3D Web Visualization of Continuous Integration
Big Data

University of Oulu
Degree Programme of Information Processing Science
Master’s Thesis
Rubini Matta Santharam
27.11.2018
Abstract

Continuous Integration (CI) is a practice that is used to automate the software build and its test for every code integration to a shared repository. CI runs thousands of test scripts every day in a software organization. Every test produces data which can be test results logs such as errors, warnings, performance measurements and build metrics. This data volume tends to grow at unprecedented rates for the builds that are produced in the Continuous Integration (CI) system. The amount of the integrated test results data in CI grows over time. Visualizing and manipulating the real time and dynamic data is a challenge for the organizations.

The 2D visualization of big data has been actively in use in software industry. Though the 2D visualization has numerous advantages, this study is focused on the 3D representation of CI big data visualization and its advantage over 2D visualization. Interactivity with the data and system, and accessibility of the data anytime, anywhere are two important requirements for the system to be usable. Thus, the study focused in creating a 3D user interface to visualize CI system data in 3D web environment.

The three-dimensional user interface has been studied by many researchers who have successfully identified various advantages of 3D visualization along with various interaction techniques. Researchers have also described how the system is useful in real world 3D applications. But the usability of 3D user interface in visualizations in not yet reached to a desirable level especially in software industry due its complex data.

The purpose of this thesis is to explore the use of 3D data visualization that could help the CI system users of a beneficiary organization in interpreting and exploring CI system data. The study focuses on designing and creating a 3D user interface for providing a more effective and usable system for CI data exploration. Design science research framework is chosen as a suitable research method to conduct the study. This study identifies the advantages of applying 3D visualization to a software system data and then proceeds to explore how 3D visualization could help users in exploring the software data through visualization and its features. The results of the study reveal that the 3D visualization help the beneficiary organization to view and compare multiple datasets in a single screen space, and to see the holistic view of large datasets, as well as focused details of multiple datasets of various categories in a single screen space. Also, it can be said from the results that the 3D visualization help the beneficiary organization CI team to better represent big data in 3D than in 2D.

Keywords
Visualization, 3D graphics, 3D web, WebGL, three.js, Continuous Integration, big data, 3D user interface.

Supervisor
PhD, Dorina Rajanen
Foreword

I would like to take this opportunity to express my gratitude to number of people who supported me on this research journey. I owe my sincerest gratitude to my supervisor Dr. Dorina Rajanen for her continuous support in research and thesis writing process. I am truly grateful for your time, endless supply of guidance and knowledge provided by you Dorina. I would like to thank Pekka Tuttila and Matti Vanhanen for their support in the organization and for their input ideas while implementing the thesis.

This master thesis work has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737494. This Joint Undertaking receives support from the European Union’s Horizon 2020 research and innovation programme and Sweden, France, Spain, Italy, Finland, Czech Republic.

I would specially thank my husband Balakumar and my son Shashvath for their excellent support and understanding me in this research journey. I am very grateful for their motivation in my difficult times.

Rubini Mattasantharam

Oulu, November 12, 2018
Contents

Abstract ......................................................................................................................... 2
Foreword ......................................................................................................................... 3
Contents ......................................................................................................................... 4
Abbreviations ............................................................................................................... 5
List of Tables ............................................................................................................... 6
List of Figures ............................................................................................................. 7
1. Introduction .............................................................................................................. 8
2. Research background and related work ................................................................. 11
   2.1 Data visualization ............................................................................................... 11
   2.2 Overview of visualization techniques ................................................................. 13
   2.3 Advantages of 3D visualization techniques ......................................................... 16
   2.4 3D web visualization ......................................................................................... 17
   2.5 Data visualization process .................................................................................. 18
   2.6 Reference model for JavaScript web-based visualization .................................... 20
   2.7 3D graphics technology on the web .................................................................... 21
   2.8 3D user interface ............................................................................................... 21
   2.9 Terminology ...................................................................................................... 23
3. Research methods .................................................................................................... 25
   3.1 Design science research approach ..................................................................... 25
   3.2 Implementation of DSR methodology .............................................................. 27
4. Architecture and Implementation ........................................................................... 29
   4.1 System architecture ............................................................................................ 29
       4.1.1 Developer’s view of system architecture ..................................................... 29
       4.1.2 Programming languages and software tools .............................................. 30
       4.1.3 Data processing model .............................................................................. 31
       4.1.4 GUI design components .......................................................................... 31
       4.1.5 User interaction ......................................................................................... 32
   4.2 Basic components of 3D graphics ...................................................................... 32
       4.2.1 Basics of coordinate system ..................................................................... 32
       4.2.2 Rendering .................................................................................................... 34
       4.2.3 Scene and camera ..................................................................................... 34
       4.2.4 Mesh, texture, material, light ..................................................................... 34
       4.2.5 Grid and axes .............................................................................................. 35
       4.2.6 Camera ........................................................................................................ 35
   4.3 Implementation .................................................................................................... 35
       4.3.1 Data collection and pre-processing ............................................................... 36
       4.3.2 GUI implementation ................................................................................... 36
       4.3.3 JavaScript library for the graphical presentation ......................................... 38
       4.3.4 Implementing the line charts for data visualization .................................... 40
5. Findings ................................................................................................................... 42
   5.1 Summary evaluation of data interpretation ......................................................... 43
   5.2 Summary evaluation of the user interface ........................................................... 44
6. Discussion and implications ..................................................................................... 46
7. Conclusions ............................................................................................................. 48
References ................................................................................................................. 50
Appendix A.1 Screenshots ......................................................................................... 56
## Abbreviations

The following is a list of abbreviations used frequently in this thesis.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two-Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three-Dimensional</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>CI</td>
<td>Continuous Integration</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>DOM</td>
<td>Document Object Model</td>
</tr>
<tr>
<td>DSR</td>
<td>Design Science Research</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphics Processing Unit</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HTML</td>
<td>Hyper Text Markup Language</td>
</tr>
<tr>
<td>IS</td>
<td>Information systems</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>SVG</td>
<td>Scalable Vector Graphics</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>VRML</td>
<td>Virtual Reality Modelling Language</td>
</tr>
<tr>
<td>WebGL</td>
<td>Web Graphics Library</td>
</tr>
<tr>
<td>X3D</td>
<td>eXtensible 3D</td>
</tr>
</tbody>
</table>
List of Tables

Table 1. Classification of the information visualization techniques (Keim, 2002) ....... 14
Table 2. Types of visualization techniques (based on Keim, 2002; Marghescu, 2007) . 15
Table 3. Hevner’s seven guidelines (Hevner et al., 2004). .......................................26
Table 4. Summary evaluation of data interpretation.....................................................43
Table 5. Summary evaluation of the data exploration .................................................44
List of Figures

Figure 1. Relation between data visualization and computer graphics (Stares, 2016) ... 12

Figure 2. A schematic diagram of the visualization process (Ware, 2004 as reproduced by Voinea, 2007) ........................................................................................................................................ 19

Figure 3. Reference model for visualization (Card et al., 1999) ............................................. 19

Figure 4. Reference model adapted for the web-based visualization (adapted from Card et al., 1999 & Guldamlasioglu, 2015) .................................................................................................. 20

Figure 5. Viewport in interactive computer graphics (Angel and Shreiner, 2014) ................. 23

Figure 6. Difference between perspective projection and orthographic projection of a 3D object (Marland, 2015) ............................................................................................................ 24

Figure 7. The research cycles in design science research (Hevner et al., 2004) ................. 25

Figure 8. The adapted design science nominal process model (Peffers et al., 2007) ... 27

Figure 9. Architecture of the implemented visualization system ........................................ 29

Figure 10. Right handed coordinate system ........................................................................ 33

Figure 11. Cube geometry shows edges, vertices, and faces ............................................. 34

Figure 12. Prototype for the 3D visualization of data .......................................................... 37

Figure 13. Screenshot for different use of colors in visualization ........................................ 37

Figure 14. Scene node map and its flow to renderer using three.js (Lyons, Dec 13) .... 39

Figure 15. Screenshot of spline charts .................................................................................. 40
1. Introduction

The modern agile and lean methodologies play a major role to speed up the software development process. The agile process aims for smaller and faster software releases through iteration. A software needs to go through the iteration processes that helps to respond quickly to the change in customer requirements. The rapid changes to software may affect the previously developed functionality and can cause errors. (Hukkanen, 2015; Kung, 2013).

Continuous integration at its core is a set of principles that can be applied to the daily working flow of development teams in software organizations (Meyer, 2014). The Continuous integration build process helps the software development team to release a quality code faster that helps to improve and maintain the quality of the product in every release. (Ravello, 2014). The term “CI system” is used here is to refer the organization system that integrates data produced by the CI team.

Fowler (2006) defined Continuous Integration (CI) as being a development practice in software development industry where members of a team integrate their code into a shared repository frequently. Every change in the code is verified by build process and tested by CI system before releasing the software (Fowler, 2006). This testing process helps the development team to detect bugs and fix the bugs in an early stage. This process helps the software organisation to have shorter and frequent release cycle. Also, this process helps to improve the software quality and increase team’s productivity. This practice includes automated software building and testing. (Shahin, Baba, Zhu, 2017).

The rapid change in the software development which undergoes the build process of CI system integration results in producing a huge amount of data in the CI system and moreover the accumulated data have various shapes and structure. The shapes and structure of data can be identified by projecting the dataset in the screen using some visualization techniques. (Vantti, 2018).

Bonyuet, Ma, Jaffrey (2004) state that software developers need the ability to see the data in a meaningful way such as graphs than in numerical way which could help them to do the analysis faster (Bonyuet, Ma, Jaffrey, 2004).

Data visualization can help to present the CI big data in an informative and effective way. The 2D visualization have been successfully applied in various complicated datasets, but when the data is represented in a 2D format it requires more screen space with increasing number of items to display on the screen. The 2D format of big data may contain hidden objects of visualization which requires more screen space to display the data in all possible views. (Dübel, Röhlig, Schumann & Trapp, 2014).

Though the dataset can be visualized in 2D, adding one more dimension called depth allows us to view the dataset in more than one angle in a single screen space. (See Appendix A.1 & A.2 to view data different angles). Moreover, 3D visualization over the web increases the accessibility of the dynamic data anywhere and the users has no need to install any software. (Bröring, 2014).

Shepherd (2008) in his study mentions that the advantage of 3D visualization is displaying objects by using less screen space, while 2D visualization requires more screen to display various categories of data. It also provides access to the hidden variables and their relations with other objects (Herman & Reznik, 2015).
To resolve the space problem and to view hidden data objects of 2D visualization, the 3D visualization technique comes in to show various perception of the big datasets by utilizing less the screen space. The data objects that are hidden in 2D visualization can be visible by projecting the data set along with the third dimension called depth, preferably in a 3D environment. A 3D environment is a 3D graphical view usually implemented in web interface via a 2D screen, and it helps us to see the holistic view of the dataset in various angles.

Interactivity is regarded as one of the most important aspect of data visualization. In this thesis work some basic interactive techniques is implemented such as interactive filtering, interactive projection, view rotation, zoom, linking and brushing in 3D web environment which can be more intuitive to the users. The interaction techniques are explained later in chapter 2.

This thesis work is focused on investigating how CI system data can be visualized in 3D environment using 3D graphic tools to support tasks such as data exploration and interpretation. Thus, the thesis addresses the following research questions:

1. How 3D visualization can help to improve the interpretation of Continuous Integration data?

2. How 3D visualization could be designed in interactive web to better support exploration of CI system data?

The term interpretation in the research question is about getting meaningful interpretation from the data representations and exploration is about getting access to different views of the data to facilitate the interpretation.

After studying through various research methods, design science research was considered as the suitable research approach to answer the above research questions. The design science research approach is explained in more detail in chapter 3. The focus of this thesis is to visualize the geometry and semantics of the CI system test data in a 3D web environment with the help of available 3D web graphic technologies.

The agile software development involves three development activities (Continuous Integration, Continuous Delivery, and Continuous Deployment). The data produced by the CI system includes meta-data (start & end times of the test build process, causes for the triggering the builds) of the builds, logs of builds. In addition to these above-mentioned data it is also important for the CI team to analyse the different kind of build failure metrics. (Hukkanen, 2015).

In multi-national software organizations, the developers are in different time zones. Thus, 3D visualization on the web has become an essentially important factor for the CI team to share and view the visualization of the CI system data.

The major advantage of web-based approaches is that they allow easy collaboration and sharing of visualization tools among different teams (Jern, 1998). Early techniques such as X3D technologies have been introduced in the past to produce 3D scenes on the web with help of plugins. (Behr, Escher, Jung, Zolner, 2009).

In this thesis, I focus on building a 3D user interface (UI) proof of concept for visualizing CI data by using 3D web graphics, by using the three.js library which is built on top of WebGL (see chapter 4). The main advantage of WebGL is that it also supports mobile platforms such as Android and iOS. The 3D graphical user interface (GUI) is a
type of interface that allows users to interact with the objects in the 3D environment. The 3D GUI acts as a medium of communication which allows users to perform 3D interaction on the computer. Bowman, Kruijff, Joseph, LaViola (2004) define 3D interaction as being “human-computer interaction in which user’s tasks are performed directly in 3D spatial context” (Bowman, Kruijff, Joseph, LaViola, 2004).

This thesis has the following structure. In chapter 2, I present the key concepts of data visualization and I review prior literature regarding data visualization, 3D data visualization, and 3D data visualization in web. I describe also the data visualization process and its reference model in detail. Also, this chapter describes related work on how 3D user interfaces should be designed.

In chapter 3, I describe the research methods used in this thesis. I focus on presenting the characteristics of Design Science Research and how it has been adopted for this study. In chapter 4, I describe about the architecture and implementation of the thesis work. Then I introduce the 3D graphics basic concepts, the libraries, technologies used, and I also describe how implementation of the thesis work is carried out in detail. In chapter 5, I discuss the results obtained from the conducted study and future work is suggested. In chapter 6, I describe the implications. In chapter 7, I conclude the thesis with final remarks.
2. Research background and related work

In this chapter, I review various literatures about data visualization. I will go through the literature about data visualization process and various visualization techniques to visualise data in the 3D environment. Also, this chapter describes the current state of data visualization in the web environment.

2.1 Data visualization

The 19th century inventions acted as a fertilizer to the beginning of modern graphics with solid growth in statistical graphics and thematic mapping. The thematic maps, such as continuous shading to show quantitative information on maps, came in use by the economic and state planning. Though the rapid growth of visualization went down in 1900’s, a stable growth of data visualization is seen after 1975 with advancement in technology (Friendly, 2008). The big changes in visualization happened in 20th century after the development of computers (Mardiney, 2016). Lohr (2012) expresses that information is continuing to accumulate and is being collected at an increasing rate.

Card, Mackinlay, & Shneiderman (1999) defined visualization as “The use of computer-supported, interactive, visual representations of data to amplify cognition”. These visual graphical representations convey the complex ideas with clarity. Shneiderman (1996) described that a simple and easy understandable visual graphic of data shows the power of visualization. Murray (2012) describes data visualization as the process of mapping information to visualization. Data visualization plays a key role by helping the users to understand larger datasets visually. Also Ware (2004) summarizes the advantages of visualization as follows:

- “Visualization provides an ability to comprehend huge amounts of data,
- Visualization allows the perception of emergent properties that were not anticipated,
- Visualization facilitates understanding of both large-scale and small-scale features of the data,
- Visualization facilitates hypothesis formation.”

Data visualization consists of two main fields of study, namely information visualization and scientific visualization.


From the above definitions, we can understand that information visualization focuses on abstract data which is a nonphysical data such as text and statistical data, whereas scientific visualization focuses on spatial data that are generated by scientific processes that covers visualizations of the real world (Rhyne, 2003; Peikert, 2007).

Figure 1 below illustrates the relationship between data visualization and computer graphics. Based on the views of Stares (2016) it can be said that the data visualization is
the process to represent the information clearly and effectively by encoding it as visual objects by using computer graphics.

![Diagram showing the relation between data visualization and computer graphics](image)

**Figure 1.** Relation between data visualization and computer graphics (Stares, 2016)

Paredes, Anslow & Maurer (2014) also refer to information visualization as being “the use of visual representations of abstract data to amplify cognition”. Chen (2017) defines abstract data as data that comes from an analysis of some type of data. Also, Chen remarks that the information visualization brings in two important aspects. The first one is, it can be used to discover new insights and knowledge from abstract data in the form of graphical representation and the second one is the representation of data that amplifies cognition.

Nielson (1996) describes that the term information visualization “has been adopted as a more specific description of the visualization of data that are not necessarily representations of physical systems which have their inherent semantics embedded in three-dimensional space” (Nielson, 1996).

Voinea (2007) describes that software visualizations are a “specialized branch of information visualization”, where the visualizations are based on the artifacts of software and its development process (Voinea, 2007).

As mentioned above from the previous literature, information visualization deals with abstract data. In this study, the data is collected from CI system software which is employed in a multi-national organization. This indicates that the study deals with information visualization of a software system. Thus, this study is also relevant from the software visualization perspective.

As shown in Figure 1 and based on the above definitions it can be said that information visualization is a subset data visualization. The data visualization is a process that involves gathering of relevant data and giving it a visual representation, which is useful for decision making, gaining insight and exploratory analysis. The visual representation of the data can be presented to the users using computer graphics. Applying computer graphics to the visualization can help to convey ideas effectively to the end user in aesthetic and intuitive way. Visualization methods such as line charts, bar charts, and scatterplots are being used to represent the data graphically and where graphics is applied to draw and render the images such as line and bar charts in a 3D format.
The data visualization can be presented in 2D and 3D format. As mentioned earlier, the visualization environment is a web-based 3D environment, thus the study focuses on 3D visualization. The following paragraphs describe the 3D web-based graphics tools and their applications.

Previously various application programming interfaces (APIs) such as Flash, and x3D VRML, have been developed to display 3D graphics content on the web for example if a user installs flash plugin it can enable web browsers to display the multimedia content such as 3D models, images and animations. VRML was considered as the major development of towards interoperability in 3D data in web environment. After VRML HTML5 turned out to be the important milestones in 3D graphics which contains the several useful elements such as canvas and scalar vector graphics (SVG). This milestone is extended that paved the way for WebGL, which is an extension of HTML5 canvas element which is now widely used in the field of web-based 3D visualization. (Chaturvedi, 2014).

The visualization is implemented in a 3D web environment that allows users to interact with the visualization system anytime, anywhere. For this purpose, in this thesis, I have used 3D web computer graphic tools that can be used to render the 3D images and present visually the CI software system data.

Teyseyre & Campo (2008) mentions that software visualization is a combination of techniques from different areas like software engineering, data mining, human computer interaction, information visualization. Thus, this thesis has conducted the analysis of software data visualization in 3D environment by combing the various 2D and 3D visualization techniques.

### 2.2 Overview of visualization techniques

To convey the information clearly and effectively to the end users, a good data visualization technique is required to display the data which holds a key to good impact. Data visualization comprises various visualization techniques which are used to communicate data or information by encoding it in a visual representation such as line charts, scatterplots and bar charts along with some interaction techniques. The main goal of data visualization must be to convey the information clearly and effectively to the end users. (Elgendi, 2017).

Keim (2002) classifies the visualization techniques according to three criteria as follows: (1) the data type to be visualized, (2) the visualization technique, and (3) the type of interaction and distortion technique. I will go through these visualization techniques used in this study in the following paragraphs. Table 1 shows the classification of visualization techniques according to Keim (2002). In the table, the techniques implemented in this thesis are highlighted with the bold.
The data type refers to the complexity of the data to be visualized (Shneiderman, 1996). Every dataset is characterized by the number of variables it represents, and these variables are called dimensions. Based on the number of dimensions in a dataset that can be visualized at once, the visualization techniques can be categorized into one-dimensional, two-dimensional, three-dimensional, and multidimensional (Marghescu, 2008, p. 29). One-dimensional data usually has one dense dimension, while the two-dimensional data has two distinct dimensions. Multidimensional data contains three or more dimensions, thus also three-dimensional data belongs to this category (Keim, 2002). Thus, the data type addressed in this thesis is a multidimensional data.

In Table 1 I have also highlighted the algorithms and software technique to mention that the data visualized in this study is a software data. Keim (2002) mentions that the visualizations support software development as follows: 1) It shows the flow of information in a program which helps to understand algorithms; 2) It enhances the understanding of written source code by representing the structure as graphs; and 3) It supports the programmers in debugging the code, for example by visualizing the errors. In this thesis, I focus on techniques that support the visualization of three-dimensional data that represent CI system data, which is a type of software data.

Here, I would like to describe the datatype of CI system data that is implemented in this thesis. The build process of CI system produces various types warnings and error information. There is different type of warnings that occur in every build. This indicates that there are different number of warning variables produced for every product. The visualization is more useful to the users when it is visualized along with date time series. This output of CI system results refers to a multidimensional data with minimum three variables such as warnings, products and datetime. The CI system data is described more detailed in chapter 4. In the following paragraphs I will discuss about the visualization techniques that are implemented in this thesis.

There are large number of visualization techniques available that can be used for visualization the data. Keim (2002) identified five categories of visualization techniques such as standard 2D/3D displays, geometrically transformed displays, icon-based

---

**Table 1. Classification of the information visualization techniques (Keim, 2002)**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Specific Techniques*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data type to be visualized</td>
<td>One-dimensional data</td>
</tr>
<tr>
<td></td>
<td>Two-dimensional data</td>
</tr>
<tr>
<td></td>
<td><strong>Multidimensional data</strong></td>
</tr>
<tr>
<td></td>
<td>Text and hypertext</td>
</tr>
<tr>
<td></td>
<td>Hierarchies and graphs</td>
</tr>
<tr>
<td></td>
<td><strong>Algorithms and software</strong></td>
</tr>
<tr>
<td>Visualization technique</td>
<td><strong>Standard 2D/3D displays</strong></td>
</tr>
<tr>
<td></td>
<td>Geometrically transformed displays</td>
</tr>
<tr>
<td></td>
<td>Icon-based displays</td>
</tr>
<tr>
<td></td>
<td>Dense-pixel displays</td>
</tr>
<tr>
<td></td>
<td>Stacked displays</td>
</tr>
<tr>
<td>Interaction and distortion technique</td>
<td>Interactive projection</td>
</tr>
<tr>
<td></td>
<td>Interactive filtering</td>
</tr>
<tr>
<td></td>
<td>Interactive zooming</td>
</tr>
<tr>
<td></td>
<td>Interactive distortion</td>
</tr>
<tr>
<td></td>
<td><strong>Interactive linking and brushing</strong></td>
</tr>
</tbody>
</table>

Note: * With bold I marked the types of visualization techniques addressed in this thesis.
displays, dense pixel displays, and stacked displays. Here, visualization techniques refer to way data is represented graphically (Marghescu, 2008, p. 29).

The following Table 2 shows different types of visualization techniques, a brief description, and examples of each type based on the review by Marghescu (2007) and Keim (2002). These visualization techniques differ with respect to the data dimensions displayed and data arrangement on the screen.

**Table 2.** Types of visualization techniques (based on Keim, 2002; Marghescu, 2007)

<table>
<thead>
<tr>
<th>Type of technique</th>
<th>Brief description and examples</th>
</tr>
</thead>
</table>
| Variations of standard 2D/3D displays | This class of visualization technique consist of most popular techniques which is very effective to present two - or three-dimensional data. (x-y)-z plots, bar charts, pie charts line graphs belong to this category.  
Example: Multiple line graphs. |
| Geometrically-transformed display | Techniques that are used to find “interesting” transformations of multidimensional datasets (Keim, 2002, p. 103).  
Examples: Scatter-plot matrix, Parallel coordinates. |
| Iconic display | Techniques that can map the attributes of the data values to features of icon. It is a symbolic representation of the data attributes such as star and cone icons.  
Example: star glyph, stick figure icons. |
| Dense pixel display | The idea of dense pixel techniques is to “map each dimension value to a colored pixel and group the pixels belonging to each dimension into adjacent areas” (Keim, 2002, p. 103).  
Example: pixel bar charts. |
| Stacked display | Various techniques are tailored together to present data partitioned in hierarchical manner.  
Examples: Embedding one coordinate system into another coordinate system, Tree maps, and cone trees. |

For this study, the **standard 2D/3D display** was considered as the most suitable visualization technique for representing the data in a 3D web environment. This technique is very effective to present one-, two- and three-dimensional data on a standard 2D or 3D display. Examples of such techniques are line graphs, pie charts, bar and column charts. In this thesis, I employ multiple line graphs to represent more than two CI data variables at a time. Multiple line graphs have been evaluated in previous studies (e.g., Marghescu, 2007) and shown that are effective for data exploration and uncovering patterns such as outlier detection, dependency analysis, comparison between data points, and classification.
Interaction and distortion techniques are employed to enable the user interaction with the data and system. The interaction techniques support the transformation of the data and visualizations as per the user exploration objectives. Basic interaction operations of visualization environment include interactive projection, filtering, zooming, linking and brushing. The distortion techniques refer to representing the data in such a way to enable focusing on a part of the data while preserving an overview of the data (Keim, 2002). In the following paragraphs I have explained the techniques such as interactive filtering, projection, zooming, linking and brushing that have been employed in this thesis.

Interactive filtering referred as “partition the data set into segments and focus on interesting subsets” (Keim, 2002). This definition by Keim (2002) is referred in this study and implemented via dropdown menus, where the user can filter what data to visualize from the list of available options from the dropdown menu. This implies that dataset partitioned into list of options allows the user to focus on subsets of dataset.

Interactive projection is to interactively change the projections on the screen, to explore various multidimensional data set. The interactive projection is achieved by filtering out which dataset of items are to be displayed on the screen viewport. In this study this technique is implemented by allowing the users to filter the data from the dropdown menu, which will be projected on the screen. The visualization of the datasets will be updated based on the user’s choice of dataset to be visualized.

Interactive zooming is referred as a selection and manipulation technique, when we are dealing with large amount of data, a default highly compressed form of dataset will be presented to provide an overview of entire dataset. Zooming technique does not only allows to display the object larger but also allows to view a variable display of data at various resolutions (Keim, Panse, Sips, 2003). In this thesis work zooming technique has been employed for example: To explore a specific product/component of interest the user can be zoom in by using mouse wheel, which shows the results in detailed form of dataset with better resolution.

Interactive brushing and linking referred as a selection technique while brushing is an interactive selection process and linking is a process which is used for communicating the selected data to the other views of the data set. For example, in this thesis the lines charts are drawn by connecting the data points which are grouped by colouring and linking all subset of points. When the user clicks on a data point the information of the corresponding data point will be displayed to the user.

2.3 Advantages of 3D visualization techniques

A variety of visualization systems have been developed to visualize various datasets. There are also many studies that compare 2D and 3D visualizations. In 2D versus 3D comparison, 2D visualization is superior to 3D visualization in terms of accuracy of reading data for graphs (Russell and Bielewicz, 2005).

However, other studies have shown that 3D visualizations are helpful in exploratory tasks. Teyssereyre and Campo (2008) described that by introducing the third dimension, 3D approaches try to create visualizations that are closer to real world objects and try to reduce the usage of the screen space. 3D provides a depth information when comparing it to 2D (Dolatabadi, Dargazany & Nouri, 2014).
Müller and Zeckzer (2017) in their findings mention that multiple views of a dataset is an important functional requirement in 3D software visualization which provides a holistic view of the system i.e., structure behaviour and evolution.

Marcus, Feng and Maletic (2003) refers to the results of various articles that describes benefits of 2D over 3D where authors indicates the following statements.

- “Displaying data in three dimensions instead of two can make it easier for users to understand the data.”
- “The error rate in identifying routes in 3D graphs is much smaller than 2D.”
- “3D representations have been shown to better support spatial.”
- “The use of 3D representations of software in new mediums, such as virtual reality environments, are starting to be explored.”

Teyseyre (2002) mentions that 3D graphical presentations provide “greater information density than 2D presentations because of a bigger physical space”. In addition, 3D graphical presentations enable a “clear perception of the relations between objects by integration of local with global views”. Also, 3D graphical presentations provide “a composition of multiples 2D views in a single 3D view”. Moreover, 3D graphics helps to represent the objects that are closer to real world objects.

Based on the above literature reviews and after studying the advantages of 3D graphical visualization, the thesis builds upon the research work to show that 3D data visualization gives better, and richer overview of data compared with 2D visualization.

2.4 3D web visualization

Rosling (2010) described how modern technology can be used to visualize trends in large data sets. Though it is a good example, his illustration was lacking in mentioning the platform where and how people could interact with data.

Behr, Escher, Jung & Zöllner (2009) highlight that 2D web and various 3D environments developed in parallel. Custom software technologies such as 3D graphics are now being used via web, while efforts have been made to implement real time 3D graphics over the internet (Evans, Romeo, Bahrehmand, Agenjo & Blat, 2014).

SVG injection, Canvas 2D and WebGL are the three ways to draw graphics inside web browser with web standards. In Scalable Vector Graphics (SVG) the nodes will be injected into the Document Object Model (DOM) using JavaScript which is then interpreted and drawn by the web browser. The HTML5 Canvas 2D allow us to draw graphics on a defined area of the webpage by using a set of 2D drawing primitives. WebGL is the extension of canvas 2D. WebGL is a 3D rendering JavaScript API which provides a 3D context for the HTML5 Canvas element (Andrews and Wright, 2014). Due to its flexibility, JavaScript is considered as the most powerful and default choice for web-based developments or web-based visualizations. (Guldamlasioglu, 2015).

The advancement in web technologies and as a current trend in web visualization HTML5 and WebGL technologies have opened the door to visualize the 3D objects in the web browser without any plugins (Jern, 1998).

Many types of fancy charts are now possible to implement with the help of computer graphics tools such as Blender, 3ds Max and with the help of programming languages
such as R and Python. (Glaab, Garibaldi, Krasnogor, 2010). In this study, I have employed Python scripts for collecting the data and three.js 3D graphics tool to present the data visually in 3D web environment.

In today’s internet of things world, web browsers are widely used as a common interface in any information technology infrastructure. Web browsers are widely used to share and retrieve information in the internet infrastructure. The advancement in browser technology has made it possible to render complex visualizations in web, thus web become an apt platform to reach out large audiences (Rizvi, 2017). In addition to the above, interactive user interface will enable users quickly to insight and explore the data in more visual way. (Keim, 2002). The user interface and interactivity will be discussed more in the following sections.

In this study, the data is characterized based on the above-mentioned information from prior literature. Firstly, the data to be visualized is a CI system software data which is employed in an organization. Secondly, the dataset had a minimum of three variables such as time, products and warnings or error information. The type of visualization technique employed is the standard 2D/3D display and more specifically it is focused on multiple line graphs.

The environment for the visualization system is a web interface designed specifically to allow 3D representation of the data and user interaction with the data and system. Though there are many standalone software programs available for data visualization, they often require a special hardware, complex installation process or lot of training is needed to understand the software. Toman and Kos (2002) mentions that the 3D web environment eliminates the requirements for hardware with no need of installation process and software setup. The real time 3D web environment through common web browser makes an application available to more number of users and helps to reduce the operating costs. (Toman and Kos, 2002).

Along with visualization techniques applying the interaction techniques make a system more effective. Interactive operations such as navigation, selection and manipulation techniques are important to present some information with more details. In this analysis interaction techniques such as interactive projection, interactive filtering, zoom, linking and brushing have been implemented in this system to make the system interactive.

Jankowski and Hachet (2013) define interactive 3D environments as “computer representations of real world or imaginary spaces through which users can navigate and in which they can interact with objects in real time”. Various interaction techniques are explained in more detail in section 2.8.

2.5 Data visualization process

Data visualization is about presenting the data graphically to enhance cognition such as understanding, insight, decision making through data exploration and communication. A process flow of how data visualization is produced and employed helps to analyse different visualizations systematically. In his book, Ware (2004) divides the data visualization process into four steps: 1) collection and storage of data, 2) pre-processing and converting the data into an easily readable format such as JSON and XML, 3) display the data on the screen, and 4) perceive the data. (See Voinea, 2007).
As shown in Figure 2, as an initial step in data visualization, the data must be gathered from the related environments. Once it is collected it needs to be preprocessed and transformed into organized data format. In this stage the data entities need to be categorized and associated with its attribute values. Graphic engines can help to map the data into visual form. Once the visual form is developed, the presented visual view need to present transformation with interaction techniques such as zoom, view rotation based on users input. Ware (2004) describes that the physical environment is “the source of data”, while the social environment is described as the way the data is “collected and interpreted”.

Ware’s (2004) information visualization process is useful to understand how information visualization is constructed from raw data to interpretation but lacks in “providing a platform to compare and contrast different information visualization systems”. An alternative model to describe the process of information visualization is provided by Card et al. (1999). Card et al. (1999) created a reference model which defines the milestones in information visualization process. A reference model can help to create a framework for the developers who build new visualizations.

As shown in Figure 3 in Card et al.’s (1999) reference model, the first step is to convert the raw data into structured data and thus obtaining data tables which include metadata. The next step is the conversion of data tables into visual structures. Visual structures are
created based on the characteristics of the data that undergoes visual mappings. In this stage, the mapping between data variables and the abstract structures can determine, for example, the x, y, z coordinate points. The graphical view can be acquired by view transformations from the visual structures. (Guldamlasioglu, 2015).

Guldamlasioglu (2015) mentions that the advantage of using a reference model is its reusability, which leads to rapid development. Implementers can create new visualizations if they understand the interactions between the data and the operations. Also, if they understand how to apply and implement the visualization techniques they can reuse the different parts of the existing system while constructing the new visualizations. (Guldamlasioglu, 2015).

2.6 Reference model for JavaScript web-based visualization

JavaScript (JS) is considered as the programming language of the web. JS can manipulate the content of a web page through DOM objects. In JavaScript most of the things are referred as objects. JavaScript has a powerful object notation called JSON which is a standard text format for strong and transporting data. JSON is a data format representing structured data which is commonly used for sending and receiving data between server and client. Guldamlasioglu (2015).

To apply the Card et al.’s (1999) reference model in the context of web visualization it is important to understand how JavaScript properties can be applied in data visualization. Based on the views of Guldamlasioglu (2015) reference model for JavaScript based visualization and Card et al. (1999) reference model for visualization, the following Figure 4 is reproduced to represent the web-based JavaScript reference model which is adapted for this thesis work.

![Figure 4. Reference model adapted for the web-based visualization (adapted from Card et al., 1999 & Guldamlasioglu, 2015)](image)

As a first step in the visualization process, the raw data will be transformed into a structured text format using a Python script i.e., the data is transformed into JSON format. JSON data format is referred here as a data table which is a structured list of attributes of the metadata. The next step is to implement the visual mapping of data that is the results in creating the visual structures of the data by applying the graphical elements such as points, lines and volumes that can be interpreted. WebGL’s three.js JavaScript library is being used in this case to draw the graphical elements such as points and lines. Next, the views are formatted with the WebGL graphical elements.
along with the HTML 5 and CSS3 elements, which gives users a visual presentation of the data. The final step is an interactive data visualized environment along with interaction techniques applied called views, where the users can perform the interactions such as zooming, linking and brushing. (Guldamlasioglu, 2015).

2.7 3D graphics technology on the web

Though there have been many plugins developed such as Adobe Flash and Java3D, the Virtual Reality Modelling Language (VRML) is the language that is used to present the 3D scenes on the web. VRML was replaced by X3D which added more support to binary and XML formats. To render the 3D on web browsers, plugins are needed to integrate the X3D (Evans et al., 2014). The incorrect choice of plugins has alienated the users and thus the use of browsers plugins is not satisfactory for the users (Behr et al., 2009).

To overcome above mentioned issue, Behr et al. (2009) introduced X3DOM a plugin free declarative 3D in browsers to present the 3D scenes on the web. The X3DOM elements were deployed in browsers by DOM technique (X3DOM). Waerner (2012) in his evaluation of 3D graphics technologies mentioned that WebGL and X3DOM are the good choices and suitable for 3D graphics on the web browsers where WebGL has the support for mobile devices as well. Mwalongo, Krone, Reina, Ertl (2016) mention that Graphics Processing Unit (GPU) based visualization techniques can help to improve the rendering performance and interactivity as the GPU could take care of the expensive computations rather than on the rendering on Central Processing Unit (CPU).

The introduction of WebGL and HTML5 technology have made 3D visualization possible on the web browser which does not require any installation of plugins. The WebGL has made a drastic progress in producing 3D Web Graphics. The WebGL API has gained popularity in a short time as it can be programmed directly via the browser using JavaScript (Evans et al., 2014). The inclusion of 3D graphical elements can make the user interface aesthetically appealing which could also ease the perception of the human visual system (Teyseyre and Campo, 2009). Additionally, WebGL is a low-level JavaScript web API that provides the hardware accelerated graphics functionality on the web. To avoid low level programming and for the ease of development several JavaScript libraries such as three.js, scene.js have been developed. (Chaturvedi, 2014).

Among the available WebGL libraries, three.js library was chosen in this thesis due to its simplicity and ease of use. Three.js is a JavaScript library that makes 3D graphics to be implemented easily in the browser without the installation of plugins. To render the graphics on the web, three.js uses WebGL API that uses the GPU to perform computations of the geometry. WebGL and three.js together allow one to create games, visualizations and 3D effects inside the browser. In this CI case study, the visualization should be designed effectively with a high-quality representation of objects in the 3D web environment. Additionally, the visualization should be designed to provide different level of visualization details.

2.8 3D user interface

Bowman, Kruijff, LaViola & Poupyrev (2005) define “A user interface acts as a medium of communication between user and computer system” that is a user interface act as a channel between users and the technology irrespective of 2D or 3D. Bowman et
al. (2005) has defined 3D User Interface as “UI that involves 3D Interaction” which he refers to types of user interaction with the dataset in 3D environment.

Bowman et al. (2005) also mentions that the 3D interaction is a task performed by the user in a 3D environment via the input devices such as mouse and keyboard that involves three dimensions called as width, height and depth. (Herman & Reznik, 2015).

According to the study of Väänänen (2014). The popularity of the 3D interface elements is lagging due to the variations of plugin support on different browsers. The efforts of 3D interface development have not been very active as there are some readymade 3D applications available for their desired purpose (Väänänen, 2014).

Bowman et al. (2004) refers to user interaction in 3D environment as “3D interaction techniques involve techniques such as navigation, and selection and manipulation”. In this thesis work the 3D interaction refers to the user interaction with the dataset presented visually by performing actions such as rotation, zooming with input devices such as mouse and keyboards.

The interactivity of the elements in the 3D UI representation is one important issue to be considered in the GUI design. Moreover, Web 3D refers to “interactive 3D technology that one can use through a web browser” (Bowman et al., 2005).

Bowman et al. (2005) describe that navigation tasks are defined into three categories such as Exploration, Search and Maneuvering. The exploration task can be described as the user investigate the environment by navigation. Search tasks involve finding and reaching a target location. The maneuvering task is all about moving a view point from one location to another location. The task involves placing a viewpoint in a more advantageous location which is characterized by short range, high precision movements to perform some specific task.

The selection and manipulation technique involve three basic tasks such as object selection, object positioning and object rotation. These refer to the tasks such as touching, positioning and orientating the object with given input device such as mouse. (Bowman et al., 2005).

The study aimed to create 3D user interface design for the CI system must be simple, lightweight and intuitive to the users and allow the users to interact with the data. Additionally, UI could allow users to customize the graphics to a certain level (such as viewing the 3D environment without lights or axes grid wall - see Appendix A.3).

In this study, I have employed the following interactive techniques: filtering, projection, zoom and linking and brushing (see Keim, 2002 and section 2.2 in the thesis), and selection and manipulation techniques for implementing rotation (see Bowman et al., 2005).

The study of data visualization and advantages of 3D together have formulated few set of criteria that can be identified by applying 3D visualization to the CI system data. By implementing 3D visualization, criteria such as comparison of various datasets, trend analysis for various products, outlier’s detection, correlation with other datasets, best possible build rates for each product are expected to meet. The results are evaluated in chapter 5 to check whether the implementation of 3D visualization met these set of criteria.
2.9 Terminology

In this section I describe some of the technical terms that are needed to understand before going through 3D graphics implementation and these terms will be used frequently in this thesis in the following chapters.

Viewport: Angel and Shreiner (2014) defined viewport as a rectangular area of the screen window. A screen-coordinate area selected for display is known as screen window. The window is measured in terms of window or screen coordinates, where the units are pixels. In window system, the graphics window is a particular type of window where graphics can be displayed or rendered. The viewport defines the area inside the graphics window “where the visual representation of data is to be displayed”.

Figure 5. Viewport in interactive computer graphics (Angel and Shreiner, 2014)

As shown in Figure 5 the viewport is the visible rectangular area of a window screen to its users. By default, it is usually the entire window. The size of the viewport can be set to smaller size via the graphic function glViewport (x, y, w, h) which is defined by four parameters such as x, y, w, h. The parameter values are defined in pixels, where x and y variables are measured relative to the left corner of the viewport and w, h gives the width and height of the viewport. The four parameters allow us to specify the positions and distances in pixel. (Angel and Shreiner, 2014).

GPU: Graphics Processing Unit is “a computer hardware component that performs graphical computations to create and manipulate images” (Eck, 2018). The GPU frees up the CPU by offloading all calculations from the CPU (Eck, 2018).

Perspective projection: Represents the way or point of view through which a user views a rendered scene in a screen via the viewport. 3D systems typically use the camera as an object that defines where the user is positioned and oriented along with real world camera properties such as the size of the field of view which defines the perspective. When a user views objects in a scene, if the objects further away in the scene appear smaller than objects close by in the front of a scene is called as perspective view or projection. (Aguilera, 2008). As shown in Figure 6, in perspective projection things further away are seen smaller in the screen, and in orthographic projection the entire object is seen in a same scale. In orthographic projection mode, the object size of the rendered image stays constant irrespective of its distance from the camera. (Aguilera, 2008; Marland, 2015).
The projection of a model defines how the form of the objects should be displayed. This depends on the kind of projection is used. There are two types of projection defined in 3D graphics system one is the parallel projection and the other one is the perspective projection. The term projection and view are used interchangeably throughout this thesis. Projection in this context means projection of 3D model into a plane surface.

The perspective projection can be further classified into one point, two-point, and three-point perspective projections. The parallel projection is further classified into oblique and orthographic projection. There are two main projections used in WebGL’s three.js library, one is the perspective projection ad other one is the orthographic projection. (Aguilera, 2008). In this thesis the perspective projection has been employed to mimic the way the human eye sees, and it is the most common projection mode used for rendering a 3D scene. (Aguilera, 2008; Marland, 2015; “three.js”, n.d.).
3. Research methods

The goal of this study is to explore, build and evaluate a 3D interactive web visualization system design of CI system data to improve the interpretation. Two research questions are addressed: 1) How 3D visualization technique on web can help to improve the interpretation of Continuous Integration big data? and 2) How 3D visualization could be designed in interactive web to better support exploration of CI system data?

Design science research methodology is chosen as the most suitable research method for this study. The research goal and questions clearly state that the thesis is focused on building and designing a 3D visualization system and its user interface to project CI system data and enable users to interact with it to facilitate the exploration and the interpretation of the data. In this chapter, there are two sections to describe the research methodology used in this thesis. The first section describes the Design Science Research in Information Systems (IS) discipline. The next section describes the Design Science Research Methodology (DSRM) along with its activities and procedures. The second section also describes how the DSRM has been adopted to this thesis context.

3.1 Design science research approach

Venable (2006) refers to the Herbert Simon’s book “The Science of the Artificial” as an important foundation for the information systems (IS) research. Following and building upon the Simon’s research, March and Smith (1995) describe different approaches to IS research and note that design science research approach needs to go through two main processes as a cycle, namely build and evaluate. Build is a problem-solving process by creating or building artifacts for a specific purpose, and evaluation is an assessment process of the artifacts created. (See Venable 2006, p.2)

Hevner and Chatterjee (2010) define that the main goal of DSR is “to create innovative artifacts addressing unsolved problems in organizations”. Artifacts created can be used to solve the organizational problems and evaluate other artifacts. Hevner, March, Park & Ram (2004) proposed an Information System Research Framework which helps to understand, execute and evaluate design science research in IS. Figure 7 shows the design science research cycle defined by Hevner et al. (2004).

![Figure 7. The research cycles in design science research (Hevner et al., 2004)](image-url)
The framework emphasizes the relevance cycle, the design cycle and the rigor cycle are three research cycles that must be present in the Design Science research. These cycles emerge as interaction between environments, DSR and knowledge base. The relevance cycle consists of identifying the problem domain and its business needs. The rigor cycle ensures the research is built upon the past knowledge as well as builds new knowledge to create innovation. The central design cycle iterates between the construction, evaluation and improvement of the artefact design.

In addition to the research cycle, Hevner et al. (2004) defined seven guidelines for conducting and evaluating the design science research. These guidelines provided the characteristics of good design science research outcomes. The seven guidelines are described in Table 3. Hevner et al. (2004) mention that among the seven guidelines “evaluation is a crucial component of the evaluation process”. Moreover, the evaluation method must be matched to the artifact and evaluation metrics. (Hevner et al. 2004).

Table 3. Hevner’s seven guidelines (Hevner et al., 2004).

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design as an artifact</td>
<td>DSR must produce a viable artefact in the form of constructs, models, methods or instantiations.</td>
</tr>
<tr>
<td>Problem relevance</td>
<td>Develop a technology-based solution that is relevant to business problems.</td>
</tr>
<tr>
<td>Design evaluation</td>
<td>The artifact must be evaluated for its utility, quality and efficacy in the problem context.</td>
</tr>
<tr>
<td>Research contribution</td>
<td>The research must provide tangible and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodology.</td>
</tr>
<tr>
<td>Research rigor</td>
<td>The research must build upon solid foundation of rigorous methods in both the construction and evaluation of the artifact.</td>
</tr>
<tr>
<td>Design as a search process</td>
<td>The design process must be a constant iteration between construction, evaluation and improvement to define a satisfying solution.</td>
</tr>
<tr>
<td>Communication of research</td>
<td>The research must be presented effectively to both technology-oriented and management-oriented audiences.</td>
</tr>
</tbody>
</table>
3.2 Implementation of DSR methodology

The DSR methodology has been adopted to carry out this study. The study followed a sequential procedure of DSR methodology starting from problem-oriented approach. Peffers, Tuunanen, Rothenberger, & Chatterjee (2007) presented a methodology as guide to effectively present and conduct their DSR in a rigorous manner by providing principles, practices and procedures to researchers. The DSRM evolved due to the lack generally accepted process models for carrying out DSR (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). Peffers et al. (2007) after examining seven representative papers presented six activities that should be carried out in the methodology. It mentioned that the researchers do not need follow the sequential order to follow the process. The researcher could start from any of the first four activities. Figure 8 demonstrates the framework of DSRM that includes the six sequential activities that are highlighted in bold text. The following list describes how the six activities of (Peffers et al., 2007) adapted to this thesis context during the research process.

Figure 8. The adapted design science nominal process model (Peffers et al., 2007)

**Problem identification and motivation:** The study is focused on visualizing the CI system data. The problem faced by the developers and managers in software organizations is that they need multiple screens to compare the big data of multiple products of different categories of data with respect to time series when they visualize the data in 2D. Tough CI team (users) of the beneficiary organization had their CI data visualized in 2D it was difficult for them to compare and analyse the different datasets in multiple screens. Also, the user needs some manual work to compare and analyse the trends of different products, where exploration and interpretation of the dataset of multiple products become difficult. This problem statement motivated the CI team of the beneficiary organization to implement 3D visualization and to identify the advantage of 3D visualization of CI system data in web environment.

**Objectives of the solution:** To answer the research problem, 3D data visualization approach would be feasible as it can resolve the space issue where multiple products data can be visualized of different categories in a single 3D environment. Implementing a 3D visualization technique in the web interface can provide an advantage of accessing
the visualization and visualized data visibility for developers in multiple locations irrespective of the location and time. The objective is to design a UI prototype for CI system data in a 3D web environment along with some interaction techniques.

**Design and Development:** The study focuses on visualizing the build rates of CI system in a 3D web environment which helps multiple datasets be compared in one single web screen space. The design and development phase comprise of implementing the design of data visualization in a 3D web environment using HTML5 UI elements and 3D graphics. Also, it focused on implementing interactivity of the data visualization such as rotation and zooming. The design cycle is executed iteratively, namely the developed design is frequently evaluated and improved until it reaches a satisfactory state.

**Demonstration:** In this research, the 3D web visualization helped to solve at least three purposes. The first comparison of multiple products in a single screen space and the next one is to identify the hidden data which are not visible in a 2D data visualization. The last one is without installing any plugins the users can view the data visualized in a web browser for different date period by choosing start date and end date of the data to be visualized.

**Evaluation:** During the implementation, the design was evaluated at frequent intervals by the experienced technical leads from the beneficiary organization to examine and suggest improvements. In the end a subjective overall summary evaluation was reported by defining the criteria met by the implemented thesis work. The evaluation part is explained in detail in chapter 5.

**Communication:** The entire research process, its activities, the importance of the problem, and the developed artifacts are presented in this thesis and communicated to the beneficiary organization.
4. Architecture and Implementation

In this chapter, first I describe architecture of the system, following that in the next section I describe the basics of graphics and 3D graphic components which are required to understand and present the 3D graphics. Next, I have described about the implementation of proof of concept, 3D graphic tools and libraries that have been employed in this study for graphical presentation.

The aim of this thesis was to create a proof-of-concept UI that helps users (software engineers) to interpret and explore CI system data. The proof-of-concept would serve the beneficiary organization to understand the feasibility and the benefits of such visualization tool and to design further an in-house 3D visualization for their CI system data.

4.1 System architecture

This section describes the architecture diagram of the implemented system for the reader to gain better understanding of the different components and explains the interaction between the components. In Figure 9 below, I have related the architecture diagram to data visualization process adopted in thesis work as I demonstrated in chapter 2 section 2.6. Each of the components will be demonstrated briefly in the following sections.

![Architecture of the implemented visualization system](image)

**Figure 9.** Architecture of the implemented visualization system

4.1.1 Developer’s view of system architecture

In this section I explain the system architecture in developer’s point of view. The following Figure 10 and paragraphs explain how the Card et al. (1999) visualization process have been adopted and how the milestones are defined for this study system architecture. Considering the definition of visualization defined by Card et al. (1999) “the use of computer-supported, interactive, visual representations of data to amplify cognition”, the whole system is defined in three layers. The first one is the Data model layer, the second one is GUI layer and the third one is the User layer.
Data model layer describes the data structure or data schema that is designed for the system. The data model layer is the source of information about the data, which defines the behavior of the data stored in the database. By using the Python scripts, the data can be collected and transformed into JSON data format for the ease of use. The data transformation is done in the data model layer. The data model layer is responsible for processing users request and returning the response via GUI.

Figure 10. System architecture in developer's view

The GUI layer is responsible for rendering the information that should presented to the user. The visual mapping and view transformation takes place in the GUI layer using the 3D graphics library, HTML 5 and css3 elements. In GUI layer the user can control what data should be displayed by filtering the appropriate dataset variables through HTML5 UI elements such as dropdown menu.

In User layer the visual and cognitive analysis take place, here cognition refers to the acquisition or use of knowledge. In the user layer, the user interacts with the GUI through the various interaction techniques such zoom, view rotation, linking and brushing, which helps to perform the visual and cognitive analysis. Thus, the visualization act as a powerful tool to help the users to perform the various cognitive processes. These layers are described in more detail in the following subsections (s. 4.1.3, 4.1.4, and 4.1.5).

4.1.2 Programming languages and software tools

Here I describe briefly the programming languages and software tools employed for the implementation of the visualization system.

HTML5 is a standard which is developed in the 2014. The new features of HTML5 have made the web more dynamic along by providing faster connection between client and server. The canvas tag element in HTML5 have enabled the graphics rendering in web. (Nordin, 2016).

Python is an easy and high level interpreted programming language unlike java or C which are compiled language. There are have been lot of frameworks developed to support web development using Python. Additionally, a vast number of libraries have been developed to extend the support of various functionalities. (Vantti, 2018). In this context Python script has been written to collect from the database and dump the data in JSON format.
WebGL is a low-level JavaScript API to render 3D graphics within any compatible browser. WebGL gives access to hardware accelerated graphics via HTML5 and canvas elements. Since it is a low-level API it is bit hard to use, Hundreds of lines need to be written to draw a simplest task. Three.js open source JavaScript library have abstracted the complexity of WebGL and allows us to create 3D content in a much easier way. (Mwalongo et al, 2016).

JavaScript is an interpreted programming language. Along with HTML and CSS technologies JavaScript has become one of the essential technologies in developing web-based projects. (Guldamlasioglu, 2015).

4.1.3 Data processing model

Data collection scripting was the very first task in this thesis work. After defining the problem statement, Python script was written to collect the data from the database of the CI system. The output of the data collection script is a structured dataset of the CI build system. Once the user submits the request from the GUI user control component, the parameters selected by the user will be passed to the data collection script. Based upon the user request parameters, Python script will collect the corresponding data from the database and the output will be a structured data.

The structured data will be then rendered and converted to a JSON format for the ease of use of data processing. This corresponds to the data transformation stage in the reference model of visualization by Card et al. (1999). Once the data is structured and received in the front-end GUI application, the 3D line charts are built to present the visual representation of data using three.js 3D graphics library on the web.

4.1.4 GUI design components

The UI was designed and implemented as simple as possible with basic functionality. To interactively visualize the data graphically based on the user selection of dataset, the interface is designed to control the interactive display of the UI according to user actions.

The user interface contains two main control components, the first control component called “User control” that contains two dropdown menus and a date picker along with a submit button. The first dropdown menu contains a list of different branches names and the other dropdown menu contains the test-set data category such as warnings, error types. The user can select the branch and the test-set from the dropdown menu and then select the start date and end date to be visualized. The filtered data selected by the user will be submitted to the backend to collect the corresponding datasets from the database and capture the mappings of the dataset. Once the data is structured it will be converted into JSON format.

The second control component can be called as “Visualization in viewport”. The main task in this component is the visualization of dataset in the viewport and it is carried out in three steps. As a first step data mapping is done by writing a JavaScript functionality and the second step is to map the data to visual structures using three.js graphics library which can be used to present data graphically on the screen. The last step is the presenting the data to the users in the viewport by combing the visual mapped data along with HTML5 and CSS3 elements called views. This is the final milestones as mentioned in the Card et al. (1999) reference model.
The most important part of GUI design is that the viewport contains the grid system representing three axes X, Y, Z and the scale of the grid. The grid system is designed and implemented using three.js graphics library. The grid system along with visual representation of data as line charts inside the viewport represents the 3D visualization of data based on the user selected input from the user control component. The visual representation of dataset along with the grid system can be referred as a 3D object in the visualization system.

The user control component and the viewport component together are presented to the users as a transformed view of the entire visualization process in the web browser GUI. The users can interact with the GUI through various interaction techniques which are explained in the next section.

4.1.5 User interaction

Interaction techniques such as filtering of dataset have been implemented in the user control component which contains the dropdown menus and date picker. In the user control component, the users can filter the data to be visualized through familiar HTML5 UI elements such as dropdown menu elements on the page, which helps to capture the mappings of the dataset resulting in information-rich 3D visualization. Thus, the user can choose the different categories of data to be visualized, also date picker is used to choose the date period of the dataset that should be visualized. This refers to the interactive projection technique of the dataset. The main category of dataset are the CI branches which contains a subset of test-set data. For example, from the dropdown menu the user can choose main branch and filter the corresponding test-set data of that chosen branch. There are various types of test-set data available such as warnings or error data, the default setting is that the test-set data will be collected for all the products under the branch if available.

In the visualization viewport component, interaction techniques such as zooming, linking and brushing and view rotation have been implemented. Zooming allows the user to view a portion of the dataset more detailed with higher resolution of the rendered image to a certain level. Linking and brushing technique is implemented when the user clicks on a data point the corresponding information of the data point will be displayed to the user. Also, the beauty of 3D visualization lies in its selection and manipulation technique which is the view rotation. The user can rotate the entire visualized data set using the input devices such as mouse and keyboard to view the data in many different angles.

4.2 Basic components of 3D graphics

Various studies have indicated that 3D computer graphics play a major role in visualization especially for three dimensional datasets. Before diving into deep technical terms of 3D visualization and 3D graphics, I would like to describe the basic graphic components in the following section.

4.2.1 Basics of coordinate system

This study focuses on the concept of three-dimensional space (3D space). To position the created object in a 3D space, a coordinate system is required. The coordinate system which is used most widely is called the Cartesian space. This coordinate system consists of mutually orthogonal coordinates or pair-wise perpendicular axes, meaning that there
is a 90-degree angle between every two axes with a meeting point in respective angles. (Buck, 2012). The meeting point is called origin and is denoted by the letter O.

In the 2D Cartesian space, a point $p$ is referred as $(x, y)$ denoting the position on the horizontal and vertical coordinates, respectively, whereas in the 3D Cartesian space a point $p$ is referred as the ordered triple $(x, y, z)$ denoting the position of the point relative to the three coordinates in the X-, Y-, and Z-axis.

In computer graphics, the orientation of the axes is so that X is the horizontal axis pointing to the right, Y is the vertical axis pointing upwards, and Z is the camera view pointing towards the viewer. This system is called right handed, as opposed to left handed where the Z axis is pointing away from the viewer. (Vince, 2006, p. 26; Pachghare, 2011).

![Right handed coordinate system](image)

**Figure 10.** Right handed coordinate system

In the right-handed system, the orientation of the axes is such that the succession of the axes follows the counter clockwise direction (i.e., X, Y and then Z). WebGL library uses the right-handed convention, however in specific situations such as working in the clip-space, WebGL uses the left-handed coordinate system (see Matsuda and Lea, 2013, p. 464).

The representation of a point $p$ is known as tuple, which represents an ordered list of elements. A tuple with three elements is called as 3-tuple or triple, i.e., $p = (x, y, z)$. The points $x$, $y$, and $z$ inside a tuple define the positions in the coordinate system. The coordinate points $(x, y, z)$ denote the distance along the coordinate axes the X-axis, Y-axis and Z-axis respectively, from the origin O, defined by the 0 point on the three coordinates, that is $(0, 0, 0)$. Thus, the position of an object can be represented in a vector in relation to the origin O.

In 3D graphics, objects are represented by three points inside a tuple. The points are called vertices and each vertex has three coordinates $(x, y, z)$. The tuple $(0, 0, 0)$ represents the origin point of the coordinate system. Tuples such as $(1, 1, 1)$ and $(2, 2, 2)$ represent a point position in relation to the origin point. This kind of definition is important to understand when we are implementing the concept in coding especially in case of plotting the values in the coordinate system. (Sloka-Frey, 2013).
4.2.2 Rendering

Rendering is the process of generating a digital output giving it a realistic appearance. A rendering component or renderer is required to present the 3D graphics on the screen. The renderer is responsible for drawing the graphics to the screen. 3D graphics are rendered from 3D models called wireframes, which define the shape of the model. The rendering process add surfaces, lighting and textures to the shape of the model to give a realistic appearance. (Christensson, 2016).

There are 2 types of rendering: real-time rendering and pre-rendering. As the name suggests, the real-time rendering is done during run time, while in pre--rendering, the rendering is done beforehand. Real-time rendering is often used in games where the next image to be rendered is unknown. Pre-rendering can be used in single image or videos which can be rendered beforehand and more time is allocated for rendering. (Slick, 2017).

When rendering, it is important to render all sides of the generated mesh for example rendering charts on all sides such as line and bar charts. Rendering refers to doing the calculations to transform the mathematical data of a scene to form a 2D image (Slick, 2017). This can be easily achieved in three.js by using a single parameter called material. In three.js, the WebGL renderer is used as a default renderer to render the 3D scenes. This is the fastest and more powerful than other renderers such as SVG and CANVAS.

4.2.3 Scene and camera

Besides rendering, for creating 3D graphics, a scene is required. Scene can be described as a container where all the objects, lights and cameras are placed. A scene holds all the data of the objects that needs to be rendered ("three.js", n.d.).

Eck (2018) describes scene as a collection of objects each with its own attributes. Camera is not just an object, it is a viewing transformation that the user would like to use. The position of the camera determines the visibility, when the scene is rendered. (Eck, 2018).

4.2.4 Mesh, texture, material, light

A mesh is a collection of vertices, edges and faces which define the shape of the object. ("Polygonmesh", n.d). In 3D graphics, the 3D object is a representation of a polygon mesh which contains the collection of vertices and polygons. ("Mesh, 3D", n.d). A 3D mesh object contains a different geometry type such as cube, polygon, sphere and different material type that describe the appearance of objects such as Lambert, Phong.

Figure 11. Cube geometry shows edges, vertices, and faces
Figure 11 illustrates the vertices, edges and faces of a cube mesh in 3D graphics. A vertex is a point in the 3D space which represents a position in terms of x, y, and z coordinates. (Eck, 2018). A texture can be described as a sheet that acts like a skin lying on the surface, and the material is assigned to the mesh to show the behaviour of the surface when it is anticipated with light. (“3D computer graphics”, n.d.). To illustrate these concepts, let us think we want to add a steel chair to a 3D scene. As a first step, we need to create a chair-shaped mesh. Then a texture will be assigned to the mesh to make it look like as if it is made of steel. This can be done, for example, by using colour, however the object would not reflect as a steel chair until the reflecting property is set. To correct this, a material would be added to the mesh, to set behaviour of the surface when it is anticipated with light. Finally, to view the object in scene a light is required, without a light only a dark and black screen would be rendered.

4.2.5 Grid and axes

A grid represents the system’s unit of measurement such as scale and size of the grid. A grid is a regular pattern of parallel lines intersecting at right angles and forming squares; it is used to identify precise positions (Alexander, 2017). Additionally, it helps to inform the user about the location of the contents in the scene. In three.js library, the axes can be presented by using a grid helper or using line class function for user defined grid. The grid is built using the box model with line class function to present the 3D coordinate system and the scale for X, Y, Z axes. The scale range on the grid is dynamic, that is the scale range changes depending on the range of the Y axis minimum and maximum value.

4.2.6 Camera

In 3D systems the camera is used as an object that defines where the user is positioned and oriented. There are different cameras available in the three.js library. The most commonly used are perspective camera and orthographic camera. In orthographic camera, projection view retains the original object size of the rendered image; that is, the object size stays constant irrespective of its distance from the camera. Perspective camera is the most common projection mode used in 3D scene rendering. The perspective camera projections are like human eyes projection of real world objects. (“three.js”, n.d.). In this thesis, I employ the perspective camera to present the 3D visualization as it would be perceived in the real world.

4.3 Implementation

This section describes the details of the design and implementation of the 3D data visualization in terms of its graphical representation. Also, the design and development of the UI is described in this section. The user interface provides an intuitive understanding of the 3D effect and helps the user to identify the advantages of applying 3D visualization to CI system data which is accumulating every day in a CI system database. Also, the model for the process of 3D visualization of CI system data is presented in this section.
4.3.1 Data collection and pre-processing

The first step in data visualization is data collection. Before going through the process, I describe the data. The CI system task is to verify the code integrated by the developers for the corresponding products and components. The CI system creates a build process to verify the successful integration of their code. This build process produces various types of information such as warnings, error information. This information will be stored in the CI system database.

In this study the focus is to visualize the resulting information that is the test-set data such as warnings, error information which is produced by the build process. For example, if the test-set data is warnings and to visualize of different types warnings, the count variable needs to be calculated. This different type warning variable let’s say w1, w2, w3 contains an array of data points [t1, t2, t3], [t2, t3], [t1, t2, t3, t4] which needs to be mapped and visualized along with its corresponding products of the branch and time series.

To visualize the variable of different types of test-sets which is an array of data points, the first step is these datasets needs to be collected from the database based on the user selection. In this study, the implementation of this user action is done with a Python script. The next step is to map and convert the data array into an easy readable format. The conversion of data format could help to ease of data processing and transformation. For this reason, the dataset is converted into JSON format in this case. Once the data is converted to JSON format it can be normalized and converted into $p(x, y, z)$ coordinate values using 3D JavaScript libraries. The coordinate values give the position of the dataset in the visualization viewport grid system. These positions are marked as data points in the grid system which are connected using line graphs.

4.3.2 GUI implementation

The GUI implementation is divided into two components. The first is the implementation of user control component and the second is the implementation is the visualization in viewport component. See Appendix A.4.

The user control component contains the HTML UI elements such as dropdown menus, date picker and a submit button. The dataset filtering and submission of the data is implemented using HTML5, CSS3, JavaScript technologies. When the user filters the data from the dropdown menu and submits the data, Python script will be called to process the request and collect the corresponding data from the data model layer. As a response to the user request, data will be rendered and returned in JSON format.

Second, to build the visualization, several steps are performed, as mentioned in section 4.1 system architecture, section 2.5, section 2.6 reference model by Card et al. (1999) and adapted reference model for the web visualization.

Once the data is filtered by the user through the dropdown menus, the filtered dataset will be processed to capture the mapping of the dataset. Once dataset is mapped, structured and transformed, it can be presented graphically in the visualization environment. The visualization is interactive, in the sense that it integrates a UI that can be manipulated by the users. The view can be rotated and zoomed, which helps view the whole dataset in various angles and in various resolutions while zooming. The projected dataset can be viewed in various angles by clicking and dragging the mouse. The mouse scroll wheel can be used to see a zoomed in and out view of the data model.
This is visual transition of data. Figure 12 shows the prototype design created before the implementation of the visualization.

**Figure 12.** Prototype for the 3D visualization of data

Stone (2016) mentions that “the most important use of colors in information visualization to distinguish one element from another”. As color can act as a strong aesthetic component, the usage of color in information display is about a function “what information are trying to convey and how color could help to enhance information displayed”. (Stone, 2016).

Sullivan (2016) mentions that the use of colors adds in another dimension of data to a graphical presentation which can also help to draw attention to specific features. (Sullivan, 2016). Colors plays a major role in data visualization by making the data understanding and interpretation easier to the user. Colors help to distinguish the different categories of the data visualized. (Healey, 1996).

**Figure 13.** Screenshot for different use of colors in visualization
The use of color label in the UI design could help the user to make cross reference easier. Figure 13 shows the color label information is displayed in the 3D GUI for the ease of understanding to user.

The choice of colors was based on the requirement of the beneficiary organization. The requirements were to group the zero values for a corresponding category of test-set data and to distinguish it separately from the other datasets red color was assigned. The other choice of colors was made randomly to suit the grid, axes and the background color. The font in the display window is instructed by the organization.

The screenshot of Figure 13 shows the implementation of different use of color to distinguish between and represent different types of test dataset. To follow the standards of beneficiary organization blue color was chosen for the background and the grid system. To represent the data and the text, colors are chosen equally important which should be easily visible with the blue background. The important point in choice of colors lies in that the overlaid data or the hidden data must be visible if they intersect each other. This is achieved by increasing the transparency of the material and the suitable choice of colors, which is shown as the dots in the Figure 13. Additionally, the connecting data points are also shown with box for the ease of understanding. When the user clicks on the data point the coordinate’s points can display information about the \( p \) (x, y, z) coordinate values such as date time, test-set variable, and product respectively.

### 4.3.3 JavaScript library for the graphical presentation

JavaScript libraries such as D3.js and three.js library is employed to present the data graphically on the GUI which is running inside a web browser. D3 stands for Data-Driven Documents and D3.js is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. (“D3.js”, n.d). Three.js is a cross-browser JavaScript library and API used to create and display animated 3D computer graphics in a web browser. Three.js library uses WebGL. Three.js library performs the rendering with the help of three renderers namely Canvas, SVG and WebGL (“three.js”, n.d). WebGL is the most commonly used renderer, while Canvas is considered the second most used. Three.js mouse interaction is supported via the OrbitControls.js API.

Both D3.js and three.js have quickly become popular tools for building 3D objects in the web. Three.js is one of the most popular library because of its good documentation and examples provided. Most importantly, it has many code examples and references which gave me ideas and information about what was possible and what was not possible in projecting the web data visualizations for this thesis work. After analyzing and studying many other libraries, three.js was found suitable to present the 3D content in the web for this thesis work. Many other available libraries with similar features are mostly game engines. The main advantage of using WebGL is that everything can be defined as object. Though three.js is also suitable for creating games, it contains only essential features to present the 3D content. Some of the features of the three.js library used in the implementation of 3D graphics are presented below (“Three.js: Features”, n.d.).

- “Renderers: WebGL, <canvas>, <svg>
- Scenes: Addition and removal of objects
- Cameras: orthographic and perspective along with various controllers
- Lights: ambient, direction, spot and hemisphere lights
- Materials: Basic, Lambert, Phong and more along with textures
- Geometry: plane, sphere, torus, cube, 3D text and tube.
- Objects: meshes, particles, sprites, lines
- Export/Import: utilities to create three.js-compatible JSON files from within: Blender, 3D Max.”

In this thesis, the dataset normalization is done using D3.js. The user interaction such as view rotation, zooming, linking and brushing are implemented in a straightforward fashion by using the three.js library. The zoom and rotation operations can be defined as a translation of camera moves, where the camera moves closer to the object when the zoom in operation occurs and the camera moves back in the zoom out operation. Linking and brushing is achieved by clicking on one of the data points where the user can see the information about the data in more detail such as product, date time, test-set value.

In the three.js library, the mouse interaction is supported via an API called Orbitcontrols.js which is available on the three.js tree, for example the location of the camera is moved in with mouse input (Li, 2014). Next let us see the basic coding elements of creating a scene that contains the 3D objects which is projected in the viewport.

```javascript
Var scene = new THREE.Scene(); // creation of scene
Var aspect = window.innerWidth / window.innerHeight
Var camera = new THREE.PerspectiveCamera(90, aspect, 0.1, 1000); // creation of camera
Var renderer = new THREE.WebGLRenderer(); // create the renderer
Renderer.setSize(window.innerWidth, window.innerHeight); // set the width and height of renderer
Document.body.appendChild(renderer.domElement()); // append the created 3D model to the screen
```

The above code shows the three.js programming code basics. The above code is necessary to create a scene, camera and the renderer. The renderer is a place where a generated 3D model will be rendered. Next let us see what kind of 3D elements are added to renderer and how the 3D objects are rendered.

![Scene node map and its flow to renderer using three.js](Lyons, Dec 13)

**Figure 14.** Scene node map and its flow to renderer using three.js (Lyons, Dec 13)
The scene node map in Figure 14 describes what kind of 3D object elements are added to the scene to render. The renderer’s job is to render all the elements together and display it in the screen. Let us go through the elements of the scene node map one by one in the following paragraph.

The first component to be considered in the scene map is the geometry, three.js library comes with many prebuilt geometry types that represent common shapes such as cube, circle, spheres and cylinders. These pre-built geometry types are derived from the base class THREE.Geometry. The next component in the scene node map is the material component that describes the appearance of the objects with surface properties such as the color, shading and textures. Three.js library supports common material types through built in classes such as MeshBasicMaterial, MeshPhongMaterial and so on. The geometry and the material element together can build a new 3D object called mesh. The next step is to build a scene object along with graphic elements such as camera, mesh and light to renderer. These graphic elements camera, mesh and light are described above in section 4.2. The scene object is the root of the node map which will be rendered by the renderer. The renderer component displays the crafted scenes in the 3D viewport which is visible to the end user.

4.3.4 Implementing the line charts for data visualization

The visual structure chosen for the data representation was line graphs more specifically multiple line graphs, where one-line graph is built for each one group of test-set of the CI system product category. To implement the visual structure and visualize the data, a grid system was built to represent the X, Y, and Z axes. The grid creation was one of the important design element of the UI presentation as it represents the three-coordinate’s position corresponding to three variables in the data, namely product components, time series and test-set data such as warnings.

![Figure 15. Screenshot of spline charts](image)

As shown in Figure 15, the X axis represents the time series, Z axis represents the CI system product components, and the Y axis represents the test-set quantitative data,
such as number of warnings or number of errors. The whole grid system represents the visualization of one branch dataset that contains different type of tests-sets.

The choice of color for the grid element and the line graphs was also important. Line graphs color was chosen to match X, Y, Z grid axes. The choice of colors for the line charts must be clearly visible inside the grid. In Figure 15, the grid lines in the Y and Z axis is a 3D graphic element that is drawn using three.js library and the number of lines to be drawn is calculated based on the maximum value of the variable represented on the Y axis, namely the number of warnings or the number of errors in the selected test dataset. See Appendix A.5.

Line chart was selected for visualizing CI system data that allow us to make comparison between the different components at the same time. Also, line charts help to compare and analyze the trend of dataset performed during a period. Line charts allow us to make comparison between the different components at the same time in a single screen space. Also, the implementation of the line charts helps to find the outliers.

Sullivan (2016) mentions that the graph creator must understand what visualization can make sense for his/her data. In this study, since the focus is to analyze the trends of data, for multiple products, multiple line charts are found as being best suited. As there many types of dataset present in one test-set array of data, one color is assigned to one type of the test-set data. For example, the warnings test-set contains different types of warnings. The color label in Figure 15 provides information about the assigned colors to each type of warnings.

Multiple line charts help to better understand the relationship between the value sets and to show the changes over time. Marghescu (2008, p. 103) showed also that multiple line graphs can be effective in tasks such as identifying relationships in data and comparing or classifying data points.

More specifically, to have an enhanced view of the data set, in this thesis, I have implemented spline charts that display smooth and fitted curves that are connected through different data points, unlike the line charts that connect the data points with straight lines. Since the CI system deals with different types of datasets a multi-series spline chart was generated to visualize all selected types of test-set in the same view.

The plotting of data is done using three.js library and the normalization of data is calculated using D3.js library. The data received form the data model layer was a JSON response which needs to be normalized. Three.js library is used to scale the data and set up the ranges for the axes. As shown in Figure 15, the line charts are drawn by connecting the different points of same category in a certain time series. The different categories of data are differentiated by various colors.

Various factors were needed to be considered in designing the UI of the visualization such as the presentation of the text, colors and its position when the users perform rotation and zooming activities. Thus, from the screen shot in Figure 15, one can see that the 3D visualization allows you to visualize multiple data series side by side each other, along with some basic interaction techniques. In some places in the view presented in Figure 15 there are series of points joined by lines which could overlap each other; this can be avoided with transparency of the object to a certain extent as well as by rotating the view in such a way that the overlapping lines become separated.
5. Findings

This study was aimed at creating a proof-of-concept (that is, a prototype) of 3D visualization for CI system dataset that could help the interpretation and exploration of CI system data. Normally, 3D visualization is used for volumetric data (data with physical geometrical properties such as length, width and height). However, in this thesis, 3D visualization was employed to render abstract data related to software development CI system data.

To accomplish the aim of the study, the design science research approach was employed. The proof of concept was designed by applying the principles of data visualization and computer graphics. These principles were implemented by employing three.js JavaScript library in front end to present 3D graphical visualization that is in the web browser viewport and Python scripts were implemented in the backend for data collection and data mapping.

The design and implementation were iterative processes, and the evaluation of the prototype in various stages was conducted with the beneficiary organization and expert feedback was considered. For example, the feedback given during the evaluation with the customer were: colors to be used or preferred, customer needed the spline charts instead of straight lines. In GUI visualization viewport the design was changed after considering the feedback from the experts. (See Appendix A.3 and A.4 indicates the design change).

In addition to the iterative process of building and evaluation, a summary evaluation was performed for the prototype. Two main sets of criteria were used for this summary evaluation, corresponding to the two research questions RQ1: How 3D visualization can help to improve the interpretation of Continuous Integration data? and RQ2: How 3D visualization could be designed in interactive web to better support exploration of CI system data?

The first set of criteria was focused on evaluating the extent to which the visualization helps in interpreting the data, namely: 1) Comparison between different test-set of different products 2) Trends analysis of various products in various date time; 3) Identify outliers. 4) To identify the best possible build rates by connecting the same group data points. 5) To identify correlation with other datasets. These types of criteria are related to the domain problem of interpreting continuous integration data. Similar approaches are found in literature reference (e.g., Koua, Maceachren, Kraak, 2005).

The second set of criteria refers to the ability of the visualization to enable exploration of the data. These are based on the principles of visualization techniques, those that 1) enable interactivity with the data and 2) enable interactivity with the visualization system.

Teyseyre (2002) mentions that “Any usability evaluation of a visualization technique must address both the evaluation of visual representation and interaction techniques by which the exploration helps to achieve the insights than a static image”. He additionally mentions that the main challenge in software visualization is that evaluation of effective mapping of dataset to provide insight and easier understanding of the dataset.

Teyseyre (2002) refers to Mack inlay’s two essential criteria to evaluate mapping of data to a visual representation 1) expressiveness 2) effectiveness. The expressiveness
criteria determine whether visual representation expresses the desired information while the effectiveness determine whether the visual representation exploits the capabilities of the output medium and human visual system.

Considering the above mentioned criteria, Table 4 and Table 5 present the summary evaluation of the implemented system in a subjective point of view.

### 5.1 Summary evaluation of data interpretation

Table 4 explain about what criteria the thesis work has met in terms of data interpretation. The data interpretation evaluates the usability of the data visualization in terms how the implemented data visualization provides insights that are useful to the users.

**Table 4. Summary evaluation of data interpretation**

<table>
<thead>
<tr>
<th>Data Interpretation</th>
<th>Comparison with other data sets</th>
<th>Trend Analysis</th>
<th>Outliers detection</th>
<th>Identify best possible build rates</th>
<th>identify correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Line charts</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In this case the multiple line graphs present the relationship between the multiple types of information through the visualization. The following paragraphs describe how the criteria for data interpretation is achieved in this thesis work.

First, the multiple lines chart can indicate different types of test-set produced for each product, this can help the users to compare the datasets with multiple products in a single screen space. This is referred to the comparison of the multiple datasets in a single screen space with the advantage of view rotation.

Second, the spline charts of test-set data tend to move in a series of highs and lows over time by connecting the same group of data points depending on the position of the data point in the coordinate system. This direction of highs and lows constitute a trend. This refers to the criteria of trend analysis of various products in various date time.

Third, line chart smoothness can indicate for which type of test-set and for which product the maximum number of highest value has occurred can be identified. For example, the user can see for which date the highest error values has occurred by clicking on the data points. This example indicates the outlier detection criteria.

Fourth, the user can filter the dataset of different CI build configuration test-set data and visualize it. This filtering of dataset could help the user to identify which configuration of test dataset has produced the best possible build rates.

Fifth, since the line charts of multiple products is project in a single screen space it is easy to compare the datasets, Also the colors help to differentiate between different types of test dataset. The color grouping of datasets along with multiple line graphs helps to understand and identify the correlation between the datasets.
5.2 Summary evaluation of the user interface

In this section, the usability of the system is evaluated as to what extent the interaction techniques implemented enable exploration of the data and the interaction with the overall system and its components. Thus, the evaluation of data exploration is formulated in terms of UI insights or UI capabilities of the developed 3D web-visualization prototype.

As presented in chapter 4, the user interface of the prototype consists of two main components. In the first component the user can filter the test-set dataset to be visualized. The filtered dataset will be projected in the viewport. The second component is the 3D viewport where the user-filtered test-set data is projected with interactive techniques such as interactive projection, filtering, zooming, rotation, linking and brushing. A summary evaluation of these techniques as to their exploratory capabilities is presented in Table 5.

Table 5. Summary evaluation of the data exploration

<table>
<thead>
<tr>
<th>Interactive / manipulation technique</th>
<th>Description</th>
<th>GUI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>User control</td>
</tr>
<tr>
<td>Filtering</td>
<td>The user can select the data to be projected using user controls (Dropdown menus and date picker)</td>
<td>Yes</td>
</tr>
<tr>
<td>Projection</td>
<td>Projecting the data based on the user actions. This action is triggered by the filtering.</td>
<td>Yes</td>
</tr>
<tr>
<td>Zoom</td>
<td>User points to the area of interest that the user wants to explore more detail using input devices such as mouse and keyboard in the visualization viewport.</td>
<td>No</td>
</tr>
<tr>
<td>Linking and Brushing</td>
<td>Brushing refers to user can select the specific data point and the linked information will be presented to the user.</td>
<td>No</td>
</tr>
<tr>
<td>View Rotation</td>
<td>View rotation is a selection and manipulation technique that enables user to view the entire 3D visual object in a different angle.</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 5 presents the various implemented interaction techniques and evaluation based on the implementation of interactivity with the data and 3D visualization in the viewport. The first column in the Table 5 describes the various interaction techniques implemented in this thesis. The second column in Table 5 describes the interactivity performed. The GUI column evaluation is based on whether the interactive technique is implemented in the user control component or in the viewport component. The data column defines whether the interaction involves the data transformation or not. The view column defines whether the interactivity involves view transformation.

When visualizing the data, the colors of line charts and the color label in GUI representing various test datasets can help the user to understand the dataset projection easily. Additionally, when the user selects a data point, the information regarding that point is also displayed on the screen (see Appendix A.5).

Filtering and projection are the interactive techniques that are implemented in the user control component while zoom, view rotation, linking and brushing technique that are implemented in the viewport component. As defined by Bowman et al. (2005) the viewport component interactive techniques can be referred as selection and manipulation techniques. The two components, along with the elements such as colors, and the interactive techniques together make a complete system for 3D visualization of the CI system data.

After analysis of the two components in the UI, the overall project can be viewed as successful and the required features for data exploration were implemented. The prototype provided some good insights of dataset using 3D graphics such as, multiple views of the dataset in various angles, the dataset differentiated through different colors and overlapped dataset can be viewed by reducing transparency of the material, which helps to identify the hidden data behind other series line chart.

To implement the 3D visualization successfully in a web environment, a deep knowledge of 3D graphics rendering was required. This knowledge was acquired by learning and applying JavaScript-based programming of 3D computer graphics using WebGL, as well as programming in Python and HTML5.

From the above process it could be said that the 3D visualization suits best to compare multiple datasets in a single screen space. An extra care is needed when the quality element comes in designing the visual elements of UI in 3D environment. Additionally, it is noted that the quality of the visual elements is better when viewing it much closer in 3D environment. As a thought from the above analysis, it can be said that the 3D visualization needs a careful design otherwise the quality control can be missed. As an overall impact, with careful design the visual capability of 3D data visualization is much better than other medium.

To further extend this work, the performance of visual elements loading can be improved by utilizing various front-end frameworks such as angular framework and react framework. These front-end technologies could be used to increase the performance of the application. Also, more user interactions such as user selection of the colors, grids, and lightings could be implemented and tested. This web-based visualization can further extend to virtual reality by projecting the data in the local space.
6. Discussion and implications

The purpose of this study was to implement a prototype of 3D visualization of the CI system data. Additionally, the study also evaluated the feasibility of implementing the visualization using the three.js tool. The implementation and the evaluation of the prototype help to find the advantages and suitability of using 3D visualization for CI system data projection in 3D. In this chapter, I discuss about the three.js tool which is used to build the 3D environment for visualization and I reflect upon the conducted analysis.

The resulting information from build process contains test set data such as warnings, error information for multiple products. This test set data contains different types of warnings, different types of errors. The different type of warnings contains different set of arrays of data points. Though the visualization of warnings variable is already implemented in 2D for every product, the users of CI team is not able to compare the data sets for multiple products without displaying the data on multiple screens.

3D visualization solves the problem of visualizing multiple variables of different test set warnings variables for multiple products along with time series in a single screen space. This states that 3D visualization is useful in the sense that CI team users can see the holistic view of the big dataset. It can help to easily identify the best build for each product in one single view. This holistic view of dataset makes the data interpretation easier. The interactive techniques such as brushing and linking, view rotation can provide more information to the users that helps in data exploration.

Though 3D visualization provides more information than 2D, the loading time of 3D graphic elements is little longer compared to 2D. The color elements need to be designed carefully to represent multiple variables of dataset. Since the visualization is presented inside a grid XYZ axis restricted space, the spacing between every product needs to be calculated properly to present the data more clearly to the users.

The study also focused on creating the user interface (UI) of the 3D visualization prototype, where HTML5 and WebGL technologies have been used. Thus, this section reflects upon the development work for this study and provides a discussion of the feasibility of the development method as well as suggestions for future work. The HTML5 and WebGL combination allows one to create 3D graphics on the web in a reasonable time of loading the textures in an optimal way. Since WebGL is a low-level API, it is quite difficult to write programs using it directly. Various WebGL based libraries have been developed for the ease of development for the programmers. Among the available 3D libraries such as scene.js, Babylon.js, as well as Python libraries, three.js was found as the most suitable library for this research.

Babylon.js was another 3D graphics library which is very similar to three.js. Comparatively three.js was simpler and more flexible to present the 3D graphic content in the web browser. In addition, three.js has a wide range of examples and animation features such as creation of scenes and animations can be found along with good documentation.

The main feature that needed to be handled very carefully is 3D text and position of the 3D objects. Though three.js offers a feature to write the animated 3D text, there is a possibility that it can make the entire application slow. Therefore, it is suggested that the 3D animation levels need to be prioritized from simple to outstanding which can help to
load and run the application in a good speed. Placing the elements in 3D graphics scene needs attention and the developer needs to be calculative in positioning. Before implementing the features, the developer needs to study the 3D graphics concepts for gaining understanding of the field and mastering the implementation.

The disadvantage of three.js is that it uses ray casting graphics functionality provided by the three.js library which helps to identify the nodes and its position, which is not very effective. Instead of ray casting functionality some other JavaScript functionality could be implemented to collect and display information about the nodes on mouse actions.
7. Conclusions

The presented proof of concept was implemented to identify the possible UI insights and the suitability of using 3D visualization CI system data projection in 3D, implementing the 3D visualization of the CI system data. Though visualizing in 3D is fun and challenging, representing the data visually is much easier to understand when compared to large table of rows and columns. As studied in this thesis the process of designing and developing visualization is an iterative process. The process involves many steps such as to discover what kind of data structure is suitable for the visualization and how it can be rendered in a 3D environment.

The selection of computer graphics tool does play an important role in 3D visualization. It is noted that designing effective visualization using a tool requires a good demonstration or a good number of examples to understand the tool to be used. In this case, three.js website did provide a vast number of examples to understand the three.js library. The three.js tool was helpful to create a good interactive 3D environment which is a basic need in any kind of 3D visualization. The HTML5 and WebGL based tool help the developers to implement the 3D visualization of three-dimensional representation of data in an interactive web environment. The WebGL three.js library which is written in JavaScript allowed to view simultaneously multiple, large three-dimensional datasets inputted in simple JSON format, which can be read and analyzed locally.

The implemented prototype lead to requirements for further improvements in the organization and it was useful to create awareness of the problem mentioned above. This analysis acted as a basis for analyzing the trends of CI system data by comparing the various categories of products in a single screen space over a period of date time series. The developed UI prototype demonstrated the value of added dimension and its robustness in web-based visualization. “The prototype can help to demonstrate the benefits of applying 3D visualization for CI system data to CI team users” this comment was mentioned by the expert from the organization. Thus, the prototype developed acted as a seed to take further action in implementing the 3D visualization in the beneficiary organization. I see that the approaches used in this thesis work provided an effective means for creating UI insights and promoting it for further development of 3D data visualization in CI system. Also, this thesis work has gave suggestions regarding the factors that needed to consider before designing and implementing the 3D visualization in web browsers. As stated in the beginning of the thesis work and to answer research questions I will go through the research questions and see what kind results has been produced based on the analysis.

RQ:1 How 3D visualization can help to improve the interpretation of Continuous Integration data?

This research question is answered by visualizing the test-set of data in an 3D environment which helps to compare one group dataset values to the other group of datasets. The 3D environment gives us one more dimension called depth which helps to visualize and compare the multiple datasets in a single screen space. This multiple comparison helps to interpret the datasets that leads to visualize multidimensional data of different categories in a same time. More specifically in the CI system study it could help to interpret the best and worst possible build rates of multiple products in the same time by watching the smoothness of the line curve. The developed proof of concept demonstrated that the 3D visualization is capable to show more information such as
multiple datasets of various products with data separated in a single holistic view, various viewing angle of the dataset leads to rich interpretation of data compared to 2D.

RQ2: How 3D visualization could be designed in interactive web to better support exploration of CI system data?

Though the data visual representation gives an idea of the interpretation of datasets, however it is insufficient for the user unless it is presented in an information rich environment to promote understanding and drive the users in better decision making. To present the datasets with instructiveness which could help the user for data exploration that is the interactivity of the system helps the user to find out more information about the dataset. For example, the interactive projection of datasets helps to user to visualize the data for various date time series. Thus, while designing data visualization the interactivity level and the interactive elements should also be defined in the early design phase. Additionally, to visualize multiple set of data sets the color coding play an important role to show the standalone groups of data and identify the outliers. Thus, the color elements should also be defined in the design phase. Since the 3D models present objects reflects the real-world objects the quality element of the 3D models should also be taken care while designing the 3D object models.

On answering the research questions Yes, the visualization of CI system in 3D environment helps to solve three main issues. 1) Screen space problem 2) can help to visualize large sets of datasets in a limited space. 3) Provides a holistic view of the entire dataset in various angles. 3D visualization solves the space problem where it is possible to visualize multiple series of dataset in a single screen space. Based on the above analysis it can be said that a 3D representation of multiple dataset can provide a quick and intuitive representation of large datasets in a single space providing more information to the users in one view.

Defining the design elements in the early stage of the implementation clearly could help the developer to implement the system in a more effective way. Based on the analysis when comes to the interactivity the basic operations such interactive filtering, interactive projection, zoom, rotation of the view, linking and brushing techniques are must in 3D visualization. As a final thought though the 3D visualization is quite successful in visualizing software system data the use of 3D graphical elements is much more effective in scientific visualization than in information visualization of software system especially in a web environment.

There are also few limitations for this analysis. The analysis is scoped only to visualization techniques, without focusing on other aspects such as web technology. Though many major browsers support WebGL, browsers like Microsoft need to increase their support for the 3D graphics within web browser. The next limitation is that since three.js library is an open source and it is still under development, some of its features keep changing and the developer needs to cross check for updates in the upcoming versions of three.js library. Due to lack of time, the thesis was limited to a subjective evaluation instead of the user experience evaluation.

Overall the use of 3D graphics in web visualization is a great and interesting field to explore further. Various studies show that the future of 3D graphics is growing drastically in virtual and augmented reality fields. This 3D web visualization of software data can be taken further and explored in 3D software visualization in web virtual reality.
References


Rosling, H. (2010). The joy of stats retrieved from https://www.bbc.co.uk/programmes/b00wgq0l


Appendix A.1 Screenshots

Appendix A.1. Screenshots of the visualization

Appendix A.2. Screenshot of the visualization in another angle
Appendix A.3. Screenshot of the visualization - user manipulation

Appendix A.4. Screenshot of user control and viewport components
Appendix A.5. Screenshot of grid lines in y and z axis