



QUALITYGATE AS AN SLA MONITOR IN BANKING ENVIRONMENT

CASE STUDY
FRONTENDART

or – how to measure what has
never been measured before

The following case study shows how the QualityGate source code analyzing and management system has been introduced into a bank environment as an SLA measurement tool – as well as the installation process and the most important experiences of the introduction and pilot period.

Preamble

The Chief Information Officer (CIO) from a major bank's IT department received the task of eliminating the live operation software errors in particular basic grade systems and optimizing development to ensure a smooth business process.

To take control over the quality of developed systems, the CIO decided on signing service-level agreements (SLAs) with the developers. QualityGate was selected to be part of the quality measurement system as a source code analyzer and management tool.

Goals

The CIO had the following expectations towards introducing the tool:

- The IT department staff should be able to use it without related expertise
- Measured software quality should be confirmable in practice:
 - lower quality source code should contain more errors
 - increased quality should reduce the average development costs

Pilot period

The bank has never used tools of this sort before, so they did not have the needed experience. To acquire this experience, they started a 3-month pilot project of introducing QualityGate to the organization.

Main targets for this pilot period:

- Precise usage of the management tool
- Calibrating measurement models and benchmark,
- Clarification of the measurement evaluation process.

FrontEndART delivered a quality model and benchmark along with QualityGate – both calibrated with international experiences in mind.

Three different basic grade systems were evaluated during the pilot project. The main steps that were performed:

1. Reference measurements, assembly of a quality concept

The first source-code quality assessments were performed using the quality model and benchmark delivered by FrontEndART. Results were discussed between the bank's IT professionals and the development companies under the supervision of FrontEndART.

2. Periodic measurements – new versions

New versions of software from third party developers were tested with QualityGate prior to live application. Upon receiving detailed results, the developers fixed problems which were considered critical, only after that could the version go live.

3. Refinement of SLA values

Target qualification values were specified per system after the pilot period. These values became the minimum threshold for accepting new versions.

Summary of measurement results

Measurement period	System 1	System 2	System 3
0 months (reference)	3,2 (low)	3,14 (low)	4,2 (high)
0-3 months	3,22 (low)	3,5 (low)	4,2 (high)
3-12 months	3,8 (passable)	4,2 (high)	4,6 (high)
Experiences			
Change in quality	+ improvement	++ huge improvement	+ improvement
Outcome of remaining errors	++ major decrease in number of critical errors	++ major decrease in number of errors	+ minor decrease in number of errors
Average time of development iteration	+ decreased	++ excessively decreased	+ slightly decreased

Conclusions

The most important conclusion is that better quality in source code can be achieved with the help of deliberate quality management – deducted the CIO

Quality vs measured values

When using supervised systems with measurably increased source-code quality, users reported of clearly positive experiences: number of daily problems were reduced and fewer errors needed to be fixed subsequently regarding further development.

Collaboration with the developers

Generally speaking, it can be said that the developers didn't have much confidence towards the quality measurement method in the beginning of the pilot period. Later, developers who collaborated with the bank could benefit from the measurement results, for these values are proof that they performed the necessary testing and that the supplied system fulfills the specified quality criteria.

Including developers in the design, development and operation of the measurement system has proven to be important.

Measurement results vs errors and development costs

Apparent correlation can be observed between the number of bugs found and the source code quality: with increasing quality, fewer errors are present in the live operation of the given system. However, in case of the development of new modules or greater functional changes, this correlation doesn't necessarily apply.

On the other hand, the measurement of development costs was much harder to perform because of the huge differences, aside from a few typical tasks. E.g. development time regarding report generation dropped significantly (~20%).

Joint examinations of measurements revealed potential error possibilities in multiple occasions – giving reference to the developers for source code improvement.

Results

Achievements made in the 3-month pilot period and in the following 6 months:

- Apparent quality improvement in the measured systems.
- These improvements noticeably reduced average number of errors found in the live running systems.
- Required time of comparable development tasks were reduced by 15%.
- The IT manager found an objective, independent technique of measurement to control the development process with – accepted by the developers and the directorate as well.