



HP Quality Center Version 10.0 Upgrade

HP Software & Solutions
Professional Services
Best Practices White Paper

January 2009
Document Version 1.3

Document Controls – Change Record

Date	Author	Version	Change Reference
12-14-08	James Vandenbossche	1.0	First draft
01-09-08	James Vandenbossche	1.1	Enhancements and edits from technical review and feedback from field experts
01-12-08	James Vandenbossche	1.2	Enhancements and edits from review and feedback from product teams and engineers
01-13-08	Sandra Herber	1.3	Editorial review

Contributors

Name	Title
James Vandenbossche	Quality Management Global Practices – HP Software & Solutions Professional Services

Table of Contents

- 1 Overview4
- 2 HP Quality Center is a Mission-Critical Application4
- 3 Enhancements and Upgrade Impact4
 - 3.1 Architecture5
 - 3.2 Asset Sharing5
 - 3.3 Versioning.....5
 - 3.4 Project Customization Management6
 - 3.5 Project Verification and Repair.....6
 - 3.6 Enhancements to Test Resource Management7
- 4 Upgrade as a Phased Approach.....7
 - 4.1 Phase 1: Preparation and Planning7
 - 4.2 Phase 2: Performing the Upgrade.....8
 - 4.3 Phase 3: Quality Center Project Configuration and Customization.....8
 - 4.4 Project and Engagement Review8
- 5 Conclusion9



1 Overview

This document provides a comprehensive overview of the HP Quality Center version 10.0 upgrade process. The release of HP Quality Center 10.0, with the simultaneous release of HP QuickTest Professional 10.0, includes many significant enhancements and new features that have resulted in an increased need to ensure that the QC upgrade is more strictly managed and controlled. As a result of these changes, the upgrade process has become more complex. More than at any time in the past, product knowledge and experience with the upgrade process are essential for a well-managed, predictable, and successful upgrade. Quality managers need an optimal plan that minimizes the risk to the business, with guaranteed safety of test data and assets against data loss and the least amount of lost productivity to live test cycles.

2 HP Quality Center is a Mission-Critical Application

The importance of HP Quality Center's availability and performance must be clear to everyone. An outage could result in the stoppage of all testing across all lines of business leveraging this quality management system. This system contains all of the requirements that need to be covered and verified, the tests designed to validate these requirements, the test sets and results of the test executions, and the defects that development must repair. QC is the repository of information and KPIs on which quality managers are dependent to make strategic decisions to improve product quality and direct their resources. An outage of even a day could result in a schedule slip of a critical release date, which can directly impact a corporation's bottom line.

Upgrading is a complex and sensitive process requiring careful planning and coordination. Each of the following six requirements must be met to assure a successful upgrade:

1. Minimize Quality Center down time and lost user productivity
2. Ensure the safety and proper upgrade of the Quality Center projects and all of the data assets they contain
3. Familiarize and mentor end users on the many new features in Quality Center 10.0
4. Configure Quality Center projects to maximize the benefits delivered by the new features
5. Maintain customizations to existing Quality Center projects as well as any integrations with external applications and systems
6. Provide essential communication to all stakeholders regarding upgrade plans, timelines, and expected impacts

3 Enhancements and Upgrade Impact

The many enhancements and new features in Quality Center 10.0 have resulted in additional complexity to the upgrade process. As with the upgrade of any mission-critical system, thorough planning and careful preparation for the Quality Center upgrade are more important than ever before. The impact of all product enhancements must be assessed for the Quality Center operations team, the project customization administrators, and the end users (testers, managers, and application teams). First changes should be assessed in terms of their support of and integration into the organization's strategic methodologies, processes, and practices, and then implemented in

a well thought out and orderly manner during the upgrade. This will often involve key management stakeholders and application teams, in addition to the quality assurance organization.

Adoption of the eagerly anticipated ability to version assets stored within the Quality Center repository will impact the way data is stored and managed within individual Quality Center projects. This, coupled with the new capability of propagating customizations to multiple projects via Templates, will drive new practices determining what data to share and when to share it. Practices for how to manage the assets over time and across teams will require refinement over time, specific to an organization's software development and quality assurance methodologies.

During the upgrade, the actual running of the Quality Center software installer is very much the same as in previous versions. Similarly, any integrations or dependencies, whether with third-party applications or other HP products, need to be considered. Potential added complexity comes from the stringent automated scanning and verification required for Quality Center projects prior to upgrading them, which may require manual adjustments and repair.

The following sections briefly describe the major product changes and enhancements, how they impact upgrade planning, and how they should be addressed.

3.1 Architecture

A full assessment of software and hardware must be made to ensure that the target architecture for Quality Center 10.0 meets any new operational requirements or recommendations. In some situations, a phased upgrade may be required (upgrade first to Quality Center 9.x and then to 10.0). In this case, the plan must also account for simultaneous operation of both the 9.x and 10.0 Quality Center instances. It is also important not to neglect any requirements dictated by specific integrations of the third-party or custom applications with Quality Center. HP Software & Solutions Professional Services will review the Quality Center operational policies and practices, and make recommendations for optimal performance and highest availability.

3.2 Asset Sharing

Quality Center 10.0 supports sharing and reuse of asset libraries across projects. Sharable libraries help you manage testing initiatives involving multiple applications and verify that changes to one application don't negatively impact another application. Specific changes can be applied to the shared assets (requirements or tests, for example) for each project while allowing the library to maintain its integrity so that individual projects can re-synch with the library as required.

To take advantage of this new feature, Quality Center projects need to have similar schemas. HP Software & Solutions Professional Services recommends deploying this new feature as a follow-on project phase separate from the Quality Center 10.0 upgrade, as it merits an assessment of existing QA processes and practices, which it may impact.

3.3 Versioning

To support asset sharing and reuse, built-in version control is provided in Quality Center 10.0 for requirements, tests, test scripts, and business components. Versioning allows distributed teams to collaborate and manage multiple versions of test assets in parallel, while providing an audit history

of changes throughout the project lifecycle. Baselining allows capture of a group of requirements, tests, or test assets at strategic points in the project lifecycle to mark specific milestones. Baselines can then be compared to assess the impact of changes and enable rollback of assets, if required. Rollback is supported by allowing the creation of a new version of the asset copied from a historical version of the asset.

Note that with the release of these new built-in version control features, the version control solutions used in past versions of Quality Center are no longer supported and the Quality Center 10.0 installer will not run. HP will provide a migration utility to input this information into the new Quality Center 10.0 version history tables. If version control has been used, it is recommended to postpone the upgrade to Quality Center 10.0 until the utility is available. Familiarization with these changes and planning should begin early in order to scope and schedule the work necessary to define and adopt new methodologies to take advantage of these new features.

3.4 Project Customization Management

One of the most important changes introduced with Quality Center 10.0 is built-in management of master template projects. The new "Template" project type allows automatic project customization synchronization from a template project to multiple production projects linked to it. Quality Center application administrators, as well as project administrators, should be mentored on these new features and participate in the planning phase prior to the upgrade. Current master template projects will need to be converted to new version 10.0 templates, and copies of production projects should be used to scope the effort necessary to adopt the new features. Quality Center project fields and workflow code will require significant review and normalization to support these new features. This work should be executed after the site upgrade as a follow-on project.

3.5 Project Verification and Repair

After Quality Center 10.0 has been installed, existing projects must be upgraded before they can be activated, enabling end-users to begin working in them again. These projects are either present after an in-place Quality Center upgrade or copied from another Quality Center installation and restored. Quality Center will not be able to activate the old projects until their database tables and data are updated to support the new features. To do this, Quality Center 10.0 introduces, for the first time, a very stringent automated verification process that each project must pass. A separate project Verification and Repair tool is provided inside the Quality Center 10.0 Site Administration tool and as a stand-alone utility for use on Quality Center 9.x instances. The use of this tool follows a two step process:

1. **Verification:** The tool analyzes the Quality Center 9.0 and 9.2 schema and data for compatibility with the Quality Center 10.0 schema. Typical issues associated with environment and schema structure are detected and reported in this step. The tool also provides recommendations on how to resolve these issues.
2. **Modification:** The tool then modifies the data and schema so that the actual project upgrade can be completed successfully.

This utility will detect all issues, but will repair only problems that require fixes that do not necessitate data loss. These other fixes must be completed manually. Examples of such problems are: additional tables, extra views, extra columns, column size mismatches, column type

mismatches, incorrect column attributes, extra indexes, extra constraints, extra triggers, and extra Oracle sequences. Assessing these issues and their impact requires a thorough understanding of the Quality Center database schema and the actual data in question.

3.6 Enhancements to Test Resource Management

Quality Center 10.0 introduces a whole new way of centrally storing and managing five QuickTest Professional and Business Process component resource types: function libraries, data tables, object repositories, recovery scenarios, and environment variables. These features add traceability from the shared resources to the tests that consume them, allowing the user to analyze the impact of change to any of those shared test resources. These new resources are stored in Quality Center's new Test Resources module. These enhancements are extended only to QuickTest Professional tests and Business Components automated by QuickTest Professional, while WinRunner will have the same support in Quality Center 10.0 as it does in version 9.2. Quality Center 10.0 supports only QuickTest Professional 10.0 and no previous versions. HP has provided a migration utility with QuickTest Professional 10.0 for upgrading Quality Center projects by mass-upgrading all QuickTest Professional based tests assets and their associated resources. Upgrading to QuickTest Professional 10.0 simultaneously with the Quality Center upgrade requires careful planning and orchestrating of the process. The recommended approach is for customers to upgrade to QuickTest Professional 10.0 while still using Quality Center 9.2, which is backward-compatible and will work without using the new test resource features until used with Quality Center 10.0. Finally, end-user training on the new features of QuickTest Professional, Business Process Testing, and the new dependencies with Quality Center is essential prior to the software upgrade.

4 Upgrade as a Phased Approach

Based on many years of experience helping our global customer base upgrade their Quality Center deployments, HP Software & Solutions Professional Services has found it most effective to plan and deliver Quality Center upgrades as a three-phase project.

4.1 Phase 1: Preparation and Planning

With the important new features introduced in Quality Center 10.0, the tasks performed as part of the upgrade preparation and planning phase have expanded from past releases. Before upgrade planning can begin, the operations staff, Quality Center application administrators, project administrators, and end-user teams must be familiarized with the new features and the value that these features bring to their quality practices and application development processes. The decisions of how and when new features are to be adopted impact the upgrade plan and schedule.

Secondly, a quality management assessment must be conducted to uncover and document all factors that could potentially impact the upgrade project. Examples of such factors include anticipated changes to the organization's software development processes, anticipated large application roll-outs on the horizon (e.g. SAP), expansion of the global QA staff, desire to adopt ITIL V3 processes, or plans to integrate with automated change control processes.

Third, an architecture review must be completed, documenting all upgrade dependencies and factoring them into the upgrade project plan and timeline. This may include hardware provisioning

or repurposing, database upgrades, integration with third-party software packages, updates to corporate security policies, or network assessment and support.

Lastly, the effort of upgrading existing Quality Center projects is dependent, as mentioned in section 3 above, on the project verification and repair process. To prevent downtime and costly surprises, all potential problems in the projects should be well understood and repaired before the actual Quality Center upgrade.

At the end of the preparation and planning phase, an upgrade plan will have been developed, agreed upon, and signed off by all required stakeholders. This document will serve as the project plan for the next phase in the project.

4.2 Phase 2: Performing the Upgrade

After the Quality Center upgrade plan has been developed and signed off, performance of the upgrade commences. The Quality Center application may be either upgraded in place or installed in a new environment. After Quality Center 10.0 has been installed, existing projects may be upgraded to the new version. Any projects storing QuickTest Professional tests must be bulk-upgraded with the migration utility provided by HP on the product installation disk.

After the desired Quality Center projects have been upgraded for version 10.0, they must be validated as ready for access by end users. At this point, end-user teams can continue with their work while phase 3 of the upgrade project commences.

4.3 Phase 3: Quality Center Project Configuration and Customization

In this phase, the Quality Center project customizations are implemented to support the new features in version 10.0 as well as to accommodate any desired workflow changes. For instance, new workflows are available in the Components and Releases modules. Adopting the new asset-sharing features requires that project fields be standardized and normalized across the desired projects. HP Software & Solutions Professional Services uses its thorough understanding of the QC project schema and internal tools to provide a safe and efficient migration. In addition, creating and deploying the new template projects may require converting 9.x project customizations to the new template project type, including workflows and project entities.

This phase of the project can potentially be quite extensive and will require significant involvement from the Quality Center's project template administrator who understands the existing QA processes and is responsible for all of the customizations in the Quality Center projects to support them.

4.4 Project and Engagement Review

A comparatively short, yet very important phase of every Quality Center upgrade project that HP Software & Solutions conducts is the engagement review. Together with all of the key stakeholders, the project requirements, tasks undertaken, problems encountered and how they were resolved, any remaining open issues and their current status, and, finally, recommendations are reviewed. A final report is provided capturing all of this information, which can be later used for reference in future upgrades, integrations, and expansions.

5 Conclusion

Quality Center is a mission-critical application, and upgrading it requires careful evaluation, planning, and execution to minimize downtime, lost productivity, and the risk of data loss. Careful planning and execution are also key to obtaining high user adoption, unlocking the greatest value from the product's features, and achieving the shortest time to value. HP Software & Solutions Professional Services provides unique product knowledge and global expertise to guarantee a successful upgrade by providing a personalized plan that takes into consideration your current and future goals, your resources and constraints, and the specific needs of your end-user community.