Software Process Improvement review & plans

Gabriele Cosmo

CERN IT-API/SI

Gabriele.Cosmo@cern.ch

Outline

- Review and plan activities:
  - Q/A & Optimization
  - Analysis & Design cycle
  - Testing
- Conclusions

The approach

- (Chosen) Domains of applicability in Geant4:
  - Q/A & Optimization activity
    - apply Q/A 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

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 ?)

Q/A & Optimization - actions

- Creation of Q/A specialized team (2 people, ≥30%)
  - "global context" activity deployed to the team
  - activity possibly independent from development
  - activity in coordination with the System Testing & Release Teams
  - activity based on *mutual trust* with developers and Coordinators
  - Team's responsibilities:
    - identify Q/A tools, also considering availability & resources
    - select *testbed* 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

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: tree *documents* in `/afs/cern.ch/sw/geant4/cvs`

→ Notify SW-Management Coordinator for new diagrams/docs on Web

Analysis & Design cycle - actions

- Improve traceability or 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

System Testing - tests

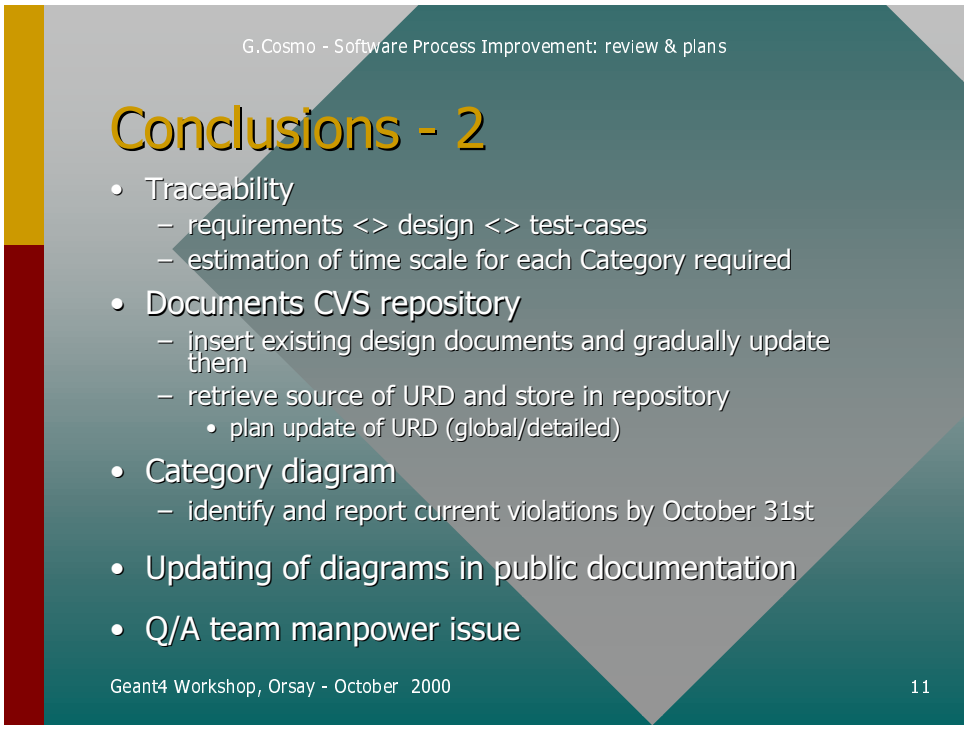
- Improvement of system and validation tests
  - → establish clear responsibility for maintenance and integration of tests in the normal development process
    - → formalized in release&testing procedures document
    - → keep updated list with descriptions
  - review and properly document tests; check 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

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, TRIUMPH)
  - → add links from central Geant4 web
- → adoption of *Tinderbox* to allow developers and testers to monitor progress of system tests and allow distributed control
  - → first working prototype already available from TRIUMPH
- integrate Q/A automation to provide developers a way to perform basic Q/A checks on code before submitting to test
  - implementation of scripts for automating CW code filtering in progress (E.Medernach)

Conclusions

- Apply SPI actions to three different domains:
  - Q/A and Optimization, Analysis & Design cycle, Testing
- Monitor progress of SPI program
  - iterate new assessments in 2001
  - extend assessment to uncovered (or partially covered) domains (testing, documentation, Software Management)
  - try improving Capability levels (try to reach Capability Level 3 assessment)
- Establish clear commitments and deadlines for the implementation of proposed SPI actions
  - Discussion here at the Category Coordinators meeting


G.Cosmo - Software Process Improvement: review & plans
Conclusions - 2
• Traceability
– requirements <> design <> test-cases
– estimation of time scale for each Category required
• Documents CVS repository
– insert existing design documents and gradually update them
– retrieve source of URD and store in repository
• plan update of URD (global/detailed)
• Category diagram
– identify and report current violations by October 31st
• Updating of diagrams in public documentation
• Q/A team manpower issue
Geant4 Workshop, Orsay - October 2000
11