| Project Name: | | |
|---|---|---|
| Time Accounting Timeline | | |
| Business objective served by this project | | |
| Significantly reduce the time required by queue coordinators and data analysts to complete the daily data quality assessment and time accounting. | | |
| Project Manager/Leader: | Project Sponsor: | PDS Version/Date: |
| Shane Walker (to transition to Florian N.) | Andrew Stephens | 1 / 2010-10-20 |

## Project Description

### Issue Statement:

The Time Accounting Timeline software provides a tool that will assist with the time accounting and data quality assessment process. The software requirements document, GS-OCS-TAT-SRD-01 by Andy Stephens and Bryan Miller, defines

> *a tool that can automatically generate a graphical timeline that can display either a Night View or Program View. The Night View shows the details of any night of observing which can be easily edited for time loss such that the changes will propagate to the appropriate programs in the OT, fault reports and night log. The Program View shows the details for a particular program, illustrating how this program's time was used throughout the semester.*

### Project Objective Statement (POS):

The goal of this project is to provide software support for completing the otherwise arduous and time consuming task of correcting time accounting and data quality assessment information. It gathers together several disparate sources of data in a convenient display and provides a single location in which they may be edited. In particular, the major features of this software will include

- A graphical "Timeline View" showing either an entire night or an entire program's worth of events related to time accounting. These include telescope slews, fault start/stop times, weather loss stop/start, twilights, observation abort, pause, etc.
- A "Tabular View" of the same information.
- Editing support for reading and writing time accounting adjustments to science programs in the Observing Database including observation-level time corrections, QA state changes, and Observing Log comments and summary information.
- Editing support for viewing and updating fault reports in Remedy, adjusting the time loss information.
- Night log view with information extracted from GEA.
- Reports including a "science breakdown" of all observations and their time charges, fault time loss, non-chargeable time, and an e-obslog text export.

### Project Flexibility:

| Flexibility Matrix | Least Flexible | Moderately Flexible | Most Flexible |
|---|---|---|---|
| Scope | | ✔ | |
| Schedule | | | ✔ |
| Resources | ✔ | | |

### Major Deliverables:

- The time accounting software itself, along with any updates to existing systems that are needed to support it.
- Operational / user documentation describing basic operation and clarifying any unusual or more involved steps.
- Technical documentation with an architectural overview.

### Assumptions:

- Software will follow an iterative development model in which functionality is rolled out for testing throughout the entire development process.
- Science support is available for answering questions, testing changes, collecting feedback from users, and prioritizing tasks. We anticiapte roughly 20 hours per 5 week iteration period, where about 12 hours are spread across stakeholders for testing and providing feedback. The remainder is an estimation of the time needed for the project scientist to participate in the development process / meetings.
- Access to Remedy / GEA from external tools is already fairly well defined.
- The software will be developed on the existing Science Program model, but isolated from it as much as possible to permit it to withstand planned changes in the coming years.
- This is a standalone tool in the same fashion as the Queue Planning Tool, albeit with a simplified installation and update procedure if possible.

### IS and IS NOT:

- **IS:** New software development to automate and simplify time accounting and quality assurance data editing.
- **IS:** The primary project of a single developer.
- **IS NOT:** A rewrite or extension of the OT, Data Manager, or any existing software system.
- **IS NOT:** A replacement for Remedy, GEA, the OT, the e-obslog or any existing software system.

## Strategy and Resources

### Milestones and Stages:

Following an initial project evaluation and elaboration phase, the development will be performed in an iterative fashion. Each iteration produces working software for evaluation. Feedback from the evaluation is then used to guide the development of the software with the general understanding that radical changes to the feature set or implementation will have an impact on other projects and must be introduced in a controlled way if necessary. Estimating times for the major features to be delivered, we anticipate six iterations ending with the "final" release of the software.

- Evaluation and Elaboration Phase
    - Initial requirements document completed by science, folding in input and clarifying questions raised by engineering.
    - Initial test plan completed by engineering.
    - Initial (prioitized) features backlog completed by science.
    - Conceptual Design Review
- Software Iteration 1
    - Features / bugs for iteration 1 selected.
    - Release 1 delivered by engineering.
- Software Iteration 2
    - Features / bugs for iteration 2 selected.
    - Testing feedback from iteration 1 received, backlog updated and prioritized by science.
    - Release 2 delivered by engineering.
- Software Iteration 3
    - Features / bugs for iteration 3 selected.
    - Testing feedback from iteration 2 received, backlog updated and prioritized by science.
    - Release 3 delivered by engineering.
- … Software Iterations …

- Final Software Iteration
    - Features / bugs for final iteration selected.
    - Features / bug backlog updated for future work.
    - User acceptance testing.
    - Final release

### Estimated Costs:

- No equipment or resources beyond those afforded for normal software development.

### Core Team Members(see Guidelines for Developing New Projects document):

- Florian Nussberger (Project Manager)
- Andrew Stephens (Project Scientist)
- Manuel Lazo (Systems Engineer)

### Extended Core Team Members:

- Florian Nussberger (software development)
- Shane Walker (consultation on existing OCS infrastructure, codebase)
- Andrew Stephens (science support, feedback processing, backlog processing)
- TBD science staff members (testing)

### Dependencies that require coordination:

- None

### Risks and Issues:

- External tool dependencies (Remedy / Nightlog). The ease with which tasks that depend on reading and updating information in tools external to the OCS depends upon the support provided for external software access (if any).

### Supplemental Resources:

- None