Starting Your PLM Implementation

Methods for success

www.aras.com
Agenda

- Getting Organized
- Resources
- Getting Started
- Project Breakdown
- Some Tips
- How Things get Sideways
- Our Recommendations
Getting Organized

► Organize your thoughts!
► Gather all relevant materials
  ▪ Existing forms
  ▪ Flow charts
  ▪ SoPs
► Identify Resources
► Keep an open mind with regards to change
  ▪ Square Peg in a Round hole
## Resources
Who do you need

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
</table>
| Project Manager    | • Direct Implementation resources  
                     • Manage project schedules  
                     • Track Status  
                     • Resolve conflicts and issues                              |
| Business Process Owner(s) | • Provide project priorities and objectives  
                             • Direct participation of resources  
                             • Resolve business process issues |
| Subject Matter Experts | • Communicate current process  
                          • Provide information details  
                          • Support user community during rollout               |
| I.T. System Support | • Support site infrastructure  
                         • Extract legacy data  
                         • Provide technical expertise                             |
| Technical Resources | • Configure application  
                         • Develop customizations  
                         • Provide technical expertise |
Getting Started

▶ Develop a Project Charter if you need funding
  ▪ Lets management know what you will do and will not do
  ▪ Defines a goal and an endpoint to measure success
  ▪ There are plenty of examples on the web

▶ Develop a Project Plan
  ▪ Lets resources know what is expected of them
  ▪ Sets the schedule for the project

▶ Get Trained
  ▪ The team needs to understand Aras Innovator

▶ Review the standard Aras Innovator Solutions
  ▪ Required for effective gap analysis
Organizing your Project

- Break your project down into phases
  - Requirements & Design
  - Construction
  - Transition
- Phases can overlap
- Work is done in a serial fashion but you don’t need to finalize a phase before moving on
Elaboration Phase
Requirements & Design

- Develop a Requirements Document
- Develop Use Cases
- Conduct Gap Analysis
- Develop design document or functional spec
Requirements Documents

Uggggh ! Really?

► They are not a waste of time even if you are the developer

► This includes “What” & “Why” for the system
  - “What” are the system requirements
  - “Why” are they requirements

► Should include high level use cases

► Intended for technical resources to understand business case

► Used as a foundation for other documents
Use Cases

Pay me now or pay me later

- More detailed than in Requirements Document
- Documents the interaction between user and system
- Start at a high level and add detail as visual prototype evolves
- Don’t worry about getting these 100% correct
- Shoot for 70-80%, then iterate with visual prototype
- These are used later to build test plans, training materials & user documentation
- Provides a point of reference for change management later
More on Use Cases

▶ Always build an index of use cases
▶ Need to be reviewed by Business Process resources
▶ Samples are available on the web
▶ A good reference
  ▪ Writing Effective Use Cases by Alistair Cockburn
▶ You will likely write 2 types of use cases
  ▪ High level (Clouds)
  ▪ Detailed (Sea Level)
Use Case Samples

Sample Index

<table>
<thead>
<tr>
<th>Implementation Phase</th>
<th>UC Number</th>
<th>Use Case Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 1</td>
<td>UC0001</td>
<td>Create Engineering Change Request</td>
</tr>
<tr>
<td>Phase 1</td>
<td>UC0002</td>
<td>Maintain Engineering Change Request</td>
</tr>
<tr>
<td>Phase 1</td>
<td>UC0003</td>
<td>Create Engineering Change Notice</td>
</tr>
<tr>
<td>Phase 1</td>
<td>UC0004</td>
<td>Maintain Engineering Change Notice</td>
</tr>
<tr>
<td>Phase 1</td>
<td>UC0005</td>
<td>Process Engineering Change Request</td>
</tr>
<tr>
<td>Phase 1</td>
<td>UC0006</td>
<td>Create new part/document</td>
</tr>
<tr>
<td>Phase 1</td>
<td>UC0007</td>
<td>Maintain Design Part</td>
</tr>
<tr>
<td>Phase 1</td>
<td>UC0008</td>
<td>Maintain Document</td>
</tr>
<tr>
<td>Phase 1</td>
<td>UC0009</td>
<td>Approve Change via Fast Track</td>
</tr>
<tr>
<td>Phase 1</td>
<td>UC0010</td>
<td>Approve Change via CRB (Change Review Board)</td>
</tr>
</tbody>
</table>

Sample Document

Adobe Acrobat Document
Design Specifications

- This is the “How” to address the requirements
- You can determine how detailed this needs to be
- Documents data model changes
  - Use Itemtype definition report to document items
- Document all the Events and methods required
  - This will save time later when trying to diagnose issues
- Used for long term understanding and maintenance
  - Very important if you plan to subcontract any work
Construction Phase

- Visual Prototypes
  - Primarily used for validating use cases and user feedback

- Behavioral Prototypes
  - Adds automation and customizations

- Data Migration

- Integrations and Interfaces

<table>
<thead>
<tr>
<th>Disciplines</th>
<th>Activities</th>
<th>Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Use Case Development</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Solution Design Doc</td>
<td></td>
</tr>
<tr>
<td>Analysis &amp; Design</td>
<td>Functional specs for customizations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Screen Mockups</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Visual Prototypes</td>
<td></td>
</tr>
<tr>
<td>Implementation</td>
<td>Import Users</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Configure Items</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Configure Forms</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Configure Workflows</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Configure Roles &amp; Permissions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Behavioral Prototypes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Develop Data Migration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interface development</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Develop Customizations</td>
<td></td>
</tr>
</tbody>
</table>
Visual Prototypes
Putting it all together

➤ This is just building things with Aras Innovator
  - Items, Forms, Workflows, Lifecycles, etc

➤ Used to solicit user feedback

➤ Built in conjunction with use cases and requirements
  - May cause you to revisit use cases and specifications

➤ Should not include automation (significant automation)

➤ Spend significant time reviewing these with users
Behavioral Prototypes
Making it all work

▶ Adds automation to the Visual Prototype
  ▪ Will likely cause you to revisit use cases and specifications
  ▪ Will introduce changes to the Visual prototype
▶ Includes building interfaces and integrations
▶ Includes unit testing
▶ Everything you need to begin full system test
# A Complete Look

<table>
<thead>
<tr>
<th>Disciplines</th>
<th>Activities</th>
<th>Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Startup</strong></td>
<td>Install Test Environment, Train Project Team, Kickoff Meeting</td>
<td></td>
</tr>
<tr>
<td><strong>Business Modeling</strong></td>
<td>High Level Requirements, Discovery workshop(s), SoW, Project Charter &amp; Solution Overview, High Level Use Cases</td>
<td></td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td>Requirements Workshops, Requirements Document(s), Gap analysis</td>
<td></td>
</tr>
<tr>
<td><strong>Analysis &amp; Design</strong></td>
<td>Use Case Development, Solution Design Doc, Functional specs for customizations, Screen Mockups, Visual Prototypes</td>
<td></td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td>Import Users, Configure Items, Configure Forms, Configure Workflows, Configure Roles &amp; Permissions, Behavioral Prototypes, Develop Data Migration, Interface development, Develop Customizations</td>
<td></td>
</tr>
<tr>
<td><strong>Test</strong></td>
<td>Prepare Test Plan, Unit Test, Full System Test</td>
<td></td>
</tr>
<tr>
<td><strong>Deployment</strong></td>
<td>Prepare User Docs, Develop User Training, Deliver User Training, Build &amp; Deploy Prod Env, Production Data Migration</td>
<td></td>
</tr>
<tr>
<td><strong>Configuration Management</strong></td>
<td>Packaging, Migrate Dev-QA-Prod, Change Management</td>
<td></td>
</tr>
<tr>
<td><strong>Project Management</strong></td>
<td>Initial Project Plan, Initial Risk Analysis, Project Planning &amp; Design Review</td>
<td></td>
</tr>
</tbody>
</table>

- **Inception**
- **Elaboration**
- **Construction**
- **Transition**
  - Initial
  - Parts
  - Change
  - Parts
  - Change
  - Interface
  - T1
  - T2

---

**Milestones**

---

**TIME**

---

* aras.com
* Copyright © 2011 Aras All Rights Reserved. Slide 16
Some Tips

- Keep permissions wide open to start
  - make a world can edit permission to facilitate the first user reviews then implement permissions

- Start with your items in a single folder on the toc

- Allow world to create most items

- Adjust form sizes in the beginning
  - it’s a better user experience

- Don’t worry about getting it 100% right
  - You Wont…. And Aras Innovator is good at making change
How Things Get Sideways

- Lack of requirements understanding or agreement on requirements
- Lack of understanding of the standard Innovator solutions
- Lack of Training
- Understanding the impact of change
  - Workflow or Lifecycle changes can impact method
- Biting off too much!
Our Recommendations

▶ Training is a MUST
▶ Engage Aras or a partner for jump start activities
  ▪ Requirements review
  ▪ Leverage our experience to recommend approaches or discuss what has been done before
  ▪ Use case development
  ▪ Initial Design Review
  ▪ Periodic reviews and questions
▶ Solve a real problem and move on to the next
Final Thoughts
Do’s and Don’ts

▶ **DO**
  - Create visual prototypes and get user validation before developing any method code
  - Develop accurate Use Cases and keep them up to date
    - They will save you time down the road!!
  - Look for “Small Wins” that provide business value

▶ **DON’T**
  - Spend a significant amount of time developing specs w/o prototyping the solution
  - Worry about not getting 100% of the detailed requirements up front: Iterate!
Starting Your PLM Implementation

Methods for success

www.aras.com