

Deliverable 1 Sample

Template document

Krzysztof Kaczmarski, Maciej Grzenda, Jarosław Legierski, Michał Okulewicz

version 2.0

12.10.2021

**Table of Contents**

[1 Abstract 3](#__RefHeading___Toc2796_2030408508)

[1.1 History of changes 3](#__RefHeading___Toc800_779315115)

[2 Vocabulary 4](#__RefHeading___Toc802_779315115)

[3 Specification 5](#__RefHeading___Toc6747_630044125)

[3.1 Executive summary 5](#__RefHeading___Toc6749_630044125)

[3.2 Functional requirements 5](#__RefHeading___Toc6751_630044125)

[3.3 Non-functional requirements 7](#__RefHeading___Toc6753_630044125)

[4 Project schedule 8](#__RefHeading___Toc6755_630044125)

[5 Risk Analysis 10](#__RefHeading___Toc6757_630044125)

[6 Bibliography 11](#__RefHeading___Toc6793_630044125)

# Abstract

This document contains a proposal of template dedicated for students’ projects conducted at the Faculty of Mathematics and Information Science Warsaw University of Technology. The template contains mandatory and optional sections that should be included in the project documentation together with their sample descriptions.

**Remarks:**

* **Exemplary descriptions of the content of the individual chapters of this document are an example and may not be copied to students’ documents.**
* **Document is intended for the exclusive use of students and faculty of Mathematics and Information Science, Warsaw University of Technology.**

## History of changes

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Description** | **Version** |
| 29.03.2016 | Jarosław Legierski | First version | 1.0 |
| 19.04.2016 | Jarosław Legierski | Modification of use cases - added APIs uusage | 1.1 |
| 28.09.2016 | Jarosław Legierski | Modification of contents of the document in accordance with the arrangements from the meeting 27.09 | 1.2 |
| 29.09.2016 | Michał Okulewicz | FURPS | 1.3 |
| 01.10.2016 | Karol Walędzik | Supplementing section of 'functional requirements' and 'change requests report' | 1.4 |
| 01.10.2016 | Jarosław Legierski | Fixed formatting + added functional tests (scenarios and results) | 1.5 |
| 01.10.2016 | Maciej Grzenda | Supplemented description in paragraphs schedule and data model | 1.6 |
| 16.10.2019 | Krzysztof Kaczmarski | Redone and reformatted | 2.0 |

# Vocabulary

**Obligatory chapter**

**(sample based on wikipedia records)**

**Abstract** - a brief summary of a research article, thesis, review, conference proceeding, or any in-depth analysis of a particular subject and is often used to help the reader quickly ascertain the paper's purpose

**Functional requirements** - In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs

**Non-functional requirements -** In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

**Risk analysis** - the science of risks and their probability and evaluation. Probabilistic risk assessment is one analysis strategy usually employed in science and engineering.

**Schedule**  - a basic time-management tool, consists of a list of times at which possible tasks, events, or actions are intended to take place, or of a sequence of events in the chronological order in which such things are intended to take place.

**Vocabulary** - a set of familiar words within a person's language. A vocabulary, usually developed with age, serves as a useful and fundamental tool for communication and acquiring knowledge

# Specification

## Executive summary

**Obligatory chapter**

In this section should be described, what is the main goal of creating a system / application, what features it has to meet, and who is main end user.

**Example description:**

This system is dedicated to collect and expose of open data sets for a wide range of developers and make them available in the form of open programming interfaces (Open APIs). Software platform and tools for exposure of open data, will be available to use with the use of free licenses (GPL, LGPL, FDL, CC).

## Functional requirements

**Obligatory chapter**

This section should be included, the requirements for the created system / application. Permitted forms: use case diagrams with description, user stories along with a description, in the alternative: usage scenarios

**Use cases**

Chapter should contain a list of use cases and their description. It is recommended to use UML to graphically represent use cases, actors, and the relationships between them. The list of cases use should be supplemented by the table with a detailed description. To complete diagrams of use cases it is recommended to use commercial tools (eg. MS Visio), or free tools (eg. Dia)

Figure 1 Use cases showing interaction of XXX with the module YYYY

Table 1 Description of use cases for actors and system YYYY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Actor | Name | Description | System response |
| Dev | Developer | A database search | Search through the data catalog in order to find a specific dataset (a web interface) | Webpage with found data sets that match the search criteria |
| Downloading  information about a Data Set | Downloading and displaying  details about a data set / function set | Webpage with a data set /  function set description |
| Data preview | Data preview | File, table, map, a reply for an API call showed on a webpage |
| Metadata preview | Downloading and displaying metadata related to the current data set | Webpage with a description of the current data set's metadata |
| Reading API information | Displaying API documentation | API documentation displayed on a webpage and/or in the form of an attachment |
| Example API use (doc) | Displaying documentation of an example API use | Documentation of an example API use displayed on a webpage and/or in the form of an attachment |
| Test API use | A button for executing an API call with example parameters (optional) | Showing the response on the webpage (data or the response in XML/JSON from WS), optional |
| Registration | Registering a user | Confirmation of registering a user – the user receives an APIKey (one APIKey per developer) |
| API use | API use in production | Synchronous communication with the platform (request-response) |
| Account removal | Removing an account from the platform | Confirmation of account removal |

**User Stories**

1. **Module 1**
   1. Usage 1 description
   2. Usage 2 description
2. **Module 2**
   1. Usage 1 description
   2. Usage 2 description

## Non-functional requirements

***Obligatory chapter***

*The chapter should contain a list of non-functional requirements and their description, preferably in the form of a table. (F)URPS + requirements classification is useful in capturing functional requirements.*

*FURPS is an acronym for:*

*• Functionality - a functional requirement presented above*

*• Usability - requirements related to the usability and ergonomics of the system*

*• Reliability - requirement related to system reliability and availability*

*• Performance - system performance requirements*

*• Supportability - system maintenance requirements*

*+ is designed to remind about specific requirements related to e.g. the need to use a specific technology, design patterns, project management methodology, data security, the existence of which depends on the specificity of the industry and system*

*The task of the teacher is to define what areas of non-functional requirements, apart from URPS, should be specified.*

*Example description:*

*Below are examples of non-functional requirements grouped into individual URPS categories*

Table 1 List of non-functional requirements for a remote repository of document records

|  |  |  |
| --- | --- | --- |
| Requirements area | Requirement No. | Description |
| Utility  ( *Usability* ) | 1 | All functionalities of the application available to the user must fit on a single screen with a resolution of 1920x1080 and a font no smaller than 12pt |
| Reliability  ( *Reliability* ) | 2 | The application is to be a *High Availability* application - available 24x7 at least 99.7% of the time between 7:00 a.m. and 7:00 p.m. with a single service break of no more than 2 hours a week on between 19:00 and 7:00. |
| 3 | The application must be resistant to the failure of any system machines and continue to work as unnoticed by the user if possible. |
| Performance  ( *Performance* ) | 4 | The application should add new objects to the system in no more than 3 seconds, with 100 object adding requests per minute |
| 5 | The application should provide *load balancing* between the system machines |
| Maintenance  ( *Supportability* ) | 6 | An instruction to restore the system from a backup will be delivered with the application |
| 7 | The system should be backward compatible with the interface of access to individual objects |
| 8 | The application should allow you to define the list of system machines from the administrator interface |

# Project schedule

***Obligatory chapter***

*The chapter should include a project schedule that includes at least milestones.*

*The purpose of creating a schedule is, of course, to plan work on the project, which in particular allows, to identify the sequence of tasks in the project. An important effect is often the statement that, for example, the project requires quick launching of the first tasks and / or their partially parallel implementation, because without taking these actions, it will be difficult or impossible to complete the project on the planned date. The project schedule contained in this point must define at least milestones, although for the above reasons it is also recommended to separate tasks with a duration of one to three weeks or more (the degree of detail depends on the duration of the entire project). The milestones include at least the following:*

*• completion of the key requirements analysis as reflected in previous sections of this document. Requirements should be approved by all stakeholders*

*• obtaining a stable solution architecture*

*• readiness of the application developed under the project for use*

*• completion of documentation and making changes resulting from tests of the application*

*Note: work on the requirements analysis does not exclude parallel work on the architecture or initial implementation.*

*The description should be in the form of a table with a list of tasks and milestones, their start and end dates, and optionally a Gantt diagram that presents this information and the relationship between tasks in a graphical form.*

***Sample description (two variants):***

*The project is planned to be implemented in accordance with the following schedule:*

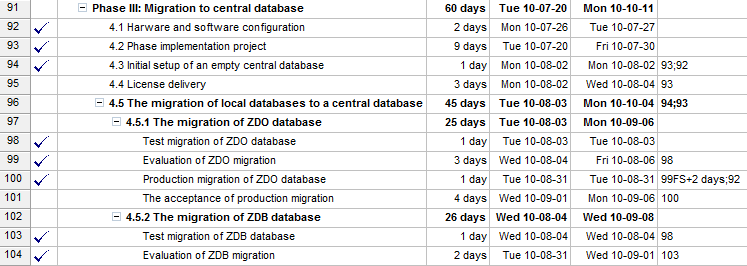


Figure 1 **Project schedule.**

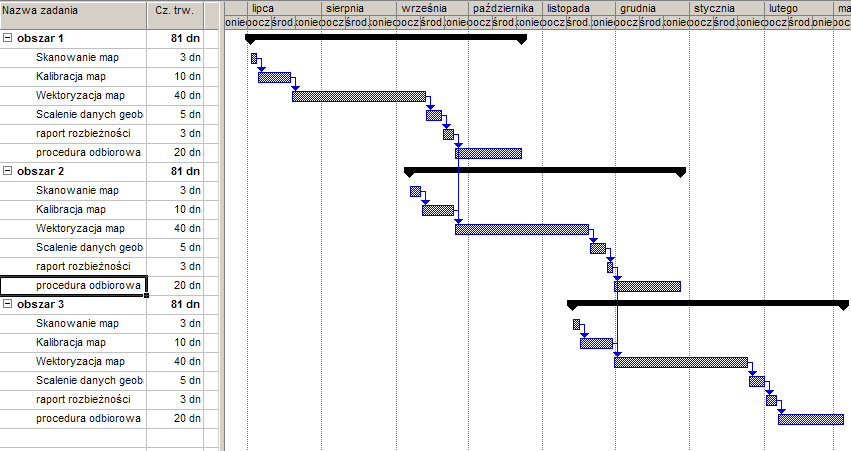


Figure 1 **Project schedule,** Gantt diagram.

# Risk Analysis

1. **Obligatory chapter**
2. Describe the points according to the SWOT methodology (Strengths, Weaknesses, Opportunities.Threats) The hazard points should be given a coarse evaluation of their likelihood and actions to counteract their occurrence.
3. ***Sample:***

|  |  |  |
| --- | --- | --- |
| 1. **SWOT** | 1. Threats | 1. Opportunities |
| 1. Internal | 1. 1. Not enough time due to other projects 2. 2 3. ... | 1. 1. 2. 2. 3. ... |
| 1. External | 1. 3. 2. 4. 3. ... | 1. 3. 2. 4. 3. ... |

1. Threat 1 : Very probable (Longer description of the sources of this threat).
2. Activities to minimize chances of occurrence.
3. Value of the threat.

# Bibliography

*Sample content*

1. Instalacja Tomcat 6.0 na CentOS 7.0 <http://www.howtoforge.com/how-to-install-tomcat-on-centos-7> [15.10.2015]
2. Instalacja Tomcat 6 z JDK 7 na CentOS RHEL Fedora Debian Ubuntu <http://linuxdrops.com/install-tomcat-6-with-jdk-7-on-centos-rhel-fedora-debian-ubuntu/> [15.10.2015]
3. Instalacja Tomcat 6 na CentOS lub RHEL <http://www.davidghedini.com/pg/entry/install_tomcat_6_on_centos> [15.10.2015]
4. Portal CKAN <http://ckan.org/> [30.10.2014]
5. Portal The Open Knowledge Foundation <https://okfn.org/> [30.10.2014]