en**vision** IT

Leaders in SharePoint Solutions

Application Lifecycle Management for SharePoint in the Enterprise

February 23, 2012

Agenda

- Introductions
- Purpose
- Visual Studio and Team Foundation Server
- Provisioning SharePoint Farms
- Agile with Team Foundation Server
- Continuous Integration
- Automated Testing
- Change Management

Peter Carson

- President, Envision IT
- SharePoint MVP
- Virtual Technical Specialist, Microsoft Canada
- Computer Engineering, UW
- peter@envisionit.com
- http://blog.petercarson.ca
- www.envisionit.com
- Twitter @carsonpeter

Purpose

- How do we build a solid SharePoint development environment for the enterprise?

- How do a number of developers work together on a SharePoint project?

- How does code and content move through a dev, test, and production lifecycle

- What is Agile and Continuous Integration, and why is it important?

Visual Studio and Team Foundation Server

- Visual Studio 2010 is the development platform for coding on SharePoint 2010
- Team Foundation Server integrates with Visual Studio to provide a repository for code, test, and other project artifacts
  - It facilitates teamwork by providing a common place for developers to work together

Provisioning SharePoint Farms
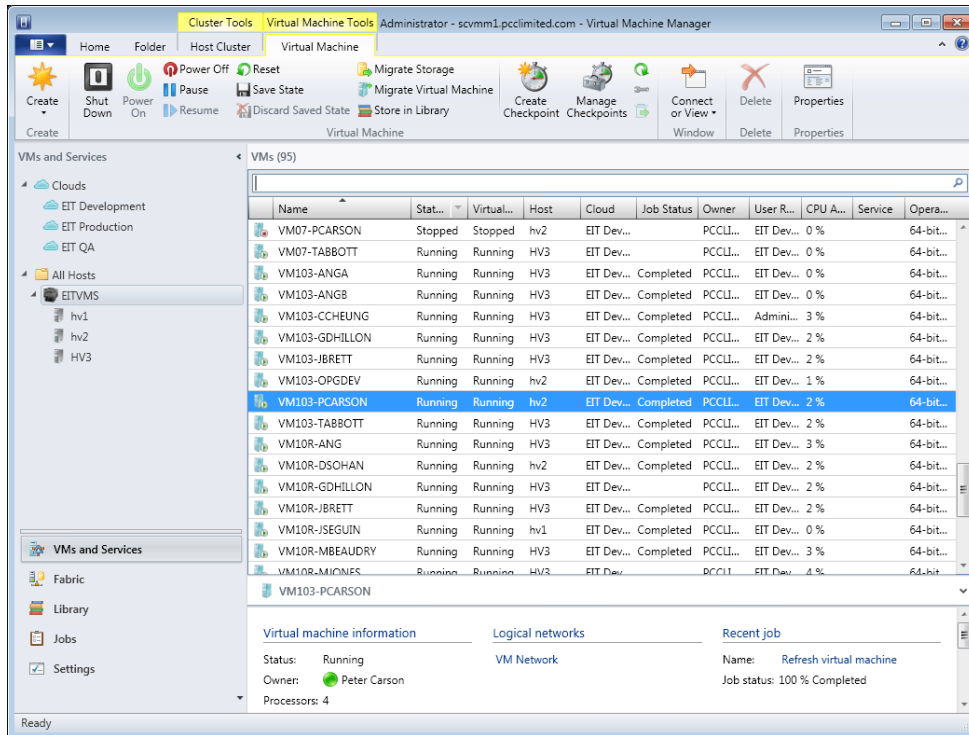
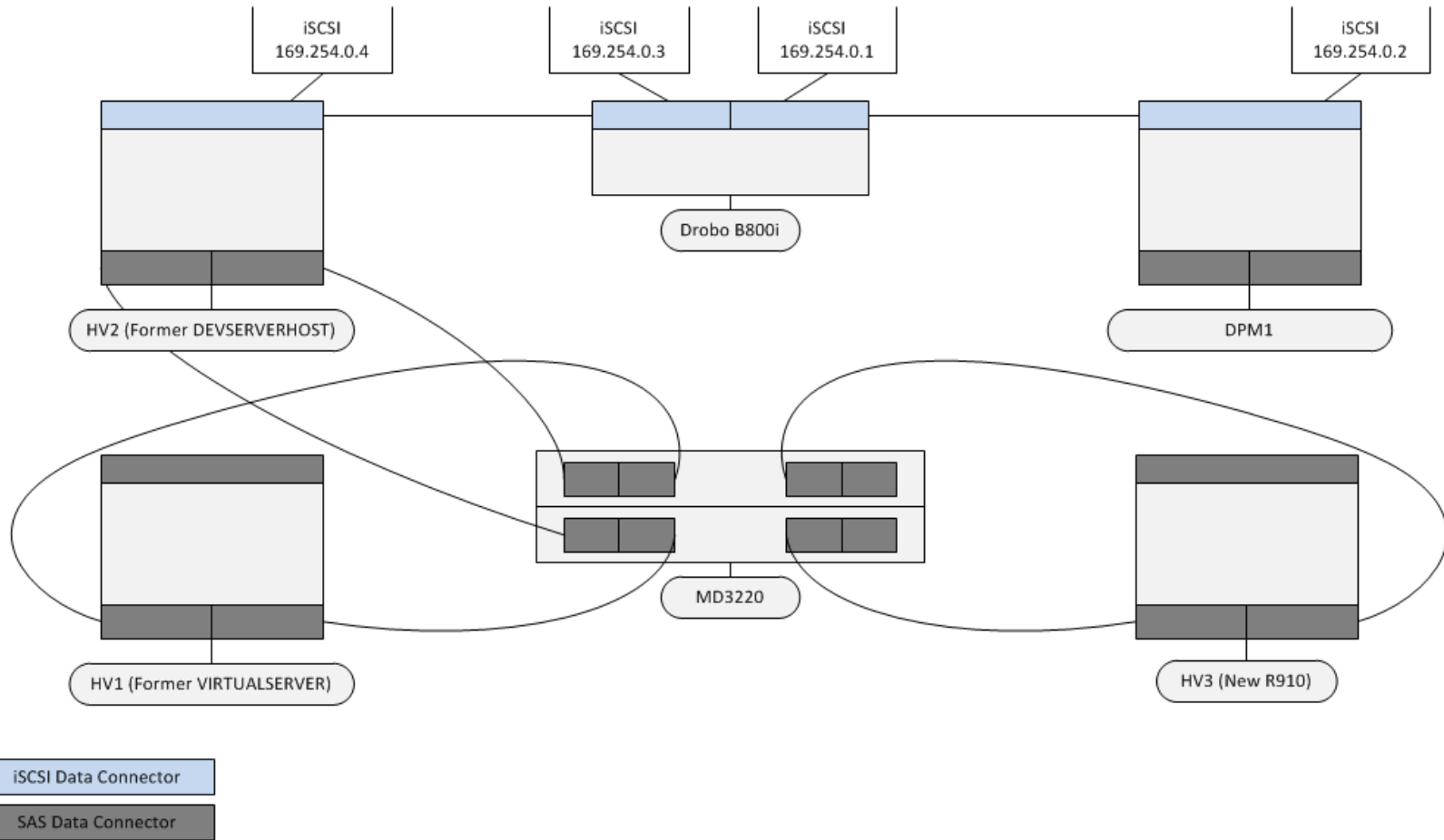# Provisioning SharePoint Farms

- Every developer needs their own complete SharePoint farm

- Teams need a dev integration farm to bring their work together on

- A full architecture needs QA and production farms as well

Envision IT Virtualization Architecture

Developer VM Template

- SharePoint Server 2010 Enterprise
  - SP1 and August CU
- Visual Studio 2010 Premium SP1
- Office 2010 Professional Plus SP1
- SharePoint Designer 2010 SP1
- Fiddler
- SQL Server 2008 R2 SP1

Creating a new Developer VM

- Open System Centre Virtual Machine Manager
- Create Virtual Machine
- Choose the Developer VM template
- Set the virtual machine name and options
- Create and start the VM

Creating a new Developer VM - Step 2

- Developer RDPs to their new VM
- Complete the SQL Server installation
  - Create the default instance
  - Set the service account(s)
  - Set the authentication mode and server administrators
- Update the AutoSPInstallerInput.xml file
  - Set the machine name for the default sites and database server
  - Set the service account passwords
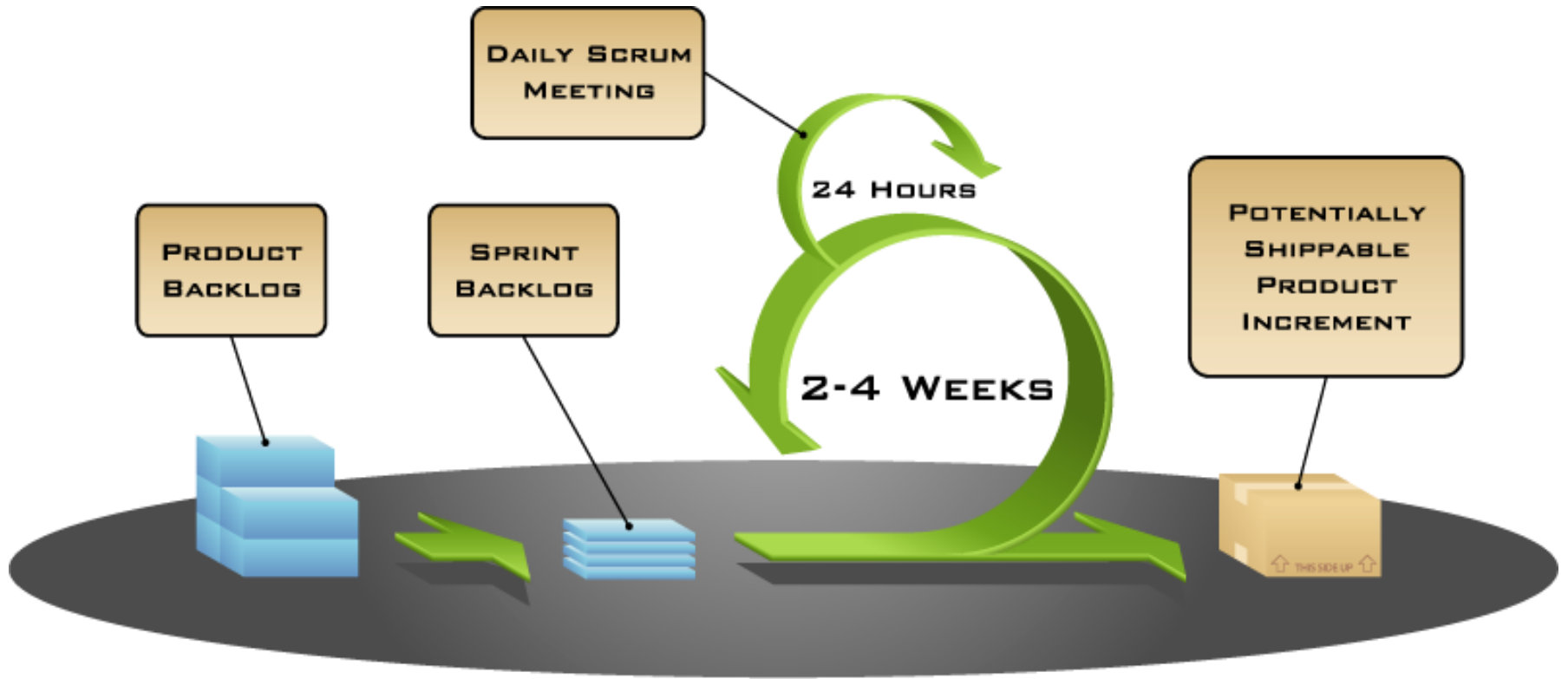- Run the AutoSPInstallerLaunch batch file
- Do a Health check review

envision IT
Leaders in SharePoint Solutions

Creating a new Developer VM - Step 3

- Start Visual Studio

- Connect to the Team Foundation Server

- Setup your workspaces

- Use a standard naming convention across your team
    - C:\Development*TFSCollection*

envision IT

Leaders in SharePoint Solutions

Other Environments

- Dev Integration, QA and Production can all be built using AutoSPInstall

- Don't need or want the Visual Studio or Office tools on these servers

- We have a base template we use for dev integration servers

- QA should match the Prod architecture as closely as possible

  - High availability with load balancing, clustered SQL, app servers, DMZ, hardening, Internet publishing

Agile Development

COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Agile with Team Foundation Server

Creating Your TFS Project

- Open Visual Studio
- Make sure you are connected to the right Team Project Collection
- In Team Explorer, right-click the top collection and create a new Team Project
- Use the Scrum 1.0 template
- Go into the Areas and Iterations
  - Delete the extraneous releases and sprints
  - Create a product backlog iteration
- Start adding your stories and tasks

Continuous Integration

Continuous Integration

Automated Testing

SharePoint Site Elements

- Site structure
  - Site columns, content types, subsites, lists, libraries
- Branding elements
  - Master pages, layouts, CSS, XSLT, supporting images
- Custom web parts
- Page and site content
- Web part instances
- Search
- Workflows
- Business Connectivity Services
- Managed Metadata

SharePoint Site Elements – Part 2

- Access, Excel, Word, Visio

- PerformancePoint

- Document conversions

- Timer jobs

- Secure Store Service

- User profiles

Site Structure

- Site columns
  - Define the fields that are used by content types
- Content types
  - Defines collections of site columns for different types of content
- Subsites
  - The hierarchy of sites that make up the site collection
- Lists and Libraries
  - The lists and libraries for each subsite

Site Structure – Part 2

- Site columns and content types can be programmatically or manually created
  - If they are going to be used in multiple site collections they should be programmatically created
  - If a single site collection, they can be created manually in a master site collection
  - GUIDs cannot be manually set, and code may need them to be consistent
- You can create them manually in SharePoint, save the site as a template, and import them into Visual Studio to create a proper WSP package

Site Structure – Part 3

- Site structure can also be done either programmatically or manually
  - Many sites are complex and it is error prone to create manually
  - We use our own SiteModelBuilder tool to create the site structure

Site Structure – Part 4

- For all site structure elements, changes can also be manual or programmatic
  - It is useful to be able to delete the whole site collection and recreate it programmatically
  - As soon as content starts being authored in SharePoint, it becomes difficult or impossible to do that
- After content authoring has started
  - Site column and content type changes need to be programmed explicitly (you can't just drop the content type and recreate it)
  - Site structure changes are usually done manually at this point
  - Changes need to be backed up and restored

Change Management
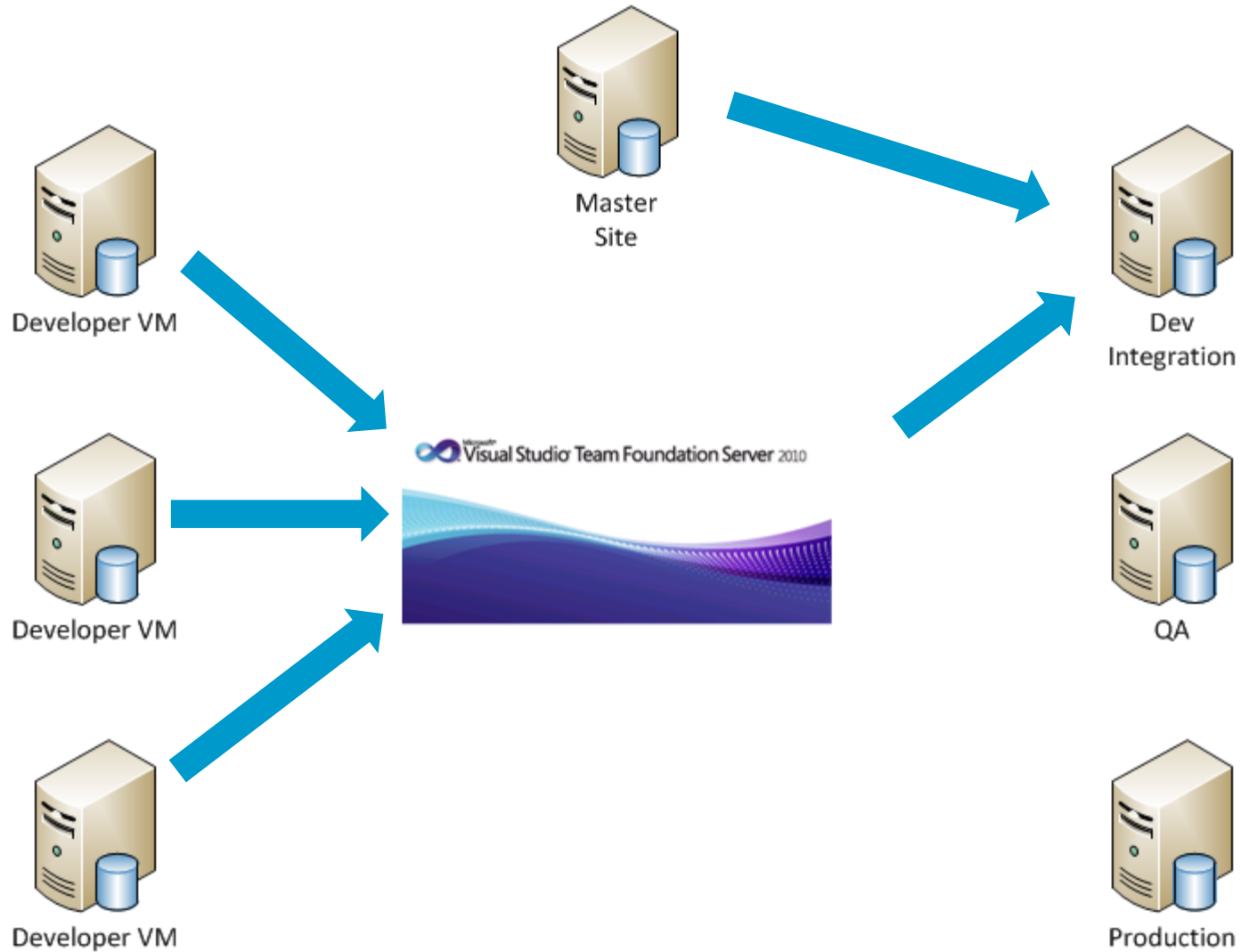
Branding Elements

- Web developers prefer to work in SharePoint Designer
  - Faster to make and see changes
- Developer is free to use SPD in their own VM
  - Working against a restored copy of the master site
- When they are ready to deploy, they package their files in a Visual Studio project
  - Results in a WSP solution file
  - This is deployed to the dev integration and other environments
- Developers never use SharePoint Designer against the master or dev integration sites

Branding Elements – Part 2

- Elements are typically in one of two modules in a branding feature
- Master page
  - Master pages
  - Layouts
- Style Library
  - CSS
  - Images
  - JavaScript
  - XSL

Page and Site Content

- Page and site content is entered directly into SharePoint by the content authors
  - Web pages, images, PDFs, list content, etc.
  - This should be done in the master site separate from where dev integration is happening
  - Care needs to be taken with this site once authors start working
  - Make sure it is part of your production backup plan
- Periodically back it up and restore it to the dev integration and developer VMs
- Ultimately it will be backed up and restored to the QA and production farms

Change Management
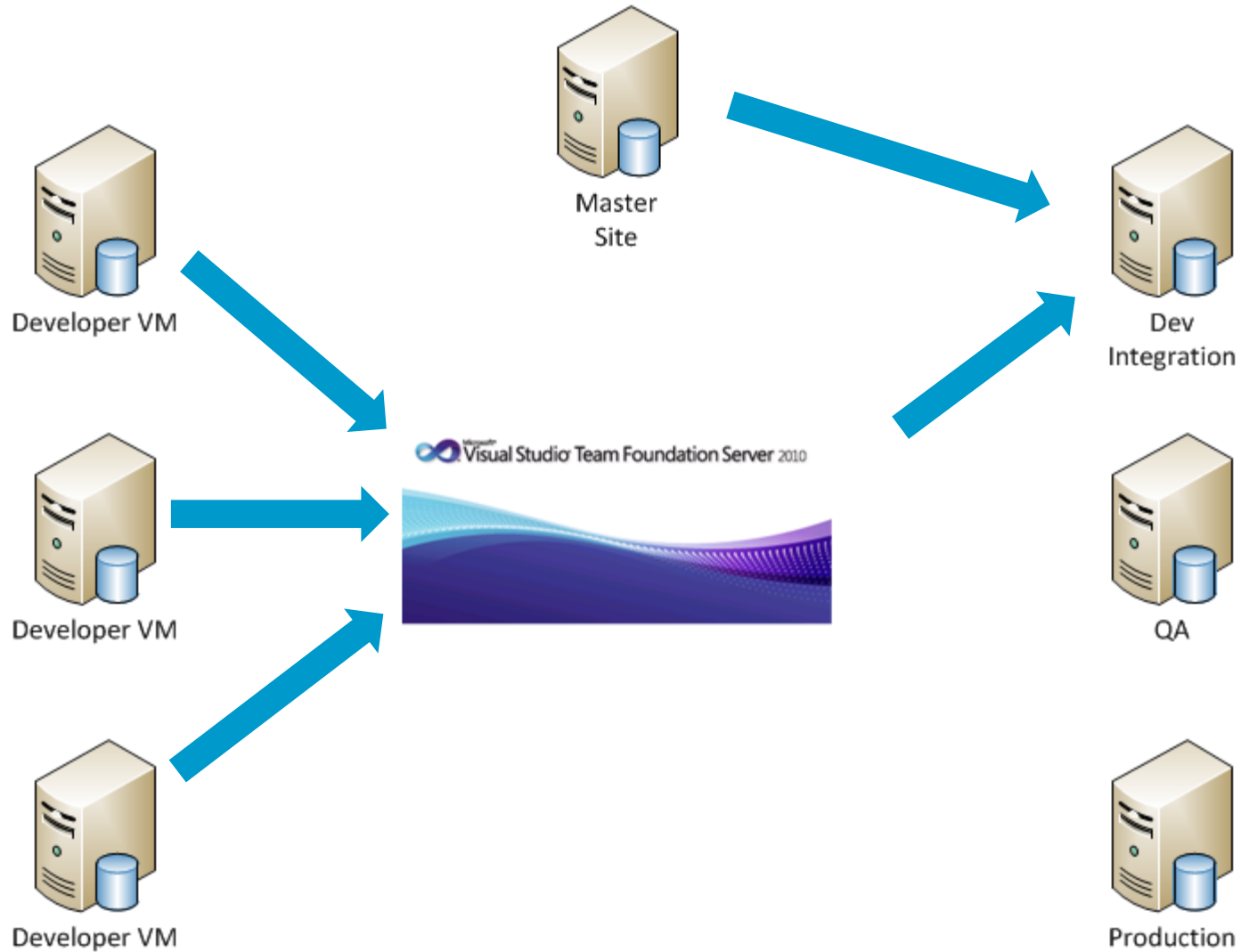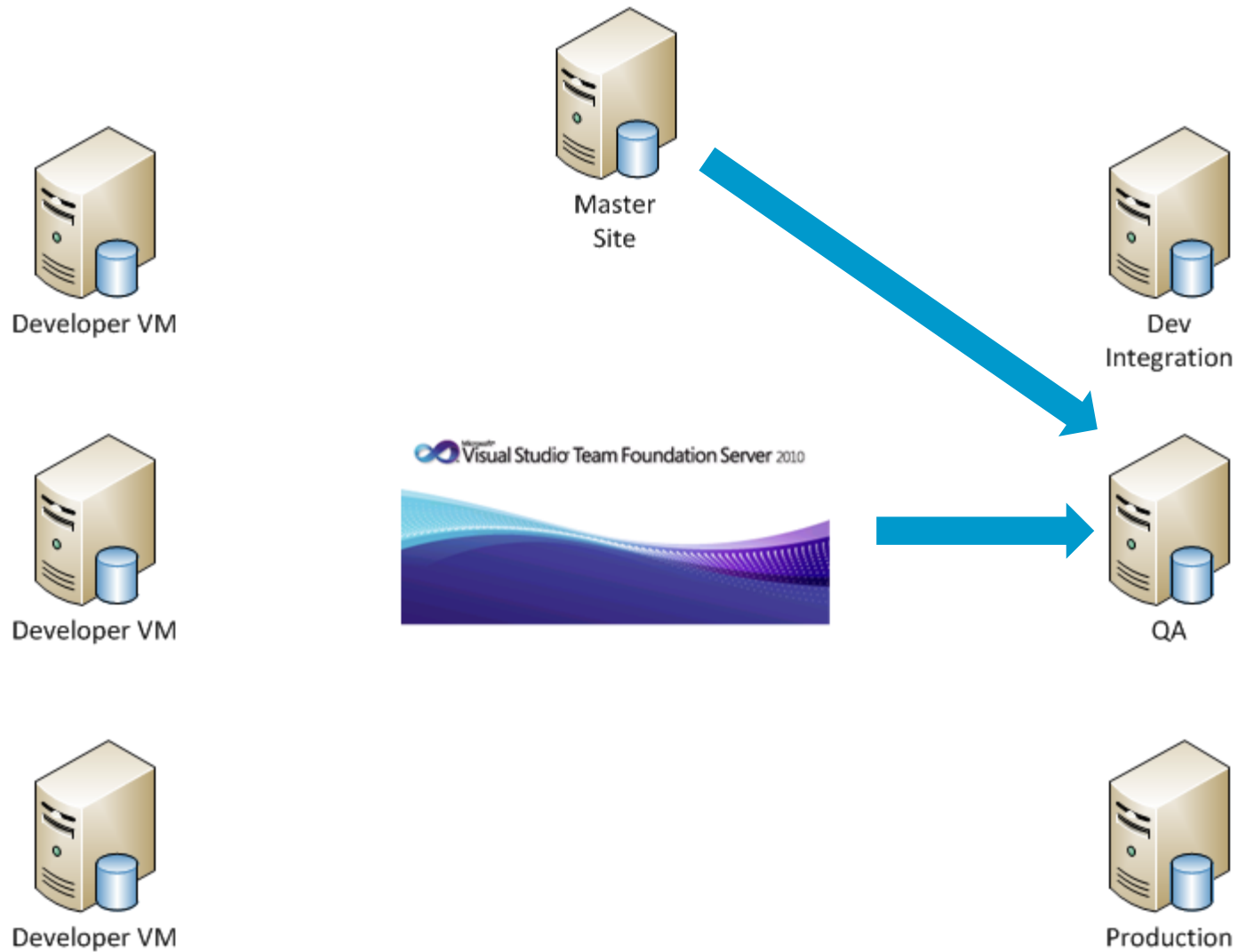
Change Management
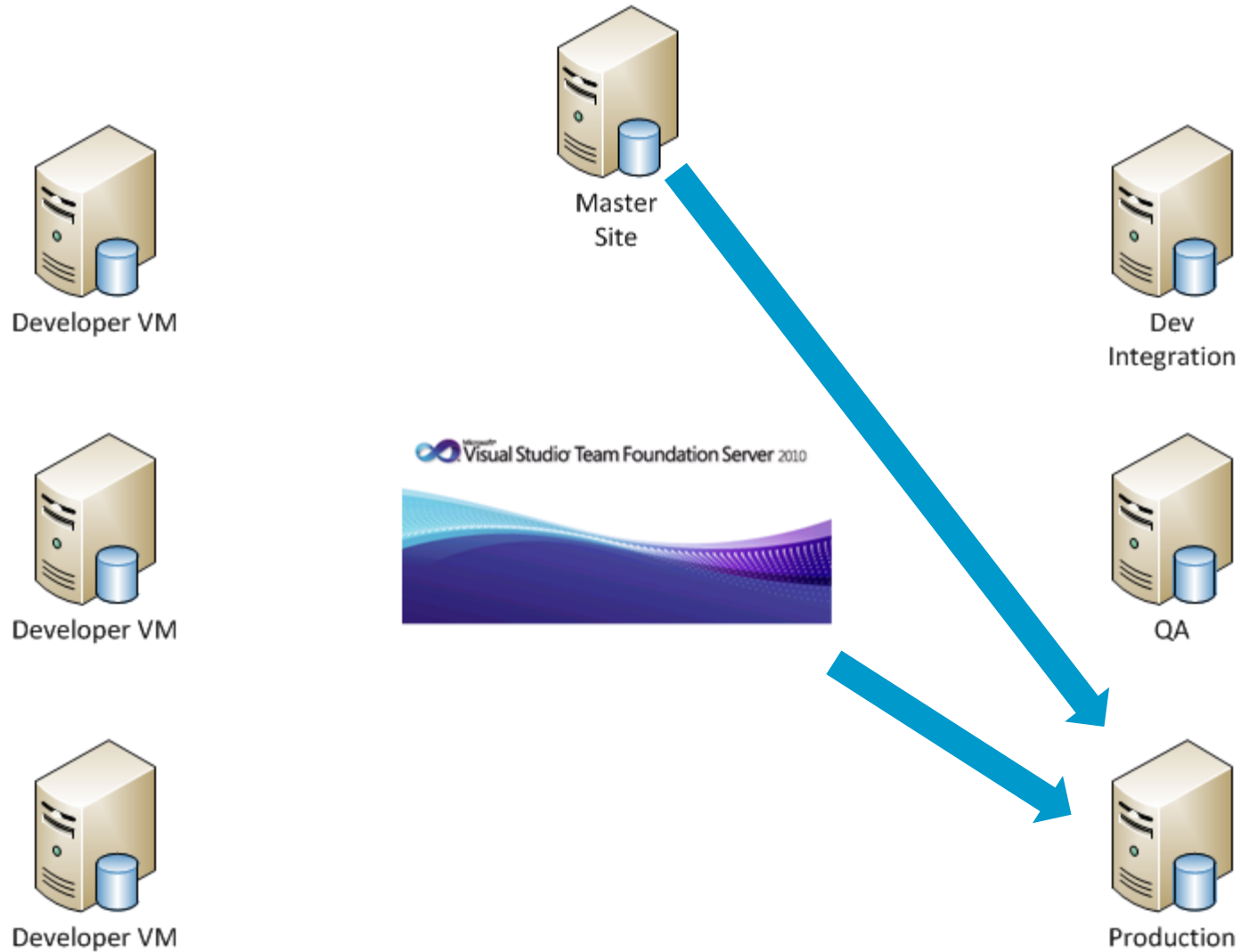
Change Management

Change Management

Web part instances

- Web parts ultimately get placed on pages
- The configuration of the web parts can often get quite complex
- You may want to disable editing of web parts for general content authors
- They end up in a grey zone between code and content

Web part instances – Part 2

- We created our own internal tool to export and import web parts
- Each web part's configuration can be exported into a .webpart file
- We scripted the extraction from the dev integration
- .webpart files are treated like code and checked into TFS for version control
- Script then applies the new .webpart files to the target environment

Search

- Use the [SharePoint Enterprise Search Migration Tool](#) when you upgrade to SharePoint Enterprise Search to migrate search-related data, such as best bets, search scopes, and site collection search settings. You can download the tool [HERE](#).

- Applies to: Microsoft FAST Search Server 2010 for SharePoint | Microsoft SharePoint Server 2010 | Microsoft Office SharePoint Server 2007

- To run the tool, open an elevated command prompt on the server and navigate to the path where the tool is installed.

  - %Program Files%\Microsoft\Search Migration Tool v1.1.1

- Migration Tool Syntax

  - searchmigrationtool.exe -action [-objectKind] [-filter=value] [-conflictBehavior=value] [-fastSearch] filename

- Usage Examples

  - searchmigrationtool -export -scope -siteFilter=http://sharepoint C:\filename.xml

  - searchmigrationtool -export -scope -bestbet C:\filename.xml

  - searchmigrationtool -import -bestbet -conflictBehavior=prompt C:\filename.xml

  - searchmigrationtool -saveSample C:\sample.xml

  - searchmigrationtool -saveSchema C:\schema.xsd

Workflows

- Workflows can be built using SharePoint Designer, Visual Studio, or third-party tools (Nintex or K2 primarily)
- Visual Studio workflow projects are automatically packaged as WSP solution files
  - Simply deploy the WSP file to the other environments
- SharePoint Designer workflows need to be deployed using SharePoint Designer
  - This means you need SPD access to production, which is not always possible
  - You can load the SPD workflow into Visual Studio to package it, but you lose the ability to edit it in SPD
- If you use third-party tools you'll need to define the process for that as well

Business Connectivity Services

- BCS can be built using either SharePoint Designer or Visual Studio
- Visual Studio BCS projects are automatically packaged as WSP solution files
  - Simply deploy the WSP file to the other environments
  - You can also export the BDC metadata model in the BDC service application
- SharePoint Designer doesn't create a WSP, and it doesn't make a complete BDC metadata model
  - You need to export and import using SharePoint Designer
- In both cases you still need to deal with connection and configuration information

Managed Metadata

- [SharePoint 2010 Managed Metadata - Moving/Copying MMS Instances / Term Stores between Environments](#) – By Andrew Connell.

- Steps are as follows:

  - Get the name of the MMS instance DB in the source or master SharePoint farm.

  - From the SQL Server that your source SharePoint farm uses and where the MMS instance DB resides, backup the database.

  - Take the backup file to your target SharePoint Farm's SQL Server and restore the database.

  - Next, make sure you grant DBO rights to the service account that is configured as the identity of the application pool that will be associated with the new MMS instance.

  - Now, on the target SharePoint farm, go through the process of creating an MMS instance except make sure you use the name of the database that you restored.

  - Verify everything looks good by going into the management page for the new MMS instance after it was created and you should see your taxonomies. To use them just make sure that the appropriate service proxies (aka: connections) are setup with the web applications where you want to use them.

Future Directions

- Visual Studio Lab Management
  - Integrates the management of HyperV environments in with Visual Studio and Team Foundation Server
  - Can automate the provisioning of lab environments for automated testing
  - Test failures can snapshot entire environments to make reproduction of errors much easier

Summary

- You're going to have lots of SharePoint environments – figure out how to build them quickly and repeatable
- Code and content often move in opposite directions through dev, QA, and Prod. Script and automate to keep this simple
- Agile is a great way to run projects
- Continuous integration with automated testing is key to its success
- There are lots of parts of SharePoint that you don't code through Visual Studio. Figure out how you move those through your environment

Links

- http://blog.petercarson.ca
- http://autospinstaller.codeplex.com/
- System Centre Virtual Machine Manager 2012 Beta http://www.microsoft.com/download/en/details.aspx?id=609
- Team Foundation Server 2010 http://www.microsoft.com/visualstudio/en-us/products/2010-editions/team-foundation-server/overview
- http://urbanturtle.com/
- http://www.telerik.com/automated-testing-tools/