Upgrading Your SharePoint Applications

DEV372

Robert L. Bogue, MS MOSS MVP

- Robert Bogue, MOSS MVP, MCSE, MCSA: Security, etc.
  - Thor Projects
  - http://www.thorprojects.com
  - Rob.Bogue@ThorProjects.com
  - Latest Book: The SharePoint Shepherd's Guide for End Users

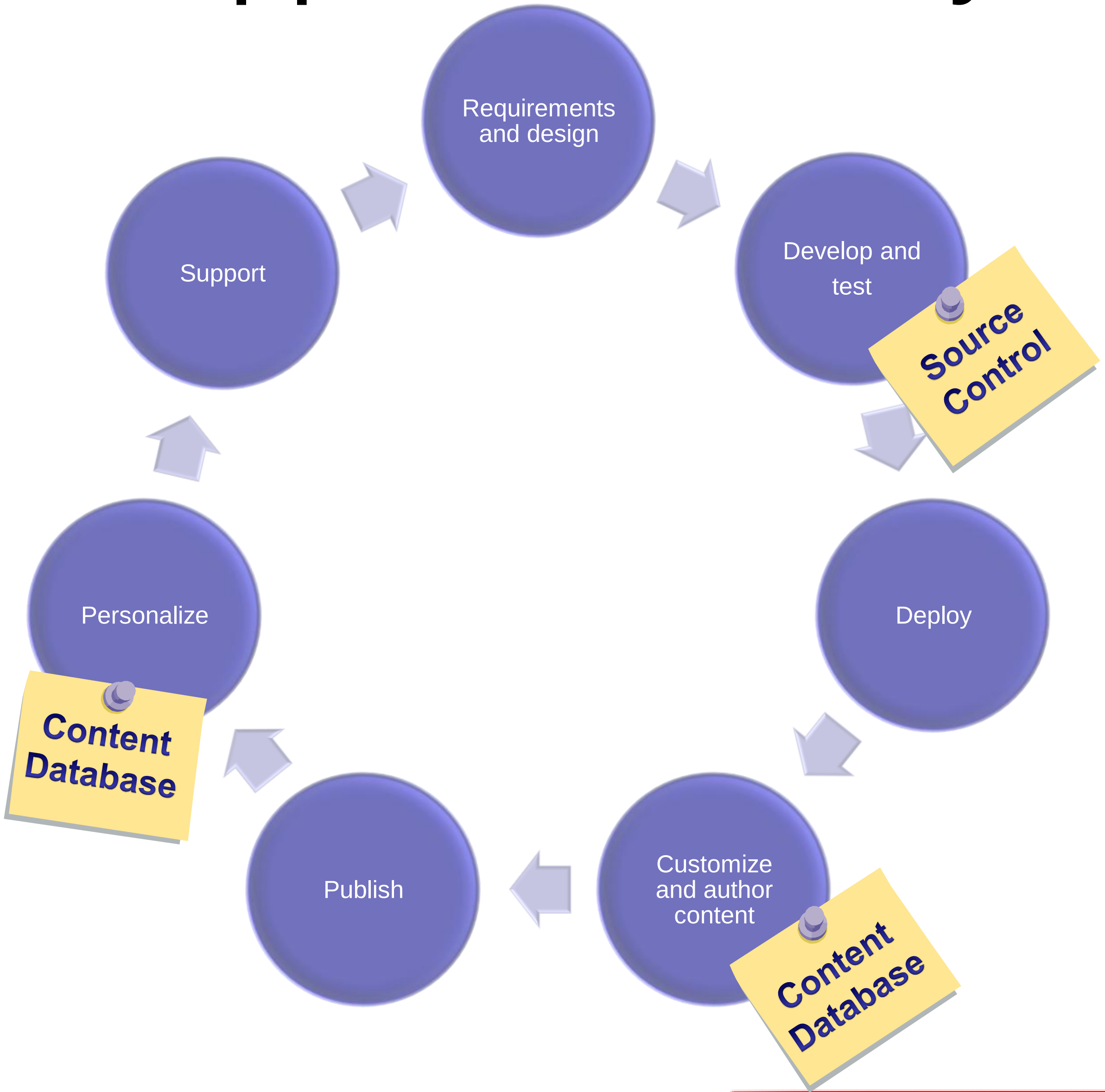
Agenda

- The challenges faced upgrading SharePoint applications
- Options to managing upgrading your application
- Upgradeable SharePoint assets

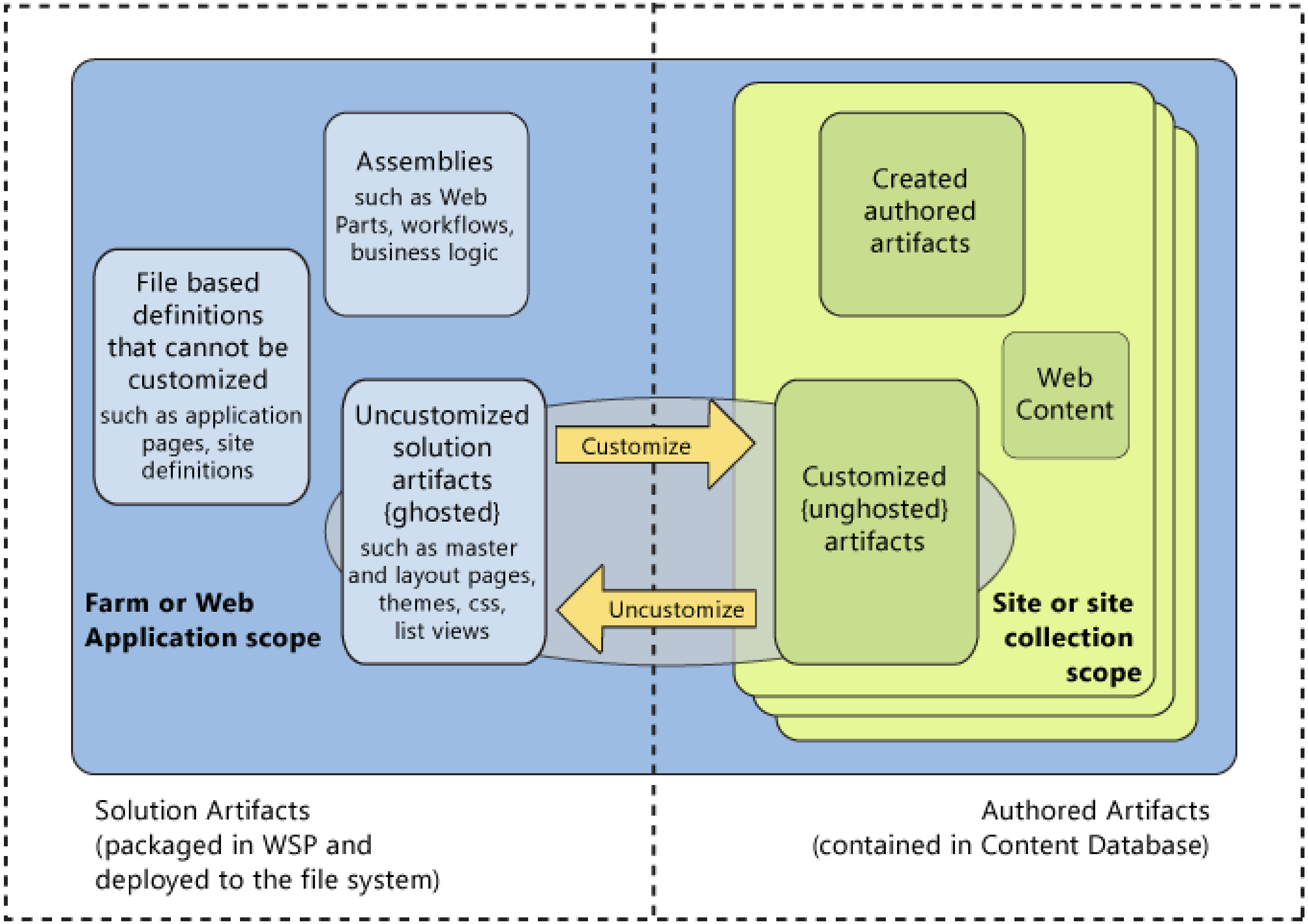
The Challenges facing upgrade

- Richer Application Lifecycle
- Code and customizations Relationship
- Feature and Solution factoring
- Approaches to installing upgrades

Richer Application Lifecycle



Code and Customization Relationship

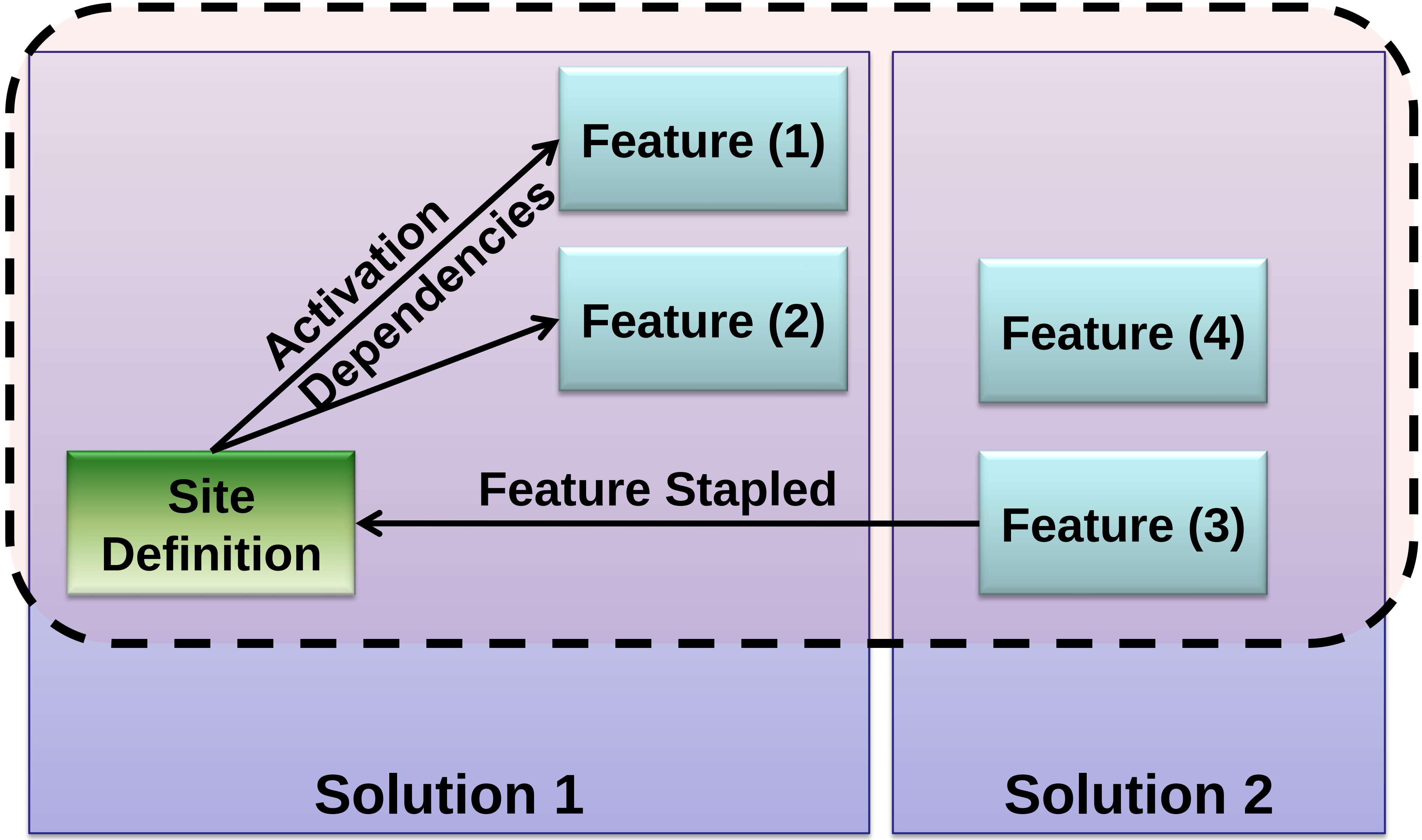

Assemblies
such as Web Parts, workflows, business logic
File based definitions that cannot be customized
such as application pages, site definitions
Uncustomized solution artifacts {ghosted}
such as master and layout pages, themes, css, list views
Customize
Uncustomize
Created authored artifacts
Web Content
Customized {unghosted} artifacts
Farm or Web Application scope
Site or site collection scope
Solution Artifacts (packaged in WSP and deployed to the file system)
Authored Artifacts (contained in Content Database)

Characteristics of Artifacts

- **Solution artifacts**
  - Files that contribute to an application
  - Packaged in a WSP
- **Authored artifacts**
  - Created with the browser or SharePoint Designer (SPD) that affect look/feel/structure
  - Reside in the Content Database
- **Web content**
  - Text and images created by content authors

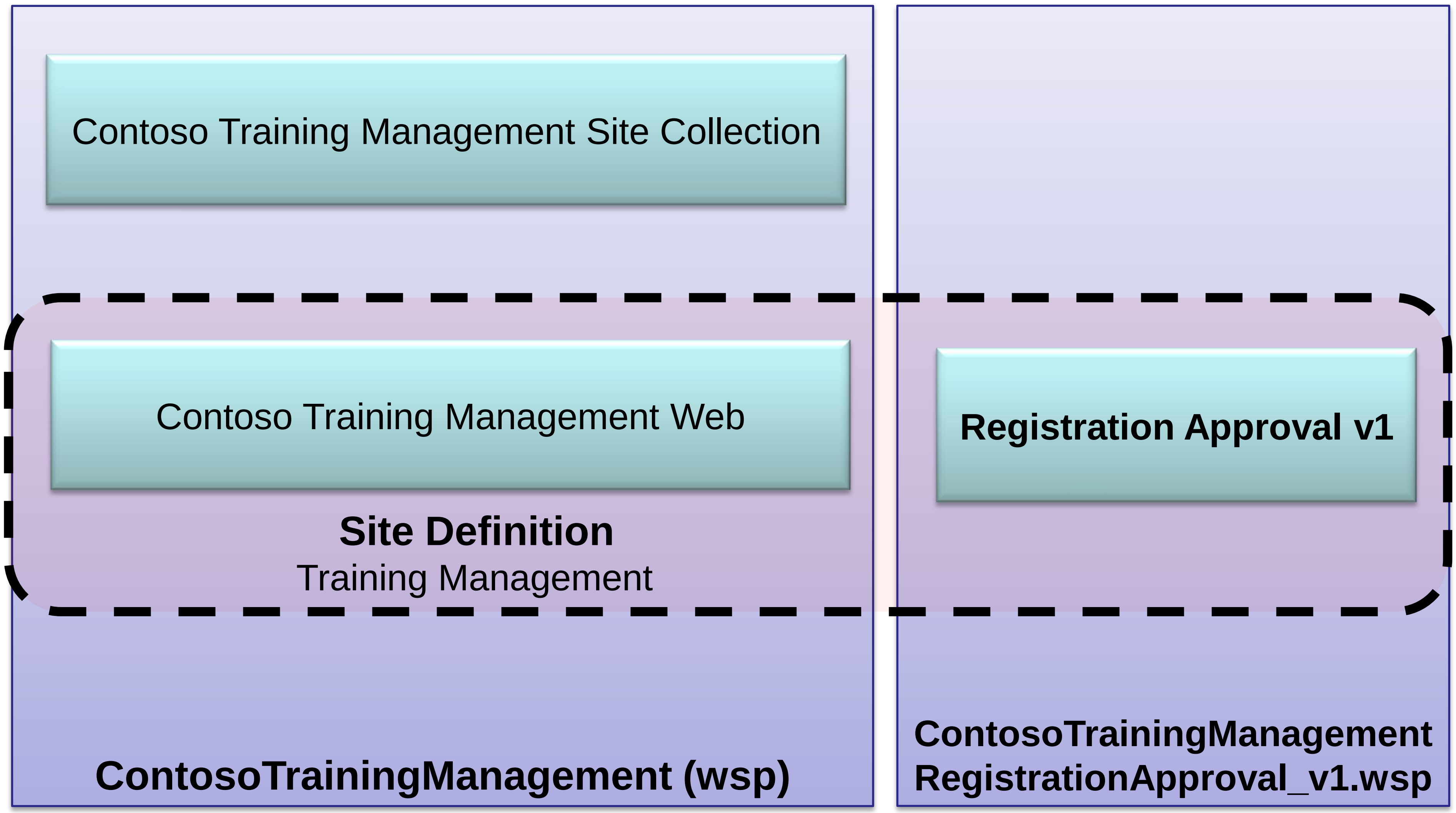
Feature and Solution Factoring

Relationship between sitedef, features and solutions



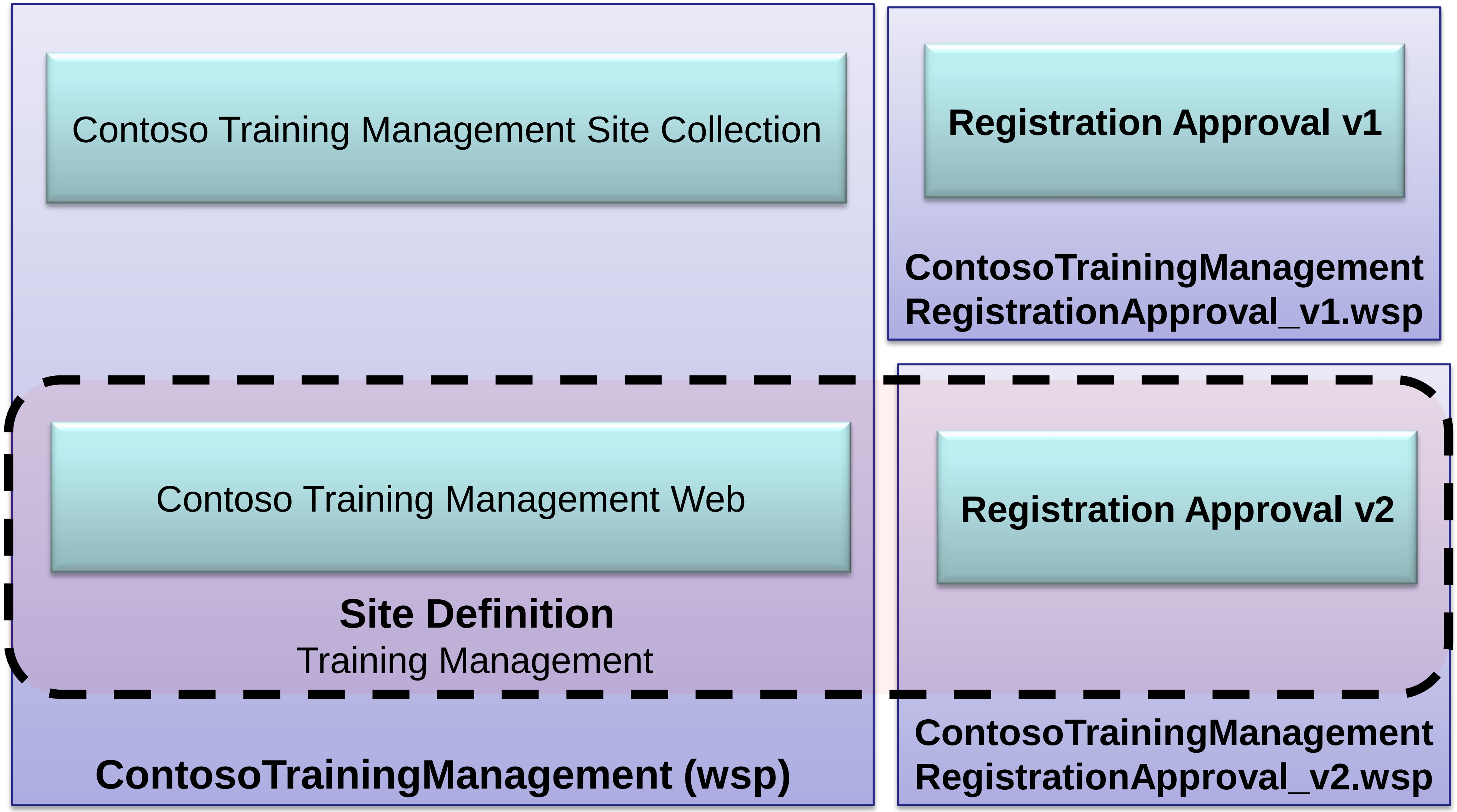
A Factoring Example

patterns and practices Reference Implementation


Contoso Training Management Site Collection
Contoso Training Management Web
Registration Approval v1
Site Definition
Training Management
ContosoTrainingManagement (wsp)
ContosoTrainingManagement
RegistrationApproval_v1.wsp

A Factoring Example

patterns and practices upgraded Reference Implementation


Contoso Training Management Site Collection
Registration Approval v1
ContosoTrainingManagement RegistrationApproval_v1.wsp
Contoso Training Management Web
Registration Approval v2
Site Definition
Training Management
ContosoTrainingManagement (wsp)
ContosoTrainingManagement RegistrationApproval_v2.wsp

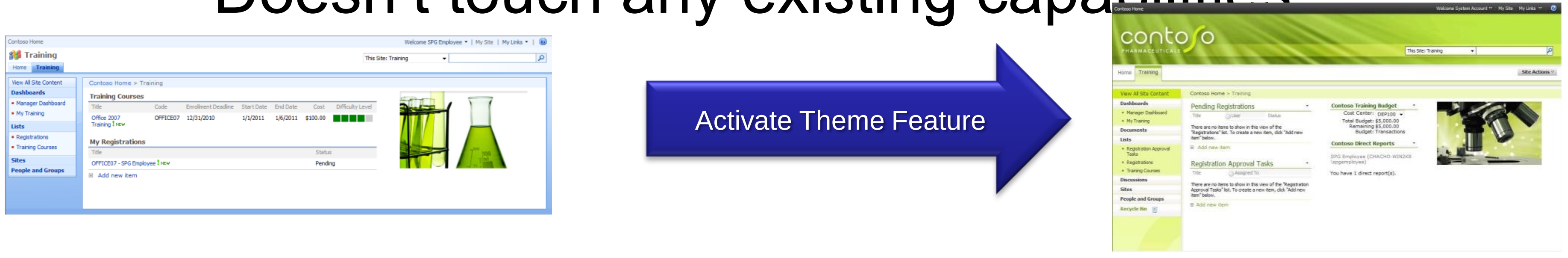
Types of Upgrades

- Adding net new functionality that constitutes a Feature
- Updating the existing features of the existing application
- Creating a new application version that can work side-by-side with an existing version

Approaches to Upgrading

Adding net new functionality that Constitutes a Feature

- The added functionality that logically makes a Feature
- The Feature does not impact installed Features
- Example: Contoso Theme
  - Can be re-used across any site
  - Doesn't touch any existing capabilities



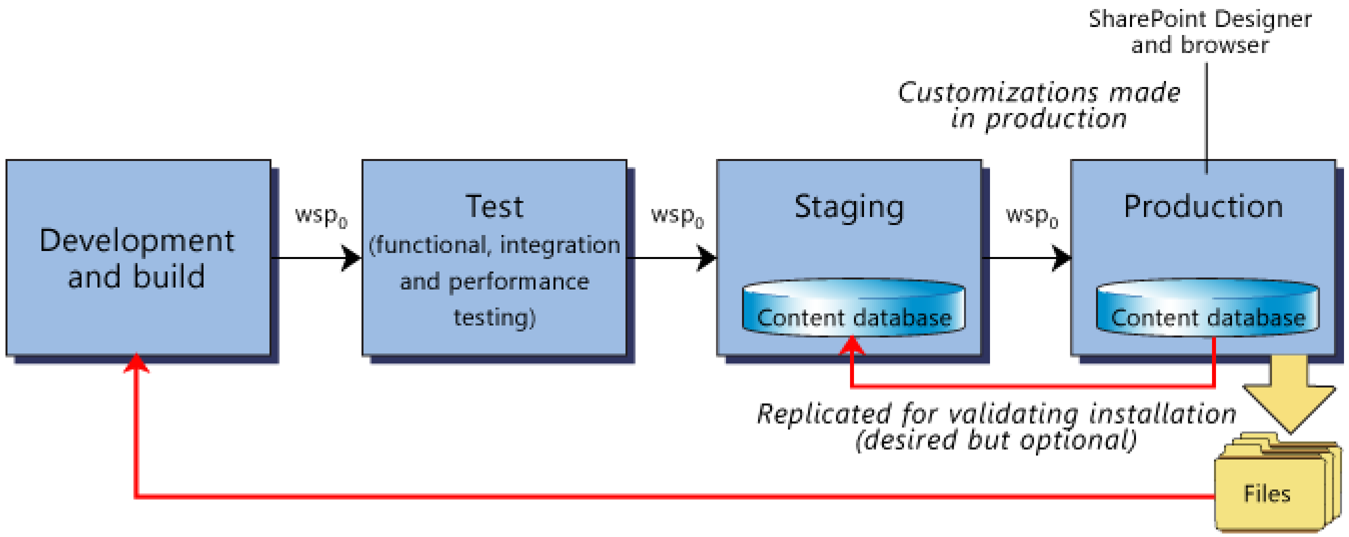
Approaches to Upgrading

Adding net new functionality

- Feature packaged and installed independently of existing application
- The Feature is applied to existing sites through the API or through the UI
- The Feature may be stapled to a site definition
  - Stapling will activate the feature for any newly created sites

Approaches to Upgrading

Updating existing features of the existing application


SharePoint Designer and browser
Customizations made in production
Development and build
wsp0
Test (functional, integration and performance testing)
wsp0
Staging
Content database
wsp0
Production
Content database
Replicated for validating installation (desired but optional)
Files

- Most types of artifacts may be upgraded using this approach
- Upgrade is logically "overlaying" files that are already created

Feature Installation and Activation

- Features are INSTALLED on every server (Server action)
  - Deploysolution handles feature installation
- Features are ACTIVATED once per scope (content DB action)

Deploying Upgrades for Solutions

- Retract and Redeploy
  - Use same ID's for solution and features
  - retract, delete, add, and deploy
  - Any added features will be installed
- Use "upgradesolution"
  - Use same ID's for solution and features
  - Performs a retract, file copy, and reinstalls assemblies to GAC
  - **Will not add new features** – can script install after upgradesolution executed.

Implications of Upgrading Applications

- Recommended approach: Reactivate features on sites and use a feature receiver
  - In many cases you must “fixup” aspects of modified artifacts through the object model
- Recommended approach: Deactivate features and use feature receiver
  - In some cases you will want to take actions when uninstalling to cleanup

Upgradeable Site Elements

- Site Columns
- Content Types
- Web Parts
- Files and Modules
- User Controls
- List Item Event Handlers
- Workflows
- Custom Actions
- List Templates
- List Definitions
- List Instances

Site Columns

- Upgrading sites with new site columns is accomplished through new site column definitions in a Feature
  - Must use DisplaceOnUpgrade attribute
- New site columns are available only after new activation or re-activation of a Feature
- Deleting site columns from a Feature not recommended.

Content Type

- Content type definitions can be created either via a Feature or via the UI.
- Lists maintain their own content type definitions in the content database.

…Content Types

Recommended

- For adding new fields to content types, add them programmatically using the SharePoint object model
- Remove fields by making the field hidden
- When programmatically upgrading content type use option to affect child content types including ones in lists

Not Recommended

- Replacing the content type definition on the file system of the Web front-end server
- Removing fields or updating the type of field in the content type definition

…Content Types

- When following the recommended approach outlined above, you cannot have a centralized XML file that contains the content type definition after the upgrade.

Web Parts

- Upgrade to web parts accomplished by deploying new version of the web part assembly
- Special considerations needed when assembly version number of the web part assembly is changed

…Web Parts

- Update the “SafeControl” element in the Web.config file of the SharePoint web application to reflect the new assembly version
- Iterate all web applications, all site collections, all webs, and all pages to update references
- Update the .webpart file for the Web Part with the new assembly version

Files and Modules

- Un-customized files can be upgraded by replacing the file with the new version in the Features
  - Changes to the file are detected after an application pool recycle
  - ~~One downside to this approach is that information about the file, such as its size and properties may not correctly reflect the actual information about the file~~

…Web Part Pages

- Additional considerations for Web Part Pages
  - Upgrades to Web Part Pages can include list view and other Web Parts
  - Web Parts previously provisioned through a Feature are not removed or merged when upgraded which leads to duplicate Web Parts

…SPFile.MoveTo

- One approach to correcting duplicate Web Parts is to provide a new name for the file being upgraded and to use the "SPFile.MoveTo" method in the SharePoint object model
  - This approach will cause you to lose all user customization to the file and is not recommended where heavy user customization will occur

…Other Approaches

- Programmatically identify and remove duplicate Web Parts through a Feature receiver
- Programmatically add new Web Parts through a Feature receiver

…Deleting files

- Deletion of files from a site can be accomplished by using the “SPFile.Delete” method in the SharePoint object model

Application Pages

- Application Pages are .aspx files stored in the virtual _layouts folder of the SharePoint Web front-end server
  - These pages cannot be customized and therefore are never stored in the content database
  - To upgrade an Application Page, deploy a new version of the .aspx file to the Web front-end server. Updates will appear after a process cache refresh

User Controls

- User controls are upgraded by
  - replacing the existing .ascx file with a new version
  - Upgrading the code behind assembly

List Item Event Handlers

- List Item Event Handlers are compiled as Assemblies and new versions can be deployed to the Web front-end servers
- What happens when the assembly version number is changed?
  - The assembly information for event handlers is persisted in the content database for each list instance and must be updated through the SharePoint object model.

Workflow

- Changes to workflow code must accommodate any existing workflow instances
  - Changes to the workflow properties or activities can cause SharePoint’s workflow serialization engine to fail

…Workflow

- In scenarios where significant changes to workflow is required, including updates to workflow properties and activities
  - Assign a new assembly version number for the workflow assembly
  - Provide a new workflow template definition for the new workflow assembly
  - Create a new workflow association using the new workflow template
  - Set the workflow association of old versions of the workflow to “No New Instances”

Custom Actions

- To upgrade an existing custom action, make changes to the feature element containing the custom action definition
  - The “Id” attribute of the “CustomAction” element must remain the same
- To remove a custom action, add a “HideCustomAction” element to a Feature

List Templates

- List templates are upgraded by deploying an updated list template definition in a Feature
- Certain properties such OnQuickLaunch, DisplayName, Description and Image can only be updated programmatically through the SharePoint object model

List Definitions

- List definitions are upgraded by deploying a new Schema.xml file in a Feature
  - Customized views cannot be upgraded through a new Schema.xml file
- Removing fields or modifying the field type from the Schema.xml file is not recommended
  - Create a new field to replace the deprecated field and mark deprecated field as hidden

List Instances

- New list instances can be provisioned to a site by providing a new “ListInstance “ element to a Feature
- Updates to any existing list instances must be performed programmatically through the SharePoint object model

Using Features

- Features allow you to create server-side, file system level customizations.
- Features allow for more modular changes to new and existing sites.
- Features make it easier to maintain and upgrade applications in SharePoint

Post conference DVD with all slide decks

Sponsored by

echoTechnology
Manage SharePoint

Thank you for attending!