**Stories approach**

**Definition of Done (delivered):**

* Implementation complete (acceptance criteria met)
* Unit/component integration tests complete and passing (green)
* Build passed on CI
* Code adheres to agreed coding standards – SonarQube with no show stoppers, critical, major issues
* Code review passed
* Merge into main line
* Build deployed into test environment
* Test cases captured in Test management tool
* Happy and unhappy path test cases have been peer reviewed, reviewed by BA, run and have passed in test environment
* End to end automation tests are updated and pass (green)
* Story tested on AWS. If not applicable comment needs to be added to the story with explanation
* Test artifacts and evidence are stored
* Story demoed and approved by the item owner or nominated deputy
* Story moved to Done (in Jira)

**Definition of Ready:**

* Story format is: (As a <user> I want to <feature> so that <benefit>)
* Acceptance criteria are clearly defined.
* User story is Independent, Negotiable, Valuable, Estimable, Small, Testable
* Following aspects of the story are identified and documented: Users, Actions, Data.
* Constraints and dependencies to other stories are identified and those stories are Done.
* Technical design is Done. It has been discussed and agreed how it will be delivered.
* Estimations for an user story should be lower or equal to 13 SP
* Story has been discussed at an 4Amigos session.
* Relevant story artifacts are provided: Business Rules, UI.

**Sprint approach**

1. The Sprint has a duration of 2 weeks
2. Business Stories can only be raised by the PO & BAs (or their proxies)
3. Technical Stories can be raised by the team, per approval with the BAs
4. Every Sprint deliverable will be signed off by the PO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Delivery Teams (Dev/Testing) | Daily Scrum / Standup | -previous day's achievements  -current day's plan  -impediments, blockers, dependencies | Daily | 15min |
| BAs  Design Leads  Devs  Testers  SMs/PM | 4-Amigos | -technical and business walk-through of user stories for Sprint  -refinement of estimations  -high-level tasks breakdown | 3 days / week | 1hr |
| PO  BAs  Design Leads  Devs  Testers  PM/SMs | Sprint Planning | -distribution of user stories per squads  -identification of potential dependencies and blockers  -reconciliation of backlog capacity with potential squads velocity | every 2 weeks | 1hr |
| PO  BAs  Devs  Testers  PM/SMs | Demo | -Real-time walk-through user stories delivered within a Sprint  -Formal process preceding the Sprint sign-off by the PO | every 2 weeks | Flexible (~30min per user story) |
| BAs  Design Leads  Devs  Testers  PM/SMs | Sprint Retrospective | usually one retro session per squad and another Big Retro with the Leads of all squads  Main outcomes on:  - What to "keep"  - What to "drop"  -Identify actions and owners | every 2 weeks |  |

**Testing approach**

1.**TC Creation**from the Acceptance Criteria of User Stories in JIRA & Requirements in Confluence

2.**Test Script Development**

* Writing in BDD fashion
* Coding in Java, Selenium Web Driver, Rest Assured, JPA
* Running with Cucumber
* Depositing them in GitHub

3.**Test Scripts**are **Integrated**in the Automatic Build – Jenkins

4.**Test Execution**at Deploy or on Demand in – Jenkins + Automation Test Framework

5.**Any found issues**will be raised in – Jira

6.**Reporting & Traceability**– Dashboards of Jira

**For User Stories and Functional Defects**  
  
**1.** Test on**feature branch** towards **master** – add a relevant comment in Jira once done (including component versions).

**2.** Test on **master** after the merge – collect the relevant test evidence, attach it to the defect/story and add a comment including component versions(AWS parameters) and link to the evidence.  
  
Naming convention: TestEvidence\_StoryID.zip.   
Note! The evidence should be archived in one zip file

**3.** Test on **release feature branch** – add a relevant comment in Jira once done (including component versions).

**4.** Testing on **release** (after the merge)

The order of the above actions may vary depending on priority (applicable for defects, depends on where the fix should be delivered first).

|  |  |
| --- | --- |
| **Test Activities (mandatory sub-tasks are marked with \*)** | |
| **Test (Analysis and) Design\*** | Internal/BA review, Zephyr tests and Cucumber features as artefacts TC specification. |
| **Exploratory Testing** | Performed on AWS environment, on feature branches |
| **Test Automation\*** | Technical enablers implemented and Cucumber features (already merged) ready to be run automatically. [Functional] Test Data - defined during the test cases design for each User Story |
| **Test Execution\*** | Performed on AWS environment, on trunk. Recommended naming convention: **TestEvidence\_StoryID-XXXX.zip.** |
| **Regression\*** | Targeted or full pack execution by running corresponding Jenkins Job |

**Raising defects**

**How I raise a defect:**

* find a bug
* discuss with the development team (and BA if required) and decide if the bug needs to be raised for tracking purposes
* if there is no need to raise it, wait for the fix from dev team
* I raise it

After I raised it,

* 1. I add all required details
  2. I assign to the developer. If the fix will come not in this sprint, I assign to BA or design lead (if technical)
  3. I monitor the defect I raised (check the status, assignment, priority)
  4. I re-test it
  5. I close it

**Statuses**:

* **Backlog** - the initial status for the created defect
* **Rejected** - this is not a defect, this is a duplicate. A comment should be added with the statement "why this item is rejected". The reporter should agree or disagree with the new status by adding a comment
* **Ready for dev** - the defect was triaged and is ready to be taken by development team
* **In Dev**, **Dev Complete** - means that the fix is in progress. this status is managed by developers
* **Ready for test** - fix is in place, it is possible to re-test and the item is assigned to the reporter
* **In test** - Defect is re-tested.
* **Test complete** - Evidence is attached, comment added.
* **Done** - no more work needs to be done.