

Verification and Validation Report: Software Engineering

Team 21, Alkalytics
Sumanya Gulati
Kate Min
Jennifer Ye
Jason Tran

March 10, 2025

1 Revision History

Date	Version	Notes
4 March 2025	1.0	Initial Draft
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

symbol	description
CSV	Comma-Separated Values
FR	Functional Requirement
LFR	Look and Feel Requirement
MG	Module Guide
MSR	Maintainability and Support Requirement
NFR	Nonfunctional Requirement
OER	Operational and Environmental Requirement
PR	Performance Requirement
SR	Security Requirement
SRS	Software Requirements Specification
ST	System Test
UHR	Usability and Humanity Requirement
VnV	Verification and Validation
WCAG	Web Content Accessibility Guidelines

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Overview	1
4	Functional Requirements Evaluation	1
4.1	Data Input and Storage	1
4.2	Data Querying and Results	2
4.3	Data Analysis	2
4.4	Data Hygiene	3
4.5	User Access	3
4.6	Data Export	4
5	Nonfunctional Requirements Evaluation	4
5.1	Look and Feel	4
5.2	Usability	5
5.3	Performance	7
5.4	Operational and Environmental	11
5.5	Maintainability and Support	11
5.6	Security	12
6	Unit Testing	16
6.1	Data Storage Module	16
6.2	Data Retrieval Module	17
6.3	Visualization Module	17
6.4	User Management Module	17
7	Changes Due to Testing	18
7.1	Changes Due to Supervisor Feedback	18
7.2	Changes Due to Failed Tests	19
7.2.1	NFR-S3	19
7.2.2	NFR-UH4	19
8	Automated Testing	19
9	Trace to Requirements	20

10 Trace to Modules	24
11 Code Coverage Metrics	24

List of Tables

1	Unit tests summary for Data Storage Module	16
2	Unit tests summary for Data Retrieval Module	17
3	Unit tests summary for User Management Module	18
4	Traceability Matrix for Test Cases and Functional Requirements	20
5	Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 1)	21
6	Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 2)	22
7	Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 3)	23
8	Traceability Matrix for Test Cases and Modules	24

List of Figures

3 Overview

This document provides a comprehensive summary of the Verification and Validation (VnV) process for the Alkalytics project. It includes reports and analysis of system testing and unit testing results, along with traceability between test cases, the software modules in the [Module Guide](#) (MG), and the requirements outlined in the [Software Requirements Specification](#) (SRS). Additionally, it highlights changes made to the implementation based on testing outcomes and user feedback.

4 Functional Requirements Evaluation

4.1 Data Input and Storage

1. FR-ST1

Control: Manual

Initial State: Database is running and ready to intake data

Input: `SAMPLE_.CSV_FILE`

*Note: Actual experiment data cannot be provided publicly.

Expected Output: The inputted data is stored in the system with proper labels and without any data loss or errors

Actual Output: Data successfully stored in the database with correct key-value pairs; no data loss or errors were detected.

Result: **Pass**

2. FR-ST1.1

Control: Manual

Initial State: Database is running and ready to intake data

Input: `SAMPLE_DATA_POINT: {NEW_EXPERIMENT_ROW}`

*Note: Actual experiment data cannot be provided publicly.

Expected Output: The inputted data is stored in the system with proper labels and without any data loss or errors.

Actual Output: The inputted data is stored in the system under the correct label and without any data loss or errors.

Result: **Pass**

4.2 Data Querying and Results

1. FR-ST2

Control: Manual

Initial State: The system is not running any jobs, and the user interface is cleared.

Input: **SAMPLE_PARAMETERS**: {Graph: Line, Experiment Dates: 2024-11-05, Parameter Type: Data Sheet, X Parameter: C1 Cond, Y Parameter: C1 Temp, X Axis Label: C1 Cond, Y Axis Label: C1 Temp, Graph Title: 2024-11-05 C1 Cond vs C1 Temp }

Expected Output: A human-readable and customizable visualization of correct results corresponding to the selected parameters from the input

Actual Output: Line graph visual displaying the data points from the data sheet with the axis custom labeled and graph title. A line of best fit is also displayed on the graph.

Result: **Pass**

4.3 Data Analysis

1. FR-ST3

Control: Manual

Initial State: The application's cleared user interface which has not yet been used to query data, with no graph showing yet.

Input: **SAMPLE_PARAMETERS**: {Graph: Scatter, Experiment Dates: 2024-09-26, Parameter Type: Data Sheet, X Parameter: C1 pH, Y Parameter: C1 Temp, X Axis Label: C1 pH, Y Axis Label: C1 Temp}

Expected Output: A small written human-readable paragraph explaining the input data.

Actual Output: A linear regression line is plotted with the resulting graph output, along with information about how well the regression line fits with the data.

Result: Neutral. The actual output does not match the original expected output, however, the linear regression line provides a clear trend analysis and has been determined to be more useful than a text-based analysis.

4.4 Data Hygiene

1. **FR-ST4**; Stretch Goal Requirement

Control: Manual

Initial State: The application's cleared user interface which has not yet been used to query data, with no graph showing yet.

Input: `SAMPLE_.CSV_FILE`

Expected Output: A log file documenting errors found in the input data and/or removals of missing data.

Actual Output: Feature not implemented; as this is a stretch goal, the functionality is currently not available.

Result: **Fail**

4.5 User Access

1. **FR-ST5**

Control: Manual

Initial State: User interface shows a login page, with no login credentials currently used

Input: `SAMPLE_VALID_EMAIL`, `SAMPLE_VALID_PASSWORD`

Expected Output: The page redirects to the page designated after login

Actual Output: Once credentials are submitted, the page redirects to the main home page of the application.

Result: **Pass**

4.6 Data Export

1. **FR-ST6**; Stretch Goal Requirement

Control: Manual

Initial State: User interface after multiple usages of data queries

Input: `USER ACTION => BUTTON CLICK`

Expected Output: Query report will be downloaded to the user's device

Actual Output: Feature not implemented; as this is a stretch goal, the functionality is currently not available.

Result: **Fail**

5 Nonfunctional Requirements Evaluation

5.1 Look and Feel

1. **NFR-LF1**

Type: User Demo, Manual

Initial State: Fully functional application ready for user interaction, starting at the login page

Input/Condition: User engagement with application

Expected Output: 90% of responses in the Navigation and Ease of Use section of the usability survey are at least 'Neither easy nor difficult'; 85% of users navigate the entire application within 10 minutes.

Actual Output: `User's answers to usability survey, time taken to complete tasks or navigate site`

Result: **Pass** OR **Fail**

2. NFR-LF2

Type: Manual

Initial State: Application running on a 15" laptop, 24" monitor, standard smart phone and/or tablet

Input/Condition: Manual tester's engagement with application

Expected Output: Number of inconsistencies found across all devices ≤ 10 .

Actual Output:

Dashbaord: 2 inconsistencies on mobile, 1 inconsistency on tablet. Not full screen cuts off half way on both tablet and mobile. Table quick view is not mobile compatible as the table is squished and not readable.

Query Page: 4 inconsistencies on mobile, 1 inconsistency on tablet. The table is not horizontally scrollable on mobile view as such cannot see the search bar and column drop down. Also cannot type in the equations field to apply them to the table. On tablet view there is abit of the table that is cut off on the right.

Upload Page: No inconsistencies.

Graphs Page: 1 inconsistency on mobile. The modal is cut off on the right. Although it does not affect the functionality as all the buttons are still reachable, it is not visually apealing.

Login Page: 0 inconsistencies.

Result: **Pass**

Although there are still inconsistency, they are mainly due to the fact that mobile compatibility is a nice to have. It is expected that almost always users will be using the app on a standard sized laptop or monitor. Hence, why there are only inconsistencies present in tablet and mobile views and not the laptop or monitor views.

5.2 Usability

1. NFR-UH1

Type: User Demo, Manual

Initial State: Fully functional application ready for user interaction, starting at the login page

Input/Condition: User engagement with application, set of tasks to complete

Expected Output: 85% of users ask for help no more than 3 times; 90% of responses to the Learning section of the usability survey are at least 'Neither easy nor difficult'.

Actual Output: User's answers to usability survey, # of times user asked usability test conductor for help

Result: Pass OR Fail

2. NFR-UH2

Type: Manual

Initial State: Fully developed application, starting at the login page

Input/Condition: Manual tester using application

Expected Output: 0 language discrepancies found in the application.

Actual Output: 0 language discrepancies found in the application.

Result: Pass

3. NFR-UH3

Type: Manual

Initial State: Application ready to take in data

Input/Condition: SAMPLE_.CSV_FILE

Expected Output: Error logs from unrecognized characters, or successful upload

Actual Output: The upload csv will still upload the unrecognized symbols without any issues. The type will just be a string. As a result, there is nothing to log.

Result: Pass

4. NFR-UH4

Type: Manual

Initial State: Application ready for use

Input/Condition: Tester engagement using third party tools

Expected Output: List of accessibility issues in accordance to Web Content Accessibility Guidelines (WCAG), and checklists for if page is screen-readable

Actual Output:

Dashbaord: All buttons and form fields are do not have alt texts or context.

Query Page: All buttons and form fields are do not have alt texts or context. Scrolling is also not possible from keyboard shortcuts.

Upload Page: Upload button does not have alt text.

Graphs Page: All buttons and form fields are do not have alt texts or context.

Login Page: Is accessible

Result: **Fail**

Most pages do not comply with the WCAG guidelines. Many of the components and buttons are missing alt texts which makes the page screen readable.

5.3 Performance

1. NFR-P1

Type: Manual

Initial State: Application navigated to upload page, ready to upload file

Input/Condition: `SAMPLE_.CSV_FILE`

Expected Output: Average upload duration \leq `UPLOAD_TIME`

`UPLOAD_TIME` = 60 seconds

Actual Output: File uploaded in 48 seconds on average, satisfying the performance requirements.

Result: **Pass**

2. NFR-P2

Type: Manual

Initial State: Application navigated to querying page, ready to make query

Input/Condition: `SAMPLE_QUERY`: {}

Expected Output: Average query response duration \leq `QUERY_TIME`
where `QUERY_TIME` = 3 seconds

Actual Output: Query response time averaged less than 0.1, satisfying the performance requirement.

Result: **Pass**

3. NFR-P3

Type: Manual

Initial State: Application ready to use

Input/Condition: Tester engagement

Expected Output: Average response time of buttons on website \leq `RESPONSE_TIME`

`RESPONSE_TIME` = 2 seconds

Actual Output:

Dashbaord: Dropdowns have a response time of 0.1 seconds

Query Page: All buttons and form fields do not have alt texts or context. Scrolling is also not possible from keyboard shortcuts.

Upload Page: Upload button takes about 2 seconds if there is an error and on average 1.75 seconds for a successful upload. It also takes 0.1 seconds to begin uploading.

Graphs Page: Graph, parameter type drop downs have a response time of 0.1. Date, x and y parameter drop downs have an average response time of 1.25 seconds

Login Page: Login and register button have a response time of 0.1 seconds

Result: **Pass**

4. NFR-P4

Type: Manual

Initial State: Application navigated to graphs page, ready to make query to generate graphs

Input/Condition: `SAMPLE_QUERY: {}`

Expected Output: Average time taken to generate graphs/visualizations \leq `GRAPH_TIME`

`GRAPH_TIME`: 10 seconds

Actual Output:

Result: **Pass** OR **Fail**

5. NFR-P5

Type: Manual

Initial State: Application navigated to query page, ready to make query

Input/Condition: `SAMPLE_QUERY : {NaCl \geq 1.0}`

Expected Output: A check for precision of numbers in different components of system. Values should be accurate to 4 decimal places.

Actual Output: Parameter values displayed with an accuracy matching that of the original spreadsheet data. All parameters exceeded the accuracy of 4 decimal places, satisfying the requirement.

Result: **Pass**

6. NFR-P6

Type: Manual

Initial State: Application running as normal

Input/Condition: Tester temporarily taking down back end

Expected Output: Error messages properly displayed

Actual Output: Displayed appropriate error messages for all components indicating that data could not be retrieved from backend services.

Result: **Pass**

7. **NFR-P7**

Type: Manual

Initial State: Application running as normal

Input/Condition: Tester temporarily disconnects internet connection

Expected Output: Previously generated plots and previous queries still working

Actual Output:

Result: **Pass** OR **Fail**

8. **NFR-P8**

Type: Manual

Initial State: Three devices ready to run application

Input/Condition: **SAMPLE_QUERY**: {}

Expected Output: System response time while under load \leq **RESPONSE_TIME_THRESHOLD**

Actual Output:

Result: **Pass** OR **Fail**

9. **NFR-P9**

Control: Data Generation, Automated

Initial State: A database with a known amount of experiment data

Input/Condition: **DUMMY_DATA**

Expected Output: Observations on system health after large payload

Actual Output: Large payload are processed with slight performance degradation, but still satisfying performance requirements.

Result: **Pass**

5.4 Operational and Environmental

1. NFR-OE1

Type: Manual

Initial State: Application running on a windows device as a web application on a Chromium based browser.

Input/Condition: Tester engagement

Expected Output: A list of all discovered issues with the application that arise due to environment compatibility

Actual Output: No issues discovered when using Chromium based browser. The experience is same as the expected.

Result: Pass

2. NFR-OE2

Type: Manual, User Demo

Initial State: Application running and ready for use on home screen

Input/Condition: User engagement

Expected Output: Survey results depicting subjective complexity of onboarding process

Actual Output:

Result: Pass OR Fail

5.5 Maintainability and Support

1. NFR-MS1

Type: Manual, User Demo

Initial State: Application running, ready for use

Input/Condition: User engagement

Expected Output: Observations on user's ability to complete tasks without support

Actual Output:

Result: **Pass** OR **Fail**

2. **NFR-MS2**

Type: Manual

Initial State: Multiple Chromium-based web browsers open

Input/Condition: Tester Engagement

Expected Output: All abnormal behaviour of web pages observed on each different web browser

Actual Output: The application ran on Firefox, Chrome and Microsoft Edge. There was no abnormal behaviour across the browsers running the app.

Result: **Pass**

5.6 **Security**

1. **NFR-SR1**

Type: Manual

Initial State: Application login page is displayed.

Input/Condition: **SAMPLE_VALID_CREDENTIALS**, **SAMPLE_INVALID_CREDENTIALS**

Expected Output: Access is granted or denied based on the validity of the credentials provided.

Actual Output: Access is granted for the **SAMPLE_VALID_CREDENTIALS**, which consisted of a registered email and correct password, and consistently denied for the **SAMPLE_INVALID_CREDENTIALS**, which consisted of the following test inputs:

- Registered email, incorrect password
- Unregistered email, correct password
- Unregistered email, incorrect password

Result: **Pass**

2. NFR-SR2

Type: Manual

Initial State: Application logged in with multiple user roles (e.g., admin, researcher, research assistant).

Input/Condition: Tester interacts with the application using on different user roles.

Expected Output: Access to query or modify data and perform sensitive operations is restricted according to user roles.

Actual Output: Only admin and researchers were able to modify data and these operations were not available to users with the role of research assistant.

Result: **Pass**

3. NFR-SR3

Type: Manual

Initial State: User is logged into the application.

Input/Condition: User remains inactive for a specified period.

Expected Output: User is automatically logged out after a predefined period of inactivity.

Actual Output: The user was not logged out after 15 minutes of inactivity.

Result: **Fail**

4. NFR-SR4

Type: Manual

Initial State: Application is open, ready for data entry and CSV upload.

Input/Condition: `VALID_DATA`, `INVALID_DATA`, `CORRUPTED_DATA`

Expected Output: All invalid inputs and CSV uploads are rejected, and only valid data entries are processed.

Actual Output: Valid data entries were processed correctly. All invalid inputs were rejected with a visual cue, and/or error message.

Result: **Pass**

5. **NFR-SR5**

Type: Manual

Initial State: Application database contains a set of unique, validated records. Application interface is open for data entry, processing, and transfer actions.

Input/Condition: **SAMPLE_DUPLICATE_RECORD**

Expected Output: Duplicate records are detected and prevented, and data accuracy is maintained during all transfer operations.

Actual Output: Duplicate records are detected and filtered out through indexing, thus data accuracy maintained throughout the transfer process.

Result: **Pass**

6. **NFR-SR6**; Out of Scope

Type: Manual/Automated

Initial State: Database is operational, with storage capacity at or below normal usage. Alert system is configured, and the administrator contact information is set up to receive notifications.

Input/Condition: Tester simulates increasing database storage usage to exceed the **STORAGE_THRESHOLD**.

Expected Output: System successfully detects when storage usage exceeds **STORAGE_THRESHOLD** and sends a timely alert to administrators.

Actual Output: Feature not implemented; notification system is considered a stretch goal.

Result: **Fail**

7. **NFR-SR7**; Out of Scope

Type: Manual/Automated

Initial State: The application is operational, with logging features configured and permissions for accessing logs assigned to administrators only.

Input/Condition: Tester performs various access and modification actions within the application, then attempts to access the audit logs with both authorized and unauthorized user accounts.

Expected Output: All actions are logged with timestamps and user identities, and audit logs are accessible only to authorized users, with proper encryption.

Actual Output: Feature not implemented; audit system is considered a stretch goal.

Result: **Fail**

8. **NFR-SR8**; Out of Scope

Type: Manual

Initial State: Application login screen is open. Administrator contact information is configured to receive security alerts.

Input/Condition: Tester attempts multiple failed logins to trigger the suspicious activity detection mechanism.

Expected Output: Application detects and blocks access after three failed attempts, sends an alert to administrators, and locks out the user temporarily.

Actual Output: Application detects failed login attempts but does not send alerts. Notification system is considered a stretch goal.

Result: **Fail**

9. **NFR-SR9**; Out of Scope

Type: Automated/Manual

Initial State: Application is running under typical workload conditions. Monitoring tools and alerts for CPU and memory usage are enabled.

Input/Condition: Tester simulates a high workload to approach system resource limits, observing the system's monitoring and optimization

response.

Expected Output: The system actively manages CPU and memory usage, preventing overload and maintaining performance stability.

Actual Output: Not conducted due to resource constraints and complexity in simulating high workload conditions in the current testing environment.

Result: **Fail**

6 Unit Testing

6.1 Data Storage Module

Test ID	Test Name	Result
6.3-UT1	Uploads experiments and data	Pass
6.3-UT2	Links data to experiments	Pass
6.3-UT3	Uploads data when experiment exists	Pass
6.3-UT4	Skips data upload if no experiment exists	Pass
6.1-UT5	Cleans data by removing empty values	Pass
6.1-UT6	Checks for duplicate experiments	Pass
6.1-UT7	Handles multiple matching experiments	Pass
6.1-UT8	Handles no matching experiments	Pass
6.3-UT9	Runs full migration process	Pass
6.3-UT10	Handles experiment upload errors	Pass
6.3-UT11	Handles data upload errors	Pass
6.3-UT12	Shows upload status indicators	Pass
6.3-UT13	Handles drag over and drop event with valid file	Pass
6.3-UT14	Handles file input change event with valid file	Pass
6.2-UT15	Handles cell editing	Pass

Table 1: Unit tests summary for Data Storage Module

6.2 Data Retrieval Module

Test ID	Test Name	Result
6.2.1-UT1	Initialize with default values	Pass
6.2.1-UT2	Fetch experiment IDs	Pass
6.2.1-UT3	Fetch experiments	Pass
6.2.1-UT4	Fetch data	Pass
6.2.1-UT5	Handle Apollo error states	Pass
6.2.1-UT6	Refetch experiments and data	Pass
6.2.1-UT7	Call RefetchExperiments function	Pass
6.2.1-UT8	Call RefetchData function	Pass
6.2.1-UT9	Filter data by search keyword	Pass

Table 2: Unit tests summary for Data Retrieval Module

6.3 Visualization Module

6.4 User Management Module

Table 3 provides a summary of the results for each test, where each test includes a link to its corresponding test file in the project repository.

Test ID	Test Name	Result
6.4-UT1		
6.4-UT2		
6.4-UT3		
6.4-UT4		
6.4-UT5		
6.4-UT6		
6.4-UT7		
6.4-UT8		
6.4-UT9		
6.4-UT10		

Table 3: Unit tests summary for User Management Module

7 Changes Due to Testing

7.1 Changes Due to Supervisor Feedback

Our supervisor suggested the current data visualization form intake could be improved by allowing the user to filter the experiment data by additional parameters, rather than only by the date it was performed. This feature will be added to the final product.

They have also requested to add a new feature that was not originally part of the requirements. This feature allows the user to compute a variety of efficiency factors related to the data, measuring the efficiencies of the membrane used in the experiment. This feature will be added to the final product, but thorough validation testing will be limited due to time constraints.

[This section should highlight how feedback from the users and from the supervisor (when one exists) shaped the final product. In particular the feedback from the Rev 0 demo to the supervisor (or to potential users) should be highlighted. —SS]

7.2 Changes Due to Failed Tests

The following sections outline changes that will be made to the implementation following a failed test or tests. This does not include tests for stretch goal requirements that failed.

7.2.1 NFR-S3

The system did not automatically log out the user after a period of inactivity. This was due to the lack of a session timeout manager, which will be implemented in the final product.

7.2.2 NFR-UH4

The system was not compliant to with the WCAG standards. This is due to the fact that the current research team using the project does not need any accessibility considerations when using the app. However this may not be true in the future, as such, updates to the current pages will be implemented using the feedback from SiteImprove in the final product.

8 Automated Testing

Pytest was used as the automated testing framework for Python. Unit test suites were written and executed using Pytest, which supports fixtures, parameterization, and code coverage metrics, making automated testing more efficient and flexible.

9 Trace to Requirements

Test ID	FR-ST1	FR-ST1.1	FR-ST2	FR-ST3	FR-ST4	FR-ST5	FR-ST6
FR-1	X	X					
FR-2	X	X					
FR-3	X	X					
FR-4	X	X					
FR-5			X				
FR-6			X				
FR-7			X				
FR-8			X				
FR-9			X				
FR-10				X			
FR-11				X			
FR-12					X		
FR-13					X		
FR-14						X	
FR-15							X

Table 4: Traceability Matrix for Test Cases and Functional Requirements

Test ID {NFR-}	LF1	LF2	UH1	UH2	UH3	UH4	OE1	OE2	MSR1	MSR2
LFR-1	X									
LFR-2		X								
LFR-3	X									
LFR-4	X									
LFR-5	X									
LFR-6	X									
LFR-7	X									
UHR-1			X							
UHR-2				X						
UHR-3					X					
UHR-4			X							
UHR-5			X							
UHR-6						X				
OER-2							X			
OER-3							X			
OER-4							X			
OER-5								X		
MSR-4									X	
MSR-6										X

Table 5: Traceability Matrix for Test Cases and Nonfunctional Requirements
(Part 1)

Test ID {NFR-}	P1	P2	P3	P4	P5	P6	P7	P8	P9
PR-1	X								
PR-2		X							
PR-3			X						
PR-4		X							
PR-5				X					
PR-6					X				
PR-7					X				
PR-8					X				
PR-9						X			
PR-10							X		
PR-11								X	
PR-12									X

Table 6: Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 2)

Test ID {NFR-}	SR1	SR2	SR3	SR4	SR5	SR6	SR7	SR8	SR9
SR-1	X								
SR-2		X							
SR-3		X							
SR-4			X						
SR-5				X					
SR-6					X				
SR-7					X				
SR-8					X				
SR-9				X					
SR-12						X			
SR-13							X		
SR-14							X		
SR-15							X		
SR-16								X	
SR-17									X

Table 7: Traceability Matrix for Test Cases and Nonfunctional Requirements (Part 3)

10 Trace to Modules

Test ID	FR-ST1	FR-ST1.1	FR-ST2	FR-ST3	FR-ST4	FR-ST5	FR-ST6
M1							
M2							
M3							
M4							
M5							
M6							
M7							
M8							
M9							
M10							
M11							
M12							
M13							

Table 8: Traceability Matrix for Test Cases and Modules

11 Code Coverage Metrics

Appendix — Reflection

1. **What went well while writing this deliverable?**

Writing the general structure of the report was straightforward as we could refer back to the tests we had written in the VnV Plan (cont. later)

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

Constructing and executing all of our tests was the biggest challenge. Due to time constraints, (cont. later)

3. **Which parts of this document stemmed from speaking to your client(s) or a proxy (e.g. your peers)? Which ones were not, and why?**

4. **In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)**