Typemock™
The Unit Testing Company

Intronis
MSP Solutions by Barracuda

INTRONIS MSP SOLUTIONS DOUBLES UNIT TEST COVERAGE IN NEW CODE

INDUSTRY

Online services

CHALLENGES

- Ensure that code quality stays high
- Improve the quality of code before it was handed off to QA

RESULTS

100%
of new code is now tested

ZERO
blocking bugs since implementing unit testing

2-3
minutes to write a new test

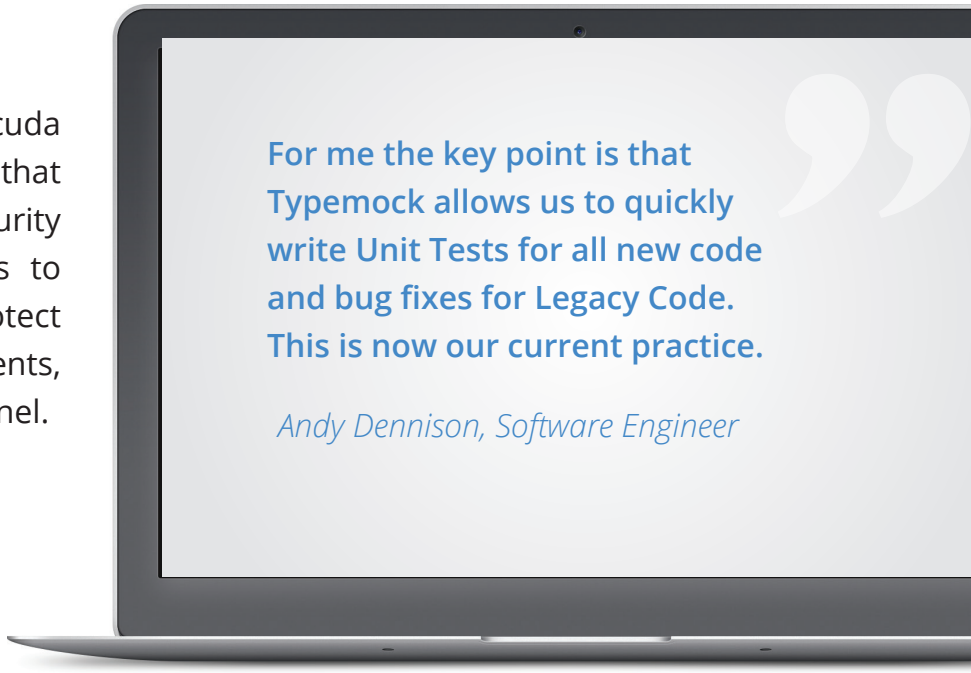
87%
of its developers feel empowered using Typemock™

SUMMARY

In its development work on its SaaS data protection solution, the Intronis ECHOplatform, the Intronis Developers Team required the power of Typemock to mock Legacy Code, including static and private methods. Redesigning Legacy Code to improve testability invites regression bugs. Bugs in QA have the potential for triggering an entire re-test cycle, resulting in expense and slipped schedules. The Intronis MSP Solutions Developer Team readily adopted Typemock; within 6 months new code test coverage doubled and writing unit tests is now their practice.

BUSINESS

Intronis MSP Solutions by Barracuda is the single-source provider that delivers a portfolio of security and data protection solutions to proactively and reactively protect a broad range of IT environments, exclusively through the IT channel.



TYPEMOCK ISOLATOR AT INTRONIS MSP SOLUTIONS

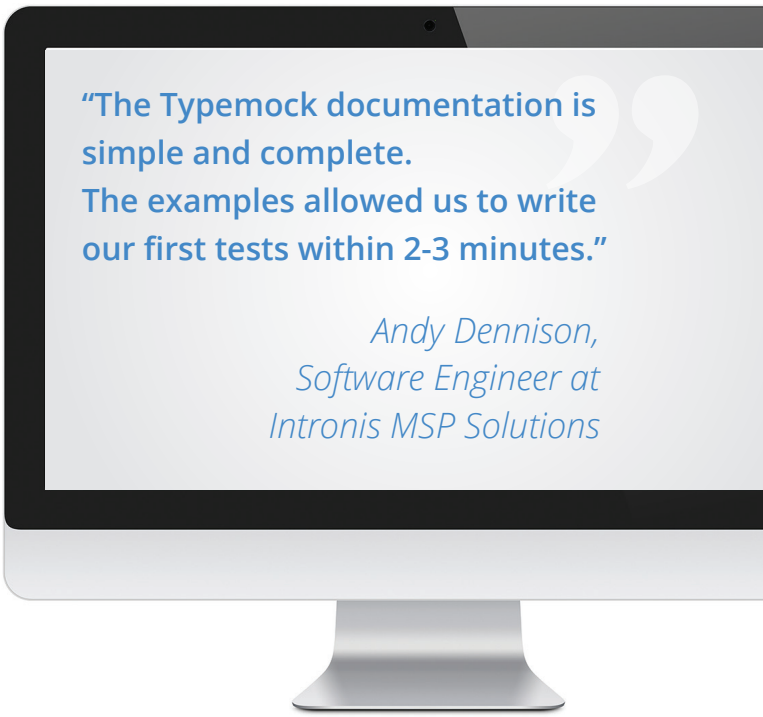
The Intronis MSP Solutions Development team chooses to use Typemock's Isolotor because it is a sharp, convenient system for unit-testing legacy code. Isolator mocks or isolates the code components which are not under test. This allows developers to construct test cases where the outcomes are the expected behavior and also expected exceptions for abnormal inputs. The result is more test cases covering a wider range of scenarios, leading to higher quality code leading in to QA.

"QA has consistently been receiving builds without system-wide blocking issues since adding unit tests with Typemock. That's a significant improvement."

Steve Lilley, Director of Quality Assurance

Typemock enables the Intronis MSP Solutions Developer Team to conduct automated unit testing of software with efficiency and ease. It has remarkably sped up development, saving the the Development team countless hours of frustrating, time-consuming and ineffective efforts while ensuring the high quality of the Intronis MSP Solution's products and services.

The Intronis MSP Solutions Developer Team is enthusiastic about Typemock's powerful ability to isolate, fake, and inject objects. They praise its consistent API which can mock everything they need, including private and static methods and fields. In addition to meeting their mocking needs, the Intronis MSP Solutions Developer Team is fond of using Typemock because it allows them to write unit tests without having to change their code. They found that Mocks are easy to specify using the Typemock™ fluent syntax. Andy Dennison expresses the team's whole-hearted support and explains why they decided to go with Typemock over open source and other mocking frameworks:



"For me, the key point is that the current development practice is that all new code and bug fixes will be covered by Unit Tests. "Mature" (i.e. with statics and large private methods) testing frequently requires the power of Typemock. This was understood during tool selection and is the prime reason for buying the tool."

Andy Dennison, Software Engineer

Dennison regards testing private methods as the *"lesser evil"* to not testing at all before making changes. He holds that the correct way to work on Legacy Code is to lock down the current behavior with Unit Tests before bug fixing, redesign or refactoring, and then to run the regression tests after the code changes have been made. To facilitate the adoption of this development process, he and his team members required the ability to test parts of code which are normally not accessible, which is why Typemock has become such a valuable tool for them. In his opinion, redesigning Legacy Code so that it can be tested is too time-consuming and invites regression bugs.

As a bonus advantage to testing with Typemock, Dennison notes the value of developers looking at real code in the testing process. His reasoning follows that when experienced developers look at real code *"they will often demonstrate best practices that may only be peripherally related to examples."* The unintended, yet very helpful, outcome of sparking developers' abilities to implement new successful processes is rare in testing solutions available to developers today, and an added gain for developer teams and their companies that test with Typemock.