System Implementation Plan

PDS 2010 System Review

March 22-24, 2010

Sean Hardman

Topics

- Approach

- Milestones

- Development Lifecycle

- Development Environment

- Resources

- Deliverables

Approach

- Based the architecture development on the findings of the three study phase working groups.
  - Architecture, Data Model and User Support
- Derive the design from the architecture, existing PDS requirements and data system surveys.
- Formed the System Architecture Working Group in July 2008.
  - Identify drivers, principles, views and overall architecture.
- Formed the System Design Working Group in January 2009.
  - Investigate standards, core technologies and open source solutions.
  - Prepare designs for each component in the system.
  - Implement and deploy according to project build schedule.

System Design Working Group

- The working group consists of the following personnel:
  - Sean Hardman (Engineering)
  - Todd King (PPI)
  - Mike Martin (Management)
  - Paul Ramirez (Engineering)
  - Alice Stanboli (Imaging)
  - Tom Stein (Geosciences)
- Periodic teleconferences held to review and discuss service designs.
- Utilize a Wiki for capturing minutes and design artifacts.
- Official artifacts are posted to the PDS Engineering Node web site.

Architecture/Design Milestones

- Presented the preliminary System Architecture to the PDS Technical Staff in September 2008.
  - Incorporated comments and suggestions into follow-on drafts.

- Presented the System Architecture to the Management Council in December 2009.
  - Directed to move forward into design.

- Initiated the design effort in January 2009.
  - Initially focused on data system surveys and design decisions.

Architecture/Design Milestones cont.

- Presented the preliminary System Design to the PDS Technical Staff in June 2009.
  - Incorporated comments and suggestions into follow-on drafts.

- Presented the System Design to the Management Council in August 2009.
  - Directed to continue detailed design effort.

- Initiated Open Source investigations and prototyping in September 2009.

Development Lifecycle

- Initiate a brief study identifying the state-of-the-practice for each component and whether there are COTS or open source solutions available.
- Identify use cases and requirements for the component.
- Prepare a design for implementing the component from scratch or for integrating a COTS or open source solution.
- Implement/integrate the component per the design with emphasis on unit test integration.
- Test the component against the requirements (component and integrated).
- Deploy with the system to the target environment (e.g., DN, EN).

Phased Development

- Phase I – Ingestion

  - Core services that will provide the infrastructure for the system and support ingestion of data.

  - Tools for design, generation and validation of products.

- Phase II – Distribution

  - Services for discovery and retrieval of registered products.

- Phase III – User Services

  - Enhanced capabilities for discovering and accessing PDS data along with support for DN-specific applications.

  - Tools for transformation and visualization of products.

Development Environment

- Development Platform
  - The majority of software will be developed with Java 2 Platform Standard Edition (J2SE) 6.0.
  - A scripting language like Python or Perl will be utilized where appropriate.
  - Developers at the Engineering Node utilize the Eclipse Integrated Development Environment (IDE).

- Operations Platform
  - Engineering Node operates PC-based commodity computers running Red Hat Linux.
  - Discipline Node's offer a variety of platforms including Linux, Solaris, Mac OS X, and Windows.

Development Environment (cont)

- ## Build Management
  - Apache's Maven is used for build management including dependency resolution and deployment package generation.
  - Looking to incorporate Continuous Integration and FindBugs (Java-based code analyzer).

- ## Configuration Management
  - The Engineering Node uses Subversion for managing software configuration.
  - Looking to configure two repositories, one for EN development and one for Node and community contribution.
  - Utilize JIRA for issue tracking (available externally).

Resources
System Design

- Engineering Node
  - System engineering and working group leadership.
  - Expect this to continue through 2012.

- Discipline Nodes
  - Support from working group members to review documents and presentations as well as attend periodic teleconferences.
  - Expect this to continue through 2011.

Resources
System Development

- ## Engineering Node
  - Open source and COTS investigations as well as prototyping started in 2010.
  - Component development starting in 2010 through 2012.

- ## Discipline Nodes
  - Begin integrating Node-specific software with the system starting in 2011 through 2012.

- ## Ames Research Center
  - User interface design and development starting in 2011 through 2012.

Deliverables

- System Architecture Specification Document
- Service Software Requirements Document (SRD)
  - General level 4 requirements for all services.
- Software Requirements and Design Documents (SRD/SDD)
  - For each component in the system.
- Software Components (per build schedule)
- Release Description Document (RDD)
- Operations Guide
  - System wide and per component.

Wrap Up

- The system architecture and design efforts have involved the Node technical staff from the beginning with periodic updates given to Node management.

- Taking a phased development approach to accommodate funding and resource availability.

- The development environment supporting system implementation utilizes modern tools and is in place.

- EN development staff is shifting focus to 2010 implementation with core service development underway.

Questions/Comments