VB DP REQUEST FLOW GUIDE

MONTHLY PROCESS

BI-WEEKLY SCRUM PROCESS

2 week=1sprint

GATHERING REQUESTS

- creating tickets in the backlog
- filling the requirements for a task according to the template

TICKETS REFINEMENT

- dIscussing implementation details, estimating tickets

PLANNING

- discussing the results and challenges of the previous 2 week
- Planning for the next week

HARD WORK

- working with requests

ROLL-OUT

- Roll-out sessions about new dashboards for each regions and Product team

DEMO

- Each 3rd week - Demo of monthly results.

Gathering Requests – Getting Started

Step 0. Define the Need

Before creating a request, it is important to clearly define:

- the business need you want to address;
- the goal of the request;
- the expected outcome of the implementation.

This initial step helps the team understand the context and propose the most effective solution.

Next, we will walk through each field that needs to be completed in the request.

Step 1. Create a Request

To create a request, you can choose one of the following options:

- Request Form – the recommended option, designed to guide you through all required fields;
- Standard Jira Ticket Creation – if you prefer working directly on the Jira board.

Step 1. Create a Request

Jira Form

all links below

Feature Request Form VBET BIA - for new dashboard, report, functionality, link

Fix request Form VBET BIA - fixing current dashboard, report, functionality, return to work after implementation link

Bug report - the current functionality is not working as expected: data does not match, the visualization is incorrect, etc. , link

Jira Request Board

board link

Work type User Story Story

Work type Task Task

Work type Bug Bug

Step 1. Create a Request

Summary *

Add name of request

Business objective *

We support markdown! Try **bold**, `inline code`, or ``` for code blocks.

Briefly describe business value request implementation will bring.

Description *

Type /ai for Atlassian Intelligence or @ to mention and notify someone.

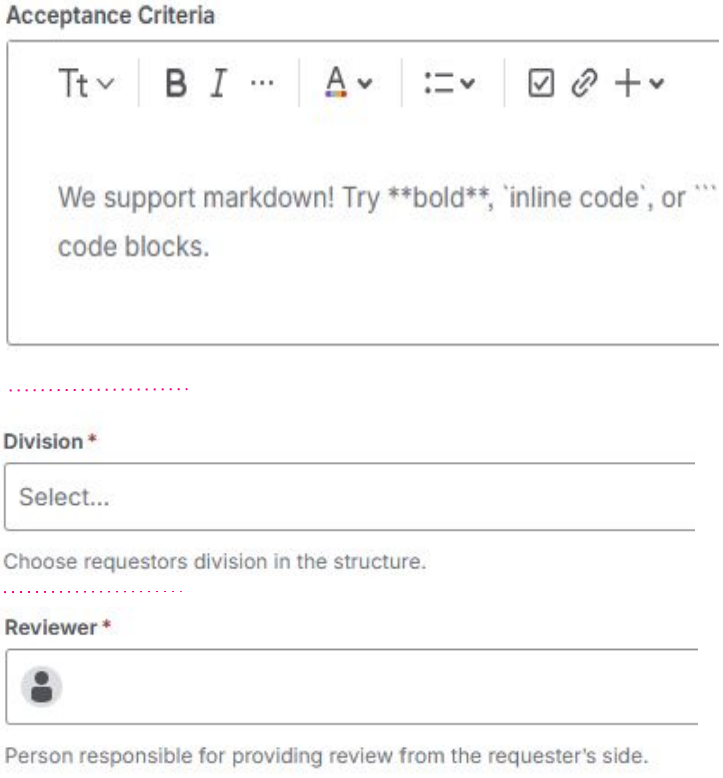
Add a short description of what needs to be done.

Add a clear and concise name for the request. The summary should **briefly describe** *what is needed* so the request is easy to identify in the backlog.

Briefly describe **why this request is important** and what business value its implementation will bring.
Focus on the expected impact (e.g. improved decision-making, risk reduction, revenue impact, time savings).

Provide a short description of **what needs to be done.**
Outline the scope of the request so the team clearly understands the expected work and boundaries.

Step 1. Create a Request



Expected Outcome

- A list of conditions or a checklist used to confirm that the request has been delivered exactly as expected.
- For dashboards: configuration, list of KPIs, data granularity, data refresh frequency, and available filters.

Select the business division or department submitting the request.
This helps identify the request owner and align priorities and communication.

Select the person responsible for reviewing and accepting the final result from the requester's side.
This person validates that the delivery meets the agreed expectations and Acceptance Criteria.

Step 1. Create a Request

Unit *

Department(s) that will use or benefit from the results of this development.

Priority

Select the priority based on the request's impact, business value, and urgency.

Please, add list the departments and business units that will use this dashboard or benefit from the implemented request.

Select the priority based on business criticality, expected business value, and urgency

- **Critical** – blocks core business operations, revenue, compliance, or security. Immediate action required.
- **Major** – high business impact. Significantly affects business processes or decision-making.
- **Medium** – moderate impact. Improves processes or analytics but is not time-critical.
- **Minor** – low impact. Cosmetic changes or minor improvements with no effect on core business flows.

Step 1. Create a Request

Selecting the correct component is important, as it determines which team will take the request and which team you will communicate with during its delivery.

Components *

BI Reporting - create a new Dashboard or change logic for the existing one. Make ad-hoc analysis
Regulatory Reporting - tasks related to the generation and reporting to regulators at the regional level

Web-data - requests related to marketing and acquisition tracking, funnel analytics, event tracking, attribution, marketing dashboards, and data validation for web and app user journeys.

Streaming - requests related to real-time data processing, business monitoring, alerts, anomaly detection, and operational or financial incidents that require immediate reaction.

Core Data - only internal technical requests. You not need to create requests there.

Responsible team

REPORTING

WEB DATA

STREAMING

CORE DATA

Work with Requests

Step 2. Refinement

At this stage, you may receive clarification questions regarding the requirements you described. Active collaboration with the team is essential, as it helps reduce uncertainties and significantly shortens the overall delivery time.

***Jira board* for check updates**

Step 3. Planning

At this stage, the team plans the request for the upcoming two-week sprint, unless it is an ad-hoc request.

Work with Requests

Step 4. Rewiev

This stage marks the completion of the request, provided the delivered result meets the **expected outcome defined in the Acceptance Criteria.** The task is moved to the **Review** status.

At this stage, the **Reviewer** specified in the request is responsible for validating the result and moving the task to **Done**. The task can remain in the Review status for up to **7 business days**.

If the task is not moved to **Done** by the **Reviewer** within this period, it will be automatically closed by the team after the waiting period ends.

If additional changes or further development related to the request are required, a **new task** or **Fix Request** should be created in the backlog using the provided link.

If the implemented functionality does not work as expected, does not meet the defined Acceptance Criteria, or if defects are identified, a **bug** must be submitted using the **Bug Report** form.