Management of Science Tools Development

S. W. Digel (SU/HEPL), J. P. Norris (NASA/GSFC)

SSC-LAT Science Tools Workshop

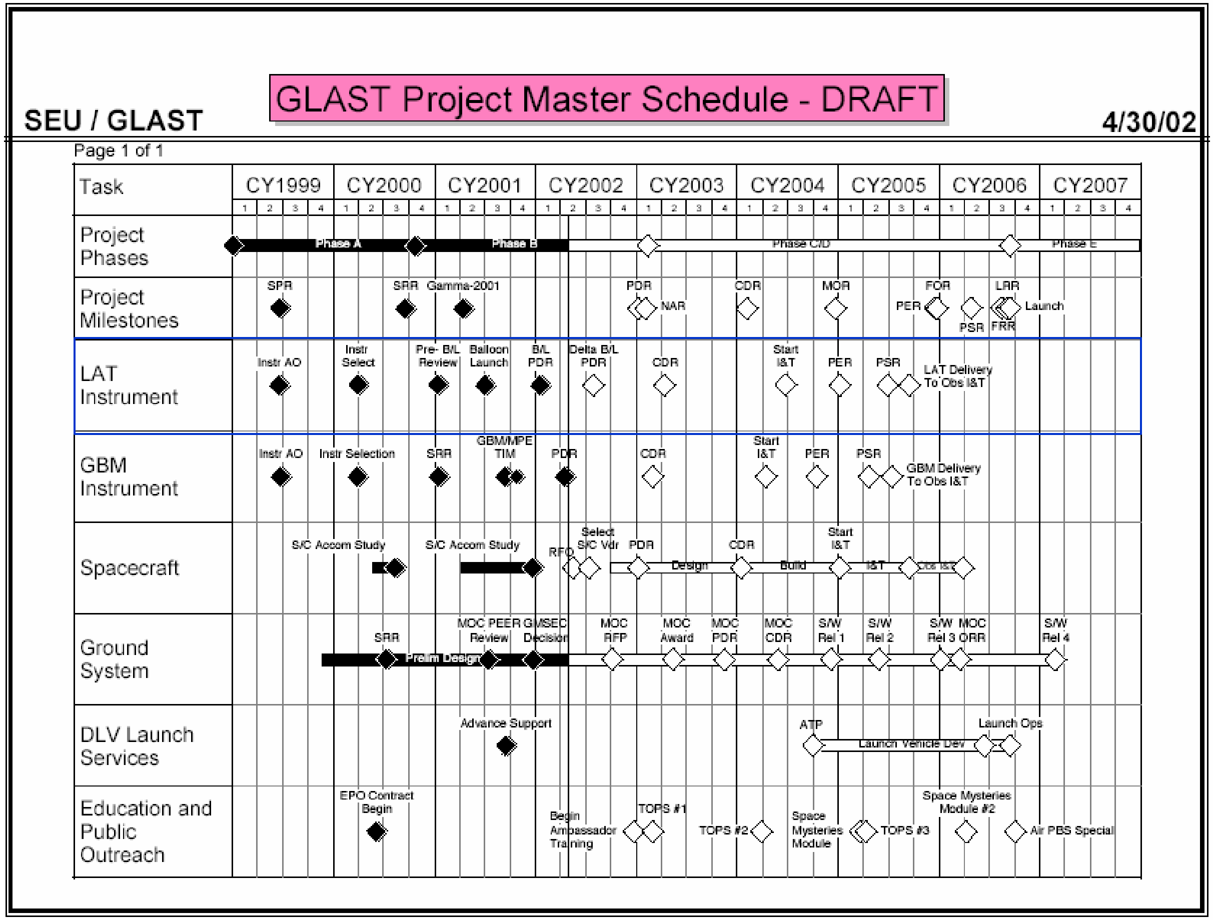
June 12-14, 2002, SLAC

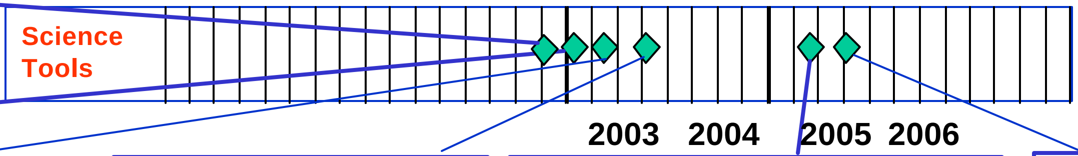
Outline

1. Schedule – external and internal milestones
2. Management plan
3. Assignments of development responsibilities for analysis tools and databases
4. Summary: Software development requirements

1. Schedule (high-level)

- **External milestones**
  - **Project NAR, LAT CDR, mission launch**
- **Internal milestones**
  - **[dates TBR]**
  - **First up: formal review of Science Tools development plan**





Science tools devel plan

SAS CDR

"Photon database technology in place"

Mock data challenge I

All req. science tools in place

MDC II

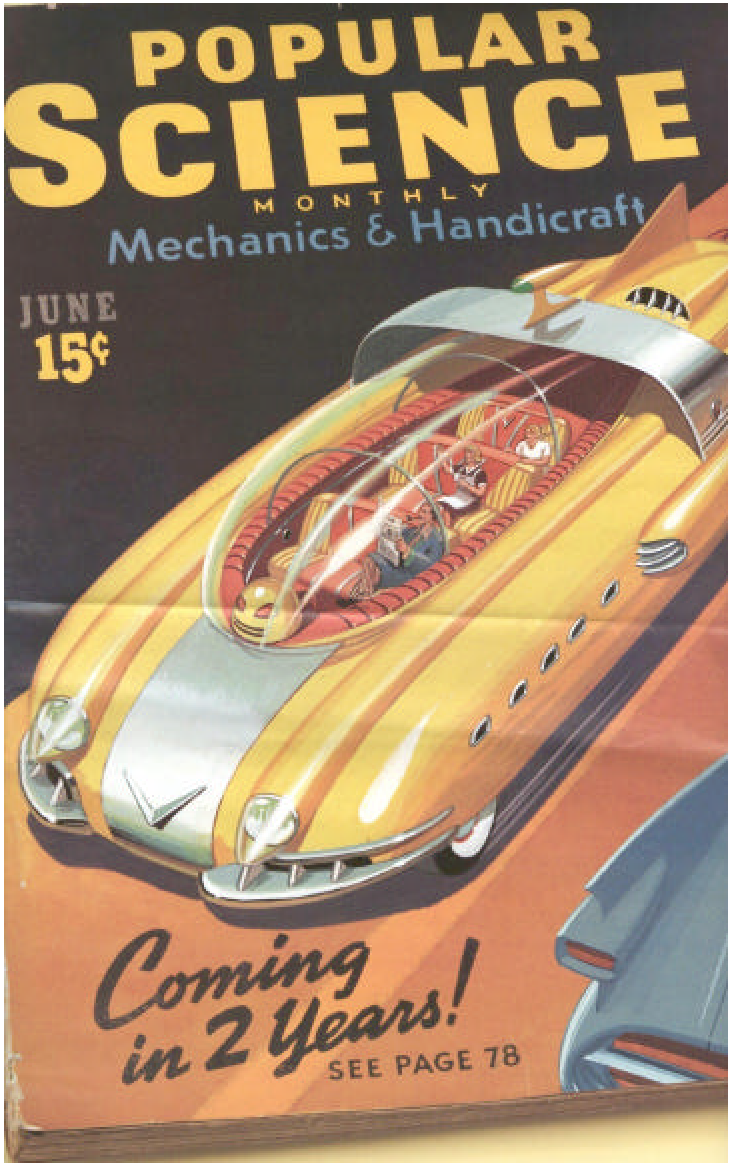
1. Schedule (cont)

- **Regarding the science tools development effort, some priorities are evident (in terms of what comes first and what needs the most work)**
  - **Observation simulators and associated pointing/livetime history and Level 1 databases**
  - **Associated utilities, including visualization tool**
  - **Likelihood analysis, GRB characterization**
- **Others can be identified as having lower priority**
  - **Tools that inherit most directly from existing software, such as pulsar phase assignment and periodicity search**
  - **Interstellar emission model – even a fairly crude one can be assumed to be perfect up until launch**
- **These considerations, plus estimated development times and FTE requirements, will be factored into the overall schedule for the development plan**

2. Management plan

- **Constraints in addition to schedule (& money):**
  - Standard analysis environment is to be jointly developed and supported by the LAT team and the SSC, the latter for use by guest investigators potentially anywhere and into the indefinite future
  - Widely distributed membership of LAT collaboration, relatively large number of developers, some likely with strong opinions, a spectrum of software development experience
  - Responsibility without authority
- **Mitigating factor beyond wishful thinking**
  - Commitment across the LAT/SSC science tools development group to the mission and the science

2. Management plan (cont 2)



- Our concept
  - Drives itself!
  - Accommodates everyone!
  - Streamlined!
  - All latest software innovations!
  - Ready in 2 years!

2. Management plan (cont 3)

- Higher-order organization: For efficiency in sharing ideas among related tools, and ideally to maintain adequate oversight of development efforts, the development will be managed by 4 coordinators
  - Databases and related utilities
  - Other utilities
  - Analysis tools
  - Observation simulation
- Responsibilities of coordinators
  - Overseeing production of requirements documents from the requirements summaries for the tools and databases
    - In particular, definition of the interfaces among the tools intra- and inter-group
  - Deriving and refining the schedule (ala Richard’s “rolling wave”)
  - Other roles – e.g., in terms of seeing that the Open Issues are addressed adequately (with documented results), and that adequate tests are developed are TBD and *open for discussion*
- Coordinators will report to the SSC-LAT working group (TBR, figuring out how to incorporate the ultimate responsibility of the LAT team in the management structure)

3. Development assignments

- With the SSC we are identifying development leads for the specific tools and databases
  - Subject to change if our understanding of the tasks changes or if something just is not working
- Depending on the complexity of the tool or database, the lead representative could be coordinating a small team.
- In the breakout sessions we will attempt to estimate development requirements and after the meeting will start organizing these teams.
  - Goal: Not too many and not too few

4. “Guidelines from Management”

1. completely research the SW task
2. determine the full requirements
3. realize an integrated schedule, and follow it
4. re-use SW where appropriate
5. divide task where appropriate, and conquer the pieces
6. team up where appropriate, since more heads better than one
7. cross-fertilize existing SW suites to evolve more functional chimeras
8. modularize, so that previously undreamt of features can be easily inserted
9. individuals complete SW assignments/milestones, making for satisfaction
10. perform necessary & sufficient tests, at all levels