Predictable software migration

Modernization of software, especially the migration of existing applications to modern technologies, is still a very unpopular subject in the IT sector. Too many migration projects failed, or produced mediocre results after years of project run-time.

The conventional migration approaches, like "off-the-shelve plus customization" and "greenfield engineering", pose a high risk on time management and budgeting. To just ignore the needed modernization of the software is obviously not a clever solution either. Outdated software causes high maintenance costs, and therefore internal cost pressure.

But modernization of software does not have to be unpredictable. A new approach to this subject creates the basis for predictability of migration projects. The key to this new approach lies in technology itself. Common projects involve migration of (by now outdated) C++ to Java. The close proximity between C++ and Java allows splitting the migration into two phases. The first phase consists of a pure language migration, the second re-designs the architecture. By splitting the process there is a possibility to make migration fast, predictable and cost efficient at the same time.

To achieve an actual benefit the language migration has to be done automatically. This automation allows an efficient and economic use of resources during the project. Firstly, the duration of the migration becomes predictable. Secondly, the automation allows for an enormous increase of migrated code quality due to the consistency of changes. This produces readable and maintainable code. During a manual migration, a specific reoccurring problem is usually solved by more than one software engineer, resulting in inconsistencies and low readability and maintainability.

The use of automation in programming language translation has its own challenges. As time is no longer a factor, the communication between the software engineering teams becomes a mission critical factor. The engineers of the old application have to be involved in the migration from the very beginning. Also, the second phase of the migration – re-architecturing the application- may still be an adventure.

Due to new migration technology, software modernization projects do no longer have to be expensive, tedious and unpredictable projects. The smart use of technology can turn the migration of an application into a "normal", cost-efficient and, most importantly, predictable project.