

# Invisible Work Detection: Methodology Specification

---

**Version 1.0 — 2026-01-28**

---

## Executive Summary

---

### The Problem

**Invisible work** is effort that occurs but is not tracked in Jira. It includes meetings, support requests, unplanned firefighting, ad-hoc fixes, technical debt work without tickets, and context-switching overhead. By definition, we cannot directly measure invisible work—it leaves no Jira footprint.

### The Approach

Instead of direct measurement, this methodology detects invisible work through **statistical fingerprints**—anomalous patterns in Jira data that suggest hidden work is occurring. When teams show high throughput variability, stale in-progress items, and low daily engagement with Jira, these signals collectively indicate that significant effort is happening outside the system.

### Key Design Decisions

- 1. CSS-Heavy Scoring:** The Invisible Work Risk (IWR) score uses a modified Composite Health Score formula weighted 75% toward Current State Score (CSS) and 25% toward Trajectory Score (TRS). Peer Growth Score (PGS) is explicitly excluded due to circular reasoning concerns.
- 2. 17 Indicators in 3 Tiers:** Indicators are organized into Primary (55% weight), Supporting (35%), and Contextual (10%) tiers based on their causal connection to invisible work.
- 3. Risk Framing:** Results are presented as “Invisible Work Risk Indicators” rather than definitive measurements, acknowledging the indirect nature of detection.
- 4. Higher Correlation Adjustment:** The methodology uses  $\rho = 0.45$  (vs. 0.30 for general CHS) to account for the higher inter-correlation among invisible work signals.

### Output

The methodology produces:

- **IWR Score** (0-100): Higher scores indicate higher risk of invisible work

- **90% Confidence Interval:** Quantified uncertainty
  - **Precision Category:** High/Medium/Low based on data quality
  - **Risk Label:** Low/Moderate/Typical/Elevated/High
- 

# 1. Conceptual Framework

---

## 1.0 Theoretical Foundation

Before diving into implementation details, it's important to establish the theoretical basis for invisible work detection. This methodology rests on three foundational premises:

### Premise 1: Capacity Conservation

Team capacity is finite. When visible work output decreases without staffing changes, the "missing" capacity must be going somewhere—either to invisible work or to reduced productivity. Mathematically:

$$\text{Total Capacity} = \text{Visible Work} + \text{Invisible Work} + \text{Slack}$$

If visible work and slack remain constant but output varies significantly, invisible work is the likely explanation.

### Premise 2: Signal Consistency

A single anomalous metric could have many explanations. However, when multiple independent signals align—high throughput variability, stale items, low Jira engagement, and mid-sprint additions—the probability that all are caused by factors other than invisible work becomes vanishingly small. This is the "constellation of signals" approach.

### Premise 3: Measurable Proxies

While we cannot directly observe invisible work, its effects are measurable. These effects manifest as:

- **Variability** in metrics that should be stable
- **Staleness** in items that should be actively worked
- **Gaps** in engagement that should be continuous
- **Anomalies** in workflow that should follow patterns

This methodology systematically measures these proxies and aggregates them into a risk assessment.

## 1.1 What Is Invisible Work?

Invisible work encompasses any effort expended by team members that does not appear in Jira. This includes:

Category	Examples
----------	----------

<b>Meetings &amp; Ceremonies</b>	Sprint planning, standups, retros, 1:1s, all-hands, cross-team syncs
<b>Support &amp; Escalations</b>	Customer support, internal support requests, production incidents
<b>Unplanned Firefighting</b>	Urgent bugs, infrastructure issues, security patches
<b>Context Switching</b>	Task switching overhead, re-orientation time, interruption recovery
<b>Communication</b>	Slack conversations, email threads, ad-hoc discussions
<b>Technical Debt</b>	Quick fixes, workarounds, "I'll ticket this later" work
<b>Learning &amp; Growth</b>	Training, documentation reading, skill development

These activities consume team capacity but often leave no trace in project management systems. A team with 8 engineers may have only 60% of their time available for tracked work due to invisible work burden.

## 1.2 Why Invisible Work Matters

Teams with high invisible work burden exhibit predictable symptoms:

### Operational Impact:

- Unpredictable velocity despite consistent staffing
- Missed sprint commitments without clear cause
- Chronic under-delivery vs. estimates

### Team Health Impact:

- Burnout from feeling "always busy but never done"
- Frustration when velocity metrics don't reflect actual effort
- Cynicism about planning processes that ignore invisible load

### Organizational Impact:

- Inaccurate capacity planning
- Unrealistic expectations from stakeholders
- Difficulty comparing team productivity

## 1.3 Detection Philosophy: Statistical Fingerprints

Since invisible work is untracked by definition, we cannot measure it directly. Instead, we detect it through its **statistical fingerprints**—patterns in Jira data that are difficult to explain without assuming significant untracked work.

The core insight is that invisible work manifests as **unexplained variability**:

$$\text{Observed Variability} = \text{Explainable Variability} + \text{Invisible Work Signal}$$

When throughput, cycle times, and WIP counts vary dramatically without corresponding changes in complexity, team composition, or process—something external is consuming capacity.

### Key detection heuristics:

1. **Variability Anomalies:** High variation in output, cycle times, and individual performance that can't be explained by visible factors
2. **Engagement Gaps:** Low Jira update frequency suggesting work happens elsewhere
3. **Staleness Patterns:** In-progress items not updated indicate attention is elsewhere
4. **Side-Door Entries:** Mid-sprint additions suggest work enters through unofficial channels

## 1.4 Important Limitations

This methodology measures risk indicators, not invisible work itself.

Claim	Validity
"Team A has high throughput variability"	✓ Directly measurable
"Team A likely has invisible work"	✓ Reasonable inference
"Team A has 20 hours/week of invisible work"	✗ Cannot be determined

### Known blind spots:

1. **Uniformly distributed invisible work:** If all teams have similar invisible work levels, relative comparison won't detect it
2. **Correlated visible complexity:** Some variability may be legitimate complexity that happens to correlate with invisible work periods
3. **Small teams:** Higher natural variability in small teams may produce false positives

### Recommendations for use:

- Present results as "Risk Indicators" not "Measurements"
- Encourage teams to validate signals against their actual experience
- Use as a conversation starter, not a verdict

---

## 2. Indicator Selection

### 2.1 Selection Criteria

Each of the 117 indicators in the Jira Health Assessment framework was evaluated against four criteria:

Criterion	Weight	Definition

<b>Causal Connection</b>	40%	Does invisible work directly cause this pattern?
<b>Specificity</b>	25%	Are there few alternative explanations for this pattern?
<b>Actionability</b>	20%	Does improving this indicator require addressing invisible work?
<b>Measurability</b>	15%	Can we reliably detect this pattern in Jira data?

Indicators scoring  $\geq 3.0$  (on a 1-5 scale) were selected. The full analysis is documented in `indicator-selection-matrix.md`.

## 2.2 Selected Indicators (17 Total)

### Primary Signals (7 indicators, 55% total weight)

These indicators have the strongest causal connection to invisible work:

Indicator	Score	Signal Interpretation
throughputVariability	4.55	Sprint output swings without demand changes = hidden work consuming capacity
memberThroughputVariability	4.35	Individual output varies = pulled into untracked activities
workflowStageTimeVariability	4.30	Same stages take different times = interruptions and context switching
sameSizeTimeVariability	4.20	Similar-sized items vary = hidden complexity or distractions
avgDailyUpdates	4.15	Low Jira engagement = work happening elsewhere
staleWorkItems	4.10	In-progress items not updated = attention is elsewhere
inProgressItemsVariability	3.90	WIP count fluctuates = ad-hoc invisible work

### Supporting Signals (7 indicators, 35% total weight)

These indicators provide corroborating evidence:

Indicator	Score	Signal Interpretation
closedWithoutComments	3.90	No discussion = decisions made off-platform

midSprintCreations	3.90	Mid-sprint additions = work enters through side doors
frequentUseVariability	3.90	Usage patterns vary = inconsistent Jira habits
collaborationVariability	3.55	Communication swings = off-platform discussions
staleEpics	3.45	Epics untouched = strategic work not tracked
bulkChanges	3.40	Bulk updates = batch housekeeping discipline
lastDayCompletions	3.40	End-of-sprint clustering = batched status updates

### Contextual Signals (3 indicators, 10% total weight)

These indicators add context but have lower specificity:

Indicator	Score	Signal Interpretation
zombieItemCount	3.60	Backlog zombies = visible work displaced by invisible
estimationVariability	3.25	Erratic estimates = scope discovered mid-flight
capacitySplitAcrossProjects	3.05	Multi-project = invisible work in other contexts

### 2.3 Indicators Not Included

Three indicators from the current Invisible Work dimension were removed:

Indicator	Reason for Removal
unresolvedEpicChildren	Measures data hygiene, not invisible work
sprintHygiene	Too composite; unclear mechanism to invisible work
siloeWorkItems	Better fit for Team Collaboration dimension

### 2.4 Why These Indicators Work: The Causal Model

Understanding *why* these indicators detect invisible work helps interpret results and avoid misuse.

#### The Throughput Variability → Invisible Work Link

Consider a team that completes 40 story points one sprint and 18 the next, with no changes in staffing, vacation, or planned work complexity. What explains this?

### Possible explanations:

1. **Invisible work consumed sprint 2 capacity** (meetings, support escalations, firefighting)
2. Work was harder than expected (but estimates were similar)
3. Team members were less productive (but no evidence of that)
4. Random variation (unlikely at this magnitude)

The first explanation is most parsimonious. The “missing” 22 points went somewhere—  
invisible work is the likely culprit.

### Mathematical framing:

Let  $V_t$  be throughput in period  $t$ . Without invisible work:  $V_t = \text{Capacity} \times \text{Productivity} \times \text{Complexity Factor}_t + \epsilon_t$

With invisible work  $I_t$ :  $V_t = (\text{Capacity} - I_t) \times \text{Productivity} \times \text{Complexity Factor}_t + \epsilon_t$

When  $I_t$  varies unpredictably,  $V_t$  inherits that variation—even if productivity and complexity are stable.

### The Staleness → Invisible Work Link

A work item in “In Progress” status for 10 days without updates suggests one of:

1. **The assignee is working on something else** (invisible work)
2. The item is blocked (should be flagged as such)
3. The assignee forgot to update Jira (symptom of invisible work culture)
4. The item is actually done but not closed (data hygiene issue)

Options 1 and 3 both indicate invisible work patterns. Staleness is a strong signal that tracked work is not receiving attention.

### The Engagement Gap → Invisible Work Link

Average daily Jira updates measure “continuous engagement” with the tracking system. Low engagement suggests:

1. **Work is happening elsewhere** (not in Jira)
2. Team culture de-emphasizes real-time tracking
3. Work items are too large (single update spans multiple days)

All three indicate invisible work risk. Either actual invisible work is occurring, or the culture creates conditions where invisible work can hide.

## 2.5 Indicator Weights

Weights are assigned by tier, with equal distribution within tiers:

Indicator	Tier	Weight
throughputVariability	Primary	7.86%
memberThroughputVariability	Primary	7.86%
workflowStageTimeVariability	Primary	7.86%
sameSizeTimeVariability	Primary	7.86%
avgDailyUpdates	Primary	7.86%
staleWorkItems	Primary	7.86%
inProgressItemsVariability	Primary	7.86%
closedWithoutComments	Supporting	5.00%
midSprintCreations	Supporting	5.00%
frequentUseVariability	Supporting	5.00%
collaborationVariability	Supporting	5.00%
staleEpicss	Supporting	5.00%
bulkChanges	Supporting	5.00%
lastDayCompletions	Supporting	5.00%
zombieItemCount	Contextual	3.33%
estimationVariability	Contextual	3.33%
capacitySplitAcrossProjects	Contextual	3.33%

**Total: 100%**

---

### 3. Mathematical Methodology

#### 3.1 Overview: CSS-Heavy Approach

Based on practitioner-statistician assessment, the Invisible Work Risk score uses a modified CHS formula:

$$IWR = 0.75 \times \text{CSS}_{\text{inv}} + 0.25 \times \text{TRS}_{\text{inv}}$$

Where:

- $\text{CSS}_{\text{inv}}$  = Current State Score for invisible work indicators
- $\text{TRS}_{\text{inv}}$  = Trajectory Score for invisible work indicators

## Why this weighting?

Component	General CHS	Invisible Work	Rationale
CSS	50%	<b>75%</b>	Current state is most actionable for invisible work
TRS	35%	<b>25%</b>	Variability indicators are noisy; reduce trajectory influence
PGS	15%	<b>0%</b>	Circular reasoning risk; adds complexity without benefit

**PGS Exclusion Rationale:** Peer Growth Score compares a team's trajectory to peers at similar starting points. For invisible work, this creates circular reasoning: if many teams start with high invisible work, improvements would be penalized by peers also improving. Additionally, the high measurement noise in variability indicators makes relative ranking unreliable.

## 3.2 Data Transformation

### 3.2.1 Variability Indicator Transformation

Variability indicators (coefficient of variation, standard deviation ratios) are typically right-skewed with a long tail. Before aggregation, apply log-transformation:

$$\tilde{X}_i = -\log(\max(X_i, 0.01))$$

The floor of 0.01 prevents undefined logarithms. The negative sign ensures that after transformation:

- Low variability (good) → High transformed value
- High variability (bad) → Low transformed value

#### Example:

- CV = 0.15 (low variability):  $\tilde{X} = -\log(0.15) = 1.90$
- CV = 0.60 (high variability):  $\tilde{X} = -\log(0.60) = 0.51$

### 3.2.2 Rate/Count Indicator Transformation

Non-variability indicators use standard direction-adjusted transformation:

$$\tilde{X}_i = d_i \times X_i$$

Where  $d_i$  is the direction coefficient:

- $d_i = +1$  if higher values indicate more invisible work risk
- $d_i = -1$  if higher values indicate less invisible work risk

Indicator	Direction
avgDailyUpdates	-1 (more updates = less risk)
All variability indicators	+1 (more variability = more risk)
staleWorkItems , staleEpics	+1 (more stale = more risk)
midSprintCreations	+1 (more mid-sprint = more risk)
closedWithoutComments	+1 (more silent closures = more risk)
lastDayCompletions	+1 (more clustering = more risk)
bulkChanges	+1 (more bulk = more risk)
zombieItemCount	+1 (more zombies = more risk)
capacitySplitAcrossProjects	+1 (more split = more risk)

### 3.2.3 Winsorization

Before standardization, winsorize extreme values:

- **Variability indicators:** P1/P99 (wider bounds due to heavy tails)
- **Rate indicators:** P2/P98 (standard bounds)

$$X_i \leftarrow \text{clip}(X_i, P_{\text{low}}, P_{\text{high}})$$

## 3.3 Current State Score (CSS) Calculation

### Step 1: Standardization

For each indicator  $i$ , compute z-score against baseline population norms:

$$z_i = \frac{\tilde{X}_i - \mu_{\text{baseline},i}}{\sigma_{\text{baseline},i}}$$

Where:

- $\mu_{\text{baseline},i}$  = Population mean for indicator  $i$  at calibration
- $\sigma_{\text{baseline},i}$  = Population standard deviation at calibration

**Note:** Baseline norms should be re-calibrated annually or when the comparison population changes significantly.

### Step 2: Z-Score Capping

Cap individual z-scores to prevent extreme values from dominating:

$$z_i \leftarrow \text{clip}(z_i, -3, +3)$$

### **Step 3: Weighted Aggregation**

Compute the raw CSS as weighted sum:

$$\text{CSS}_{\text{raw}} = \sum_{i=1}^m w_i \cdot z_i$$

Where  $w_i$  is the weight for indicator  $i$  from Section 2.4, and  $\sum w_i = 1$ .

### **Step 4: Variance Calculation**

The variance of the aggregate depends on weights and indicator correlations:

$$\text{Var}(\text{CSS}_{\text{raw}}) = \sum_{i=1}^m w_i^2 \cdot (1 - \bar{\rho}) + \bar{\rho}$$

For invisible work indicators, use  $\bar{\rho} = 0.45$  (higher than general CHS  $\bar{\rho} = 0.30$ ).

With the tier-based weighting:

- $\sum w_i^2 = 7 \times (0.0786)^2 + 7 \times (0.05)^2 + 3 \times (0.0333)^2$
- $\sum w_i^2 = 0.0432 + 0.0175 + 0.0033 = 0.064$
- $\text{Var}(\text{CSS}_{\text{raw}}) = 0.064 \times (1 - 0.45) + 0.45 = 0.035 + 0.45 = 0.485$

### **Step 5: Scaling**

Scale to achieve SD  $\approx 15$  on a 0-100 scale:

$$k_{\text{css}} = \frac{15}{\sqrt{\text{Var}(\text{CSS}_{\text{raw}})}} = \frac{15}{\sqrt{0.485}} \approx 21.5$$

$$\text{CSS}_{\text{inv}} = 50 + k_{\text{css}} \times \text{CSS}_{\text{raw}}$$

### **Step 6: Bounding**

Bound to [5, 95] to avoid extreme scores:

$$\text{CSS}_{\text{inv}} = \text{clip}(\text{CSS}_{\text{inv}}, 5, 95)$$

#### **Interpretation:**

- CSS = 50: Average invisible work risk relative to baseline
- CSS = 65: Risk is 1 SD above average
- CSS = 35: Risk is 1 SD below average

## **3.4 Trajectory Score (TRS) Calculation**

### **Step 1: Period Segmentation**

Divide the assessment period into early and recent windows:

- **Early period:** First 4–6 time periods (sprints/weeks)
- **Recent period:** Last 4–6 time periods

### Step 2: Effect Size per Indicator

For each indicator  $i$ , calculate the trajectory as standardized effect size:

$$\text{trajectory}_i = \frac{\bar{X}_{i,\text{recent}} - \bar{X}_{i,\text{early}}}{\sigma_{i,\text{pooled}}}$$

Where:

- $\bar{X}_{i,\text{recent}}$  = Mean of indicator  $i$  in recent period
- $\bar{X}_{i,\text{early}}$  = Mean of indicator  $i$  in early period
- $\sigma_{i,\text{pooled}}$  = Pooled standard deviation across all periods

### Step 3: Trajectory Capping

Cap individual trajectories more aggressively for invisible work (due to noise):

$$\text{trajectory}_i \leftarrow \text{clip}(\text{trajectory}_i, -2, +2)$$

This is tighter than general CHS ( $\pm 3$ ) because variability indicators are noisier.

### Step 4: Weighted Aggregation

$$\text{TRS}_{\text{raw}} = \sum_{i=1}^m w_i \cdot \text{trajectory}_i$$

### Step 5: Aggregate Capping

$$\text{TRS}_{\text{raw}} \leftarrow \text{clip}(\text{TRS}_{\text{raw}}, -3.0, +3.0)$$

Tighter than general CHS ( $\pm 4.5$ ) to reduce trajectory noise influence.

### Step 6: Scaling

$$\text{TRS}_{\text{inv}} = 50 + 10 \times \text{TRS}_{\text{raw}}$$

$$\text{TRS}_{\text{inv}} = \text{clip}(\text{TRS}_{\text{inv}}, 20, 80)$$

### Interpretation:

- TRS = 50: Stable, no significant change
- TRS = 60: Improving (risk decreasing)
- TRS = 40: Worsening (risk increasing)

## 3.5 Final Risk Score Aggregation

## Composite Calculation

$$IWR = 0.75 \times CSS_{inv} + 0.25 \times TRS_{inv}$$

## Standard Error Calculation

By propagation of uncertainty (assuming component independence):

$$SE(IWR)_{raw} = \sqrt{0.75^2 \times SE(CSS_{inv})^2 + 0.25^2 \times SE(TRS_{inv})^2}$$

Apply inflation factor for component correlation:

$$SE(IWR) = 1.30 \times SE(IWR)_{raw}$$

The 1.30 factor (vs. 1.20 for general CHS) accounts for higher correlation between CSS and TRS for invisible work indicators.

## Confidence Interval

90% confidence interval:

$$IWR \pm 1.645 \times SE(IWR)$$

## 3.6 Risk Score Interpretation

Since we're measuring **risk** rather than **health**, the scale interpretation is inverted:

IWR Score	Risk Label	Interpretation
≤ 30	Low Risk	Work well-captured in Jira; signals are consistent
31-45	Moderate Risk	Most work visible; some monitoring warranted
46-55	Typical Risk	Average visibility; room for improvement
56-70	Elevated Risk	Suggests hidden work streams; investigation recommended
> 70	High Risk	Likely substantial invisible work; action needed

**Mapping to Health Score:** For UI consistency with other dimensions, convert:

$$\text{Health Score} = 100 - IWR$$

## 3.7 Precision Categories

Based on standard error and data coverage:

SE(IWR)	Coverage	Precision	UI Treatment
≤ 5	≥ 90%	High	Full display
5-10	70-89%	Medium	Show with caveat
> 10	< 70%	Low	Flag as provisional

**Note:** These thresholds are wider than general CHS (80/65/45 vs. 85/75/50) due to inherent measurement noise in variability indicators.

---

## 4. Edge Case Handling

### 4.1 Missing Indicators

#### Coverage Threshold

Teams must have valid data for at least **60%** of weighted indicators:

$$\text{coverage} = \sum_{i \in \text{valid}} w_i$$

If coverage < 60%, the IWR score is not calculated. Display: "Insufficient Data."

#### Reweighting for Partial Data

When  $60\% \leq \text{coverage} < 100\%$ , reweight available indicators:

$$w'_i = \frac{w_i}{\sum_{j \in \text{valid}} w_j}$$

Flag the score as "Limited Data" when coverage < 80%.

### 4.2 Small Sample Sizes

#### Minimum Data Requirements

Data Availability	Calculation Approach	Flag
< 4 weeks	Not calculated	"Insufficient Data"
4-7 weeks	CSS-only (no TRS)	"Provisional"
≥ 8 weeks	Full CSS + TRS	Standard

#### Small Team Adjustment

Teams with < 5 members have higher natural variability. Apply shrinkage toward population mean:

$$IWR_{\text{adjusted}} = \alpha \times IWR + (1 - \alpha) \times 50$$

Where:

$$\alpha = \frac{n_{\text{members}}}{n_{\text{members}} + 3}$$

For a 3-person team:  $\alpha = 3/6 = 0.5$ , so the score is shrunk 50% toward the mean.

## 4.3 Extreme Values

### Individual Indicator Outliers

After winsorization and z-score capping, individual indicators are bounded to  $\pm 3$  SDs.

### Team-Level Outliers

If > 50% of indicators are at ceiling ( $z = 3$ ) or floor ( $z = -3$ ), flag for manual review. This pattern is unusual and may indicate:

- Data quality issues
- Team in exceptional circumstances
- Methodology calibration needed for this team type

## 4.4 New Teams

Teams with < 8 weeks of history:

1. Calculate CSS-only score
2. Set TRS = 50 (neutral)
3. Use modified formula:  $IWR = 0.85 \times CSS + 0.15 \times 50$
4. Flag as "Provisional - New Team"
5. Increase SE by 25%

## 4.5 Holiday/Vacation Periods

Periods with < 50% normal team activity should be:

1. Excluded from TRS calculation (early/recent period assignment)
2. Flagged in CSS calculation
3. Noted in output: "Assessment period includes reduced-activity weeks"

## 4.6 Methodology Transitions

When teams transition methodologies (e.g., Scrum to Kanban):

1. Flag first 4 weeks after transition
2. Increase SE by 30%
3. Note: "Recent methodology change may affect signal reliability"

## 5. Scenario Analysis

This section presents 10 detailed scenarios to validate the methodology produces sensible results. Each scenario includes context, input data, step-by-step calculations, and validation against expected outcomes.

### Validation Criteria

For each scenario, we assess:

1. **Face validity:** Does the score match intuition about invisible work?
2. **Discrimination:** Does the methodology distinguish different situations?
3. **Robustness:** Is the score stable to small input changes?
4. **Actionability:** Does the result guide meaningful action?

### Scenario 1: Team with Genuine High Invisible Work

**Context:** Engineering team of 7, heavily burdened by production support and meetings.

#### Input Data:

Indicator	Raw Value	Z-Score
throughputVariability	CV = 0.55	+2.1
memberThroughputVariability	CV = 0.48	+1.8
workflowStageTimeVariability	CV = 0.42	+1.5
sameSizeTimeVariability	CV = 0.38	+1.2
avgDailyUpdates	1.8/day	-1.9
staleWorkItems	35%	+2.0
inProgressItemsVariability	CV = 0.40	+1.4
Supporting indicators	(moderate)	avg +1.0
Contextual indicators	(moderate)	avg +0.8

#### Calculation:

- $CSS_{\text{raw}} = 0.55 \times 1.57 + 0.35 \times 1.0 + 0.10 \times 0.8 = 0.86 + 0.35 + 0.08 = 1.29$
- $CSS_{\text{inv}} = 50 + 21.5 \times 1.29 = 77.7$
- $TRS_{\text{inv}} = 52$  (slightly worsening)
- $IWR = 0.75 \times 77.7 + 0.25 \times 52 = 58.3 + 13.0 = 71.3$

**Result:** IWR = 71 (High Risk)

**Validation:** This is correct. The team shows strong signals across all tiers, consistent with reported high support burden.

---

## Scenario 2: Team with High Variability but No Invisible Work

**Context:** Team of 5 working on R&D projects with genuine complexity variation.

### Input Data:

Indicator	Raw Value	Z-Score
throughputVariability	CV = 0.50	+1.9
memberThroughputVariability	CV = 0.30	+0.5
workflowStageTimeVariability	CV = 0.45	+1.7
sameSizeTimeVariability	CV = 0.35	+1.0
avgDailyUpdates	5.2/day	+0.8
staleWorkItems	8%	-0.5
inProgressItemsVariability	CV = 0.25	+0.2
Supporting indicators	(low)	avg -0.3
Contextual indicators	(low)	avg -0.2

### Calculation:

- High throughput/stage variability but low staleness and good engagement
- $CSS_{raw} = 0.55 \times 0.74 + 0.35 \times (-0.3) + 0.10 \times (-0.2) = 0.41 - 0.11 - 0.02 = 0.28$
- $CSS_{inv} = 50 + 21.5 \times 0.28 = 56.0$
- $TRS_{inv} = 50$  (stable)
- $IWR = 0.75 \times 56 + 0.25 \times 50 = 54.5$

**Result:** IWR = 55 (Typical Risk)

**Validation:** Correct. Despite high throughput variability, the team's strong engagement (low staleness, high updates) pulls the score toward typical. This is the methodology working as intended—variability alone doesn't confirm invisible work; the full signal pattern matters.

---

## Scenario 3: Team with Low Jira Engagement but No Hidden Work

**Context:** Small team of 3 developers, minimal process overhead, prefer code over tickets.

**Input Data:**

Indicator	Raw Value	Z-Score
throughputVariability	CV = 0.22	-0.3
memberThroughputVariability	CV = 0.20	-0.4
workflowStageTimeVariability	CV = 0.18	-0.6
sameSizeTimeVariability	CV = 0.25	+0.1
avgDailyUpdates	1.2/day	-2.2
staleWorkItems	28%	+1.5
inProgressItemsVariability	CV = 0.15	-0.7
Supporting indicators	(mixed)	avg +0.5
Contextual indicators	(low)	avg -0.1

**Calculation:**

- Low variability across the board, but low engagement
- $CSS_{raw} = 0.55 \times (-0.37) + 0.35 \times 0.5 + 0.10 \times (-0.1) = -0.20 + 0.18 - 0.01 = -0.03$
- $CSS_{inv} = 50 + 21.5 \times (-0.03) = 49.4$

**Apply small team adjustment:**

- $\alpha = 3/(3+3) = 0.5$
- $CSS_{adjusted} = 0.5 \times 49.4 + 0.5 \times 50 = 49.7$
- $TRS_{inv} = 50$  (stable)
- $IWR = 0.75 \times 49.7 + 0.25 \times 50 = 49.8$

**Result:** IWR = 50 (Typical Risk), with small-team flag

**Validation:** Correct. The low engagement signals (staleness, low updates) are offset by excellent variability scores. The methodology correctly identifies this as a team with strong predictability despite minimal Jira interaction—not an invisible work problem.

**Scenario 4: Team Improving Invisible Work Discipline**

**Context:** Team implemented a “capture everything” policy 8 weeks ago.

**Input Data:**

Period	throughputVariability	staleWorkItems	avgDailyUpdates
Weeks 1-4	CV = 0.48	32%	2.1/day
Weeks 5-8	CV = 0.30	15%	4.5/day

### Trajectory Calculation:

- $\text{trajectory}_{\text{throughput}} = (0.30 - 0.48)/0.15 = -1.2$  (improving)
- $\text{trajectory}_{\text{stale}} = (15 - 32)/8 = -2.1$  (improving, capped at -2.0)
- $\text{trajectory}_{\text{updates}} = (4.5 - 2.1)/1.0 = +2.4$  (improving, capped at +2.0)

### Combined:

- $\text{TRS}_{\text{raw}} \approx -1.4$  (negative = improving for risk score, but we measure improvement as positive)
- Since lower risk is better:  $\text{TRS}_{\text{inv}} = 50 + 10 \times 1.4 = 64$

### CSS from recent data:

- Recent variability is low:  $\text{CSS}_{\text{inv}} = 45$  (below average risk)

### Final:

- $\text{IWR} = 0.75 \times 45 + 0.25 \times 64 = 33.75 + 16.0 = 49.75$

**Result:** IWR = 50 (Typical Risk), trending toward Low

**Validation:** Correct. The team's current state ( $\text{CSS} = 45$ ) shows improvement, and the trajectory ( $\text{TRS} = 64$ ) confirms the positive trend. The composite acknowledges both where they are and that they're heading in the right direction.

---

## Scenario 5: Team with Partial Data Availability

**Context:** New project; some indicators not available.

### Available Indicators (70% coverage):

- All 7 Primary indicators: available
- 4 of 7 Supporting indicators: available
- 0 of 3 Contextual indicators: available

### Reweighting:

- Original weights: Primary = 55%, Supporting (4/7) = 20%, Contextual = 0%
- Coverage = 55% + 20% + 0% = 75%
- Reweighted: Primary = 55/75 = 73.3%, Supporting = 20/75 = 26.7%

### Calculation with available data:

- Primary indicators avg z-score: +1.2
- Supporting indicators avg z-score: +0.8

- $CSS_{\text{raw}} = 0.733 \times 1.2 + 0.267 \times 0.8 = 0.88 + 0.21 = 1.09$
- $CSS_{\text{inv}} = 50 + 21.5 \times 1.09 = 73.4$

**Adjusted SE:** Inflate by 20% due to reweighting

**Result:** IWR = 68 (Elevated Risk), flagged "Limited Data"

**Validation:** Correct. The score is calculated but appropriately flagged. The team should not be penalized for unavailable data, but uncertainty is acknowledged.

---

## Scenario 6: Team with Extreme Outlier Values

**Context:** Team had production incident causing extreme metrics in one sprint.

**Input Data:**

Indicator	Week 1-3	Week 4 (incident)	Week 5-8
throughputVariability	CV = 0.25	CV = 1.2	CV = 0.28
staleWorkItems	10%	85%	12%

**Without winsorization:** Week 4 would produce z-scores > 5

**With winsorization (P1/P99):**

- CV = 1.2 is winsorized to P99  $\approx 0.85$
- Stale 85% is winsorized to P99  $\approx 65\%$
- Resulting z-scores: +3.0 (capped)

**Calculation:**

- CSS calculated on all 8 weeks, with winsorization dampening the outlier
- $CSS_{\text{inv}} \approx 58$  (moderately elevated)

**Result:** IWR = 56 (Elevated Risk)

**Validation:** Correct. The winsorization and capping prevent a single bad week from dominating. The score is elevated but not extreme, appropriately reflecting "mostly good with one bad period."

---

## Scenario 7: Cross-Project Team with Split Capacity

**Context:** Team works across 3 Jira projects, capacity fragmented.

**Input Data:**

Indicator	Value	Notes
capacitySplitAcrossProjects	0.65	High fragmentation

throughputVariability	CV = 0.40	Elevated
memberThroughputVariability	CV = 0.55	High
avgDailyUpdates	3.8/day	Normal
staleWorkItems	18%	Slightly elevated

#### Calculation:

- The capacity split indicator contributes to contextual tier (+1.5 z-score)
- Primary indicators show mixed signals (high member variability, okay throughput)
- $CSS_{raw} \approx 0.75$
- $CSS_{inv} = 50 + 21.5 \times 0.75 = 66.1$

**Result:** IWR = 62 (Elevated Risk)

**Validation:** Correct. The methodology appropriately flags cross-project teams as higher risk, as invisible work often hides in "the other project."

---

## Scenario 8: New Team with 4 Weeks History

**Context:** Newly formed team, limited data.

#### Available Data:

- 4 weeks of indicator data
- TRS cannot be calculated (requires 8 weeks for early/recent comparison)

#### Calculation:

- CSS-only:  $CSS_{inv} = 52$  (slightly elevated)
- TRS = 50 (neutral, by rule)
- Modified formula:  $IWR = 0.85 \times 52 + 0.15 \times 50 = 44.2 + 7.5 = 51.7$
- SE inflated by 25%

**Result:** IWR = 52 (Typical Risk), flagged "Provisional - New Team"

**Validation:** Correct. The score is conservative (shrunk toward neutral) and appropriately flagged.

---

## Scenario 9: Team During Holiday Period

**Context:** Assessment spans December with 3 weeks of reduced activity.

#### Activity Levels:

Week	Activity vs. Normal
Week 1-2	100%

Week 3-5	40% (holiday)
Week 6-8	100%

### Handling:

1. Weeks 3-5 excluded from TRS calculation
2. TRS uses Weeks 1-2 (early) vs. Weeks 6-8 (recent)
3. CSS includes all weeks but flags the assessment

**Result:** IWR = 48 (Typical Risk), noted "Assessment period includes reduced-activity weeks"

**Validation:** Correct. Holiday periods don't artificially inflate risk scores.

---

## Scenario 10: Team Transitioning Scrum to Kanban

**Context:** Team switched from 2-week sprints to Kanban flow 3 weeks ago.

### Pre-Transition (Weeks 1-5):

- Sprint-based metrics available
- Variability indicators tied to sprint boundaries

### Post-Transition (Weeks 6-8):

- Continuous flow metrics
- Different variability calculation basis

### Handling:

1. Flag first 4 weeks post-transition
2. Increase SE by 30%
3. TRS reliability reduced (comparing different metric paradigms)

**Result:** IWR = 55 (Typical Risk), flagged "Recent methodology change may affect signal reliability", SE = 9.2 (elevated)

**Validation:** Correct. The methodology doesn't penalize the transition but acknowledges reduced confidence.

---

## 5.11 Sensitivity Analysis

Beyond individual scenarios, we must understand how sensitive the methodology is to parameter choices and input variations.

### Weight Sensitivity

**Question:** How much does changing tier weights affect final scores?

**Test:** Compare scores under three weight schemes:

Scheme	Primary	Supporting	Contextual
Default	55%	35%	10%
Primary-heavy	70%	25%	5%
Balanced	45%	40%	15%

### Results for Scenario 1 (High Invisible Work):

Scheme	CSS_raw	CSS_inv	IWR
Default	1.29	77.7	71
Primary-heavy	1.42	80.5	74
Balanced	1.18	75.4	69

**Observation:** Score varies by  $\pm 3$  points across reasonable weight alternatives. This is acceptable—the team is consistently identified as “High Risk” regardless of weighting scheme.

### Results for Scenario 2 (High Variability, No Invisible Work):

Scheme	CSS_raw	CSS_inv	IWR
Default	0.28	56.0	55
Primary-heavy	0.38	58.2	57
Balanced	0.20	54.3	53

**Observation:** Score varies by  $\pm 2$  points. Team remains in “Typical Risk” category. The methodology correctly distinguishes this from Scenario 1 under all weight schemes.

### Correlation Parameter Sensitivity

**Question:** How sensitive is the score to the assumed correlation  $\rho$ ?

**Test:** Vary  $\rho$  from 0.30 (general CHS) to 0.60 (very high correlation)

$\rho$	Var(CSS_raw)	k_css	CSS (Scenario 1)
0.30	0.35	25.4	82.8
0.45 (default)	0.49	21.5	77.7
0.60	0.62	19.0	74.5

**Observation:** Higher assumed correlation produces more conservative (lower) scores. The default  $\rho = 0.45$  is a reasonable middle ground. The score varies by  $\sim 8$  points

across the full range—material but not category-changing for most teams.

**Recommendation:** If empirical correlation data becomes available, recalibrate  $\rho$  based on actual indicator correlations. Until then, 0.45 is defensible for invisible work indicators.

## Single Indicator Sensitivity

**Question:** Can a single indicator dominate the score?

**Test:** For Scenario 1, set each primary indicator to  $z = 0$  while keeping others unchanged.

Zeroed Indicator	CSS_raw	CSS_inv	Change
None (baseline)	1.29	77.7	—
throughputVariability	1.12	74.1	-3.6
memberThroughputVariability	1.15	74.7	-3.0
avgDailyUpdates	1.14	74.5	-3.2
staleWorkItems	1.13	74.3	-3.4

**Observation:** No single indicator changes the score by more than 4 points. The tiered weighting (7.86% per primary indicator) effectively prevents single-indicator dominance. A team cannot game the score by improving just one metric.

## Missing Data Sensitivity

**Question:** How does missing data affect score reliability?

**Test:** Progressively remove indicators and observe score and SE changes.

Coverage	Missing Indicators	IWR	SE(IWR)	Precision
100%	None	71	4.2	High
85%	3 contextual	71	4.8	High
75%	2 primary, 2 supporting	69	6.1	Medium
65%	4 primary, 2 supporting	66	8.3	Low
55%	—	Not calculated	—	—

**Observation:** The score is stable down to ~75% coverage. Below that, both the score and confidence degrade. The 60% threshold for calculation is appropriate—scores below this coverage are too uncertain to be meaningful.

## Stress Test: All Teams Improve

**Question:** What happens when all teams in the population improve their invisible work practices?

**Scenario:** All teams reduce variability by 20% and increase Jira engagement by 15%.

**Expected Behavior:**

- CSS scores should improve (lower risk) for all teams
- Relative ranking may stay similar, but absolute scores reflect real improvement
- This is by design: CSS uses fixed baseline norms, not current population percentiles

**Result:** If baseline norms are set at pre-improvement levels, all teams show score improvement. When norms are recalibrated (annually), the "new normal" becomes the baseline.

**Implication:** Unlike percentile-based scoring, this methodology allows for "everyone wins" scenarios where industry-wide improvement is visible in scores.

**Stress Test: Ceiling Effects**

**Question:** What happens for teams already performing well?

**Scenario:** Team with all indicators at  $z = -1.5$  (well below average risk)

**Calculation:**

- $CSS_{raw} = -1.5$  (all indicators weighted average)
- $CSS_{inv} = 50 + 21.5 \times (-1.5) = 17.8$
- Bounded to minimum 5
- IWR =  $0.75 \times 5 + 0.25 \times 50 = 16.3$

**Result:** IWR = 16 (Low Risk, at floor)

**Implication:** Teams at the floor have limited room to show improvement in IWR. For such teams:

1. Display "Excellent - Minimal invisible work risk detected"
2. Focus TRS on maintaining low risk rather than improving
3. Consider IWR "problem solved" and focus on other dimensions

---

## 5.12 Scenario Summary

#	Scenario	IWR	Risk Level	Key Validation
1	High invisible work	71	High	Correct detection
2	High variability, no invisible work	55	Typical	False positive avoided

3	Low engagement, small team	50	Typical	Shrinkage applied correctly
4	Improving discipline	50	Typical	Trend captured
5	Partial data	68	Elevated	Limited data flag works
6	Extreme outlier	56	Elevated	Winsorization effective
7	Cross-project team	62	Elevated	Fragmentation captured
8	New team	52	Typical	Provisional handling correct
9	Holiday period	48	Typical	Reduced activity handled
10	Methodology transition	55	Typical	Reliability flag appropriate

**Overall Assessment:** The methodology produces sensible, interpretable results across diverse scenarios. It correctly identifies high-risk teams, avoids obvious false positives, handles edge cases gracefully, and maintains stability under parameter perturbations.

---

## 6. Implementation Notes

### 6.1 Minimum Data Requirements

Requirement	Threshold	Handling
Historical data	$\geq 4$ weeks	Below: "Insufficient Data"
Indicator coverage	$\geq 60\%$ weighted	Below: "Insufficient Data"
TRS data	$\geq 8$ weeks	4-7 weeks: CSS-only
Team size	$\geq 3$ members	Below: Apply shrinkage

### 6.2 Computational Considerations

All calculations are feasible within Forge constraints:

- No iterative optimization required
- No external library dependencies beyond basic math
- Single-pass aggregation for each component
- Total calculation < 100ms for typical team

### 6.3 Caching Strategy

Data Element	Cache Duration	Invalidation
Baseline norms ( $\mu, \sigma$ )	1 year	Annual recalibration
Team raw metrics	1 week	New data ingestion
IWR scores	1 week	New assessment run
Indicator z-scores	1 week	New raw metrics

## 6.4 Recommended Refresh Frequency

Use Case	Refresh	Rationale
Dashboard display	Weekly	Balance freshness with stability
Trend analysis	Monthly	Reduce noise in trajectory
Annual review	Quarterly	Sufficient data for robust TRS

## 6.5 Integration with CHS

The IWR score can be converted to a health score for dimensional display:

$$\text{Invisible Work Health} = 100 - \text{IWR}$$

This maintains consistency with other dimensions where higher = healthier.

## 7. Limitations and Future Work

### 7.0 Statistical Properties and Assumptions

Before discussing limitations, it's important to document the statistical assumptions underlying this methodology.

#### Distributional Assumptions

**Variability indicators** are assumed to follow a log-normal distribution after transformation. This is appropriate because:

- Coefficient of variation is bounded below by 0
- Empirically, CV distributions show right-skew with occasional extreme values
- Log transformation normalizes the distribution for z-score calculation

**Rate indicators** (e.g., stale item percentage, daily updates) are assumed approximately normal after winsorization. This is reasonable for rates bounded between 0 and 1 when extreme values are capped.

**Composite score** distribution approaches normality by the Central Limit Theorem, given that we're aggregating 17 indicators. Even if individual indicators are non-normal, the weighted sum tends toward normal.

## Independence Assumptions

The methodology assumes:

1. **Indicators within a team are correlated** (captured by  $\rho = 0.45$ )
2. **Teams are independent** for baseline norm calculation
3. **Time periods within a team are correlated** (captured in TRS calculation)

Violation of assumption 2 (e.g., teams in same department share invisible work sources) could bias baseline norms. This should be monitored empirically.

## Stationarity Assumptions

The methodology assumes indicator distributions are approximately stationary over the assessment period. Non-stationarity (e.g., organization-wide process change) could invalidate TRS calculation. The methodology flags this via methodology transition handling (Section 4.6).

## 7.1 Construct Validity Limitations

The primary uncertainty is whether these indicators actually measure invisible work or something else:

Concern	Impact	Mitigation
Variability may reflect legitimate complexity	False positives	Supporting indicators provide cross-validation
Invisible work uniformly distributed	Undetectable	Relative comparison assumes variation exists
Team self-report bias	Validation challenge	Use objective calendar/meeting data where available

**Recommendation:** Conduct validation studies correlating IWR scores with:

- Team self-reported invisible work hours
- Calendar meeting load
- Slack/communication platform activity

## 7.2 Known Blind Spots

1. **Invisible work in meetings that produce visible output:** Productive meetings that create Jira tickets aren't "invisible" but consume time
2. **Efficient invisible work:** Some teams handle support efficiently with minimal disruption
3. **Cultural differences:** Some teams prefer minimal documentation regardless of invisible work

## 7.3 Future Enhancements

Enhancement	Benefit	Complexity
Calendar integration	Direct meeting burden measurement	High (data access)
Communication analysis	Slack/Teams activity correlation	High (privacy)
Individual-level scoring	Identify specific capacity drains	Medium (additional metrics)
Predictive modeling	Forecast invisible work trends	Medium (ML infrastructure)
Benchmark expansion	Industry-specific norms	Low (data collection)

## 7.4 Validation Study Design

Recommended validation approach:

### Phase 1: Pilot Study (10 teams, 3 months)

**Objective:** Establish basic construct validity

#### Data Collection:

- Weekly self-reported invisible work hours (survey)
- IWR scores calculated weekly
- Qualitative feedback on score reasonableness

#### Analysis:

- Pearson correlation between IWR and reported hours
- Target:  $r > 0.5$  (moderate correlation)
- Identify indicators with strongest/weakest signal

#### Success Criteria:

- Correlation is statistically significant ( $p < 0.05$ )
- Teams with self-reported high invisible work show elevated IWR
- No systematic bias by team size or methodology

### Phase 2: Calibration Study (50 teams, 6 months)

**Objective:** Refine methodology parameters

#### Data Collection:

- Expanded team sample across departments
- Calendar integration (meeting hours) where available

- Communication platform activity (optional)

#### **Analysis:**

- Multiple regression: IWR ~ reported\_hours + meeting\_hours + slack\_activity
- Identify redundant indicators (high correlation, low unique contribution)
- Recalibrate tier weights based on empirical signal strength
- Establish population norms ( $\mu_{\text{baseline}}$ ,  $\sigma_{\text{baseline}}$ )

#### **Success Criteria:**

- $R^2 > 0.30$  for multiple regression model
- All primary indicators show significant unique contribution
- Population norms are stable (low variance across subgroups)

### **Phase 3: Deployment (Organization-wide)**

**Objective:** Production rollout with monitoring

#### **Data Collection:**

- Ongoing IWR scores for all teams
- Periodic (quarterly) validation surveys
- Team feedback on actionability

#### **Analysis:**

- Monitor false positive rate (teams flagged high risk but report low invisible work)
- Monitor false negative rate (teams with high invisible work but low IWR)
- Track score stability over time

#### **Success Criteria:**

- False positive rate < 15%
- False negative rate < 20%
- 70% of teams find score "reasonable" or "accurate"

## **7.5 Interpretation Guidelines for Practitioners**

When presenting IWR scores to teams, use the following communication guidelines:

#### **DO say:**

- "Your invisible work risk indicators suggest there may be work happening outside Jira"
- "These patterns are consistent with teams that report high meeting load or support burden"
- "Consider whether your team's Jira practices capture all meaningful work"

#### **DON'T say:**

- "You have 40% invisible work"
- "Your score proves you're not tracking work properly"
- "Teams with your score have exactly X hours of untracked work"

**Frame as conversation starter:** The IWR score should prompt discussion, not judgment. Questions to explore:

1. "What work consumes your time that doesn't appear in Jira?"
2. "Are there recurring meetings or support duties that should be tracked?"
3. "Could we create better tickets for interrupt-driven work?"

### Acknowledge limitations:

- "This is a risk indicator, not a measurement"
- "High variability can have causes other than invisible work"
- "Your team knows your work better than any metric"

---

## 8. Glossary

---

Term	Definition
<b>Invisible Work</b>	Effort that occurs but is not tracked in Jira
<b>IWR (Invisible Work Risk)</b>	Composite score (0-100) indicating likelihood of invisible work
<b>CSS (Current State Score)</b>	Component measuring current indicator levels vs. baseline
<b>TRS (Trajectory Score)</b>	Component measuring trend over time within assessment period
<b>PGS (Peer Growth Score)</b>	Peer comparison component (not used in IWR)
<b>Coefficient of Variation (CV)</b>	Standard deviation divided by mean; normalized variability measure
<b>Z-score</b>	Number of standard deviations from the mean
<b>Winsorization</b>	Capping extreme values at specified percentiles
<b>Baseline norms</b>	Population mean and standard deviation at calibration time
<b>Effect size</b>	Standardized difference between two means

---

## Appendix A: Complete Indicator Reference

---

### A.1 Indicator Details

ID	Indicator	Type	Direction	Weight
----	-----------	------	-----------	--------

1	<code>throughputVariability</code>	Variability	+1	7.86%
2	<code>memberThroughputVariability</code>	Variability	+1	7.86%
3	<code>workflowStageTimeVariability</code>	Variability	+1	7.86%
4	<code>sameSizeTimeVariability</code>	Variability	+1	7.86%
5	<code>avgDailyUpdates</code>	Rate	-1	7.86%
6	<code>staleWorkItems</code>	Rate	+1	7.86%
7	<code>inProgressItemsVariability</code>	Variability	+1	7.86%
8	<code>closedWithoutComments</code>	Rate	+1	5.00%
9	<code>midSprintCreations</code>	Rate	+1	5.00%
10	<code>frequentUseVariability</code>	Variability	+1	5.00%
11	<code>collaborationVariability</code>	Variability	+1	5.00%
12	<code>staleEpics</code>	Rate	+1	5.00%
13	<code>bulkChanges</code>	Rate	+1	5.00%
14	<code>lastDayCompletions</code>	Rate	+1	5.00%
15	<code>zombieItemCount</code>	Count	+1	3.33%
16	<code>estimationVariability</code>	Variability	+1	3.33%
17	<code>capacitySplitAcrossProjects</code>	Rate	+1	3.33%

## A.2 Indicator Calculation Details

**`throughputVariability`** : Coefficient of variation (CV) of completed story points per sprint over the assessment period.  $CV = \sigma/\mu$  where  $\sigma$  is standard deviation and  $\mu$  is mean. Higher CV indicates more variability.

**`memberThroughputVariability`** : For each team member, calculate their throughput CV. Then average across members. This captures individual-level variation that may be masked by team aggregates.

**`workflowStageTimeVariability`** : CV of time spent in "In Progress" or equivalent status. Calculated per work item, then aggregated. High variability suggests inconsistent work patterns.

**`sameSizeTimeVariability`** : Group work items by story point bucket (1, 2, 3, 5, 8, 13). Within each bucket, calculate CV of cycle time. Average across buckets. High variability means same-sized work takes unpredictably different times.

**`avgDailyUpdates`** : Count all Jira updates (status changes, comments, field edits) and divide by calendar days in assessment period. Compare to team size to normalize.

**staleWorkItems** : Percentage of "In Progress" items not updated in the configured stale threshold (default: 5 days). Excludes items flagged as blocked.

**inProgressItemsVariability** : CV of daily WIP count. Count items in active workflow states each day, calculate CV of that time series.

**closedWithoutComments** : Percentage of resolved items with zero comments. Indicates decisions made off-platform.

**midSprintCreations** : For sprint-based teams, percentage of completed items that were created after sprint start. High rates indicate "side door" work entry.

**frequentUseVariability** : CV of daily update counts. Stable teams have consistent daily Jira activity; variable teams have "burst" patterns.

**collaborationVariability** : CV of average comments per item per sprint. Measures consistency of communication patterns.

**staleEpics** : Percentage of in-progress epics not updated in the stale threshold. Strategic work visibility signal.

**bulkChanges** : Percentage of updates that occur in bulk operations (>10 items modified within 5 minutes). Indicates batch housekeeping rather than real-time tracking.

**lastDayCompletions** : Percentage of sprint completions occurring on the final day. High rates suggest batch status updates rather than continuous tracking.

**zombieItemCount** : Count of backlog items with no updates in 180+ days but still open. Normalized by total backlog size.

**estimationVariability** : CV of (actual\_time / estimated\_time) ratio. High variability means estimates are unreliable.

**capacitySplitAcrossProjects** : Herfindahl-Hirschman Index (HHI) of work distribution across projects. Lower HHI = more fragmentation = more risk of invisible work in "other" projects.

---

## Appendix B: Formula Quick Reference

---

### Transformation

#### Variability indicators:

$$\tilde{X}_i = -\log(\max(X_i, 0.01))$$

#### Rate indicators:

$$\tilde{X}_i = d_i \times X_i$$

### CSS Calculation

$$z_i = \frac{\tilde{X}_i - \mu_{\text{baseline},i}}{\sigma_{\text{baseline},i}}$$

$$\text{CSS}_{\text{raw}} = \sum_{i=1}^m w_i \cdot z_i$$

$$\text{Var}(\text{CSS}_{\text{raw}}) = \sum_{i=1}^m w_i^2 \cdot (1 - \bar{\rho}) + \bar{\rho}$$

$$k_{\text{css}} = \frac{15}{\sqrt{\text{Var}(\text{CSS}_{\text{raw}})}}$$

$$\text{CSS}_{\text{inv}} = \text{clip}(50 + k_{\text{css}} \times \text{CSS}_{\text{raw}}, 5, 95)$$

## TRS Calculation

$$\text{trajectory}_i = \frac{\bar{X}_{i,\text{recent}} - \bar{X}_{i,\text{early}}}{\sigma_{i,\text{pooled}}}$$

$$\text{TRS}_{\text{raw}} = \text{clip}\left(\sum_{i=1}^m w_i \cdot \text{clip}(\text{trajectory}_i, -2, +2), -3, +3\right)$$

$$\text{TRS}_{\text{inv}} = \text{clip}(50 + 10 \times \text{TRS}_{\text{raw}}, 20, 80)$$

## Final Score

$$\text{IWR} = 0.75 \times \text{CSS}_{\text{inv}} + 0.25 \times \text{TRS}_{\text{inv}}$$

$$\text{SE}(\text{IWR}) = 1.30 \times \sqrt{0.75^2 \times \text{SE}(\text{CSS})^2 + 0.25^2 \times \text{SE}(\text{TRS})^2}$$

$$90\% \text{ CI} = \text{IWR} \pm 1.645 \times \text{SE}(\text{IWR})$$

## Health Score Conversion

$$\text{Invisible Work Health} = 100 - \text{IWR}$$

## Appendix C: Scenario Calculation Details

See Section 5 for detailed worked examples.

Summary of scenario results:

#	Scenario	IWR	Risk Level	Notes
1	High invisible work team	71	High	Correct detection

2	High variability, no invisible work	55	Typical	False positive avoided
3	Low engagement, small team	50	Typical	Shrinkage applied
4	Improving discipline	50	Typical	Trend captured
5	Partial data	68	Elevated	Limited data flag
6	Extreme outlier	56	Elevated	Winsorization effective
7	Cross-project team	62	Elevated	Fragmentation captured
8	New team (4 weeks)	52	Typical	Provisional flag
9	Holiday period	48	Typical	Reduced activity handled
10	Methodology transition	55	Typical	Reliability flag

---

## Revision History

---

Version	Date	Changes
1.0	2026-01-28	Initial specification

---

Document version 1.0. *Invisible Work Detection Methodology Specification.*