QA Test Plan

QA Process:

The purpose of this document is to describe all the processes, tools and changes that we would like to put in place to bring some consistency and accountability to the Footsteps2Brilliance software development process, and to apply principles of Software Quality Assurance at the same time. By doing so, we would have solidified regular builds/delivery of new versions, while also minimizing risk of new issues or regression defects by first having QA'd new builds first. It is also the responsibility of QA during this process to provide feedback as end users to ensure not only quality, but user friendliness as well. The below processes will help ensure these goals are met going forward.

2 WEEK SPRINTS

2 week sprints are the most common for small to medium sized organizations wanting to adopt Agile software development methodologies and after reviewing what is currently in place, that is something we will try to implement. Due to specific release schedule of the apps/builds, sprint end will not necessarily mean release, but it will introduce stability and scheduling. Each sprint will bring new features, bug fixes, and increased performance over the last. Specifically, the first week would be dedicated to development while the second to QA. The sprint is kicked off on a Monday (Sprint Planning meeting) and is concluded the following week Friday (Sprint Retrospective meeting), once the new build is signed off by QA. Also, due to specific development flows (backend and frontend, web vs mobile apps, large test matrix), some adjustments might be necessary, but for now we will try with this standardized approach.

TICKET SCREENING/CYCLE

Current ticket tool is KanbanFlow which is failing to deliver all functionalities of a proper issue tracking tool. There is no proper issue prioritization, no backlog, no proper issue cycle. JIRA has already been chosen as a new tool. JIRA will be properly set up with all the required fields, a backlog, current sprint, etc to allow for a smooth transition to true Agile software development. Ticket screening, their reporting, and processes along the way are very structured, to maximize efficiency. Specifically, as mentioned above, a new sprint is kicked off on a Monday by reviewing all the tickets in the backlog (new and old), analyzing the highest priority items, and assigned them to a developer in the current sprint. Sprints have a 1 week DEV allotment and 1 week QA allotment. If there is remaining capacity in the sprint even after all the higher priority new tickets have been assigned, the backlog grooming process continues and older/lower priority issues can be assigned to use up approximately 90% resources immediately. Why 10% should be reserved for “other” in sprint tasks is explained below.

While tickets are in the backlog, they should be assigned to a F2B lead for review and in “New” status. They should stay in that status until assigned to a specific developer, until they begin working on the ticket, updating its status to “In Progress” accordingly. Each developer should only have 1 “In Progress” ticket in their queue at 1 time, so it is easily visible what they are currently working on. When they believe they have fixed the bug or implemented the new feature according to the specifications, they need to assign the ticket to QA for verification. At this stage, the ticket status should be updated to “Test” with a build number the changes are available in being specified by the developer in a comment. The ticket stays assigned to DEV until a build is available for QA to test that ticket on, at which point they verify the fix/new feature and close the ticket (“Fixed”). If they are still seeing the issue or they don’t believe the new feature was implemented exactly per the specifications, the ticket repeats the cycle and is re-opened (“New”). Status names and the actual bug progress can be adjusted so it suits better to all parties involved. To illustrate this flow, below are a few example scenarios;

QA reports new issue -> backlog, new status -> new sprint begins -> issue is screened, assigned to developer -> developer later in the week begins work on ticket, in progress status -> ticket is completed and assigned to QA, test status -> QA verifies fix/new feature, ticket is closed

- or

QA reports new issue -> backlog, new status -> new sprint begins -> issue is screened, assigned to developer -> developer later in the week begins work on ticket, in progress status -> ticket is completed and assigned to QA, test status -> QA tests ticket but still an issue or not per specifications, new status -> QA assigns ticket directly back to the responsible developer in the current sprint with a description and/or screenshots of where they problem lies -> developer tries again to correct the issue, in progress status developer locates and corrects issue, reassigns to QA for testing, test status -> QA verifies that this time around the issue is corrected, closes ticket.

Other variations are of course possible as well, but the above 2 are the most common. The 10% capacity you reserved earlier in the sprint is to allow for some regression issues or back and forth between the developers and QA, which is likely to take place (2nd example).

QA FOCUS

QA focus will be on Clever Kids app, as this will allow of introduction of new processes from scratch. QA will also provide support on other applications if needed - testing of fixed tickets, new app versions and similar, but we will try to introduce biggest changes through CK app. Through CK, we will try to dissect the app into manageable parts - modules (books, games, etc), frontend and backend. Each of these parts will receive full QA treatment - documentation, test cases, processes and similar, so it will easily transfer to other applications as well.

QA PRIORITIES

These are the things that should be addressed in near future - without any prioritization:

1. Setting up JIRA - involving QA in JIRA administration, maintenance and issue flow monitoring
2. Migrating issues from KanbanFlow - while the general idea is to start ‘from scratch’, some issues that are lingering should be transferred to JIRA
3. Adding more people to QA - due to the amount of work to be done, introducing more QA engineers makes sense and it will provide a very positive feedback on app quality.
4. Involving QA with devs - the earlier QA is involved with development process, earlier on bugs can be caught and fixed. Developers and QA engineers will be more comfortable to work with somebody they know and trust, making the whole testing process smoother.
5. Involving QA with current QA team - this will bring mutual benefit. QA engineers can bring QA specific knowledge to the table, while current QA team can bring app specific knowledge
6. Developing test matrix - creating a matrix of all platforms vs all app versions. Identifying the most important platforms/versions so we can bring more effort into testing what is really important and will have biggest impact overall
7. Introducing “release notes” as a way of easily tracking changes between builds - all the changes (code commits) between 2 builds (versions) should be documented, so QA can see immediately what the changes are and make sure that those are addressed first.
8. Developing test plan for testing of various modules - after identifying the major parts of the app, each of those should get its own test cases, prioritization, testing plan and similar
9. Investigating automated testing opportunities - after working on the app for some time, we will identify what can be automated, so we can receive better, quicker and more unified feedback for every build made.