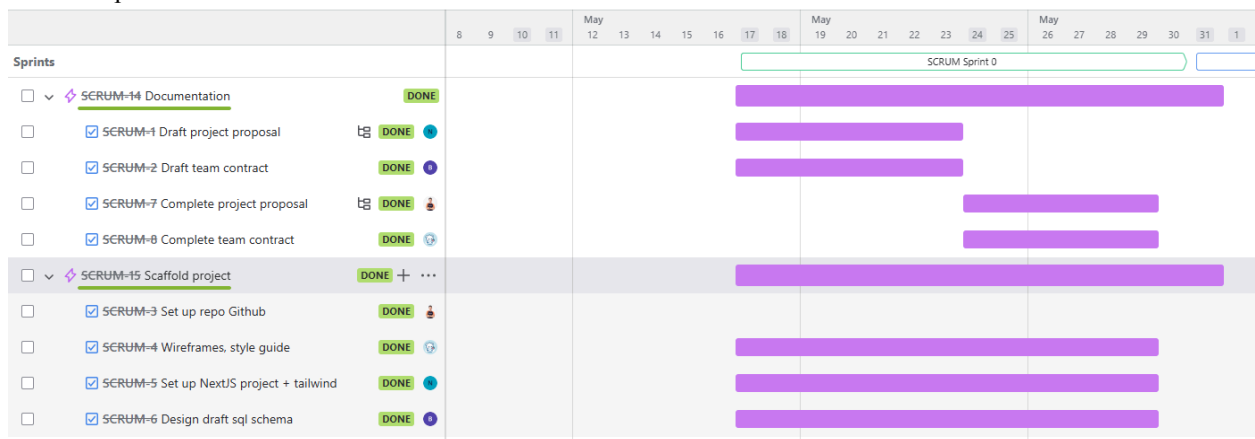
4.2.3 Releases

The Jobify MVP will involve the following software releases:

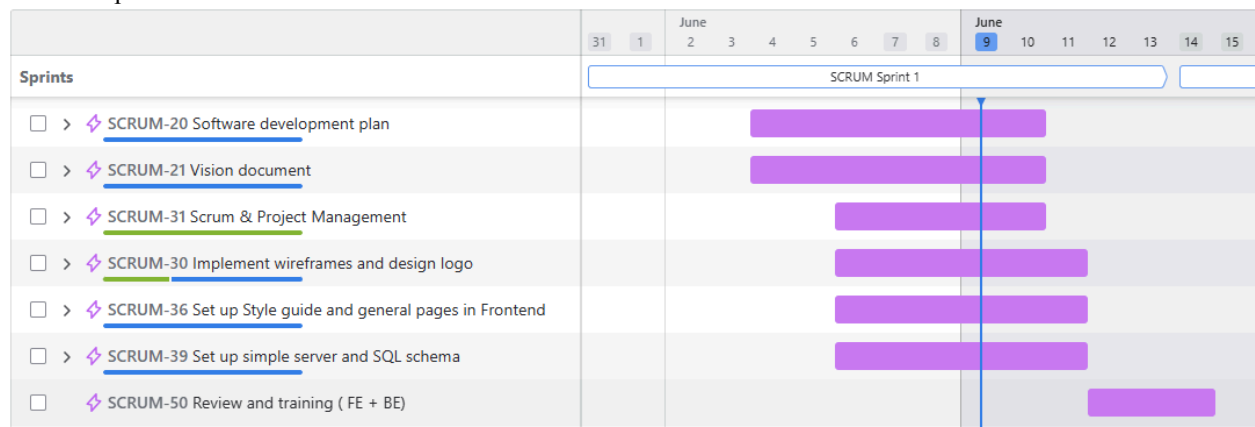
Release	Type	Description
Alpha Release	Internal Demo	Basic workflows implemented; core features functional (end of Iteration 4)
Beta Release	Stakeholder Demo	All MVP features integrated; partial testing completed (end of Iteration 5)
MVP Production Release	Final Delivery	Fully tested MVP deployed to production (end of Iteration 6)

4.2.4 Project Schedule

Sprint 0:

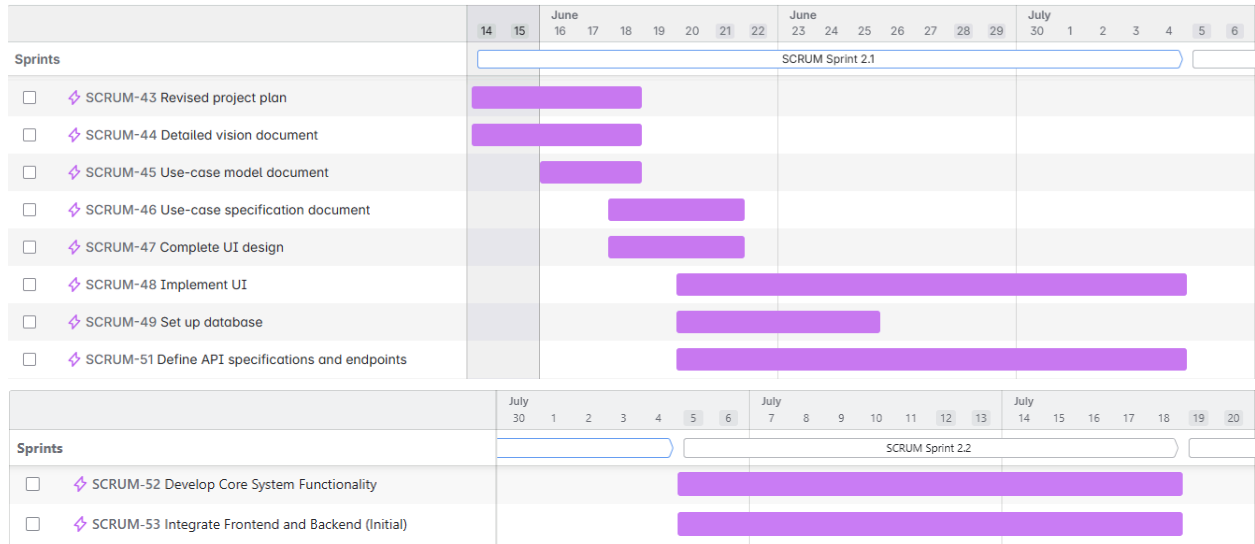


Sprint 1:

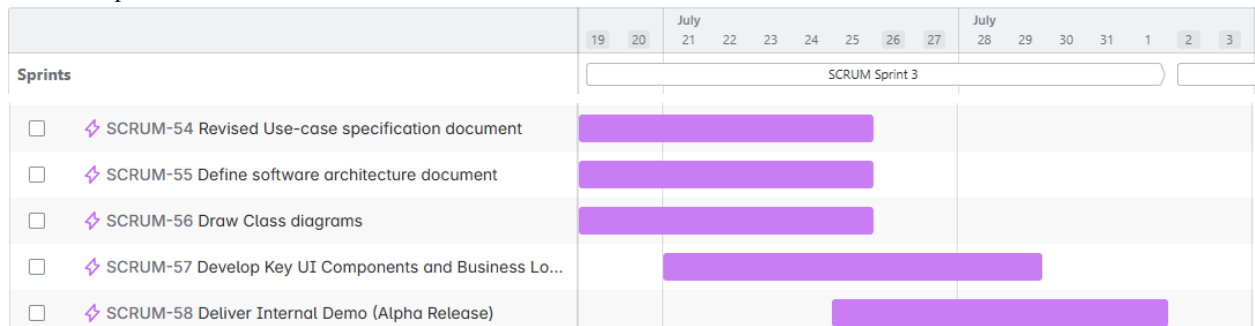


Sprint 2:

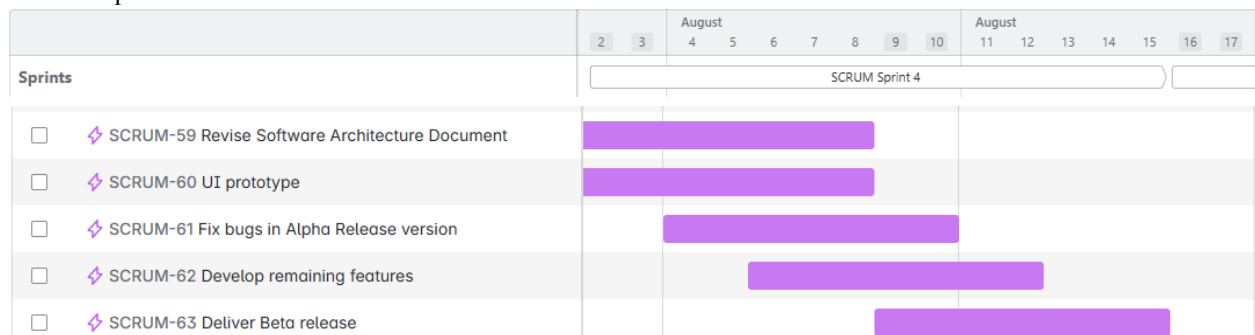
Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	



Sprint 3:

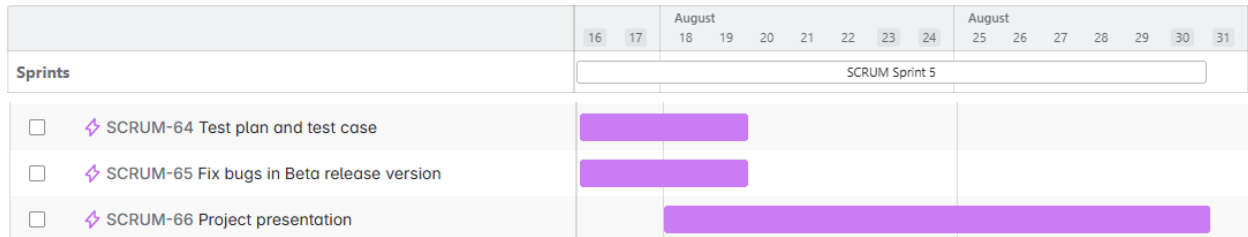


Sprint 4:



Sprint 5:

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	



4.2.5 Project Resourcing

The Jobify MVP will be developed by a compact, agile team of 4 members, covering all required roles through multi-skilled contributors. The table below outlines the team structure, responsibilities, and phase allocations:

Role	Team Member	Key Responsibilities	Assigned Phase(s)
Project Assistant <i>(also Fullstack Support)</i>	1	Project coordination, documentation, RUP process oversight, full-stack development support for backend and frontend as needed	All phases
Backend Developer	1	API development, database schema design, server-side logic implementation	Elaboration, Construction
Frontend Developer	1	React-based UI development, frontend logic, integration of APIs, and responsive layout implementation	Elaboration, Construction
UI/UX Designer	1	Design user interface layouts, create UI mockups, ensure usability and consistency	Inception, Elaboration

Resource Allocation by Phase:

Phase	PA	Backend Dev	Frontend Dev	UI/UX Designer
Inception (Week 1-4)	✓	–	–	✓
Elaboration (Week 5-10)	✓	✓	✓	✓
Construction (Week 11-14)	✓	✓	✓	–

Training Requirements:

Training Topic	Target Team Members	Completion Deadline
Agile Process & RUP for Small Projects	All team members	End of Week 2
Jobify System Architecture Overview	PA, Backend, Frontend Developers	End of Week 6

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

CI/CD Pipeline Setup	PA, Backend, Frontend Developers	End of Week 8
----------------------	----------------------------------	---------------

4.3 Project Monitoring and Control

The Jobify project will employ a robust monitoring and control framework to ensure adherence to scope, schedule, and budget, while maintaining high quality and managing risks effectively. Given the compressed timeline for an MVP, control mechanisms will be even more stringent to prevent scope creep and ensure timely delivery of core features.

4.3.1 Requirements Management

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process. This ensures that changes are systematically evaluated for their critical impact on the MVP scope, schedule, and resources before implementation.

4.3.2 Reporting and Measurement

At the end of each iteration, updated cost and schedule estimates, along with summary metrics reports, will be produced to reflect the current project status and inform planning for subsequent phases.

In alignment with the RUP Guidelines: Metrics, a minimal yet effective set of project metrics will be collected weekly to support informed decision-making and early issue detection. These include:

Earned Value of Completed Tasks:

- Used to reassess the remaining schedule and budget, and to identify potential scope adjustments based on actual progress.
- Defect Trends (Open vs. Closed): Tracked via trend graphs to visualize defect resolution velocity and help estimate the remaining effort required for stabilization.
- Acceptance Test Case Pass Rate: Visualized through trend analysis to demonstrate functional completeness and progress toward release readiness.

In parallel, overall project costs will be continuously tracked and compared against the allocated budget, ensuring early visibility into potential overruns and enabling timely corrective actions.

4.3.3 Risk Management

Risks will be identified in the Inception Phase using the steps identified in the RUP for Small Projects activity “Identify and Assess Risks”. Project risk is evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

Risk Ranking (High, Medium, Low)	Risk Description and Impact	Mitigation Strategy and/or Contingency Plan
High	Team lacks experience with selected tech stack (NextJS, Tailwind, Node.js, PostgreSQL) → May delay initial development.	Organize hands-on workshops, pair programming, and assign early training tasks. If necessary, simplify architecture or switch to a more familiar stack.
High	Unstable integration across frontend, backend, and database → Broken CI/CD pipelines, and regular integration checkpoints.	Establish clear integration interfaces early. Use API mocking, CI/CD pipelines, and regular integration checkpoints.
High	Unexpected behaviors during testing (e.g., job application workflow errors) → Delayed QA and rework.	Allocate buffer time for exploratory testing. Use real-world test scenarios early. Document edge cases in acceptance
High	Scalability issue: Job matching or application submission lags with high user traffic.	Perform stress testing early. Add load balancing or caching strategies. Consider scaling the backend with containerization if needed.
High	Unclear or overlapping responsibilities among team members → Task duplication or omission.	Use detailed task breakdown in planning sessions. Assign clear ownership via tools like Jira or Trello. Review responsibilities regularly.

Jobify	Version: <1.6>
Software Development Plan (Small Project)	Date: <13/06/2025>
SPD	

Risk Ranking (High, Medium, Low)	Risk Description and Impact	Mitigation Strategy and/or Contingency Plan
High	Coin balance may not sync correctly due to API or logic bugs.	Ensure backend logic includes transaction safety (atomic operations + rollback on error).
Medium	Lack of effective communication within the team → Misaligned expectations and slower problem-solving.	Encourage open channels (Slack, daily standups). Foster a supportive environment to raise blockers early. Assign a communication lead if needed.
Medium	Core reusable components (e.g., resume parser, job recommendation model) underperform or can't be reused as planned.	Identify alternatives (e.g., open-source libraries or APIs). Refactor to fit Jobify context or defer to later releases.
Medium	Scope creep due to stakeholder requests (e.g., add new features mid-sprint).	Lock MVP scope with a signed-off requirement set. Introduce a change control process to assess impact before approval.
Low	Critical team members become unavailable due to illness or personal issues.	Cross-train team members on essential modules. Maintain sprint-wise knowledge documentation. Assign backups for key tasks.
Low	Chosen tool or platform (e.g., PostgreSQL or deployment service) becomes unreliable.	Regularly back up data. Prepare alternate providers or service migration plans. Monitor platform status proactively.

4.3.4 Configuration Management

Appropriate tools will be selected that provide a database of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.