**ASP.NET MVC 2 in Action**

1. **Part 1:  High-speed fundamentals**
   1. High-speed beginner ramp up (Ben) **reviewed 11/29**
      * MVC pattern explanation
      * GuestBook sample (very concise)
        + New project walk-through
        + 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
        + TempData
        + PRG
   2. Presentation Model (Matt) **reviewed 11/29**
      * Purpose of the presentation model - Martin Fowler's article defining presentation model pattern, Evans quote on useful models
      * Strongly-typed view model
      * Display model
      * Input model
      * Combined display & input model for complex page
   3. View (Jimmy) **reviewed 11/29**
      * View data
      * View model
      * View data binding
      * EditorFor & levels of controls
   4. Controller (Eric) **reviewed 11/29**
      * Storyboard flow x   Post Redirect Get for success / Dump Model for failure
      * Mapping domain model into view model (manually)  x
      * Accepting input x -> **add views and use the default template instead of the empty project template**
      * Model State & Temp data
      * Controller Base class - one page -
      * **Side bars -** 
        + **~~Happy Path sidebar from chapter 1.~~**
        + **~~page 19 Icontroller note~~**
        + ~~page 30  note on httpcontext~~
        + ~~chapte 3 page 2~~
        + testing controller drop in.
      * Testing Controllers. pg 13
   5. Consuming 3rd party components (Eric) **reviewed 11/29**
      * MvcContrib grid - remove WithActionSections sample.
      * Consume the login portable area (redo to be the Account controller and membership provider)
   6. IIS Hosting (Ben) **reviewed 12/6**
      * Basic IIS 7/7.5 hosting
      * IIS 6 hosting
        + Extensionless route
   7. Levering existing ASP.NET features *(Jeffrey)*
2. **Part 2:  Journeyman techniques (Jeffrey writes Part 2 intro)**
   * + Learning and applying industry knowledge
     + Legacy code - Feathers
     + Clean Code - Robert Martin
     + DDD - Eric Evans
     + Continuous Integration book
     + Uncle Bob Agile Principles, Patterns & Practices, etc in C#
   1. The domain model (Matt) **reviewed 12/6**
      * Lift from chapter 2 (change to Product/Order/OrderLineItem/Customer)
   2. Extending the Controller (**Eric**)
      * IController and requirements for Controller (list)
      * Requirements for Filters  - NOTE FROM chapter 7 PAGE 17
      * Requirements for a method to be an action (list)
      * Return value requirements (any object)
      * Custom action results for delayed execution (code sample) - CSV ,    Logout Sample -show test for this bad boy ,  - **Ready For Review**
      * Selectors: Get/Post/**ChildAction  (explain the difference between Action and ChildAction)**/Authorize/**RequireHttps**/NonAction (code sample) - testing of the https

* + - Updated ActionResult Lists (resharper ctrl-alt b)
    - chapter 3 notes - 21 &22 notes
    - chapter 6 - page 19 , note 27
    - chapter 12 - pg6 why seperate filters.
  1. Advanced view techniques (Jimmy)
     + RenderAction
     + Params wrapper for querystrings
     + Automating views with InputBuilders
     + Spark
     + .Net 4 auto encode html string
  2. Security (Matt)
     1. Restrict Access to actions by Role
     2. XSS & XSRF attacks
        1. how they work
        2. how to protect against them type here -Jeffrey Palermo 1/3/10 9:06 PM
     3. Getting JSON data via POST instead of GET (and overriding) - Needs review for RC
  3. AJAX (take 1stEd, change example from CodeCampServer, test & revise - should not be a rewrite) (Ben)
  4. Controller factories (Ben)
     + Wire up StructureMap
     + Wire up Ninject
     + Wire up Windsor
  5. Model binders & value providers (Jimmy)
     + Leverage domain model type binding in actions
     + Add value provider so that framework also looks in Application, Session, and Server Variables
  6. ModelState validation (Jimmy)
     + DataAnnotations
     + ModelMetaData (custom validations besides attributes)  ... like Castle Validators?
     + ASP.NET AJAX client-side validations
     + page chapter 4 - page 27,
     + chapter 13 - page 11 attributes on Domain Model

1. **Part 3:  Mastering ASP.NET MVC**
   1. Routes (body of work should be aggregating code samples and light wordsmithing of existing chapter) (Ben)
      * Lightly revise existing routes chapter for accuracy
      * Add MvcContrib route testing (harvest section 12.3.1 from Chapter 12)
   2. Deployment techniques (Eric)
      * Harvest section 10.5 =  from Chapter 10 (XCopy using NAnt & Tarantino DB migrations)
      * MSDeploy? ( nartate CCS Deployment)
   3. Mapping with AutoMapper *(brand new)* (Matt)
      * Show life without automapper
      * Map from domain to display model
      * Map using a custom converter and "after map"
   4. Keeping controllers lightweight (Matt)
      * Apply Filters in Controller Factory vs. attribute on supertype
      * AutoMapToModel action result?
      * leveraging an application bus for a simple hub & spoke architecture
      * text on skinny controllers, duplication makes controllers fat
   5. Full system testing (Jimmy)
      * WatiN testing from current recipes chapter
      * Subcutaneous testing
   6. Organizing large apps with Areas *(brand new) (Jimmy)*
      * Basic area example and when to use an area
      * Secure an area with <location/> node in web.config
      * Generate links between areas
   7. Building and distributing components using portable areas (Eric)
      * Componentize AccountController & Membership Provider from default template
      * Distribute portable area
      * Install in ASP.NET MVC project   - **Show sample of receiving a Bus Message and writeout user login**
      * Install in WebForms project  - side bar from chapter 1 "Integrating with WebForms"
      * Distribute RSS Widget (Html.Rss(string url)) with portable area - where Html.Rss is a wrapper of an Html.Action( ) call.
   8. Decoupled data access using NHibernate *(Jeffrey)*

Part 4 : Cross-cuttong advanced topics

* 1. Route Debugger (Ben)
  2. Customizing Visual Studio for ASP.NET MVC(Jeffrey)
  3. Testing Practices (Jeffrey)
  4. Autocomplete recipe (Ben)