Sensur av hovedoppgaver

Universitetet i Sørøst-Norge
Fakultet for teknologi og maritime fag

Prosjektnummer: **2020-8**
For studieåret: **2019/2020**
Emnekode: **SFHO3201-1 19H Bacheloroppgave**

Prosjektnavn
Extended Reality Inspection (XRI)

Utført i samarbeid med: Kongsberg Defence & Aerospace AS

Ekstern veileder: Alf Pettersen

Sammendrag:
The interest in Mixed Reality Head Mounted Displays (MR HMDs) is rapidly expanding across medical and industrial settings. The main objective of this project was to evaluate the possibility of improving workflow, efficiency and reliability of an inspection process in industrial production setting with Mixed Reality Technology and HoloLens.

Stikkord:
- Mixed Reality technology
- Microsoft HoloLens
- Composite Manufacturing

Tilgjengelig: JA

Prosjekt deltagere og karakter:

Navn	Karakter
Alina Steshenko	
Piri Babayev	
Markus Liavik Kanton	
Ola Matre	
Tobias Mellum	

Dato: 15. juni 2020

________________	________________	________________
Navn	Karoline Moholth	Navn
Intern Veileder	Intern Sensor	Ekstern Sensor

Bachelor's thesis

EXTENDED REALITY INSPECTION

Submitted by

Name	*Student No*
Alina Steshenko	216980
Ola Matre	131865
Piri Babayev	212310
Tobias Mellum	212107
Markus Liavik Kanton	212659

May 2020

Abstract

The interest in Mixed Reality Head Mounted Displays (MR HMDs) is rapidly expanding across medical and industrial settings. Microsoft HoloLens was the first commercial device to merge virtual and augmented reality features in a hybrid technology known as Mixed Reality. The market of devices with 3D hologram-like interfaces is increasing, and with Microsoft's second generation of the HoloLens on the doorstep it is perfect time to explore the potential and limitations of the MR technology. This paper presents research conducted by undergraduate students as a part of a Bachelor's project. The goal was to test and determine the possibility of improving workflow, efficiency and reliability of an inspection process in an industrial production setting using HoloLens. A proof of concept application was developed and built with Unity game engine to assess this evaluation. HoloLens allows one to increase productivity with hands-free access to complex data, and the prototype's features such as the direct transmission of read-values from industrial micrometer to the application along with image capturing to document the process show exciting possibilities. While MR technology and HoloLens itself are promising, the research indicates that at least the first generation of the headset requires improvements in areas such as tracking accuracy and ergonomics, among others, before it is ready for deployment in an industrial setting. In contrast, HoloLens 2, offers improvements in those areas and shortens the gap towards a satisfying solution. Complete evaluation of the second generation headset is not covered by this paper, thus a conclusion could not be drawn on whether the improvements in tracking accuracy and other areas are adequate.

Contents

List of Figures

List of Tables

1 Introduction

The XRI project is a collaborative project between the University of South-Eastern Norway (USN) and Kongsberg Defence & Aerospace AS (KDA), and was carried out by a team of undergraduate students as a part of their bachelor's thesis. Information about the team members along with the team's working agreement can be found in appendices B and C, respectively.

The team was guided by an *Internal Supervisor* and *Internal Sensor* from the USN, Professor José M. M. Ferreira and University lecturer Karoline Moholth Mcclenaghan, and an *External Supervisor*, Technical Director at KDA, Alf Pettersen.

The assignment was to evaluate the possibility of improving efficiency and reliability of a quality control process at KDA using MR technology and HoloLens.

The report is divided into six main sections besides the **Introduction** and **Conclusion**:

- **The Problem** section presents the project's background and introduces the task at hand. Information about the client and project requirements are listed here.
- **Technology** section provides technical background by addressing hardware and software under consideration.
- **Project Management and Development Process** section, as the title implies, is about project planning and development framework. Project model, quality assurance and risk management are discussed here.
- **Software Architecture, User Story Mapping and Design** describes the transformation process from the project requirements to a software design and architecture.
- **Implementation** section explains how the software design and architecture plans from the previous section were carried out, as well as the production of the fixture prototype to accommodate the software application.
- **Review and Future Work** concludes the conducted work and research in a more detailed manner than the following **Conclusion** section, addressing parts from the **Implementation** section and presenting recommendations for future development.

2 The Problem

This chapter is an overview of information about the client, project background, project description and customer requirements.

2.1 KONGSBERG Aerostructures

Kongsberg Aerostructures, referred to as customer or client in this paper, is located in Kongsberg. It is a sub division in Kongsberg Defence & Aerospace AS (KDA) which is a part of the world-renowned KONGSBERG ASA. The division employs approximately 500 workers divided over three main areas [1]:

- Production
- Program
- Support (Economy, marked, technology)

The production is split in three, all of which has a department for manufacturing engineering:

- Mechanical
- Composites
- Special Processes

2.1.1 Composites

The Composites department have two decades of extensive experience in composite manufacturing, producing advanced composite parts for aerospace applications to a diverse and thorough client base. [2]

KONGSBERG ASAs manufacturing process of composite panels includes the following production steps [2]:

1. Material Handling, Cutting and Kitting
2. Laminating/Lay-Up in Clean Room
3. Bagging

4. Autoclave/Demould
5. Bonding
6. High Precision Milling (PMM)
7. Co-ordinate Measuring Machine (CMM)
8. In-House Ultrasound and X-RAY Inspection
9. Prime and Paint
10. Mechanical Assembly
11. Final Inspect

Production process step *11. Final Inspect* is where the client has facilitated the project for the student group discussed in this paper.

2.1.2 Final Inspect Step

The final inspect step is where the final quality control of the different composite panels is performed. The inspector checks that all prior operations are signed off, and that all applicable documentation following each panel is collected and that it corresponds with customer's and internal requirements. [2]

The next operation performed in the final inspection step is thickness measurements at several points on the composite panel, and it is here the customer wishes the team to research and evaluate possible improvements. The following description of the current process is based on information provided by the client and an inspector at the Kick-off meeting (see the meeting abstract, Appendix D).

The number of measurements performed on a panel depends on what type of panel it is - there are approximately 40 different panels and each panel type can vary in both size (from 0,3 to 2,0m) and form. The operation is exceedingly time consuming and prone to error since the entire process is done manually. The most time demanding actions are:

- Printing out model schematics (used for reference of measurement tolerance and location) and collecting the print which is far from the workstation (100 m).

- Measuring each point with a standard micrometer (amount of measuring points varies starting from 5 and up to 100 depending on the part).
- Listing values of each measurement on the model drawing.
- Enter all measured tolerances from the model drawing to SAP on a computer workstation.

The operation can be unreliable for the following reasons:

- The location of the measurement points are precisely the same for the same types of panels, but the area of where a measurement has been performed can vary.
- There is no way to verify the values measured after the operation.
- Human error can also be a factor: inaccuracy, uncertainty, etc.

2.1.3 Technical Department

KDA's Technical Department works with continuous improvements in all departments to streamline and optimize production processes in order to maintain their international competitiveness.

Technical Director at KDA, Alf Pettersen (hereafter named client), is the head of the Technical Department. He also has the overall responsibility for all student projects, whether it is their Summer Project for Students, Bachelor Projects, etc. and is the team's contact person for the bachelor project.

As seen in Appendix D The Technical Department is interested in a more efficient and testable solution that can eliminate unnecessary mistakes and decrease time spent on this procedure.

2.2 Project Assignment

The following description of the project assignment is based on information provided by the customer at the Kick-off meeting (see the meeting abstract, Appendix D).

The client wishes to evaluate the possibility of improving the inspection process by implementing Mixed Reality (MR) technology. The client already has two models of 1st generation HoloLens available and two 2nd generation HoloLens have been ordered for testing purposes. Given that two 1st Generation Hololenses are already available, the client suggested to develop and implement a thickness measuring system using Microsoft's Hololens 1/2 in order to:

- Visualize guidelines on exactly where the measurements should be taken.
- Visualize work instructions on how to execute the operation unambiguously.
- Let the operator account for each measuring action.
- Capture the exact measurement point when measured with the built-in camera.

The project group also needs to research more dynamic instruments for the thickness measurement process, evaluate the solutions and eventually substitute the present manual method with it. These instruments need to be able to communicate with a system that registers and compares the measurement data with nominal measurements and tolerances so the operator receives visual information on whether the results are acceptable.

The end goal for the project is to increase the efficiency and reliability for both the operator and the documentation process, as well as reduce the error occurrence. To maintain these goals the client has suggested the following list of sub-assignments to be reached by the end of the project period:

- Evaluation of digital instruments for measuring thickness.
- Create an interface from the measurement device to the computer system - cable or wireless.
- Construction of fixtures for positioning of parts for measurements, as well as eventual support tools.
- Evaluation of different MR Technologies and hardware.
- Develop software that comprises:
  - Identification of the object that is to be measured
  - Visualize guidelines on where to measure through MR HMDs
  - Registration of the measurement data
  - Documentation photos of the measurements

Kongsberg Defence & Aerospace (KDA) will contribute with test objects with relevant measurement needed for testing of the equipment. KDA will acquire and finance MR and measurement equipment.

2.2.1 Customer Requirements

At the project start the team has hosted a kick-off meeting with Jose Ferreira (internal advisor from USN), Alf Pettersen (the client) and an operator involved in the Final Inspect step in the Composites Department at KONGSBERG Aerospace. At the meeting the operator described the current process for the thickness measurement operation from start to finish, and together with the client, discussed their vision of a potential solution. Using the input provided during this meeting (Appendix D) the team has noted the following list of systems' requirements (sorted in no particular order):

- The system shall make the measurement process more time efficient than the current method.
- The system shall reduce the possibilities of human error.
- The system shall improve the recording and documentation of measured points.
- The system shall guide the inspector to every measurement point.
- The system shall be easy to use regardless of age, technical background, etc.
- The system shall not use Wi-Fi.

3 Technology

This section introduces the concepts of Extended and Mixed Reality and addresses evaluation and application of relevant technology such as Microsoft HoloLens, thickness measurement equipment and fixtures.

3.1 Extended and Mixed Reality

Figure 1: A picture of Pokemon Go in action

When the terms *extended reality* (XR), *virtual reality* (VR) and *augmented reality* (AR) are used in this thesis the team takes basis in definitions set by the *XR Association* (XRA). Microsoft, Google, Samsung, Oculus and other leaders within the industry [3] are part of this association. Another term that is used in the scope of this thesis is *mixed reality* (MR), which will be discussed shortly.

XRA defines XR as an umbrella term that

> *encompasses augmented reality, mixed reality, virtual reality, and other forms of alternate, expanded, or immersive reality applications, including those not yet invented.* [4, p.4]

In contrast, AR is an environment that extends the real world with a virtual layer. An example of an AR application is Pokemon Go, where virtual Pokemons are projected onto a real-world display as seen in [5, Fig. 1] AR software uses techniques to find the AR device's position relative to real life surroundings. [4]

Figure 2: Virtual reality in action

VR is an environment that replaces the real world with a digital three dimensional reality, in which users can orient using tracking technologies of the VR device. Tracking can either be done by six degrees of freedom (6DOF) or by three degrees of freedom (3DOF). Both types involve the orientation of the device in a 3 dimensional plane, while 6DOF type also tracks the position change, or movement, of the device. Some VR devices are also able to interact with the virtual surroundings, as seen in Fig. 2. [4]

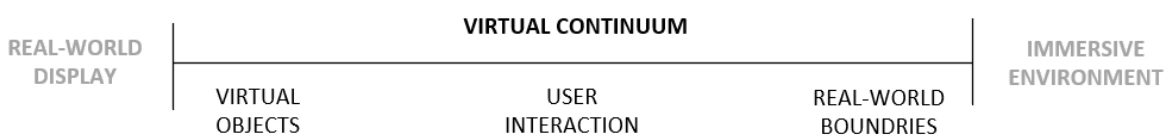


Figure 3: A diagram showing the virtual continuum

The term MR was first defined by Paul Milgram and Fumio Kishino in 1994 [6]. In doing so, they have introduced another concept called a *virtual continuum* (VC) [6]. As seen in Fig. 3, VC is a line spectrum describing different VR related environments where one extremum being real environments, for example a conventional video display, and the other being a total immersive virtual environment. MR is being defined as everything in between these two extrema where both a virtual environment **and** a real world environment is present in one display, thus categorizing all AR and the instances of VR where users interact with the scene as instances of MR.

3.2 Microsoft HoloLens - 1st and 2nd generation

Initially, the project was intended to be developed for HoloLens 2, the newest of the mixed reality headsets from Microsoft. It is the best option on the market in terms of technical specifications, as shown in Appendix 9, the comparison of HoloLens devices and two competitors: Magic Leap and NReal.

Unfortunately, due to a significant setback in the delivery time of the HoloLens 2, the client was not able to obtain the headset in time, and the team was left with a choice: wait for the HoloLens 2 and develop using an emulator or develop for HoloLens 1 which the team was given full-time access to. Given that the delivery time of the HoloLens 2 was not known, the team decided to proceed with the development for the HoloLens 1 knowing that an upgrade to HoloLens 2 would be required at a later point if the project's concept is to be adapted into the production process at KDA (Sec. 2.1).

One of the main reasons the team recommends using HoloLens 2 instead of 1 besides the technical advantages is ergonomics. Even though functionality desired by the client can be achieved with HoloLens 1, it is quite heavy with an uncomfortable fit-system and somewhat poor weight distribution. After testing it first hand, we do not recommend using it for a full working day.

The team had a chance to test HoloLens 2 at TechnipFMC. It has enormous improvements in ergonomics compared to its ancestor. It feels lighter and less bulky. Even though it is lighter by only 10 grams, it is the weight distribution that matters. With a new dial-in fit system designed for extended use, it feels more balanced with the gravity shifted back, closer to the centre of your head. Better balance of the headset, ensures among other things, less load on the nose, something that our team found to be one of the downsides of HoloLens 1.

Another significant improvement of HoloLens 2 is the possibility of flipping up only the front visor. It makes it much smoother and more comfortable to do then lifting the whole outer part of the headset as in the first version.

Last but not the least, another improvement that is worth mentioning is the field of view. At the first look, it seems that HoloLens 1 has a bigger clear front visor than HoloLens 2, but it is only a small portion of that visor that is used as a display. Holograms that are visible when looking straight ahead can disappear or get cut off when turning to the side. While the first generation has a 34-degree diagonal field of view, the second generation expands it to 52-degrees. Even though it still is not a completely immersive experience, it is a huge improvement.

The team would also like to point out how useful the eye-tracking feature of the HoloLens 2 is in terms of unfriendliness and accurate visual alignment. For more information see Sec. 3.3.4.

The team was set to develop a software for HoloLens 1 that needed to be somewhat version-independent / easily updated for use with the HoloLens 2. Main objectives to keep in mind regarding the device upgrade are processor/compile dependencies and versions of the development platform Unity, as explained in more detail in Sec. 3.3.

3.3 Software

This section contains research of relevant software available for *Universal Windows Platform* (UWP) application development and consecutive deployment on HoloLens device, as well as collaboration tools used by the team during development.

3.3.1 Development Platform, Environment, Framework and Tools

Unity is a real-time 3D development platform that is recommended by Microsoft for developing mixed reality apps. The competitor engine *Unreal* is coming in with the support for the HoloLens while Unity has been supporting it for years and thus is better supported and documented by Microsoft. [7][8][9] There are also a number of toolkits available for Unity development that make HoloLens development much easier, as for example *Mixed Reality Toolkit* (MRTK).

The MRTK is a cross-platform toolkit for building Mixed Reality experiences for VR and AR[10]. It is installed on top of the Unity and supports both HoloLens (1st gen) and HoloLens 2. HoloLens (1st gen) executes applications on an x86 processor in contrast to HoloLens 2, which uses an ARM processor (see Appendix 9). Unity 2018 LTS (Lomg Term Support) supports compiling ARM32 apps while Unity 2019.x supports compiling ARM32 and ARM64 apps. Even though ARM64 applications are generally preferred because of the difference in performance, MRTK v2 always guarantees support for Unity 2018 LTS but does not necessarily guarantee support for every iteration of Unity 2019.x. [11] That is why the team chose to start developing in Unity 2018 LTS.

Visual Studio is used to build and deploy an application to HoloLens. For this project the newest available version was used, which, at the time of writing, is Visual Studio 2019.

HoloLens can run Windows 10 and UWP applications [12]. *Scripting backend* is a framework that powers scripting in Unity, and UWP supports only two: *.NET* and *IL2CPP* (Intermediate Language To C++) [13]. .NET scripting back-end is being deprecated in Unity 2018 and is removed in Unity 2019, so IL2CPP was selected in order to keep this project maintainable and up-gradable in the future [11]. With IL2CPP, Unity converts intermediate language code from scripts and assemblies to C++ before creating a native binary file for the chosen platform. Luckily, Unity asserts that IL2CPP increases performance, security, and platform compatibility of the projects. [14]

3.3.2 Communication and Collaboration Tools

The team used several communication and collaboration tools during the project, and could really appreciate the variety, cross-platform compatibility and integrability of the digital tools during the COVID quarantine (see Sec. 4.3.1).This section will briefly introduce the tools the team has used during the project.

Jira by Atlassian [15] was, among other things, used as the project management tool and for user story mapping. It was one of the team's most essential administrative tools and is described in more detail in the chapters to follow (see Sec.4.1 and Sec. 5.2) .

Communication tools became more important than ever during the COVID quarantine. One tool that is worth mentioning besides Zoom, that was used for video conferencing, is a platform for instant messaging Slack [16]. The messaging tool became the main communication arena during the quarantine and made team members more available to each other during working hours. In addition to direct messages, the team could categorise communication into channels as seen in [Fig. 4], which made it easy to organize and look up messages by topics at any time.

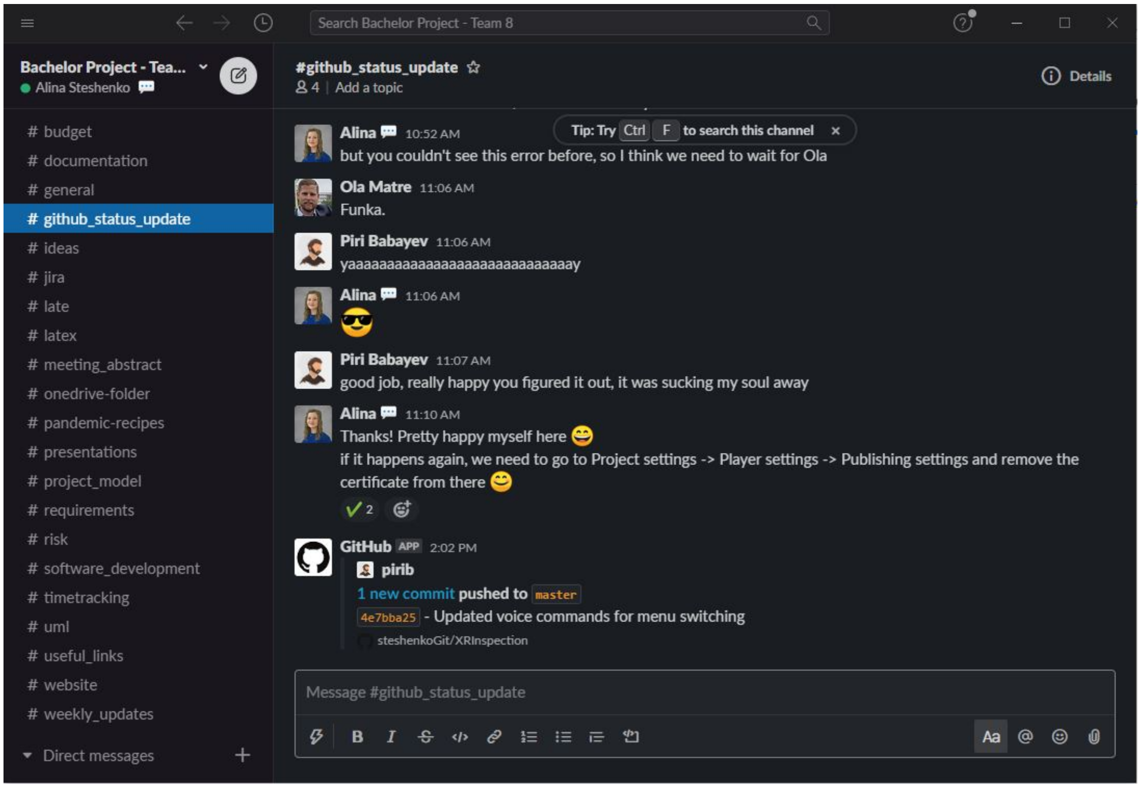


Figure 4: Instant messaging tool Slack

Slack also features "linking" of other tools. One that was especially useful to connect to Slack was GitHub [17].

GitHub is a cloud-based code hosting and management platform for Git repositories that allows teams to work together on projects. The version control software Git that GitHub service is build on, is an open-source system that was started by the creator of Linux, Linus Torvalds. It keeps record of changes to files, who made them, and what were the changes. It also stores the state of the project at each *commit*. A commit is a snapshot of the projects current state, capturing the integrity of the projects files. Collaborators can anytime roll back to an earlier version where a commit is made if needed. [18]

By connecting GitHub to Slack and creating a dedicated channel for GitHub updates, the entire team was notified of any commits to the project with the title and description of the commit as seen in Fig. 4.

Branching is another useful Git-feature that the team took advantage of. When creating a GitHub repository, it has one default branch, also know as the master-branch. Each developer with write access can create a copy (a branch) of the current project in order to isolate development work and, for example, work on individual feature or functionality without affecting other branches in the repository. The branch can later be merged with the master-branch. [18]

Even though GutHub provides a graphical interface it does not provide a visual and intuitive overview of the repository's branches. Luckily, GitHub together with their partners has created a *Student Developer Pack* [19] which gives access to a tool called *GitKraken* [20] for free.

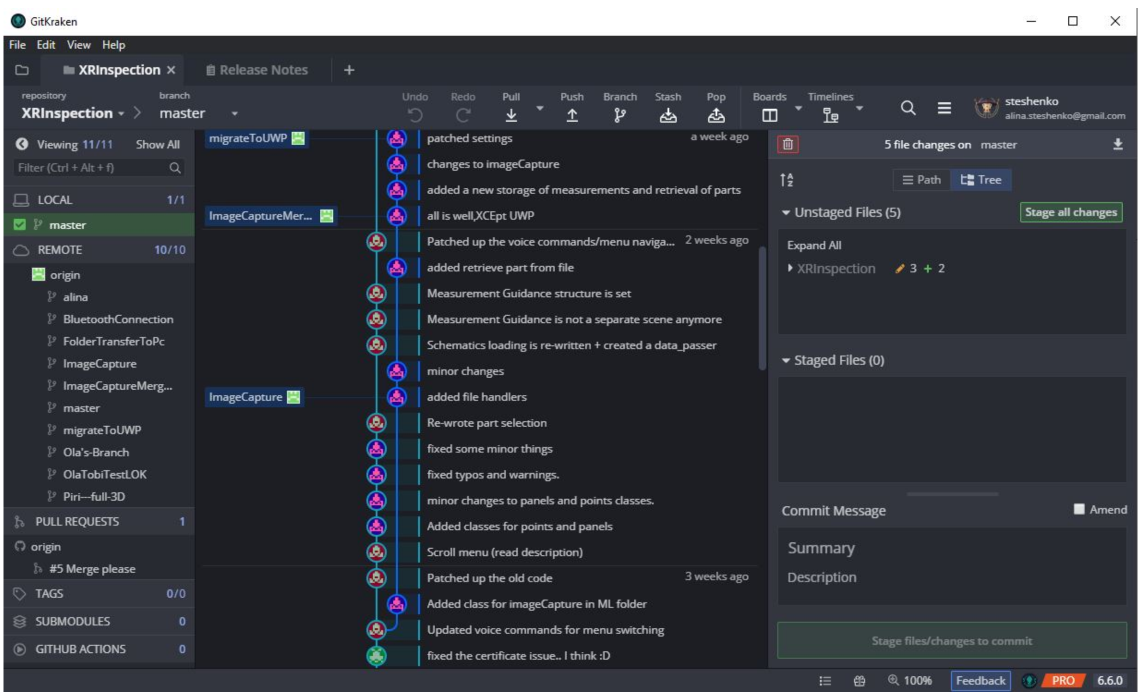


Figure 5: Git client with GUI: GitKraken

GitKraken is a Git client with intuitive Git *Graphical User Interface* (GUI). By connecting the app to a GitHub account, team members had access to the project's repository with in-app merge tool and visual commit history (as seen in Fig. 5) which made Git workflow much easier.

3.3.3 Object Recognition and Tracking Software

One of the main objectives of this project is holographic guidance through measuring points on a part/panel in the real world. Image recognition and tracking software is needed to achieve this in order to detect and track a part/panel in space and estimate where the measuring points are.

At the time of writing, MRTK doesn't support image recognition yet. Microsoft has a Software Development Kit for QR code detection and tracking on Windows Mixed Reality devices but it is supported only by the Hololens 2 [21].

Image recognition for the Hololens is going to be natively supported in Unity in the future as announced at the Unity Blog this January [22].

So at the time this paper was written there were two classical tools for image recognition and tracking that could be used with HoloLens 1 and Unity: *Vuforia* [23] and *HoloLensARToolKit* [24]. The ARToolKit was researched and tested first because it is free, smaller and less resource-demanding than Vuforia Engine that has many additional features and is quite expensive.

3.3.3.1 HoloLensARToolKit

`HoloLensARToolKit` uses square marker patterns to overlay AR content on the real world. It is an experimental project that integrates ARToolKit [25] with HoloLens. It is written in native C/C++ code (with an additional wrapper for development on Unity) by Long Qian, a Ph.D student at the Department of Computer Science at Johns Hopkins University with research focus on Augmented Reality and Medical Robotics [26]. The source code is open for use in compliance with GNU Lesser General Public License (LGPL v3.0). [24][27][28]

An application developed with the `HoloLensARToolKit` does not require internet connection. `HoloLensARToolKit v0.3` was released on 19th of April 2020 and according to the documentation supports both HoloLens 1 and HoloLens 2. [24]

The ARToolKit has two features that could be relevant for this project. First one is the concept of marker recognition that can be customized and used to uniquely identify parts. It is described in more detail in Sec. 6.2.1. Second feature is placing of AR content into physical world that can be used for overlaying of digital content onto a physical part and is described in Sec. 6.3.2.

The following conditions influence the recognition performance:

- Light conditions.
- Distance. The distance between the marker and the camera influences the performance. The current version works only within +/- 1m from the marker [29].
- Clear sight. Covering up marker with hand or other objects will interrupt tracking.
- Complexity of the marker. More complex markers are harder to recognize and track.

3.3.3.2 Vuforia

Unity versions until recently have a built-in Vuforia Engine built in package. It will no longer be natively distributed and directly supported by Unity as of Unity 2019.3 [30]. Vuforia Engine is a platform that can be used to develop and deploy AR software to places digital content onto physical objects. It is cross-platform, supports both generations of the HoloLens devices and is intended for both commercial and industrial uses. [31][23]

An application developed with Vuforia does not require internet connection unless it is using a cloud database which was deemed unnecessary for the XRI project. [32]

While ARToolKit can detect, track and place AR content onto an image, Vuforia Engine offers additional functionalities of recognition, tracking and placing digital content onto a physical **object**. Vuforia has many features that allow attaching digital content to specific objects [33], and for this project, *Model Targets* feature [34] especially interesting.

According to Vuforia, Model Targets can be used for creating AR experiences on large or small complex objects by using their CAD data. It uses edges of the object for both detection and tracking, and objects can be recognized instantly but not simultaneously. The size of an object can vary from a small toy to a building. [23].

There are, however, some drawbacks. One of the downsides of Vuforia is the pricing. A Vuforia Development License Key is free, but a subscription is needed for deployment.

Deployment license options can be found on Vuforia's pricing page [35] and at the writing moment, the Pro-plan is the only one including Model Targets feature. The price of the Pro license is not announced and is specified by Vuforia upon request only. The next-best plan, *Basic + Cloud*, costs $99 a month and should give some idea about the pricing of Vuforia's top licensing.

Computer vision is still in its infancy and all object recognition and tracking solutions on the market, including Vuforia, suffer from similar problems that have to be taken into consideration to optimize the performance:

- Variation in object's surface is required and objects with shinny and reflective surface are not recognized. Objects that are transparent or are mainly black will not do either. Objects in a single uniform color or without any texture in the surface can be detected but are said to be difficult to track. [36]
- Light conditions.
- The speed the headset is moving at when facing the object.
- Distance between the object and the camera.
- Orientation and complexity of the object's shape.
- Object tracking works best on static objects.
- If the object is symmetrical it is likely to not be able to identify the angle at which you are looking at the object.

3.3.4 HoloLens Camera Calibration

There are several camera calibrations that address different causes/problems of inaccurate hologram viewing experience. If you are new to the technology of head-mounted displays for AR applications, it might be confusing to differentiate between them and knowing where to start. This section addresses the tip of the iceberg but should clarify some of the doubts and give a good starting point. This paper addresses three camera calibrations:

- Camera calibration due to distortion (for the ARToolKit).
- Camera-projection calibration (see Sec. 6.3.2.2).
- Camera calibration due to user's unique eyes.

Camera calibrations are not required but is highly recommended for accurate image and object tracking. Both Vuforia and HoloLensARToolKit have camera calibration guides that can be used for calibrating HoloLens' external camera. [37][38]

To understand how this type of camera calibration works, we first need to explain how the coordinate systems are organized. For simplicity, a 2D image/marker is used as an example, [39, Fig. 6].

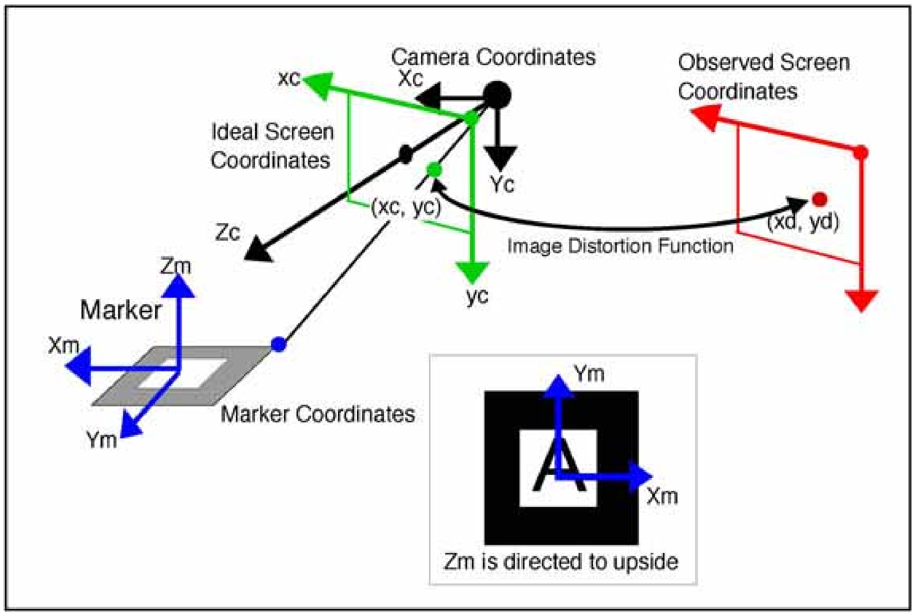


Figure 6: Computer vision coordinate systems and distortion

The marker coordinates Xm, Ym and Zm are related via ideal screen coordinates xc and yc to Camera coordinates Xc, Yc and Zc. Transformation between marker and camera coordinates can be described with following formula [40]:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & W_x \\ V_{21} & V_{22} & V_{23} & W_y \\ V_{31} & V_{32} & V_{33} & W_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \tag{1}$$

V-elements define the rotation and W-elements define the translation.

Observed screen coordinates *xd*, *yd* can deviate from the ideal screen coordinates due to distortions (see Fig. 7) and perspective projection matrix. And this is where the image distortion function comes in, in which the distortion factor can be set and tuned by camera calibration. [40]

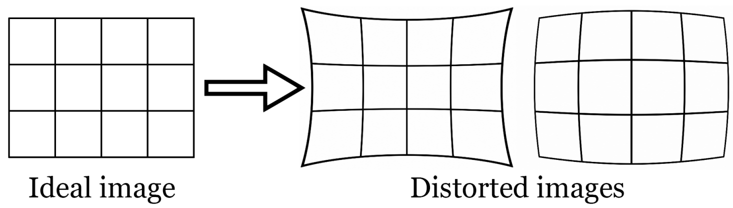


Figure 7: Image distortion

The following example is from calibrating a HoloLens front-facing camera using *OpenCV* [41] by following HoloLens Camera Calibration guide [38] provided by Long Qian (see sec. 3.3.3.1). Fig. 8 shows three images:

1. The first image is the original image of a chessboard marker captured with HoloLens using `HoloLensFrameSaver` application and extracted via Windows Device Portal [42].

2. The python `calibrate.py` script finds contours of the chessboard marker in the original image and extracts the edges and corners of the marker.

3. The distortion caused by the lens is corrected using the formula mentioned above by the same `calibrate.py` script.

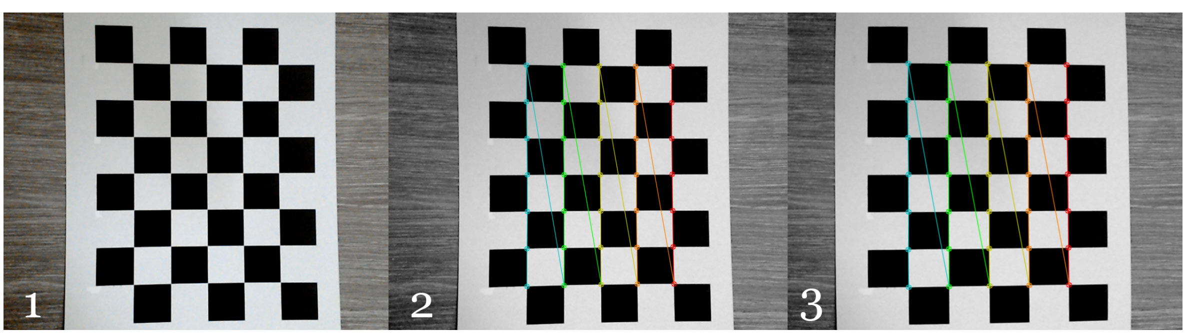


Figure 8: Corrected image distortion

A human eye might not perceive the difference right away. When something is really a straight line but the camera picks it up as a curve it is clearly distorted by the lens. Notice how the edges of the chessboard in the first two images of the Fig. 8 are curved inwards, and how the lines are corrected after calibration.

The *calibrate.py* creates a calibration file of two different types: *data.json* and *data.yaml*. The calibration file is device-specific and the calibration should be done for every device. In this case the OpenCV format calibration was converted and used with ARToolKit as mentioned in Sec. 3.3.3.1.

This type of camera calibration fixes relatively minor distortion issues but is really important in cases where precision matters as it does in this project.

Sec. 6.3.2.2 provides more information about manual calibration/camera-projection calibration and offsetting virtual content.

Another factor that affects hologram viewing experience is the users unique eyes, so for hyper accurate visual alignment you need technology that is able to detect the users eyes. Both HoloLens 1 and 2 are provided with calibration tools pre-installed on the devices but they use different calibration technologies and techniques. The HoloLens 1 uses *interpupillary distance* (IPD) while the HoloLens 2 uses eye-tracking and doesn't need IPD. [43]

Every user must do the calibration routine. This process is more user friendly on HoloLens 2 than HoloLens 1 since it initiates automatically when using device for the first time, saves the users profile and recognises the user next time he/she puts on the headset. Calibration data is stored locally on the device and is not associated with any account information. [43]

HoloLens 1, on the other hand, prompts the calibration only on set up of the device and then the user must manually start the calibration process if desired. The data stored is only from the latest calibration, meaning that the calibration must be done every single time you switch between users.

3.4 Thickness Measurement Technology

This section is an overview on technologies used for thickness measurements in this project. It describes the current solution of a conventional digital micrometer and the proposed new solution implementing a communication interface with the HoloLens-device.

3.4.1 Current Use

As stated in Appendix D, the customer uses conventional micrometers (which will be discussed shortly) with digital display to measure the thickness around the edges of panels used for aeroplanes. The panels are made of *carbon fiber reinforced plastic* (CFRP), with one smooth side (outside of panel, called OML) and one rough side (inside of panel, called IML). When cured, the CFRP panels are very stiff and brittle. The panel-sizes measures to 0.3-2m, with 5-100 points to be measured on each panel. The way the operators use the instrument is by taking up to several measurements in close proximity to a predefined measuring point and then take the average of the measurements. The average value is then compared to the tolerance interval defined for that point. There is currently no interface between the instrument and computers, but Customer has ordered a digital micrometer from Insize (article 3539-253A, as seen in [44, Fig. 9]) with transmitting and receiving equipment (article 7315-30 in [44]) that can be interfaced with a conventional computer.

3.4.2 Existing Methods

Different micrometers are evaluated on specifications such as *accuracy*, meaning the maximum difference between the true value and the measured value when applied within the range of the instrument, *range*, the interval from the minimum value and the maximum value where the instrument has predefined data documentation, and *resolution*, the smallest step the instrument can detect. [45]

3.4.2.1 Micrometer

A micrometer is a type of caliper that uses a calibrated screw to measure an object from two sides, displaying either mechanically or digitally. In the case of a mechanical display, the measurement is read from the sleeve of the instrument. While in the case of a digital display, the measurement is read from an LED/LCD-screen. [46]

Figure 9: Insize 3539-253A digital micrometer

Modern micrometers typically have an accuracy of 4-10μm for ranges of <100mm, with a resolution of 10μm [47].

When evaluating alternatives for the current micrometer used by our client, the team has focused on finding instruments that is capable of wireless communication. Due to security demands at the site of operation, short-range wireless communication is the only acceptable wireless alternative. Since HoloLens uses Bluetooth for short-range wireless communication, the team's focus has been on finding a micrometer that are capable of connecting to Bluetooth.

Since HoloLens 1 does not support any USB-connection with external accessory devices, the micrometer needs to be connected directly to the built-in Bluetooth-receiver as opposed to connecting the micrometer via an external Bluetooth-receiver that is connected to the micro-USB connection of the HoloLens [48].

As alternatives to the current Insize micrometer (see section 3.4.1) the team found only one alternative that was capable of measuring as deep into the panels as needed as well as connecting via Bluetooth. Insize was found being a better alternative due to price.

The team also assessed different Bluetooth-transmitters for the project, where every alternative found was working on only a limited range of micrometers or demanded a unique type of receiver to work. The latter is also the case for Insize. [49][50][51][52]

The alternatives the team has evaluated are:

- Modify current Bluetooth transmitters so they are able to connect directly to HoloLens.
- Connect via an external computer to HoloLens.
- Build our own Bluetooth transmitter with the ability to connect to HoloLens.

The first one is difficult, if not unfeasible, because the Bluetooth-transmitters built for micrometers are, according to Insize (e-mail correspondence), communicating on non-standard broadband. The second alternative raises some security issues, since it requires additional software on the operators computer and the HoloLens. The last alternative requires some work, but can be achieved using off-the-shelf components for prototyping.

3.4.2.2 Universal Asynchronous Receiver/Transmitter

Universal Asynchronous Receiver/Transmitter (UART) is a protocol for serial communication. A UART-protocol is a series of on- and off-states, or bits, that come in packets (as seen in Fig.10) with one start-bit, a number of data-bits (most commonly eight bits or one byte), potentially a parity-bit and one or two stop-bits. The bits are transmitted in a certain *baud rate*, which means the number of bits transmitted per second, and is often in the range of a couple of thousands bits per second to several hundred thousands of bits per second. [53]

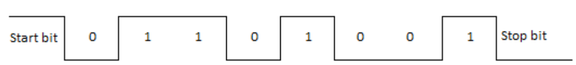


Figure 10: UART-packet containing the data 0x69

3.4.2.3 Bluetooth Connection

For this project, *Bluetooth Low Energy* (BLE) will be used as wireless communication. BLE is like traditional Bluetooth, a type of short-range radio communication developed by *Bluetooth Special Interest Group* (Bluetooth SIG) that functions on a frequency band defined to be 2400-2483.5 MHz. The difference is that BLE is a more energy effective method built for long lasting communication such as with sensors or computer peripherals (keyboards, mouse, joystick, etc.). [54][55][56]

The *Human Interface Device Profile* (HID-profile) is a Bluetooth profile that is mainly developed to be a wireless alternative to wired USB-connection where devices are used for human interaction. HID uses many of the same principles as USB-devices. Of the most common usage is keyboard and mouse which commonly uses USB-connection to connect with computers. In a network using HID-profile, units are defined as Bluetooth HID devices or hosts, where the devices are the peripherals of the hosts. In this project, the micrometer with Bluetooth-transmitter works as a device while the HoloLens being the host. The devices and hosts connects using a *Service Discovery Protocol* (SDP) and communicate via HID interrupt and control channels where HID input, output and feature data is transmitted and received as reports. [57]

3.5 Fixtures

A fixture is a general term for a tool or device that holds or support a product during a working operation. It is commonly used in a variety of manufacturing processes and vary very significantly in size and rigidness depending on the products form, shape, size, weight, etc.

For the project assignment it may be necessary to fix the composite parts in certain positions so that the HoloLens always knows where the part is and in which orientation it is in. Therefore, to quickly see if using fixtures is necessary and identify what types of fixtures is applicable to the final solution the group split it into three different parts:

- No fixture
- Dynamic fixtures
- Static fixtures

3.5.1 No Fixtures

As seen in Appendix D, the client described that the current solution for thickness measurements does not involve any fixtures to keep the panels in place while controlling them. The panels are placed loosely on top of a regular table and are easily moved on top of these. To be able to handle the panels gently and not cause any damage the tables follow a short set of rules:

- The tables need to be covered with cardboard and bubble wrap to create a soft surface;
- The panels can never lay partly on the outside of the tables, if the panel is too big for one table, two tables must be connected so there is no risk of the panel falling off;
- Panels can never lay on top of each other without a bubble wrap sheet separating them (this is more applicable for transporting panels between the shelves where they are stored and the work station).

3.5.2 Dynamic Fixtures

Dynamic fixtures is a very broad category covering possibilities for fixing a part in several different positions/orientations. They may not always be the most suitable solution for high precision operations such as machining, probe measuring, etc.

Examples of reasons to use a dynamic fixture:

- Give part multiple lockable positions;
- Works for smaller and or lighter parts;
- Can be designed to work for multiple different parts;
- Locks part for easier processing.

3.5.3 Static Fixtures

A static fixture is useful when an operation is done by either an operator, machine, etc. demands high precision and no movement on the part that's being worked on.

Examples of reasons to use a static fixture:

- Low tolerances;
- Repeatability;
- Lock part in a single position;
- Works for big and or heavy parts.

Using static fixtures are very useful for some operations, but might limit others. For some operations it might be necessary to move or rotate the part to certain positions for easier access and this can't be done with the static fixtures. An example of a static fixture can be seen in [2, Fig. 11]

Figure 11: Static Mechanical Fixture

4 Project Management and Development Process

This section introduces theory and application of project model elements, principles of project management and product development that build up the framework of the hybrid model chosen for this project.

In Fig. 12 we present a simplified diagram for our project. Rather than bursting into the explanation of what each of the blocks represents and how one should interpret their relationship to each other, we will opt to ”show, don’t tell” technique. We will examine each concept represented here in detail, using them as building blocks culminating in the big picture.

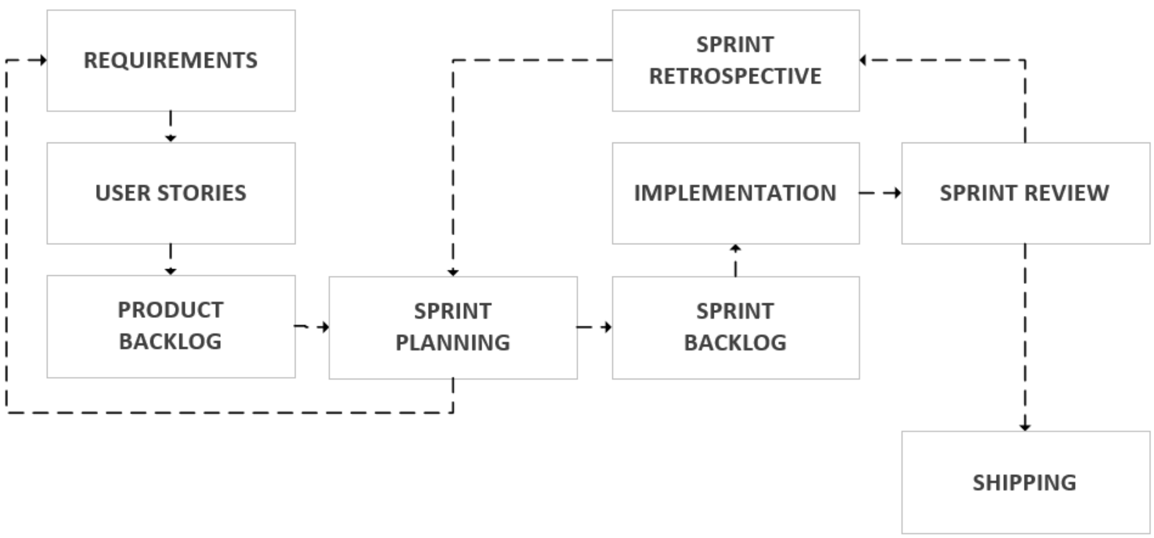


Figure 12: The project model

4.1 Scrum and Agile Methodology

After a round of discussions the team has decided to adopt Scrum and Agile combo as our project model. An Agile Team Working Agreement was established early in the project to encourage teamwork, self-organization and accountability. It is continuously refined by the Team through the project's duration and is included as Appendix **??**.

Agile methodology is a practice based on an iterative and incremental product development and testing, where system requirements and potential solutions evolve through development life cycle of the project. [58]

When one mentions Agile, Scrum springs to mind immediately, and for a good reason. It is a lightweight process framework that divides the project into shorter cycles called Sprints. [58] Having shorter cycles makes the process more adaptive and responsive to changes, resulting in a better end solution. The idea is to frequently deliver a working software which is the primary measure of progress. Early and continuous delivery of software and the responsiveness to requests supports customer collaboration and satisfaction with the product. [59]

Before delving deeper into the details of Scrum and Agile, we will shortly mention the tools we have decided to use to maximize the efficiency of these methodologies. Based on our team member's experiences our first stop was Trello. [60] Being a flexible tool for project management Trello fitted our needs quite perfectly for the first 3 weeks, but due to the lack of support of Scrum artifacts (discussed shortly after) it has been replaced by Jira. [15] Explicit support for management of these aforementioned artifacts such as Sprints, Backlogs, and User Stories to name a few, tipped the scales in favor of Jira.

The following discussion will consider vital components, or as we call them artifacts, of our Scrum model along with screen captures displaying their usage in the tools described earlier. Important Scrum artifacts that were adapted to the hybrid project model, as well as their application with the help of a management tool called Jira, are explained in the following chapters.

4.1.1 User Stories

User stories are important Scrum artifacts that are used to define functional requirements at a high-level. They are short, simply described from a point of view of our customer, and in our case, we are using the popular convention that takes the form: *As a [persona], I [want to], [so that].* The goal of User Stories is to facilitate discussion within our team, and give our team a better understanding of what the needs user wants the solution to take care of. [61]

4.1.2 Product Backlog

The Product Backlog is a dynamic list of all known things that need to be done to achieve the project's objectives. It includes administrative tasks, implementation oriented subtasks, user stories and other artifacts that are continuously refined and updated, evolving along with the product as the team gains a better understanding of the desired outcome [58].

Figure 13: Product backlog in Jira

4.1.3 Sprints

One iteration of the development cycle is called a Sprint. Sprint length chosen for this project is just one week long - the decision has been made to keep it short so the team can keep up with changes and adapt to new situations on the fly. [62]

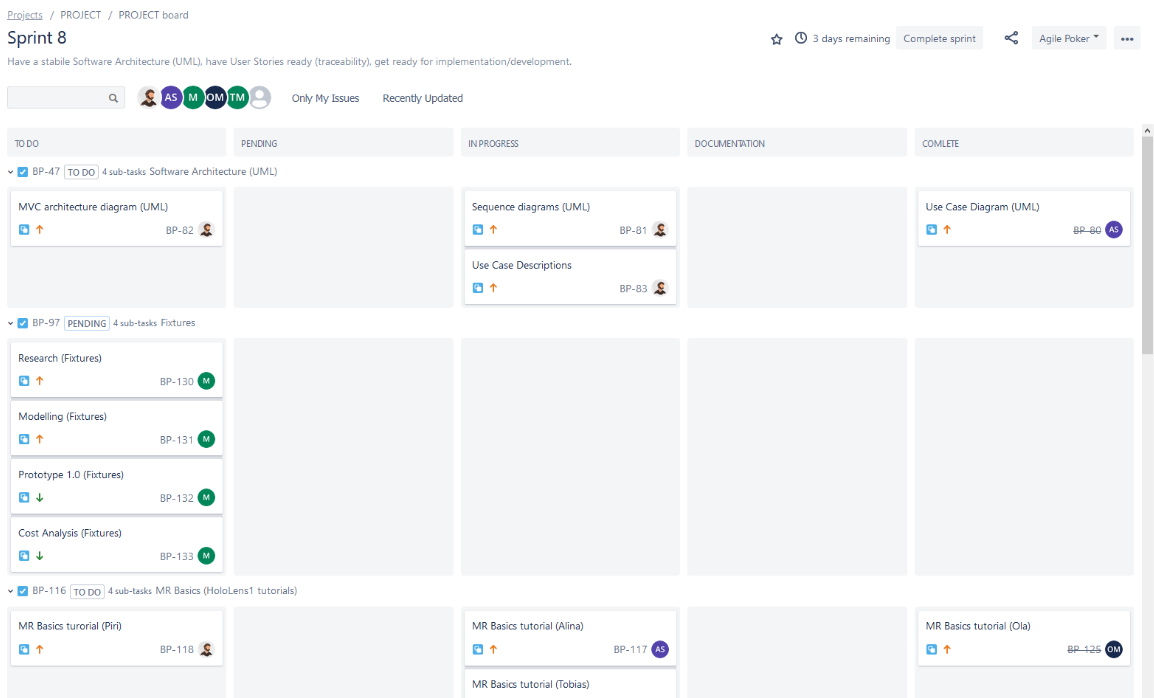


Figure 14: Sprint board in Jira

4.1.3.1 Sprint Backlog and Planning

At the beginning of every sprint, the team has a Sprint Planning meeting. During the meeting, we set the objectives for the sprint, estimate workload and choose items to work on from our Product Backlog. At this meeting, user stories are decomposed into manageable, "bite-size" tasks, and the subset of the Product Backlog selected for the sprint is called a Sprint Backlog.

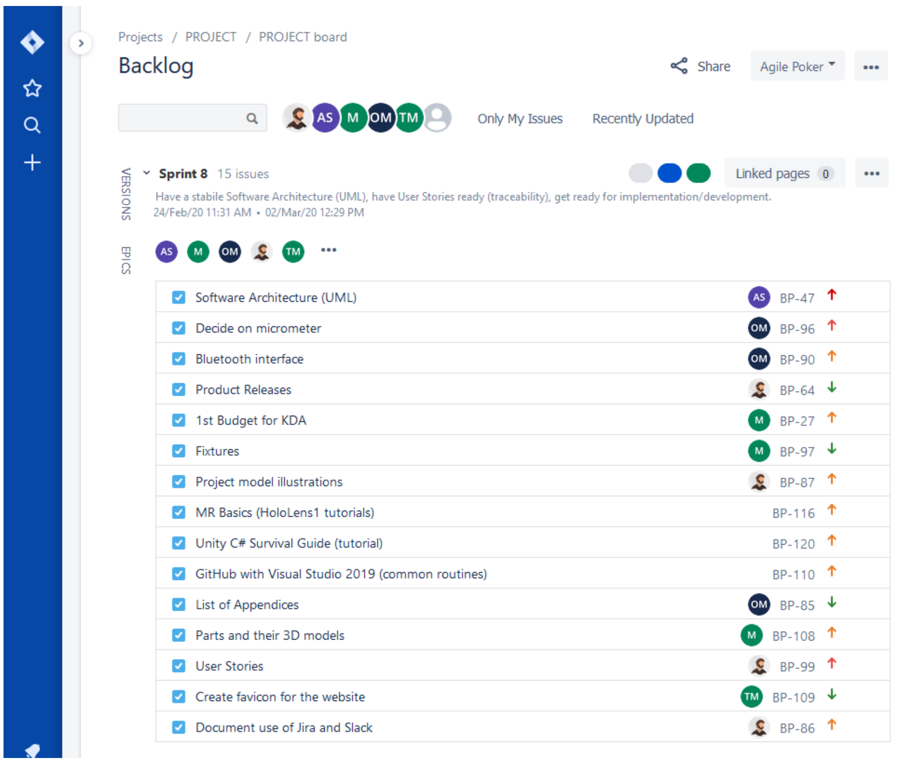


Figure 15: Sprint backlog in Jira

4.1.3.2 Sprint Review and Retrospective

Sprint review is a meeting taking place in the end of the week where the team discusses past week's sprint and reviews it. It helps the team to estimate better the workload for the next sprint and brings the team to the same page about the state of the project. Retrospective, on the other hand, is a weekly sit-down that gives the team an opportunity to reflect on, as well as share their mood of, the past week.

4.2 Behaviour and Test Driven Development

This chapter addresses software development methodologies that make the foundation for the testing framework applied in this project as well as it's development model.

Behaviour-Driven Development (BDD) is an iterative development cycle that can be integrated within the Scrum framework, and focuses on functionality of the system from a user perspective as the basis for software tests. [63]

For this project the team chose a BDD test case format of scenario-oriented acceptance criteria as shown in Fig. 16 for a given functionality, which are expressed as Features derived from User Stories (Sec. 4.1.1). Every test case describes a set of behaviours that the customer can expect from the system. Much as the story itself, acceptance criteria is keyword-based and line-oriented, and is written in plain English language so it is easy to read for the customer and other "non-technical" participants of the project.

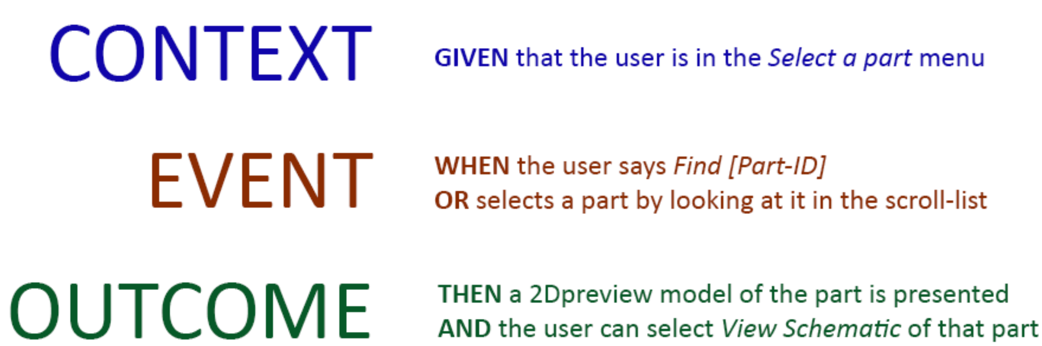


Figure 16: BDD - Test case example

Test-Driven Development (TDD), also known as "test-first programming", is one of the widely used practices in Agile Development [64]. It has a short development cycle similar to BDD, but in contrast, is programming language oriented. In order to understand what TDD is we first need to explain what Unit Tests are.

Unit Test is a test of the smallest **testable** part (unit) of a software in isolation. The main objective of Unit Testing is to validate that each component of the software performs as expected [64]. Note that doing unit testing does not mean you are doing TDD, however, TDD is not possible without unit testing.

In traditional unit testing the test is written **after** the piece of code to be tested, while TDD approach is to write a unit test **before** writing any implementation code.

DevOps published an interesting article by a blogger Dave Farinelli where he described what we believe should be the core mindset of every quality-oriented developer team:

> "Quality software gets built when there's an acknowledgment that quality test code should receive the same amount of attention and resources as quality production code, as they are equally essential in development." [65]

[66, Fig. 17] illustrates three basic steps of the TDD cycle [64]:

1. Write a single unit test that will immediately fail.
2. Write the necessary (nothing less, nothing more) code to pass the test.
3. Refactor the code to improve the implementation.

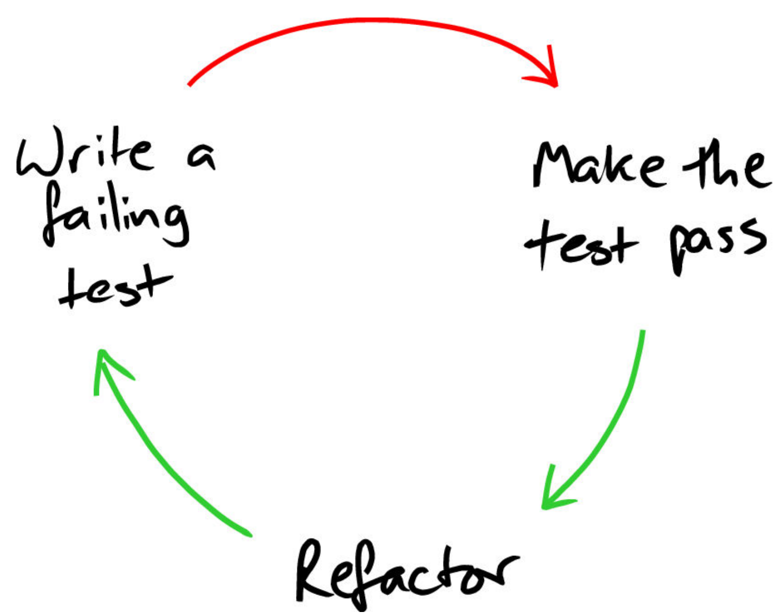


Figure 17: The fundamenal TDD cycle

A common misconception is that BDD and TDD are adversaries, when in fact they complement each other. BDD focuses on the right functionality of the system, while TDD focuses on right implementation of that functionality. By combining both BDD and TDD our development team does not only focus on building a system that does the right thing but also that the system does that thing right!

Fig. 18 illustrates the development model created for this project. It is implemented in Jira and described in more detail in Sec. 5.2.

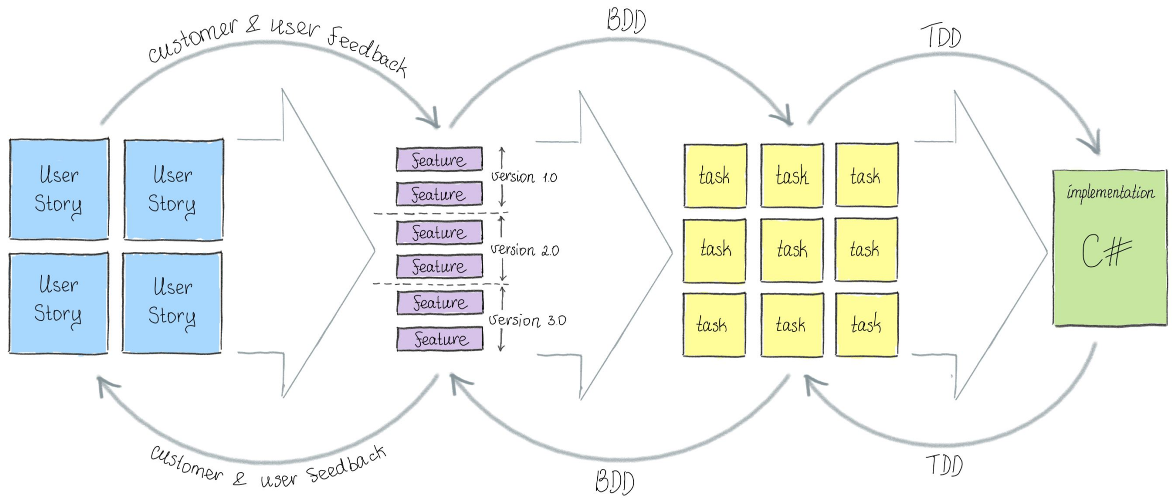


Figure 18: The development process model

4.3 Risk Management

Risk is "any activity, occurrence, or decision in business or personal life that involves uncertainty" [67]. The managements of risks is restricted to uncertain events that can have adverse effects on the project. "The practice of risk management includes planning the approach, identifying and analyzing risks, response planning and implementation, and ongoing monitoring of risks" [68] . Risk management is also a term that means different things to different businesses [69]. This projects' description of risk management will be further described in this section.

The team created a template for risks that sets several demands to writing the description of each risk. This ensures that every event that proposes a risk, is thoroughly examined and understood, while their structure is consistent. A baseline understanding of each risk is advantageous when making an appropriation to the probability and severeness of consequences. Each risk is described by the event triggering it, the source of the event, the assets affected, the consequences of that event, its probability and its degree of implications against the projects' goal. Furthermore, every risk should come with a strategy for avoidance and mitigation [68]. As seen in the risk management template Appendix F the strategy is specified as pre- and post- event.

Different risks have been sorted into categories. The categories considered are defined in the following way:

- Project risks - the ones associated with the success of the project that are not covered in the other groups;
- Product risks - those associated with how the product works;
- Technical risks - directly associated with hardware or software
- External risks - those that our team can not prevent.

The Risk matrix is used to present qualitative and semi-quantitative data, but not mathematically precise data [70]. Risk is subjective meaning a specific event can be considered as having severe consequences by one person, while the second person might find those consequences less severe [67]. This observation led to a matrix using three levels of consequences and three levels of probability of occurrence to the aforementioned events, in contrast to using numerous levels of nuance. Consequences are divided into three colors, namely, green: Barely noticeable on the end product, yellow: significant damage to the product, and red: product may become obsolete or unusable. The three levels of probability, in contrast, are - low: most likely not going to happen, medium: as likely to happen as not to happen, and high: most likely going to happen.

An initial thorough risk analysis followed by iterative reassessments fits well into an agile method of work with frequent sprints. Definition of a risk statement includes describing the source, assets affected, the event(s) triggering the risk and the consequences this event has on our objectives [70]. A combination of probability and severity of consequences is a usual approach for prioritizing the events [69]. This project has prioritization divided from one to three where three is the risks the team should pay closest attention to, as seen in Fig. 19

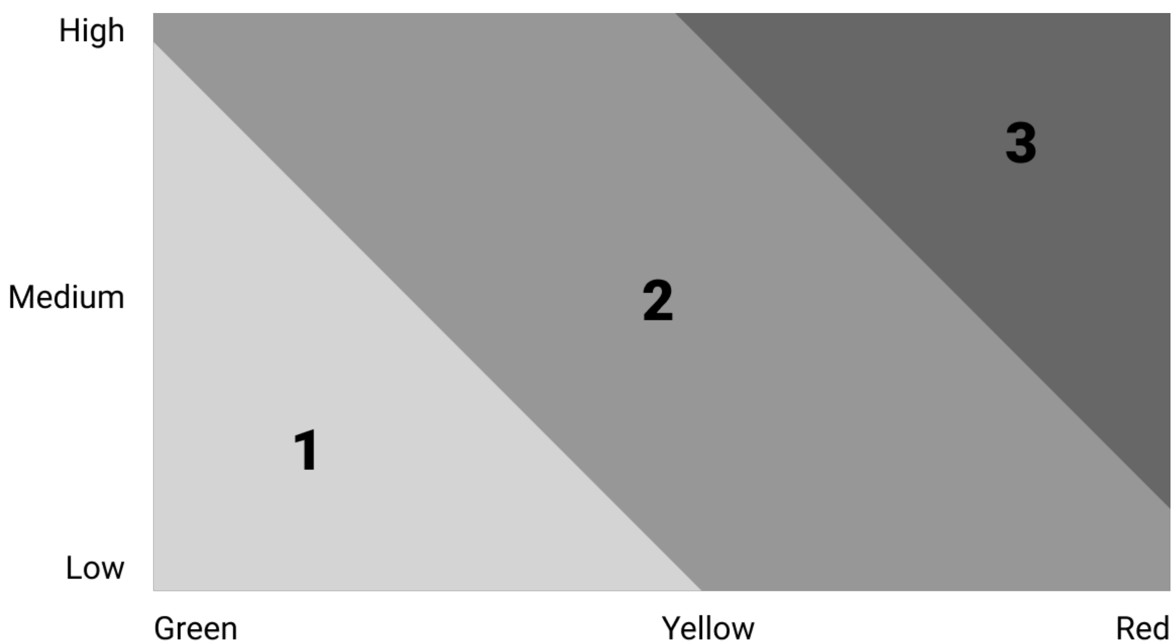


Figure 19: Risk management - prioritizing

Management of risks includes activities to identify and assess risks, describe them, analyse their attributes, implement a planned response. The team's process for risk management is: Identify new risks, define their statement, define their probability and severity, lay strategy, document actions taken, as seen in Fig. 20. An assessment is executed at the start of each sprint. Iterating this process makes the team able to assess and attend risks that are amendable at the given stage of the project [68].

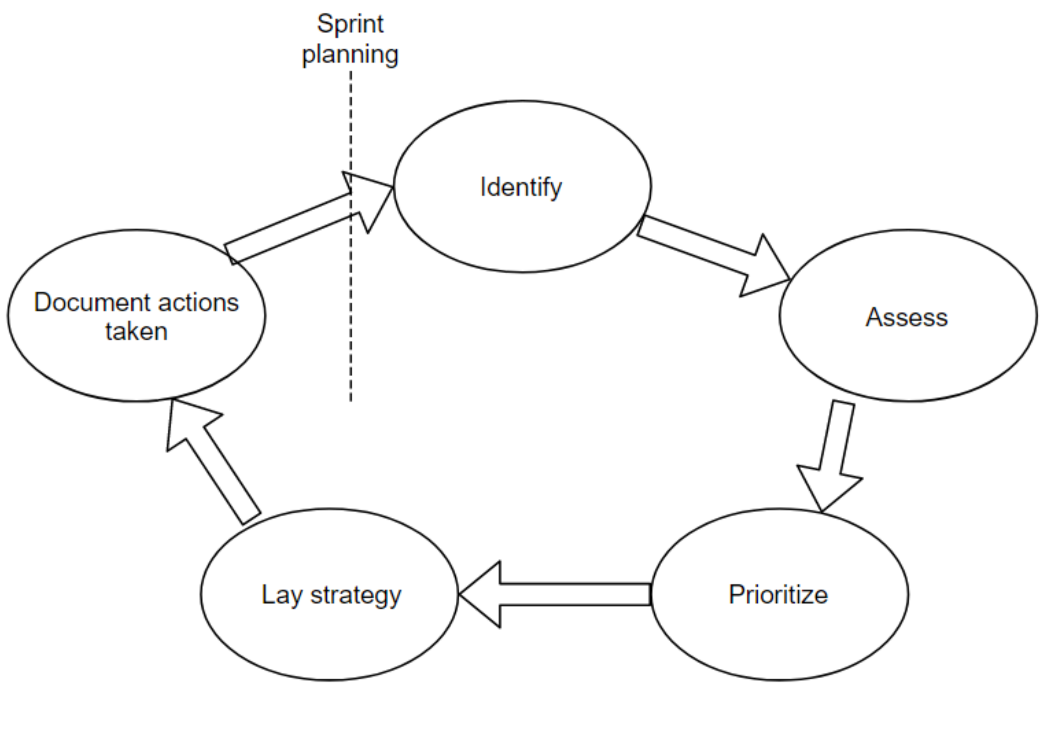


Figure 20: Iterative risk management

As the project developed events were added and removed from the risk table and their statements were altered. The last version of the risk table which covered the conclusion of the project is seen in Appendix F.

4.3.1 Quarantine

In March 2020 all universities in Norway closed by instructions of the government as a measure to mitigate the impact of a newly discovered virus Covid-19 [71]. People are instructed to stay isolated as much as possible. Many firms keep their employees at home and there is a risk for shops relevant to our project to close. Necessary equipment for our project are not always available when we need it and key personnel, like employees at USN and the Kongsberg Group is only available through remote communication tools. The team has discussed precautions to mitigate the effects of unavailable equipment and personnel.

4.3.1.1 Equipment

All features of the software needs to be tested on the HoloLens device and some features need continuous interaction with the device to be developed. Since the team is not allowed to physically meet there is the challenge of only one team member having access to the HoloLens. The team uses GitHub services as described in Sec. 3.3.2 which allows continuous sharing of code and online collaboration, so the team member with the HoloLens can carry out tests of software that the whole team is working on. This proved to be a big hurdle due to the time it takes to deploy software to the HoloLens. Always depending on the team member who had the physical HoloLens lead to longer stretches of work where essentially two people had to work on the same task when this would have been unnecessary under normal circumstances.

There is also risk of delay in processing orders and shipping time due to the quarantine as well, therefore equipment such as micrometer, hardware and mechanical components is being ordered as early as possible. One of the features of the first product release, as seen in Sec. 6.1 was that the measurement value should be displayed in the HoloLens. The micrometer the client is using was not available for version 1 of the system, therefore this feature was postponed to version 3.

4.3.1.2 Communication

Prior to the Corona crisis the team often relied on personal interactions when updating, explaining and visualising project outtakes for each other. This method of problem solving is no longer possible, which demands each individual of the group to be more transparent with their work flow and progress.

Adapting to a period where online communication is the only solution has not been a huge issue for the team since it was facilitated for remote working due to group members living far off campus. Which means, the group had many of the facilities for this situation already in place.

Social interaction is a useful part of teamwork as it can alleviate stress, create communion to the project and lower the bar for sharing challenges. The team is bound to work separately, in their individual home spaces, this is a contrast to their usual work flow and creates some additional heft because of the forced environment. We adapted to this by increasing interactions on Slack and Skype.

4.3.1.3 Presentations

The original format of presentations was changed to delivering a video of the team presenting the project. This format is not familiar to the group members, and it requires different techniques to capture and hold the attention of the audience, since there is no face to face, two way interaction with the audience etc. . The team uses Camtasia to record our parts. The main frame in the picture will be illustrations or slides of what is being presented, while there will be a smaller frame with a video shot of the presenter to keep the presentation personal. These parts are shot individually and then edited together to a single video.

5 Software Architecture, User Story Mapping and Design

This chapter is about the journey from customer requirements to stable software architecture and product design. Each of the three sections addresses different aspects of the process with the same goal - understand customer's needs and design a product that fulfils those needs.

5.1 Software Architecture

As the previous chapters have illustrated the project first and foremost is a software solution. Given that success of the bachelor's project relies heavily on how the solution competently conforms to the requirements and standards we have set, it is paramount that the team has invested adequate amount of time and effort into designing the software architecture. Our main attention thus has been diverted to Unified Modelling Language (UML) - a visual language that helps teams to evaluate and design object-oriented systems [72].

Building a product that is to become a solution to a fairly complex problem is difficult. This discussion is outside the scope of this thesis so this paragraph would keep it to two reasonably important points UML helps to solve: helping the team to achieve the right understanding of the solution and building the software foundation for it. We will shortly discuss these points below and later in more detail how each individual UML diagrams help. [73]

Working in teams is bound at some point to reach the moment where mental images held by different team members diverge and the famous phrase "Wait, that is not what I had in mind" is inevitably spoken. These moments need to be embraced and not shunned as they add clarity to the common understanding and push the development of the solution in the right direction.

One of the most dreaded moments in software developers' lives is the very beginning of the development - the steps taken now decide the outline of the system. Reverting back the changes and making amends into the existing framework becomes more and more costly as the team progresses into development process. [74] The importance of this step cannot be exaggerated and the colossal investment of time and effort in it is fully justified. Having said that it is not surprising that the team has encountered difficulties in the process of the decision making - the team was always torn between two extremes: on the one hand including a lot of details in the diagrams needlessly complicates the system and made it harder to read and understand, on the other simplifying the system left it being ambiguous and vague. The final diagram represents what we consider to be the golden middle.

5.1.1 Use Cases

Use Cases are a good starting point in any systems' architecture design - they are a simple, straightforward way of presenting of what is happening in the system [72]. They also help derive the understanding of how the system functions on a more technical level.

The diagram has an *Actor* - the party that interacts with the product, it is the user of the product and in our case it is an Inspector. The bubbles linked to it represent actions that can be taken by the actor to interact with the system. The word *Include* on the arrow indicates that a part of the system from which the arrow is pointing is contained within the pointed subsystem the user is taking an action upon. [72]

Use Case diagrams will be presented first, along with the short description of the process for each of the *Base Case* - the bubbles that are directly linked to the actor. The descriptions themselves are presented as short stories - this format helps us shape the right understanding of the process, and how we should approach the software design.

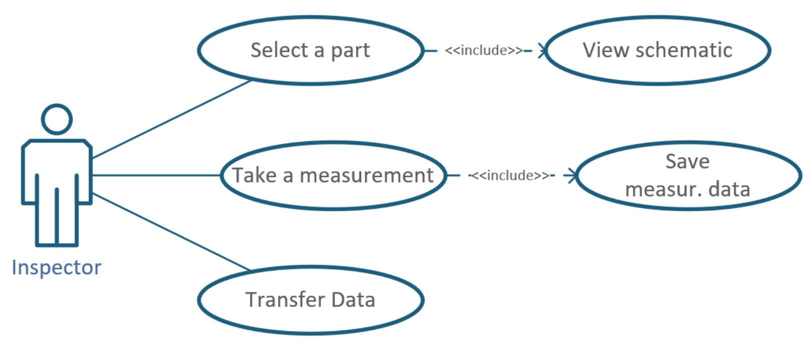


Figure 21: The main use case diagram

The main UML Use Case diagram includes all active actions the user can take. Descriptions of the Base Cases are presented on the next page.

1. **Select Part**:

   Inspector puts on the HoloLens and runs the Extended Reality Inspection software. He/she is then presented with a choice on how to select a part to work with next – they can choose to Resume the unfinished work on a part, choose a part to start measurement from scratch, or choose from a list of previously done work. They can do so by interacting with the holographic menu or using voice commands, or by scanning a barcode found on the schematics of that part. The system then displays a holographic schematic of that part with relevant information, such as, part ID, number, shelf number and thickness measurement tolerance of measuring points.

2. **Take a Measurement**

   Inspector chooses a point and uses a digital micrometre to measure it. The measurement data is sent over by Bluetooth to the headset and the Inspector reviews it during the entire operation. The system calculates and stores the average measurement value. After performing a satisfactory amount of measurements the Inspector finishes measurement of the point using either using voice command or hand gestures.

3. **Transfer Data**

   Once the inspector wants to transfer data to a stationary computer, they select "Ready to Transfer" in the main menu and the system generates a report of all the finished parts.

The team argue that *Select a Part* includes a lot of additional critical information, so to show its significance a separate diagram has been created, presented below.

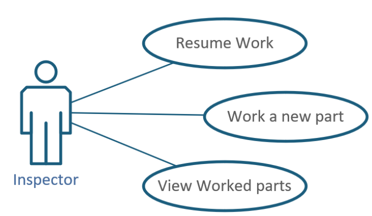


Figure 22: Use case diagram - Select a part

5.1.2 Sequence Diagrams

Use Case diagrams represent static relationships between the actor and the use cases, but they do not facilitate the design of object-oriented design. That responsibility is reserved for Sequence Diagrams. These diagrams are used to show how different classes interact with each other by sending messages. Sequence Diagrams are related to a particular use case and show only classes that are used in a given interaction. They do not present a complete picture, but give an outline of how classes interact. [72]

The Sequence Diagrams are presented by Base Case and show how the actions performed in the Use Case diagrams work out in the solution.

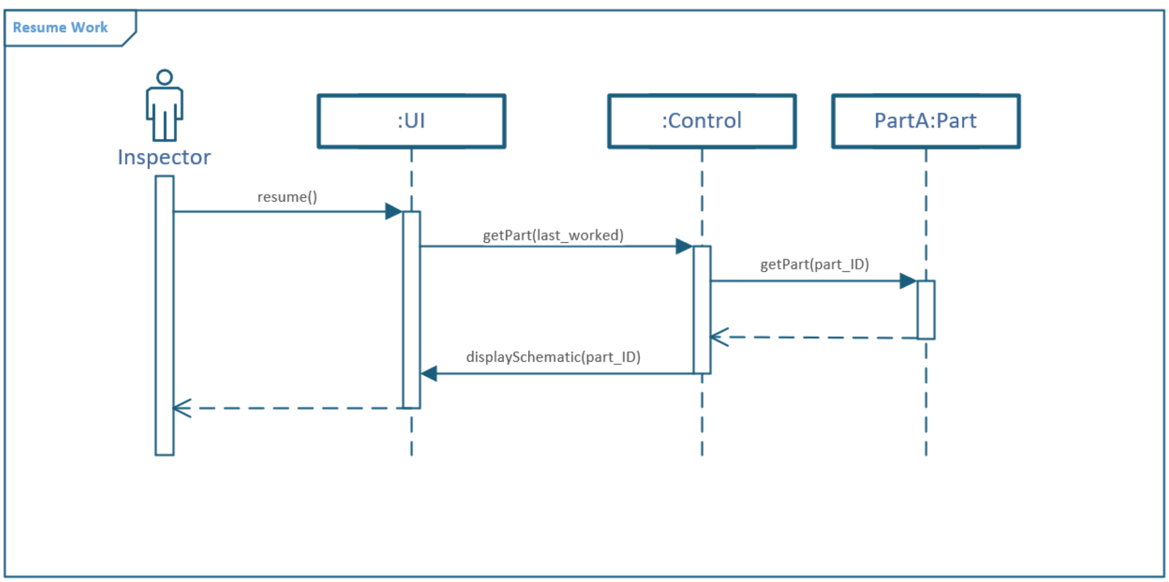


Figure 23: Sequence diagram - Select a part, Resume work

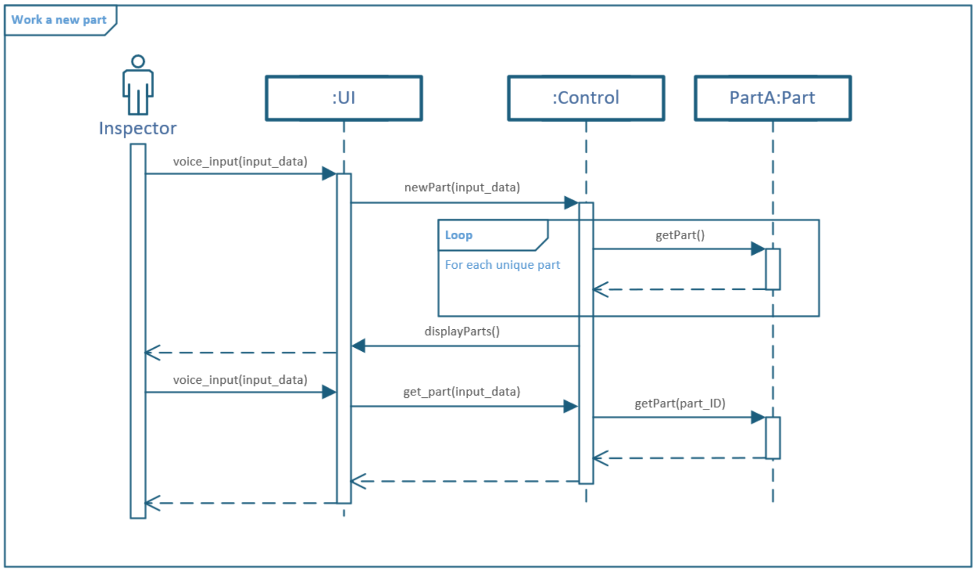


Figure 24: Sequence diagram - Select a part, Work a new part

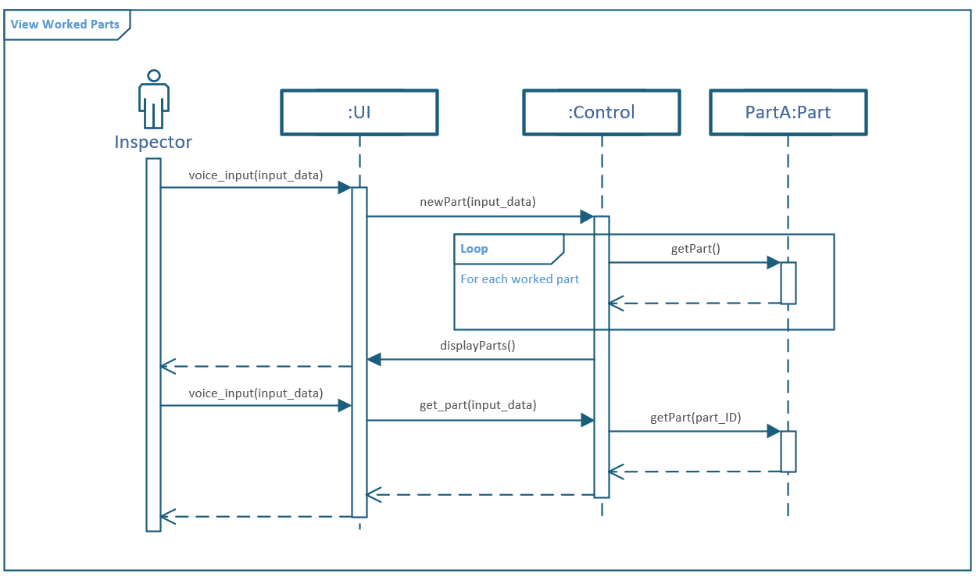


Figure 25: Sequence diagram - Select a part, View worked parts

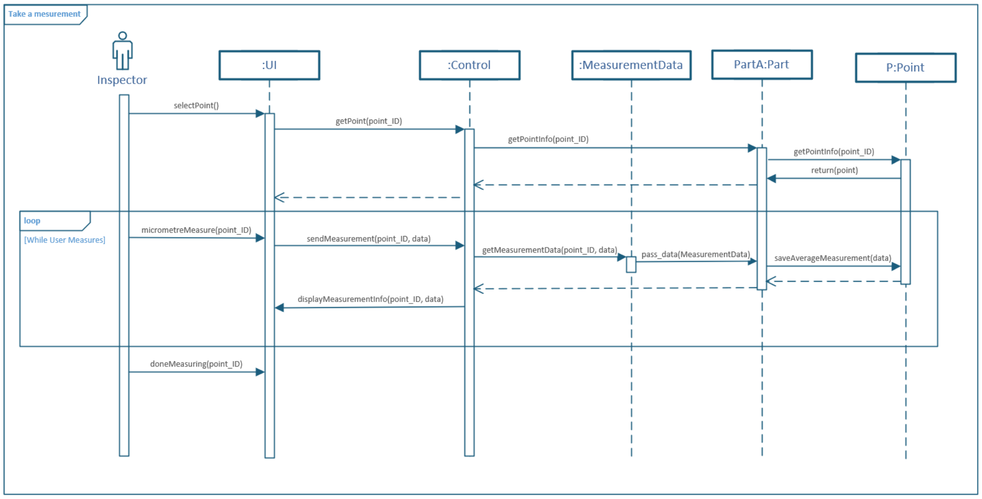


Figure 26: Sequence diagram - Take a measurement

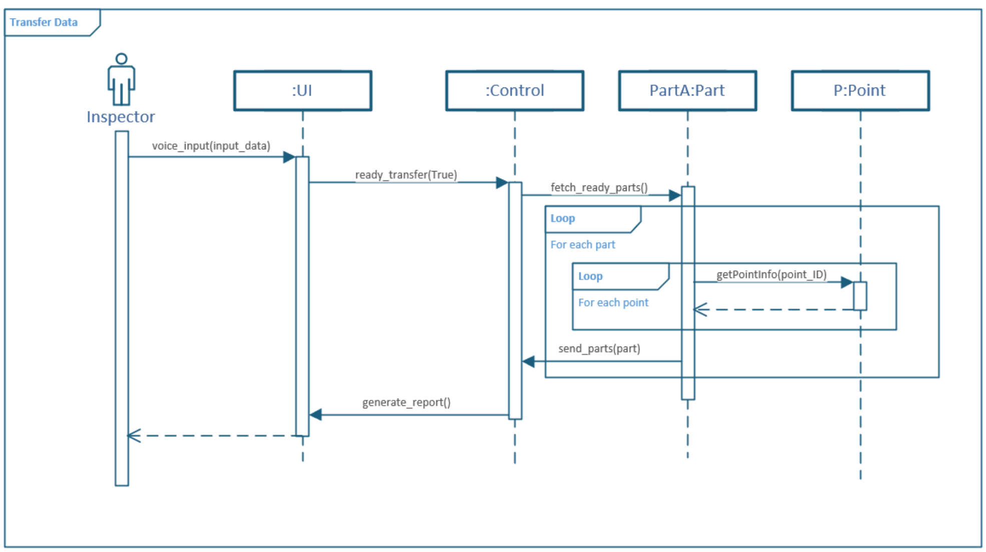


Figure 27: Sequence diagram - Transfer data

5.1.3 Model-View-Controller Architecture

Model-View-Controller Architecture is the cherry on top of the rest of the diagrams. It's simplicity is deliberate, main purpose being to simply put all into perspective. It consists of two parts: to display Data Elements in the software and the connections between them.

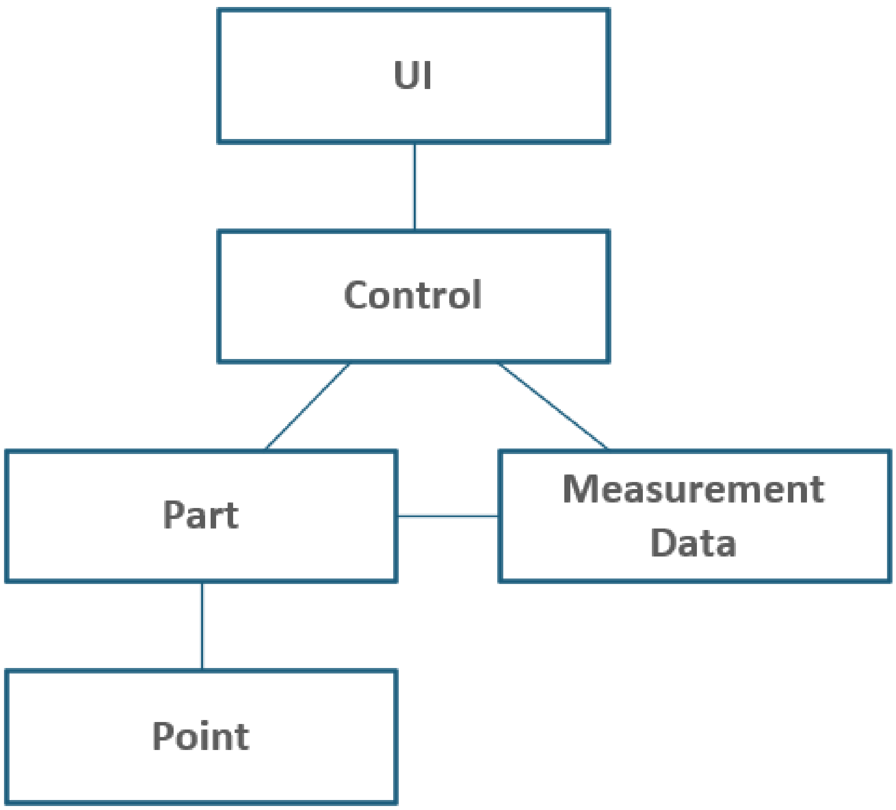


Figure 28: Model-View-Controller architecture

5.2 User Story Mapping

As mentioned in Sec. 4.1.2, the Product Backlog can contain all things that are needed to be done to achieve the project's objectives. Meaning that the Backlog could have everything from simple administrative tasks such as *"Fix page numbering in the document"* to more complex Scrum artefacts such as User Stories representing a functionality or a feature to be implemented. This is not very practical as the big picture of the product development is easily lost when mixing administrative and technical tasks in a flat backlog.

Initially, this was intended to be solved by having two backlogs: a Product Backlog for the product development and a Project Backlog for everything else regarding the project. However, this alternative introduced another potential challenge - keeping track of two different Backlogs, as well as two separate Sprint Boards, is inconvenient since it makes it harder to keep an overview of the total amount of work in a given sprint and introduces a risk of overhead in human resources.

The solution was a Jira add-on by Easy Agile called User Story Map [75]. It provided not only a way of keeping an overview of the product development without separating it from the rest of the project's tasks, but it also gave the team an intuitive, collaborative and visual way of understanding and defining user's journey with the product.

5.2.1 The Backbone and The Skeleton

Figure 29 shows the first draft of the User Story Map made by the team.

At the top is what is called the *Backbone* of the map. It provides the general structure, captures the high level activities the user will accomplish while using the product and represents "the essential capabilities the system needs to have". [76] The cards "hanging down" are the "ribs" of our skeleton. They are User Stories that represent the functional requirements (Sec. 4.1.1).

The vertebrae of the Backbone are user activities that have a narrative flow from left-to-right [77]. In this case, the activities were derived from the discussions the team had while working on the Use Case Diagram where the development team talked through each user action (or activity) in order to capture the functionality of the system under development.

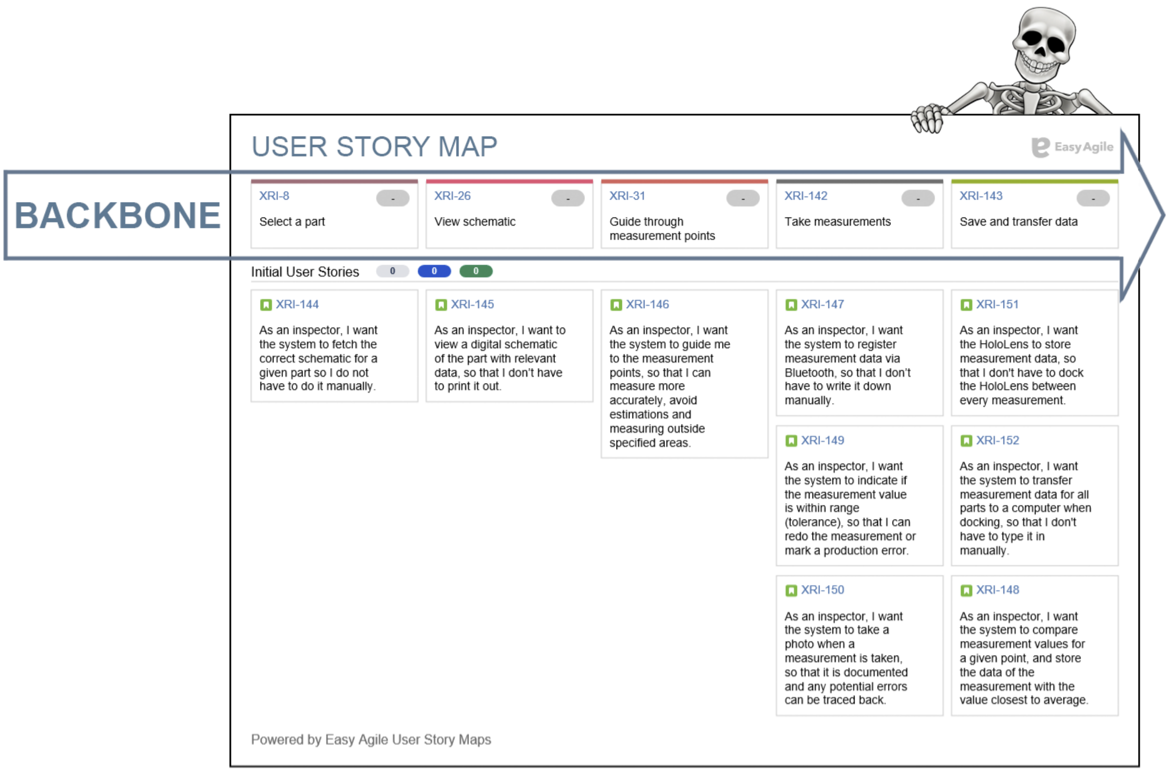


Figure 29: User story map and it's backbone

The map was reviewed and approved by the customer and an user representative, before the team moved on to the next step.

5.2.2 Features and Release Planning

After establishing a stable structure of functional requirements, the team focused on deriving features and acceptance criteria needed in order to meet them.

The map was split into three release slices, and features were derived from the initial User Stories as shown in Fig. 30, supporting incremental release strategy of Agile development (Sec. 4.1). While the backbone just "is", the features in the narrative flow within each product release version are the "walking skeleton" which describes the smallest possible module that could be build to give end-to-end functionality [76].

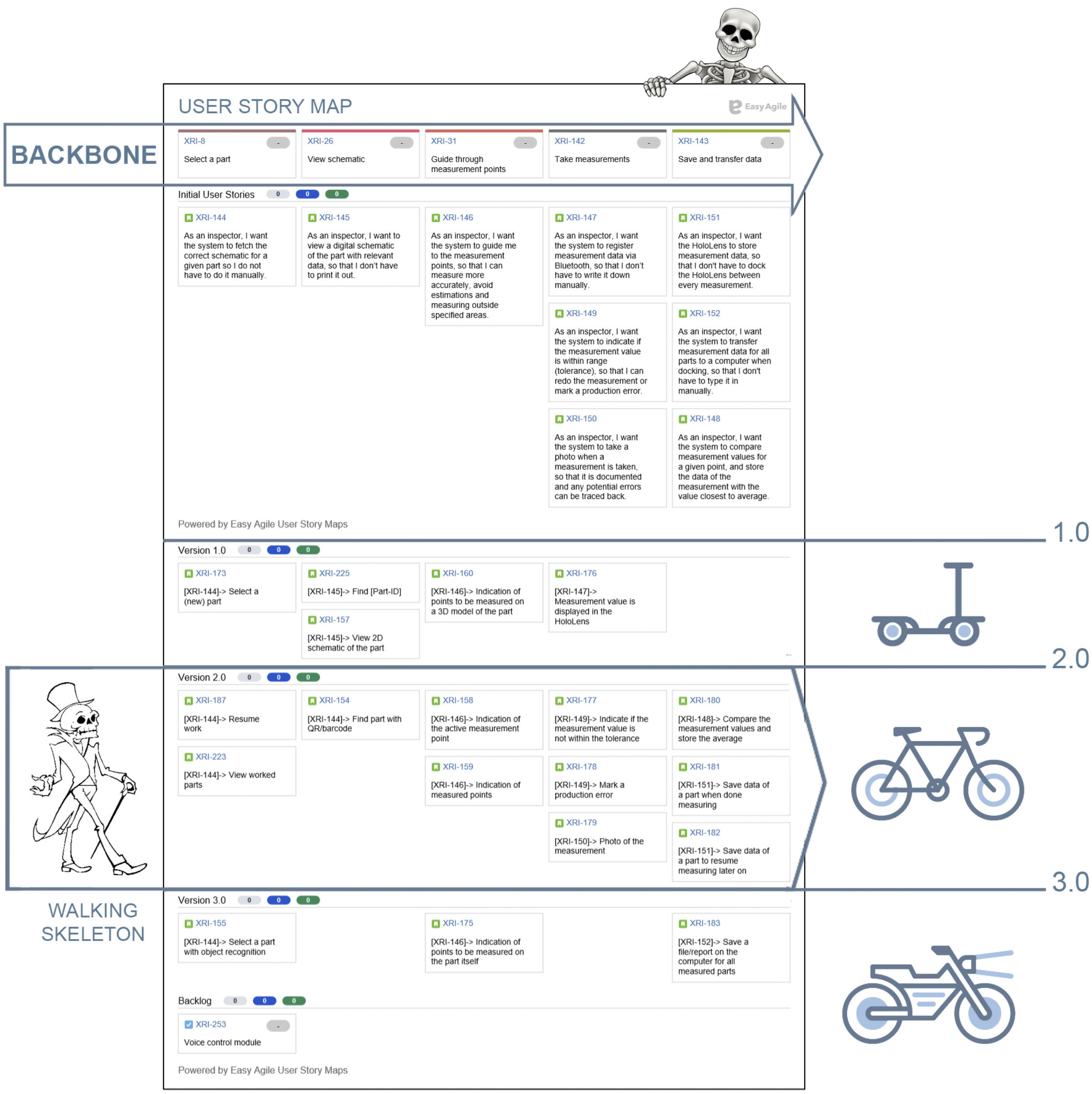


Figure 30: User story map with release versions

Every feature has a behavioral test case in a format of acceptance criteria and was split into manageable, more specific tasks (Fig. 31). This ensures traceability from the requirements to the implementation and back.

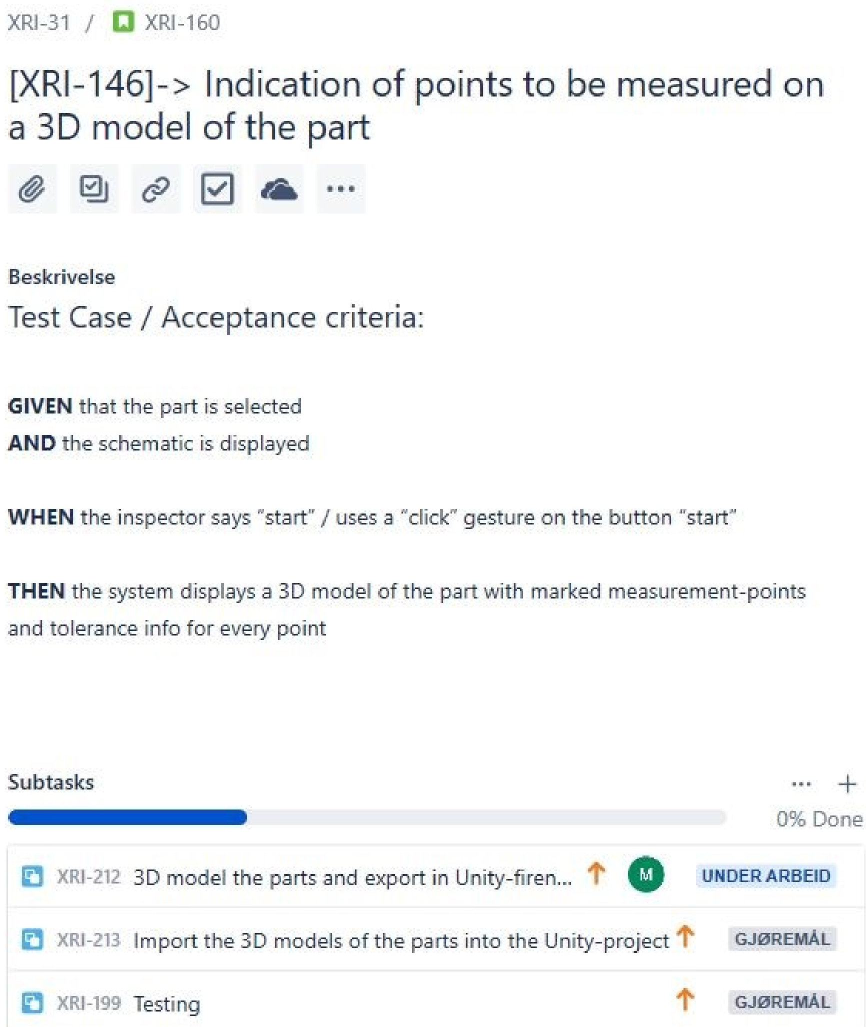


Figure 31: Example of a feature in Jira

The product map by Easy Agile also supports dividing the map with sprint into "swim-lanes", allowing the team to plan implementation of the features by sprints and avoiding a flat backlog (Fig. 32).

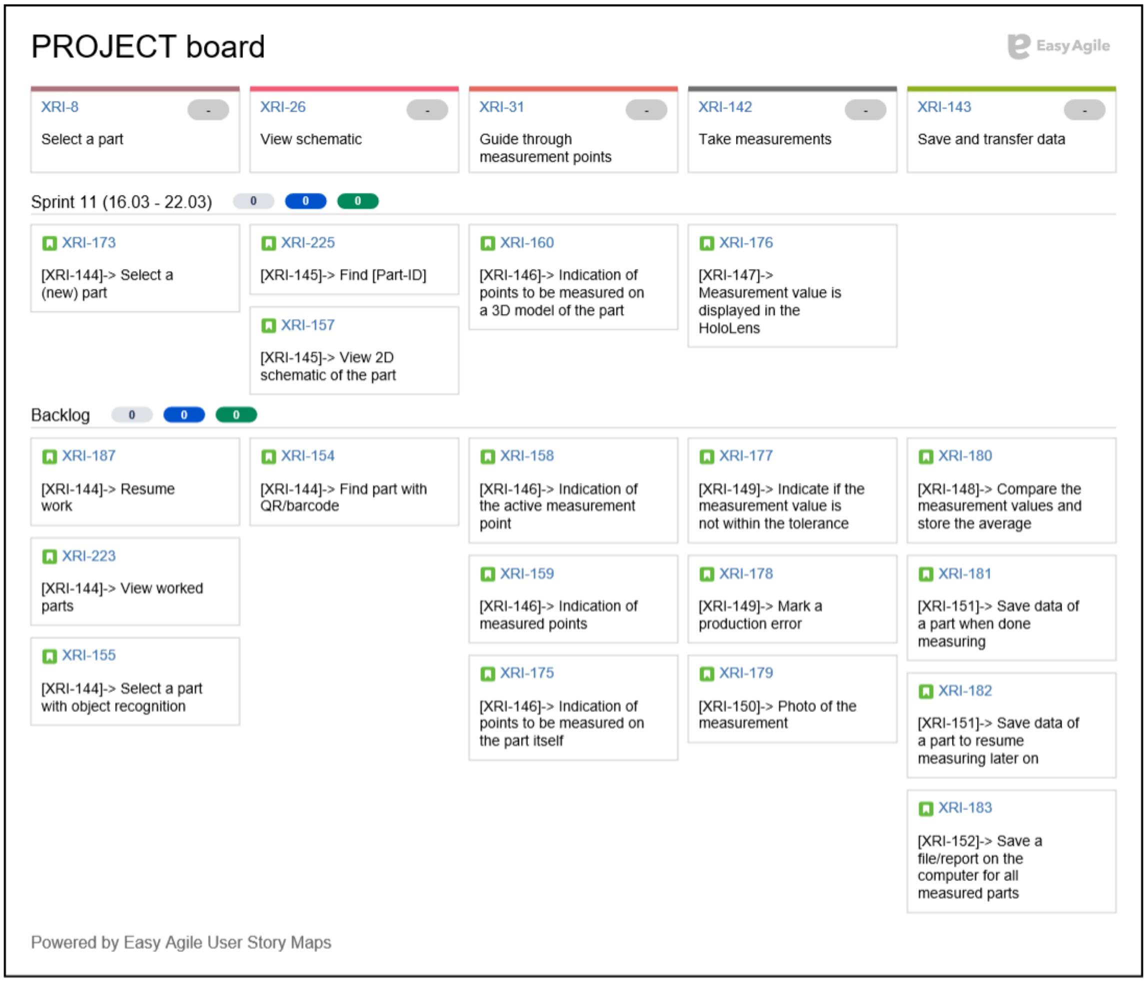


Figure 32: Sprint view of the product map

5.3 User Interface Design

Having thought about the functionality of the solution the next step was to start charting out a rough user interface of it. The reasoning to make this step next one was to understand whether team's reasoning up until this moment translates well into a practical application and to make sure that the problem with mental images discussed in UML chapter is once again embraced. The goal was to create a design that puts team's previous work together and make a user- and Mixed Reality-friendly design.

Designing a *User Interface* (UI) and a *User Experience* (UX) for Mixed Reality Headset deserves a separate discussion. Designing with Hololens in mind creates some limitations: using overly bright colors is exhausting for the eyes of the user, while dark colors are next to invisible; the familiar navigation methods used in all modern devices such as clicks and scrolls needs to be minimized in their usage - they are tiring since the user needs to keep their hand stretched in front of the headset for it to register the movement; and the interface elements needs to be sufficiently big to be easy to read and understand. Having said that, there are also the opportunities that the team was thrilled to explore: operating in Mixed Reality gives us whole two new dimensions to use in the design - the 3D missing from all common devices, plus the actual spatial reality around us; the speech recognition and gaze tracking add new ways to interact with the interface elements; and support for custom gestures can add a whole new layer of control.

An additional point of discussion is the process of designing a UI Prototype - the focus here is not to create the final design, but rather the framework that the team can build upon. The prototype illustrates the interface elements needed in the solution, their approximate positions, and the way the team expects the user to interact with them. The actual colors, positions, 3D elements etc. will be added as the understanding of the solution the team wants to build and the tools used, evolves.

Features outlined in MVP 1.0 are the ones the team also decided to include in the first prototype. Below are three application windows represented. Next few paragraphs discuss all three of them in brief and explain some of the important interface elements. The interface parts that we do not discuss have been added considering the future iterations of the product. Based on the previous experiences of the team members the decision has been made to use an interface design tool called Figma [78].

Start Menu, Fig. 33, is the launch menu the user sees once the application is booted up. On top left next to the microphone icon the application displays the tips for the voice commands that the user can use. Inspector can also use the navigation control menu (buttons in the middle to the left) and using "click" gesture navigate through the application. The last worked part is displayed in the center, along with the relevant information.

Figure 33: User interface prototype - Start menu

Select a Part Menu, Fig. 34, is, as the name hints, the menu from which the inspector gets to select a part to work on. The list in the bottom is a scrollable menu, which the user can navigate through by using the arrows on the left and right. When the user looks at the elements in the menu, that active mode in the middle is swapped, and the information about it is updated.

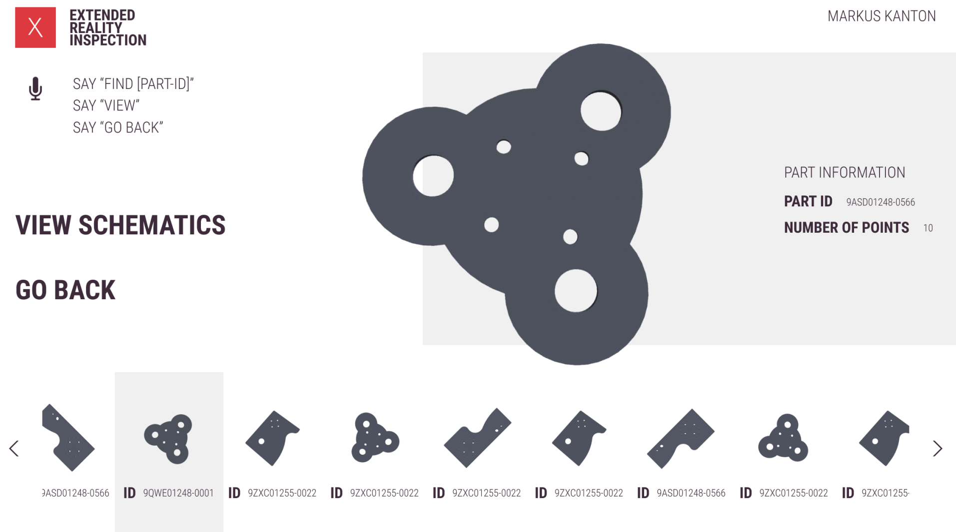


Figure 34: User interface prototype - Select a part menu

View a Part Menu, Fig. 35, is where the inspector gets more information about the part, can view the Schematic of the part and the information about points location. "Start Measuring" button will lead to a new window where the inspector gets to use the micrometre to interact and measure the points.

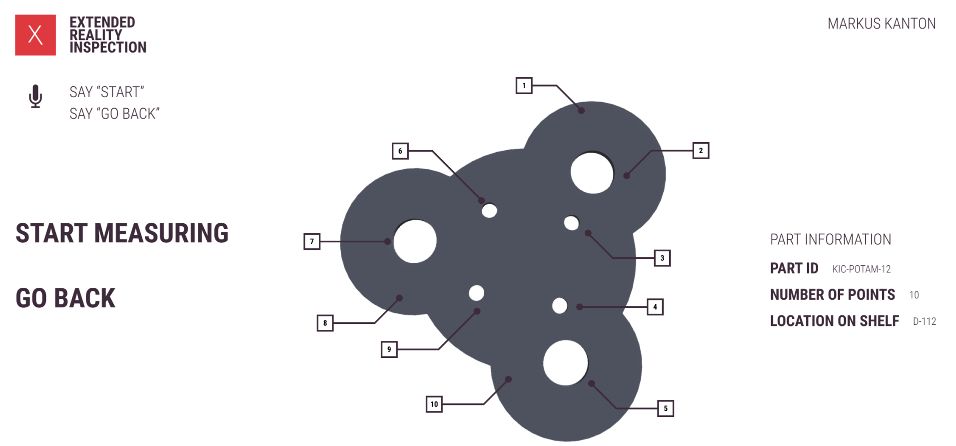


Figure 35: User interface prototype - View a part menu

6 Implementation

As mentioned in Sec. 5.2.2, the development was split into three product release versions. Sec. 6 consists of three main sub-sections, one for each product release, and it focuses on how features of those release versions were implemented. Every sub-section presents an overview of features planned for that version release and is followed by description of the work done by the team to implement them. Some of the features illustrated in the release overview are a part of the same functionality, or intervene with each other, and are therefore grouped together in sections to follow. The section is concluded with description of fixture prototypes development to accommodate the XRI application.

6.1 Product Release Version 1.0

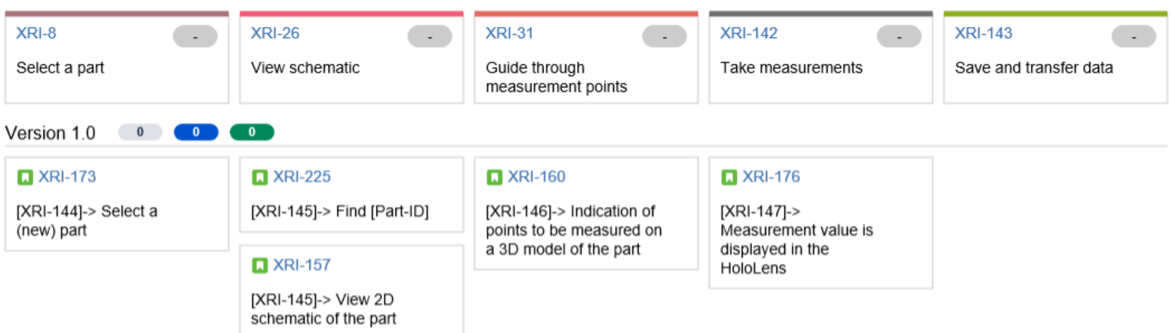


Figure 36: Product release version 1.0 features

Fig. 36 shows the product map for the first product release and the features that were planned to be implemented. The main goal here was to paint with broad strokes - create a user interface design and establish a foundation for it in Unity, get to understand how to share and communicate our work, and get to know our tools better. The team agreed that underestimating the importance of a good groundwork might cost us greatly later on, so Product Version 1 was designed to be the team's "soil probing" time. And yet, the team was a bit too ambitious still, in hoping to implement features XRI-160 and XRI-176 in time for the first deadline. These were postponed and moved to the second release (see Sec. 6.2). It did not, however, prevent the team from achieving the main objective of the first product release.

6.1.1 Menu Navigation and Voice Commands

Both functionalities discussed in this sub chapter were chosen to be the first ones to be implemented. The Menu navigation would allow the team to better understand both the hierarchy and inheritance systems in Unity, while voice commands would lay all the groundwork for other features using it in the future releases, as well as give possibility for team members without HoloLens 4.3 device interact with the software. Acceptance criterias for the features (see Fig. 37 and 38) show another good reason to discuss these functionalities together, since quite a few of them are linked together.

XRI-157: View 2D schematic of the part
Test Case / Acceptance criteria:

GIVEN that the user is in "Select a part" menu
AND the user has chosen the part

WHEN the user selects "View Schematic" with "click" gesture
OR says "Veiw Schematic"

THEN the system switches to "View a part" menu
AND a 2D schematic of the part and its info is presented

Figure 37: Acceptance criteria of feature XRI-157

XRI-173: Select a (new) part
Test Case / Acceptance criteria:

GIVEN the user is the "Start" menu

WHEN the inspector uses click-gesture or says "Select a part"

THEN the system switches to "Select a part" menu

XRI-225: Find [Part-ID]
Test Case / Acceptance criteria:

GIVEN that the user is in "Select a part" menu

WHEN the user selects a part from the scroll-list by looking at its elements

THEN a 2D preview model of the part is presented
AND the user can select "View schematic" of that part

Figure 38: Acceptance criteria of Selecting a part feature

6.1.1.1 Menu Navigation

In a nutshell, the menu navigation system has been made with two things in mind: a well-rounded, self-encapsulated module that allows a fast way of introducing new menu elements to the system, and ability to use it in different parts of software with ease.

Its usage boils down to simply adding new transition rules, and defining new parent menu elements. Menu navigation is used both in voice commands and when the user interacts with the menu elements using gestures.

6.1.1.2 Subtitle 2 Voice commands

Voice commands are an integral part of any Extended Reality system - as discussed in Sec. 5.3 HoloLens presents new possibilities in terms of user interaction with the software, and ability to maneuver around the application without raising hands is indeed an attractive idea. Menu navigation, finding the correct part, choosing a point, saving the work and other features of the program can be done using voice commands. The implementation in itself is rather simple, thanks to the public library provided by Microsoft and Unity and, in brief, are a collection of words that the user can use, as well as the actions that those commands perform. Given the fact that speech recognition can be a bit inconsistent in its performance given the variety of human dialects, accents and pronunciations , the team used short, easy to pronounce, and self-descriptive words in the application. [79]

6.2 Product Release Version 2.0

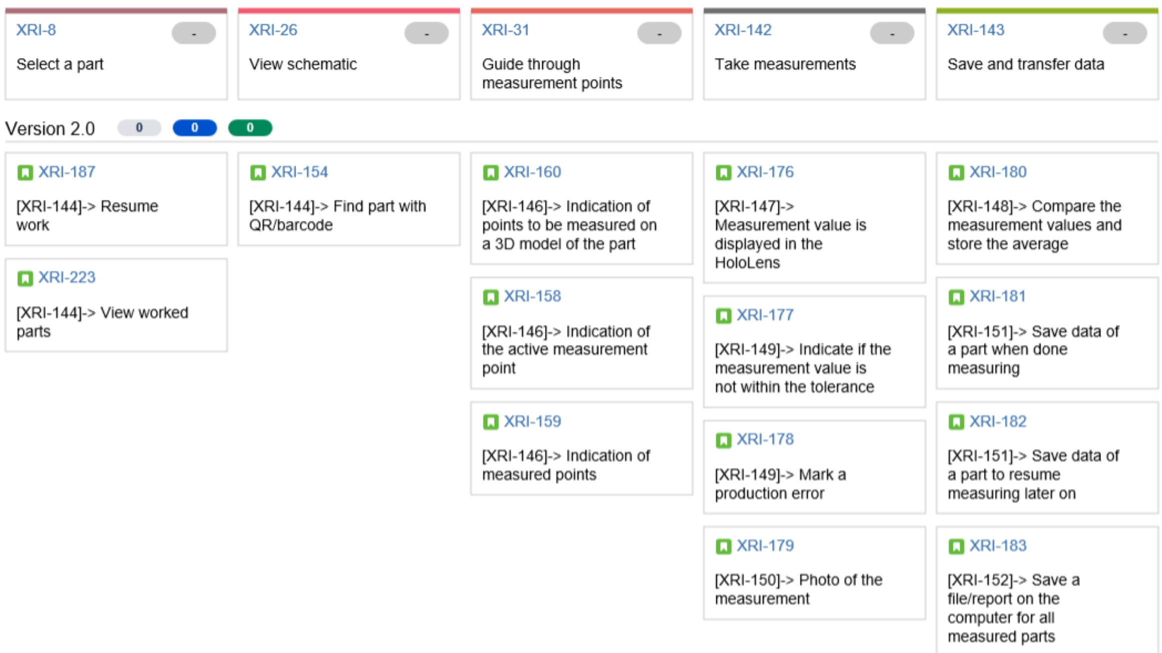


Figure 39: Product release version 2.0 features

Fig. 36 shows the product map for the second product release. This version was planned to be the most feature-rich of the three and the main objective was to have a minimum viable product that satisfies the requirements (see Sec. 2.2.1).

The basic menu application developed in the first version was intended to be advanced by selecting parts using barcodes (see Sec. 6.2.1) and extended with the possibility of interacting with holograms of parts and be guided through measurements points on digital models (see Sec. 6.2.2).

The second version had also an objective of setting up a data management system for saving and retrieving measurement data as well as handling and documenting the results received from the micrometer (see Sec. 6.2.3).

Establishing connection between the micrometer and the HoloLens was not an easy task which success depended on several external factors. Due to obstacles described in section 6.3.1 as well as the COVID quarantine (see Sec.4.3.1) the XRI-176 feature was, once again, moved to the next release.

Some of the nice-to-have but not essential features were stripped away altogether due to time limitations and challenges created by the COVID situation. XRi-167 *Resume work* and XRI-223 *View worked parts* were among those features.

6.2.1 QR/Barcode scanning feature

Every part that is inspected has a barcode that can be used for identification. That is why selecting a part by scanning a QR/Barcode is one of the features (XRI-154) of the second product release version, that was researched and evaluated by the team. Selecting a part by looking at a barcode also makes the application more user friendly than selecting it from a scroll-menu or saying long ID-numbers to select it with a voice command (see Sec. 6.1.1).

The envisioned functionality is described by the acceptance criteria in Fig. 40.

XRI-154: Find part with QR/barcode
Test Case / Acceptance criteria:

GIVEN that the user is in the "Select a part" menu
AND the QR/barcode is within the field of view

WHEN the inspector says "Find by code"

THEN the system displays a digital schematic of the part QR/barcode is linked to

Figure 40: Acceptance criteria for the QR/Barcode scanning feature

QR/Barcode recognition is not something that is supported by HoloLens natively, at least not by the 1st generation. Microsoft offers a Software Development Kit for QR code detection and tracking on Windows Mixed Reality devices but it is supported only by the Hololens 2, as mentioned in Sec. 3.3.3 *Image Recognition and Tracking Software*.

There is a QR/barcode library that is made specifically for decoding and generation of barcodes within images called ZXing. It was originally implemented in Java and later ported to support different platforms, including UWP and should, in theory, be possible to implement with HoloLens. [80] The main challenge with this is that documentation is near nonexistent and the library is rather old. There are some references of successful implementation of QR scanning with HoloLens using ZXing library with Unity, most of them are online forums, and are considered to be unreliable sources. However, there is one blog that is referenced to in almost all forums regarding the subject and is worth mentioning.

Mike Taulty works in Microsoft's Commercial Software Engineering group that builds Mixed Reality/HoloLens applications on UWP, and has been working with development languages, tools, frameworks and platforms for Microsoft for twenty years and in the industry since 1991. [81] In Desember 2016 Mr. Taulty wrote a blogpost *Windows 10, UWP, QR Code Scanning with ZXing and HoloLens* [82] that received a lot of attention on forums like *Windows Mixed Reality Developer Forum* [83] and *Stack Overflow* [84]. It seems that many developers were able to use ZXIng with HoloLens by following Taulty's guidance at the time it was written and until around year 2017.

The challenge with Taulty's implementation today is that is outdated and there are dependencies that prevent porting the solution directly to a newer version of Unity. Taulty's project was build in Unity 5.5, a version of Unity released in 2016 [85], and uses HoloToolKit, an older version of MRTK (see Sec. 3.3.1) from 2016. A lot of things have changed since then and in addition to Unity and tool kit dependencies there are building tools dependencies in Visual Studio as well.

Even though Taulty's solution seems as a good starting point, implementing QR scanning with HoloLens using ZXing today requires more expertise and human resources than our team could provide in the short amount of time given. Considering that the XRI project will need to be ported to HoloLens 2 which supports QR code detection with a SDK provided by Microsoft [21] it does not seem as a good use of time and resources either.

With all that being said, QR/Barcode scanning is a process of image detection/recognition. Since this project requires image detection and tracking to achieve holographic guidance through measuring points on a part/panel in real world, it should be possible to use image processing functionality of the same software (see Sec. 3.3) to recognize QR/barcodes if it does not contradicts it's licensing.

6.2.1.1 ARToolKit markers

ARToolKit was used to demonstrate the concept of barcode recognition with HoloLens. There are two marker images designed to work with ARToolKit for HoloLens by default: *Hiro* and *Kanji*. The markers can be found in the GitHub repository of the HoloLensARToolKit [24].

The toolkit can recognize two types of markers: **pattern markers** like Hiro and Kanji and **barcode/matrix code markers**, described by an ID.

The pattern markers can be customized and there are some trivial guidelines/rules for developing new ones, found in the ARToolkit documentation or as result of recognition performance (see Sec. 3.3.3.1):

- The marker should be asymmetric for correct marker orientation.
- The marker should have a border for correct detection and not to blend in with the surroundings.
- The marker should not be too complex (a bit ambigious, as no definition on what is considered complex has been given).
- The marker can be colored since ARToolKit translates it to grayscale internally but a high contrast marker (black and white) is preferable since it is considered less complex.

After designing a marker it can be used with ARToolKit by translating it to a binary file (for example by using a program called **mk_patt** [86]) and then indicating the filename of the new pattern marker description in the `ARUWPMarker.cs` script of the toolkit.

Barcode/matrix code markers recognized by ARToolKit can be generated using several online tools. Marker Generator by Eden Networks Ltd [87] is one of the tools that can be used and provides many options for generation matrix codes with up to 4194303 possibilities (with 5x5 barcode dimension). 3x3 barcode dimension (without error checking and correction) gives 63 possibilities and is the best option since is trivially less complex.

Due to delivery deadline of the project this feature was tested as a stand alone concept and not integrated into the main application.

6.2.2 Indication of Measurement Points on a Digital Model

The measurement guidance system is arguably the most important part of the system - this is where most of the work is done by the operators. It first fetches information about the parts and measurement points (as discussed in the previous sections), instantiates those elements as holograms, and displays relevant information.

In case as such, the system has to deal with two separate types of objects - the part under inspection and the points to be measured. Fortuitously, this provided the group with a natural way of encapsulating both the information and functionality, just as it was anticipated in UML design 5.1. The part and point instantiation ensures that all relevant information is available in the MonoBehaviour classes [88] and user can use it by interacting with the objects. Thus, for example, upon selecting a point, the information regarding it is displayed and the system is ready to receive commands regarding the point itself, such as taking measurement or moving to the next point. These directly comply with Acceptance criterias shown in Fig. 41 and have universally been agreed within the team to be a user friendly solution.

XRI-160: Indication of points to be measured on a 3D model of the part
Test Case / Acceptance criteria:

GIVEN that the user is in "View a part" menu

WHEN the inspector says "Start"
OR selects "Start Measuring" with a "click" gesture

THEN the system displays a 3D model of the part with marked measurement-points and tolerance info for every point

XRI-158: Indication of the active measurement point
Test Case / Acceptance criteria:

GIVEN that the 3D model of a part is displayed
AND the points to be measured are indicated

WHEN the inspector selects a point

THEN the system marks the point as active

XRI-159: Indication of measured points
Test Case / Acceptance criteria:

GIVEN that the inspector is done measuring a point
AND has registered measurement data

WHEN the inspector says/selects "done"/"next"

THEN the system marks the point as measured
AND indicates points that are measured and are to be measured

Figure 41: Acceptance criteria for Indication of measurement points on a digital model features

Lastly, depending on the measurement value stored in a given point, the system signals the state of the point, namely, *not measured*, *measuring*, *within tolerance*, *outside tolerance* and communicates that with the operator by turning points to appropriate colors - black, orange, green, and red, respectively. This helps operator to note, for example, exactly how many points are left to measure and whether there are any production errors as seen in Acceptance criteria Fig. 42.

XRI-177: Indicate if the measurement value is not within the tolerance
Test Case / Acceptance criteria:

GIVEN that the measurement point is active

WHEN the measurement data is registered

THEN the system indicates if the measurement data is / is not within the tolerance
AND stores the intermediate data

Figure 42: Acceptance criteria for Indication of measurement value outside of tolerance

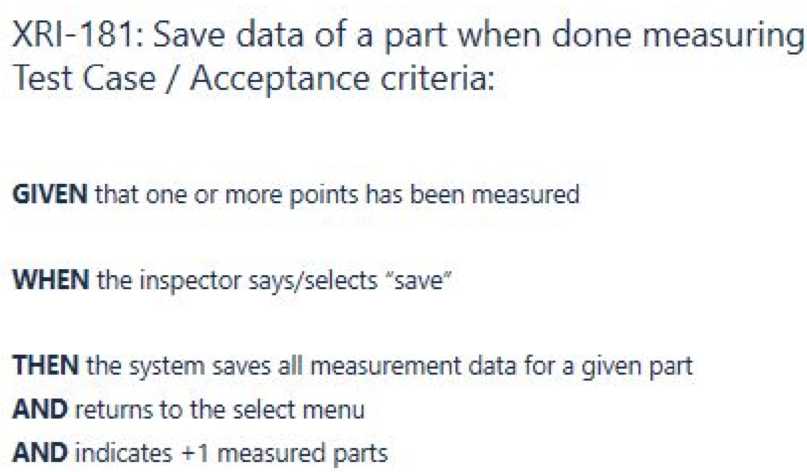
6.2.3 Documenting Measurements and Data Management

One of the requirements presented in Sec. 2.2.1 is to improve efficiency of the work process and the documentation of measurements. This implies that the system needs a reliable and accessible persistent storage of the data produced by the application. Persistent storage is storage of permanent data. [89].

6.2.3.1 Data storage

When the user starts the measurement process the application retrieves the required information from a file depending on the chosen panel. The information is stored in a plain text file and is not hard coded inside the application. The idea of this was to allow developers and users to update metrics like tolerance and position of points outside of the application and even add new panels without having to redeploy the entire application. As of now the positioning of points demands some trial and error and the user needs some knowledge of Unity engine to add new parts to the system - measuring parts only work for the composite panels that has been added and calibrated manually by the team.

In line with acceptance criteria XRI-181 (see Fig. 43) - when the measurements of a specific panel is saved, these measurements together with their respective point IDs is saved to a single *Comma-Separated Values*-file (CSV-file). CSV is a file type that separates fields of information with commas - the system use the common application as described in [90]. This file format was chosen because users can easily go into a document and view, or manually change different values without extensive knowledge of computers. The file type is also widely compatible with different text readers and can be opened in a popular spreadsheet reader, Microsoft Excel.


XRI-181: Save data of a part when done measuring
Test Case / Acceptance criteria:

GIVEN that one or more points has been measured

WHEN the inspector says/selects "save"

THEN the system saves all measurement data for a given part
AND returns to the select menu
AND indicates +1 measured parts

Figure 43: Acceptance criteria for saving the measurement data

From the users' perspective the term "directory" is a location for storing files [91]. Every time the micrometer register a measurement, the HoloLens takes a picture to document where the measurement were taken. From the users' perspective the term "directory" is a location for storing files [91]. The file with measurements and a directory of the belonging pictures is stored unaccompanied in a directory, name of which is constructed using the panels ID, along with date and time of the storing. Looking at the directories - one level up from the directory with a measurement file, there is a directory listing all the directories with stored measurements that has been done on panels with this distinct Panel ID. One level up from this - all the panels ID's which has measurements stored to them are listed, as seen in Fig. 44. This allows the user to locate and access previously stored sessions of measurements quickly, even when there are hundreds of them. The directory structure is created as panels are measured and there are no empty, redundant directories. To extend the system any number of new panels can be added without changes to the file system or how measurements are stored.

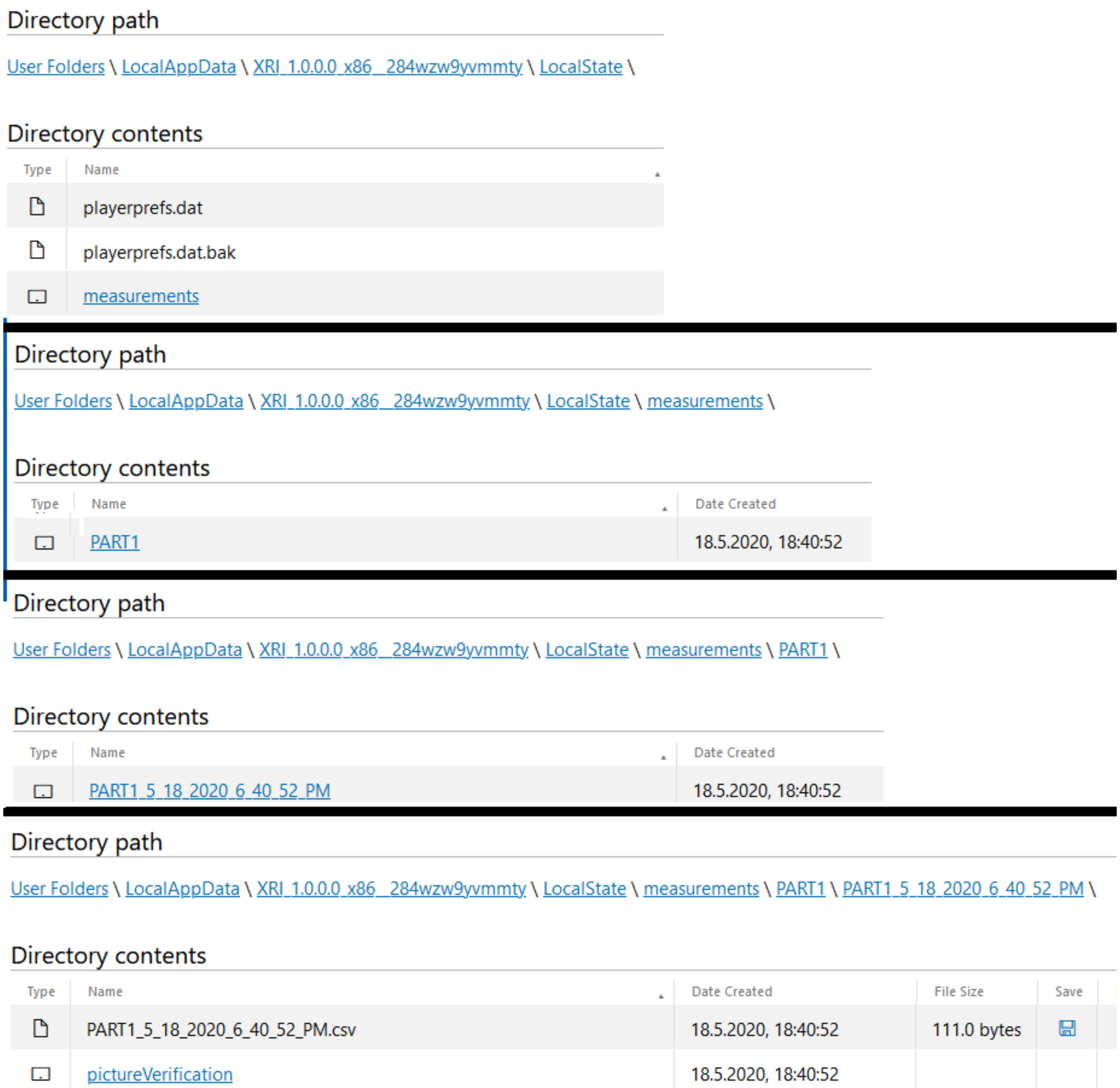


Figure 44: Snapshot of the directories structure

When a measurement is sent from the micrometer to the XRI-application its value is evaluated on the application side. The measurement value is controlled to be inside the expected tolerance of that specific point (see Sec. 2.2.1). If the measurement is outside the tolerance - the system alerts the user in accordance to the acceptance criteria as seen in Fig. 45.

XRI-178: Mark a production error
Test Case / Acceptance criteria:

GIVEN that the measurement point is active
AND the measurement value from the micrometre has been registered

WHEN the registered data is indicated not within the tolerance

THEN the system alerts the inspector
AND gives him an option of marking the part as faulty or redo the measurement

Figure 45: Acceptance criteria for marking a production error

The average value of the measurements taken for each point are stored as described in acceptance criteria in Fig. 46. At the same time, photo verifications are stored for every single measurement in accordance with the acceptance criteria in Fig. 49. Photos are named by its point ID and the time it was taken. This data is not moved into the persistent storage before the user uses the *save* instruction.

XRI-180: Compare the measurement values and store the average
Test Case / Acceptance criteria:

GIVEN that the measurement point is active
AND at least one measurement is registered

WHEN the the inspector says/selects "save"

THEN the system compares the registered values
AND saves the value closest to average
AND indicates the point as measured

Figure 46: Acceptance criteria for comparing measurement values and storing the average

For the user to retrieve measurement data from persistent storage, the user must use *Windows Device Portal* (WDP) - this application allows the user to access HoloLens files over network connection or USB [92]. Inside the applications installation directory there is a measurements directory which holds all of the measurement's data.

6.2.3.2 Implementation of storage and data management

In the systems context which is to verify where a measurement is taken - a minor loss of image quality is acceptable in exchange for the reduced size. The images are stored as JPEG-files. JPEG was chosen since it provides compression of the images at a ratio of up to 10:1, uncompressed over compressed - with minor loss in perceptible image quality [93].

The code in `fileHandler` provides two functionalities - storing measurements and retrieving a part. Store measurements takes a list of points and a panel ID and stores these to the appropriate location together with the belonging pictures - creating all necessary directories if needed. `retrievePart()`-functionality takes a panel's ID and parses the information from that panel's file and creates a list of points that holds all the data needed to perform a measurement on that panel.

There is a vast variety of options when handling files on Microsoft Windows operating system and the resources on the web proves to be ambiguous and often hard to get working on HoloLens because of file access restrictions and deprecated solutions. Plowing through these cost the team a lot of time. The group discovered two distinct ways to deal with files that work using Unity for HoloLens - Microsoft's own API called `Windows.Storage` [94] and `System.IO.File` - the latter one is preferred according to an official blog post [95] by Unity, and thus has been the team's choice as well. File access is restricted on HoloLens and there is no file explorer as on the common Windows Operating System used on the majority of PCs. Thus, all the directories and the CSV-files containing measurement data is created and stored within the installation directory.

6.3 Product Release Version 3.0

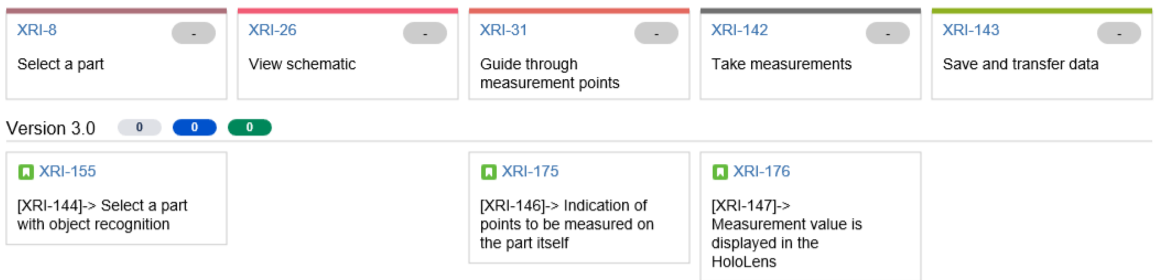


Figure 47: Product release version 3.0 features

In addition to refactoring existing features, the team worked on features shown in Fig. 47 during the third product release.

Selecting a part with object recognition (XRI-155) was a nice-to-have feature that was way too ambitious to implement during the project timeline. Given that it was not required, the team decide to drop it altogether and focus on more important matters.

XRI-176 feature, communication between the micrometer and the HoloLens turned out to be more time-consuming to implement then first expected. It was postponed all the way from the first to the last product release but was also one of the team's greatest achievements during the project (see Sec. 6.3.1)

XRI-175, precisely overlaying AR content on real world objects to achieve indication of measurement points and guidance on an actual physical part was a very demanding feature since it comprehends some of the most advanced technologies in the field. It a central part of this project and is described in last section, 6.3.2).

6.3.1 Communication Between Micrometer and the XRI application

The communication between the micrometer and the application installed in the HoloLens is mainly a one way communication after connection has been attained. The micrometer sends a measurement value to the HoloLens, which displays the value without returning any data (see acceptance criteria in Fig. 48). In addition to sending data, the same line of communication can be used to notify the HoloLens that a measurement is done which needs to be documented by recording an image (see acceptance criteria in Fig. 49). This means that data should arrive in a format that the HoloLens can recognize and with a latency low enough for the HoloLens to record a timely image for use as documentation.

XRI-176: Measurement value is displayed in the HoloLens
Test Case / Acceptance criteria:

GIVEN that the 3D model w/measurement points is displayed

WHEN the inspector pushes the micrometer button

THEN the system displays measurement value

Figure 48: Acceptance criteria for receiving measurement value from the micrometer

XRI-179: Photo of the measurement
Test Case / Acceptance criteria:

GIVEN that the measurement point is active

WHEN the measurement data from micrometre is registered

THEN the system takes a photo

Figure 49: Acceptance criteria for taking a photo of the measurement

6.3.1.1 Initial Solution

Since HoloLens 1 is unable to communicate with peripherals through its USB-C connection [48], and WiFi being prohibited at the site where the system will operate, the team viewed Bluetooth as a viable alternative. After deciding on using the client's already obtained Insize 3539-253A digital sheet metal micrometer the team first went for a solution with a self-produced Bluetooth-module built around the *Adafruit Feather nRF52 Bluefruit* (seen in Fig. 50), which would be able to communicate directly with the micrometer through serial communication channel using the micrometer's 2.5mm auxiliary connector, and then passing the measurement data on to the HoloLens through Bluetooth using HID-profile [96][48]. The most significant problem the team ran into was obtaining the protocol that was needed to communicate with the micrometer. In the end, this turned out to be too time consuming, which made the team turn to a solution using Insize Bluetooth-devices (they are advertised as Bluetooth-devices while actually communicating using Texas Instruments' CC2530 which is a Zigbee device, only similar to Bluetooth [97]).

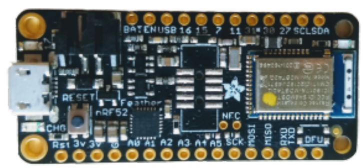


Figure 50: Adafruit Feather nRF52 Bluefruit

6.3.1.2 Final Solution

The Bluetooth-devices, a transmitter (Insize 7315-30) and a receiver (Insize 7315-2) that the clients already used in other projects, is not able to communicate with standard Bluetooth-devices such as the built-in Bluetooth-transceiver in HoloLens directly [48], which made it less favourable to include them in the final solution. After investigating the protocols with a logic analyzer and consulting with the manufacturer through e-mails during the attempted development of the first solution, the team discovered how to obtain measurement values through the use of the Insize-devices in two different ways.

One way is using a USB-host (an example seen in [98, Fig. 51]) connected to the Insize-receiver, which could read the data and pass it on to the Adafruit-device left from the initial solution, which again passes the data on to the HoloLens in the same way as in the initial solution.

Figure 51: Mini USB Host Shield 2.0

The advantage of this solution is that no soldering on the Insize-devices is required. The disadvantages of this solution is that it uses two Bluetooth-transmissions in serial connection which can cause latency to a degree that results in the HoloLens recording images of the measurement later then required.

The second alternative was to connect to the circuit-board where the wires connected to the micrometer is soldered to read data from the micrometer, which is only possible when the receiver is powered and able to hand-shake with the transmitter. From the connection, the Adafruit-device can read off values and pass it on to the HoloLens.

The advantage of this solution compared to the previously mentioned solution with a USB-host is that this solution only has one Bluetooth-transmission between the micrometer and the HoloLens. The disadvantage is that it requires soldering in the circuit-board of the Insize-transmitter which can cause damage to it. After consulting with the client about the advantages and disadvantages of the different solutions, the team chose to implement the second alternative.

The Bluetooth-module prototype is assembled on a prototype through-hole circuit-board with minimal emphasize on compact design and more emphasize on making the assembly as reversible as possible. As seen in Fig. 52, the prototype includes the Adafruit-device(1) that is powered by a 3.7V Li-Po-battery(2) and receives serial data from the micrometer on its RX-pin through the Insize-transmitter(3). The Insize-transmitter hand-shakes with the Insize-receiver(4) to enable communication with the micrometer, therefore the Insize-receiver is powered by the battery through the Adafruit-device. The board also includes a switch(5) that needs to be in 'off'-position during uploading of new scripts to the Adafruit-device, and a test-button(6) that can

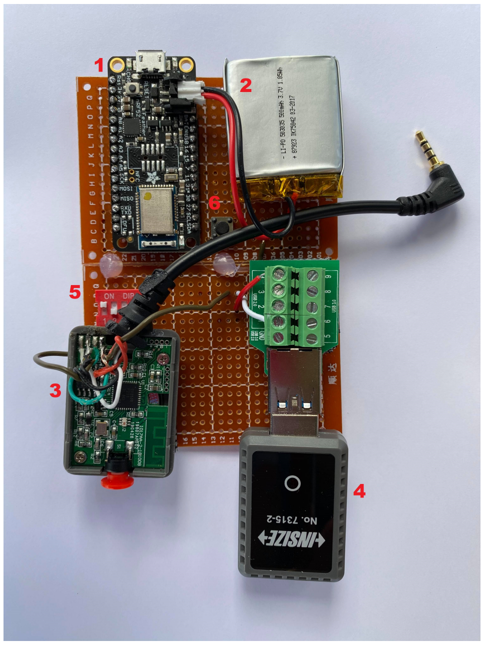


Figure 52: Front of prototype Bluetooth-module

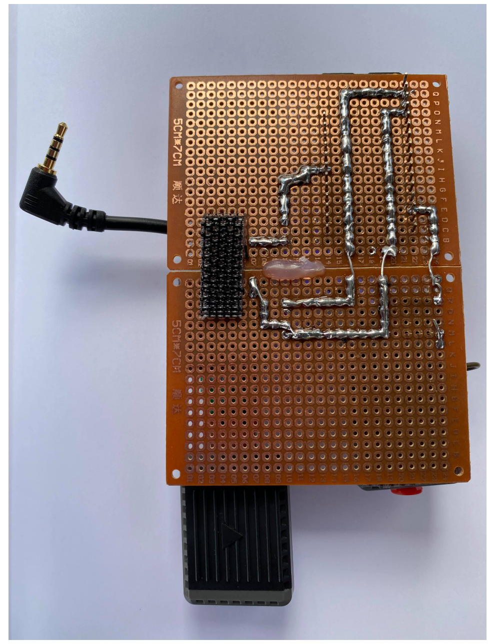


Figure 53: Backside of prototype Bluetooth-module

be used to send hard-coded data to the HoloLens for debugging purposes. The initial reason to add a test-button was to test the communication between the Adafruit-device and the HoloLens, parallel to soldering the connections in the Insize-transmitter circuit-board. The soldering and wiring between components is, as seen in Fig. 53, done on the backside of the circuit-board, where a piece of Velcro is glued on, so one can attach it to the micrometer for testing in operation.

Velcro is also used to attach the Insize Bluetooth-modules and the battery to the board for easier disassembly, as seen in Fig. 54.

According to [99] and analyzing of the output read through a logic analyzer the measurement value is transmitted from the micrometer to the Insize-transmitter using a UART-protocol with a baud rate of 2.4kB/s containing a start-bit, 7 data-bits and two stop-bits. The format of the protocol for measurement-value output in inches is described in Tab. 1. Where *CR* (0xD in HEX) and *LX* (0xA in HEX) represent `carriage return` and `newline` [100].

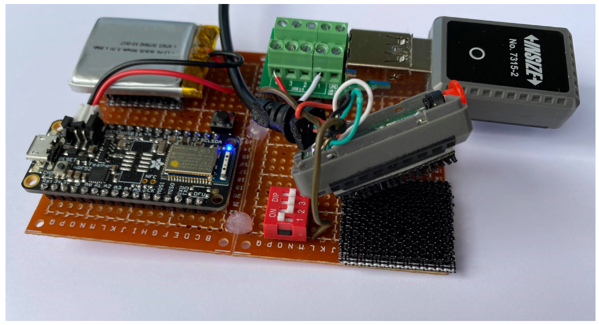


Figure 54: Components attached using Velcro

Table 1: Output data format micrometer for inches

Index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Value	Space	'0'-'9'	'.'	'0'-'9'	'0'-'9'	'0'-'9'	'0'-'9'	'0'-'9'	CR	LX

6.3.1.3 Software Implementation - Bluetooth-module

The Bluetooth-module software implementation was done through *Micrometer_communication.ino*. The code is an edited version of *hid_keyboard.ino* by GitHub-users *hathach* and *toddtreece* found at [101] which already had working code that connects the device with Bluetooth-hosts such as the HoloLens. When the Adafruit-device detects data from its RX-input, `Serial.Available()` returns `true` so the device can read data through `Serial.readBytes()`, which will assign to `data_string` the first 10 characters received. The 10 characters corresponds to the 10 bytes sent by the protocol mentioned above. The script uses a method called `keyPress()` of the class `BLEHidAdafruit` that sends a character passed as an argument [102] to send measurements to the HoloLens that is interpreted in the same way that it would interpret keyboard key-presses.

Every measurement value starts with a `whitespace`-character as a frame-header which it inherits directly from the micrometer protocol, as long as the number is positive. When the micrometer sends a negative value the first character in the character-array will be a minus-sign, which is not interpreted as the start of a value by the HoloLens. This is acceptable, if not wanted, since negative measurement values never occur in valid measurements. The last character in the array is the `endline`-character '\n' which functions as the frame-tail of the character-array. In addition to the frame-header and frame-tail, the characters representing the actual value are separated by 'x'-characters. The function of these separation-characters will be explained in the next paragraph. The script also contains an ISR-function, `sendTestMeasurementISR()`, that controls the test-button on the Bluetooth-module. When the function is triggered by a button-press, a test-value is sent to the HoloLens which adds a tool for troubleshooting problems with the micrometer-communication.

6.3.1.4 Software Implementation - HoloLens

The software implementation of the micrometer-communication for the HoloLens is implemented in *micrometer_measurement.cs*. The implementation is built around the `Event`-class from the `UnityEngine`-library which implements objects that store information about events such as user inputs (key presses, mouse actions) and rendering events. Every time such events occur, the method `OnGUI()` that is inherited through `MonoBehaviour` is called. To obtain the array of characters sent from the Bluetooth-module, an object of class `Event` is instantiated with the variable name 'e' and gets assigned the last registered event through the static property `Event.current`. Primarily the frame-header, frame-tail and characters from the measurement-value is relevant to register, but since the `OnGUI()`-method gets called for other events as well, `Event.current` regularly bring empty characters or repetitions of the previous character. Such as the array ' ', '0', '.', '0', '2', '9', '0', '1', '\n' , representing "0.02901", register as the sequence ' ', null, null, '0', '0', null, null, '.', '.', '0', ... , '\n', '\n' . To make sure the script records the right sequence, the characters sent from the Bluetooth-module gets separated by 'x'-characters (as stated above), so that the program waits for an 'x' after receiving a new character from the measurement. This makes the program ignore repeated characters. [103][104]

When the HoloLens receives the frame-header the program instantiates a component of type `ImageCapture` which makes the HoloLens record an image as soon as possible after a measurement is taken. When the frame-tail arrives, the measurement gets parsed from a `TextMesh.text`, that displays and stores the character-array from the Bluetooth-module, and passed through the method `takeMeasurement()` that store the value in a `Point`-object.

6.3.2 Overlaying AR content onto physical objects

Due to the delivery deadline of the project, and the extent and complexity of this feature, it was tested as a stand-alone concept and not integrated into the main application. The team considered it to be important to test and verify the technology before integrating it into the main solution.

Besides demonstrating the barcode recognition concept using HoloLens, the ARToolKit is actually a good starting point on learning more about placing AR content onto physical objects. There were two possible SDKs that could be used for this functionality, ARToolKit and Vuforia Model Targets (see Sec. 3.3.3), and since there was no time to bet on two horses the team decided to proceed with the `HoloLensARToolKit` for reasons listed in Sec. 3.3.3.

The idea was to place a barcode/marker for a given part onto a specific place on the table at the working station. The HoloLens will recognize it using ARToolKit and project an identical hologram of the part it identifies. The hologram can then be fixed in place by using Unity's WorldAnchor API [105] (application programming interface). One can then align the actual part with the hologram and secure the part using fixtures 6.4.

The "sub-project" started out as a sample scene of the `HoloLensARToolKit` (see Sec. 3.3.3.1) in Unity 2019.2.21f1. The team encountered a fair amount of debugging and installation of Visual Studio 2015 and 2017 building tools (v140) to solve dependencies issues and successfully build and deploy to the device using USB.

6.3.2.1 HoloLensARToolKit Configurations

After creating an empty 3D project in Unity and configuring project settings for MR development [106], the *HARToolKitUWP.unitypackage* from HoloLensARToolKit GitHub repository [24] was imported into it's *Assets*. New scene called *HL1ARToolKit_demo* was created and it's hierarchy consists of the following GameObjects (Fig. 55):

1. **Main Camera** has the setup recommended by Microsoft for HoloLens projects [106]. Besides the MR settings it has a child object *Preview Plane* that is used by the *ARUWP Video Script* linked in the *ARUWP Controller* object. It is a video preview holder that is anchored in the top left corner for the user's perspective.

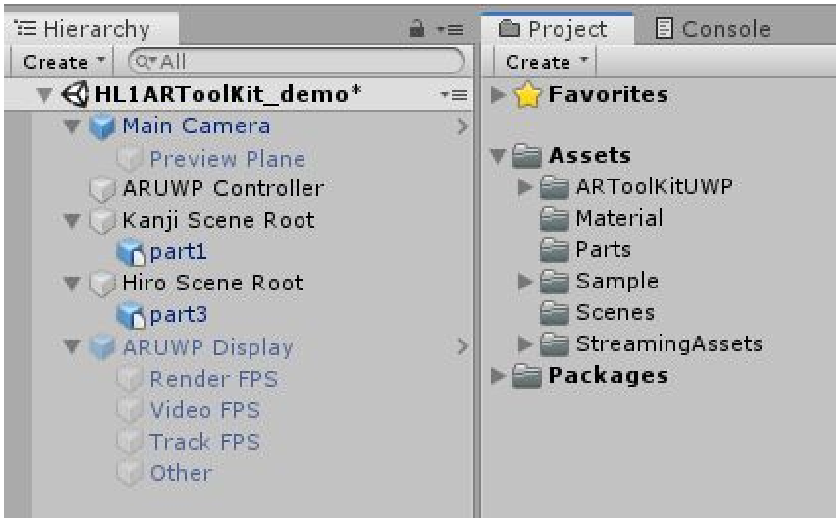


Figure 55: ARToolKit demo project: hierarchy and assets

2. **ARUWP Display** has several child objects that regard frame rate information which are useful for performance testing and has a *Canvas* element that is used for camera rendering. The *ARUWP Display* is a *prefab* from *ARToolKitUWP-Sample.unitypackage* [24] and can be found in *Sample* folder under *Assets*. It should be disabled in real applications since the video preview will naturally draw resources from the recognition and tracking.

3. **ARUWP Controller** is the most essential object that comprises important script elements:

   (a) **ARUWP Video (Script)** contains *ARWPVideo class* which uses Windows MediaCapture APIs [107] to handle video access and allows *ARUWP Controller (Script)* to control video initialization, start, stop and enabling preview. *Video Parameter* matching the device should be selected here. **1344 x 756** resolution profile with maximum supported frame rate (30 fps) was chosen based on Microsoft's Mixed Reality Documentation on Platform capabilities and APIs [108].

   (b) **ARUWP Controller (Script)** initializes camera parameters. The camera calibration file mentioned in Sec. 3.3.4 is linked here. The ARToolKit comes with default camera calibration files for for HoloLens 1 and 2 which can be found in *StreamingAssets* under *Assets*. However, it is recommended to generate a calibration file that is specific for the device.

Pattern Detection Mode in the *ARUWP Controller (Script)* specifies what kind of marker the detection algorithm should look for. There are three different modes to chose from:

- AR_TEMPLATE_MATCHING_MONO works for pattern markers.
- AR_MATRIX_CODE_DETECTION works for code markers.
- AR_TEMPLATE_MATCHING_MONO_AND_MATRIX should be selected if there is a need for detecting both.

For more information about the types of markers and how they can be generated please see Sec. 3.3.3.1.

(c) **ARUWP Marker (Script)** component should be added for every unique marker. The *ARUWPMarker class* represents an ARToolKit marker which type can be configured with four options but only two of them (single markers) are relevant for this project:

- **single** for a single pattern marker. The marker file that is linked here have to be a pattern file (.patt) and stored in *StreamingAssets* folder in project's *Assets*.
- **single_barcode** for a single barcode marker specified by *Barcode ID*.

For more information about the types of markers and how they can be generated please see Sec. 3.3.3.1.
The ARToolKit algorithm provides marker position and orientation that is applied to the virtual object for rendering (also called Scene Root) by configuring *Visualization Target*. *ARUWPTarget class* (*ARUWP Target Script*) represents that object and is attached to the scene's root.

All scripts (a, b and c) can be found in *Assets* → *ARToolKitUWP* → *Scripts*

4. **Hiro Scene Root** and **Kanji Scene Root** act as sub-scenes and the actual 3D models of the parts are attached to them. Transformation of the root object is automatically updated by the algorithm so the "initial" rotation and scale of the 3D model of the part is configured in the child object, not the parent/root object. The 3D models are exact replicas of the physical parts and are scaled to 1 in X, Y and Z directions. The 3D objects of the parts are modeled in SolidWorks and converted from CAD to *.fbx* files.

6.3.2.2 Manual Camera-projection Calibration

Fig.56 demonstrates the offset the team experienced when testing the ARToolKit sample scene *HL1ARToolKitCoords*. The visualization of the virtual object had to be adjusted and aligned better with the marker and the following steps should be conducted after the camera calibrations steps described in section 3.3.4.

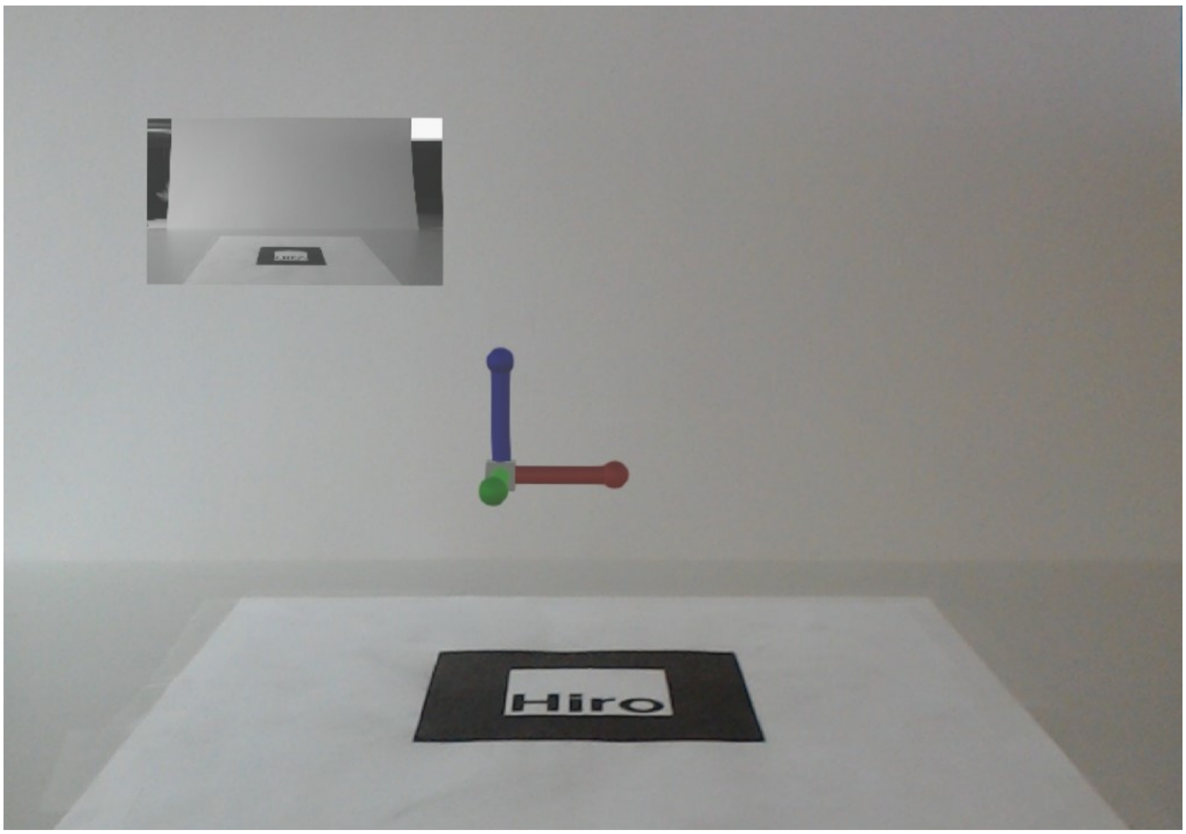


Figure 56: H1ARToolKitCoords sample scene - virtual object offset

`calibrationMatrix` in the *ARUWPMarker.cs* script is used to offset the tracking results. It is initialized with `Matrix4x4.identity`[109]. The identity matrix effectively does nothing when applied, meaning no offset is applied. The last column is doing translation only (no rotation or other transformation). The first coordinate of the 4 coordinates of the last column is the X translation, the second is Y translation, Z for the third one and the fourth component is always a 1 (the homogeneous coordinate) (see Sec. 3.3.4).

The goal was to align the axis object to the dead center of the marker visually in 3 dimensions. To manually update the `calibrationMatrix`, `ManualUpdateCalibrationMatrix()` method was defined in the *ARUWPMarker.cs* and called in its `Start()` method.

The following calibration procedure was quite tedious and time demanding. The manually coded calibration matrix is very sensitive and the offset had to be corrected by tiny increments of only one coordinate at a time with 3 digit precision. After every change of coordinate the project had to be build and deployed to the headset to verify the correction.

The author of the HoloLensARToolKit, Long Qian, together with three other professors at Johns Hopkins University published a paper called *Alignment of the Virtual Scene to the Tracking Space of a Mixed Reality Head-Mounted Display* [29]. This paper is about this exact misalignment problem and how they tried to fix it. The best way to perfect alignment (L. Qian is claiming 4mm accuracy) is to do the exact steps written in the paper. Unfortunately, they did not share the files to do the *Multipoint Single 3D Object Method* calibration described in the paper. The team contacted Long Qian regarding this matter but he did not wish to share the code used for the paper. It is something that is possible to recreate though, but not in the limited time the team had left at the writing point.

6.4 Fixture Prototypes

This section contains the most important information for the fixture modelling and manufacturing, from idea to reality. The actual process is quite copious and it has been a very iterative process with a lot of back-and-forth communication with the client and operators to arrive to a final prototype that is presentable. As there has been a lot of modelling in SolidWorks all models and assemblies are attached with the digital thesis submission.

The following list shows the most important tools used for producing the different fixture prototypes:

- Solid Works 2019
- Ultimaker 3D printers
- Ultimaker Cura 4.5
- FlashPrint
- FlashForge 3D printers

6.4.1 Prototype Development

In the prototype development phase of the project the group had high emphasis on dialogue with the client to create a fixture solution that would serve the needs of the operators when using the system. It was decided to conduct an interview with the client to understand what they saw as the key requirements for the fixture system. This was done via e-mail and phone communication due to the limitations imposed by the COVID-19 pandemic (see Sec. 4.3). The following list covers the fixture systems key customer requirements:

- The fixture must be able to fix multiple panels with different sizes (ranging from 0,5x0,5m to 3,0x1,0m) and curvatures.
- The fixture must be user friendly for the operator to work with.
- The fixture must be steady when placing and working with the panels.
- The fixture must have minimal possibility to damage the panels while mounting and dismounting.

From the start of the project, what kind of fixture the client needed was a very open task, but the group soon realized the importance of keeping the part still for the HoloLens system to work (explained in more detail in magnet chapter below). Therefore, developing a fixture prototype has been a very iterative process where input from the client and the operators have been the biggest factors for redesigns, implementations and even restarting completely on new fixture designs.

Due to the COVID-19 pandemic restrictions starting in February the group had to shift the focus from the production of the prototypes, to securing access to all the required machines, tools, mechanical components, etc., to ensure a continuous workflow of modelling-and-creating parts for the fixture prototypes (see Sec. 4.3). As of 19.03.2020 the group was able to get a hold of a 3D printer, the produced parts from campus, some M5 bolts and nuts, and a caliper. By week 17 the group also got a hold of additional mechanical components, such as screws, nuts, calipers of different dimensions, tools, etc..

The client also issued some concerns that the team had to focus on while working on the fixtures prototype. A summary of this is shown here:

"The simplest parts we produce might not need a fixture at all. Alternatively smaller soft blocks the part can lay steady on. As of now all parts are placed on plastic covered tables and soft "blocks" are used to elevate some panels where it's needed to be able to use the caliper." (see Appendix D)

For the subchapters regarding the *Ball Point Fixtures* the group decided to only include the production process of version 1 and 6, as the changes done from versions 2 to 5 are small changes that were done iteratively within the group, as well as with input from the client, and thus the group deemed that those versions were redundant to be included in the report. Nevertheless, the models for all versions' parts and assemblies are included in the digital thesis submission.

6.4.1.1 Carbon Fiber Panels

To test the different fixtures it was necessary to produce smaller carbon fiber parts that are somewhat similar to the ones at KDA. The reason we wanted to use similar parts as for KDA is to understand the parts' material properties and how they correspond to different actions, as for example magnetic force. This instantly creates a realistic system compared to if we were to use cardboard models and not entirely understand

the product we are making the system for.
The first batch consisted of three different parts (as seen in Fig. 57) which all had one identical set of holes. The production of this has been documented in Appendix G and as a video which can be seen at www.xri.no/production.

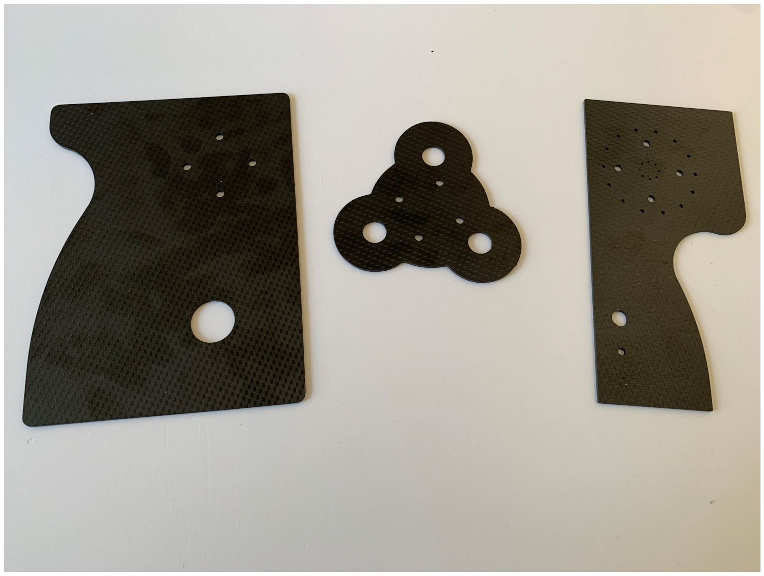


Figure 57: Produced carbon parts

There is a lot of ways to produce carbon fiber panels, for example with wet lay-up, resin infusion, preimpregnated fiber sheets, etc.. The carbon fiber panels used in the project are made from preimpregnated carbon fiber sheets. Which means that the fiber sheets are filled with a optimal amount of epoxy resin from the manufacturer creating a product with high strength, toughness and structural performance at low weight. [110] This production process is the same as the one used at KDA, which makes the material differences from the different carbon fiber sheets the only variation.

6.4.1.2 3D Printing Parts

During the project period the group had access to a FlashForge Finder 3D printer, as seen in Fig. 58. [111] This printer mainly prints with Polyactic Acid-filament (PLA-filament) which is a bioplastic with a melting point between 190 to 220. [111] The group has also been able to get one of our personal contacts to let us print with Thermoplastic Polyurethane-filament (TPU-filament) as well, which is used for it's elastic properties. [111]

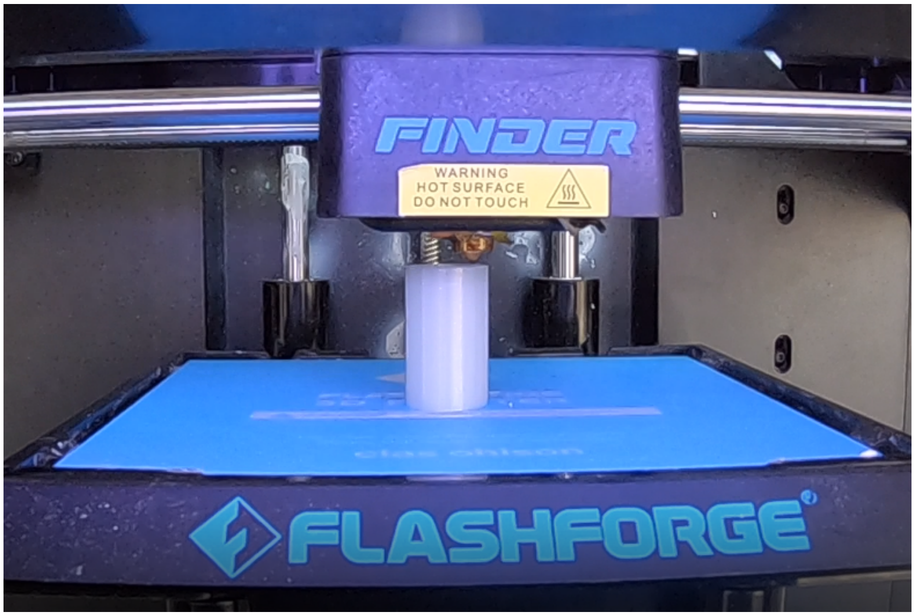


Figure 58: Manufacturing Part with FlashForge Finder Printer

Using a 3D printer was a simple way of creating and testing most of the parts going into the fixture prototypes, since it's a short way from conceptually designed parts and systems in SolidWorks to getting the part created in real life. Luckily, pricewise it is a very affordable solution as well.

Even though it is very easy to manufacture prototype parts using 3D printers there are a lot of different factors that all have a say in the final printed product's material properties. The most important factors the grouped tweaked for each part produced and will be discussed in the respective sub-chapters. These properties are described below:

- **Fill Density:** Determines the interior solidity of the model. This again determines the strength of the model, where low fill density makes it weaker and more prone to, among others, tensile and shear stresses or torsion. The density option varies between 0 to 100%, but we never used it below 10% as the printer is not able to print in mid air and this is the lowest point for this when printing solid parts. Using 100% creates a completely filled part.
- **Fill Pattern:** Here the group had four options, *Line*, *Triangle*, *Hexagon* or *3D infill* pattern. We were recommended by lecturers with experience with the

printers to use either the triangle and hexagon patterns as these are the most optimal when comparing strength to production time. The triangle pattern is quicker to produce but weaker than the hexagon pattern and is therefore more optimal when creating bigger parts for testing quickly.

- **Brim Enabled:** For some parts printed there was an issue where the parts didn't melt to the print plate. In some cases this was solved by enabling the *Brim* as this creates a larger surface area around the part with melted plastic. This is easily removable when the print is finished.
- **Pause Heights:** The printer can pause at specific heights, enabling the group to print nuts, knurled nuts, etc. into the parts where this was necessary.

6.4.1.3 Machining Parts

Manufacturing parts at the workshop in KDA was also a possibility at the later stages of the project. As they have milling and CNC machines, waterjet cutter, drilling machines, etc. which we didn't have available since the machining lab at USN was closed due to COVID-19 (see Sec. 4.3). Because of the strict regulations at KDA we were not able to get any pictures or videos from these processes.

6.4.2 Hole Mount Fixture

The very first solution was modelled to show out idea of fixing the parts in a specific position, using the holes that were machined into them, as seen in Fig. 59. This is similar to the parts at KDA since some parts have the same hole dimensions that could be used for this purpose.

The panels would be fixed as it would mount on four plugs with a slightly smaller diameter than the holes. On the upper side of the panel four "heads" would be placed upon the plugs, and screws would hold them together by mounting to the knurled nuts printed into the main "mount".

This model was shown as an example at the first presentation but was retired shortly after when discussing with the client and operators because it would be realistic only for a small amount of similar parts at KDA.

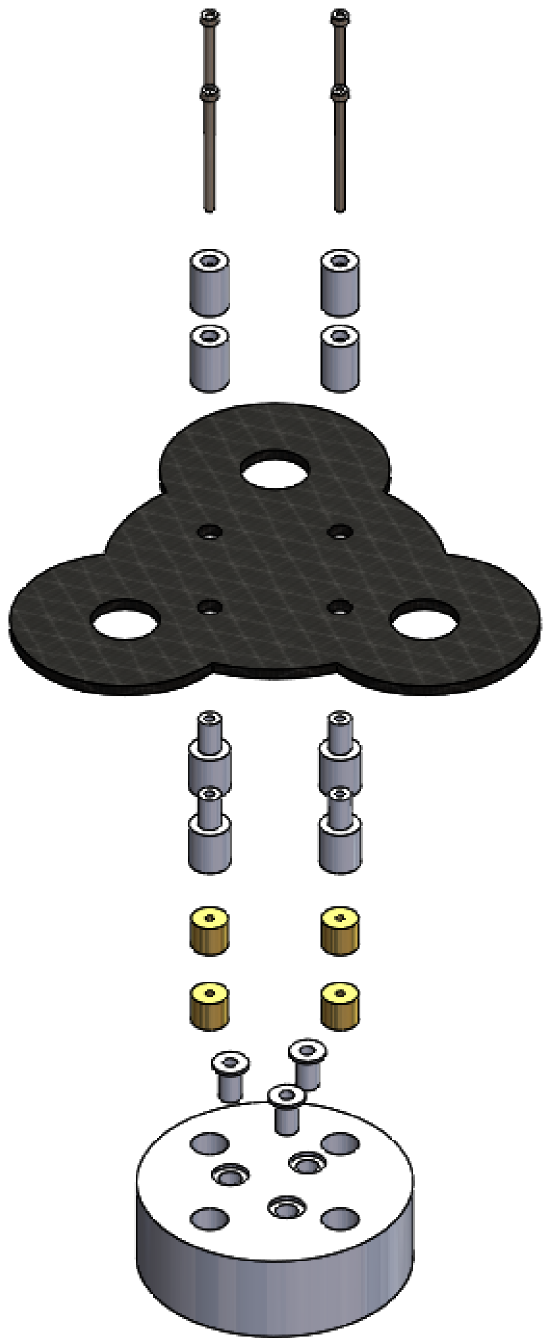


Figure 59: Exploded view - First Concept Model

6.4.3 Clamp Fixture

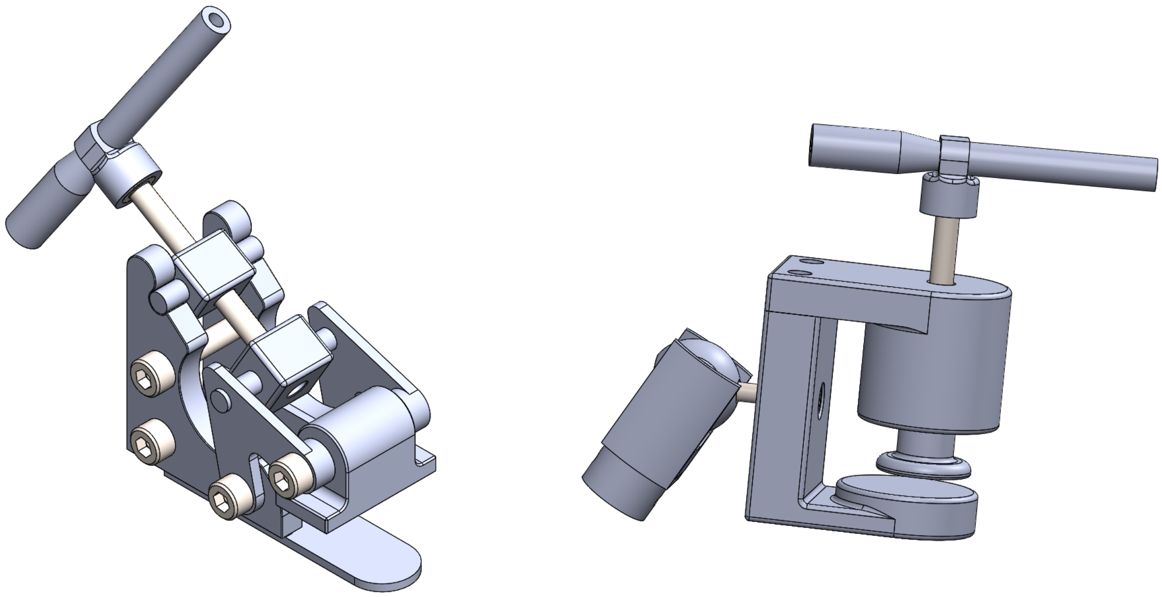


Figure 60: Clamp Fixture SolidWorks Models - V1: Left , V2: Right

The clamp fixtures seen in Fig. 60 was the team's second attempt. The idea was to use clamps to hold the composite panels in specific positions. The clamps would then be mounted on arms with a given number of joints which could vary the position of each clamp, making it usable for multiple parts. The second version clamp was also mounted on a ball joint, making it able to get to more complex positions than the first version.

The client and operators weren't convinced with this solution and we discussed other possibilities as explained further in the next chapter.

6.4.4 Ballpoint Fixture - Version 1

During a meeting midway in the project the client suggested to use simple ball points, that the operator would rest the panels on while working on them. From this point on, using ball points has been the only focus when designing fixtures. Therefore, the modelling has revolved around creating a prototype that gives the operator a flexible fixture that can be used for various parts.

In total, 6 versions of the Ballpoint Fixtures has been made - the first version was created to show the capabilities of the idea itself, and versions 2 to 6 (as shown in Fig. 61) were created with improvements to simplify the production and making a more complete final product.

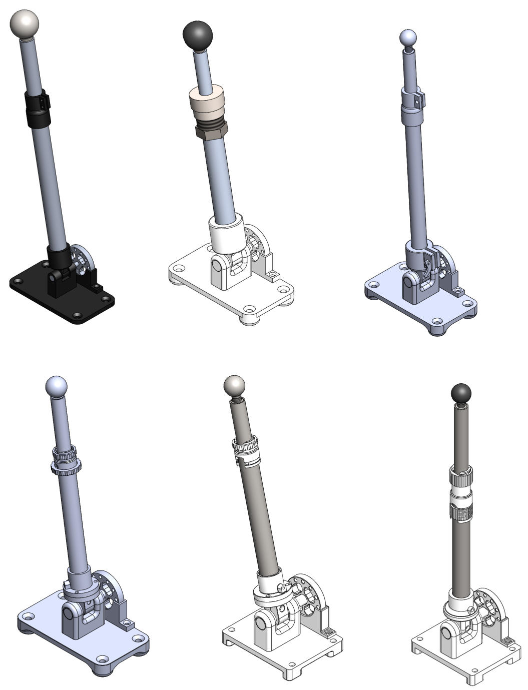


Figure 61: Fixture evolution - top left: V1 - bottom right: V6

The first version was fairly simple but as a consumer product it would be too complicated and expensive to produce in mass numbers. Regardless, this was created with the mindset of portraying the fixtures functionality for the client.

The system was made with a rotational joint that could be locked in certain orientations by rotating the wheel and entering a locking pin through the alternative holes near its perimeter.

The aluminium profiles was chosen for the sole purpose of creating a telescopic function where the outer aluminium profile is locked in the pole mount and the inner aluminium profile has a outer diameter smaller than the outer profiles inner diameter which lets it slide through it, and that, in return, gives the possibility of adjusting the length of the pole.

6.4.4.1 Rotation Joint - Version 1

The schematic for the *Rotation Joint* can be found in Appendix I.

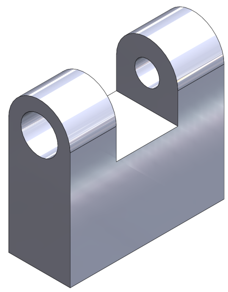


Figure 62: Prototype V1.0 - Rotation Joint

The rotation joint shown in Fig. 62 was made this way to simply be able to showcase the planned functionality of the systems' rotational capabilities. It lets the rotation pin enter and hold in the exact position so the rotation bracket can be placed in the center plane of the rotation joint, which gives the fixture symmetry. It also has two holes underneath with nuts printed into so that machine screws can mount it to the bottom plate.

6.4.4.2 Wheel - Version 1

The schematic for the *Wheel* can be found in Appendix I.

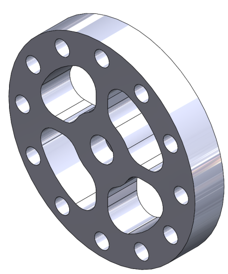


Figure 63: Fixture prototype V1.0 - Wheel

The wheel shown in Fig. 63 was created to be able to stop the rotation at set intervals by using a locking pin. The pin holds it in position by locking the movement with the wheel joint part described later in the sub chapter. The wheel is connected with the rotation pin using a M5 machine screw.

6.4.4.3 Wheel Joint - Version 1

The schematic for the *Wheel Joint* can be found in Appendix I.

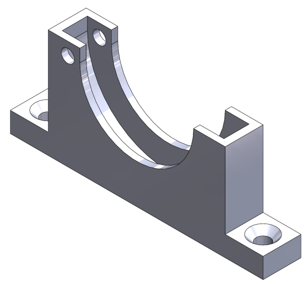


Figure 64: Fixture prototype V1.0 - Wheel Joint

The *Wheel Joint* shown in Fig. 64 was created to have a fixed placement for the wheel. The inside surface was created with circular "bumps" to give smaller surface area and less friction when rotating wheel.

6.4.4.4 Rotation Pin - Version 1

The schematic for the *Rotation Pin* can be found in Appendix I.

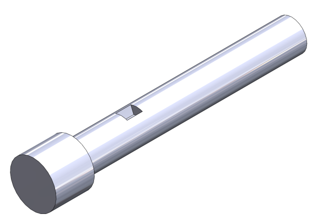


Figure 65: Fixture prototype V1.0 - Rotation Pin

The *Rotation Pin* shown in Fig. 65 was created to connect the *Rotation Joint*, the *Rotation Bracket* and the *Wheel*. It was created with a "slot" for the pole mount to be fixed in the center position. This solution was a very temporary solution as it was a very complex and difficult way of solving a relatively easy problem.

6.4.4.5 Rotation Bracket - Version 1

The schematic for the *Rotation Bracket* can be found in Appendix I.

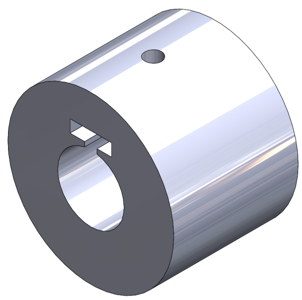
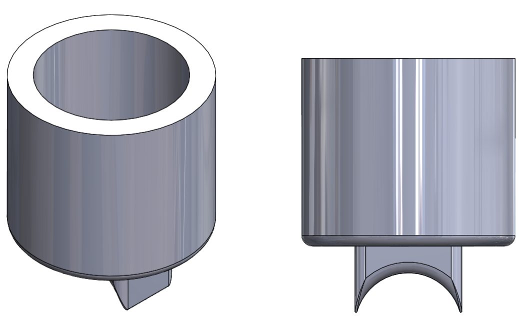


Figure 66: Fixture prototype V1.0 - Rotation Bracket

The *Rotation Bracket* shown in Fig. 66 was created to connect the *Rotation Pin* and the *Pole Mount*. The mounting solution was a complex and difficult alternative which evolved in later versions.

6.4.4.6 Pole Mount - Version 1

The schematic for the *Pole Mount* can be found in Appendix I.
The *Pole Mount* shown in Fig. 67 was created to connect the *Outer Aluminium Pole* to the system. The mount has a tight fit Ø30 dimension to hold the pole in place. The connection to the *Rotation Bracket* was a solution which would be complicated to manufacture, and therefore it needed to be changed in later revisions.



(a) (b)

Figure 67: Fixture prototype V1.0 - Pole Mount

6.4.4.7 Bottom Plate - Version 1

The bottom plate shown in Fig. 68 was created as an anchor for each individual fixture "mount". Ideally this would have legs to stand on rather than the the machine screws.

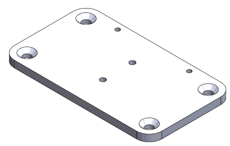


Figure 68: Fixture prototype V1.0 - Bottom Plate

6.4.4.8 Outer Aluminium Pole - Version 1

The schematic for the *Outer Aluminium Pole* can be found in Appendix I.

Figure 69: Fixture prototype V1.0 - Outer Aluminium Pole

The *Outer Aluminium Pole* aluminium pole shown in Fig. 69 was designed with a length of 300mm and has a tight fit to the *Pole Mount*, which it is also locked to using a set screw. It also has a thight fit the *Pole Clamp* which is tightened even harder when rotating the *Pole Clamp Ring - Large*.

6.4.4.9 Inner Aluminium Pole - Version 1

The schematic for the *Inner Aluminium Pole* can be found in Appendix I.

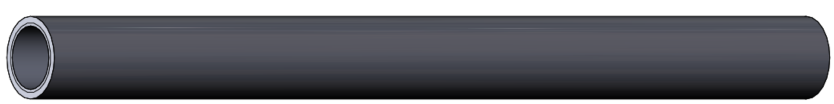


Figure 70: Fixture prototype V1.0 - Inner Aluminium Pole

The *Inner Aluminium Pole* shown in Fig. 70 was designed with a length of 300mm but had an outer diameter 1 mm smaller than the *Outer Aluminium Poles'* inner diameter, allowing it to pass through. It was also threaded in the upper end with M22 threads to mount the *Steel Ball Plug*.

6.4.4.10 Steel Ball Plug - V1.0

The schematic for the *Steel Ball Plug* can be found in Appendix I.

Figure 71: Fixture prototype V1.0 - Steel Ball Plug

The *Steel Ball Plug* shown in Fig. 71 was designed so the operator could easily place the composite parts on top of multiple of them. It has a M22 fine pitch thread which matches the *Inner Aluminium Pole* which enables the operator to adjust the height of the fixture more precise.

6.4.5 Ballpoint Fixture - Version 6

The modelling and testing resulted in the last and final version, version 6, as seen in Fig.72.

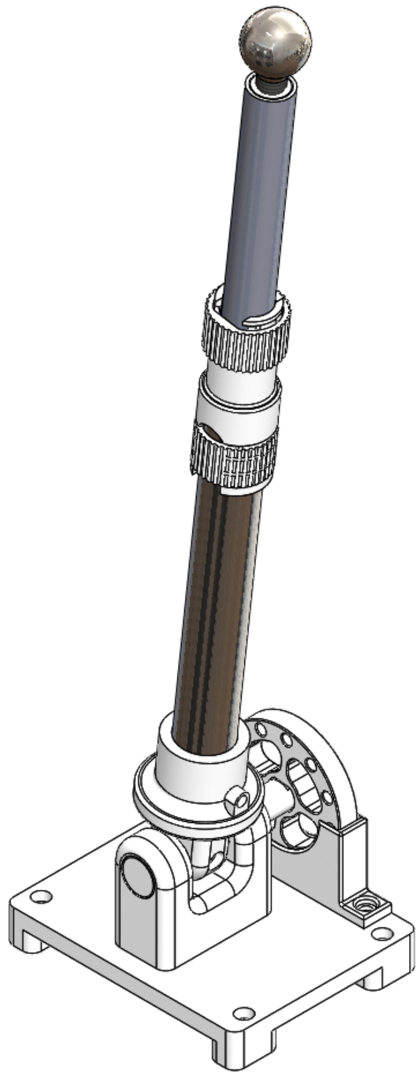


Figure 72: Final fixture assembly - V6.0

The fixture is made of three main sections: the *base plate*, the *telescopic pole* and the *Magnet Holders*. The schematic for all assemblies and parts for *Ballpoint Fixture - Version 6* can be found in Appendix J.

6.4.5.1 Version 6 - Base Plate

The schematic for the *Base Plate* can be found in Appendix J.

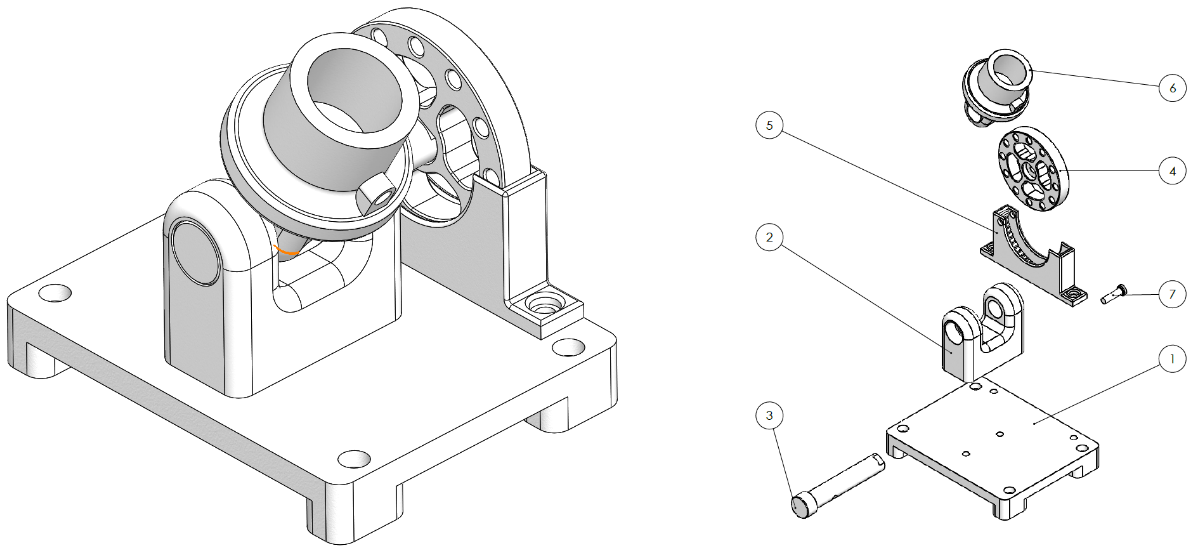


Figure 73: Fixture prototype V6.0 - Base Plate

Table 2: Bill of Material List - Base Plate

ITEM NO.	PART NUMBER	QTY.
1	Bottom Plate	1
2	Rotation Joint	1
3	Rotation Pin	1
4	Wheel	1
5	Wheel Joint	1
6	Pole Mount	1
7	Locking Pin	1

The *Base Plate* seen in Fig. 73 is the section which is mounted to the operator table and it gives the rotational functionality for the telescopic pole. It is made of the parts as seen in Table 2.

Item No.: 1 - Bottom Plate

The schematic for the bottom plate can be found in Appendix J.

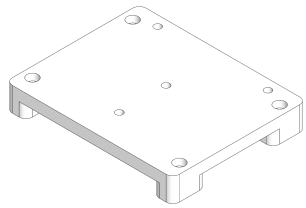


Figure 74: Fixture prototype V6.0 - Bottom Plate

The *Bottom Plate* seen in Fig. 74 is the part that is directly mounted to the operators work table. This is done using four M5 machine screws through each "leg". It also has connection points to the *Rotation Joint* and the *Wheel Joint*, also made compatible for M5 machine screws for simplicity and accessibility.

Most important print specifications:

- Fill density: 100%
- Print Quality: Standard

100% fill density was used to ensure a solid fixture foundation. Standard print quality creates a more precise extrusion of plastic than low quality, but it has lower precision than high and hyper quality. For a part like the bottom plate, which does not need high tolerances, standard print quality is sufficient.

Item No.: 2 - Rotation Joint

The schematic for the "rotation joint" can be found in Appendix J.

The *Rotation Joint* seen in Fig. 75 connects to the bottom plate as two M5 nuts are printed into the solid body. These nuts are accessible for the M5 machine screws from the *Bottom Plate*.

The *Rotation Joint* is also designed in a manner where the *Rotation Pin* can be mounted so that joint's mount hole is perfectly aligned with the center plane of the *Rotation Joint*. Most important print specifications:

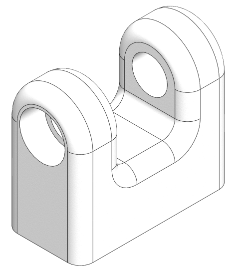


Figure 75: Fixture prototype V6.0 - Rotation Joint

- Fill density: 50%
- Fill Pattern: Hexagon
- Print Quality: High

50% fill density was used since the nuts connecting with the machine screws create a strong enough joint for the connection with the *Bottom Plate*. For the *Rotation Joint* hole placement and tolerances is a bit more critical. High print quality showed to be sufficient.

Item No.: 3 - Rotation Pin

The schematic for the *Rotation Pin* can be found in Appendix J.

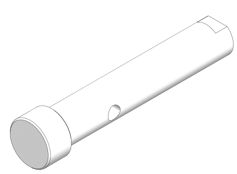


Figure 76: Fixture prototype V6.0 - Rotation Pin

The *Rotation Pin* seen in Fig. 76 slides through the *Rotation Joint* until the head touches the inner face of the latter. As for the *Rotation Joint*, one M5 nut is printed into the shaft of the *Rotation Pin*, allowing an M5 machine screw connect it to the *Wheel*. The flattened section is a safety lock to ensure that the angle of both the *Rotation Pin* and the wheel is the same. The through hole with a nut shape on one side makes it possible to connect the pole mount to the rest of the section using an M5 machine screw and nut.

Most important print specifications:

- Fill density: 100%
- Print Quality: High

Since the *Rotation Pin* must tolerate shear tension due to the applied torsion from the pole weight it was necessary to use 100% infill. The *Rotation Pin* needs to align its holes with holes on other parts. High print quality is used to ensure this.

Item No.: 4 - Wheel

The schematic for the *Wheel* can be found in Appendix J.

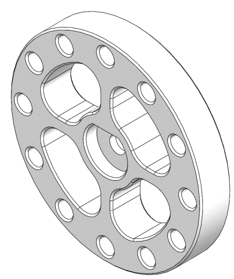


Figure 77: Fixture prototype V6.0 - Wheel

The *Wheel* seen in Fig. 77 is connected with the *Rotation Pin* and enables the operator to lock the pole in different orientations. The *Wheel* lays loose in the *Wheel Joint* which lets it rotate freely. The *Locking Pin* can easily be slid through the hole in the *Wheel Joint* and any of the holes along the *wheels* perimeter.
Most important print specifications:

- Fill density: 50%
- Shell Count: 2
- Fill Pattern: Hexagon
- Print Quality: High

Item No.: 5 - Wheel Joint

The schematic for the *Wheel Joint* can be found in Appendix J.

The *Wheel Joint* seen in Fig. 78 connects with the bottom plate using two M5 machine screws and nuts. It locks the *Wheel* in a specific plane and lets it rotate around its center axis which is the same as the *Rotation Pins'* center axis.

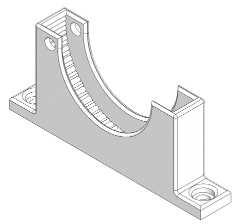


Figure 78: Fixture prototype V6.0 - Wheel Joint

Item No.: 6 - Pole Mount

The schematic for the *Pole Mount* can be found in Appendix J.

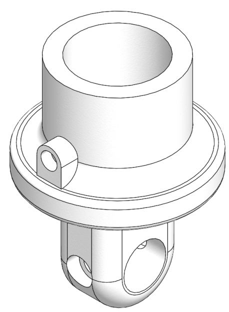


Figure 79: Fixture prototype V6.0 - Pole Mount

The *Pole Mount* is connected to the *Rotation Pin* with a through bolt which locks them together, making them rotate at the same rate around the same axis. The "Ø25x2 steel pole" has a tight fit in the "pole mount" and is locked in using a M3 set screw.

Item No.: 7 - Locking Pin

The schematic for the *Locking Pin* can be found in Appendix J.

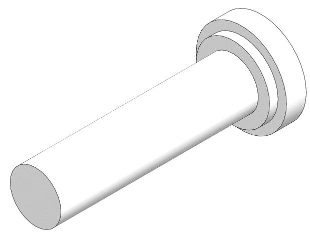


Figure 80: Fixture prototype V6.0 - Locking Pin

The *Locking Pin* has the simple function of locking the fixture in a certain orientation by sliding through the hole in the *Wheel Joint* and any of the holes along the *Wheels'* perimeter.

6.4.5.2 Version 6 - Telescopic Pole

The schematic for the *Telescopic Pole* can be found in Appendix J.

Table 3: Bill of Material List - Telescopic Pole

ITEM NO.	PART NUMBER	QTY.
8	Ø25x2mm Steel Pole	1
9	Pole Clamp	1
10	Pole Clamp Ring - Small	1
11	Pole Clamp Ring - Large	1
12	Ø20x1,5mm Steel Pole	1
13	Ø18, M12 Insert	1
14	Steel Ball Plug	1
15	Steel Ball Rubber Cover	1

The *Telescopic Pole* seen in Fig. 81 is the section which allows the operator to adjust the length of each pole in the fixture system. It is made of the parts seen in Table 3.

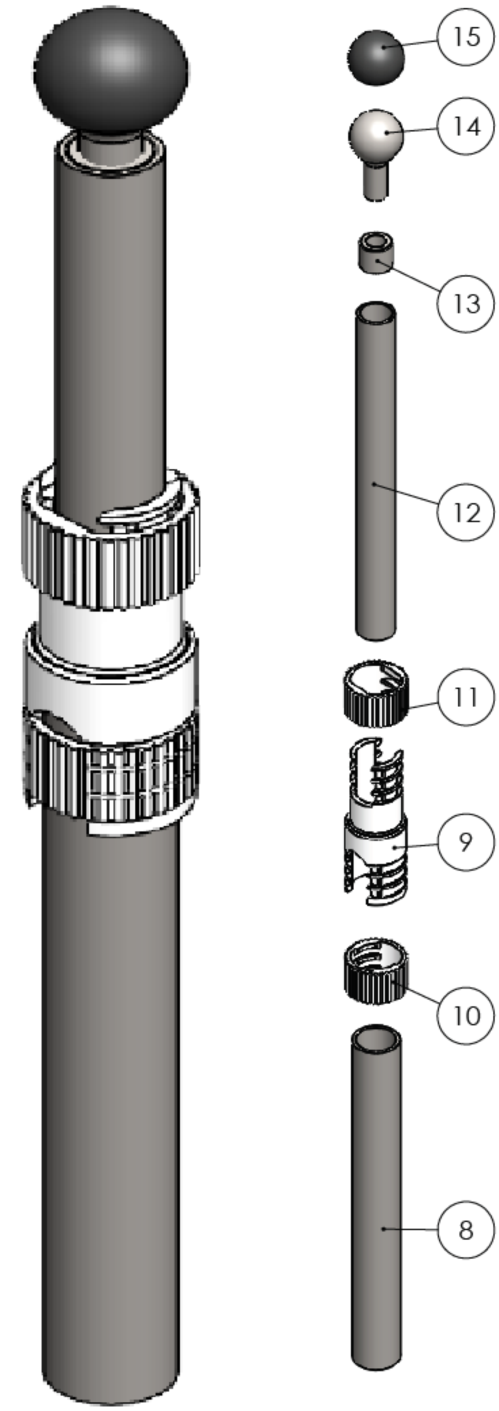


Figure 81: Fixture prototype V6.0 - Telescopic Pole

Item No.: 8 - Ø25x2mm Steel Pole

The schematic for the *Ø25x2mm Steel Pole* can be found in Appendix J.

Figure 82: Fixture prototype V6.0 - Ø25x2mm Steel Pole

The *Ø25x2mm Steel Pole* is of a generic St37 steel quality. It is cut to a length of 200mm and has a tight fit to the *Pole Mount*, which it is also locked to using a set screw. It also has a tight fit with the *Pole Clamp* which is tightened even harder when rotating the *Pole Clamp Ring - Large*.

The inner diameter of the pole is Ø21, which gives the *Ø20x1,5mm Steel Pole* a 0,5mm gap in every direction which lets it slide back and forth with no friction applied from the outer pole.

Item No.: 9 - Pole Clamp

The schematic for the *Pole Clamp* can be found in Appendix J.

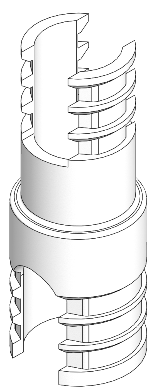


Figure 83: Fixture prototype V6.0 - Pole Clamp

The *Pole Clamp* seen in Fig. 83 is designed so that the larger steel pole has a tight fit already before tightening it further with the *Pole Clamp Ring - Large*, which is done by rotating it.

The same procedure was used when designing it to fit with the *Ø20x1,5mm Steel Pole*, except here we have used a looser fit so that it can slide easily back and forth when placing it for the composite panels. The same tightening function is used here as well with the *Pole Clamp Ring - Small*.
Most important print specifications:

- Fill density: 100%
- Print Quality: Standard

The *Pole Clamp* is not subjected to any high stresses, but since its walls are so thin it was almost as efficient to print it with 100% fill density as with 20% infill. Also, this makes the part unable to "flex" when the tightening rings are used, giving more pressure directly onto the poles.

Item No.: 10 - Pole Clamp Ring - Small

The schematic for the *Pole Clamp Ring - Small* can be found in Appendix J.

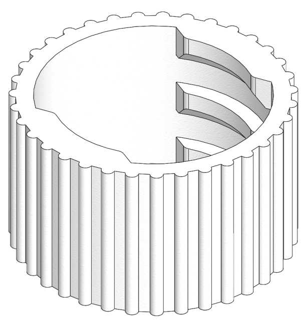


Figure 84: Fixture prototype V6.0 - Pole Clamp Ring - Small

The *Pole Clamp Ring - Small* seen in Fig. 84 is designed to slip over the outer surface of the *Pole Clamp* and pressing its walls towards the center with extruded areas, making it squeeze tight to the *Ø20x1,5mm Steel Pole*. The rough outside surface was design this way to make it easier to turn when it is tightened.
Most important print specifications:

- Fill density: 100%
- Print Quality: High

Since the *Pole Clamp Ring - Small* tightens around the pole clamp making it as solid as possible is essential for it not to flex or break when in the tightened position. Also, when printing this part to avoid using support structure we created inside chamfers in 45°, this process is less exposed when using high or hyper print quality compared to low or standard.

Item No.: 11 - Pole Clamp Ring - Large

The schematic for the *Pole Clamp Ring - Large* can be found in Appendix J..

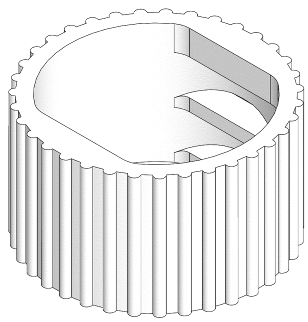


Figure 85: Fixture prototype V6.0 - Pole Clamp Ring - Large

The *Pole Clamp Ring - Large* shown in Fig. 85 is created the exact same way as *Pole Clamp Ring - Small* and can be read about in the *Item No.: 10 - Pole Clamp Ring - Small* chapter above.

Item No.: 12 - Ø20x1,5mm Steel Pole

The schematic for the *Ø20x1,5mm Steel Pole* can be found in Appendix J.

Figure 86: Fixture prototype V6.0 - Ø20x1,5mm Steel Pole

The *Ø20x1,5mm Steel Pole* shown in Fig. 86 is created the exact same way as *Ø25x2mm Steel Polel* and can be read about in that chapter section.

Item No.: 13 - Ø18, M12 Insert

The schematic for the *Ø18, M12 Insert* can be found in Appendix J.

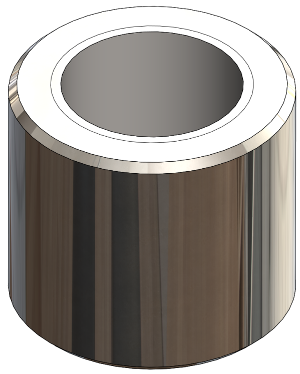


Figure 87: Fixture prototype V6.0 - Ø18, M12 Insert

The *Ø18, M12 Insert* is manufactured out of a hardened Sis2541 steel alloy which gives it great wear-and-corrosion resistance along with longer durability than regular S37 steel. [112] For our application the insert is pressed into the *Ø20x1,5mm Steel Pole* with a k17 tolerance, which is a relatively tight fit. In addition, it is also threaded so that the operators can fine tune the height of ball point. Both of these functions is fatiguing for the insert which is why it is critical that the material properties of the alloy fits the purpose.

Item No.: 14 - Steel Ball Plug

The schematic for the *Steel Ball Plug* can be found in Appendix J.

The *Steel Ball Plug* is manufactured out of a with good magnetic properties, unlike, for example stainless steel. This is essential for the *Magnet Holder* so it can pull towards it, as well as squeeze and hold the composite panels in place. It has a M12 fine pitch thread which matches the *Ø18, M12 Insert* which enables the operator to adjust the height of the fixture more precisely.

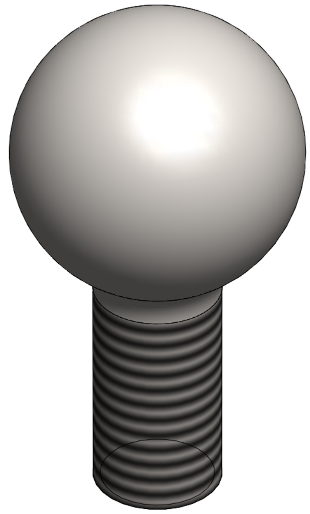


Figure 88: Fixture prototype V6.0 - Steel Ball Plug

Item No.: 15 - Steel Ball Rubber Cover

The schematic for the *Steel Ball Rubber Cover* can be found in Appendix J.

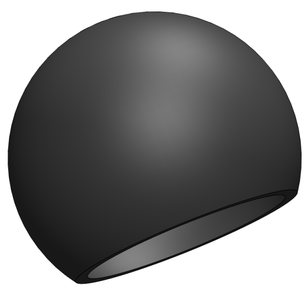
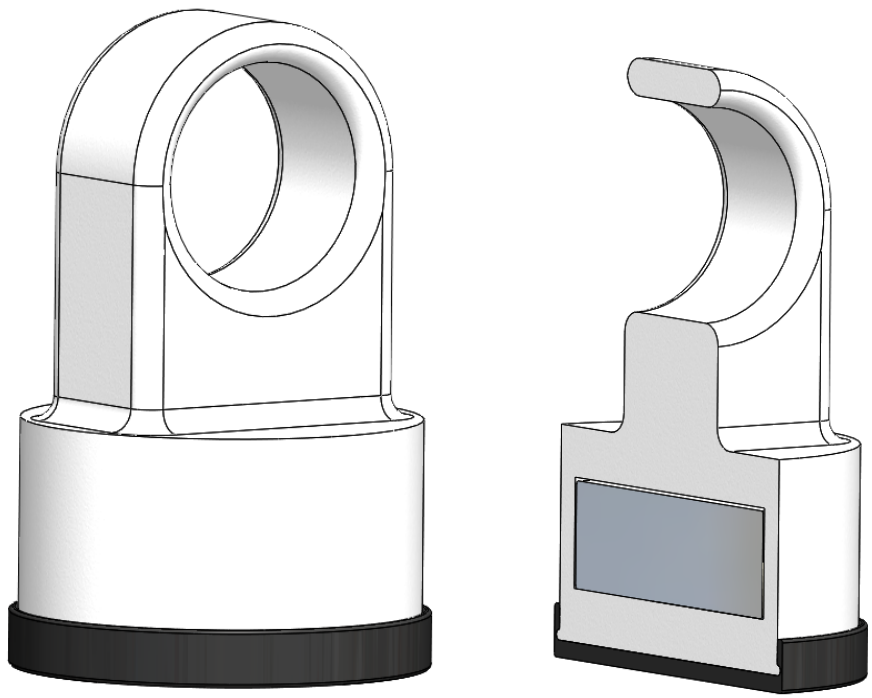


Figure 89: Fixture prototype V6.0 - Steel Ball Rubber Cover

The *Steel Ball Rubber Cover* is printed using thermoplastic elastomer (TPU), which is a form of flexible plastic. This cover is pressed over the ball point which creates a protective layer between the composite panels and the steel ball preventing scratches, damage, etc. to the panels.



(a) (b)

Figure 90: Final magnet assembly

6.4.5.3 Magnet Holders

Disk magnets [113] Ø30x10 mm with pull force of 20 kg were printed into a magnet holders, as seen in Fig. 90. The holders were supplied with printed PPU rubber covers to prevent damage to the composite panels. The magnets are used to secure the inspection panels in specific positions in order to aid inspectors with their work and ensure the precise alignment of AR content onto the parts (see Sec. 6.3.2). The final magnet assembly is shown with section view in Fig. 90 (b).

6.4.6 Assembled Fixture

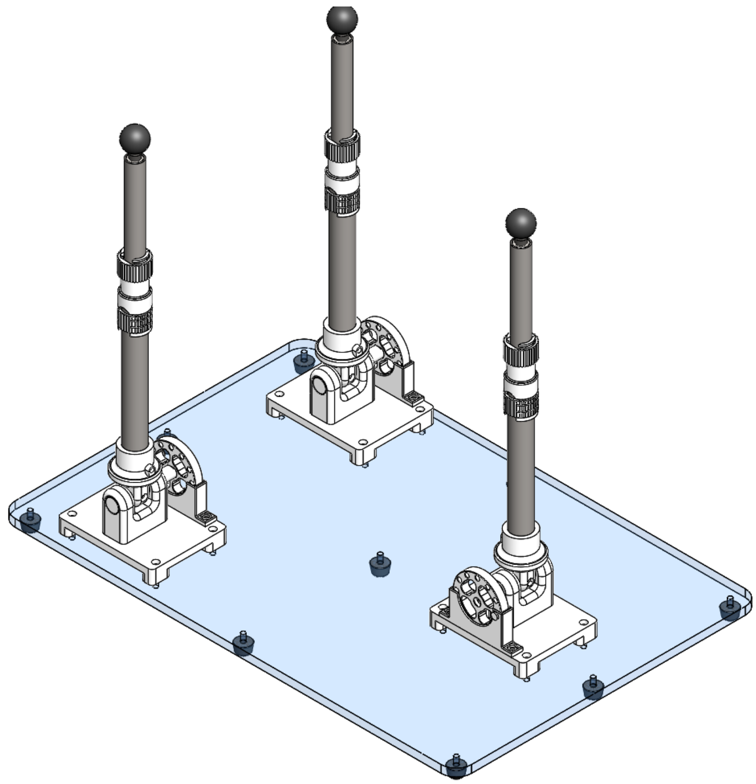


Figure 91: Fixture prototype V6.0 - Complete SolidWorks Model

Depending on the size of the panel being worked on the operator can use multiple ball point fixtures, as shown in Fig. 91. In this case the plexi glass table which had laser cut holes for mounting the *Ball Point Fixtures* and for its legs is used instead of an actual working table which is used at KDA.

The final manufactured fixture is shown in Fig. 92 (a), and Fig. 92 (b) shows it holding a composite panel in position using the *Magnet Holders*.

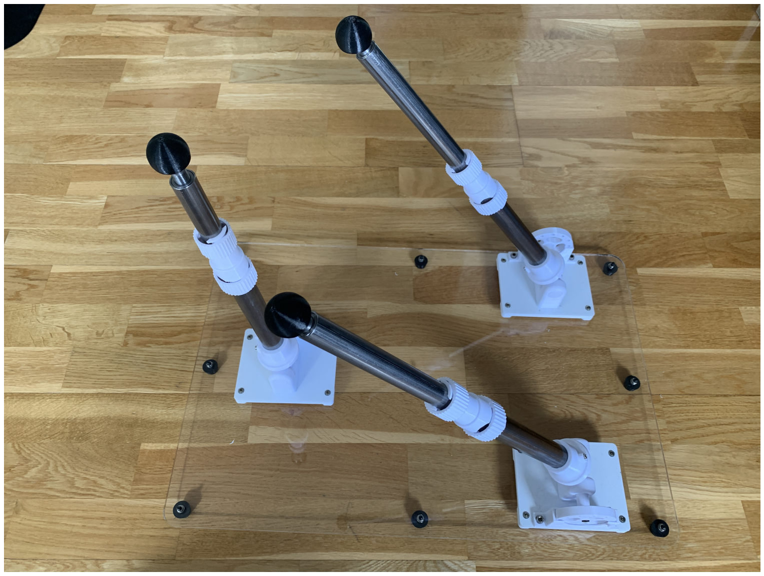


Figure 92: Fixture prototype V6.0 - Complete Fixture Model

Figure 93: Fixture prototype V6.0 - Fixture Holding Composite Panel

7 Review and Future Work

Having reached the end of the project one cannot help but reflect on not only the results achieved, but also on the potential improvements. This chapter's goal is to briefly summarize two broad topics - evaluate and review of the overall work done, and recommendations for further development. To keep it short and concise, the team has decided to categorize it into five sub-chapters, that we have deemed to be of most importance.

7.1 The XRI Application

Considering the fact that the team has set its main goal to demonstrate opportunities created by utilizing HoloLens, the team has focused on primarily on implementing the requirements set by the client. This has proven to be effective in proving the concept, but has weakened our grip on other elements that could make an application successful. These quality-of-life improvements are the main focus of this chapter, built atop the analysis of work done.

While working on the software the development team has put a priority on writing an encapsulated code that would allow to easily expand the functionality of the application and add new features on demand. All modules discussed in the sub-sections of Chapter 6 are self-consistent, and will give opportunity to upgrade the application with ease. To further test new potential features, the team highly suggests to utilize the existing structure, as it will speed up the process.

Regarding the user interface design and user experience, team's recommendation is to switch to a full 3D menu navigation system, that will utilize the space between the inspector and the measured part. Some simple tests showed that adjusting the position and angle of the menu elements relative to the position of the inspector, the direction they are facing and the location of the part provide a smoother navigation experience, make it easier to work on the part, and the holograms become less obtrusive when one is focusing on the physical objects. This will also allow to completely remove the white background, which as we noted in the User Interface Design (see Sec. 5.3) could bring discomfort to the user's eyes.

Sec. 6.2.3, Data Storage, discusses how the current file saving system works, and to improve the user-friendliness, the measurement-files directory, ideally, should be automatically moved to the user's computer. Luckily, *Windows Device Portal* (WDP) comes with an API to handle files [42]. The assessment from reading the documentation is that it should be possible to create a feature to the current application that would automatically the move the measurements directory to the computer.

7.2 Computer Vision and MR Head-mounted Displays

One of the most comprehensive and at the same time interesting technologies with lots of potential that the team encountered during the project is computer vision (CV) for MR head-mounted displays (HMD). Computer vision in itself is a complex scientific field, add the XR technology with the markets' top end device, such as HoloLens, in the mix and you have an ocean of possibilities.

The team had slim to none prior knowledge or experience with CV, OHMD and MR technologies which resulted in a very steep learning curve. The research conducted with the limited time and resources is not groundbreaking, but hopefully, can provide the customer with a solid background to consider further research and development.

Two image recognition and tracking development tools were reviewed, HoloLensAR-ToolKit and Vuforia (see Sec. 3.3.3). Given that there were no time to bet on two horses, the team decided to stick with the ARToolKit (see Sec. 6.3.2) for several reasons. Besides the reasons listed in Sec. 3.3.3, such as price and complexity of the Vuforia software, ARToolKit is an open-source software which gives the team more freedom to peek behind the scenes and understand how it works. The question that the customer is eventually left with is whether to invest in further development of a solution that is solely fitted to the company's needs, or lease an expensive off-the-shelf solution that has more functionalities than required, that still has to to be customized and fitted to the process.

Keep in mind that Vuforia is a high-profile business company and things are often not as easy as they look in the professional promotion videos and YouTube. Computer vision is still in its infancy, and all solutions on the market suffer from the same problems, as mentioned in Sec. 3.3.3.2. However, the team would have liked more time to test the Vuforia's Model Targets feature further and experience more of these limitations ourselves, in order to give a more thorough assessment.

Regarding future development with ARToolKit, it is HoloLens 2 compatible. Minor differences is that the camera calibration file for the ARToolKit (see Sec. 3.3.4) and the Video Parameter in the ARUWP Video (Script) of ARUWP Controller object (see Sec. 6.3.2.1) would need to be updated to 1504 x 846 resolution profile based on Microsoft's Mixed Reality Documentation on Platform capabilities and APIs [108].

The most significant and crucial task remaining is the manual camera-projection calibration described in Sec. 6.3.2.2 and the most reliable approach to achieve a perfect alignment would be to recreate the *Multipoint Single 3D Object Method* mentioned in Sec. 6.3.2.2. Note, however, that the author calibrated v0.3 version of the ARToolKit for the HoloLens 2, so the offset issue the team experienced might not be that big of an issue when ported for HoloLens 2 after all.

The next step could be, as mentioned in Sec. 6.3.2, fixing projected holograms in space using a WorldAnchor.

No matter what solution is chosen it should be integrated into the main application after it has been ported for HoloLens 2.

7.3 Porting for HoloLens 2

The team kept the objective of easily upgradable software thorough the development process to ensure compatibility and seamless transition to HoloLens 2, as mentioned in Sec. 3.2 and 3.3.1. Besides the aspects mentioned there, it is recommended to take a look at Microsoft's guidelines on the porting process for HoloLens 1 application to HoloLens 2 device which can be found in Microsoft's Mixed Reality Documentation [11].

7.4 Micrometer Communication

The development of the micrometer communication has not been a linear path, but the team believes that the work has provided a proof of concept when it comes to connecting a micrometer with the HoloLens headset in a way that is applicable to many other measurement-tools and computer-devices.

Initially, prototype development was done using a breadboard, and divided into two parts; communication between the micrometer and the Bluetooth-transmitter, and communication between the Bluetooth-transmitter and the XRI-application.

With the Adafruit Bluetooth-device as a basis, the team started testing data-transmission of hard-coded data to .txt-documents through text-editors on mobile phones and PC's due to limited accessibility of the HoloLens headset (see Appendix F). Accessibility of the micrometer was also limited at first, so the team tried to get as much information about the protocols as possible through testing when the micrometer was available and through e-mail correspondence with Insize. The most time-consuming part of the process was figuring out the protocol from the micrometer. This was due to multiple instances of miscommunication with Insize where received data sheets lacked crucial information that the team tried to fill in through trial and error. At the end, the team had to move from the initial plans on communicating with the micrometer without the Insize Bluetooth-devices to assembling a system around both the Insize-devices.

After finally getting data from the micrometer to the Adafruit-device, the focus was on reading data to the XRI-application and parsing it to float-values.

For the micrometer communication, a natural next step in development will be to get the protocol needed for the Adafruit-device to communicate with the micrometer directly. This will make the design more compact and cheaper since the Insize Bluetooth-devices would not be needed anymore. This system will also work on a PC in the same way as the Insize Bluetooth-system does on its own, but without the need of a receiving external module connected to a computer.

Another option available is implementing standby-mode for the Adafruit-device, which will increase battery-life and decrease the frequency of down-time due to charging.

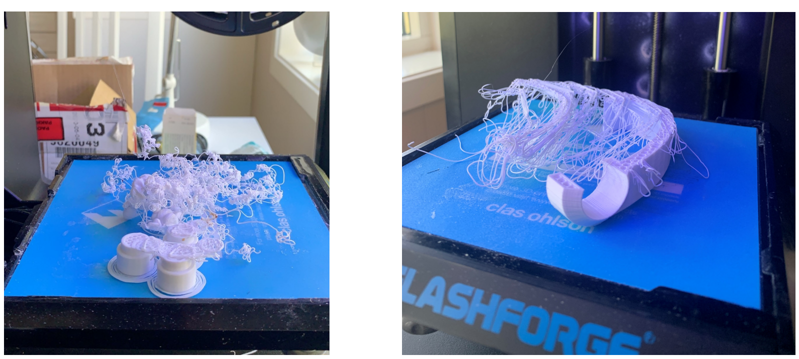
There are also unused potential in the Adafruit-device concerning data management. Through simple coding, the device can save measurements temporarily, handle both metric and imperial units and communicate with the HoloLens using any ASCII-characters. An example of application can be to mount an undo-button to the Bluetooth-module, which can send a message to the HoloLens for it to delete the last measurement.

7.5 Fixtures

Creating the fixture prototype has been a timely and at times difficult task. As the project is dominated by software development and the group consists of three computer engineering students, one electrical engineering student and one mechanical student. At the project start it was unsure if it was necessary to use fixtures for the system, as no one from the group had any particular experience with the HoloLens software and the precision of the alignment technology used. This created a task with very little framework and limits, which made it hard to start with anything specific from the beginning. During the project period it showed to be a crucial implementation, as the object recognition software on the HoloLens proved to be inadequate as is.

The COVID-19 pandemic (see Sec. 4.3) made it particularly difficult to create anything physical as the school went into complete lock down, making it impossible for the group to create parts at campus and hard to order parts and tools (as metal profiles, screws, nuts, washers, etc.) from there as we weren't able to physically see what was available. This created a great deal of delay as we had a lot of back-and-forth communication with the campus representatives responsible for order and availability of materials.

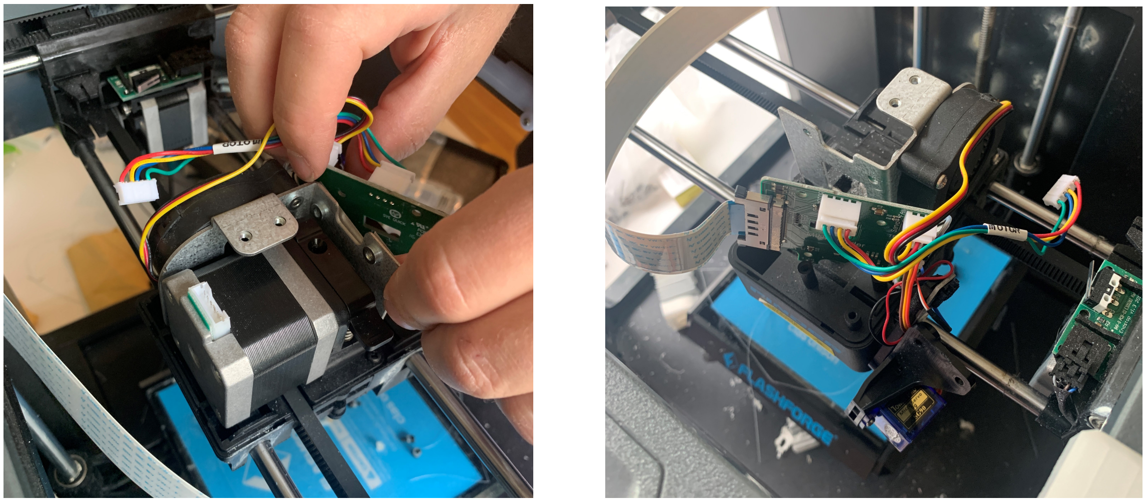
The lock down also affected the project's client, KDA, which became even stricter regarding access to their facilities. Therefore, the mechanical team, who normally have access to the facility wasn't allowed on the premises for approximately two months. This limited the opportunity to talk to the machine workers there and asking them to help with manufacturing parts for the project. All this uncertainty made ordering parts for the project quite problematic, time consuming, and at times, impossible. Luckily, we were able to take advantage of our contact network and thereby borrow a 3D-printer for the remainder of the project period, making it possible to actually produce a physical product. Unfortunately, we also had some problems with the printer as it struggled to pull the filament thread through the extruder. Resulting in print failures (see Fig. 94).



(a) (b)

Figure 94: Failed 3D print

This was a production delaying problem and a lot of time was put into fixing it, taking time from other tasks that were directly project related. As shown in Fig. 95, it was necessary to open the printer head and take it apart for cleaning purposes, switching of almost defect parts, tightening springs, etc..



(a) (b)

Figure 95: Fixing the 3D printer

The last month of the project, the mechanical team was allowed back to KDAs facilities and get some manufacturing of some of the components done by the operators there. Most of this was already communicated via e-mail, but it proved important to have the physical dialog to make sure that the schematics told the operator what exactly the group wanted for the final products.

As the prototype is finalized and manufactured, the team believes it shows a proof of concept that is presentable as a final product for the project. To follow up with the prototype we recommend that the client evaluates the fixture's functions and continues the work on a fixture with the same functionalities more functional with the part sizes that is in their production process.

8 Conclusion

Can the quality control process at KDA (see Sec. 2.1.2) be improved in terms of efficiency and reliability with MR technology and Microsoft HoloLens? This was the initial question the team has faced with, which is also well-illustrated by the requirements expressed by the customer (see. Sec. 2.2.1). Unfortunately, one cannot wrap up a 200 page report with a simple yes or no. The difficulty in doing so, of course, lies in the richness of the detail and nuances this project has been riddled with. So instead, the conclusion will present team's main findings in broad stokes, in hopes that it gives a better insight into the project.

In-house testing showed that the efficiency can be indeed enhanced with a head mounted computer (HMC), such as HoloLens, and the XRI application substantiates this statement. Hands-free access to information in demand, wireless transmission of the micrometer measurements to the application, and digitization of the data transfer can do wonders for the inspectors, decreasing human errors and increasing convenience and speed of the entire process, while image capturing of the measurements with the built-in camera improves tracebility and documentation of the process. Fixing the composite panels in specific positions proved to be a crucial factor for the HoloLens' object tracking software to function properly, while also giving the operator an easier working experience as the panel are locked in convenient positions for the measurement process. With a proper introduction, the system is easy to use regardless of age and technical background, while meeting security standards of the client.

In contrast, there are still areas that require improvements before the product is ready for deployment in a wide scale industrial setting. The fixture components must be designed for a higher carry load, as the plastic components are fragile and varies a lot in quality as is. The fixture itself may be scaled to fit the panel sizes at KDA better and the mounting brackets for the working table (*Bottom plate*) may be changed for an easier deployment. First generation HoloLens has quite uncomfortable head-fit system and it is not suitable to be worn for very long at a time. Also, improvements in tracking accuracy are required to successfully reduce the possibility of human errors during the measurement process, and implement a guidance system that overlays AR content onto physical objects. Many of the shortcomings of the HoloLens 1 are rectified in the second generation of the device, HoloLens 2, and an evaluation of the new headset is earnestly recommended to draw a full conclusion on whether the improvements in tracking accuracy and other areas are adequate for the industrial usage.

References

[1] *Aerostructures - Smart Manufacturing*, 2019.
Available: https://www.kongsberg.com/kda/products/aerostructures/ [Accessed Jan. 20, 2020].

[2] *Aerostructures - Composites*, 2019.
Available: https://www.kongsberg.com/kda/products/aerostructures/pt-composites/ [Accessed Feb. 5, 2020].

[3] *About the XRA*. XR Association, 2020.
Available: https://www.xra.org/ [Accessed Feb. 8, 2020].

[4] *XR Primer 2.0: A Starter Guide for Developers*. XR Association, 2019.
Available: https://xra.org/wp-content/uploads/xra-primer-2.0-web-v2.pdf [Accessed Feb. 8, 2020].

[5] Matthew Corley. SANTA CRUZ, CALIFORNIA - JULY 12, 2016: The hit augmented reality smartphone app "Pokemon GO" shows a Pokemon encounter overlain on a real world trail in the forest. Shutterstock, 2016.
Available: https://www.shutterstock.com/nb/image-photo/santa-cruz-california-july-12-2016-451598275 [Accessed Feb. 19, 2020].

[6] P. Milgram and F. Kishino. *A Taxonomy of Mixed Reality Visual Display. IEICE Transactions on Information Systems*, E77-D, No.12, 1994.

[7] *Unity development overview*. 2020.
Available: https://docs.microsoft.com/en-us/windows/mixed-reality/unity-development-overview [Accessed April 18, 2020].

[8] *Unity*.
Available: https://unity.com/ [Accessed April. 18.04, 2020].

[9] *Unreal Engine 4 Documentation - Microsoft HoloLens Development*.
Available: https://docs.unrealengine.com/en-US/Platforms/AR/HoloLens/index.html [Accessed April 18, 2020].

[10] *Getting started with MRTK*.
Available: https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/GettingStartedWithTheMRTK.html [Accessed April 18, 2020].

[11] *Get your existing app ready for HoloLens 2*. Microsoft, 2019.
Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-porting-guide [Accessed Mar. 17, 2020].

[12] *What's a Universal Windows Platform (UWP) app?*
Available: https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide [Accessed April 18, 2020].

[13] *Scripting Backends*. Unity Documentation.
Available: https://docs.unity3d.com/2018.4/Documentation/Manual/windowsstore-scriptingbackends.html [Accessed April 18, 2020].

[14] *Universal Windows Platform: IL2CPP scripting back end*.
Available: https://docs.unity3d.com/2018.4/Documentation/Manual/windowsstore-il2cpp.html [Accessed April 18, 2020].

[15] *Jira — Issue Project Tracking Software — Atlassian*.
Available: https://www.atlassian.com/software/jira [Accessed Feb. 26, 2020].

[16] *Where work happens — Slack*.
Available: https://slack.com/ [Accessed Feb. 26, 2020].

[17] *GitHub*.
Available: https://github.com/ [Accessed Mar. 20, 2020].

[18] *GitHub Help Documentation*. GitHub.
Available: https://help.github.com/en/github [Accessed May 15, 2020].

[19] *GitHub Student Developer Pack*.
Available: https://education.github.com/pack [Accessed May 15, 2020].

[20] *GitKraken Git GUI*.
Available: https://www.gitkraken.com/git-client [Accessed May 15, 2020].

[21] *QR code tracking*. Microsoft.
Available: https://docs.microsoft.com/en-us/windows/mixed-reality/qr-code-tracking [Accessed April 18, 2020].

[22] T. Krogh-Jacobsen. *Unity 2019.3 is now available*. Unity Blog, 2020.
Available: https://blogs.unity3d.com/2020/01/28/unity-2019-3-is-now-available/ [Accessed April 18, 2020].

[23] *Vuforia Engine - The Worls's Most Widly Deployed AR.* Vuforia.
Available: https://engine.vuforia.com/engine [Accessed April 21, 2020].

[24] L. Qian. *HoloLens with ARToolKit v0.2.* GitHub.
Available: https://github.com/qian256/HoloLensARToolKit [Accessed April 18, 2020].

[25] P. Lamb. *ARToolKit Homepage.*
Available: http://www.hitl.washington.edu/artoolkit/ [Accessed April 20, 2020].

[26] L. Qian. *About me.* Personal blog of Long Qian, 2019.
Available: https://longqian.me/aboutme/ [Accessed April 18, 2020].

[27] L. Qian. *ARToolKit on HoloLens.* Personal blog of Long Qian, 2017.
Available: longqian.me/2017/01/20/artoolkit-on-hololens/ [Accessed April 18, 2020].

[28] *GNU Lesser General Public License.*
Available: https://www.gnu.org/licenses/lgpl-3.0.en.html [Accessed April 18, 2020].

[29] N Navab P. Kazanzides E. Azimi, L. Qian. *Alignment of the Virtual Scene to the Tracking Space of a Mixed Reality Head-Mounted Display.* Journal: arXiv preprint arXiv:1703.05834.
Available: https://arxiv.org/pdf/1703.05834.pdf [Accessed May 15, 2020].

[30] A. Mahendran M. Fuad. *Unity XR platform updates.* Unity Blog, 2020.
Available: https://blogs.unity3d.com/2020/01/24/unity-xr-platform-updates/ [Accessed April 19, 2020].

[31] *Using Vuforia Engine with Unity.*
Available: https://docs.microsoft.com/en-us/windows/mixed-reality/vuforia-development-overview [Accessed April 18, 2020].

[32] *License Manager FAQ.* Vuforia Developer Library.
Available: https://library.vuforia.com/articles/FAQ/License-Manager-FAQ [Accessed April 30, 2020].

[33] *Vuforia - Attach digital content to specific objects.*
Available: https://engine.vuforia.com/content/vuforia/en/features.html [Accessed April 22, 2020].

[34] *Introduction to Model Targets in Unity.*
Available: https://library.vuforia.com/features/objects/model-targets.html [Accessed April 22, 2020].

[35] *Vuforia Pricing.*
Available: https://developer.vuforia.com/pricing [Accessed April 21, 2020].

[36] *Model Targets Supported Objects CAD Model Best Practices.* Vuforia Developer Library.
Available: https://library.vuforia.com/content/vuforia-library/en/articles/Solution/model-targets-supported-objects.html [Accessed April 30, 2020].

[37] *External Camera Calibration.* Vuforia Developer Library.
Available: https://library.vuforia.com/content/vuforia-library/en/articles/Solution/external-camera-calibration.html [Accessed April 21, 2020].

[38] L. Qian. *HoloLens Camera Calibration.* GitHub, 2020.
Available: https://github.com/qian256/HoloLensCamCalib/tree/7a243c2af858c55f13083919ee4a42bc276e988b [Accessed April 22, 2020].

[39] *ARToolKit - Coordinate systems.*
Available: http://128.95.75.17/artoolkit/documentation/cs.htm [Accessed April 21, 2020].

[40] M. Billinghurst H. Kato. *Marker tracking and HMD calibration for a video-based augmented reality conferencing system.* IEEE, 2002.
Available: https://ieeexplore.ieee.org/document/803809 [Accessed April 21, 2020].

[41] *OpenCV.*
Available: https://opencv.org/ [Accessed April 28, 2020].

[42] *Using the Windows Device Portal.*
Available: https://docs.microsoft.com/en-us/windows/mixed-reality/using-the-windows-device-portal [Accessed April 24, 2020].

[43] *Improve visual quality and comfort.* Microsoft.
Available: https://docs.microsoft.com/en-us/hololens/hololens-calibration [Accessed May 14, 2020].

[44] *Insize Sheet Metal Micrometer*. Indiamart.
Available: https://www.indiamart.com/proddetail/insize-sheet-metal-micrometer-20812053962.html [Accessed Feb. 2, 2020].

[45] W. Bolton. *Mechatronics - Electronic Control Systems in Mechanical and Electrical Engineering*. Pearson Educational Limited, 6 edition, 2015.

[46] G. Schuetz. *Micrometer Accuracy*. Mahr Metrology, unknown.
Available: https://www.mahr.com/en/Services/Production-metrology/Know-how/Gaging-Tips/Hand-Tools-Gaging-Tips/?ContentID=19087&Overview=0 [Accessed Feb. 6, 2020].

[47] *Measuring Instruments - Catalogue no. 203*, 2017-2018.

[48] Microsoft Corporation. Hardware Accessories. 2018.
Available: https://docs.microsoft.com/nb-no/hololens/ [Accessed Feb. 10, 2020].

[49] *iGaging Bluetooth Smart Adapter Dongle - 35-BT-28*. Penn Tool Co., 2020.
Available: https://www.penntoolco.com/igaging-bluetooth-smart-adapter-dongle-35-bt-28/ [Accessed Mar. 20, 2020].

[50] *TLC-BLE Bluetooth transmitter*. Tesa Technology, 2020.
Available: https://tesatechnology.com/en-gb/products/connectivity/wireless-transmission-systems/tlc-ble-bluetooth-transmitter_p40479.htm#/collapseComplementaires [Accessed Mar. 20, 2020].

[51] *Wireless Measurement Data Communication System - U-Wave*. Mitutoyo America Corporation, 2015.
Available: https://www.mitutoyo.com/wp-content/uploads/2015/10/U-WAVE.pdf [Accessed Mar. 20, 2020].

[52] *Wireless Measurement Read WiMER Series 1*. Motionics LLC, 2020.
Available: https://store.motionics.com/products/wireless-measurement-read-wimer-series-1 [Accessed Mar. 20, 2020].

[53] Circuit Basics. *Basics of UART Communication*. Circuit Basics.
Available: https://www.circuitbasics.com/basics-uart-communication/ [Accessed May 20, 2020].

[54] *Radio Versions.* Bluetooth Special Interest Group, 2020.
Available: https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/radio-versions/ [Accessed Mar. 19, 2020].

[55] A.Huang and L.Rudolph. Bluetooth for Programmers. 2005.
Available: http://people.csail.mit.edu/rudolph/Teaching/Articles/PartOfBTBook.pdf [Accessed Mar. 19, 2020].

[56] D.Coleman S.Mistry and A.Allan. Make: Bluetooth. Make: Community, 2015.
Available: https://learning.oreilly.com/library/view/make-bluetooth/9781680451108/ [Accessed Mar. 19, 2020].

[57] *Human Interface Device Profile 1.1.* Bluetooth Special Interest Group, 2015.
Available: https://www.bluetooth.com/specifications/archived-specifications/ [Accessed Mar. 19, 2020].

[58] J. Greene and A. Stellman. *Head First Agile.* O'Reilly Media, Inc., 2017.

[59] M. Cohn. *Succeeding with agile.* Addison-Wesley Professional, 2010.

[60] *Trello.*
Available: https://www.trello.com [Accessed Feb. 26, 2020].

[61] M. Rehkopf. *User Stories with Examples and Template.*
Available: https://www.atlassian.com/agile/project-management/user-stories [Accessed Feb. 6, 2020].

[62] M. Rehkopf. *Sprints.*
Available: https://www.atlassian.com/agile/scrum/sprints [Accessed Feb. 6, 2020].

[63] J. F. Smart. *BDD in Action.* Manning Publications, 2014.

[64] P. Hamill. *Unit Test Frameworks.* O'Reilly Media, 2004.

[65] D. Farinelli. *The Benefits of Test-driven Development.* DevOps, 2018.
Available: https://devops.com/the-benefits-of-test-driven-development/ [Accessed Jan. 17, 2020].

[66] N. Pryce and S. Freeman. *Growing Object-Oriented Software, Guided by Tests.* Addison-Wesley Educational Publishers Inc, 2009.

[67] B. Yehling and A. Cortez. *The Complete Idiot's Guide to Risk Management*. Alpha, 2010.

[68] Project Management Institute. *The Standard for Risk Management in Portfolios, Programs, and Projects*. Project Management Institute, 2019.

[69] K. Baxter. *Fast Track to Success: Risk Management*. Pearson Business, 2010.

[70] J. Talot. *Whats right with risk matrices*. DevOps, 2018. Available: `https://www.juliantalbot.com/post/2018/07/31/whats-right-with-risk-matrices` [Accessed Jan. 17, 2020].

[71] Helsedirektoratet. *Closing of educational institutions*. Available: `https://www.helsedirektoratet.no/nyheter/helsedirektoratet-har-vedtatt-omfattende-tiltak-for-a-hindre-spredning-av-covid-19/` [Accessed Mar. 20, 2020].

[72] S. Bennett. *Schaum's outline of UML*. McGraw-Hill, 2005.

[73] R. Miles and K. Hamilton. *Learning UML 2.0*. O'Reilly Media, Inc., 2006.

[74] G. Bonnema. *Systems design and engineering - facilitating multidisciplinary development*. Abingdon: CRC Press, 2017.

[75] *Easy Agile User Story Maps for Jira*. Available: `https://marketplace.atlassian.com/apps/1212078/easy-agile-user-story-maps-for-jira?hosting=cloud&tab=overview` [Accessed Mar. 19, 2020].

[76] J. Patton. *The New User Story Backlog is a Map*. Jeff Patton Associates, 2008. Available: `https://www.jpattonassociates.com/the-new-backlog/` [Accessed Feb. 29, 2020].

[77] P. Economy J. Patton. *User Story Mapping*. O'Reilly Media, 2014.

[78] *Figma: The collaborative interface design tool.* Available: `https://www.figma.com/` [Accessed Mar. 18, 2020].

[79] *Unity - Scripting API: PhraseRecognitionSystem* . Available: `https://docs.unity3d.com/ScriptReference/Windows.Speech.PhraseRecognitionSystem.html` [Accessed May 15, 2020].

[80] M. Jahn. *ZXing.Net*. NuGet, 2019.
Available: `https://www.nuget.org/packages/ZXing.Net` [Accessed April 21, 2020].

[81] M. Taulty. *About*. Personal blog of Mike Taulty.
Available: `https://mtaulty.com/about/` [Accessed April 20, 2020].

[82] M. Taulty. *Windows 10, UWP, QR Code Scanning with ZXing and HoloLens*. Personal blog of Mike Taulty.
Available: `https://mtaulty.com/2016/12/28/windows-10-uwp-qr-code-scanning-with-zxing-and-hololens/` [Accessed April 20, 2020].

[83] *Windows Mixed Reality Developer Forum*.
Available: `https://forums.hololens.com/` [Accessed April 20, 2020].

[84] *Stack Overflow*.
Available: `https://stackoverflow.com/` [Accessed April 20, 2020].

[85] A. Lian. *Unity 5.5 is ready for you*. Unity Blog, 2016.
Available: `https://blogs.unity3d.com/2016/11/29/unity-5-5-is-ready-for-you/` [Accessed April 20, 2020].

[86] *ARToolKit - Using Other Patterns*.
Available: `www.hitl.washington.edu/artoolkit/documentation/devmulti.htm` [Accessed May 21, 2020].

[87] Eden Networks Ltd. *Marker generator for ARToolKit v4.5.3 and later*.
Available: `https://au.gmented.com/app/marker/marker.php` [Accessed May 21, 2020].

[88] *Unity - Scripting API: MonoBehaviour* .
Available: `https://docs.unity3d.com/ScriptReference/MonoBehaviour.html` [Accessed May 15, 2020].

[89] The free dictionary. *persistentStorage*.
Available: `https://encyclopedia2.thefreedictionary.com/persistent+storage` [Accessed May. 19th, 2020].

[90] Y. Shafranovich. *csvDesc*. The Internet Society, 2005.
Available: `https://tools.ietf.org/html/rfc4180` [Accessed May. 19th, 2020].

[91] William Stallings. *Operating systems, internals and design principles*. Pearson, 9 edition, 2018.

[92] Microsoft. *WDP*.
Available: https://docs.microsoft.com/en-us/windows/uwp/debug-test-perf/device-portal?redirectedfrom=MSDN [Accessed May. 19th, 2020].

[93] Qing Yu Jiebo Luo and M.E Miller. *JpegCompression*.
Available: https://ieeexplore.ieee.org/document/1038063 [Accessed May. 19th, 2020].

[94] Microsoft. *windowsStorage*.
Available: https://docs.microsoft.com/en-us/uwp/api/windows.storage?view=winrt-18362 [Accessed May. 19th, 2020].

[95] Microsoft. *fileUnity*.
Available: https://forum.unity.com/threads/unityengine-windows-file-vs-system-io-file.577363 [Accessed May. 19th, 2020].

[96] T.Ha and K.Townsend. *Bluefruit nRF52 Feather Learning Guide*. Adafruit, 2017.
Available: https://learn.adafruit.com/bluefruit-nrf52-feather-learning-guide/introduction [Accessed Mar. 19, 2020].

[97] Texas Instruments. *A True System-on-Chip Solution for 2.4-GHz IEEE802.15.4 and ZigBee Applications*. Texas Instruments.
Available: http://www.ti.com/lit/ds/symlink/cc2530.pdf?&ts=1589969846800 [Accessed May 20, 2020].

[98] Banggood. *Mini USB Host Shield 2.0 ADK SLR Development Tool Geekcreit for Arduino*. Banggood.
Available: ?https://www.banggood.com/Mini-USB-Host-Shield-2_0-ADK-SLR-Development-Tool-p-1175308.html [Accessed May 20, 2020].

[99] Insize. *Communication Instruction*.
Available: Received by e-mail. [Accessed April 23, 2020].

[100] *ASCII Table and Description*.
Available: http://www.asciitable.com/[Accessed May 20, 2020].

[101] hathach and toddtreece. *hid_keyboard.ino*. GitHub.
Available: https://github.com/adafruit/Adafruit_nRF52_Arduino/blob/master/libraries/Bluefruit52Lib/examples/Peripheral/hid_keyboard/hid_keyboard.ino [Accessed May 20, 2020].

[102] hathach. *BLEHidAdafruit.h*. GitHub.
Available: https://github.com/adafruit/Adafruit_nRF52_Arduino/blob/master/libraries/Bluefruit52Lib/src/services/BLEHidAdafruit.h [Accessed May 20, 2020].

[103] Unity Technologies. *Event*. Unity Technologies.
Available: https://docs.unity3d.com/ScriptReference/Event.html [Accessed May 20, 2020].

[104] Unity Technologies. *OnGUI*. Unity Technologies.
Available: https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnGUI.html [Accessed May 20, 2020].

[105] *WorldAnchor*. Unity Documentation.
Available: https://docs.unity3d.com/ScriptReference/XR.WSA.WorldAnchor.html [Accessed May 13, 2020].

[106] *MR Basics 101: Complete project with device*. Microsoft.
Available: https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-101 [Accessed May 13, 2020].

[107] *Windows.Media.Capture Namespace*. Microsoft.
Available: https://docs.microsoft.com/en-us/uwp/api/windows.media.capture?view=winrt-19041 [Accessed May 14, 2020].

[108] *Platform capabilities and APIs - Camera and capture*. Microsoft.
Available: https://docs.microsoft.com/en-us/windows/mixed-reality/locatable-camera [Accessed May 13, 2020].

[109] *Scripting API - Matrix4x4*. Unity Documentation.
Available: https://docs.unity3d.com/ScriptReference/Matrix4x4.html [Accessed May 14, 2020].

[110] hexcel.com. *Prepregs Resins*. hexcel.com.
Available: https://www.hexcel.com/Products/Prepregs-and-Resins/ [Accessed May 23, 2020].

[111] clasohlson.com/no. *FlashForge Finder 2.0 3D-Skriver*. clasohlson.com/no. Available: https://www.clasohlson.com/no/FlashForge-Finder-2.0,-3D-skriver/p/38-9154 [Accessed May 23, 2020].

[112] Natasha Gilani. *Properties of Hardened Steel*. sciencing.com. Available: https://sciencing.com/properties-hardened-steel-6301296.html [Accessed May 23, 2020].

[113] Supermagneter.no. *Diskmagnet Ø 30 x 10 mm*. Supermagneter.no. Available: www.supermagneter.no/disk-magnet-o-30-mm-x-10-mm [Accessed May 20, 2020].

Appendices

List of Appendices

Appendix A Document Version Control

Table 4: Document version control

Version	Date	Author	Brief summary of changes
0.1	06.01.20	Alina	Document setup (LaTex)
0.2	07.01.20	Alina	1st draft of **Team Working Agreement**
0.3	Date	Piri	Review of **Team Working Agreement**
0.4	Date	Alina	Update of **Team Working Agreement**
0.5	Date	Markus	Initial **Project description**
0.6	25.01.20	Alina	**Document Version Control** setup
0.7	26.01.20	Alina	Reorganised document structure
0.8	29.01.20	Markus	**Project description** restructured to **KONGSBERG Aerostructures** and **Project Assignment**
0.9	31.01.20	Alina	Reorganised document structure
0.10	31.01.20	Alina	**Project model and development process**
0.11	1.02.20	Alina	**Scrum and Agile Methodology**
0.12	2.02.20	Alina	**Behaviour and Test Driven Development**
0.13	6.02.20	Markus	**Requirements**
0.14	7.02.20	Markus	**Fixtures**
0.15	7.02.20	Tobias	**Risk management**
0.16	8.02.20	Alina	Update of **References**

Version	Date	Author	Brief summary of changes
0.17	8.02.20	Alina	Update of **Project model and development process**
0.18	9.02.20	Alina	**Project Schedule**
0.19	9.02.20	Piri	Update of **Scrum and Agile Methodology**
0.20	9.02.20	Piri	**MR Headset Comparison**
0.21	9.02.20	Alina	Formatting **MR Headset Comparison**
0.22	10.02.20	Ola	**Technology**
0.23	10.02.20	Ola	Fixed **Risk Management Framework**
0.24	10.02.20	Ola, Piri, Alina	Final review
1.0	11.02.20		Ready for the First Revision
1.1	19.02.20	Ola	Set margins between text and page number
1.2	20.02.20	Ola	Update of figures in **Technology**
1.3	21.02.20	Alina	**Team overview**
1.4	08.03.20	Alina	**User Story Mapping**
1.5	09.03.20	Alina	**Microsoft HoloLens**
1.6	16.03.20	Alina	Organized document structure
1.7	18.03.20	Alina	Introduction to **Software Architecture, User Story Mapping and Design**
1.8	18.03.20	Alina	Update of **Behaviour and Test Driven Development**

Version	Date	Author	Brief summary of changes
1.9	18.03.20	Markus	Update of **Fixtures**
1.10	18.03.20	Markus	**Composite Panel Production**
1.11	18.03.20	Piri	**Software Architecture**
1.12	18.03.20	Piri	**User Interface Design**
1.13	19.03.20	Tobias	**Quarantine**
1.14	19.03.20	Tobias	**Collaboration**
1.15	19.03.20	Ola	**Thickness Measurement Technology**
1.16	20.03.20	Ola	Hyperlinks for **Document Version Control**
1.17	21.03.20	Tobias	**Product Release Version 1.0**
1.18	21.03.20	Tobias	Introduction to **Implementation**
1.19	22.03.20	Alina	Final review
2.0	23.03.20		Ready for the Second Revision
2.1	05.04.20	Alina	New Introduction to **Implementation**
2.2	20.04.20	Alina	**Software**
2.3	22.04.20	Alina	**Development Platform, Environment, Framework and Tools**
2.4	23.04.20	Alina	**Object Recognition and Tracking Software**
2.5	25.04.20	Alina	**HoloLens Camera Calibration**
2.6	26.04.20	Alina	Restructured and updated **Product Release Version 1.0**
2.7	26.04.20	Alina	Added images to **Implementation**

Version	Date	Author	Brief summary of changes
2.8	30.04.20	Alina	**Vuforia**
2.9	11.05.20	Markus	**Fixture Modelling Appendix**
2.10	11.05.20	Alina	**Product Release Version 2.0**
2.12	11.05.20	Alina	**Product Release Version 3.0**
2.13	11.05.20	Alina	**QR/Barcode scanning feature**
2.14	13.05.20	Alina	**Overlaying AR content onto physical objects**
2.15	15.05.20	Alina	**HoloLensARToolKit Configurations**
2.16	15.05.20	Alina	**Manual Camera-projection Calibration**
2.17	18.05.20	Alina	Structure **Implementation** and **Review and Future Work**
2.18	19.05.20	Alina	**Computer Vision and Optical Head-mounted Displays**
2.19	19.05.20	Ola	**Communication Between Micrometer and the XRI application**
2.20	19.05.20	Ola	**Micrometer Communication**
2.21	19.05.20	Piri	**Menu Navigation and Voice Commands**
2.22	19.05.20	Tobias	**Documenting Measurements and Data Management**
2.23	19.05.20	Piri	**Indication of Measurement Points on a Digital Model**
2.24	20.05.20	Piri	**The XRI Application**
2.25	20.05.20	Alina	**Abstract**

Version	Date	Author	Brief summary of changes
2.26	22.05.20	Alina	**Introduction**
2.27	23.05.20	Alina	**Conclusion**
2.28	23.05.20	Alina	Formatting of figures & fixing references
2.29	23.05.20	Markus	**Fixture Prototypes**
2.30	23.05.20	Markus	**Fixtures Review**
2.31	24.05.20	Ola	Cleaning references
2.32	24.05.20	Piri	Final Review
3.0	25.05.20	Alina	Final Document

Appendix B Team Overview

Team 8 - 2020 – Extended Reality Inspection

Workspace: Innovasjonsloftet, Room 114

Alina Steshenko
alina.stehshenko@gmail.com
+47 909 92 147
Computer and software engineering
(Embedded Systems)

Responsibilities:
Software design and architecture, development and testing; Jira Master and project management; LaTex documentation; Presentations and graphic design.

Piri Babayev
babayevpiri@hotmail.com
+47 944 72 557

Computer and software engineering
(Embedded Systems)

Responsibilities:
Scrum Master; UI/UX design; Software Development; Documentation revision; Graphic design.

Tobias Mellum
tobias.mellum@gmail.com
+47 908 51 914

Computer and software engineering
(Embedded Systems)

Responsibilities:
Risk management; Software Development, Software architecture, Documentation, Data Management.

Ola Matre
olamatre@gmail.com
+47 472 68 811

Cybernetics and Mechatronics

Responsibilities:
Micrometre-HoloLens communication; Documentation standards; Software development and testing

Markus Liavik Kanton
markuslkanton@gmail.com
+47 478 46 081

Mechanical Engineering

Responsibilities:
Modelling and 3D printing; Project management and time tracking; Budget and accounting; Website; Video editing and presentations.

Intern veileder: **Jose M. M. Ferreira**
jose.ferreira@usn.no

Ekstern sensor og veileder: **Alf Pettersen**
+47 995 90 069
alf.pettersen@kongsberg.com

Appendix C Team Working Agreement

The purpose of this agreement is to develop optimal productivity flow and establish clear objectives, both for the team as a whole and for individual team members. Working disciplines, or "ground rules" in this document are the team's agreement as to how they will work together and communicate, plan, set, and achieve common goals.

C.1 Working Hours and Time Tracking

Core hours in the period of 06.01.20 - 15.03.20	
Monday	10.00 - 15.00
Wednesday	09.00 - 15.00
Friday	10.00 - 15.00
	+ 8 hours outside of core hours
Total:	24 hours

Core hours in the period of 16.03.20 - 02.06.20	
Monday, Tuesday	09.00 - 15.00
Other weekdays	10.00 - 15.00
	+ 13 hours outside of core hours
Total:	40 hours

- Team members shall be present (at the workspace) during core hours unless agreed otherwise.
- Team members shall be punctual and inform the rest of the team as soon as possible if running late.
- If a team member is absent during core hours he/she shall be available via phone, Slack and Google Hangouts if possible.
- Each team member shall track his or her working hours every day. Team members are expected to be truthful about their recorded workhours.

- 30 minutes lunch break and 15 minutes regular breaks are included in the working hours. Should a break exceed the agreed amount of time, it shall be deducted from the tracking of working hours.

C.2 Workspace

Team's primary workspace is room 114 at Innovasjonsloftet. Team's secondary workspace is Innovasjonslabben at Technology Park. The team shall agree on which workspace will be used when during the week.

- Be respectful of the team space.
- Keep it clean.
- Keep the volume low and take disruptive conversations elsewhere.

C.3 Meetings

Following guidelines are for meetings with the client and internal advisor:

- A formal invitation and an agenda shall be sent to all participants by the meeting's chairperson at least two days before the meeting.
- A team member shall take notes at each meeting and send a formal meeting abstract to all participants within 24 hours of the meeting.
- Role as a secretary and as a chairperson shall be rotated between team members. The secretary shall be the chairperson at the next meeting.
- Each team member shall be dressed appropriately for the meetings.
- The team shall be focused during the meetings and stay off devices unless there is an emergency.

C.4 Presentations

- Documentation shall be sent to internal and external sensors, internal and external advisors minimum two working days before the presentation.
- Each team member shall be dressed appropriately, be prepared and show up at an appropriate time before presentations.

C.5 Team Norms

Shared responsibility - each team member has an equal voice. In case of absence a team member shall trust his/her teammates' collective decision.

Commitment - work together and support each other in our common goals. Don't be afraid to ask for help.

Quality and performance - strive to do our best to achieve the goals and milestones set by the team and to deliver our best work to solve the task set by the client.

Communication - to the best of our ability and without unreasonable delays. The team shall try to resolve any disagreements and disputes within the team before involving a third party. No hidden agendas.

Open feedback - be willing to provide and accept constructive criticism on methods, core, processes etc.

Appendix D Kick-off Meeting Abstract

Meeting with the client (Team No 8)

Title	Kick-off
Date	24.01.2020
Time	10:00 – 12:00
Location	KONGSBERG Innovation Center
Chairperson	Markus Kanton
Notetaker	Alina Steshenko
Attendees	**Team members:** Alina Steshenko, Piri Babayev, Ola Matre, Tobias Mellum, Markus Kanton **Company representatives:** Alf Pettersen (the client), Espen Janshaug (operator), Åse Kleven (contact person at Innovasjonslabben) **Observers:** Jose M. M. Ferreira (team's internal advisor from USN, via Skype)
Agenda	1. Introduction 2. Status update on the team's work so far (Alina) 3. Project description, stakeholder and user interviews 4. Project expectations (Piri) 5. HoloLens status update (Åse) 6. Contract signing 7. Other matters
Summary	1. **Short introduction around the table.** 2. **Alina updates on the team's work:** ◦ Week 2: Intensive lecture week with info from USN. ◦ Official project start at 13th of January

- 24 working hours a week until Easter exams, full 40 working hours a week after that.

Two administrative sprints so far with focus on project planning:

- Established routines and formed a Team Working Agreement
- Templates: documentation, time tracking etc.
- Project management tools
- Project model: A hybrid model with Agile approach incl. several Scrum elements
- Established Risk management routines and documentation

Working on:

- Setting the project schedule and milestones
- Technology research
- Establishing Product development process
- Testing, Verification and Validation Strategy

3. **Project description, stakeholder and user interviews:**
An inspector takes thickness measurements in various places of different outer panels of a plane. This is done to ensure that aeroplane's dynamics are not affected. For example, when two panels with different thickness are assembled next to each other it creates an unwanted edge that would affect aerodynamics of the plane.

About the panels:

- Size ranges from 30 centimetres up to 2 meters.
- Smooth green upside, irregular white inside.
- Around 40 different types of panels.

About the working station:

- The panels are stored on shelfs in the working area prior to inspection.
- A panel to be inspected is placed on a table.

- A computer is placed two meters away from the table.

About the current inspection process:

- Inspector prints out a document for every panel that needs to be measured. The document contains a drawing of the panel, indicators of where on the panel thickness should be measured and the measurement tolerance for every given point.
- Number of measurement points on a panel vary from 5 to 100.
- The thickness is measured by the inspector manually with a micrometre. It is in inches and has 3 decimal place accuracy.
- The inspector writes down the result of every measurement on the printed-out drawing.
- After taking a measurement of all indicated point of the panel the inspector logs in to SAP from the computer placed at the working area, and manually types the results written on the drawing into the system. The drawing is then discarded.

Client's wishes on improving the inspection process:

- Improved efficiency and reduced error occurrence:
    - A better way of guiding the inspector on where to measure than the need of printing out a drawing for every panel.
    - A better way of recording the results from the measurements than writing them down on a drawing and then manually typing them into the system.
- Better documentation:
    - For example, a photo of where the measurement was taken.

Client's wishes in general:

- Evaluation of practical use of HoloLens in collaboration with operators
- User Target Group should be of all ages and with different background, even though there are only two inspectors now.

- The product should be HoloLens version independent

Comments:

- For security reasons, WiFi cannot be used during the inspection control process. Bluetooth is acceptable. Emil Moholt can be contacted for other questions regarding IT Security.

4. **Piri conducted a Project expectations exercise.**

Results:

The team
- Help get Hololens 2
- Honest feedback
- Help us test the product
- Good Communication between all the parties
- Keep in mind that it is a student project
- A solution that both the client and the inspectors are happy with

Espen
- User-friendly
- Efficient
- Quality assurance

Åse
- Keep the KIC clean after usage
- Always ready to help you out

Alf
- Solution to guide operator to measure at right location
- Solution to document measurement - thickness data and location
- Evaluation of practical use of Hololens in collaboration with operators

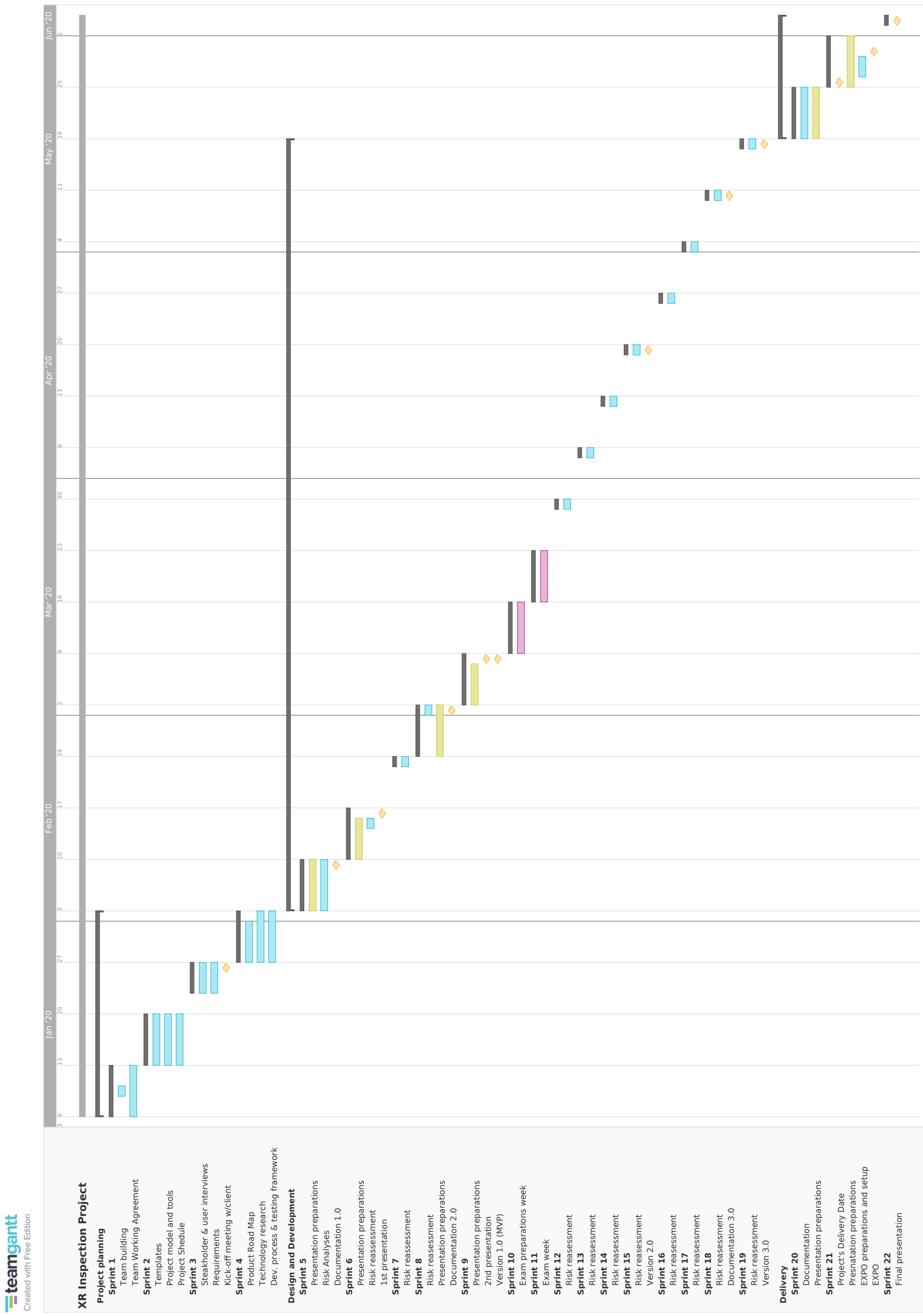
Jose
- Good and specific requirements from the client

5. **Åse informs about HoloLens 2.**
   Unfortunately, the companies contact person in USA still has not ordered HoloLens 2. We are hoping to borrow a HoloLens 2 from TechnipFMC that has ordered 6 unites that should arrive in next couple of weeks. More information will be available in week number 6.

6. **Contract is signed by the team members and the client.**
   Markus will obtain remanding signature from USN.

7. **Other matters:**
   The team shall send a status update document on a weekly basis. Summary of this meeting shall be sent to all participants. The team shall start on formalising requirements and user stories as soon as this summary is approved by the client.

Appendix E Project Schedule



Appendix F Risk Management Framework

Risk Matrix

Probability	Severity		
	HIGH	HIGH	HIGH
	MED	MED	MED
	LOW	LOW	LOW

Explanations:
HIGH - Most likely going to happen.
MED - Both likely to happen and not happen.
LOW - Most likely not going to happen.

RED - The product may become obsolete or unusable.
YELLOW - Significant damage to the product.
GREEN - Barely noticeable on the end product.

ID	Event	Source	Asset	Consequences	Risk product	Strategy
	Project risks:	Risks associated with the success of our project that is not covered in the other groups.				
R1	Risky event	What is the source of the risk	What asset is affected	Consequences of the event		
R1.1	Lack of communication	Group members, faculty staff, client.	Product, project.	Tasks are not finished in time. Decreased productivity.	LOW	Pre Risk: Use of Slack as the official communication channel for the team. Divide topics into different channels in Slack. Post Risk: Be strict and frequent in sharing communication issues.
R1.2	Unfinished tasks	Group members not prioritizing the project. Unrealistic tasks, spending to much time on writing tests.	Product , grade.	Tasks are not finished in time. Decreased productivity.	HIGH	Pre Risk: Delegate less urgent tasks to the members. Post Risk: Converse why tasks where not finnished.

R1.3	Failure to follow project methodology.		Product, grade.	Progression might suffer and we wont get to test the methodology fully.	LOW	Pre Risk: Have a clear definition of our project methodology and communicate uncertainties Post Risk: Set a meeting where everyone share their understanding of the methodology.
R1.4	Team members forgetting their parts while presenting	stress/anxiety, lack of preparation.	grade		N/A	Pre risk: Always have 2 people knowing each part of the presentation, start rehearsals in adequate time before presentations.
R1.5	Team members cant work as excpected because of the present virus.	Quarantine, Illness, loss of close relations	Product	Decreased productivity, higher workload on team members.	MED	Pre risk: Set of alternative routines. Post risk: Use tools to cooperate online and communicate issuses and solutions.
R1.6	Equipment unavailable	Quarantine, closed shops, unavailable key people.	Product	Some tasks might not be solvable.	MED	Pre risk: Acquire all necessary equipment for the project as soon as possible. Post risk: Work on other tasks/features, consider changing the product.
R1.7	3d printer failing.	Either user error or system error	Fixtures	Fixures not ready for presentation.	LOW	Pre risk: Clean printer head from excess plastic, change old mechanical components. Post Risk: Outsource printing to Richard

	Product risks:	Risks assosiated with how the product works				
ID	**Event**	**Source**	**Asset**	**Consequence**	**Risk product**	**Strategy**
R2						
R2.1	Product does not meet the client's needs	Insufficient understanding of needs and the process to fulfill the clients needs	product, project	Unsatisfied customer	MED	Pre Risk: Frequently discuss customer needs with customer. Start testing early. Post Risk: Consider mitigation options

R2.2	Requirements not met	Bad project methodology or methodology not used correctly.	product, project	Product might be useless to our customer	LOW	Pre Risk: Revisit requirements frequently. Use good test methods. Start testing at an early stage.
R2.3	Low performing application	Our teams' technical skills	Product	Product might be useless, operators find it inconvenient to use.	LOW	Pre Risk: Use available resources to unity and hololens like steven and official guidelines. Post Risk: Investigate issues, restucture and optimize.
R2.4	Technology overengineered with redundant functionality.	Not implementing our project methodology strict. Not tracing requirements and needs to functionality.	product	Probably decreased performance of our product.	LOW	Pre Risk: Trace requirements and wants to every functionality and subsystem Post Risk: Consider removing redundant functionality
R2.5	Operators finding the user interface too complicated to use.	Product does not fit operators' technical skills.	product	Some operators might reject using our product.	LOW	Pre Risk: Maintain dialogue with operators. Arrange user tests. Post Risk: Redesign user interface.
R2.6	Developers writing bad unit tests	Lack of experience with tdd and restricted time to learn.	Product	Units might not get automated tests.	HIGH	Pre risk: The entire team should verify in Jira that they have finished studying unit test
R2.7	Insufficient photo documentation	Short time left to integrate the functionality and test alternative solutions.	Product	Not all measurements will have good photos	MED	Solution: Make pictures available to the operator as he measures.

	Technical:	Risks directly assosiated with hardware or software				
ID	**Event**	**Source**	**Asset**	**Consequence**	**Risk product**	**Strategy**
R3						
R3.1	Our implementation requires more storage than available on the hololens	Using some quantity of high quality images for object recognition.	Our implementation and its functionality	Object recognition less precise or non existing.	LOW	Pre Risk: We get approx 54 Gb after the OS. Make an assessment on how much we need for viable OR. Optimalization techiques regarding graphics
R3.2	Fps drops	Resource heavy application, e.g object recognition.	Our implementation.	Weakened user experience	MED	Pre Risk: Look for areas that can later be optimized Post Risk: Optimize resource demanding parts of the application
R3.3	Application crash regularly	Lack of knowledge with hololens, not applying proven methods, insufficient testing.	Software	Weakened user experience, inefficient work process.	MED	Pre Risk: Use officail guidelines to development aswell as local resources like Steven Bos. We are using unit testing. Post Risk: Investigate and debug system.
R3.4	Bad choices of external libraries/technology.	Lacking knowledge and experience with the tools and technology	Our implementation and product.	Suboptimal solution.	LOW	Pre Risk: Using the available resources and local knowledge. Post Risk:Consider to change technology.
R3.5	Implementation is not scalable and maintainable.	Choice of development techniques.	Continuation of development for the team and later projects for the client.	Product can not easily be used or continued by the client.	LOW	Pre Risk: Use proven methods for scalable applications, we are using unit testing.
R3.6	Technology not fit for security concerns. (hololens or externally developed technology)	Dependancies in applied libraries and hololens communication .	Our implementation and project.	The client can not use the intended technology	LOW	Pre Risk: Communicate with the clients security officer.

R3.7	Hardware components malfunctioning	Production faults, user errors, wear.	Project, product	Project progression might suffer.	MED	Pre risk: order duplicate components. Post Risk: Order new components

	External risks:	Risks that our team can not prevent.				
ID	**Event**	**Source**	**Asset**	**Consequence**	**Risk product**	**Strategy**
R4						
R4.1	Innovation within AR technology may leave our product redundant	Innovation of new technology	Our product	Our product might be left redundant by more potent core technology	LOW	
R4.2	Hololens 2 device not available for development	Delayed delivery	Testing and development	Some requirements might be harder to test without the device. The emulator might not reveal all issues that the device does.	N/A	
R4.3	Regulatory changes in the organization.	NSM, Kongsberg group, change in politics.	Requirements to security.	Reassessment of functionality and choice of technology	MED	Pre Risk: Communicate with clients security managers. Post Risk: Look for option that are in line with new regulations.

Highlighted risks for the final phase of the project:
Insufficient time for integration of software,
application crashes,
insufficient documentation,
operators not fond of the user experience with the prototype of the micrometer extension,
github merges.

Appendix G Composite Part Production

This appendix will in some level of detail elaborate on how the composite parts for the project has been made. For a more visual showcase of the process a video can be found on www.XRI.no/production For more information about the material at use: https://www.hexcel.com/Resources/DataSheets/Prepreg and download PDF for 8552 - Mid-Toughened, High Strength, Damage-Resistant, Structural Epoxy Matrix.

- **Step 1 - Unfreeze material**

For panel production a *pre-impregnated* (prepreg) carbon fiber weave mat with Hexcel's epoxy matrix HexPly 8552 is used. The mat is a living product which, in this case, means that it has a quality expiration date for how long it can be in room temperature, and how long it can be frozen before curing. Thus, to extend the time where the material is usable, the mat is kept frozen at -30°. Therefore, before opening the protecting bag the material defrost for a minimum of 12 hours.

- **Step 2 - Release Tool Surface**

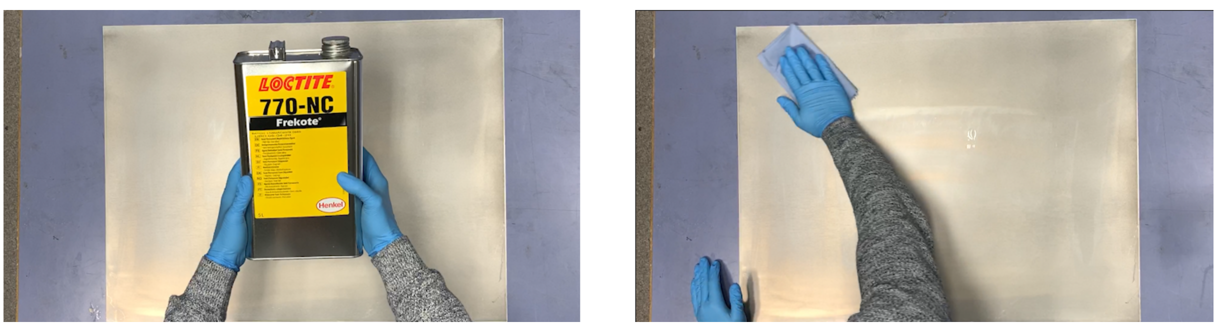


Figure 96: Releasing tool surface with 770-NC

To ensure that the resin doesn't bond the carbon fiber panel with the tool the release coating 770-NC Freekote (as seen in Fig.97) is used, which creates membrane at the tool surface and ensures the release of the finished product after curing.

Table 5: The formula for releasing the tool surface.

Release surface	**Time interval**
1	5 min
2	5 min
3	5 min
4	5 min
5	15 min

The formula used for releasing the tool surface can be seen in Tab.5.

- **Step 3 - Cut Plies**

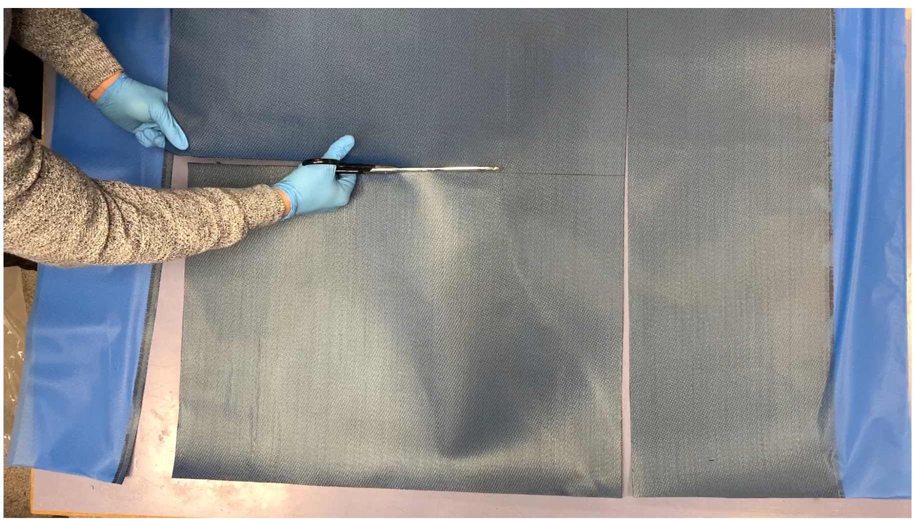


Figure 97: Cutting plies with scissors for carbon fiber

To create the end product it is necessary to cut out the plies in the orientation required, e.g. what angle the weave is located in. Usually in a production process for this kind of advanced material, this is done with a very precise cutting system. Since the quality of the product and the precision of the plies orientation is not that critical for this project it is sufficient to do it by hand, as demonstrated in Fig.97.

- **Step 4 - Lay-up**

When manufacturing composite panels it is important to keep the ply orientation structure of the panel symmetrical to eliminate internal stresses. To ensure this parts are produced using a lay up demonstrated in Fig.98.

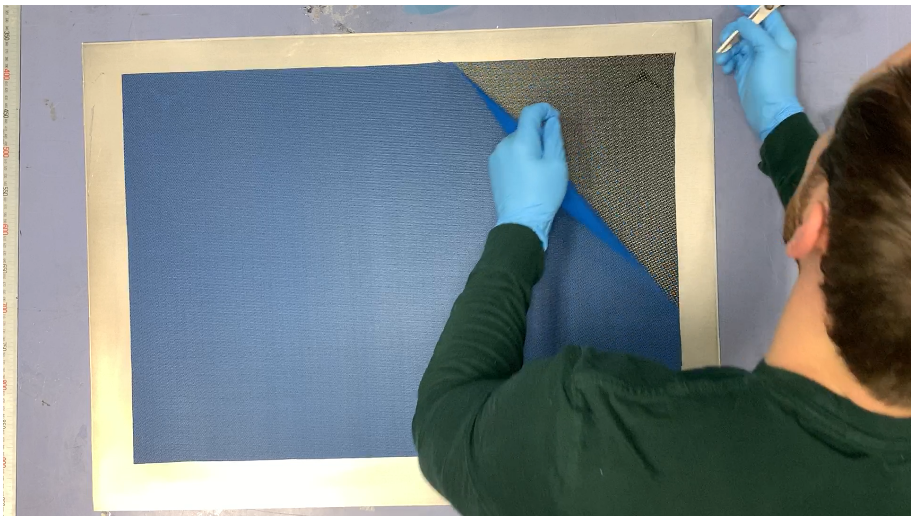


Figure 98: Lay up of part

The lay up follows the steps seen in Tab.6

Table 6: The steps of lay up.

Lay up step	**Ply orientation**
1	90°
2	+45°
3	-45°
4	0°
5	0°
6	-45°
7	90°
8	15 min

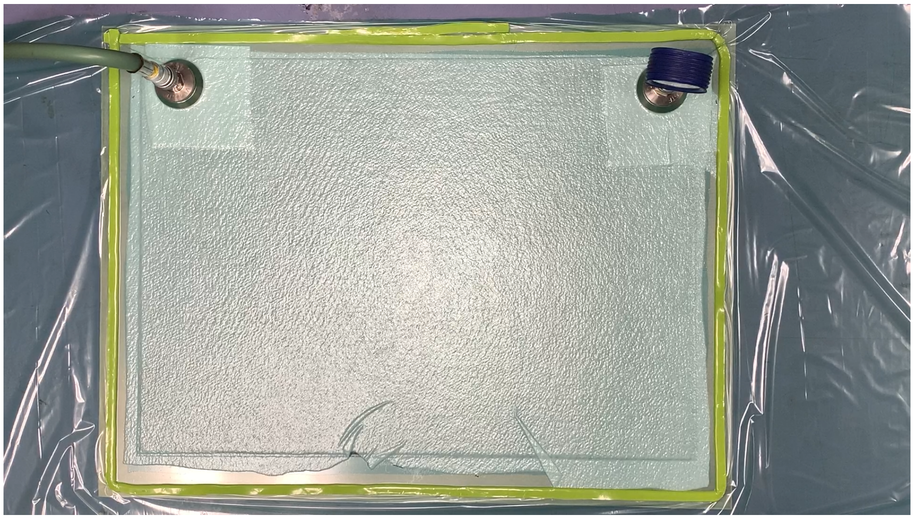


Figure 99: Debulking composite part

To enhance the bonding and remove potential cavities between plies during the lay up process debulking (vacuum bagging) is executed, as demonstrated in Fig.99, using the time intervals stated in Tab.7.

Table 7: Time intervals for debulking.

After ply	**Time interval**
1	5 min
5	15 min
8	15 min

- **Step 5 - Bagging**

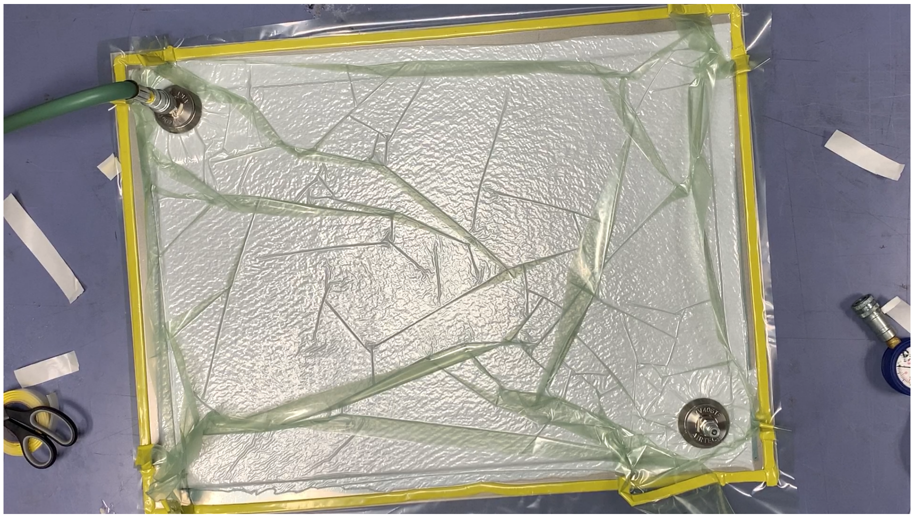


Figure 100: Finished curing bag

Since the product needs to be cured at high temperatures, and external pressure, more durable materials are required than what is the case for the debulk operations. This material is more expensive and therefore used only for curing. The finished curing bag can be seen in Fig.100

- **Step 6 - Curing**

Figure 101: Autoclave at Krag Technology Lab

The resin in prepreg carbon fiber mats always cure better at elevated temperatures and pressures. The optimal cure formula varies for different types of resin. The autoclave in Krag Composite Lab (as seen in Fig.101) gives the group the ability to cure the panels with similar quality as the ones made at KDA. For the Hexcel 8552 Hexply curing is done following the steps in Tab.8

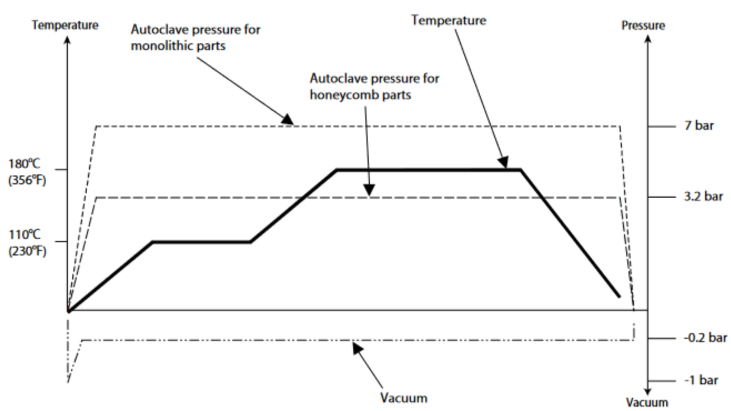


Figure 102: Curing cycle for honeycomb and monolithic components

Table 8: Curing for Hexcel 8552 Hexply.

Curing step	**Temperature [°C]**	**Vacuum [bar]**	**Pressure [bar]**	**Time [min]**
Start	20	-0,90	0	0
1	120	-0,20 (when pressure is 1 bar)	5	60
2	120	-0,20	5	60
3	180	-0,20	5	30
4	180	-0,20	5	120
5	20	-0,20	0	< 32
Stop	20	0	0	0

Figure 103: Demoulding composite panel

- **Step 7 - Demould**

After the cure cycle is complete we use protective gloves to remove bagging material (as seen in Fig.103) and to burr the sharp edges of the panel.

- **Step 8 - Machining**

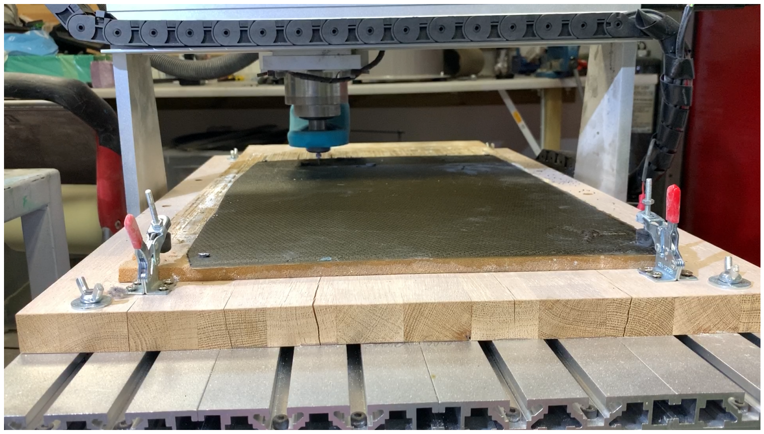


Figure 104: Machining carbon fiber parts

We used a two axis milling machine (as seen in Fig.104) to mill out different parts (as seen in Fig.105) with similar attributes for testing purposes.

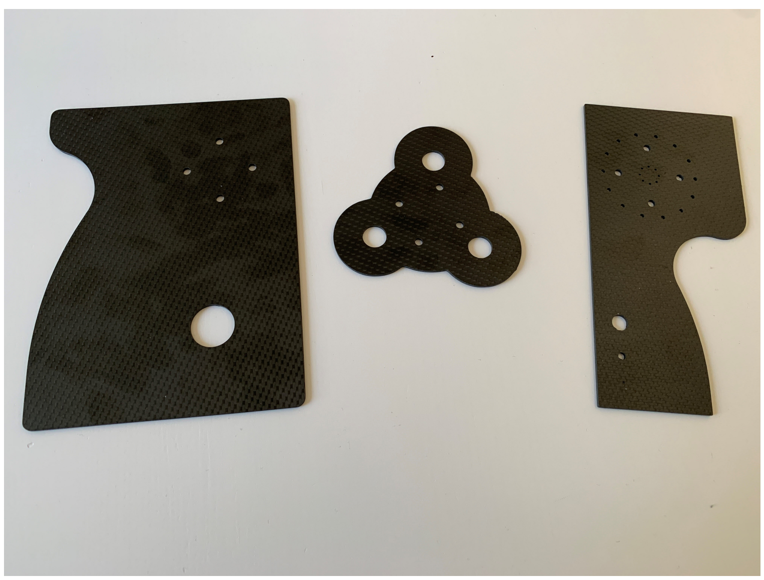


Figure 105: Finished carbon fiber parts

Appendix H MR Headset Comparison

Table 9: MR headset comparison

	Hololens 1	**Hololens 2**	**Magic Leap**	**Nreal**
About	First generation of Hololens headset is the first try in conquering the augmented reality horizons – it features a standalone PC integrated into the headset and allows users to interact with projected graphics using their hands.	The new generation of Hololens is a massive improvement with better ergonomics and calculating power.	Instead of mounting a small dedicated computer on the headset Magic Leap offers a separate clip-on solution that is cabled with the glasses. It also includes a separate hand controller, similar to the ones used in VR headsets.	The lightest solution is Beijing-based start-up that offers you to tether your glasses to an Android smartphone or a PC and all computing is done off-device.

	Hololens 1	**Hololens 2**	**Magic Leap**	**Nreal**
CPU	Intel 32-bit x84 architecture with TPM 2.0 support (1GHz).	Snapdragon 850 - 64-bit ARMv8 (2.9 GHz)	2x Denver 2.0 - 64-bit 4x ARM Cortex A57 64-bit Together up to 1.7GHz. 2x A57s and 1x Denver accessible to applications.	Qualcomm Snapdragon 845 – 64-bit ARM LTE (1.7GHz).
HPU/GPU	HPU 1.0	HPU 2.0	GPU Nvidia Parker SOC – 64-bit ARM	Depends on connected Android or PC device
Main Memory	2 GB RAM	4-GB LPDDR4x system DRAM	8GM (4GB available to apps)	Depends on connected Android or PC device
Storage	64 GB Flash	64-GB UFS 2.1	128GB (95GB available for apps)	Depends on connected Android or PC device

	Hololens 1	**Hololens 2**	**Magic Leap**	**Nreal**
Field of View	34 degrees	52 degrees	50 degrees	52 degrees
Camera	2MP photo / HD video camera	8MP, 1080p video	No information available	No information available
Wire Connection	Micro USB 2.0	USB-C with USB-PD (Power Delivery) tech	USB-C	USB-C
Wireless	• Wi-Fi 802.11ac; • Bluetooth 4.1 LE.	• Wi-Fi 802.11ac; • Bluetooth 5.	• Wi-Fi 802.11ac; • Bluetooth 4.2	No information available
Weight	566g	579g	316g + 415 g for clip on computer	88g + 23g controller + 170g computing unit
Battery life	2-3 hours	2-3 hours	Up to 3.5 hours	No information available

	Hololens 1	**Hololens 2**	**Magic Leap**	**Nreal**
Other Specifications	• Gaze tracking; • Gesture input; • Voice support	• Eye-tracking; • Gesture input; • Head tracking; • Voice support – command and control on device, natural language with internet connectivity.	• Eye-tracking; • Gesture input;	• Eye-tracking; • SLAM (Simultaneous Localization and Mapping); • 6DoF (Degrees of Freedom) tracking; • Plane detection; • Image tracking;
Development Platform	• Unity; • Unreal; • WebVR; • Native;	• Unity; • Unreal; • WebVR; • Native;	• Unity; • Unreal Engine; • Lumin; • Lumin Web Platfrom;	• NRSDK; • Unity; • Unreal Engine; • Android;

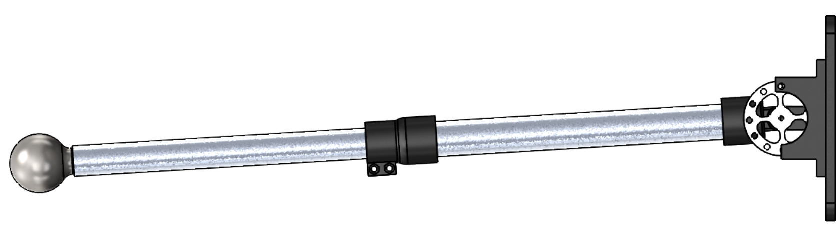
	Hololens 1	**Hololens 2**	**Magic Leap**	**Nreal**
Accessibility	Users can use glasses with the headset	Users can use glasses with the headset	Users cannot use glasses with the headset (prescription lenses are sold separately).	Users cannot use glasses with the headset (prescription lenses are sold separately).
Market Price	USD 5,000	USD 3,500	USD 2,295	Pre-order for USD 499 - Available for sale in early 2020 Pre-order USD 1199 - developer Kit
Producer	Microsoft	Microsoft	Magic Leap	Hangzhou Tairuo Technology Co.

	Hololens 1	Hololens 2	Magic Leap	Nreal
Comments	Not officially sold by Microsoft anymore.	The only device with Bluetooth 5.0 which has a much higher range and bandwidth compared to Bluetooth 4.1	The headset must be connected to a Lightpack, a clip-on ellipse-shaped computer, where all the processing power is. Includes a separate controller used to interact with projected Mixed Reality.	The company is based in Beijing, China, which potentially can be a security alert for our customer. Requires a separate Android mobile device to perform all computations on. The same Android device along with a separate controller is used to interact with the augmented reality.

	Hololens 1	Hololens 2	Magic Leap	Nreal
Pros	• Does not require any peripheral devices; • Existing online material (tutorials, manuals, examples, etc.);	• Supports Bluetooth 5.0; • Does not require any peripheral devices; • Computing Power;	• Lightweight;	• Lightweight; • Cheap;
Cons	• Heavy; • Poor FOV; • Requires user to get used;		• No gesture recognition; • Accessibility; • Sold only in six cities in the world so far; • A separate computer needs to be carried around;	
Source	Link	Link	Link1 Link2	Link

Appendix I Ball Point Fixture Schematics - Version 1

ITEM NO.	PART NUMBER	QTY.
1	bottom-plate	1
2	rotation-joint	1
3	wheel-joint	1
4	wheel	1
5	rotation-pin	1
6	rotation-bracket	1
7	bracket-1	1
8	pole-mount	1
9	aluminium-pole-OD30-ID26-L300	1
10	pole-clamp	1
11	aluminium-pole-OD26-ID18-L300	1
12	steel-ball-plug	1



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:

FINISH: -

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION -

	NAME	SIGNATURE	DATE
DRAWN	Markus Ianton		08.05.20
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: -

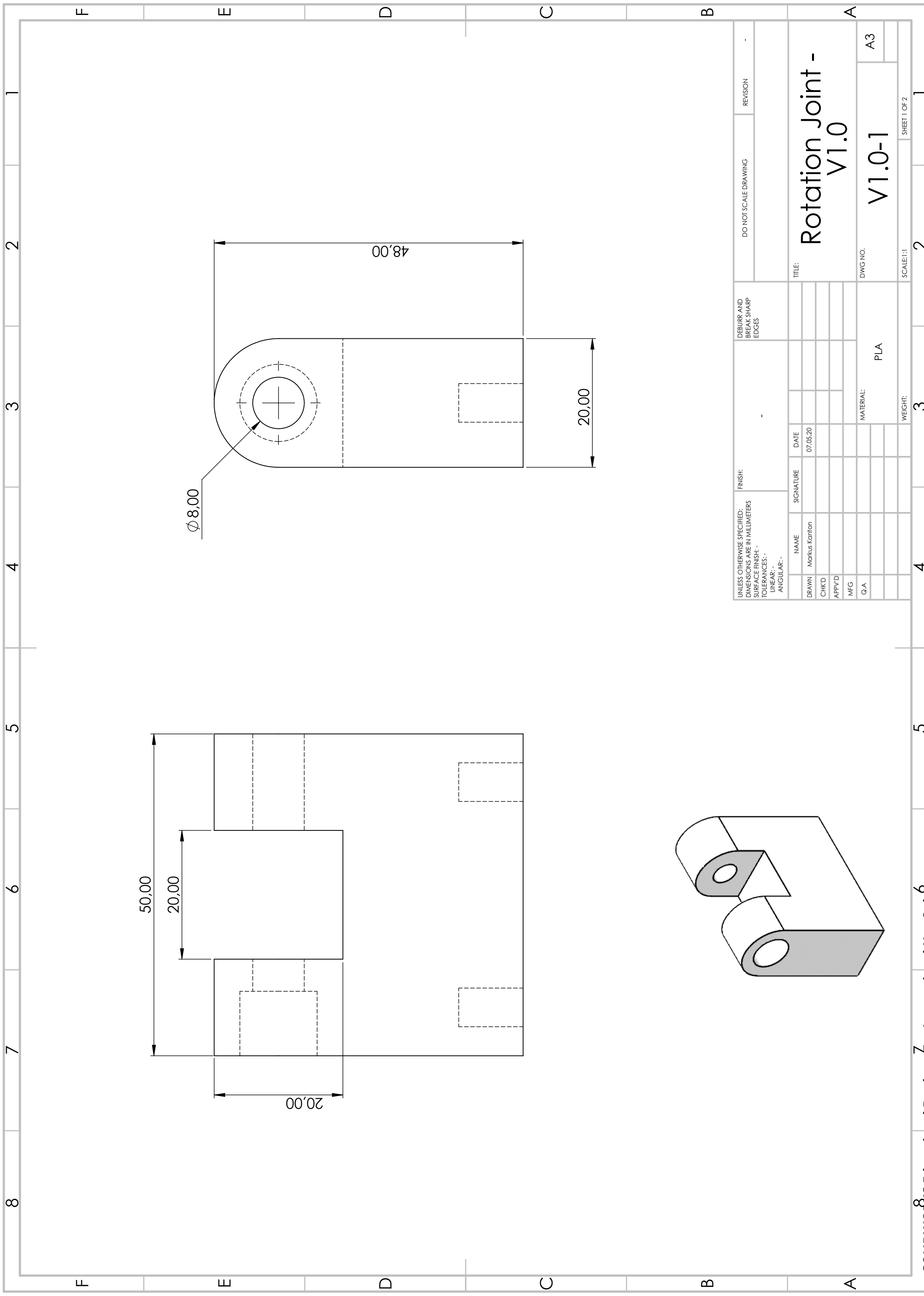
WEIGHT:

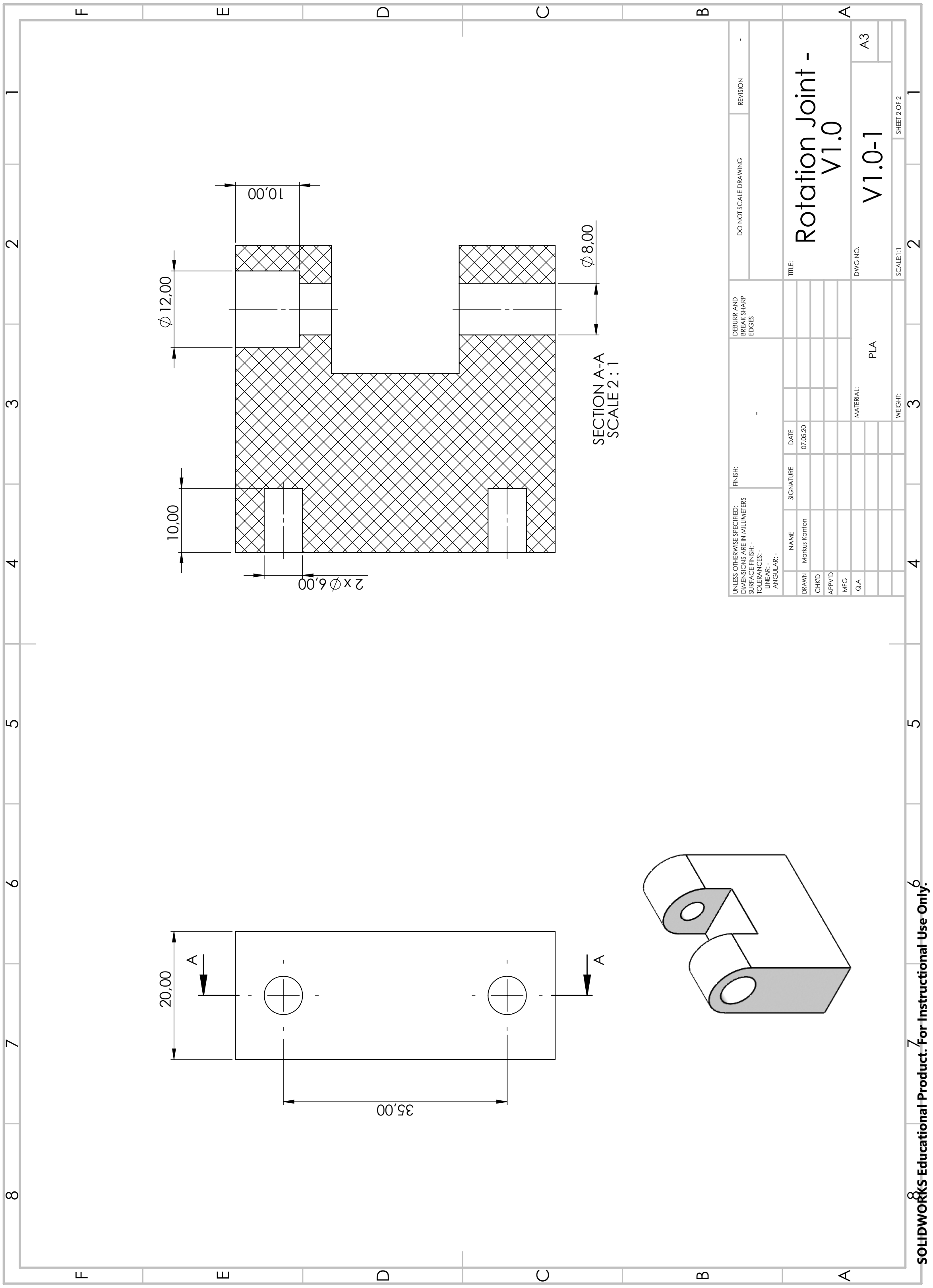
TITLE: Ball Fixture

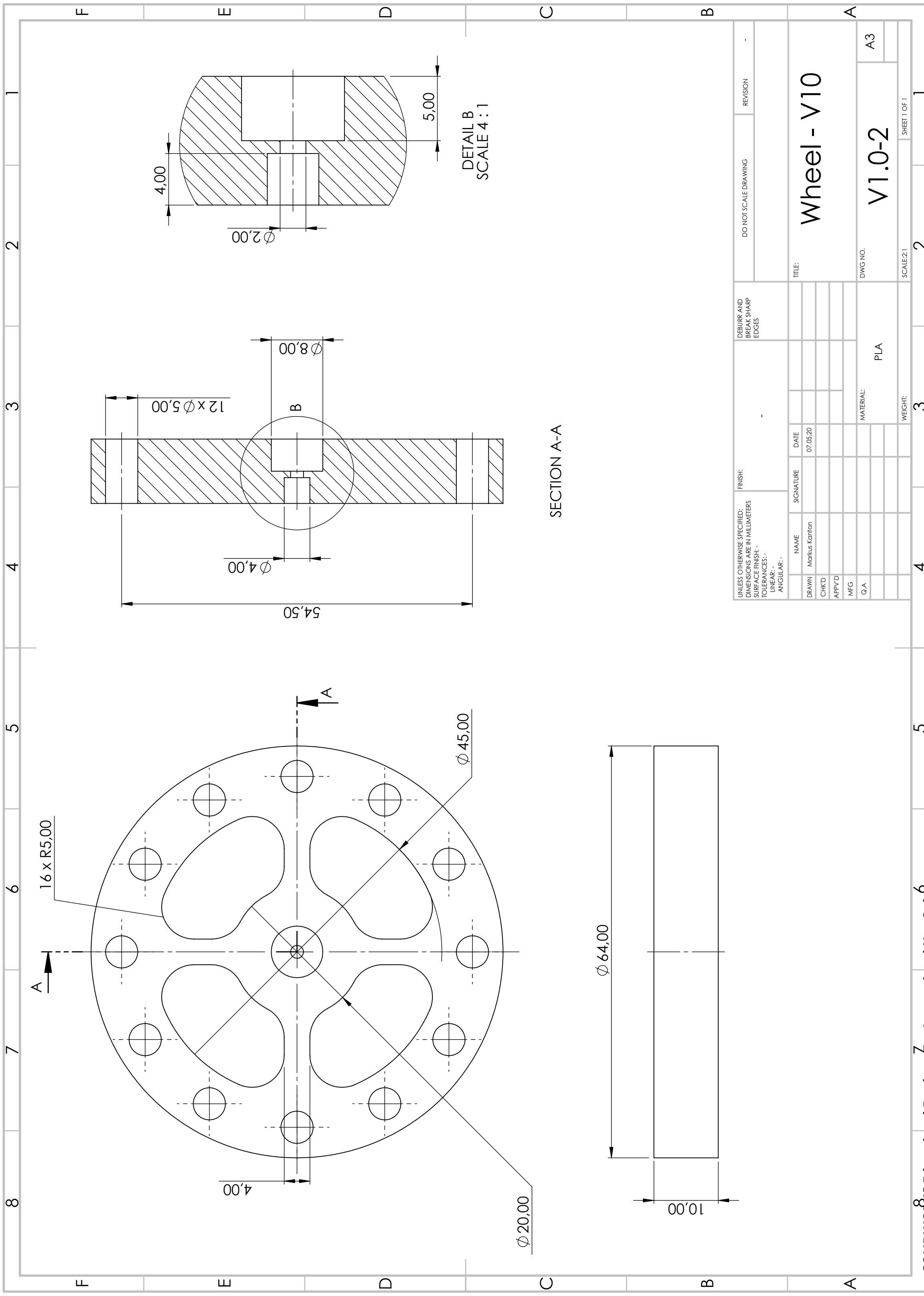
DWG NO. V1.0-ASSE

A3

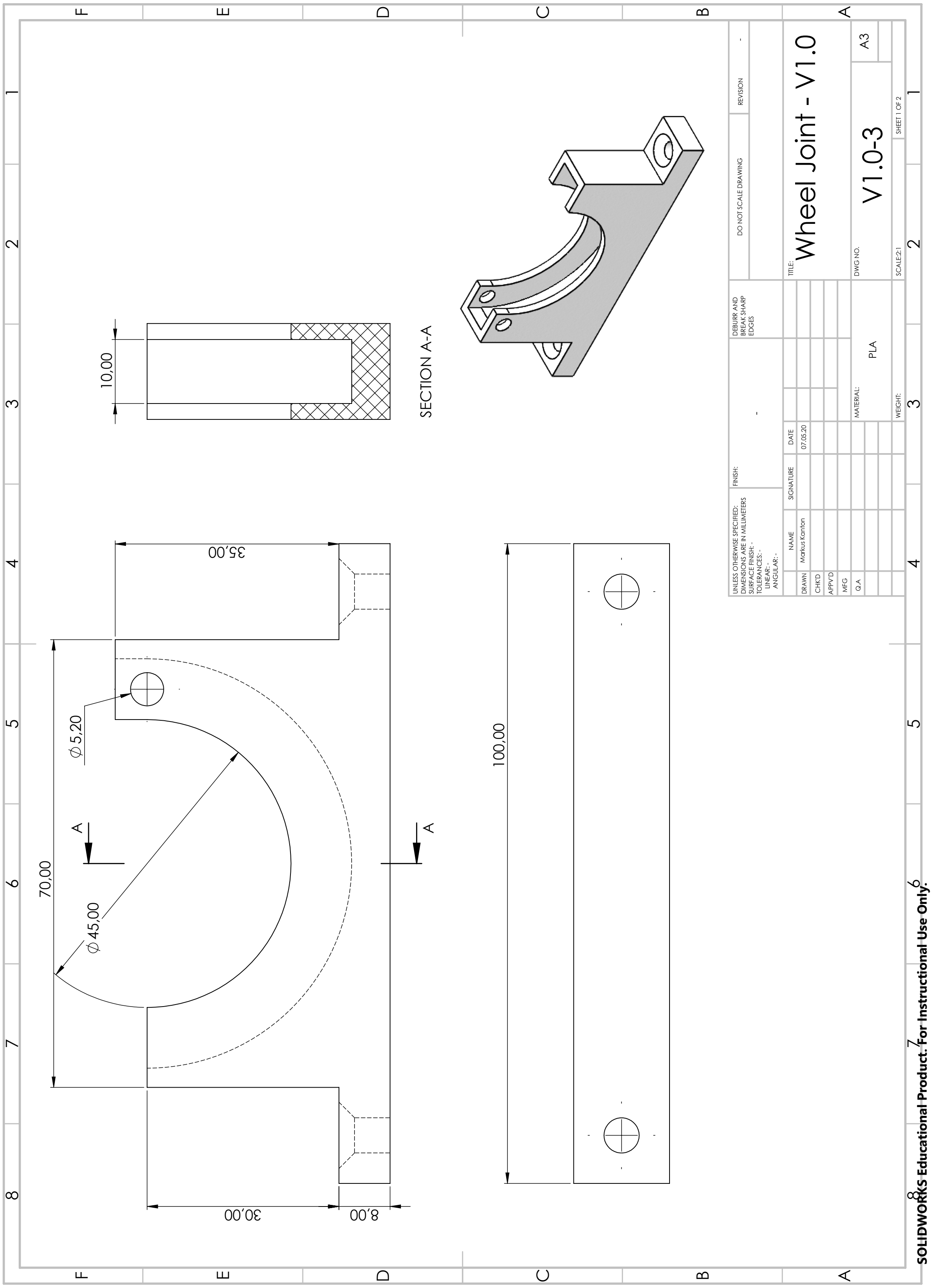
SCALE:1:5 SHEET 1 OF 1

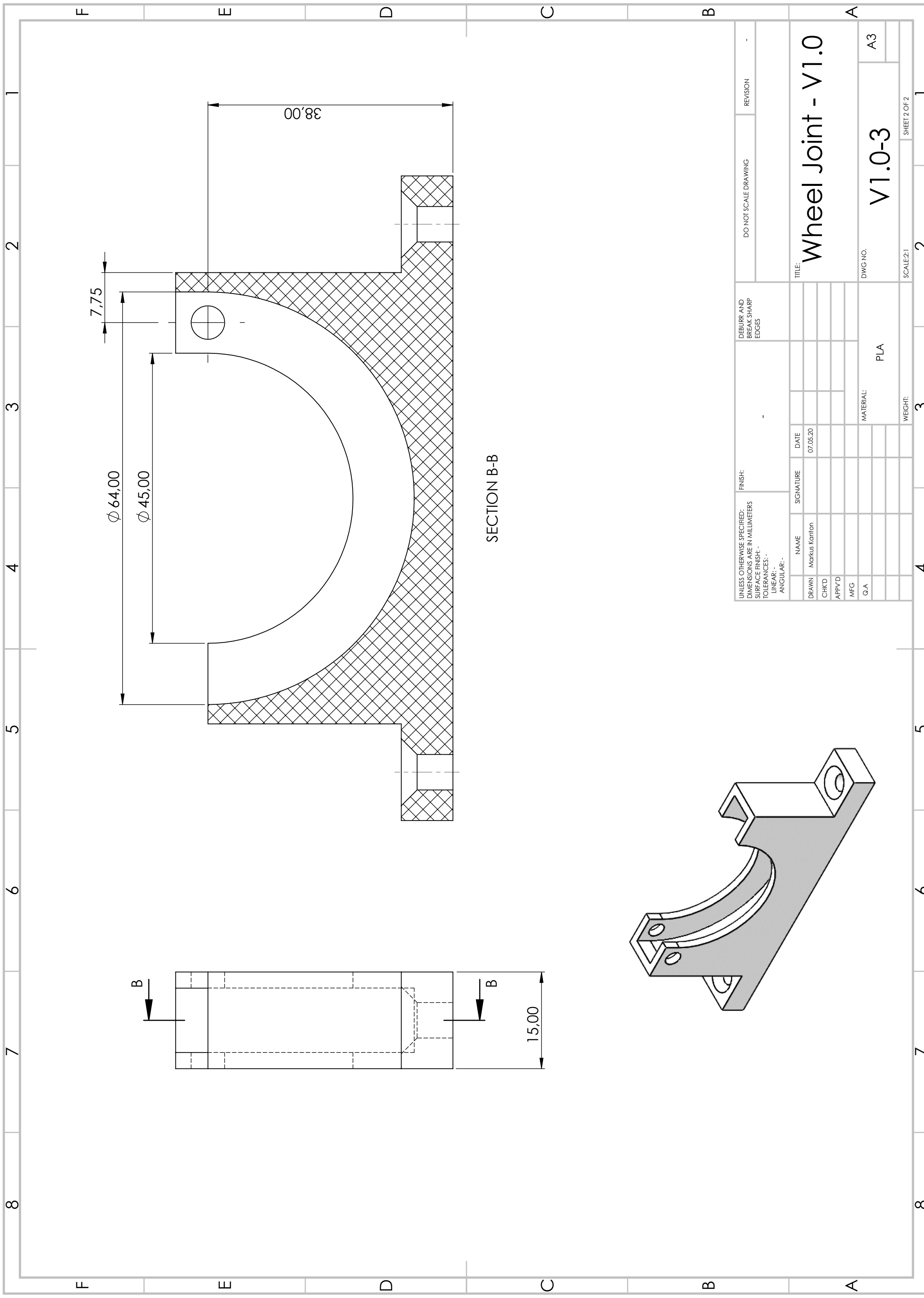

48,00
20,00
⌀8,00
50,00
20,00
20,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: -
TOLERANCES: -
LINEAR: -
ANGULAR: -
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
07.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
PLA
WEIGHT:
TITLE:
Rotation Joint - V1.0
DWG NO.
V1.0-1
A3
SCALE:1:1
SHEET 1 OF 2
SOLIDWORKS Educational Product. For Instructional Use Only.

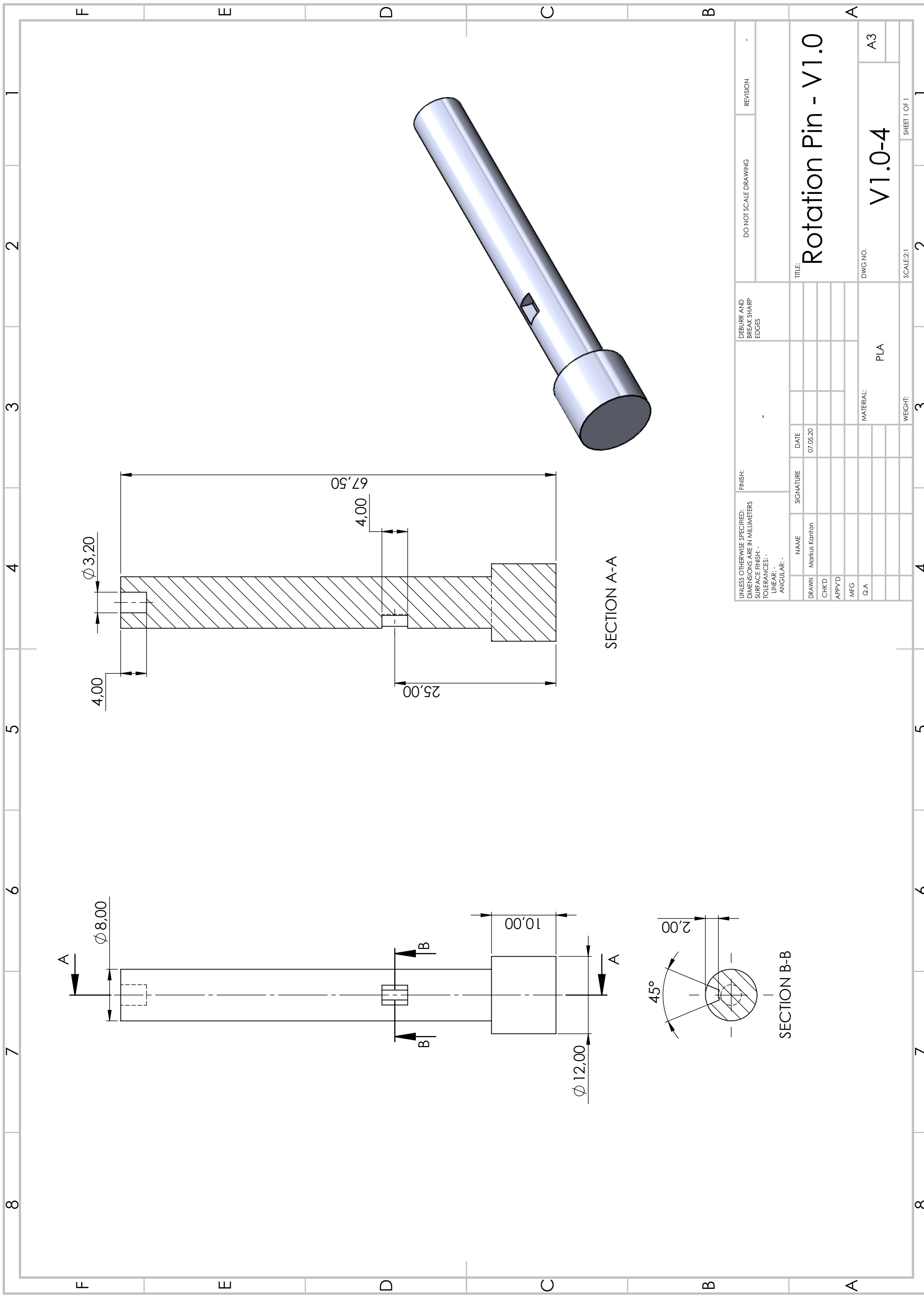

SECTION A-A
SCALE 2 : 1
10,00
Ø 12,00
Ø 8,00
2 x Ø 6,00
20,00
35,00
A
Rotation Joint - V1.0
V1.0-1
A3
PLA
Markus Kanton
07.05.20
DO NOT SCALE DRAWING
SOLIDWORKS Educational Product. For Instructional Use Only.

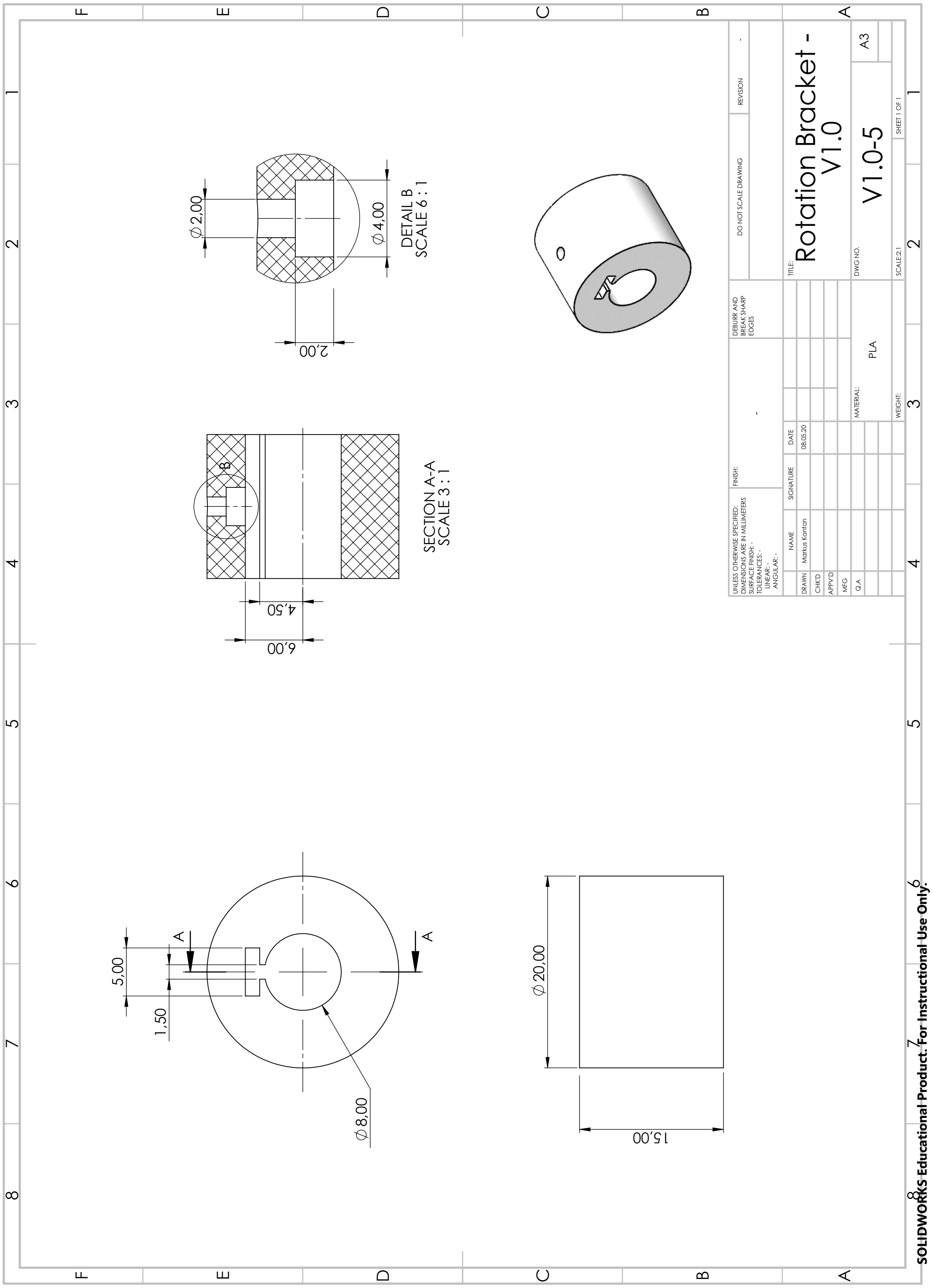


Wheel - V10
V1.0-2
DWG NO.
TITLE:
A3
SCALE:2:1
SHEET 1 OF 1
WEIGHT:
MATERIAL:
PLA
REVISION
-
DO NOT SCALE DRAWING
DEBURR AND BREAK SHARP EDGES
FINISH:
-
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: -
TOLERANCES: -
LINEAR: -
ANGULAR: -
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
07.05.20
CHK'D
APPV'D
MFG
Q.A
SECTION A-A
DETAIL B
SCALE 4 : 1
5,00
4,00
⌀2,00
⌀8,00
12 x ⌀5,00
⌀4,00
54,50
B
A
A
⌀45,00
16 x R5,00
⌀64,00
4,00
⌀20,00
10,00
SOLIDWORKS Educational Product. For Instructional Use Only.

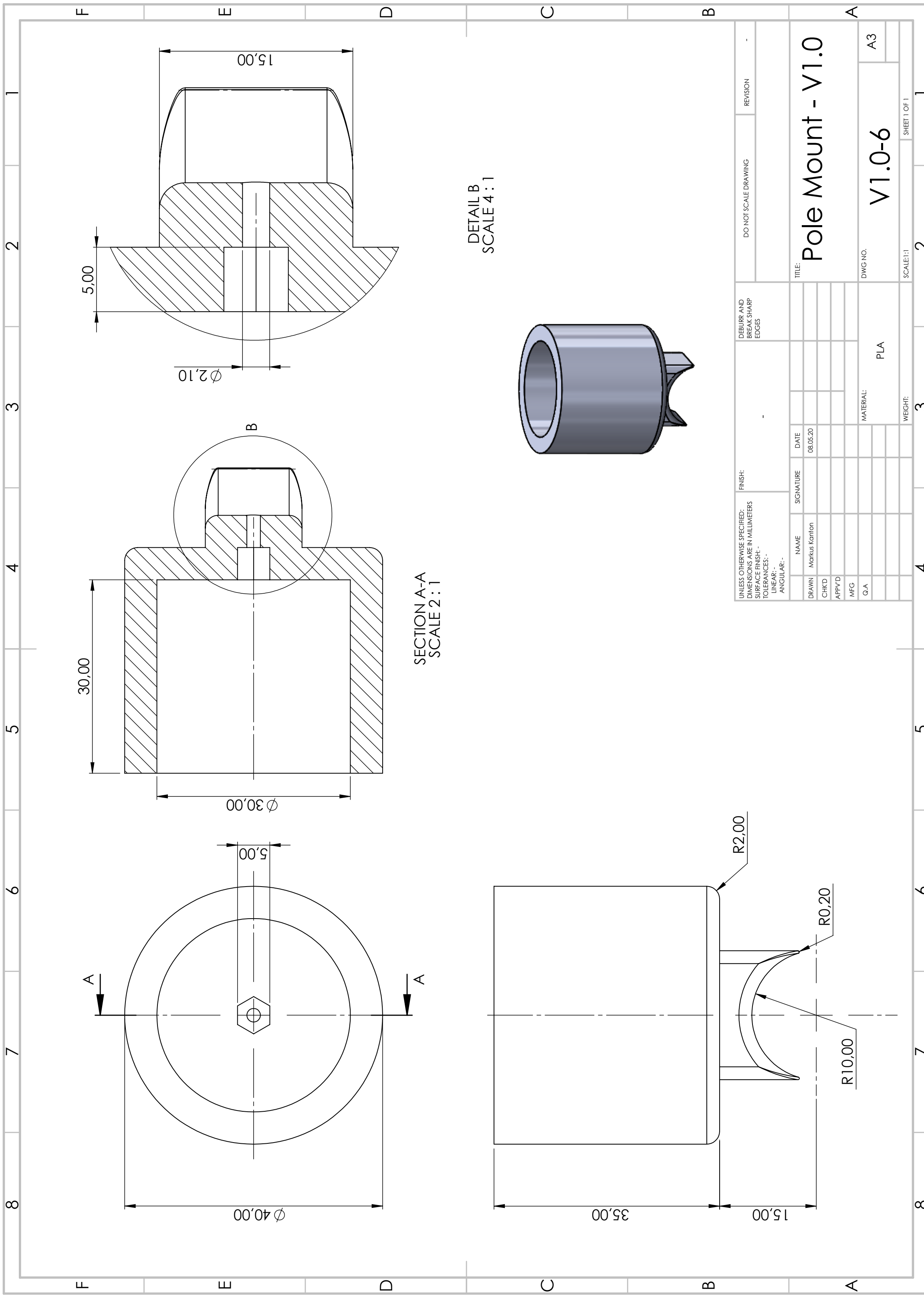

Wheel Joint - V1.0
V1.0-3
A3
PLA
SECTION A-A
A
10,00
35,00
70,00
⌀5,20
⌀45,00
100,00
30,00
8,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: -
TOLERANCES: -
LINEAR: -
ANGULAR: -
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
07.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
DWG NO.
SCALE:2:1
SHEET 1 OF 2
SOLIDWORKS Educational Product. For Instructional Use Only.


38,00
7,75
⌀64,00
⌀45,00
SECTION B-B
B
B
15,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: -
TOLERANCES: -
LINEAR: -
ANGULAR: -
FINISH:
-
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
-
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
07.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
PLA
TITLE:
Wheel Joint - V1.0
DWG NO.
V1.0-3
A3
WEIGHT:
SCALE:2:1
SHEET 2 OF 2
SOLIDWORKS Educational Product. For Instructional Use Only.

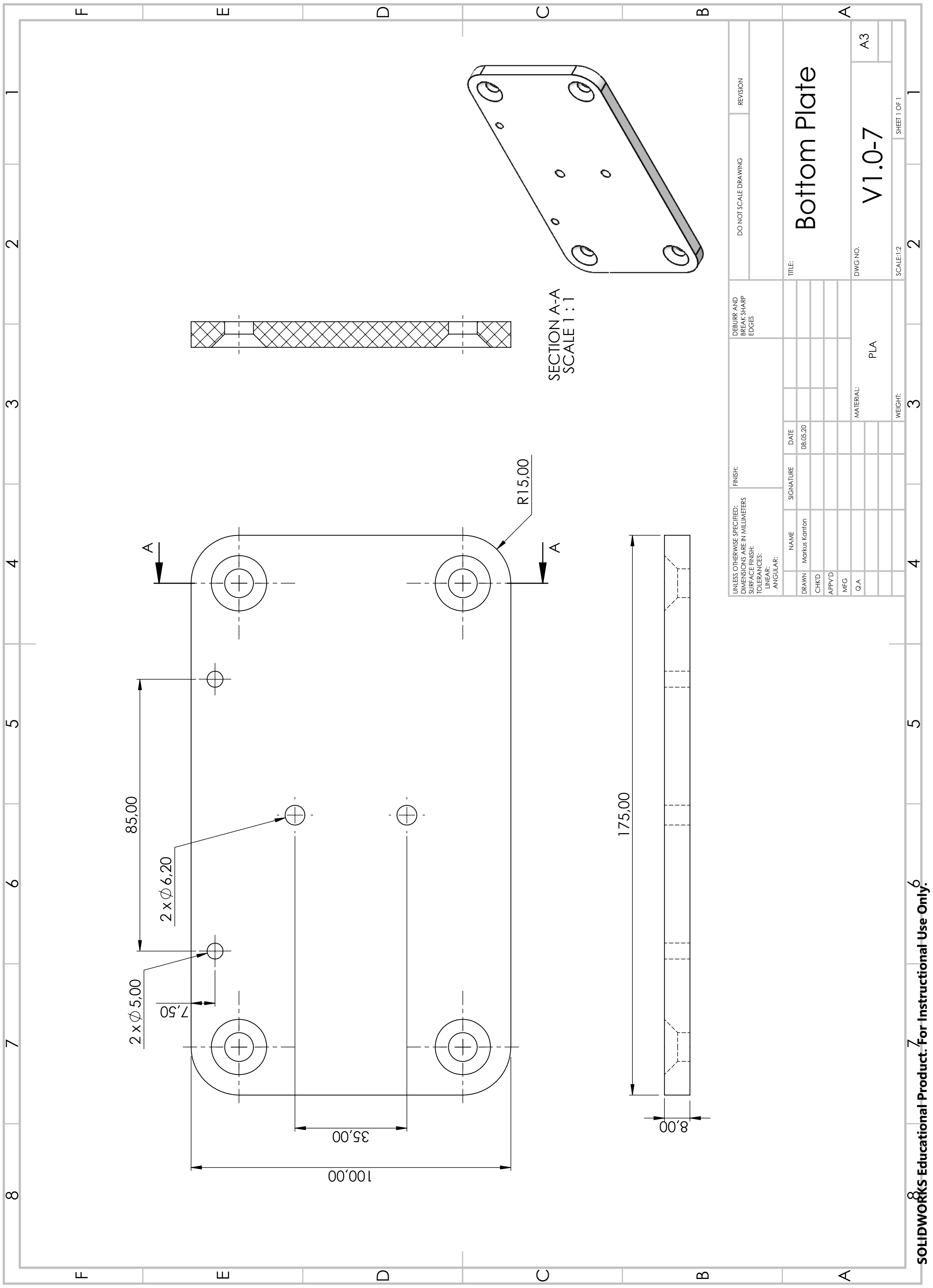

Rotation Pin - V1.0
V1.0-4
DWG NO.
TITLE:
A3
SCALE:2:1
SHEET 1 OF 1
MATERIAL:
PLA
WEIGHT:
DO NOT SCALE DRAWING
REVISION
DEBURR AND BREAK SHARP EDGES
FINISH:
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: - TOLERANCES: - LINEAR: - ANGULAR: -
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
07.05.20
CHK'D
APPV'D
MFG
Q.A
SECTION A-A
SECTION B-B
67,50
25,00
4,00
4,00
Ø3,20
Ø8,00
Ø12,00
10,00
2,00
45°
SOLIDWORKS Educational Product. For Instructional Use Only.

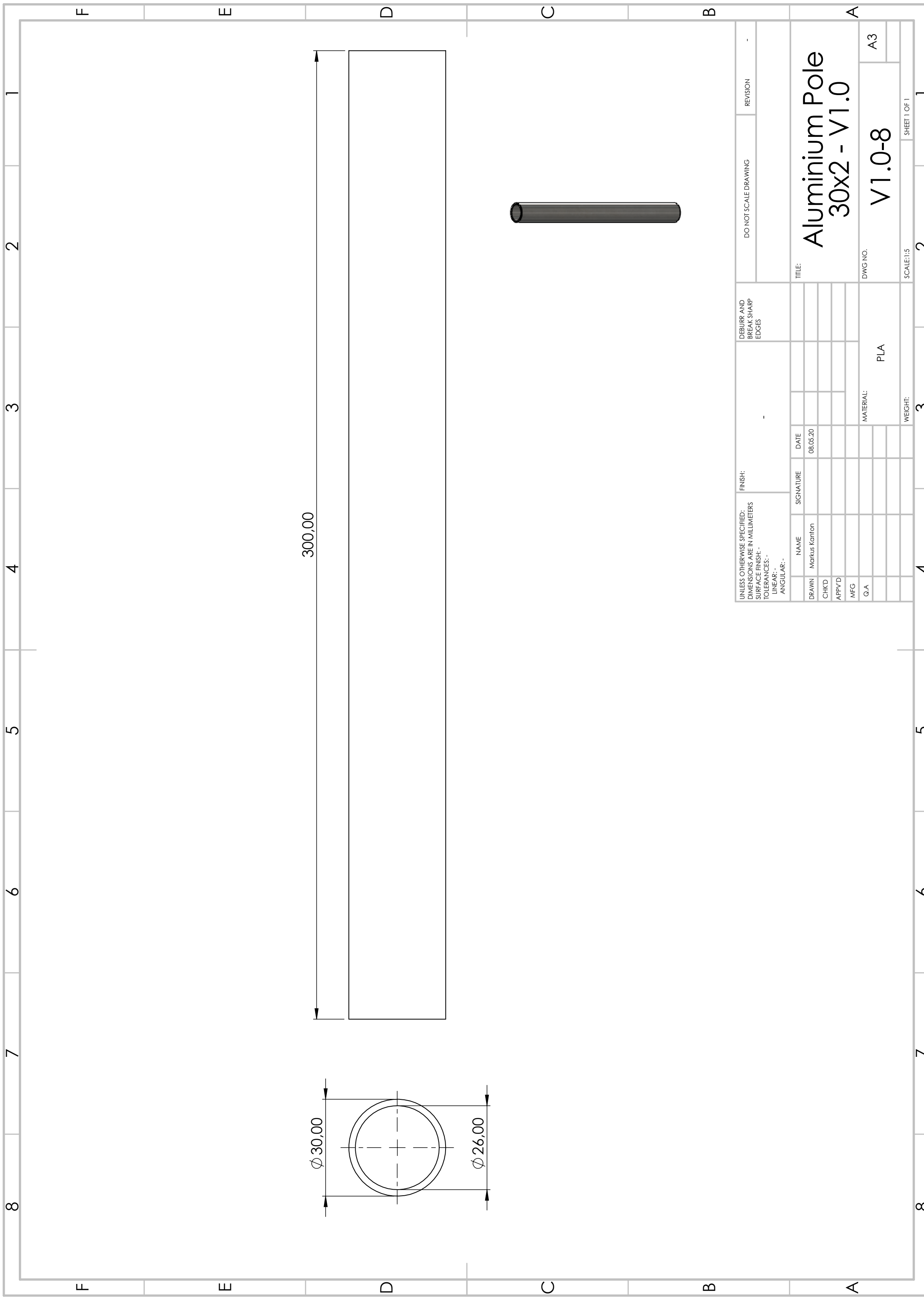


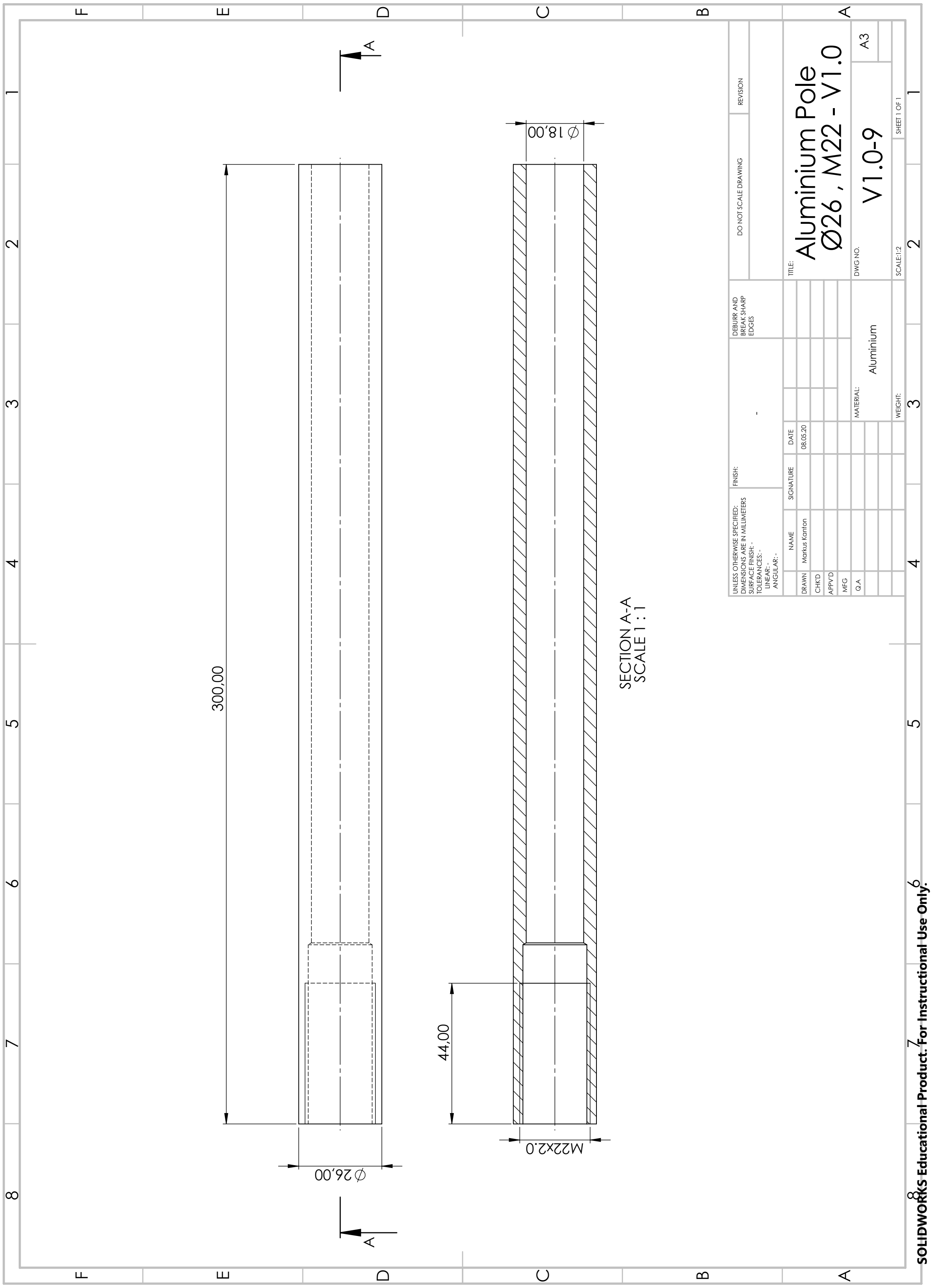
SECTION A-A
SCALE 3 : 1
DETAIL B
SCALE 6 : 1
Ø2,00
Ø4,00
2,00
4,50
6,00
5,00
1,50
Ø8,00
Ø20,00
15,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
DRAWN
Markus Kanton
08.05.20
TITLE:
Rotation Bracket - V1.0
MATERIAL:
PLA
DWG NO.
V1.0-5
A3
SCALE:2:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.



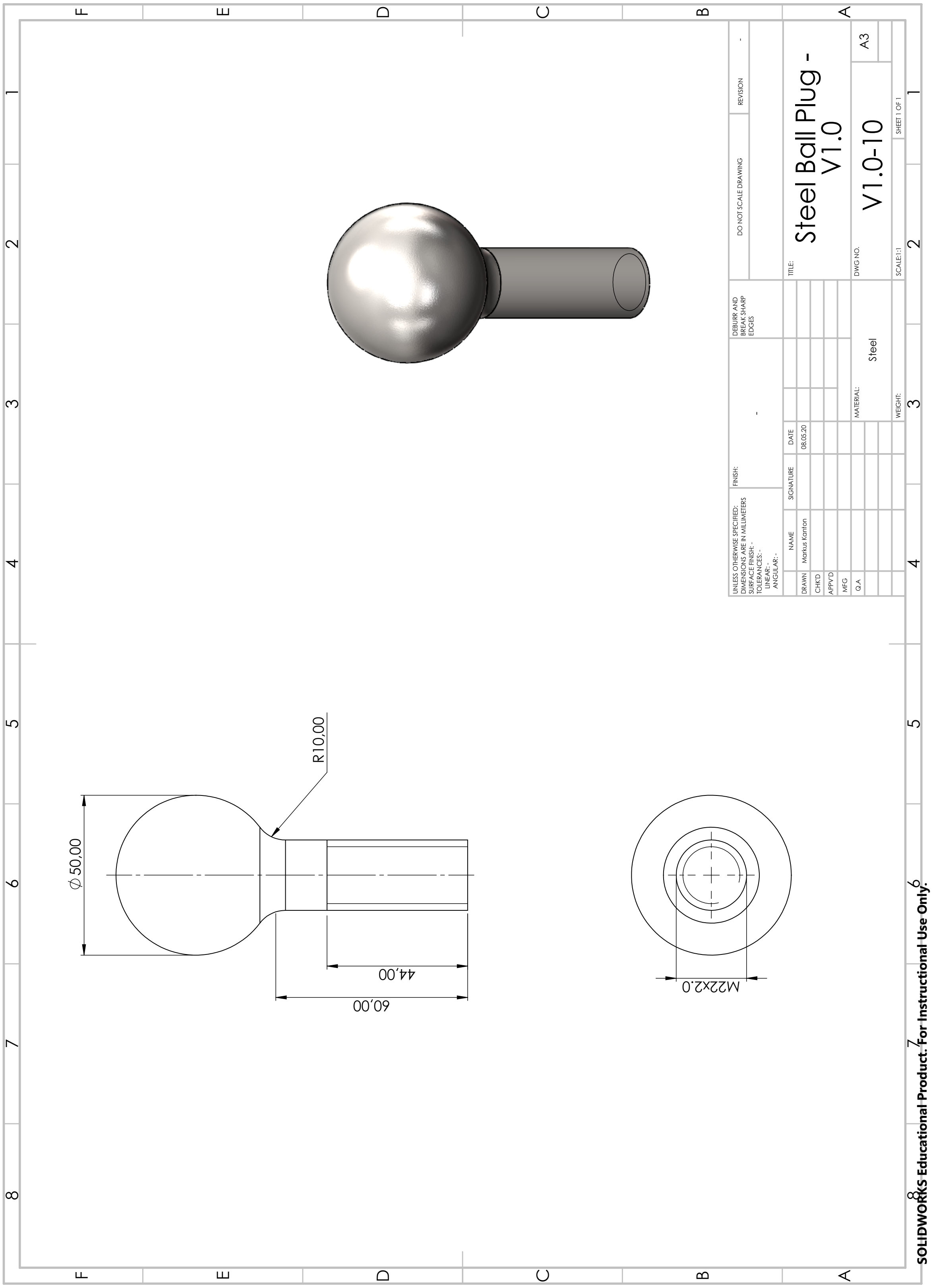
15,00
5,00
⌀2,10
DETAIL B
SCALE 4 : 1
B
30,00
⌀30,00
SECTION A-A
SCALE 2 : 1
5,00
A
A
⌀40,00
R2,00
R0,20
R10,00
35,00
15,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: -
TOLERANCES: -
LINEAR: -
ANGULAR: -
FINISH:
-
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
-
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
08.05.20
CHK'D
APPV'D
MFG
Q.A
TITLE:
Pole Mount - V1.0
MATERIAL:
PLA
DWG NO.
V1.0-6
A3
WEIGHT:
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


Bottom Plate
V1.0-7
A3
SECTION A-A
SCALE 1 : 1
R15,00
85,00
2 x ⌀ 6,20
2 x ⌀ 5,00
7,50
175,00
35,00
100,00
8,00
A
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
08.05.20
CHK'D
APPV'D
MFG
Q.A
TITLE:
MATERIAL:
PLA
DWG NO.
WEIGHT:
SCALE:1:2
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

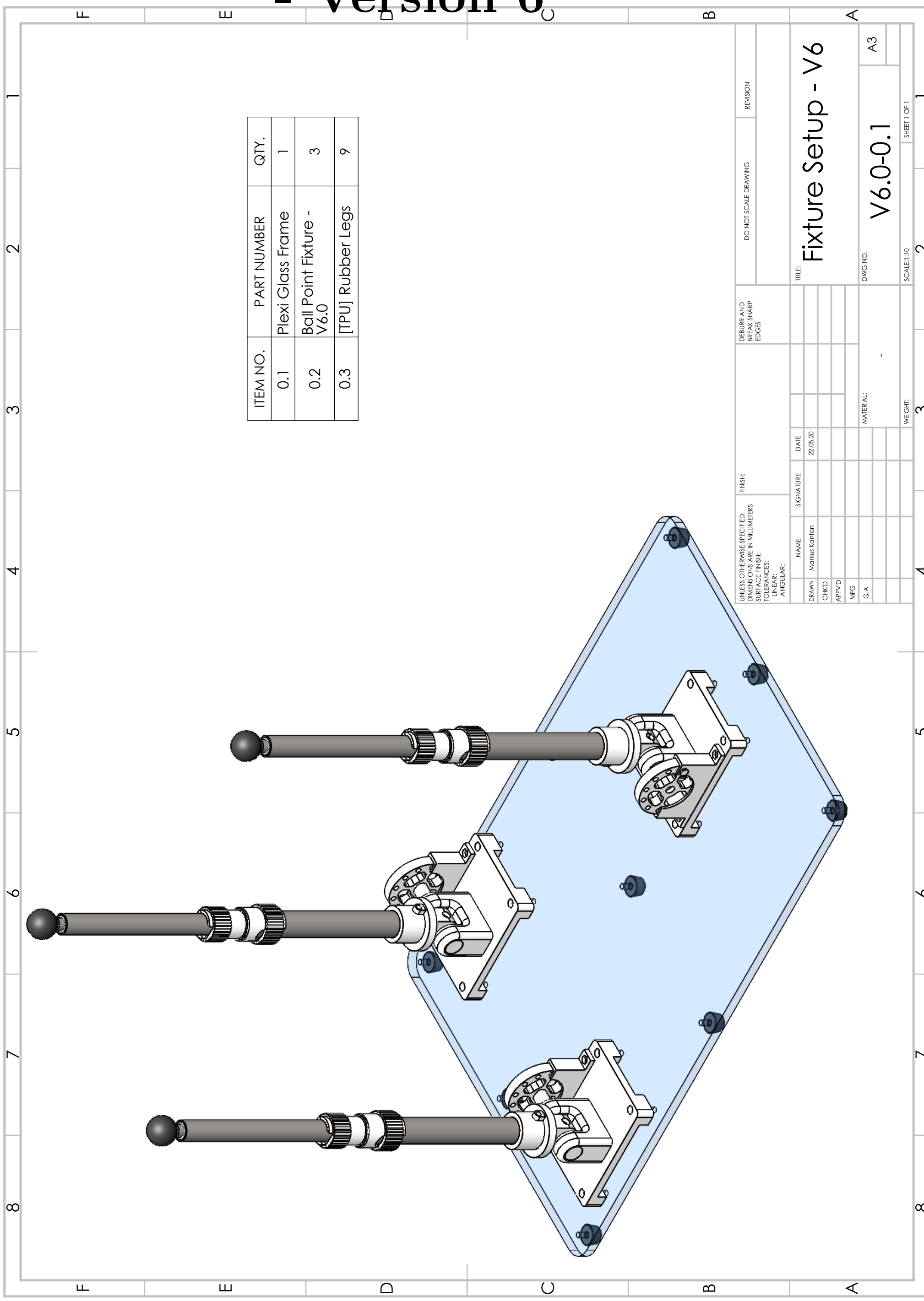

Aluminium Pole
30x2 - V1.0
V1.0-8
A3
DWG NO.
TITLE:
SCALE:1:5
SHEET 1 OF 1
WEIGHT:
MATERIAL:
PLA
DO NOT SCALE DRAWING
REVISION
DEBURR AND BREAK SHARP EDGES
FINISH:
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
08.05.20
CHK'D
APPV'D
MFG
Q.A
300,00
∅30,00
∅26,00
SOLIDWORKS Educational Product. For Instructional Use Only.

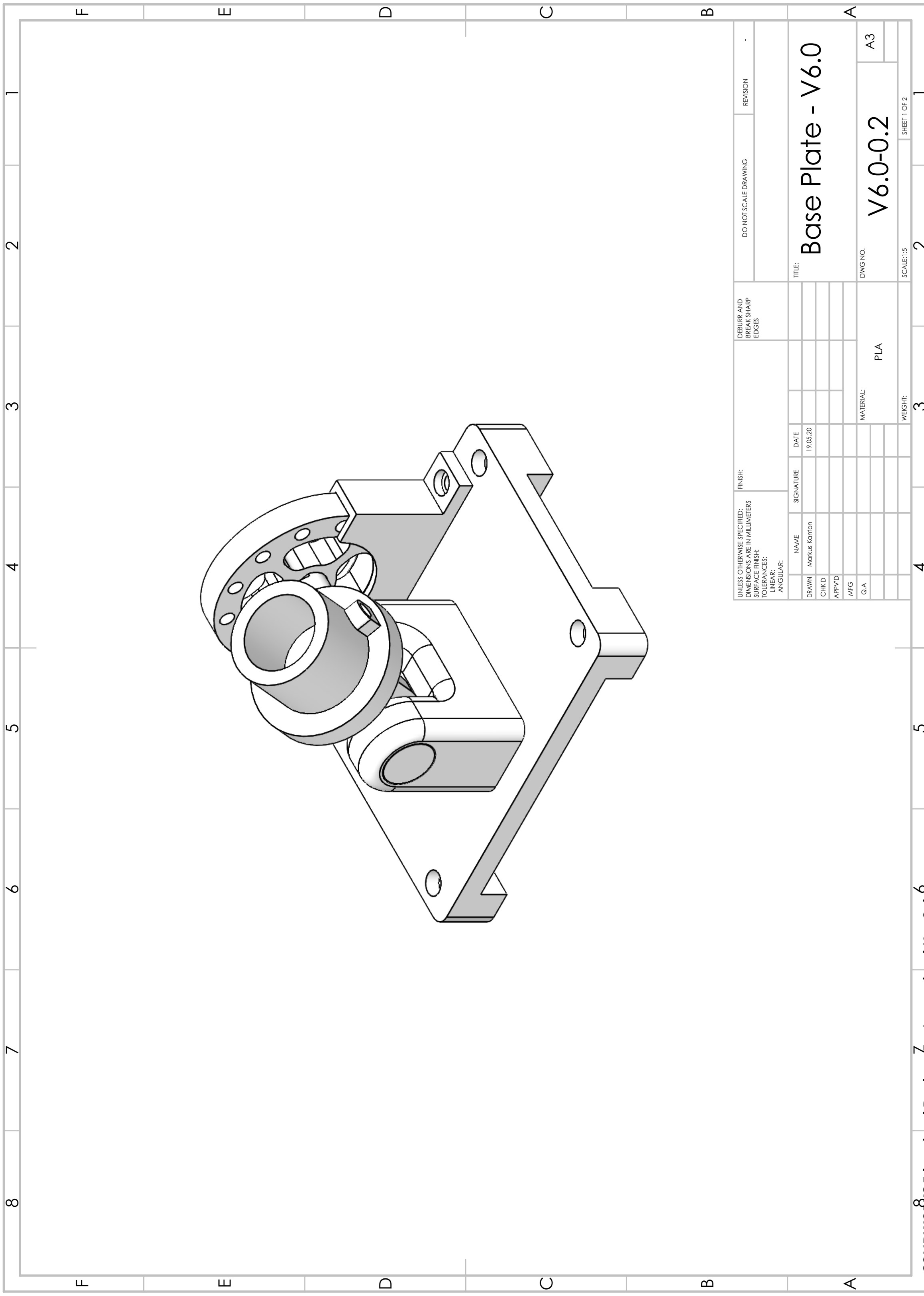


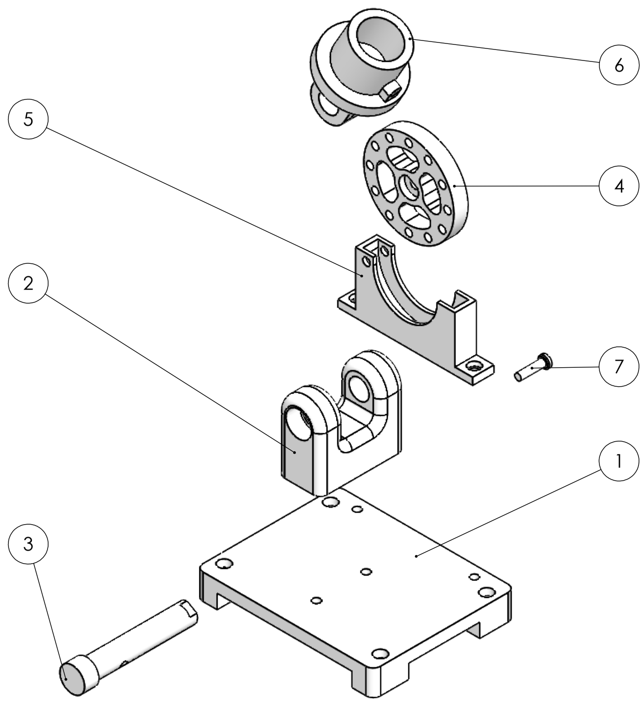
A
A
300,00
∅26,00
∅18,00
44,00
M22x2.0
SECTION A-A
SCALE 1 : 1
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: -
TOLERANCES: -
LINEAR: -
ANGULAR: -
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
08.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
Aluminium
TITLE:
Aluminium Pole
Ø26 , M22 - V1.0
DWG NO.
V1.0-9
A3
WEIGHT:
SCALE:1:2
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


Ø50,00
R10,00
44,00
60,00
M22x2,0
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: -
TOLERANCES: -
LINEAR: -
ANGULAR: -
FINISH:
-
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
-
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
08.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
Steel
WEIGHT:
TITLE:
Steel Ball Plug - V1.0
DWG NO.
V1.0-10
A3
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

Appendix J Ball Point Fixture Schematics - Version 6




UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
-
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
19.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
PLA
TITLE:
Base Plate - V6.0
DWG NO.
V6.0-0.2
A3
WEIGHT:
SCALE:1:5
SHEET 1 OF 2
SOLIDWORKS Educational Product. For Instructional Use Only.



ITEM NO.	PART NUMBER	QTY.
1	Bottom Plate	1
2	Rotation Joint	1
3	Rotation Pin	1
4	Wheel	1
5	Wheel Joint	1
6	Pole Mount	1
7	Locking Pin	1

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN	Markus Kanton		19.05.20
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE: Base Plate - V6.0

MATERIAL: PLA

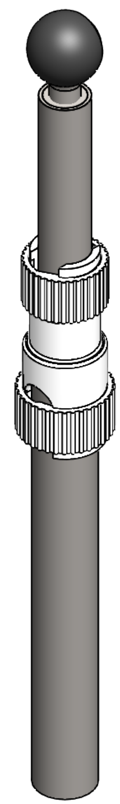
DWG NO. V6.0-0.2

A3

WEIGHT:

SCALE:1:5

SHEET 2 OF 2



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE	
DRAWN	Markus Kanton		22.05.20	
CHK'D				
APPV'D				
MFG				
Q.A				

TITLE: Telescopic Pole - V6

MATERIAL: -

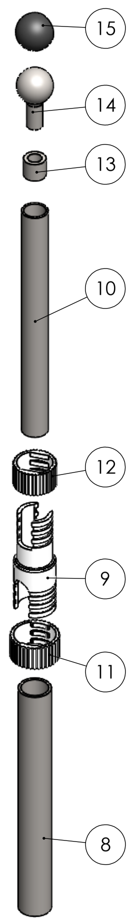
DWG NO. V6.0-0.3

A3

WEIGHT:

SCALE:1:5

SHEET 1 OF 2



ITEM NO.	PART NUMBER	QTY.
8	Ø24x2mm Steel Pole	1
9	Pole Clamp	1
10	Ø20x1,5mm Steel Pole	1
11	Pole Clamp Ring - Large	1
12	Pole Clamp Ring - Small	1
13	Ø18, M12 Insert	1
14	Steel Ball Plug	1
15	Steel Ball Rubber Cover	1

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE
DRAWN	Markus Kanton		22.05.20
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: -

WEIGHT:

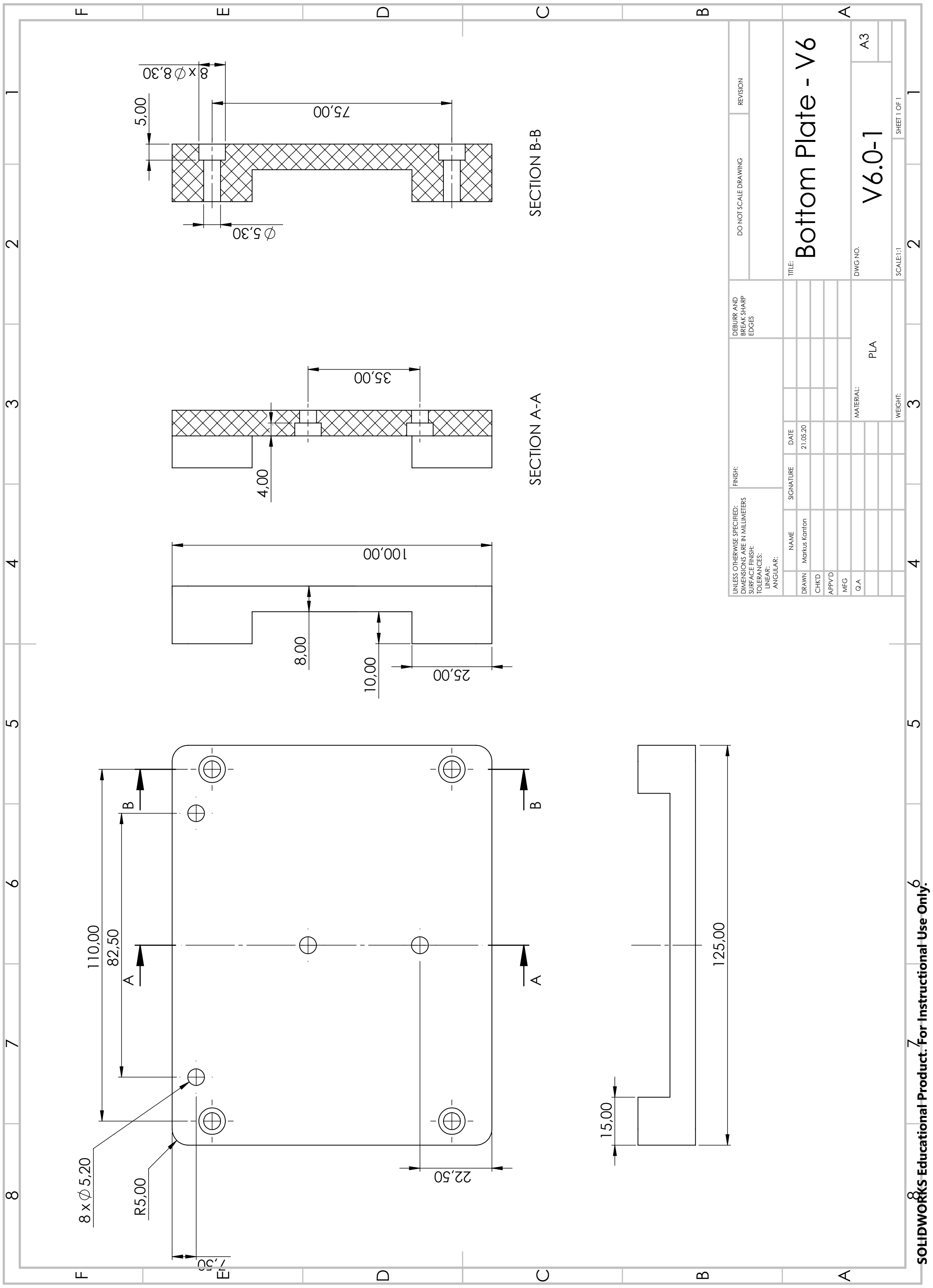
TITLE: Telescopic Pole - V6

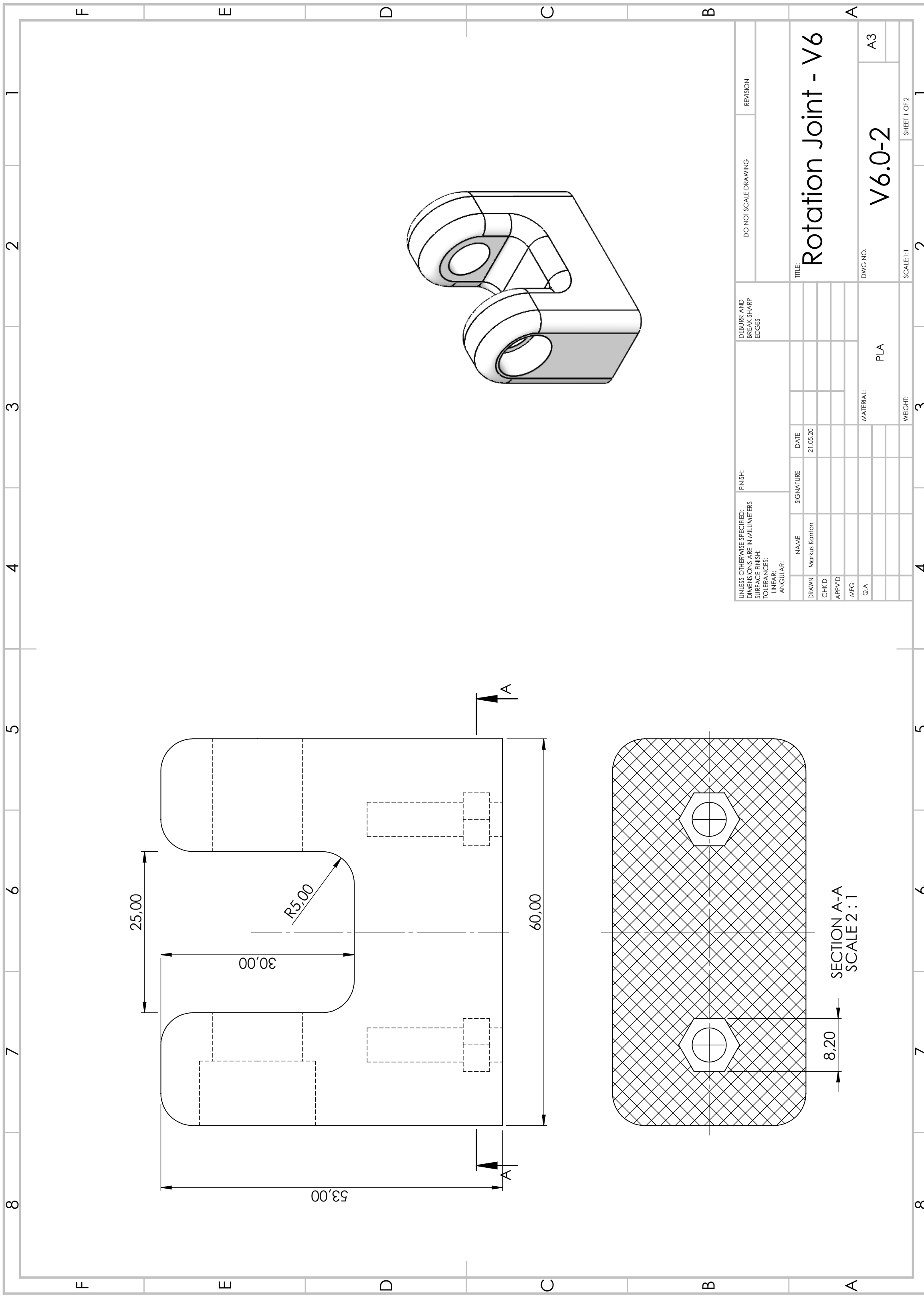
DWG NO. V6.0-0.3

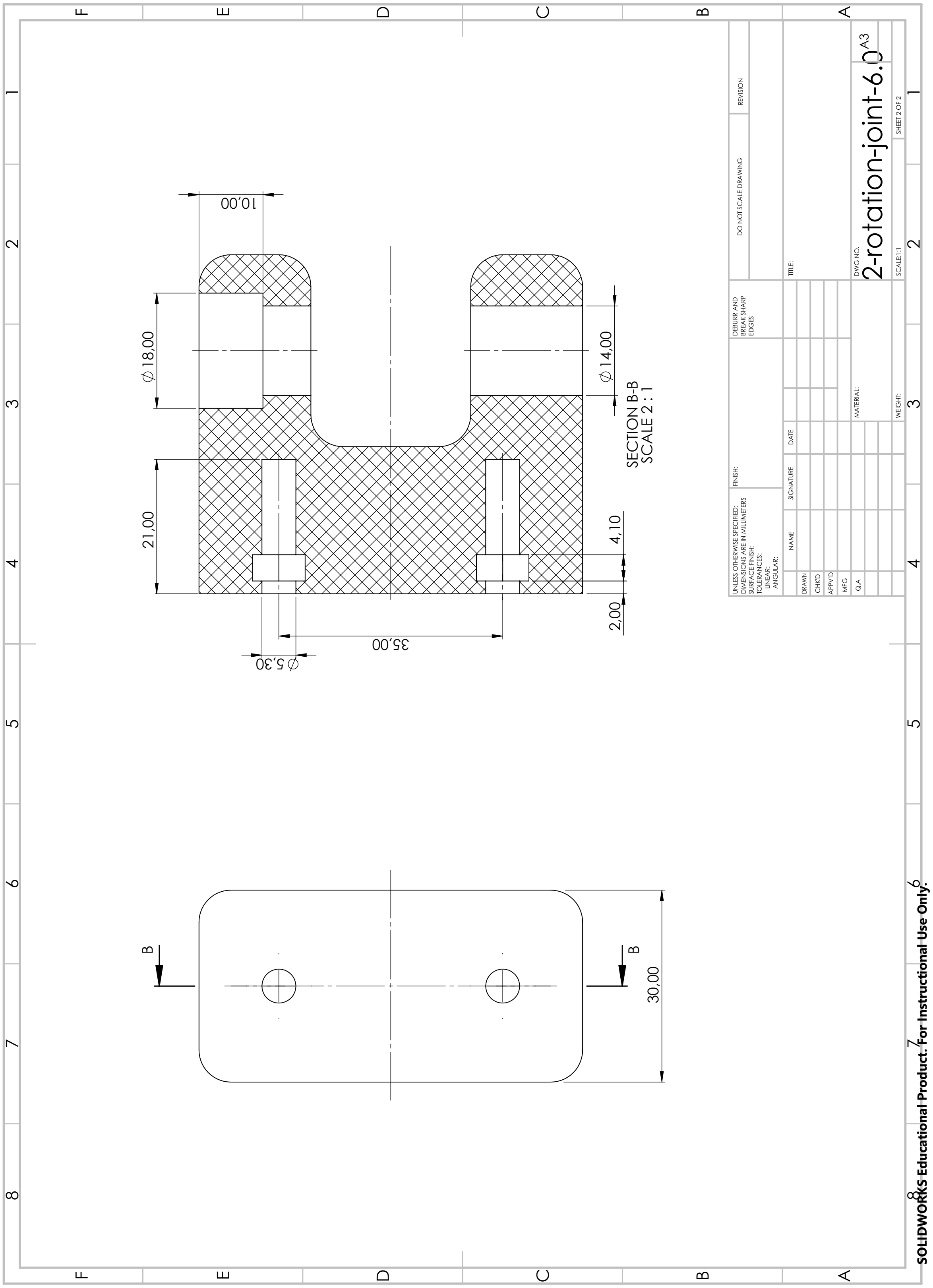
A3

SCALE:1:5

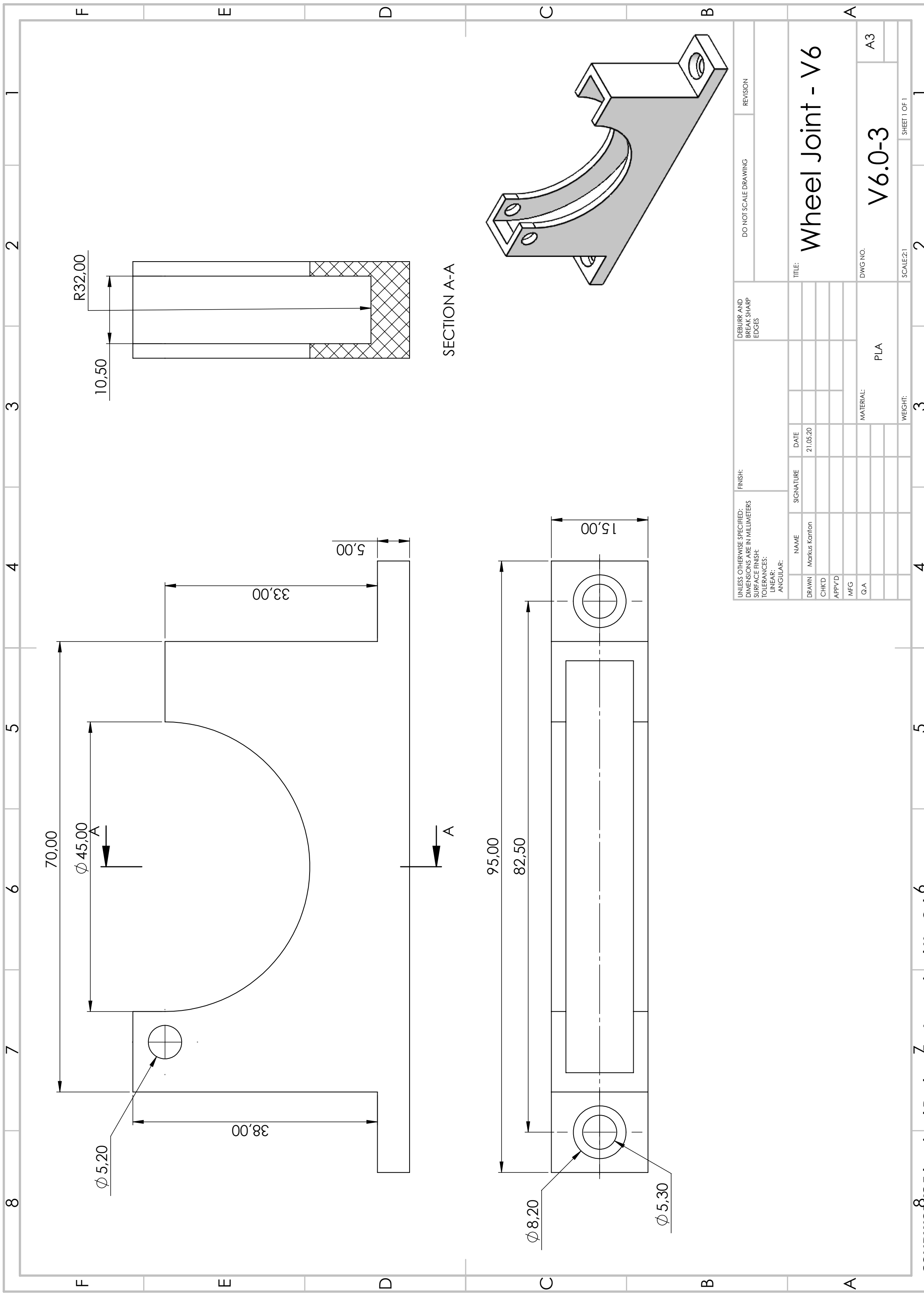
SHEET 2 OF 2

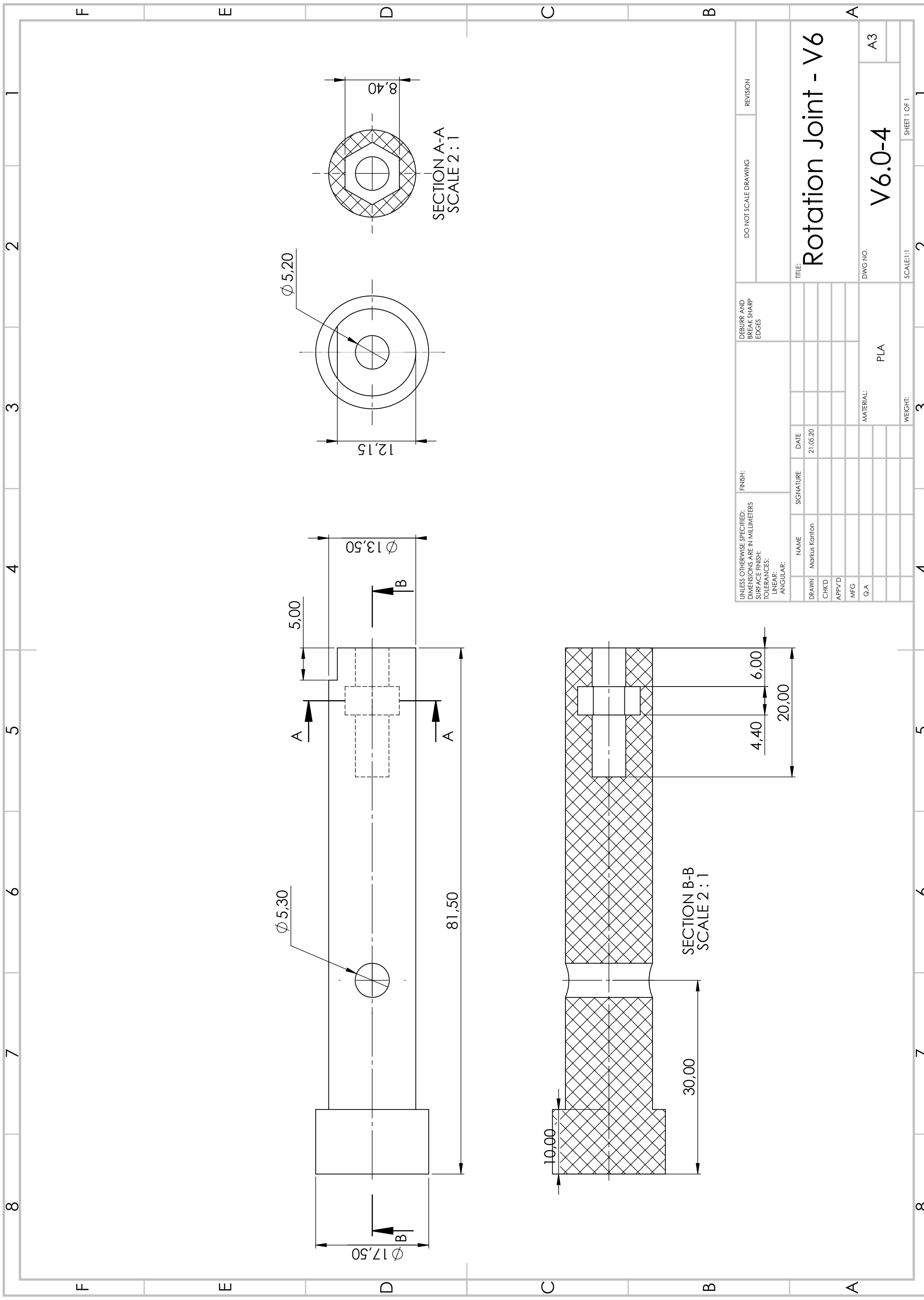

Bottom Plate - V6
V6.0-1
A3
PLA
SECTION A-A
SECTION B-B
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
TITLE:
DWG NO.
SCALE:1:1
SHEET 1 OF 1
WEIGHT:
MATERIAL:
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
21.05.20
CHK'D
APPV'D
MFG
Q.A
8 x ⌀8,30
5,00
75,00
⌀5,30
35,00
4,00
100,00
8,00
10,00
25,00
110,00
82,50
8 x ⌀5,20
R5,00
7,50
22,50
15,00
125,00
SOLIDWORKS Educational Product. For Instructional Use Only.

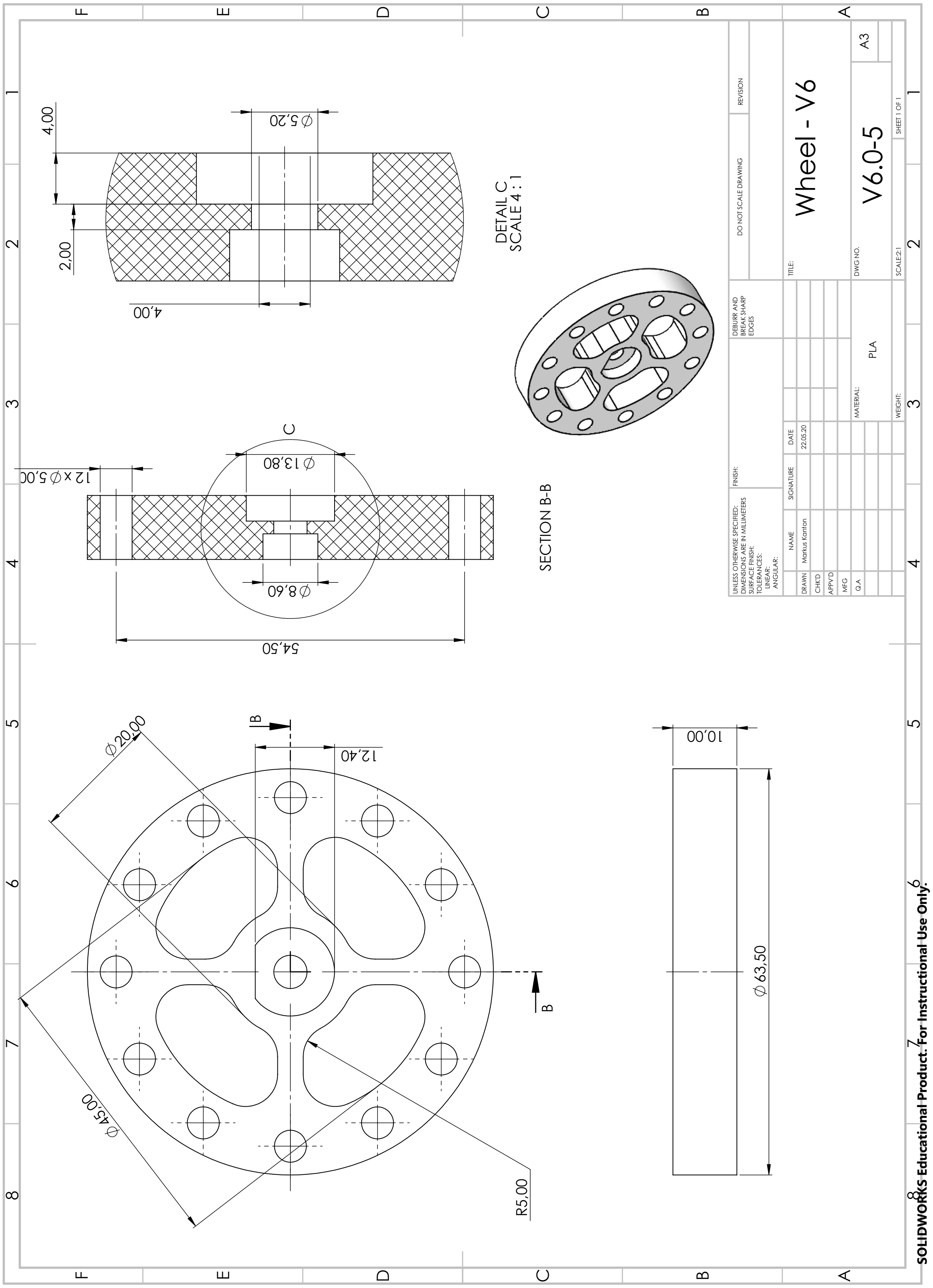

Rotation Joint - V6
V6.0-2
DWG NO.
TITLE:
A3
SHEET 1 OF 2
SCALE:1:1
WEIGHT:
MATERIAL:
PLA
DO NOT SCALE DRAWING
REVISION
DEBURR AND BREAK SHARP EDGES
FINISH:
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
21.05.20
CHK'D
APPV'D
MFG
Q.A
SECTION A-A
SCALE 2 : 1
25,00
30,00
R5,00
60,00
53,00
8,20
A
SOLIDWORKS Educational Product. For Instructional Use Only.

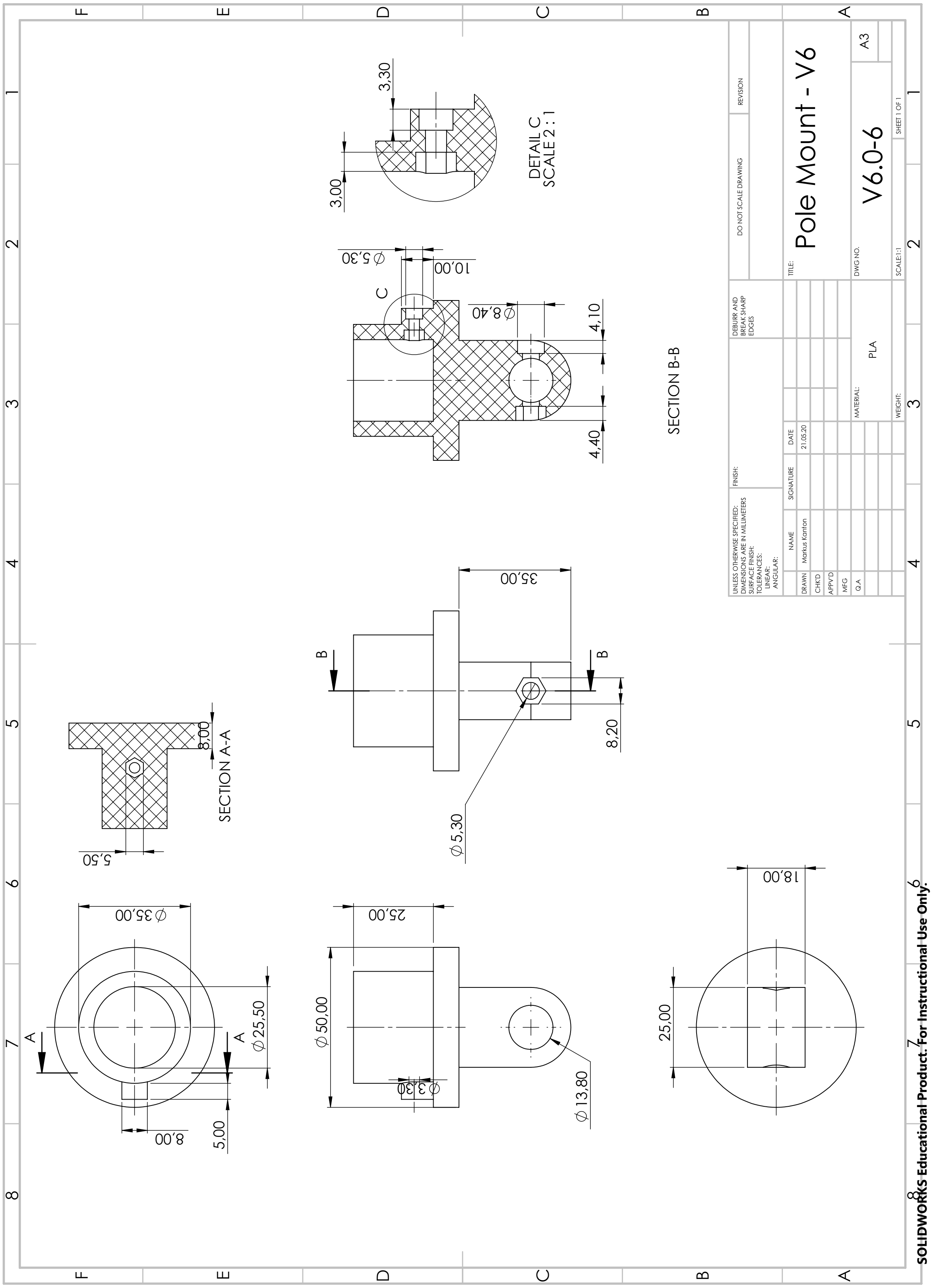


10,00
⌀18,00
⌀14,00
21,00
4,10
2,00
35,00
⌀5,30
SECTION B-B
SCALE 2 : 1
B
B
30,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
TITLE:
DWG NO.
2-rotation-joint-6.0
A3
WEIGHT:
SCALE:1:1
SHEET 2 OF 2
SOLIDWORKS Educational Product. For Instructional Use Only.

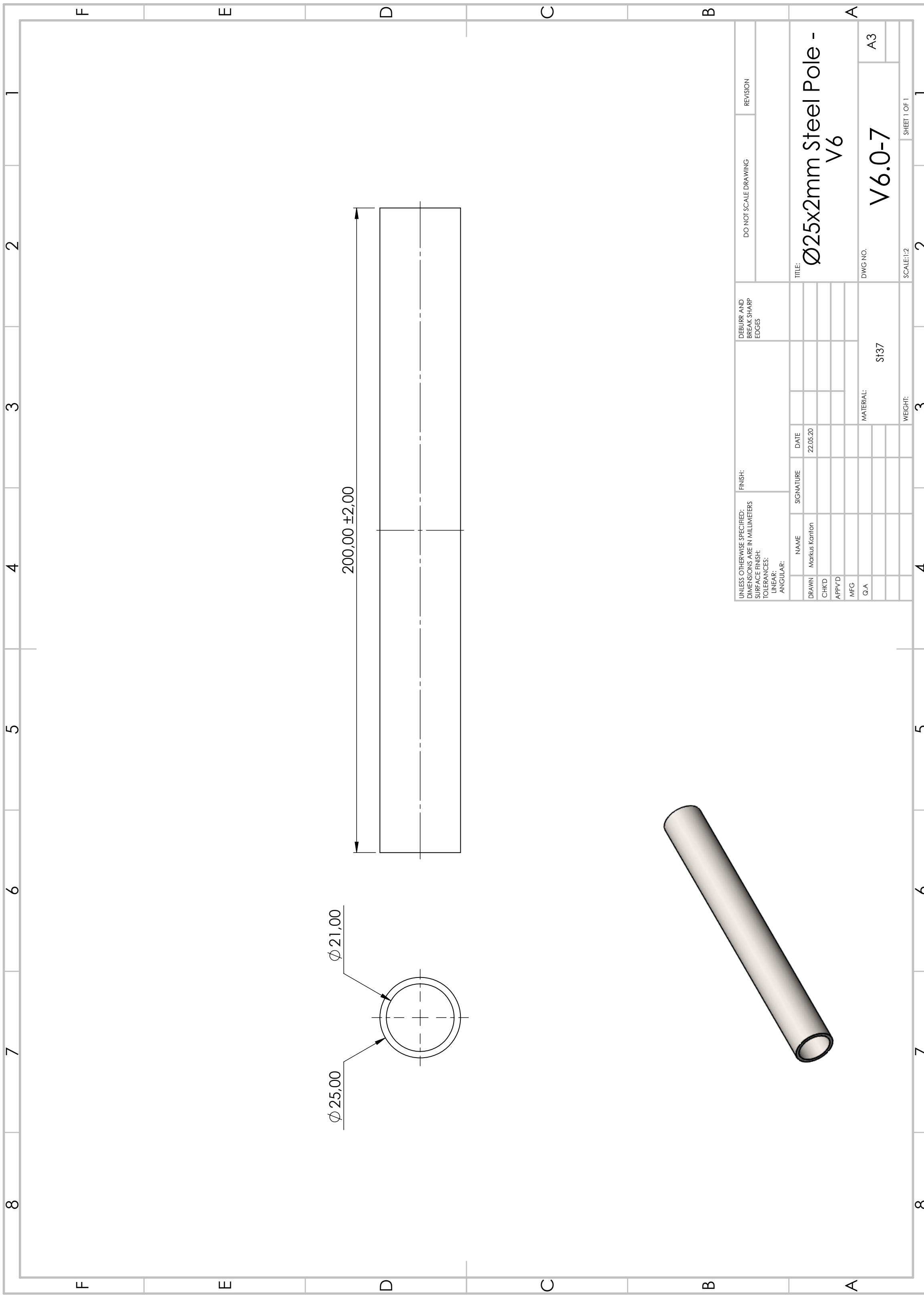

R32,00
10,50
SECTION A-A
5,00
33,00
70,00
⌀45,00
A
A
⌀5,20
38,00
15,00
95,00
82,50
⌀8,20
⌀5,30
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
21.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
PLA
WEIGHT:
TITLE:
Wheel Joint - V6
DWG NO.
V6.0-3
A3
SCALE:2:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

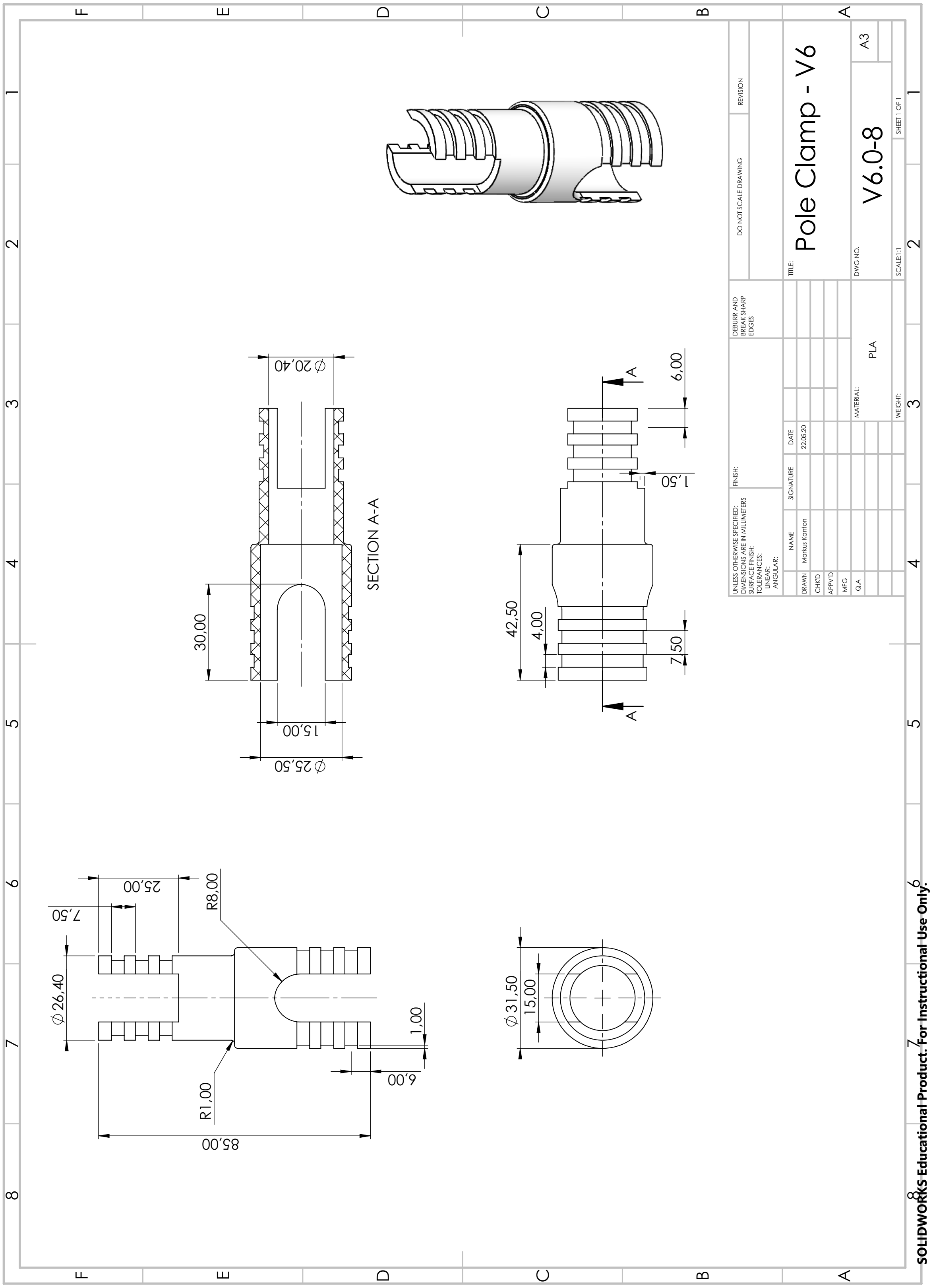

Rotation Joint - V6
V6.0-4
A3
PLA
SCALE:1:1
SHEET 1 OF 1
DWG NO.
MATERIAL:
WEIGHT:
TITLE:
DRAWN Markus Kanton 21.05.20
NAME
SIGNATURE
DATE
CHK'D
APPV'D
MFG
Q.A
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
SECTION A-A
SCALE 2 : 1
SECTION B-B
SCALE 2 : 1
8,40
Ø 5,20
12,15
Ø 13,50
5,00
Ø 5,30
81,50
Ø 17,50
6,00
4,40
20,00
30,00
10,00
SOLIDWORKS Educational Product. For Instructional Use Only.

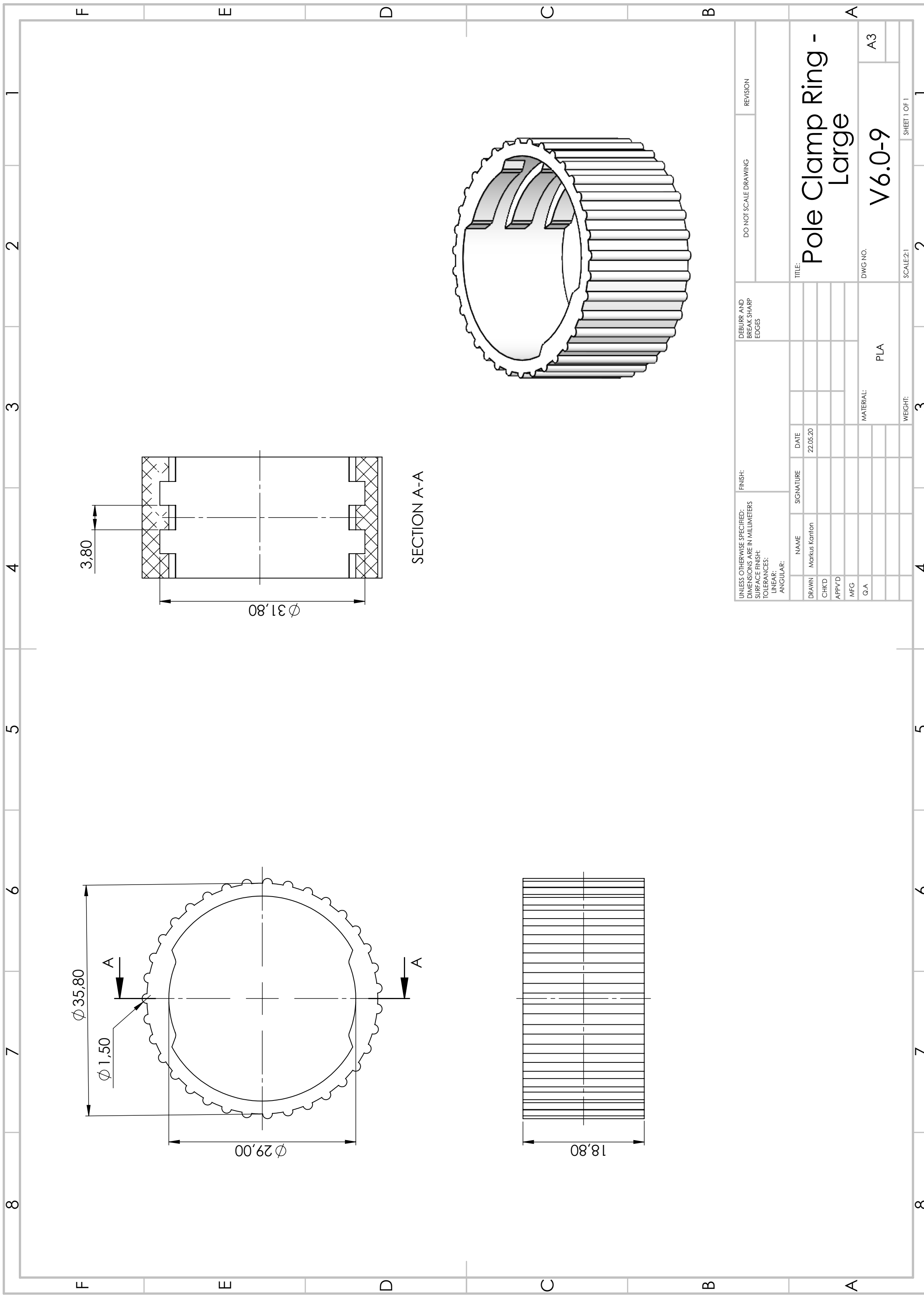

Wheel - V6
V6.0-5
DETAIL C
SCALE 4 : 1
SECTION B-B
4,00
2,00
∅5,20
4,00
∅13,80
12 x ∅5,00
∅8,60
54,50
∅20,00
12,40
10,00
∅63,50
∅45,00
R5,00
B
C
DO NOT SCALE DRAWING
REVISION
DEBURR AND BREAK SHARP EDGES
FINISH:
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
22.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
PLA
WEIGHT:
TITLE:
DWG NO.
SCALE:2:1
SHEET 1 OF 1
A3
SOLIDWORKS Educational Product. For Instructional Use Only.

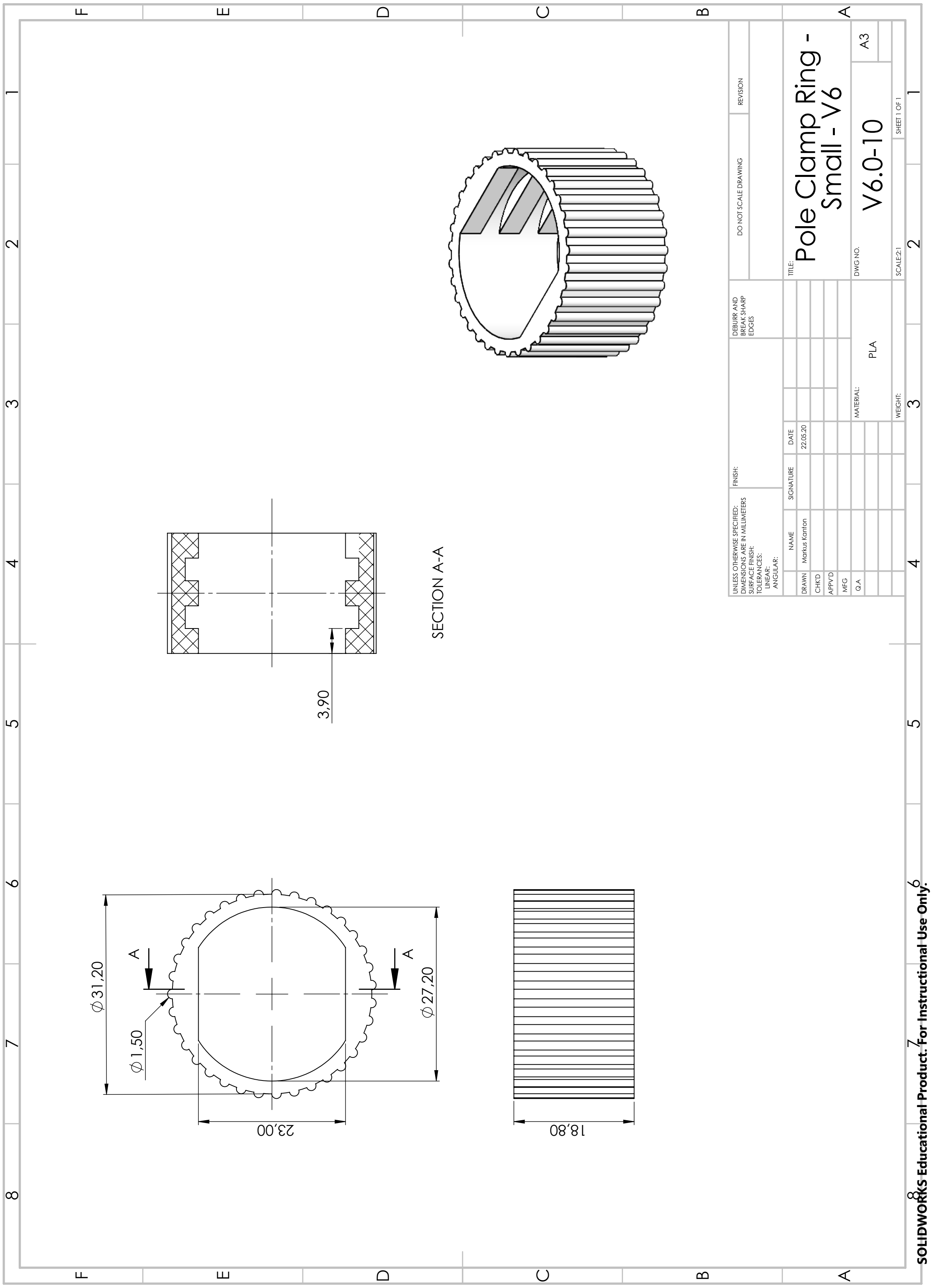


Pole Mount - V6
V6.0-6
SECTION A-A
SECTION B-B
DETAIL C
SCALE 2 : 1
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
21.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
PLA
WEIGHT:
TITLE:
DWG NO.
A3
SCALE:1:1
SHEET 1 OF 1
3,30
3,00
⌀5,30
10,00
⌀8,40
4,10
4,40
35,00
8,20
8,00
5,50
⌀35,00
⌀25,50
8,00
5,00
25,00
⌀50,00
⌀3,30
⌀13,80
18,00
25,00
SOLIDWORKS Educational Product. For Instructional Use Only.

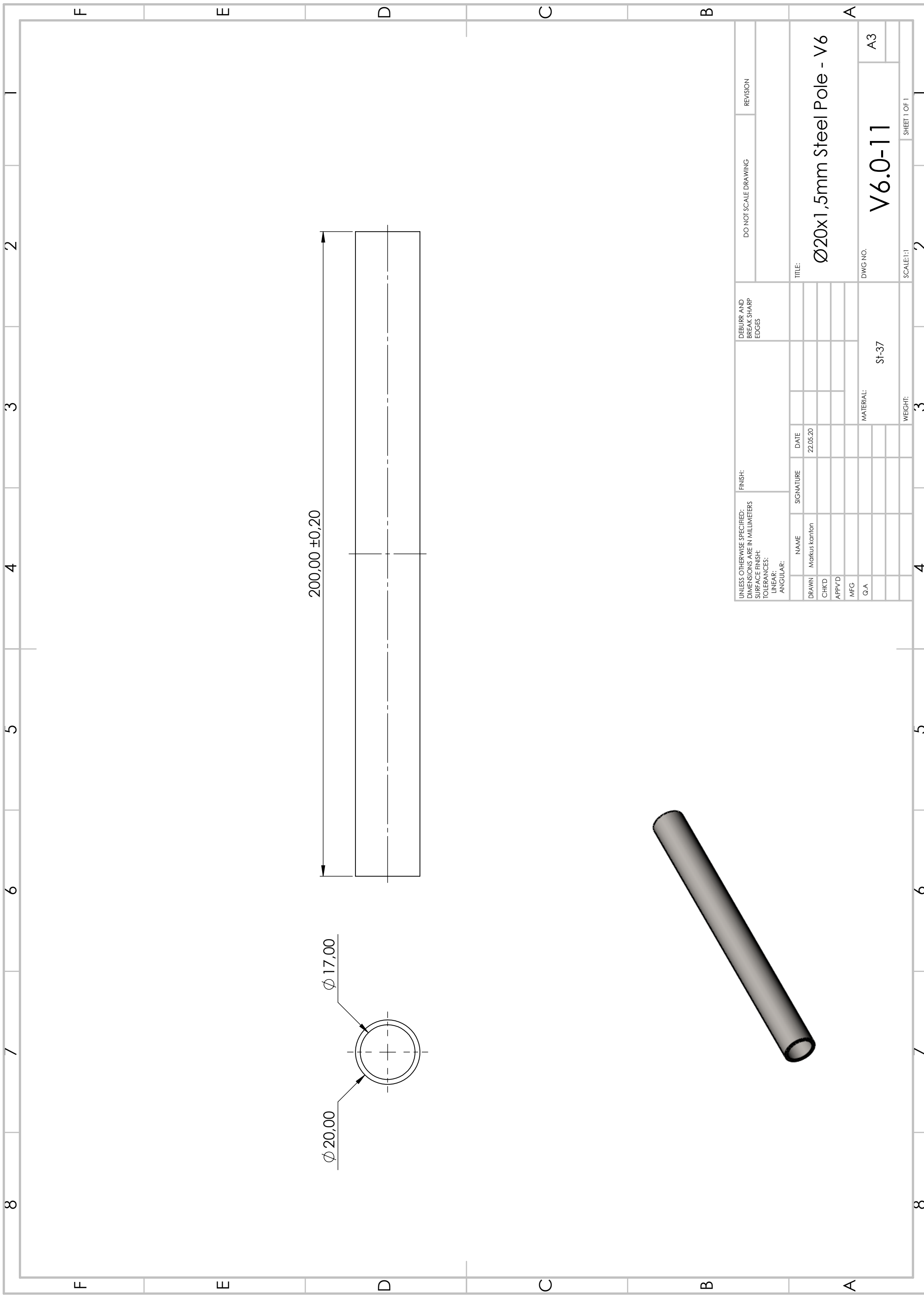

Ø25x2mm Steel Pole - V6
V6.0-7
200,00 ±2,00
Ø21,00
Ø25,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
22.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
S137
WEIGHT:
TITLE:
DWG NO.
A3
SCALE:1:2
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.

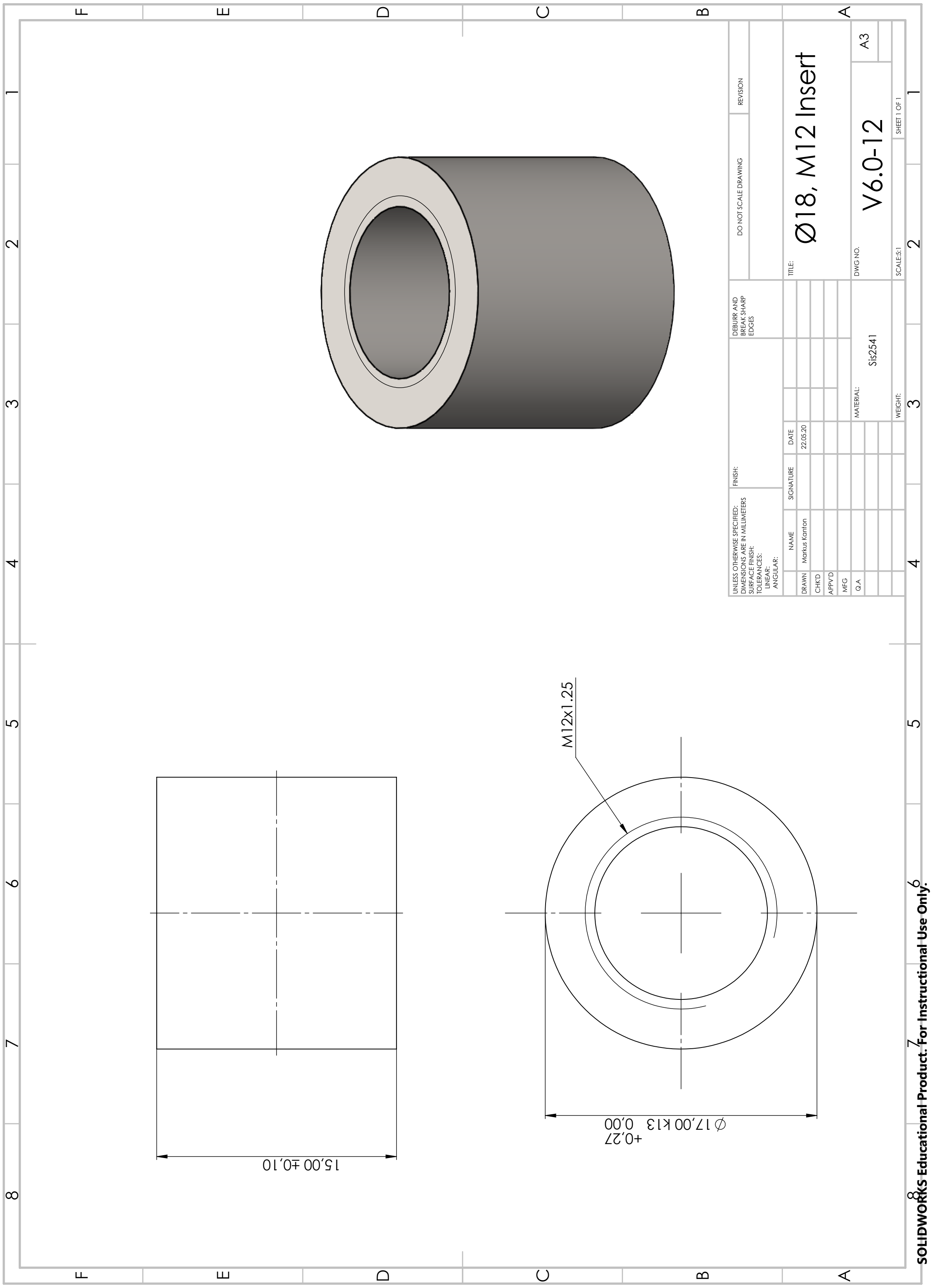

Pole Clamp - V6
V6.0-8
DWG NO.
TITLE:
A3
SCALE:1:1
SHEET 1 OF 1
WEIGHT:
MATERIAL:
PLA
DO NOT SCALE DRAWING
REVISION
DEBURR AND BREAK SHARP EDGES
FINISH:
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
22.05.20
CHK'D
APPV'D
MFG
Q.A
SECTION A-A
SOLIDWORKS Educational Product. For Instructional Use Only.

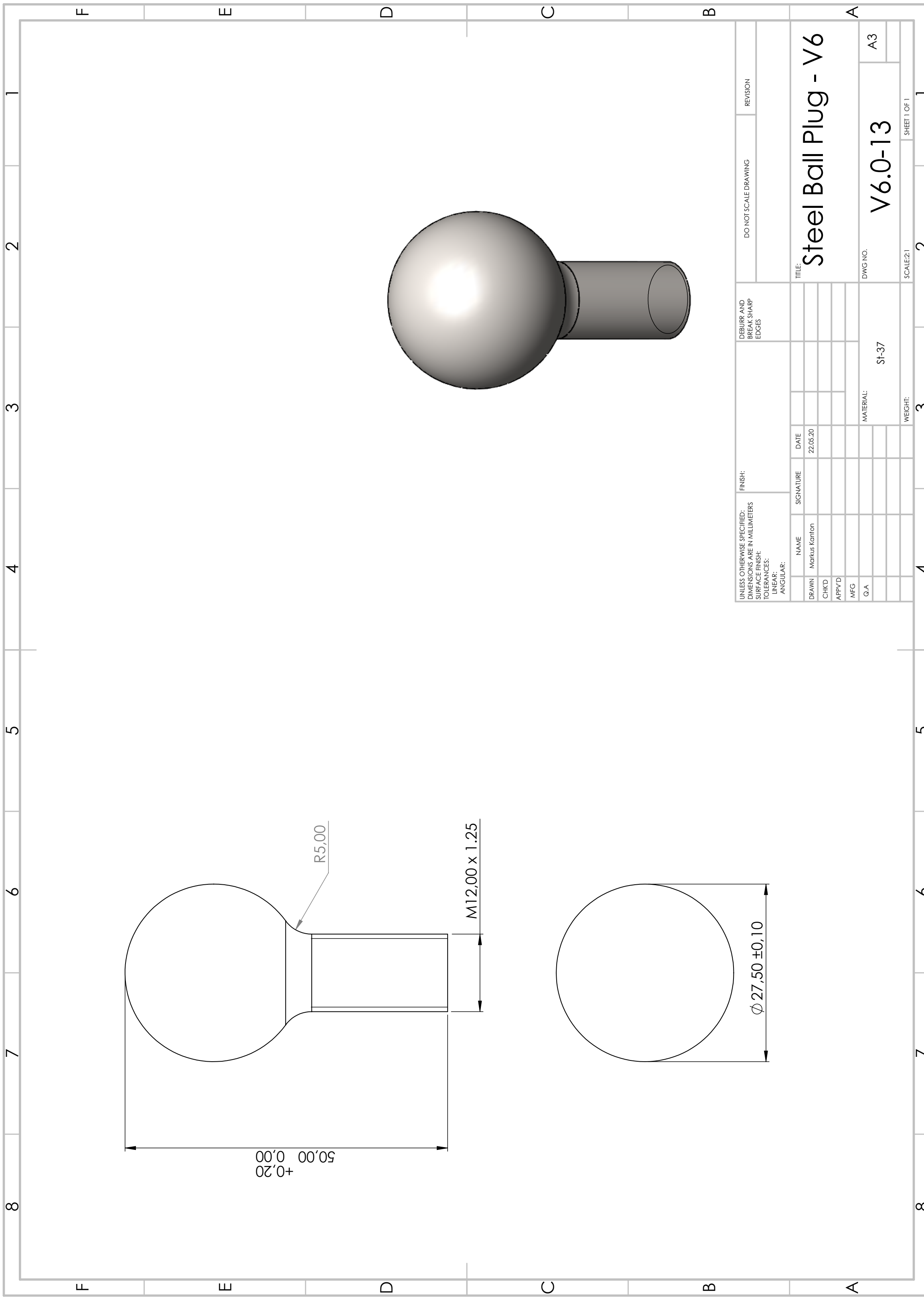

Pole Clamp Ring - Large
V6.0-9
PLA
SECTION A-A
3,80
Ø31,80
Ø35,80
Ø1,50
Ø29,00
18,80
A3
DO NOT SCALE DRAWING
REVISION
DEBURR AND BREAK SHARP EDGES
FINISH:
TITLE:
DWG NO.
MATERIAL:
WEIGHT:
SCALE:2:1
SHEET 1 OF 1
NAME
SIGNATURE
DATE
DRAWN
22.05.20
CHK'D
APPV'D
MFG
Q.A
SOLIDWORKS Educational Product. For Instructional Use Only.



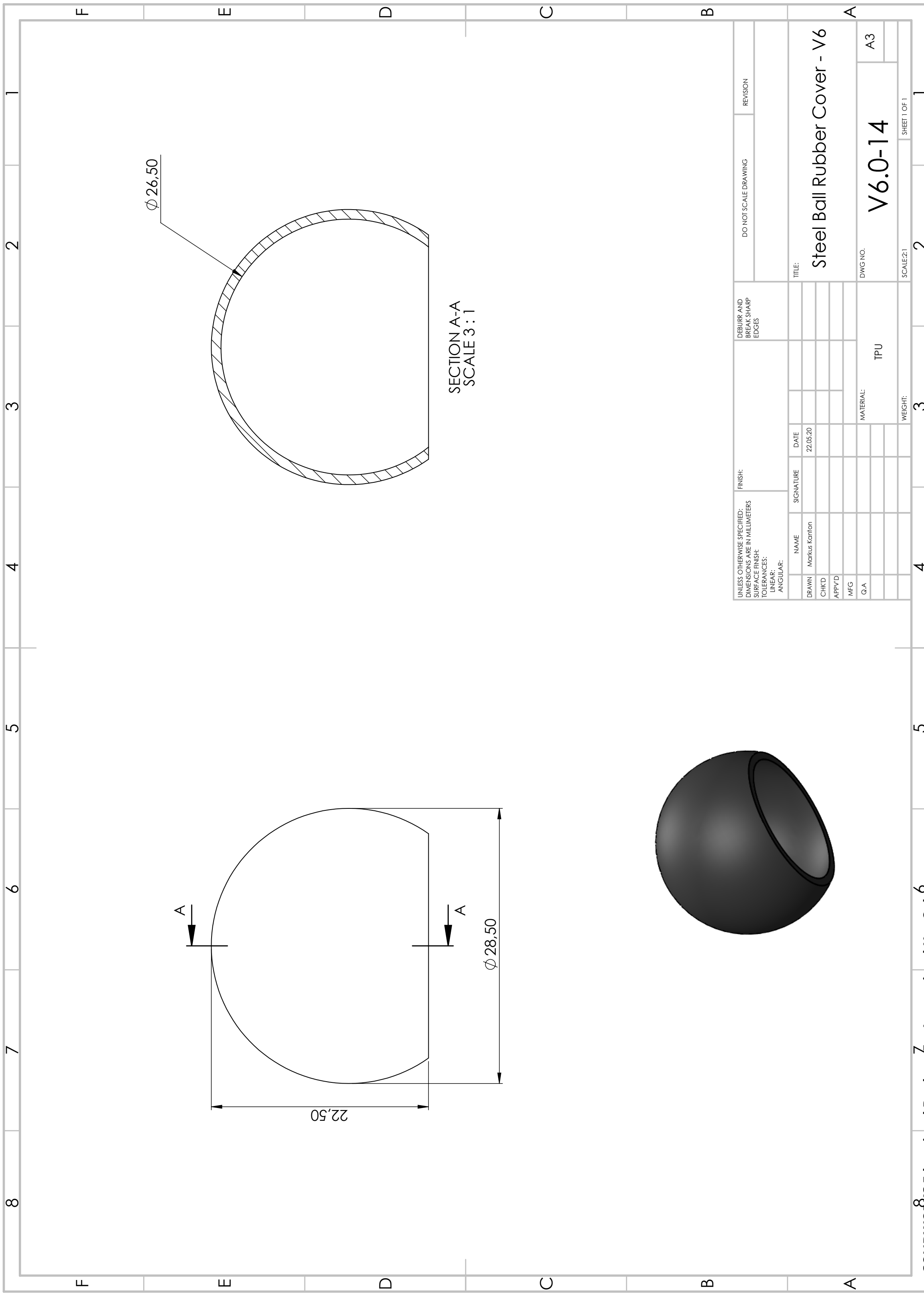
Pole Clamp Ring - Small - V6
DWG NO. V6.0-10
A3
SCALE:2:1
SHEET 1 OF 1
MATERIAL: PLA
WEIGHT:
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN Markus Kanton 22.05.20
CHK'D
APPV'D
MFG
Q.A
SECTION A-A
3,90
Ø31,20
Ø1,50
Ø27,20
23,00
18,80
A
A
SOLIDWORKS Educational Product. For Instructional Use Only.


200,00 ±0,20
∅17,00
∅20,00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
22.05.20
TITLE:
Ø20x1,5mm Steel Pole - V6
MATERIAL:
St-37
DWG NO.
V6.0-11
A3
SCALE:1:1
SHEET 1 OF 1
WEIGHT:
SOLIDWORKS Educational Product. For Instructional Use Only.


Ø18, M12 Insert
V6.0-12
TITLE:
DWG NO.
A3
SCALE:5:1
SHEET 1 OF 1
MATERIAL:
Sis2541
WEIGHT:
DO NOT SCALE DRAWING
REVISION
DEBURR AND BREAK SHARP EDGES
FINISH:
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
22.05.20
CHK'D
APPV'D
MFG
Q.A
M12x1.25
15,00 ±0,10
Ø17,00 K13 0,00 +0,27
SOLIDWORKS Educational Product. For Instructional Use Only.



Steel Ball Plug - V6
V6.0-13
TITLE:
DWG NO.
A3
SCALE:2:1
SHEET 1 OF 1
MATERIAL:
St-37
WEIGHT:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
FINISH:
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
22.05.20
CHK'D
APPV'D
MFG
Q.A
R5,00
M12,00 x 1.25
Ø27,50 ±0,10
50,00 +0,20 0,00
SOLIDWORKS Educational Product. For Instructional Use Only.


SECTION A-A
SCALE 3 : 1
Ø 26,50
Ø 28,50
22,50
A
A
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
Markus Kanton
22.05.20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
TPU
WEIGHT:
TITLE:
Steel Ball Rubber Cover - V6
DWG NO.
V6.0-14
A3
SCALE:2:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.