Caleb Stark, 2/23/2025, Module 8 Assignment 2 DevOps, Change Approval Process

Change management is a necessary process in the development of software but it does not come without some pitfalls and areas of concern. Overall, a good change management process will allow for seamless additions to be made to the code base and implemented into the live environment with minimal impact on those using the code. When there is impact a good change approval process, we should be able to help with finding the origination of the issue and be able to make a fix quickly as well. The issue is that when it does not go well then not only can you see impacts on the quality of the approved code but also bottlenecks and delays that can cause misses in deadlines, or lead to much larger final production pushes that can be hard to manage when errors and bugs appear.

Before looking at these pitfalls lets first breakdown the goals of the change approval process it a few different ideas of what it looks like. Change approval process exists as a point to decrease the risk of implementation from changes made to the code base. This process does remove bugs from occurring but instead lessens the risk of a bug appearing and if one does come that its overall impact is minimal. To implement this process, you may look to add review process with a peer or board of reviewers, implement testing environments, regularly integrate changes into the production environment, and ensure that you have effective monitoring to alert of errors if they are to occur. When it comes to these various options for change approval it may be hard to know where you should start for implementation, the key here is determining what your environment requires. Automated testing is a key component of change approval for and program of reasonable scale, this testing can be used to check code prior to implementation, automate approved code to be integrated into the live environment, and finally be able to detect issues and try to limit overall impact.

When it comes to the downsides of change approval process there are some big areas to consider watching to ensure that the process does not become to bloated. One of these areas is determining different levels of severity for any changes to determine how each change is handled. If you treat every change the same and require the most rigorous checks even on code that had little to no change then you will cause backlogs to build especially if this rigorous check requires a persons input. The next big issue is putting too much trust or responsibility onto a single point of failure such as change approval boards. These boards may take a significant amount of time to complete a check on your code especially if they have not been involved in the planning or creation of the stated code. This board or similar styles of change approval could actually result in the releases of code to become more volatile and cause larger issues. This can come up because as the approval backlog fills up we can miss key information that should be implemented right away which then would hopefully save us all in the long term.

Overall I think that a change approval process should always exist for software that will require long lasting support. By having the process be the norm people can expect the prices to stay similar and not much else change with our lives.

# References

Dora. (n.d.). *Streamlining change approval*. Retrieved from DORA: https://dora.dev/capabilities/streamlining-change-approval/

*Guide to the IT Change Requests Approval Process*. (n.d.). Retrieved from MyndBend: https://www.myndbend.com/guide-to-the-it-change-requests-approval-process/

LaunchDarkly. (n.d.). *The downsides of heavy change management*. Retrieved from LaunchDarkly: https://launchdarkly.com/guides/reconciling-change-management-and-continuous-delivery/the-downsides-of-heavy-change-management/