DevOps SDLC

We will now move into the DevOps culture, where we will begin at the same point that the traditional silo commenced.

Breaking down the walls

Here the developers are still writing the code, but now the developers perform their code in stages. Breaking the code into these stages run automated steps that build, integrate, test and deploy that section of the code. Because of this automation, the QA's can review the work immediately, testing the code and handing it off to production for deployment. As each change is small, it is a continuous process that is fast and harmonious.

The process

This environment is not perfect. There will still be human error in making the product, but when the code is sent for production and something is broken, continuous monitoring will detect the problem almost immediately and notify the team that there is an issue. This rapid deployment of changes ensures that the latest features and bug issues are resolved for the stakeholders with the minimum time delay between creating the code and its deployment.

Get faster, deploy small

Because of the DevOps setup, the team will be able to roll back and deploy the previous version of the code exceptionally quick, with minimal downtime and little effect on the stakeholder or customer. If the traditional SDLC was used in this case, any rollback would be catastrophic. The large amounts of code between deployments could lead to significant and lengthy downtimes or delay in rolling out the new features or improvements.

Summary

As the DevOps team roll back to their previous version, and the changes are not significant, it can allow them to fix and then redeploy the code, ensuring speed of delivery and that the team can still maintain the stability of the product. The rapid turnaround to any potential problems in the lifecycle is dealt with quickly and often without the customers noticing there was an issue.