**Task 1**

**The difference among (ceo , coo , cto , cfo ):-**

1. **Ceo: Chief Executive Officer.**

**The person who has most authority in an organization or the business**

1. **Coo: Chief Operating Officer.**

**Senior who responsible for day to day administrative and operational function in a business**

1. **Cto: Chief Technical Officer.**

**Executive level person who responsible for overseeing the development in technology needs in the business**

1. **Cfo: Chief Financial Officer.**

**The executive senior who responsible for managing the financial action of the business**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Task 2**

**Difference among ( unit testing , system testing , integration testing , acceptance testing )**

**1-Unit Testing: test all parts of the system individually and check if it proper operation or not**

**2- System Testing: test the way which all components of the system contact with each other**

**3-Integration Testing: test which all components of the system are tested as one group**

**4-Acceptance Testing: last phase in software testing that test what degree the system meets with the client needs**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Task 3**

**V\_model: is SDLS where execution process happens in sequential way**

**Also known as verification and validation model**

**The advantages of the V-Model method:**

* **This is a highly-disciplined model and Phases are completed one at a time.**
* **Works well for smaller projects where requirements are very well understood.**
* **Simple and easy to understand and use.**
* **Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.**

**The disadvantages of the V-Model method :**

* **High risk and uncertainty.**
* **Not a good model for complex and object-oriented projects.**
* **Poor model for long and ongoing projects.**
* **Not suitable for the projects where requirements are at a moderate to high risk of changing.**
* **Once an application is in the testing stage, it is difficult to go back and change a functionality.**
* **No working software is produced until late during the life cycle.**

**Task 4**

**Best Monitoring Tools for Servers**

**1. Sematext Monitoring**

**2. Prometheus and Grafana**

**3. SolarWinds Server & Application Manager**

**4. Dynatrace**

**5. Datadog**

**6. New Relic**

**7. ManageEngine OpManager**

**8. AppDynamics**

**9. Sensu Go**

**10. Nagios XI**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Task 5**

**Ci/Cd ( continuous integration / continuous delivering ):-**

**CI/CD falls under DevOps (the joining of development and operations teams) and combines the practices of continuous integration and continuous delivery. CI/CD automates much or all of the manual human intervention traditionally needed to get new code from a commit into production, encompassing the build, test (including integration tests, unit tests, and regression tests), and deploy phases, as well as infrastructure provisioning. With a CI/CD pipeline, development teams can make changes to code that are then automatically tested and pushed out for delivery and deployment.**[**Get CI/CD right**](https://about.gitlab.com/blog/2020/07/06/beginner-guide-ci-cd/)**and downtime is minimized and code releases happen faster.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Task 6**

Phases of the DevOps lifecycle

* **Plan : The planning phase is exactly what it sounds like: planning the project’s lifecycle. In contrast to conventional methods to the development lifecycle, this model assumes that each stage will be repeated as necessary. In this manner, the DevOps workflow is planned with the likelihood of future iterations and likely prior versions in mind.  
  This implies that we will likely have information from past iterations that will better inform the next iteration, and that the present iteration will likewise inform the next iteration. This stage often involves all teams to ensure that no area of the planning is ignored or forgotten.**
* **Code : The developers will write the code and prepare it for the next phase during the coding stage. Developers will write code in accordance with the specifications outlined in the planning phase and will ensure that the code is created with the project’s operations in mind.**
* **Build :Code will be introduced to the project during the construction phase, and if necessary, the project will be rebuilt to accommodate the new code. This can be accomplished in a variety of ways, although GitHub or a comparable version control site is frequently used.  
  The developer will request the addition of the code, which will then be reviewed as necessary. The request will be approved if the code is ready to be uploaded, and the code will be added to the project. Even when adding new features and addressing bugs, this method is effective.**
* **Test : Throughout the testing phase, teams will do any necessary testing to ensure the project performs as planned. Teams will also test for edge and corner case issues at this stage. An “edge case” is a bug or issue that only manifests during an extreme operating event, whereas a “corner case” occurs when many circumstances are met.**
* **Release : The release phase occurs when the code has been verified as ready for deployment and a last check for production readiness has been performed. The project will subsequently enter the deployment phase if it satisfies all requirements and has been thoroughly inspected for bugs and other problems.**
* **Deploy : In the deploy phase, the project is prepared for the production environment and is operating as planned in that environment. This would be the responsibility of the operations team; in DevOps, it is a shared responsibility. This shared duty pushes team members to collaborate to guarantee a successful deployment.**
* **Operate : In the operating phase, teams test the project in a production environment, and end users utilise the product. This crucial stage is by no means the final step. Rather, it informs future development cycles and manages the configuration of the production environment and the implementation of any runtime requirements.**
* **Monitor : During the monitoring phase, product usage, as well as any feedback, issues, or possibilities for improvement, are recognized and documented. This information is then conveyed to the subsequent iteration to aid in the development process. This phase is essential for planning the next iteration and streamlines the pipeline’s development process.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Task 7**

Difference among ( Mvp , Mvc ,Mvvm ) :-

1-Mvp : **minimum variable product**

**a product with enough features to attract early-adopter customers and validate a product idea early in the product development cycle**

2-Mvc : **model view controller**

**the name of a methodology or design pattern for successfully and efficiently relating the user interface to underlying data models**

3-Mvvm : **model-view-view-model**

**a software design pattern that is structured to separate program logic and user interface controls**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***