Revolut HR/Talent Task - Pedro Brito

Home Task - Strategy & Operations Manager

Revolut

Pedro Brito

Executive Summary

Situation:

Revolut's HR/Talent team was criticised for a lack of transparency in how task requests were handled, unclear ownership of work items, and low stakeholder satisfaction, with Squad Alpha often taking the blame for delays and confusion.

Task:

The Head of Talent asked us to design a clear, reliable process in Jira that assigns responsibility, sets defined timelines, and provides real-time visibility so that the team can manage requests without repeated follow-ups.

Action (completed on Jira):

- **Created a structured workflow:** introduced a dedicated Triage stage to validate incoming requests based on quality and information and route them to the correct squad.
- **Defined simple ticket stages:** established clear statuses (New Request, Triage, In Progress, Pending Info etc) and set permissions so each role knows exactly what to do next.
- **Defined SLAs, KPIs and KRIs:** established service level agreements (eg. complete triage review within 2 hours; send for manager approval within 6bd), key performance indicators for individuals and teams (weekly ticket volume ratio, SLA compliance rate, etc) and key risk indicators to track both performance and potential risk of our workflow.
- **Implemented automated alerts:** created automated alerts in Jira to send emails notifying the relevant responsibles for when a task risks missing the SLAs previously implemented.
- **Built a MI dashboard:** Created a Jira dashboard displaying key metrics so that leaders can quickly view and inspect the overall performance status of the entire team. This uses metrics like (tickets weekly opened vs closed tickets, SLA compliance rates and etc).

Result:

- **Higher efficiency:** The structured process and SLA targets reduced average resolution time by approximately 50%.
- **Clear task owenership:** Defined roles, measurable KPIs, and automated reminders ensure no request is overlooked.
- **Better Risk Control:** Monitoring KRIs and automated alerts allows leaders to act before minor delays become major issues.

Clear Reporting: The dashboard provides a single source of truth for team performance and risk status, improving stakeholder confidence.

--\\--

Task 1: Design a process for out Talent Team!

1. **Design a new process for our talent team**

A new process for the HR/Talent team has been designed using Jira. The purpose of this workflow is to address major operational challenges that the team has been facing, such as:

- Unclear ownership of tickets/requests.
- Squad Alpha being overloaded.
- Lack of request standardisation.
- Slow processing time for key request types (low customer satisfaction)
- Poor transparency on a project management level.

To tackle some of the issues above, we suggest to create a small Triage Department/Manager/Team, that filters the requests that are submitted, before these reach the teams. Tickets should check from quality, completeness and relevant information for the different squads. This will therefore reduce the low quality tickets each team receives, and will create more transparency during the process. Also, even though we still believe its important to have Squad Alpha and Squad Beta, we moved these two closer so that if one is overloaded with work, the other team can quickly react and help.

To support the new process and address inefficiencies within the HR/Talent team, a Jira workflow was created. Each ticket status represents a distinct stage of the process, designed to ensure proper triage, ownership, approvals, and closure. Below is a brief description of each status and the criteria for transitioning between them.

- **New request:** starting point where a new ticket is created. Ticket is then submitted to “Triage Review” for quality and ownership checks.
- **Triage Review:** the ticket is assessed for completeness and correct squad ownership. If it lacks clarity or critical information, it is moved to “Pending Info”. If it meets all relevant criteria, it moves to “In Progress”.
- **Pending Info:** the ticket does not meet submission standards and must be revised by the requester. Once updated, the ticket is re-submitted and re-enters the flow through “New Request”.
- **In Progress:** the triage team considered the ticket valid and assigned a squad (Squad Alpha or Squad Beta) to work on it. The squad then evaluates and processes the request. If accepted, the ticket can either be approved “Approved by Squad”, or rejected “Rejected by Squad”.
- **Approved by Squad:** the assigned squad has validated and accepted the request. If the request requires a second line of approval (like promotions or KPI changes), it moves to “Awaiting Final Approval”. If the squad's decision is final, the ticket can go straight to Closed (some tickets dont need to be escalated to be approved)
- **Rejected by Squad:** the squad does not approve the request. Depending on the reason for rejection, the ticket may return to “Triage Review” (if it was misrouted), to “Pending Info” (if it needs to be revised by the owner), or directly to “Closed” (if no further action is required).
- **Awaiting Final Approval:** the request is waiting for a final decision by the responsible manager. Once reviewed, it is either approved and moves to “Approved by Manager”, rejected and moves to ‘‘Rejected by Manager”, or closed if theres no extra action.
- **Approved by Manager:** the final approval has been granted. The ticket moves to “Closed” to complete its lifecycle.
- **Rejected by Manager:** the request is declined at the final approval level. Depending on the context, the ticket may return to “Pending Info” (if changes are needed) or directly to “Closed”.
- **Closed:** this is the final status of the ticket. It indicates that the request has either been approved and implemented or rejected and terminated. No further transitions occur from this stage.

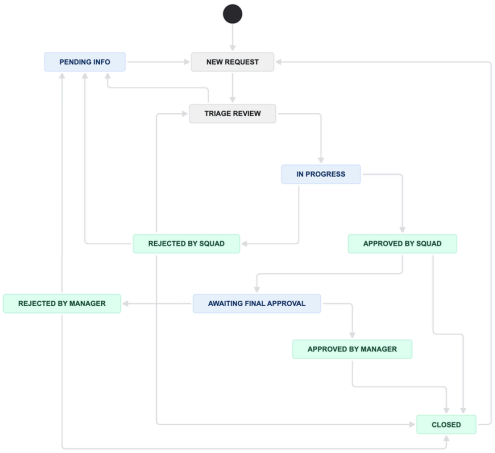


Figure 1: Jira workflow

2. **Design and specify issue layout**

In order to support the workflow presented above and ensure clarity, consistency, and efficiency not only during the triage step but throughout the entire process, the following fields have been selected for use. These fields were chosen based on their relevance to the creation and evaluation of a ticket. This means that some fields that will appear to the Squad Manager, for example, won't appear for the Requester. While some fields are mandatory by default in Jira, others have been deliberately set as mandatory to ensure users provide the necessary information when creating an issue. When creating an issue, the fields below will be visible:

Field Name	Purpose	Mandatory to fill when submitting an issue?
Priority	Defines urgency of the request helping the Triage team.	Yes
Summary	Brief title describing nature of request	Yes

Description	Detailed explanation of the request. Crucial for triage team.	Yes
Reporter	identifies requester's name	*Jira auto-filled*
Request Type	used by triage to know the type of ticket (promotion, performance KPI, performance results)	Yes
Needs Manager Approval?	indicates if ticket needs manager approval before final closure	Yes
Attachment	allows user to upload docs s supporting their ticket	No

Table 1: fields used when creating a request with the current workflow

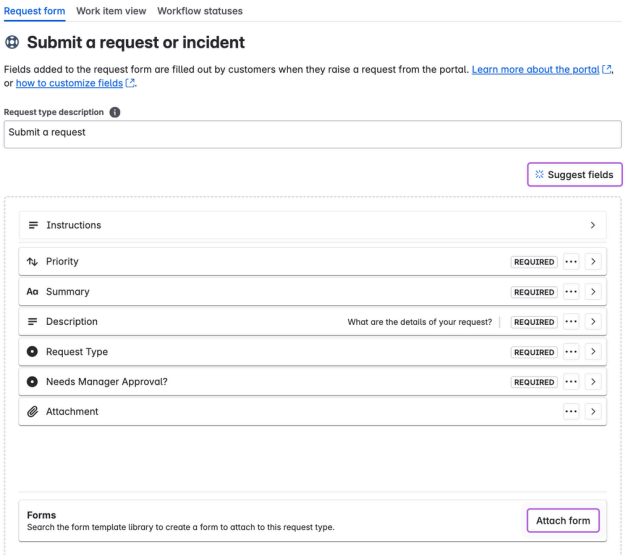


Figure 2: submitting a request layout

3. **Design Roles in Workflow**

Applying conditions: To ensure a structured and accountable ticket lifecycle, workflow conditions were applied to restrict who can transition issues between specific statuses (new request to triage review, fr example). This setup prevents unauthorized transitions, ensures role-based responsibilities are respected, and contributes to process reliability across the HR/Talent flow. Therefore, the following conditions were implemented (use figure 1 as guidance):

- **New Request** to **Triage Review**
  Only the original ticket creator of the ticket can perform this transition, ensuring initial ownership.
- **Triage Review** to **In Progress / Pending Info**
  Only the Triage Manager can assess and assign tickets to squads or return them for missing information.
- **In Progress** to **Approved/Rejected by Squad**
  Only members of the assigned squad (Alpha or Beta), and their manager, can decide on the outcome of the request during execution.
- **Approved by Squad** to **Approved/Rejected by Manager**
  Only the designated Manager has permission to issue the final approval or rejection of the squad's proposal.
- **Rejected by Squad** to **Triage Review**
  Only the Triage Manager can reassess and redirect tickets that were incorrectly assigned or deprioritized.
- **Pending Info** to **Triage Review**
  Only the original reporter can confirm that the missing information has been added and resubmit the ticket.

This configuration balances flexibility with control, ensuring that tickets progress only when the responsible party takes action. See example below where only the riage manager can move a ticket from Triage Review to In Progress.

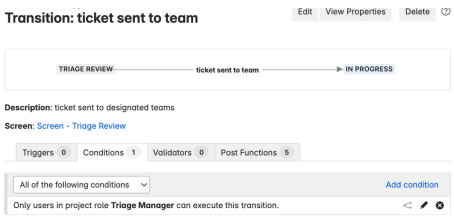


Figure 3: Example of conditions applied to a transition in Jira

Applying fields: Fields are editable during specific stages of the workflow, depending on the role involved and the purpose of each status. Edits are allowed at the beginning of the ticket's lifecycle and while the ticket is being assessed or worked on. Once the ticket reaches a decision point, such as being approved, rejected, or closed, editing is restricted to preserve integrity and traceability.

More specifically:

- **Fields are editable** when the ticket is in:
  - **New Request** – The requester can submit or update the initial information.
  - **Triage Review** – The triage manager can adjust and assign the request.
  - **Pending Info** – The original requester can revise and resubmit missing or incorrect information.
  - **In Progress** – Squad members working on the task can add or update relevant details.
  - **Awaiting Final Approval** – The assigned manager can review and modify fields if needed before final approval.

To ensure each role only interacts with the relevant fields at the right stage of the process, we created custom screens linked to specific workflow statuses. This allows us to control who can update which field and when. Each screen is assigned to a role and tied to a status transition. This structure helps maintain clarity, data quality, and accountability:

- **Requesters:** can input initial details (e.g. summary, description) but cannot change them afterward.
- **Triage Managers:** can assign squads, update request types, and set priorities during the triage phase.
- **Squad Members:** can comment, add attachments, modify request type (if incorrect) and change manager approval requirement.
- **Squad Managers:** can comment and modify or add attachments.

The image below outlines which fields appear in each screen, and which role owns them. This was applied to Jira as well. The higher the hierarchy, the less work is necessary to do on the ticket:

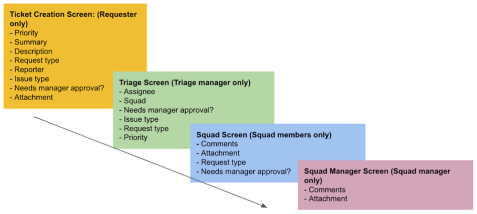


Figure 4: Workflow screens with fields showing up for each ticket stage.

4. **Process map tool designed on Lucidchart and adhering to BPMN 2.0 rules.**

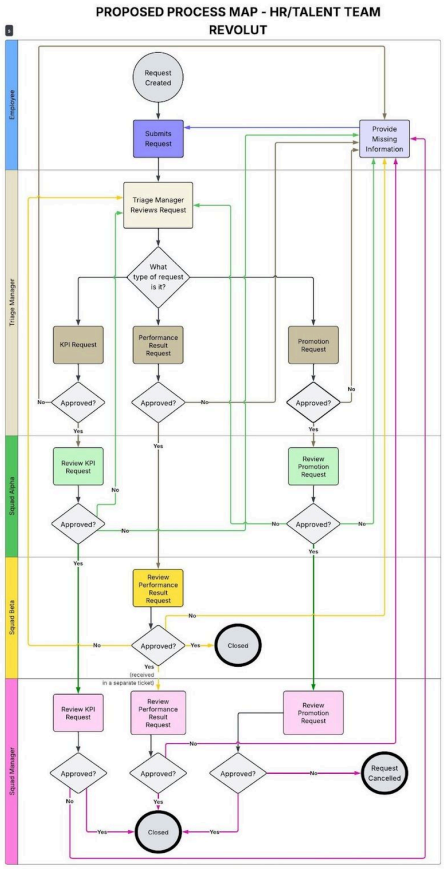


Figure 5: suggested process map, created on Lucidchart, containing swimlanes and adhering to BPMN 2.0 rules.

5. **Define the KPIs and SLAs**

Defining SLAs for the project:

The Talent/HR team previously operated without a structured triaged process, causing:

- inconsistent resolution times, especially for more complex requests (high standard deviation)
- no clear accountability across the different roles.
- lack of visibility of who was actually responsible for the delays within the teams
- Limited ability to track team and individual team members performance.

To address the points highlighted above a structured SLA system would be introduced to our workflow with suggestive time targets (that can be later adjusted based on actual performance).

Process Step		**SLA Target**	**Justification**
Triage Review		Within 2 hours	Ensures requests are reviewed quickly and routed to the right team without delay.
Squad Execution	**KPI requests**	Within 6 business days	(prev. ~12 days) with better structure, we expect time to complete a task to drop by half
	Promotion requests	within 6 business days	(prev. ~12 days) with better structure, we expect time to complete a task to drop by half
	Performance results	2 business days	Faster to handle. No big change.
Pending info - Requester		within 3 business days	prevents process from getting stagnated when returning to

		Requester
Manager Approval	within 3 business days	reduce delays in final decision

Table 2: SLA targets implemented on Jira

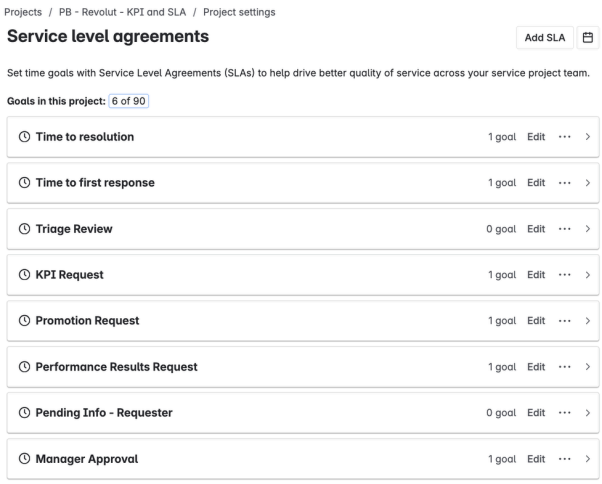


Figure 6: SLA targets implemented on Jira

With the information above, a variety of reports were created where these metrics can be compared to each other. Image below shows an example of the time a ticket remains inside each team/report type:

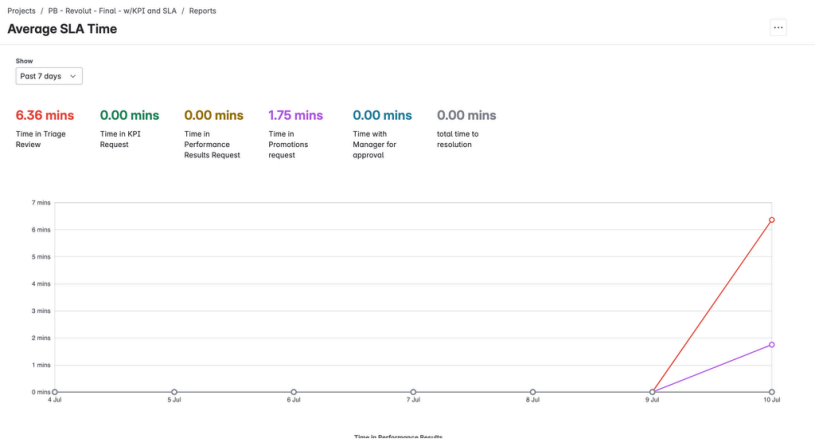


Figure 7: Average SLA time report

KPIs to measure team and individual performance

To complement the SLA structure defined above, it's now crucial to define the team and individual KPIs. On a team-level, these KPIs should measure how effectively and efficiently the team deals with requests. On an individual-level, these KPIs should mostly focus on the contribution of each team member, the quality and how efficient they are. These metrics aim to fix some of the issues we saw originally. Some of the KPIs highlighted below have been added to Jira using Filters:

- lack of accountability across team roles
- inconsistent resolution times with high standard deviation
- limited visibility on individual performance
- no clear tracking mechanism for leadership

Team KPIs

% of tickets resolved within SLA	tracks how well the team adheres to the SLA targets set for each process step.
Average time to resolution	the actual time it takes between request creation and completion
% of tickets reopened	indicates potential quality issues in the resolution process
Weekly ticket volume ratio	total number of tickets resolved per week, divided by total number of tickets created. Helps tracking workload
% of tickets returned to requester	indicates the quality of initial submissions.
Individual KPIs	
Tickets handled (per user)	number of tickets where user performed at least one step
SLA compliance rate (per user)	% of tickets completed within the SLA during their assigned step
Average handling time	average time a user spends working on a ticket
Rejection or rework rate	% of tickets rejected or returned for rework
Open ticket aging	number of tickets assigned to the user that have been open beyond average time

Table 3: individual and team KPIs crucial for our workflow

6. **Design the controls for the process using Key Risk Indicators (KRI)**

With the KPIs and SLAs already in place, it's now crucial to implement some level of control over these parameters. In other words, it's using those SLAs in an active manner so that it creates an alert when risk thresholds are breached. To achieve this, Jira Automations were used.

Alerts were created every time our SLA time frames were breached. Once this happens, we forced the system to send an automatic email, with some relevant information, warning the relevant people about this. See some examples below of what has been implemented.

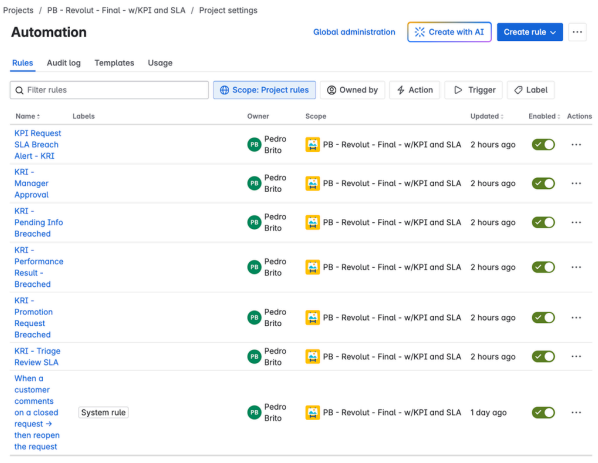


Figure 8: alerts implemented in Jira using Automation

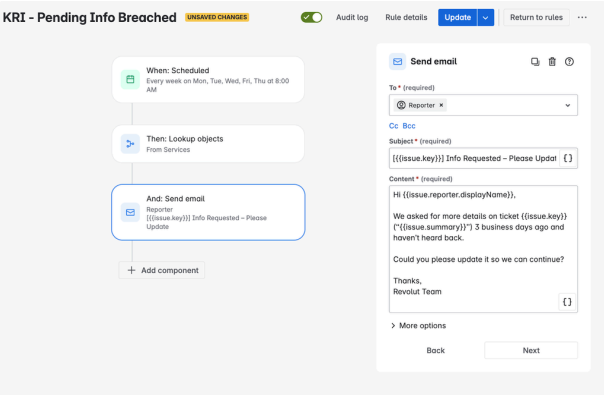


Figure 9: Email alert configuration for then “Pending Info” SLA is breached.

7. **Suggest Jira automations and/or operational changes to reduce time and cost spent for the process**

Auto-transition by via comment:

- Approving, rejecting or moving a ticket around still involves some manual labor (even with transitions in place).
- It could be quicker and easier if a manager or Squad member could approve, reject a ticket by just writing a comment “approved”, “rejected” in the comment box.

Auto-triage using AI

- Our suggestion relies on a team to perform the triage process.
- This is always prone to errors, especially due to the tight deadlines (SLA imposed) we created.
- Using machine learning to auto-triage tickets can save time, resources and improve overall quality.
- The machine would learn from successful tickets that have never been rejected, the trends and what a high quality ticket must have.

User feedback

- To track feedback from the process, at the end of each status, the user should be capable of adding comments on how the process should be improved.
- The comments should then be directed to the owner of the process for updating.

8. **Risk Control Self Assessment**

A Risk Control Self Assessment (RCSA) table was created to translate our process design, SLAs and KPIs into a clear set of controls that can be monitored, enforced and continuously improved. If we can measure it, we can improve it. These can be considered as guardrails for our system, allowing us to know where the weaknesses of our process are and where to look at when taking action.

The table links each workflow step to its primary failure mode or delay (the KRI), the corresponding SLA, and any implemented automations. It also shows which controls are active in Jira and which are still pending, proposes next-step improvements, and assigns a severity rating to each risk. This structure enables the business to prioritize and address issues as they arise, acting quickly and efficiently.

Control Objective	Key Risk (KRI)	KPI/SLA	Control Implemented	Implemented in Jira?	Next steps	Severity
Triage Review ≤ 2 h	Requests sitting un-triaged >=2h causing delays	Triage Review SLA Breached	Email to triage manager when SLA target is breached	Y	Monitor breach rate weekly. escalate if >5%	High
Squad Alpha Execution ≤ 6 bd (KPI/Promotion)	Squads exceed 6 bd on KPI/Promotion tasks,	KPI Request SLA Breached” “Promotion Request SLA	Alert email to squad member & squad manager	Y	After 1 month, tune SLA based on actual median	Very High

	delaying overall cycle	Breached	when SLA breach detected		completion times	
Squad Beta Execution ≤ 6 bd (KPI/Promotion)	Performance results tasks take > 2 bd, slowing notifications	Performance Results SLA Breached	Alert email to squad member & squad manager when SLA breach detected	Y	Report average times; adjust SLA if too aggressive/lenient	High
Pending Info ≤ 3 bd (Reporter)	Tickets stuck pending info > 3 bd, freezing progress	Pending Info SLA Breached	Alert email to ticket owner reminding to complete ticket.	Y	Add “pending too long” gadget to dashboard	Medium
Squad Manager Approval ≤ 3 bd	Manager decisions delayed > 3 bd, preventing ticket closure	Manager Approval SLA Breached	Alert email to Squad Manager when SLA breached	Y	Consider auto reminders every 24 h	Very High
Minimize reworks & reopens	High % of tickets reopened indicates quality issues	% Tickets Reopened	Creation of a weekly filter to track this	N	Quality team to understand issue if it happens	Medium
Workload vs. intake	Surge in new requests floods squad capacity.	weekly volume ratio (resolved/created)	Report “Created vs Resolved” by week	N	Interchange tickets between Squads to balance workload.	Very High
Improve Initial Quality	High % of tickets returning to requester means low efficiency	% Returned to Requester	Monthly check number of tickets going to Pending Info	Y	Understand the feedback on why they got rejected	Medium

Table 4: Risk Control Self Assessment table

9. **Document the Process in Confluence**

The link to our Confluence is the following:

Revolut HR/Talent Task - Pedro Brito

Link to associated project in Jira is the following:

10. **Design a MI Dashboard**

The dashboard created for our project consolidates some of our most critical metrics and alerts in real time. On the left, the Created vs Resolved line chart shows weekly ticket volume alongside resolution counts, helping us instantly spot workload imbalances. Next to it, the % Tickets Returned to Requester table highlights issues stuck in the "Pending Info" state, so we can understand about the root-cause feedback loops. The middle column surfaces our "Tickets Created This Week" ratio and, below it, a Service Project SLA Report gadget that displays current SLA compliance percentages for each process step (triage, KPI requests, performance results, promotion requests, manager approval). On the right, rows of Issue Statistics gadgets break out each request type's Breached and Completed counts, allowing us to see at a glance which workflows need attention.

By grouping SLA met vs. breached filters alongside real-time counts of unresolved or reopened tickets, the dashboard functions as an operational control panel. It alerts us immediately to SLA breaches (our key risk indicators), tracks resolution times, and surfaces trends in ticket volume. This layout ensures our team can focus on the biggest process friction points, whether that's too many late manager-approvals, spikes in promotion requests, or tickets pending requester info.

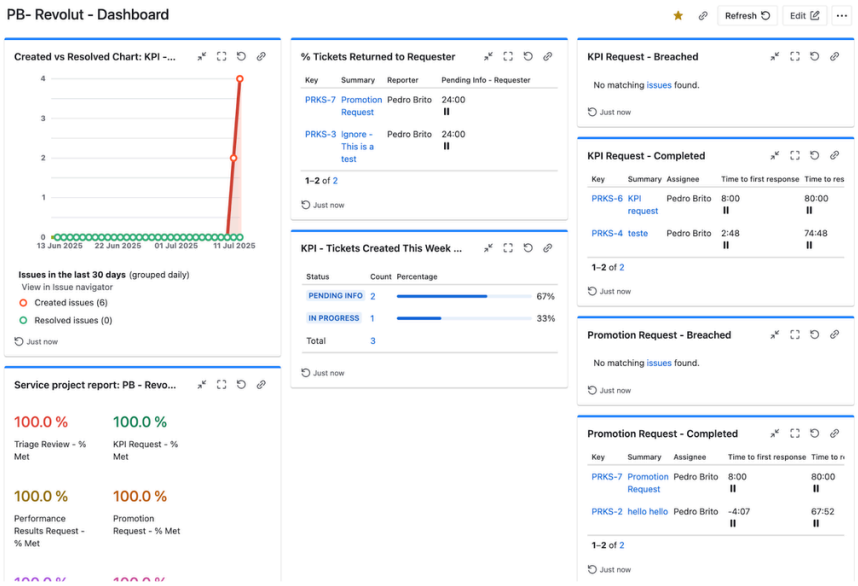
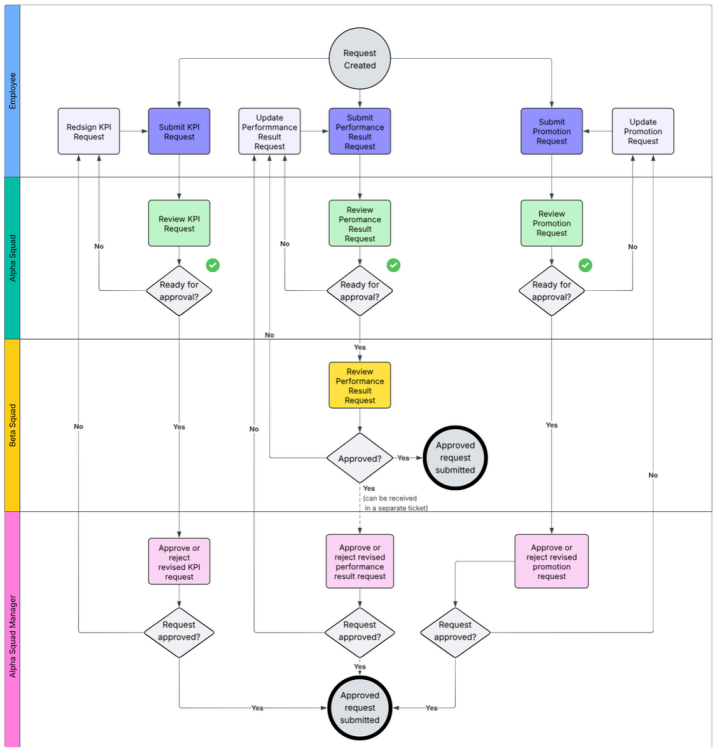


Figure 10: Current dashboard view for our Revolut project.

Appendixppendipmroilf

ORIGINAL PROCESS MAP - HR/TALENT TEAM
REVOLUT



Appendix 1: Original process HR/Talent team process map, highlighting requests cycle.