Software Project Plan

1. Project Overview

Project name	{{Project name}}
Sponsor	{{Sponsor / decision-maker}}
Project manager	{{Project manager}}
Start date	{{Start date}}
Target end date	{{Target end date}}
Status	{{Not started / On track / At risk}}

Keep this block on page one so anyone can see the project, its owner, and where it stands at a glance. Name one accountable project manager and one sponsor who can unblock decisions.

2. Objectives & Success Criteria

- Ship {{the feature / release / migration}} by {{date}} meeting {{a measurable bar — e.g. <0.5% error rate, no P1 regressions}}
- Stay within the agreed scope and {{budget / engineering capacity}}
- Hand off a documented, supportable system to {{the on-call / ops team}}

Write objectives you can measure — "launch the new checkout by Q3 with under 1% error rate", not "improve checkout". If you cannot tell whether it is done, rewrite it.

3. Scope

In scope

- {{The features, services, or systems being built or changed}}
- {{Environments: dev, staging, production}}
- {{Testing: unit, integration, UAT}} and the release itself
- {{Documentation and handoff}}

Out of scope

- {{Adjacent systems you will not touch}}
- {{Nice-to-have features explicitly deferred to a later release}}
- {{Work owned by another team}}

Naming what is out of scope is what actually prevents scope creep. Be explicit about the work people will assume is included but is not.

4. Milestones & Timeline

Milesto	Owner	Target date	Status
Requiren & design sign-off	{{Tech lead}}	{{Date}}	{{Status}}
Build complete (feature freeze)	{{Eng lead}}	{{Date}}	{{Status}}
QA / UAT passed	{{QA lead}}	{{Date}}	{{Status}}
Go-live / release	{{Release manager}}	{{Date}}	{{Status}}
Post-rele review & handoff	{{PM}}	{{Date}}	{{Status}}

Milestones are checkpoints, not tasks — a handful of dates that prove the project is moving. Each should be a clear “done / not done”, with one owner.

5. Tasks & Work Breakdown

Task	Owner	Start	Due	Status
{{Write the technical design doc}}	{{Owner}}	{{Start}}	{{Due}}	{{Status}}
{{Build {component}}}	{{Owner}}	{{Start}}	{{Due}}	{{Status}}
{{Write tests and set up CI}}	{{Owner}}	{{Start}}	{{Due}}	{{Status}}
{{Run UAT and fix defects}}	{{Owner}}	{{Start}}	{{Due}}	{{Status}}
{{Prepare release: runbook, rollback, monitoring}}	{{Owner}}	{{Start}}	{{Due}}	{{Status}}

Break the work into tasks small enough to track in a week or less. Give each an owner and a due date; a task with no owner will not happen.

6. Roles & Responsibilities

Role	Name	Responsibility
Project / product manager	{{Name}}	Owns scope, schedule, and stakeholder updates
Tech lead	{{Name}}	Owns the design and technical decisions
Engineers	{{Names}}	Build, test, and ship the work
QA	{{Name}}	Owns the test plan and release sign-off

Assign responsibilities to roles so the plan survives a team change. Be clear who decides, who does the work, and who needs to be kept informed.

7. Risks & Mitigation

Risk	Impact	Likelihood	Mitigation	Owner
{{Dependency on another team or vendor slips}}	{{High}}	{{Med}}	{{Mitigation}}	{{Owner}}
{{Migration data issues}}	{{High}}	{{Med}}	{{Dry-run + rollback plan}}	{{Owner}}
{{Scope creep mid-build}}	{{Med}}	{{Med}}	{{Change-request process}}	{{Owner}}

List the few risks that would actually derail the project, rate impact and likelihood (High/Med/Low), and name the action and the owner. Review this table at every status check.

8. Budget

Item	Estimate	Actual
{{Cost item — labor, tools, materials}}	{{Estimate}}	{{Actual}}
Contingency	{{Estimate}}	{{Actual}}

Estimate the main cost lines and add a contingency for the unknowns. If your project has no budget, keep the

section and write “N/A” — reviewers expect to see it.

9. Approval

Project Manager Name: _____ Signature: _____ Date: _____	Sponsor Name: _____ Signature: _____ Date: _____

Both sign-offs confirm the plan, scope, and budget are agreed before work starts.