Agile methodology features:

1. For frequent releases for small features.
2. Testing activity is not separate phase as it is parallel to development.
3. Approach towards test cases: Only high level scenarios but without steps.
4. Participate in daily sprint planning, stand-ups,
5. To provide fast feedback.
6. Components of Scrum team:

Product owner: Maintains product backlog

Scram master: Maintains and manage sprint

Scram team: Developers, designers and QA

1. Sprint: It is a iteration
2. Sprint length: Depends, Min one week and maximum one month.
3. Product owner prioritize the backlog items but Daily meeting by team to prioritize testing during scrum and spring meeting.
4. Epic: A user defined feature:
5. Story: Epic is divided into Stories:
6. Task: Further breakdown of stories:
7. Product backlog: It contains a list of all desired features and is owned by the product owner
8. Sprint backlog: It is a subset of the product backlog owned by development team and commits to deliver it in a sprint. It is created in Sprint Planning Meeting
9. Burn up Chart: It shows the progress of stories done over time
10. Burn down Chart: It shows how much work was left to do overtime

**Agile practice in my project:**

**Stages:**

1. **Initial inception or Planning Meeting** (requirement identification, team building, stakeholder occurrence on product etc. etc.) meeting with the product owner and development team to **define the sprint** (or time boxes or iteration). E.g. dcb . **Everyone in the team involves.**

**Outcomes**:

* 1. Defining Sprint
  2. Duration of Sprint.
  3. Resources identification etc.

1. During the initial NEW sprint planning or set up, we QA team prepares the test environment such as servers building for dcb.

1. **Daily stand up call** with scrum team to prioritize and pick up the most importance features/requirements to be delivered in current sprint and then removing it from product backlog and then picking up the next work items. Daily meeting is also conducted to track the in progress items.

**Activities:**

* Prioritizing and selection of prioritize items.
* Discuss progress
* Task assignment
* Brain storming sessions
* Removing the done items.

1. Development takes 2 week for 1 sprint to develop a logical product (e.g dcb) and handed over to QA team as a part of default build and then we used to run the test cases around on this sprint along with the automated regression test cases on sprint/build so ensure that nothing is broken.

Some set of manual performance tests are also conducted to verify the resources usage such as memory leakage, site responsiveness (example: web viewer connection on multiple systems).

High level activities which QA team covers in a particular spring:

1. Functional testing
2. Test cycle execution: (Mostly dedicated resources to each cycle execution).
3. Daily Regression testing using automation tools
4. Performance testing (manual).
5. Bug fixes testing
6. Story testing
7. Execution of high level tests.
8. Various Build status
9. Final system testing
10. Failover and recovery testing
11. Conformance testing (such as various OS testing).
12. Static documentation testing.
13. Demo to customers.
14. After clearing from our QA bucket, **a RC** is declared and **this RC** build goes through the user acceptance tests. After clearing the acceptance test, this build has been initiated to transition phase and is given to our few selected customers for beta testing by deploying at their production servers. During the transition, we used to support users by giving them trainings After successful beta testing, a GA is declared along with all final documentation such as help files, user manuals etc.

How do you estimates the QA efforts:

It is twofold:

1. **For Fixed test items/cycles:**

For some test cycles (**which usually remains fixed all across current or previous iterations**) such as regression, failover and back up procedures, system or conformance testing, performance testing On the basis of previous iteration velocity: (e.g. we extract the time for the test cases design and execution of same test cycles executed in pervious iteration).

1. **For stories:**

**Depends:**

**2.1 Complexity**

**2.2 Associated use cases**

**2.3 Type of testing**

**2.4 Priority**

**Challenges in testing in agile environment?**

1. **Frequent builds:** Continuous integration of code by constantly changing of code by multiple check in a day or **code refactoring.**
2. **In adequate test coverage:** Due to frequent change in requirements.
3. **Inaccurate estimation:** due to frequent change in requirements.
4. **Distributed teams.**
5. **Have to do more and more regression testing** (both automation and manual: manual ex: spy objects which cannot be covered in automation) due to great risk of regressing of defects.
6. **Less documentations** as testers have to rely on face to face communication.
7. **Less duration of meeting** so cannot get the complete picture of requirement from dev or BA. Has to go through the JIRA to read the ambiguous stories (such as dcb stories).
8. **Strict deadlines or over commitment:** Can be compromised with quality.
9. **Developers has much say.**
10. **Addition of new stories at the end of cycle.**

To overcome above challenges “Risk based testing is the best approach along with automation”.

**Sprint:** It is a strict time based activity in which iteration of workable software has to be delivered at any cost. E.g dcb back end and next target is dcb GUI.

**Spring End:**

At the end of the sprint, test completion report is presented to product management team which includes the following activities and information.

1. **Lessons learned**
2. **Total number of test cases (passed, failed, blocked)**
3. **Environments tested**
4. **Components tested**
5. **Blockers.**
6. **Deferred bugs**
7. **Metrics**

**7.1** Test case design productivity metric:

**7.2** Test Effectiveness %

**7.3 Test Execution Coverage**

**7.4 Test Execution productivity**

**7.5 Error Discovery Rate or it is also called defect density ratio**

**7.6 Defect Removal Efficiency %**