**Agile Manifesto**

* Individuals & interactions over processes and tools
* Working software over detailed documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan

**Agile Advantages**

* **Adaptive team** responds to changing requirements.
* **Time and effort** not wasted
* **Face to face communication** with customer 4 input
* Documentation is concise and saves time
* High quality **software in least possible time** duration & satisfied customer. *80/20 rule*

**Agile Disadvantages** *Methodology*

* Difficult to assess the effort required at beginning of the life cycle for larger projects
* Lack of emphasis on designing and documentation
* Project can easily get off track if customer is not clear what final outcome that they want
* Not Suitable for large systems that require detailed documentation and

**Key Practices in Agile** *Practices*

Code is **extensible, low-defect code** with **simple** **robust design** that works for features currently implemented, well-factored & well-protected by unit tests

* Pair Programming
* Test Driven Development
* Refactoring ( rigorous, regular)
* **Code** **Base** **Shared**– single standard
* **Integration continuous**
* **Design** Simple

**Key Processes In Agile** *Processes*

* XP - **Extreme Programming**
* **Scrum**
* Sprint **Review** and **Retrospective**
* **Test Driven Development**
* Adaptive Development – Changing Requirements

**Pair Programming Advantages**

* **Communication skills, Ideas** are plentiful;, &

**Learn from** different perspectives

**Pair Programming Disadvantages**

* **Conflict** and debate break out among the group
* One person **overloaded** with project work
* Difficult to **collaborate** differently ideas

**Sprint Review** *Involves the Client*

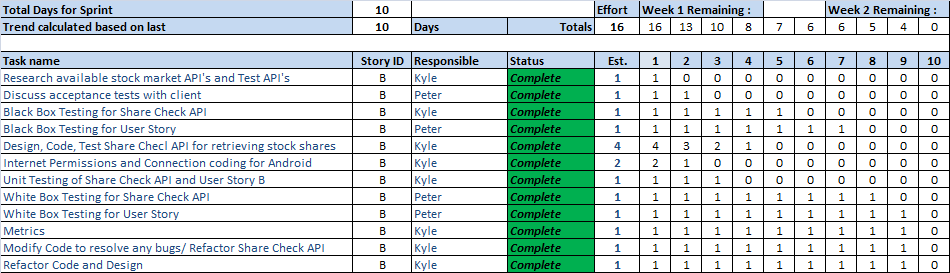
* Review completed / uncompleted work
* Present the completed work to the stakeholders
* Sprint Backlog

**Sprint Retrospective** *Scrum Team*

* **All team members** reflect on the past sprint
* Make **continuous process improvements**
* *What went well during the sprint?*
* *What could be improved in the next sprint?*
* **Product Backlog**

**Sprint backlog**  *Identifying main features*

* Task **Name**, Person **Responsible**, and **Status**
* **Effort** of task and **Time Remaining**
* **Burn down chart** – estimated velocity



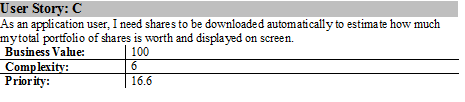
**Customer Role in Product Backlog**

* Decide **business value** for each story before each sprint & Ultimately decide which to implemented

**Conventional management vs. Agile Management**

* Deliverables pass through **a chain of approvals**
* **Formally document** each phase of the life cycle.
* **Traditional management** processes and Procedures
* Application **users are NOT always accessible** to you.
* The project team cannot **adapt to changes** in working durations, overlaps & shifts.

**User Story**

****

* **Business Value / Complexity = Priority**

*Reverse Side: Acceptance Tests & notes e.g. any key terms.*

**Three C’s of User Stories**

* Card - used for planning and prioritization.
* Conversation - actual requirement communicated.
* Confirmation - is the Customer Tests

*It allows us to confirm that we have implemented the requirement properly. The tests provide the detail of the story and much of the detailed documentation of the project.*