

# *TETRA, as a set of techniques and tools for calculating technical debt principal and interest*

Boris Kontsevoi  
President & CEO  
Intetics Inc.  
Naples, FL, USA  
[boris@intetics.com](mailto:boris@intetics.com)

Elizabeth Soroka  
Inbound Marketing Director  
Intetics Inc.  
Minsk, Belarus  
[e.soroka@intetics.com](mailto:e.soroka@intetics.com)

Sergei Terekhov  
Director Quality Assurance  
Intetics Inc.  
Minsk, Belarus  
[tsv@intetics.com](mailto:tsv@intetics.com)

*The paper covers the company's proprietary approach to the assessment of software product quality and measurement of technical debt. In the paper, the authors state that the software product quality is the amount of various technical debts embedded into the product and delivered to end-users. Similar to the fact that any debt can be paid, the software product quality can be efficiently managed and measured by a set of techniques, parameters, and dimensions. The authors share the details of the company proprietary approach to technical debt management done with the help of the Technical Debt Reduction plAtform, TETRA. They share the measurement parameters, tools and major dimension for the assessment.*

*Keywords — technical debt, metrics, measurements, quality, test, testing, bug, fixing, software.*

If you have been in the software development industry for a long period, the chances you have had to explain the notion of software quality are quite high. The major stumbling block is that software quality is often understood differently. Some define quality as the level of customer satisfaction, others say, it is about meeting the requirement of the customer. For the tech world, it is the state of software being free from defects. To us, the latter is closer to the real world, though it often raises even more questions. At Intetics, we decided to fix the misunderstanding by carrying out intensive research on what the quality of a software product is, and how we can manage it efficiently. As a result, we came to an opinion that the quality of a software product is the amount of various technical debts embedded into the product and delivered to end-users.

Industry best practices state that technical debt is any code added now that will take more work to fix at a later time, typically with the purpose of achieving rapid gains. The source code is really one of the most important aspects in software. It is well known that the higher the number of problems in the source code, the more redevelopment is required. This is how the industry understands the technical debt, and we all agree. However, having dealt with it for many years, we started thinking about technical debt in a different way. We have come to an idea that technical debt is “*a combined product non-compliance with technical guidelines and business objectives that negatively impact business results.*”

With this idea in our minds, we spent over 6000 hours of research, defining the critical dimensions of software product quality. These dimensions became the skeleton of our software product assessment platform, which we called TETRA, Technical DebT Reduction plAtform. Why TETRA? The technical debt is like a tiny tetra fish that exists in thousands of various species. It's like that small, insignificant, sometimes “locally beautiful” trade-offs of quick solutions that are easy to implement instead of applying the overall best solution. When technical debt piles up, we pay an “interest” in the form of harder maintainability, mediocre user experience, development team productivity decline, and overall higher costs. We got all those little fish in one aquarium and created the TETRA platform. How do we know what really matters in software products and can we assess both the technical and the business sides?

They are measured and assessed by a certain set of techniques and tools, chosen on the basis of best practice analysis, Intetics original research, and expertise acquired during multiple projects. The result of these measurements gives the opportunity to calculate the technical debt of the overall software product and its parameters. To acquire the big picture, we assess the product as per eight dimensions in the following way.

## I. DIMENSION 1.

**Source code quality** is the assessment of the quality of the product source code by the follower major parameters:

- **Cyclomatic Complexity** (methods, classes, files) is a measure of the number of linearly independent paths through a program module.
- **Duplications** (lines, blocks, files) is the number of physical lines touched by duplications.
- **Code coverage** (lines, branches) is the percent of source code covered by unit tests that help check individual units of the source code to evaluate whether they are fit for use.
- **Rules compliance means** the compliance of the source code to a coding standard and best practices of a particular programming language.

- **SQALE** rating is the rating that is given to a project related to the value of your Technical Debt Ratio.

To measure these metrics, we use the SonarQube tool [1] and additional created plugins.

## II. DIMENSION 2.

**Usability, UI & Documentation** is the evaluation of the product user interface, usability, and documentation. The process includes the expert assessment, check-list and the product assessment by learnability, memorability, efficiency, error tolerability and likeability metrics [2]. To increase the accuracy of the assessment, we use tools like W3C Markup Validation Service, Usabilla, Browser Shots, etc.

## III. DIMENSION 3.

**Security** is the valuation of product vulnerabilities according to CVE, CERT, CWE, OSVDB, OWASP [3], and BID. For the security assessment, we use both automated and manual approaches. The assessment is performed with the help of the tools like Burp Suite, OWASP ZAP, and Nessus [4].

## IV. DIMENSION 4.

**Performance.** We run product evaluation based on customer requirements done by load testing and virtual users' activity simulation. Using tools like JMeter, LoadRunner, WebLoad and LoadUI Pro, we assess a set of performance parameters, among those stand average response time, peak response time, concurrent users and a set of parameters for the server state assessment.

## V. DIMENSION 5.

**Business logic** assessment is the evaluation of how well a software product complies with the required business processes based on the feedback of different focus groups like end-users, management, customers, community, etc. The evaluation is based on the six main metrics: effectiveness, project workflow quality, data safety, simplicity, business rules and policy, and competitiveness.

## VI. DIMENSION 6.

**Architecture quality** includes the expert assessment of product software core structure and data model. In the process of the assessment, we analyze design patterns, services, exception handling, infrastructure and recycling components, layers, etc [5], [6].

## VII. DIMENSION 7.

**Data quality** includes the verification of product resistance to bad data, exception handling, and bad data prevention. The assessment is based on the metrics like accuracy, consistency, validity, currency, completeness, timeliness, accessibility. We use the Talend Open Studio for Quality and manual testing to ensure the assessment accuracy.

## VIII. DIMENSION 8.

**Open Source code use.** The assessment detects inclusions of open source components to the product and helps to mitigate copyright, security, legal, and operational risks that can result of such use. In the result of the assessment, we define the list of the open source code in the product, licenses, and copyrights, and analyze the open source software version. The assessment is performed with the help of tools like Black Duck and Palamida.

Following this approach to analyze and measure the technical debt, we create the final comprehensive report, providing the description of every product weakness and the state of the product quality.

The important thing about TETRA is that it is not merely for technology experts. TETRA is an asset for everyone working with the product. Developers and testers get a non-prejudiced assessment of their work, managers receive feedback on the product's overall capacity, while users get the reliable and well-performing product. Moreover, the processes used for the product assessment train the team to follow best practices during the product development. This increases team proficiency and consequently the quality of the product.

The great thing about the platform is that it provides the product owners and investors with proper business information. The fair analysis of the product tells companies whether the product is ready for the market launch. In case the product already works as the company offering, owners can estimate its efficiency and value on the market. As for the investors, TETRA allows them to evaluate the state of a product for purchase, better define the product's market value and assess the investment risks of the intended transaction.

The ultimate goal of TETRA is to quantitatively evaluate the condition of a product to achieve a better result. Without a comprehensive measurement, there is no way to understand the quality and control it. TETRA measurements answer the question of how well the requirements of development teams, product owners, users, investors, and market are met. This is about the efficient control of your product.

## REFERENCES

- [1] SonarQube Documentation, Metrics Definitions, <https://docs.sonarqube.org>
- [2] ISO/IEC 25063:2014: Systems and software engineering -- Systems and software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability, in press.
- [3] OWASP Foundation, "OWASP Top 10 – 2017, The Ten Most Critical Web Application Security Risks", in press.
- [4] CISQ Code Quality Standards, <https://it-cisq.org/standards/code-quality-standards/>
- [5] Microsoft, "Microsoft Application Architecture Guide, 2nd Edition", in press.
- [6] Paul Clements, Rick Kazman, and Mark Klein, "Evaluating Software Architectures: Methods and Case Studies", in press.