



Promoting IT Efficiency

QAIassist
Integrated Methodology
Deliverable Descriptions

Table of Contents

1.	CONTEXT.....	3
2.	INTEGRATED METHODOLOGY - PRE-CURSOR.....	4
2.1.1	<i>Business Case.....</i>	4
3.	QAIASSIST - PROJECT MANAGEMENT (PM)	5
3.1.1	<i>Project Charter.....</i>	5
3.1.2	<i>Project Plan/Schedule.....</i>	5
3.1.3	<i>Project Roles & Responsibilities.....</i>	5
3.1.4	<i>Project Deliverables.....</i>	5
3.1.5	<i>Project Configuration Management Plan.....</i>	6
3.1.6	<i>Project Quality Assurance Plan.....</i>	6
3.1.7	<i>Project Procedures.....</i>	6
3.1.8	<i>Project Risk Definition Form.....</i>	6
3.1.9	<i>Project Risk Log.....</i>	6
3.1.10	<i>Project Issues Definition Form.....</i>	6
3.1.11	<i>Project Issue Log.....</i>	7
3.1.12	<i>Project Change Request Form.....</i>	7
3.1.13	<i>Project Change Request Log.....</i>	7
3.1.14	<i>Project Status Report.....</i>	7
3.1.15	<i>Unit Test (UT) Authorization.....</i>	7
3.1.16	<i>System Integration Test (SIT) Authorization.....</i>	7
3.1.17	<i>User Acceptance Test (UAT) Authorization.....</i>	8
3.1.18	<i>Project Close-out Report.....</i>	8
4.	SOFTWARE DEVELOPMENT	9
4.1.1	<i>Detailed Business Requirements.....</i>	9
4.1.2	<i>Requirements Traceability Matrix.....</i>	9
4.1.3	<i>High Level Solution Design.....</i>	9
4.1.4	<i>Detail Solution Design.....</i>	9
4.1.5	<i>Programming Specification.....</i>	9
4.1.6	<i>Training and Support Plan.....</i>	10
5.	TESTING	11
5.1.1	<i>Testing Strategy.....</i>	11
5.1.2	<i>User Acceptance Test (UAT) Plan.....</i>	11
5.1.3	<i>User Acceptance Test (UAT) Evaluation.....</i>	11
5.1.4	<i>User Acceptance Test (UAT) Defect Log.....</i>	11
5.1.5	<i>System Integration Test (SIT) Plan.....</i>	12
5.1.6	<i>System Integration Test (SIT) Evaluation.....</i>	12
5.1.7	<i>System Integration Test (SIT) Defect Log.....</i>	12
5.1.8	<i>Unit Test (UT) Plan.....</i>	12
5.1.9	<i>Unit Test (UT) Evaluation.....</i>	12
5.1.10	<i>Unit Test (UT) Defect Log.....</i>	13

1. CONTEXT

This document provides an overview of the QAIassist Integrated Methodology.

A description has been provided for each and every deliverable of the methodology - the deliverable descriptions are arranged in accordance with the appropriate QAIassist methodology (Project Management, Software Development, Software Testing).

2. INTEGRATED METHODOLOGY - PRE-CURSOR

The QAassistant Integrated Methodology has been created to provide organizations an aid to effectively and consistently deliver quality business solutions in a timely manner. It relies on a continued and combined effort between business and technical resources throughout the life of a project.

Prior to utilizing the disciplines and deliverables of the QAassistant Integrated Methodology, a formal business need must exist. Once identified, the business need can be clarified, and authorized. Once authorized, the business need provides the information necessary to make a decision about whether a project should proceed. It provides an analysis of the costs, benefits, and risks associated with a proposed investment and offers reasonable alternatives and a recommended solution. Once approved, it provides a baseline to monitor progress and measure results.

2.1.1 Business Case

The *Business Case* deliverable is used to identify, document and initiate a project. It originates out of a business need and acts to provide a high level description of the business requirements. It is used as an entry point into the organizational integrated (project management, development, testing) methodology and is referred to throughout the life of the project.

3. QAIASSIST - PROJECT MANAGEMENT (PM)

The QAIassist Project Management (PM) Methodology is dependent on having authorization that a business need does exist, a Business Case has been documented, and the necessary Stakeholders have provided formal approval and authorization to initiate a project.

The QAIassist Project Management (PM) Methodology focuses on the overall management, oversight, and delivery of a project - this includes initiating, planning, executing, controlling and closing a project. The Project Management Methodology defines four (Initiate, Plan, Execute & Control, and Closeout) unique phases. Specific deliverables exist within each PM phase (Initiate, Plan, Execute & Control, and Closeout). Progression and iterations through the PM phases and deliverables is dependent on the conditions and characteristics of each unique project.

3.1.1 Project Charter

The ***Project Charter*** deliverable is used to establish a formal project. It is the initial deliverable prepared for a project and defines why the project was initiated, the scope of the project, the purpose & objectives of the project, the project milestones and a high level estimate on the effort and cost associated with the project. The ***Project Charter*** acts as the "footing" for the project.

3.1.2 Project Plan/Schedule

The ***Project Plan/Schedule*** deliverable provides an overview of how the project will be performed from start through completion. It defines the milestones, the deliverables to be completed, the risks associated with the project, the activities to be performed, and the estimated costs associated with the project. The ***Project Plan/Schedule*** acts as "map" for how the project will be completed.

3.1.3 Project Roles & Responsibilities

The ***Project Roles & Responsibilities*** deliverable identifies and describes the specific roles required to complete the project. It defines the project organizational structure, the "titles" of the project resources, and the roles and activities to be performed by the project team members. The ***Project Roles & Responsibilities*** deliverable acts to identify "who" will be contributing to the project.

3.1.4 Project Deliverables

The ***Project Deliverables*** deliverable identifies all of the deliverables that will be completed by the project team. Each of the project deliverables to be completed are described. Once identified, these project deliverables act as the basis for creating the project work breakdown structure (WBS) and schedule.

3.1.5 Project Configuration Management Plan

The *Project Configuration Management Plan* deliverable defines how the project "deliverables " and "configuration items" will be identified, maintained, distributed and audited throughout the life of the project. The **Project Configuration Management Plan** ensures the project will have the procedures in place to ensure there is integrity in the product being delivered.

3.1.6 Project Quality Assurance Plan

The *Project Quality Assurance Plan* deliverable identifies how "quality" is going to be incorporated into the end product and the processes used to create the end product. The **Project Quality Assurance Plan** ensures specific "quality assurance" activities (reviews, product and process audits) are being performed throughout the life of the project.

3.1.7 Project Procedures

The *Project Procedures* deliverable is used to define all of the procedures the project team will utilize throughout the life of the project. *Project Procedures* can range from approving deliverables to migrating code to reporting status. The *Project Procedures* ensure the project team has applied an appropriate level of rigor for administering the project.

3.1.8 Project Risk Definition Form

The *Project Risk Definition Form* deliverable is used to identify and describe "risks" that may force the project to an immediate conclusion due to schedule, costs, quality. The *Project Risk Definition Form* deliverable ensures all of these "risks" are identified, monitored and mitigated throughout the project life.

3.1.9 Project Risk Log

The *Project Risk Log* deliverable is used to identify and mitigate risk throughout the life of the project. Risks associated with scope, cost and quality are monitored throughout the life of the project ensuring the project will deliver according to its original objectives. The *Project Risk Log* is used to monitor and mitigate any factors that will have an adverse effect on the scope, cost, and quality of the project.

3.1.10 Project Issues Definition Form

The *Project Issue Definition Form* deliverable is used to identify and describe "issues" that may force the project to an immediate conclusion due to schedule, costs, quality. The *Project Issue Definition Form* deliverable ensures all of these "issues" are identified, monitored and can be mitigated throughout the life of the project.

3.1.11 Project Issue Log

The ***Project Issue Log*** deliverable is used to identify and mitigate "issues" throughout the life of the project. Issues associated with project scope, team procedures, communication, schedule, roles are identified and logged. The ***Project Issue Log*** is used to monitor and mitigate any factors that will have an adverse on the performance of the project team.

3.1.12 Project Change Request Form

The ***Project Change Request*** deliverable is used to identify and describe changes that may append, change or delete functionality of the system as it is being developed. As the project evolves through the lifecycle, new or changing information may alter the requirements for the system. These changes must be administered to ensure the end product reflects the business need. The ***Project Change Request*** deliverable ensures all of these changes are documented and can be assessed and "approved" changes can be built into the system prior to it being placed in the production environment.

3.1.13 Project Change Request Log

The ***Project Change Request Log*** is used to identify and monitor all additional or altered functionality to be incorporated into the project. The ***Project Change Request Log*** ensures the evolving user requirements are documented throughout the life of the project. Once documented, they can be assessed as to incorporating them in the original project or being incorporated in a later release.

3.1.14 Project Status Report

The ***Project Status Report*** deliverable is used to monitor and communicate the progress of the project and also acts as the mechanism to identify project "risks", "issues", "change requests". The ***Project Status Report*** is applicable to evaluating the projects actual progress against the planned progress and can be rolled up to provide a status to the Project Owner/Sponsor.

3.1.15 Unit Test (UT) Authorization

The ***Unit Test Authorization*** deliverable signifies the "approval" that all project coding, deliverables and work products have satisfied the required ***Unit Test Evaluation Criteria*** and can be migrated to the System Integration Test (SIT) environment. Approval of the ***Unit Test Authorization*** deliverable ensures all appropriate items are placed under proper configuration management and can be used as the initial basis for performing the next (SIT) level of testing.

3.1.16 System Integration Test (SIT) Authorization

The ***System Integration Testing (SIT) Authorization*** deliverable signifies the "approval" that all project coding, deliverables and work products have satisfied the required ***System***

Integration Test Evaluation Criteria and can be migrated to the User Acceptance Test (UAT) environment. Approval of the ***System Integration Test Authorization*** deliverable ensures all appropriate items are placed under proper configuration management and can be used as the initial basis for performing the next level (UAT) of testing.

3.1.17 User Acceptance Test (UAT) Authorization

The ***User Acceptance Test (UAT) Authorization*** deliverable signifies the "approval" that all project coding, deliverables and work products have satisfied the required ***User Acceptance Test Evaluation Criteria*** and can be migrated to the production (live for user) environment. Approval of the ***User Acceptance Test Authorization*** deliverable ensures all appropriate items are placed under proper configuration management and can be used as the initial basis for monitoring any additional maintenance or functionality on the application.

3.1.18 Project Close-out Report

The ***Project Close-out Report*** deliverable is used to provide an objective assessment of how the project evolved. It documents the "favorable" and "unfavorable" aspects of the project. The ***Project Close-out Report*** is intended to assist future project teams with "lessons learned".

4. SOFTWARE DEVELOPMENT

The QAassist Software Development (SD) Methodology focuses on defining, designing, building, testing and delivering a business solution. The QAassist Software Development Methodology defines five (Systems Analysis, Design, Build, Test and Release) unique phases. Specific deliverables exist within each SD phase (Systems Analysis, Design, Build, Test and Release). Progression and iterations through the SD phases and deliverables is dependent on the conditions and characteristics of each unique project.

4.1.1 Detailed Business Requirements

The ***Detailed Business Requirements*** deliverable is used to provide clarity on the business need that is to be addressed by the application. The ***Detailed Business Requirements*** deliverable provides the development team the business parameters they will use to deliver the necessary business functionality.

4.1.2 Requirements Traceability Matrix

The ***Requirements Traceability Matrix*** deliverable is used to ensure all user defined requirements are documented and incorporated into the application/system. The ***Requirements Traceability Matrix*** acts as inventory of all user requirements which can be referenced to ensure the final application/system satisfies the user needs.

4.1.3 High Level Solution Design

The ***High Level Solution Design*** deliverable is used to define the boundaries of the application to be delivered. The ***High Level Solution Design*** illustrates the data and process flows, the high level functionality to be incorporated in the application, the sub-subsystems and functions required to satisfy the business needs of the application, and the standards to be applied in developing the application.

4.1.4 Detail Solution Design

The ***Detail Solution Design*** deliverable(s) are an extension of the ***High Level Design*** deliverable - each function defined in the ***High Level Solution Design*** design is further clarified with a separate and unique ***Detail Solution Design*** deliverable. Each and all of the specific functions necessary to deliver the business requirements are identified and documented. Each of the ***Detail Solution Design*** deliverables addresses the necessary (functional, technical and administrative) activities to be incorporated into the application, the computer programs that provide that functionality, and interfaces with other application functions.

4.1.5 Programming Specification

The ***Programming Specification*** deliverable(s) are an extension of the ***Detail Solution Design*** deliverables - each program/module defined in a ***Detail Solution Design*** deliverable is further clarified a unique ***Programming Specification*** deliverable. Each ***Programming Specification*** deliverable defines the purpose and context for the program/module, the

environment it will operate in, and the detailed design to be incorporated into the program/module.

4.1.6 Training and Support Plan

The *Training and Support Plan* deliverable provides the description of how the end users are going to be trained in using the final application/product and the support they will receive once the application has been made operational. It specifies the methods of training, the required curriculum, the course content to be delivered, and mechanisms used to deliver the training.

5. TESTING

The QAIassist Software Testing (Testing) Methodology focuses on identifying the business solution criteria, verifying the business solution reflects the business requirements, and validation that the functionality addresses the business need. The QAIassist Software Development Methodology defines five (Systems Analysis, Design, Build, Test and Release) unique phases. Specific deliverables exist within each "Testing" phase (Systems Analysis, Design, Build, Test and Release). Progression and iterations through the "Testing" phases and deliverables is dependent on the conditions and characteristics of each unique project.

5.1.1 Testing Strategy

The **Testing Strategy** deliverable defines how all of the testing activities are to be executed throughout the life of the project. It identifies the testing tasks to be completed in each of the testing environments (unit, integration, user acceptance) the testing standards to be applied through with all testing environments, the testing tools to be used, the testing deliverables to be completed, and the acceptance criteria used for testing.

5.1.2 User Acceptance Test (UAT) Plan

The **User Acceptance Test (UAT) Plan** deliverable defines how all of the testing activities are to be executed prior to the application/product being migrated into the production environment. It identifies the testing tasks to be completed in User Acceptance Testing environment (unit, integration), the testing standards to be applied within the User Acceptance Testing environment, the testing tools to be used, the testing deliverables to be completed, and the acceptance criteria used for User Acceptance testing.

5.1.3 User Acceptance Test (UAT) Evaluation

The **User Acceptance Test (UAT) Evaluation** deliverable is used to document the "expected" user acceptance criteria results against the "actual" User Acceptance test results. Functionality that "passes" these tests is ready for the production environment. Functionality that "fails" these tests are recorded and require further activity from the project team.

5.1.4 User Acceptance Test (UAT) Defect Log

The **User Acceptance Test (UAT) Defect Log** deliverable is used to document and monitor all of the "failed" tests from the User Acceptance Evaluation. Each "failed" test is assessed and communicated to the project team who are required to make the necessary changes to rectify the "failed" test.

5.1.5 System Integration Test (SIT) Plan

The ***Systems Integration Test (SIT) Plan*** deliverable defines how all of the testing activities are to be executed prior to the application/product being migrated into the user acceptance test (UAT) environment. It identifies the testing tasks to be completed in System Integration Testing environment (unit, integration), the testing standards to be applied within the System Integration Testing environment, the testing tools to be used, the testing deliverables to be completed, and the acceptance criteria used for the System Integration testing.

5.1.6 System Integration Test (SIT) Evaluation

The ***System Integration Test (SIT) Evaluation*** deliverable is used to document the "expected" System Integration Test Acceptance criteria results against the "actual" Systems Integration test results. Functionality that "passes" these tests is ready for the User Acceptance testing environment. Functionality that "fails" these tests are recorded and require further activity from the project team.

5.1.7 System Integration Test (SIT) Defect Log

The ***System Integration Test (SIT) Defect Log*** is used to document and monitor all of the "failed" tests from the ***System Integration Test Evaluation Criteria*** deliverable. Each "failed" test is assessed and communicated to the project team who are required to make the necessary changes to rectify the "failed" test.

5.1.8 Unit Test (UT) Plan

The ***Unit Test (UT) Plan*** deliverables (one per program/module) defines how all of the testing activities are to be executed prior to the application/product being migrated into the System Integration Test (SIT) environment. It identifies the testing tasks to be completed in Unit Testing environment, the testing standards to be applied within the Unit Test environment, the testing tools to be used, the testing deliverables to be completed, and the acceptance criteria used for the unit testing.

5.1.9 Unit Test (UT) Evaluation

The ***Unit Test (UT) Evaluation*** deliverable is used to document the "expected" Unit Test Acceptance criteria results against the "actual" Unit Test results. Functionality that "passes" these tests is ready to be migrated to the System Integration Testing (SIT) environment. Functionality that "fails" these tests are recorded and require further activity from the project team.

5.1.10 Unit Test (UT) Defect Log

The ***Unit Test (UT) Defect Log*** is used to document and monitor all of the "failed" tests from the ***Unit Test Evaluation*** deliverable. Each "failed" test is assessed and communicated to the project team who are required to make the necessary changes to rectify the "failed" test.

*QAassist is the industry recognized benchmark in information technology (IT) methodologies for small and mid-sized business (SMB's) – including the certification and support of practitioners delivering QAassist IT Methodology solutions. Visit [QAassist's website](http://www.qaassist.com)—
www.qaassist.com*