

## Module Review and Takeaways

In this module, you have seen how teams of developers, software architects, and business analysts collaborate to design an MVC web application that meets the needs of users. You can gather functional and technical requirements by talking to stakeholders and creating use cases, usage scenarios, and user stories. The model, view, controller, and other aspects of the design depend on these requirements. You have also seen how these design activities are completed in projects that use the agile methodology or extreme programming.



**Best Practice:** In Agile Development and Extreme Programming projects, developers discuss with users and stakeholders throughout development to ensure that their code will meet changing requirements. Even if you are not formally using these methodologies, it is good practice to regularly communicate with users.



**Best Practice:** When you design an ASP.NET MVC web application, start with the model, and then plan controllers, actions, and views. The controllers, actions, and views that you create each depend on the model.

### Common Issues and Troubleshooting Tips

Common Issue	Troubleshooting Tip
When you create a very detailed project plan, much of your work is wasted when requirements change late in the project.	

### Review Question(s)

**Question:** You want to support both English and Spanish in your web application. You have both Spanish-speaking and English-speaking developers and want to ensure that views remain readable as easily as possible. Should you use multiple view files or multiple resource files to globalize your site?

### Real-world Issues and Scenarios

You should bear in mind that when you select a project methodology, few projects follow a neat plan in real situations. Of the methodologies described in this module, agile development and extreme programming are the most flexible and respond when plans change in the middle of development. However, even with these methodologies, changing circumstances result in wasted development time and your project budget should include a contingency to cope with such changes.

Furthermore, when working with agile development and extreme programming projects, project managers must take care to avoid project creep or scope-creep. This occurs when people add new requirements when development takes place. Project creep results in projects that are over-budget and late.

### Tools

Microsoft Office Visio: You can use Visio to create all types of UML software design diagrams, including Domain Model diagrams and LDMs. You can also use it to create wireframes.

Visual Studio 2015: You can create class diagrams such as LDMs in Visual Studio 2015.