SIG TECHNICAL DEBT MANAGEMENT AT SCALE

Automatic assessment of the ROI on Technical Debt removal

Assessing your team's technical debt, helping to pay it off, and calculating the ROI of the repayment

The Challenge: Technical debt is slowing down your business

Software systems are often used for many years. During these years of operation, these systems are continuously changed. There are many different ways to make changes to a software system. Some ways are much better than others in that they provide easy ways to make changes later. Sometimes, a less-than-optimal approach is taken. This is often due to a lack of time or knowledge on the part of a developer.
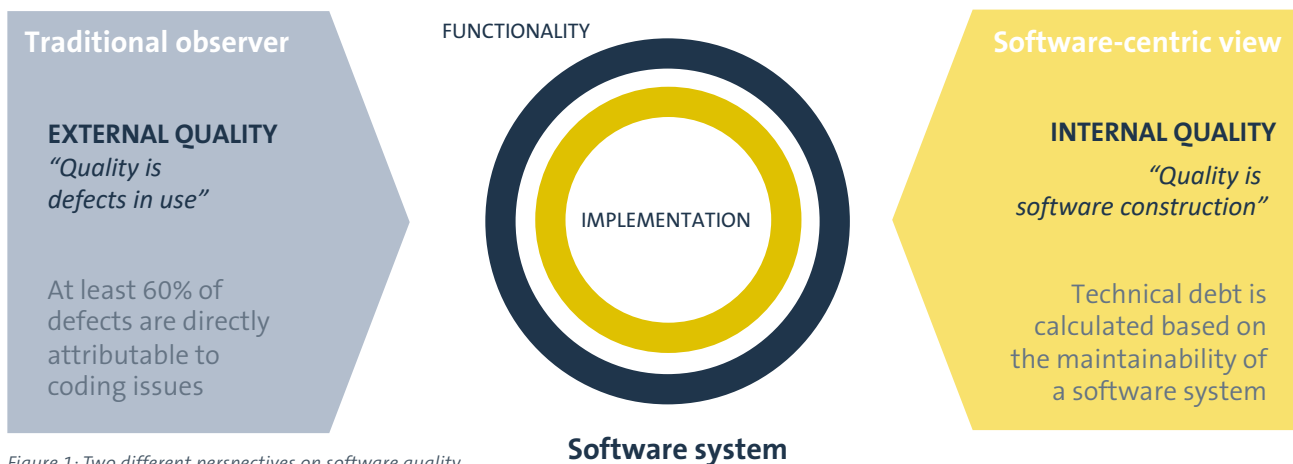

Traditional observer

EXTERNAL QUALITY
"Quality is defects in use"

At least 60% of defects are directly attributable to coding issues

FUNCTIONALITY

IMPLEMENTATION

Software-centric view

INTERNAL QUALITY
"Quality is software construction"

Technical debt is calculated based on the maintainability of a software system

Software system

Figure 1: Two different perspectives on software quality

Such easy solutions and quick fixes were often meant to be temporary, but no time was taken to properly solve or integrate them. The sum of these inconsistencies is often referred to as technical debt. Every time a developer needs to make changes to a system, the technical debt in the system incurs interest on his or her work. This interest slows down development. Comparing an average legacy system to a well-built new system, with little technical debt, the speed of development is three times faster in the latter.

Over their lifetime, systems accrue an enormous amount of technical debt. Repairing all this technical debt is not feasible. Not only is it too expensive, it would also prevent developers from working on what really matters for business: functionality. In addition, often, not even

a system's developers have a clear overview of all of its technical debt. Nor can they quantify the effect of removing a specific instance of technical debt. As a consequence, it is very challenging to convince business to allocate time for technical debt removal.
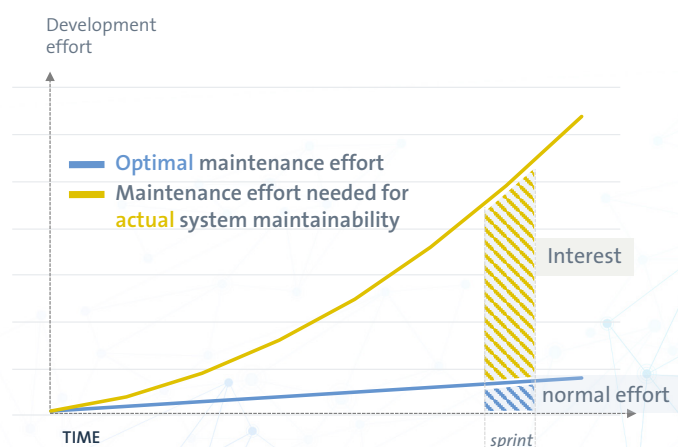

Development effort

— **Optimal** maintenance effort
— Maintenance effort needed for **actual** system maintainability

Interest

normal effort

TIME

sprint

Figure 2: The interest of technical debt

The Result: Technical debt is not structurally addressed by development teams

Performing a comprehensive inventory of technical debt requires finding all instances of maintainability violations of all systems in your landscape. In addition, comparability will be a challenge with the many different technology stacks that are typically in use. Immediately after building this overview, it will be outdated due to continuing development. An even harder challenge is that the list of problems will vastly outweigh the available time and budget. Where to start with solving problems? What is the most efficient use of those limited resources?

The Solution: Automated ROI calculation of refactoring candidates

A maintainability violation in a software system is a sub-optimal solution that slows down software development. The solution for such a violation is rewriting the code related to that violation. This rewriting, or refactoring, costs time and effort, but it removes the maintainability violation and thereby reduces the technical debt in the system. Future development should now be a little bit faster.

But not all refactorings are created equal. Some deliver much more future development productivity increases than others. In short, a refactoring-specific ROI should inform your decision where to use your limited resources.

Such a ROI calculation should, for example, take into consideration whether the code to be improved is likely to be worked on again in the future, what the risk of the refactoring is or what the resulting maintainability increase on the system of a whole will be.

Promising refactorings, which we call refactoring candidates, are presented to the development team for final consideration.

SIG has developed an approach in which these factors can be calculated and prioritized automatically and continuously, as systems are developed. By taking a portfolio-level approach, refactoring candidate prioritization can be performed over multiple systems to efficiently distribute a limited refactoring budget.

SIG

Why SIG

We are a group of software enthusiasts, who over the past 20 years have operationalized and optimized pragmatic software quality management. Only SIG has the expert personnel and tools to carry out technical debt assessment over massive industrial IT landscapes quickly, efficiently, and effectively.

We've been performing this kind of software analysis for more than 15 years, and we now have more than 36 billion lines of code in the benchmark data we use to assess the technical debt in your code base.

To further ensure our technical debt assessments are in line with how developers see their system, engagement with SIG includes a session with the involved teams to validate the understanding we gather about the inner workings of the code base and system architecture with the very people who built it.

We support many large international organizations in the continuous and structural improvement of their development productivity. In addition, we carry out more than 300 cross-border software development assessment projects annually. SIG is specifically experienced in translating technical findings to razor sharp advice to all layers of management – whether they understand software or not.

Deliverables

We deliver comprehensive technical assessment of the maintainability of the systems in scope. Our reports are geared towards qualitative, fact-based information that highlights the biggest risks and opportunities refactoring potential. Typically, these fall into five types:

1. **A report on the current level of technical debt.**

2. **A repay estimation. Technical debt, quantified in terms of both time and money, that explains in detail how many person-months of work would be required to pay off the debt you currently have over your IT system.**

3. **A calculation of the ROI on paying off the technical debt.**

4. **A quality benchmark that rates the teams' software against industry-relevant quality criteria for companies of its volume and technology class.**

5. **A reconstructed architecture of the IT portfolio that works from the IT assets themselves to give an accurate picture of the real-world environment, exposing flaws in the idealized architecture represented in documentation and giving you an invaluable post-transaction guide for maintenance, development, and integration activities.**

The Benefits of Working with SIG

Drawing from SIG's vast experience in the field, our consultants deliver focused IT advice that fits the purpose of having a clear pragmatic way in paying down your landscape's technical debt. Findings are to-the-point, with clear action items assigned — and always based on measurements against the industry's largest benchmark, which contains more than 36 billion lines of code.

Clients specifically report the following benefits:

- **A pragmatic prioritized overview of actions to take**

- **Goal-oriented interactions, as opposed to yet another hours-based report with impractical recommendations**

- **Identification of risks, along with the ability to take timely mitigating actions**

- **A nuanced, true understanding of development teams' daily challenges**

# SIG Technical Debt Management Monitoring
Why Monitor?

After assessing the amount of technical debt your system or portfolio contains, SIG indicates the refactoring candidates that will make the highest impact on the amount of debt you are continuously repaying.

However, a system will inevitably keep degrading and having a grip on your technical debt is imperative to stay in control of this process. Therefore, several years ago, SIG chose to open up the ability for development teams to work with SIG's own proprietary tooling, Sigrid®. Sigrid is a software assurance platform that gives you continuous insight into the quality of your software portfolio with, yet again, clear pragmatic indicators on how to achieve the quality goal you have set for your team. In this manner, software quality is embedded in the way of

working. Our consultants will check in with you regularly to indicate possible improvements, so you stay in full control of both your overspend as well as the quality of your software.

We have looked at this overspend in conjunction with the frequency of system monitoring, conducting research on teams that have analyzed the technical debt one time only; inconsistently; and consistently monitored their technical debt and progression on paying it down.
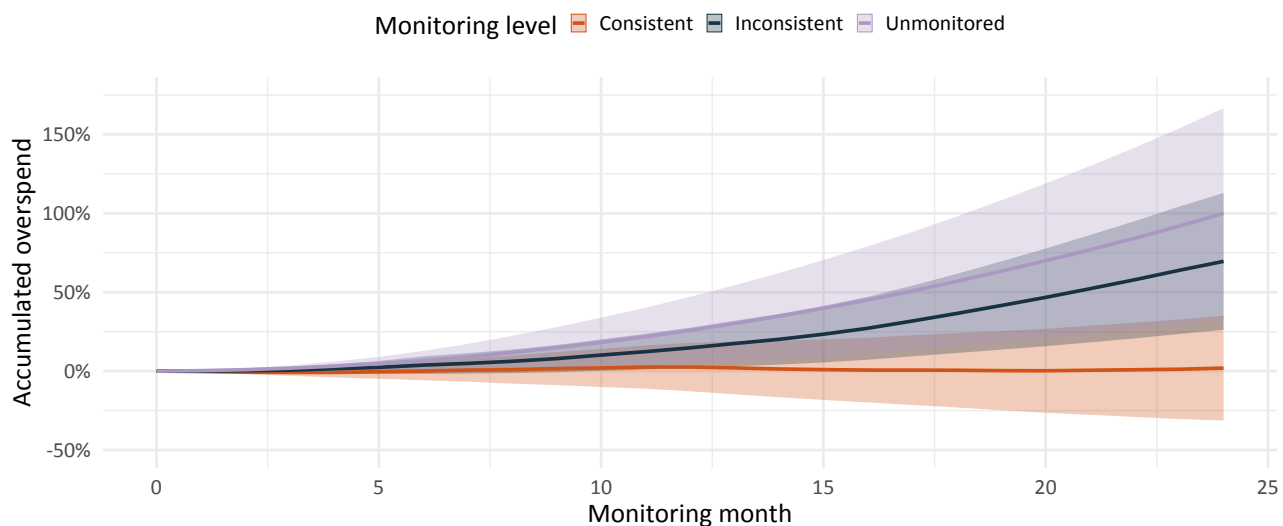
True control over code quality - *at vendor or internal* - using SIG's monitor Sigrid

Holistic quality insight and advice
(maintainability, security, privacy, reliability, performance)

Actionable advice that matters
(No more long lists of meaningless violations)

Ease of use
(Just upload code, works out of the box)

Continuous insight on all levels
(From meaningful portfolio KPI's to code-level issues)

Integrated best-of breed tooling
(No learning curves, no failed adoptions, single point of truth)

Integrated on-demand services
(Coaching, helpdesk, expert code-reviews, consultancy)

With Sigrid, SIG helps organizations improve their software health in every stage of the software life cycle – on a continuous basis or with a single assessment.

BUY

Vendor Selection

Outsourcing

M & A

RFP Review Quote Challenge

Productivity Monitoring

IT Due Dilligence

Early Effort Estimation

Solution Development

Innovation

Cloud Readiness Scan

Sigrid

Maintainability
Security
Performance
Reliability
Productivity

Architecture Design Review

Architecture & Design

BUILD

Portfolio Monitor Software Risk

Software Risk Monitor

Maintenance

Legacy Transformation

Development

OPERATE

Improvement

Sigrid enables the visibility & recommendations you need

SIG

About SIG

Software Improvement Group (SIG) helps business and technology leaders drive their organizational objectives by fundamentally improving the health and security of their software applications. SIG combines its proprietary tools and software analysis data with its consultants' expertise to help organizations measure, evaluate and improve code quality – whether they're building, buying or operating software.

SIG has the world's largest software analysis database with more than 36 billion lines of code across more than 280 technologies. The expert consultants at SIG use this data to evaluate an organization's IT assets on maintainability, scalability, reliability, complexity, security, privacy and other mission-critical factors. The SIG laboratory is the only one in the world accredited according to ISO/IEC 17025 for software quality analysis.

Founded in 2000, SIG is headquartered in Amsterdam with regional offices in New York, Copenhagen, Antwerp and Frankfurt. For more information, please visit www.softwareimprovementgroup.com

SIG Software Improvement Group

Fred. Roeskestraat 115

1076 EE Amsterdam

The Netherlands

www.softwareimprovementgroup.com

marketing@softwareimprovementgroup.com