



Company: Stago Group

Stago, founded in 1945, is a French medical technology group that employs more than 2,200 people worldwide. Since 1962, Stago has dedicated its entire resources to developing the single field of coagulation. Permanently attuned to the medical and scientific communities, Stago is actively interested in all advances in this domain and in all the implications of such progress. Through this strategic approach, the company now has many major patents and innovations.

Stago employs several dozen programmers who develop the company in-house software.

Challenges and Objectives

Stago was looking to improve development quality to minimize development time, to reduce the amount of resources required for every project, and to eliminate business disruption due to code quality issues.

Controlling Code Quality

In 2010, Stago started developing in .NET and rapidly implemented processes to ensure code quality and architectural integrity. The team started looking for ways to automate and systematize quality standards.

As the .NET team grew, the requirements needed to effectively supervise an increasing amount of code production expanded in complexity and criticality.

Writing Custom Coding Conventions

The software architect team at Stago wanted to make sure that all their coding conventions were strictly and continuously respected. They needed a tool that not only let them easily add their custom coding conventions as custom rules, but also let them ensure such conventions were respected both by each individual .NET developer environment (in Visual Studio) and during all build process.

Detecting and Fixing Issues Early

Stago's team recognized that the monetary and time cost of fixing these issues increases exponentially the later they are detected in the application lifecycle. It is usually

accepted that a bug costs 10 times more if it goes unnoticed through Quality Assurance (QA).

An issue reported during the Quality Assurance process at maintenance time or, even worse, at production time, will typically take several man-hours/days from support and development teams to investigate and fix.

Moreover, this has a negative impact on the end user experience and business processes. Negatively impacting IT credibility as executives and end users only remember the times when things broke down.

Conversely, an issue caught within Visual Studio at development time will typically only take a few minutes to be fixed on the spot by its developer. There are no impacts on future code maintainability, nor on the end user experience or business continuity. This capability of fast detection and fix was one of the key requirements for Stago.

Another important objective was to ease the relationship between the developers and the architects responsible for enforcing coding conventions and quality standards. Constant reminders of the same quality requirements not respected by developers consumed valuable architects and developers' time. It also strained the relationship within the .NET team.

Solutions

After having conducted an extensive evaluation between all the possible products on the market, the Stago IT engineers selected NDepend. Since then, all Stago .NET developers have been equipped with NDepend in their development environment and it is also available on all build machines.

"We selected NDepend after reviewing all the other options on the market and it quickly became the backbone of our development effort. It ensures we always deliver superb quality code, on time, on budget and with the minimum required level of effort. Since we adopted NDepend we have saved the equivalent of several man-years in time and cost. We have also suffered no breakdown of our applications in production."

- Fabien Prestavoine, Architect

The key selection criteria were:

Easy to Write Custom Coding Conventions

NDepend is the only tool that makes it very easy to develop custom coding rules on .NET code. With NDepend, a coding rule is just a C# LINQ query written against a code model fluent API.

The Stago team has written around a hundred custom code rules that are checked continuously against the code produced by the developer team. It is done both in the developer environment and during the build process.

Check Issues Quickly and Frequently

NDepend is also integrated in Visual Studio and has an automatable module that can be integrated into the build process.

All custom coding rules are frequently checked on the developer's machine and are considered critical by Stago's Architects. Both NDepend analysis and rule execution typically takes less than 5 seconds.

As a result, all violations are fixed as soon as they have been detected. Even more importantly, developers

quickly learned to avoid making mistakes and triggering rule violations, effectively producing squeaky-clean code on the first try.

As requirements evolve, new rules are created every time the architects discover a non-wanted practice they want to eliminate.

Pain-free Tooling

NDepend proved to be easy to get started and easy to use. Stago's team implemented it themselves and started to use it without any external training or consulting in a matter of days. Today, Stago does not even include a specific NDepend explanation when training new developers.

NDepend proved to be stable in the long term and most of questions Stago users had were already answered by the help embedded in the tool or on the online knowledge base. They have not experienced any issue during 5 years of daily and intensive use, including after version upgrades.

Testimonial

"As a developer in a team operating in agile mode, we are often focused on the result and the satisfaction of the customer needs at the end of each sprint. Sometimes that means a mass production of a code, respecting little or not at all the good coding practices and transgressing architecture rules implemented by the architects. Fortunately, tools like NDepend are there, to remind us that this is not the proper way to do it, and enforcing the rules before the commits."

I highly recommend this tool for other teams."

- Mourad Chibane R&D Team

Benefits

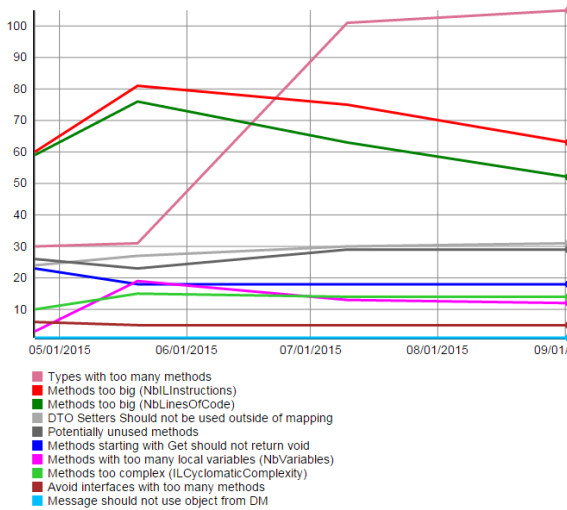
All .NET Developments are in the Green

Quality and architecture standards match Stago's requirements and are continuously respected. Every single violation becomes obvious and is immediately fixed before committing the source code.

Significant Gain in Team Productivity and Predictability

Architects and quality engineers are relieved from a lot of the quality control work. Catching problematic issues before QA and maintenance time, code evolution has become a continuously smooth process. Stago developers no longer spend time reworking the same pieces of code over and over again.

Over the last 4 years, all delivery deadlines were met with ease. A major factor in reaching such an outstanding process was having no surprises concerning code quality during testing and maintenance. Developers' skills have also markedly improved with a continuous decrease of bad practices detected as they write code.



Improved Relationships Between Architects and Developers

NDepend automatically pinpoints mistakes when they are introduced by developers in their own code. Architects are no longer seen as "the bad guys" who have "nothing else to do than to blame developers" when

something goes wrong. Architects, on the demand of the developers, are now the senior guys who provide guidance and advice about how to avoid NDepend flagging issues. As a result, team dynamics have markedly improved for the better.



Stago's Architecture Team: Thomas Radioyes Jérémie Yardin, Jessica Castejon, Fabien Prestavoine, Xavier Solau, Frédéric Huchon (absent)

Ultimately, the Stago .NET development team uses NDepend as the backbone of their quality process.

Testimonial

"Not only are we developing better and faster applications for less money, but we also have a much better relationship between the developers and their managers. By letting the developers proactively and easily clean their code, NDepend really eliminated the main source of tension in our jobs.

NDepend also proved to be a completely reliable company and I am only too happy to warmly recommend NDepend to other development groups."

- Fabien Prestavoine, Architect