



Promoting IT Efficiency

Integrated Methodology Deliverable Descriptions



Copyright © 2013, QAlassist C2013

This publication may not be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the copyright holder.



Table of Contents

1.	CONTEXT	4
2.	CONTEXT DIAGRAM.....	5
3.	INTEGRATED METHODOLOGY - PRE-CURSOR.....	6
3.1.1	<i>Business Case.....</i>	<i>6</i>
4.	QAIIASSIST - PROJECT MANAGEMENT (PM).....	7
4.1.1	<i>Project Charter</i>	<i>7</i>
4.1.2	<i>Project Plan</i>	<i>7</i>
4.1.3	<i>Project Schedule (Work Breakdown Structure – WBS)</i>	<i>7</i>
4.1.4	<i>Project Roles & Responsibilities.....</i>	<i>7</i>
4.1.5	<i>Project Deliverables</i>	<i>8</i>
4.1.6	<i>Project Configuration Management Plan</i>	<i>8</i>
4.1.7	<i>Project Quality Assurance Plan.....</i>	<i>8</i>
4.1.8	<i>Project Procedures.....</i>	<i>8</i>
4.1.9	<i>Project Risk Definition Form.....</i>	<i>8</i>
4.1.10	<i>Project Risk Log</i>	<i>8</i>
4.1.11	<i>Project Issues Definition Form</i>	<i>9</i>
4.1.12	<i>Project Issue Log.....</i>	<i>9</i>
4.1.13	<i>Project Change Request Definition</i>	<i>9</i>
4.1.14	<i>Project Change Request Log</i>	<i>9</i>
4.1.15	<i>Project Status Report</i>	<i>9</i>
4.1.16	<i>Team Status Report</i>	<i>9</i>
4.1.17	<i>Unit Test (UT) Authorization.....</i>	<i>10</i>
4.1.18	<i>System Integration Test (SIT) Authorization</i>	<i>10</i>
4.1.19	<i>User Acceptance Test (UAT) Authorization</i>	<i>10</i>
4.1.20	<i>Project CloseOut Report</i>	<i>10</i>
5.	SOFTWARE DEVELOPMENT	11
5.1.1	<i>Detailed Business Requirements</i>	<i>11</i>
5.1.2	<i>Requirements Traceability Matrix.....</i>	<i>11</i>
5.1.3	<i>High Level Solution Design</i>	<i>11</i>
5.1.4	<i>Detailed Solution Design.....</i>	<i>11</i>
5.1.5	<i>Programming Specification</i>	<i>12</i>
5.1.6	<i>Code</i>	<i>12</i>
5.1.7	<i>Training and Support Plan.....</i>	<i>12</i>
5.1.8	<i>Unit Test (UT) Defect Log</i>	<i>12</i>
5.1.9	<i>Unit Test (UT) Authorization.....</i>	<i>12</i>
6.	TESTING.....	13
6.1.1	<i>Testing Strategy</i>	<i>13</i>
6.1.2	<i>User Acceptance Test (UAT) Plan</i>	<i>13</i>
6.1.3	<i>User Acceptance Test (UAT) Evaluation Criteria.....</i>	<i>13</i>
6.1.4	<i>User Acceptance Test (UAT) Defect Log.....</i>	<i>14</i>
6.1.5	<i>User Acceptance Test (UAT) Authorization</i>	<i>14</i>
6.1.6	<i>System Integration Test (SIT) Plan</i>	<i>14</i>
6.1.7	<i>System Integration Test (SIT) Evaluation Criteria.....</i>	<i>14</i>
6.1.8	<i>System Integration Test (SIT) Defect Log.....</i>	<i>14</i>
6.1.9	<i>System Integration Test (SIT) Authorization</i>	<i>14</i>
6.1.10	<i>Unit Test (UT) Plan.....</i>	<i>15</i>
6.1.11	<i>Unit Test (UT) Evaluation Criteria</i>	<i>15</i>

1. CONTEXT

This document provides an overview of the QAssist Integrated Methodology.

A description has been provided for the deliverables of the methodology - the deliverable descriptions are arranged according to the lifecycles (Project Management, Software Development, Software Testing) of the methodology.

QAIassist Integrated Methodology

	Initiate	Plan	Execute & Control	Closeout	
Project Management	<p>Business Case Detailed Business Req'ts</p>	<p>Project Charter Project Plan/Schedule/WBS Roles & Responsibilities Project Deliverables Config. Management Plan Quality Assurance Plan Project Procedures</p>	<p>All previously created PM deliverables Project Risk Definition Project Risk Log Project Issue Definition Project Issue Log Project Change Request Definition Project Change Request Log</p>	<p>Team Status Report Project Status Report Unit Test (UT) Authorization System Integration Test (SIT) Authorization User Acceptance Test (UAT) Authorization</p>	<p>Project Closeout</p>
Software Development	<p>Entry Deliverables Business Case Project Charter Detailed Business Requirements</p> <p>Exit Deliverables High Level Solution Design Reqmt's Traceability Log</p>	<p>Entry Deliverables High Level Solution Design Reqmt's Traceability Log</p> <p>Exit Deliverables Detailed Solution Design Programming Specifications Reqmt's Traceability Log</p>	<p>Entry Deliverables Detailed Solution Design Programming Specifications Reqmt's Traceability Log</p> <p>Exit Deliverables Code Training & Support Plan Reqmt's Traceability Log</p>	<p>Entry Deliverables Code (UT) Plan (UT) Evaluation Criteria Reqmt's Traceability Log</p> <p>Exit Deliverables (UT) Evaluation Criteria (UT) Defect Log (UT) Authorization Reqmt's Traceability Log</p>	<p>Entry Deliverables (UT) Authorization (UAT) Plan</p> <p>Exit Deliverables (UAT) Evaluation Criteria (UAT) Defect Log (UAT) Authorization Reqmt's Traceability Log</p>
Software Testing	<p>Entry Deliverables Business Case Project Charter Detailed Business Reqmt's</p> <p>Exit Deliverables Testing Strategy (UAT) Plan (UAT) Evaluation Criteria</p>	<p>Entry Deliverables High Level Solution Design Testing Strategy</p> <p>Exit Deliverables (SIT) Plan (SIT) Evaluation Criteria</p>	<p>Entry Deliverables Detailed Solution Design Programming Specifications Reqmt's Traceability Log</p> <p>Exit Deliverables (UT) Plan (UT) Evaluation Criteria Reqmt's Traceability Log</p>	<p>Entry Deliverables (UT) Authorization (SIT) Plan (SIT) Evaluation Criteria</p> <p>Exit Deliverables (SIT) Evaluation Criteria (SIT) Defect Log (SIT) Authorization Reqmt's Traceability Log</p>	<p>Entry Deliverables (SIT) Authorization (UAT) Plan</p> <p>Exit Deliverables (UAT) Evaluation Criteria (UAT) Defect Log (UAT) Authorization Reqmt's Traceability Log</p>
	Systems Analysis	Design	Build	Test	Release

2. CONTEXT DIAGRAM

3. INTEGRATED METHODOLOGY - PRE-CURSOR

The QAlassist Integrated Methodology has been created to provide organizations an aid to effectively and consistently deliver quality applications and 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 QAlassist 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.

3.1.1 Business Case

The **Business Case** deliverable is used to identify, document and establish a project definition. 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 QAlassist Integrated (project management lifecycle, software development lifecycle, software testing lifecycle) Methodology and is referred to throughout the life of the project.

4. QAI ASSIST - PROJECT MANAGEMENT (PM) LIFECYCLE

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

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

The QAI Assist Project Management (PM) lifecycle is applicable to a host of environments (IT and non IT). It is also integrated and can be leveraged to manage the QAI Assist Software Development (SD) lifecycle and the QAI Assist Software Testing (ST) lifecycle.

4.1.1 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.

4.1.2 Project Plan

The **Project Plan** deliverable is used to provide an overview of how the project will be performed from start through completion. It defines the milestones, the plans (ie QA, CM, Training) 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 acts as "map" for how the project will be completed.

4.1.3 Project Schedule (Work Breakdown Structure – WBS)

The **Project Schedule** deliverable is used to define and prioritize the activities and deliverables to be completed by the project team throughout the completion of the project. It is also used to monitor and communicate the progress and status of the project team against those activities and deliverables.

4.1.4 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, the roles to be performed by the project team members, and the responsibilities of each project team member.

4.1.5 Project Deliverables

The **Project Deliverables** deliverable identifies and defines all of the deliverables that will be worked on and completed by the project team. The project deliverables defined within this deliverable are reflected on the **Project Schedule** (work breakdown structure - WBS) deliverable.

4.1.6 Project Configuration Management Plan

The **Project Configuration Management Plan** deliverable defines how the project "deliverables" and "configuration items" will be maintained, distributed and audited throughout the life of the project. The **Project Configuration Management Plan** ensures there will be an appropriate level of rigor applied to ensure the authorized project work products, artifacts and deliverables are stable, secured, maintained and available.

4.1.7 Project Quality Assurance Plan

The **Project Quality Assurance Plan** deliverable identifies the processes, procedures and protocols that will be employed by the project team to ensure an appropriate level of rigor is applied to the "quality" throughout the life of the project. The **Project Quality Assurance Plan** defines the specific "quality assurance" activities (reviews, product and process audits) that will be performed by the project team.

4.1.8 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** are used to guide how the project team will work together to complete the project. Depending on the needs of the project team, project procedures can range from approving deliverables to migrating code to reporting status. The **Project Procedures** ensure the project team applies an appropriate level of rigor in administering the project.

4.1.9 Project Risk Definition Form

The **Project Risk Definition** deliverable is used to identify and describe a specific "risk" that may affect the project schedule, costs, and quality. The Project Risk Definition deliverable ensures that project team members have the means to identify and document a "risk" associated with the project. Once each "risk" is identified and documented it can be monitored and mitigated throughout the life of the project.

4.1.10 Project Risk Log

The **Project Risk Log** deliverable is used to identify and mitigate all (a collection of all the identified "risks") of the project "risks" 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 objectives. The **Project Risk Log** is a repository that contains a summary of each and all the individually identified risks associated with the project – it is used to document and monitor the status of all project risks (scope, cost, and quality).

4.1.11 Project Issues Definition Form

The **Project Issue Definition Form** deliverable is used to identify and describe a specific "issue" requiring resolution by the project team – left unaddressed “issues” can become elevated into Project Risks. The Project Issue Definition deliverable ensures each separate "issue" is identified. Once identified each “issue” is documented, monitored and mitigated throughout the life of the project.

4.1.12 Project Issue Log

The **Project Issue Log** deliverable is used to identify and mitigate all (a collection of all the identified “issues”) of the project “issues” throughout the life of the project. Issues associated with scope, cost and quality are monitored throughout the life of the project ensuring the project will deliver according to its objectives. The **Project Issue Log** is used to monitor and mitigate any and all issues that will have an adverse effect on the scope, cost, and quality of the project.

4.1.13 Project Change Request Definition

The **Project Change Control** deliverable is used to identify and describe changes that may append, change or delete functionality of the product/application as it is being developed. As the project evolves through the software development lifecycle and software testing lifecycle, new or changing information may alter the requirements for the application/system being developed. These changes must be administered to ensure the end product reflects the business need. The **Project Change Control** deliverable ensures all “suggested/recommended” functionality changes are documented - once documented, these changes can be assessed/evaluated. Authorized functionality changes can be built into the system/application prior to it being placed in the production environment or bundled and incorporated into another release of the application.

4.1.14 Project Change Request Log

The **Project Change Control Log** is used to identify and monitor all additional or required functionality to be incorporated into the project. The **Project Change Control 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.

4.1.15 Project Status Report

The **Project Status Report** deliverable is used to communicate the progress of the project - this includes project “risks”, “issues”, “change requests”. It illustrates actual progress against the planned progress and is used to provide a status to the Project Owner/Sponsor/Stakeholder.

4.1.16 Team Status Report

The **Team Status Report** deliverable is used to identify, monitor and communicate how project team members are progressing against the planned activities and deliverables that have been assigned to them – including “risks”, “issues”, “change requests”. The individual Team Status Reports are also used as the basis for completing the Project Status Report.

4.1.17 Unit Test (UT) Authorization

The **Unit Test Authorization** deliverable signifies "approval" that all the necessary application code has 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.

4.1.18 System Integration Test (SIT) Authorization

The **System Integration Testing (SIT) Authorization** deliverable signifies the "approval" that all project code has satisfied the required System Integration Test Evaluation Criteria and can be migrated to the User Acceptance Test (UAT) environment. Approval of this 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.

4.1.19 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 once it is in the production environment.

4.1.20 Project CloseOut Report

The **Project CloseOut 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 CloseOut Report** is intended to assist future project teams with "lessons learned".

5. SOFTWARE DEVELOPMENT (SD) LIFECYCLE

The QAlassist Software Development (SM) lifecycle is dependent on having authorization that a business need does exist, a **Business Case** deliverable has been documented, and the necessary Stakeholders have provided formal approval and authorization to initiate a project.

The QAlassist Software Development (SD) lifecycle focuses on defining, designing, building, and unit testing an application/business solution. The SD lifecycle consists of five (Systems Analysis, Design, Build, Test and Release) unique phases - specific deliverables exist within each phase. Progression and iterations through the Software Development Lifecycle phases and deliverables is dependent on the conditions and characteristics of each unique project.

The QAlassist Software Development (SD) lifecycle is closely integrated with the QAlassist Software Testing (ST) lifecycle – application requirements and design established using the SD Lifecycle are referenced and used as testing criteria within the ST Lifecycle.

5.1.1 Detailed Business Requirements

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

5.1.2 Requirements Traceability Matrix

The **Requirements Traceability Matrix** deliverable is used to ensure all user defined requirements are documented and incorporated into the application/system. It acts as the repository for all user requirements - it can be referenced and crosschecked to ensure all user requirements have been incorporated into the application before it is released into the production environment.

5.1.3 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 into 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.

5.1.4 Detailed Solution Design

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

5.1.5 Programming Specification

The **Programming Specification** deliverable(s) are an extension of the **Detailed Solution Design** deliverables - each program/module defined in a **Detailed Solution Design** deliverable is further clarified as 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. The **Programming Specification** deliverables act as the basis for developing the code for the application.

5.1.6 Code

The programming specification(s) are used to develop the application code. The code will incorporate the technical standards and all the necessary functionality as defined by the business requirements for each program/module.

5.1.7 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.

5.1.8 Unit Test (UT) Defect Log

The **Unit Test (UT) Defect Log** is used to document and monitor all of the "failed" tests against the **Unit 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.

5.1.9 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.

6. SOFTWARE TESTING (ST) LIFECYCLE

The QAlassist Software Testing (ST) lifecycle is dependent on having authorization that a business need does exist, a **Business Case** deliverable has been documented, and the necessary Stakeholders have provided formal approval and authorization to initiate a project.

The QAlassist Software Testing (ST) lifecycle focuses on identifying the business solution criteria, verifying the business solution reflects the business requirements, validation that the functionality addresses the business need, and delivering the required functionality into a production environment. The ST lifecycle consists of five (Systems Analysis, Design, Build, Test and Release) unique phases - specific deliverables exist within each phase. Progression and iterations through the Software Testing Lifecycle phases and deliverables is dependent on the conditions and characteristics of each unique project.

The QAlassist Software Testing (ST) Lifecycle is closely integrated with the QAlassist Software Development (SD) Lifecycle – testing criteria established and applied using the ST Lifecycle is derived from the requirements and design created from the SD Lifecycle.

6.1.1 Testing Strategy

The **Testing Strategy** deliverable lays out a global and holistic perspective of testing for the project/application – it defines the high level testing activities that will 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 across all testing environments, the testing tools to be used, the testing deliverables to be completed, and the standards used to identify the acceptance criteria used for testing.

6.1.2 User Acceptance Test (UAT) Plan

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

6.1.3 User Acceptance Test (UAT) Evaluation Criteria

The **User Acceptance Test (UAT) Evaluation Criteria** deliverable is used to document the "expected" User Acceptance Test evaluation criteria prior to conducting user acceptance testing. The UAT testing criteria is used to evaluate "expected" results versus "actual" results. Functionality that "passes" the user acceptance tests is ready to be migrated to the production environment. Functionality that "fails" these user acceptance tests are recorded and require further activity from the project team.

6.1.4 User Acceptance Test (UAT) Defect Log

The **User Acceptance Test (UAT) Defect Log** deliverable is used to document and monitor all of the User Acceptance "failed tests" (actual results based on testing versus expected results from the evaluation criteria). Each "failed" test is assessed and communicated to the project team who are required to make the necessary changes to rectify the "failed" test.

6.1.5 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.

6.1.6 System Integration Test (SIT) Plan

The **Systems Integration Test (SIT) Plan** deliverable defines how all of the System Integration testing activities are to be executed prior to the application/product being migrated into the User Acceptance environment. It identifies the testing standards to be applied within the SIT environment, the testing tools to be used, the testing deliverables to be completed, and the standards used to define the SIT evaluation criteria.

6.1.7 System Integration Test (SIT) Evaluation Criteria

The **System Integration Test (SIT) Evaluation Criteria** deliverable is used to document the "expected" SIT test evaluation criteria prior to conducting System Integration testing. The System Integration test criteria is used to evaluate "expected" results versus "actual" results. Functionality that "passes" the SIT tests is ready to be migrated to the User Acceptance Test (UAT) environment. Functionality that "fails" these SIT tests is recorded and requires further activity from the project team.

6.1.8 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.

6.1.9 System Integration Test (SIT) Authorization

The **System Integration Testing (SIT) Authorization** deliverable signifies the "approval" that all project code has satisfied the required System Integration Test Evaluation Criteria and can be migrated to the User Acceptance Test (UAT) environment. Approval of this 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.

6.1.10 Unit Test (UT) Plan

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

6.1.11 Unit Test (UT) Evaluation Criteria

The **Unit Test Evaluation Criteria** deliverables (one per program/module) is used to document the "expected" Unit Test evaluation criteria prior to conducting the Unit Testing. The Unit Test criteria is used to evaluate "expected" results versus "actual" results. Functionality that "passes" the UT tests is ready to be migrated to the System Integration Test (SIT) environment. Functionality that "fails" these Unit tests is recorded and requires further activity from the project team.