



Promoting IT Efficiency

Integrated Methodology Deliverable Descriptions



Copyright © 2011, QAlassist C2011

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

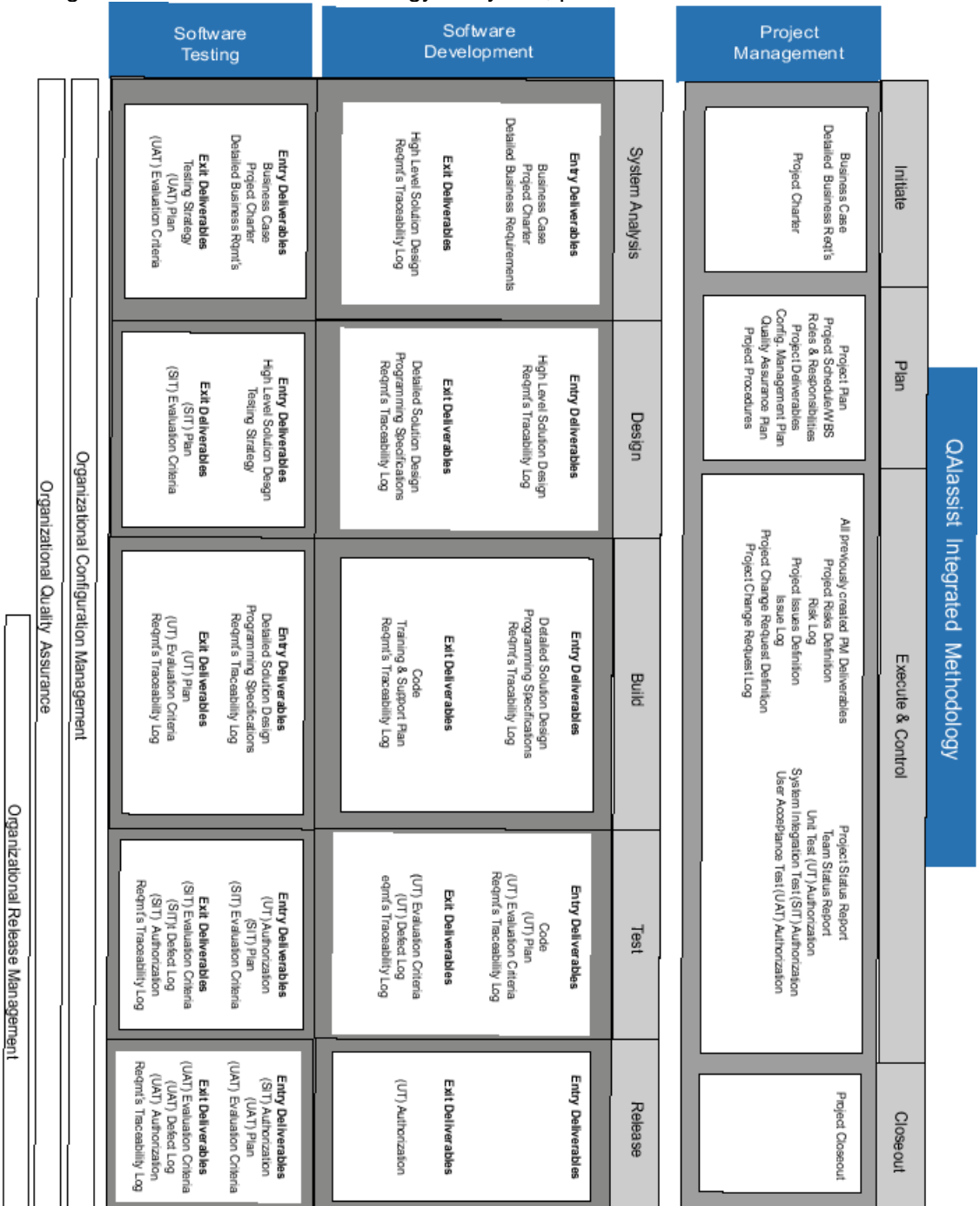
1. CONTEXT

This document provides an overview of the QAlassist Integrated Methodology.

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

2. CONTEXT DIAGRAM

The context diagram illustrates the methodology lifecycles, phases and deliverables.



© QAlassist 2011

3. INTEGRATED METHODOLOGY - PRE-CURSOR

The QAlassist 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 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, development, testing) 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 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 - this includes initiating, planning, executing, controlling and closing a project. To deliver this PM lifecycle, the QAI Assist Integrated 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.

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, and the roles to be performed by the project team members.

4.1.5 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 schedule (work breakdown structure - WBS).



4.1.6 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.

4.1.7 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.

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** 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 "risks" that may effect the project schedule, costs, quality. The Project Risk Log deliverable ensures all of these "risks" are identified, monitored and can be mitigated throughout the life of the project.

4.1.10 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 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.

4.1.11 Project Issues Definition Form

The **Project Issue Definition Form** deliverable is used to identify and describe "issues" and matters requiring resolution by the project team – left unaddressed "issues" can become elevated into Project Risks. The Project Issue Log deliverable ensures all of these "issues" are identified, monitored and can be mitigated throughout the life of the project.

4.1.12 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.

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 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 Control** 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.

4.1.14 Project Change Request Log

The **Project Change Control Log** is used to identify and monitor all additional or altered 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.

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.

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 (SD) lifecycle focuses on defining, designing, building, testing and releasing a business solution. To deliver this SD lifecycle, the QAlassist Integrated 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.

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 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.

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 lifecycle focuses on identifying the business solution criteria, verifying the business solution reflects the business requirements, and validation that the functionality addresses the business need. To deliver this "Software Testing" lifecycle, the QAlassist Integrated Methodology defines five (Systems Analysis, Design, Build, Test and Release) unique phases. Progression and iterations through the "Software Testing" phases and deliverables is dependent on the conditions and characteristics of each unique project.

6.1.1 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 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 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 within the UAT environment, 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. This 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 testing activities are to be executed prior to the application/product being migrated into the User Acceptance environment. It identifies the testing tasks to be completed in the SIT environment, 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. This 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 are recorded and require 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. This 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 are recorded and require further activity from the project team.

