DeCare℠
Systems Ireland, Ltd

CPM Toolkit

Cutting Application Development Costs Using DSI's CPM Toolkit

December 18, 2008

Prepared by:

Brendan Lawlor
Process Architect
DeCare Systems Ireland

Tel +353 21 4925 158
blawlor@decaresystems.ie

TABLE OF CONTENTS

PROPRRIETARY RIGHTS NOTICE

EXECUTIVE SUMMARY

Continuous Integration (CI) is a proven cost-cutter. Applying CI to performance testing promises an even greater percentage return, due to the nature of performance issues. But those savings are only realised if we apply the idea in a way that is informed by our shop floor experience of CI in particular and process in general.

Where Does a Software Business Lose Money?

Two big culprits are project overruns and loss of return business from unhappy customers. When it comes to happy customers, performance is a factor that punches above its weight. The thing about performance is that it is one of the first aspects that a user perceives in your system, and it elicits a strong emotional reaction: If your system is particularly fast, your customer will tend to look more kindly on other issues. If your system performs poorly, all the bells and whistles in the world won't save you.

The causes of project overrun are many and varied, but one of the old reliables is delays in QA, fixing bugs that should have been found during development. Bugs found late in the development cycle are particularly expensive to fix because their symptoms appear a long way from their causes. We spend much more time diagnosing than curing. It is this realisation that has led to the widespread adoption of Continuous Integration and Unit Testing in the software development community. CI and Unit Testing, with their emphasis on identifying bugs early and close to their source, have proved their value to the corporate bottom line. The argument for applying the same approach to performance testing is even stronger. Performance related bugs typically have multiple and complex causes. The traditional approach to tracking them down - firing up a tool like Quest Software's JProbe® and applying it over the entire codebase - represents a high overhead. We can say that the time spent finding performance bugs is typically an order of magnitude greater than time spent fixing them.

These are matters where consensus on cost savings can be arrived at quickly: Performance is a vital factor in keeping our customers happy. And Continuous Integration, with its philosophy of finding problems when they are cheapest to solve, has proved itself over the years. It can be readily seen that the two should be combined to produce even greater savings.

What is harder to see however, is HOW they should be combined. While the principles are simple, if we apply them incorrectly we stand to make things worse rather than better. The experience-tempered application of these principles is what the CPM Toolkit is all about.

CPM Toolkit Does the Right Thing the Right Way

There are a number of ways in which the CPM Toolkit tool avoids the pitfalls of a more naive approach to continuous performance testing.

1. *CPM Toolkit Reuses Unit Tests*

CPM Toolkit applies its performance analysis to running unit tests and there are two major cost savings to this approach:

a. No new costs - the performance test piggyback on existing unit tests.
b. Further reduced diagnosis time - problems are pinpointed to small sections of code from the beginning.

2. *CPM Toolkit Analyses Trends*

One of the real difficulties with performance testing is that it is hard to set a threshold value between pass and fail. Many software quality metrics suffer from this problem - they yield very precise results but with almost no guide to what a good target value should be. CPM Toolkit flags changes in trends over multiple builds, rather than enforcing arbitrary thresholds. When we couple performance measurement with the trend analysis that CI provides, we get something new.

3. *Navigable Data Presentation*

Performance testing will not by itself reduce costs. In fact, it represents yet another data stream to be processed by developers and Project Managers, What is important about CPM Toolkit is that it presents information in the correct way: simple at first, but allowing navigation to the details. Top level data can alert us to a potential problem, and easy navigation can quickly confirm, as well as indicate a potential solution.

How Much Time Will CPM Toolkit Save You?

Your reputation is not a number, and the damage you can do to it by delivering under-performing systems cannot be measured. But we can begin to put some rough estimates of project overruns caused by catching performance problems too late.

Standard Calculation of CI Savings

This simple calculation of time saved by using CI (not specific to performance CI with CPM Toolkit) makes the following assumptions:

1. If a bug can be found and fixed in 1 hour during construction-time unit testing, the relative cost of finding and fixing it later is

- 250% more during development system testing
- 400% more during QA
- 600% more during Production

(These take into account response time, communication overhead, diagnosis of problem, and knock-on changes forced on other code).

2. Of bugs not found during construction

- 45% are found in Development System Testing
- 45% are found in QA
- 10% are found in Production

This gives the following result:

1 hr in Dev = (0.45 * 2.5) + (0.45 * 4) + (0.1 * 6) = 3.5 hrs *after Dev*

or stated another way:

Every person-day spent fixing bugs caught by Continuous Integration represents a saving of 2.5 person days.

This is a very conservative value and would be much higher again for performance-specific bugs.

Other Savings

The benefits of the tool should not only be seen in terms of delays avoided, but also in terms of increasing the efficiency of your existing development team. CPM Toolkit acts like another pair of eyes on the code, flagging problems as they appear and freeing your development teams to spend more time on more directly productive activity.

Cost of Using CPM Toolkit

As already mentioned above, CPM Toolkit is smart about the way it implements CI for Performance, and reduces costs to a minimum.

Because CPM Toolkit reuses existing unit testing infrastructure, there is just a once-off setup cost of CPM Toolkit that can be spread over all projects that use it, so that it becomes negligible compared to those projects' overall effort.

The cost of assimilating the new information is also negligible - it is simply another aspect of CI's reporting. It only attracts attention and effort if a potential problem is found, i.e. if the performance trend of the results deviates significantly.

The navigable nature of the results make it easy to find the test case code which deviates in performance.

About CPM Toolkit

The CPM Toolkit introduces Continuous Performance Management into a Continuous Integration environment. CPM Toolkit automates the collection of performance, memory usage and code coverage data to generate performance benchmarks, while providing project management level visibility across development projects.

- Combine the power of JProbe with a *Continuous Integration (CI)* environment to incorporate cost effective performance management early in the development lifecycle.
- Establish *performance benchmarks* and easily track performance deviations from build to build.
- Prioritise performance monitoring across the entire *development team* by extending the results of JProbes' Analysis Engines to the project team.
- Analyse performance behaviour *throughout the development phase* to establish normal patterns of behaviour and build a performance profile of your application.
- Implement a *"find early, fix early"* practice to prevent expensive and time consuming performance tuning in production.
- Provide a greater level of *confidence* prior to moving code into system test, and subsequently user acceptance testing by removing performance issues at the development stage.
- Gain *project management level visibility* across all projects under development using the CPM-at-a-glance dashboard.

About DeCare Systems Ireland

DeCare Systems Ireland (DSI) was established in 1998 and is a subsidiary of DeCare Dental LLC, one of the largest dental benefit management companies in the United States. DSI is an enterprise software development company specialising in architecting, developing and integrating custom .Net and Java applications. With over 140 software technology professionals on staff, DSI provides unique, customer-specific, cost-efficient solutions for clients including Amazon, Avon, Expedia and a number of large US healthcare insurance carriers. Serving both private and public sector organisations, DSI focuses on Architecture Blueprinting, eSolutions, Bespoke Application Development and Application Performance Management.

DeCare Systems Ireland Ltd
Building 1
University Technology Centre
Curraheen Road
Cork
Ireland

Phone: +353 21 4925 100 | Email: info@decaresystems.com | Web: http://www.decaresystems.com | Blog: http://blog.decaresystems.com