Business Intelligence Implementation Plan

John Wensink

MIS480 Senior Capstone

Colorado State University-Global Campus

Chris den Heijer

January 10, 2020

Business Intelligence Implementation Plan

As our Systems Development Life Cycle (SDLC) enters into the implementation phase, I wanted to take a moment to outline the plan to effectively manage the significant change that our organization is going through. In this short briefing, I will outline the change plan in terms of testing and resources, technical and non-technical documentation for our userbase as well as the development team, risk mitigation techniques, and resources, as well as timetables in case problems come up with deploying our Business Intelligence (BI) solution to our various companies across the enterprise. Any time there is a significant change to the way people are accustomed to doing business there is likely to be pushback from the user base and we must be prepared to provide continuous education and support so that adoption is widespread, fast, and lasting.

**Testing: Plans and Resources**

We will begin testing in a phased approach with smaller, more technically proficient focus groups who can work closely with the project leaders in a dev-ops type of environment. Initial testing scripts will be worked with a pass-fail result, and any failing tests will need to be reconciled with a program developer to ensure that the bug that caused the error can not be repeated. Once all the initial testing scripts are consistently passing (Tutej & Dubey, 2012) the testing will enter the second phase which will be wider in scope and duration. Here, a larger group of employees from our innovations lab will work closely with their manager to implement and document successes and failures in a sandbox environment where the system can be introduced to live production work and the inevitable bugs can be worked out. This team of employees has been hand-selected for their attention to detail as it pertains to innovative process implementation. Our organization is fortunate to have the innovation lab as a resource as they are experienced testers who have given the green light to every recent change innovation in the past ten years.

**Documentation: Technical and Non-Technical**

This project’s documentation can not live solely in the minds of a small team of core developers. Every step along the way down to the production floor will have its own set of change documentation all the way down to the #comments written into the code itself. Having a timestamped log of version changes that correspond to daily backups which we can revert to in case of failure will be an invaluable asset should the system fail as a result of recursive code improvements. Further downstream, the project manager’s log should be all-encompassing including the developer’s logs, DevOps changes, innovation team changes, as well as support tickets documenting troubles or errors (Uk, 2017). Non-technical documentation and training will need to exist in user-friendly formats both digital and print that will be easily accessible and simple for a layman to understand. It is easy to forget how reliant teams in the change management department come to rely on technical jargon and assume above-average technical skills when it comes to business information systems. If possible, a step-by-step walkthrough of the process, as well as a troubleshooting guide, should be within easy access of any employee who requests it. Graphics are always helpful and color printouts will be required over black and white.

**Risk Mitigation and Implementation Timeline**

The redundant backups and thorough change logging are just one of a set of risk management best practices that our organization would be wise to implement. As this is an untested system we should start by loading training data into the system and contract a full network penetration test to ensure our member’s sensitive data will not be at risk. Perhaps our in-house team could accomplish this task, but for audit and compliance purposes it should be completed by a disinterested third party. We expect the development to be finalized by the end of this week and next week we will start to slowly implement the new BI system. If all goes according to plan we expect to have major bugs and errors ironed out within 90 days and adoption by the innovation lab should be producing statistically significant data within about 30 days, with process improvements starting to be shown around the two-month mark. In Q4 of 2021, our organization will push for enterprise-wide adoption, with any departments having issues with the implementation of putting their requests in before the end of the year. Should assistance be requested, our DevOps team will be on standby to provide assistance to any department that requests it.

Although the implementation of such a dynamic change is intimidating, there is no other team that I would rather have doing the planning and work. Attention to detail has been a strength of our department in the past and will continue to be called upon as we pursue daily progress over immediate perfection. By this time next year implementation should be in full swing and the fruits of our labor can be realized.

References

Tutej, M., & Dubey, G. (2012). Testing and Quality Assurance in Software Development

Life Cycle Models. Retrieved December 14, 2020, from <https://www.ukessays.com/essays/information-technology/testing-and-quality-assurance-in-software-development-life-cycle-models.php>

Uk, I. (2017, January). Importance of Testing in Software Development Life Cycle.

Retrieved December 14, 2020, from <https://www.ijser.org/researchpaper/Importance-of-Testing-in-Software-Development-Life-Cycle.pdf>