

TEMPLATE

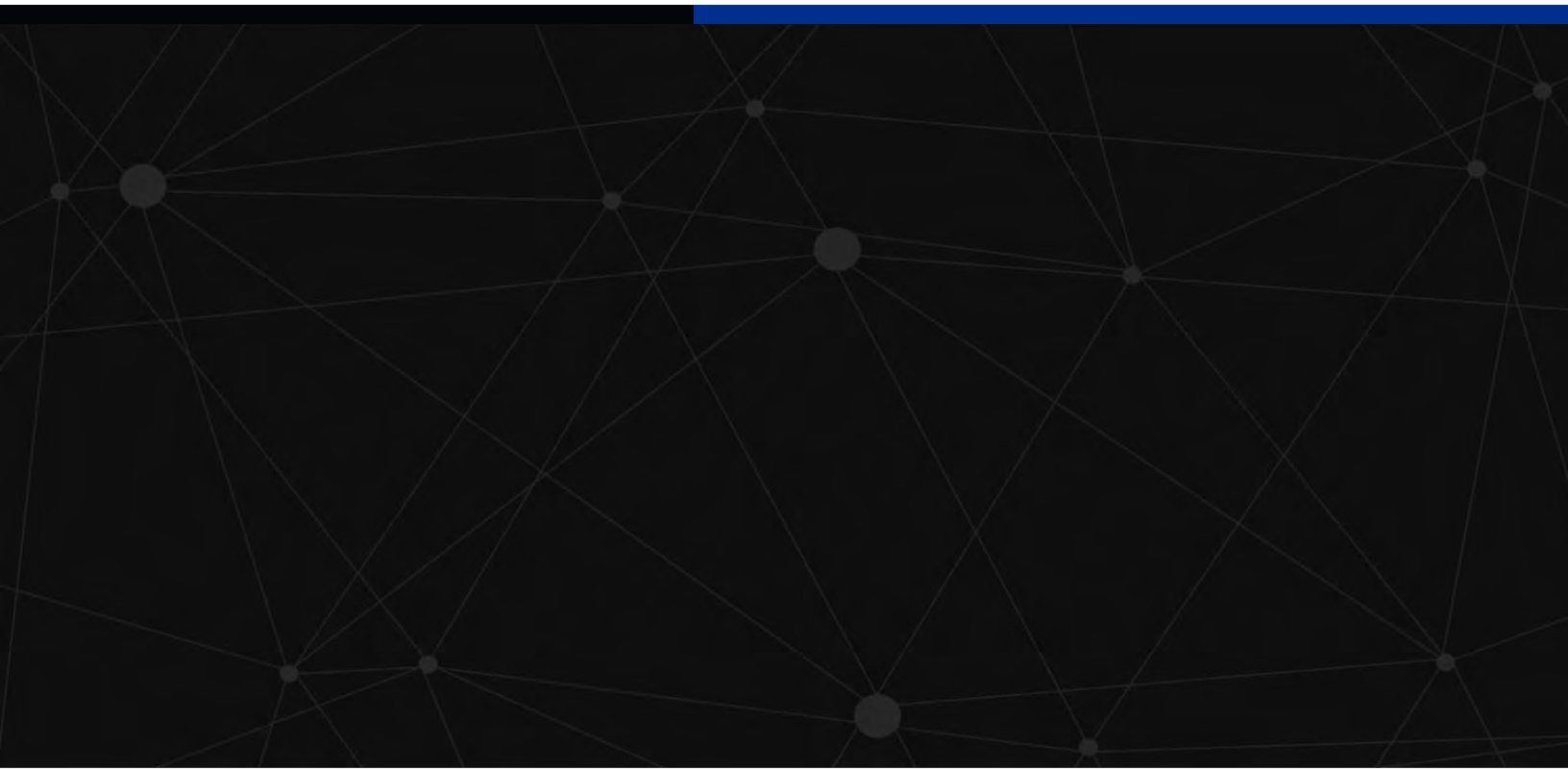
---

# SOFTWARE REQUIREMENTS SPECIFICATION

SRS TEMPLATE

## Table of Contents

Introduction .....	2
The Four Major Steps in Creating a Software Requirements Specification.....	2
How to Write Your Software Requirements .....	3
Who On Your Internal Team Should Be Involved .....	5
Software Requirement Specification Template.....	9



## Introduction

Welcome to QAT Global's Software Requirements Specification (SRS) Template! This template has been thoughtfully designed to assist our valued customers in providing the necessary details about their software projects. The SRS is a critical document that outlines the software solution's requirements, functionalities, and constraints. By completing this template, you will be able to provide our team with a comprehensive understanding of your project, enabling us to deliver a tailored and successful software solution.

The SRS template serves as a structured framework to capture essential information about your software project. It guides you through various sections, covering all critical aspects, including functional and non-functional requirements, user expectations, data management, external interfaces, and more. By completing each section thoughtfully, you will provide us with valuable insights into your project's objectives, scope, and specific needs.

This template is designed to be user-friendly and intuitive, with clear instructions provided for each section. It also includes example content and placeholders to guide you in formatting your responses effectively. We encourage you to provide as much detail as possible to ensure a comprehensive understanding of your requirements.

By utilizing this SRS template, you can establish a solid foundation for successful software development. It will enable effective communication, collaboration, and alignment between your organization and our development team. The completed SRS will be a reference point throughout the project, ensuring that the final solution meets your expectations and requirements.

We recommend completing the SRS template at the earliest stages of your project. This will help you to articulate your software needs clearly, avoid misunderstandings, and expedite the development process.

Thank you for choosing QAT Global as your software development partner. We are confident that this SRS template will facilitate a comprehensive understanding of your project, paving the way for a successful collaboration and the delivery of a high-quality software solution. Should you have any questions or require assistance, our team is here to support you throughout the process.

Let's embark on an exciting journey towards transforming your software vision into reality!

## The Four Major Steps in Creating a Software Requirements Specification

Understanding the four major steps involved in creating an SRS is essential for ensuring the success of software projects. Each step plays a vital role in capturing accurate and actionable requirements, from eliciting requirements to validating them. In this article, we will delve into the four major steps of creating an SRS, highlighting their significance and emphasizing the iterative nature of the process. Project owners and software development teams can foster collaboration, mitigate risks, and lay the foundation for a successful software solution by following these steps. These four major steps are:

## 1. Eliciting Requirements:

This step involves gathering and identifying requirements from various stakeholders, including clients, users, domain experts, and business analysts. Techniques such as interviews, workshops, surveys, and observations can be used to collect requirements. The goal is to understand the needs, expectations, and constraints of the software system.

## 2. Analyzing Requirements

In this step, the collected requirements are analyzed and refined to ensure they are clear, consistent, and feasible. This involves organizing, categorizing, and prioritizing requirements, resolving conflicts or ambiguities, and identifying dependencies and constraints. The analysis phase helps create a solid foundation for the subsequent steps.

## 3. Documenting Requirements

Once the requirements are analyzed and validated, they are documented in the SRS. This step involves documenting the functional and non-functional requirements, user stories, use cases, system constraints, and any other relevant information. The SRS serves as a reference document that provides a detailed description of what the software system should do and how it should behave.

## 4. Validating Requirements

The final step is to validate the requirements documented in the SRS. This involves reviewing the SRS with key stakeholders, including the client, users, and development team, to ensure that the requirements accurately capture their needs and expectations. Feedback and input from stakeholders are collected, and any necessary revisions or updates to the SRS are made. This iterative process continues until the requirements are deemed complete, consistent, and acceptable to all parties involved.

It's important to note that the process of creating an SRS is typically iterative, with feedback and revisions being incorporated throughout the steps. Collaboration and effective communication between stakeholders is crucial in ensuring the requirements' accuracy and completeness.

## How to Write Your Software Requirements

Writing a software requirement involves clearly articulating what the software system should do or how it should behave. Here are some guidelines to follow when writing software requirements:

1. Be Clear and Specific
  - State the requirement in an unambiguous manner.
  - Use precise and specific language to avoid any misunderstandings.
  - Avoid vague terms or ambiguous phrases that can lead to multiple interpretations.
2. Use a Consistent Format
  - Follow a consistent format or template for writing requirements to ensure clarity and uniformity.

- Include a unique identifier or reference number for each requirement for easy tracking and referencing.
3. Focus on Functionality
    - Specify the desired functionality or behavior of the software system.
    - Describe what the system should do, the inputs it should accept, the processing it should perform, and the outputs it should generate.
  4. Use Action Verbs
    - Start each requirement with an action verb to clearly indicate what needs to be done.
    - Use verbs like "create," "display," "calculate," "validate," etc., to describe the intended actions or operations.
  5. Be Testable and Measurable
    - Ensure the requirement can be objectively tested and validated to determine if it has been successfully implemented.
    - Specify any acceptance or success criteria that can be used to measure the fulfillment of the requirement.
  6. Avoid Implementation Details
    - Focus on the "what" rather than the "how" when describing requirements.
    - Avoid prescribing specific implementation details, technologies, or design decisions in the requirement. Leave room for the development team to determine the most appropriate implementation approach.
  7. Consider User Perspectives
    - Write requirements from the user's perspective, describing the desired outcome or experience.
    - Use user-centric language and terminology to ensure both technical and non-technical stakeholders easily understand the requirements.
  8. Include Necessary Context
    - Provide sufficient context or background information to help the development team understand the purpose and rationale behind the requirement.
    - Include any dependencies, constraints, or relevant system behavior that may impact the requirement.
  9. Review and Validate
    - Review each requirement for completeness, consistency, and feasibility.
    - To ensure shared understanding and agreement, validate the requirements with relevant stakeholders, including clients, users, and the development team.
  10. Maintain Traceability
    - Establish traceability between requirements and other project artifacts, such as design documents, test cases, and user stories.
    - Maintain a traceability matrix or traceability links to track the relationship between requirements and other project elements.

Remember, effective communication and collaboration with stakeholders throughout the requirement gathering and documentation process is crucial for ensuring the accuracy and completeness of the requirements.

## Who On Your Internal Team Should Be Involved

When writing the Software Requirements Specification (SRS), involving key members of the client's internal team who possess the necessary knowledge and expertise is crucial. The following roles within the client's organization should typically be involved:

### Project Sponsor/Owner:

- The project sponsor or owner should provide high-level guidance and overall project objectives.
- They play a critical role in defining the project scope and ensuring alignment with business goals.

### Subject Matter Experts (SMEs):

- SMEs from various functional areas of the organization should be involved.
- They have in-depth knowledge of the business processes, domain-specific requirements, and industry best practices.
- SMEs contribute to accurately defining the functional and non-functional requirements.

### Business Analysts:

- Business analysts act as a bridge between the client's team and the development team.
- They work closely with stakeholders to elicit, analyze, and document requirements.
- Business analysts ensure that the SRS reflects the client's needs and helps in achieving the desired business outcomes.

### Product Managers/Owners:

- Product managers/owners play a critical role in defining the product vision and understanding user needs.
- They contribute to the identification of user requirements and help prioritize features and functionalities.

### System/Technical Architects:

- System architects or technical architects provide technical expertise and insights.
- They ensure that the SRS aligns with the technical feasibility and architecture of the system.
- Architects help define the system's high-level structure and guide the decision-making process.

#### Quality Assurance (QA) Team:

- The QA team should be involved to ensure that the SRS includes requirements related to testing, quality assurance, and compliance.
- QA team members can provide insights into testability, acceptance criteria, and quality standards.

#### IT Operations/Infrastructure Team:

- The IT operations or infrastructure team should be involved in specifying any technical requirements or constraints related to the system's infrastructure, deployment, and maintenance.

#### Legal/Compliance Team:

- If there are legal or compliance requirements specific to the software system, involving the legal or compliance team is essential.
- They can provide guidance on regulatory standards, data privacy, security, or other legal considerations.

Having cross-functional representation within the client's team is important to ensure comprehensive and accurate requirement gathering. Collaboration and open communication between these team members contribute to a well-defined and aligned SRS.

QAT Global's business analysts can work closely with the client's internal team to facilitate requirement elicitation, documentation, and validation processes. They can guide the team through industry best practices, provide insights, and ensure that the SRS effectively captures the client's vision and goals.

## SRS Template Instructions

To help us understand your project requirements thoroughly, please follow these instructions when completing the Software Requirements Specification (SRS) template:

1. Introduction:
  - Provide a brief but clear overview of your project, including the project name, description, and objectives.
  - Keep the introduction concise, focusing on key information.
2. Scope:
  - Clearly define the boundaries and limitations of the software system.
  - Specify what is included and what is excluded from the system.
  - Provide a comprehensive understanding of the project's scope.
3. Functional Requirements:

- List all the functional requirements of the software system.
  - Each requirement should be described clearly and concisely.
  - Include specific features, functionalities, and behaviors expected from the system.
  - Provide sufficient details to ensure a shared understanding of the requirements.
4. Non-Functional Requirements:
    - List all the non-functional requirements of the software system.
    - Include performance, security, usability, reliability, and any other relevant requirements.
    - Specify any specific metrics or benchmarks for measuring non-functional requirements.
    - Use clear language and provide quantifiable measures wherever possible.
  5. User Requirements:
    - Describe the needs, goals, and expectations of the system's end users.
    - Specify user personas, user workflows, or any specific essential user requirements.
    - Clearly articulate the user's perspective and desired user experience.
  6. System Architecture:
    - Provide an overview of the system architecture, including high-level diagrams and descriptions of components, subsystems, and interfaces.
    - Specify any preferred technologies, platforms, or frameworks (if any).
    - Use diagrams or visual aids to enhance understanding where necessary.
  7. Data Requirements:
    - Identify the data entities, attributes, and relationships within the system.
    - Describe data formats, storage requirements, and any data management considerations.
    - Be specific about data sources, data flow, and any data manipulation requirements.
  8. External Interfaces:
    - Describe any interfaces with external systems, APIs, hardware devices, or software components.
    - Specify the protocols, data formats, and communication mechanisms required for seamless integration.
    - Identify any dependencies or expectations regarding external interfaces.
  9. Assumptions and Constraints:
    - List any assumptions made during the requirements-gathering process.
    - Clearly state any constraints or limitations that may impact the development or use of the software system.
    - Provide sufficient context to ensure all stakeholders clearly understand the project's assumptions and constraints.
  10. Project Timeline and Deliverables:
    - Provide a high-level project timeline, milestones, and deliverables.
    - Specify any dependencies or critical dates that should be considered.
    - Ensure the timeline is realistic and feasible within the project's constraints.
  11. Stakeholders:



- Identify the key stakeholders involved in the project.
- Specify their roles, responsibilities, and any specific expectations or requirements they may have.
- Clearly define the communication channels and points of contact for each stakeholder.

12. Technical Requirements:

- Identify any specific technical requirements or constraints.
- Include hardware, software, and infrastructure considerations.
- Be specific about the technical environment necessary for the software system.

13. Testing and Quality Assurance:

- Describe any specific testing requirements or quality assurance processes.
- Specify the types of testing (e.g., unit, integration, system) and any test data requirements.
- Clearly outline any specific quality assurance expectations or criteria.

14. Deployment and Support:

- Outline any specific deployment considerations.
- Specify any support or maintenance requirements post-deployment.
- Communicate any expectations or requirements related to deployment and ongoing support.

15. Documentation:

- Specify the desired documentation deliverables, such as user manuals, technical documentation, or any other specific documentation requirements.
- Clearly state the expected format or structure for the documentation.

Please ensure that your responses in the template are detailed and comprehensive, providing sufficient information to enable our team to understand your requirements fully. Use clear and concise language, avoiding unnecessary jargon or ambiguity.

If you have any questions or need clarification on any sections of the template, please reach out to your assigned QAT Global delivery manager or account manager for assistance.

We look forward to working with you to deliver a successful software solution that meets your needs and expectations.

# Software Requirement Specification (SRS) Template

Please provide the requested information for each section and include any additional details that would help us understand your project requirements thoroughly. You may require assistance from the technical team at QAT Global to complete some sections of the SRS, like architecture, data requirements, external interfaces, timeline, and other details, so please work with your designated contact to complete these sections.

Remember to consult with your QAT Global delivery manager or account manager to ensure the template is tailored to your specific project needs.

[Project Name]

## Software Requirements Specification (SRS) Template

### 1. Introduction:

- a. Project Name:
- b. Project Description:

- c. Project Objectives:

2. Scope:

a. Describe the boundaries and limitations of the software system.

b. Specify what is included and what is excluded from the system.

3. Functional Requirements:

- a. List the functional requirements of the software system.
- b. Include specific features, functionalities, and behaviors expected from the system.
- c. Provide a description for each requirement.

4. Non-Functional Requirements:

- a. List the non-functional requirements of the software system.
- b. Include performance, security, usability, reliability, and other relevant requirements.
- c. Specify any specific metrics or benchmarks for measuring non-functional requirements.

5. User Requirements:

- a. Describe the needs, goals, and expectations of the system's end users.
- b. Specify user personas, user workflows, or specific user requirements.

6. System Architecture:

- a. Provide an overview of the system architecture.
- b. Include high-level diagrams and descriptions of components, subsystems, and interfaces.
- c. Specify the preferred technologies, platforms, and frameworks (if any).

7. Data Requirements:

- a. Identify the data entities, attributes, and relationships within the system.
- b. Describe data formats, storage requirements, and data management considerations.



8. External Interfaces:

- a. Describe the system's interfaces with external systems, APIs, hardware devices, or software components.
- b. Specify protocols, data formats, and communication mechanisms.

9. Assumptions and Constraints:

- a. List any assumptions made during the requirements-gathering process.
- b. Identify any constraints or limitations that may impact the development or use of the software system.

10. Project Timeline and Deliverables:

- a. Provide a high-level project timeline, milestones, and deliverables.
- b. Specify any dependencies or critical dates.

11. Stakeholders:

- a. Identify the key stakeholders involved in the project.
- b. Specify their roles, responsibilities, and any specific expectations or requirements they may have.

## 12. Technical Requirements:

- a. Identify any specific technical requirements or constraints.
- b. Include hardware, software, and infrastructure considerations.

13. Testing and Quality Assurance:

- a. Describe any specific testing requirements or quality assurance processes.
- b. Specify the types of testing (e.g., unit, integration, system) and any test data requirements.

14. Deployment and Support:

- a. Outline any specific deployment considerations.
- b. Specify any support or maintenance requirements post-deployment.

15. Documentation:

- a. Specify the desired documentation deliverables (e.g., user manuals, technical documentation).
- b. Include any specific requirements for documentation.



At QAT Global, Your Success is Our Mission. For nearly 30 years, we've partnered with businesses to deliver custom software solutions that empower innovation, scale, and long-term success. Whether you're modernizing existing systems or developing new applications, our global team of experts is committed to delivering Quality, Agility, and Transparency—every step of the way.

We specialize in custom software development and offer flexible IT staffing solutions, including client-managed teams and team members and QAT Global-managed teams. With offices across the US, Brazil, and Costa Rica, we provide the expertise and collaboration you need to achieve your technology goals.

Join forces with a trusted partner who delivers excellence with integrity. Schedule a consultation with QAT Global today and let us help you build the custom software solutions that drive your success.

We Do It Right.

---

Get dozens of in-depth guides and more on business & leadership, custom software development, nearshoring, digital transformation, staff augmentation, and more by visiting:

[www.qat.com](http://www.qat.com)



**QAT GLOBAL**  
Quality • Agility • Technology