Matcher development process summary

Matcher software version: v5.2.0

Resources: human

- We employ an in-house team of programmers, co-ordinated by our Product Manager
- We employ an in-house team of testers, co-ordinated by our Test Manager
- Occasional use of external consultant programmers where specific skills or experience are required to complement our internal resources

Resources: systems

- Microsoft Azure DevOps
- Microsoft Planner
- Other Microsoft Office 365 applications

User requirements

- User requirements derive from a variety of sources, including our internal product experts, consultant product experts, distributors, end users, prospects, third parties (e.g. EMR suppliers, fertility industry competent authorities) and monitoring of competitor products
- We proactively seek input and feedback, from which we derive some user requirements, including from key account development forums, end user questionnaires, ‘show and tell’ sessions with key end users and prospects
- We also derive some user requirements form reactive input and feedback, e.g. support tickets, informal feedback, meetings with various parties at trade exhibitions
- The top-level roadmap lives in Microsoft Planner
- Details live in DevOps
- Prioritisation meetings are typically held every 2 weeks, including estimations for each user story, assigning user stories to releases, and breaking them down to tasks

Methodology

- We use an Agile methodology
- Major version releases are approximately every 3 years
- Minor version releases are approximately every 6 months

Sprints

- Sprints are typically 2 weeks long
- They start with a kick-off meeting to move tasks within DevOps from the release backlog into the sprint and allocate tasks to staff members
- Sprints end with a meeting to discuss the work covered and the work going into the next sprint
- Each sprint is followed-up with a ‘show and tell’ session to obtain initial feedback, explain new functionality to colleagues, and identify documentation that needs to be updated

Builds

- We use a version numbering system of *major.minor.patch.internal*
- At any one time there may be multiple current relevant version numbers
  - **Prod** is the version last released to end users.
  - **Internal** is the version last released internally.
  - **Development** is the version currently being developed.

  - **Beta/UAT** when a release is handed to a beta/UAT location that version will become the latest beta/UAT version

Testing: internal system/unit

- Each developer tests their own code before marking a task as completed in DevOps
- Unit testing of completed tasks by the testing team occurs within the sprint and is documented and tracked within DevOps
- If the task passes testing, the tester moves the task to 'closed'
- If testing identifies any issues or bugs with the task, the tester creates a bug. This bug is moved into the version sprint backlog to be allocated to a future sprint
- If a tester passes a task or bug, it is moved to 'closed' and assigned to the 'beta/UAT' sprint (iteration) for the release

Testing: internal beta

- Once all tasks and bugs to be worked on within the release have been closed, an internal beta build version is created
- Internal beta/UAT testing will be driven by the tasks and bugs on the UAT print/iteration
- The internal beta/UAT testers include a combination of the testing team, our product experts and consultant product experts
- Any new bugs/issues found that require fixing within the current version will be created and placed into a 'pre-release' sprint
- Any minor bugs/issues noted that are determined to be tolerable, i.e. of low impact to end users, will be added to the tasks to be worked on in the next version release, and any required work-around instructions documented and communicated to end users as part of the training and/or installation process.

Testing: external UAT

- Once all tasks to be worked on within the pre-release sprint have been closed, an external UAT build version is created
- Any new bugs/issues found that require fixing within the current version will be created and placed into a 'pre-release' sprint
- We use EmbryoTools (www.embryotools.com) as a team of consultant product experts to quality control, test and evaluate the version as end users
- Any new bugs/issues found that require fixing within the current version will be created and placed into a 'pre-release' sprint
- Any minor bugs/issues noted that are determined to be tolerable, i.e. of low impact to end users, will be added to the tasks to be worked on in the next version release, and any required work-around instructions documented and communicated to end users as part of the training and/or installation process
- Any potential new user requirements derived from EmbryoTools' consultancy feedback and recommendations will be added to the top-level roadmap in Microsoft Planner, and/or created as new user stories in the backlog

Sign-off

- A version is only signed-off for release when all testing has been completed satisfactorily, including the development of any bugs/issues that have been added to the pre-release sprint, and when all known bugs/issues and required work-around instructions have been documented