Cognitive State Representation and Visualization of Human Brain

Software Requirements Specification
v1.0

Simple Labs

Atakan KAYA, 1746114
Barış NASIR, 1746288
Özlem Ceren ŞAHİN, 1746668
Bahattin TOZYILMAZ, 1746395

October 30, 2013
This document contains the software requirements specification for the “Cognitive State Representation and Visualization of Human Brain” software. The document is prepared according to the “830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.”.

This Software Requirements Specification provides a complete description of all the software requirements and views of the “Cognitive State Representation and Visualization of Human Brain”.

The first section of this document includes scope and purpose of project and gives overall description of it.

The following sections include detailed description and requirements of the project.
## Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changed</th>
<th>A/D/M</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>30 Oct 2013</td>
<td>-</td>
<td>-</td>
<td>Initial version</td>
</tr>
<tr>
<td>1.1</td>
<td>29 Dec 2013</td>
<td>2.2</td>
<td>M</td>
<td>Language, Platform change</td>
</tr>
<tr>
<td>1.1</td>
<td>29 Dec 2013</td>
<td>3.2</td>
<td>M</td>
<td>Language change</td>
</tr>
<tr>
<td>1.1</td>
<td>29 Dec 2013</td>
<td>4</td>
<td>M</td>
<td>Updated to current design</td>
</tr>
<tr>
<td>1.1</td>
<td>29 Dec 2013</td>
<td>6.2</td>
<td>M</td>
<td>Updated to current design</td>
</tr>
</tbody>
</table>

*A: Added, M: Modified, D: Deleted*
CONTENTS

1 Introduction .................................................. 6
  1.1 Problem Definition ...................................... 6
  1.2 Purpose .................................................. 7
  1.3 Scope .................................................... 7
  1.4 Definitions, Acronyms, and Abbreviations .......... 8
  1.5 References .............................................. 8
  1.6 Overview ................................................ 8

2 Overall Description .......................................... 9
  2.1 Product Perspective ..................................... 9
    2.1.1 System Interfaces .................................. 9
    2.1.2 Hardware Interfaces ................................. 9
    2.1.3 Memory ............................................. 9
  2.2 Constraints .............................................. 9
  2.3 Assumptions and Dependencies ....................... 10

3 Specific Requirements ........................................ 11
  3.1 Functional Requirements ............................... 11
    3.1.1 User Use Cases ................................... 11
    3.1.2 Use Case UC01: Load File .......................... 12
    3.1.3 Use Case UC02: Zoom In and Out ................. 12
    3.1.4 Use Case UC03: Rotation ........................... 13
    3.1.5 Use Case UC04: Four Regions ..................... 14
    3.1.6 Use Case UC05: Show Side-by-Side ............... 14
    3.1.7 Use Case UC06: Configure Colors ................. 15
    3.1.8 Use Case UC07: Transparency Adjustment ....... 16
    3.1.9 Use Case UC08: Change Voxel Size ............... 16
    3.1.10 Use Case UC09: Layer Depth Adjustment ....... 17
  3.2 Non-Functional Requirements ......................... 17
    3.2.1 Performance Requirements ......................... 17
    3.2.2 Design Constraints ................................ 18

4 Data Model and Description ............................... 19
  4.1 Data Description ....................................... 19
    4.1.1 Data Objects ...................................... 19

5 Behavioral Model and Desc. .............................. 20
  5.1 Description for Software Behavior ................... 20
  5.2 State Transition Diagrams ............................. 21

6 Planning .................................................... 22
  6.1 Team Structure ......................................... 22
6.2 Estimation (Basic Schedule) ............................................. 22
6.3 Process Model ............................................................. 22

7 Conclusion ................................................................... 25
1 INTRODUCTION

This document is a software requirement specification for a representation and visualization of cognitive state of human brain project. In this document, firstly we are going to define the problem and introduce the purpose and the scope of this document. Secondly, we are going to give an overall description. After these steps, we are going to state specific requirements, data models and behavioral models with their descriptions consecutively. Finally, we are going to present our planning, team structure, basic schedule and process model.

1.1 PROBLEM DEFINITION

Developing ways to study mental representations has been a long-standing and intriguing challenge to computer scientists. MVPA is a very recent approach that few researchers have implemented in neuroimaging research. MVPA uses machine learning algorithms to train on fMRI data, and can be used to classify the type of information being processed or cognitive operation that is being carried out, based on distributed patterns of neural activity in the brain. In its current state, MVPA is a very advanced and promising approach that can enable researchers to answer questions about cognitive operations and brain function that could not be assessed with other approaches. However, given its recent nature, the algorithms researchers use for MVPA has much room to improve. In this project, we propose an interdisciplinary collaboration that aims to fill this gap in the literature. Specifically, we propose several fMRI studies, whereby the feasibility of the current machine learning algorithms in the literature can be tested, and the algorithm with maximum performance can be identified. With this approach, via seeking out the machine learning approach that can best describe neural activity in the brain associated with cognitive processes, our aim is to build a model, called Local Voxel Networks that can be used to classify neural activity assessed by fMRI during cognitive operations. The suggested approach has three main steps:

*Step 1:* Acquiring the fMRI data from experiments

*Step 2:* Enhancing the fMRI data by image preprocessing techniques to prepare for MVPA techniques

*Step 3:* Developing a new model for fMRI image and comparing it to the available techniques

Above, there is the definition of "Tubitak Project 112E315" which is in process at the Image Processing and Pattern Recognition Laboratory of Computer Engineering Department, Middle East Technical University. During our senior project, we are going to be part of this project. Since it is a huge project and there is time limitations, our focus is going to be
implementing the 2nd step of the aforementioned study only.

1.2 PURPOSE

This software requirement specification document is defining the detailed description of the architecture, functionalities and specifications of the project. This document is going to serve as a guideline for both the development team and the users.

Since more than one version of this document are going to be released, there may be modifications to adapt changes of requirements and specifications of the project.

The target audience of this project is mostly academicians and medical institutes. Cognitive state representation and visualization of human brain is fundamentally important in neuroanatomy, neurodevelopment, cognitive neuroscience and neuropsychology.

Additionally, as stated in the project proposal, physicians can benefit from this tool to assess how risky brain surgery or similar invasive treatment is for a patient and to learn how a normal, diseased or injured brain is functioning. They can map the brain with this tool to identify regions linked to critical functions such as speaking, moving, sensing, or planning. This is useful to plan for surgery and radiation therapy of the brain. Clinicians also can benefit from this to anatomically map the brain and detect the effects of tumors, stroke, head and brain injury, or diseases such as alzheimer's.

1.3 SCOPE

This project's final product is a high-end desktop/compute cluster application, named "Cognitive State Representation and Visualization of Human Brain". The software is responsible for analysis and visualization of fMRI data.

Cognitive neuroscience is an emerging area or study. These studies aim to understand how human brain actually works. However, brain imaging tools generate data that is generally too big for a human to analyze. The product will be used in cognitive neuroscience studies to present this data in a more understandable manner. Software will first read brain imaging data, and then analyze it to see connections, merge closely connected components and visualize the data so that it is easy to inspect and see connections.
This project will be done by 4 people in 2 semesters.

1.4 Definitions, Acronyms, and Abbreviations

<table>
<thead>
<tr>
<th>3D</th>
<th>3 Dimensional</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC</td>
<td>Functional Connectivity</td>
</tr>
<tr>
<td>fMRI</td>
<td>functional Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>GLU</td>
<td>OpenGL Utility Library</td>
</tr>
<tr>
<td>GLUT</td>
<td>OpenGL Utility Toolkit</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphichs Processing Unit</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>METU</td>
<td>Middle East Technical University</td>
</tr>
<tr>
<td>MVPA</td>
<td>MultiVoxel Pattern Analysis</td>
</tr>
<tr>
<td>OpenGL</td>
<td>Open Graphics Library</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
</tbody>
</table>

1.5 References


1.6 Overview

The following chapters of this software requirement specification document includes six major topics. Firstly, overall description describes the product perspective, product functions, constraints and dependencies. In the next chapter, specific requirements are identified. After that data model and behavioral model are given in consecutive chapters. Finally planning and estimated schedule is stated. Document is ended with a conclusion.
2 OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

2.1.1 SYSTEM INTERFACES

The system developed will need to read brain scan (fMRI) data, convert these scanned data to brain regions and analyze these regions. To achieve this with best modularity, we plan to decouple these subsystems. So there will be an input subsystem, an analysis subsystem and a visualization subsystem; all of them connected via well-defined data interfaces.

2.1.2 HARDWARE INTERFACES

The process needed for visualizing the human brain data is computationally expensive. The project needs a compute cluster or a high-end processor. Again, memory demand is high. Visualization part also needs an OpenGL capable graphics card. Also, some calculations may be sideloaded to GPU, so a good graphics card will benefit the project.

2.1.3 MEMORY

The project uses input data in the range of gigabytes. This data has to be read once and converted to a more compact format, maybe compressed in the process.

Assuming 80,000 nodes and a 80,000 × 80,000 adjacency list, the software needs at least 80,000 × 80,000 × sizeof(float) memory for one frame of data. That is 25,600,000,000 bytes for the best case, nearly 24 gigabytes of continuous data.

For secondary storage purposes, the project has much more challenging requirements. A typical dataset contains 30 or more frames. So, a single dataset takes up 720 gigabytes of data with an uncompressed file format.

2.2 CONSTRAINTS

First and foremost constraint is the performance of the program. C++ is the de facto language for developing performance-critical applications. However, we choose to implement our program in C#. This is mostly due to our selection of 3D engine and extensibility of the program.
As stated before, the project is being developed for research purposes. Project team queried several researchers about their setup to find most suitable configuration. It was concluded that the project should be developed for Windows platform. Quality advantage of Windows graphics card drivers aided this choice.

Apart from stated two constraints, there are no other constraints for this project.

2.3 ASSUMPTIONS AND DEPENDENCIES

First of all, the project assumes that the host computer will have enough RAM to load a single frame and enough secondary storage capacity to store whole file in a single disk. Also, it is assumed that host computer(or the client) will have a graphics card with OpenGL capabilities.

There are open-source projects on similiar fields of research. They have a great potential for code reuse. However, most of these projects are written in Python or Matlab languages. Consequentially, they use Matlab file formats.

The project is expected to be mostly self contained, using libraries that already exist in the development environment client uses. Project team will try to minimize external dependencies. OpenGL/GLU/GLUT trio will definitively be used. For the aforementioned reason, the project will also depend on a library for Matlab interoperability.

Other assumptions can be listed as:

- A linux environment
- Working OpenGL/GLU/GLUT libraries
- Preferably an OpenCL capable GPU or a High Performance Computing cluster
3 SPECIFIC REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

3.1.1 USER USE CASES

Cognitive State Representation and Visualization of Human Brain

- Layer Depth Adjustment
- Change Voxel Size
- Transparency Adjustment
- Configure Colors
- Show Side-by-Side
- Four Regions
- Rotate
- Zoom In and Out
- Load File
3.1.2 Use Case UC01: Load File

Actor

Load File

<table>
<thead>
<tr>
<th>Actor</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The user loads the data taken from fMRI machine.</td>
</tr>
<tr>
<td>Trigger</td>
<td>The user presses the &quot;Load File&quot; button.</td>
</tr>
</tbody>
</table>
| Basic Path | 1. The user presses "Load File" button.  
2. The user chooses the data to be analyzed.  
3. 3D image of the choosen data is shown to the user. |
| Alternative Paths | None |
| Post-Condition | 3D image of the choosen data is shown to the user. |
| Other      | None |

3.1.3 Use Case UC02: Zoom In and Out

Zoom In and Out

actor

<table>
<thead>
<tr>
<th>Actor</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Simple Labs 12 of 25</td>
</tr>
</tbody>
</table>
3 SPECIFIC REQUIREMENTS

<table>
<thead>
<tr>
<th>Actor</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The user can zoom in and zoom out to 3D image of the brain.</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>The user presses either &quot;Zoom In&quot; or &quot;Zoom Out&quot; buttons</td>
</tr>
</tbody>
</table>
| **Basic Path** | 1. The user presses zoom in (zoom out) button.  
2. Zoom in (zoom out) image of the brain is shown to the user. |
| **Alternative Paths** | 1. The user wheels up (down) the mouse for zoom in (zoom out). |
| **Post-Condition** | Zoom in (zoom out) image of the brain is shown to the user. |
| **Other** | The tool does not allow the user to zoom in or zoom out infinitely. |

3.1.4 USE CASE UC03: ROTATION

![User Interface Diagram]

<table>
<thead>
<tr>
<th>Actor</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The user can rotate the image 360 degree around its center to any direction.</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>The user presses the &quot;Rotate&quot; button to activate rotation.</td>
</tr>
</tbody>
</table>
| **Basic Path** | 1. The user presses the "Rotate" button to activate rotation.  
2. The user clicks (left click) anywhere on the image and drags the mouse to any direction that she/he wants.  
3. The rotated image is shown to the user as she/he drags the mouse.  
4. The user releases the left click when wanted angle is captured. |
| **Alternative Paths** | None |
| **Post-Condition** | The rotated image is shown to the user. |
| **Other** | None |
3.1.5 Use Case UC04: Four Regions

Actor

Description: The user can do selection between four brain lobes. Only the selected lobes will be visible.

Trigger: The user presses the "Lobe Selection" button.

Basic Path:
1. The user presses the "Lobe Selection" button.
2. A popup window appears. It contains 4 main lobes (frontal, parietal, temporal, occipital) of brain.
3. The user does her/his selection.
4. The popup window is disappeared and only the selected lobe is appeared on tool.

Alternative Paths: None

Post-Condition: The popup window that contains lobes of brain.

Other: None

3.1.6 Use Case UC05: Show Side-by-Side

Actor

Description: The user can do selection between four brain lobes. Only the selected lobes will be visible.

Trigger: The user presses the "Lobe Selection" button.

Basic Path:
1. The user presses the "Lobe Selection" button.
2. A popup window appears. It contains 4 main lobes (frontal, parietal, temporal, occipital) of brain.
3. The user does her/his selection.
4. The popup window is disappeared and only the selected lobe is appeared on tool.

Alternative Paths: None

Post-Condition: The popup window that contains lobes of brain.

Other: None
3 SPECIFIC REQUIREMENTS

<table>
<thead>
<tr>
<th>Actor</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>With this feature, the user can see the brain from different sides in one window.</td>
</tr>
<tr>
<td>Trigger</td>
<td>The user clicks the &quot;Side by Side&quot; button.</td>
</tr>
<tr>
<td>Basic Path</td>
<td>1. The user clicks the &quot;Side by Side&quot; button.</td>
</tr>
<tr>
<td>Alternative Paths</td>
<td>None</td>
</tr>
<tr>
<td>Post-Condition</td>
<td>The window is divided into six pieces to show the image from different views, namely, front, left, right, rear, up and bottom.</td>
</tr>
<tr>
<td>Other</td>
<td>None</td>
</tr>
</tbody>
</table>

3.1.7 USE CASE UC06: CONFIGURE COLORS

![Diagram of use case UC06: Configure Colors]

<table>
<thead>
<tr>
<th>Actor</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Changes color distribution for color blinds.</td>
</tr>
<tr>
<td>Trigger</td>
<td>The user clicks the &quot;Change Color Distribution&quot; button.</td>
</tr>
<tr>
<td>Basic Path</td>
<td>1. The user clicks the &quot;Change Color Distribution&quot; button.</td>
</tr>
<tr>
<td>Alternative Paths</td>
<td>None</td>
</tr>
<tr>
<td>Post-Condition</td>
<td>Color distribution is rearranged according to color blinds.</td>
</tr>
<tr>
<td>Other</td>
<td>None</td>
</tr>
</tbody>
</table>
3.1.8 Use Case UC07: Transparency Adjustment

<table>
<thead>
<tr>
<th>Actor</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Changes transparency of the voxels so deeper layers can be seen.</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>The user changes &quot;Transparency&quot; slider.</td>
</tr>
<tr>
<td><strong>Basic Path</strong></td>
<td>1. The user changes &quot;Transparency&quot; slider.</td>
</tr>
<tr>
<td><strong>Alternative Paths</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Post-Condition</strong></td>
<td>The transparency of voxels is changed.</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td>None</td>
</tr>
</tbody>
</table>

3.1.9 Use Case UC08: Change Voxel Size

<table>
<thead>
<tr>
<th>Actor</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Changes unit size of voxels in case size is miscalculated.</td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
<td>The user changes &quot;Voxel size&quot; slider.</td>
</tr>
<tr>
<td><strong>Basic Path</strong></td>
<td>1. The user changes &quot;Voxel size&quot; slider.</td>
</tr>
<tr>
<td><strong>Alternative Paths</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Post-Condition</strong></td>
<td>Voxel size is changed.</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td>None</td>
</tr>
</tbody>
</table>
3.1.10 USE CASE UC09: LAYER DEPTH ADJUSTMENT

<table>
<thead>
<tr>
<th>Actor</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Allows user to view inner layers of brain.</td>
</tr>
<tr>
<td>Trigger</td>
<td>The user adjusts the &quot;Layer Depth&quot; slider.</td>
</tr>
<tr>
<td>Basic Path</td>
<td>1. The user changes the value of the slider by moving the cursor right or left.</td>
</tr>
<tr>
<td>Alternative Paths</td>
<td>1. The user clicks left (right) button located on the left (right) side of the slider.</td>
</tr>
<tr>
<td>Post-Condition</td>
<td>Inner layer or outer layer is shown to the user according to the user’s input.</td>
</tr>
<tr>
<td>Other</td>
<td>None</td>
</tr>
</tbody>
</table>

3.2 NON-FUNCTIONAL REQUIREMENTS

3.2.1 PERFORMANCE REQUIREMENTS

As we mentioned in the project proposal, dealing with large data causes some performance issues. The data contains chunk of voxels and our aim is connecting related voxels. Functional Connectivity (FC) is the statistical dependence between remote neural elements or regions across time. FC consist of two main steps:

1. Find correlation between two voxels’ time series using a correlation metric.
2. Construct correlation matrix (connectivity matrix) by using each pairs’ correlation response.

However, connectivity matrices are expensive in voxel level. For example, if we have 8142 voxels in our data set then we will have approximately 33 million functional relations. If number of voxels reaches 82926 then we will have 3428 million functional relations. Then, suppose computation time of each functional connectivity metric takes approximately one
second. For the input that contains 8142 voxels, total computation time reaches up to 383 days (with 256-way perfect parallelization it will take 1.5 days). Even 8000x8000 sized correlation matrix is expensive where our domain spans approximately 80000 voxels and their relations. To handle this problem we will apply GPU parallelization for speed-up and reformulate correlation metric for GPU. Also, we will use C# and OpenGL where other products and projects are using MatLAB.

3.2.2 Design Constraints

During the development of the project, we will mostly use our own computers. However because of the nature of the complexity of the brain, we are going to work with huge data. Therefore time by time, our computers and processors may not be enough to evaluate and process the data and when necessary we are going to have access to the computers in Image Processing Laboratory at Computer Engineering Department of METU.

The fMRI data is going to be provided by Prof. Dr. Fatoş Yarman Vural and during evaluation of the data Connectome and BrainNet Viewer are going to be used to create a background for this project. Final 3D image which will be shown in the graphical user interface is going to be obtained by using OpenGL. During the development of the project C# is going to be used as programming language.

Additionally UML standards will be used for the diagrams and IEEE standards will be used for the reports.
4 DATA MODEL AND DESCRIPTION

4.1 DATA DESCRIPTION

4.1.1 DATA OBJECTS

In this section of the requirements document, the relationships of interconnections and access among entities are specified. These relationships include information sharing, order of execution and parameterization of interfaces.
5 Behavioral Model and Desc.

5.1 Description for Software Behavior

The project's workflow is pretty simple. It follows this simple pattern:

1. Read data
2. Process data for visualization
3. Draw data

When the program opens, it should check whether command line contains any options or input files. If they exist, the program should make necessary changes and/or load the input file. User can cancel the action and close the program at this stage.

Program may or may not support loading from different file formats, e.g: Matlab file format, CSV format or plaintext formats. Readers for all these formats will be written and produce output in the same structure. When users select a file, program will begin loading the data. This process can be interrupted and paused.

After data is loaded, program will begin processing data. Processed data may be saved for caching purposes, since the processing part will be the most burdensome computation that will be done by the program. In the process, raw data will be compressed by lossy techniques, resulting nodes will be cross checked for any potential correlations. When this process is done, data will be converted to brain coordinates. This process can be paused or cancelled.

Visualization part is the next step. Brain coordinates will be converted to physical coordinates and shown on the screen. This may seem like a simple task, but user can still change settings at this stage. For example, voxel size may be changed, requiring all triangles to be reset.
5.2 State Transition Diagrams

![State Transition Diagram]

- Application
- Worker thread
  - Wait for input file
  - Load file
  - Analyze
- Visualisation
  - Exit Button
  - Change setting
  - Exit Button
6 PLANNING

6.1 TEAM STRUCTURE

Prof. Dr. Faroş Yarman Vural - Project Advisor
Dr. Atilla Özgit - Advisor
Mehmet Çelik - Advisor
Özlem Ceren Şahin - Researcher, Developer
Bahattin Tozyılmaz - Researcher, Developer
Atakan Kaya - Researcher, Developer
Barış Nasır - Researcher, Developer

6.2 ESTIMATION (BASIC SCHEDULE)

The project is planned to be finished by June 2014. There will be 4 revisions. At each revision, the program will become more usable and bug-free.

At first revision, it is planned to have basic visualization functionalities. We will work with a toy dataset to understand data fields.

At second revision, processing pipeline will be implemented. This pipeline will enable us to do some basic operations on dataset. Hardware acceleration will not be used at this point.

Third and fourth revisions will be focused based on client feedback.

6.3 PROCESS MODEL

One of the most important thing when developing a software product is to decide on a methodology conforming the needs and the requirements of the project. In this section, these software development methodologies will be briefly discussed and both the advantages and the disadvantages of them will be compared and finally the chosen methodology for this project will be further explained.

The most straightforward methodology is "Waterfall". It is known as the traditional approach to software development. In this method, the phases are conducted for one time and should be done correctly as going back is seen very rare. For instance, the planning
will be performed in the very beginning of the project and re-planning will not be possible when the team goes on to the next phase, thus this requires a thorough understanding of the project as a whole and foreseeing the possible obstacles on the way.

Waterfall methodology might be the best choice for small-scale projects that include only a few deliverables. However, as the tasks get more complex, reviewing the phases will be inevitable. Moreover, the tasks should be prioritized to meet the time and resource constraints. Consequently, waterfall methodology would be highly risky as since it would prevent to realize the value of the project and leave the testing until the very end and may not handle any requirements change on the way.

At this point, agile software development comes into question since it welcomes changing requirements, yields sustainable development and working software is the principal measure of the progress. One of the most well-practiced process models is called Scrum which is an iterative and incremental agile software development framework. The comparison of Waterfall and Scrum frameworks can be seen in Figure 6.1.

It is essential to question whether Scrum is appropriate for a research project since it is originally created for development projects. For this, understanding the nature of our project is vital. None of this project’s actions, practices or techniques are simple or repeatable, its predictability is limited and it is difficult to control. The activities may require constant changes in directions, new tasks may be added or it may face unforeseen and challenging issues. In other words, our project is an "empirical" process that cannot be well defined and uncertainty is inevitable. Marchesi et al. suggests Scrum for the management of the projects that has these properties.

As it is stated by Marchesi et al., Scrum has three roles, three ceremonies, and three artifacts. The roles defined by the Scrum framework are Product Owner, Scrum Master and Scrum Team. The Product Owner will have the control of the whole project, and will ensure that the artifacts delivered by the process are in line with the research project aims and the
required deliverables. Obviously, Prof. Dr. Faroş Yarman Vural will have this role. Scrum Master is the person who is responsible for enforcing the rules of the process and ensuring that the process is used as intended. The responsible for this role will be one of the team members, Bahattin Tozyılmaz. Finally, the Scrum Team is defined as the anyone who is involved in the realization of this project.

The three ceremonies are defined as Sprint Planning, Sprint Review, and Daily Scrum Meeting. In the beginning of the sprints, our team will hold a Sprint Planning session, after that, we will work on the requirements of the sprint by supporting them with Scrum Meetings and finally we will review the sprint. Sprints will roughly cover a month; however, Scrum Meetings will not be held daily since all the team members are under tight schedule for their other work.

Finally, the three artifacts are called Product Backlog, Sprint Backlog, and Burndown Chart. The Project Backlog is a prioritized list of project requirements with estimated times to turn them into parts of project deliverables and/or completed system functionality and the Sprint Backlog a list of tasks that defines the team's work for each Sprint. We will record our work activities to the Burndown Charts in order to estimate the time that will be required by the remaining work and to see the whole picture about our project.
7 CONCLUSION

This Software Requirement Specification document is prepared to give requirement details of the project "Cognitive State Representation and Visualization of Human Brain". Tool's description and functionalities are specified with some details. Interaction between the user and system is defined. Scheduling timeline and task sharing between team members are arranged. Also, technologies to be used and additional researches to be done are understood by team members.