**Planning document**

**Scalable Continuous integration and Continuous deployment using devops**

1. **Objective:** Establish “Scalable Continuous integration and Continuous deployment by using DevOps” at different phases. This document provides high level guideline to achieving the intended solution.
2. **Description:**

This document describes the goal to achieve in 3 phases. Which also includes:

1. Roles of responsibilities of each team member.
2. Project management tools to be used and its roles.
3. Timeline to achieve each phase.
4. End product final assessment.
5. Identifying improvements for future enhancement.
6. **Phase I:**

Goal of this phase is to:

1. Understand various tools which will be part of the initial product.
2. Setup of Git server, Jenkins server (using docker),
3. Understand the roles of the management tools.
4. Basic developer environment setup.
5. Basic pipeline setup. Which includes:

Git, MS-scripts, Jenkins, VS codebase setup, TestRunner (with Nunit tests) and DB setup.

1. Jenkins scripts are used as “Infrastructure as code” to trigger the build, run tests on different machines and push the passed builds to shared location.
2. **Phase II:**

Goal of this phase is to:

1. Enhance the solution to have Dockers, Containers and Cloud.
2. Artifactory or Harbor tool to store the end product with automatic versioning.
3. Enhance pipeline to support both Git and Svn.
4. Enhance TestRunner to support MSTests.
5. Various dashboard tools to show the pipeline health and status. Which are
6. Pipeline health: html page shows the health of the pipeline. If expected results are not available, if pipeline is broken, expected machines are not up then that immediately reflects here.
7. Test results activity: Html tool pulls information from the DB and show the live status of the test suites running in the various machines. This is required for developers to have quick feedback whether expected tests are passed or not.
8. Test results dashboards: Shows the code coverage for each archive, test results passing/failing trend per week (or for demanded duration).
9. **Phase III:**

When this phase is reached, we already have a CI-CD running. Here most of the work will be related to cloud. Making the solution scalable. When end user wants to run X tests and want to have results in Y hours, then solution shall scale itself at various phases by including adjusting CPU usage at build servers, increasing number of VM’s in the cloud and provide the results in stipulated time.

(Manikanta: I don’t have concrete technical details like how to achieve this, but I know for sure that this can be achieved).

1. **Timeline: TBD**
2. **Roles and Responsibilities of individuals: TBD**
3. **Way of working (WOW):**

At the beginning two people will be working on one task. Going forward, at end of the phase I, individuals will work on different things. Everything shall be synced up among all of you, load shall be shared equally. One who is working on that area will be the expert in that but rest all aware how it works. In this way, one can reach the goal faster.

There is no hard and fast rule, considering magnitude of the work which comes up amid of the development, WOW can be altered. The one proposed is ideal WOW in the Agile process.

**How we can be a great team?**

1. Understand what to do before taking up work.
2. Always do a spike and have a workable code, before taking up the product development.
3. Challenge others with your knowledge.
4. Open for constructive criticism.
5. Respect each other’s opinion and work.
6. Appreciation is optional. (That shouldn’t be the motivation, in my view).
7. **Documents to be delivered at end of the solution development:**

It is required to understand the documents required to deliver at end of the product development (which includes time line to deliver them), to college and concerned authorities. The reason being:

1. Documentation consumes lots of time. Being aware of what docs required upfront helps us to have better planning.
2. No matter how much good work you do, if you don’t know how to present yourself and your product, it’s gonna eat dust.
3. Depending on what we are presenting, we can also alter dashboard tools we develop, which benefits us to present our solution in better way.
4. I would like to stress below point in your documentation.

“Take away of this project.” :

1. You are going to learn software development tools and also project management tools.
2. WOW is of industrial standards.