

PROJECT QUALITY PLAN



Uqla Phase 4

#	Student Name	Responsibilities
1	Norah Almubarak	Introduction to the project, purpose of quality plan for the project, general constraints, quality goals, functional requirements, system and integration requirements, requirements metrics.
2	Lujain Albattah	Project scope, planned verification tests, functional requirements system, and integration requirements, metrics introduction.
3	Aljawharah Alotaibi	Planned reviews, functional requirements, system and integration requirements, design metrics.
4	Yara Aljasir	Planned acceptance tests, functional requirements, system and integration requirements, implementation metrics.
5	Raghad Alotibi	Planned configuration management, functional requirements, system and integration requirements, testing metrics.
6	Sarah Alshali	Planned validation tests functional requirements, system and integration requirements, maintenance metrics.

Instructor Name:
L.Monira Aldolaimi

1. Introduction to the project

Nowadays, with health and fitness taking center stage in our lives, the demand for personalized and innovative solutions is more pronounced than ever. Uqla seamlessly connects trainees with certified coaches, all while utilizing AI analysis for detailed health insights to provide a holistic fitness experience. What sets Uqla apart is its integration of video recognition technology, a feature that actively monitors trainees' exercise sessions in real-time.

The quality plan will provide a clear vision for Uqla's quality activities and goals, as well as the standards and practices aimed at guaranteeing the quality of our work.

This document serves as the quality plan for Uqla, it will detail our approach to ensure that we meet user's expectations. We went through the purpose of quality plan, and the scope of the project. Then we identified the general constraints, functional requirements, system integration requirements, and quality goals that must be met. Then we listed all the activities of planned reviews, planned verification tests, planned validation tests, and planned acceptance tests. Finally, we planned the configuration management, and wrote the project team responsibilities.

2. Purpose of Quality Plan for project

The purpose of the Quality Plan for Uqla project is to ensure that the app meets predefined quality standards. It outlines quality objectives, criteria, and assurance activities such as testing and compliance checks. The plan also helps mitigate risks, ensures compliance with regulations, and facilitates continuous improvement throughout the project lifecycle. Ultimately, it aims to deliver a high-quality app that meets user expectations and enhances user satisfaction.

3. Project Scope

Uqla is an iOS application developed with aims to connect individuals with qualified coaches, allowing them to receive customized workout plans, real-time feedback on their exercise form, and ongoing support for achieving their fitness goals. And enhancing the coaching process by providing coaches with valuable insights into the trainees' overall health and habits, enabling continuous modification and monitoring of their fitness plans.

Our target users are individuals who are seeking to forge a healthier lifestyle, and coaches who are looking for a place to offer their expertise and services.

We will achieve our goal by fulfilling the requirements, while following the quality plan during each phase to guarantee thorough documentation and quality for every project deliverable.

4. General Constraints

1. The system shall integrate with Health app and MyFitnessPal.
 2. The system shall support iOS platform.
 3. The system shall enable online transaction by integrating with Apple pay.
 4. All scripts shall be written in Swift language.
 5. The data type that should transfer between the system components should be in JSON format.
-

5. Functional Requirements

5.1. Trainee Requirements

- 5.1.1. The trainee shall be able to search for coaches based on their specialty, name, and availability.
 - 5.1.2. The trainee shall be able to view the coach's profile and credentials.
 - 5.1.3. The trainee shall be able to send a coaching request to their chosen coach.
 - 5.1.4. The trainee shall be able to pay the subscription fees.
 - 5.1.5. The trainee shall be able to view his personalized workout schedule.
 - 5.1.6. The trainee shall be able to view detailed instructions for each exercise in their schedule.
 - 5.1.7. The trainee shall be able to view his live sessions schedule.
 - 5.1.8. The trainee shall be able to view video demonstrations for each exercise in their schedule.
 - 5.1.9. The trainee shall be able to exercise using the video recognition for each exercise in their schedule.
 - 5.1.10. The trainee shall be able to record their exercise.
 - 5.1.11. The trainee shall be able to request a live session with the coach.
 - 5.1.12. The trainee shall be able to track his progress towards fitness goals.
 - 5.1.13. The trainee shall be able to track his health information.
 - 5.1.13.1. The trainee shall be able to track his nutrition.
 - 5.1.13.2. The trainee shall be able to track his water intake.
 - 5.1.14. The trainee shall receive notifications.
 - 5.1.15. The trainee shall be able to add comments on each workout exercise.
-

5.2. Coach Requirements

- 5.2.1. The coach shall be able to view trainee requests.
 - 5.2.1.1. The coach shall be able to accept/reject trainee requests.
- 5.2.2. The coach shall be able to view trainee profile information.
- 5.2.3. The coach shall be able to create workout plans for each trainee.
- 5.2.4. The coach shall be able to update the trainee workout plans.
- 5.4.5. The coach shall be able to access his trainee's workout schedules.
- 5.4.6. The coach shall be able to respond to trainee comments within 24 hours of receiving them.
- 5.4.7. The coach shall be able to conduct follow-up live sessions.
- 5.4.8. The coach shall be able to track the trainee's progress.

5.3. User Requirements

- 5.3.1. The user shall be able to create an account.
- 5.3.2. The user shall be able to log into the system.
- 5.3.3. The user shall be able to log out of their account.
- 5.3.4. The user shall be able to edit their account.
- 5.3.5. The user shall be able to review the app.
- 5.3.6. The user shall be able to communicate through messaging.
- 5.3.7. The user shall be able to communicate through video calls.
- 5.3.8. The user shall be able to reset their password.

6. System and Integration Requirements

- 6.1. The system shall securely send a randomly generated password to a new user's phone number.
- 6.2. The system shall access the device's camera to provide video recognition capabilities.
- 6.3. The system shall notify trainee in case of wrong exercise form.
- 6.4. The system shall securely create reports for trainees showcasing their progress.

7. Quality goals

Requirements	McCall's Quality Factors
7.1. The system shall be available 99.0% of the time.	Reliability
7.2. The user shall learn how to use the system's core features in less than 10 minutes.	Usability
7.3. The system shall interoperate with Health app to retrieve health data. 7.4. The system shall interoperate with MyFitnessPal app to retrieve diet data.	Interoperability
7.5. The system shall be able to pass the testing in less than 6 weeks.	Testability
7.6. The system shall display red lines indicating incorrect form.	Correctness
7.7. The system shall send an OTP (one time password) for user password reset.	Integrity
7.8. The software documentation shall adhere to the IEEE documentation guidelines.[1]	Maintainability
7.9. The Uqla application shall not use more than 10% of battery power in two hours' time.	Efficiency

8. Planned Reviews

Project Reviews

A Project Review is a process of examining and evaluating activities, documents, diagrams, and implementations at the end of every software development life cycle phase, to evaluate the progress and success of the project at any stage of the development process.

1. Requirements Gathering and Analysis

- Software Requirements Specification (SRS) review.
- Non-functional requirements review.
- List of all ministry of sport recommendations.

2. Software Design:

- Use case diagram.
- Deployment diagram.
- Class diagram.

3. Implementation:

- Software source code.
- MyFitnessPal and Health interface implementation.
- Database implementation.

4. Testing:

- Unit testing for the software source code
- Integration testing for MyFitnessPal and health
- Interactive Application Security Testing (IAST)

5. Maintenance:

- Adding new functionalities to the system
- Data Backup and Recovery
- Configuration and Infrastructure Review

Walkthroughs

These are group work sessions in which the walkthrough team validates the deliverable using previously defined scripts, presentations, question & answer sessions, and brainstorming sessions, if appropriate.

1. Requirements Gathering and Analysis

- **Software Requirements Specification (SRS) review:** Are all the requirements clear and consistent?
- **Non-functional requirements review:** Are the requirements documented, managed, controlled, and measurable?
- **List of all ministries of sport recommendations:** Are all recommendations covered and up to date?

2. Software Design:

- **Use case diagram:** Do all actors exist?
- **Deployment diagram:** Does the diagram visualize the physical hardware and software of a system?
- **Class diagram:** Are all relationships between classes specified?

3. Implementation:

- **Software source code:** Does the source code implement all required functional requirements? Does the function contain bugs and defects?
- **MyFitnessPal and Health interface implementation:** Does the implementation comply to the design phase specifications?
- **Database implementation:** Do all tables have proper foreign/primary keys?

4. Testing:

- **Unit testing for the software source code:** Do all functions operate with the specified specifications and as expected? Are tests thorough in terms of the functional requirements?
- **Integration testing for MyFitnessPal and health:** Are MyFitnessPal and health integrations tested for iOS compatibility? How compatibility between system and service versions is ensured by unit tests?
- **Interactive Application Security Testing (IAST):** Are security vulnerabilities being identified? Are these vulnerabilities being monitored over all the time?

5. Maintenance:

- **Adding new functionalities to the system:** Have the new features been added without interfering with the functionality that already exists?
- **Data Backup and Recovery:** Are backup procedures routinely examined to guarantee the recoverability and integrity of data? Are backups carried out at the best intervals to reduce data loss and maintain system performance?
- **Configuration and Infrastructure Review:** Does the infrastructure adequately support the additional system requirements? Is performance monitoring implemented to promptly detect and resolve any inefficiencies or bottlenecks inside the system?

Inspection

The team members should inspect all the documents and ensure all the aspects are correct and complete.[5]

9. Planned Verification Tests [4]

1. Test if both trainees and coaches can easily understand and navigate the application within 1 hour.
2. Test if the video recognition technology used for real-time exercise monitoring provides prompt feedback and assessment for trainees, with response times not exceeding 2 milliseconds.
3. Test that trainees' and coaches' data, including exercise sessions, progress, and coaching plans, is securely stored in the cloud storage system.
4. Test if the system response time does not exceed 3 seconds.

10. Planned Validation Tests [4]

1. Test if the application allows user to successfully register by providing necessary information, such as name, password, email, and relevant training details.
2. Test if user can log into the application securely using their registered credentials.
3. Test if the application tracks and records trainees' exercise sessions, capturing relevant data such as duration, intensity, form analysis, and progress.
4. Test if trainees can communicate with their assigned coaches by exchanging feedback, questions, training updates, and personalized coaching plans.
5. Test if coaches have access to trainees' health dashboard and analytics.

11. Planned Acceptance Tests [4]

1. **Design Phase Acceptance Test:** Conduct an acceptance test during the design phase where trainees and coaches can provide feedback on the user interface, overall usability, and coaching features. Incorporate necessary improvements and enhancements based on their feedback.
2. **Implementation Phase Acceptance Test:** Perform an acceptance test after the implementation phase, allowing trainees and coaches to interact with the actual application. Identify and address any detected issues, usability concerns, or coaching functionality gaps before the final release.
3. **Release Phase Acceptance Test:** Conduct an acceptance test after the release of the application, where trainees and coaches can evaluate the product's functionality, user experience, and coaching effectiveness in a real training environment. Promptly address any identified faults or bugs in subsequent updates or patches to ensure an optimal training and coaching experience.

12. Planned configuration management.

Software Storage

The application will use AWS as a storage for storing application data, the data will be stored in Amazon S3, a cloud-based object storage service provided by AWS.

By seamlessly integrating Amazon S3 with the application, users can directly perform read and write operations, facilitating efficient data management while maintaining standardized organization within the storage infrastructure.

Security and Backups

Uqla will implement robust encryption techniques to guarantee user data protection:

- **Advanced Encryption Standard (AES):** Ensure that sensitive data, including user passwords, health information, and other personal information, is consistently encrypted and kept locally on the device.
- **TLS (Transport Layer Security):** Secure the connection between the Uqla (iOS app) and the server backend, applied at the app's initial setup.
- **Open Authorization (OAuth):** Integrate OAuth in Uqla to provide secure authentication protocols to ensure users can safely sign in using their existing login credentials from Facebook, Google, or Apple ID.

To prevent the risk of data loss, automated backups of the user-generated content and application database are performed, we will be using Amazon S3 Continuous backups that allow us to restore to any point in time within the last 35 days, After the first full backup of existing data is complete, changes in S3 bucket data are tracked as they occur.

The tracked changes allow us to use PITR (point-in-time restore) for the retention period of the continuous backup.[6]

Version Control

Uqla's version control will be managed through GitHub, a code hosting platform for version control and collaboration. Each feature will have its own branch, facilitating parallel development. When a feature is completed, it will undergo a review process via GitHub's pull request mechanism. Team members will create pull requests at the end of the day, describing changes before merging them back to the desired branch. This process ensures effective coordination, consistency, and adherence to the project schedule. Upon completion of the development phase, feature branches will be merged back to the development branch, which contains all completed features not yet released. When enough features are ready in the development branch, a release will be created and pushed to the project's master branch, containing production-ready code. This approach ensures smooth collaboration, streamlined development, and reliable version management.

13. Quality Assurance Process Metrics

Software metrics manage the estimation of software processes which can be used to guide and evaluate software development, QA metrics provide measurements that QA engineers, developers, and other stakeholders use to track the progress of testing, optimize quality assurance processes.[13]

Here are the matrices that we will be using for our project Uqla:

1. Requirements Phase:

- Requirement Stability Index (RSI): It gives indication on effectiveness of the Requirements gathering process. It's a comparison of change to requirements (added/ deleted/ modified) and the original requirements, it should be measured at overall project level.[14]

2. Design Phase:

- Development Errors Removal Effectiveness (DERE): It measures how well development teams identify and fix errors during development (design and code), it gives an insight on how well testing strategies were during development.[14]

3. Implementation Phase:

- Code Reuse (CRe): It measures how much of the code is reused across the project, it gives an insight on the project quality as reused software are guaranteed more since they are tested and ready to integrate.[15]

4. Testing Phase:

- Defect Density: It measures the number of defects found within a software's lines of code, helps analyze the performance of the program and emphasize the higher-risk program components.[14]

5. Maintenance Phase:

- Full availability (FA): It measures the percentage of time the software is available, helps in estimating the reliability of the software, ensuring minimal downtime and uninterrupted service for users.

Phase 1: Requirements

- Measuring requirements phase quality [7]

Metric Name	Metric	What does it do?	Data needs to be collected
Requirement Stability Index (RSI)	$RSI = 1 - ((\text{Number of changed requirements} + \text{Number of deleted requirements} + \text{Number of added requirements}) / \text{Total number of initial requirements}) \times 100$	RSI ensures the stability of requirements over the development process; it ranges from 0 to 1. Higher values indicate greater stability and lower volatility in requirements.[8]	<ol style="list-style-type: none"> 1. Number of changed requirements. 2. Number of deleted requirements. 3. Number of added requirements 4. Number of Initial requirements.

Phase 2: Design

- Measuring design phase quality [9]

Metric Name	Metric	What does it do?	Data needs to be collected
Development Errors Removal Effectiveness (DERE)	$DERE = NDE / (NDE + NYF)$	Merge development-stage error records with failure records from the initial year of regular operation.	<ol style="list-style-type: none"> 1. NDE = Total development (design and code) errors detected during development. 2. NYF = Number of software failures detected during one year of maintenance service.

Phase 3: Implementation

-measuring implementation phase quality [10]

Metric Name	Metric	What does it do?	Data needs to be collected
Code Reuse (CRe)	$CRe = ReKLOC / KLOC$	Measure the level of reusability that the code possesses by the number of thousands of reused lines of code with respect to the thousands of lines of code of the system.	<ol style="list-style-type: none"> 1. ReKLOC= Number of thousands of reused lines of code. 2. KLOC=Thousands of lines of code.

Phase 4: Testing

-measuring testing phase quality [11]


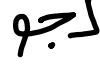



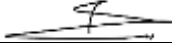
Metric Name	Metric	What does it do?	Data needs to be collected
Detect Density	Defect Density = Total Defect/ Software Size	Measures the number of flaws found in software or other parts over the period of a development cycle and splits it by the software's size. It is used to determine whether the software will be released.	<ol style="list-style-type: none"> 1. The size of software or code (could be measured by FP (function points) or LOC (Line of code)). 2. Total number of bugs.

Phase 5: Maintenance

-measuring maintenance phase quality [12]

Metric Name	Metric	What does it do?	Data needs to be collected
Full availability (FA)	$FA = (NYSerH - NYFH) / NYSerH$	Measures software system's availability by how long it is available or able to perform the desired functions. The effectiveness of the program increases with its length of availability.	<ol style="list-style-type: none"> 1. NYSerH = The total available service hours of the system in a year. 2. NYFH = Total number of hours in a year when at least one function is unavailable or fails.

14. Project Team Quality Responsibilities

Name	Role, Task	Signature	Date
Norah Almubarak	Developer, Integration Control		20-4-2024
Lujain Albattah	Developer, Verification Tests		20-4-2024
Aljawharah Alotaibi	Quality Assurance Member, Planning Reviews		20-4-2024
Yara Aljasir	Quality Assurance Member, Acceptance Tests		20-4-2024
Raghad Alotaibi	Quality Assurance Member, Configuration Management		20-4-2024
Sarah Alshali	Tester, Validation Tests		20-4-2024

References:

- [1] Quality Requirement Documentation Guidelines for Agile Software ... (n.d.-f). <https://ieeexplore.ieee.org/document/9810243/>
- [2] (n.d.). *Types of Software testing*. GeekForGeeks. <https://www.geeksforgeeks.org/types-software-testing/>
- [3] Heads-Up: Amazon S3 Security Changes Are Coming in April of 2023 | AWS News Blog Anon 2022 aws.amazon.com <https://aws.amazon.com/blogs/aws/heads-up-amazon-s3-security-changes-are-coming-in-april-of-2023/>
- [4] The Types of Testing That Lead Your Fitness App to Perfection
Kate Zalozna 2020 Medium <https://kate-ponomarenko.medium.com/the-types-of-testing-that-lead-your-fitness-app-to-perfection-d62b59b08acf>
- [5] Biolchini, J., Mian, P. G., Natali, A. C. C., & Travassos, G. H. (2005, May 5). *Systematic Review in Software Engineering*. PESCE. https://edisciplinas.usp.br/pluginfile.php/7930850/mod_resource/content/5/ConceitosRevisaoSistematica_Biolchini.pdf
- [6] (n.d.). *Amazon S3 backups*. AWS. <https://docs.aws.amazon.com/aws-backup/latest/devguide/s3-backups.html>
- [7] Jain, A. (2023, September 11). *How to measure requirements quality*. Visure Solutions. <https://visuresolutions.com/requirements-management-traceability-guide/how-to-measure-requirements-quality/>
- [8] Simplilearn. (2024, February 16). Top project metrics and process metrics in project management for 2023-24: Simplilearn. Simplilearn.com. <https://www.simplilearn.com/project-and-process-metrics-article#:~:text=Requirement%20Stability%20Index%3A%20Provides%20visibility,for%20a%20unit%20of%20input.>

[9] "System Analysis & Design - System Design," 28 July 2021. [Online]. Available:

https://www.tutorialspoint.com/system_analysis_and_design/system_design.html

[10] A. Tyler and R. E. Glasgow, "Implementing Improvements: Opportunities to Integrate Quality Improvement and Implementation Science," *Hospital Pediatrics*, vol. 11, no. 5, pp. 536–545, May 2021, doi: <https://doi.org/10.1542/hpeds.2020-002246>.

[11] "GeeksforGeeks. (n.d.). Software Testing - Defect Density, from <https://www.geeksforgeeks.org/software-testing-defect-density/> "

[12] 9, O., & Raza, M. (2019, October 9). *System reliability & availability calculations*. BMC Blogs. <https://www.bmc.com/blogs/system-reliability-availability-calculations/>

[13] Son, H. (n.d.). *QA Metrics in Software Testing: Definitions, Types, Formulas & Examples*. TestRail. <https://www.testrail.com/ga-metrics/>

[14] Gnanasambandam, C., Harrysson, M., Alharith , H., Keovichit, J., & Srivastava, S. (2023, August 17). *Yes, you can measure software developer productivity*. McKinsey&Company. <https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/yes-you-can-measure-software-developer-productivity#/>

[15] (2023, May 4). *What is code reuse and why is it important?* OpsLevel. <https://www.opslevel.com/resources/what-is-code-reuse-and-why-is-it-important>