**MoSCoW Technique / Kano Model**

Tool: GitHub

Gather User Feedback

 **Survey/Feedback Questions**:

* "What features do users find most valuable?"
* "What features are missing or need improvement?"
* "How do users feel about the UI/UX?"
* "What bugs or recurring issues have users reported?"
* "What new features or improvements do users suggest?"

 **Collect Feedback**: Use tools like:

* **Surveys** (Google Forms, Typeform)
* **Customer reviews** (on GitHub)
* **Support tickets or forums**

 **Analyze Feedback**: Review all the gathered feedback to identify common themes, missing features, and improvements. Categorize feedback into feature requests, UI/UX concerns, bugs, and suggestions.

**Organize Features Based on Feedback**

 **Feature Listing**: From the feedback, compile a list of features to prioritize. For example, if feedback suggests AI code reviews, integration with more CI/CD tools, or security improvements, list those as features.

 **Categorize**: Group features logically. For example:

* **Collaboration** (e.g., real-time code editing, code reviews)
* **UI/UX Improvements** (e.g., customizable dashboards)
* **Security** (e.g., 2FA, SSO)
* **Performance** (e.g., faster code search, CI/CD integrations)

**Implement MoSCoW Model**

| Feature | Priority |

|-------------------------------------|--------------|

| Enhanced code search | Must-Have |

| AI-assisted code review | Should-Have |

| Customizable dashboards | Could-Have |

| Integration with CI/CD tools | Could-Have |

| Enhanced security features (2FA) | Must-Have |

| Advanced analytics and reporting | Could-Have |

**Implement Kano Model**

| Feature | Category | Satisfaction Impact |

|-------------------------------------|-----------------|---------------------|

| Enhanced code search | Performance | High |

| AI-assisted code review | Delighter | High |

| Customizable dashboards | Delighter | Moderate |

| Integration with CI/CD tools | Performance | Moderate |

| Enhanced security features (2FA) | Basic Need | High |

**Combine MoSCoW and KANO Models for Prioritization**

 **Review Both Models**: Compare features in both MoSCoW and KANO models to validate priority. For example, features that are both **Must-Have** (MoSCoW) and **Basic Needs** (KANO) should be implemented immediately.

 **Cross-Reference**: Create a final prioritized list based on both models.

* For instance, **Enhanced security features (2FA)** may be both a **Must-Have** (MoSCoW) and a **Basic Need** (KANO), so it should be the top priority.