The Monte Carlo Method: Versatility Unbounded in a Dynamic Computing World
Chattanooga, Tennessee, April 17-21, 2005, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2005)

EXPERIENCE WITH SOFTWARE PROCESS IN PHYSICS PROJECTS

S. Guatelli, B. Mascialino, M. G. Pia, M. Piergentili
INFN - Sezione di Genova
Via Dodecaneso 33, 16146 Genova (Italy)
Susanna.Guatelli@ge.infn.it; Barbara.Mascialino@ge.infn.it;
MariaGrazia.Pia@ge.infn.it; Michela.Piergentili@ge.infn.it

L. Moneta, I. Papadopoulos, A. Pfeiffer
CERN
CH-1211 Geneva (Switzerland)
Lorenzo.Moneta@cern.ch; Ioannis.Papadopoulos@cern.ch;
Andreas.Pfeiffer@cern.ch

ABSTRACT

The adoption of a rigorous software process is well known to represent a key factor for the quality of the software product and the most effective usage of the human resources available to a software project. The Unified Process, in particular its commercial packaging known as the RUP (Rational Unified Process) has been one of the most widely used software process models in the software industry for a number of years. We present the application of the Unified Process and of the RUP to a variety of Monte Carlo simulation projects in High Energy Physics, space science and medical physics environments. We illustrate how the UP/RUP provide a flexible process framework, that can be tailored to the different needs of individual software projects. We describe the experience with different approaches (top-down and bottom-up) to the implementation of the process in software organisations. We document a critical analysis of the effects of the adoption of the UP/RUP, and discuss the relative benefits of the public (UP) and commercial (RUP) versions of the process.

Key Words: Software Process, Software Engineering, Unified Process, Rational Unified Process

1 INTRODUCTION

The software process encompasses the set of activities, methods, practices, and transformations that people use to develop and maintain software and associated products. The process is a key element to ensure that a project matches its defined objectives: it provides a disciplined approach to assigning tasks and responsibilities within a development organization. By defining a discipline for the software development, it plays a fundamental role in achieving software quality and an effective usage of the resources available to a project, with the goal of producing high-quality software that meets the needs of its end users within a predictable schedule and budget. Several software process models have been developed in the past years, and standards for the evaluation of software capability maturity have been defined. The role of software process is well established in the professional software environment as the key to achieve software quality; and has been recognized for several years in physics environments characterized by mission critical software applications, such as space science. However this discipline is relatively new to the domain of particle physics software, and is often perceived with some skepticism in the physics research environment [1]. The Unified Process [2] is one of the most widely adopted process frameworks in the industrial environment. In spite of its large spread in the professional software domain, it is scarcely known in the physics research environment. In this paper we present the first application of the Unified Process in the context of a high energy physics project, and the further experience gained in further applications to a variety of software development projects in various physics research domains. We show how a process instance derived from the Unified Process framework has been tailored to the peculiar characteristics of a physics research environment.

2 THE UNIFIED SOFTWARE FRAMEWORK

The Unified Process for software development is based on proven family of best practices, which supports the complete software development life cycle. It is use-case driven, architecture-centric, iterative, and incremental. The Unified Process takes full advantage of the industry-standard Unified Modeling Language [3], and demonstrates how the notation and process complement one another. The Rational Unified Process or RUP [4] product is a software engineering process based on the Unified Process. The RUP has two dimensions: the first dimension represents the dynamic aspect of the process as it is enacted, and it is expressed in terms of phases, iterations, and milestones; the second dimension represents the static aspect of the process: how it is described in terms of process components, disciplines, activities, workflows, artifacts, and roles. The software lifecycle of the Rational Unified Process (RUP) is decomposed over time into four sequential phases, each concluded by a major milestone. The Inception phase focuses on defining the vision of the product, its business case and the scope of the project. The Elaboration phase focuses on planning the necessary activities and resources, and and designing the architecture. The Construction phase centers on building the product, and evolving the vision, the architecture and the plans. The lifecycle terminates with the Transition phase, in which the software product is delivered to its user's community. The software development consists of a series of iterations, through which the software under development evolves incrementally. Each iteration is concluded by the release of the product, which may be a subset of the complete vision, but useful from some engineering or user perspective. Each release of the code is complemented by a set of supporting artifacts: release description, user's documentation, plans, etc. The static component of the process is articulated into nine disciplines: Business Modeling, Requirements, Analysis & Design, Implementation, Test, Deployment, Environment, Project Management, Configuration and Change Management. Each phase - and each iteration - spans all the disciplines, even if with a different emphasis along the software development lifecycle. Artifacts are work products that are produced and used during a project; they capture and document project information. The RUP can provide a valuable roadmap towards achieving a higher process maturity. It has been demonstrated [5] [6] that it matches the requirements for reaching the levels 2 and 3 defined by the SEI Capability Maturity Model [7] and the ISO/IEC 15504 standard [8]. The Rational Unified Process provides a framework that can be customized to each software development organization's specific needs. This characteristic of the RUP as a flexible, customizable process is particularly relevant for its application in the peculiar environment of a physics research project.

3 THE SOFTWARE PROCESS IN THE PHYSICS RESEARCH ENVIRONMENT

We recall here some peculiar features that characterize a software development process in the physics research environment. With respect to the industry environment, where the organization management has direct control on the staff members developing a software project, the research environment is characterized by a collaborative agreement among various groups, often geographically spread, contributing to a software development, while each individual is usually responsible for her/his research activity. In this context the project management roles and activities largely rely on the goodwill of the team members. Even the adoption of a software process cannot be enforced as a top-down approach in such an environment, but has to be achieved through a consensus of the whole team. Convincing the developers' team of the benefit of adopting a rigorous software project is often difficult: the main justification of the widely spread scepticism is the fact that past experiments achieved successful scientific results without paying much attention to the software development process, or to software in general. Another peculiar feature of the physics research environment with respect to the commercial one consists in the fact that the developers' and users' community largely overlap, or even coincide: physicists develop software for their own usage, or to be used in the experiments they are involved in, rather than software products to be deployed to customers' companies or to the general public. Because of this overlap of roles, disciplines like requirements engineering may not be rigorously addressed, taking for granted that the developers know the needs of the community from their own experience. At the same time, the direct domain knowledge of the software developers facilitates the communication with the software users and the understanding of their requirements, which is often one of the main issues in commercial software development projects. The domain knowledge of the software developers in the physics research environment is seldom complemented by a solid background in software engineering: the software developers are usually physicists, rather than software professionals; therefore an effective software process must address the developers' training with great care. The adoption of a software process is still quite unusual in the physics research environment; a relevant exception is the Geant4 project [9], that has paid great attention to adopting a sound software process since its early stage. In most cases the software process adopted does not follow any of the widely known process models used in the industrial practice; rather it usually consists of a home-made process model, sometimes inspired to a widely spread model or to generic guidance documents, such as the ISO 15504 roadmap for software process improvements.

4 THE PILOT PROJECT: INTRODUCING THE RUP IN ANAPHE

Anaphe [10] was a project in the CERN IT Division for the development of an object oriented analysis tools system, especially addressed to the requirements of the LHC experiments currently under construction. The project was characterized by a sound architecture, and was built to respond to well known user requirements, deriving from the past experience in the scientific community and the foreseen challenging features of the LHC analysis environment. However, the development team did not follow any software process explicitly, and the achievement of the project goals was entirely left to the personal efforts of the individual developers. The project suffered of repeated failures to match the release schedule, and the released versions were often unstable for usage in analysis applications. A software process improvement programme was launched in 2002, having as main goals improving the reliability of the project schedule and the quality of the released software. The Rational Unified Process was adopted as a process framework: to our knowledge, this pilot project represents the first deployment of the RUP in the high energy physics domain. This first project represented first of all an exploration of the capabilities of the RUP to address the peculiar characteristics of a scientific research environment, and an opportunity to gain direct experience in using it. The RUP was first introduced in the Anaphe project when it was already at an advanced development stage: therefore, the conventional dynamic dimension of the process had to be adapted, skipping the Inception and Elaboration phases as such, and jumping directly into the Construction phase. However, since the previous history of the project had neglected some activities and artifacts which were considered essential, the construction phase addressed some key artifacts usually typically produced in the first two phases, Because of the peculiar situation of the project, the pilot project focused in defining and improving the process in the areas considered most critical for the success of the project. However, the adoption of the RUP had a beneficial effect even in the disciplines in which the project already had a sound approach, for instance emphasizing the documentation of the architecture and detailed design, and the traceability of requirements through other artifacts, such as design models and code implementation. The problem of delayed and unstable releases was addressed by following the guidelines of the RUP in the disciplines identified as correlated to such effects. The first iteration of the software process improvement focused on the Project Management discipline. A software development plan was agreed among the team members and documented in detail. A risk list was compiled, identifying the risks the project may encounter and analyzing them in terms of their probability to occur and their foreseen impact. Mitigation strategies were devised for the risks rated in the top rank with respect to the combination of both variables. Among the core disciplines, the focus of the software process improvement addressed especially Implementation and Test: this choice was motivated by the fact that delays in the release schedule were often attributed to the difficulties generated by the integration of the whole software system at a late development stage, often exposing faults in many of the components. In the context of Implementation, some activities were introduced in the process according to the corresponding RUP workflow. The activity “Perform Unit Tests” allowed to trap errors and to correct defects at an early stage of the software development in each class, rather than in the final system integration. The activity “Plan System Integration” was introduced, defining a process and responsibilities for the system integration, rather than leaving it the individual effort of the project manager. The activity “Perform Integration” was refined, based on the outcome of the “Plan System Integration”. A system of nightly builds was set up to address integration problems as early as possible, thus mitigating the risks related to the discovery of integration problems at a critical stage. The risk of releasing unstable, unreliable software was mitigated by improving the software testing according to the RUP guidelines. Feedback from users was recognized as an important input to assess the quality of the product in the development phase. To extend the user community, the guidelines of the RUP in the Deployment discipline were followed, complementing the release of the code with other artifacts, such as training material in the form of documented examples. The effects of the adoption of the RUP were visible within a few months since its adoption, even if the software process improvements had gone through a limited number of iterations and the development case was still largely incomplete. Thanks to the improvements in the Project Management, Implementation and Test disciplines, the delay in the release schedule dropped from the order of a few months to a few days; the quality of the software became sufficiently reliable for Anaphe to be adopted by Geant4 Advanced Examples as the analysis tool used in the certification of their release and recommended to Geant4 users. These first successes provided an encouraging feedback on the suitability of the RUP for application in a physics research project. The RUP pilot project terminated less than one year since its start, when Anaphe was stopped as an independent project and merged into the PI [11] project of the LHC Computing Grid Application Area [12].

5 APPLICATIONS OF A RUP-BASED SOFTWARE PROCESS IN PHYSICS RESEARCH PROJECTS

Following the successful results with the pilot project in Anaphe, the Rational Unified Process has been adopted in several software projects in a variety of physics research environments: Geant4 Low Energy Electromagnetic Physics [13] and Geant4 Electromagnetic Physics Validation [14] (core simulation software projects), the Statistical Toolkit [15] (a mathematical library for data analysis), Brachytherapy Dosimetry System [16] and IMRT Dosimetry System [17] (simulation for oncological radiotherapy), REMSIM [18] (radioprotection in interplanetary missions) and Bepi Colombo simulation [19] (planetary astrophysics). The experience gained in such a large number of projects has allowed to tailor the process to the specific needs of the projects with a good refinement. A subset of artifacts has been identified as essential among the huge set proposed by the RUP, and particularly suitable to the peculiar case of a software process in a physics research environment. The essential artifacts are listed in Table 1. Several metrics are collected along the development process, and represent a valuable reference on which to build a quantitative process control. The benefits of adopting a sound software process, and in particular an instance of the RUP specifically tailored to the project, are evidenced by the success rate of the projects to which it has been adopted. A successful project is defined as one completed on time, delivering all the planned features and functions, respecting the original budget (that in the case of a scientific project coincides with the allocation of human resources). The success rate measured on the physics research projects adopting the RUP is 95%, to be compared to the success rate of 23% reported by the Standish Group [20].

6 CONCLUSIONS

The Rational Unified Process has been adopted for the first time in the field of physics research. A pilot project in Anaphe has demonstrated the benefits it provides to improve the software quality and the respect of the development schedule. The application in a variety of software development projects in various research domains (core physics software, space science, astrophysics, mathematics, oncological radiotherapy) has resulted in a success rate of the projects of 95%.

7 REFERENCES

1. J. Knobloch, M. G. Pia, “Trends in computing,” *IEEE Transactions on Nuclear Science*, **51 (5)**, pp. 2050-2055 (2004).
2. I. Jacobson, G. Booch, J. Rumbaugh, *The United Software Development Process,* Addison

Wesley Longman (1998).

3. G. Booch, I. Rumbaugh, I. Jacobson *UML User Guide,* Addison Wesley Longman (1998).
4. P. Kruchten, *The Rational Unified Process, an Introduction,* Addison Wesley Longman (2000).
5. ‘The Rational Unified Process - An Enabler for Higher Process Maturity,” (2003).
6. “Rational Software Corporation, Assessing the Rational unifed Process against ISO/IEC15504-5: Information Technology - Software Porcess Assessment Part 5: An Assessment Model And Indicator Guidance,” (2000).
7. M. Paulk et al. “Capability Maturity Model for Software, Version 1.1. Software Engineering Institute, Pittsburgh, PA SEI-93-TR-024,” (1993).
8. “ISO/IEC Joint Technical Committee 1 (JTC1), ISO/IEC DTR 15504 Software Process Assessment.,” (2003).
9. G. Cosmo, “Geant4 software process,” *Proceeding of CHEP 2001 Conference,* Beijing China, September 2001, pp. 469-472 (2001).
10. O. Couet, B. Ferrero-Merlino, Z. Molnar, J. T. Moscicki, A. Pfeiffer, M. Sang, “ANAPHE-OO Libraries and tools for data analysis,” *Proceeding of CHEP 2001 Conference,* Beijing China, September 2001, pp. 219-222 (2001).
11. A. Pfeiffer et al., “The Physicist’s Interface Project,” *Proceeding of IEEE NSS 2004 Conference,* Roma Italy, October 2004, (2004).
12. A. Pfeiffer et al., “Overview of the LCG Application Area,” *Proceeding of IEEE NSS 2004 Conference,* Roma Italy, October 2004, (2004).
13. S. Chauvie et al., “Geant4 Low Energy Electromagnetic Package,” *Proceeding of IEEE NSS 2004 Conference,* Roma Italy, October 2004, (2004).
14. K. Amako et al., “Validation of Geant4 Electromagnetic Physics against Protocol Data,” *Proceeding of IEEE NSS 2004 Conference,* Roma Italy, October 2004, (2004).
15. G. A. P. Cirrone, S. Donadio, S. Guatelli, A. Mantero, B. Mascialino, S. Parlati, A. Pfeiffer, M. G. Pia, A. Ribon, P. Viarengo, “A Goodness-of-Fit Statistical Toolkit,” *Transactions on Nuclear Science,* **51 (5)**, pp. 2056-2063 (2004).
16. S. Guatelli et al., “Technology transfer from HEP computing to the medical field: overview and application to dosimetry,” *Proceeding of 9th Topical Seminar on Innovative Particle and Radiation Detectors Conference,* Siena Italy, May 2004, (2004).
17. F. Foppiano et al., “A Geant4-Based Simulation of an Accelerator’s Head Used for Intensity Modulated Radiation Therapy,” *Proceeding of IEEE NSS 2004 Conference,* Roma Italy, October 2004, (2004).
18. S. Guatelli et al., “Dosimetry for Interplanetary Missions,” *Proceeding of IEEE NSS 2004 Conference,* Roma Italy, October 2004, (2004).
19. A. Mantero et al., “A Library for Simulated X-Ray Emission from Planetary Surfaces,” *Proceeding of IEEE NSS 2004 Conference,* Roma Italy, October 2004, (2004).
20. “The Standish Group International Inc., Extreme Chaos,” (2001).

Table I: Essential artifacts

Phase	Artifacts
Inception	Vision Risk List Development Plan Development Case CVS Repository
Elaboration	User Requirements Architecture Model Design Model Architecture Model Test Plan Test Suite Prototype Tools Guidelines
Construction	The system Documentation
Transition	The product Release Notes Training Material