Agile Priority Index: A Formula-Based Approach For Feature Prioritization

Saikumar V L

Abstract- *In the fast-paced and ever-changing world of software development, effectively deciding which features to work on first is crucial to ensure that the most valuable and impactful functionalities are delivered to users in a timely manner. This paper introduces the "Agile Priority Index," a fresh approach that uses a formula to help Agile software development teams prioritize features. The formula combines important factors such as feature value, priority, effort, complexity, resource skillset, and risk to calculate the Agile Priority Index. By using this index, development teams can make objective and collaborative decisions about feature prioritization, leading to efficient resource allocation and increased customer satisfaction. The proposed Agile Priority Index provides an adaptable and data-driven solution to improve decision-making processes in Agile project management, promoting successful and customer-centric software development practices.*

I. INTRODUCTION

The Agile Priority Index (API) is a formula-based method designed to prioritize features efficiently in Agile software development.

By integrating multiple parameters such as feature value, priority, and complexity, API provides an objective ranking system for development teams.

Utilizing API can optimize resource allocation and enhance customer satisfaction through the delivery of high-value features.

II. GLOSSARY

2.1 Agile

An iterative software development methodology that values customer collaboration, adaptability, and incremental product delivery.

2.2 Feature Prioritization

The process of assessing and ranking software features based on their significance and impact to determine development order.

2.3 Agile Priority Index (API)

A metric derived from a formula integrating feature value, priority, effort, complexity, resource skillset, and risk to prioritize software features in Agile projects.

2.4 Effort

A numerical value representing the estimated resources and time required for feature implementation.

2.5 Complexity

A measure of the intricacy and difficulty involved in developing a particular software feature.

2.6 Resource Skillset

An evaluation of the expertise level of the resources assigned to work on a feature, ranging from Junior to Senior.

2.7 Risk

A numerical value indicating the potential challenges and uncertainties associated with implementing a specific software feature.

2.8 Iterative Development

The process of repeatedly refining and enhancing software features through multiple cycles to deliver frequent, incremental improvements.

2.9 Customer-Centric

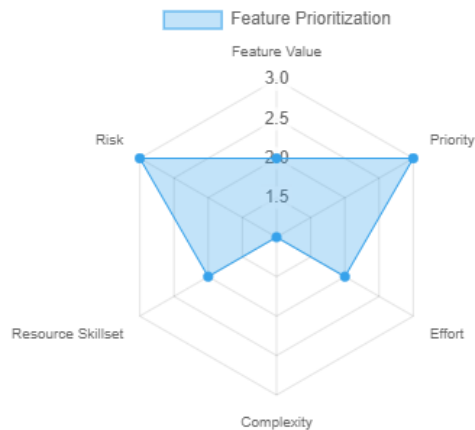A development approach focused on meeting customer needs and delivering value to enhance user satisfaction.

Fig 1.1 – API sample value spider chart

III. BENEFITS OF IMPLEMENTING API

The Agile Priority Index (API) formula serves as a valuable aid in Agile project management, providing development teams with a systematic approach to prioritize software features.

3.1 Data-Driven Decision Making

The API formula introduces objectivity into the feature prioritization process. By assigning numerical values to parameters, the formula facilitates data-driven decision-making. This aids in minimizing subjective biases and ensures that prioritization is based on a well-defined methodology. Development teams can confidently allocate resources to high API-rated features, resulting in a streamlined development pipeline.

3.2 Effective Resource Allocation

With the API formula at their disposal, Agile teams can allocate resources more effectively. By identifying features with higher API values, teams can focus on implementing functionalities that offer maximum value and impact to end-users. This resource optimization reduces waste, accelerates development cycles, and enhances the team's productivity.

3.3 Enhanced Customer Satisfaction

The API formula's ability to prioritize features based on their value to end-users ensures that customer-centric development takes precedence. High API-rated features are given priority, leading to quicker delivery of functionalities

that align with user needs and preferences. This approach fosters higher customer satisfaction and loyalty, crucial for achieving business success.

3.4 Iterative Improvement

As development progresses, the API formula can be utilized iteratively to reevaluate feature prioritization. As new information becomes available, teams can update parameter values and recalculate API scores, allowing for agile adaptation to changing project dynamics. This iterative approach ensures that the most up-to-date and impactful features remain in focus throughout the project lifecycle.

IV. CALCULATION

4.1 Formula

$$API = (f * p) / (R * r * e)$$

1. The feature value (f) and priority (p) are subjective measures provided by stakeholders or the development team. Feature value represents the importance or impact of the feature, while priority reflects its relative importance compared to other features.
2. The resource skillset (R_skillset) and complexity (C) are objective measures. Resource skillset represents the expertise   level of the resource (Junior, Intermediate, or Senior), and complexity indicates the complexity of the feature (Less Complex, Intermediate, or More Complex).
3. The resource skillset availability / complexity (R) combines the resource's skillset level with the complexity of the feature to determine the suitability of the resource for the task.
4. The risk factor (r) allows teams to consider the level of risk associated with implementing a particular feature.
5. The effort (e) represents the estimated effort required to complete the feature.

By plugging in the appropriate numerical values for each parameter into the formula, we can calculate the API for each feature, facilitating better feature prioritization in Agile development.

4.2 Parameter Values

f: Feature Value - *Values: 1 (Low), 2 (Medium), 3 (High)*
p: Priority - *Values: 1 (Low), 2 (Medium), 3 (High)*
R: Resource skillset availability / Complexity
   a) Resource skillset level multiplied by the complexity level of the feature.

b) R represents the combined effect of the resource's skillset and the complexity of the feature.
c) Values for Resource skillset (R_skillset): 1 (Junior), 2 (Intermediate), 3 (Senior)
d) Values for Complexity (C): 1 (Less Complex), 2 (Intermediate), 3 (More Complex)

R = *R_skillset* / C

e: Effort - *Values: 1 (Low), 2 (Medium), 3 (High)*
r: Risk - *Values: 1 (Low), 2 (Medium), 3 (High*

4.3 Illustration

Example Features: Feature A, Feature B
Parameter Values:

Feature Value (f):

Feature A: 3 (High)
Feature B: 2 (Medium)

Priority (p):

Feature A: 2 (Medium)
Feature B: 3 (High)

Effort (e):

Feature A: 2 (Medium)
Feature B: 1 (Low)

Complexity (C):

Feature A: 3 (High)
Feature B: 2 (Medium)

Resource Skillset (R_skillset):

Feature A: 2 (Intermediate)
Feature B: 3 (Senior)

Risk (r):

Feature A: 2 (Medium)
Feature B: 1 (Low)

Calculating the API Value for Each Feature:

For Feature A:
API = (f * p) / ([R_skillset / C] * r * e)
API = (3 * 2) / (2 / 3 * 2 * 2)
API = 6 / 2.67
API = 2.25

For Feature B:
API = (f * p) / ([R_skillset / C] * r * e)
API = (2 * 3) / (3 / 2 * 1 * 1)
API = 6 / 1.5
API = 4

Based on the calculated API values, the priority consideration for the two features is as follows:

Feature A: **API Value = 2.25**
Feature B: **API Value = 4**

Since a higher API value indicates higher priority, Feature B has a higher API value and, therefore, higher priority compared to Feature A.

V. NEED OF RESOURCE SKILLSET PARAMETER

Resource skillset is vital in requirement prioritization, allocating tasks to experts for efficient development and risk management. It optimizes resource use, promotes collaboration, and aligns with stakeholder expectations for successful feature delivery.

Table 4.1 – API sample calculation table

Feature	Feature Value (f)	Priority (p)	Resource Skillset (R_skillset)	Complexity (C)	Risk (r)	Effort (e)	R (R_skillset / C)	API
A	3 (High)	2 (Medium)	2 (Intermediate)	3 (High)	2	2 (Medium)	0.67	**2.25**
B	2 (Medium)	3 (High)	3 (Senior)	2 (Medium)	1	1 (Low)	1.5	**4**

VI. INSIGHT CATALYST

a) If **(R_skillset / C) < 1**: For each major role in the development team (e.g., junior, intermediate), create specific "spike" values for the corresponding parameters. Assign higher weights to certain parameters to reflect the expertise level.

b) If **(R_skillset / C) = 1:** Create "spikes" if needed based on specific considerations, even though the resource's skillset is at an intermediate level.

c) If **(R_skillset / C) > 1**: No "spikes" required, and the standard API formula is used as is.

Please note that, $R = R\_skillset / C$

VII. OBJECTIVE - SUMMARY

The primary aim of utilizing the Agile Priority Index (API) is to establish a measurable metric for feature prioritization in software development projects. By integrating various parameters, including feature value, priority, resource skillset, complexity, risk, and effort, the API enables data-driven decision-making, and optimizing resource allocation for efficient and effective development and delivery of valuable features in Agile projects.

REFERENCES

[1] MoSCoW Prioritization - link
[2] Weighted Shortest Job First - https://scaledagileframework.com/wsjf/
[3] Requirements Prioritization - link