**Whitepaper**

**Title: Dynamic Kanban Task Management System with Quantitative Prioritization and Adaptive Resource Allocation**

Hamza Khan
Software Engineer, NUCES FAST University
Email: hamzaskhaan@gmail
Website: www.hamzaskhan.com
Date: November 10, 2024

---

## Abstract

In agile project management, maintaining focus on high-impact tasks while effectively managing resource limitations and dynamically evolving priorities is essential for efficient and successful sprint execution. This paper introduces a novel Kanban-based system designed to calculate a quantitative prioritization for tasks based on impact, effort, customer value, and business alignment, with an additional dynamic priority adjustment for overdue tasks. The system is implemented as a Python program that simulates resource allocation, progress tracking, and adaptive prioritization over customizable sprints, thus offering a practical approach to optimizing work-in-progress (WIP) within agile sprints. This solution provides a flexible, resource-conscious Kanban approach that allows project managers to manage tasks and resources effectively while adapting to changing project requirements.

---

## Introduction

Kanban boards are widely used in agile project management to visualize and manage workflow. While Kanban typically allows continuous, flexible task prioritization, agile teams often face challenges when it comes to prioritizing tasks that yield the highest value and managing work-in-progress (WIP) limits. This whitepaper presents a Kanban-based Python program designed to tackle these issues with a structured approach to task prioritization, dynamic resource allocation, and sprint management. Key features include:

1. **Quantitative prioritization**: Each task's priority is calculated based on weighted parameters.
2. **Dynamic WIP limits**: Work is limited based on sprint capacity, preventing resource overload.
3. **Customizable sprint lengths**: Sprints can be set to 2 or 4 weeks to accommodate varying project timelines.
4. **Adaptive resource management**: Resources are assigned to tasks in order of priority, with adaptive adjustments for overdue tasks.

---

## Methodology

This system is implemented in Python using an object-oriented approach, encapsulating all Kanban elements, such as **tasks**, **resources**, **sprints**, and **the Kanban board**, into distinct classes with clearly defined attributes and methods.

### Key Elements

1. **Quantitative Task Prioritization**
   To determine task priority, the program applies a formula to compute a **priority score** based on the following parameters, each rated on a scale from 0 to 5:
   - **Impact**: The potential impact of the task on the project's overall success.
   - **Effort**: The estimated effort required to complete the task, inversely affecting priority.
   - **Customer Value**: The direct benefit of the task to customer satisfaction or end-user needs.
   - **Business Alignment**: The task's alignment with strategic business goals.

   The formula used for priority score calculation is:

$$\mathrm{PS} = (0.4 \times \mathrm{I}) + (0.3 \times \mathrm{CV}) + (0.2 \times \mathrm{BA}) - (0.1 \times \mathrm{E}) + (\mathrm{OP})$$

Where,
PS = Priority Score
I = Impact

CV = Customer Value
BA = Business Alignment
E = Effort
OP = Overdue Penalty

The **overdue penalty** is dynamically added when a task exceeds its expected timeline, calculated as `0.05 * overdue_days`, ensuring overdue tasks rise in priority.

2. **Dynamic Resource Allocation**
   Resources are defined with an **efficiency factor** (ranging from 0 to 1), impacting how quickly each resource completes tasks. When tasks are assigned, resource efficiency is considered in calculating task duration, with higher-efficiency resources completing tasks faster. The program allocates resources to the highest-priority task available, ensuring that high-value work is prioritized.
3. **WIP Limits and Sprint Backlogs**
   The program enforces a **WIP limit** to avoid overloading resources. Sprints are initiated with a selected backlog of high-priority tasks up to the WIP limit. Tasks exceeding the WIP limit remain in the product backlog, awaiting future sprints.
4. **Customizable Sprint Length**
   The system allows project managers to set sprint durations, with 2-week or 4-week options, aligning with industry standards and project needs.

---

## Program Structure

The system comprises several classes to implement the Kanban process:

1. **Task**:
   - Stores task properties such as title, impact, effort, customer value, business alignment, estimated completion days, and priority score.
   - Calculates a **priority score** using the formula above, and updates it if a task becomes overdue.
2. **Resource**:
   - Represents a team member or resource with a unique efficiency level.
   - Tracks current task assignment and remaining task duration, dynamically adjusting based on resource efficiency.
3. **Sprint**:
   - Represents a sprint with customizable duration and a WIP limit.
   - Manages task assignments, dynamically allocates resources based on priority, and tracks day-to-day progress.
4. **Kanban**:
   - The main Kanban board, managing the product backlog and sprints.
   - Adds tasks to the backlog, initiates sprints with high-priority tasks, and handles sprint progression.

---

## Execution Flow

1. **Task Creation and Prioritization**:
   - Tasks are created with assigned values for impact, effort, customer value, and business alignment.
   - Each task's priority score is calculated and stored.
2. **Resource Assignment**:
   - Resources are created with unique efficiency ratings, impacting their task completion rate.
   - Tasks are assigned based on priority, ensuring efficient resources focus on high-priority tasks.
3. **Sprint Initialization**:
   - A sprint is created with a specified duration and a WIP limit.
   - High-priority tasks from the product backlog are added to the sprint backlog, up to the WIP limit.
4. **Dynamic Priority Adjustment**:
   - Each day, tasks' priority scores are updated if they become overdue, with an added penalty, ensuring delayed tasks get attention.
5. **Resource Progression and Task Completion**:
   - Each resource works on assigned tasks daily, reducing remaining days for task completion.
   - Once a task is completed, the resource is freed and re-assigned to the next highest-priority task.

---

### Example Usage

In the Python program, the following example code demonstrates initializing a Kanban board, creating tasks, adding them to the backlog, and starting a sprint.

```
if __name__ == "__main__":
    resources = [Resource("Alice", 1.0), Resource("Bob", 0.8), Resource("Charlie", 0.6)]
    kanban_board = Kanban(resources, wip_limit=3)
    kanban_board.add_to_backlog(Task("Design UI", 5, 2, 4, 3, 5))
    kanban_board.add_to_backlog(Task("Backend API", 4, 3, 5, 5, 8))
    kanban_board.add_to_backlog(Task("User Testing", 3, 2, 4, 4, 6))
    kanban_board.add_to_backlog(Task("Documentation", 2, 3, 2, 4, 7))
    sprint = kanban_board.start_sprint(duration_weeks=2)
    kanban_board.progress_sprint(sprint)
```

---

### Discussion

The proposed system enables project managers to maintain a structured approach to task prioritization and sprint planning, with a focus on high-impact tasks and resource optimization. Key advantages include:

- **Data-driven task prioritization**: Quantitative priority scores enable consistent and objective decision-making.
- **Dynamic adaptability**: Tasks can be reprioritized based on real-time factors like overdue status.
- **Resource efficiency**: Resources are assigned tasks in order of priority, maximizing output.

### Future Enhancements

Future improvements could include:

- **Integration with machine learning**: To predict task durations and priority scores based on historical data.
- **Automated sprint review and feedback loops**: Allowing managers to adjust WIP limits and resource allocation based on completed sprints.
- **User interface**: A UI for ease of use and real-time visualization of task progress and resource allocation.

---

### Conclusion

This Kanban-based Python program provides a dynamic, quantitative approach to task prioritization and sprint management, empowering project managers to manage tasks and resources more effectively in an agile setting. By integrating data-driven prioritization with real-time resource allocation and WIP limits, this system addresses common project management challenges, enabling teams to stay focused on high-impact work and meet project goals efficiently.

---

This whitepaper serves as a theoretical guide for project managers looking to implement a quantitative prioritization model in their Kanban workflows.