

## Case Study: GARAIO Code Coverage Up To 75% With Typemock Isolator

### The Company

Founded in 1994, GARAIO AG develops and provides software solutions based on Microsoft technologies. As one of the few of Switzerland's Microsoft Gold Certified partners, the company provides its wide range of solutions to middle to large-size enterprises and to public administrations.

GARAIO boasts years of expertise and vast knowledge in the implementation and development of Microsoft-based projects adapted to the individual needs of each customer. GARAIO attaches great importance to the quality of its developments, and it is the company's vision to be leader in technology-rich projects for today's complex market requirements.

*"One of the tools we tested would work on well designed applications, but Typemock was the only one that could test an application that was not designed with testing in mind."*

**Jörg Jenni, Senior Software Architect, GARAIO AG.**

### TDD Challenge

GARAIO implements test driven development (TDD) as they found it to be much faster and enforceable than any other development methodology. Not only do the company's senior developers take advantage of mocking techniques, but they also motivate junior developers to use it in their daily testing tasks.

Since GARAIO introduced TDD, they faced several problems. A major difficulty was dealing with dependencies during unit testing, which proved to be too complex and costly to maintain. GARAIO discovered that they had to break up test dependencies by mocking all objects that were not subject to a particular test. Since both .NET and C# can't do this, they began searching for a mocking solution.

*"We can now test more efficiently components that do not have a testable design. With the help of Typemock, we managed to reduce our test setup code dramatically."*

**Marco Studer, Senior Software Developer, GARAIO AG**

### Other Solutions Failed to Rise to the Challenge

GARAIO first tested other solutions but found their behavior inconsistent and that they significantly affected their system design. For example, one of the tools they tried forced them to change the properties of their classes to virtual properties to facilitate mocking.

### Discovering Typemock Isolator

GARAIO wanted to unit test their code without significantly increasing the number of developers conducting the tests. What the company did change, however, was their test coverage and the importance they attached to unit testing. With

Typemock's Isolator, GARAIO's team of six finds unit testing a natural part of their work routine, while other teams in the company are still struggling to introduce the TDD way of developing.

*"One of the tools we tested would work on well designed applications, but Typemock was the only one that could test an application that was not designed with testing in mind."*

**Jörg Jenni, Senior Software Architect, GARAIO AG.**

## Facts and Figures

Typemock's Isolator afforded GARAIO a myriad of benefits:

The company increased code-test coverage from an estimated 10% in their last project prior to the introduction of Isolator to 75% in their current project.

In a previous project before Isolator, it had taken GARAIO some five years to complete more than 2,000 tests; with Isolator, in only six months in their current project they carried out over 1,500. This translates to a remarkable 700% increase, from 466 to 3,336 tests per year!

In GARAIO's previous project, four out of eight developers were actively implementing unit testing (some using the recently introduced Isolator); today, six out of six developers unit test with Isolator.

## Results

Today, GARAIO fully implements TDD with Isolator, without exception. In addition to reducing the lines of code necessary to set up mocks and tests, the company has substantially reduced mocking and testing costs. Moreover, developers can now implement straightforward mocking, without having to spend time on interfaces or virtual components.

*"While we have not yet invested resources in quality metrics, we firmly believe that Typemock Isolator enables us to release products that comply with the high standards of quality our company is committed to."*

**Christoph Hilty, Software Developer, GARAIO AG.**

Typemock Isolator for .NET unit testing enables GARAIO's developers to focus their energy on developing quality code. The Isolator technology isolates code such that it is no longer necessary to refactor or restructure code to make it testable.

There is no doubt that Isolator enables GARAIO's developers to meet the uncompromising standards of quality to which the Swiss company is committed. The introduction of Typemock's Isolator into their R&D department is in line with the company's vision of adapting the newest and most innovative technologies into their projects and solutions.

## Industry

Microsoft-based software solutions house

## Challenges

Breaking dependencies during unit testing  
Reducing testing complexity and high costs

## Results

700% increase in the number of tests conducted during the development process!  
Increase in test coverage from an estimated 10% to 75%!  
Improved quality at lower costs

## Benefits

Less time spent testing and/or fixing bugs  
Repeatable unit tests reduce development time and increase efficiency and quality

## Summary

Implementing TDD (test-driven development) as their development methodology, Swiss company GARAIO needed a testing solution that would not force them to deal with dependencies during unit testing – a very complicated and expensive process. After trying several solutions, they selected Typemock’s Isolator, the only tool they found that could do unit testing the way they wanted even for applications not designed with testing in mind. This made their unit testing easy and cost effective, and they have never looked back!

Comments? Write to [webmaster@typemock.com](mailto:webmaster@typemock.com)  
Copyright © Typemock Ltd. 2004-2008. All Rights Reserved | [Legal](#)