



# Enterprise Resource Planning (ERP) functional testing best practices:

Ten steps to ERP systems reliability



## Table of contents

Introduction .....	.2
Step 1: Initiate planning and gather requirements. ....	.3
Step 2: Define test goals and select appropriate tests. ....	.3
Step 3: Define objectives to meet testing goals. ....	.4
Step 4: Identify functionality test cases. ....	.4
Step 5: Document key business processes. ....	.5
Step 6: Develop modular test components. ....	.5
Step 7: Set up a test lab. ....	.5
Step 8: Understand and leverage the “smoke test.” ....	.6
Step 9: Execute regression tests. ....	.6
Step 10: Analyze defects and create test reports. ....	.6
HP QuickTest Professional software .....	.7
Functional testing of ERP applications .....	.7
For more information .....	.8

---

## Introduction

Enterprise Resource Planning (ERP) software applications enable you to manage a wide range of essential business functions, including planning products, purchasing parts, maintaining inventories, interacting with suppliers, providing customer service and tracking orders. Some ERP solutions also may include application modules for finance and human resources. Although these applications usually do not directly generate revenue, they enable you to use existing customer data in an effective, efficient manner. This helps streamline your business activities, allowing you to deliver higher-quality service to new and existing clients.

ERP applications often use a single, centralized data repository for all modules. Therefore, when these applications have performance issues they can potentially impact all areas of your business that use the common repository. This connectivity creates the need for robust testing and monitoring programs to improve the health of applications critical to your company's mission.

When testing ERP applications, you should always take a holistic approach. Business process transactions impact many departments and divisions within a company, and modules within the ERP application itself. When validating process functionality, it is important to capture tests of these business processes within automated testing solutions for rapid repeatability. The inherent complexity of crossing the multiple business stacks makes it critical to functionally test every ERP application.

During ERP implementation, it can be difficult to satisfy business requirements before you release the application into production. You should test and verify these applications so they perform optimally. Through thousands of customer engagements, HP Software has compiled a list of best practices to help you verify the functionality of essential business applications. The following sections describe the ten key steps you can use to design and implement effective functional testing programs for your key ERP applications.

## Step 1: Initiate planning and gather requirements.

Planning is one of the most critical stages of functional testing in any environment. This step is even more important with ERP applications because of the complexity of the environment and the intricate business requirements that drive these implementations. Poor planning can lead to confusing results and inadequate test coverage. Well-devised planning allows you to measure and enhance your testing efforts for greater return on investment (ROI).

Many companies purchase prepackaged ERP solutions to allow for quick integration of all areas of business administration. However, you should not deploy an ERP package without first customizing it to your business. It is important to first know your business's requirements since these are typically used to develop customized ERP applications, to test the system and its customizations, and to obtain final sign-off. By carefully defining these requirements in the beginning, you can better plan and manage the tests to meet your needs. The requirements can then be linked to the tests and actual test results (identified defects) to provide complete functional testing coverage.

## Step 2: Define test goals and select appropriate tests.

By creating primary testing goals, you can determine the particular type of testing that is required. You can also generate test objectives, project plans and team structure from these goals. Listed below are a number of different types of validation tests that you should perform when function testing an ERP implementation.

- **Data mapping:** Since many ERP implementations are tightly integrated with backend legacy systems, it is essential that you test the data mapping between the ERP application and the legacy system. Corrupt or invalid data may be located inside the legacy systems, which could cause display issues in the application.
- **Business process tests:** Testing can validate that your business processes are functioning correctly from build to build. Because workflow is essential to enforcing business rules, these tests should cover all navigations and direct functions throughout the entire integrated system. You must fully test the application's business rules and triggers so that all rules are executed properly.
- **Access control systems:** ERP access control systems determine what views you can use and what data you can see in those views. Using supply chain and partner portals will increase security concerns. Testing the security from your perspective helps enforce the validation rules. Data-driven tests will allow IT to use the same script with different log-in credentials to validate security rules.
- **Regression testing:** Each time that a "code drop" is deployed, you should regression test the functionality of each object inside the applet, including the existence, functionality, values, etc. A code drop refers to any new release of the ERP application, patches and/or hot fixes.

### Step 3: Define objectives to meet testing goals.

After you have defined all goals, you should then develop a set of objectives that will achieve these goals. Listed below is a common set of initial objectives:

- Analyze the application functionality and identify critical business processes. For example, a critical business process within an ERP application is the creation of a “service request.”
- Establish a “smoke test” to quickly perform this testing within the development build cycle. The smoke test should test the critical business functionality or delve deeply into the functionality of the application. For example, can you create an activity that is to be attached to a “trouble ticket?”
- Run the smoke test after each official build.
- Start to create automated tests to reduce the manual effort of running the smoke test.

After you have achieved the initial objectives, you should create and agree to a set of follow-up objectives. An example is listed below:

- Analyze the application to identify the functionality that would broaden the testing effort to include more than 75 percent of the total application functionality. (Achieving 100 percent script automation is very difficult, since automated testing cannot report on such issues as usability testing.)
- Build automated tests that pass on a consistent basis to reduce the testing effort.

### Step 4: Identify functionality test cases.

When identifying test cases, remember the application must achieve critical business functionality. Most corporations do not go live with basic or standard implementations because of the unique business requirements of each enterprise. Therefore, you will need to exhaustively test the areas that are being customized in order to facilitate functionality in production. One of the main benefits of ERP applications is their ability to integrate with existing legacy systems to satisfy your business needs. However, these integrations must be heavily tested because they are not part of the standard (or non-customized) implementation.

Initially, you should not test the same functionality in many different ways. The application is designed to be flexible enough to allow you many different ways to accomplish your daily tasks. It is important to always deploy test cases that enable requirement-driven, user-path coverage. Initial tests should have some of the following common characteristics:

- They should test mission-critical business functionality.
- They should test key business processes of the application.
- They should identify the test areas of the ERP application that have been customized.
- The functionality should be stable and not under major development.
- The tests should be candidates for a “smoke test.”

Once you have constructed and successfully run the initial automated tests, your objectives will change and the set of tests will expand as the application functionality matures. You may also add tests to routinely inspect problem areas in the application, such as interfaces to legacy systems.

## Step 5: Document key business processes.

When you document business processes that will become test scripts, first gather all of the information relevant to the test case. Each test case needs to have a stated purpose that relates back to the area of the business that is being tested. The test objective should satisfy a requirement or set of requirements. You should also document the logical steps needed to navigate through the system to fulfill the test's purpose. Since the test cases are being used to measure the success or failure of your business processes, the documentation should contain a description of what needs to be verified for a test to be successful.

Along with the navigation and verification required for test cases, you also need valid set(s) of data values to successfully navigate through the test case. This data can be master data (data taken from a database), user-created input data that can be added as you proceed, or prepared data that is seeded into the database before the script is created.

## Step 6: Develop modular test components.

Creating modular test scripts allows you to unit test your modules and the customizations made to the ERP application modules before the entire system is completed. Then, they can be combined into test sets to accomplish specific testing goals. HP offers a next-generation functional testing solution, HP Business Process Testing software, which enables you to manage all test cases in terms of business components and end-to-end process validations.

## Step 7: Set up a test lab.

We recommend you set up a separate quality assurance (QA) test lab as part of your overall strategy for testing and refining the ERP application. By having all machines configured to the desired specifications, you may reduce any issues caused by poorly configured machines. This also allows you to use the lab to perform the unit test prior to promoting the new code to the next release.

## Step 8: Understand and leverage the “smoke test.”

With most ERP applications, businesses waste a lot of testing effort on incomplete builds. This is common when a development organization performs a build and passes it to the test team, and then testing progresses for a few days. As a result, the software build frequently has significant and fundamental problems and doesn't warrant in-depth testing. Unfortunately, the test team has already wasted a few days discovering this instability, while the development staff proceeded to add more new features to the unstable baseline.

One way you can avoid this situation is by building a “smoke test” that consists of your critical business functionality. This is often a combination of manual and automated tests created in a short period of time that will run quickly (usually in less than one hour). Running the smoke test will give your development team immediate feedback on the build's quality and can help your team focus on more serious problems. The smoke test's scripts can be taken from the unit tests that your developers have already created.

## Step 9: Execute regression tests.

You should run regression test sets comprised of the critical business processes every time a new build of the ERP application is released. A regression test set delves deeper into application functionality than a smoke test, which focuses solely on the core business functionality. Because the updates are provided by the vendor, customizing the applications can have an adverse effect on functionality and performance. As a result, these applications must be tested with every build.

## Step 10: Analyze defects and create test reports.

You will be able to detect the readiness of the ERP application by the number of defects found within the system. As the test sets are executed, you must track and analyze failures that occur within the tests. A robust functional testing solution should provide the ability to track and report on all of the defects located within the business process. Your team can use this information to measure and manage how defects are prioritized, fixed, retested and closed.

You must also thoroughly document all testing processes and outcomes with comprehensive reports. This will enable your test teams to accurately analyze test results, and to reuse test cases and scripts in the future.

## HP QuickTest Professional software

HP QuickTest Professional software is an advanced automated testing solution for building functional and regression test suites. It addresses every major software application and environment, including ERP applications from Oracle® and SAP. This automated testing solution uses keyword-driven testing approaches to capture, verify and replay user interactions automatically, and helps you quickly identify and report on application defects while providing advanced functionality for tester collaboration.

With HP QuickTest Professional's keyword-driven approach, you have full access to the underlying test and object properties through an integrated scripting and debugging environment that is round-trip synchronized with the keyword view.

HP Business Process Testing is an automated and manual functional testing framework solution for test design, test creation, test maintenance, test execution and test data management. It helps you align testing with business objectives and dramatically reduces the test cycle times. It achieves this by leveraging business subject-matter experts in the test design and quality optimization process, automating the test-plan documentation creation and streamlining test maintenance for application change.

HP Business Process Testing works together with HP QuickTest Professional and uses a "script-less" framework testing design that satisfies the needs of both technical and non-technical users. It allows you to deploy higher-quality applications faster, cheaper and with less risk.

## Functional testing of ERP applications

By using HP QuickTest Professional and HP Business Process Testing, you can develop and use consistent, repeatable testing processes that make it faster, cheaper and easier to predetermine ERP application readiness. After the initial functional test plan is completed, you can use HP software to automatically verify the integrity of all business transactions within ERP applications. HP software approaches ERP application testing from the business process perspective. These solutions test the ERP application by operating it like a real user would—performing step-by-step operations, such as updating inventory information or ordering a part from a supplier.

While capturing business processes during the test creation phase, HP QuickTest Professional and HP Business Process Testing separate the ERP business logic from the input data to allow you to vary selections and data entry based on a list of choices. Testing the application repeatedly with the same data doesn't always yield realistic results. To truly verify the functionality of an application, you need different sets of data to emulate real-life actions of many users. HP software allows you to input data directly into the test or import it from a database to create realistic data-driven testing scenarios. This capability allows you to analyze the actual ERP business process using variable input data.

Packaged ERP applications are usually very complex. Creating a simple customization for a record can unexpectedly impact other records or overall performance. When updates are released, even for simple customizations, you must thoroughly test all business processes, not just where the change occurred. This allows you to measure the actual impact of the updates and avoid defects.

## For more information

HP software offerings enable you to adopt robust functional testing programs for all of your ERP solutions. The solutions empower your testing team to create sophisticated test suites with less training; help to enable correct application functionality across all environments, data sets and business processes; and allow you to fully document and replicate defects for developers. This enables your team to fix defects faster and meet aggressive production deadlines.

For more information about functional testing of ERP solutions, HP QuickTest Professional, HP Business Process Testing, or other HP software and services, contact your local HP Software sales representative or visit [www.hp.com/go/software](http://www.hp.com/go/software).

---

To learn more, visit [www.hp.com](http://www.hp.com)

© Copyright 2007 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Oracle is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

4AA1-3653ENW, JULY 2007

