White Paper

CresTech

An Agile Approach to the Bottle Neck Game of Performance Engineering

Confluence

24 October 2013

To overcome the problems associated with the traditional approach, we have come up with the agile methodology for executing the performance engineering process. In this case, the basic principles of Agile are used to get a quick ROI and maximum benefits. Agile performance engineering has major advantages over other rival options such as the Waterfall approach to performance engineering.

Author – Ravinder Singroha

ravinder.singroha@crestechglobal.com

Abstract

In order to sustain growth momentum, draw new customers and retain existing ones, 21st century organizations need robust IT infrastructure, and software applications which are available 24x7x365 and can drive business. Such high reliability, availability, performance, and optimum response times can only become a reality when software products and applications are thoroughly tested before use.

Traditionally, performance engineering was conducted at the end of the development lifecycle just before the product went live, as companies were not keen to invest in the environment or tools for performance engineering. What organizations didn't realize was the fact that discovering problems—particularly design and architectural flaws—in the software, at the end of the product cycle, would lead to higher costs in fixing these problems.

To overcome the problems associated with the traditional approach, we have come up with the agile methodology for executing the performance engineering process. In this case, the basic principles of Agile are used to get a quick ROI and maximum benefits. Agile performance engineering has major advantages over other rival options such as the Waterfall approach to performance engineering.

Business Case

Some of the statistics revealed by the study are mind-boggling in their demonstration of impatience. For example, one in four people abandon surfing to a website if its page takes longer than four seconds to load. That's just four "Mississippi's," guys. Four in 10 Americans give up accessing a mobile shopping site that won't load in just three seconds (which is roughly the time taken to read to the period at the end of this sentence). Given that shopping sites tend to have to be image-centric, and thus may take longer to load.

The greater majority of users also won't wait in line (unless they have to) for more than 15 minutes. Fifty percent wouldn't go back again to an establishment that kept them waiting for something. So you'd better serve them swiftly the first time if you want their repeat commerce, no matter what Groupon deal you can cook up. Surprising as all this may be, the implications of this impatience are even more socking. Amazon's calculated that a page load slowdown of just one second could cost it $1.6 billion in sales each year. Google has calculated that by slowing its search results by just four tenths of a second they could lose 8 million searches per day--meaning they'd serve up many millions fewer online adverts.

The growing complexity and sophistication of software is also creating applications performance problems for companies. Forrester Research studies show that around 85 percent of organizations with revenues in excess of USD 1 billion reported glitches in applications performance. Performance engineering is therefore the need of the hour

Solution

How we can identify and monitor bottlenecks in software developments projects in order to prevent inefficiency?

"Organizations are deploying the Agile performance engineering approach to ensure zero performance defects, always available, and deliver high quality software products and eventually which will leads to around zero performance problem and around zero business loss due to performance issues."

The Agile approach is an iterative and collaborative process, where performance engineers work alongside the development teams to identify the bugs and fix them faster. The approach recommended by Impetus has multiple phases that are integrated into the SDLC.

The proactive Agile approach can be used both when creating new applications and upgrading existing ones. The advantage of Agile-based performance engineering is its ability to meet product performance goals faster and more efficiently. By identifying the bugs in the software earlier in the development cycle, the Agile approach helps companies save costs and bring zero defect, high quality products quicker to market.

Using the Agile approach organizations can efficiently undertake the following activities during software development:

- Capture their non-functional software requirements
- Develop performance strategy
- Identify and implement the processes to ensure performance
- Identify and isolate performance bottlenecks
- Fix the application and system performance issues
- Release the product early without any doubt about its performance in the production environment.

The Agile approach is emerging as the preferred route to performance engineering by organizations keen on minimizing the risk of product failures and succeeding in the marketplace. Performance Engineering is an important activity for projects executed using any methodology. It becomes more critical for projects using SCRUM or any other agile methodology.

This section will suggest various performance engineering activities that should be planned for and executed in the stages of a scrum lifecycle.

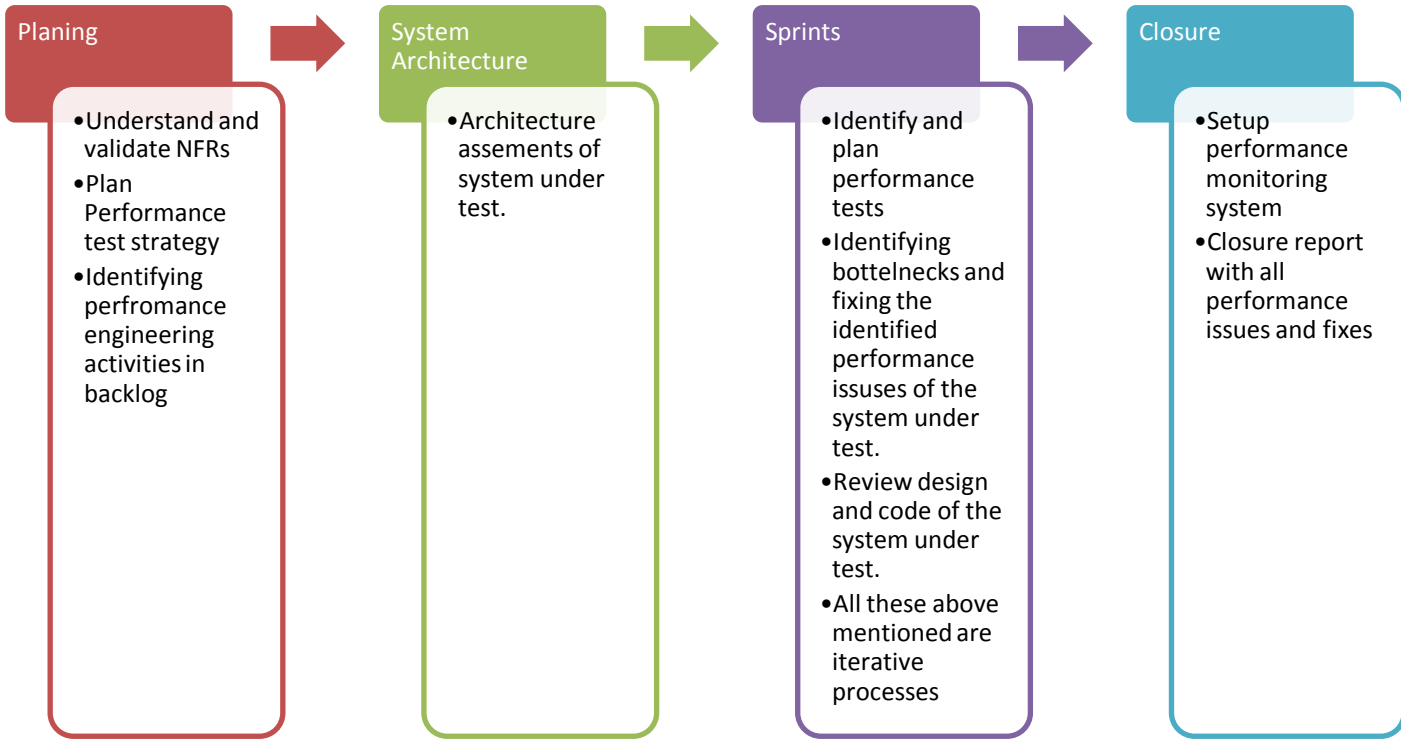

Figure 1: Performance engineering in SCRUM

Planning Stage

This phase involves understanding the requirements, and planning for the 'sprints' ahead. The various performance engineering activities that should be planned during this phase are listed and explained below

Understand and validate Non Functional Requirements

This area everyone would agree is important and critical, but is the one often missed or partly done due to various constraints. Efforts spent here will help the team to validate the performance requirements and be realistic about it.

Performance Test Strategy

It is important to define the test strategy, which will include details like

- Scope of Performance Testing
- Infrastructure requirements
- Methodology used for testing
- Workload characteristics used for testing
- What are the tool requirements and licensing cost?
- What are the metrics that are captured?
- How will the results be shared?

Identify Performance engineering activities in product backlog

While we are identifying feature lists for the product backlog, it is also important to identify the various performance engineering activities like workload modeling, performance testing, and performance assessment etc. Many a times these activities are inherently assumed and not planned, resulting in insufficient planning and poor execution.

System Architecture Phase

Defining architecture for the system determines the success of the code to meet the requirements.
While there has been a lot of focus on validating the architecture for the various functional and business requirements, it is sad that reviewing architecture for NFR's is not planned as a matter of priority

Architecture Assessment – The Performance viewpoint

Every software development project should undergo an architecture assessment. It is more important in an agile methodology, as the inherent nature of the model is dynamic and having an architecture that meets the requirements will enhance the success of the project.
During the architecture definition stage, most often than not, we perform a qualitative assessment of the architecture. Such assessments help understand how the various quality of service requirements like performance are addressed in the architecture. Some of the well known assessment methodologies are

- Architecture Tradeoff Analysis Method
- Cost Benefit Analysis Method
- Active Reviews for Intermediate Design

Architecture reviews conducted at this stage will provide us with areas of architecture that have to be modified / updated to achieve better performance. It should be ensured that the architecture is updated and bought to an agreeable shape before the sprints could begin.
The above step is critical as any significant changes to the architecture during the sprints will adversely affect the progress.

Sprints

Sprints are the building blocks of any scrum methodology. Shippable, production quality software are developed, tested and deployed during each sprint. A strict control on Quality of Service parameters during the sprints will ensure that the application meets the Quality of Service requirements.

Review Design and Code

There should be a constant endeavor to review the design and code during the sprints for existing and potential performance issues. Automating code review using tools like FxCop, which has rule sets to identify performance issues can be used to expedite the process.
With this iterative and consistent approach to identifying and fixing performance issues, there is a continuous improvement to the quality of code.

Identify and Automate Performance Tests

The team identifies and closes on the list of features that will be implemented during each sprint.
During this process, we should also plan for conducting the performance test for the features that have been implemented.
As a best practice, we should identify the list of features for performance testing a sprint ahead. This will aid the team in automating the performance tests that can be used in the subsequent sprint.
Automation of tests is mandatory as manual effort will lead to delays and errors, which will prove costly in a time-boxed sprint.

Identify bottlenecks and fix performance issues

Every performance test is executed with the focus of identifying bottlenecks in the application. Fixingthe bottlenecks should be done during the course of each sprint, so as not to pile up the issues and buggy code which leads to performance issues.
Performance issues discovered during a sprint should be planned as features that should be implemented during subsequent sprints.

Performance Engineering Sprint

Initial sprints will witness an evolution of architecture and design. There will be changes to the functionality and refactoring to the application code. As the sprints become more mature, which are characterizes by their longer duration, we should consider on planning a parallel sprint for performance engineering activities.
We can also consider a parallel performance engineering sprint for every 2-3 normal sprints, so as to have a considerable functionality implemented for performance testing and assessment. By being a parallel, independent sprint, a performance engineering sprint does not affect the progress of the application development. Also, it provides an isolated environment for the performance engineering team to analyze the application performance. Activities like performance testing, performance assessment, capacity projection can be carried out in this sprint.

Closure

This phase involves deploying the complete application in the production environment, and obtaining sign off's from the stakeholders.

Setup Performance Monitoring Systems
The team should plan on using systems and tools to monitor and report performance issues, so that they can be resolved quickly.

Challenges in Executing Performance Engineering in SCRUM

Implementing performance engineering processes in scrum has its own set of challenges as follows:

- Engaging the customer – Being qualitative rather than quantitative
- Automate tests – Manual is not on
- Mindset – The toughest nut!

It has become the need of the hour in the current IT industry to apply and accept performance engineering as an integral part of any agile software delivery process. Timely identification of performance bottle neck issues can entirely change the game and help score the development party the winning streak. Investment in license procurement of tools which help automate the performance testing pays in the long run. It's time now to change our mindsets and aim towards more qualitative delivery approaches of the services and products to the stakeholders. Performance engineering is to be adapted in each part of the SCRUM for a healthy and an efficient delivery of the software solution.

Appendix

ROI : Return on investment

Agile: Agile methodology is an alternative to traditional project management, typically used in software development. It helps teams respond to unpredictability through incremental, iterative work cadences, known as sprints. Agile methodologies are an alternative to waterfall, or traditional sequential development.
Scrum: Scrum is an iterative and incremental agile software development framework for managing software projects and product or application development. Its focus is on "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" as opposed to a "traditional, sequential approach". Scrum enables the

creation of self-organizing teams by encouraging co-location of all team members, and verbal communication between all team members and disciplines in the project.

Sprint: A sprint is the basic unit of development in Scrum. The sprint is a "timeboxed" effort; that is, it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one week and one month, although two weeks is typical.

NFR: Non functional requirements

References

Agile Software Development - http://en.wikipedia.org/wiki/Agile_software_development
Performance Engineering - http://en.wikipedia.org/wiki/Performance_engineering
SCRUM Development Process – Ken Schwaber
Realizing Continuous Performance - Management Best Practices Documentation – Steve Haines
http://www.infosys.com/infosys-labs/innovation-co-creation/Pages/index.aspx
http://en.wikipedia.org/wiki/Scrum_(software_development)
http://www.smartcompany.com.au
http://www.fastcompany.com/1825005

About Author:

Ravinder Singroha works as a Lead Analyst in the PSG (Performance and Security) group of Crestech Software System Pvt. Ltd.(www.crestechglobal.com). He has experience of more than 5 years in performance engineering and software development complete life cycles of different domains like telecom, finance etc. He lives in Delhi NCR (India). In his other life he is an avid cricketer and has lately discovered the joys of fatherhood.