**SSW 555 Agile Methods for Software Development**

**Quiz 3: Scrum**

**Name: Mrunal Phadke**

**---------------------------------------------------------------------------------------------------------**

**1.  What is the difference between defined and empirical processes?**

**Answer:**

* Scrum is founded on empirical process control theory, or empiricism.
* Empiricism asserts that knowledge comes from experience and making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk.
* **An empirical process is seen as a black box and you evaluated it’s in and outputs. This is done by defining checkpoints that should occur at predefined points.** **It is used for handling processes that are complex and not very well understood.**
* As checkpoints in Scrum we have daily scrums, sprint planning meeting, sprint review meeting and sprint retrospective. Since each sprint is time boxed, the checkpoints will always occur at a predefined time.
* Inspection is done at each checkpoint so you can adapt the process with the information collected through observation, experience and experimenting.
* A defined process is derived from first principles, meaning in science that it needs to adhere to laws of nature and other fundamental and well defined laws.
* **The defined process control model requires that every piece of work be completely understood. Given a well-defined set of inputs, the same outputs are generated every time. A defined process can be started and allowed to run until completion, with the same results every time.**
* A defined process is an amount of tightly coupled steps where the output from one step is the input to the next step and where no observation or evaluation of the output is done to feedback to the process. A defined process when started will run to the end without any checkpoint. The output from a defined process should always be the same or with little variance given the same input to the process. Eg : Waterfall model

**---------------------------------------------------------------------------------------------------------**

**2. What is the difference between the Product Backlog and the Sprint Backlog?**

**Answer:**

* **Product Backlog:**
* **The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product.** The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.
* A Product Backlog is never complete. The earliest development of it only lays out the initially known and best-understood requirements.
* The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. As long as a product exists, its Product Backlog also exists.
* The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate and value.
* **Sprint Backlog:**
* **The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal.** The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality into a “Done” Increment.
* The Sprint Backlog makes visible all of the work that the Development Team identifies as necessary to meet the Sprint Goal.
* The Sprint Backlog is a plan with enough detail that changes in progress can be understood in the Daily Scrum. The Development Team modifies the Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint. This emergence occurs as the Development Team works through the plan and learns more about the work needed to achieve the Sprint Goal.
* As new work is required, the Development Team adds it to the Sprint Backlog. As work is performed or completed, the estimated remaining work is updated. When elements of the plan are deemed unnecessary, they are removed. Only the Development Team can change its Sprint Backlog during a Sprint. The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint, and it belongs solely to the Development Team.

**----------------------------------------------------------------------------------------------------------**

**3. Why are features "locked" during a sprint?**

**Answer:**

* Scrum is process of implementing Agile, where features are delivered in a ‘n’ day sprints (where n is any positive integer).
* In scrum you define the requirements for a ‘n’ day sprint and work on them with vigor for ‘n’ days without being sidetracked by other things or having things re-prioritized.
* A specific feature is not recognized as being completed until it is analyzed, designed, coded, tested, re-factored and documented. At the end of the ‘n’ day sprint, most features defined in the ‘n’-day sprint should be completed.
* If some did not get finished (because of being underestimated), the uncompleted features can be moved to a later sprint. A sprint is considered successful if all the completed features have high quality and can be put into production (or beta) upon ending the sprint.
* An important component of Scrum is using a time-box approach, where meetings and events have a definite time period and this time-box is strictly enforced.
* **Once the features are locked in for the ‘n’-day sprint, no changes are allowed** (new features cannot be introduced until the next sprint).
* Features and priorities need to locked to define that sprint. If features and priorities keep on changing, it destroys the purpose of developing sprints (plans).
* **Objective of Sprint is to maintain focus on delivering one thing that has highest priority**.
* **Locking the features helps to avoid any interruptions and achieve high developer productivity**.
* Thus, **to achieve a definitive work plan for the development team to follow for some next number of ‘n’ days, features need to be locked.**

**-----------------------------------------------------------------------------------------------------------**

**4. What is the purpose of a Burn‐Down Chart?**

**Answer:**

* **The burndown is a chart that shows how quickly you and your team are burning through your customer's user stories.** It shows the total effort against the amount of work we deliver each iteration.
* Basically it can be defined as graphical view of remaining work Versus time. It may show estimated and actual values
* In this chart, the slope of line is Velocity.
* The Burn down chart is mostly used to predict end of project.
* It looks something like this :

