WIPRO
Applying Thought

USE AUTOMATION TO DELIVER BETTER SOFTWARE QUALITY

"Application's not responding again! Now, we are only ten tech-support calls away from another free pizza!"

Aman Chandra, Head of Tools Group, Wipro
Kavitha Sridhar, Senior Practice Manager

Table of Contents

Abstract

In today's world, where IT drives business, an organization's reputation can take a beating if its software system is flawed. Managing modern enterprise applications, which are implemented by integrating a set of custom/industry standard solutions and platforms, is quite a task in itself. Discovering software flaws across these is no less a challenge either. This paper helps you understand the structural quality of your applications that would enable you to identify system glitches using an automated and holistic framework.

Introduction

Let's see if you recognize this situation: the HR application you are using slows down, becomes unresponsive, and you think, "Our network is becoming worse by the day. It's time IT upgraded the network or added bandwidth or found some solution." But ask the network administrator and he tells you: the application just barreled a huge volume of data down the network when all that the receiving server required to process the task was 20% of the data.

What should be fixed then? Network bandwidth or the amount of data the application sends? We think the structural quality of the application needs urgent improvement. The question is: How do you know if your application performance is below par and needs improvement?

Scratch under the surface of the HR application we were just talking about, and you are likely to find it has structural issues. Structural quality is a great measure of application health.

An HR application glitch may cause some user dissatisfaction but may not blot-out bottom-lines. But an application deployed to manage B2C situations, like the Black Friday shopping frenzy, may do so. Retailers cannot afford to let anything go wrong with their applications at that time because every transaction pumps up the bottom-line.

IT is getting complex

Today's applications are complex. There are multiple components that make up an organization's IT: multiple technologies and devices, legacy components married to latest technology platforms that are married to a variety of databases and networks[1]. These applications run in different environments (cloud and data center) accessed from multiple devices, making it almost impossible to identify the root cause of a problem. It also makes it difficult to cover all possible states, paths and data values during testing.

According to a study by research firm Aberdeen, application performance can impact up to 9% of corporate revenue[2]. Naturally, businesses should be concerned about application performance.

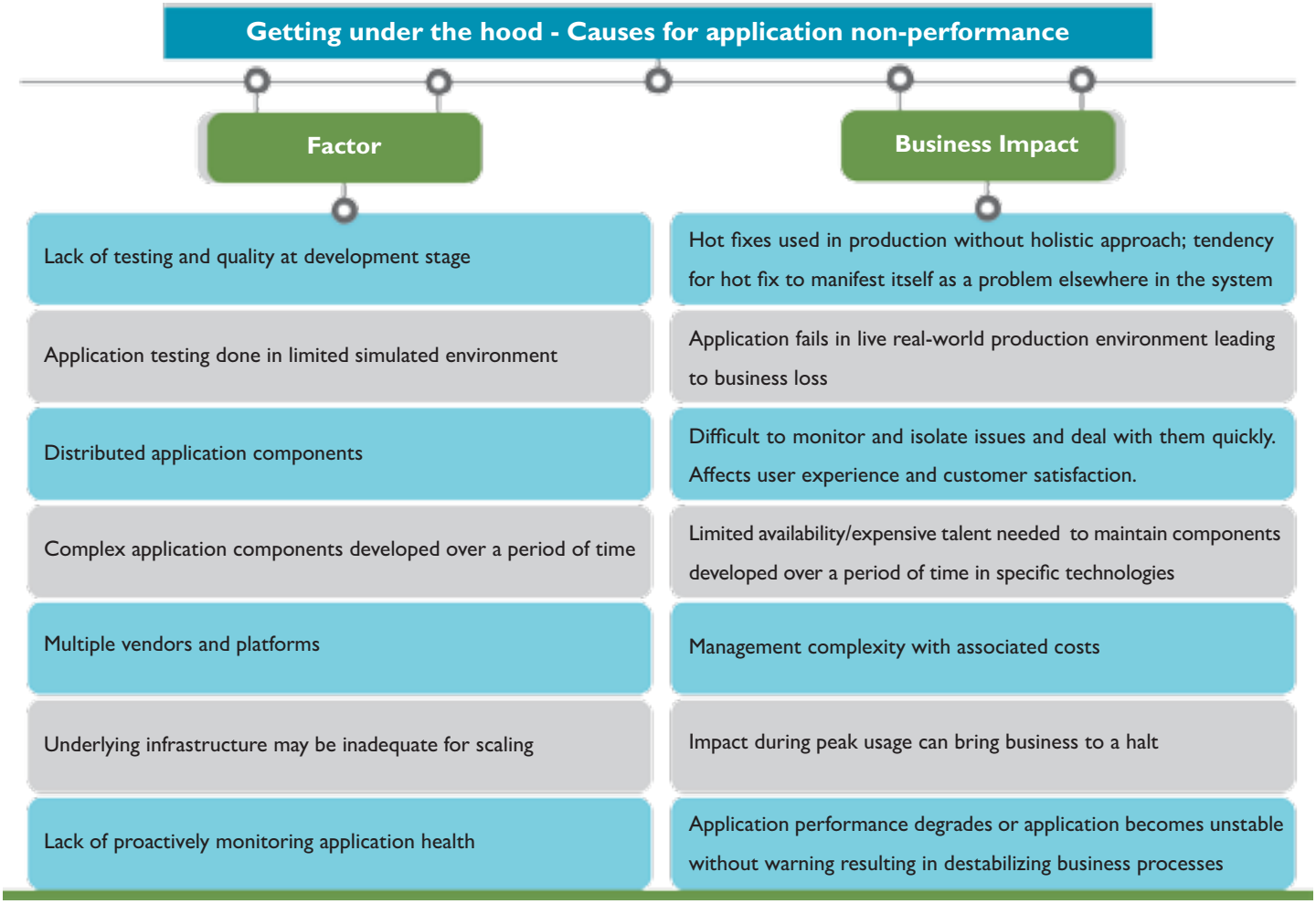


Figure 1- CAUSES FOR APPLICATION NON-PERFORMANCE

So, what can be the possible causes for your applications' non-performance (see Figure-1). Complexities further grow as businesses deal with application exigencies in a piecemeal manner. This results in technical debt as there comes a time where organizations do not have even a single person left to understand the underlying complexities of the system because it was developed, upgraded, managed and maintained by several hundred different hands. These result in losses at critical moments – when the business grows, new offerings made, during heavy work-load or when usage spikes.

[1] We are reminded of Glass's Law from his book *Facts and Fallacies of Software Engineering* which states that for every 25% increase in functionality, there is a 100% increase in terms of complexity in that system.

[2] The Performance of Web Applications: Customers are Won or Lost in One Second. Aberdeen: Reference number 5136.

Measure application complexities

Application performance is dependent on several factors such as the hardware, network, architecture, database, programming practices, sophistication of integration and the BI layer. Default configuration like the number of database connections; code deadlocks; resource utilization like network, CPU and memory; default password, misconfigurations, SQL injections; unawareness of resource bounds, blocking calls that result in application failure – just about anything can impact applications.

Another factor is technical debt. This refers to technical violations or the consequences of deviating from best practices for software architecture and programing, leading to inefficient usage of resources and sub-optimal processing of the data.

Technical debt needs a proactive as well as a reactive response. If the debt is not squared, it lingers. Technical debt is not easy to diagnose and the long-term impact is far reaching. For businesses, it often crops up as the future cost of maintaining an application.

How can we then simplify the application landscape?

Are your applications meeting the following criteria?

Application Performance helps meet varying usage patterns (e.g.: holiday demand, month-end spike in financial transactions, annual peaks in online tax filing etc.). This involves testing the application (after some proactive checks such as transaction/call flow analysis, profiling etc.) for forecast loads on critical business processes and transactions. It helps answer questions like, "Is the application response-time satisfactory and meets the expectations of the end user/customer?"

Application Security & Resilience ensures that disruptions don't result in loss and applications restart automatically at the last transaction (or in a worst case at a pre-determined restart point). From a user standpoint, there is only a delay in response-time. There is no loss of transactions or data. In addition, there is no requirement for a fresh sign-in.

Application Scalability refers to the ability of systems to manage a growing number of transactions. Most systems can scale provided the resources scale as well. However, you need answers to questions such as, "Does my application require resources to grow exponentially, linearly, or logarithmically?" These are important considerations, depending on the expected growth.

Figure 2 - HEALTH CARD TO MEASURE APPLICATIONS

The first step is to identify and define the health parameters of an application (see Figure 2), then set the acceptable performance thresholds based on the business environment. For example, what is the level of Application Performance a business requires and what is its current level?

The answer is Structural Quality in the form of a quantitative measure such as Structural Quality Index (SQI[3]) for the application. Once the SQI is articulated, the diagnosis and solution can prevent over reaction and over management. More such parameters identified in Figure 2 can be evaluated using a health-check software[4], thereby automating retrieval of technical quality insights of your applications.

[3] SQI is derived from the Software Quality Model that lists requirements like usage of resources, queries, etc., aligning to ISO 25010 that characterizes the software quality.

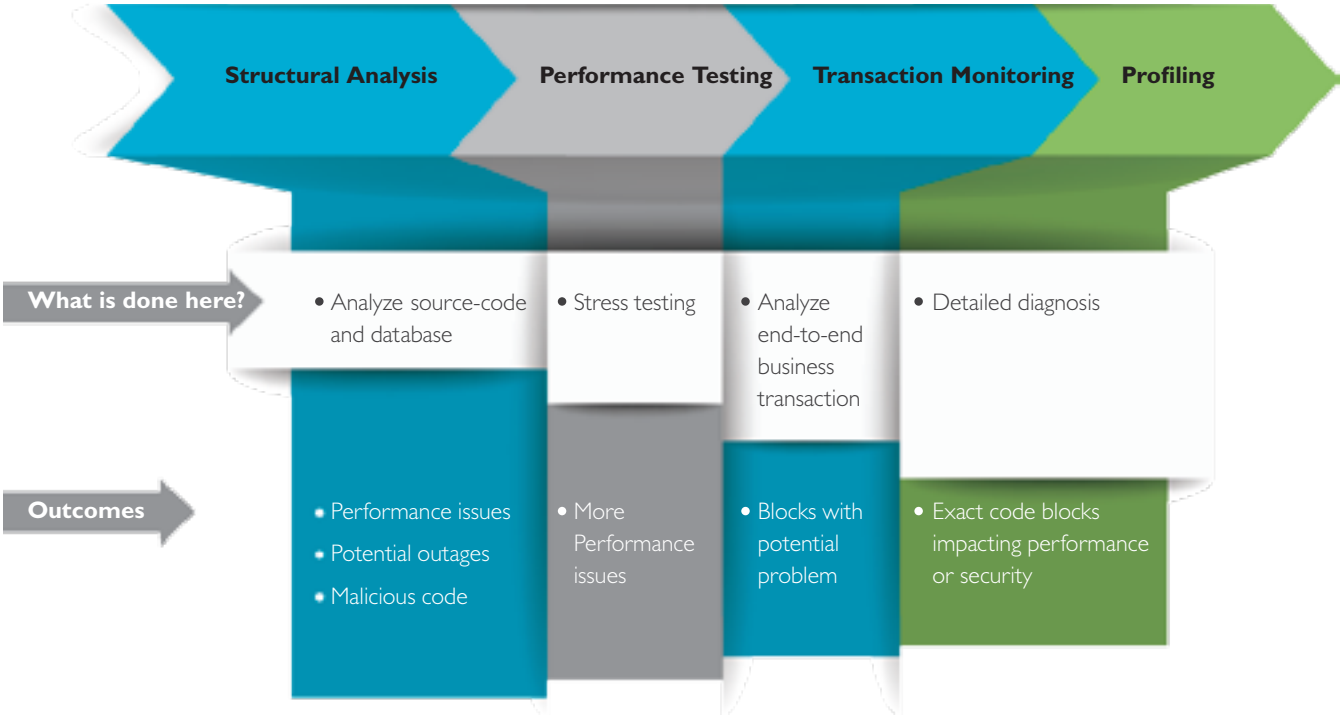
[4] Please refer to Wipro AppInsight for more details.

Performing a holistic analysis

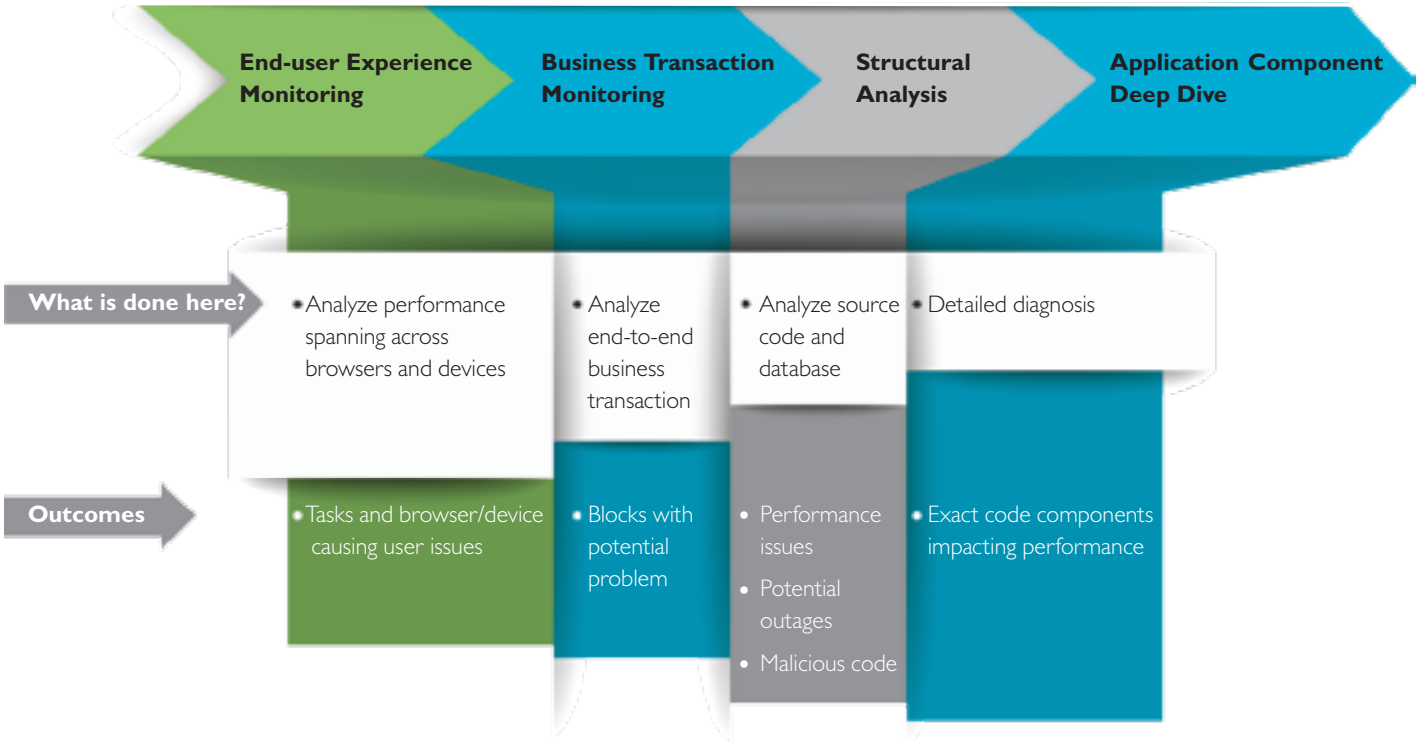
It is possible to get an understanding of your applications and follow it up with a holistic analysis after you get insights into quality issues. Thereafter, you can get the optimal fixes done based on business priorities.

Based on the stage the software is in -- development, pre-production, production or maintenance -- the analysis can be proactive or reactive.

Proactive approach



Reactive approach



Proactive Assessment (During development/pre-production)

Proactive Readiness Assessment looks at the application in development/pre-production environments. The goal is to use varying levels of proactive measures to identify code components and stacks that create bottlenecks and impede performance. As application life (or usage) increases, proactive assessment and maintenance help bring down the risk of disruptions or outages. The sequence of pro-active assessments are:

Structural analysis: At application level, it digs into the source code in development to identify patterns that indicate performance issues, outages and malicious code across technologies. The analysis should ideally identify, rank and prioritize code components and stacks that create bottlenecks and impede performance. In proactive readiness assessment, structural quality is addressed during the application development stage. This helps in early identification of issues around performance, security and robustness. The problems can be fixed at source. The cost of fixes at this point will be minimal.

Performance Testing: Live production environments are quite different from development and pre-production environments. In the absence of access to the live production environment, stress results using performance testing in pre-production environment is used to evaluate quality of an application.

Transaction Monitoring: When several components and platforms are brought together to deliver a single functionality, the level of complexity increases manifold. Business Transaction Monitoring looks at running applications in pre-production stage. It points to blocks - like hardware, application, database or network – that can lead to risks.

Profiling: Is used to perform detailed diagnosis of the components to locate code blocks impacting performance.

Reactive Assessment (During production/maintenance)

When the application is live and shows glitches or impacts other components negatively, the reactive solution is applied to minimize possible down-time. Here again, two of the proactive assessments can be used -

End-user Experience Monitoring: End-users can access the Web applications using various browsers (IE, Chrome, etc.) and from different devices. This monitoring done through software can point out the tasks and related browsers/device causing the performance-related issues that end-users face.

Business Transaction Monitoring: This monitoring is similar to the one used in proactive assessment, but here we need to monitor the production environment transactions instead and zero in on the problematic blocks, e.g. hardware, application, database or network that could lead to risks. Further diagnosis through structure analysis enables tracing of the components within the block that is impacting the performance.

Structural Analysis: On identification of blocks in application or database during business transaction monitoring, structural analysis is done to unearth the patterns in the code that leads to the blockage. E.g. Incorrect usage of index or queries could be an outcome of such an analysis.

Application Component Deep Dive: This is used to perform detailed diagnosis and pin-point the code components in the blocks impacting performance.

Conclusion

Organizations intensify testing efforts when critical applications crash or begin to show signs of malfunction. However, it is better to pre-empt such glitches and manage applications by using an automated framework that can help arrest drain on revenues.

The automation framework described in this paper enables reduction in the effort spent on downstream activities of application development like testing, review and rework activities. This helps reduce cost of application maintenance. The framework based on proactive and reactive application assessments can be used for even the most complex of IT application landscapes.

About the Authors

Aman Chandra has over 21 years of experience in IT industry and 7 years in various lead positions in areas of Delivery Excellence, Tools Consulting, Service Industrialization, Process Optimization and Competency Development. He has been heading the Tools Group at Wipro's Global Transformation Unit, since April 2011.The Tools Group provides solutions in Application Lifecycle Management (ALM), Software Quality Automation and DevOps space, enabling project teams improve overall productivity, velocity and service quality.
Aman can be reached at aman.chandra@wipro.com

Kavitha Sridhar handles AppInsight platform in Wipro. The AppInsight platform at the Tools Group leverages both commercial and home-grown tools to automate application structural quality measurement and analysis. Kavitha's responsibilities include need-realization, solution definition; tools-enabled assessment, interpretation of results and recommendations. She has more than 16 years of experience in software development, end-to-end software engineering and application quality assurance tools consulting.
Kavitha can be reached at kavitha.kandaswamy@wipro.com

This thought leadership artefact is from Global Transformation (GT) Office at Wipro. GT focusses on transformation initiatives for Wipro's key clients. GT has also been instrumental in evangelization of next-generation IT platforms at Wipro. Contact us gt.office@wipro to discuss and know more.

About Wipro Ltd.

Wipro Ltd. (NYSE:WIT) is a leading Information Technology, Consulting and Business Process Services company that delivers solutions to enable its clients do business better. Wipro delivers winning business outcomes through its deep industry experience and a 360 degree view of "Business through Technology" - helping clients create successful and adaptive businesses. A company recognized globally for its comprehensive portfolio of services, a practitioner's approach to delivering innovation, and an organization wide commitment to sustainability, Wipro has a workforce of over 140,000, serving clients in 175+ cities across 6 continents. For more information, please visit **www.wipro.com**

DO BUSINESS BETTER

CONSULTING | SYSTEM INTEGRATION | BUSINESS PROCESS SERVICES

WIPRO LIMITED, DODDAKANNELLI, SARJAPUR ROAD, BANGALORE - 560 035, INDIA. TEL : +91 (80) 2844 0011, FAX : +91 (80) 2844 0256, Email: info@wipro.com

North America Canada Brazil Mexico Argentina United Kingdom Germany France Switzerland Nordic Region Poland Austria Benelux Portugal Romania Africa Middle East India China Japan Philippines Singapore Malaysia South Korea Australia New Zealand

IND/BRD/MAR 2014 - BCA 2015