



Quality Assurance Manual

Version 1.01

Revision History

[illegible]

Table of Contents

Table of Contents

1	Introducing Quality Assurance.....	4
2	Using This Manual.....	5
2.1	Overview.....	5
2.2	Glossary of Terms.....	7
2.3	Related Documents.....	8
2.4	Document Status Summary.....	9
3	First Steps.....	10
3.1	Getting Started.....	10
3.2	Setting up a Project.....	10
3.3	The Main Quality Assurance Steps.....	11
3.4	Variations on a Theme.....	12
3.4.1	QA for a Small Project.....	12
3.4.2	QA for a Medium Project.....	13
3.4.3	QA for a Large Project.....	14
4	Quality Assurance Processes Overview.....	15
4.1	Concept.....	15
4.2	Define.....	16
4.3	Design.....	16
4.4	Plan.....	17
4.5	Build.....	18
4.6	Integrate.....	19
4.7	Test.....	19
4.8	Install.....	20

1 Introducing Quality Assurance

Quality Assurance helps us achieve the following goals:

Quality	Products and services meet the relevant quality standards. <ul style="list-style-type: none"><input type="checkbox"/> Customer requirements are clearly established.<input type="checkbox"/> Customer requirements are met.<input type="checkbox"/> Systems are reliable.<input type="checkbox"/> Systems are maintainable.
Efficiency	Delivery on time and within budget. <ul style="list-style-type: none"><input type="checkbox"/> Projects are planned effectively.<input type="checkbox"/> Progress is monitored and compared with plan.<input type="checkbox"/> Effective corrective action taken when required.<input type="checkbox"/> Projects are completed within budgeted cost.<input type="checkbox"/> Projects are appropriately documented.
Development	Steady improvements in project management. <ul style="list-style-type: none"><input type="checkbox"/> We develop better ways of doing things.<input type="checkbox"/> Lessons learned are incorporated into future projects.

2 Using This Manual

2.1 Overview

The Quality Assurance procedures in this manual are based on the following broad model.

CONCEPT	<i>Develop the initial idea</i>
DEFINE	<i>Find out what it has to do</i>
DESIGN	<i>Determine how to build it</i>
PLAN	<i>Make plans for building it</i>
BUILD	<i>Build it</i>
TEST	<i>Test it</i>
INSTALL	<i>Install it</i>
MAINTAIN	<i>Keep it going</i>

In practice things get a bit more complex, but the underlying framework is proven. Some common variations on the basic theme are:

This Manual is a guide

This manual describes a suggested framework for quality assurance. It is not mandatory that all aspects of the framework are implemented. Each project will have different characteristics and different requirements. The manual is designed offer flexibility and a mechanism for tailoring quality processes to particular circumstances. This manual is based around the requirements for assuring quality in a new project from its conception to implementation.

A Quality Plan is Mandatory

A quality plan will be created and agreed for all projects. Once the plan is agreed, the processes that it describes will be mandatory.

Iterative process

Whilst coding is being done, typically we will be getting feedback from the customer. This usually results in further refinements to the Specify stage, and may also result in changes to the Design and Cost stages.

This is accepted as normal practice, and project management must control this process in such a way as to enable the work to proceed smoothly.

Parallel work

Some of the above, or parts thereof are often done in

parallel, such as creating test plans whilst design is underway, or starting the User Manuals during the specification stage.

2.2 Glossary of Terms

VVP	<p>Validation and Verification Plan</p> <p>A quality assurance document which describes how the deliverables are going to be validated and verified as being fit for delivery to the customer.</p>
ITP	<p>Integration Test Plan</p> <p>A quality assurance document which describes how the system is going to be tested as an integrated whole, prior to customer acceptance testing and/or delivery.</p>

2.3 Related Documents

2.4 Document Status Summary

Description:	Catalyst IT Quality Assurance Manual
Current Version:	1.01
Location:	boole:/general/catalyst/QA/doc/reference/QA_Manual.sxw
Last saved:	9/12/2005 11:47 a12/p12
Printed:	9/11/2005 9:44 a11/p11

3 First Steps

3.1 Getting Started

Initial Assessment

A project is normally initiated following some form of sales work, discussion of feasibility, the likely profitability, and the strategic importance of the project.

There may have been verbal discussions with the potential customer(s), and possibly the exchange of letters, tender documents, specifications, and other material.

Create Project

At some point the decision is taken whether to move from discussion to a 'real' project.

So that information can be captured for later use without being lost, do this as soon as possible.

Once a project has been created it can be used to store all relevant RFP's, RFI's, e-mails, estimates, preliminary design notes, concept diagrams etc. in electronic format, and these can then be used later on, if formal documents such as requirements specifications need to be written.

3.2 Setting up a Project

A basic HOWTO on this process.

3.3 The Main Quality Assurance Steps

The following phases represent the Quality Assurance steps that a project can go through in its life-cycle. Depending on project size, a subset of these might be appropriate, and this is entirely flexible, just as long as it is documented in the Project QA Plan at the outset.

Phase	Description	Approval reqd.
Concept (all)	A QA Project is created. Conceptual and functional requirements are defined. From these, a preliminary project plan and costing is produced, and the proposed Project QA Plan is defined (a statement of what QA entails, for this project).	Internal Customer
Define (L,M)	Acquire detailed requirements from the customer, and produce a specification document appropriate to the project size.	Internal Customer
Design (all)	Describe how the requirements are going to be implemented.	Internal
Plan (all)	<ul style="list-style-type: none"> Produce a project plan, a maintenance plan and a costing of the work, based on the requirements and design. Produce other plan(s) as required by the QA Plan for the project. This will consist of an overall Verification and Validation Plan, containing details of what test plans will be produced, and the testing schedule, plus the test plans themselves. 	Internal Customer
Build (all)	The production of all required deliverables. Activities will include management of development, progress tracking, and provision for feedback from customer requiring change control on any of the elements above. Normally no QA steps will be required during development, however projects may implement these if required (eg. staged test-plan sign-offs).	
Integrate (L,M)	A phase where development is completed and the system is integrated and acceptance-tested by the development team. I.e. a 'ready for delivery' phase.	Internal
Test (L)	Customer acceptance testing. A project-specific phase where the customer verifies the delivered system meets the requirements. Implies that test plans have been created.	Customer
Install (all)	The 'go live' phase. Install the system on the target environment. Customer usually acknowledges delivery at this point. This is also the hand-over to the maintenance team.	Internal Customer

Key: L – Applicable to Large projects, M - Medium, S – Small, all – All projects (L,M,S)

3.4 Variations on a Theme

The Quality Assurance Steps might appear to be a very complicated and tortuous path to follow, especially for small projects, but never fear, this is where QA can be made flexible. The basic rule is, as long as you state, in clear terms, what the QA process for your project is, and get it QA-approved (signed-off) as such, then you are compliant with the QA standard.

If we consider that the list of steps given in Section 3.3 is the full list, then a lengthy project would implement all of these. We can then perhaps define two other common scenarios to cater for small and medium-sized projects respectively.

At this point we are only considering the big picture – the overall process steps. In later sections of this manual you can find detail on each step, and these details can be varied according to project size as well, so we can tune the QA process considerably to fit.

3.4.1 QA for a Small Project

A small project might be one which involves 1-2 developers, and might typically be of a full-time duration of 1-4 weeks. These steps might then be appropriate:

Concept --> Design --> Plan --> Build --> Install

- ❑ **Concept:** The small project will have its preliminary concept and functional requirements documented and signed off.
- ❑ **Define:** Small projects do not usually produce a separate requirements specification. Instead they rely on the concept/functional specification produced above, and ad-hoc querying of the user.
- ❑ **Design:** A description of how the system is going to be implemented. For a small project this might be no more than a couple of pages or so.
- ❑ **Plan:** Production of a final project plan which documents when the project will be finished, how much it costs, and important milestones. Production of a maintenance plan.
- ❑ **Build:** Development of the small project proceeds with developers coding and unit-testing to the requirements specification.
- ❑ **Integrate:** Usually unnecessary for small projects.
- ❑ **Test:** Usually no formal acceptance testing is done for small projects.
- ❑ **Install:** For a small project there will normally be a single point where the application is finished, and where the customer asked to do final testing and verify that it is working to their requirement. Maintenance takes over site administration.

3.4.2 QA for a Medium Project

A medium project might be one which involves 3-5 developers, and might typically be of a full-time duration of 5-10 weeks. In this case the following steps might be enough:

Concept --> Define --> Design --> Plan --> Build --> Integrate --> Install

With a medium-sized project we have all the steps of a big project minus the 'Test' phase. In addition, the detail to which the documentation goes is expected to be much less than for a large project, and some elements of the various phases will not be required.

This means that the medium project has requirements specifications produced, but isn't tested in a truly formal way either in-house or by the customer. This type of project is normally 'integrated' at the end of development by testing it from the requirements specification directly.

- ❑ **Concept:** The medium-sized project will have its preliminary concept and functional requirements documented and signed off.
- ❑ **Define:** Medium projects will normally be required to produce a separate requirements specification. This will involve detailed discussions with the customer with a view to documenting a complete set of testable requirements and use-cases which accurately describe the system.
- ❑ **Design:** For a medium project, the design document should provide details of the basics, such as database schema design, functional design, interface design, and operational matters.
- ❑ **Plan:** Production of a final project plan which documents when the project will be finished, how much it costs, and important milestones. Production of a maintenance plan.
- ❑ **Build:** Production of all deliverables. Development of the medium project proceeds with developers coding and unit-testing to the requirements specification.
- ❑ **Integrate:** The medium project should be tested against the specification in a rigorous way, prior to final delivery to the customer. This phase can also include customer testing on a staging environment, prior to putting the site live.
- ❑ **Test:** Usually no formal acceptance testing for medium projects.
- ❑ **Install:** The medium project is put live, and the customer asked to verify that it is working to the requirements. Maintenance takes over site administration.

3.4.3 QA for a Large Project

A large project might be one which involves 5 or more developers, and might typically be of a full-time duration of over 10 weeks.

Apart from involving all of the QA steps, the large project is expected to produce documentation across the board which is much more detailed than for a medium project.

- ❑ **Concept:** The large project will have its preliminary concept and functional requirements documented and signed off.
- ❑ **Define:** Large projects will be required to produce a requirements specification. This will involve detailed discussions with the customer with a view to documenting a complete set of testable requirements and use-cases which accurately describe the system.
- ❑ **Design:** For a large project, the design document should provide detailed descriptions of the database schema design, including what each data item is for, a functional design, an interface design, and an operational design.
- ❑ **Plan:** A project plan which documents when the project will be finished, how much it costs and how it will be maintained. An overall Verification and Validation Plan will be produced which contains details of the test plans to be written, and the testing schedule. The test plans for a large project will, at minimum, include a plan which tests all the use-cases described in the requirements specification. Additional plans may be called for, for example to implement regression tests, or to provide for staged milestone testing during development. Developers may also be called on to produce unit test plans for each portion of work undertaken.
- ❑ **Build:** Production of all required deliverables. Development of the large project proceeds with developers coding and unit-testing to the requirements specification.
- ❑ **Integrate:** The large project will be tested using the test plans developed earlier, so that any problems can be rectified.
- ❑ **Test:** The customer undertakes acceptance testing using the test plans developed as part of the project.
- ❑ **Install:** The project is put into production, or otherwise delivered to the customer, who is then asked to verify that it is installed correctly and working to the requirements. Maintenance takes over site administration.

4 Quality Assurance Processes Overview

The following sections briefly describe each of the possible quality assurance phases, and what can take place in them.

For each phase a schedule of possible *Activities* is presented, which should be treated as a guide rather than as a definitive or exhaustive list. The *Outputs* are the tangible products of each phase, such as documents etc, and the *QA processes* are the approvals or reviews which are the essence of the quality process.

4.1 Concept

Activities:

- ☐ Create project.
- ☐ Identify project stakeholders.
- ☐ Document our initial understanding of what is required.
- ☐ Document possible solution(s).
- ☐ Determine preliminary work breakdown structure.
- ☐ Make preliminary effort/cost estimates.
- ☐ Assign project resources.
- ☐ Nominate the project manager.
- ☐ Determine the appropriate QA plan.

Outputs:

- ☐ Quality assurance plan.
- ☐ Preliminary project plan.
- ☐ Concept document.
- ☐ Functional specification.
- ☐ Preliminary design.
- ☐ Feasibility study.
- ☐ Risk analysis.

QA processes:

- ☐ Review of quality plan.
- ☐ Review of concept document.
- ☐ Review of functional specification.
- ☐ Review of preliminary project plan.
- ☐ Review of preliminary design.
- ☐ Review of feasibility study.
- ☐ Review of risk analysis.

Notes: This phase usually represents a handover from a marketing/pre-sales team, to a project implementation team. Before handover can be achieved, the customer should have performed a QA signoff on the concept document/functional specification, the preliminary project plan and costing, and also on how the quality assurance process will be undertaken. It is **particularly important** that the customer is encouraged to recognise the preliminary nature of the outputs, and appreciates that these **will** change during the next phases.

4.2 Define

Activities:

- ☐ Expansion of preliminary requirements.
- ☐ Additional detailed requirements gathering, JAD sessions etc.
- ☐ Place an emphasis on testable statements and definitive use-cases.
- ☐ Optionally look forward to potential design(s) and modify requirements accordingly.

Outputs:

- ☐ Requirements specification documentation.

QA processes:

- ☐ Review of requirements documentation.

Notes: The requirements specification is a critical piece of the project jigsaw. The contents must be written in such a way as to make it easy to test the system by referring to them. To facilitate this, requirements should be written in a highly structured format with each item corresponding to a numbered section for easy reference. In addition, each requirement item should ideally be written in such a way as to be 100% testable; eg. instead of stating a webpage should be 'quick' to save, put down a measurable requirement such as 'must save in under 2 seconds'.

4.3 Design

Activities:

- ☐ Derive and document the implementation specifics, from the requirements.
- ☐ Determine the operational environment (hardware, software)

Outputs:

- ☐ Design documentation.

QA processes:

- ☐ Review of design documentation.

Notes: The design is an essential document for any size of project, but may vary a great deal in detail according to the project. Small projects might just provide a page or two describing the basic implementation details for example. The requirement is to provide enough detail to allow peer review of the intended implementation, and also to allow on-going maintenance tasks to be scoped. The maintenance document must contain everything that both System Administrators and Maintainers need to know in order to provide on-going support and maintenance for the system.

4.4 Plan

Activities:

- ☐ Review requirements, design and formulate work breakdown structure.
- ☐ Refinement of work-package estimates, project plan.
- ☐ Negotiation for resources.
- ☐ Refinements to QA plan.
- ☐ Production of overall plan for testing the system.
- ☐ Production of test plans.
- ☐ Creation of test environments, infrastructure, data.
- ☐ Creation of disaster recovery plan.

Outputs:

- ☐ Maintenance plan.
- ☐ Project plan (including WBS).
- ☐ Validation and verification plan.
- ☐ Test plans.
- ☐ Testing schedule.
- ☐ Test environment.
- ☐ Disaster recovery plan.
- ☐ Assembled project team.

QA processes:

- ☐ Review of maintenance plan.
- ☐ Review of project plan.
- ☐ Review of validation and verification plan.
- ☐ Review of acceptance test plans.
- ☐ Review of disaster recovery plans.

Notes: The project plan is another keystone phase. It sets out the work packages to be done, details of their cost, and it provides the delivery schedule. All of these are peer reviewed, and then agreed with the customer via the QA approval process. This phase also finalises the project team, and the Project QA Plan which will be adopted for the remainder of the work.

The verification and validation plan (VVP) contains details of the way testing is going to be implemented for the project. It should contain details of what test plans are to be written, and also the testing schedule for implementing the testing. Most importantly it should describe the way that quality assurance is to be implemented for testing.

4.5 Build

Activities:

- ☐ Application building.
- ☐ Project tracking.
- ☐ Project progress meetings.
- ☐ Progress reporting to customer.
- ☐ Code reviews.
- ☐ Unit testing.
- ☐ Module testing.
- ☐ Actioning change requests.
- ☐ Tracking issues, resolving issues.

Outputs:

- ☐ Progress reports, milestones.
- ☐ Change requests.
- ☐ User manual.
- ☐ Maintenance manual.
- ☐ Implemented system.

QA processes:

- ☐ Review of maintenance manual.
- ☐ Peer reviews of the codebase.
- ☐ Internal test plan approvals.
- ☐ Site acceptance test plan approvals.
- ☐ Change request approvals.
- ☐ Review of user manual.

Notes: The QA processes which are implemented during this phase are described in the Quality Assurance Plan. For small/medium sized projects this might well be minimal, and general project management might go ahead unfettered by any kind of QA approvals process. However for some projects it may be desirable to build QA validation into the development, such as staged site acceptance testing and sign-off for each tested module/milestone. This might involve customer approvals, or it might just be done internally, depending on the project.

4.6 Integrate

Activities:

- ☐ Assembly of all modules, data and equipment.
- ☐ Testing of the integrated system.
- ☐ Data conversion from legacy system.
- ☐ Loading of base data.
- ☐ Volume testing.
- ☐ Performance testing.
- ☐ Soak testing.
- ☐ Planning for installation.

Outputs:

- ☐ Integrated system.
- ☐ Change requests.
- ☐ Installation plan.

QA processes:

- ☐ Review of integration test results.
- ☐ Review of site acceptance test results.
- ☐ Review of installation plan.
- ☐ Change request approvals.

Notes: The QA processes in this phase might vary significantly by project. The golden rule is that the integrated system should be tested in-house in exactly the same way that the customer will be testing it. A medium-sized project would probably just therefore the reviewer to go through the requirements specification document and test rigorously to the statements made in it. However, a large project would have a customer acceptance test plan, and this would be used instead.

4.7 Test

Activities:

- ☐ Set up testing environment, data etc.
- ☐ Acceptance testing.

Outputs:

- ☐ Test results documentation.
- ☐ Change requests.

QA processes:

- ☐ Review of acceptance test results.
- ☐ Change request approvals.

Notes: The work which went into the specification and test plans should make this phase fairly simple to action. Generally the acceptance testing will be done by the customer, and test results used to generate change requests and testing approvals/sign-off.

4.8 Install

Activities:

- ☐ Setting up the production system and its environment.
- ☐ Preparations to hand over to the system administrators and maintainers.
- ☐ Evaluation of QA processes in this project.

Outputs:

- ☐ Production system.
- ☐ Training material.
- ☐ Post-installation feedback on QA processes.

QA processes:

- ☐ Review of handover to system admin & maintainers.
- ☐ Review of maintenance plan.
- ☐ Review of delivered system.
- ☐ Post-installation review.

Notes: This phase should be very straight-forward if all other phases of the project have gone smoothly. The integration phase should have done just about all of the work.