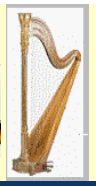





Geant 4

Software Process: Physics

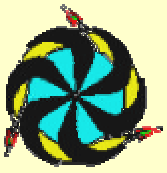
Maria Grazia Pia

INFN Genova

on behalf of the Geant4 Collaboration

Geant4 Review

CERN, 18-23 June 2001



Budker Inst. of Physics
IHEP Protvino
MEPHI Moscow
Pittsburg University
Univ. of Helsinki
Univ. of Barcelona

Outline

- User Requirements
  - Capture
  - Specification, Analysis, Maintenance
  - Approval, prioritisation
- OOAD
- Testing and validation
- Traceability
- SPI

- *Various methodologies followed by the Physics WGs*
- *Adapted to the scope of each WG, its management etc.*
- *Also adapted to the user communities addressed*

A rigorous software process is applied

- in support of a better quality of the software
- especially relevant in the physics domain

UR: capture

Various methodologies adopted

- Elicitation through interviews and surveys
  - Useful to ensure that UR are complete and there is wide agreement
- Joint workshops with user groups
- Direct requests from users to WG coordinators or members
- Use cases
- Analysis of existing Monte Carlo codes
- Study of past and current experiments
- Prototyping
  - Useful especially if requirements are unclear or incomplete
  - Prototype based on tentative requirements, then explore what is really wanted

- Not only functional requirements, users also ask for
  - Documentation
  - Proof of validation of the physics models and their implementation
  - Examples of application in real-life set-ups
- The requirements derive from many sources, in diverse domains
  - HEP, astrophysics, space, medical etc.

User requirements evolve

...and we should be able to cope with their evolution!

UR: specification, analysis and maintenance

An example

Specification: PSS-05 standard

- **UR 4.1** The user shall be able to simulate polarised Compton scattering.
- **Need**: Essential
- **Priority**: Needed by end 2001
- **Stability**: Stable
- **Source**: INFN-Argentinian telescope, UNH
- **Clarity**: Clear
- **Verifiability**: To be verified

Analysis: in WG workshops

Maintenance:
under configuration management,
in CVS repository

GEANT4 LOW ENERGY ELECTROMAGNETIC PHYSICS

Posted on the WG web site

User Requirements Document

Status: in CVS repository

Version: 2.4
Project: Geant4-LowE
Reference: LowE-URD-V2.4
Created: 22 June 1999
Last modified: 26 March 2001
Prepared by: Petteri Nieminen (ESA) and Maria Grazia Pia (INFN)

Other methodologies adopted
eg. Hadronic Physics: UR described in a publication

UR approval, prioritisation

Internal requirements

- Affect a single physics WG, not other categories
- Handled within the WG
  - WG coordinator decides
  - Decided in WG meetings
  - Informal discussions within the WG
- Then reported to the TSB
  - Usually as part of the WG plans for a Geant4 release
- Recorded
  - In WG public documents
  - In WG internal documents
  - In WG coordinator's private documents

Collaboration-wide requirements

- Affect more than one class category, or relevant architectural issues
- Discussed in the TSB
- Recorded in the TSB minutes

OOAD

Spiral approach

The life-cycle model adopted for most domains in Geant4 is both iterative and incremental

↓ *especially relevant in the physics domain!*

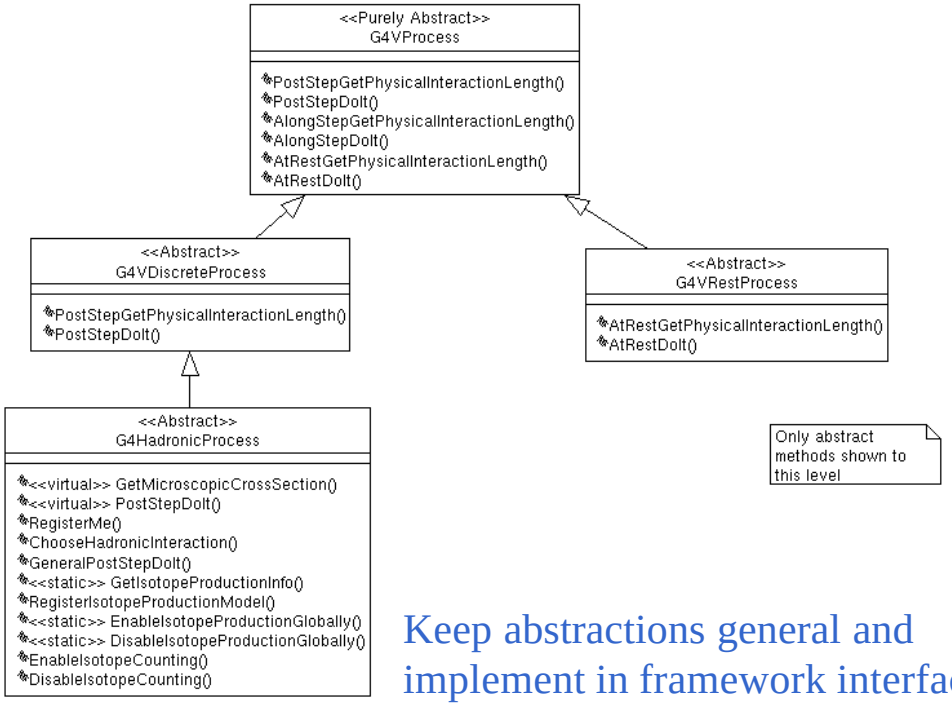
Booch methodology for OOAD

UML notation

– has been chosen as the common *language* for documentation of designs and internal design reviews
– old documents in Booch notation are being progressively updated and converted to UML

Rational Rose CASE tool

– extensively used for the initial generation of design documents
– regularly used as part of the software life-cycle by some WGs
– where required, also for reverse engineering
(Hadronic Physics, Standard EM, LowE EM initially)



Keep abstractions general and implement in framework interfaces

Design process

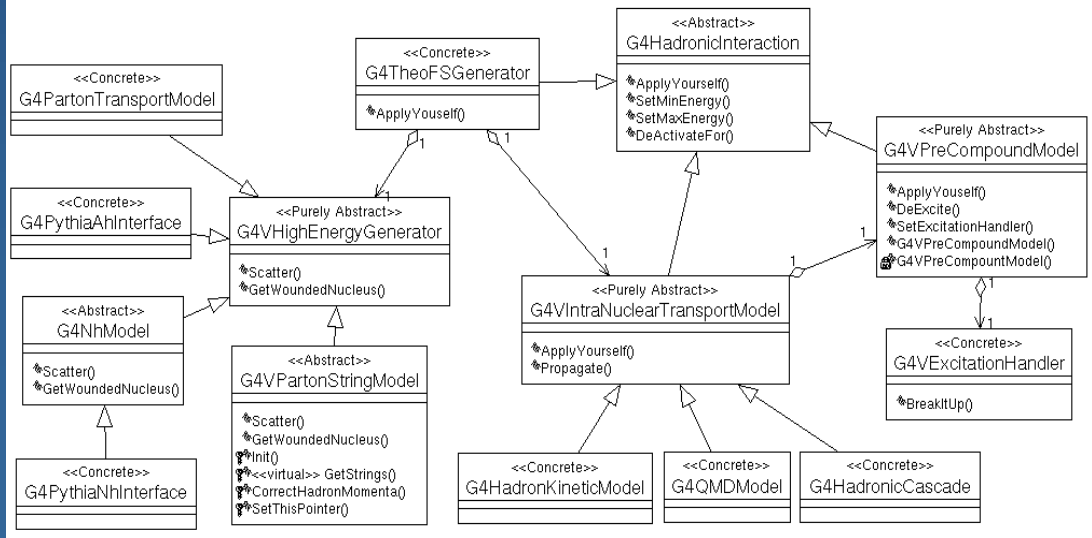
Framework functional requirements are obtained through use-case analysis

Framework components are found through grouping use-cases into independent bundles

An example of OOAD in the physics domain

A *Russian dolls* approach to framework design

Address more specific use-cases in specialized frameworks, that are implementing the interfaces of the more general frameworks



Maria Grazia Pia, INFN Genova

Testing & QA

Various levels of testing

- Unit testing
- Cluster testing
- System/integration testing
- Acceptance testing
- Physics validation

Other tools and methods to improve quality:

- Automated code checking *(Code Wizard)*
- Code reviews
- Defect analysis and prevention

Documentation of procedures, essential to a rigorous software process →

Posted on the WG web site

Version 2

27 May 2001

The Role of Testing in the Software Process of the Geant4 Low-Energy Electromagnetic Physics Working Group

P. Nieminen and M.G. Pia

1 Introduction

Testing forms a vital part of the software process in developments as advanced and complex as those currently in progress in the Geant4 Low-Energy e-m physics Working Group. The purpose of this document is to outline the procedures to be followed regarding testing both during development of new software, and during updates and corrections to existing code.

2 Testing objectives and goals

The objective of testing is to ensure the new, or updated, code performs as intended. Testing should reveal any potential deviancies from expected behaviour of the code both from physics and performance point of view. The goal is high-quality code ready for public release, ultimately leading to easier maintenance and substantial timesaving for developers in the course of the software lifecycle.

3 Test designs and testing schedules

3.1 Test requirements

1. Testing should be performed according to agreed and documented procedures.
2. Traceability through requirements-design-implementation-tests should be implemented.
3. The design should be tested for satisfying the user requirements.
4. The code implementation should be tested for compliance with the design.
5. The code should be tested for correct functionality.
6. The code should be tested for compliance with Geant4 coding guidelines.
7. The code should be tested for satisfactory quality, clarity and readability.
8. Every class of the lowenergy category shall be exercised in an appropriate system test (directly or indirectly).
9. The code should be tested on all Geant4 supported platforms.
10. The code shall be submitted to the entire set of tests above to be considered for release.
11. Tests and test tools should be documented.
12. The test code should be kept under configuration management (in Geant4 CVS repository).
13. Reference outputs, data sets for validation tests etc. should be kept in appropriate agreed locations, accessible to the whole WG.
14. Test tools should be maintained.
15. Modifications of the tests (including test tools, reference outputs, data sets etc.) should be performed according to agreed and documented procedures.
16. The most recent test results should be made available to WG coordinators for code to be included in a monthly global tag or in a Geant4 public release, according to the guidelines described in the "Testing process" section.

Unit tests

Examples

```
/afs/cern.ch/user/p/pia/Vol4/geant4/source/processes/electromagnetic/lowenergy/test
MGsungeant > ls *.cc
G4BremsstrahlungTest.cc              G4PhotoelectricTest.cc
G4ComplexTest.cc                     G4RayleighTest.cc
G4ComptonTest.cc                     G4StoppingPowerTest.cc
G4LowEnergyGammaConversionTest.cc    G4eIonisationTest.cc
G4LowEnergyPolarizedComptonTest.cc   G4hLowEnergyTest.cc
G4LowEnergyTest.cc                   G4hTestLossTableProduction.cc
G4MeanFreePathTest.cc                G4hTestStoppingPower.cc
```

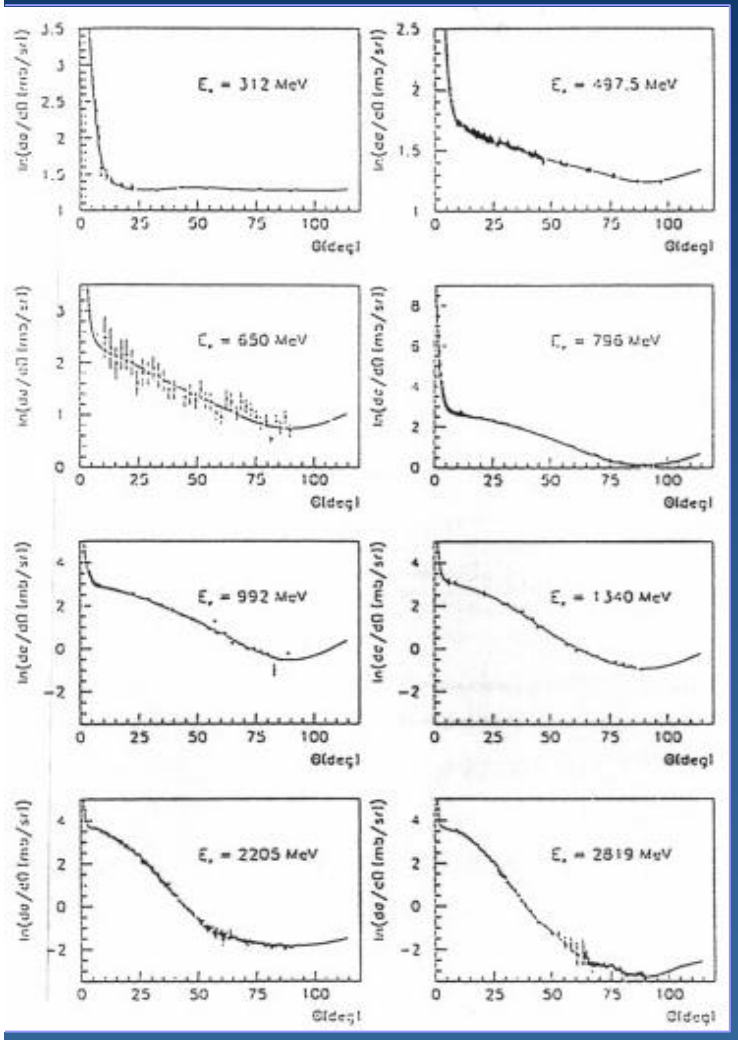
```
/afs/cern.ch/user/p/pia/Vol4/geant4/source/processes/electromagnetic/test
MGsungeant > ls
CVS      History  TestEm0  TestEm1  TestEm2  TestEm3  TestEm4  TestEm5
```

```
/afs/cern.ch/user/p/pia/Vol4/geant4/source/processes/hadronic/models/generator/de_excitation/test
MGsungeant > ls *.cc
EvaporationTest.cc              G4ExcitationHandlerTest.cc      G4PhotonEvaporationTest.cc
G4CompetitiveFissionTest.cc     G4GammaDeexcitationTest.cc
G4EvaporationProbabilityTest.cc G4NuclearLevelManagerTest.cc
```

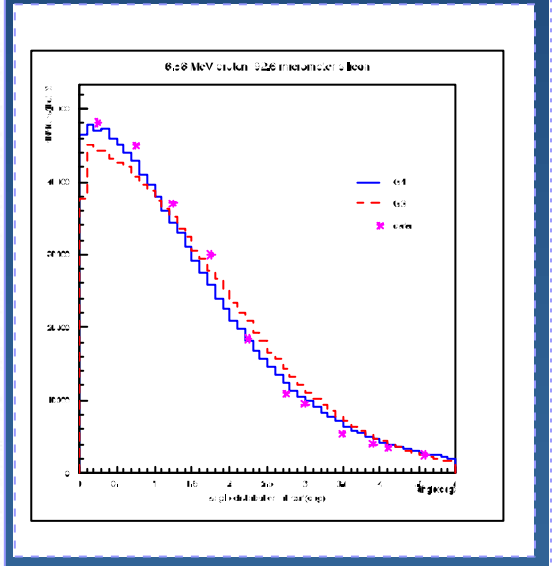
Validation

- Comparison with experimental data
- At various levels: details of physics models and global features

Some examples

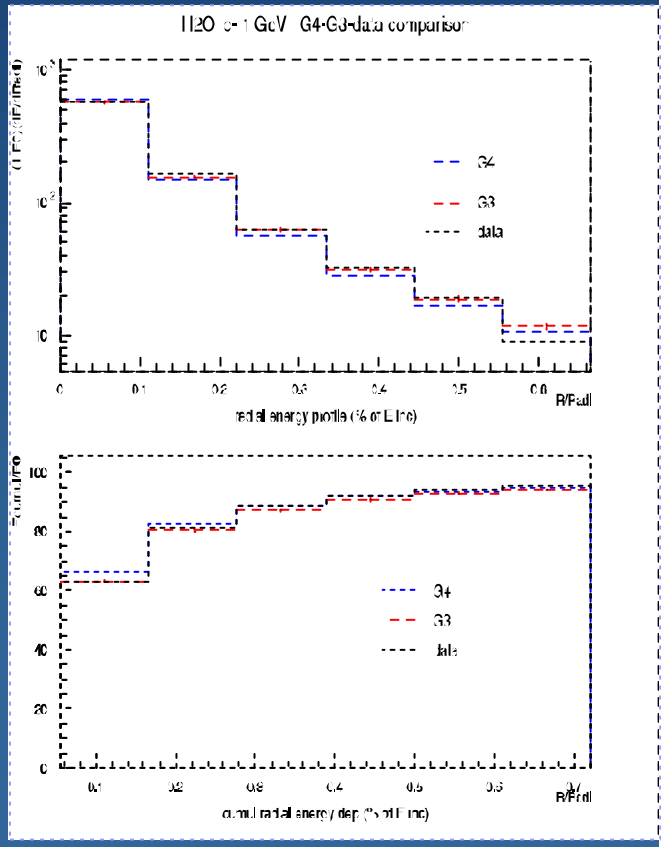


Multiple scattering



p elastic scattering on hydrogen

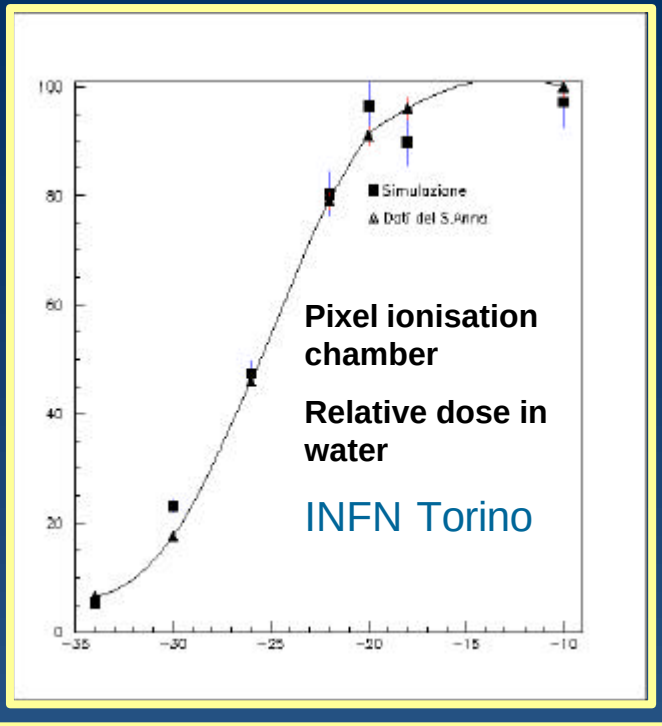
Shower profile

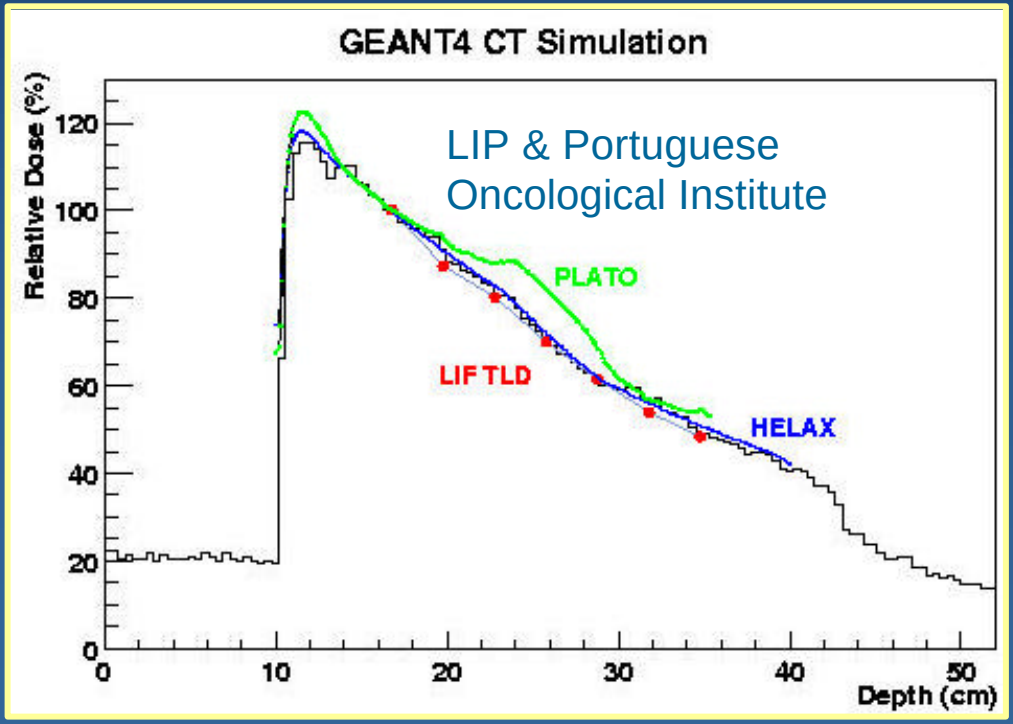


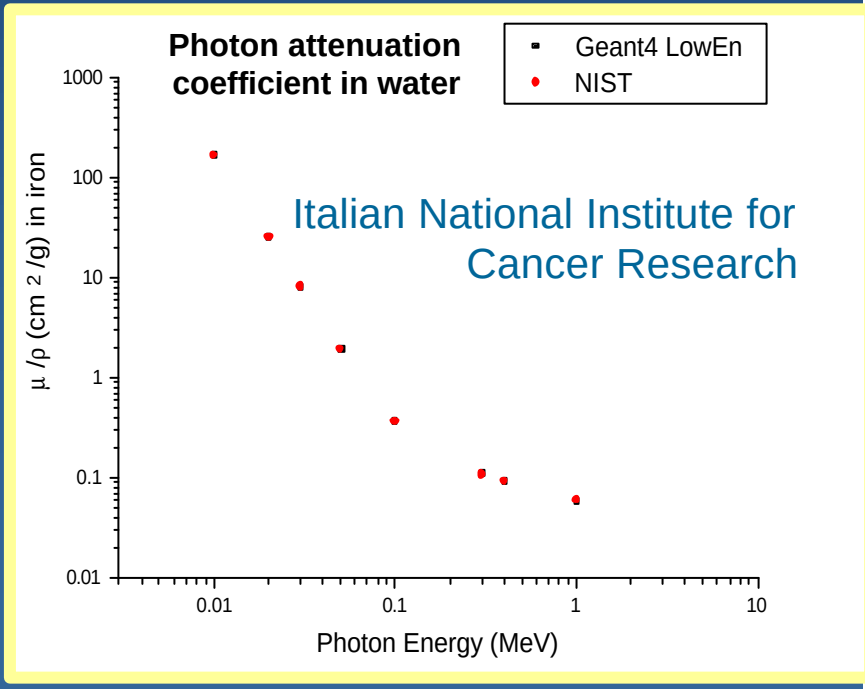
Collaboration with user groups in testing

Independent validation of physics models and implementation

Some examples







Traceability

Traceability between

- Requirements
- Design
- Implementation
- Tests

Traceability process

One of the objectives of the SPI programme in progress

Already fully implemented and documented in one of the Physics WGs

In progress in other WGs

Emphasis at the Geant4 Workshop in July

Example of traceability map

UR	Design	Implementation	Unit test	System test
1.1	LowE.mdl e/photon	G4LowEnergyBremsstrahlung G4LowEnergyIonisation G4LowEnergyCompton G4LowEnergyPhotoelectric G4LowEnergyRayleigh G4LowEnergyGammaConversion	G4LowEnergyGammaConversionTest G4BremsstrahlungTest G4IonisationTest G4ComptonTest G4Photoelectric G4RayleighTest	Test14
1.2	Future	Future	Future	Future
1.3	Future	Future	Future	Future
1.4	Future	Future	Future	Future
A.1	LowE.mdl e/photon	See 1.1	See 1.1	Test14
A.2	LowE.mdl e/photon	See 1.1	See 1.1	Test14
A.3	LowE.mdl e/photon	See 1.1	Not available yet	Test14
A.4	LowE.mdl e/photon	Future	Future	Future

SPI

- Improvements motivated by the assessment in 2000
  - ➔ focus on design and QA
- Tailor Geant4-wide processes to the specific context of WGs, or even of projects
- Aim: a continuously improving software process

Version 1

27 May 2001

Guidelines of the the Development and Tag Process of the Geant4 Low-Energy Electromagnetic Physics Working Group

P. Nieminen and M.G. Pia

1 Introduction

This document provides guidelines for a development and tag process. It applies to developments in the lowenergy category, in system tests and examples under the responsibility of the Low-Energy Electromagnetic Physics Working Group.

2 Development and tag process

2.1 Lowenergy category

The following items are under the responsibility of the developer:

1. Perform the development and tag process in agreement with the present document.
2. Code developments should be carried on according to the plans agreed by the WG, respecting the agreed priorities.
3. Responsibilities for code developments are agreed in WG meetings and documented in the minutes of the meetings.
4. Code developments should be carried on according to Geant4 coding guidelines.
5. Check out a copy of the design from the repository.
6. Consult with the WG coordinators if there are any questions about understanding the design documents.
7. Start the development from the most recent monthly global reference tag, with on top the most recent lowenergy tag recommended by the WG coordinators for group's use.
8. Update the work environment regularly to the announced monthly global reference tags, lowenergy tags and tags of system tests relevant to the lowenergy category.
9. Implement the code according to the design.
10. Embed in the development process the testing process described in a separate document.
11. Implementation and testing should be
12. Commit the code under development to the CVS repository frequently, and well in advance with respect to the scheduled release, even if it is not fully functional yet; the minimal requirement is that the code should compile clear of errors and warnings on at least two supported platforms.
13. The procedure to commit code to CVS is:
    - Update the work environment to CVS head: *cvs update -A -d -P*
    - Add new files (if pertinent): *cvs add filename*
    - Commit the new code to the CVS repository: *cvs commit -m "Comment" filename*, where "*comment*" is a meaningful description of the development being checked in.
    - Inform the WG coordinators about the new code committed to the repository.
14. Adding new directories to the CVS repository may be done only in agreement with WG coordinators.
15. Issue "private" tags as frequently as needed. *Private tags are category tags with a format different from the official Geant4 format. Only WG coordinators should issue official category tags.*

Relevant activities in 1999-2000

User Requirements

- Collection, specification and analysis of new requirements
- Maintenance of WG URD
- *(for WGs having specific URD)*

The maintenance of the general Geant4 URD is part of the SPI in progress

OOAD

- Design iterations
- OOAD cycles for new features
- Move to UML notation

Testing & QA

- New tests
- Extension of coverage of existing tests
- Collaboration with users in validation
- Design and code reviews

SPI

- Traceability
- Internal training to SPI