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

* High-speed beginner ramp-up
  + Fundamentals
    - MvcHandler
    - UrlRoutingModule
    - Basic routes
    - IController and Controller
    - Basic display, viewdata, and binding
  + 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
  + (rework to use order/customer/product)
* The role of the Controller
  + Controls the storyboard flow
    - No validation, no detailed work. Accepts input and passes on. Get away from framework types as fast as possible.
  + Examples of good controllers
  + Examples of bad controllers
  + Action vs. action result.
  + Edit/Get and Edit/Post (two actions with same name)
  + Filters and applying them at controller factory level. No layer supertype for controllers
  + TempData.
* Views
  + Basics
    - Viewdata
    - Fundamental binding & html helpers
    - Strongly typed
    - Display & editor templates
    - Rendering a simple form
  + 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
* URL Routing (as is)
  + Routes for areas
* Extensibility points
  + Controller factory
  + Modelbinders
  + IValueProvider
  + Model metadata & validation providers
* AJAX & Web Services
  + Ajaxify any form
  + Stream view model as JSON to AJAX call
  + Hijax
  + Expose web services w/ xml result
* Hosting & Deployment
  + IIS 7/7.5 deployment
  + MSDeploy w/ Powershell deployment
  + Mono & OpenSuse deployment
  + IIS6 deployment
  + Azure hosting
* Best practices (adapt)
  + Improve on full-system testing
* Building, distributing, and consuming MVC components
  + Intro to Areas & Portable Areas
  + Building portable areas
  + Packaging and distributing portable areas
  + Consuming
* Storing and retrieving data with NHibernate
* File Upload processing
  + Upload XML & put in queue
  + Upload/process small file with validation.
  + Xml process web service w/ command line client
* Leveraging existing features.
* Scaling to complex sites.