



## INNOVATION REPORT

# Extending the lifetime of software-intensive systems

.....

Project leader: John Pleunis (Philips Healthcare)

***Application of evolutionary software-development techniques, tools and processes can cut total development costs and increase product lifetimes. The SERIOUS project addressed the issues involved in such evolution – a relatively unexplored territory in software engineering. While a series of industrial case histories clearly demonstrated the benefits, a major challenge was found in changing mindsets.***

Software-intensive systems evolve throughout their lifetimes. The overall quality will reach a predefined level in the course of the creation phase, after which the software quality will degrade if no special measures are taken. It is therefore necessary to establish a continuing development plan to prolong the lifetime of the product to overcome this degradation.

A more mature approach is to pay attention to all quality aspects during the complete life cycle of the product. This technique, focused on retaining the software investment, is called evolutionary software development.

### **Facing up to the challenges**

The European software industry faces many challenges. These include increasing market pressure to come up with new products and features faster, and to create product lines with more product differentiation. There is also a growing need to integrate existing functionality into combined systems, and to interconnect products so they can share information.

A new approach is required to software development to meet the increasing pace of developing software assets and their increased complexity. Changing software that has been adapted and modified several times needs different software-engineering skills from developers than required when developing software from scratch.

Increased complexity and size also impacts development costs. For many companies, an increase in costs is only acceptable if system lifetimes can be prolonged. However, this model is no longer valid if the rate at which systems change increases and therefore the costs required to implement these changes also rise.

A more effective way of dealing with system change is essential – making obvious the need for a move towards an evolutionary software-development process. Evolutionary development explicitly takes into account optimisation of the benefits and costs during the whole life cycle, including the phases after delivery of the first product. It gives companies a strategic means to evolve software, guaranteeing increasing product quality during its whole lifetime.

However, even with software evolution, it is important to maintain overall software quality during the development process as evolving systems tend to decrease in quality. Currently, most organisations focus on development of new features without taking into account the consequences of these changes.



## INNOVATION REPORT

Long-term profitability requires that quality aspects, such as maintainability and extendibility, must be clearly addressed in all phases to ensure an optimal life cycle. Focusing on only adding a single feature to the system may cause problems in other, even untouched, areas of the software. To overcome this, anticipatory development activities are needed early-on.

### Developing methodologically-sound techniques

SERIOUS set out to achieve a breakthrough that could turn current practices of handcrafted evolution into methodologically-sound techniques supported by tools that could be adopted by industry – including development processes for the evolution of operational software-intensive systems. Such issues have had little attention before.

The project involved investigating the technology required, addressing the process issues and measuring the quality of the system. Key aspects were:

- **Techniques, tools and models to improve quality aspects related to software evolution** – SERIOUS aimed to improve techniques to integrate software refactoring and software analysis techniques smoothly internally within the day-to-day software process, leading to a true evolutionary software engineering paradigm;
- **Processes and methods to improve software-development models in industry** – the focus in most development models is on the creation phase. Requirements for managing the period after delivery are often not taken into account in the creation phase. SERIOUS aimed to address processes and methods, applicable in industry, which explicitly take into account such evolutionary requirements; and
- **Identification, definition and embedding of software-evolution-related quality aspects** – SERIOUS aimed to define quality aspects in a level of detail that can be applied and/or used in the early phases of software development and continued to be usable in the evolution phase.

Academic techniques, tools and models related to software evolution were applied in a series of industrial case studies to evaluate their usefulness. The results of these studies have been gathered in a handbook on refactoring which is now available publicly.

Furthermore, process and methods that improve the software development models in industry were gathered and applied at industrial partners that deal with evolutionary development. This collection of best practices has been put together in a process patterns handbook which is now also available publicly.

Finally, SERIOUS studied the quality aspects of software. The results were embedded in tools which now allow partners to monitor the internal quality aspects of their systems in real time.

While the main technical challenges involved finding the correct techniques for refactoring an existing system, and measuring the quality of that system during development, a major challenge from an organisational point of view was to change the mindset required for the developers.

### Practical results demonstrated

Many of the case studies carried out resulted in a change in techniques, tools and processes used by the business units where the case studies took place – and beyond.



## INNOVATION REPORT

Highlights included:

- Refactoring according to the handbook patterns helped Alcatel-Lucent ISAM Broadband Access Server to gain over 50% reduction in memory use and 33% start-up time improvement in an important subsystem, extending the lifetime of the product for at least two years;
- The SERIOUS approach helped Philips X-Ray Imaging halve time-to-market for the introduction of new imaging-processing algorithms from one year to six months;
- Implementation of the SERIOUS refactoring approach in the ESI PLUM product line variation management tool results in estimated savings of 50% after the fourth product line product for a customer using the tool to move to a software product line (SPL);
- The SERIOUS approach for refactoring .NET applications to support multiple languages resulted in a doubled adoption rate for an Ibermatics HumanCare application;
- Changed product line road mapping practices at Philips Healthcare are expected to reduce the overall innovation cost for X-ray products by 10% as a result of 70% fewer problem reports in systems integration and a 40% improvement in innovation lead-time; and
- The now widespread practice of earlier testing in combination with refactoring at Alcatel-Lucent results in a change in defect distribution along the verification phases. Defects are detected earlier in the development cycle, and so are cheaper to repair.

The result of SERIOUS will help maximise the long-term value of investments in software through use of evolutionary development models that take into account the whole life cycle of the product. Such an approach will cut costs and increase product lifetimes while boosting the competitiveness of the European software-intensive systems industry.