

Software Process Improvement a common approach in IT/API

Gabriele Cosmo

CERN IT/API-SI

Gabriele.Cosmo@cern.ch

G.Cosmo - Software Process Improvement: a common approach in IT/API

Outline

- Common objectives
- Common methods
- SPI: current status
 - GEANT4
 - ANAPHE
- SPI: actions planned for year 2001
- Conclusions

CERN, Geneva - March 2001

2

G.Cosmo - Software Process Improvement: a common approach in IT/API

Common objectives for SPI

- Understand, determine and establish applicable procedures to Software development and maintenance of the software products.
- Make SPI a Software Process *life-cycle driven*.
 - ⇒ Primary life-cycle processes:
 - guarantee that the code quality will not degrade with time: apply SPI actions associated with a regular QA activity
 - assure that coupling will not increase with the growing complexity of the software
 - ⇒ Improve overall usability and robustness of applications: improve quality, maintainability and reliability of the code.
 - ⇒ Assure continuity and integration of regular system testing within the normal Software development activity.

CERN, Geneva - March 2001 3

G.Cosmo - Software Process Improvement: a common approach in IT/API

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)

CERN, Geneva - March 2001 4

Benefits ?

- Theory says that without proper engineering:
 - ① Results may not be exactly those one expects
 - ② Results obtained are not worth the time spent to achieve them and spent to reiterate always on the same problems

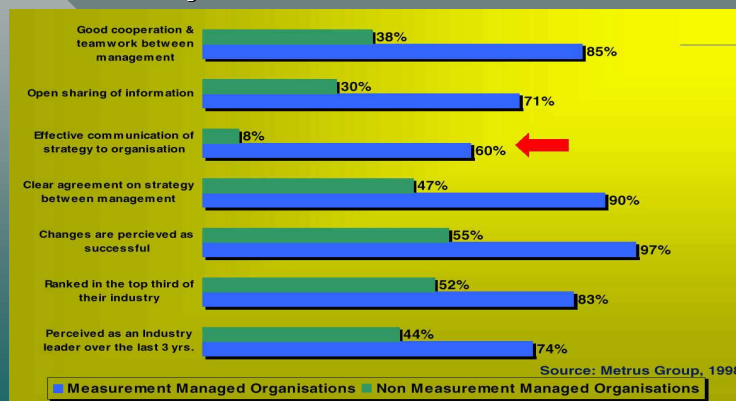


CERN, Geneva - March 2001

5

Benefits ? Some examples...

- expected results are:
 - reduction of total overrun time
 - reduction of financial loss caused by overrunning cost estimates and missing milestones



CERN, Geneva - March 2001

(*) from ESI (European Software Institute) Balanced IT Scorecard

6

G.Cosmo - Software Process Improvement: a common approach in IT/API

Software Processes

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

- **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**

Development

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

CERN, Geneva - March 2001 7

G.Cosmo - Software Process Improvement: a common approach in IT/API

Common approach & methods

- Adopt a well defined process model to address Software Process issues: SPICE, CMM, ...
- Choose a limited set of domains where to apply an assessment and a SPI program.
 - Identify current roles and responsibilities.
 - Identify purposes, goals and priorities.
- Agree on a strategy for changing/tailoring Processes
 - Gradually apply SPI actions to concerned areas.
- Do not focus only on technical issues !
 - Quality of products embedded in the knowledge of the staff
 - Direct relation between: Quality of products, Processes producing them, People performing processes.
- Define measures to quantify impact of improvement
 - Consider monitoring progress of the SPI program and iteration along with Software life-cycle.

CERN, Geneva - March 2001 8

G.Cosmo - Software Process Improvement: a common approach in IT/API

Organizational alignment

- Use de-facto standard certified channels (process models) 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

CERN, Geneva - March 2001 9

G.Cosmo - Software Process Improvement: a common approach in IT/API

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

CERN, Geneva - March 2001 10

G.Cosmo - Software Process Improvement: a common approach in IT/API

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 "0 - Incomplete" to "5 - Optimizing":
 - Each level characterizes the level of understanding and control by which the Process is performed
 - 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*

CERN, Geneva - March 2001 11

G.Cosmo - Software Process Improvement: a common approach in IT/API

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: 2 levels
- Distributed resources and funds
- Needs for development: homogeneous computing environment, methods and tools

CERN, Geneva - March 2001 12

G.Cosmo - Software Process Improvement: a common approach in IT/API

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

CERN, Geneva - March 2001 13

G.Cosmo - Software Process Improvement: a common approach in IT/API

The Geant4 strategy

- 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

CERN, Geneva - March 2001 14

G.Cosmo - Software Process Improvement: a common approach in IT/API

The Geant4 strategy 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 written procedures and *mutual trust* with developers and Coordinators
- Improve automation: integrate with tools for testing

CERN, Geneva - March 2001 15

G.Cosmo - Software Process Improvement: a common approach in IT/API

Geant4: Q/A & Optimization - activities

- Adoption of specialized tools and scripts
 - Monitoring of dynamic memory allocation
 - (*improve overall usability and robustness of the applications at run-time*)
 - ⊗ identify test-bed applications
 - ⊗ define responsibilities (global context: associated to Release)
 - ? Performance monitoring and profiling
 - (*improve run-time performance and avoid redundancy*)
 - ⊗ identify test-bed applications, define responsibilities
 - Source code filtering for conventions and coding rules violations
 - (*improve quality and reliability of the code*)
 - ⊗ implement project's specific coding rules to apply to scripts/tool
 - ? Source code filtering for metrics analysis
 - (*improve quality, maintainability and portability of the code*)
 - ⊗ define responsibilities and methods, identify tools
 - ? Test coverage analysis
 - (*improve quality of testing*)
 - ⊗ identify test-bed applications
 - ⊗ identify tools and methods (global context: associate to Testing ?)

CERN, Geneva - March 2001 16

G. Cosmo - Software Process Improvement: a common approach in IT/API

Geant4: Q/A & Optimization - actions

- Creation of Q/A specialized team (2 people, $\geq 30\%$)
 - “global context” activity deployed to the team
 - activity possibly independent from development
 - activity in coordination with the System Testing & Release Teams
 - Team’s responsibilities:
 - identify Q/A tools, also considering availability & resources
 - select *test-bed* applications and care for maintenance/upgrade in collaboration with STT and Category Coordinators
 - perform a complete analysis every 1-2 months and delegate to Category Coordinators fixes to the code
 - identify coding rules and implement them through scripts/tool
- Identify resources (tools/people) from external groups in the Collaboration
- Improve automation: integrate with tools for testing

CERN, Geneva - March 2001 17

G. Cosmo - Software Process Improvement: a common approach in IT/API

The Geant4 Strategy: 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

- General Actions:
 - Periodically review the global category diagram
 - check for violations/changes and additions
 - inform TSB of any architectural change for new developments
 - 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

- Currently in place: CVS tree *documents* on AFS
- Notify SW-Management Coordinator for new diagrams/docs on Web

CERN, Geneva - March 2001 18

G.Cosmo - Software Process Improvement: a common approach in IT/API

Geant4: OOAD - assessment results

- Improve traceability of requirements with design and test-cases
- Promote internal training (books, monitoring, ...)
 - requirements ↔ design ↔ implementation cycle
 - design methodology and CASE tools
- Provide/adopt tools for work-flow management, size & effort estimation
 - training for effective usage of the tools
- Define standards for design documents to be provided and published AND maintain them !
 - CVS repository for design documents (architectural/detailed sources, URD, specifications, ...)
- Adopt *change management* for design faults/updates during development/maintenance

CERN, Geneva - March 2001 19

G.Cosmo - Software Process Improvement: a common approach in IT/API

The Geant4 strategy: System Testing

- Improvement of system and validation tests
 - establish clear responsibility for maintenance and integration of tests in the normal development process
 - formalized in release&testing procedures document
 - keep updated list with descriptions
 - review and properly document tests; check correspondence with URD and use-cases
 - identify most common use-cases
 - create map for testing coverage and keep it updated
 - involve category coordinators
 - adopt/improve regression and statistical tests
 - establish methods and tools
 - provide a clear time-table for deliverables

CERN, Geneva - March 2001 20

G.Cosmo - Software Process Improvement: a common approach in IT/API

Geant4: System Testing - automation

- adoption of *Bonsai* to automate testing activity and CVS tags submission through Web
 - system already in production (see our *tag-database*)
- adoption of *LXR* for online browsing of code through Web
 - first prototype implemented and available (KEK, TRIUMF)
- adoption of *Tinderbox* to allow developers and testers to monitor progress of system tests and allow distributed control
 - first working prototype already available from TRIUMF
- integrate Q/A automation to provide developers a way to perform basic Q/A checks on code before submitting to test
 - implemented scripts for automating CW code filtering

CERN, Geneva - March 2001 21

G.Cosmo - Software Process Improvement: a common approach in IT/API

The ANAPHE class library

- More than 200 classes distributed in 15 Packages
 - I/O and foundation classes
 - Histogramming, minimizing/fitting
 - Visualization
 - Interactive analysis
- Adoption of *de-facto* Standards
 - strict collaboration with external providers and projects: RD45 (object persistency), NAG (numerical libraries), Qt (custom 2D graphics), TGS (basic 3D graphics), Geant4, ...
- Requirements elicitation and support for a wide user community (HEP)

CERN, Geneva - March 2001 22

G.Cosmo - Software Process Improvement: a common approach in IT/API

ANAPHE: the strategy

- Choose ISO/IEC 15504 Standard as model for addressing Software Process Improvement
- Areas of application of SPI:
 - OO Analysis & Design Software cycle
 - Testing
 - Quality Assurance
 - Documentation

CERN, Geneva - March 2001 23

G.Cosmo - Software Process Improvement: a common approach in IT/API

ANAPHE: OOAD Software cycle

- Based on UML engineering methodology
 - *Component based* - communication defined through abstract interfaces
 - *Use Case driven* - system modeled according to usage scenarios
 - *Architecture centric* - system structure defined by its most important characteristics
 - *Iterative & incremental* life-cycle model
- ⇒ Area of improvement: *Software construction*
 - Improve the current production system
 - Building environment and automation, establish procedures
 - Evaluate alternative Software release tools

CERN, Geneva - March 2001 24

G.Cosmo - Software Process Improvement: a common approach in IT/API

ANAPHE: Testing & QA

- Improvement of *Unit* and *System* testing
 - ⇒ Establish and document procedures:
 - Make *Unit* testing a pre-condition for every official build of a package
 - Make *System* testing a pre-condition for every new release of the Software
- Quality Assurance
 - ⇒ Define QA test suite, establish procedures
 - Make QA a pre-condition for every official build of a package

CERN, Geneva - March 2001 25

G.Cosmo - Software Process Improvement: a common approach in IT/API

ANAPHE: Documentation

- ⇒ Generation of User Manuals
 - Improve automation and consistency between code and documentation, by adopting specific tools
 - *Doxygen* for Software Reference Manual
 - *DocBook* for User Guides
 - Assure coherent and uniform formatting style, by defining appropriate *templates*

CERN, Geneva - March 2001 26

G.Cosmo - Software Process Improvement: a common approach in IT/API

Conclusions

- Geant4, ANAPHE: challenging projects where to apply a Software Process Improvement (SPI) program
- ① Use experience and expertise to identify the correct actions to apply for SPI
- ② Identify the key-areas/domains where SPI needs to be applied. Use a de-facto standard assessment model (SPICE ISO/IEC 15504-5) wherever needed
- ③ Keep in mind goals of SPI and don't focus only on technical issues

CERN, Geneva - March 2001 27

G.Cosmo - Software Process Improvement: a common approach in IT/API

Conclusions - 2

- Apply SPI actions to three different domains:
 - Q/A and Optimization, Analysis & Design cycle, Testing (GEANT4)
 - Q/A, Analysis & Design cycle, Testing and Documentation (ANAPHE)
- Monitor progress of SPI program: use a common approach
 - iterate new assessments in future
 - extend assessment to uncovered (or partially covered) domains (testing, documentation, Software Management)
 - try improving Capability levels
 - try to reach Capability Level 3 assessment ?
- Assign manpower for future SPI activities
 - member in the Collaboration/Group
 - external consultancy

CERN, Geneva - March 2001 28

G.Cosmo - Software Process Improvement: a common approach in IT/API

References

- ANAPHE:
 - <http://cern.ch/proj-anaphe-doc/softproc>
- GEANT4:
 - http://cern.ch/geant4/milestones/software_process

CERN, Geneva - March 2001 29