DevOps Service Delivery - Work Intake Process

The following is a proposed Service Delivery Model for SOM IT Devops

Information

The following is a proposed workflow for overall service delivery for devops - this proposal does not materially change the core devops process. It does modify the way we would intake, triage, and sequence and link the work together. We would also need some additional "pre-kanban" statuses.

1. **New Client Request** - If it's a bug, we should create a **bug** ticket, do a quick triage and get it straight to Kanban backlog. Otherwise it should come in as a request ticket to the Desk Project if the client does actively work their own Jira project (such as AASL). If the client has a active Jira Project they can enter their request as an Improvement issue type.

2. **Triage** - Request/Improvement is assigned to Business Analyst who will triage (assistance from managers as needed), and can move it to **Requirements Gathering** or work with manager to Bitbucket **Want Do**.

3. **Requirements (or Unprioritized)** - BA works with clients to build user stories or a requirements document. BA helps client form a plan around a set of priorities. Once the request has enough requirements to be actionable it moves to Ready for Sequencing.

4. **Sequencing** - BA meets with Developers to create an **EPIC**, and individual tasks assigned appropriately, and sequenced as actionable work. Client can prioritize at the EPIC level and monitor through dashboards.

5. **Backlog** - Backlog is the developers backlog. It should contain bugs, which the Managers should assign and triage, or it will contain ALL tasks in an EPIC that is starting.

6. **Development to Deployment sign-offs** - The client must sign off on the development at the following environment stages:
   a. **Multi-Dev** - When the developer is at a place where they want the client to review the work to ensure it meets their business need. This is usually isolated to the specific change being requested.
   b. **Test/UAT** - When the developer has merged the individual change with the system, the client is obligated to test the implementation thoroughly. This the final test and sign-off before the change is moved into production.
   c. **Production** - Once the change has been moved into production, the client should review the deployed functions and verify with the developer that the request can be closed.

7. **Change Tickets** - The BA will create change tickets in the Desk Project to reflect the deployments, and will close them with references to the tickets for development. This will give the Service Desk knowledge of the applications that have been updated when troubleshooting. This is a service management best practice and a requirement requested by Connie.

8. **Managing Epic Status** - The BA manages the EPIC by moving tasks individually from backlog **To Be Picked**, as sequenced. The managers will manage bugs and their internal projects by assigning them and moving them to be picked. The developers work out of To Be Picked.

9. **Dashboards** - Clients monitor their work through Dashboards. Managers review progress of their initiatives through Dashboards. Kanban focuses on the work blocked or nearing completion so the BA and managers are alert to Epics needing attention.

**PROS**

* Devops process does not materially change
* Devops team has fewer backlog and to be picked items, and are working from a curated list
* Clients have visibility to their requested work, and visibility it takes for IT to complete a given request
* Clients can prioritize at the epic level if work backs up
* Clients must participate in requirements definition
* Managers have a view of defects vs new requests via epics
* Managers can balance the defect vs internal project work by selectively moving work to be picked
* Creates an appropriate service delivery model with defined roles of client, analyst, developers and managers
* The documentation the BA creates helps support the Service Catalog
* By tying bugs to the Application catalog, devops has incentive to keep it current
* Creating change tickets creates visibility to the work for the Service Desk, and is a best practice.

**CONS**

* Client may perceive Triage as a roadblock or additional JIRA stages as delays - in reality it means their items do not sit in backlog for as long
* Initially work is front-loaded with BA and Client; we would need a transition to this model
* Change - no one likes change
* The only programmatic changes would be to the Request issue type in the desk project. And the improvement issue type in client JIRA projects. We could also propose a new ticket issue type for all projects to reflect "service improvement". Dell, Matthew would need to scope that component

---

High Level Workflow
Reference


Help us improve

Your Rating: Results: 15 rates

Related articles

- New York Times journal information
- CylanceProtect "Installation Token" requested
- TurningPoint Polling Solution
- Logitech for Breakout Rooms
- Instructions for Bloomberg remote access