

Aluno: Lucas Spicollys Leão de Lima

2.5 - Across the process. Management and control of progress, quality, change, and risk, are processes that can appear in any moment. Knowing this, it's essential to be prepared to use those activities when necessary. (Pressman Chapter 2, page 18)

4.10 - Because software process provides stability, control, and organization to an activity that can, if left uncontrolled, become quite chaotic. So even that your software or a component is correct, when you follow process models, solve problems can be easily, the flow of activities, actions and tasks works well and you especially when you use agile development. (Pressman Chapter 4).

5.3 - Agile process are iterative or incremental. Once each iteration is a kind of small project, manage them become easier. That is, when problems occur in small scale, we can identify issues and control them fast. It's possible to complete a task with just one iteration, but an entire project, I think isn't, because iterations are core of agile process, so it's hard to think about the agile project without them.

<https://searchcio.techtarget.com/definition/Agile-project-management>

7.9 - In the context of process schedule, a high granularity refers the level of detail of the schedule. That is, a high granularity schedule have all tasks described with minimum details and a low granularity schedule have more general tasks. (Pressman Chapter 7 page 114)

7.13 - According Pressman (page 123), a successful test is one that uncovers an as-yet-undiscovered error, that is, when you realize a test and you didn't found an error, probably you are very lucky or you didn't the test well. So it's indispensable to be thorough when you do a test.

8.3 – When it happens the each customer defends its own point of view, proposing conflicting requirements and saying that their version is “essential for our special needs”. According Pressman (page 135) we have to reconcile these conflicts through a process of negotiation. In combination of asking and raking the priority of requirements and using iterative process, the internal conflicts can be eliminated.

8.17 – “It encompasses five tasks: (1) distributed debugging uncovers errors and determines their cause, (2) run-time verification determines whether software matches its specification, (3) run-time validation assesses whether the evolving software meets user goals, (4) business activity monitoring evaluates whether a system satisfy business goals, and (5) evolution and co-design provides information to stakeholders as the system evolves.” (Pressman, pages 160 and 161)

10.7 – It is a model of class representation grouped by packages that relate to each other or show a state or behavior pattern. It might be used to categorize elements of the requirements model (as an example, use cases or analysis cases). (Pressman page 199)

11.2 – A sequence diagram shows the execution of an use case for the application, and the objects that are involved in each use case related. It can show paths that an object or actor does through the use case. State diagram shows various states that are valid for an object (it could be anything, like a method or an entire class). It shows what actions are valid for a given object, depending on what state it is.

<https://softwareengineering.stackexchange.com/questions/102346/difference-between-statechart-and-sequence-diagram>