System Verification and Validation Plan for Software Engineering

Team #22, TeleHealth Insights
Mitchell Weingust
Parisha Nizam
Promish Kandel
Jasmine Sun-Hu

November 4, 2024

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

[The intention of the VnV plan is to increase confidence in the software. However, this does not mean listing every verification and validation technique that has ever been devised. The VnV plan should also be a **feasible** plan. Execution of the plan should be possible with the time and team available. If the full plan cannot be completed during the time available, it can either be modified to "fake it", or a better solution is to add a section describing what work has been completed and what work is still planned for the future. —SS]

[The VnV plan is typically started after the requirements stage, but before the design stage. This means that the sections related to unit testing cannot initially be completed. The sections will be filled in after the design stage is complete. the final version of the VnV plan should have all sections filled in. —SS]

Contents

List of Tables

[Remove this section if it isn't needed —SS]

List of Figures

[Remove this section if it isn't needed —SS]

1 Symbols, Abbreviations, and Acronyms

symbol	description
T	Test

[symbols, abbreviations, or acronyms — you can simply reference the SRS (Author, 2019) tables, if appropriate —SS]

[Remove this section if it isn't needed —SS]

This document ... [provide an introductory blurb and roadmap of the Verification and Validation plan —SS]

2 General Information

2.1 Summary

[Say what software is being tested. Give its name and a brief overview of its general functions. —SS]

The software being tested TeleHealth Insights, is a web-based at-home bilingual speech assessment system with video and audio analysis features. The system is designed to provide clear guidance to parents when administering the assessment to their children, in an environment where speech-language pathologists (SLPs) are unavailable. By streamlining the assessment pro- cess, the project aims to provide a convenient and comprehensive solution for SLPs to assess and support their patient's speech and language development remotely.

TeleHealth System comprises of several software components to efficiently provide a platform for children assessments to occur while actively recording for clinicians to review and analyze.

2.1.1 User Interface

We must ensure that the interface is easy to navigate for all stakeholders involved, the parents, clinicians as well as the parents. The system should be simple and testing will include ensuring there is an intuitive layouts, clear instructions and labels, as well as a logical workflow.

2.1.2 Video and Audio Recording

Ensure set up of hardware devices used on application is intuitive and done properly.

2.1.3 Video and Audio Analysis

Integrate web-based application to store recorded audio and video footage. The system will be tested to ensure it is able to appropriately analyze and

identify flags and bias when conducting assessments. The analyzed data should be presented in a easy to understand representation

2.2 Objectives

The Vnv Plan will focus on the following objectives

2.2.1 Ensure Accuracy and Correctness of Software Reading and Identifying Bias

The main objective of this VnV Plan is ensure correctness of the system's ability to record video and audio of our clients completing speech therapy assessments, store and train an AI model to to be able to detect any disturbances and bias in the assessment taking process. A report or dashboard should be made to present its findings and ensure the results are accurate.

2.2.2 Security and Authentication

Second main objective of this VnV plan is to ensure security and strong authentication capabilities of the software.

- Check security and anonymity of all patient data (recording), ensuring its secure storage of information and access.
- Check software correctness, ensuring authorized users are allowed in the system, and their appropriate role.

  Parents should be able to view the assessment and complete it with their child.

  Clinicians should be able to view assessment results and recordings

  Admins should be able to view and create accounts.
- Check the software correctly blocked unauthorized users and grants them access to the assessment portal and assessment results.

2.2.3 Demonstrate Usability

The last main object of the VnV plan is to check that the application is easy and intuitive to use. The main goal of the system is to create an application that stores correct results and allows users to complete their task quickly

and efficiently. Given that the system will be used by parents, children, and medical professionals, the UI needs to be easy to understand and use.

Out of scope Objectives

[State what is intended to be accomplished. The objective will be around the qualities that are most important for your project. You might have something like: "build confidence in the software correctness," "demonstrate adequate usability." etc. You won't list all of the qualities, just those that are most important. —SS]

[You should also list the objectives that are out of scope. You don't have the resources to do everything, so what will you be leaving out. For instance, if you are not going to verify the quality of usability, state this. It is also worthwhile to justify why the objectives are left out. —SS]

[The objectives are important because they highlight that you are aware of limitations in your resources for verification and validation. You can't do everything, so what are you going to prioritize? As an example, if your system depends on an external library, you can explicitly state that you will assume that external library has already been verified by its implementation team. —SS]

2.3 Challenge Level and Extras

[State the challenge level (advanced, general, basic) for your project. Your challenge level should exactly match what is included in your problem statement. This should be the challenge level agreed on between you and the course instructor. You can use a pull request to update your challenge level (in TeamComposition.csv or Repos.csv) if your plan changes as a result of the VnV planning exercise. —SS]

[Summarize the extras (if any) that were tackled by this project. Extras can include usability testing, code walkthroughs, user documentation, formal proof, GenderMag personas, Design Thinking, etc. Extras should have already been approved by the course instructor as included in your problem statement. You can use a pull request to update your extras (in TeamComposition.csv or Repos.csv) if your plan changes as a result of the VnV planning exercise. —SS]

2.3.1 Challenge Level

Our challenge level is general as the project scope is limited in terms of how much research is required. The required domain knowledge is basic web-design in a stack of our choice. We are also planning on using open-source large language models for audio and video processing.

2.3.2 Extras

Our project has the following extras;

- **User Documentation:** Providing users with a guide on how to use and better understand the system.
- **Usability Testing:** Receive user feedback on usability of design of the ap- plication, including improvements on how the system looks and functions. This will help to ensure an intuitive and easy system for both clinicians and parents/children to navigate through.

2.4 Relevant Documentation

[Reference relevant documentation. This will definitely include your SRS and your other project documents (design documents, like MG, MIS, etc). You can include these even before they are written, since by the time the project is done, they will be written. You can create BibTeX entries for your documents and within those entries include a hyperlink to the documents. —SS]

The two main relevant documentation that helped guide us in System Verification and Validation Plan (VnV) was Software Requirements Specification (SRS) and Development plan.

- The SRS document was crucial to define descriptions, rationals and fit criteria for all of the systems main functional and non functional requirements. This outlines the basis of what needs to be verified and tested in the VnV Plan.
- Development Plan: Defined the main Goals, Objectives and extras for the system, aiding the team to understand what needs to be done and focused on.

Author (2019)

[Don't just list the other documents. You should explain why they are relevant and how they relate to your VnV efforts. —SS]

3 Plan

[Introduce this section. You can provide a roadmap of the sections to come. —SS]

3.1 Verification and Validation Team

[Your teammates. Maybe your supervisor. You should do more than list names. You should say what each person's role is for the project's verification. A table is a good way to summarize this information. —SS]

3.2 SRS Verification Plan

[List any approaches you intend to use for SRS verification. This may include ad hoc feedback from reviewers, like your classmates (like your primary reviewer), or you may plan for something more rigorous/systematic. —SS]

[If you have a supervisor for the project, you shouldn't just say they will read over the SRS. You should explain your structured approach to the review. Will you have a meeting? What will you present? What questions will you ask? Will you give them instructions for a task-based inspection? Will you use your issue tracker? —SS]

[Maybe create an SRS checklist? —SS]

3.3 Design Verification Plan

[Plans for design verification —SS]

[The review will include reviews by your classmates —SS]

[Create a checklists? —SS]

The design verification plan outlines the strategies that the team will use to verify the usability and correctness of our system. The following outlines the plan for verifying the User Interface of the software.

- Shall conduct a quick review with supervisor after design documents (MIS) have been completed

- Conduct peer review sessions from classmates to provide critical suggestions on improvements of the system.
- Conduct final formal review with clinician Dr. Du and the team, following defined SRS and MIS documents.

The following checklist will be used to verify the system's design documents
❑Are all requirements traceable to at least one feature/module in the MIS?
❑Do all modules and components follow SOLID design principles?
❑Have the creation of all modules been tracked via issues and closed once reviewed?

3.4 Verification and Validation Plan Verification Plan

[The verification and validation plan is an artifact that should also be verified. Techniques for this include review and mutation testing. —SS]

[The review will include reviews by your classmates —SS]

[Create a checklists? —SS]

3.5 Implementation Verification Plan

[You should at least point to the tests listed in this document and the unit testing plan. —SS]

[In this section you would also give any details of any plans for static verification of the implementation. Potential techniques include code walkthroughs, code inspection, static analyzers, etc. —SS]

[The final class presentation in CAS 741 could be used as a code walkthrough. There is also a possibility of using the final presentation (in CAS741) for a partial usability survey. —SS]

The following outlines the plan for verifying implementation of the system, using both static and dynamic techniques

- Walkthrough of key components of the system with the supervisor (UI design, user authentication, video and audio analysis)
- Walkthrough of each components with other teammates

- Running unit tests (to be implemented in VnV Plan) to verify that the implementation matches the specified design outlined in MIS. This will be done automatically (dynamic) using Github Actions for each pull request made, following
- Running system tests (both functional and non functional) described in the VnV plan to verify that the implementation meets requirements
- Running Static analyzers including linters to help discover any potential errors or bugs in the code. Linters and tools will also be used to help make the code more organized and readable.
- Major code commits will be reviewed by at least one other team member before merging to main branch to ensure consistency and minimize chance of conflicts.

3.6 Automated Testing and Verification Tools

[What tools are you using for automated testing. Likely a unit testing framework and maybe a profiling tool, like ValGrind. Other possible tools include a static analyzer, make, continuous integration tools, test coverage tools, etc. Explain your plans for summarizing code coverage metrics. Linters are another important class of tools. For the programming language you select, you should look at the available linters. There may also be tools that verify that coding standards have been respected, like flake9 for Python. —SS]

[If you have already done this in the development plan, you can point to that document. —SS]

[The details of this section will likely evolve as you get closer to the implementation. —SS]

3.7 Software Validation Plan

[If there is any external data that can be used for validation, you should point to it here. If there are no plans for validation, you should state that here. —SS]

[You might want to use review sessions with the stakeholder to check that the requirements document captures the right requirements. Maybe task based inspection? —SS]

[For those capstone teams with an external supervisor, the Rev 0 demo should be used as an opportunity to validate the requirements. You should plan on demonstrating your project to your supervisor shortly after the scheduled Rev 0 demo. The feedback from your supervisor will be very useful for improving your project. —SS]

[For teams without an external supervisor, user testing can serve the same purpose as a Rev 0 demo for the supervisor. —SS]

[This section might reference back to the SRS verification section. —SS]

4 System Tests

[There should be text between all headings, even if it is just a roadmap of the contents of the subsections. —SS]

4.1 Tests for Functional Requirements

4.1.1 System Set Up

- FR-ST-SS1

Control: Manual

Initial State: User is logged into the system and has access to the assessment information page.

Input: User navigates to the page where assessment information is displayed before starting hardware checks.

Output: User is able to view relevant information about the assessment before beginning any hardware checks.

Test Case Derivation: Users need to be aware of the assessment requirements and goals prior to starting the hardware setup to ensure they are prepared.

How the test will be performed:

1. Log in as a user and navigate to the assessment information section.
2. Verify that the page displays necessary details about the assessment, such as the purpose, requirements, and overview.

3. Confirm that the information is accessible and readable to the user.

- FR-ST-SS2

Control: Manual

Initial State: User is on the hardware check section of the system with audio devices connected.

Input: User initiates the audio hardware check through the system.

Output: User receives confirmation that the audio input and output devices are functioning correctly.

Test Case Derivation: Verifying that audio devices are functional before the assessment prevents issues of recording audio during the assessment, which could effect proper identification of bias.

How the test will be performed:

1. Start the audio hardware check.
2. Confirm that the system prompts the user to test both audio input (microphone) and output (speakers/headphones).
3. Verify that the system indicates successful audio detection when the test is performed.
4. Test with a non-functional audio device and confirm that the system displays an appropriate error.

- FR-ST-SS3

Control: Manual

Initial State: User is on the hardware check section of the system with a video device connected.

Input: User initiates the video hardware check through the system.

Output: User receives confirmation that the video capturing device is functioning correctly.

Test Case Derivation: Ensuring video functionality prevents disruptions in assessments that require visual input or interaction and ensures proper recording is taken for accurate data analysis

How the test will be performed:

1. Start the video hardware check.
2. Confirm that the system activates the video device and displays a live feed.
3. Verify that the system confirms successful video detection if the feed is displayed correctly.
4. Test with a non-functional video device and confirm that the system displays an error.

- FR-ST-SS4

Control: Manual

Initial State: User has successfully completed audio and video hardware checks.

Input: User proceeds to the tutorial section after completing hardware checks.

Output: User is directed to a tutorial that explains the assessment process in a step-by-step manner.

Test Case Derivation: Providing a tutorial ensures that users understand the assessment process, improving accuracy and compliance.

How the test will be performed:

1. Complete the audio and video hardware checks.
2. Verify that the system automatically navigates the user to a tutorial section upon completion of the hardware checks.
3. Confirm that the tutorial provides clear, step-by-step instructions on how to proceed with the assessment.

- FR-ST-SS5

Control: Manual

Initial State: User has completed the tutorial and is ready to begin the assessment.

Input: User initiates the start of the assessment through the system.

Output: User is taken to the first assessment question, and the assessment begins.

Test Case Derivation: Allowing users to start the assessment on their own terms helps them feel prepared and reduces errors.

How the test will be performed:

1. After completing the tutorial, select the option to start the assessment.
2. Confirm that the system directs the user to the first assessment question.
3. Verify that the assessment interface is properly displayed and ready for the user to begin answering.

4.1.2 Assessment Interface

- FR-ST-AI1

Control: Automatic

Initial State: User has started the assessment and is ready to begin answering questions.

Input: User initiates the assessment.

Output: System begins recording both audio and video, with an indicator showing recording is active.

Test Case Derivation: Audio and video recording are essential for capturing user responses accurately, and an indicator reassures users that recording is in progress.

How the test will be performed:

1. Start the assessment as a user.

2. Confirm that the system begins recording audio and video.
3. Verify that a visible indicator shows that recording is ongoing.

- FR-ST-AI2

Control: Automatic

Initial State: User is on a new question within the assessment.

Input: System progresses to a new question in the assessment.

Output: The system plays the corresponding audio prompt for the new question.

Test Case Derivation: The audio prompt ensures users understand each question and assessment is running as intended.

How the test will be performed:

1. Progress to a new question in the assessment.
2. Verify that the system automatically plays the corresponding audio prompt.

- FR-ST-AI3

Control: Automatic

Initial State: User has progressed to a new question in the assessment.

Input: System loads a new question.

Output: System displays all possible answer options for the user to select from.

Test Case Derivation: Presenting options ensures the user can make a response selection based on the question, system must mark answer select as right or wrong

How the test will be performed:

1. Go to a new question in the assessment.
2. Confirm that all answer options associated with the question are displayed.

- FR-ST-AI4

Control: Manual

Initial State: User is viewing the answer options for a question in the assessment.

Input: User selects one of the displayed answer options.

Output: System highlights or otherwise indicates the user's selected option.

Test Case Derivation: Visual confirmation of selection minimizes user error and allows the user to review their choice before confirmation.

How the test will be performed:

1. Select an answer option for a question.
2. Verify that the system indicates the selected option visually (e.g., highlighting).

- FR-ST-AI5

Control: Manual

Initial State: User has selected an answer option and is ready to proceed.

Input: User confirms their selection.

Output: System moves the user to the next question or stage.

Test Case Derivation: Confirmation helps prevent unintended answers, ensuring accuracy in user responses.

How the test will be performed:

1. Select and confirm an answer for a question.
2. Verify that the system progresses to the next question or stage upon confirmation.

- FR-ST-AI6

Control: Automatic

Initial State: User is in the process of answering questions within the assessment.

Input: User enters and exits each question.

Output: System records timestamps for entry and exit for each question.

Test Case Derivation: Timestamping provides valuable tracking information for synchronizing recordings and analyzing response timing.

How the test will be performed:

1. Begin the assessment and progress through several questions.
2. Verify that the system logs entry and exit timestamps for each question.

- FR-ST-AI7

Control: Automatic

Initial State: User has reached the final question in the assessment.

Input: User completes the final question and confirms the selection.

Output: System displays a message informing the user that the assessment is complete.

Test Case Derivation: Notifying the user of completion provides a clear end to the assessment and allows users to exit confidently.

How the test will be performed:

1. Complete the final question in the assessment and confirm the selection.
2. Verify that the system displays a completion message.

4.2 Tests for Nonfunctional Requirements

4.2.1 Security

The test cases below ensures that the system meets essential security requirements including authentication of users and encryption of confidential data in the system

- SR-ST-AC1

  Type: Dynamic

  Initial State: System has multiple user roles: Admin, Parent, and Clinician.

  Input/Conditiont: User with Admin role attempts to create and assign accounts to clinicians

  Output/Results: Only Admin users can access and execute functions related to clinician account creation

  How the test will be performed:

  1. Log in as an Admin and attempt to create and view clinician accounts. Verify that the action succeeds
  2. Log in as a non-Admin (e.g., Parent or Clinician) and try to access the same function. Confirm that access is denied with an appropriate error message.

- SR-ST-AC2

  Type: Dynamic

  Initial State: Parent role created and available for testing.

  Input/Condition: User with Parent role logs in and attempts to complete assessments.

  Output/Results: Parent users can create their account, complete assessments, and log out successfully.

  How the test will be performed:

  1. Log in as a Parent user and attempt to start and complete an assessment. Verify successful completion and logout.

2. Attempt to access administrative functions (e.g., creating clinician accounts). Verify that access is denied with an appropriate message.

- SR-ST-AC3

Type: Dynamic

Initial State: Clinician role with restricted access is created and available for testing.

Input/Condition: User with Clinician role logs in and attempts to view assessment results.

Output/Results: Clinician users can view assessment results but cannot start or complete assessments as a Parent user would.

How the test will be performed:

1. Log in as a Clinician and attempt to view completed assessment results. Confirm access is granted.
2. Attempt to start or complete an assessment and confirm access is denied with an error message indicating unauthorized action.

SR-ST-AC4

- **Type:** Dynamic

Initial State: System with user login functionality.

Input/Condition: Users attempt to log in with correct and incorrect credentials.

Output/Results: Users can only log in with correct credentials; unauthorized access attempts are denied.

How the test will be performed:

1. Attempt to log in with valid credentials for multiple roles (Admin, Parent, Clinician). Confirm successful login.
2. Attempt to log in with incorrect credentials (e.g., incorrect username or password). Confirm that login is denied and an error message is shown.

- SR-ST-P1

  Type: Static

  Initial State: System ready for pre-release review.

  Input/Condition: Review documentation to ensure adherence to data protection and privacy laws in the region.

  Output/Results: Confirm all applicable data protection requirements are met.

  How the test will be performed:

  1. Conduct a documentation review with legal and compliance teams, focusing on privacy policies, data handling, and retention practices.
  2. Verify that all data collection, storage, and usage adhere to regional privacy laws (e.g., GDPR if applicable).

- SR-ST-P2

  Type: Static, Automated

  Initial State: System with sensitive data handling enabled.

  Input/Condition: Examine data in transit and at rest.

  Output/Results: Data remains encrypted according to standard encryption protocols during transit and at rest.

  How the test will be performed:

  1. Use network monitoring tools to capture data packets during transmission to ensure data is encrypted.
  2. Review database configuration to verify that data at rest is encrypted. Decrypting should only be possible by authorized clinician accounts.

- SR-ST-P3

  Type: Static

  Initial State: System configured for data collection.

  Input/Condition: Examine data storage for PII.

Output/Results: System does not store any personal identifiable information beyond username and assessment recordings.

How the test will be performed:

1. Conduct a database inspection and audit to ensure no PII (e.g., address, date of birth, names) is stored.
2. Verify data schema to confirm that only usernames and recordings are stored.

- SR-ST-IM1

Type: Dynamic

Initial State: System is in the account creation phase, requiring users to set up passwords.

Input/Condition: User attempts to create an account with both weak and strong passwords.

Output/Results: Account creation is completed only when a strong password, meeting specified security criteria, is entered. Weak passwords are rejected with an error message detailing the password requirements.

How the test will be performed:

1. Attempt to create an account with a weak password (e.g., fewer than 8 characters, no special characters or numbers). Verify that the system rejects the password and displays an error message with the password requirements.
2. Attempt to create an account with a password that meets all the specified criteria (e.g., at least 8 characters, containing upper and lowercase letters, numbers, and special characters). Verify that account creation is successful.
3. Repeat with different combinations of weak and strong passwords to confirm consistent enforcement of the password policy.

4.2.2 Compliance

The test cases below ensures that the system meets essential compliance requirements including security of the system, following the rules of the clinician and law.

- SR-ST-STD1

  Type: Static, Automated

  Initial State: User assessment data, including video and audio recordings, is stored in the system.

  Input/Condition: Examine the storage configuration and security measures applied to user assessment data.

  Output/Results: User assessment data is securely stored and associated only with usernames, ensuring no additional personally identifiable information is included. Data at rest is protected by strong encryption.

  How the test will be performed:

  1. Review database schema to confirm that assessment data is linked only to usernames and does not include any additional identifiable information.
  2. Conduct an inspection of the encryption protocols used for stored data to verify compliance with security standards.
  3. Attempt unauthorized access to stored assessment data to ensure encryption and security measures prevent access by unauthorized users.

4.3 Traceability Between Test Cases and Requirements

[Provide a table that shows which test cases are supporting which requirements. —SS]

5 Unit Test Description

[This section should not be filled in until after the MIS (detailed design document) has been completed. —SS]

[Reference your MIS (detailed design document) and explain your overall philosophy for test case selection. —SS]

[To save space and time, it may be an option to provide less detail in this section. For the unit tests you can potentially layout your testing strategy here. That is, you can explain how tests will be selected for each module. For instance, your test building approach could be test cases for each access program, including one test for normal behaviour and as many tests as needed for edge cases. Rather than create the details of the input and output here, you could point to the unit testing code. For this to work, you code needs to be well-documented, with meaningful names for all of the tests. —SS]

5.1 Unit Testing Scope

[What modules are outside of the scope. If there are modules that are developed by someone else, then you would say here if you aren't planning on verifying them. There may also be modules that are part of your software, but have a lower priority for verification than others. If this is the case, explain your rationale for the ranking of module importance. —SS]

5.2 Tests for Functional Requirements

[Most of the verification will be through automated unit testing. If appropriate specific modules can be verified by a non-testing based technique. That can also be documented in this section. —SS]

5.2.1 Module 1

[Include a blurb here to explain why the subsections below cover the module. References to the MIS would be good. You will want tests from a black box perspective and from a white box perspective. Explain to the reader how the tests were selected. —SS]

1. test-id1

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

Input:

Output: [The expected result for the given inputs —SS]

Test Case Derivation: [Justify the expected value given in the Output field —SS]

How test will be performed:

2. test-id2

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input:

   Output: [The expected result for the given inputs —SS]

   Test Case Derivation: [Justify the expected value given in the Output field —SS]

   How test will be performed:

3. ...

5.2.2 Module 2

...

5.3 Tests for Nonfunctional Requirements

[If there is a module that needs to be independently assessed for performance, those test cases can go here. In some projects, planning for nonfunctional tests of units will not be that relevant. —SS]

[These tests may involve collecting performance data from previously mentioned functional tests. —SS]

5.3.1 Module ?

1. test-id1

   Type: [Functional, Dynamic, Manual, Automatic, Static etc. Most will be automatic —SS]

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

5.3.2 Module ?

...

5.4 Traceability Between Test Cases and Modules

[Provide evidence that all of the modules have been considered. —SS]

References

Author Author. System requirements specification. https://github.com/..., 2019.

6 Appendix

This is where you can place additional information.

6.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

6.2 Usability Survey Questions?

[This is a section that would be appropriate for some projects. —SS]

Appendix — Reflection

[This section is not required for CAS 741 —SS]

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

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. What knowledge and skills will the team collectively need to acquire to successfully complete the verification and validation of your project? Examples of possible knowledge and skills include dynamic testing knowledge, static testing knowledge, specific tool usage, Valgrind etc. You should look to identify at least one item for each team member.

4. 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?