**Teamwork II. (Agile Software Development)**

**Team 7**

Binh Bui – e1300518

Nguyen Cong Danh – e1400467

Thanh Vuong – e1400489

1. **What is the main idea of Agile software development model?**

Agile development is a different way of managing IT development teams and projects, based on iterative and incremental development. The requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. Agile development values:

* Individual and interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan progress

Agile methods are considered as lightweight methods, which are perceived as providing a little control, much less document oriented than heavyweight methods, the team considers the source code to be the project documentation.

Here are some of the advantages of lightweight methodologies:

* They accommodate change well.
* They are people oriented rather than process oriented. They tend to work with people rather than against them.
* They are complemented by the use of dynamic checklists.
* The project teams are smaller.
* They rely on working in a team environment.
* They foster knowledge sharing.
* Feedback is almost instantaneous.
* The project manager doesn’t need to develop “heavy” project documentation but instead is able to focus on the absolute necessary documentation (e.g., the project schedule).
* They are learning methodologies. After each build or iteration, the team learns to correct issues on the project, forming an improvement cycle throughout the project.

1. **What are the main principle of Agile software development model?**
   * Customer satisfaction by early and continuous delivery of useful software
   * Requirements are flexible to change throughout the process, even in the late stage
   * Prototypes are produced frequents (time measured in weeks), which customers can track it
   * Working software is the principal measure of progress, reduce the heavy load and needs of documentation
   * Small daily meeting weekly meeting between team members
   * Face-to-face conversation is the best form of communication.
   * Projects are built around motivated individuals, who should be trusted.
   * Continuous attention to technical excellence and good design.
   * Self-organizing, cross-functional team

* Regular adaptation to become more effective
* The development is incremental and iterative.

1. **What are similarities and differences between RUP and Agile software development models?**

* Similarities:
  + They are both iterative model.
* Differences:

|  |  |  |
| --- | --- | --- |
|  | **RUP** | **SCRUM** |
| **Cycle** | Divided into 4 major phases (Inception, Elaboration, Construction, and Transition). Even though the phases can be concurrent, it is generate that specific activities will peak during certain phases. | Each iteration is a complete cycle. |
| **Artifact** | Scope Document, Requirement Document, System Architecture Document, Development Plan, Test Plan… | Documents are reduced a lot in SCRUM, instead it has only a few as Product Backlog, Sprint Backlog. |
| **Scope** | Scope are set clear at the beginning of the project, though during the process, it might be revised but changes are strictly controlled. | SCRUM uses Product Backlog instead of scope, which is re-defined and re-evaluated at the end of each sprint. |
| **Planning** | Formal planning is used, it has a detailed list of activities and tasks with a more specific end-date and milestones. | No detailed Project Plan. Each iteration workload and date is determined at the beginning and the end of the sprint. Usually the SCRUM master and Product Owner decides the date. |

1. **What are similarities and differences between Scrum, XP and FDD programming models?**

* Similarities
  + Highly Collaborative
  + High focus on communication and collaboration
  + Completely developed and tested features in short iterations
  + Changes are easy to adapt
  + Revision and Evaluation are frequent
* Differences

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Scrum** | **XP (Extreme Programming)** | **FDD (Feature Driven Development)** |
| Duration | • Each iteration (called Sprints) last from two weeks to one month long. | • Each iteration lasts from one or two weeks long. | • Each iteration cycle is usually 2 weeks. |
| Planning | • No detailed project plan.  • Each iteration workload and date is determined at the beginning and the end of the sprint.  • Usually the SCRUM master and Product Owner decides the date. | • The work is processed under a strict priority order.  • Features are developed prioritized by the customer and the team is required to set the cornerstone on that. | • FDD has well defined control points (milestones) to track the progress of every task.  • Those control points (Walkthrough, Design, Design Inspection, Coding, Code Inspection and Promote to build) to help project manager to track the progress of every task (feature) |
| Requirement | • Does not prescribe any engineering practice | • Test-driven development, simple design, Automated testing, Pair programming. | • Does not specify any particular engineering practice |
| Team | • Self-organizing & Cross-functional team | • There are general roles such as Tracker, Customer, Coach, Manager but they are not intensively focused on. Also, anyone can be the role Doomsayer. | • Feature team have recognized roles (Project Manager, Chief Architect, Class Owner,..) |