**Agile:**

* + Introduction
  + Agile Project Management
  + What is Scrum?
  + Functionality of Scrum
  + Components of Scrum
    - Scrum Roles
    - The Process
    - Scrum Artifacts

**>>Introduction:**

* Classical methods of software development have many disadvantages:
  + huge effort during the planning phase
  + poor requirements conversion in a rapid changing environment
  + treatment of staff as a factor of production
  + New methods: Agile Software Development

**>>Manifesto for Agile SD**

* Based on the Manifesto for Agile Software Development
  + **Individuals and interactions** over processes and tools
  + **Working software** over comprehensive documentation
  + **Customer collaboration** over contract negotiation
  + **Responding to change** over following a plan

**>>Agile Project Management**

* Qualities:
  + Minimize risk 🡪 short iterations
  + Real-time communication (prefer face-to-face) 🡪 very little written documentation
  + Indicated for unpredictable / rapidly changing requirements

**>>Agile Methods**

* Agile methods:
  + Scrum
  + Extreme Programming
  + Adaptive Software Development (ASD)
  + Dynamic System Development Method (DSDM)
  + …
* Agile Alliance
  + A non-profit organization promotes agile development

**>>What is Scrum?**

* Definition from rugby football:
* a scrum is a way to restart the game after an interruption, where the forwards of each side come together in a tight formation and struggle to gain possession of the ball when it is tossed in among them

**>>Scrum - an agile process**

* SCRUM is an agile, lightweight process for managing and controlling software and product development in rapidly changing environments.
  + Iterative, incremental process
  + Team-based approach
  + developing systems/ products with rapidly changing requirements
  + Controls the chaos of conflicting interest and needs
  + Improve communication and maximize cooperation
  + Protecting the team form disruptions and impediments
  + A way to maximize productivity

**>>Functionality of Scrum**



**>>Components of Scrum**

* + Scrum Roles
  + The Process
  + Scrum Artifacts

**>>Scrum Master**

* Represents management to the project
* Typically filled by a Project Manager or Team Leader
* Responsible for enacting scrum values and practices
* Main job is to remove impediments

**>>The Scrum Team**

* Typically 5-10 people
* Cross-functional (QA, Programmers, UI Designers, etc.)
* Members should be full-time
* Team is self-organizing
* Membership can change only between sprints

**>>Product Owner**

* Acts like one voice (in any case)
* Knows what needs to be build and in what sequence this should be done
* Typically a product manager

**>>The Process**

* Sprint Planning Meeting
* Sprint
* Daily Scrum
* Sprint Review Meeting

**>>Sprint Planning Meeting**

* A collaborative meeting in the beginning of each Sprint between the Product Owner, the Scrum Master and the Team
* Takes 8 hours and consists of 2 parts (“before lunch and after lunch”)

**>>Parts of Sprint Planning Meeting**

* 1st Part:
  + Creating Product Backlog
  + Determining the Sprint Goal.
  + Participants: Product Owner, Scrum Master, Scrum Team
* 2nd Part:
  + Participants: Scrum Master, Scrum Team
  + Creating Sprint Backlog

**>>Pre-Project/Kickoff Meeting**

* A special form of Sprint Planning Meeting
* Meeting before the begin of the Project

**>>Sprint**

* A month-long iteration, during which is incremented a product functionality
* NO outside influence can interference with the Scrum team during the Sprint
* Each Sprint begins with the Daily Scrum Meeting

**>>Daily Scrum**

* Is a short (15 minutes long) meeting, which is held every day before the Team starts working
* Participants: Scrum Master (which is the chairman), Scrum Team
* “Chickens” and “Pigs”
* Every Team member should answer on 3 questions

**Questions**

* What did you do since the last Scrum?
* What are you doing until the next Scrum?
* What is stopping you getting on with the work?

**>>Daily Scrum**

* Is NOT a problem solving session
* Is NOT a way to collect information about WHO is behind the schedule
* Is a meeting in which team members make commitments to each other and to the Scrum Master
* Is a good way for a Scrum Master to track the progress of the Team

**>>Sprint Review Meeting**

* Is held at the end of each Sprint
* Business functionality which was created during the Sprint is demonstrated to the Product Owner
* Informal, should not distract Team members of doing their work

**>>Scrum Artifacts**

* Product Backlog
* Sprint Backlog
* Burn down Charts

**>>Product Backlog**

* Requirements for a system, expressed as a prioritized list of Backlog Items
* Is managed and owned by a Product Owner
* Spreadsheet (typically)
* Usually is created during the Sprint Planning Meeting
* Can be changed and re-prioritized before each PM

**>>Estimation of Product Backlog Items**

* Establishes team’s velocity (how much Effort a Team can handle in one Sprint)
* Determining units of complexity.
  + Size-category (“T-Shirt size”)
  + Story points
  + Work days/work hours
* Methods of estimation:
  + Expert Review
  + Creating a Work Breakdown Structure (WBS)

**>>Product Backlog**

* Is only a FORECAST!-> is not exact

**>>Sprint Backlog**

* A subset of Product Backlog Items, which define the work for a Sprint
* Is created ONLY by Team members
* Each Item has it’s own status
* Should be updated every day
* No more then 300 tasks in the list
* If a task requires more than 16 hours, it should be broken down
* Team can add or subtract items from the list. Product Owner is not allowed to do it
* Is a FORECAST!
* Is a good warning monitor

**>>Burn down Charts**

* Are used to represent “work done”.
* 3 Types:
  + Sprint Burn down Chart (progress of the Sprint)
  + Release Burn down Chart (progress of release)
  + Product Burn down chart (progress of the Product)

**>>**Burn down Charts

* X-Axis: time (usually in days)
* Y-Axis: remaining effort

**>>Sprint Burn down Chart**

* Depicts the total Sprint Backlog hours remaining per day
* Shows the estimated amount of time to release
* Ideally should burn down to zero to the end of the Sprint
* Actually is not a straight line
* Can bump UP

**>>Release Burn down Chart**

* Will the release be done on right time?
* X-axis: sprints
* Y-axis: amount of hours remaining
* The estimated work remaining can also burn up

**>>Product Burn down Chart**

* Is a “big picture” view of project’s progress (all the releases)

20.41