

bynx

software without boundaries

A Case Study





Would you dare test your source code?

A chance meeting on a plane bound for the Netherlands resulted in two men discussing software quality - specifically how a defined standard (over and above ISO 25010) is much needed.

“Your software should work for your business, not the other way around where your business is a slave to it.”

is how Mervyn Blank, UK sales director of source code testing company **Omnex** put it. He added: “that can happen when it’s unreliable because code is poorly structured, architected and contains a host of other low standard elements.”

The other gentleman was Mark Binks, group managing director of fleet and leasing management software provider **Bynx**. The company’s flagship product **bynxFLEET** manages in excess of 1 million vehicles worldwide and has been in use by customers for over 25 years. During that time, the product has undergone continuous refinement (as is standard practice).

Customers depend on **bynxFLEET’s** reliability so quality standards are engrained in the company’s culture. Stringent coding practices, quality control and testing regimes are commonplace. But Mark Binks is convinced the software development industry itself could and should go further.

Laying down the gauntlet

The conversation inevitably resulted in Blank asking Binks if he’d considered source code testing for **bynxFLEET**.

Bink’s first reaction was to dismiss the idea as unnecessary. As founder of the company, his guidance and direction has steered it from humble beginnings in South Africa in 1988 to the global success it is today, employing 80 people worldwide with customers in 22 markets.

Binks takes up the story: “Initially, I thought I was being ‘sold to’ but the more I listened as Mervyn explained the lesser-known and appreciated aspects of testing, the more interested I became. The fact that there could be issues in our software of which we’re unaware just because of some simple labelling discrepancies, for example, struck me. What was also revealing was what he spelled out about licensing problems associated with open source programming.

Some of these things I hadn’t even thought about. Mervyn made clear that finding holes in source code was only a small element of what **Omnex** testing does. I thought: this could actually help us validate the caliber of our product and design the next phase. I was confident our code would come through clean as we place a lot of emphasis on programming and risk remediation.

I concluded it was time to put my money where my mouth is. If it turned out my confidence was well placed, we have something we can talk about in sales and marketing. If misplaced, then great, we have guidance on what work we need to do to enrich **bynxFLEET** and get it ready for the next-generation.”



The Test

Omnext performed source code analysis, which is the automated testing of software code for the purpose of debugging.

Source code is comprised of statements created using a text editor or visual programming tool. These are then saved in a file and this creates the software program. The source code is the most permanent form of a program but the program itself can be modified, changed, improved and upgraded at any time.

The modules chosen for testing in this case were **bynxFLEET** Short-term Rentals and Fines Management, which were chosen because they are transactional in nature and process high volumes of activities daily. This makes them some of the most hard-working elements within **bynxFLEET** and therefore testing them delivered a good, overall indication of the solidity of the platform as a whole. The applications are primarily developed in Java and Oracle.

Omnext SaaS automated analysis was performed on the source code to evaluate maintainability, security, IP infringement risks and quality.

Initially, **Omnext** performed a 'fit test'. The company uses different industry accepted methods, scales and metrics for this, which ensures every area of code is uncovered, tested and nothing stays hidden.

In static analysis, such as this, debugging is done by examining the code without executing the program to reveal errors at an early stage in development, often eliminating the need for multiple revisions later.

Blank explains:

“**Bynx** wanted to gain detailed insight into the quality of the application's code in order to minimize technical debt. This ensures compliance with industry standards and highlights if it contains any security or infringement risks.”



The problems with software development

Quality source code is key to the success of all software product stakeholders. Inherent source code quality will ultimately deliver a product with, among other attributes, flexibility, maintainability and usability. It's important all of these characteristics are treated equally to eliminate the risks and challenges associated with software development.

Bynx' quality standards are based on the objectives of:

- A clear definition of purpose for the application
- Simplicity of use
- Ruggedness (hot on error reporting and difficult to misuse)
- Reliability
- Extensibility
- Brevity
- Efficiency (fast but secure)
- Cost-efficiency
- Conformity to industry standards

Measuring the flexibility and maintainability of software, which Omnnext does to within the framework of ISO 25010, ensures unbiased results and reveals detailed analysis of technical debt.

Security risks that are traceable to the development process can exist in source code but many more vulnerabilities only come to light when an attack occurs. Testing can uncover these so they can be solved preventatively.

Using open source components in commercial software development is a normal practice that delivers positive benefits: cost-efficiency, speed of development and productivity. But it can also result in licensing issues as it is possible some components are dual licensed. Some versions of the open source components could be outdated, which increases the risk of exploitation, security vulnerabilities, 'buggy-ness' and poor maintainability.

The benefits - seeing your software in a new light

Bynx was happy with the results of the test, which concluded that **bynxFLEET** is highly maintainable. For the customer, this means the risk of high maintenance costs is low and the application is robust and easy to change and configure. The test also demonstrated that **bynxFLEET** is well structured and agile. From a customer perspective, it makes **bynxFLEET** a safe and secure choice.

Helpfully, the test highlighted where the company can reduce complexity and improve error handling in Oracle. It uncovered hidden security and open source vulnerabilities, which have now been fully addressed.

"Without the test, we may not have become aware of these so soon." Binks concludes.

For **Bynx**, testing the validity and quality of its source code makes sense, not only to validate the company's quality standards and coding practices but to highlight where further improvements can be made cost-effectively.

Someone once said: "No meeting is ever by chance" and in this case, Blank and Binks made sure that was true.

