**Contents**

1 Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms and Abbreviations 3

1.4 References 3

1.5 Overview 3

2 Overall Description 4

3 Specific Requirements 4

3.1 Functionality 4

3.1.1 <Functional Requirement One> 4

3.2 Usability 4

3.2.1 <Usability Requirement One> 4

3.3 Reliability 4

3.3.1 <Reliability Requirement One> 4

3.4 Performance 4

3.4.1 <Performance Requirement One> 4

3.5 Maintainability 4

3.5.1 <Maintainability Requirement One> 4

3.6 Design Constraints 4

3.6.1 <Design Constraint One> 4

3.7 On-line User Documentation and Help System Requirements 4

3.8 Purchased Components 4

3.9 Interfaces 4

3.9.1 User Interfaces 4

3.9.2 Hardware Interfaces 4

3.9.3 Software Interfaces 5

3.9.4 Communications Interfaces 5

3.10 Licensing Requirements 5

3.11 Legal, Copyright, and Other Notices 5

3.12 Applicable Standards 5

4 Supporting Information 5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author(s)** |
| 21.10.2017 | 0.01 | Introduction | Valerii Pozdiaev |
|  |  |  |  |
|  |  |  |  |

# Introduction

[The introduction of the **Software Requirements Specification (SRS)** should provide an overview of the entire **SRS**. It should include the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the **SRS**.]

[Note: The Software Requirements Specification (**SRS**) captures the complete software requirements for the system, or a portion of the system. This document describes a typical **SRS** outline for a project using only traditional natural-language style requirements – with **no use-case modelling.**.]

[Many different arrangements of an **SRS** are possible. Refer to [IEEE830-1998] for further elaboration of these explanations, as well as other options for organizing an **SRS**.]

## Purpose

Данный документ является спецификацией требований программного обеспечения для системы автоматизации согласования договоров. Документ описывает саму систему, функциональные и нефункциональные требования, ограничения и системные интерфейсы. Документ предназначен для всех членов проектной команды, а также для действующих и потенциальных заказчиков данной системы.

## Scope

Система автоматизации согласования договоров – это программный продукт, позволяющий организовать работу с электронными договорами разных форматов.

Система предоставляет следующие возможности:

* Создание электронного договора на основе шаблона для соответствующего типа сделки
* Создание учетной записи пользователя с правами на создание договора и/или согласование
* Проведение процесса согласования договора

Система не работает с другими типами электронных документов и не поддерживает, помимо процесса согласования, другие форматы работы с электронными договорами.

Основными преимуществами данного продукта являются:

* Автоматизация и ускорение процесса согласования договора
* Уменьшение ошибок, возникающих при согласовании в бумажном формате
* Шаблонное создание договоров разных типов.

## Definitions, Acronyms and Abbreviations

В документе применяются следующий аббревиатуры:

* САСД: система автоматизации согласования договоров
* ПО: программное обеспечение
* ПС: процесс согласования

Определения:

**Менеджер контракта.** Пользователь системы, который создал контракт по выбранному шаблону и отправил его на согласование.

**Согласовывающее лицо.** Пользователь системы, который согласовывает часть договора, соответствующую его компетенции.

**Задача согласования.** Активность, в которой согласовывающее лицо подтверждает указанную в данной задаче часть договора.

**Процесс согласования.** Состоит из задач, количество которых соответствует числу требуемых согласований. Задачи в процессе расположены линейно и выполняются по порядку.

## References

Данная спецификация составлена в соответствии со стандартом [830-1993 - IEEE Recommended Practice for Software Requirements Specifications](http://www.utdallas.edu/~chung/RE/IEEE830-1993.pdf)

## Overview

Глава 1 содержит обзор программного продукта, описываемого в данном документе.

Глава 2 включает описание системы, ее общие функции и ограничения.

Глава 3 описывает все функциональные и нефункциональные требования к системе

[This subsection should describe what the rest of the **SRS** contains and explain how the document is organized.]

# Overall Description

[This section of the **SRS** should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

• product perspective

• product functions

• user characteristics

• constraints

• assumptions and dependencies

• requirements subsets]

# Specific Requirements

[This section of the **SRS** should contain all the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modelling, these requirements are captured in the Use-Cases and the applicable supplementary specifications.]

## Functionality

[This section describes the functional requirements of the system for those requirements which are expressed in the natural language style. For many applications, this may constitute the bulk of the **SRS** Package and thought should be given to the structure of this section. This section is typically structured by feature, but alternative structures may also be appropriate, for example, structure by user or by subsystem. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modelling tools, etc., are employed to capture the functionality, this section will refer to the availability of that data, indicating the location and name of the tool that is used to capture the data.]

### <Functional Requirement One>

[The requirement description.]

## Usability

[This section should include all of those requirements that affect usability. For example,

• specify the required training time for a normal users and a power user to become productive at particular operations

• specify measurable task times for typical tasks or base the new system’s usability requirements on other systems that the users know and like

• specify requirements to conform to common usability standards, such as Microsoft’s GUI standards]

### <Usability Requirement One>

[The requirement description goes here.]

## Reliability

[Requirements for reliability of the system should be specified here. Some suggestions follow:

• Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, etc.

• Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.

• Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?

• Accuracy—specify precision (resolution) and accuracy (by some known standard) that is required in the system’s output.

• Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand of lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).

• Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]

### <Reliability Requirement One>

[The requirement description.]

## Performance

[The system’s performance characteristics should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

• response time for a transaction (average, maximum)

• throughput, for example, transactions per second

• capacity, for example, the number of customers or transactions the system can accommodate

• degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)

• resource utilization, such as memory, disk, communications, etc.

### <Performance Requirement One>

[The requirement description goes here.]

## Maintainability

[This section indicates any requirements that will enhance the maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, maintenance utilities.]

### <Maintainability Requirement One>

[The requirement description goes here.]

## Design Constraints

[This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, etc.]

### <Design Constraint One>

[The requirement description goes here.]

## On-line User Documentation and Help System Requirements

[Describes the requirements, if any, for on-line user documentation, help systems, help about notices, etc.]

## Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc. so that the software can be developed and verified against the interface requirements.]

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behaviour, etc. ]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this **SRS** but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, etc.]

## Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notice, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, safety, security, etc.]

# Supporting Information

[The supporting information makes the **SRS** easier to use. It includes:

• Table of contents

• Index

• Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the **SRS** should explicitly state whether or not the appendices are to be considered part of the requirements.]