

<epam>

Data Quality

Introduction to Quality Assurance



TRAINING
CENTER

— <epam> —



Agenda

- Software Testing
- JIRA introduction

Why is Testing Necessary

UNCOVERS DEFECTS BEFORE IT IS DELIVERED TO CUSTOMER

UNTESTED SOFTWARE COSTS THE
CUSTOMER BUSINESS FAILURE TOO



PROVIDE FEEDBACK TO ALL
STAKEHOLDERS

AS LATER THE STAGE IN WHICH BUG IS IDENTIFIED,
MORE IS THE COST TO RECTIFY IT



PROVIDE FEEDBACK



REDUCE IMPLEMENTATION
COST

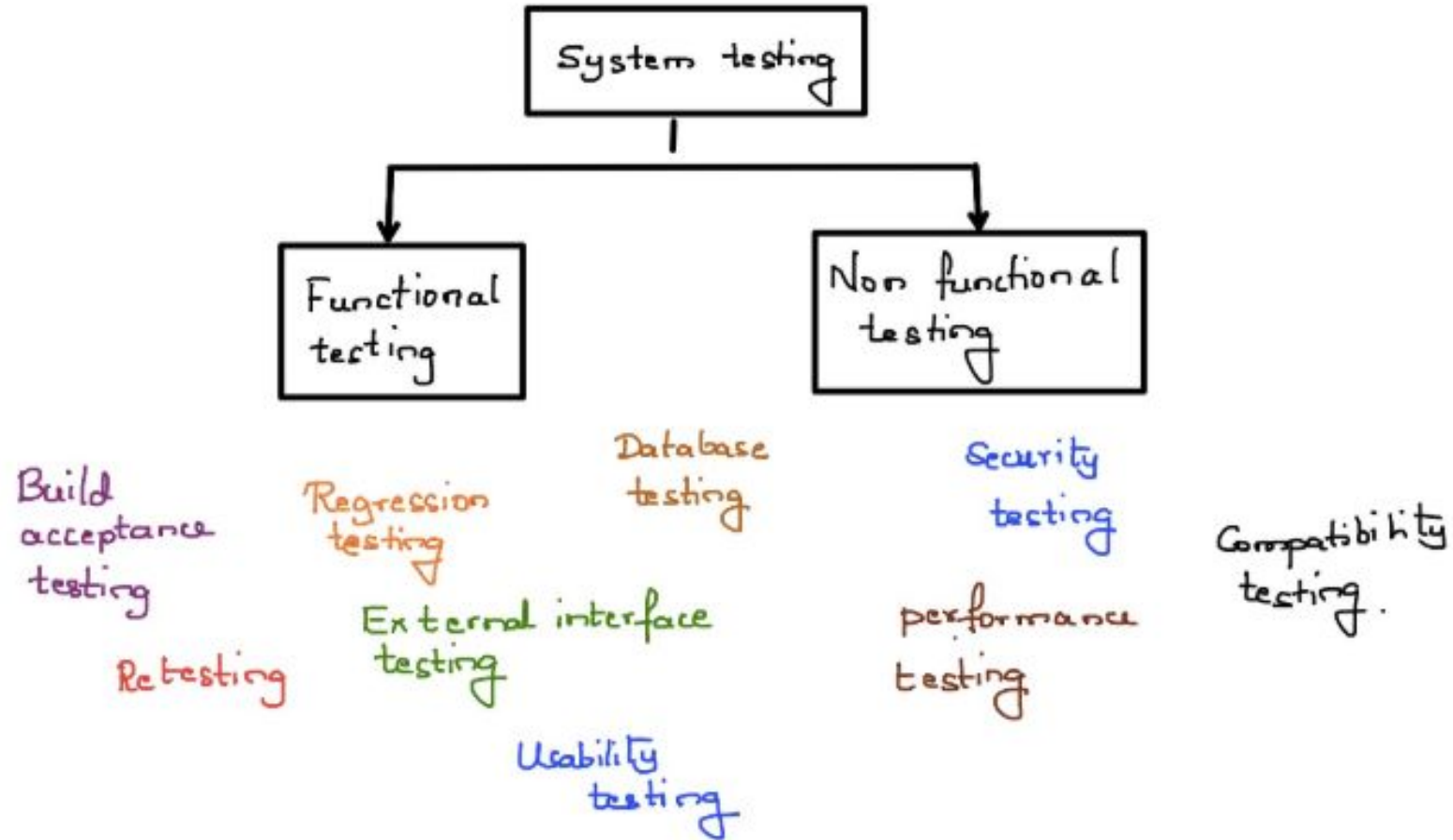


GAINING END USERS
CONFIDENCE



STAY IN
THE BUSINESS

Main Software Testing Types



Functional Testing

The functions are “**what**” the system should do.

Functional tests tend to answer the question of “**does this particular feature work**”.

Functional
testing



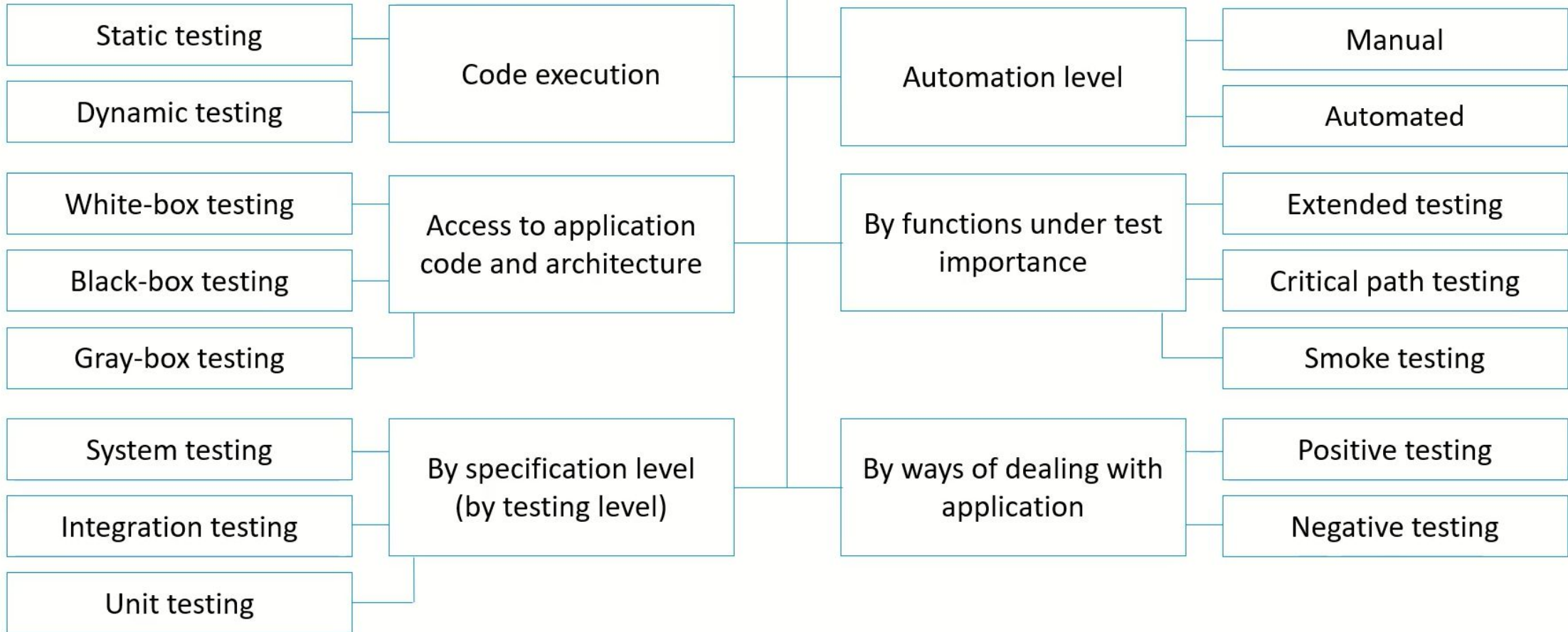
Non-Functional testing

Non-functional tests tend to answer the question of “**how well**” the system behaves.

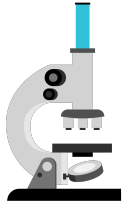


Functional Testing

Classification by...



Test types by testing depth



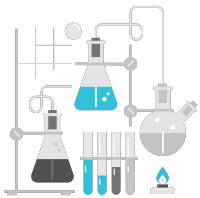
Smoke

a subset of all defined test cases that cover the main functionality of a component or system, the most crucial functions.



Critical path

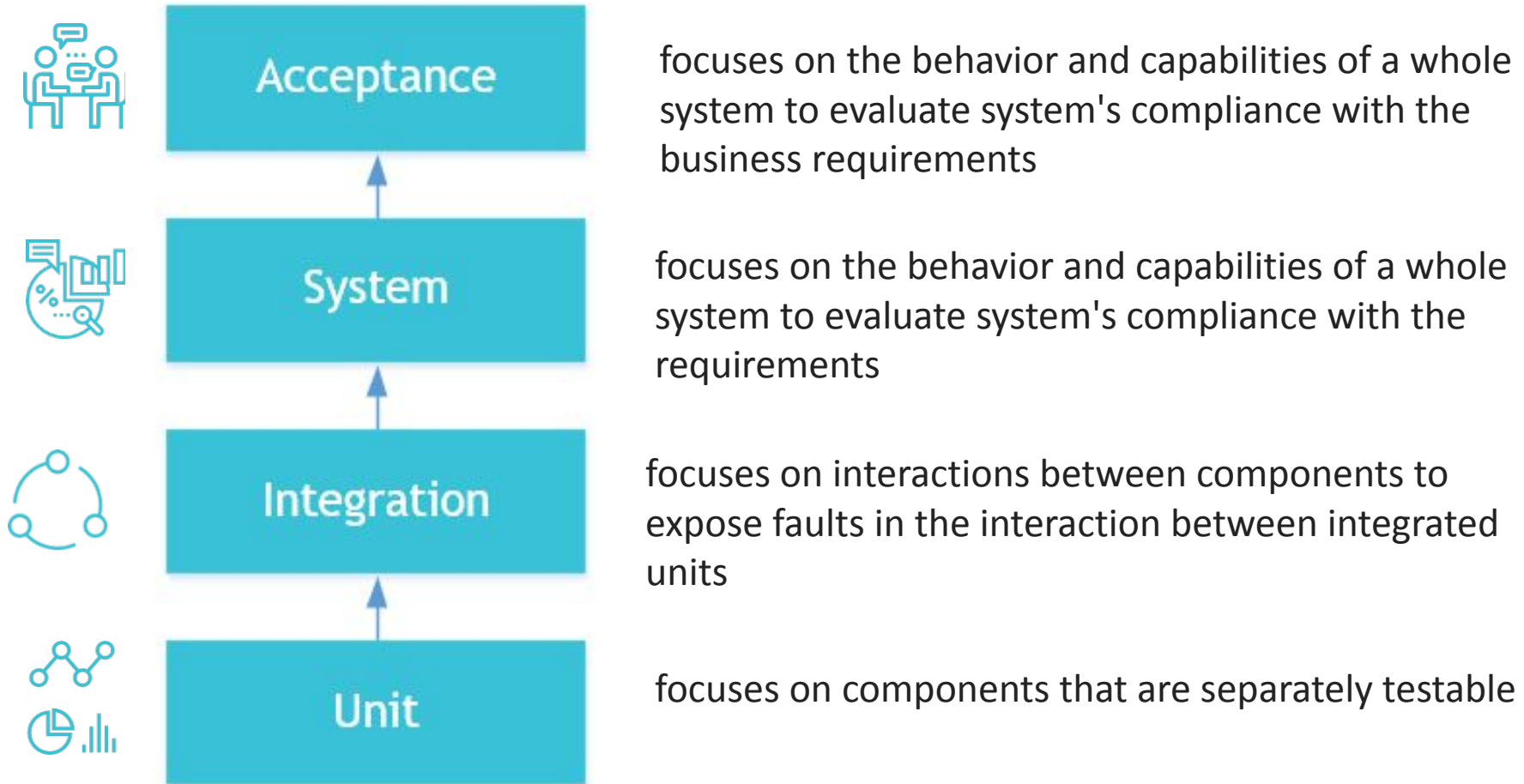
test cases that cover the functionality used most of the time by the majority of users.



Extended
path

test cases that cover the “nice-to-have” functionality (not used most of the time by the majority of

Software Testing Levels



Software Testing Levels



UNIT

- Who: DEV, QA
- When: Component is developed
- Why: To validate that each unit of the software performs as designed
- How: White-box testing



INTEGRATION

- Who: DEV, QA
- When: Units to be integrated are developed
- Why: To expose faults in the interaction between integrated units
- How: White, Black or Grey-box testing



SYSTEM

- Who: QA
- When: Separate units are integrated into System
- Why: To evaluate system's compliance with the specified requirements
- Black-box testing



ACCEPTANCE

- Who: Business users
- When: Component is developed or units are integrated into System
- Why: To evaluate system's compliance with the business requirements; assess whether it is acceptable for the delivery
- Black-box testing

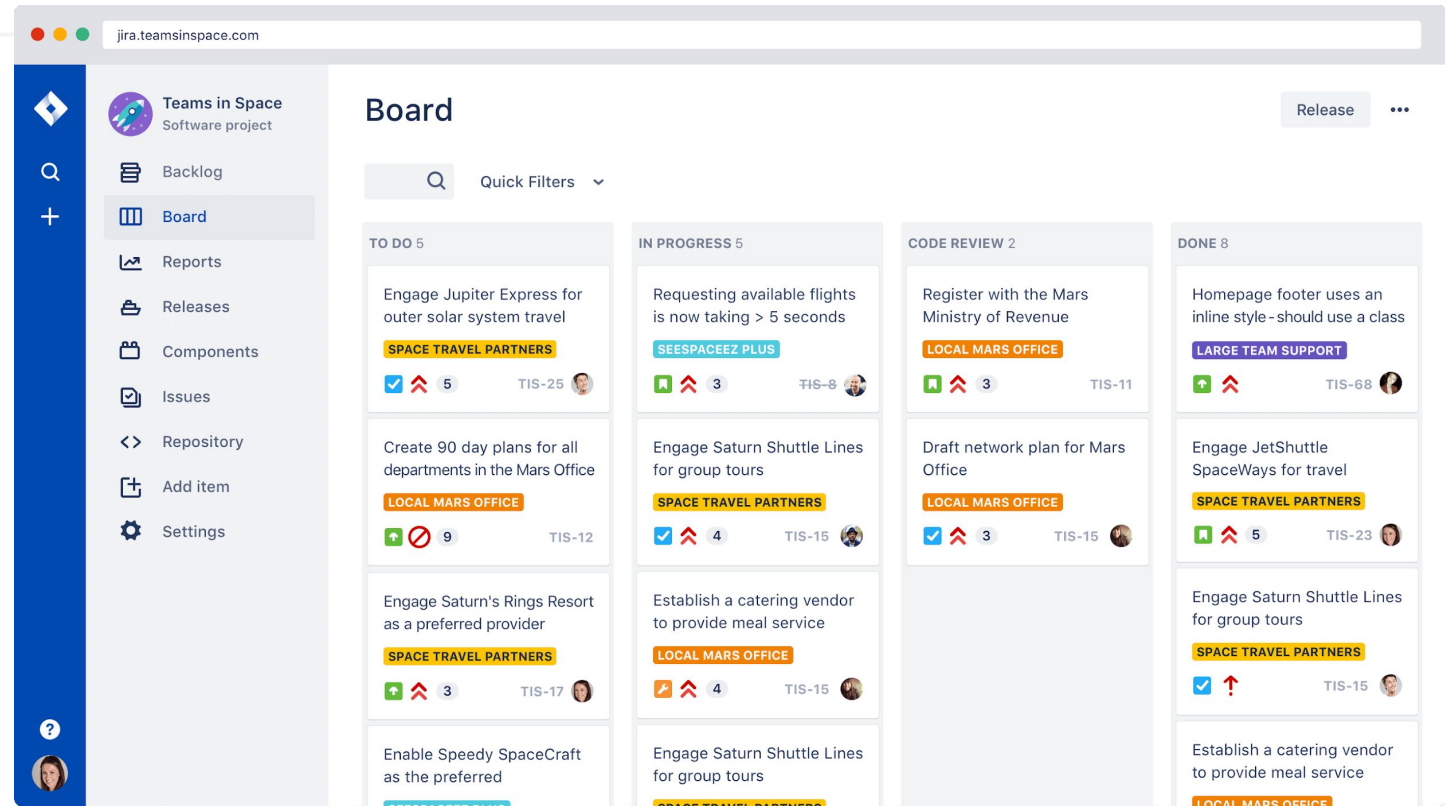
What is JIRA?

JIRA is ...

- ✓ a web Atlassian tool
- ✓ the tracker for team planning
- ✓ configurable and extendable tool

JIRA ...

- ✓ adapts to business processes
- ✓ maintains your historic data
- ✓ used to prioritize, assign, track, discuss, report and watch issues\tasks



JIRA advantages and disadvantages

ADVANTAGES

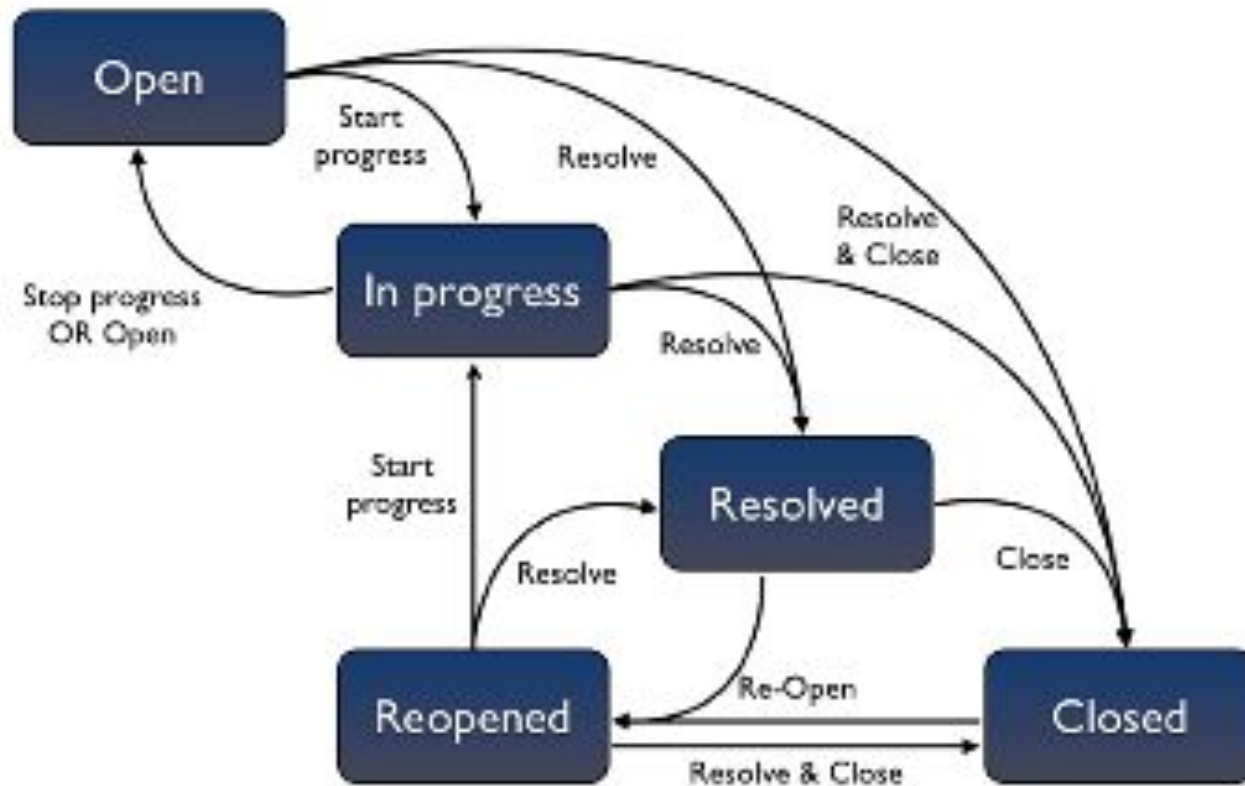
- Teamwork facilitation
- Full visibility of the supported processes for the team and process/unit stakeholders
- Planning
- Time-reporting
- Tasks prioritizing
- Clear KPIs for People Management

DISADVANTAGES

- JIRA Performance
- Lack of training on project management
- Time and efforts spent on «Getting acquainted stage»

Workflow in JIRA?

A JIRA workflow is the set of statuses and transitions that an issue goes through during its lifecycle.



What is an issue?

Issue – any task that requires an action from a person.

Issue types:



Bug — a problem which impairs or prevents the functions of the product.



Task — a task that needs to be done.



Sub-task - are useful for splitting up a parent issue into a number of smaller tasks that can be assigned and tracked separately.



Story – in JIRA Agile, a story is represented as an issue, and individual tasks within the story are represented as sub-task.



Epic - An epic captures a large body of work. It is essentially a large user story that can be broken down into a number of smaller stories. It may take several sprints to complete an epic.

Standard actions on issue

- Create
- Edit
- Assign
- Comment
- Log work
- Attach
- Link
- Clone
- Delete

The screenshot shows the JIRA interface for an issue titled 'TEST' in the 'EPM-HRM' project. The issue ID is 'EPMHRM-2117'. The 'More' dropdown menu is open, displaying a list of actions. Several actions are highlighted with red boxes: 'Edit', 'Comment', 'Assign', 'Log Work' (at the top of the dropdown), 'Log Work' (in the middle of the dropdown), 'Attach Files', 'Move', 'Link', and 'Clone'. Other actions visible in the dropdown include 'Mind Map', 'Plan Time', 'Start Tracker', 'View Worklogs', 'Agile Board', 'Rank to Top', 'Rank to Bottom', 'Attach Screenshot', 'Add Vote', 'Voters', 'Stop Watching', 'Watchers', 'Create Sub-Task', 'Convert to Sub-Task', 'Labels', and 'Delete'. The main issue details are partially visible, showing fields for Type, Priority, Affects Version/s, Component/s, and Labels. The 'Tempo' section shows a date range of '01/Jul/15 - 31/Jul'. The 'Structure' is set to 'Global'. The 'Activity' section shows 'All Comments' and a message 'There are no comments yet'. The 'Log Work' button is also visible in the bottom right of the issue details area.

Creating an issue

The screenshot shows the 'Create Issue' form in JIRA. The form is titled 'Create Issue' and has a 'Configure Fields' button in the top right corner. The fields are as follows:

- Project***: A dropdown menu with a blue icon on the left.
- Issue Type***: A dropdown menu with 'Task' selected and a question mark icon.
- Summary***: A text input field.
- Priority**: A dropdown menu with 'Trivial' selected and a question mark icon.
- Component/s***: A dropdown menu with a question mark icon and a hint: 'Start typing to get a list of possible matches or press down to select.'
- Assignee**: A dropdown menu with 'Automatic' selected and a question mark icon. Below it is the text 'Assign to me'.
- Environment**: A rich text editor with a toolbar containing 'Style', 'B', 'I', 'U', 'A', 'A', 'Link', 'Image', 'List', 'List', 'Smiley', '+', and 'Up'. Below the editor is a 'Visual' tab and a 'Text' tab. A hint below reads: 'For example operating system, software platform and/or hardware specifications (include as appropriate for the issue).'
- Description**: Another rich text editor with the same toolbar as the Environment field.

At the bottom of the form, there is a checkbox for 'Create another', a blue 'Create' button, and a 'Cancel' button.

- 1. Project.** Every issue is associated with a project.
- 2. Issue Type.** This field allows you to select the kind of issue you want to create, such as a bug or task, etc.
- 3. Summary.** A summary should be a very brief summation of the issue. The summary acts as the title of the issue.
- 4. Priority.** You can set the priority of this issue. Setting the priority can help your team during the triage process.
- 5. Assignee.** Here you can choose whom to assign the issue to.
- 6. Environment.** OS/browser/product configuration.
- 7. Description.** Here's where you can describe the issue in as much detail as you'd like.
- 8. Create.** Once you click 'Create', your new issue will be added to JIRA.

Resolving an issue



1. Press **Resolve** button in the upper panel bar
2. Move the necessary Task from one column to another (Tracking board)


Resolve Issue

A screenshot of the 'Resolve Issue' dialog box. It features an information icon and the text: 'Resolving an issue indicates that the developers are satisfied the issue is finished.' Below this is a 'Resolution*' dropdown menu with 'Please select...' and a help icon. Underneath is a 'Fix Version/s' dropdown menu. At the bottom is a 'Comment' field with a rich text editor toolbar.


3. And don't forget about Issue Resolution. JIRA by default will mark an issue to be Unresolved when the resolution field is not having any value on it.
4. Adding comments to an issue is a useful way to record additional detail about an issue and collaborate with team members.

Log work on issue

 Log Time

 [Issue ID]

Description

User*  [User Name] .. ▾

Period

Date*

From To

Worked*

Remaining estimate

Log another

LOG WORK ON ISSUE

To log work, use the following format:

- **w** - weeks
- **d** - days
- **h** - hours
- **m** – minutes



Time tracking

Original Estimate (blue) — the amount of time originally anticipated to resolve the issue.

Remaining Estimate (orange) — the remaining amount of time currently anticipated to resolve the issue.

Time Spent (green) — the amount of time logged working on the issue so far.



People

Assignee:  Liudmila Glazunova
Reporter:  Liudmila Glazunova
Votes: 0
Watchers: [1 Stop watching this issue](#)

Dates

Created: Yesterday
Updated: Just now

Time Tracking

Estimated:  16h
Remaining:  14h
Logged:  2h

Collaborators

Liudmila Glazunova 2h / 0h

KANBAN board

Kanban is a catalyst for change through small, incremental improvements to your existing process – be it scrum or otherwise. Rooted in lean manufacturing, Kanban is a signaling system that can be effectively applied to software development, DevOps, IT operations, HR processes and many other processes.

The screenshot shows a Jira Kanban board titled "EPMH-MSTD People Management Tracking board". At the top, there are navigation tabs: "Backlog", "Kanban board", "Reports", "Board", and "View in Tempo". Below the tabs is a "QUICK FILTERS" section with various filters like "Recognition", "Training", "Survey", etc. The board is divided into four columns: "41 To Do", "29 In Progress", "173 Resolved", and "430 Done". A dropdown menu shows "This week 17 issues". The "To Do" column contains two issues: EPMHMSTD-505 (To improve of internal communication process) and EPMHMSTD-681 (MSTD_Adaptation talk-Yaraslau Shabanets1_2015-07-28). The "In Progress" column contains two issues: EPMHMSTD-4 ("Welcome to MSTD" presentation) and EPMHMSTD-507 (QPF & ASMT Assistance). The "Resolved" column contains one issue: EPMHMSTD-678 (MSTD-BA_Adaptation talk-Natallia Ivaka1_2015-07-21). The "Done" column is empty.

SCRUM board

Teams who use scrum plan work in *sprints*, short blocks of time in which the team can estimate and return value back to the customer.

Software development teams use scrum to deliver complex solutions in a structured, predictable way.

The product owner manages the *backlog*, ensuring the team is working on the highest-priority items first. With JIRA Agile, the team can organize stories in sprints, in larger epics, and across release versions.

Projects / DQETEST / SCRUM DQE

9 days remaining Complete sprint

TO DO

- Second story (DQET-2)
- Bug1 (DQET-3)

IN PROGRESS

- First issues (DQET-1)

DONE

Projects / DQETEST / SCRUM DQE

Backlog

Share

Backlog 4 issues

Create sprint

First issues	DQET-1	↑
Second story	DQET-2	↑
Bug1	DQET-3	↑
Bug2	DQET-4	↑

+ Create issue

<epam>

Q & A

