**CS692, R2 – Jenkins Progress Report**

**Team: 4**

**Jenkins Admin: Bisheng Zeng**

**Iteration 3: Planned and Completed Tasks**

|  |  |
| --- | --- |
| Planned Tasks | Completed Tasks |
| 1. Implement Jenkins Blue Ocean for enhanced visualization of pipelines. 2. Integrate static code analysis tools into Jenkins pipeline for continuous code quality monitoring. 3. Set up and configure deployment to production environment. 4. Conduct load testing on the deployed applications. 5. Establish automated backup and recovery procedures for Jenkins configuration. 6. Implement additional role-based access controls for more granular permissions. 7. Conduct training sessions for the team on new Jenkins features and best practices. | 1. Implement Jenkins Blue Ocean for enhanced visualization of pipelines. 2. Integrate static code analysis tools into Jenkins pipeline for continuous code quality monitoring. 3. Set up and configure deployment to production environment. 4. Conduct load testing on the deployed applications. 5. Establish automated backup and recovery procedures for Jenkins configuration. 6. Implement additional role-based access controls for more granular permissions. 7. Conduct training sessions for the team on new Jenkins features and best practices. |

**Iteration 4: Planned and Completed Tasks**

|  |  |
| --- | --- |
| Planned Tasks | Completed Tasks |
| 1. Enhance existing CI/CD pipelines to include additional automated tests. 2. Integrate additional code quality and security analysis tools into the Jenkins pipeline. 3. Explore and implement Docker containerization for consistent builds. 4. Develop a rollback mechanism for quick recovery in case of failed deployments. 5. Automate deployment to additional environments (e.g., staging or production) based on successful QA tests. 6. Implement performance monitoring tools for tracking the efficiency of builds and deployments. 7. Conduct a Jenkins system audit to identify and plan for any necessary upgrades or updates. 8. Document all changes and update the Jenkins user manual accordingly. | 1. Enhance existing CI/CD pipelines to include additional automated tests. 2. Integrate additional code quality and security analysis tools into the Jenkins pipeline. 3. Explore and implement Docker containerization for consistent builds. 4. Develop a rollback mechanism for quick recovery in case of failed deployments. 5. Automate deployment to additional environments (e.g., staging or production) based on successful QA tests. 6. Implement performance monitoring tools for tracking the efficiency of builds and deployments. 7. Conduct a Jenkins system audit to identify and plan for any necessary upgrades or updates. 8. Document all changes and update the Jenkins user manual accordingly. |