Towards Quality-Driven Software Migration

Jan Jelschen, Gaurav Pandey, Andreas Winter
Carl von Ossietzky Universität, Oldenburg, Germany
{jelschen,pandey,winter}@se.uni-oldenburg.de

Long-running software systems suffer from software erosion, due to their constant evolution to meet new or changing requirements, severely limiting their maintainability. Migrating software systems, i.e. transferring legacy systems into modern environments and technologies without changing functionality [FWE^{+}12], is a key technique of software evolution, and serves to keep existing software systems operational. Structured migrations allow for transferring established software solutions to cutting edge technology, without having to consider the significantly higher risks of developing a new system from scratch.

Reliable predictions regarding long-term trends of software systems' quality are a prerequisite to make sensible decisions between software migration or redevelopment strategies, which enable a system's continued development, evolution, and operation. Especially when planning to migrate software systems, and choosing appropriate tools, a prognosis of project-specific quality properties of the system *after* migration is required. To assess and compare software quality before and after language migrations, like from COBOL to Java, a major obstacle to overcome is the paradigm shift between procedural and object-oriented programming that occurs (or should occur) when translating between these languages. For example, any measures which try to capture object-oriented properties like encapsulation will only provide meaningful results on the target system, with nothing to compare to on the source system side, i.e. a direct comparison is impossible.

Recognizing these challenges, two main research questions can be identified, which lead to two sets of central objectives regarding *quality criteria* and *quality prediction*, to address them:

1. How to measure and compare the quality of legacy and migrated software systems?
2. How to predict the quality of software migration results?

Quality Criteria. A basic prerequisite for quality measurements is having a set of criteria appropriate for assessing and comparing the quality of software systems under migration, and knowing which criteria are relevant for a specific project. Whereas criteria for incremental quality monitoring in plain forward engineering projects, utilizing a fixed development environment, are known, comparable criteria applicable for legacy *and* migrated systems are unknown. Therefore, a *catalog of quality criteria* has to be assembled, elicited from literature review and expert interviews, and a process to define a migration project's goals, and select suitable, measurable criteria has to be created.

Quality Prediction. To help a cooperation to decide whether to migrate, or choose a different strategy (e.g. re- or new-development), or to tailor a migration process to specific needs, information is needed on the resulting system's quality properties *before* carrying out the migration. A quality prediction model has to be derived by measuring software systems before and after migration, and looking for correlations. Such a model will serve to project quality properties after a migration, given the source system, and the migration toolchain with its parameters. It can also be used to refine such parameters in an iterative process, to meet certain quality requirements in the target system.

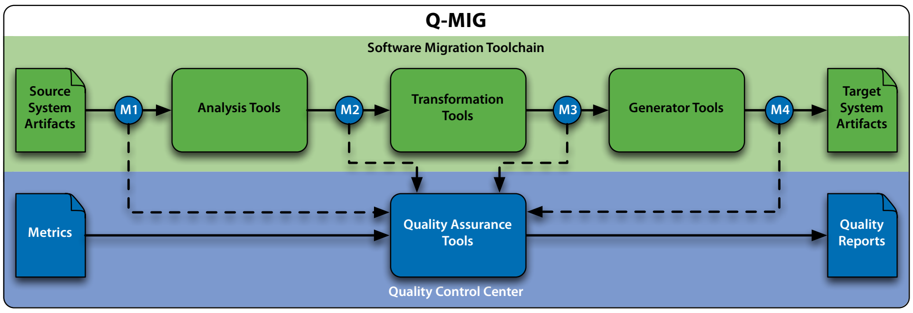

Figure 1: The Q-MIG software migration toolchain and quality control center.

Figure 1 depicts a generic *software migration toolchain*, equipped with a *quality control center*. The former takes source system artifacts and transforms them into target system artifacts, according to the migration scenario's parameters, using appropriate analysis, transformation, and code generation tools. The latter complements the migration process by tapping into it at monitoring points, evaluating quality criteria using metrics at different steps, assessing and comparing the quality using quality assurance tools, and generating quality reports as a result.

To realize the quality control center, the central questions subject to research are *a*) which criteria are relevant for a given migration scenario, *b*) how to measure and interpret them in a way that correlates with experts' assessments, and *c*) how to sensibly compare them, considering programming paradigm shifts.

These issues will be addressed by the project *Q-MIG: Building a Quality-Driven, Generic Tool-Chain for Software Migration*[1], which commenced on January 1, 2014. Q-MIG is a joint project of *pro et con Innovative Informatikanwendungen GmbH* (focusing on the *software migration toolchain* [EU11]) and *Carl von Ossietzky University* (focusing on the *quality control center*). For integration, a service-based approach will be used [JMOW13]. Q-MIG aims at enabling project managers and software developers to make early strategic decisions regarding the realization of software migrations, and to already carry out targeted quality assuring measures during the migration.

At the workshop, we would like to discuss how to predict comparable (internal) quality criteria for software migration projects.

References

[EU11] Uwe Erdmenger and Denis Uhlig. Ein Translator für die COBOL-Java-Migration. *Softwaretechnik-Trends*, 31(2), 2011.

[FWE+12] Andreas Fuhr, Andreas Winter, Uwe Erdmenger, Tassilo Horn, Uwe Kaiser, Volker Riediger, and Werner Teppe. Model-Driven Software Migration - Process Model, Tool Support and Application. In Anca Daniela Ionita, Martin Litoiu, and Grace Lewis, editors, *Migrating Legacy Applications: Challenges in Service Oriented Architure and Cloud Computing Environments*. IGI Global, Hershey, PA, USA, 2012.

[JMOW13] Jan Jelschen, Johannes Meier, Marie-Christin Ostendorp, and Andreas Winter. A Description Model for Software Evolution Services. In *CoNaIISI*, Cordoba, November 2013. RIISIC.

[1] Q-MIG is funded by the Central Innovation Program SME of the German Federal Ministry of Economics and Technology (Bundesministerium für Wirtschaft und Technologie - BMWi). Funding code: KF3182501KM3.