Software Requirements Specification for SE 4G06: subtitle describing software

Team 3, Tangle
Calvyn Siong
Cyruss Allen Amante
Edward Gao
Richard Li
Mark Angelo Cruz

October 6, 2024

Contents

1 Purpose of the Project **vi**
- 1.1 User Business vi
- 1.2 Goals of the Project vi

2 Stakeholders **vi**
- 2.1 Client vi
- 2.2 Customer vi
- 2.3 Other Stakeholders vi
- 2.4 Hands-On Users of the Project vi
- 2.5 Personas vi
- 2.6 Priorities Assigned to Users vi
- 2.7 User Participation vii
- 2.8 Maintenance Users and Service Technicians vii

3 Mandated Constraints **vii**
- 3.1 Solution Constraints vii
- 3.2 Implementation Environment of the Current System vii
- 3.3 Partner or Collaborative Applications vii
- 3.4 Off-the-Shelf Software vii
- 3.5 Anticipated Workplace Environment vii
- 3.6 Schedule Constraints vii
- 3.7 Budget Constraints vii
- 3.8 Enterprise Constraints viii

4 Naming Conventions and Terminology **viii**
- 4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project viii

5 Relevant Facts And Assumptions **viii**
- 5.1 Relevant Facts viii
- 5.2 Business Rules viii
- 5.3 Assumptions viii

6 The Scope of the Work **viii**
- 6.1 The Current Situation viii
  - 6.1.1 Software Engineering Practices viii
  - 6.1.2 Physics Engine Integration ix

6.2 The Context of the Work x
6.2.1 Software Engineering Practices x
6.2.2 Physics Engine Integration xi
6.3 Work Partitioning xii
6.4 Specifying a Business Use Case (BUC) xiv

7 Business Data Model and Data Dictionary **xiv**
7.1 Business Data Model xiv
7.2 Data Dictionary xiv

8 The Scope of the Product **xv**
8.1 Product Boundary xv
8.2 Product Use Case Table xv
8.3 Individual Product Use Cases (PUC's) xv

9 Functional Requirements **xv**
9.1 Functional Requirements xv

10 Look and Feel Requirements **xv**
10.1 Appearance Requirements xv
10.2 Style Requirements xv

11 Usability and Humanity Requirements **xv**
11.1 Ease of Use Requirements xv
11.2 Personalization and Internationalization Requirements xvi
11.3 Learning Requirements xvi
11.4 Understandability and Politeness Requirements xvi
11.5 Accessibility Requirements xvi

12 Performance Requirements **xvi**
12.1 Speed and Latency Requirements xvi
12.2 Safety-Critical Requirements xvi
12.3 Precision or Accuracy Requirements xvi
12.4 Robustness or Fault-Tolerance Requirements xvi
12.5 Capacity Requirements xvi
12.6 Scalability or Extensibility Requirements xvii
12.7 Longevity Requirements xvii

13 Operational and Environmental Requirements **xvii**
13.1 Expected Physical Environment xvii
13.2 Wider Environment Requirements xvii
13.3 Requirements for Interfacing with Adjacent Systems xvii
13.4 Productization Requirements xvii
13.5 Release Requirements xvii

14 Maintainability and Support Requirements **xvii**
14.1 Maintenance Requirements xvii
14.2 Supportability Requirements xviii
14.3 Adaptability Requirements xviii

15 Security Requirements **xviii**
15.1 Access Requirements xviii
15.2 Integrity Requirements xviii
15.3 Privacy Requirements xviii
15.4 Audit Requirements xviii
15.5 Immunity Requirements xviii

16 Cultural Requirements **xviii**
16.1 Cultural Requirements xviii

17 Compliance Requirements **xix**
17.1 Legal Requirements xix
17.2 Standards Compliance Requirements xix

18 Open Issues **xix**

19 Off-the-Shelf Solutions **xix**
19.1 Ready-Made Products xix
19.2 Reusable Components xix
19.3 Products That Can Be Copied xix

20 New Problems **xix**
20.1 Effects on the Current Environment xix
20.2 Effects on the Installed Systems xix
20.3 Potential User Problems xx
20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product xx

20.5 Follow-Up Problems xx

21 Tasks **xx**

21.1 Project Planning xx

21.2 Planning of the Development Phases xx

22 Migration to the New Product **xx**

22.1 Requirements for Migration to the New Product xx

22.2 Data That Has to be Modified or Translated for the New System xx

23 Costs **xxi**

24 User Documentation and Training **xxi**

24.1 User Documentation Requirements xxi

24.2 Training Requirements xxi

25 Waiting Room **xxi**

26 Ideas for Solution **xxi**

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Purpose of the Project

1.1 User Business

Insert your content here.

1.2 Goals of the Project

Insert your content here.

2 Stakeholders

2.1 Client

Insert your content here.

2.2 Customer

Insert your content here.

2.3 Other Stakeholders

Insert your content here.

2.4 Hands-On Users of the Project

Insert your content here.

2.5 Personas

Insert your content here.

2.6 Priorities Assigned to Users

Insert your content here.

2.7 User Participation

Insert your content here.

2.8 Maintenance Users and Service Technicians

Insert your content here.

3 Mandated Constraints

3.1 Solution Constraints

Insert your content here.

3.2 Implementation Environment of the Current System

Insert your content here.

3.3 Partner or Collaborative Applications

Insert your content here.

3.4 Off-the-Shelf Software

Insert your content here.

3.5 Anticipated Workplace Environment

Insert your content here.

3.6 Schedule Constraints

Insert your content here.

3.7 Budget Constraints

Insert your content here.

3.8 Enterprise Constraints

Insert your content here.

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

Insert your content here.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

Insert your content here.

5.2 Business Rules

Insert your content here.

5.3 Assumptions

Insert your content here.

6 The Scope of the Work

6.1 The Current Situation

For the scope of this capstone, there are two main areas of focus: Software Engineering practices and physics engine integration. Both of these have current states which will be improved upon over the course of this project.

6.1.1 Software Engineering Practices

The TPG (Tangled Program Graph) project is currently managed with basic software engineering practices. The codebase is hosted on GitLab, and Dr.

Kelly's research group uses Git branches to separate and manage their work. This current setup does allow for parallel work and multiple contributors. However, several key practices are missing:

- Unit Testing: There are no unit tests in place. This absence means that code changes are not systematically validated, increasing the risk of introducing bugs and regressions.
- Continuous Integration/Continuous Deployment (CI/CD): The project lacks automated pipelines for building, testing, and deploying code. Without CI/CD, integrating changes can be time-consuming and error-prone.
- Pull Request Templates and Standards: There are no standardized templates or guidelines for pull requests, leading to inconsistencies in code reviews and collaboration.
- Open Source License: The project has not yet adopted an open-source license, which can deter external contributions and limit the software's usage.
- Issue Management: There is no formal system for tracking bugs, feature requests, or tasks, making project management less efficient.

The current structure of the TPG codebase may not be optimized for use as an open-source library. Current researchers need to run shell scripts as the entry point, which can be a barrier to entry for those unfamiliar with the system. Additionally, MacOS with ARM based chips may have extra difficulty in onboarding to this project due to the many dependency conflicts in this current state. This approach limits accessibility and may discourage potential users and contributors. More research into code structure and analysis of how other open source libraries allow for their frameworks to be integrated into open source contributor workflows should be studied.

6.1.2 Physics Engine Integration

The TPG framework has been validated using OpenAI Gym's classic control problems such as CartPole, Acrobot, and Pendulum. These environments are stationary, meaning their transition functions (the rules determining the next state given a current state and action) do not change over time. In contrast, real-world environments are typically non-stationary. Their transition functions can evolve due to external factors, requiring agents to adapt continuously. Currently, the TPG framework needs to evolve and be adapted to more dynamic environments.

6.2 The Context of the Work

6.2.1 Software Engineering Practices

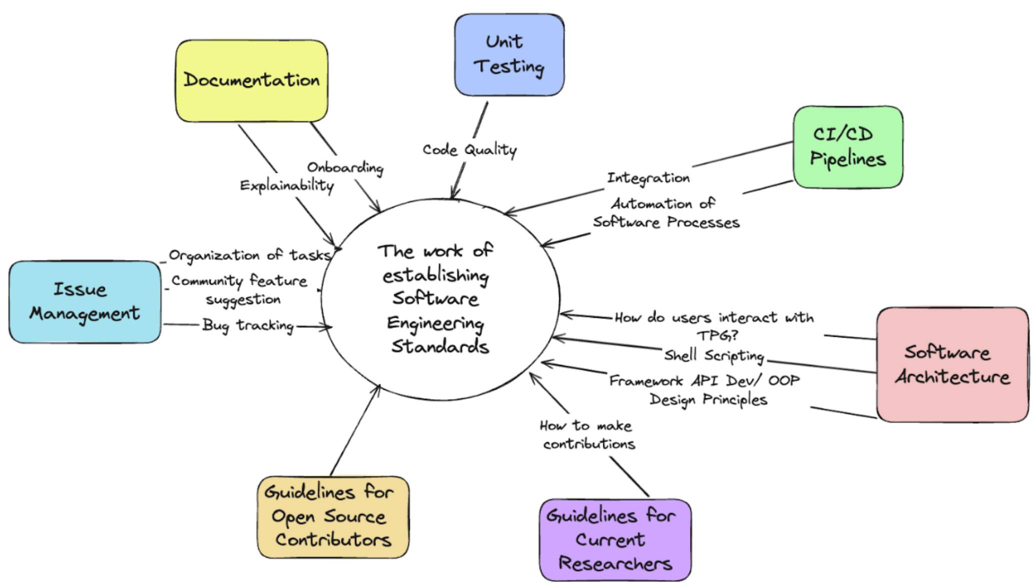

Figure 1: System Context

- Unit Testing: Focused on improving code quality, unit tests ensure that individual pieces of the codebase work as expected.
- CI/CD Pipelines: The integration and automation of software processes, such as building and testing, will allow for continuous integration and deployment of changes, ensuring the project remains robust as it scales.

- Software Architecture: This examines how users interact with TPG, the use of shell scripting, and how the framework is structured using API development and object-oriented programming (OOP) principles. It emphasizes improving design principles for a better developer and user experience.

- Documentation: This is critical for onboarding new developers, ensuring explainability, and providing clear, thorough project documentation.

- Issue Management: Introducing a formal system to track bugs, feature requests, and community suggestions will help organize tasks and streamline project development.

- Guidelines for Open Source Contributors: Establishing clear guidelines will provide a roadmap for external contributors to participate in the project, increasing collaboration and contributions.

- Guidelines for Current Researchers: This component covers how researchers and developers within the project can contribute effectively, ensuring consistency and alignment with the project's goals.

6.2.2 Physics Engine Integration

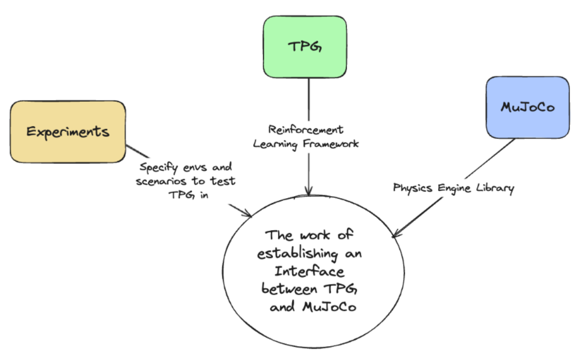

Figure 2: System Context

- TPG (Tangled Program Graphs): TPG is a reinforcement learning framework. In this context, its role is to act as the core system that will be tested and integrated with dynamic environments. Establishing the interface between TPG and MuJoCo will allow the TPG framework to be tested in physically realistic simulations.
- MuJoCo (Multi-Joint dynamics with Contact): MuJoCo is a high-performance physics engine used for modeling and simulating dynamic systems. In this diagram, it represents the external library that provides the physics-based environments required for testing TPG. The interface with TPG will enable the reinforcement learning agents created using TPG to interact with complex, real-world-like physics simulations provided by MuJoCo.
- Experiments: Experiments define the environments and scenarios where TPG will be tested. This component represents the experimental setups that specify the parameters for evaluating TPG's performance in various MuJoCo-based scenarios. These experiments are critical for determining how well TPG adapts to different physics-based tasks, environments, and scenarios.

6.3 Work Partitioning

Table 1: A sample long table.

Event Name	**Inputs**	**Outputs**	**Summary**

Continued on next page

Table 1 – continued from previous page

Event Name	Inputs	Outputs	Summary
Contributor wants to merge code changes they made	New changes (code that was modified in a PR)	New code is successfully integrated with the main code	CI - Continuous Integration practices into the repo, ensuring all devs can make changes in a seamless manner and be up to date while concurrent work is occuring
Contributor wants to evaluate the code they wrote	New code blocks that are written by a contributor	Test functions are created to evaluate new code	Automated tests are generated for new code blocks that are written ensuring code robustness
Contributor wants to onboard and use TPG	N/A	Contributor is able to run the framework	Seamless onboarding experience that allows a new contributor/user to get up and running

Continued on next page

Table 1 – continued from previous page

Event Name	Inputs	Outputs	Summary
Contributor wants to perform an experiment to test the TPG framework in MuJoCo	TPG, MuJoCo	Functioning simulation of an experiment integrated with a physics engine	New experiments to be conducted in more realistic scenarios evolving the development of this reinforcement learning framework
Contributor wants to get visual results of from test data	TPG, MuJoCo	Graphs of the recent experiment	Evaluation of the experiment is crucial to improving the framework and allowing it to become good at multi task reinforcement learning tasks

6.4 Specifying a Business Use Case (BUC)

Insert your content here.

7 Business Data Model and Data Dictionary

7.1 Business Data Model

Insert your content here.

7.2 Data Dictionary

Insert your content here.

8 The Scope of the Product

8.1 Product Boundary

Insert your content here.

8.2 Product Use Case Table

Insert your content here.

8.3 Individual Product Use Cases (PUC's)

Insert your content here.

9 Functional Requirements

9.1 Functional Requirements

Insert your content here.

10 Look and Feel Requirements

10.1 Appearance Requirements

Insert your content here.

10.2 Style Requirements

Insert your content here.

11 Usability and Humanity Requirements

11.1 Ease of Use Requirements

Insert your content here.

11.2 Personalization and Internationalization Requirements

Insert your content here.

11.3 Learning Requirements

Insert your content here.

11.4 Understandability and Politeness Requirements

Insert your content here.

11.5 Accessibility Requirements

Insert your content here.

12 Performance Requirements

12.1 Speed and Latency Requirements

Insert your content here.

12.2 Safety-Critical Requirements

Insert your content here.

12.3 Precision or Accuracy Requirements

Insert your content here.

12.4 Robustness or Fault-Tolerance Requirements

Insert your content here.

12.5 Capacity Requirements

Insert your content here.

12.6 Scalability or Extensibility Requirements

Insert your content here.

12.7 Longevity Requirements

Insert your content here.

13 Operational and Environmental Requirements

13.1 Expected Physical Environment

Insert your content here.

13.2 Wider Environment Requirements

Insert your content here.

13.3 Requirements for Interfacing with Adjacent Systems

Insert your content here.

13.4 Productization Requirements

Insert your content here.

13.5 Release Requirements

Insert your content here.

14 Maintainability and Support Requirements

14.1 Maintenance Requirements

Insert your content here.

14.2 Supportability Requirements

Insert your content here.

14.3 Adaptability Requirements

Insert your content here.

15 Security Requirements

15.1 Access Requirements

Insert your content here.

15.2 Integrity Requirements

Insert your content here.

15.3 Privacy Requirements

Insert your content here.

15.4 Audit Requirements

Insert your content here.

15.5 Immunity Requirements

Insert your content here.

16 Cultural Requirements

16.1 Cultural Requirements

Insert your content here.

17 Compliance Requirements

17.1 Legal Requirements

Insert your content here.

17.2 Standards Compliance Requirements

Insert your content here.

18 Open Issues

Insert your content here.

19 Off-the-Shelf Solutions

19.1 Ready-Made Products

Insert your content here.

19.2 Reusable Components

Insert your content here.

19.3 Products That Can Be Copied

Insert your content here.

20 New Problems

20.1 Effects on the Current Environment

Insert your content here.

20.2 Effects on the Installed Systems

Insert your content here.

20.3 Potential User Problems

Insert your content here.

20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Insert your content here.

20.5 Follow-Up Problems

Insert your content here.

21 Tasks

21.1 Project Planning

Insert your content here.

21.2 Planning of the Development Phases

Insert your content here.

22 Migration to the New Product

22.1 Requirements for Migration to the New Product

Insert your content here.

22.2 Data That Has to be Modified or Translated for the New System

Insert your content here.

23 Costs

Insert your content here.

24 User Documentation and Training

24.1 User Documentation Requirements

Insert your content here.

24.2 Training Requirements

Insert your content here.

25 Waiting Room

Insert your content here.

26 Ideas for Solution

Insert your content here.

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?