

Software Risk Management Plan – Sample

1. Introduction

This document describes the risk management plan for the project titled “**Camera-Based Collision Detection and Distance Measurement System using Python.**” The purpose of this document is to identify potential risks that may occur during the design, development, testing, and deployment of the system and to define strategies to minimize or control those risks.

The system uses a camera to detect multiple bodies, identify collision using bounding box intersection, calculate the distance between bodies using Euclidean distance, and display alerts in real time.

Effective risk management ensures smooth project execution and successful completion within time and resource constraints.

2. Risk Identification

The following types of risks have been identified:

2.1 Technical Risks

- Inaccurate object detection due to poor lighting conditions.
- Errors in collision detection logic.
- Incorrect distance calculation due to pixel-to-real-world conversion issues.
- Camera hardware malfunction or failure.
- Performance issues in real-time video processing.

2.2 Schedule Risks

- Delay in coding due to lack of experience with computer vision libraries.
- Time required for debugging and testing may exceed planned schedule.
- Dependency on team coordination for task completion.

2.3 Resource Risks

- Unavailability of required hardware (camera or system).
- Limited computing power (low RAM or CPU).

- Lack of access to stable internet (for updates or libraries).

2.4 Cost Risks

- Additional cost for better camera or system hardware.
- Unexpected software tool or library dependency costs (if any).

2.5 Operational Risks

- System failure during live testing.
- User misunderstanding of system output.
- False collision detection in crowded scenes.

3. Risk Analysis

Each identified risk is analyzed based on **probability** and **impact**:

| Risk Type | Description | Probability | Impact |
|-------------|----------------------------|-------------|--------|
| Technical | Object detection error | Medium | High |
| Technical | Distance calculation error | Medium | Medium |
| Schedule | Delay in development | Medium | High |
| Resource | Camera malfunction | Low | Medium |
| Operational | False collision detection | Medium | High |

Probability Levels: Low / Medium / High

Impact Levels: Low / Medium / High

4. Risk Mitigation

Strategies to reduce or eliminate risks:

4.1 Technical Risk Mitigation

- Use reliable libraries like OpenCV and NumPy.
- Perform multiple test runs under different lighting conditions.
- Calibrate camera for accurate distance measurement.

- Use bounding box center points for distance calculation.

4.2 Schedule Risk Mitigation

- Divide project into small modules (camera input, detection, collision logic, distance calculation).
- Follow weekly milestones.
- Regular progress reviews.

4.3 Resource Risk Mitigation

- Use backup camera device if primary one fails.
- Run project on systems with minimum required hardware.
- Keep offline copies of required libraries.

4.4 Operational Risk Mitigation

- Add threshold values to avoid false collision detection.
- Display warnings instead of final alerts for uncertain collisions.
- Provide simple user interface with clear messages.

4.5 Security & Privacy Mitigation

- Do not store unnecessary video data.
- Restrict access to system.
- Use only local camera feed for testing.

5. Risk Monitoring

Risk monitoring is done continuously throughout the project lifecycle.

5.1 Monitoring Activities

- Weekly review meetings to discuss risks.
- Testing system performance regularly.
- Monitoring accuracy of collision detection.
- Checking camera hardware and software updates.

5.2 Risk Review Process

- New risks will be added when identified.
- Existing risks will be re-evaluated after mitigation actions.
- Documentation will be updated accordingly.

5.3 Responsibility

- Project team members are responsible for reporting risks.
- Team leader monitors overall project risk.
- Mentor reviews risk handling strategies.

6. Conclusion

Risk management plays a crucial role in the successful development of the **Camera-Based Collision Detection System**.

By identifying risks early and applying proper mitigation strategies, the project ensures reliability, accuracy, and timely completion.

Continuous monitoring and evaluation will help minimize technical failures and improve overall system performance.