Release Process Documentation

Overview

Audityzer follows a structured release process that ensures quality, stability, and proper communication with the community. This document outlines the complete release workflow from planning to deployment.

Release Types

Major Releases (x.0.0)

- **Frequency**: Quarterly
- **Content**: Breaking changes, major features, architecture updates
- **Planning**: 6-8 weeks in advance
- **Testing**: Comprehensive testing including beta releases

Minor Releases (x.y.0)

- **Frequency**: Monthly
- **Content**: New features, enhancements, non-breaking changes
- **Planning**: 2-3 weeks in advance
- **Testing**: Standard testing suite + feature validation

Patch Releases (x.y.z)

- **Frequency**: As needed
- **Content**: Bug fixes, security patches, minor improvements
- **Planning**: 1 week or immediate for critical issues
- **Testing**: Focused testing on affected areas

Hotfix Releases (x.y.z+1)

- **Frequency**: Emergency only
- **Content**: Critical security fixes, production-breaking bugs
- **Planning**: Immediate
- **Testing**: Minimal but focused testing

Release Planning

Release Calendar

```
Q1 2025:
- v1.3.0 (Major) - March 15
- v1.3.1 (Patch) - March 30
- v1.3.2 (Patch) - April 15

Q2 2025:
- v1.4.0 (Minor) - April 30
- v1.4.1 (Patch) - May 15
- v1.5.0 (Minor) - May 30
- v2.0.0 (Major) - June 30
```

Feature Planning

```javascript
// Release planning configuration
const releasePlan = {
  "v1.3.0": {
    type: "major",
    targetDate: "2025-03-15",
    features: [
      "cross-chain-bridge-testing",
      "ai-vulnerability-detection-v2",
      "real-time-dashboard",
      "community-portal"
    ],
    breakingChanges: [
      "api-v2-migration",
      "config-format-update"
    ],
    dependencies: [
      "security-audit-completion",
      "performance-benchmarks"
    ]
  },

  "v1.4.0": {
    type: "minor",
    targetDate: "2025-04-30",
    features: [
      "marketing-automation",
      "enhanced-monitoring",
      "plugin-system"
    ],
    breakingChanges: [],
    dependencies: [
      "v1.3.0-stable"
    ]
  }
};
```

Release Workflow

1. Pre-Release Phase

Feature Freeze

```
# Create release branch
git checkout unified-main
git pull origin unified-main
git checkout -b release/v1.3.0

# Update version numbers
npm version 1.3.0 --no-git-tag-version

# Update changelog
npm run changelog:generate

# Commit version updates
git add .
git commit -m "chore: prepare v1.3.0 release"
```

Quality Assurance

```
# Run comprehensive test suite
npm run test:all
npm run test:security
npm run test:performance
npm run test:e2e

# Generate test reports
npm run test:coverage
npm run test:report

# Security audit
npm audit
npm run security:scan
```

Documentation Update

```
# Update documentation
npm run docs:build
npm run docs:validate

# Update API documentation
npm run api:docs

# Generate migration guide
npm run migration:guide
```

2. Release Candidate Phase

Create Release Candidate

```bash
# Tag release candidate
git tag v1.3.0-rc.1
git push origin v1.3.0-rc.1

# Build release candidate
npm run build:production
npm run package:release

# Deploy to staging
npm run deploy:staging
```

Community Testing

```javascript
// Release candidate announcement
const rcAnnouncement = {
  title: "Audityzer v1.3.0 Release Candidate Available",
  content: `
    We're excited to announce the release candidate for Audityzer v1.3.0!

    ## New Features
    - Cross-chain bridge testing for LayerZero and Stargate
    - AI-powered vulnerability detection v2
    - Real-time security dashboard
    - Community portal integration

    ## Testing Instructions
    1. Download: npm install audityzer@1.3.0-rc.1
    2. Test your workflows
    3. Report issues: https://github.com/Audityzer/audityzer/issues

    ## Feedback Deadline
    Please provide feedback by March 10, 2025
  `,
  channels: ["discord", "reddit", "twitter", "github"]
};
```

Beta Testing Program

```bash
# Deploy to beta environment
npm run deploy:beta

# Invite beta testers
npm run beta:invite

# Monitor beta metrics
npm run beta:monitor
```

3. Release Phase

Final Preparations

```
# Merge release branch to main
git checkout main
git merge release/v1.3.0 --no-ff

# Create final tag
git tag v1.3.0
git push origin main --tags

# Update unified-main
git checkout unified-main
git merge main
git push origin unified-main
```

Build and Package

```
# Build production artifacts
npm run build:production

# Create distribution packages
npm run package:npm
npm run package:docker
npm run package:binary

# Sign packages
npm run sign:packages

# Upload to registries
npm publish
docker push audityzer/audityzer:1.3.0
```

Deployment

```
# Deploy to production
npm run deploy:production

# Update CDN
npm run cdn:update

# Update documentation site
npm run docs:deploy

# Notify monitoring systems
npm run monitoring:notify
```

4. Post-Release Phase

Release Announcement

```
// Release announcement template
const releaseAnnouncement = {
  title: "  Audityzer v1.3.0 Released!",
  content: `
    We're thrilled to announce the release of Audityzer v1.3.0!

    ##   What's New

    ### Cross-Chain Bridge Testing
    - LayerZero protocol testing
    - Stargate Finance integration
    - Radiant Capital support
    - Multi-chain validation

    ### AI-Powered Security Analysis
    - Enhanced vulnerability detection
    - Pattern recognition improvements
    - Automated remediation suggestions
    - 95% accuracy rate

    ### Real-Time Dashboard
    - Live security monitoring
    - Interactive threat visualization
    - Custom alert configuration
    - Performance metrics

    ### Community Portal
    - Discord integration
    - Contributor onboarding
    - Community challenges
    - Reward system

    ##   Improvements
    - 40% faster scanning performance
    - Reduced memory usage by 25%
    - Enhanced error handling
    - Better TypeScript support

    ##   Bug Fixes
    - Fixed memory leak in AI analysis
    - Resolved dashboard loading issues
    - Improved error messages
    - Fixed cross-chain sync issues

    ##   Migration Guide
    See our migration guide: https://docs.audityzer.com/migration/v1.3.0

    ##   Thank You
    Special thanks to our contributors and community members!

    ##   Download
    - npm: npm install audityzer@1.3.0
    - Docker: docker pull audityzer/audityzer:1.3.0
    - Binary: https://releases.audityzer.com/v1.3.0
  `,
  channels: ["blog", "discord", "reddit", "twitter", "linkedin"]
};
```

Monitoring and Support

```
# Monitor release metrics
npm run metrics:monitor

# Track adoption rates
npm run adoption:track

# Monitor error rates
npm run errors:monitor

# Prepare support documentation
npm run support:prepare
```

Release Automation

CI/CD Pipeline

```yaml
# .github/workflows/release.yml
name: Release Pipeline

on:
  push:
    tags:
      - 'v*'

jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '16'
      - name: Install dependencies
        run: npm ci
      - name: Run tests
        run: npm run test:all
      - name: Security scan
        run: npm run security:scan
      - name: Performance test
        run: npm run test:performance

  build:
    needs: test
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Build packages
        run: |
          npm run build:production
          npm run package:all
      - name: Upload artifacts
        uses: actions/upload-artifact@v3
        with:
          name: release-packages
          path: dist/

  publish:
    needs: build
    runs-on: ubuntu-latest
    steps:
      - name: Publish to npm
        run: npm publish
        env:
          NODE_AUTH_TOKEN: ${{ secrets.NPM_TOKEN }}
      - name: Publish Docker image
        run: |
          docker build -t audityzer/audityzer:${{ github.ref_name }} .
          docker push audityzer/audityzer:${{ github.ref_name }}
        env:
          DOCKER_USERNAME: ${{ secrets.DOCKER_USERNAME }}
          DOCKER_PASSWORD: ${{ secrets.DOCKER_PASSWORD }}

  deploy:
    needs: publish
    runs-on: ubuntu-latest
    steps:
      - name: Deploy to production
        run: npm run deploy:production
        env:
          DEPLOY_TOKEN: ${{ secrets.DEPLOY_TOKEN }}
      - name: Update documentation
        run: npm run docs:deploy
      - name: Notify monitoring
        run: npm run monitoring:notify
```

Automated Release Notes

```javascript
// scripts/generate-release-notes.js
const { Octokit } = require('@octokit/rest');

class ReleaseNotesGenerator {
  constructor(token) {
    this.octokit = new Octokit({ auth: token });
  }

  async generateReleaseNotes(fromTag, toTag) {
    const commits = await this.getCommitsBetweenTags(fromTag, toTag);
    const prs = await this.getPullRequestsBetweenTags(fromTag, toTag);

    const features = this.categorizeCommits(commits, 'feat');
    const fixes = this.categorizeCommits(commits, 'fix');
    const breaking = this.categorizeCommits(commits, 'BREAKING');

    return this.formatReleaseNotes({
      features,
      fixes,
      breaking,
      contributors: this.getContributors(commits)
    });
  }

  formatReleaseNotes({ features, fixes, breaking, contributors }) {
    return `
##   New Features
${features.map(f => `- ${f.description}`).join('\n')}

##   Bug Fixes
${fixes.map(f => `- ${f.description}`).join('\n')}

## ⚠ Breaking Changes
${breaking.map(b => `- ${b.description}`).join('\n')}

##   Contributors
${contributors.map(c => `- @${c.username}`).join('\n')}
    `;
  }
}
```

Version Management

```
// scripts/version-manager.js
class VersionManager {
  constructor() {
    this.currentVersion = require('../package.json').version;
  }

  calculateNextVersion(changeType) {
    const [major, minor, patch] = this.currentVersion.split('.').map(Number);

    switch (changeType) {
      case 'major':
        return `${major + 1}.0.0`;
      case 'minor':
        return `${major}.${minor + 1}.0`;
      case 'patch':
        return `${major}.${minor}.${patch + 1}`;
      default:
        throw new Error('Invalid change type');
    }
  }

  updateVersionFiles(newVersion) {
    // Update package.json
    const packageJson = require('../package.json');
    packageJson.version = newVersion;
    fs.writeFileSync('package.json', JSON.stringify(packageJson, null, 2));

    // Update other version files
    this.updateDockerfile(newVersion);
    this.updateHelmChart(newVersion);
    this.updateDocumentation(newVersion);
  }
}
```

Quality Gates

Pre-Release Checklist

- [ ] All tests passing (unit, integration, e2e)
- [ ] Security scan completed with no critical issues
- [ ] Performance benchmarks meet requirements
- [ ] Documentation updated and reviewed
- [ ] Breaking changes documented
- [ ] Migration guide prepared
- [ ] Changelog updated
- [ ] Version numbers updated across all files

Release Checklist

- [ ] Release branch created and tested
- [ ] Release candidate deployed and tested
- [ ] Community feedback incorporated

- [ ] Final security review completed
- [ ] Production deployment tested
- [ ] Rollback plan prepared
- [ ] Monitoring alerts configured
- [ ] Support team notified

Post-Release Checklist

- [ ] Release announcement published
- [ ] Documentation site updated
- [ ] Monitoring metrics normal
- [ ] Community feedback monitored
- [ ] Support tickets triaged
- [ ] Next release planning initiated
- [ ] Retrospective scheduled
- [ ] Lessons learned documented

Rollback Procedures

Automated Rollback

```bash
# scripts/rollback.sh
#!/bin/bash

PREVIOUS_VERSION=$1
ROLLBACK_REASON=$2

echo "Initiating rollback to version $PREVIOUS_VERSION"
echo "Reason: $ROLLBACK_REASON"

# Stop current services
npm run services:stop

# Deploy previous version
npm run deploy:version $PREVIOUS_VERSION

# Verify rollback
npm run verify:deployment

# Notify team
npm run notify:rollback "$PREVIOUS_VERSION" "$ROLLBACK_REASON"

echo "Rollback completed successfully"
```

Manual Rollback Steps

1. **Identify Issue**: Determine the scope and impact
2. **Decision Point**: Decide between hotfix or rollback
3. **Communication**: Notify team and stakeholders
4. **Execute Rollback**: Deploy previous stable version
5. **Verification**: Confirm system stability
6. **Post-Mortem**: Analyze and document lessons learned

Metrics and Monitoring

Release Metrics

```javascript
// monitoring/release-metrics.js
const releaseMetrics = {
  deploymentTime: measureDeploymentTime(),
  errorRate: measureErrorRate(),
  adoptionRate: measureAdoptionRate(),
  rollbackRate: measureRollbackRate(),
  timeToDetection: measureTimeToDetection(),
  timeToResolution: measureTimeToResolution()
};

// Key Performance Indicators
const kpis = {
  deploymentFrequency: 'weekly',
  leadTimeForChanges: '< 1 week',
  meanTimeToRecovery: '< 1 hour',
  changeFailureRate: '< 5%'
};
```

Release Dashboard

```javascript
// dashboard/release-dashboard.js
const ReleaseDashboard = () => {
  return (
    <Dashboard>
      <MetricCard
        title="Deployment Frequency"
        value="2.3 per week"
        trend="up"
      />
      <MetricCard
        title="Lead Time"
        value="4.2 days"
        trend="down"
      />
      <MetricCard
        title="MTTR"
        value="23 minutes"
        trend="stable"
      />
      <MetricCard
        title="Change Failure Rate"
        value="2.1%"
        trend="down"
      />

      <ReleaseTimeline releases={recentReleases} />
      <AdoptionChart data={adoptionData} />
      <ErrorRateChart data={errorData} />
    </Dashboard>
  );
};
```

Communication Strategy

Stakeholder Communication

```
// communication/stakeholders.js
const stakeholders = {
  internal: {
    developers: ['slack', 'email'],
    qa: ['slack', 'jira'],
    devops: ['slack', 'pagerduty'],
    management: ['email', 'dashboard']
  },

  external: {
    community: ['discord', 'reddit', 'twitter'],
    enterprise: ['email', 'support-portal'],
    partners: ['email', 'partner-portal'],
    media: ['press-release', 'blog']
  }
};
```

Communication Templates

```
## Internal Release Notification
Subject: Audityzer v1.3.0 Released to Production

Team,

Audityzer v1.3.0 has been successfully deployed to production.

**Key Changes:**
- Cross-chain bridge testing
- AI vulnerability detection v2
- Real-time dashboard

**Monitoring:**
- Error rates: Normal
- Performance: Within SLA
- User feedback: Positive

**Next Steps:**
- Monitor for 24 hours
- Collect user feedback
- Plan v1.3.1 if needed

## External Community Announcement
  Audityzer v1.3.0 is now live!

We're excited to share the latest improvements to our Web3 security platform...
```

Continuous Improvement

Release Retrospectives

```
// retrospective/template.js
const retrospectiveTemplate = {
  whatWentWell: [
    "Automated testing caught issues early",
    "Community feedback was valuable",
    "Deployment went smoothly"
  ],

  whatCouldImprove: [
    "Release notes could be clearer",
    "Beta testing period too short",
    "Documentation updates delayed"
  ],

  actionItems: [
    {
      item: "Improve release notes template",
      owner: "docs-team",
      dueDate: "next-release"
    },
    {
      item: "Extend beta testing period",
      owner: "qa-team",
      dueDate: "next-release"
    }
  ]
};
```

Process Optimization

- **Automation**: Increase automation in testing and deployment
- **Feedback Loops**: Improve feedback collection and analysis
- **Documentation**: Enhance documentation and communication
- **Training**: Provide training for team members
- **Tools**: Evaluate and adopt better tools and practices

This release process ensures that Audityzer delivers high-quality releases while maintaining community engagement and operational excellence. For questions or suggestions, please join our Discord community (https://discord.gg/audityzer).