1)REST, web servisleri geliştirmek için kullanılan mimari ilkeler bütünüyken; SOAP, başka platformlarda ve başka teknolojilerde geliştirilen uygulamaların birbiri ile anlaşabilmesini sağlayan bir protokoldür. Kullanımı esnek ve hafif olması REST mimarisinin öne çıkan özelliğiyken, güvenlik ve uyum konularında SOAP daha avantajlıdır. REST, birden fazla veri formatıyla çalışabilirken, SOAP yalnızca XML kullanımına izin verir.

2)Acceptance Testing, bir yazılımın istenen şartları sağlayıp sağlamamasıyla ilgilenir. Ürünün son kullanıcıya teslim edilecek düzeyde uygun olması için gerekli bütün senaryolar üzerinde durur. Functional Testing ise yazılımda var olan özelliklerin kontrol edilmesidir. Ortaya konulan işin tasarlanıldığı gibi çalışıp çalışmamasına bakılır.

3)Mocking, unit test yapılırken kullanılan bir işlemdir. Test edilecek yapının başka nesnelere veya metotlara bağlı olması durumunda, bu bağımlılıkların yerlerine taklitleri yapılır. Bu sayede test edilecek yapı izole edilerek, sadece ona odaklı işlemler yapılır.

4)Ne kadar test yazılması gerektiği, yazılımcının tecrübesine ve projenin kendisine göre değişiklik gösterir. Test yazmaya yeni başlamış birisinin kendini bu konuda geliştirmesi gerektiğinden, bir yerden başlayarak buna alışması ve niceliği dert etmemesi gerekir. Tecrübeli bir yazılımcı ise, proje için gerekli faktörleri analiz etmeli ve bunları test etmelidir.

5)HTTP PUT, genel olarak var olan bir kaynağı güncellemek için kullanılır, eğer kaynak yoksa kendi yaratır. Ne kadar istek atılırsa atılsın, dönen yapı aynıdır. Kullanıcı, PUT isteğini kullanırken kaynağı kendi belirler. HTTP POST ise var olan kaynağın altına yeni alt kaynaklar eklemek için kullanılır. Atılan istek kadar kaynak yaratılacağından, her seferinde farklı bir yapı döner. Kullanıcı, hali hazırda var olan kaynağa istek atar.

6)Safe HTTP Metotları, kaynaklar üzerinde değişiklik yapmayan, salt okuma işlemi yapan metotlardır. GET, HEAD ve OPTION örnek olarak gösterilebilir. Zıt anlamlısı olan Unsafe HTTP Metotları, kaynaklar üzerinde değişiklik yaparlar. POST, PUT ve DELETE ise burada örnek olarak verilebilir.

7)HTTP Basic Authentication basit bir güvenlik önlemidir. İstemci, isteği attığı sunucuya erişmek için Authorization başlığı altında kullanıcı adı ve şifreyi base-64 şifrelemesi yaparak gönderir. Sunucu, gönderilen bilgileri kontrol ederek; doğruysa geri dönüş yapar, yanlışsa 401 (Unauthorized) hatasını döner.

8)RestTemplate, RESTful Web servislerini kullanmak için oluşturulan senkron istemcidir. HTTP istekleri işlemek için kullanılıp, atılan istekleri ve dönütleri nesnelere dönüştürür. Kullandığı HTTP kütüphaneleri sayesinde, detaylı bilgi gerektiren aktarım protokolleri arka planda yapılarak kullanıcıdan soyutlanır.

9)@Controller, @Component yapısının özelleşmiş halidir. İşaretlenen sınıfın web isteklerini işleyen bir görevi olduğunu belirtir. Dönüt olarak View şeklinde dönüş yapar. @RestController ise @Controller yapısının özel bir halidir. @Controller ve @ResponseBody yapılarını içerir. İçeriğindeki @ResponseBody yapısı sayesinde XML ya da JSON kullanan objeler döndürür.

10)DNS Spoofing saldırısı, kullanıcıyı ulaşmak istediği yerin kötü niyetli bir taklidine yönlendirerek yapılan, kişisel verileri veya önemli bilgileri çalmayı hedefleyen bir saldırı türüdür.Korunmak için ekstradan bir katman olan DNSSEC(DNS Güvenlik Uzantısı) eklenerek süreç daha güvenli hale getirilebilir.

11)Content Negotiation, bir kaynağın aynı URL adresinde farklı biçimlerde sunulabilmesine olanak sağlayan bir konsepttir. Atılan HTTP isteğinin yapısı incelenerek, genelde kullanıcının tercihleri üzerinden var olan en uygun kaynak dönülür. Veri biçimi, kodlama türü, dil vb gibi alan başlıkları sayesinde değişiklikler yapılır.

12)Statelessness, kullanıcının paylaştığı verilerin sunucu tarafında tutulmaması üzerine kurulan bir kısıtlamadır. Bu yüzden istemci tarafından sunucuya atılacak olan her bir isteğin gerekli bilgilere sahip olması gerekir. HTTP istekleri birbirlerinden izole bir şekilde atıldıklarından, kullanıcılar gerekli bilgileri saklama ve iletmeyle sorumludur.

13)CSRF saldırısı, kimliği doğrulanmış olan kullanıcıların bilgileriyle, kendi istekleri dışında kötü niyetli kişilerce işlemler yapılmasıdır. Açıklardan faydalanarak yapılan bu saldırı, genellikle sosyal mühendislik yöntemleri kullanılarak gerçekleştirilir. Bu saldırının önüne geçmek için ise CSRF Token yöntemi kullanılır. Token rastgele oluşturulan büyük bir değer olup her bir oturum için yeniden oluşturulur.

14)HTTP Request 5 ana yapıdan oluşur:

- Verb: Atılacak isteğin türünü belirtir. GET, POST, DELETE vb.
- URI: Kaynağın hangi lokasyonda olduğunu belirtir.
- Version: HTTP isteğinin versiyonunu belirtir.
- Request Header: İstekle alakalı üst bilgilerin bulunduğu alandır.
- Request Body: İstek içeriğinin bulunduğu yer.

HTTP Response ise 4 ana yapıdan oluşur:

- Status/Response Code: Atılan isteğe göre sunucudan dönen koddur. 404 Not Found vb.
- Version: HTTP isteğinin versiyonunu belirtir.
- Response Header: Dönütle alakalı üst bilgilerin bulunduğu alandır.
- Response Body: Dönüt içeriği bu alanda bulunur.