Feature Requests Policy - Submission & Evaluation

Document Information

- **Policy Type**: Product Development and Feature Management
- **Version**: 2.4
- **Effective Date**: January 2025
- **Review Date**: July 2025
- **Owner**: Product Management Team

Feature Request Submission Process

Idea Submission Guidelines

Feature requests can be submitted through our product portal, customer support channels, or directly through in-app feedback mechanisms. All submissions require detailed descriptions including use case scenarios, expected benefits, and target user groups.

Submission categories include new features, enhancements to existing functionality, integration requests, and user experience improvements. Each submission receives a unique tracking number and acknowledgment within 24 hours of receipt.

Documentation Requirements

Complete feature requests include problem statements, proposed solutions, success metrics, and business justification. Technical requirements should specify platform compatibility, performance expectations, and security considerations.

Visual mockups, workflow diagrams, and user journey maps enhance request evaluation and increase implementation likelihood. Supporting documentation including market research, competitor analysis, and user feedback strengthens submission quality.

Community Voting and Feedback

Upvoting System and Mechanics

Our community voting platform allows customers to upvote feature requests with weighted voting based on subscription tier and account tenure. Premium customers receive 3 votes per request while standard customers receive 1 vote.

Voting periods are continuous with quarterly tallies for roadmap planning. Customers can change votes and add comments with voting activity tracked for transparency and community engagement metrics.

Feedback Collection and Analysis

Community feedback includes detailed comments, use case examples, and implementation suggestions from diverse user perspectives. Feedback analysis identifies common themes, technical constraints, and market demand patterns.

Customer advisory boards provide structured feedback sessions with representative user groups for complex feature evaluation. Advisory feedback includes detailed requirement analysis and implementation priority recommendations.

Evaluation and Prioritization

Priority Scoring Framework

Feature requests are evaluated using a comprehensive scoring system considering market demand, technical feasibility, strategic alignment, and resource requirements. Scoring criteria include customer impact, revenue potential, and competitive advantage.

Priority scores are recalculated quarterly with updated market conditions, technical capabilities, and business priorities. High-priority features receive detailed feasibility analysis and resource allocation planning.

Internal Review Process

Technical feasibility assessments include architecture reviews, security evaluations, and performance impact analysis. Engineering teams provide implementation estimates and technical risk assessments for prioritized features.

Product management reviews include market analysis, competitive positioning, and strategic alignment evaluation. Cross-functional review teams ensure comprehensive evaluation covering technical, business, and user experience perspectives.

Roadmap Planning and Communication

Quarterly Roadmap Development

Product roadmaps are developed quarterly with input from customer feedback, market analysis, and strategic business objectives. Roadmap planning includes feature prioritization, resource allocation, and timeline estimation.

Roadmap transparency includes public sharing of planned features, development timelines, and progress updates. Customers receive quarterly roadmap presentations with Q&A sessions and detailed feature explanations.

Timeline Communication and Updates

Feature development timelines are communicated through roadmap publications, customer newsletters, and product update announcements. Timeline updates include progress reports, scope changes, and delivery date adjustments.

Delayed features receive detailed explanations including technical challenges, resource constraints, and revised timeline estimates. Customers are notified of significant timeline changes with alternative solutions and workaround suggestions.

Beta Testing and Early Access

Beta Program Participation

Beta testing programs provide early access to new features for select customers with structured feedback collection and bug reporting processes. Beta participants are chosen based on feature relevance, technical expertise, and feedback quality history.

Beta testing includes dedicated support channels, regular feedback sessions, and direct communication with development teams. Beta participants receive priority consideration for related feature requests and enhanced product influence.

Prototype Access and Feedback

Advanced prototypes are shared with key customers for early feedback and validation testing. Prototype access includes detailed documentation, usage guidelines, and structured feedback collection processes.

Prototype feedback directly influences final feature design and implementation decisions. Customers providing valuable prototype feedback receive recognition and early access to production releases.

Request Lifecycle Management

Status Tracking and Communication

Feature request status categories include Submitted, Under Review, Approved, In Development, Testing, and Released. Status updates are communicated through the product portal, email notifications, and customer dashboard updates.

Detailed status information includes development progress, testing results, and expected release timelines. Customers can track their submitted requests with real-time status updates and milestone notifications.

Duplicate Request Management

Duplicate feature requests are identified through automated matching and manual review processes. Duplicate requests are merged with original submissions to consolidate voting and feedback.

Customers submitting duplicate requests are notified of existing submissions with options to add additional context or vote for existing requests. Duplicate consolidation ensures accurate demand measurement and prevents vote fragmentation.

Rejection and Alternative Solutions

Request Rejection Criteria

Feature requests may be rejected due to technical limitations, strategic misalignment, resource constraints, or low market demand. Rejection decisions include detailed explanations and alternative solution recommendations.

Rejected requests remain in the system for future consideration with status updates when circumstances change. Customers can resubmit rejected requests with additional justification or modified requirements.

Alternative Solution Recommendations

Rejected features often have alternative solutions including existing functionality, third-party integrations, or workaround procedures. Alternative recommendations include detailed implementation guides and support resources.

Custom development options are available for enterprise customers with specific requirements that don't align with general product direction. Custom solutions include dedicated development resources and specialized support agreements.

Development Process and Timeline Management

Feasibility Review and Technical Assessment

Technical feasibility reviews are conducted by senior engineering staff with expertise in relevant system components and architecture. Reviews include performance impact analysis, security considerations, and integration complexity assessment.

Feasibility assessments provide detailed implementation estimates including development time, testing requirements, and resource allocation needs. Complex features receive multi-phase development planning with milestone-based delivery approaches.

Development Timeline Communication

Feature development timelines are communicated through multiple channels including product roadmaps, customer newsletters, and dedicated feature announcement emails. Timeline estimates include development phases, testing periods, and deployment schedules.

Timeline updates are provided bi-weekly for features in active development with detailed progress reports and milestone achievements. Customers receive immediate notification of significant timeline changes with explanations and revised delivery estimates.

Release Management and Deployment

Changelog and Release Documentation

Comprehensive changelogs document all feature releases with detailed descriptions, usage instructions, and compatibility information. Release notes include feature benefits, implementation details, and migration guides for existing customers.

Documentation includes video tutorials, setup guides, and frequently asked questions to support customer adoption. Release documentation is available in multiple formats including web-based guides, PDF downloads, and interactive tutorials.

Phased Release Strategy

New features are released using phased deployment strategies starting with limited customer groups and gradually expanding to all users. Phased releases enable real-world testing and feedback collection before full deployment.

Release phases include internal testing, beta customer deployment, and general availability with monitoring and feedback collection at each stage. Phased releases reduce risk and ensure feature stability before widespread adoption.

Community Engagement and Feedback

Customer Survey and Input Collection

Regular customer surveys collect feedback on feature preferences, usage patterns, and satisfaction levels. Survey results influence development priorities and resource allocation decisions.

Targeted surveys are conducted for specific feature categories with detailed user experience analysis and improvement recommendations. Survey feedback is integrated into product planning cycles and communicated back to participants.

Product Advisory Board Input

Customer advisory boards provide structured input on feature development priorities and strategic product direction. Advisory board members represent diverse customer segments and provide detailed feedback on proposed features.

Advisory board meetings include feature demonstrations, development progress updates, and strategic product discussions. Board recommendations are formally documented and integrated into product planning processes.

Testing and Quality Assurance

Beta Testing Coordination

Beta testing programs are carefully managed with selected customers representing diverse use cases and technical environments. Beta testing includes structured feedback collection, bug reporting, and performance monitoring.

Beta test coordination includes regular check-ins, feedback sessions, and direct communication with development teams. Beta participants receive priority support and early access to feature updates and improvements.

Quality Assurance and User Acceptance

Comprehensive quality assurance testing covers functionality, performance, security, and user experience across all supported platforms. QA testing includes automated test suites and manual testing scenarios.

User acceptance testing involves real customer scenarios with actual data and workflows. UAT feedback directly influences final feature adjustments and release readiness decisions.

Backlog Management and Planning

Feature Backlog Organization

Product backlogs are organized by feature category, priority level, and estimated development effort. Backlog items include detailed specifications, acceptance criteria, and success metrics.

Backlog grooming sessions occur weekly with cross-functional teams reviewing priorities, updating estimates, and refining requirements. Backlog transparency includes customer visibility into planned features and development progress.

Sprint Planning and Execution

Development sprints are planned with clear objectives, deliverables, and success criteria. Sprint planning includes resource allocation, dependency management, and risk assessment.

Sprint reviews include demonstration of completed features, stakeholder feedback, and iteration planning. Sprint retrospectives identify process improvements and development efficiency enhancements.

Notification and Alert Systems

Development Progress Notifications

Customers receive automated notifications for features they've voted on or submitted, including development start, milestone completion, and release announcements. Notification preferences can be customized by feature category and update frequency.

Progress notifications include detailed development updates, testing results, and expected release timelines. Customers can track feature progress through personalized dashboards and email summaries.

Feature Release Alerts

Feature release alerts are sent to all customers with detailed information about new functionality, setup instructions, and benefits. Release alerts include links to documentation, video tutorials, and support resources.

Targeted alerts are sent to customers who specifically requested released features with personalized messaging and implementation guidance. Release alerts can be customized based on customer preferences and subscription levels.

Stakeholder Communication and Involvement

Cross-Functional Collaboration

Feature development involves collaboration between engineering, product management, design, and customer success teams. Cross-functional communication ensures comprehensive feature development and customer-focused outcomes.

Regular stakeholder meetings include progress updates, challenge discussions, and decision-making sessions. Stakeholder involvement extends throughout the development lifecycle from conception to release.

Customer Success Integration

Customer success teams provide valuable input on feature adoption, usage patterns, and customer satisfaction. Success team feedback influences feature refinements and customer onboarding processes.

Success team involvement includes customer communication, adoption support, and feedback collection. Success teams serve as customer advocates throughout the development process.

Popular Feature Tracking and Analysis

Feature Usage Analytics

Released features are monitored using comprehensive analytics including adoption rates, usage patterns, and customer satisfaction metrics. Analytics data informs future development decisions and feature improvements.

Usage analytics include detailed user behavior analysis, feature interaction patterns, and performance metrics. Analytics insights are shared with development teams for continuous improvement planning.

Customer Satisfaction Measurement

Feature satisfaction is measured through surveys, usage metrics, and customer feedback collection. Satisfaction measurements inform iteration planning and feature enhancement priorities.

Regular satisfaction reviews include customer interviews, focus groups, and detailed feedback analysis. Satisfaction data is used to identify successful features and areas requiring improvement.

Release Timeline and Delivery

Delivery Commitment Management

Feature delivery commitments are made carefully with realistic timeline estimates and clear scope definitions. Delivery commitments include contingency planning for technical challenges and resource constraints.

Commitment updates are communicated proactively with detailed explanations for any changes. Customers receive regular progress reports and revised delivery estimates when necessary.

Post-Release Support and Iteration

Released features receive ongoing support including bug fixes, performance improvements, and user experience enhancements. Post-release support includes dedicated engineering resources and customer success involvement.

Feature iterations are planned based on customer feedback, usage analytics, and identified improvement opportunities. Iterative improvements are communicated through regular product updates and enhancement announcements.

Community Feedback and Engagement

Community Platform Management

Our community platform serves as the central hub for feature discussions, feedback collection, and user collaboration. Community members can participate in feature discussions, share use cases, and provide implementation feedback.

Community moderation ensures constructive discussions with clear guidelines for respectful communication and productive feedback. Moderators facilitate discussions, answer questions, and escalate technical issues to appropriate teams.

User-Generated Content and Ideas

Community members can submit detailed feature proposals including mockups, user stories, and implementation suggestions. User-generated content is evaluated for feasibility and integration into official development planning.

Outstanding community contributions receive recognition through featured posts, contributor badges, and direct engagement with product teams. Community-generated ideas that become features are acknowledged in release announcements and documentation.

Feedback Collection and Analysis

Structured Feedback Criteria

Feedback collection follows structured criteria including use case descriptions, user impact assessments, and implementation priority rankings. Structured feedback enables consistent evaluation and comparison across different feature requests.

Feedback categories include functionality requests, user experience improvements, integration needs, and performance enhancements. Each category has specific evaluation criteria and response timelines.

Multi-Channel Feedback Integration

Feedback is collected through multiple channels including community forums, customer support interactions, sales team input, and user research sessions. Multi-channel feedback is consolidated for comprehensive analysis and priority assessment.

Feedback integration includes automated sentiment analysis, topic clustering, and trend identification. Integrated feedback reports provide comprehensive views of customer needs and market demands.

Voting and Ranking Systems

Community Voting Mechanisms

Community voting uses weighted systems based on user engagement, subscription level, and expertise in relevant feature areas. Voting mechanisms include upvoting, detailed comments, and use case sharing.

Voting periods are continuous with quarterly analysis for development planning. Community members can track voting results and see how their input influences product decisions.

Feature Ranking and Prioritization

Feature ranking combines community votes, technical feasibility, strategic alignment, and resource requirements. Ranking algorithms ensure balanced consideration of customer demand and business priorities.

Regular ranking reviews include community feedback, market analysis, and competitive positioning. Ranking transparency helps customers understand development priorities and decision-making processes.

Public Roadmap and Transparency

Roadmap Publication and Updates

Public roadmaps are published quarterly with detailed feature descriptions, development timelines, and progress indicators. Roadmap updates include scope changes, timeline adjustments, and completion status.

Roadmap transparency includes explanations of development decisions, technical challenges, and resource allocation. Customers can track feature progress and understand development priorities.

Community Input on Roadmap Planning

Community members can provide input on roadmap priorities through surveys, focus groups, and feedback sessions. Community input influences quarterly roadmap reviews and priority adjustments.

Roadmap planning sessions include community representatives, customer advisory board members, and key stakeholders. Community input is formally documented and integrated into planning processes.

Idea Management and Collaboration

Idea Submission and Collaboration

Community members can submit ideas collaboratively with multiple contributors and iterative refinement. Collaborative idea development includes discussion threads, document sharing, and collective editing.

Idea collaboration tools include commenting systems, document versioning, and contributor attribution. Collaborative ideas receive enhanced visibility and development consideration.

Idea Comments and Discussion

Feature ideas include commenting systems for detailed discussion, clarification, and enhancement suggestions. Discussion moderation ensures productive conversations and actionable feedback.

Comment threading enables organized discussions with topic-specific conversations and expert input. Discussion summaries are provided to development teams for comprehensive understanding.

Rejected Ideas and Alternative Solutions

Rejection Communication and Reasoning

Rejected ideas receive detailed explanations including technical limitations, strategic misalignment, or resource constraints. Rejection communications include alternative solutions and workaround suggestions.

Rejection decisions are documented with rationale and future consideration possibilities. Rejected ideas remain accessible for reference and potential reconsideration when circumstances change.

Alternative Solution Development

Rejected features often have alternative approaches including existing functionality, third-party integrations, or custom development options. Alternative solutions include implementation guides and support resources.

Alternative solution development includes community collaboration and customer success team involvement. Alte

Contact Information

- **Feature Requests**: features@company.com
- **Product Feedback**: feedback@company.com
- **Beta Program**: beta@company.com
- **Product Portal**: portal.company.com

Policy Updates

This policy is reviewed quarterly by the Product Management Team and updated based on customer feedback, development process changes, and market conditions. Policy changes are communicated through customer announcements and product portal updates.