SIEMENS
Ingenuity for life

Polarion QA benefits guide
Improving test management and quality assurance

Testing of software and hardware is indispensable for avoiding problems. Companies are hiring more and more test managers and acquiring software tools to automate testing processes. These tools should support test managers and not constrain them, allowing them to organize their work in the best way for their organization. This paper describes the benefits of Siemens PLM Software's Polarion QA™ test management solution.

Contents

Pains of a test manager

In 2009, Ford had to recall 4.5 million cars because of a defective cruise control. In 2005, Mercedes had to recall 1.3 million cars to test the electronic braking system of their E-Class and CLS models. In 1996, the first Ariane 5 rocket detonated shortly after takeoff because of an error in the control software, which cost about $370 million. In 1994, an Intel Pentium processor caused a rounding error that cost Intel about $475 million.

Some of these incidents might well have been avoided if they were tested correctly. Nowadays testing of software and hardware is indispensable for companies. To try to avoid such problems, companies hire more and more test managers who work with software tools to support their tests. These tools should support test managers and not constrain them, allowing them to organize their work in the best way for their organizations.

One of the most common tools for testers is Microsoft® Excel®, which has both pros and cons.

Pros:
- You can easily change test scripts

- You can send them to partners who can execute the test scripts as well

- It is easy to copy/paste the tests that are conducted most often

Cons:
- Excel's reporting features are limited if you have to show which test cases are executed, failed or passed

- Collaborating concurrently with someone on a document isn't possible

- Linking test cases to requirements isn't possible in an easy way

- No version management of your test cases

Test managers and testers are increasingly hitting the limits of Microsoft Excel and are looking for a tool that covers or improves the pros, and mitigates the cons of the list above. Polarion QA, Siemens PLM Software's test management solution, is rapidly being recognized as an outstanding option.[1]

Let us look now at some benefits of the Polarion QA test management solution, explained with use cases that are common in testing and test management.

Top eight benefits of Polarion

1. Design your testing process to your needs

A company that has been working successfully with defined processes for years doesn't want to change their processes because a new tool doesn't support the way they work best. It should be easy to implement your processes in a new tool. In almost the same manner, test cases can have different attributes and content such as requirements, defects, etc. It is easy to implement your processes or look and feel in Polarion QA by configuring your own work items.

Work items

A work item is the Polarion QA term for an artifact of your development process. A work item can be anything you want to track in your project. Polarion QA comes with several pre-defined work item types for test cases, software requirements, defects, activities and change requests. Custom work item types for work products, safety goals, etc. can be defined as required.

Work item data fields, custom fields

Each work item has a number of default data fields, which are used to describe and categorize the item, assign it to someone, incorporate it into project planning and tracking, set its status, and so forth. Custom fields can be defined for any work item type, enabling tracking of and querying on any kind of information. For each work item type the look and feel is completely customizable.
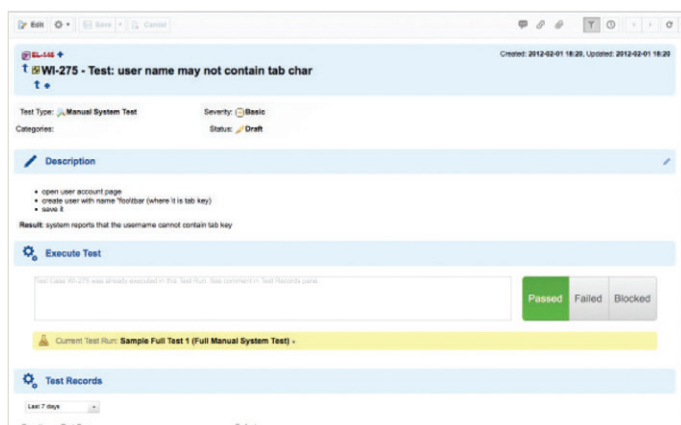

2. Control the lifecycle of test cases

When the pressure is on, it is so easy for people to forget some process step, take shortcuts, or even bypass or circumvent something in order to (supposedly) save time or cost. But as mentioned in the introduction, that could be a recipe for disaster down the road. And if your company must comply with strict government regulations and standards, not only must process be followed, but you must be able to prove it for auditors. Here again, a process based on office documents quickly becomes cumbersome and error-prone.

With Polarion's integrated workflows, customizable to support any process, standard or regulatory norm, steps simply cannot be missed or bypassed at crunch time. People don't have to know the whole process, or who owns each step. They simply process assigned tests, tasks, etc. and mark their status. Items automatically move on to the next workflow step, everyone who needs to know is notified, and an audit trail showing who changed what, when, and why is automatically recorded in the underlying history.

Work item lifecycle/workflow

A test case can have its own lifecycle or workflow definition. A "workflow" is a set of statuses and status transitions, transition conditions and dependencies that a work item passes through in its lifecycle. Each of its elements, statuses, transitions, conditions and dependencies can be customized. This means that you can customize workflow to support any process.
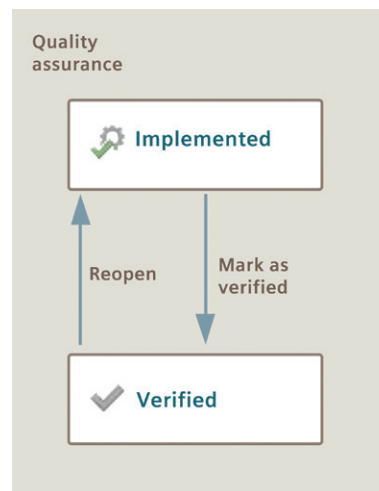
3. Ensure that all relevant software requirements are covered by test cases and all test cases are consistent with their requirements

If you don't test every requirement, defects can occur after the product is released. This can be very expensive as you can see from the examples at the beginning of this paper. You need to guarantee that every requirement will be covered by a test case. So all test cases must be connected to the requirements they verify.

This connection is accomplished by linking work items in Polarion QA.

Link between work items, link attributes/roles

Linking test cases to requirements or other work items is the key to taking advantage of traceability and impact analysis features. Test cases can be linked inside one project, between different projects and even between different repositories. This allows you to relate test cases for different products and/ or product variants and to get traceability and impact information that is not limited by project scope. Furthermore you want to know which test cases are available for your requirements.

A link between work items is defined and categorized by "link roles." Link roles are distinguished by their names (relates to, implements, verifies, etc.). They can have different semantics if needed, and can be customized to meet your specific needs. Using links, it is easy to manage your complete testing flow from the concept phase down to your hardware and software testing and related activities, work products, risk items and/or requirements.

Suspect links

From time to time, test cases or requirements must be updated or changed. But what happens to the test cases that are refinements of the updated test cases? The testers must be notified about the changes. Maybe the refinements need an update, too.

A "Suspect" link can be a solution. Work items have a "suspect" attribute that affects links and an "auto-suspect" option that, when enabled, automatically sets the suspect attribute to true for new links between work items. You can toggle the suspect attribute on and off using the "Suspect" check box available for each linked work item in the "linked work items" section of the work item viewer/editor.

Linked work items with the auto-suspect option in effect.

This project is stable at the moment. Now suppose someone makes a change to the Work Item WI-4 as shown below.

Work item changes at level 2

Notification propagation when a work item changes.

When a work item is changed at this level, the links to work items that WI-4 impacts become suspect, and the owners of the impacted Work Items (WI-1, WI-2, WI-3) are notified by email.[2] These owners can assess whether the changes to WI-4 impact their work items. Notice that the owner of WI-5 is not notified because changes to WI-4 don't impact WI-5. If work item WI-5 were to be changed, then the change would impact WI-4 and the owner of that item would be notified (see figure below).

Suspect link

Notification propagation when a work item changes at a different impact level.

The owner of WI-4 can then determine whether or not changes are needed to that work item. If WI-4 is changed as a result of the change to WI-5, then the owners of WI-1, 2 and 3 are notified as shown previously.

4. Customizable test specifications from ready-made templates

Most companies have constant structures for how a document should look or what information is stored in it. You don't want to create a new document from scratch every time. Rather, you want to re-use the existing templates.

Maybe your company has a test plan or something similar that is used every time you create a new project or test run, with an equal structure and the same test cases or work items. You don't have to set up a new document each time.

Re-using a document

You can re-use an existing document[3] or any revision of an existing document and change its content however you need to.

To determine the number of a revision of a document to re-use, check the original document's history. You can re-use documents in the same project, in the same space or a different space, or in a different project in any space in that project. You can either create a new, standalone copy of a re-used document, or a "derived" copy in which the text/image content can only be modified from the base document, but which allows changes to some work item fields in the copy – "status," for example.

You can re-use multiple documents in a single operation, optionally keeping relative work item links between a group of documents that are re-used at once. For example, if you have a specification document containing requirements and a document containing a set of test cases that cover the functional requirements, you can re-use both the requirements document and the document with test cases, and have test cases from the re-used test case document linked to the functional requirements in the re-used requirement document.

5. Get reports for your testers and manager that show results of your testing progress

You may need a list of all test cases that should be tested and their results, along with a report of which test cases were executed, passed or failed. Furthermore it should be possible to use the test cases more than one time in different test runs. Maybe you want to define test run templates that can be re-used.

If your test managers want to have reports on your QA activities, the reports should be generated automatically and without much effort.

Creating test runs

A test run is a special system object created from a test run template, initially provided by Polarion QA and customizable by you. A test run represents an instance of the running of a set of tests. It defines the set of test cases executed, specifies some information about the test environment and build version tested, displays information about the results of the tests' execution and enables a tester to execute the tests specified in the test run.
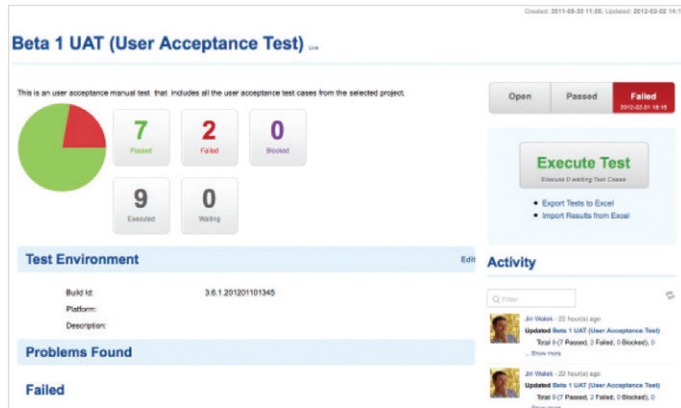
Test reports.

If you're a project lead, project manager or IT executive, Polarion's Live Dashboards provide an up-to-date information summary that enables you to see how things are going with any project. Dashboards present near or actual real-time aggregate information based on the actual progress of planned requirements, tasks, change requests, defect fixes, etc. Of course, reviewing dashboards is no substitute for close contact with your team, but project dashboards and the features that compile the information displayed can reduce the amount of time you and your team spend in meetings reporting on progress, so you can focus on problems and solutions instead. Dashboards can also eliminate the need for developers to write up information, and for managers to review and compile information from periodic status reports.

Polarion also provides managers with the ability to generate a variety of reports with output to both online and offline formats.
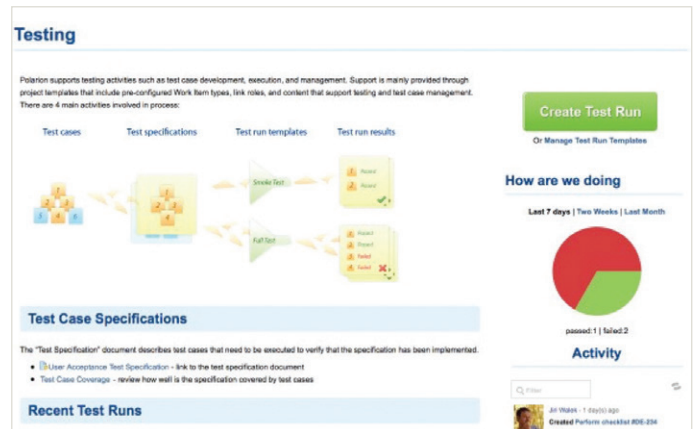
Testing.

6. Use your existing test cases

You probably already have a set of test cases that you want to start managing in Polarion, perhaps a list of test cases that are stored in an Excel file. This is a common scenario, and one that Polarion QA makes quite easy to handle.

Importing from Excel

Microsoft Excel workbooks containing artifacts such as requirements, test cases, defects, etc. can be imported to Polarion QA to create tracked and managed work items of the respective type. During import you have the option to create work items in a document, or as items in the project's tracker. After import into a document, you have the option to use the Word Round-trip features to collaborate with external stake-holders who use Microsoft Word. After import to the tracker, you have the option to use Excel Round-trip to collaborate with external stakeholders who use Excel.

Import Excel.

7. Work in your familiar environment

If you need to execute tests offline because it isn't possible to be online every time you or someone else executes them, Polarion QA provides a solution for this scenario.

Excel Round-trip

When you export any set of work items using the "xlsx: Microsoft Excel" option, the output is written to a special "round-trip" format for Microsoft Excel. An external user may optionally edit work item content in Excel and/or create new work items for import into Polarion. A Polarion QA user with the necessary permissions can then import the edited Excel workbook back to Polarion QA, updating the work item content from the externally edited Excel worksheet, and/or creating new work items defined externally in the exported Excel workbook.

If the work items are exported from a document with the necessary export options, it is also possible for an external user to change the document's structure externally in Excel. The structural changes are reflected in the online document when the Excel worksheet changes are re-imported to Polarion QA.

Excel export may be run from any scope: repository, project group or project. Re-importing of an Excel workbook that contains modified work items can also be done in any scope. However, re-importing of Excel workbooks that contain new work items is only allowed in the project scope.

Exported work item fields that are writable in the portal may be optionally locked in the "Export Work Items" dialog during the export procedure so that users of the exported Excel file cannot easily change the field value. If an advanced Excel user does manage to change the value of a locked field in Excel, the change is ignored when the Excel file is imported back into Polarion QA. (Note that fields which are read-only in the portal display the "locked" icon in the dialog, and are not "unlocked" when clicked.)
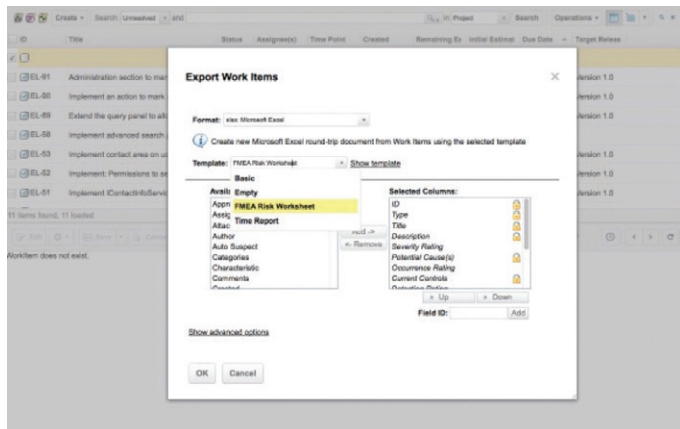
8. Integration with existing test automation tools

You may have specialized automated testing tools, and you may want to import results of testing with these tools into Polarion QA, preferably with links to requirements and other test cases. You will find that Polarion QA handles this need quite easily.

Automated testing

Simply configure any tool capable of exporting test results to the open xUnit format to write its results file to a specific location on your network, and specify that location in Polarion QA. Every time a new results file is written, Polarion QA automatically creates a new test run artifact, which logs that instance of testing. The test run definition (based on a user-definable template) determines which test cases were run, and of course you will have linked them to relevant requirements, so you have complete traceability.

The outcome of the executed testing is immediately and clearly visible in Polarion QA, showing overall status (passed/failed), and if failed, showing which test cases succeeded, which failed, and which were blocked. Failed tests result in automatic creation of a defect item, auto-linked to the failed test case for traceability, and auto-assigned to the appropriate developer(s), who receive an email notification.[4]

With Polarion QA, you leverage your existing investment in test automation and gain improved visibility and transparency with timely information, plus complete, automatic history of your testing activity.

Import Excel.

References

1. For example, Polarion QA was named Best Quality Tool at the Software Quality Days conference in Vienna, Austria, January 2012.
2. The change is also reflected immediately in the Activity Stream, which item owners can see when logged in to the portal.
3. Polarion's LiveDoc documents provide the familiarity and ease of use of office documents, but contain trackable artifacts such as test cases that can be managed with project workflow.
4. Beyond a user-definable failure threshold, a single summary defect is created and assigned, rather than individual defects for each failed test.

Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower,
Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software
Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over 15 million licensed seats and more than 140,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

For more information about Polarion, please visit www.siemens.com/polarion

www.siemens.com/plm