

Oct 2008 - Pragmatic Software Newsletters

Automating Your Regression Test Cases

Companies that develop and maintain software can dramatically improve the quality of their software releases by creating regression test cases that ensure that existing features are not broken with new releases. This newsletter discusses:

- How to create regression test cases
- When to automate regression test cases
- Best practices for automation analysis

Creating Regression Test Cases

Once a software product has been released to production, each new release of the software could cause existing features to fail. To prevent this, it is wise to create a set of regression test cases that are run with each new release. Below are some best practices when developing a regression suite:

1. **Categorize by Functional Area** – Your software product most likely has different sets of functional areas (e.g. Invoicing, Billing, etc). When creating regression test cases, categorize them by functional area so that you can ensure you have good test coverage for each functional area of your software.
2. **Regression Test Case Design** – Regression test cases do not normally need to test bounds, invalid data entry, etc – normally they will be designed to test the software the way it is designed to work. The reason for this is that when the feature was originally designed, it should have been thoroughly tested for bounds, invalid data, etc. An exception to this is if you find that new releases tend to break existing features from a validation perspective. If this is the case, keep some specialized regression test cases to ensure that the validations are still in place.
3. **Revisit the Regression Set with each New Release** – Upon implementing a new release of your software, it is wise to recognize new features shipped with the new release and to create a new set of regression test cases that test the new features. If you do not revisit your regression test cases with each new release of your software, the regression test cases will become stale and out dated.

When to Automate Regression Test Cases

Many companies run their regression test cases manually, so when does it make sense to begin automating your regression test cases? It makes sense to automate your test cases when you can no longer run the regression test cases on each build created. For example, if you are doing daily or weekly builds of your code to quality assurance and you cannot quickly run your regression test cases with each build, it is time to consider automating them.

To automate test cases, you must purchase an automated testing tool. There are many great tools on the market, including Automated QA Test Complete (<http://www.TestComplete.com>), HP Quick Test Pro, HP Win Runner, Rational Robot and Rational Functional Test, just to name a few. We normally recommend Automated QA Test Complete, as it is competitively priced and has similar features as the others. Once you have purchased an automated tool, you can use the tool to create your regression test cases.

How does Automation Work?

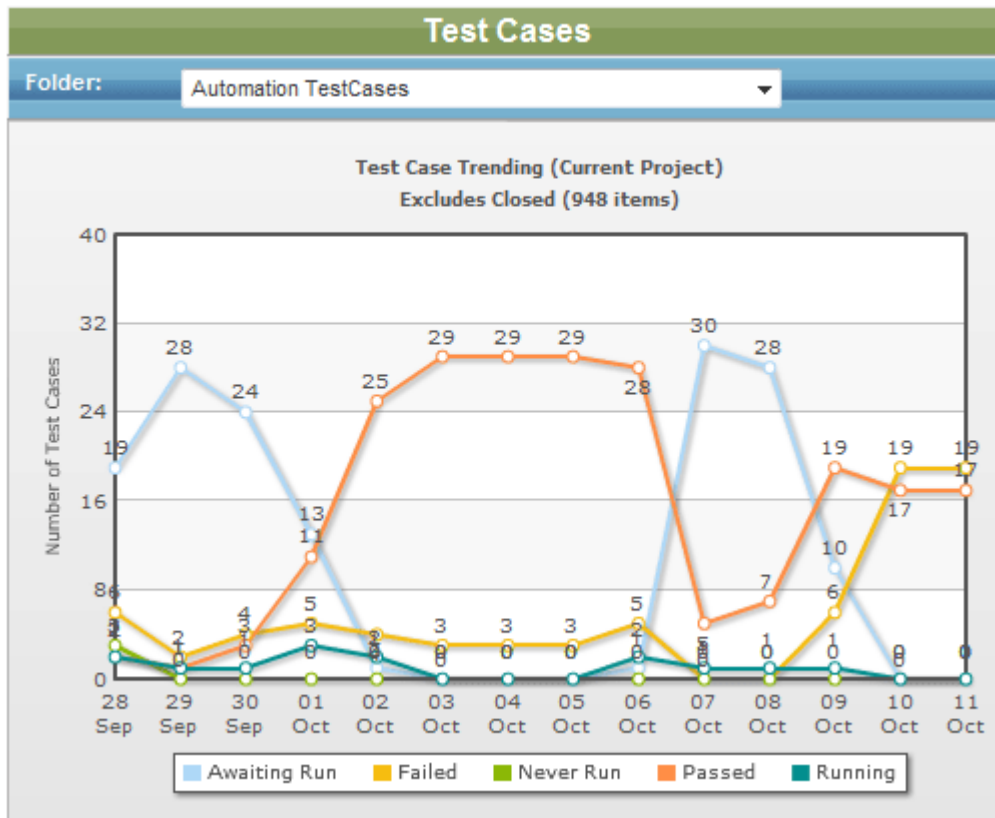
Each test case becomes a script. Many tools have record and playback features where you can turn the recorder on, open your software and perform the actions for a test scenario, then save the recording. This is a great way to learn the scripting engine, but it is not usually adequate to create well designed automated test scripts. Normally, you will want to have a technically minded software quality engineer in your organization that creates and maintains the automated scripts, as using these tools require knowledge of the tool, programming skills and great trouble shooting skills.

If you are initially creating your automation strategy, it is wise to consult with an automation expert to ensure best practices for your automation design. There are many companies that specialize in this; we have worked extensively with **Star-QA** (<http://www.star-qa.com>) with great results. Contracting with an automation expert can save effort and costs in the long term, as they will normally work with you to provide an automation framework that will be reusable and can provide training to your software quality engineer(s), allowing them to make great strides with their automation skill set in very little time. Another advantage of working with an automation expert is that they can implement "keyword driven automation". This simply means that they can create a set of re-usable automation scripts that can be invoked by name (Login, AddOrder, AddtoCart, etc), allowing less technical team members to create new sets of automated scripts.

Best Practices for Automation Analysis

Once your regression test cases are automated, they should be automatically run upon new builds of your software. If you can do daily builds of your software into your quality assurance environment, this is ideal. Once the automation is running daily, you will need a way to quickly determine how many automation test cases were run, how many passed and how many failed. For failed tests, you will want to drill into the detailed logs to determine what caused the failure.

Software Planner (<http://www.SoftwarePlanner.com>) is an ALM tool that can manage this process. Software Planner integrates with all the major automated testing tools including Automated QA Test Complete, HP Quick Test Pro, HP Win Runner, Rational Robot, and Rational Functional Test. By integrating automated testing into Software Planner, you can launch the tests from within Software Planner, create test sets, analyze the results (which tests passed or failed), and automatically send emails upon test completion. You can also trend these results using graphical dashboards. Below is an example of a dashboard that shows trending of your automation runs:



As you can see from the graph above, the past 2 days (Oct 10/11) has introduced a problem in the code because 19 automated test cases failed while 17 passed. Looking at the graph, you can see that the the issue was introduced on Oct 9, as all test cases passed from Oct 2 - Oct 6. This type of information is invaluable for quality assurance teams.

If you would like to see how Software Planner integrates with the major automated testing tools, watch a narrated movie at <http://www.pragmaticsoftware.com/guidedtours/automatedtesting.htm..>

Summary

As we have seen, automating your regression test cases can be valuable. You should see a return on investment within one release of your software after implementing an automation test strategy. This will be achieved by:

- **Quicker Releases** – By having your regression test cases run automatically, your software quality team can concentrate on testing new features of your software and less time regressing existing features.
- **Higher quality releases** – Your software releases will have fewer bugs and require less customer support because they will be of higher quality.
- **Happier Customers** – Your customers will be happier and more willing to serve as testimonials for future prospects.

Helpful Templates

Below are some helpful templates to aid you in developing software solutions on-time and on-budget:

- **Pragmatic Agile Development** - <http://www.pragmaticsw.com/PADOverview.pdf>
- **Software Development /QA Templates** - <http://www.pragmaticsw.com/Templates.asp>
- **Software Planner** - <http://www.SoftwarePlanner.com>
- **Agile Training** - <http://www.PragmaticSW.com/Services.asp>

About the Author

Steve Miller is the President of Pragmatic Software (<http://www.PragmaticSW.com>). With over 23 years of experience, Steve has extensive knowledge in project management, software architecture and test design. Steve publishes a monthly newsletter for companies that design and develop software. You can read other newsletters at <http://www.PragmaticSW.com/Newsletters.asp>.