Software Process Improvement in Geant4

Gabriele Cosmo
CERN IT-API/SI
Gabriele.Cosmo@cern.ch

Outline

- The Geant4 Project
- Software Processes
- Improvement Strategy
- The SPICE ISO/IEC-15504-5 model
- Applicability to Geant4
- The Geant4 approach
- Conclusions

The Geant4 Project

- Started as CERN R&D project in December 1994
  - **Aim**: realize a software toolkit for Simulation in HEP based on modern Object-Oriented Software Engineering techniques and methodologies, Programming Languages and International Standards.
- Evolution:
  - April 1997: first *alpha* release (R&D)
  - July 1998: first *beta* release (R&D)
  - December 1998: first public *Production* release

The Geant4 Project - organization

- A world-wide Collaboration
  - Collaboration of more than 100 scientists from over 40 Institutions and Laboratories, participating in more than 10 experiments world-wide
  - Applications ranges from HEP to low-energy and nuclear Physics, Astrophysics, Medical application.
  - A MoU (Memorandum of Understanding) defines all terms of the Collaboration.

The Geant4 Project - distributed development

- More than 1200 classes distributed in 17 Categories (Software components in the Booch terminology)
- Hierarchical structure of complex Categories
- Development teams organized according to domain Category definition, from the design Category diagram
- Centralized coordination of domain Categories
  - domain decomposition <> geographical location
  - assignment of responsibilities and Support
- Distributed resources and funds
- Need for: homogeneous computing environment, methods and tools

Software Processes

- ***Primary Life Cycle***
- ***Supporting Life Cycle***
- ***Management Processes***
- ***Organizational Life Cycle***
- ***User-supplier Processes***

- Development
  - System Requirements Analysis and Design
  - Software Design
  - Software Construction
  - Software Integration

- Documentation
- Configuration Management
- Quality Assurance
- Testing
- Verification & Validation
- Joint Review
- Problem Resolution
- Project tasks Management
- Risk Management
- Improvement Process
- Process Establishment
- Human resource Management
- Infrastructure
- User Support, Distribution

Software Processes - elements for Improvement

- Process establishment
  - Identify current roles and responsibilities
  - Assess currently performed Processes
  - Agree on a strategy for changing/tailoring Processes
- Process improvement
  - Identify purposes, goals and priorities
  - Define measures to quantify impact of improvement
- Process assessment
  - See next slide...

Software Processes - process assessment

- Define an assessment method
- Identify the scope of the assessment
- Plan the assessment for each individual component
- Validate the retrieved information
- Identify strong and weak areas
- Archive and version the results
- Identify priorities for improvement from the final assessment's ratings

SPICE ISO/IEC-15504-5

Improvement Strategy

- Adopt well defined process models to address Software Process issues: SPICE, CMM, …
- Do not focus only on technical issues !
- Software Development: *a knowledge intensive industry*
  - Quality of products embedded in the knowledge of the staff
  - Direct relation between:
    - Quality of products
    - Processes producing them
    - People performing processes

Improvement Strategy - goals

- **Process Effectiveness**
  - activities performed in the Process are adequate to produce the desired results (Process compliance, flexibility)
- **Process Stability**
  - reduce performance variation, to allow a Process to behave in a predictable way (Process control, support, training)
- **Process Efficiency**
  - optimize the amount of resources needed to achieve the required outcomes (Process improvement, automation)
- **Process Capability**
  - produce predictable results in a predictable manner (Process maturity, organizational alignment)

The SPICE ISO/IEC-15504-5 model

- Since 1993, SPICE (Software Process Improvement and Capability dEtermination) developed a standard framework for Software Process assessment within ISO (International Organization for Standardization)
- It proposes 6 levels of maturity (*capability levels*) from “Incomplete” to “Optimizing”:
  - Each level characterizes the level of understanding and control that the Process is being carried out
  - It represents a set of *co-working* attributes providing a major enhancement of capability in the performance of a Process
  - Levels: *Incomplete, Performed, Managed, Established, Predictable, Optimizing*

Organizational alignment

- Use de-facto standard certified channels for software Improvement
- Consult external projects and organizations to learn strengths and weaknesses of adopted solutions for software development
- Allow adoption of key software technologies aligned with tools and products available in the organization
- Promote training and innovation in software technology

Applicability to Geant4

- Last Software Process assessment applied to the Geant4 project: October 1998 (SPICE model)
- Need to understand and determine applicable procedures to software development and maintenance in the “production” phase of the software product
- Complexity factors
  - Different applicability levels for different Category domains
  - Distributed development teams and resources
  - Complex coordination and control for *support* activities
  - Dynamic environment
  - Limited manpower

The Geant4 approach

- Consider Process Improvement as a gradual process
  - Identify the key areas needing Improvement (level 3)
  - Avoid too much formality: weaknesses also identified through experience in the organization
  - Allow for a continuous Improvement, life-cycle driven
- (Chosen) Domains of applicability in Geant4:
  - Q/A & Optimization activity
    - applied to the software product in either global and component domain related context
  - Analysis & Design software cycle
    - identify the well established OOP procedure for development and maintenance
  - Testing
    - assure constant improvement and continuity to system testing

The Geant4 approach - Q/A & Optimization

- By adoption of specialized tools and scripts:
  - Monitoring of dynamic memory allocation applied to test-bed applications
  - Performance monitoring and profiling
  - Source code filtering for conventions and coding rules violations
  - Source code filtering for metrics analysis
  - Test coverage analysis on test-bed applications
- Deploy "global context" activity to a specialized team
  - not involved in development
  - in coordination with the System Testing team
  - based on *mutual trust* with developers and Coordinators
- Improve automation: integrate with tools for testing

The Geant4 approach - Analysis & Design cycle

Goal: *guarantee that the code quality will not degrade with time. Assure a coherent development where coupling will not increase with the complexity of the software*

- Periodically review the global category diagram
  - check for violations/changes and additions
- Actions to be performed by Category Coordinators
  - periodically review URD, possibly starting from “use cases”
  - review/identify areas where A&D software cycle need to be applied
  - review consistency of code with design
  - supervise Category activity and organize training
- Collect architectural/detailed design and URD documents and define a clear procedure for maintenance and update

The Geant4 approach - System Testing

- Improvement of system and validation tests
  - establish clear responsibility for maintenance and integration of tests in the normal development process
  - review and properly document tests; check correspondence with URD and use-cases
  - adopt/improve regression and statistical tests
- Automation
  - adoption of *Bonsai* to automate testing activity and CVS tags submission through Web
  - adoption of *LXR* for online browsing of code through Web
  - adoption of *Tinderbox* to allow developers and testers to monitor progress of system tests and allow distributed control
  - integrate Q/A automation to provide developers a way to perform basic Q/A checks on code before submitting to test

Conclusions

- ◆ Geant4: a challenging project where to apply a Software Process Improvement (SPI) program
- ❶ Use experience and expertise to identify the correct actions to apply for SPI
- ❷ Identify the key-areas/domains where SPI needs to be applied: through a de-facto standard assessment model (SPICE ISO/IEC 15504-5)
- ❸ Keep in mind goals of SPI and don’t focus only on technical issues