

Matrix of Common Project Risks

Risk	Mitigating Strategy
Insufficient resources available to perform the work	<ul style="list-style-type: none"> ■ Explore various channels to secure resources, including hiring new staff or involving subcontractors, or consider training staff who currently lack sufficient skills. ■ Consider alternative implementation approaches or rescheduling and reprioritizing work. ■ Hire top talent.
High turnover on the project team	<ul style="list-style-type: none"> ■ Investigate reasons for turnover and provide feedback on possible corrective measures to management. ■ Improve team cohesion through proactive communication. ■ Work to establish a project environment for success.
Poor team dynamics	<ul style="list-style-type: none"> ■ Involve interactive team management to identify issues and act as facilitator to resolve team issues. ■ Implement processes to escalate conflict resolution to senior management if needed.
Friction between the project team and the customer	<ul style="list-style-type: none"> ■ Establish clear lines of communication between the project team and the customer. ■ Proactively manage communication. ■ Develop issue logs and plans to track and resolve issues. ■ Follow up on action items. ■ Ensure all project status information is accurate and up-to-date.

Risk	Mitigating Strategy
Contractor failure	<ul style="list-style-type: none"> ■ Check references. ■ Assess abilities prior to hiring. ■ Provide a scope of work that clearly identifies responsibilities. ■ Actively manage the contractor relationship.
Overly optimistic schedule	<ul style="list-style-type: none"> ■ Incorporate adequate time for planning, design, testing, bug fixing, retesting, changes, and documentation, and properly account for nonworking time such as weekends, holidays, and staff vacations. ■ Solicit feedback from the technical team when scheduling work. ■ Properly account for schedule dependencies including stakeholder dependencies that are not directly controllable.
Poorly defined requirements	<ul style="list-style-type: none"> ■ Develop clear, complete, detailed, cohesive, attainable, and testable requirements that are agreed to by all players. ■ Use prototypes to help nail down requirements. ■ In "agile"-type environments (fluid, changing continually), frequent coordination with customers/end users is necessary.

Risk	Mitigating Strategy
Scope creep	<ul style="list-style-type: none"> ■ Work closely with customers when developing requirements. ■ Use issue logs for customer communication. ■ Implement change control and configuration control mechanisms that identify the processes and approvals needed to implement change. ■ Be prepared to defend against excessive changes and additions once development has begun, and be prepared to explain consequences. ■ Use incremental development practices.
Inadequate design	<ul style="list-style-type: none"> ■ Insist on approved requirements prior to initiating design. ■ Provide specifications on design standards. ■ Allow sufficient time for design activities. ■ Conduct design reviews.
Poor software quality	<ul style="list-style-type: none"> ■ Insist on validating requirements and design specifications. ■ Require walk-throughs and inspections when appropriate. ■ Initiate review and testing early on; retest after fixes or changes. ■ Plan for adequate time for testing and bug fixing. ■ Analyze the causes of errors with the objective of implementing process improvements. ■ Use formal tools to track software discrepancies including their resolution.
Base technology/tools not ready for deployment	<ul style="list-style-type: none"> ■ Explore alternate implementation approaches. ■ Incorporate cost and schedule contingencies into the project baseline.