**ASP.NET MVC in Action, 2ed**

1. Part 1:  High-speed fundamentals
   1. High-speed beginner ramp up (Ben)
      * MVC pattern explanation
      * New project walk-through.
      * GuestBook sample
        + create a controller
        + Add a view (show Visual Studio tooling)
        + basic form post
        + ViewDataDictionary
        + html helpers
      * GuestBookWithModel sample
        + create a model
        + strongly typed view data
        + Add a strongly-typed View (VS Tooling)
        + basic model binding
        + DisplayFor / EditorFor
        + Basic Validation
      * ReSharper 5.0's support for MVC
   2. View fundamentals (Jimmy)
      * View data
      * View model
      * View data binding
      * EditorFor & levels of controls
   3. Role of the controller (Eric)
      * Storyboard flow x
      * Mapping domain model into view model (manually)  x
      * Accepting input x
   4. The presentation model (Matt)
      * Purpose of the presentation model
      * Strongly-typed view model
      * Display model
      * Input model
      * Combined display & input model for complex page
   5. Consuming 3rd party components (Eric)
      * MVC futures html helpers (what futures are left after the MVC 2 release?)
      * MvcContrib grid
      * Consume the login portable area (redo to be the Account controller and membership provider)
   6. IIS Hosting (Ben)
      * Basic IIS 7/7.5 hosting
      * Installation on Server Core 2008 R2
      * IIS 6 hosting
        + Extensionless route
2. Part 2:  Journeyman techniques
   1. Part 2 introduction (2 pages) Learning and applying industry knowledge
      * Legacy code - Feathers
      * Clean Code - Robert Martin
      * DDD - Eric Evans
      * Continuous Integration book
   2. The domain model
      * Lift from chapter 2 (change to Product/Order/OrderLineItem/Customer)
   3. Actions & ActionResults
      * Requirements for a method to be an action
      * Return value requirements (any object)
      * Custom action results for delayed execution
      * Get/Post/ChildAction/Authorize/RequireHttps/NonAction
   4. Advanced view techniques (Jimmy)
      * RenderAction
      * Using master pages with partials
      * Params wrapper for querystrings
      * jquery UI CSS framework
      * Automating views with InputBuilders
      * Spark
      * .Net 4 auto encode html string
   5. Security
      * Restrict Access to actions by Role
      * Protecting against cross-site scripting
      * Getting JSON data via POST instead of GET (and overriding)
   6. AJAX
   7. Controller factories
   8. Model binders & value providers
   9. ModelState validation
   10. File upload processing
   11. Levering existing ASP.NET features
3. Part 3:  Mastering ASP.NET MVC
   1. Routes
      * Route Testing
   2. Deployment techniques
   3. Full system testing
   4. Mapping with AutoMapper
   5. Keeping controllers lightweight
   6. Organizing large apps with Areas
   7. Building and distributing components using portable areas
   8. Decoupled data access using NHibernate
   9. Scaling to complex sites

UrlRoutingModule & MvcHandler

Appendix:  Ruby/Rails

**OLD ToC below:**

High-speed beginner ramp-up

o       Fundamentals

       MvcHandler

       UrlRoutingModule

       Basic routes

       IController and Controller

       Basic display, viewdata, and binding

o       How you should really program

       Dependency management –> solution template

       Display/input models

       Messaging to domain layer

       Convention-based views & InputBuilders

       Domain Model vs. Presentation Model

o       (rework to use order/customer/product)

       The role of the Controller

o       Controls the storyboard flow

       No validation, no detailed work.  Accepts input and passes on.  Get away from framework types as fast as possible.

o       Examples of good controllers

o       Examples of bad controllers

o       Action vs. action result.

o       Edit/Get and Edit/Post (two actions with same name)

o       Filters and applying them at controller factory level.  No layer supertype for controllers

o       TempData.

       Views

o       Basics

       Viewdata

       Fundamental binding & html helpers

       Strongly typed

       Display & editor templates

       Rendering a simple form

o       Advanced views

       Partial views w/ master pages (no ASCX)

       Creating a view convention w/ InputBuilders

       jQuery UI CSS framework

       RenderAction() for common areas w/ HtmlHelper wrapper like : Html.ShoppingCartSummary()

       Params class wrapper for query string args

Strongly Typed View Models and Automapping.

       URL Routing (as is)

o       Routes for areas

       Extensibility points

o       Controller factory

o       Modelbinders

o       IValueProvider

o       Model metadata & validation providers

Action Filters

Action Result

       AJAX & Web Services

o       Ajaxify any form

o       Stream view model as JSON to AJAX call

o       Hijax

o       Expose web services w/ xml result

       Hosting & Deployment

o       IIS 7/7.5 deployment

o       MSDeploy w/ Powershell deployment

o       Mono & OpenSuse deployment

o       IIS6 deployment

o       Azure hosting

       Best practices (adapt)

o       Improve on full-system testing

       Building, distributing, and consuming MVC components

o       Intro to Areas & Portable Areas

o       Building portable areas

o       Packaging and distributing portable areas

o       Consuming

       Storing and retrieving data with NHibernate

       File Upload processing

o       Upload XML & put in queue

o       Upload/process small file with validation.

o       Xml process web service w/ command line client

       Leveraging existing features.

       Scaling to complex sites.