Copyright © 2002-2012 QAvantage

All rights reserved. This documentation is protected by copyright and distributed under a license that restricts its use, copying, and distribution. No part of this documentation may be reproduced in any form or by any means without the prior written authorization of QAvantage. While every precaution has been taken in the preparation of this documentation, QAvantage assumes no responsibility for errors or omissions. This documentation and the features described herein are subject to change without notice. The information provided in this material may be changed or updated without notice. QAvantage may also make improvements and/or changes in the products and/or the programs described in this information at any time without notice.

QAvantage assumes no responsibility regarding the accuracy of the information that is provided by QAvantage and use of such information is at the recipient’s own risk. QAvantage provides no assurances that any reported problems may be resolved with the use of any information that QAvantage provides. By furnishing the information, QAvantage does not grant any licenses to any copyrights, patents or any other intellectual property right of QAvantage or its licensors.

RTIME is a registered trademark and/or trademark of QAvantage in the United States and/or other countries.

All other trademarks are the property of their respective owners.

The example organizations, companies, persons, server and object ids, and email addresses depicted herein are fictitious. No association with any real or actual entity in intended or should be inferred.
Table of Contents

ABOUT THIS GUIDE ......................................................................................................................... 5
QUICK LINKS ................................................................................................................................. 5
RTIME FEATURES ......................................................................................................................... 5
RTIME BENEFITS ......................................................................................................................... 6
GUIDE CONVENTIONS.................................................................................................................... 6

   User Interaction .......................................................................................................................... 6
   Notes ........................................................................................................................................ 6
HOW THIS GUIDE IS ORGANIZED ............................................................................................... 7

DON’T PASS THROUGH YOUR SDLC WITHOUT CONFIRMING YOU’VE MET THE COMPANY-STANDARD
EXIT AND ENTRANCE CRITERIA OF EACH PHASE .............................................................. 7

COMPONENTS OF THE SYSTEM DEVELOPMENT LIFECYCLE .................................................... 8
COMMON SDLC METHODOLOGIES .............................................................................................. 9
COMMON SDLC PROJECT INITIATIVE TYPES .............................................................................. 9
COMMON SDLC PROJECT PHASE DEFINITIONS (WATERFALL) ................................................... 10
COMMON SDLC PHASE DELIVERABLES ....................................................................................... 13
COMMON SDLC DELIVERABLE TASKS ....................................................................................... 13
STANDARD METHODOLOGY COMpanY BEST PRACTICES RECAP ........................................ 16

   Methodology .............................................................................................................................. 16
   Phases ...................................................................................................................................... 16
   Deliverables ............................................................................................................................... 17
PROJECT HIERARCHY, DEFINITION AND SCOPECING ............................................................ 18
CREATE A PROJECT ...................................................................................................................... 18
DEFINE, REFINE AND APPROVE PROJECT REQUIREMENTS ................................................ 19
COMMUNICATION PLAN ............................................................................................................ 20
OVERVIEW .................................................................................................................................. 20
PROJECT ESTIMATING AND SCHEDULING .................................................................................. 22
DETERMINING REQUIREMENT COMPLEXITY ........................................................................... 22
SETTING DELIVERABLE LEVEL OF EFFORT ............................................................................ 23
DERIVING PHASE EXPERIENCE-ADJUSTED HOURS ................................................................. 23
ESTIMATION TYPES .................................................................................................................... 23
PROJECT WORK PLAN MANAGEMENT ...................................................................................... 23

   Managing your RTIME™ Generated MS Project Plan ............................................................. 24
TEAM STRUCTURE, ROLES AND RESPONSIBILITIES ............................................................... 25
PROJECT LEADERSHIP ROLE AND RESPONSIBILITIES .......................................................... 25
ADMINISTRATOR ROLE AND RESPONSIBILITIES ................................................................. 26
BUSINESS ANALYSTS .................................................................................................................. 26
TEST EXECUTORS ........................................................................................................................ 26
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>QAV CRITICAL SUCCESS FACTORS</td>
<td>27</td>
</tr>
<tr>
<td>Risk Mitigation Strategies</td>
<td>27</td>
</tr>
<tr>
<td>Improved Planning</td>
<td>27</td>
</tr>
<tr>
<td>Increased Customer Role</td>
<td>27</td>
</tr>
<tr>
<td>Tightened Project Control</td>
<td>27</td>
</tr>
<tr>
<td>Maintaining a High Performance Team</td>
<td>27</td>
</tr>
<tr>
<td>Assumptions</td>
<td>28</td>
</tr>
<tr>
<td>Human Resources</td>
<td>28</td>
</tr>
<tr>
<td>Processes</td>
<td>28</td>
</tr>
<tr>
<td>Communications</td>
<td>28</td>
</tr>
<tr>
<td>Environment, Data and Change Control</td>
<td>28</td>
</tr>
<tr>
<td>Management</td>
<td>29</td>
</tr>
<tr>
<td>Constraints</td>
<td>29</td>
</tr>
<tr>
<td>Budgetary</td>
<td>29</td>
</tr>
<tr>
<td>Schedule</td>
<td>29</td>
</tr>
<tr>
<td>Resources</td>
<td>29</td>
</tr>
<tr>
<td>Software, Environment and Hardware</td>
<td>29</td>
</tr>
<tr>
<td>Issue Management</td>
<td>30</td>
</tr>
<tr>
<td>Requirement-related Issue Creation</td>
<td>30</td>
</tr>
<tr>
<td>Dependency Management</td>
<td>30</td>
</tr>
<tr>
<td>Internal Dependencies</td>
<td>30</td>
</tr>
<tr>
<td>External Dependencies</td>
<td>30</td>
</tr>
<tr>
<td>Quality Gates and Primary Deliverables</td>
<td>31</td>
</tr>
<tr>
<td>Requirements Definition and Business Approval</td>
<td>31</td>
</tr>
<tr>
<td>Assembling a High-Performance Quality Assurance Organization</td>
<td>34</td>
</tr>
</tbody>
</table>
About This Guide

This chapter provides a general overview of the features and benefits of RTIME, how information is presented in this guide, and how this guide is structured.

This guide is intended to provide end-users with detailed recommended Best Practices for successful SDLC management whether using QA Vantage RTIME™ collaboration platform or other mechanism. Please also review the complete RTIME™ user guide for detailed instructional guidance on utilizing this Best in Class technique.

Quick Links

You can quickly link to other sections in each chapter by using the Quick Links box on the first text page in the chapter.

RTIME Features

RTIME offers many features that simplify project management, improve communications, minimize defects, and ensure that the outcome of your projects meet the intended business requirements. Features include support for:

- Release Content Management
- Project and Resource Management
- Requirements Management
- Software Development Management
- Test Case Management
- Component Management
- Communication & Collaboration
- Out-of-the-Box Integration
**RTIME Benefits**

RTIME™ is business requirements-driven and is designed for use by all business, development, QA, Training and management team members. Whatever development methodology you use, managing resources, meeting aggressive timelines, ensuring development of precise business requirements, and handling test case management has never been this straightforward. RTIME™ allows you to:

- Improve product development accuracy.
- Develop non-ambiguous requirements with pinpoint accuracy.
- Eliminate mistakes due to poor or missed communications.
- Simplify project management.
- Improve software quality.
- Reduce development time and costs.
- Unite all project participants with a common information sharing platform.
- Provide project visibility for all stakeholders.
- Provide superior capabilities at a fraction of the cost of other commercially available tools.
- Deliver proven results with an easy-to-use user interface.

**Guide Conventions**

**User Interaction**

Text that is presented in italics and bold indicates user interaction. This includes buttons, operations and text entry fields:

Click **OK** to accept.

**Notes**

Notes are presented in the following format:

Note: This is an RTIME™ Best Practices note.
## How This Guide is Organized

This *RTIME Best Practices* guide is divided into the following chapters:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chapter 2:</strong> Industry Standard SDLC Terminology</td>
<td>Provides general information about Industry Standard SDLC terminology, Testing Phases, Common Deliverables supporting SDLC phases, phase ownership and accountability.</td>
</tr>
<tr>
<td><strong>Chapter 3:</strong> Project Hierarchy, Definition and Scoping</td>
<td>One clear aid in estimating is grouping and identifying similar projects: past, current &amp; future</td>
</tr>
<tr>
<td><strong>Chapter 4:</strong> Communication Plan</td>
<td>No Tool or Technique can succeed without a focus on Communication and Team Collaboration. The communication plan defines roles, responsibilities, channels, and forums.</td>
</tr>
<tr>
<td><strong>Chapter 5:</strong> Project Estimating and Scheduling</td>
<td>Although CMMI is not described in the context of this primer, repeatability is a key factor in becoming CMMI compliant. RTIME gets you there.</td>
</tr>
<tr>
<td><strong>Chapter 6:</strong> Team Structure, Roles and Responsibilities</td>
<td>Whether using traditional waterfall or Agile with a hint of SCRUM, defining the team members roles and responsibilities leads to accountability</td>
</tr>
<tr>
<td><strong>Chapter 7:</strong> QAV Critical Success Factors</td>
<td>Describes some of the ‘softer’ critical items that require discipline and commitment to enable projects to succeed.</td>
</tr>
<tr>
<td><strong>Chapter 8:</strong> Quality Gates and Primary Deliverables</td>
<td>Don’t pass through your SDLC without confirming you’ve met the company-standard exit and entrance criteria of each phase.</td>
</tr>
<tr>
<td><strong>Chapter 9:</strong> Assembling a High-Performance Quality Assurance Organization</td>
<td>It’s a fact: Your company’s most key asset is your human resources. Make certain you select those with the credentials that match up to building a High-Performance Quality Assurance Organization.</td>
</tr>
</tbody>
</table>
Components of the System Development Lifecycle

Quality Assurance rigor through the entire System Development Lifecycle (SDLC) and across the enterprise varies by project depending on the type of initiative, methodology and phases being executed. For instance, a project focused on improving back-end database performance might not require extensive Usability Testing. A project focused on altering the footer information on legal contracts might not require extensive performance testing. This chapter serves to level-set RTIME-related terminology for quicker assimilation for all project participants. Take notice that the majority of this Best Practices document, while written in the context of supporting the RTIME™ collaboration platform, can also be implemented independent of any technology tool.

This chapter also provides the basis and structure for how your project requirements are developed and implemented. The structure is represented in terms of current industry-standard methodologies, life-cycle phases and deliverables produced from each phase. RTIME™ can easily be configured to support additional application life-cycle methodologies.

Project Management Hierarchy

Portfolios
Projects
Phases
Test and Validation Infrastructure
Requirements, Approvals, Signoffs
Task Management
Sources and Traceability
Company-wide Assets
Organizational Communications
Reporting

Note: RTIME™ provides current methodologies and associated SDLC phases as part of your installation. Additional methodologies, phases and deliverables can be defined by your RTIME™ administrator at any time.
Common SDLC Methodologies

Although there are countless variations of SDLC methods and techniques, RTIME™ is pre-configured with several of the more commonly used. For each Methodology, a standard set of life-cycle phases is defined. Some phases are incorporated into multiple Methodologies. Other phases are specific to a given methodology. RTIME™ allows each client to customize their methodologies and associated phases.

Common SDLC Project Initiative types

RTIME™ accommodates any variation of Initiative Types, as entered by the RTIME Administrator. Upon installation, 10+ initiative types representing common business practices are loaded into the Admin-defined Initiative Types Table.

RTIME™ Best Practices:
Consistent use of each initiative type for like2like projects helps to fine tune the project estimation process on future projects.

1. Change Management Request (CMR)

Represent a request from the business to IT for IT products or services. The size of a CMR is greater than 120 hours of effort performed by IT. QAV evaluates the CMR for Business Requirements and to assess and estimate the complexity of the testing effort.

2. Compliance/Regulatory (C/R)

These requests come in from legal, regulatory or compliance departments and are typically mandated by Federal, State or local governments.

3. Customer Change Request (CCR)

CCRs are requests from the business to IT for IT products or services. The size of a CCR ranges from 25 hours to 120 hours of effort performed by IT. QAV evaluates the CCR for Business Requirements and to assess and estimate the complexity of the testing effort. Any project requiring more than 120 hours is considered a New Development and must be submitted with the appropriate CMR form.

4. Customer Patch Hotfix (CPH)

These requests come in directly from external customers who are typically looking for a fix to a previously deployed product or software release.
5. **Customer Project Request (CPR)**

   These requests come in directly from customers who are typically looking for a *new or enhanced* product or service.

6. **Incident Repair Request (IRR)**

   These requests come in directly from internal departments who are looking for a *fix* to a previously deployed product or software release.

7. **Major Release (MaR)**

   Major Releases are scheduled throughout the year at planned intervals. Projects scheduled within a Major Release have a longer development cycle and larger impact across a company’s enterprise.

   **RTIME™ Best Practices:**
   1. Create a new portfolio for each scheduled Major Release. (Admin zone)
   2. Define Releases that correspond with the Release Date. (Admin zone)
   3. Establish Projects of Initiative Type = Major Release, Release = Release Date (Managers zone)

8. **Minor Release (MiR)**

   MiRs are scheduled, as necessary, to support the Lines of Business’ products and services. MiR projects are reactive, in nature, to the market’s demands, and therefore time is of the essence. These projects are known for generating quick-hit revenues. QAV evaluates the MiR for Business Requirements and to assess and estimate the complexity of the testing effort.

9. **Work Order Request (WOR)**

   WORs handle internal IT requirements for resources in order to upgrade infrastructure used to support the Lines of Business. This Initiative Type is broken out as a means to assess general enhancements to the technology infrastructure supporting various projects and Lines of Businesses.

   Depending on the methodology chosen for each project, associated phases and deliverables require participation and specific documentation from the requesting organization. Failure to provide the appropriate information could result in a delay in migration to production.

   **RTIME™ Best Practices:**
   1. Choose the methodology with Phases and Deliverables that most closely match the objective of the project. (Managers zone, Phases tab)
   2. Use the Web-based dashboard to analyze the present distribution of project types

**Common SDLC Project Phase definitions (Waterfall)**

- **System Concept (SC)** – In this project inception phase, Key Business Objectives are defined, project stakeholders are identified, business cases are originated and submitted for approval (if appropriate), and a high-level timeline is drawn to be elaborated and refined in General Design.

- **General Design (GD)** – Business Cases are refined and approved, impacted systems identified, Workflow and Dataflow diagrams are developed and socialized, high-level prototype mock-ups are also distributed. Requirements are drafted and approved as appropriate.
**Detailed Design (DD)** – Design Specifications are written to satisfy the General Design requirements. All detailed Requirements are generated, finalized and approved in this phase.

**Development (Dev)** – Software is constructed per the Detailed Design specifications and approved Requirements. Any refinement to the approved requirements is reflected in RTIME™ as the single plan of record for the project.

**Unit Testing (UT)** - Unit Testing is performed by the Development team throughout the Life-Cycle of product development.

**Unit Integration Testing (UIT)** – Integration Testing is performed by the Development team prior to migrating the software components into the Testing Environment. It is also utilized by the Development team as part of the regression testing performed following a defect correction. This testing phase assembles multiple components validated in the Unit Test phase, and validates a single Business Function or transaction. Once the unit testing is completed, the testing of Logical Units (LUs) will be performed. IT will define the test cases, execute, and document the results. The QAV will review the results as entrance criteria into subsequent testing phases.

**System Integration Testing (SIT)**

*Usability Testing* is prepared and executed by the QA organization. Its primary function is to validate the basic User Interface presentation, flow and completeness. It does not focus on Business Functionality but instead ensures that the front-end is fully functional: No missing links, missing pages, missing frames, inoperable option frames, non-functioning combo boxes, browser incompatibilities etc.

*Component/Functional Testing* consists of testing a set of Logical Units (LUs), that when combined, satisfy a Business Process (BP). QA develops and executes tests targeting the functionality of these BPs, which is expressed in terms of Functional, or Business. Its primary function is to validate that the developed functionality corresponds to the business expectations, as defined in the Requirements. Component testing consists of a sequence of tests that encompass all components of an application. These tests verify that the application provides all business functions defined in the requirements document and that these functions are defect free. The approach utilized during Component testing consists of the following test methods:

- **Positive function tests**, where the function is tested for successful completion with valid data;
- **Negative function tests**, where the function is tested to reject invalid data or violations of functional business rules;
- **Data tests**, which focus on capability and variations of input and output that would be encountered in the customer’s environment (“Black Box”);
Business Scenarios, where robust samples of typical business situations are tested.

- **External Vendor Interface Testing** – The EVI Testing ensures that the new or enhanced application is compatible with all external entities of the project being tested.

- **User Acceptance Testing (UAT)** – Upon completion of SIT, the application is transferred to the business units for User Acceptance Testing (UAT). The business is responsible for developing the test cases and plan for conducting this phase of testing. User Acceptance Testing (UAT) is prepared and executed by the Business Organization, with support from the QAV. Its primary function is to validate that the developed functionality corresponds to the business expectations. UAT ensures that the new or enhanced application processes the business functions correctly in a production-like environment. It is executed in an orderly manner, and relative metrics are recorded and reported. The business functionality topics are contributed by the Business community, however, existing component Test Cases satisfying similar functionality may be re-used. Any additional test cases required could be developed and executed by QAV and then the UAT team. These additional cases could represent gaps in the existing Component Test bed and should be added accordingly.

- **Quality Verification (QV)** – An additional testing phase, Quality Verification, can be executed in a Production-like environment prior to migrating to Production. This phase gives the IT and QA departments added security that the product in its entirety can and will be successfully migrated to production.

- **Regression Testing (RT)** – Software Build & Quality Gate confirmation
  
  **Software Build** – This phase is designed as a key segment to the RTIME Technique. The actual process varies, depending on your software build schedule. Based on the Build Release Notes, search for the Test Cases corresponding to the Defects listed. These cases are set to the 'In Regression' status to reflect the progress made by the IT staff.

  **Quality Gate/Phase** – Once a Scenario/Case/Script has executed successfully and has been automated, it is a candidate to be added to the Phase Promotion Regression Test. Using RTIME™’s Copy feature, the Test Lead and the Business Subject Matter Experts select test cases with robust capability coverage to include in the Regression Test bed. The regression test for a particular Test Phase grows as the testing progresses. A full regression test is executed to validate Quality Gate criteria into each subsequent phase of testing.

- **Post Release Testing (PR)** – Depending on the complexity of the system, certain features are unable to be simulated or fully test in a Test Environment. This Phase focuses on proving not only that the New Functionality works as it did in test, but also that the integration of it into production causes predictable (as expected) results.
Common SDLC Phase Deliverables

RTIME™ is pre-configured with many of the current Methodologies, their relevant Phases and associated deliverables. As this information is all RTIME Administrator defined, any of the pre-defined settings can be revised, enhanced or replaced altogether by each client’s standards. This section suggests the deliverables needed to implement a typical RTIME™ Project Engagement.

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>Factor</th>
<th>Order</th>
<th>Successor</th>
<th>Resource Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solicit User Stories(XP)</td>
<td>1.00</td>
<td>10</td>
<td>1</td>
<td>Architect</td>
</tr>
<tr>
<td>Assess Business Status</td>
<td>1.00</td>
<td>11</td>
<td>1</td>
<td>Senior Analyst</td>
</tr>
<tr>
<td>Describe Current Business Process</td>
<td>1.00</td>
<td>12</td>
<td>1</td>
<td>Architect</td>
</tr>
<tr>
<td>Define Desired Business Process</td>
<td>1.00</td>
<td>13</td>
<td>1</td>
<td>Developer</td>
</tr>
<tr>
<td>Define/Refine Roles and Responsibilities</td>
<td>1.50</td>
<td>14</td>
<td>1</td>
<td>Senior Tester</td>
</tr>
<tr>
<td>Define and Refine Business Rules</td>
<td>1.33</td>
<td>15</td>
<td>1</td>
<td>Junior Tester</td>
</tr>
<tr>
<td>Choose System Metaphor(XP)</td>
<td>1.00</td>
<td>15</td>
<td>1</td>
<td>Senior Analyst</td>
</tr>
<tr>
<td>Determine Impacted Stakeholders</td>
<td>1.00</td>
<td>20</td>
<td>1</td>
<td>Senior Analyst</td>
</tr>
<tr>
<td>Define Stakeholder needs in terms of Requirements</td>
<td>1.00</td>
<td>21</td>
<td>1</td>
<td>Architect</td>
</tr>
<tr>
<td>Develop Requirements/Obtain LOB Approx</td>
<td>1.00</td>
<td>21</td>
<td>1</td>
<td>Senior Analyst</td>
</tr>
<tr>
<td>Manage the Project Scope and Requirements</td>
<td>1.00</td>
<td>22</td>
<td>1</td>
<td>Developer</td>
</tr>
<tr>
<td>Refine System Requirements</td>
<td>1.50</td>
<td>23</td>
<td>1</td>
<td>Senior Tester</td>
</tr>
<tr>
<td>Request and Obtain Requirement Approvals</td>
<td>1.33</td>
<td>24</td>
<td>1</td>
<td>Junior Tester</td>
</tr>
<tr>
<td>Conduct Architecture Analysis</td>
<td>1.00</td>
<td>30</td>
<td>1</td>
<td>Junior Tester</td>
</tr>
</tbody>
</table>

Common SDLC Deliverable tasks

The deliverables listed below are a sample set of tasks designed to support a medium size project and must be modified for a specific project’s needs. It does not take into consideration any order of magnitude (+ or -) needed to support concurrent projects.

### Design Specifications

- Confirm known Managers, Stakeholders and Analysts for the project are provisioned in RTIME
- Provide Quality Gate Exit Criteria
- Identify Project sponsorship
- Develop Key Business Objectives
- Determine additional Stakeholders
- Construct high-level conceptual design and obtain appropriate feedback and signoff
## Components of the System Development Lifecycle

<table>
<thead>
<tr>
<th>Project Life-Cycle Activities</th>
<th>Owner</th>
<th>Other Participants</th>
<th>Tasks</th>
</tr>
</thead>
</table>
| General                        | Business | QAV and IT         | • Evaluate Entrance Criteria  
• Determine impacted systems and additional Stakeholders  
• Construct General Design, obtain feedback and signoff. |
| Detailed                       | IT | Business and QAV | • Determine any additional impacted systems and identify modules requiring change  
• Work with projects and business community to define Logical Units and Business process matrices  
• Construct Detailed Design, obtain feedback and approval  
• Evaluate Entrance criteria for next phase |

### Development and Unit Testing

| Quality Gate Entrance Evaluation | IT | QAV and Business | ✔ Review and confirm all requirements are reviewed and approved by Stakeholders  
✔ Confirm no Medium or High issues exist against requirements  
✔ Confirm all additional project participants are active in RTIME with the appropriate security role |
|----------------------------------|----|------------------|-------|
| Test Planning                    | IT | Business and QAV | • Identify and provide IT organization resources security level ‘3’ providing unrestricted access to RTIME Developers zone  
• Build test schedules for each project  
• Support the Business team in defining business Requirements Documentation  
• Support IT in analysis where applicable.  
• IT responsible for producing the technical specifications prior to coding.  
• Testing requirements assessed and scoped.  
• Estimation grids for project planning communicated per communication plan. |
| Unit Test Execution              | IT | QAV              | • Provide IT with Standard Unit Test Templates  
• Review Unit Test Plans  
• Provide IT with Quality Gate Criteria for entering into Beta Test  
• Conduct Walkthroughs with IT and Business Unit  
• Evaluate Q-Gate criteria for go/no go to Integration |

### Integration Testing

| Quality Gate Entrance Evaluation | QAV | IT and Business | ✔ Review existing Modules against LU Matrix  
✔ Prepare Entrance into Integration  
✔ Review Unit Test Cases for Completeness  
✔ Validate Software Transfer to Integration Libraries  
✔ Validate all Data is propagated to correct databases by reviewing query results against all loaded data entities |

---

Page | 14
Copyright © 2002-2012 QA\textregistered advantagemark
## COMPONENTS OF THE SYSTEM DEVELOPMENT LIFECYCLE

<table>
<thead>
<tr>
<th>Project Life-Cycle Activities</th>
<th>Owner</th>
<th>Other Participants</th>
<th>Tasks</th>
</tr>
</thead>
</table>
| **Define Logical Units**      | IT    | QAV                | • Develop Technical Test Requirements  
• Define Logical Units (LU)  
• Develop Integration (LU) Test Scripts  
• Build Integration Test Cases (LU)  
• LU Priority as to expedite BP testing to Beta  
• Execute LU Test Scripts  
• Create Logical Unit Matrix  
• Certify All LUs have executed |
| **Define Business Processes** | QAV   | Business and IT    | • Develop Time estimate from original Requirements  
• Evaluate New/Mod/Del impact to Production  
• Develop Business Test Requirements  
• SME Involvement-Requirement validation  
• Define Business Processes (Test Scenarios)  
  ○ Develop BP Test Scripts  
  ○ Review BP Scripts for Completeness  
• Define In-Scope Test Cases  
• Declare Out of Scope Test Cases  
• Create Business Process Matrix  
• Review Test case  
• Tie software component strings to Business Processes  
• Initiate Communication plan for external organizations impacted |
| **Execute Integration Test**  | IT    | QAV                | • Review release notes  
• Execute Test Cycle 1:n  
• Certify Test Pass/Fail  
• Document Findings  
• Review Findings  
• Update Test Plans, if necessary |
| **Quality Gate Exit Evaluation** | QAV | | • Certify all Beta Test scripts are complete  
• Compile all Integration Test Scripts  
• Submit Integration Test Scripts to QAV |
| **User Acceptance Testing**   |       |                    |       |
| **Quality Gate Entrance Evaluation** | Business | QAV | • Evaluate Beta Test execution Packets  
• Evaluate any Level 1 and 2 defects  
• Certify Release to User Acceptance Test |
| **Execute User Acceptance Test** | Business | QAV | • QAV Participation includes:  
  ○ Supporting Test Script generation  
  ○ Reviewing UAT Test Plan  
  ○ Support Documenting of Test Results  
  ○ Perform Bug Triage  
  ○ Certification of UAT Completion  
• QAV Initiates Communication Plan  
  ○ Identify Business Areas - Web site, DCT, rate lock, MIPS, Gallagher, external services, MI, Appraisal, Calculations, Brain, |
### Standard Methodology Best Practices recap

#### Methodology
- **RTIME™** comes preloaded with many popular techniques
- You may use any of the pre-defined methods or create your own
- Only your **RTIME™** administrator can add, edit, or remove methodologies and will do so based on senior management’s recommendation
- Methodologies are comprised of Project Phases and selection dictates timelines based on those Phases

#### Phases
- **RTIME™** comes preloaded with many popular life-cycle project phases
- You may use any of the pre-defined phases or create your own
- Only your **RTIME™** administrator can add, edit, or remove life-cycle phases and will do so based on mid-level management’s recommendations
- Phases are comprised of Deliverables which are weighted by default
- Phases are also weighted by default, and this weight is used to calculate the project timeline.

---

<table>
<thead>
<tr>
<th>Project Life-Cycle Activities</th>
<th>Owner</th>
<th>Other Participants</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brass, pricing, product restrictions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• BU executes Test Scripts and Enters TeamTrack items, as appropriate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• BU Executes Regression Testing across business channels, as necessary</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Quality Gate Emotional Quality Gate Entrance | IT | QAV | Signoff for promotion to production |
| Post Release Testing | QAV | IT | To provide support and perform Post Release testing. |
| | | | • Business Unit will be consulted on release coverage. |

---

![Phase Table](chart.png)

<table>
<thead>
<tr>
<th>Phase</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elaboration - Prototype</td>
<td>1.15</td>
</tr>
<tr>
<td>Construction-Design &amp; Build Increment I</td>
<td>2.30</td>
</tr>
<tr>
<td>Construction-Test Increment</td>
<td>3.45</td>
</tr>
<tr>
<td>Construction-Design &amp; Build Increment II</td>
<td>2.30</td>
</tr>
<tr>
<td>Construction-Test Increment II</td>
<td>3.45</td>
</tr>
<tr>
<td>Transition-User Acceptance Test</td>
<td>1.45</td>
</tr>
<tr>
<td>Transition-Documentation</td>
<td>1.00</td>
</tr>
<tr>
<td>Transition-Release/Deploy</td>
<td>1.00</td>
</tr>
<tr>
<td>Requirement Definition</td>
<td>1.15</td>
</tr>
<tr>
<td>Requirement (Backlog)</td>
<td>1.00</td>
</tr>
<tr>
<td>Design Specifications</td>
<td>1.15</td>
</tr>
<tr>
<td>Product Development</td>
<td>2.30</td>
</tr>
</tbody>
</table>

---

*Copyright © 2002-2012 QA vant age*
Deliverables

- RTIME™ comes preloaded with many popular deliverables
- You may use any of the pre-defined deliverables or create your own
- Only Your RTIME™ administrator can add, edit, or remove life-cycle deliverables and will do so based on project or program management’s recommendations
- Deliverables are also weighted by a factor, and this weight is used to calculate the project timeline.

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Desired Business Proc</td>
<td>1.00</td>
</tr>
<tr>
<td>Define/Refine Roles and Res</td>
<td>1.50</td>
</tr>
<tr>
<td>Define and Refine Business Req</td>
<td>1.33</td>
</tr>
<tr>
<td>Choose System Metaphor (XP)</td>
<td>1.00</td>
</tr>
<tr>
<td>Determine Impacted Stakeholder</td>
<td>1.00</td>
</tr>
<tr>
<td>Define Stakeholder needs in</td>
<td>1.00</td>
</tr>
<tr>
<td>Develop Requirements/Obtain</td>
<td>1.25</td>
</tr>
<tr>
<td>Manage the Project Scope curr</td>
<td>1.00</td>
</tr>
<tr>
<td>Refine System Requirements</td>
<td>1.50</td>
</tr>
<tr>
<td>Request and Obtain Require</td>
<td>1.33</td>
</tr>
</tbody>
</table>
Project Hierarchy, Definition and Scoping

Consistency and reusability provide huge cost savings when implemented across the enterprise. Most of the controls, rules and rigor built into RTIME™ provide each customer with a means for delivering just that. The previous chapter described how to establish a consistent, company-wide project management infrastructure. This chapter focuses on defining Project-specific information within the domain of the enterprise-wide infrastructure.

<table>
<thead>
<tr>
<th>Step</th>
<th>What</th>
<th>Who</th>
<th>How</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Locate an existing or request a new Portfolio be added by the RTIME Administrator</td>
<td>Group Leader or Project Manager</td>
<td>Evaluate existing Portfolios. If none apply, request a new one from your RTIME™ Administrator</td>
<td>Inception of Project Idea</td>
</tr>
<tr>
<td>2.</td>
<td>Add General Project Info including assignment of project to Project Manager</td>
<td>Group Leader</td>
<td>Right-Click Portfolio and complete the General Tab form. Inform selected PM via Rteam.</td>
<td>Following Portfolio ID</td>
</tr>
<tr>
<td>3.</td>
<td>Assign and Notify Stakeholders</td>
<td>Project Manager</td>
<td>Select the Approvals tab and assign stakeholders from the list of stakeholders. Notify your RTIME™ Administrator if a resource is absent from the list of stakeholders.</td>
<td>Following notification of Project</td>
</tr>
<tr>
<td>4.</td>
<td>Select Methodology and review associated Phases</td>
<td>Project Manager</td>
<td>Select the Project Phases tab. Beneath the list of phases is a dropdown for choosing a methodology to be used with this project.</td>
<td>For quick estimation or full MS Project creation</td>
</tr>
<tr>
<td>5.</td>
<td>Review assigned deliverables</td>
<td>Project Manager</td>
<td>Select the Deliverables tab. Remove any standard deliverables not applicable to the project.</td>
<td>For quick estimation or full MS Project creation</td>
</tr>
<tr>
<td>6.</td>
<td>Catalog Project Sources</td>
<td>Project</td>
<td>Select the Project Sources tab and create entries for each project source.</td>
<td>Ongoing as new</td>
</tr>
<tr>
<td>Step</td>
<td>What</td>
<td>Who</td>
<td>How</td>
<td>When</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Manager</td>
<td>Note that project sources must reside on a network shared drive in order for other team members to have access to them.</td>
<td>documentaion surfaces or supplemental documents are created internally to support the project.</td>
</tr>
<tr>
<td>7.</td>
<td><strong>System Mapping</strong></td>
<td>Project Manager</td>
<td>Click the System Mapping tab and associate any systems with this project. If a system is not defined, notify the RTIME™ Administrator.</td>
<td>Following Proof of Concept, General Design or similar.</td>
</tr>
</tbody>
</table>

### Define, Refine and Approve Project Requirements

<table>
<thead>
<tr>
<th>Step</th>
<th>What</th>
<th>Who</th>
<th>How</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Familiarize yourself with existing project documentation.</td>
<td>Project Manager</td>
<td>Analyze the functions and documentation pertaining to the project. Organize the information into meaningful and business-related subject matter areas.</td>
<td>After a project’s business case or budget is approved.</td>
</tr>
<tr>
<td>2.</td>
<td><strong>Build your Requirement Classes</strong> based on the subject matter areas established in the previous step. Plan such that all requirements pertaining to the Requirement Class can be organized in that one group.</td>
<td>Project Manager</td>
<td>Right-Click the Project and select ‘Define New Requirement Class’. Enter the Requirement Class Title, Type* and description. Contact your RTIME Administrator if the current Requirement Class Types don’t satisfy the group purpose.</td>
<td>As summary information becomes available.</td>
</tr>
<tr>
<td>3.</td>
<td>Develop Requirements from existing documentation. Conduct walkthroughs and obtain detailed information from project stakeholders and subject matter experts.</td>
<td>Business Analyst, Project Manager</td>
<td>Review all existing documents and isolate requirements. Right-Click a Requirement Class corresponding to the functionality within the requirement. Complete the general tab and save. Associate any Assumptions, Constraints, Issues or dependencies.</td>
<td>As detailed information becomes available.</td>
</tr>
<tr>
<td>4.</td>
<td>Obtain Stakeholder approval for Requirements</td>
<td>Project Manager</td>
<td>Managers Zone, Approvals tab. Select any of the names in the left list and assign as stakeholders. Select each stakeholder and click ‘Reqs Ready’ button in the ‘Stakeholder Notification’ frame.</td>
<td>After all requirements are defined and ready to be designed.</td>
</tr>
</tbody>
</table>
Communication Plan

Overview

Most likely, many of your projects impact applications and users across multiple business channels. Because of this, it is extremely important that all resources are notified of potential impacts early on in the development life cycle. To facilitate this communication, your RTIME™ Administrator will request a Single Point of Contact (SPOC) from all LOBs and System owners.

RTIME™ Best Practices:
1. Single Points of Contact (SPOCs) are Project Stakeholders or System Owners defined as RTTeam members with a security level of Stakeholder or higher.
2. SPOCs are auto-notified through RTIME when a Project Manager designates them as a Project Stakeholder or when a System they own is impacted by a project.
3. All associate contact information is available under the RTTeam menu option, where a user can chat or send email to an individual team member or RTTeam group.

The following table lists just a sampling of various means to communicate:

<table>
<thead>
<tr>
<th>Communications Item/Event</th>
<th>Vehicle</th>
<th>From:</th>
<th>To:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Documentation</td>
<td>RTIME™ Chat w/Portal</td>
<td>Analyst</td>
<td>Owner</td>
</tr>
<tr>
<td>Test Artifacts including Scenarios and Traceability Matrices</td>
<td>RTIME Reporting Hub or Microsoft Excel</td>
<td>Test Team</td>
<td>Development and business partners</td>
</tr>
<tr>
<td>Status Updates</td>
<td>e-Mail or RTIME™ Chat from Associate Dropdown list</td>
<td>Manager</td>
<td>Analyst</td>
</tr>
<tr>
<td>Reporting of Bugs</td>
<td>e-Mail from Issue form</td>
<td>Tester</td>
<td>Developer</td>
</tr>
<tr>
<td>General Inquiries</td>
<td>e-Mail or RTIME™ Chat</td>
<td>Any Member</td>
<td>Any Member</td>
</tr>
</tbody>
</table>
SUGGESTED RECURRING MEETINGS

<table>
<thead>
<tr>
<th>Communications Item</th>
<th>Participant</th>
<th>Frequency</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Workshops</td>
<td>RTIME™ Chat w/Portal</td>
<td>Analyst</td>
<td>Owner</td>
</tr>
<tr>
<td>Team Status Meeting</td>
<td>Testing Manager</td>
<td>Weekly</td>
<td>Analyst</td>
</tr>
<tr>
<td>Project Status Meeting</td>
<td>Testing Manager</td>
<td>Weekly</td>
<td>Test Lead</td>
</tr>
<tr>
<td>Business Unit/QA Walkthru</td>
<td>BU, Test Lead, Dev</td>
<td>As required</td>
<td>Various</td>
</tr>
</tbody>
</table>

COMMUNICATION PLAN EXECUTION

Early detection and resolution of conflicts and dependencies will ensure a smooth transition through RTIME™ Quality Gates. There are several stages to executing the communication plan through a SDLC engagement. These stages or communication points are a critical component in achieving successful people collaboration. It's important to note that in some SDLC models, these events occur concurrently. RTIME allows the Project Manager to not only build the phase dependencies for each methodology but to also select the methodology for each project.

<table>
<thead>
<tr>
<th>Communications Item</th>
<th>Participants</th>
<th>Timing</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Workshops</td>
<td>PM, Stakeholders, SMEs, Business Analysts, IT Manager</td>
<td>Prior to Detailed Design</td>
<td>QAV or PM Manager</td>
</tr>
<tr>
<td>Detailed Design Walkthrough and Approval</td>
<td>PM, Key Stakeholder, Designer, IT Manager</td>
<td>Prior to development</td>
<td>Designer</td>
</tr>
<tr>
<td>System Test plan Walkthrough and Approval</td>
<td>QAV/PM Manager, IT,</td>
<td>1 week Prior to System Test</td>
<td>QAV or Test Lead</td>
</tr>
<tr>
<td>User Acceptance Test</td>
<td>Lines of Business, QAV</td>
<td>1-2 weeks prior to UAT</td>
<td>QAV</td>
</tr>
<tr>
<td>Production Release</td>
<td>IT, QAV</td>
<td>Start of Code Freeze</td>
<td>IT</td>
</tr>
</tbody>
</table>

COMMUNICATION TEAM CONTACT BY ORGANIZATION

Each Business Unit has contacts required to be included on various correspondences. All team members should be defined in RTIME™, whether power users or casual report runners. The following information is required to register a user to send and receive RTIME communications:

<table>
<thead>
<tr>
<th>Organization</th>
<th>Network ID</th>
<th>First Name / Last Name</th>
<th>Email Address</th>
<th>Security Level</th>
<th>Contact Number</th>
</tr>
</thead>
</table>
Project Estimating and Scheduling

RTIME™ estimates are calculated from ‘the bottom up’, using a Complexity assigned at the individual Requirement level, a Level of Effort established at the Deliverable level and experience-adjusted hours set at the Phase level (Aggregate of Complexity * Deliverable * Phase). That amount is then rolled up into an estimate for each individual project Phase. Finally, each project is rolled up to the Program-level for management reporting.

Determining Requirement Complexity

A Requirement Complexity can be set to Low, Medium or High. Each complexity level is assigned a weight in the Admin zone and this weight is applied across the enterprise. Adhering to these Best Practices across the enterprise ensures accurate and consistent estimations for projects managed within, or external to, RTIME™. Consider how many of these traits apply to the requirement:

- Impacts a single web page, form, report or other presentation
- Impacts multiple pages, forms etc. of the same presentation type
- Impacts multiple presentation types (web pages and reports)
- Impacts multiple applications with a system
- Impacts multiple systems across an organization
- Impacts multiple organizations across the enterprise
- Represents completely new functionality
- Represents enhancements to existing functionality
- Represents minor changes to existing functionality
- Requires a single or independent variable to implement feature
- Requires several variables to implement feature
- Requires many variables to implement feature
As these are just some of the potential influences towards determining complexity, most important is communication and agreement within a department as to what constitutes Low, Medium or High complexity. The key here is consistency applying rules to gauge complexity.

**RTIME™ Best Practices:**

1. A major design benefit of RTIME Requirement Management is in the ‘limitation’ of 3 levels of detail. This helps enforce consistency of project planning and estimating across multiple organizations. Reporting, searching and traceability are also improved by this hierarchy. Use the Requirement Class, Requirement Class type or qualify the Requirement if additional ordering is required.

**Setting Deliverable Level of Effort**

RTIME™ deliverables seed data provides for a baseline Level of the more popular SDLC methodologies. These values will need adjusting as experience dictates depending on current skills and lessons learned from previous projects.

**Deriving Phase Experience-Adjusted Hours**

RTIME™ seed data provides for a baseline number of hours for the phases most common to the more popular SDLC methodologies. These values will need adjusting as experience dictates depending on current skills and lessons learned from previous projects.

**Estimation Types**

The following Estimation Types have been adopted for QAV’s Best Practices for use in estimating effort and duration (which translates into labor costs). The accuracy level is tied directly to the accuracy and detail of requirements defined to RTIME™.

**Definitive (± 25%)** – An approximation of Time and Cost calculation based on adjusted and tuned factor settings as described in the previous section. Definitive estimates result from established list of Phases, Deliverables and the ‘Approved’ Requirements implemented by Phase and Deliverable. This Estimation Type assumes project requirements have passed through Quality Gate 1 (QG1) – Requirements Evaluation.

**Preliminary** – An Estimate developed by a project lead and or subject matter experts based on a general understanding of a JDI-type (± 20%), CMR-type (± 40%) or LSP-type (± 60%). This type of estimate will be used for preliminary commitments. Requirements are entered into RTIME™ using the ‘Speed Add’ feature which allows requirements to be added upon entry of only 2 fields: Title and Complexity.

**RTIME™ Best Practices:**

1. Each manager should adhere to similar rules for setting Requirement Complexity
2. Schedule periodic consistency reviews or assign the auditing to the RTIME Administrator.

**Project Work Plan management**

RTIME™ directly interfaces with MS Project to provide mechanized project work plans for each individual project. The Project Plan is managed by the Project Lead and administered by the Project Lead or Project Administrator. This serves as the primary management tool for scope, schedule, resource and cost. Progress and status is tracked and reported against the
baselines. Future iterations of the project plan as generated are adjusted, based on variations to the baseline. These subsequent iterations become the baseline for managing future initiatives. Public access is available per the Communication Plan. The activities associated with managing and administering the project plan includes:

- Generating the Project Plan from within RTIME™
- Fine-tuning the RTIME™-related deliverables
- Integrating non-RTIME™-related deliverables
- Combining individual MS Project Plans into a Project Master
- Requesting auto-updates via the MS Project™<RTIME™ gateway
- Requesting and Updating manual status from deliverable owners
- Reporting Status through RTIME™ and MS Project™

### Managing your RTIME™ Generated MS Project Plan

<table>
<thead>
<tr>
<th>What</th>
<th>Who</th>
<th>When</th>
<th>Where</th>
<th>How</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate the Project Plan from within RTIME™</td>
<td>Project Manager</td>
<td>After Phases and Requirements have been defined</td>
<td>RTIME™ Managers Zone</td>
<td>Right-Click Project</td>
</tr>
<tr>
<td>Fine-Tune the RTIME™-related deliverables</td>
<td>Project Manager</td>
<td>After generating the Project Plan from within RTIME™</td>
<td>MS Project</td>
<td>Manually adjust plan</td>
</tr>
<tr>
<td>Integrate non-RTIME™-related deliverables</td>
<td>Project Manager</td>
<td>After generating Project plan from within RTIME™</td>
<td>MS Project</td>
<td>Manually add tasks and deliverables to plan</td>
</tr>
<tr>
<td>Combining individual MS Project plans into a Master</td>
<td>Program Manager</td>
<td>As part of building release content and ongoing updates</td>
<td>MS Project</td>
<td>Using the RTIME™ master plan, insert individual project plans</td>
</tr>
<tr>
<td>Request Updates via the MS Project™&lt;RTIME™ gateway</td>
<td>Project Manager</td>
<td>Whenever a revised cost, effort or status update is needed</td>
<td>MS Project</td>
<td>Click the Update Via RTIME™ Gateway</td>
</tr>
<tr>
<td>Request and apply updates of non-RTIME™ deliverables</td>
<td>Project Manager</td>
<td>Ad-hoc or standard weekly schedule, depending on needs</td>
<td>MS Project</td>
<td>Manually adjust plan</td>
</tr>
<tr>
<td>Status Reporting through RTIME™</td>
<td>Members with RTIME™ access</td>
<td>As Needed or at the request of PM to review and approve Requirements, designs or tests</td>
<td>RTIME™</td>
<td>Two Robust Reporting methods: 1) RTIME™ Application Menu bar or 2) RTIME™ Web-based Reporting.</td>
</tr>
<tr>
<td>Status Reporting through MS Project</td>
<td>Members with MS Project</td>
<td>When requiring Plan vs. Actual budget status, resource leveling, Gantt Time Lines, or other MS Project-type reporting is needed</td>
<td>MS Project</td>
<td>Access any of the RTIME™ MS Project Gateway Customized Views or filters</td>
</tr>
</tbody>
</table>
Team Structure, Roles and Responsibilities

The QAV team is constructed with all of the staff needed to Manage and Execute the activities stated in this Charter document. This team supports 3 primary roles: 1) Leadership, 2) Administration 3) Test Planning and Executing. Ideally, QAV also employs a project coordinator, general banking Subject Matter Experts, one technical engineer who will be responsible for the environment and SQL queries, technical assistance and common tools used by the testing team and a standards and support analyst.

Project Leadership Role and Responsibilities

The leadership of QAV consists of 1 Project Manager, 1 Team Lead and depends on the invaluable support of IT and Business Management. Areas of responsibility for the Leadership Team are:

- Development of Project Charter
- Procurement or Development of Requirements and Testing tools
- Procurement or Development of status and metrics reporting tools
- Monitoring and Reporting of testing outcomes
- Development and Enforcement of standards and procedures
- Generate single status report to the steering committee and management
- Training of the project team in methodology and best practices
- Maintain all project plans and report schedule variances
- The coordination and development of the projects testing plan
- Identify the staffing needs
- Certify the software has past the Quality Gate acceptance criteria
- Establish specific promotion criteria
- Lock down software upon Certification
- Run the day to day testing tasks in the beta environment
- Providing junior-level subject matter expertise to perform as apprentices for Senior Business resources
- Developing service level agreements with our serviced entities
Administrator Role and Responsibilities

- The project administrator is responsible for collecting team status via the MS Project Plan.
- In addition to project plan updates, the administrator maintains the Issues and Dependencies list.

These activities require an extremely pro-active resource, to pursue and persist on getting actual work effort captured into the project plan and close monitoring of the Issues and associated dependencies.

Business Analysts

- Define Requirements reflective of the initiative to be tested
- Develop Test Scenarios (Business-like usage of area being tested)
- Develop Test Cases including supporting Test Data and Expected Results.
- Obtain buy-in from the appropriate business resource, as defined in the Communications Plan
- Support Test Executors
- Create Logical Units and Business Process linkage to Approved Business Requirements Matrixes
- Consider and Factor External Vendors and all associated Interfaces
- Coordinate with External Service vendors for creating regression test bed.
- Prepare and Deliver Execution Signoff package to SMEs.

Test Executors

- Support Test Analysts with Test Case Inventory
- Review and Resolve any Script discrepancies with the Test Planners
- Execute test scripts, validate results, Create Ticket item, if necessary
- Update Test Case tracking tool with status, Actual Results and any failure reasons. The Test Executor must take care in setting the appropriate test case status
- Manage individual Regression Test pool by tracking Software Builds
QAV Critical Success Factors

The QAV team is constructed with all of the staff needed to Manage, Execute or Support the activities stated in this document. This team supports 3 primary roles: 1) Leadership, 2) Administration 3) Test Planning and Executing. Ideally, QAV also employs a project coordinator, general banking Subject Matter Experts, one technical engineer who will be responsible for the environment and SQL queries, technical assistance and common tools used by the testing team and a standards and support analyst.

Risk Mitigation Strategies

Risk Management focuses on four key areas: 1) Improved Planning, 2) Increased Customer Role, 3) Tightened Project Control, 4) Building a High Performance Team

Improved Planning

- Detail in Work Breakdown Structures and associated Activity Lists
- Apply Configuration Management to products within project
- Obtain Timely approvals and sign-offs
- Conduct full post-project evaluation reflected in process improvement
- Monitor changes in business environment

Increased Customer Role

- Increase Business customer involvement
- Increase communication with sponsor/customer
- Create a customer steering group
- Have customer or designated committee rule on all change requests
- Carefully define objectives and requirements

Tightened Project Control

- Establish consistent tracking methods
- Produce regular progress reports
- Introduce and enforce formal change management/control
- Report on key milestone meetings
- Establish an issues reporting and resolution process

Maintaining a High Performance Team

- Select an experienced project leader
- Obtain team participation in planning
QAV CRITICAL SUCCESS FACTORS

➢ Perform regular technical reviews w/experts
➢ Increase communication within the Project and across the Program

Assumptions

These assumptions provide some of the common considerations for most projects. They are tracked and maintained in the RTIME Collaboration platform at the Project or Requirement level. Those pertaining to a given project should be reviewed throughout the SDLC, as they are the basis for critical business and financial decisions. Note: this are just a few examples of project assumptions

Human Resources

➢ All resources identified in the statement of work resource planning document will be provided no later than [enter date]. Until the QAV team is fully staffed, its potential is reduced.
➢ Throughout the existence of QAV, all resources displayed on the QAV Organization Chart are dedicated to the Testing group as Full Time Equivalents, 100%.
➢ Although budget for QA resources will be allocated to the project from various budgets, management of those resources will be the responsibility of the QAV Leadership Team.
➢ There will be no delays resulting from unavailability of the business resources when their support is required
➢ Timely availability of systems support people
➢ Timely availability of database support

Processes

➢ All System development projects and maintenance will adhere to the corporate systems development life cycle (SDLC).
➢ QAV becomes engaged with an initiative simultaneous to IT’s design phase
➢ Testing Group will use the Code Test Strategy defined in this Charter. IT agrees to deliver Logical Units as required by the QAV in support of this strategy.
➢ QAV is engaged during the requirement phases of all projects in which time Test Case scope and requirements mapping will occur
➢ A System Development Life-Cycle (SDLC) management platform is in place and used for all projects requesting QAV

Communications

➢ Once migrated to the Beta Test Environment, the current system code will be frozen to reduce risk during testing. If the freeze cannot be maintained, complete communication on the changes that are made, will be available to the testing team.
➢ There are stable and thorough requirements and design specifications available to the QAV

Environment, Data and Change Control

➢ 2 isolated test environments are made available no later than [enter date]
➢ Change and configuration management processes and controls are used consistently by IT to minimize incorrect software version migration.
➢ Timely availability to realistic test data
There is stable software in the test environment and dedicated resources for quick turnaround of software fixes

There will be continued availability of all current systems that supply data to and receive data from the applications tested by QAV.

The Beta version migrated to the QAV environment by IT meets the exit criteria for the developer-conducted Integration testing

**Management**

- QAV is granted veto rights to reject any migration to its Testing Environment should IT or Business fail to provide the specific Quality Gate documents, as agreed to in this charter.
- "Team-based" approach is used for the project, i.e. there are no walls between IT, Business Analysts, Business User Community, Net Ops, DB Ops and other parties affected by the QAV.
- All required organizations will fully participate in the projects and provide a project sponsor and the needed Subject Matter Experts (SME)

**Constraints**

These Constraints provide some of the common considerations for most projects.

**Budgetary**

- The budgeting of QAV is typically provisioned by the IT budget. Any decision by the QAV leader regarding resources, Software or Equipment must be approved by the IT manager.

**Schedule**

- Socializing the QAV global strategy for rollout is scheduled for [enter Date] through [enter date] The preparation effort culminates in the release of this QAV Charter document.
- Individual QAV Testing Project Schedule constraints are reported by project per the detailed protocol of the QAV Communication Plan.

**Resources**

- Team members are required to plan any vacation time, at a minimum, 1 month in advance. To ensure there will be minimal impact to the project, the desired vacation time should be discussed with appropriate management (functional and project).
- Should a member of the team become unable to perform their assigned activities and/or meet their accepted responsibilities, it is imperative that they notify the appropriate management (functional and project) in a timely manner.
- Risk to the successful implementation and execution of QAV lies primarily in the human resource domain. The processes and discipline surrounding a solid Requirements and Testing methodology brings little value without the resources necessary to implement and execute.

**Software, Environment and Hardware**

- **Testing Software** - Your Company’s standard testing protocol is will integrate seamlessly into RTIME™. QAV’s adapts this standard and makes every attempt to remain compliant. If need arises, QAV communicates any deviation.
QAV CRITICAL SUCCESS FACTORS

- **Environment** - To unleash the full potential of QAV, a supported Beta Test environment should be created, ideally mirroring current production.
- **Hardware** - No additional Hardware is expected or required, with the exception of what would be necessary to implement the Software or Environmental needs previously stated.

**Issue Management**

Issues are identified and submitted for action. Submissions are logged and tracked. A unique, sequential identifier is generated for each issue. Each issue has an owner assigned to ensure accountability and pro-active response. The issues log is reviewed in the project status and/or technical review meetings.

There are 2 distinct types of RTIME issues:
1. Requirement-related issues.
2. Test-related issues.

**Requirement-related Issue Creation**

Issues are created against requirements for various reasons. Here are a few:

- A discrepancy exists between various project participants as to the implementation requirements necessary to satisfy the business function.
- Relationship between Issue severity and Requirement Status

Upon creation of an Issue, an automated issue tracking system submits an e-mail to the assigned Issue Owner. The owner has 48 hours to provide a mitigation strategy and redirect the issue or accept full accountability for the resolution.

**Dependency Management**

Cross-project dependencies account for the primary source of risk to each testing endeavor. QAV’s project plan manages these dependencies on two levels: Internal and External

**Internal Dependencies**

An Internal Dependency exists between various resources or teams within a single organization/Channel/Business Unit. These dependencies are easier to manage and react to, since the prioritization of activities lies within a single management organization. Links for Internal Dependencies are tracked within the MS Project plan, and the appropriate parties are notified as described in the Communication Plan. Status is requested as described in the Communication Plan.

**External Dependencies**

An External Dependency exists between one internal organization and another internal organization OR one internal organization and one non-internal (external to your company) corporation. These dependencies inject far more risk into the project and corresponding deliverables, and therefore require constant proactive attention. Status is requested as described in the Communication Plan.
Quality Gates and Primary Deliverables

Presented here are 4 groups of activities, each with 1 or more Quality Gate evaluation checklist. Quality Gates are validation points at each project phase.

1. **Requirements Definition and Business Approval**
2. **Test Case Creation, Execution and Signoff**
3. **User Acceptance Test**
4. **Post-Release Activities**

**Requirements Definition and Business Approval**

- **Quality Gate 1 (QG1) – Requirements Evaluation**
  - Business Channel to prepare and provide final approval
  - Prioritization only occurs after Requirements are obtained
  - Documentation must be in Standard Guide format
  - QAV will not accept any project without fully documented requirements
  - Review for applicability and completeness relative to bug fix
  - Validate IT Sign-off
  - Approve Migration into Beta Environment

- **Requirement Definition Activities**
  - Contact Business and Application Owner if necessary (See Communication Plan for appropriate contact information)
  - Analyze requirements for business unit
  - Document Assumptions, Constraints, Issues and Dependencies
  - Create Requirement Class entry
  - Review Existing Requirements relative to Bug Fix functionality
  - Determine Applications Associated with Change Request
  - Determine dependencies across the Organization requesting the change.
  - Determine any dependencies external to the Organization, but Internal to your company
  - Determine any Dependencies External to your company
Review any applicable Existing Testing Scenarios from the Regression Test Repository
Develop any additional Testing Scenarios and corresponding Test Data
Obtain Business Approval of Requirements

**Technical Specification Activities**
- IT Development to provide this documentation before QAV proceeds to conduct further analysis

**Testing Effort Estimation and Prioritization Activities**
- Analyze Technical Specification
- Determine Complexity
- Estimate Testing Level of Effort
- Prioritize across projects
- Prioritize within project

**Business Walkthru Activities**
- Technical specifications and Testing Analysis to be reviewed by Business Owner
- Determination is made on accuracy of analysis
- Approval of Requirements is granted from the Business Owner to proceed with Code and Test

**Test Case Creation, Execution and Signoff**

- **Quality Gate 2 (QG2) – Test Preparation Evaluation**
  - Obtain Unit Test results
  - Requirements Documentation Approved
  - Technical Specifications Approved
  - Testing Analysis Approved

- **Test Case Scenario Preparation Activities**
  - Review Functional Impact Report (FIR)
  - Complete Test Case Estimation based on FIR
  - Select Appropriate Regression Test Cases based on FIR
  - Develop any new test cases based on functionality
  - Incorporate new test cases into regression test bed, as appropriate
  - Determine Release and Communicate to Development and Business Unit
  - Coordinate the schedule and work allocation for the project
  - Create Test Case(s) based on Approved Requirements; Include Execution Steps, Test Data and Expected Results

- **Quality Gate 3 (QG3) – Test Execution Evaluation**
  - Evaluate Prepared Test Cases and Mapped Requirements
  - Resolve any Assumptions, Constraints, Issues or Dependencies
  - Confirm and Satisfy any Test Case Pre-Requisites
  - Execute Test Case and Validate Results
  - Enter Defect report into TeamTrack, if necessary
  - Assemble Test Results and submit to Business Owner for Signoff
  - Document necessary tracking information in support of UAT Q-Gate
  - Provide Quality Gate UAT form to the Business Owner per Communication Plan
Execute Test Scenarios Activities

- Evaluate Loan Information
- Review Back-end Database
- Execute Test Cases
- Evaluate results
- Regression Test Failed cases
- Update Status
- Close TeamTrack Items

User Acceptance Test

- **Quality Gate 4 (QG4) – UAT Evaluation**
  - Evaluate Beta Test execution Packets
  - Evaluate any Level 1 and 2 defects
  - Certify Release to User Acceptance Test

User Acceptance Test Activities

- Support Business Unit with Development of Test Scenarios
- Business Unit to Create and Execute test cases
- Support Final Signoff

Post-Release Activities

- **Quality Gate 5 (QG5) – Release Evaluation**
  - QAV to provide release management proper signoffs
  - Assist release management in scheduling of release to production
  - Provide Release Contacts for Testing

Post Release Activities

- Review Requirements and Test Case Information for any Post-Release Requirements
- Review and Validate JDI assumptions and conditions
- Provide Updates per the Communication Plan with any pertinent information regarding the Post-Release performance
- Post-Production Evaluation (PPE) is issued for every project reaching the QAV. QAV provides its relevant information as prescribed in the Communication Plan. It documents:
  - A management or executive-level summary of the project (based on Charter)
  - Success of the project as measured against objectives
  - Lessons learned
## Assembling a High-Performance Quality Assurance Organization

A common and proven composite of resources to successfully implement software releases is described within. The infrastructure can expand or contract based on the size and frequency of your releases. This is only a recommendation based on previous success stories from QAVantage, and describes the skills, roles and responsibilities of a traditional QA organization.

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
<th>Skills Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support Team Lead</td>
<td>➢ Plan and manage testing activities</td>
<td>➢ Project management knowledge and skills (2-5 years)</td>
</tr>
<tr>
<td></td>
<td>➢ Assist in developing testing policy and procedures</td>
<td>➢ Results focused</td>
</tr>
<tr>
<td></td>
<td>➢ Aids in developing testing strategy</td>
<td>➢ Problem-solving skills</td>
</tr>
<tr>
<td></td>
<td>➢ Allocate resources across Testing projects</td>
<td>➢ Team oriented</td>
</tr>
<tr>
<td></td>
<td>➢ Define high-level testing requirements, i.e. resources, testing facility, data, hardware, etc.</td>
<td>➢ Coaching / mentoring skills</td>
</tr>
<tr>
<td></td>
<td>➢ Develop testing standards specific to Beta Testing</td>
<td>➢ Strong testing methodology background</td>
</tr>
<tr>
<td></td>
<td>➢ Develop, coordinate, and monitor test execution plans</td>
<td>➢ Knowledgeable of Banking systems</td>
</tr>
<tr>
<td></td>
<td>➢ Lead testing design and analysis phase</td>
<td>➢ Strong Writing skills</td>
</tr>
<tr>
<td></td>
<td>➢ Coordinate and lead defect triage</td>
<td>➢ Knowledge of Q/C</td>
</tr>
<tr>
<td></td>
<td>➢ Work closely with IT and Business Customers to support communication</td>
<td>➢ 3 – 5 years MS Office</td>
</tr>
<tr>
<td>Role</td>
<td>Responsibility</td>
<td>Skills Set</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>process</td>
<td>➢ Provide testing reports to Project Management Team</td>
<td>➢ Web Literate</td>
</tr>
<tr>
<td></td>
<td>➢ Review testing deliverables produced by the Test Team, as needed</td>
<td></td>
</tr>
<tr>
<td>Technical Support Specialist</td>
<td>➢ Report directly to the Support Team Lead</td>
<td>➢ Developer with 5 – 7 Years experience</td>
</tr>
<tr>
<td></td>
<td>➢ Provide technology knowledge and expertise as needed by the Test Team</td>
<td>➢ SQL knowledge</td>
</tr>
<tr>
<td></td>
<td>➢ Assess readiness of the testing environment for testing execution</td>
<td>➢ Web Development Knowledge</td>
</tr>
<tr>
<td></td>
<td>➢ Develop release notes based on information gathered from the Phoenix Management</td>
<td>➢ Good Writing Skills</td>
</tr>
<tr>
<td></td>
<td>➢ Develop technology standards, if needed</td>
<td>➢ Understanding of Testing methodology</td>
</tr>
<tr>
<td></td>
<td>➢ Review testing strategy, test design and analysis deliverables to ensure that it conforms to Project's architectural and development standards</td>
<td>➢ Good Mentoring Skills</td>
</tr>
<tr>
<td></td>
<td>➢ Build SQL Queries as needed to validate testing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>➢ Support Testing</td>
<td></td>
</tr>
<tr>
<td>Senior Test Analyst</td>
<td>➢ Report directly to the Project Test leads</td>
<td>➢ Senior Testing person 3 – 5 years experience</td>
</tr>
<tr>
<td></td>
<td>➢ Develop testing patterns and testing models, e.g. business scenario checklist, decision table, test coverage, test cycles checklist, etc.</td>
<td>➢ Strong Knowledge of Testing Methodology (3-5)</td>
</tr>
<tr>
<td></td>
<td>➢ Implement testing standards, policy and procedures</td>
<td>➢ Web application Knowledge (2)</td>
</tr>
<tr>
<td></td>
<td>➢ Define test data requirements</td>
<td>➢ Good Writing Skills (1)</td>
</tr>
<tr>
<td></td>
<td>➢ Define testing environment requirements</td>
<td>➢ Team player</td>
</tr>
<tr>
<td></td>
<td>➢ Develop test coverage matrix</td>
<td>➢ MS office literate (2-3)</td>
</tr>
<tr>
<td></td>
<td>➢ Develop test cases</td>
<td>➢ Good Mentoring Skills</td>
</tr>
<tr>
<td></td>
<td>➢ Perform test execution, capture test results and report defects</td>
<td>➢ Ability to write test scripts (2)</td>
</tr>
<tr>
<td></td>
<td>➢ Generate detailed test reports, e.g. defect metrics, test coverage metrics, status</td>
<td>➢ Analytical skills to diagnose problems (2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>➢ Banking Knowledge (+)</td>
</tr>
<tr>
<td>Role</td>
<td>Responsibility</td>
<td>Skills Set</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>report, etc.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>➢ Participate in the defect triage, as needed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>➢ Work closely with the various Lines of Business</td>
<td></td>
</tr>
<tr>
<td>Test Executor</td>
<td>➢ Organize Test Case information</td>
<td>➢ Experience with Web-site testing</td>
</tr>
<tr>
<td></td>
<td>➢ Execute Tests and Validate Results</td>
<td>➢ Well organized</td>
</tr>
<tr>
<td></td>
<td>➢ Produce and Communicate Defects to IT</td>
<td>➢ Methodology oriented</td>
</tr>
<tr>
<td></td>
<td>➢ Eliminate redundancy in prepared cases</td>
<td></td>
</tr>
</tbody>
</table>