Spring mvc (Framework)

* Entkopplung der Applikationskomponenten
* Dependency Injection: Den Objekten werden die benötigten Ressourcen und Objekten zugewiesen. Man muss nicht selbst suchen.
* Aspektorientierte Programmierung: Dadurch kann man vor allem technische Aspekte wie Transaktionen oder Sicherheit isolieren und den eigentlichen Code davon frei halten. Spring bringt dabei neben Spring AOP eine gute Unterstützung für AspectJ und dessen *Pointcut Expression Language* mit.
* Vorlagen dienen dazu, die Arbeit mit einigen Programmierschnittstellen (APIs) zu vereinfachen, indem Ressourcen automatisch aufgeräumt sowie Fehlersituationen einheitlich behandelt werden.
* Dadurch wird ein POJO-basiertes Programmiermodell möglich, bei dem die POJOs zusätzlich in verschiedenen Umgebungen (auf einem Server oder in einer Client-Anwendung) lauffähig sind.
* (Erleichtert Transaktionsmanagement mit der Datenbank mittels Hibernate)

ASP.NET MVC

Vorteile

* Visual Studio komplette IDE, die keine Konfiguration erfordert  
  Sehr umfangreiche, aber trotzdem gut bedienbare Entwicklungsumgebung
* Erfahrung in C#, ASP.NET  
  Torsten hat im ersten Semester viel C# gemacht.
* Wenig Konfigurationsaufwand  
  Man kommt schnell und einfach zu einer lauffähigen Applikation, die sehr wenig Konfiguration erfordert.
* Relativ einfach zu lernen und zu benutzen
* Gute Dokumentation <http://asp.net>  
  Viele Tutorials und HowTo’s

Nachteile

* Hosting ist „teurer“ als Spring MVC
* Angewiesen auf Microsoft Frameworks  
  Gibt allerdings auch z.B. Spring oder Hibernate für .NET