



Promoting IT Efficiency

QAI
assist

White Paper

Integrated Methodology Overview



Copyright © 2010, QAlassist C2010

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.	PURPOSE	5
2.1	SCOPE	5
2.2	USERS AND BENEFITS	5
2.3	ORGANIZATIONAL STANDARDS.....	5
3.	CONTEXT DIAGRAM.....	6
4.	PROJECT MANAGEMENT (PM) - GUIDELINES.....	7
4.1	PM - PRE-REQUISITES	7
4.2	PM - UTILIZATION	7
4.3	PM - PHASES	7
4.3.1	<i>Initiate</i>	7
4.3.2	<i>Plan</i>	7
4.3.3	<i>Execute & Control</i>	8
4.3.4	<i>Closeout</i>	8
5.	SOFTWARE LIFECYCLE (DEVELOPMENT & TESTING) - GUIDELINES	9
5.1	SOFTWARE LIFECYCLE - PRE-REQUISITES	9
5.2	SOFTWARE LIFECYCLE - UTILIZATION	9
5.3	SOFTWARE LIFECYCLE - PHASES	9
5.3.1	<i>Systems Analysis</i>	9
5.3.2	<i>Design</i>	9
5.3.3	<i>Build</i>	11
5.3.4	<i>Test</i>	11
5.3.5	<i>Release</i>	11
5.4	SOFTWARE LIFECYCLE - INTEGRATION	11
5.4.1	<i>End User/Client</i>	11
5.4.2	<i>Project Management</i>	11
5.4.3	<i>Software Testing (Unit/System Integration/User Acceptance)</i>	12
5.4.4	<i>IT (Quality Assurance Process) Audit</i>	12

1. CONTEXT

The QAIassist Integrated Methodology provides an efficient and easy to use framework for increasing overall IT efficiency and quality of the software development processes and deliverables – it decreases development and maintenance costs while increasing customer satisfaction.

Additional benefits of the QAIassist Integrated Methodology include:

- explicitly linking management and software development, and maintenance/support activities to their business objectives
- expanding the scope of and visibility into the product lifecycle and software development, and maintenance/support activities to ensure that the product or service meets customer expectations
- incorporating lessons learned from additional areas of best practice (e.g., measurement, risk management)
- implementing more robust high-maturity practices
- addressing additional organizational functions critical to their products and services to more fully comply with relevant industry or regulatory audit standards

Every project that is initiated has its own unique set of circumstances and characteristics. For optimal results, the integrated methodology can be “tailored” to the needs of the specific project.



2. Purpose

This document defines the project management (initiate, plan, execute & control, closeout), and software development and testing (systems analysis, design, build, unit test, release) lifecycles.

2.1 Scope

The QAassistant Integrated Methodology consists of a comprehensive suite of deliverables that can be applied by project development and maintenance teams to consistently deliver quality applications on time and within budget. It is scalable and can be applied in whole or in part within multiple delivery (prototyping, agile, RAD, spiral, waterfall) environments.

2.2 Users and Benefits

This QAassistant Integrated Methodology can be leveraged by all corporate resources requiring information systems (IT) development/maintenance/support of their business applications. Business and Project Managers represent the primary owners of the methodology. Other audiences may include

IT Clients

- With transparency into the practices and methods applied to deliver (development, and maintenance/support) business solutions. IT Clients gain a further understanding of the methods/deliverables required to deliver and support their applications.

IT Development teams

- In designing and implementing new business applications. The methodology provides the business (user) community, project management and technical (IT) resources an inventory of deliverables in assisting the development of the business solution.

IT Maintenance/Support teams

- To ensure all the necessary business functions are incorporated into existing applications. The methodology provides an inventory of deliverables that can be customized to meet the business needs of specific (maintenance/support) projects.

IT (QA Process) Audit

- The pre-defined baseline suite of deliverables (work products) that are available within the methodology. An internal audit function will review the activities and deliverables being prepared and delivered by the project team against that baseline.

IT Senior Management

- Have a common mechanism for monitoring the utilization and efficiency of IT resources in the development and maintenance/support of business solutions.

2.3 Organizational Standards

The ABC Company Integrated Methodology incorporates two unique disciplines.

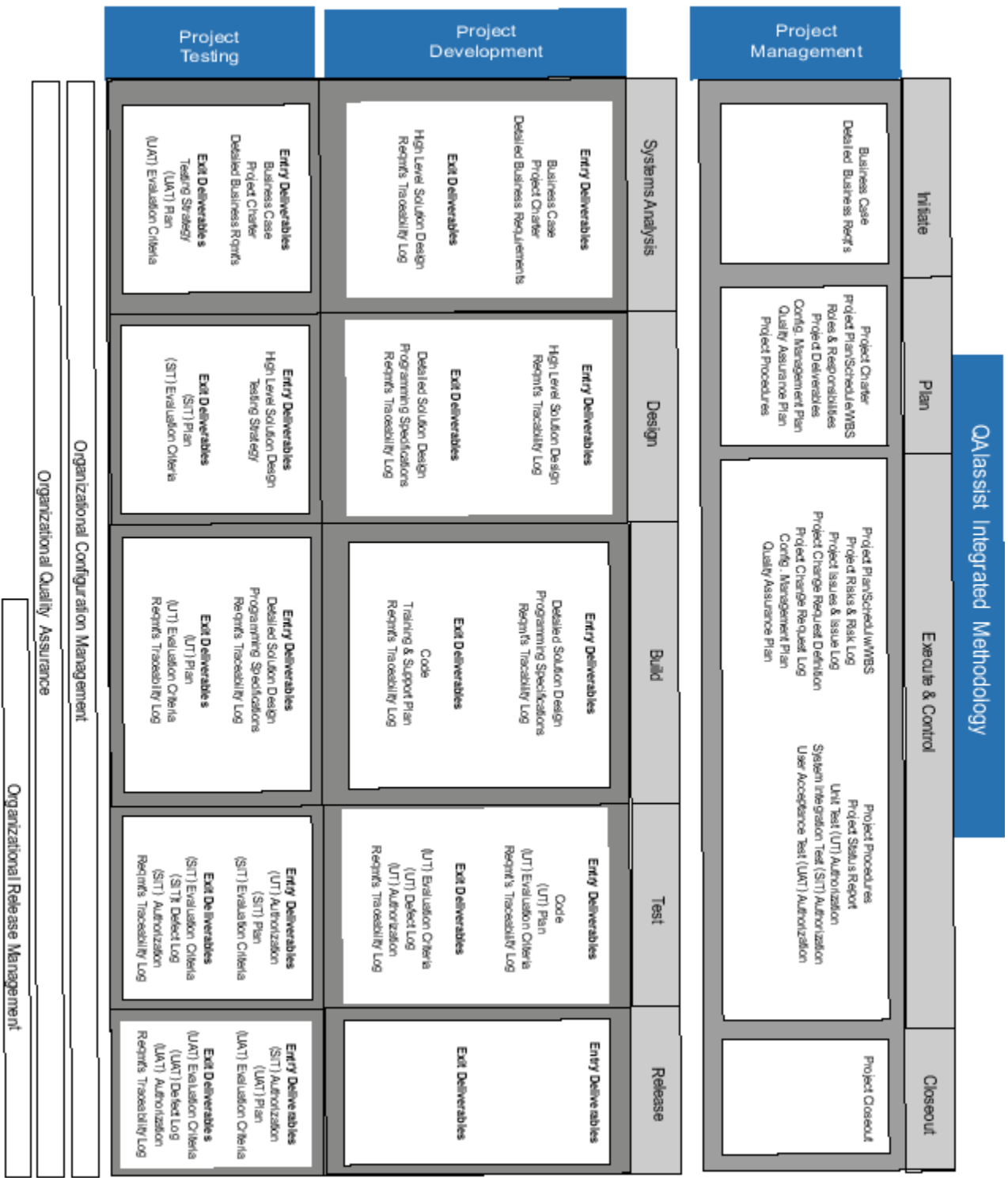
The **Project Management** discipline ensures the organization applies an appropriate level of planning and oversight on its projects - it consists of four specific phases (Initiate, Plan, Execute & Control, Closeout).

The **Software Lifecycle** discipline ensures project teams have the tools to consistently deliver quality applications on a timely basis - it consists of five specific (Systems Analysis, Design, Build, Test, Release) phases.



3. CONTEXT DIAGRAM

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



4. Project Management (PM) - Guidelines

4.1 PM - Pre-requisites

As a pre-requisite in defining a project, a **Business Case** must be developed and authorized.

Once the formal **Business Case** has been authorized, a Project Manager can begin work with members of the business community to initiate the project.

4.2 PM - Utilization

The Project Management discipline:

- Defines the objectives and scope of the project
- Identifies the resources and roles to be completed on the project
- Establishes a plan designed to ensure successful implementation of the business solution
- Assesses progress against project schedules, milestones and metrics
- Takes corrective action when there is a deviation from the project cost, schedule, quality

4.3 PM - Phases

The PM discipline consists of four (Initiate, Plan, Execute & Control, and Closeout) unique phases. Within each phase, PM deliverables have been identified as well as the activities required to complete these deliverables. Progression/iteration through the PM phases is dependent on obtaining the appropriate authorizations from the designated project stakeholders.

4.3.1 Initiate

Initiate is the first phase to be performed within the project management discipline. It is the process of formally recognizing that a project (new development or maintenance) exists and has obtained the appropriate level of authority to continue. The purpose of this phase is twofold. First, to assess and determine a business need. Second, to translate business requirements into notational form or models that will enable the project team to confirm the project requirements can be fulfilled. This is an iterative process between the business representative and the technical IT resources. The Project Manager performs this exercise and through analysis is able to define the **Business Case** and translate it into further granularity resulting in defining **Detailed Business Requirements**. This phase is considered complete when both the Business Case and the Detailed Business Requirements have been completed and the project has received the appropriate level of authorization to proceed.

4.3.2 Plan

Plan is the second phase to be performed within the project management discipline. It is the process of recognizing that a project (new development or maintenance) has been authorized and requires further resources to devise, maintain and execute a workable **Project Plan** to ensure the business solution is effectively implemented. The workable **Project Plan** must address project scope, resource requirements, project team roles, deliverables to be prepared throughout the project, a schedule to define how and when the project will be completed, and the activities to be applied to ensure "quality" is incorporated into the solution.

4.3.3 Execute & Control

Execute & Control is the third phase to be performed within the project management discipline. It is the process of coordinating the activities of the project team resources to ensure the project can be completed according to the **Project Plan**, monitoring the progress of the project activities against the **Project Plan**, and taking the appropriate corrective action when the project is deviating from **Project Plan**. The Project Manager prepares and utilizes a number of specific deliverables to ensure project procedures are available to the project team, the project management deliverables are maintained throughout the life of the project, deviations to scope, schedule and resources are addressed in a timely fashion.

4.3.4 Closeout

Closeout is the final project management phase. Its purpose is to document a true reflection of how the project evolves from start date through to its completion so that future projects can benefit from the knowledge and experience gained on the project. Future project teams can then leverage this knowledge to increase the efficiencies on delivering business solutions to their clients.

5. Software Lifecycle (Development & Testing) - Guidelines

The software lifecycle is composed of two integrate disciplines. First, a development lifecycle used to design and construct software. Second, a testing discipline that is applied to ensure the software being developed satisfies both the business need and the defined requirements.

5.1 Software Lifecycle - Pre-Requisites

As a pre-requisite into the Software LifeCycle (SLC) the **Business Case**, **Project Charter** and **Detailed Business Requirements** deliverables should be completed and authorized.

When the Business Case, Project Charter, and Detailed Business Requirements deliverables have been prepared and authorized project resources can be assigned and begin to perform the activities/deliverables of the SLC.

5.2 Software Lifecycle - Utilization

The software lifecycle provides the framework for:

- Analyzing business requirements
- Proposing a functional design to satisfy business requirements
- Designing technical architecture
- Creating/Testing technical code required to satisfy requirements

5.3 Software Lifecycle - Phases

5.3.1 Systems Analysis

The purpose of this Systems Analysis phase is two-fold. First, to translate business requirements into notational form or models that will enable the project team to confirm the project requirements can be fulfilled. This is an iterative process between the business representative and the technical IT resources. Second, to establish a strategy that will guide the testing activities on the project.

5.3.2 Design

The Design phase ensures the application is designed (functional & technical) according to authorized **Detailed Business Requirements** and **High Level Solution Design** deliverables established during System Analysis. This acts as the foundation for developing/constructing the application and also provides the criteria for unit testing the code for the application.

The Design phase focuses on the refinement of the data, application and technology models defined in the High-Level Solution Design deliverable, and incorporates other factors (e.g. data and non-functional requirements, testing strategy) that must be considered in designing the solution. The solution designed is refined to a level where individual software, hardware and data components are defined and documented.

The Design phase also is used to create the pre-defined testing criteria and conditions to be applied for system integration testing.

When this discipline is complete, it will be possible to generate comprehensive estimates for identifying the activity of resources for application development, maintenance/support and testing activities.



5.3.3 Build

The Build phase ensures the application is built according to the contents of the authorized ***Detailed Business Requirements, Requirements Traceability Log, and Programming Specifications*** deliverables completed during Design. This phase will also ensure the appropriate level of rigor is performed through utilization of the ***Unit Test Plan*** deliverable to ensure the application adheres to the authorized ***Detailed Solution Design*** deliverable. The aim is to produce readable, testable and maintainable code in the source control applications/systems.

5.3.4 Test

This phase ensures the application has successfully performed all the necessary activities defined in the ***Unit Test (UT) Plan*** and the ***System Integration Test (SIT) Plan***. The aim is to ensure the code that has been built in the development environment has been properly tested to ensure it satisfies all the cases and conditions defined with the ***Unit Test (UT) Evaluation Criteria*** and the ***System Integration Test (SIT) Evaluation Criteria***.

5.3.5 Release

The Release phase ensures the application has been tested according to the authorized User Test Criteria. The aim is to ensure the code that has been prepared in the development environment and has successfully passed through the Testing phase and can be assessed against the ***User Acceptance Evaluation Criteria***. Once satisfying the criteria the application can be placed into the production environment and utilized by the user community.

5.4 Software Lifecycle - Integration

The Software Development Lifecycle has been designed to promote integration, cohesion and communications of existing organizational functions. Including

5.4.1 End User/Client

End users/clients utilize the SLC to act as the interface to communicate with IT management and staff. They recognize the SLC provides them the mechanism they require to effectively define their business needs and requirements and that these requirements guide all IT development and maintenance effort. The SLC ensures linkage and communication exists between the business resources and the IT resources throughout the life of every project.

5.4.2 Project Management

Provides a foundation for the planning, monitoring, oversight and implementation of the required business requirements. Although development and maintenance/support projects differ in their characteristics, both must be planned, monitored and controlled. This includes identifying the deliverables that must be produced, the resources to work on those deliverables, the ongoing progress made on the deliverables, issues and risks associated with creating the deliverables, and application of organizational process (quality assurance, configuration management, etc). The SLC deliverables act as the basis for the Project Plan/Schedule – required deliverables are assigned by the project manager to the project team members.



5.4.3 Software Testing (Unit/System Integration/User Acceptance)

Although development and maintenance/support projects differ in their characteristics and the mechanisms required to ensure this safety/integrity, success in either environment is dependent on establishing a comprehensive set of business requirements (user participation) ensuring all conditions associated with those requirements have been sufficiently tested (including Performance & Capacity – P&C testing), and knowledge that all necessary release procedures have been applied prior to migration to the production environment. The SLC affords project teams (development and testing resources) the deliverables they require to ensure the product/application being delivered has been tested and adheres to the authorized requirements.

5.4.4 IT (Quality Assurance Process) Audit

The purpose of Quality Assurance (QA) is to provide management with appropriate visibility into the process being used by the software project and of the products being built and designed. The objective of organizational QA is to verify that SLC activities being performed by the project development and maintenance teams comply with the approved organization's SLC processes and procedures. The SLC provides the organizational QA function a baseline to conduct project audits (reviewing the completion of the SDL deliverables by the project team).

QAassist is the industry leader in "Promoting IT Efficiency" to small and mid-sized businesses (SMB's). Our scalable solutions provide SMB's the tools they required to increase operational performance and bottom line through greater "IT Efficiency". Our flagship "Integrated IT Methodology" product provides SMB's a cost-effective means to consistently deliver products, projects, and operations on time and within budget. We design, deliver and support our "IT Efficiency" solutions in affiliation with industry recognized IT Consulting Firms, IT Tool Vendors, and IT Training Organizations.

