Sean Hodges

1. Integration – Its hard to keep track of code from different sources
   1. Docker can help with integration because it keeps all the necessary parts of an application, such as libraires and other dependencies all in one container. When developers integrate code from different sources it will be kept all in one place, so it is super easy to keep track of
2. Testing – Keeping consistent tests is a challenge
   1. Docker provides repeatable test environments so whenever an update is made to the code the previous test cases will still be right where they were left, making it super easy to test older code again
3. Deployment – Releasing on time, dropping new releases, “it works on my machine”
   1. When a new product/release is deployed, Docker containers allow users to commit changes to Docker images and version control them, which means that if the new release breaks the code, it can be rolled back to a previous working version
   2. Docker reduces deployment to seconds using containers
   3. Docker keeps consistent environments so the same container can be used from development to production
4. Customer Support – Not always reliable
   1. Docker has a very reliable customer support team so if there is ever an issue or problem that needs to be solved, there is someone from Docker who can help
   2. With the Docker container that Spotify used, Spotify was able to keep all of their services up and running, which improved the user experience for the customers
5. Internal and External Hackers – Security for the users information
   1. The applications running on containers are isolated from each other, which gives the user complete control over the traffic flow and management
   2. No container can look into another container to see what it is doing

Sources:

DZone article on the assignment