NoCodeJam XP Weighting Logic Specification

Version 2.0 | Last Updated: January 2026

1. Purpose

This document defines the platform-wide XP calculation logic for NoCodeJam. XP is system-calculated only and cannot be manually set by authors or AI tools. This ensures fairness, consistency, and prevents gaming of the reward system.

2. Design Principles

- Outcome-Based: XP awarded for completion, not attempts
- Effort-Weighted: Time and difficulty are primary factors
- Non-Exploitable: No XP farming mechanisms
- Predictable & Fair: Transparent formula applied consistently
- Composable: Pathway XP derives from challenge XP

3. Author-Defined Inputs

Authors provide these inputs when creating challenges:

- Difficulty: Beginner / Intermediate / Advanced
- Estimated Time Box: In minutes (range: 10-240 minutes)
- Challenge Type: Build / Modify / Analyse / Deploy / Reflect

4. System-Defined Inputs

These values are controlled by the platform and cannot be changed by authors:

- XP Multipliers (difficulty and challenge type)
- XP Caps and Floors (25 XP minimum, 250 XP maximum)
- Completion Bonuses
- Anti-Abuse Limits

5. Difficulty Multipliers

Difficulty Level	Multiplier
Beginner	1.0

Created by James Jones 2026
https://github.com/ThunderOpsAI

Intermediate	1.4
Advanced	1.8

Rationale: Multipliers reward increased effort and complexity of advanced challenges while keeping beginner challenges accessible.

6. Time-Based XP Formula

Base Time XP = Estimated Minutes × 2

This formula converts time investment into XP. Each minute of estimated effort is worth 2 XP before multipliers are applied.

Time Box Constraints:

- Minimum: 10 minutes
- Maximum: 240 minutes (4 hours)
- Granularity: 5-minute increments recommended
- Validation: Time estimates must be tested by 3+ reviewers with acceptable variance of ±20%

7. Challenge Type Modifiers

Challenge Type	Modifier	Rationale
Reflect	0.8	Lower technical complexity
Analyse	1.0	Baseline difficulty
Modify	1.1	Requires code comprehension
Build	1.2	Creates new functionality
Deploy	1.3	Highest complexity, production concerns

8. Final Challenge XP Formula

Challenge XP = Base Time XP × Difficulty Multiplier × Challenge Type Modifier

Final XP is rounded to the nearest whole number and clamped between 25 XP (minimum) and 250 XP (maximum).

Rounding Rule: Round to nearest integer (0.5 rounds up)

Created by James Jones 2026
https://github.com/ThunderOpsAI

9. Worked Examples

Example 1: Beginner Build Challenge

- Difficulty: Beginner (1.0)
- Time: 30 minutes
- Type: Build (1.2)

Calculation:

1. Base Time XP = 30 × 2 = 60
2. Apply Multipliers = 60 × 1.0 × 1.2 = 72
3. Round = 72 (already whole number)
4. Clamp = 72 (within 25-250 range)

Final XP: 72

Example 2: Advanced Deploy Challenge

- Difficulty: Advanced (1.8)
- Time: 90 minutes
- Type: Deploy (1.3)

Calculation:

5. Base Time XP = 90 × 2 = 180
6. Apply Multipliers = 180 × 1.8 × 1.3 = 421.2
7. Round = 421 (nearest integer)
8. Clamp = 250 (exceeds maximum, capped at 250)

Final XP: 250

Example 3: Intermediate Reflect Challenge

- Difficulty: Intermediate (1.4)
- Time: 15 minutes
- Type: Reflect (0.8)

Calculation:

9. Base Time XP = 15 × 2 = 30
10. Apply Multipliers = 30 × 1.4 × 0.8 = 33.6
11. Round = 34 (nearest integer)
12. Clamp = 34 (within 25-250 range)

Final XP: 34

Example 4: Edge Case - Below Minimum

- Difficulty: Beginner (1.0)

Created by James Jones 2026
https://github.com/ThunderOpsAI

- Time: 10 minutes
- Type: Reflect (0.8)

Calculation:

13. Base Time XP = 10 × 2 = 20
14. Apply Multipliers = 20 × 1.0 × 0.8 = 16
15. Round = 16
16. Clamp = 25 (below minimum, raised to 25)

Final XP: 25

10. Learning Pathway XP

Base Calculation
Pathway XP is calculated as the sum of all challenge XP within that pathway.

Optional Completion Bonus
An optional +5% completion bonus may be applied at the platform level, capped at 150 XP maximum.

When Bonus Applies:

- Applied automatically to all pathways at launch
- Can be disabled for specific pathways if needed
- Platform administrators control bonus activation

Example Pathway XP Calculation:
5 challenges: 50 + 75 + 100 + 120 + 85 = 430 XP
Completion bonus: 430 × 0.05 = 21.5 → 22 XP
Total Pathway XP: 430 + 22 = 452 XP

11. Anti-Gaming Rules

These rules prevent exploitation of the XP system:

- XP Awarded Once Per Challenge: Completing the same challenge multiple times yields XP only on first completion
- No XP for Retries: Failed attempts do not award XP, only successful completion
- No Standalone Reflection XP: Reflections are required for challenge completion but do not award additional XP beyond the challenge XP
- No XP for Tool Usage Alone: Using recommended tools does not award bonus XP
- No Duplicate XP: Identical challenges (even across different pathways) award XP only once
- Plagiarism Detection: XP can be revoked if plagiarism is detected
- Minimum Engagement Time: System tracks actual time spent; speed-running below reasonable thresholds may trigger review

12. XP Visibility Rules

- Pre-Challenge: Learners see expected XP before starting
- Post-Completion: XP awarded and displayed immediately upon successful completion
- Author Restrictions: Authors cannot edit or override XP values
- Leaderboard Updates: XP updates leaderboards in real-time
- Audit Trail: All XP calculations are logged with inputs for transparency and debugging

13. XP Recalculation Policy

When Formula Changes

- Formula changes require platform administrator approval
- Notice period: 2 weeks minimum before implementation
- Communication: Announcement to all active learners

Handling Existing Learners

- Grandfathering: Learners keep XP earned under old system
- New Challenges: New formula applies to challenges completed after change date
- Optional Recalculation: Learners may opt-in to recalculation if new formula is more favorable

14. Challenge Type Decision Tree

Use this guide to classify challenges:

Created by James Jones 2026
https://github.com/ThunderOpsAI

Build: Creating something new from scratch. Example: Build a landing page, Create an API, Design a database schema

Modify: Improving or extending existing code/project. Example: Add a feature to existing app, Refactor code for performance, Extend API with new endpoint

Analyse: Evaluating, debugging, or understanding existing work. Example: Debug broken code, Review security vulnerabilities, Analyse performance bottlenecks

Deploy: Publishing or configuring for production. Example: Deploy to cloud platform, Set up CI/CD pipeline, Configure DNS and SSL

Reflect: Critically assessing learning or process. Example: Write post-project reflection, Compare approaches, Identify learning gaps

Hybrid Challenges: For challenges combining types (e.g., Build + Deploy), use the primary activity type (whichever takes more time/effort).

15. XP Health Metrics

The platform monitors these metrics to ensure XP system health:

- Average XP Per Challenge: Target range 50-150 XP
- XP Distribution by Difficulty: Ensure balanced representation
- XP Inflation Over Time: Detect if average XP increases without justification
- Outlier Detection: Flag challenges with unusually high/low XP relative to difficulty/time
- Completion Time vs. Estimate: Track if actual times align with estimates (target: ±20%)

16. Future Extensions

These features are under consideration but not yet implemented:

- Streak Bonuses: Daily/weekly completion streaks award bonus XP
- Seasonal Events: Limited-time XP multipliers during special events
- Community Challenges: Collaborative challenges with shared XP pools
- XP Multiplier Events: Platform-controlled temporary XP boosts (e.g., 1.5× weekend)
- Skill Mastery Bonus: Extra XP for demonstrating consistent skill proficiency

Note: All future extensions must maintain the core principle that XP remains system-controlled and non-exploitable.

Created by James Jones 2026
https://github.com/ThunderOpsAI

Appendix: Quick Reference

XP Formula Summary

Challenge XP = (Minutes × 2) × Difficulty Multiplier × Type Modifier
Rounded to nearest integer, clamped to [25, 250]

Quick Lookup Tables

Difficulty	Multiplier
Beginner	1.0
Intermediate	1.4
Advanced	1.8

Challenge Type	Modifier
Reflect	0.8
Analyse	1.0
Modify	1.1
Build	1.2
Deploy	1.3

Version History

- v2.0 (January 2026): Added worked examples, recalculation policy, health metrics, decision tree, time box constraints
- v1.0 (Original): Initial XP formula specification

Created by James Jones 2026
https://github.com/ThunderOpsAI