USER-CENTERED DESIGN CSP 588

Participation – P1

Name: Satyam Rajput (A20537375)
Email Id: srajput3@hawk.iit.edu

Participation – 1

1. Discuss (at least 200 words) the role of “agility” in improving the user experience.

   The **Agile** method followed in the **Engineering Life Cycles** emphasizes **iteration** as a fundamental principle to drive the continuous improvement of products and services. This iterative method incorporates both **macro** and **micro** levels of evolution, enabling more deliberate evolution of versions at a macro level and quick prototyping improvement at a micro level. The principal aim is to guarantee the regular provision of significant **chunks** of high-quality work products, prioritising their usability and usefulness.

   – Micro: quickly evolve prototypes

   – Macro: more methodically evolve versions

   Chunks should be usable and useful (prioritize)

   Agile UX

Agile UX strongly emphasises concept and possibility exploration early in the design process, with the delivery of functional elements coming later. This methodology permits the appearance of necessary modifications as the design develops, promotes the delivery of features with limited scope, encourages learning through experimentation and feedback, creates team spirit, and guarantees the delivery of a useful user experience.

Explore Concepts and Possibilities Early, Deliver Working Elements Later :

Agile UX promotes early design concepts and possibility exploration, allowing teams to analyse different ideas and strategies before getting on a particular design direction. Teams can gather valuable insights and iterate on concepts before finalising the user experience by delaying the delivery of working elements to later stages.

Rapid Delivery of Small Scope Minimizes Straying from User Need:

Agile UX's quick delivery of features with limited scope reduces the possibility of breaking too much from user needs. Teams can maintain alignment with user requirements and make incremental improvements based on user feedback by concentrating on delivering small, targeted elements.

Learning by Trying (Feedback):

Agile UX promotes teams to adopt a "learn by trying" mentality, actively seeking out user feedback and iterating designs in response to feedback from users and real-world usage.

Required Changes Reveal Themselves

Agile UX is iterative and exploratory, so as the design develops, necessary adjustments to the user experience and design become apparent. This enables teams to modify and enhance the user experience in response to changing project requirements, new information, and user feedback.

Getting It Done Quickly, but Getting It Done Right

Agile UX places a strong emphasis on the necessity of providing the user experience promptly without sacrificing quality. Teams can strike a balance between quality and speed by emphasising continuous improvement and quick delivery, ultimately resulting in an impactful and productive user experience.

2. Discuss (at least 200 words) the “engineering” challenges imposed by providing agility.

We know that according to **Engineering Life Cycles**
In The waterfall model
– Forced completion of each phase takes place yet quick changes in the System and version could be an issue.

Iteration and agility
– Evolve quickly which requires time and efforts for resolving complexity for stable change and subtle environment.

Discipline is **still required to ensure the timely delivery** of **useful versions**

Organizational Challenges

Structural Adaptation: Agile principles might not be compatible with conventional hierarchical structures. Cross-functional teams must be reorganised and given more authority by organisations to promote cooperation and self-organization

Alignment with Business Objectives: Engineering efforts must be in line with business objectives to demonstrate agility. It is crucial to make sure that agile practices help the company and its clients receive value.

Technical Challenges

Systems: Technical debt and old systems can hinder engineering agility. Reworking and updating old code to make it more flexible is a major technical task.
Integration and Testing: To guarantee that frequent changes have no impact on the overall stability of the system, agile engineering requires strong integration and testing procedures.

Operational Challenges

Continuous Delivery: Demonstrating agility requires the establishment of a strong continuous delivery pipeline. To enable frequent and dependable software releases, this entails automating the build, test, and deployment processes.

Managing Complexity: Agile engineering often involves handling growing complexity as a result of ongoing modifications and changing specifications. Securing a balance between stability, scalability, and agility is a major operational challenge.

Balancing Agility with Stability:

Finding a balance between stability and agility is necessary to make sure that quick changes don't compromise the system's endurance and dependability.

Real-World Example

To overcome the problem of frequent changes, continuous integration and deployment pipelines are set up in the software domain to automate testing and deployment.

Engineering agility requires embracing innovative practices, addressing technical, operational, and cultural challenges, and making trade-offs. Examples from the real world show how different engineering specialities address these problems, highlighting the creativity and adaptability that engineering's agility promotes.