# Risk analysis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Code** | **Risk** **event** | **Consequence** | **Measures** | **P** | **C** | **Priority** |
| R1 | If multiple get sick at the same time? | Delay and loss of work.    More work for those who are available.    High stress level. | Minimize loss of work by saving often.    Work a few extra hours during that period and actively use Taiga to distribute tasks to stay on track.    Lower stress levels by having good communication and planning within the group. | Unlikely | Critical | 10 |
| R2 | If someone suddenly wants to quit? | Delay, poor solutions, and lower trust within the group. | Being honest, follow-up in stand-up meetings, and actively using out dashboard. | Rare | Critical | 5 |
| R3 | Various problems with group work. | Delays, poor communication, attendance. | Good work method, communication, and status meetings.    Social gathering. | Moderate | Major | 12 |
| R4 | The group is not able to achieve an optimal solution as the client envisioned. | Bad results (grade / further understanding of the problem).    Bad conscience / self-confidence. | Maintain good communication with the client.    Have a good working process so that any errors can be detected early. | Unlikely | Critical | 10 |
| R5 | New nationwide pandemic. | Lockdown so that we do not meet physically.    The chance of more people getting infected and becoming sick. | Work remotely, use Teams and other tools available actively.    Have good hygiene habits. | Unlikely | Major | 8 |
| R6 | Global component shortages. | Delays in work/test and results. | Ask the client about components / alternative components.    Find alternative solutions. | Rare | Insignificant | 1 |
| R7 | The war in Ukraine. | Increased fuel prices cause hesitation to drive to school. | Driving together or using public transportation. | Unlikely | Minor | 4 |
| R8 | Issues related to software drivers and versions | We may be unable to answer the customer's | Use older versions | Likely | Minor | 8 |

**OpenCV**

OpenCV is an open-source computer vision and machine learning library that is used to analyze and manipulate visual data in real-time. It is used for a wide range of applications, including image and video processing, facial recognition, object detection, and tracking. OpenCV provides a set of powerful tools and algorithms which make it easy to work with images and videos, including tools for image filtering, feature detection, and machine learning. It is widely used in industries like robotics, surveillance, and automotive. OpenCV is particularly useful with Python due to its ease of use, making it a popular choice for developers and researchers who want to work with computer vision applications.

OpenCV has numerous advantages and some of them are:

* Open-source and free to use and modify
* Supported by a vast community of developers
* Highly efficient with optimized algorithms for real-time visual data processing

However, there are some drawbacks to using OpenCV including:

* Steep learning curve, especially for those new to computer vision and machine learning
* Somewhat complex to integrate with certain systems
  + Depend on the specific requirements and capabilities of the system being used
* Primarily a low-level library, challenging to use for higher-level applications

Despite these limitations, OpenCV remains a powerful and versatile tool for visual data analysis and manipulation.

There are different techniques used in OpenCV and some of them are

**Blob detection:**

In OpenCV, "blobs" refer to groups of connected pixels that share similar characteristics, such as color or texture. Blob detection is a common computer vision technique used for image segmentation, object tracking, and feature extraction.

<https://learnopencv.com/blob-detection-using-opencv-python-c/>

**Contour detection:**  
Contours refer to the boundaries of objects in an image or video. They are often used for object detection, recognition, and classification. In OpenCV, contours can be found using the "findContours()" function, which detects and extracts the boundaries of objects in an image.

<https://learnopencv.com/contour-detection-using-opencv-python-c/>

More information about techniques used in OpenCV can be found here:

<https://learnopencv.com/getting-started-with-opencv/>

Kilder:

Risk:  
<https://safetyculture.com/topics/risk-assessment/5x5-risk-matrix/>

OpenCV:

<https://www.geeksforgeeks.org/opencv-overview/>

<https://mobilunity.com/blog/hire-opencv-developers/>

<https://www.slideserve.com/zanta/tutorial-on-matlab-and-opencv>