Best Practices for Enterprise Data Management

Greg Pleiss
Don Hong
Topics

1. Procurement Approaches
2. Necessary Features and COTS Architecture
3. Deployment Considerations
4. Summary
5. Questions
Procurement Approaches
Enterprise System Procurement Approaches

A spectrum of system implementation patterns

- **Custom System**: Built entirely with custom code
- **Component System**: Uses stand-alone COTS components with significant development
- **COTS System**: Configured with minimal development
Comparing Approaches

Custom
- Custom built to meet business goals
- Emphasis on software development
- Design based on detailed functional requirements
- Considerable development time / effort
- Static system

COTS Components
- Custom system, using some COTS elements
- Emphasis on component-based software development
- Design based on detailed functional requirements
- Reduced development time / effort
- Some capability evolves with COTS releases

COTS System
- Orchestrates COTS to meet business goals
- Emphasis on workflows and configuration
- Design based on business goals and COTS capability
- Minimized development time / effort
- Evolving system with COTS releases

💡 COTS System follows standard system development lifecycles, but activities are different than traditional developmental systems
What is the COTS Approach?

“Going with the grain”

- **Maximizes investment** in *commercial off the shelf* software in a GIS system
- **Meets business goals** by leveraging COTS
  - Configures and extends COTS
  - Avoids developing software
- **Provides immediate capability**... continually improving via COTS release cycles
- **Engages users** early and often to iteratively improve system
A Different Way of Thinking

- **Challenges common procurement models**
  - Requirements focused on *business goals* vice detailed feature functions
  - Select COTS that best meets business goals and implement “with the grain” for best results
  - Schedule linked to COTS release cycles

- **Asks users to consider new business processes**
  - New workflows based on COTS strengths (usually better)
  - People resist change
  - Organizations resist change

- **Avoid temptation to overly customize**
  - Best intentions of many people often push systems towards customization
ArcGIS System Concept

The backbone of a COTS based, enterprise data management system…

ArcGIS
Using & Managing
Map & Geographic
Information
• Mapping
• Analysis
• Collaboration
• Editing
• Compilation

Cloud Services
Enterprise Services
Local

…with flexibility to support wide range of user roles
Enterprise GIS System Pattern @ ArcGIS 10

Power to configure how content is produced, shared, and used

Map & Data Products

Services (Maps, Analysis)

Field Edits & Partner Input

Basemaps (ArcGIS Online)

Authoritative Content Producers

Configurable & shared
- workflows
- production tools
- business rules
Necessary features and COTS architecture
Enterprise Data Management

Key features

Production Management

Geodata Management and Maintenance

Content Dissemination

Production

Geodatabase

Web Services

Publication Geodatabase

Web Services
Production Management

Configuring COTS to support work tasking, reporting, and management

• Provide transparency
  - Task and manage production
  - Provide status and reporting

• Utilize role-based functionality
  - Targeted user experience
  - Keep it simple

• Extend the reach of GIS workflows
  - Simple, web-based access
  - Enable non-GIS users
Production Management

Providing transparency and extending the reach of GIS workflows

**ArcGIS Server Web APIs**
- Flex API
- Silverlight API
- JavaScript API

**ArcGIS Server Workflow Manager**
- Reporting Services
- Tasking Services
- Management Services
- Execution Services

**ArcGIS Server**
- Basemaps
- Operational Layers

**Production**

Managers & Executives

Web Applications
Geodatabase Design Best Practices

Delivering high performance, application-ready geodata

• Emphasize end-user capabilities
  - Define detailed information products
  - Build to deliver, avoid the “what ifs”

• Embrace federated compilation and maintenance
  - Be realistic, prioritize compilation capabilities
  - Leverage data content provided by the enterprise
  - Integrate content at the service level, not RDBMS

• Design for COTS usability
  - Leverage geodatabase concepts (FCs, subtypes, etc)
  - Balance tradeoffs for maintenance and dissemination
Enterprise Geodata Management

Providing options for distributing geodata management activities

**Geodatabase Versions**
- Direct editing
- Basis for replication

**Connected Replicas**
- Workgroups
- Two-way exchange

**Disconnected Replicas**
- Ideal for outsourcing work
- XML-based transfer
- Two-way exchange

**Check Out Replicas**
- File-based GDB
- Inhouse or Outsourced
- One-time exchange

---

**Production Geodatabase**
- Read-Write Access
- Tuned for Editing
- Authoritative Content

**Publication Geodatabase**
- Read-Only Access
- Tuned for Services
- Approved Content
Data Maintenance

Configuring COTS to support workflow-driven, rule-based production

• Model new and existing processes
  - Integrate GIS and non-GIS
  - Standardize and streamline
  - Continually evolve workflows

• Configure a common rule-base
  - Feature templates
  - Map templates
  - Quality Control rules
Data Maintenance

Providing standardized workflows and rules-based data maintenance

ArcGIS Workflow Manager
- Workflows
- Application Integration
- Task Assistant Manager

ESRI Production Mapping
- QA/QC Rules
- Feature Templates
- Metadata Rules
- Map Templates

ArcGIS Web APIs
- Flex
- Java Script
- Silver light

ArcGIS Mobile
- iOS
- Windows
- Android

ArcGIS Server Workflow Manager
- Tasking Services
- Management Services
- Execution Services
- Mapping Services

Production

ArcGIS Desktop

ArcGIS Server
Content Dissemination

Using COTS to support a variety of end-users and applications

- Deliver diverse information products
  - Visualization
  - Analysis
  - Geodata

- Enable search & discovery
  - Standards-based (i.e. OGC)
  - ArcGIS Online ‘tags’

- Support internal and external users
  - Web services (connected)
  - Traditional media (disconnected)
Content Dissemination

Providing well-defined information products to the enterprise and beyond

**Connected**

- Many Users
- Cloud
- Many Applications
- ArcGIS Server Web APIs
  - Flex API
  - Silverlight API
  - JavaScript API

**Disconnected**

- Distributed Publication Node

**ArcGIS Server & GeoPortal Extension**

- Search & Discovery
- OGC Services
- Geoprocessing Services
- Map Services
- Geodata Services
A Complete System for Enterprise GIS

Managing, maintaining, and delivering geodata to the enterprise

Production Management

Geodata Management and Maintenance

Content Dissemination

(Connected/Disconnected)
Deployment Considerations
COTS Impacts to System Development Process

Traditional Approach

Planning → REQ → Design → DEV → Test → Implement

COTS Approach

Planning → REQ → Design → DEV → Test → Implement

Demos for Users

Configuration

Users Exercise System

Users Exercise System

Accelerated Delivery
Planning

Identify clear project goals

• Focus on business objectives and goals
• What are you trying to accomplish?
  - Improving productivity
  - Increasing revenue
  - Reducing cost
  - Supporting regulatory requirements

• All system requirements and technical decisions should consider business goals

Remember you are trying to solve a business problem not engineer a new software module
Planning

Prepare for change

- Plan project using multiple phases/iterations
- Plan for heavy user interaction and validation of capabilities
- Allows for integrating new COTS releases
- Provides an opportunity for intermittent training

Phases or iterations should focus on completing a use case or scenario
COTS Based Phased Planning

Phase 1: Out of Box
- Implement base COTS release
- Data Model – first release
- Workflows - initial configuration
- Usage Patterns – Identify
- User Roles – Define
- Architecture - establish initial infrastructure

Phase 2: Refine
- Implement incremental COTS Release
- Data Model – modify based on input
- Workflows – elaborate and create new
- Usage Patterns - refine
- Architecture - Integrate with other systems

Phase 3: Finalize
- Implement Final COTS Deployment Release
- Data Model - finalize
- Workflows - finalize workflows
- Architecture – plan for system implementation
- Training – develop plan

Continuous Learning & Improvement

25
Requirements

Do not start with a blank slate – understand & leverage existing COTS functionality

- Focus on the “what” NOT the “how”
- Embrace the change – do not modify COTS technology to support an inefficient process
- Keep it simple
Design

Orchestration of COTS functionality

- Short focused interactive design sessions
- Heavy use of prototyping and lab environment
- End users along with COTS experts
  - Defining user roles
  - Defining workflows
  - Identifying efficiencies
  - Understanding end user access
- Consider future COTS releases

💡 Design should emphasize delivering immediate capabilities
COTS Design/Configuration Process

Configure
- Build base workflows
- Streamline COTS usage
- Obtain user feedback

Design
- Configuration Team focuses on workflows
- User role definition
- Usage patterns

Prototype
- COTS experts fully engaged with users
- Review existing COTS tools
Develop

Configuration of workflows and business rules

- Based on user input and feedback
- Targeted to configuration not customization
- 90% configuration 10% (or less) development
Test

Functionality proven via use case validation

- Frequent user validation
- Multiple user groups to verify usage patterns
- Prioritize user feedback – leverage phases
- Use validation as training opportunity

Real time feedback to Configuration Team
Implementation

Consider COTS release schedule in implementation plan

- Understand you are changing process not just technology
- Leverage COTS training
- Data flow patterns
- IT Specifications
- Security considerations
- Administration
  - Who? What? Skills?
- Don’t forget about the data!
COTS Based Phased Implementation

Pilot Phase
- Training Users
- Migrate to new data model
- Verify Security Model
- Implement Base Architecture
- Migrate Data for AOI

Deployment Phase 1
- Configure Workflows
- Expand Workforce Training
- Other Workgroup Users
- Refine Operational Procedures
- Ramp Data Migration

Deployment Phase 2
- Implement Incremental COTS Release
- Execute Workforce Training
- Expand to Other Users
- Complete Data Migration
- Refine Operational Procedures
Summary
Benefits of a COTS Approach

- Minimized cost to implement
- Shorter schedule – immediate exercising by users
- Reduced risk (technology proven by user-base)
- Lowered maintenance costs
- Evolves with future COTS functionality
- Wider range of qualified people to use and maintain system
Summary

• COTS configuration is a more efficient way to approach system development

• The COTS Approach redefines system development process, with focus on mission instead of building software

• ArcGIS 10 provides a platform for configuration and rapid development – making the COTS Approach viable for many systems
Thank you!

More Info:
- COTS Approach ArcNews article
- COTS Approach white paper

Contact us:
- Greg Pleiss, gpleiss@esri.com
- Don Hong, xhong@esri.com

• Please fill out the online evaluation
  - http://www.esri.com/sessionevals
Questions?