Issue tracking

Jana Kostičová

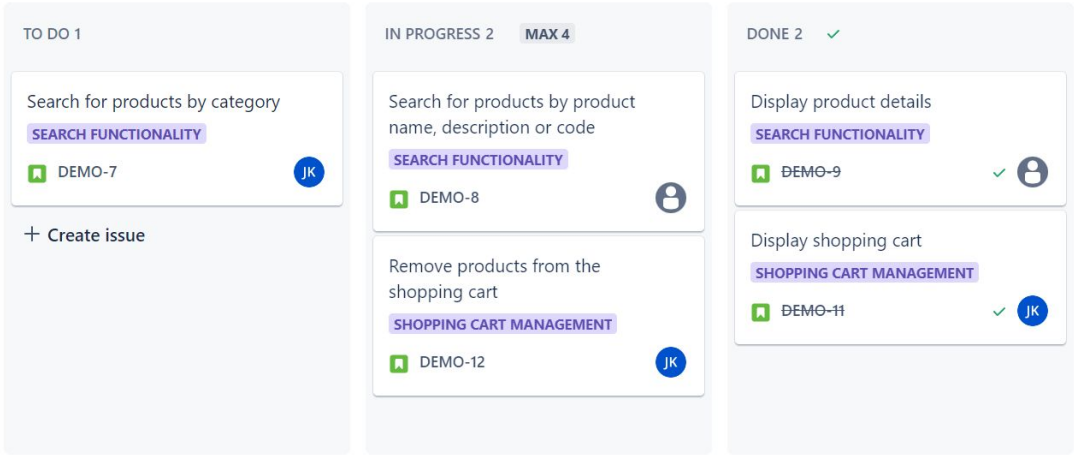
Issue

= any work item that needs to be documented, tracked and resolved

- Software bugs
- Feature requests
- Tasks
- Internal IT problems
- Requirements
- Stories
- Epics
- …

issue types

My issues:



Information about (my) issues scattered across different resources

Issue tracking system (ITS) = a centralized system for managing all aspects of (software development) issues

Issue attributes

Content

- ID
- Reporter
- Title, Description
- Product, Component
- Type (feature request, bug, task, ..)

Planning-related

- Status
- Assignee (individual or group)
- Priority / severity
- Release
- Start date, due date
- Estimates

Relationships to other issues

- Parent / child (hierarchy)
- Depends on
- Blocks
- Relates to

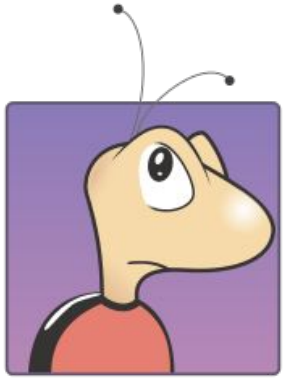
Other

- Time spent
- Links to VCS commits / pull requests
- Comments
- … many more

Issue Tracking Systems - features

- Issue types and issue attributes
- Issue relationships
- Issue lifecycle (workflow)
- Reporting (both current status and history)
- Integration (especially with Git)

Bugzilla



- First released in 1998
- Open-source, developed by Mozilla
- Primarily focused on bug tracking
- Originally on-premise solution, now several service providers offer cloud-based deployment
- Examples: https://bugzilla.mozilla.org/, https://gcc.gnu.org/bugzilla/

Features

- Issue types: Everything is a bug, custom bug types can be defined (Bug, Defect, Enhancement, Task, Change Request)
- Issue relationships: Only “Depends on” and “Blocks” relationships, no hierarchy
- Issue lifecycle: Default: *New -> (Unconfirmed) -> Assigned -> Resolved -> (Verified) -> Closed -> (Reopened)*
    - Customizable - new statuses can be introduced, transitions can be modified, different workflows for different project categories can be set, etc.
- Reporting: Basic tabular and graphical summary reports, customizable
- Integration: No built-in integration with VCS - possible through 3rd party tools (e.g. GitZilla) or manual linking

Usage & limitations

- Tracking and managing software defects and / or enhancements
- Rich customization options
- More suitable for maintenance activities compared to developing entirely new software
- Not suitable for project management and agile workflows
- Limited reporting (comparing to more modern ITS)

Jira

- First released in 2002
- Commercial software, developed by Atlassian (free plan / up to 10 users)
- Originally issue tracking tool for developers
- Later extended with (agile) project management features
- On-premise / cloud
- Example: https://issues.apache.org/jira/projects/CASSANDRA/,

Features

- Issue types: Epic, Story, Task, Bug, Subtask; customizable
- Issue relationship:
  - Issue hierarchy (parent/child): Epic → {Story, Task, Bug} → Subtask; only partially customizable
  - Issue links: Relates to, Blocks / Is blocked by, Clones / Is cloned by. Duplicates / Is duplicated by; customizable
- Issue lifecycle: *To Do -> In Progress -> Done;* customizable
- Reporting: Rich view and reporting options, drill down based on issue hierarchy
- Integration: Built-in integration with Bitbucket & other tools in Atlassian ecosystem, integration with external VCS (e.g., GitHub) through plugins
- Atlassian Marketplace released in 2012 (3rd party plugins): marketplace.atlassian.com, Atlassian Community: community.atlassian.com, support channels

Jira (cont.)

Usage

- Tracks issues from different viewpoints: development teams, project managers, product managers,...
- Rich support for agile development
- Robust reporting
- Suitable both for new software development and maintenance
- Scales well to large projects

Limitations

- Steeper learning curve
- Costs must be taken into account (also for
- Default issue hierarchy is only partially customizable - this can be limitation for some projects

GitHub Issues

- GitHub released in 2008
- Originally it provided only simple issue tracking capabilities
- Later few planning features were added, but it still remains very lightweight ITS
- Available for free plan users
- Example: https://github.com/facebook/react/issues

Features

- Issue types: No explicit issues types. Issue has a predefined field set that can be customized at the project level. It is possible to differentiate between issues by adding new field (e.g., “Issue type”) and define its values as necessary.
- Issue relationships: No hierarchy, no explicit relationships (only via links in comments)
- Issue lifecycle: Mandatory: Open → Closed + Custom lifecycle can be added (default: Todo → In Progress → Done)
- Reporting:
- Integration: Tight integration with GitHub repositories and workflows (it is mandatory to link issue to some repository)

Usage & limitations

- Great for small teams using Github
- Lightweight, easy to use and configure
- Offers basic functionality for agile workflows
- Insufficient if more complex project management features and reports are necessary
- Mostly not scalable for large projects

References

- https://www.bugzilla.org/
- https://www.atlassian.com/software/jira
- https://github.com/features/issues
- Wikipedia - Comparison of issue tracking systems