

Deployment Plan – MakerSpace 2023

1. Goals
 - a. All features defined by Dr. Gourd will be implemented
 - b. Website will be tested, and different scenarios will be played out to simulate real users interacting with the project.
2. Roles
 - a. Bug fixing - Bugs will be dealt with based on availability and who finds the bug; no specific person is assigned to a part of the code
 - b. Testing - Every member will simulate a student user, student-worker and admin and try to 'break' the code to maximize realistic scenarios we are unaware of.
 - c. Feature control - Also distributed among group members
3. User-development Communications
 - a. A google form link is available where users will be able to report any bugs/errors they encounter while using the app.
 - b. There is also an emergency button that notifies the development team on a discord channel everytime it is pressed.
4. Task Dependency
 - a. We will be implementing a Finish-to-Start task dependency
5. Bug Tracking
 - a. A google form will be used to track bug submissions and will be dealt with based on the user's perceived importance level if it coincides with that of the development team's.
6. Training
 - a. There will be no necessary training to use the application and required trainings for makerspace will be provided by admins
 - b. Admins will have to join the emergency button discord in order to be notified about emergency button activations
7. Post-deployment Changes
 - a. Gourd :) (Thanks)
8. Business Continuity Development Plan
 - a. Version control will be vital to future development, if any problems occur during updates or a major bug is discovered, all users will be reverted back to an old version of the website (Thanks Microsoft)

[REFERENCE AND CAN BE USED IN POINT 8]

1. Minimal disruption to business practices:
 - Develop a clear plan and schedule for deployment, communicate it to all stakeholders, and stick to it as closely as possible.
 - Conduct thorough testing and quality assurance before deployment to identify and resolve any issues or bugs.
 - Use a phased approach to deployment, starting with a small group of users or a limited functionality, and gradually expanding to the entire system.
 - Provide training and support to users to ensure they are comfortable and prepared to use the new system.
 - Monitor the system closely after deployment to quickly identify and resolve any issues that arise.
2. Recovering lost data:
 - Regularly back up all data to a secure location.
 - Implement a disaster recovery plan that outlines how data will be recovered in the event of a disaster or system failure.
 - Use data recovery software or services if necessary to retrieve lost data.
3. Halting deployment and rolling back:
 - Establish clear criteria for halting deployment, such as a major system failure or significant data loss.
 - Develop a rollback plan that outlines the steps necessary to return to the previous system and data state.
 - Test the rollback plan in advance to ensure it is effective and efficient.
 - Communicate the decision to halt deployment and roll back to all stakeholders as quickly and clearly as possible.
4. Point of no return:
 - The point of no return is typically when data has been irreversibly modified or lost, or when a significant portion of the system has been deployed and cannot be easily rolled back.
 - To minimize disruption in this scenario, it is important to have a clear communication plan in place to quickly notify all stakeholders of the issue and any necessary actions they should take.
 - It is also important to have a plan for mitigating the impact of the issue, such as implementing a temporary workaround or providing additional support to affected users.