



*ACE 2011 International*

# Starting Your PLM Implementation

Methods for success

[www.aras.com](http://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 relevent 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



Role	Responsibility
Project Manager	<ul style="list-style-type: none"><li>• Direct Implementation resources</li><li>• Manage project schedules</li><li>• Track Status</li><li>• Resolve conflicts and issues</li></ul>
Business Process Owner(s)	<ul style="list-style-type: none"><li>• Provide project priorities and objectives</li><li>• Direct participation of resources</li><li>• Resolve business process issues</li></ul>
Subject Matter Experts	<ul style="list-style-type: none"><li>• Communicate current process</li><li>• Provide information details</li><li>• Support user community during rollout</li></ul>
I.T. System Support	<ul style="list-style-type: none"><li>• Support site infrastructure</li><li>• Extract legacy data</li><li>• Provide technical expertise</li></ul>
Technical Resources	<ul style="list-style-type: none"><li>• Configure application</li><li>• Develop customizations</li><li>• Provide technical expertise</li></ul>

# Getting Started

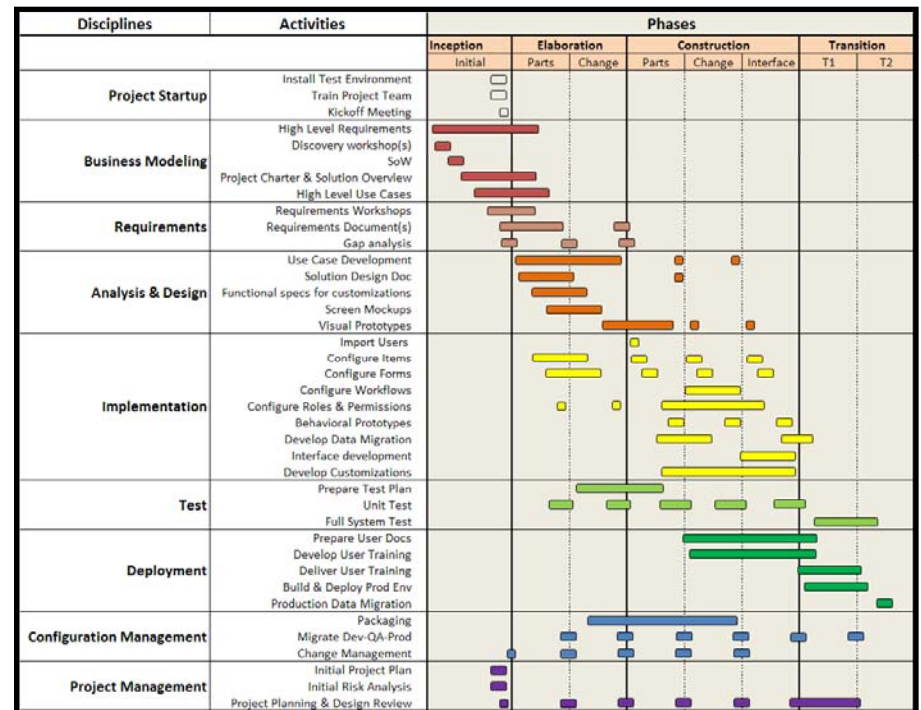


- ▶ **Develop a Project Charter if you need funding**
  - Lets management know what you will do and will not do
  - Defines a goal and an endpoint to measure success
  - There are plenty of examples on the web
- ▶ **Develop a Project Plan**
  - Lets resources know what is expected of them
  - Sets the 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

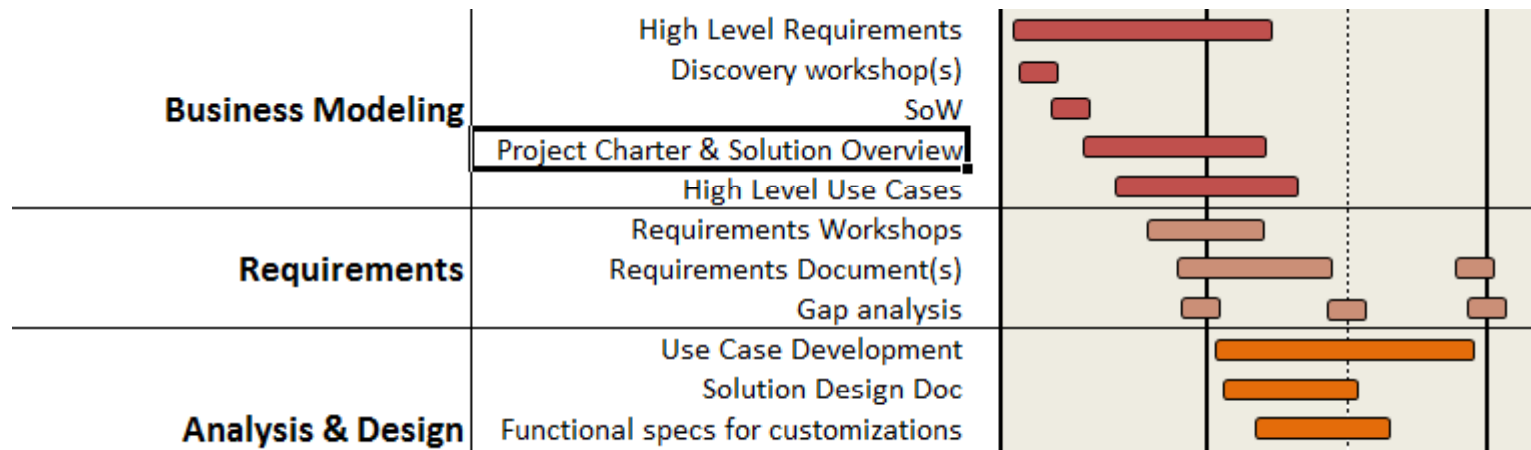


# Elaboration Phase

## 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

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

## ▶ 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

Disciplines	Activities	Phases					
		Inception	Elaboration		Construction		
		Initial	Parts	Change	Parts	Change	Interface
<b>Analysis &amp; Design</b>	Use Case Development		[Orange bar]				
	Solution Design Doc		[Orange bar]				
	Functional specs for customizations		[Orange bar]				
	Screen Mockups		[Orange bar]				
<b>Implementation</b>	Visual Prototypes			[Orange bar]		[Orange dot]	[Orange dot]
	Import Users				[Yellow dot]		
	Configure Items		[Yellow bar]		[Yellow bar]	[Yellow bar]	[Yellow bar]
	Configure Forms		[Yellow bar]		[Yellow bar]	[Yellow bar]	[Yellow bar]
	Configure Workflows		[Yellow bar]		[Yellow bar]	[Yellow bar]	[Yellow bar]
	Configure Roles & Permissions		[Yellow dot]	[Yellow dot]	[Yellow bar]	[Yellow bar]	[Yellow bar]
	Behavioral Prototypes				[Yellow bar]	[Yellow bar]	[Yellow bar]
	Develop Data Migration				[Yellow bar]	[Yellow bar]	[Yellow bar]
	Interface development						[Yellow bar]
	Develop Customizations				[Yellow bar]	[Yellow bar]	[Yellow bar]

# 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**

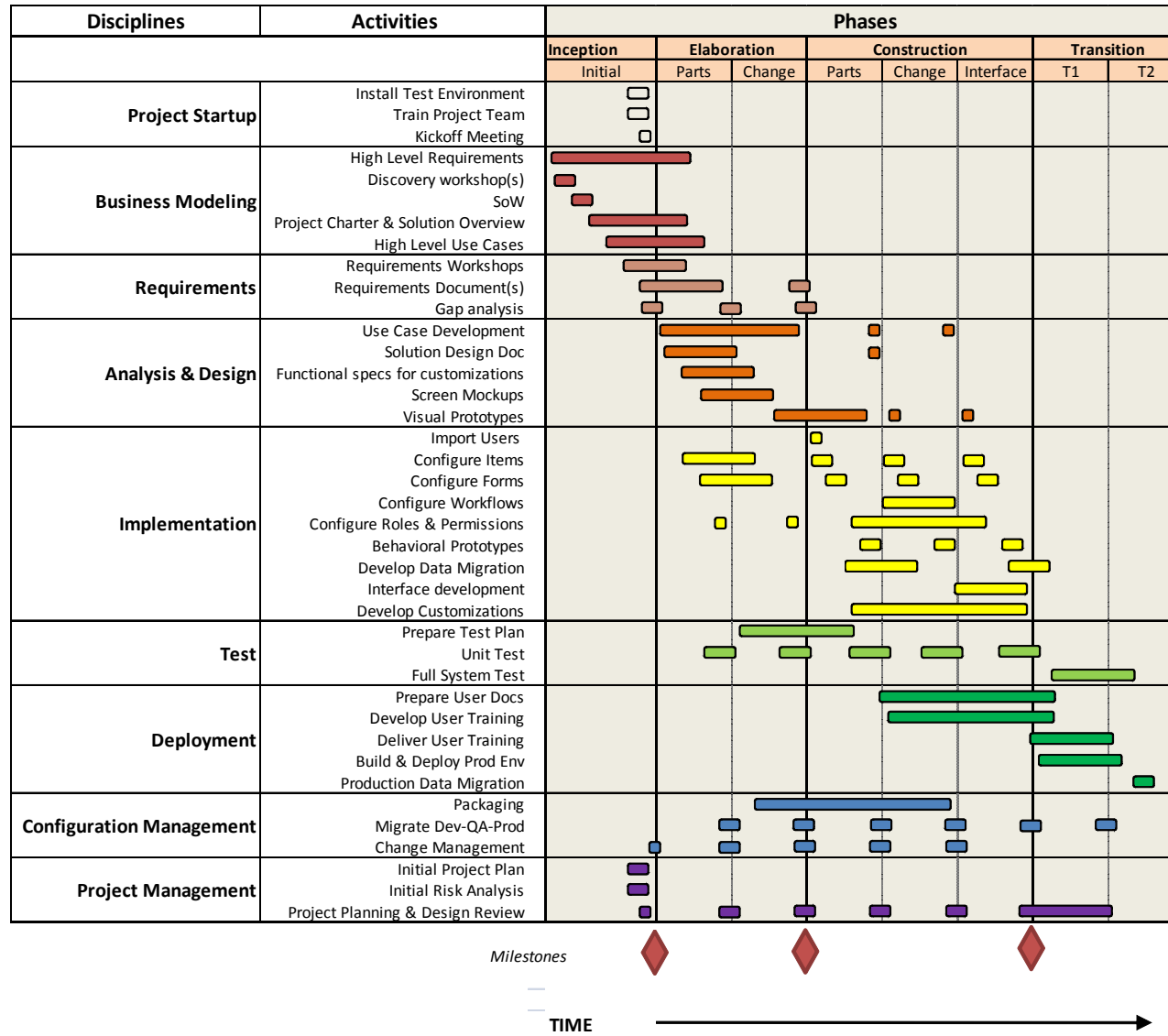
# Behavioral Prototypes

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





# 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 !



*ACE 2011 International*

# Starting Your PLM Implementation

Methods for success

[www.aras.com](http://www.aras.com)

