GEM TN-93-450

Computing Subsystem Steering Committee Meeting

FNAL - SSCL (Video Conf.)

John Hilgart
Superconducting Super Collider Laboratory

July 22, 1993

Abstract;

Attendees and presentations of the Computing Subsystem Steering Committee Meeting held at the Superconducting Super Collider Laboratory on July 22, 1993.

Computing Subsystem Steering Committee

22 July 1993

FNAL - SSCL (video Conf.)

Name	ms	e-mail
Gary Word	2005	Gary_Word@ssc.gov
John Hilgart	2003	hilgart@sscvx1
George Yost	2005	Yost@sscvx1
John Womersley	2005	womersley@ssc.gov
Lee Roberts	2003	lroberts@sscvx1
Irwin Sheer	2300	sheer@sscvx1
Larry Cornell	2003	cornell@sscvx1
Ken McFarlane	2005	mcfarlan@sscvx1

Jim Dunlea (at FNAL)

Organization of Project Management Files for computing subsystem

o Currently: nothing in place

o Need place which supports the way we have decided to do software project management

o Proposal:

a single afs volume:

/usr/ssc/gem_dev/

below which is

```
.../gem_dev/PROJECT/pdoc/
                        ~~/overview.html~~
                        /cvs
                        /src
                        /doc
                         /SUB_PROJECT/pdoc
                                        no cvs?
                                        etc.
```

Overview Document

o Paragraph overview of project, preferably in html, or alternatively in plain text. Examples will be provided.

Suggested name:

overview.html (html version)
overview.text (plain text version)

What's in pdoc (acl–protected)

o Management documents as agreed to in last CSSC:

project charter
project initiation plan
project design documents
project reviews
PERT and GANTT charts
etc.

These documents probably need to be stored twice: in a universal display format (.ps) as well as the native word processor format, e.g., rtf. Direct html may be appropriate for shorter documents. ~~ascii~~ plain ~~text~~ mit

File protections

o Create appropriate work groups for each project, whose members could read and write to the sensistive pdoc area

o WWW access is still possible for this area, for those logged onto an ssc.gov domain computer

o The rest of the PROJECT area could be world-readable but group writable

Example Projects

o Software technology choices

o Geometry definition language

o 3-D event display

o everything else from the projects list Gary and Irwin prepared But some refinement needs to be done for optimal project names.

WWW's role

o There will be a projects link under the GEM home page, from which:

+ The overview document of all projects will be accessible as a single document

+ All project directories will be available through a hyperlink to the /usr/ssc/gem_dev directory (for world read access)

+ A "developers only" link will give developers access to the pdoc areas for reading

From mcfarlan@diamond.gem.ssc.gov Wed Jul 21 02:38:47 1993
Return-Path: <mcfarlan@diamond.gem.ssc.gov>
Received: from sunova.ssc.gov by pdsf.ssc.gov (4.1/SMI-4.1)
id AA21651; Wed, 21 Jul 93 02:38:46 CDT
Received: from diamond.gem.ssc.gov by sunova.ssc.gov (4.1/SMI-4.1-DNI-7.0.1)
id AA10365; Wed, 21 Jul 93 02:34:19 CDT
Message-Id: <9307210734.AA10365@sunova.ssc.gov>
Received: by diamond.gem.ssc.gov
(16.8/16.2) id AA05871; Wed, 21 Jul 93 02:38:42 -0500
Date: Wed, 21 Jul 93 02:38:42 -0500
From: Ken McFarlane <mcfarlan@diamond.gem.ssc.gov>
Apparently-To: Gary_Word@ssc.gov
Status: RO

SYSTEM DEVELOPMENT (draft)
K. McFarlane, July 1993.

1. Overview

This file contains a suggested project life cycle, templates of the documents which would be produced during the life cycle, and will sometime contain an example. The approach is a combination of standard ones and the spiral approach of Boehm ["A Spiral Model of Software Development and Enhancement," Barry W. Boehm, IEEE Computing...].

The spiral nature is defined by using the same set of considerations and the same phases at each stage of the life cycle, though the phases may be quite different in subject matter. (The life cycle has stages, each stage has five phases.) Each stage can be the last, if that is deemed appropriate. The approach is intended to accomodate hardware, as well as software, projects. It is more flexible than standard life-cycles alone, and therefore more suited to our needs.

In the spiral model as described here, each stage begins with statements, from the previous stage, of:

1. Goals
2. Constraints
3. Resources allocated
4. Schedule

The first phase is to identify:

5. Alternatives
6. Dependencies
7. Risks

The second phase:

8. Evaluates alternatives
9. Formulates a risk resolution plan
10. Carries out the risk resolution plan

This phase could include development of detailed requirements, system simulations or building and testing of prototypes as appropriate to the phase.

The third phase will be the development and acceptance of

11. A 'next-level product,'

based on the results of steps 8-10 above, which might range from a set of requirements and specifications in the early stages to a completed system in the final stage.

The fourth phase consists of:

12. Plan for next phase, including items 1-4 above.

The fifth (zeroth) phase is a management review resulting in:

13. Approval of the plan and additional commitments.

Figure 1 shows the 'coordinate system' for the spiral. The starting point is on the left, and the spiral goes clockwise (->-):

```
  Identify alternatives,          |        Evaluate alternatives,
  dependencies,risks              |        Resolve risks
  Steps 5,6,7                .----->------.  Steps 8,9,10
 _________________________|_    |      |
 Review goals, constraints | \_  |      |
 resources, commitments    |    \|______V_________________________
 Steps 1,2,3,4/13     .    O   _/|      |
 ______________________|_______/  |      |
                       |          |      |
                       '--------<--------'   Develop, verify
   Plan next phase                |          next-level product
   step 12 (i,ii,iii,iv)          |          Step 11
```

The goal is to provide a flexible system that includes:
- o Structured techniques for developing a complete system
- o All managerial planning and implementation activities
- o Identification and resolution of risks
- o Allocation of resources
- o Identification of dependencies
- o Definition of major and intermediate deliverables
- o Standard schedule and quality-control points
- o collaboration and project management approval points.
- o Description of all data entering and leaving the system
- o Description of all processes (including manual) that transform the data
- o Definition of all required hardware (CPUs, memory, networks, mass storage)

Management of risk plays a central role in Boehm's spiral model, and he lists the top ten software risk items and the risk management techniques. Here we just list the risks, and give the overall management plan.

1. Personnel shortfalls
2. Unrealistic schedules and budget
3. Developing the wrong functions
4. Developing the wrong user interface
5. Gold plating
6. Continuous changes in requirements
7. Shortfalls in externally furnished components (dependencies)
8. Shortfalls in externally-performed tasks
9. Real-time performance shortfalls
10. Straining technical capabilities

Management plan:
1. Identify top ten risks
2. Create plan to resolve each risk item
3. Update list, plan and results monthly
4. Highlight risk item status in monthly project reviews
5. Carry out planned corrective action.

2. System Development Stages

Many system development projects will have four major stages: project planning, design, implementation, and maintenance and enhancement. The implementation stage can be multiple. Here we describe these stages in the 'spiral' language.

2.1 Project Planning

The project planning activity is itself a three-stage process. Management review occurs at each stage.

2.1.1 PROJECT CHARTER
First, a project charter is created, including items 1-4 above. This will come from outside the project, e.g. from ECMC, and may be the result of a zeroth turn of the spiral. The resources will include the people charged with completing steps 5-12, resulting in the project initiation plan.

2.1.2 PROJECT INITITATION PLAN
The project initiation plan is produced in the first stage; it will address items 5-12, with the following results for particular items:

5. Among the alternatives should be an analysis of the current approach
11. Conceptual design
12. Plan for next stage, including:
    i) Goals
    ii) Constraints
    iii) Resources needed
    iv) Schedule
    (These will be translated into 1,2,3,4 for the next stage).

The plan should include a plan for developing requirements, a life-cycle plan, and a recommended methodology.
The management step, 13, may modify these and add additional commitments, leading into the next stage.

2.1.3 PROJECT PLAN
The project plan is more detailed than the initiation plan, and will require more resources to develop. It will include:

11. Requirements and requirements validation
    Preliminary analysis
12. Description of the specific methodology to be used
    Cost and schedule estimate
    R&D plan
    Management plan
    Review plan.
    Points i-iv for the next stage

2.2 Project Design

The project design phase will produce a REVISED PROJECT PLAN including:

10. R&D carried out to supplement the top-down analysis approach with a bottom-up approach, prototyping elements needed in the final system.
11. New essential model
    Process specifications
    Data specifications (or object specifications)
    Hardware specifications
    Draft operations manual
    Draft procedures (user's) manual
12. Configuration management plan
    Testing plan
    Implementation plan; including a schedule of walk-throughs and reviews

The major deliverables will be:
- o revised project plan
- o revised requirements document
- o Specifications documents
- o draft manuals
- o reports on the R&D, and
- o implementation plan that includes:
  - revised cost and schedule estimates
  - procurement plan.

2.3 Implementation

In this phase, which may be carried out twice, once for a major prototype

and again for the final system, a full system is implemented that satisfies the requirements. This includes:

11. Next-level product:
    Hardware and software procurement
    Software development
    Installation
    Testing
12. Maintenance plan
    Upgrade plan
13. Final QUALIFICATION REVIEW.

Major deliverables include:
- o installation specifications
- o a final requirements document
- o data conversion specifications
- o data and process specifications,
- o libraries (if object-oriented, class libraries)
- o training specifications
- o test specifications
- o final operations and maintenance manuals
- o the qualification review report
- o an enhancement or upgrade plan
- o and a working system.

2.4 Maintenance/Enhancement

This is not a single stage, but two distinct parallel phases. The maintenance stage for hardware has obvious components: repairs, replacements of obsolete parts, etc. For software and hardware, there will be changes needed due to clarification of requirements and rectification of deviations from requirements.
Enhancement consists of adding new functionality or performance to the system, and will go through a process of planning, control, and review.

Draft templates follow.

DRAFT TEMPLATES

A set of draft templates follows, for the project charter, the project initiation plan and the project plan. Templates for the revised project plan, the implementation and maintenance or enhancement plans have still to be worked up, as have templates for reviews, walkthroughs, cost documents, and schedule documents. The mechanism for version control and inclusion of documents is to be defined.

Note that the project name and/or the
project directory define the names of the standard set of files, so that any tools developed to manipulate the files can access them with that information alone. For example, the name of the 'xplan' file defining the project schedule, resources and dependencies would be PROJECT.prj, while the derived PERT chart might be PROJECT.pert or PROJECT.pert.tex.
The templates are separated by ^L characters.

Project name:
Project directory:

P R O J E C T C H A R T E R

[version information]

Project initiation team:
Date:

1. Goals

2. Constraints

3. Resources

4. Schedule
   Start date:
   Identification phase expected finish date:
   Risk resolution expected finish date:
   Next-level product expected finish date:
   Plan completion expected date:
   Review date:

P R O J E C T I N I T I A T I O N P L A N

[version information]

<include project charter>
Date:

5. Alternatives
   1. Analysis of current approach(es)

6. Dependencies (libraries, applications, etc.)

7. Top ten risks (include risk of unavailable resources, dependencies,etc):
   1.
   2.
   3.
   4.
   5.
   6.
   7.
   8.
   9.
   10.

8. Plan for, and results of, evaluation of alternatives

9. Risk resolution plans
   1.
   2.
   3.
   4.
   5.
   6.
   7.
   8.
   9.
   10.

10. Risk resolution plan results
    1.-10.

11. Conceptual design/preliminary analysis

12. Plan for next stage
    1. Requirements development plan
    2. Life-cycle plan
    3. Recommended methodology
    4. Recommendations for next stage:
       i) Goals
       ii) Constraints
       iii) Resources needed.
       iv) Schedule

13. Review Results
    A) New Commitments
    B) Approved next-stage charter(s) (projects may be split at this phase):
    Name of (new) project:

1. Goals
2. Constraints
3. Resources
4. Schedule

Project name:
Project directory:
Project initiation plan:

P R O J E C T P L A N

[version information]

Project planning team:
Date:

<include review results>

5. Alternatives

6. Dependencies (libraries, applications, etc.)

7. Top ten risks (include risk of unavailable resources, dependencies,etc):
   1. Cost
   2. Schedule
   3.
   4.
   5.
   6.
   7.
   8.
   9.
   10.

8. Plan for, and results of, evaluation of alternatives

9. Risk resolution plans
   1. Cost: estimating methodology,etc.
   2. Schedule: estimating methodology, etc.
   3.
   4.
   5.
   6.
   7.
   8.
   9.
   10.

10. Risk resolution plan results
    1.-10.

11. Report of Requirements development activity

    Report of requirements validation activity

    Preliminary analysis results
    (CASE tool files:)

    Cost and Schedule estimate

12. Plan for development/design stage
    1. Specific methodology
    2. R&D plan
    3. Management plan

4. Review plan
5. Recommendations for next stage:
   i) Goals
   ii) Constraints
   iii) Resources needed.
   iv) Schedule

13. Review Results
    A) New Commitments
    B) Approved next-stage charter(s) (projects may be split at this phase):
    1. Goals
    2. Constraints
    3. Resources
    4. Schedule

Minutes of the Computing Subsystem Acting Steering Committee of July 22nd, 1993

Present: Gary Word
John Hilgart
George Yost
John Wommersley
Lee Roberts
Irwin Sheer
Larry Cormell
Ken McFarlane
Jim Dunlea (by video-conference)

Systems Development, Ken McFarlane (see attached transparency copies included in the limited-distribution-GEM-technical-note generated from this meeting)

We have a 500 man-year effort for GEM computing systems. How do we ensure we have a flexible enough procedure to accomodate the different kinds of development?

Ken presented two software-development models. The waterfall model is an 8-stage process which shows sequential, chronological project development in an ordered fashion. This doesn't reflect how we really develop software in our community. Boehm (IEEE computing, 1988) has a model more suited to our style: the spiral model. Here a project normally starts out small-scale, with enough resources allocated to finish one spiral turn at a time. The spiral stages are (briefly):

1) State goals and resources required.
2) Survey alternatives. State risks.
3) Risk reduction and resolution
4) prototype
5) document, implement
6) planning
7 = 1') management review. New resources for second turn
2') etc.

Ken's transparencies also include proposed templates for three documents which would be required of every software project:

1) Project charter
2) Project initiation plan
3) Project Plan

As these templates are all less than 2 full sides, they don't appear too burdensome, if you're willing to accept that there should be some documentation. This is a contentious issue as Irwin worries that three documents would be unacceptable for physicists who have always avoided documenting anything. The threshold becomes too high. Jim Dunlea also felt that the amount of documentation should be reduced. Ken said this desire can be accomodated for smaller projects where it may be appropriate to merge the three documents into one. The amount of documentation has to scale with the project's ambitiousness.

Larry Cormell felt strongly that Ken's was the right approach, pointing out that the SSC project will be reviewed like no other science project in history. HEP software development is also high-risk in all the categories one looks at. The documentation provides some risk reduction. Another way to reduce risk: breakdown of projects. Experience has shown that 3 - 5 people working on a project has much better chance of success than 100 people. The unfortunate military types have to produce 30 documents per project, in conformance with DOD 2167A .

Organization of Project Management Files for computing subsystem, J.H. (see attached presentation)

Briefly, an area is proposed for development and project documentation which would be an afs area:

```
/usr/ssc/gem_dev/
```

under which, for each PROJECT, would be

```
/usr/ssc/gem_dev/PROJECT/pdoc/ <- for project mgmt documentation
                         /cvs  <- cvs repository for this project
                         /src  <- source
                         /doc <- user guides, etc.
                         /SUB_PROJECT/pdoc <- if applicable
                             etc.
```

The committe members disagreed on the level of file protections which should be applied to pdoc, and the PROJECT's directory. J.H. pleaded for world-read access to the PROJECT area, except for pdoc, but Irwin thought this was too open. These two have been tasked to come up with a proposal for protections.

Another part of the proposal was to use WWW integrally to give read access to documents. A global document which contains an overview of all projects should also be available.
It was decided to use either plain-text or direct html for project documents. J.H. has offered to help the first few documenters get started with html.

Other Business

We have been an acting committee until now. Ken will take up this issue with a GEM leader to make the CSSC an official body.

Next meeting: August 26th.

Submitted by John Hilgart