Dev Ops Assignment 1

Collaboratively done by Akshat Singhal and Isha Ipsita

1.) **The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is the first SDLC model that was used for software development. List and explain the phases of the model and give a couple of advantages and disadvantages of the Model.**

Answer 1.) Waterfall model was among the first models to be used for software development practices, It contains a multitude of well-connected non iterative steps each step feeding input into the other, hence the name linear sequential life cycle model.

Phases of Waterfall or SDLC:

- Study of resources ie. technical and financial feasibility :

  This stage includes team leaders visit the client and understanding roughly the requirement, and understanding input-output data to be produced. The constraints, conditions they need to levie upon the work. Look into solutions and have an understanding of the problem statement. Solutions are examined on cost, time and what fits the best for the requirements.

- Requirement Analysis:

  In this stage exact needs and requirements are understood and this phase ends with the formation of an SRS document. The SRS document mainly contains functional as well as non-functional requirements and the goal of implementation.

- Design Phase:

  Divided into two phases high level design and low-level design

- Coding and Unit Testing:

  In the software development lifecycle (SDLC), the testing phase is where you focus on research and discovery. During the testing phase, developers check to see if their code and programming meet the needs of the customer. While it is impossible to fix all of the faults discovered during the testing phase, the results of this phase can be used to reduce the number of flaws in the software programme. The project team must first create a test plan before testing can commence. The test plan outlines the types of testing you'll do, as well as the resources you'll need, how the software will be tested, who should be the testers at each stage, and test scripts.

- Integration and system testing: comprises of alpha, beta and acceptance testing

  Alpha being done by the development team, beta by a set of known customers and acceptance by the main client. It is carried out in a planned manner by the system test plan document.

- Maintenance:

  The main goal in software development is the end goal i.e., the client satisfaction, by system maintenance troubleshooting is done and the client is given a functional up to date software that is functional in all aspects, maintenance is also of corrective, preventive and adaptive types.

***2.)* Give a brief synopsis of what the Agile method is, and list the 12 principles of this game changing methodology ?**

Answer 2.) Agile methodologies are a type of project management process, mainly used for software development, where demands and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers.

The Agile methodology is a collection of principles that value adaptability and flexibility. Agile aims to provide better responsiveness to changing business needs and therefore focuses on enabling teams to deliver in workable increments.

Stemming from the values and principles of the Agile Manifesto, it was created as a response to the inadequacies of traditional development methods such as the Waterfall Model. The software industry is a highly competitive market due to the fact that software is something that can be continuously updated. This means that developers need to constantly improve and innovate their products to keep on top of the game—and the linear, sequential approach of the Waterfall method just wasn’t cutting it.

12 Principles of this agile methodology are:-

1. Our highest priority is to satisfy the customer through an early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in the development. Agile processes harness change for the customer's competitive advantages.
3. Deliver working software frequently, from a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is a face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors,developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity- the art of maximizing the amount of work not done- is essential.
11. The best architectures, requirements and design emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Question 3.) **Agile manifesto has 4 key values. One of these values is usually overlooked. Give a brief explanation for each value and define the one value that is usually overlooked and explain the major importance of it.**

Answer 3.) Agile Manifesto values :-

1.) Individual and team interactions over processes and tools

People drive the development process since they are the ones who respond to changing business needs and develop processes and tools in response to change.If a team prioritizes processes, technology or tools then the individuals in the team become less responsive and meeting the customer requirement becomes a difficult task. If individuals are valued over processes and tools, the communication becomes fluid and the interactions happen, based on the change in business requirements.

2.) Working software over comprehensive documentation.

Create and deliver value. Agile does not rule out the need for documentation. Agile emphasizes that the document is streamlined in a way that it gives the exact picture to the developer of what exactly is needed to build the software, without getting lost in the intricacies.

The requirements are documented as user stories so the developer gets clarity on the business requirements. This eventually becomes a guide for the developer to start building the exact functionalities. Agile also emphasizes that delivering working pieces of the software at frequent intervals matters the most to the customer.

3.) Customer collaboration over contract negotiation

Negotiation enables the customer and the product manager to work out the details of the product delivery. There is room for re-negotiation as well. In traditional software processes are more before the start of the development process and after the product is completely ready.

The development team and the customer can make sure that the requirements are met at each and every point in time during the development process.

4.) Responding to change over following plan

A Change was considered expensive and avoidable by the traditional software development methodologies.

Changes in customer's business needs-direct impact on developer's plans.

A process of method tailoring describes the Agile approach towards change. It is defined as a process or capability of a human agent to determine a system development approach for a specific project situation through responsive changes in, and dynamic interplay between contexts, intention and method fragments.

Question 4.) **There are different aspects of a quality software team. We have the Developmental team and we have Operations. What do each bring to the table? What are the benefits from each side?**

Answer 4.) The main goal in the software development world is to bring to the client a well functional software that meets the ever-changing demands of the user and has been trialed and tested systematically and deemed flawless. Development team handles the functionalities and actual functioning and writing of the code; hence their role is to keep changing and writing code. Whereas the Operations team have on their backs the role of making the code operational and hence crave for stability. The roles of the two teams are fundamentally clashing; here comes the role of dev ops using which the clash between the two teams could be minimized and software could be developed and delivered smoothly according to client satisfaction and need.

Question 5.) What is Devops? How does it differ from Agile? What are the benefits, similarities and usefulness for creators?

Answer 5.) DevOps is a software development practice that promotes collaboration between development and operations, resulting in faster and more reliable software delivery. Commonly referred to as a culture, DevOps connects people, process, and technology to deliver continuous value.

- DevOps is a practice of bringing development and operations teams together whereas Agile is an iterative approach that focuses on collaboration, customer feedback and small rapid releases.
- DevOps focuses on constant testing and delivery while the Agile process focuses on constant changes.
- DevOps requires a relatively large team while Agile requires a small team.

- DevOps leverages both shifts left and right principles, on the other hand, Agile leverages shift-left principle.
- The target area of Agile is Software development whereas the Target area of DevOps is to give end-to-end business solutions and fast delivery.
- DevOps focuses more on operational and business readiness whereas Agile focuses on functional and non-function readiness.

__Question 6.)__ Breakdown what CAMS stands for. Give an example of each part of this amazing methodology?

__Answer 6.)__ The CAMS model was invented by Damon Edwards and John Willis, authors of the famous Podcast DevOps Cafe. CAMS stands for Culture, Automation, Measurement and Sharing.

Culture
DevOps culture emphasizes on **offering high quality service**.

Adapting to a DevOps culture means transforming into an organization that focuses on delivering a high-quality service. Service not only means making the product available to the customer, but also it is to ensure that the product is in the best of its shape and it constantly meets the requirements of the customers.

Service mindset is a critical aspect of DevOps culture, since the products developed with service mindset will meet the following conditions:

- Quality is given utmost importance right from the source.
- Early discovery of errors, especially right at the activity where these are created.
- Solving problems is easier and cheaper.
- Happy customers.

Examples include:

- Enquiring users regarding the proper functioning of the product at regular intervals and providing insurance service. This is followed by electronic companies like **LG**.
- Upgrading the software of products previously purchased. This is followed by most software-based companies like **Tesla** upgrading its cars regularly**.**

Automation

Technology advancements are leading to **automation of routine tasks**.

Job surveys conducted across the sectors have started showing the shift towards automating routine jobs and how today there is a heavy need for skills to automate. Today, automation is applicable to different industry sectors such as manufacturing, healthcare, transportation and IT.

Automated processes are repeatable. Cost of process execution is minimized. In other words, automation results in predictable and standardized outcomes. These concepts can be applied to software delivery processes.

Examples include:

- Using a task management software like **Facebook workspace** to better manage work.
- Using meeting scheduling software to better manage collisions in meeting

Measurement

Measurement is fundamental for continuously offering value and improvement. Tracking important metrics and offering feedback helps improve the process.

Measurement and monitoring have to be integrated into your day-to-day processes as part of DevOps implementation. Three ways of feedback model are:

- The First Way focuses the entirety of the system and **ensures that none of the defects at a local level affect the overall flow**, performance or stability. This method ensures there is one single flow of feedback.

- The Second way is about creating **bidirectional feedback loops** of moving from left to right and right to left. For this to happen, the feedback loops should be shorter in order to receive and send feedback sooner. This method helps in having a deeper understanding of the process.
- The Third way of measurement is to **create an environment of continuous learning**, feedback and room for failure. In this method, we build an MVP, receive feedback and make changes accordingly. It rewards teams for risk taking despite the failure.

Examples include:

- Regular **employee surveys** to capture the overall sentiment. Most companies do so at regular intervals.
- Capturing the issues reported by end users. For Software products, this is done using issue trackers like **GitHub issues**.

Sharing

Sharing is one of the most important aspects of the CAMS model.

DevOps by definition is the collaboration of development and operations and sharing is fundamental to this. Implementing DevOps involves constant interaction and most importantly knowledge sharing. Every team within an organization has encountered failures with regard to the people, processes, tools, projects etc. This knowledge has to be shared within an organization among the team members to ensure successful implementation of DevOps.

Examples include:

- To ensure such an environment, companies provide time outside of work so that employees can mix.
- Formation of teams plays a major role in this as well.

Question 7. You have graduated. You are about to start your amazing career. What motivates you to be a successful engineer? What is important to you, what will be important to your future goals and what are parts of you that you need to reflect on to improve yourself as a good coder and a better person?

Answer 7.) The pursuit of engineering has always compelled me, the ability to create something is a role of an engineer and I shall forever be motivated in making, doing or developing stuff that are meant for the greater good. What is more important to me I could answer as always evolving, code has taught me to evolve continuously as a person, as a learner; always strive for better methods that are more effective, short, that save time and computational space. For my future goals, I would like to keep learning and making clarity in my understanding to turn into a good coder, especially having an exceptional understanding in the algorithms that govern the crux of the code. Also, if I could not be too ambitious may be find flaws in the existing ones and make new ones under my own name. As I am on my way to be a good coder I must acknowledge and understand code, as for me code is a vortex of possibilities that can make life easier and be incorporated everywhere, if u see closely our DNA is also a piece of code and that excites me.