CIShell Tools: The Road Ahead
Web Client Visualizations

Past and Present
The Past

- Old frameworks
  - Flash-based visualizations
  - Google API

- Problems
  - Hefty upfront cost
    - API keys
    - Deployment
  - Speed
  - Scalability
The Present

• Modern frameworks
  – MVC backend
  – Vector rendering libraries
  – Responsiveness with HTML5

• Solutions
  – Almost no barrier for entry
  – Fast analysis and rendering
  – Scalable
  – Flexible
The Future
CI Shell Framework

An evolving platform
Foundations

• CIShell
  – CNS Core framework
  – Open Source
  – Plug-and-play
  – Supports variety of
    • Data formats
    • Analysis algorithms
    • Visualizations
Foundations

- **OSGi**
  - Industry standard plugin platform for Java
  - Widely supported (especially for Eclipse)
  - Live plugin updates
Foundations

- NWB/EPIC
- MapSustain
- Plugins
Ongoing Work

• Sci2
  – Full-range tool
    • Input formats
    • Analysis plugins
    • Data output
    • Visualization
  – Toolchain component
Ongoing Work

- Introspection/Preparation
  - What is CIShell?
  - What does it need to do?
  - What are the core plugins?

- Internal Changes
  - New data structures
  - JSON plugins
  - New plugin development tool
  - Web service interface
The Future

• Dependency Updates
  – Java 8
  – External libraries

• Unified platform
  – Web services
  – Appliance embedding
  – Better desktop experience
    • Workflow editor
    • General usability
    • Documentation
The Future

• Scalability
  – Service ecosystem interoperation
  – Data sources
    • Re-working general database plugins
    • Parallel/Stream processing via Hadoop/Storm
  – Workflows
    • Workflow logic structures
    • Stream-compliant workflow building
CIShell Tooling and Infrastructure

Workspaces in progress
Development Methodology

Kanban

Scrum

Tooling and Infrastructure

CI Shell Tools: The Road Ahead: Tooling and Infrastructure
Source Code Management and Version Control

- Network-based
- Difficult to merge
- Collaboration challenging
- Typical solution: “overwrite” old version with new

- Distributed
- Easy branching and merging
- Better group development
- Several successful workflows
Issue Tracking

• And:
  – Confluence
  – Email
  – Word-of-mouth

• Everything!
• Plus:
  – Integration with Github
Code Deployment

- Sourceforge is dated
- Only using Github public to distribute releases
- Github Enterprise
- Supports Continuous Integration and Deployment
- Facilitate collaboration with external developers
Build System

- Tycho
- Dependent on Eclipse IDE
- Rapid deployment
- Supports Continuous Integration
- Integration with Github
Future Tools and Infrastructure
Methodological Goals

• Git workflow
  – Gitflow
  – Github Workflow

• Fully test-backed development
  – Will be adding unit and integration tests as part of refactoring and updating efforts, as well as new development
  – Potential for Test-Driven Development/Behavior-Driven Development
  – Will facilitate Continuous Integration

• Continuous Integration
  – Fast, automated testing and deployment of the latest changes
  – Faster regression discovery, faster bug fixes
Possible Tooling Changes

- Development Environment
  - eclipse → IntelliJ IDEA

- Build System
  - Jenkins → TeamCity
Thank You!