**SHIVAM SHANDILYA**

**2017UCO1595**

* **Write short notes on following**

1. **Scrum**

The scrum framework is a heuristic software development framework that is based on continuous learning and making adjustments according to fluctuating factors. It acknowledges that the development team doesn’t know everything at the start of a project and will evolve through experience. Scrum is structured to help teams naturally adapt to changing conditions and user requirements, with re-prioritization built into the process and short release cycles so your team can constantly learn and improve.

Artifacts are something that we make, like a tool to solve a problem. In Scrum, these three artifacts are a product backlog, a sprint backlog, and an increment with your definition of “done”. They are the three constants in a scrum team that we continue to revisit and invest in overtime.

1. **Lean Development**

The Lean Software Development process aims at removing all things unnecessary, all the extra stuff, from software engineering. Its principles are largely similar to those of Agile development; [some even consider the two methodologies](https://martinfowler.com/bliki/AgileVersusLean.html) to be almost equivalent to each other. The only difference that exists between these two software development processes is that [lean is more convergeable](https://gupea.ub.gu.se/bitstream/2077/38521/1/gupea_2077_38521_1.pdf) in terms of development strategy.

Overall, there are 7 principles of Lean software development, each aiming to quicken delivery and bring higher value to end-user:

* Eliminating Waste
* Building Quality In
* Amplifying Knowledge
* Delaying Commitment
* Delivering Fast
* Respecting people
* Optimizing the whole thing

**3. Extreme programming (XP)**

Extreme Programming is one of the many Agile frameworks applied by companies for their software development process. But it’s key feature is its emphasis on technical aspects of software development which distinguishes Extreme Programming from the other approaches.

Extreme Programming has simple rules that are based on 5 values:

1. **Communication:** Everyone on a team works jointly at every stage of the project.
2. **Simplicity:** Developers strive to write simple code bringing more value to a product, as it saves time and efforts.
3. **Feedback:** Team members deliver software frequently, get feedback about it, and improve a product according to the new requirements.
4. **Respect:** Every person assigned to a project contributes to a common goal.
5. **Courage:** Programmers objectively evaluate their own results without making excuses and are always ready to respond to changes.

**4. Adaptive Software Development (ASD)**

Adaptive Software Development is a software development process which embodies the principle that continuous adaptation of the process to the work at hand is the normal state of affairs.

Adaptive Software Development is used to make teams more adaptable to changing customer demand, requirements and market needs. Specifically, it does this by promoting lightweight planning and ongoing learning, through a three-phase process: speculate, collaborate, and learn.

**5. Feature Driven Development**

An Agile methodology for developing software, Feature-Driven Development (FDD) is customer-centric, iterative, and incremental, with the goal of delivering tangible software results often and efficiently. FDD in Agile encourages status reporting at all levels, which helps to track progress and results.

FDD allows teams to update the project regularly and identify errors quickly. Plus, clients can be provided with information and substantial results at any time. FDD is a favorite method among development teams because it helps reduce two known morale-killers in the development world: Confusion and rework.

Typically used in large-scale development projects, five basic activities exist during FDD:

1. Develop overall model
2. Build feature list
3. Plan by feature
4. Design by feature
5. Build by feature