Kanbanstudy
Targeting success

A Guide to the

KANBAN BODY OF KNOWLEDGE (KBOK™ GUIDE)

APPENDIX A. ADDITIONAL LEARNING

The Practical Implementation Guide for Managing Workflows using Kanban

(Includes Examples from popular digital Kanban tools, facilitates integration with other Agile frameworks, and recommends ways to use AI for increased productivity.)

APPENDIX A. ADDITIONAL LEARNING

A.1 Scaling Kanban and Integrating with Scrum (ScrumBan)

ScrumBan is a hybrid agile framework that combines elements of Scrum and Kanban to provide an effective method for managing work. It incorporates Scrum's iterative processes alongside Kanban's principles of visualizing Workflows and promoting continuous progress. This combination offers a structured yet flexible approach to organizing and completing Tasks.

When starting an initiative, teams can choose whether to use Scrum or ScrumBan (Scrum combined with Kanban) to manage their work. This guide explores how Kanban can be applied to manage Tasks across all teams within an organization. The use of Scrum for managing projects, IT operations, and DevOps is discussed in detail in the *A Guide to the Scrum Body of Knowledge (SBOK® Guide), Fifth Edition*, and in the white paper *Extending Scrum to IT and DevOps*.

Typically, Scrum projects and Kanban Workflows are managed separately by different teams. However, at the enterprise level, businesses may have portfolios and programs that involve multiple Scrum projects and Kanban Workflows. To scale Kanban across an enterprise, it is essential that those responsible for managing portfolios and programs have visibility into all the work being performed by Scrum and Kanban Teams within the company. This may include multiple Prioritized Product Backlogs, Kanban Backlogs, and releases.

Figure A.1 shows a Kanban Board interface for 'Kanban Initiative 1,' displaying Tasks and user stories categorized by status (To Do, In Progress, Done). Various Tasks, such as 'Task Group,' and user stories like 'User Story 2' and 'User Story 1,' are visible, with some marked as complete and others showing ongoing progress, indicating a collaborative Workflows environment.

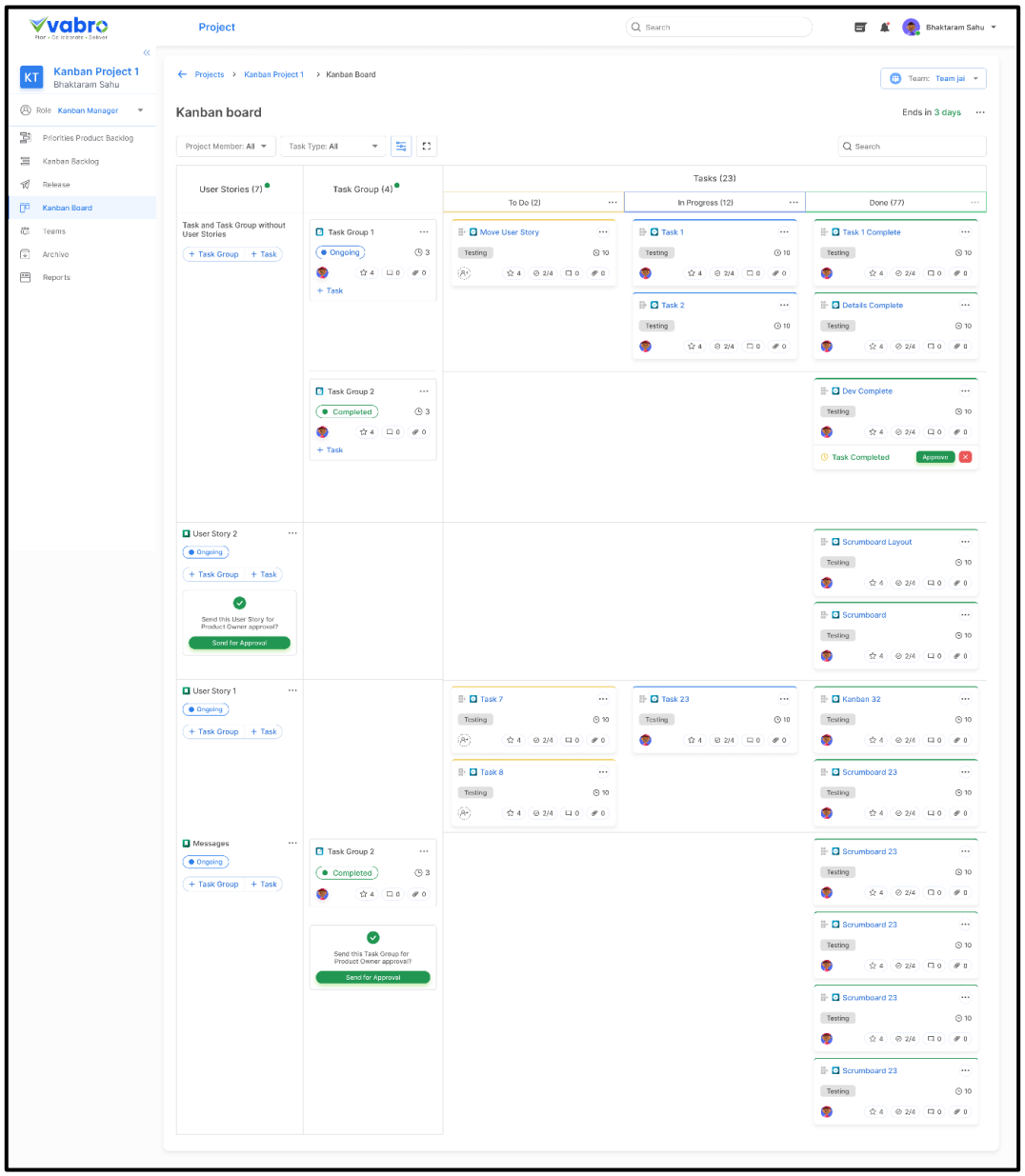

Figure A. 1: Use of Kanban Board in Scaling Kanban (Source: Vabro)

Figure A.2 shows a Kanban Board backlog in Vabro with various Tasks categorized by Task Group, user story, and category. It indicates their estimated time, assigned team, and current status. The Tasks are organized across different boards and assigned to various team members, reflecting the progress and Workflows of an ongoing project.

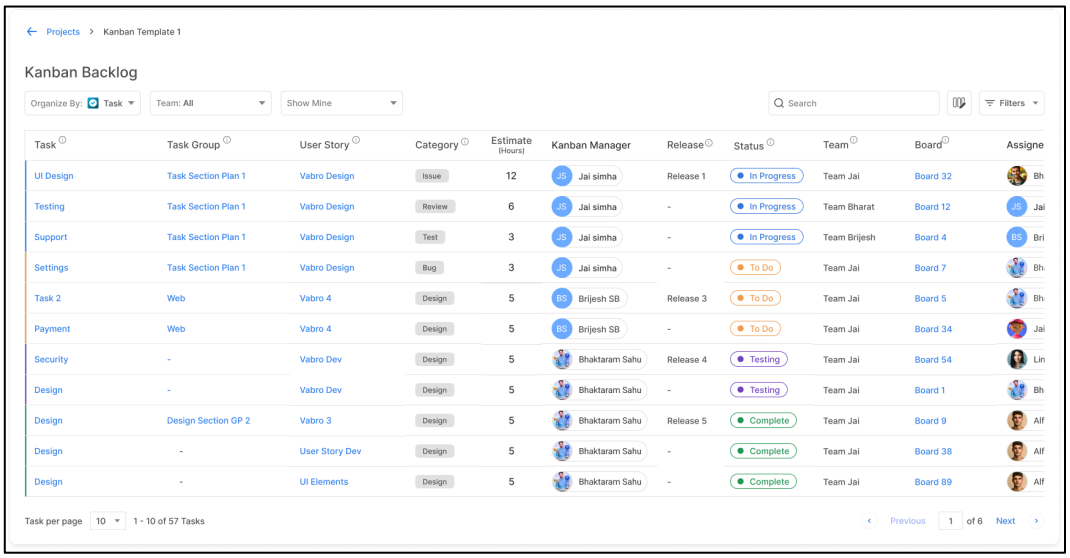

Figure A.2: Use of Kanban Backlog in Scaling Kanban (Source: Vabro)

A.1.1 Differences between Kanban Board and Scrum Board

Kanban Boards and Scrumboards are both visual tools used to manage work, but they serve different purposes. The key differences between a Kanban Board and a Scrum Board are:

- Kanban Boards allow team members to add Task Groups and Tasks directly to the board.
- Scrum Boards are used to track Tasks associated with user stories that have been pulled into the Sprint Backlog by the Scrum Master (to be worked on during the sprint).
- Kanban Boards can have any number of columns, offering more flexibility. This flexibility is not typically available with Scrum Boards, which are primarily used for tracking progress. Scrumboards generally have three main columns, with one or two additional columns that the Scrum Master can add to monitor specific progress details.
- Kanban Boards can specify a Work in Progress (WIP) limit within each column. The WIP limit is the maximum number of Tasks allowed in that column. WIP limits are not defined for Scrumboards as Scrumboards are used to depict the different stages of work within a Sprint (which is a time-boxed development cycle).
- Kanban Boards include completed work, planned work (yet to be started), work that has been put on hold, and more. Tasks that are no longer relevant to the current active work being done by the team can be archived. This ensures that WIP (Work-in-Progress) limits remain meaningful, as they apply only to relevant Tasks within each column of the Kanban Board. In contrast, Scrum Boards are time-based and focus solely on work in the current Sprint. They are reset at the end of each Sprint, which typically lasts one to four weeks.

A.1.2 Using Kanban for Broader Organizational Collaboration

When Kanban Boards interact with other Kanban Boards across an organization, it creates a cohesive ecosystem where dependencies and cross-functional Workflows are managed efficiently. To set up an effective Kanban structure for seamless interaction across the organization, the following steps can help ensure transparency, efficient dependency tracking, and streamlined communication:

A.1.2.1 Cross-Board Visibility

- Linked Kanban Boards: Establish links between boards for different teams (e.g., development, QA, marketing) to represent dependencies visually. Most Kanban tools allow linking cards across boards, making it easy to track which Tasks from one team depend on or impact Tasks from other teams.
- Shared Dependency Columns: Consider adding a "Waiting on Another Team" or "External Dependency" column on each board to capture when a Task is blocked by another team's work. This signals to the broader organization that coordination is required and flags potential issues before they become blockers.

A.1.2.2 Dependency Mapping and Signals

- Dependency Cards: Use special dependency cards or tags to highlight Tasks that are dependent on work from another team. For instance, if a feature being developed requires specific customer data from the sales team, the development Kanban Board should reflect this dependency. This visual indicator ensures that teams are aware of dependencies and can prioritize accordingly.
- Color-Coded Indicators: Apply color-coded tags to quickly signal which team is involved in each dependency (e.g., blue for development, red for design, green for marketing). This visual approach helps stakeholders across boards easily spot dependencies and plan their Workflows accordingly, ensuring a smoother coordination across teams.

A.1.2.3 Using Forms for Cross-team Collaboration

- Forms for Collaboration: Forms enable users and team members to collaborate across initiatives, workspaces, and organizations. Forms can be created at the Board level and made accessible to users. Once a user submits a form, a relevant Task is automatically generated at the Board level. Teams can then pick up these Tasks and work on them completely. Upon completion, the user is notified about the result. Additionally, automation can be applied to further streamline and automate the form handling process.
- Form Grouping and Access Levels: Forms can be organized based on various levels, such as Organization, Workspace, Workflows, or User. They can also be categorized according to the origin or function of the form, such as HR, Finance, Tech, etc. Users can view the status of forms and have the ability to edit or delete them as needed.
- Private vs. Public Forms: Forms can be set to either private or public access. Public forms allow any user to submit by accessing a link or embedding the form on third-party sites, similar to Google Forms. Private forms, on the other hand, are restricted and have dedicated pages created by the Kanban Manager or Kanban Team at the board level for more controlled access.

- A typical form can include the following information:
    - Name
    - Description
    - Form Link
    - Who can use the Form (Access Type: private or public)
        - Current board members
        - Members from this project
        - Members from this workspace
        - Current workspace members
        - Members from this organization
        - Anyone with the link
    - Target Artifact such as Task Group/Task
    - Artifact Name
    - Artifact Description
    - Artifact Priority
    - Artifact Category
    - Artifact Assignee
    - Form Level such as User/Team/Project/Workspace/Organization
    - Are form edits allowed after submitting?
    - Is form withdrawal allowed after submitting it?
    - Created by, Created on will be shown once form is created.
    - Form resolution options
- Once forms are submitted, Task Groups/Tasks will be created at the designated location or board for the respective teams. If permitted, the Kanban Manager or a Kanban Team Member can reject the form request before a Task Group or Task is created. In the Backlog and on the Board, an indicator can be displayed to differentiate between auto-generated Task Groups and manually created ones.
- Once approved or completed, the form is considered finished, and the user will be notified. When approving or sending the Task Group for approval, the Kanban Manager or a Kanban Team Member can choose from the resolve options defined during the form setup.

Figure A.3 shows a digital form interface from Airtable, titled 'New Product Launches,' in edit mode within a workspace. The form includes fields for 'Launch Name,' 'Status,' 'Launch Date,' 'Description,' 'Tags,' 'Start Date,' 'Deadline,' 'Flagged,' and 'Assets,' indicating that it is designed for managing and tracking new product launches.

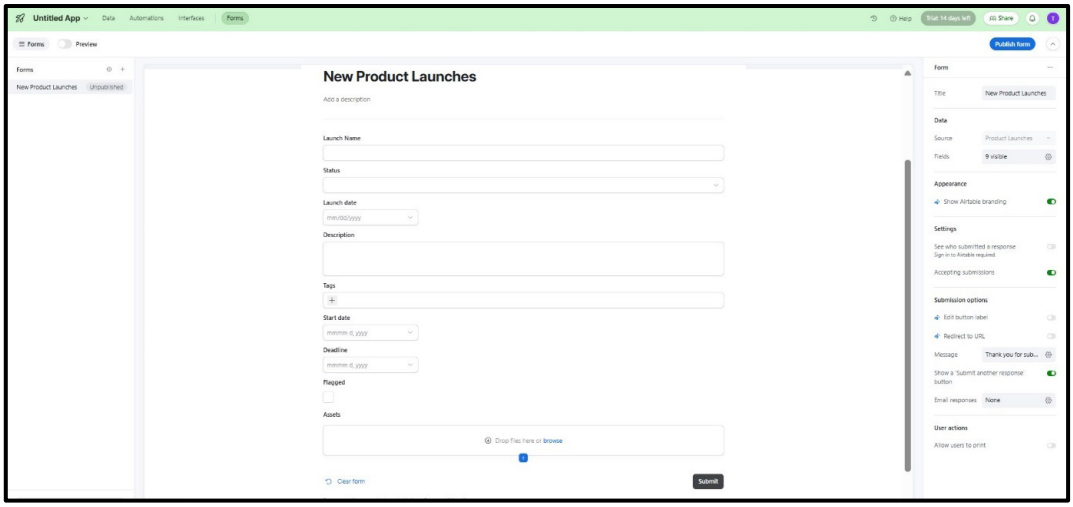

Figure A.3: A Variant of Forms used in Kanban (Source: Airtable)

Figure A.4 shows a form for the 'Automation' screen within a Nifty workspace, focused on editing a 'Project Request Form' with specific fields like 'Name (First and Last),' 'Email,' and 'Project Description.' It includes options to mark fields as required and select input types. The interface allows users to customize form fields and their properties to streamline project requests.

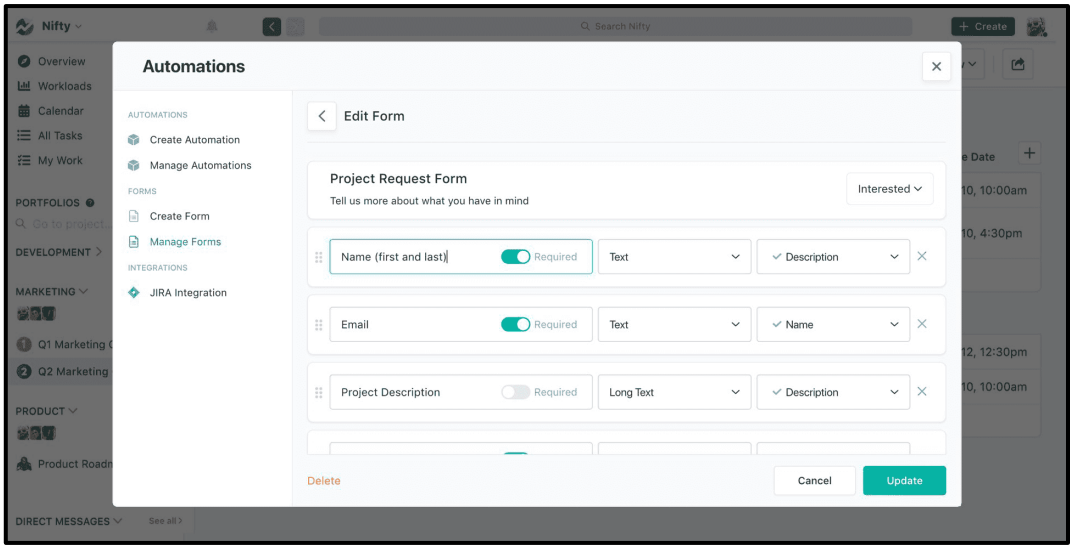

Figure A.4: A Variant of Forms used in Kanban (Source: Nifty)

Figure A.5 shows a form for the 'Automation' screen within a Nifty workspace, focused on editing a 'Project Request Form' with specific fields like 'Name (First and Last),' 'Email,' and 'Project Description.' It includes options to mark fields as required and select input types. The interface allows users to customize form fields and their properties to streamline project requests.

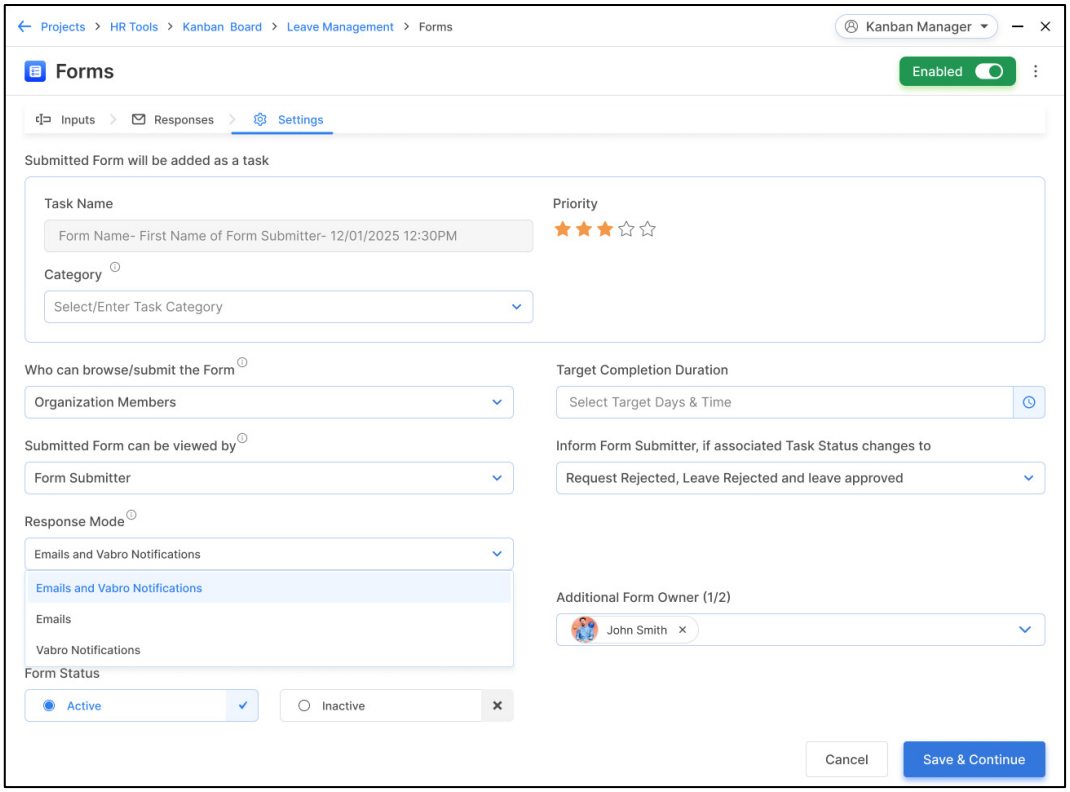

Figure A.5: Use of Forms Settings for Flexibility (Source: Vabro)

A.1.2.4 Example of How Forms can be set up in an Organization

Assume that the HR team creates a form for Leave Management allowing organization members to apply for leave.

Workspace	Human Resources
Project	Manage HR Operations (Kanban Workflows)
Board	Leave Management [Columns: Pending, Under Review, on Hold, Complete]
Product Owner	HR 1
Kanban Manager	HR 2
Kanban Team Member	HR 3 and HR 4

Table A-1: Example of Setting up Forms in an Organization

HR can set the following options in the Form:

<table>
<tr><td>Name</td><td>Apply Leave</td></tr>
<tr><td>Description</td><td>A form for organization members to apply leave.</td></tr>
<tr><td>Target Artifact Type</td><td>Task<table><tr><td>Artifact Name</td><td>Leave Request - [USER_NAME]</td></tr><tr><td>Artifact Description</td><td></td></tr><tr><td>Artifact Priority</td><td>4 Stars</td></tr><tr><td>Artifact Category</td><td>Leave</td></tr><tr><td>Artifact Assignee</td><td>HR 2</td></tr></table></td></tr>
<tr><td>Approve the form submission request before creating the artifact?</td><td>No</td></tr>
<tr><td>Form Status</td><td>Active or Inactive</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Submitted Forms can be viewed by</td><td>Only me/Members in Kanban Board (Select a Board)</td></tr>
<tr><td>Who can use the form/Form Access/Permissions?</td><td>All Members from this organization:</td></tr>
<tr><td>Notify all affected members via email and notification?</td><td>Yes</td></tr>
<tr><td>Confirmation Message</td><td>Thank you for submitting your leave request. It has been forwarded to the respective teams. Please check your email for further details and updates.</td></tr>
<tr><td>Limit number of responses</td><td>No limit</td></tr>
<tr><td>Form edits allowed after submitting</td><td>No</td></tr>
</table>

Form withdrawal allowed after submitting	Yes
Form resolve Options	Approved Default: Your leave has been approved from [START_DATE] to [END_DATE] for [DURATION] days. Rejected Default: Your leave has been rejected from [START_DATE] to [END_DATE] for following reasons [Enter reason here]

Next, HR can create the form with the necessary details, including employee name/ID, email address, leave start/end dates, duration, reason for leave, and any other relevant information.

A.1.2.5 Regular Cross-Team Syncs and Standups

- Schedule short, regular sync meetings or standups between dependent teams to review cross-board statuses. During these syncs, teams can use their Kanban Boards as a reference to discuss dependencies and provide updates on progress or potential delays."
- Escalation Workflows for Dependencies: Establish an escalation process for unresolved dependencies that exceed a certain time limit. Teams can agree on a standard protocol, such as flagging the Task or moving it to a dedicated escalation column to bring it to leadership's attention.

A.1.2.6 Utilize Swimlanes for Cross-Team Projects

- For projects involving multiple teams, create swimlanes specific to each team or project phase. This keeps Tasks organized while illustrating the flow of work across different functions."
- Cross-Functional Swimlanes: For larger initiatives, incorporating cross-functional swimlanes within the board allows teams to see the entire Workflows without needing to jump between boards. This is particularly helpful for product and project managers who oversee the end-to-end process.

A.1.2.7 Align Work-in-Progress (WIP) Limits Across Teams

- If work is passed between teams, set shared WIP limits or align them based on typical throughput to prevent overloading any single team in the Workflows. A manageable WIP for each team ensures smooth workflow, reduces bottlenecks, and enhances the efficiency of cross-board interactions.

A.1.2.8 Dedicated Channels for Cross-Board Communication

- Use integrated communication channels within Kanban tools (e.g., Slack, Teams, or in-app comments) for Tasks involving other teams. A centralized discussion space for each Task facilitates updates, reduces redundant communication, and ensures that relevant information is easily accessible to all parties.

A.1.2.9 Track Key Metrics on Dependencies

- Measure the time Tasks spend in dependency-related stages, the number of Tasks dependent on other teams, and the average resolution times for these dependencies. Analyzing these metrics helps identify bottlenecks and informs process improvements across teams.

A.1.2.10 Service-Level Agreements (SLAs) for Inter-Board Dependencies

- Define SLAs or agreed timeframes for each team to complete Tasks that will impact another board. SLAs foster accountability across the organization, enabling teams to coordinate and plan more effectively around each other's timelines. Integrating multiple Kanban Boards using these methods helps teams manage dependencies, share accountability, and ultimately deliver projects more efficiently across all functions.

A.2 Using Kanban for DevOps

Along with developing new products and services, organizations typically have operational support and maintenance functions incorporated into their production environments. These operational or “Ops” functions were historically carried out by specialized teams or resources that worked differently from development teams. They also used different IT software tools and frameworks for managing operational activities. However, many organizations today have opted to combine development and operations functions into DevOps, integrating development and operational support activities.

Operational work typically involves recurring or repetitive Tasks, such as maintaining or supporting established products or services. This also applies to business operations like new employee hiring, financial reporting, payroll processing, as well as IT operations, such as customer support, call centers, systems monitoring, infrastructure management, and routine procedures. Operational work can be predictable, with set rules and procedures in place to address specific situations. On the other hand, project or development work produces a unique product or service and has a definitive beginning and end. By nature, project work is less predictable, as it consists of unique activities or Tasks specific to the project. Examples include creating a new software application or developing new functionality for an existing application.

Organizations can apply Kanban principles and artifacts to DevOps processes to streamline Workflows, improve collaboration, and increase efficiency. Using Kanban for DevOps involves defining and visualizing the Workflows of development and operational processes, limiting work in progress, prioritizing requirements, monitoring progress, and making continuous improvements.

Figure A.6 shows a Vabro interface for the "Prioritized Product Backlog," featuring various user stories, such as "Vabro mob design" and "Vabro dev." Each user story is assigned to different teams and boards, marked with priority stars and status indicators. The interface allows for effective organization and filtering of user stories, with options to create new ones or manage upcoming releases.

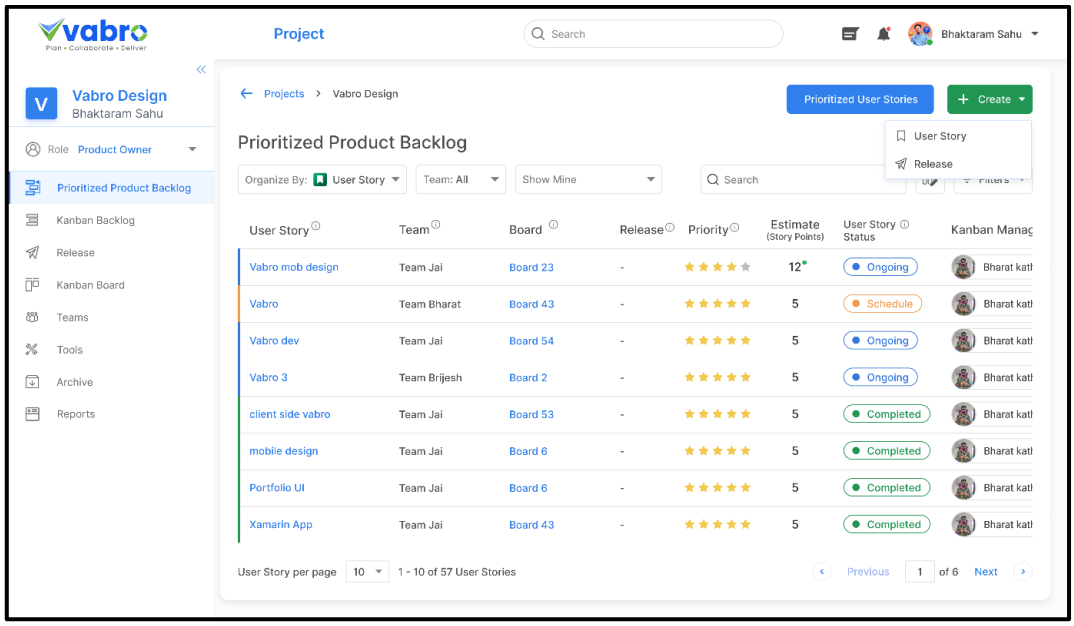

Figure A.6: Applying Kanban for DevOps (Source: Vabro)

A.2.1 Applying Kanban to DevOps

Kanban can be applied to DevOps in the following specific ways:

- Kanban Backlog Creation: When starting a Kanban initiative, the Product Owner can immediately create a Kanban Backlog for the initiative. Creating a Prioritized Product Backlog is optional; the Product Owner may choose to create one only if there's a need to define User Stories for specific requirements.
- User Stories Addition: The Product Owner adds User Stories to the Kanban Backlog or the Prioritized Product Backlog. The Kanban Manager pulls high-priority User Stories from the Prioritized Product Backlog to the Kanban Board. The Kanban Manager can pull in any number of User Stories at any time, but for each User Story, a target completion date should be set. The Kanban Manager or the Kanban Team can also estimate the effort required for each User Story, which helps in determining the amount of work to be completed based on the team's cycle time and velocity.
- Completion and Review: Once the User Stories pulled from the Prioritized Product Backlog are completed, they are reviewed by the Kanban Manager and then submitted to the Product Owner for approval. User Stories that are rejected by the Product Owner due to non-completion or non-compliance with the Acceptance Criteria are moved back into the Prioritized Product Backlog. These stories can then be worked on by the same or another Kanban Team involved in the initiative.

Summary of Using User Stories in Kanban:

- Creation: User Stories follow a predefined structure and serve as a simple way to document requirements and desired end-user functionality. These requirements are short, simple, and easy to understand, improving communication among business stakeholders and enabling better estimations by the team. User Stories help teams grasp user needs and expectations for the deliverables to be created. The Product Owner is responsible for creating and prioritizing User Stories in the Prioritized Product Backlog.
- Work Status: As the team works toward delivering results, the Kanban Team uses the Kanban Board. Team members break down User Stories into Task Groups and/or Tasks, which are then worked on by the team.
- Approval: Approval of User Stories, Tasks, or Task Groups can be sought from the Product Owner or another designated role/personnel, depending on the approval process set at the time the User Stories or Task Groups are created.

In some cases, a Kanban Board may not have any User Stories pulled from the Prioritized Product Backlog. In such instances, the board will display only the Task Groups and Tasks created by the Kanban Team. The Kanban Manager has the ability to view, edit, or delete Task Group Templates for the Kanban Team.

A.2.2 Kanban DevOps Roles

The typical roles in Kanban can be adapted by incorporating additional responsibilities when applying Kanban principles to a DevOps environment. The Kanban Team, as described in section 3.1, consists of the Product Owner, Kanban Manager, and Team Members.

Product Owner—In a DevOps environment, the Product Owner may have the following additional responsibilities, in addition to the standard duties associated with Kanban:

- Creating, Adding, and Prioritizing User Stories: The Product Owner is responsible for creating, adding, and prioritizing User Stories in the Kanban Backlog (or alternatively in the Prioritized Product Backlog) if User Stories are needed. It is important to note that defining User Stories is optional when applying the Kanban method.
- Defining Acceptance Criteria: The Product Owner defines the Acceptance Criteria for each User Story and ensures that the team understands the business value of the work they are doing.

Kanban Manager— When applying Kanban to DevOps processes, in addition to the typical responsibilities associated with Kanban, the Kanban Manager may have the following additional responsibilities:

- Pulling User Stories: Pulling User Stories from the Kanban Backlog (or Prioritized Product Backlog) to the Kanban Board for the team to work on.
- Estimating and Targeting: Ensuring that all User Stories are estimated by the team and have a specified target completion date before they are worked on.
- Reviewing and Submitting: Reviewing and submitting completed User Stories to the Product Owner for approval.
- Incident Management: Acting as the Incident Manager, responsible for reviewing and managing incidents and issues as they arise.
- Release Management: Serving as the Release Manager to help streamline DevOps releases.

Team Members— When applying Kanban to DevOps processes, in addition to the typical responsibilities associated with Kanban implementation, the Kanban Team Members may have the following additional responsibilities:

- Estimating User Stories: Estimating User Stories and specifying target completion dates.
- Creating Task Groups and Tasks: Creating Tasks and Task Groups to deliver the User Stories pulled into the Kanban Backlog by the Kanban Manager (from the Prioritized Product Backlog set up by the Product Owner). Creating Tasks and Task Groups to deliver User Stories pulled into the Kanban Backlog by the Kanban Manager (from the Prioritized Product Backlog set up by the Product Owner)

A.2.3 Kanban DevOps Releases

The Product Owner and/or Kanban Manager can optionally specify Releases. Releases represent functionalities or results to be delivered together at a specified time (e.g., new or improved features to be made available to stakeholders). To ensure that a Kanban Release is completed on time, multiple Task Groups, Tasks, and User Stories may need to be completed.

If Kanban Releases are created by the Product Owner or Kanban Manager, there should be an option to add Tasks to a Kanban Release when creating a User Story, Task Group, or individual Task. This ensures that all work related to the Kanban Release is tracked and contributes to the successful delivery of the corresponding deliverables.

Figure A.7 displays a "Portfolio Release Dashboard" in Vabro, showcasing a list of software releases for the "Vabro Portfolio". It includes the release names, release dates, progress in terms of completed features, and overall status (Ongoing, Scheduled, Completed). For example, "Vabro Release 1" and "Release V1.0" are marked as ongoing, "Release V3.0" and "Release V2.0" are scheduled, while "Release V4.0" has been completed.

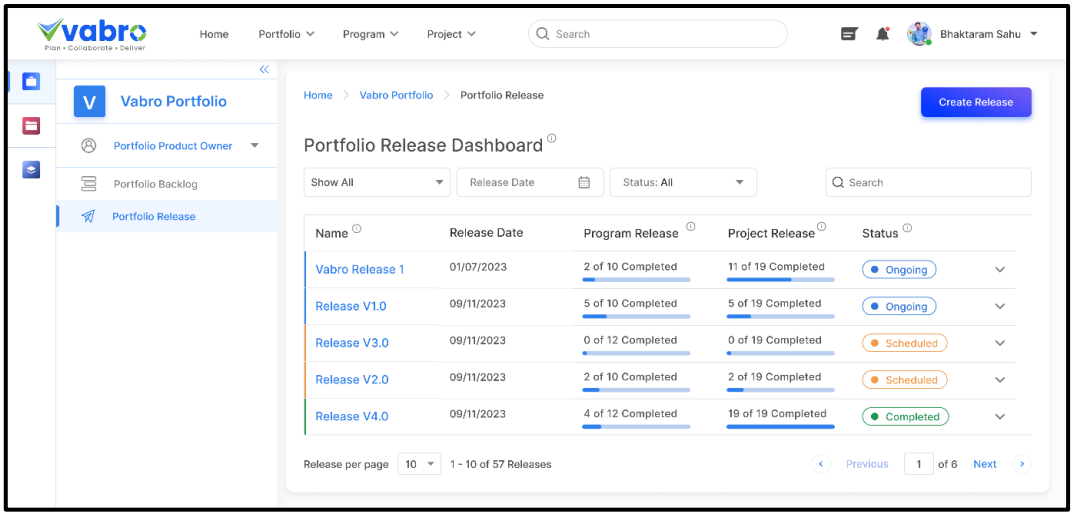

Figure A.7: Release Management in Kanban for DevOps (Source: Vabro)

The Product Owner specifies which User Stories are part of a Release. For independent Task Groups and Tasks not associated with a User Story, the Kanban Manager and/or Kanban Team Members can define which Tasks and Task Groups contribute to the results of the Release. User Stories, along with independent Tasks and Task Groups, can be pulled into a Release at any point before the Release is completed. A Release is considered done when all User Stories, Task Groups, and Tasks identified for the Release have been completed. By integrating Kanban with Scrum and DevOps, organizations can create a powerful combination that leverages the best of each framework. This synergy offers flexibility, continuous delivery, and high productivity for development teams.

- **Kanban with Scrum for Enhanced Workflows Efficiency**
  - Managing Work-in-Progress (WIP): Scrum organizes work into sprints with set timeframes and goals, while Kanban's WIP limits help prevent overload within a sprint, allowing the team to focus on completing high-priority Tasks without compromising quality.
  - Improving Flow Within Sprints: Kanban's pull-based system enables team members to take on Tasks as they become available, enhancing Workflows. Within a Scrum sprint, the Kanban Board displays Tasks at different stages, making it easy to visualize progress and identify potential bottlenecks. This visualization can increase Task completion rates and provide more accurate sprint forecasts.
  - Enhanced Sprint Planning: Teams can use Kanban data (such as cycle time and throughput) to better estimate the work for upcoming sprints. This data-driven approach helps Scrum teams improve the accuracy of their sprint planning and capacity estimation, leading to more predictable and reliable sprint outcomes.

- **Kanban and DevOps for Continuous Delivery**
  - Visualizing the DevOps Pipeline: Kanban aligns well with the continuous integration and continuous delivery (CI/CD) model of DevOps. By representing each stage of the pipeline on the Kanban Board (e.g., development, testing, deployment), teams can visually track the movement of code from commit to production. This allows for real-time monitoring and ensures smoother transitions between development, testing, and deployment.
  - Reducing Deployment Cycle Times: The constant flow of Tasks in Kanban, combined with DevOps automation, helps reduce deployment cycle times. Automating testing and deployment stages allows teams to quickly move items across the board, shortening the feedback loop and enabling faster delivery.

- Emphasizing Continuous Improvement: Kanban promotes continuous process refinement and improvement. By continuously monitoring Kanban Metrics such as lead time and cycle time, DevOps teams can identify areas of the deployment pipeline that require optimization, fostering ongoing improvement.

- **Daily Stand-ups and Kanban for Sprint Transparency**
  - Kanban Visualization in Standups: In Scrum's daily standup, using a Kanban Board to display Tasks that are in progress, completed, or blocked provides a quick, visual overview of the sprint's current status. This enhances transparency, making it easier to discuss roadblocks and next steps. It also allows for swift course corrections within the sprint, benefiting both Scrum and DevOps teams.
  - Kanban Signals for Task Progress: Kanban uses visual signals, such as color-coding, to indicate Task progress and priority levels. These signals can be particularly useful during daily Scrum standups by highlighting urgent issues, Tasks needing immediate attention, or dependencies. This visualization helps teams prioritize effectively and allocate resources where they're most needed.

- **Aligning Scrum's Retrospectives with Kanban's Continuous Improvement**
  - Retrospective Insights: Both Scrum and Kanban emphasize continuous improvement. In Scrum, retrospectives are held at the end of each sprint, whereas Kanban encourages a continuous review of metrics such as WIP, cycle time, and throughput. Combining these insights allows teams to review both sprint outcomes and Workflows efficiency, providing a broader perspective on potential improvements.
  - Addressing Process Bottlenecks: Scrum retrospectives can leverage Kanban data to identify and address bottlenecks. If specific Tasks consistently remain stuck in a particular stage, teams can adjust their processes, redefine WIP limits, or focus on resolving dependencies. This blend of practices creates a more adaptable approach to iteration and improvement.

- **Kanban Metrics and DevOps for Continuous Feedback**
  - Leveraging Kanban Metrics: Metrics like lead time, cycle time, and throughput help identify areas for improvement, enabling DevOps teams to address inefficiencies in the pipeline. These metrics provide insights into how quickly code moves through each stage, helping refine the CI/CD pipeline for faster, more reliable delivery.
  - End-to-End Monitoring: DevOps relies on continuous feedback throughout the development lifecycle. Kanban enhances this by visually representing feedback stages and Tasks awaiting validation. By incorporating feedback into the Kanban Workflows, teams can quickly address issues that arise post-deployment, improving product reliability and performance.

- **Blending Roles and Responsibilities**
  - Cross-Functional Collaboration: Kanban Boards facilitate collaboration between Scrum and DevOps teams by providing a shared view of Tasks. Developers, testers, and operations personnel can all use the same board to track their work, enhancing visibility and cross-functional alignment.
  - Defining Clear Ownership: In DevOps, responsibilities can often be blurred between development and operations teams. Kanban Boards help by assigning clear Task ownership and specifying each team member's role at every stage. This alignment between Scrum's role clarity and Kanban's visual Task ownership enhances accountability.

- **Scaling Kanban, Scrum, and DevOps**
  - Supporting Cross-Team Visibility: In larger organizations, multiple teams often work concurrently on different parts of a product. Kanban's visual system enables teams to track dependencies and identify potential blockers between Scrum teams and DevOps functions, fostering better coordination.
  - Method Flexibility: Kanban can be used at both the team and portfolio levels to provide real-time insights into the status of various teams' work. This scalability makes it easier for organizations to maintain alignment across Scrum, DevOps, and other teams working on different initiatives.

Combining Kanban with Scrum and DevOps creates a Workflows that balances structured planning with flexibility and continuous delivery. By integrating these frameworks, organizations can build a more resilient and adaptive development environment that meets the needs of today's fast-paced tech landscape.

A.3 Aligning Kanban with OKRs and Organizational Goals

OKRs (Objectives and Key Results) and organizational goals are essential strategic tools that companies use to drive and measure success in business initiatives. Organizational goals are broad, strategic objectives that outline a company's vision and the key outcomes needed to achieve success. They serve as guiding principles for all departments and teams.

OKRs, on the other hand, provide a structured framework for setting and tracking goals. They consist of two main components: Objectives and Key Results. Objectives are qualitative, inspirational statements that define what an organization aims to achieve. Each objective typically has multiple key results, which are measurable outcomes that indicate progress toward the achievement of that objective.

Objectives Example:
Objective: Increase customer satisfaction.
Objectives are broad, qualitative, and inspirational statements that define what the organization wants to achieve.

Key Results Example:
For the objective "increase customer satisfaction," the key results could include:

- Improve the Net Promoter Score (NPS) from 75 to 90.
- Achieve a 97% positive feedback rate from customers on support tickets.

Key results are quantitative, measurable outcomes that indicate how the organization will achieve the stated objectives. They are specific and actionable, providing a clear way to track progress.

A.3.1 Aligning the Kanban method with OKRs and Organizational Goals

Aligning Kanban with OKRs and broader organizational goals can significantly enhance a company's efficiency, focus, and adaptability. Kanban, with its visual, flow-based approach, is effective for managing work-in-progress and improving productivity. On the other hand, OKRs provide clear, measurable goals that help drive progress.

Steps involved in aligning the Kanban method with OKRs and organizational objectives include:

- Define the Organizational Goals: Establish clear strategic goals that reflect the vision of the organization. These goals should be aligned with the overall mission and vision of the company.
- Set Objectives and Key Results (OKRs): Break down the broader organizational goals into specific, measurable objectives. Key results should be actionable and measurable, helping track progress toward each objective.
- Map OKRs to Kanban Workflows: Integrate OKRs with the Kanban Board by assigning relevant Tasks or Work Items that align with the objectives and key results. Each Work Item should contribute to achieving specific OKRs.
- Monitor and Adapt: Regularly review the flow of work on the Kanban Board and evaluate how the completion of Tasks aligns with the progress of OKRs. Adapt Workflows or processes as necessary to ensure alignment with organizational goals.
- Foster Continuous Improvement: Use Kanban's continuous improvement principles, such as analyzing bottlenecks and optimizing Workflows, to ensure that work processes are continuously refined and more effectively aligned with OKRs.

Figure A.8 shows the ClickUp setup screen, where a user is selecting features of interest, such as Chat, Automations, Tasks & Projects, Dashboards, Goals & OKRs, Docs & Wikis, Forms, and Time Tracking, among other options. The interface indicates that the user is customizing their workspace by selecting the tools and functionalities that best suit their needs.

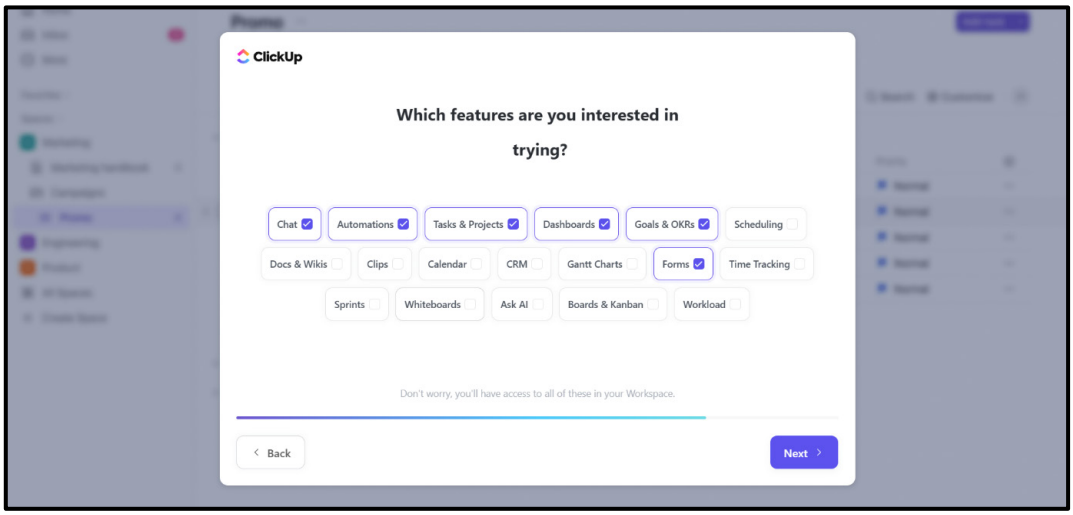

Figure A.8: Use of OKRs and other Features in Kanban (Source: ClickUp)

A.3.2 Defining Objectives and Key Results (OKRs) and Goals in a Kanban Context

When aligning Kanban with OKRs and goals, objectives can be defined at various levels, such as the organization, workspace, and Workflows. Workspace serves as the top level in the project delivery or product development hierarchy, allowing organizations to define, plan, track, and monitor Workflows in a centralized space. It helps streamline collaboration and ensures focus on delivering value at each stage of the process. In digital Kanban tools, workspaces are further broken down into projects or Workflows, offering flexibility to users. Multiple workspaces can be created or joined, but each workspace remains independent, enabling tailored management of each initiative.

Objectives at these levels can be defined as follows:

- At the organizational level, an Organization Admin can:
  - Define and manage both Organization-Level and Workspace-Level OKRs and goals. Objectives should include descriptive titles, detailed descriptions, and target outcomes/Key Results.
  - Set progress tracking settings and assign weightage for company and workspace goals. The following items, which contribute specific percentages to the objective, can be utilized:
    - Workspace
    - Project
    - Workflows
  - Maintain visibility into all workspace, Workflows, and individual OKRs (if set to public).
  - Monitor workspace goals to track their alignment with organizational objectives.
  - Ensure company objectives are linked to the progress of associated workspace, and Workflows goals.
- At the Workspace level, a Workspace Admin can:
  - Define and manage Workspace-level and Workflows OKRs.

- Set progress settings and weightage for Workspace and Workflows goals. Relevant items contributing to the objective percentage can include:
  - Workflows
  - User Stories (for DevOps)
  - Task Groups (for Kanban)
- Link Organizational Goals to Workspace Goals as Parent Goals through an approval process.
- Provide visibility into all Workflows and their associated OKRs within the Workspace. They can also view linked Organization Goals.
- Ensure Workspace goals are aligned with and support the broader company objectives.

- At the Workflows level, a Product Owner or Kanban Manager can:
  - Define and manage Workflows-level OKRs.
  - Set progress tracking and assign weightage to Workflows goals. The following items, which contribute a specific percentage to the objective, can be utilized:
    - Relevant user stories
    - Task Groups
  - Link Workspace Goals to Workflows Goals as Parent Goals through an approval process.
  - Provide visibility into all user stories, Task Groups, and associated OKRs within the Workflows.
  - Ensure that Workflows goals are aligned with company objectives.
- At the individual level, a team member can:
  - Define and manage personal OKRs privately.
  - Link Task Groups and Tasks they are working on to their individual OKRs.
  - Provide visibility into their own OKRs and assigned Tasks.
  - Ensure that Workflows goals are aligned with company objectives.

Figure A.9 shows a Vabro platform dashboard for tracking organizational and individual IT goals, focusing on "Organization Goals/Objectives" and a visible progress board. It highlights key results and their progress toward objectives such as "Increase Customer Satisfaction by 50%," "Improve Customer Response Time," and "Enhance Ticket Resolution Efficiency," each with specific timelines and assigned personnel.

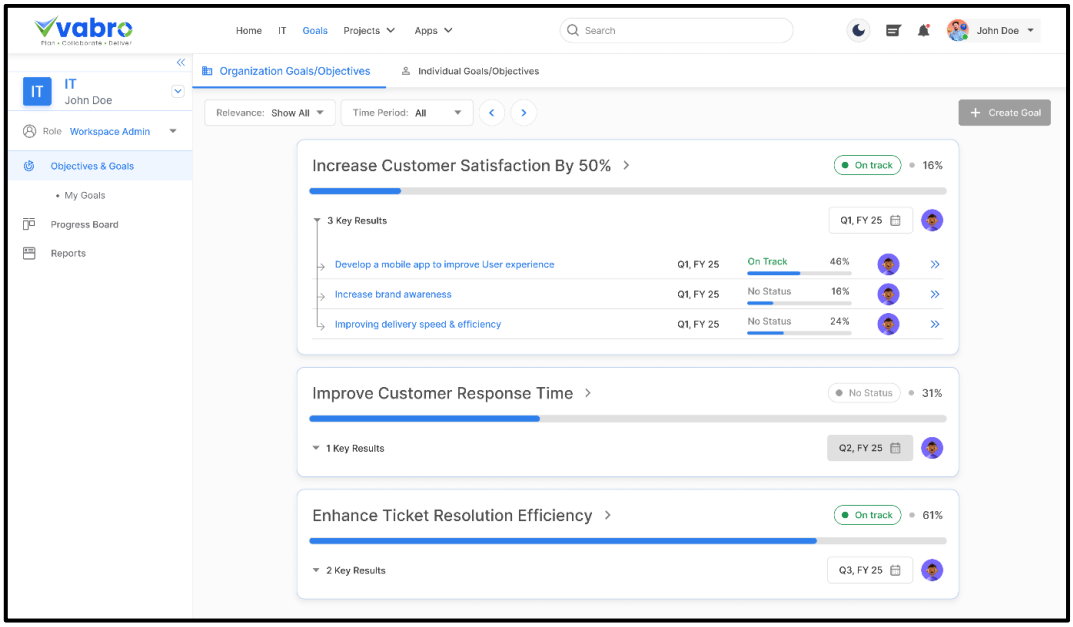

Figure A.9: Defining OKRs for an Organization in Kanban (Source: Vabro)

Figure A.10 shows a goal-tracking dashboard in Jira, focused on "Increase Customer Satisfaction." It displays progress and related Workflows, highlighting two sub-goals: "Develop & Launch VMFoods Mobile App" and "Expand Online Delivery to Tier 2 Cities," each with progress bars and assigned timelines.

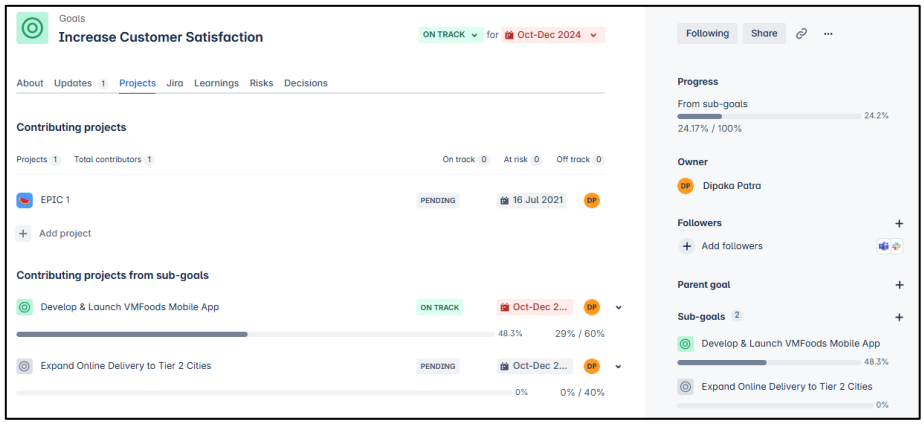

Figure A.10: Company-level OKRs (Source: Jira)

Figure A.11 shows a goal-tracking dashboard for VMFoods, Inc., displaying the company's strategic objectives and their progress. It highlights key goals such as "Increase Sales" and "Increase Customer Satisfaction," along with their sub-goals and assigned teams, indicating the timeline and current status of each objective.

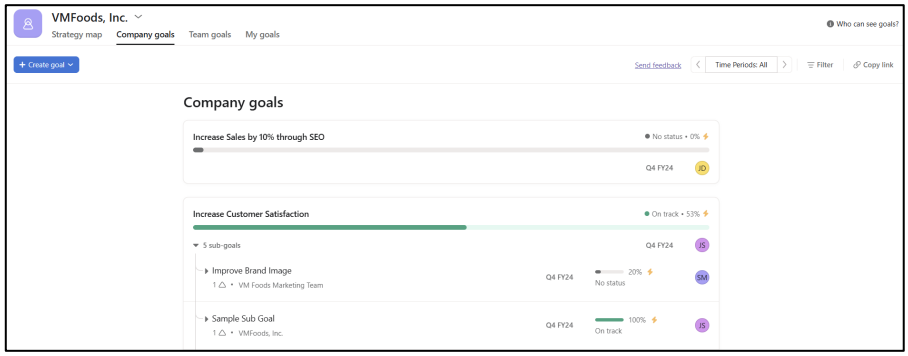

Figure A.11: Company Goals (Source: Asana)

Figure A.12 shows a goal-tracking interface in ClickUp, focused on increasing brand awareness and engagement, with current progress at 9%. It lists target metrics, such as increasing website traffic by 20% and growing social media followers by 30%, showing progress indicators and Task breakdowns for each.

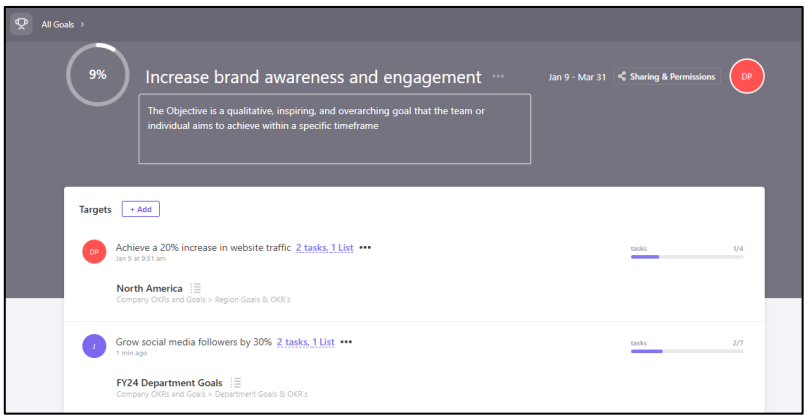

Figure A.12: Example of a Company-level OKR (Source: ClickUp)

Objectives should be ambitious, qualitative goals that drive the team's purpose and align with the organization's vision. For example, an organizational-level objective could be 'Enhance product quality and customer satisfaction'.

Key Results: These are quantitative measures that indicate progress toward the objective, such as "Reduce production defects by 20%" or "Achieve 90% positive customer feedback." With Kanban, OKRs and goals can be integrated by associating specific Tasks on the Kanban Board with these key results. Each column on the board, representing different stages of work, becomes a visual roadmap toward the objective.

Figure A.13 shows a progress tracking interface in Vabro for the key result, "Develop a mobile app to improve User Experience," which is part of the goal to "Increase Customer Satisfaction by 50%." The completion rate is 54%, and the Workflows is on track. The interface displays associated projects, Workflows, and related documents, along with a progress graph and a comment section for team communication.

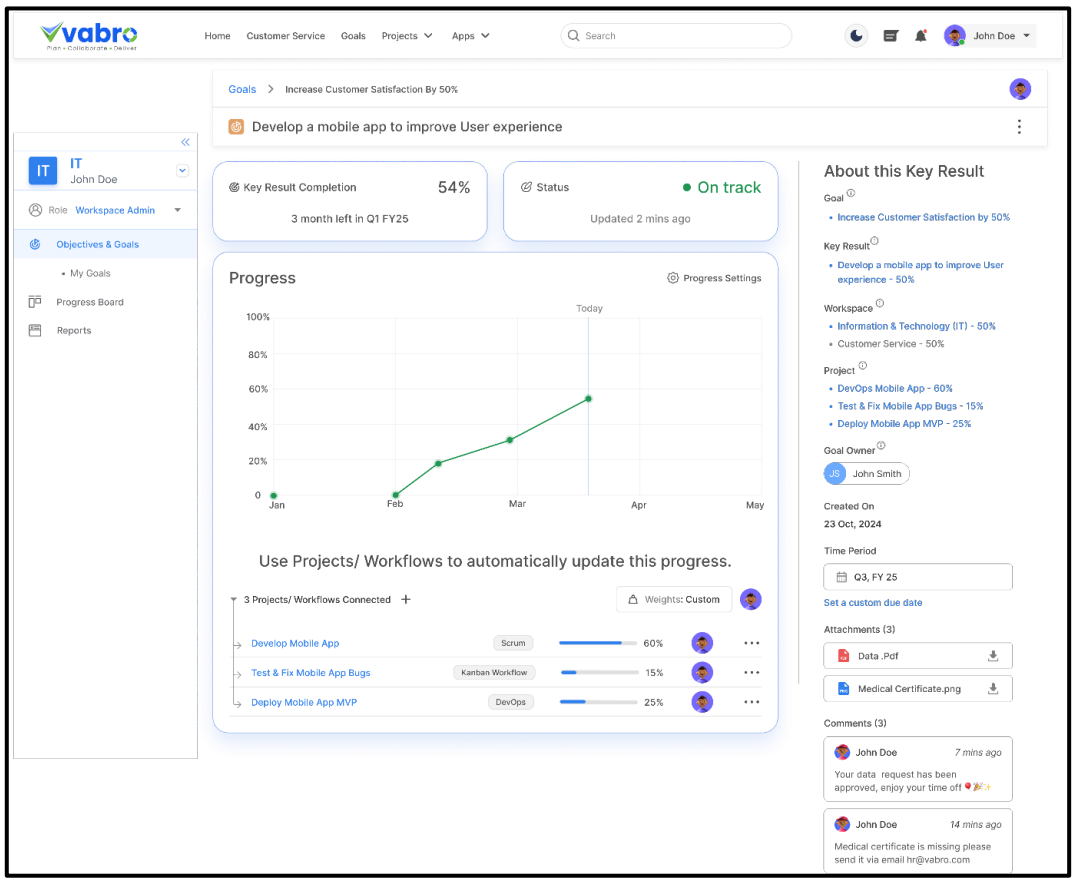

Figure A.13: Tracking OKRs in Kanban (Source: Vabro)

A.3.3 Using Kanban for Transparency in OKRs and Goals Alignment

Using Kanban for Transparency in OKRs and Goals Alignment involves:

- Visualizing Progress: The Kanban Board provides a clear view of ongoing Tasks, their priorities, and how they align with the organization's goals. When Tasks are directly linked to key results, teams gain better awareness of their contribution to larger objectives.

- Prioritizing Work: By placing high-impact Tasks at the top of the Kanban Workflows, teams can prioritize OKR-related work first. This focus ensures that resources are directed toward what matters most for achieving objectives.

A.3.4 Kanban WIP Limits to Support OKRs and Goals

Using WIP limits to support OKRs and organizational goals helps in:

- Reducing Overload: Limiting work-in-progress (WIP) on the Kanban Board helps teams avoid Task overload and maintain a steady pace toward key results. By focusing on fewer Tasks that contribute to OKRs, the team can improve both the quality and efficiency of their output.
- Encouraging Flow: When Tasks move more fluidly through the Kanban system, the team can better assess progress toward OKRs. WIP limits encourage consistent work toward objectives while minimizing bottlenecks.

A.3.5 Continuous Improvement (Kaizen) for Alignment with OKRs and Goals

In Kanban, aligning the effort of continuous improvement (Kaizen) with OKRs and goals is achieved through:

- Retrospectives: Regular retrospectives allow the team to assess progress toward OKRs and adjust their Kanban processes as needed. This practice fosters a continuous improvement (Kaizen) mindset, supporting iterative progress toward objectives.
- Process Tweaks: By analyzing Workflows patterns, teams can adjust WIP limits, re-prioritize Tasks, or streamline processes to better align with the organization's goals and improve overall performance against OKRs.

A.3.6 Integrating Feedback Loops for Enhanced Alignment with OKRs and Goals

In Kanban, the integration of feedback loops for enhanced alignment with OKRs and goals is achieved in the following ways:

- Data-driven Decisions: Kanban metrics, such as lead time, cycle time, and throughput, provide valuable data for tracking progress on OKRs. If a particular key result is lagging, this data can guide adjustments to resource allocation or process improvements.
- Frequent Check-Ins: Embedding regular feedback loops within the Kanban process allows the team to make small, informed adjustments to stay on track toward OKRs, quickly adapting to any changes in business needs.

A.3.7 Regular Review of OKRs and goals for Realignment

Regular Review of OKRs and Goals for Realignment involves:

- Quarterly OKR Revisions: Since OKRs are typically set on a quarterly basis, Kanban Boards should be adjusted periodically to reflect shifting priorities. When OKRs are reviewed, teams can add or modify Tasks to align with any new objectives or key results.
- Goal Realignment: As organizational objectives evolve, the Kanban Board should evolve as well. Teams should regularly review the alignment between the board's Tasks and the updated OKRs, ensuring that day-to-day work continues to contribute to long-term goals.

By combining the flexibility of Kanban with the goal-focused structure of OKRs, organizations can maintain a strong sense of purpose while adapting dynamically to changing demands. This alignment fosters a culture where both immediate priorities and long-term objectives are transparently connected and continuously pursued.

A.4 Lean Kanban

The Lean concept optimizes an organization's system to produce valuable results based on its resources, needs, and alternatives while reducing waste. Since these factors are dynamic and constantly evolving, a Lean organization evaluates the entire system rather than focusing only on individual components and continuously fine-tunes its processes. Lean's foundation asserts that reducing the length of each cycle (iteration) increases productivity by minimizing delays and aiding in early error detection. Consequently, shorter cycles reduce the total effort required to complete Tasks.

Kanban is designed to reduce idle time in the process of business value creation. The Kanban system determines what the process needs and when it needs it, using visual aids as a signaling system to guide future actions. It is closely associated with the design of pull systems and the concept of delivering just-in-time goods. Lean Kanban integrates the visualization methods defined by Kanban with Lean principles. Adopting Lean Kanban practices and principles brings several benefits to an organization, such as creating a manageable workload based on team capacity, reducing disruptions and delays by minimizing downtime due to errors, and fostering cross-alignment by building a culture of engaged workers. Overall, Lean Kanban helps achieve better process control through process improvement.

In a broad sense, Lean is a set of values and principles guiding successful product development, while Kanban is a process tool for applying these values and principles in practice. Lean Kanban combines these practices and tools to create value from product concept to delivery.

A.4.1 Core Values of Lean Kanban

The core values of Lean Kanban are as follows:

- **Continuous Improvement** Lean Kanban adopts the concept of Kaizen, rooted in the Japanese philosophy of business that emphasizes continuous improvement. "Kaizen" consists of two parts: "Kai," meaning an idea for change or an action to rectify something, and "Zen," meaning "good." This concept is based on the belief that there is always room for improvement in business, ultimately leading to the successful elimination of waste and excess.

- **Visualization of Work** A Kanban Board is a popular tool for implementing the Kanban methodology and increasing productivity. The board helps teams organize their work using cards (Kanban), each representing a different feature, and placed on the board. It aids in the visualization, control, and optimization of Workflows. Kanban Boards are flexible and can be customized to match the Workflows and associated needs.

- **Respect for People and Processes** Lean Kanban values each person and process within a business as essential contributors to the organization. Every idea has the potential to add to the company's success. Lean promotes an environment of recognition and respect, which is crucial for clear communication. This respect encourages people to share their opinions and suggestions, leading to potential improvements.

- **Limit Work in Progress** This value advocates for using buffers to smooth out variability in the effort required to complete a Task. The goal is to avoid overloading the system.
  The key element is limiting new Tasks based on the system's available capacity, keeping total work within predetermined WIP limits. Developers must maintain a smooth process flow, avoiding excessive Tasks in WIP. A new Task is only pulled when the current Task is completed.

- **Eliminate Waste and Target Zero Defects** Lean Kanban utilizes statistical quality control and an adaptive human-centric structure to keep team members motivated to achieve zero defects in their work.

- **Standardize Work and Manage Workflows** Lean Kanban emphasizes standardizing Workflows or processes to ensure consistency, quality, and efficiency in results. Managing Workflows involves monitoring and reporting each Task, focusing on the speed and smoothness of movement. Fast movement creates quick value, while smooth movement minimizes delays and makes the process more predictable. Developers following Kanban break down work into smaller chunks to complete Tasks faster and facilitate better Workflows.

- **Empower Teams and Encourage Long-Term Thinking** Lean Kanban prioritizes empowerment at all levels—individual, team, and organizational. Empowerment fosters improvement and sustained growth, encouraging long-term thinking focused on sustainable growth and development rather than short-term gains.

A.4.2 Lean Kanban Practices

The core practices of the Lean Kanban method are as follows:

Optimize the Whole

- Optimizing part of a system will eventually optimize the entire system. This includes focusing on the entire value stream, delivering complete products, and implementing long-term thinking throughout the organization. A value stream is a list of steps to create value for both the customer and the team. The value stream must be as realistic and detailed as necessary to visualize and understand the Workflows. It should also be able to adapt as the process and context change.

Eliminate Waste

- In the Toyota Production System (TPS), waste is termed "muda" and refers to anything in the system that does not add value to the customer. Visualizing the Workflows or process is essential for identifying and eliminating waste. Accurately describing the current value stream and articulating the desired value stream for the product cycle is critical, especially in the early stages, to prevent failure due to early errors. Waste can arise from building the wrong thing, failing to learn, or practices that impede the process.

Build in Quality

- Frequent defects indicate flaws in the overall process. After identifying the starting point in the value stream, the next step is to categorize work types, such as features, bugs, Tasks, and change requests. Improving quality involves:
- Determining Workflows by modeling the work rather than the workers and representing activities in the order they are performed.
- Breaking dependencies and designing the system to accommodate feature additions at any stage, using historical data and empirical techniques to anticipate demand and adjust the system.
- Ensuring final verification of the system is error-free.

Learn Constantly

- Lean Kanban assumes that nothing is perfect, and there is always room for improvement. Continuous learning is facilitated by:
- Developing a system that responds rapidly to change.

- Making process or Workflows tolerant of change.
- Clearly defining when and what changes are required, acknowledging the financial implications of decisions.
- Learning from mistakes and challenging preset standards to improve continuously.
- Using empirical techniques to reach conclusions, favoring quantitative over qualitative methods.

Engage Everyone

- Organizations gain a competitive edge through human resources, as insight, creativity, and intellect are scarce resources. To maximize these, organizations should:
- Encourage team autonomy.
- Promote specialization by providing challenges, constructive feedback, and growth opportunities.
- Engage individuals by showing the purpose of each Task and its role in the bigger picture, thereby boosting productivity.

The Practical Implementation Guide for Managing Workflows using Kanban

The Kanban Body of Knowledge (*KBOK™ Guide*) offers guidelines for successfully implementing Kanban, a widely used Agile methodology for managing business workflows. Originally developed in manufacturing, Kanban is now applied across various industries and sectors, including software development, healthcare, education, human resource management, retail, sales and marketing, finance, and more. It works for organizations of all sizes, from small businesses to large enterprises.

The *KBOK™ Guide* is built on insights from thousands of workflows across industries, with significant input from the global Kanban community and the VMEdu® Global Authorized Training Partner Network, comprising over 2,000 companies in more than 50 countries. Its development was a collaborative effort involving experts and practitioners from diverse fields.

The *KBOK™ Guide* is a comprehensive yet easily accessible framework for managing workflows with Kanban. It includes practical examples of Kanban implementation using popular IT tools, helping readers apply the methodology in their organizations. The guide also covers how Kanban integrates with other Agile frameworks such as Scrum, DevOps, OKRs, and Lean. Recommendations about how Artificial Intelligence can be used to increase productivity in Kanban workflows are also included in the *KBOK™ Guide*.

The *KBOK™ Guide* serves as a resource for both experienced Kanban practitioners and professionals new to workflow management. It's also suitable for those with no prior Kanban experience. The widespread adoption of the *KBOK™ Guide* framework standardizes how Kanban is applied to workflows globally and significantly helps organizations improve their overall productivity and return on investment.

Kanbanstudy
Targeting success

ISBN 979-8-9898916-1-0
9 798989 891610