

# mitarbeiterpuls



## DEVELOPER BRIEFING

### Implementierungsleitfaden v2.1 – PULS-Berechnung und Datenmodell

Version 2.1 (Beta) | Onyva Innovation GmbH | Streng vertraulich

#### **⚠ Wichtig: Diese Version ersetzt alle vorherigen Briefings**

Dieses Dokument beschreibt die v2.1-Logik. Neu gegenüber v2.0: Befragungsrhythmus 14-täglich (nicht wöchentlicher), optionale Freitextfrage als Item 11 vollständig spezifiziert, explizite Anonymitätspflicht für Freitextausgabe im Dashboard.

## 1. Überblick – Änderungen v2.0 → v2.1

Thema	v2.0	v2.1 (aktuell)
Befragungsrhythmus	Wöchentlich	14-täglich
Freitextfrage	Nur DB-Feld, keine UI-Spezifikation	Vollständig: Position, Text, Anonymität, Hinweistext
Anonymität Freitext	Nicht explizit beschrieben	Explizit: kein Zeitstempel, kein Team-Tag im Output
Version	2.0	2.1

## 2. Fragebogen v2.1 – alle 11 Items

Befragungsrhythmus: alle 14 Tage. Items 1–10 Pflicht, Item 11 optional.

**Skala Items 1–10: 1 = stimme überhaupt nicht zu | 10 = stimme voll zu**

#	DB-Feld	Code	Dimension	Aussage
1	e1	E1	Engagement	Ich war in dieser Woche bei der Arbeit motiviert.
2	e2	E2	Engagement	Meine Arbeit hatte fuer mich in dieser Woche Sinn und Relevanz.
3	f1	F1	Fuehrung	Meine direkte Fuehrungskraft hat mir in dieser Woche Orientierung gegeben.
4	f2	F2	Fuehrung	Ich habe mich durch meine Fuehrungskraft fair und respektvoll behandelt gefuehlt.
5	b1	B1	Belastungsbalance	Mein Arbeitspensum war in dieser Woche gut bewaeltigbar.
6	b2	B2	Belastungsbalance	Ich konnte Belastung und Erholung in dieser Woche gut ausbalancieren.
7	n1	N1	Energie	Ich habe mich in dieser Woche energiegeladen gefuehlt.
8	n2	N2	Energie	Ich hatte in dieser Woche ausreichend mentale Kraft fuer meine Aufgaben.
9	bi1	BI1	Bindung	Ich moechte auch in den naechsten 12 Monaten Teil dieses Unternehmens sein.
10	bi2	BI2	Bindung	Ich wuerde dieses Unternehmen als Arbeitgeber weiterempfehlen.

### Item 11 – Optionale Freitextfrage (nach den 10 Pflicht-Items)

#### Fragetext im UI:

„Moechten Sie noch etwas teilen – was gut lief oder was Sie bewegt hat?“

(optional – anonym)

#### Anforderungen an die Implementierung:

- Position: NACH Item BI2, niemals zwischen den 10 Pflicht-Items
- Eingabetyp: Freitext, mehrzeiliges Textfeld
- Pflichtfeld: NEIN – darf nicht erzwungen werden
- DB-Feld: comment\_text TEXT NULL (bereits im Schema vorhanden)
- Anonymitaet: kein user\_id, kein Zeitstempel, kein Team-Tag im Output
- Dashboard: nur aggregiert und anonymisiert, nie einzeln anzeigen
- PULS-Berechnung: Freitextantworten gehen NICHT in den PULS ein

#### Pflicht-Hinweistext direkt unter dem Eingabefeld im UI:

*"Ihre Antwort ist vollstaendig anonym. Ein Rueckschluss auf Ihre Person ist technisch nicht moeglich."*

## 3. Datenmodell

---

### 3.1 Rohantworten-Tabelle

Kein user\_id. Das Feld comment\_text war bereits in v2.0 vorhanden und wird jetzt aktiv verwendet.

```
CREATE TABLE pulse_responses (  
  id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  created_at  TIMESTAMPTZ NOT NULL DEFAULT now(),  
  company_id  UUID NOT NULL,  
  team_id     UUID NULL,  
  survey_id   UUID NOT NULL,  
  period_start DATE NOT NULL,  
  period_end  DATE NOT NULL,  
  -- 10 Pflicht-Items (SMALLINT 1-10, NULL wenn nicht beantwortet)  
  e1  SMALLINT CHECK (e1 BETWEEN 1 AND 10),  
  e2  SMALLINT CHECK (e2 BETWEEN 1 AND 10),  
  f1  SMALLINT CHECK (f1 BETWEEN 1 AND 10),  
  f2  SMALLINT CHECK (f2 BETWEEN 1 AND 10),  
  b1  SMALLINT CHECK (b1 BETWEEN 1 AND 10),  
  b2  SMALLINT CHECK (b2 BETWEEN 1 AND 10),  
  n1  SMALLINT CHECK (n1 BETWEEN 1 AND 10),  
  n2  SMALLINT CHECK (n2 BETWEEN 1 AND 10),  
  bi1 SMALLINT CHECK (bi1 BETWEEN 1 AND 10),  
  bi2 SMALLINT CHECK (bi2 BETWEEN 1 AND 10),  
  -- Item 11: optionale Freitextantwort  
  comment_text TEXT NULL  
);  
  
-- WICHTIG: comment_text niemals mit user_id oder Zeitstempel zusammen  
ausgeben
```

### 3.2 Aggregationstabelle

```
CREATE TABLE pulse_period_metrics (  
  id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),  
  company_id  UUID NOT NULL,  
  team_id     UUID NULL,  
  period_start DATE NOT NULL,  
  period_end  DATE NOT NULL,  
  engagement  NUMERIC(3,1),  
  fuehrung    NUMERIC(3,1),  
  belastungsbalance NUMERIC(3,1),  
  energie     NUMERIC(3,1),  
  bindung     NUMERIC(3,1),  
  puls       NUMERIC(3,1),  
  n_engagement INTEGER NOT NULL DEFAULT 0,  
  n_fuehrung  INTEGER NOT NULL DEFAULT 0,  
  n_belastungsbalance INTEGER NOT NULL DEFAULT 0,  
  n_energie   INTEGER NOT NULL DEFAULT 0,  
  n_bindung   INTEGER NOT NULL DEFAULT 0,
```

```

visible          BOOLEAN NOT NULL DEFAULT TRUE,
created_at      TIMESTAMPTZ NOT NULL DEFAULT now()
);
-- Freitexte verbleiben in pulse_responses.comment_text
-- Ausgabe nur separat, anonymisiert (siehe Abschnitt 5)

```

## 4. Berechnungslogik

---

### 4.1 Drei Ebenen

Freitextantworten (comment\_text) gehen in keine Berechnungsebene ein.

- Ebene 1: Personenwert je Dimension = (Item\_1 + Item\_2) / 2, nur wenn beide Items beantwortet.
- Ebene 2: KPI = arithmetischer Mittelwert aller gueltigen Personenwerte, n >= 5.
- Ebene 3: PULS = geometrischer Mittelwert der fuef KPIs, nur wenn alle fuef vorliegen.

### 4.2 PULS-Formel

```

PULS = round(
  exp(
    (1.0 / 5.0) * (
      ln(engagement_kpi)      +
      ln(fuehrung_kpi)       +
      ln(belastungsbalance_kpi) +
      ln(energie_kpi)        +
      ln(bindung_kpi)
    )
  ),
  1 -- eine Nachkommastelle, Skala 1.0 bis 10.0
)

```

### 4.3 TypeScript Referenzimplementierung

```

function round1(v: number): number {
  return Math.round(v * 10) / 10;
}

function geometricMean(values: number[]): number | null {
  if (!values.length || values.some(v => v <= 0)) return null;
}

```

```

    return Math.exp(values.reduce((s, v) => s + Math.log(v), 0) /
values.length);
}

function getPersonScores(row: ResponseRow): DimensionScores {
    return {
        engagement:      row.e1  && row.e2  ? (row.e1 + row.e2) / 2 :
null,
        fuehrung:       row.f1  && row.f2  ? (row.f1 + row.f2) / 2 :
null,
        belastungsbalance: row.b1  && row.b2  ? (row.b1 + row.b2) / 2 :
null,
        energie:        row.n1  && row.n2  ? (row.n1 + row.n2) / 2 :
null,
        bindung:        row.bi1 && row.bi2 ? (row.bi1 + row.bi2) / 2 :
null,
        // comment_text wird hier nicht verarbeitet
    };
}

function aggregatePeriod(rows: ResponseRow[], minN = 5): AggregateResult {
    const ps = rows.map(getPersonScores);
    const vals = (key: keyof DimensionScores) =>
        ps.map(r => r[key]).filter((v): v is number => v !== null);
    const eng=vals('engagement'), fueh=vals('fuehrung'),
bel=vals('belastungsbalance');
    const ener=vals('energie'), bind=vals('bindung');
    const kpi = (v: number[]) =>
        v.length >= minN ? round1(v.reduce((a,b)=>a+b,0)/v.length) : null;
    const E=kpi(eng), F=kpi(fueh), B=kpi(bel), N=kpi(ener), BI=kpi(bind);
    const pulsInput = [E,F,B,N,BI].filter((v): v is number => v !== null);
    const puls = pulsInput.length===5 ? round1(geometricMean(pulsInput!)) :
null;
    return { engagement:E, fuehrung:F, belastungsbalance:B, energie:N,
bindung:BI, puls,
        nEngagement:eng.length, nFuehrung:fueh.length,
nBelastungsbalance:bel.length,
        nEnergie:ener.length, nBindung:bind.length };
}

```

## 5. Freitext – Anonymisierte Dashboard-Abfrage

Freitextantworten werden NIEMALS mit identifizierenden Merkmalen ausgegeben.

```

-- Anonymisierte Freitexte fuer Dashboard
SELECT comment_text
FROM pulse_responses
WHERE
    company_id = :company_id
    AND period_start = :period_start

```

```

AND comment_text IS NOT NULL
AND comment_text != ''
AND (
  SELECT COUNT(*) FROM pulse_responses
  WHERE company_id = :company_id
  AND period_start = :period_start
  AND comment_text IS NOT NULL
) >= 5 -- Anonymitaetsschwelle
ORDER BY RANDOM(); -- keine chronologische Reihenfolge

-- NIEMALS ausgeben: created_at, team_id oder survey_id zusammen mit
comment_text

```

### ⊖ Was bei Freitext niemals ausgegeben werden darf

Kein Zeitstempel, kein team\_id, kein Bereich, keine chronologische Reihenfolge. Ausgabe: reiner Text, zufaellig sortiert, erst ab n >= 5 Freitextantworten.

## 6. Status- und Trendlogik

### 6.1 Ampel

```

type Status = 'green' | 'yellow' | 'red';
function getStatus(v: number): Status {
  if (v >= 7.0) return 'green';
  if (v >= 6.0) return 'yellow';
  return 'red';
}
// Gilt fuer alle 5 KPIs und den PULS gleichermassen

```

### 6.2 Trend

```

function getTrend(current: number, previous: number | null): number | null
{
  if (previous === null) return null;
  return Math.round((current - previous) * 10) / 10;
}
// > 0 Pfeil hoch gruen +0.3
// < 0 Pfeil runter rot -0.2
// == 0 neutral
// null keine Vorperiode

```

## 7. Missing-Data und Validierungsregeln

---

1. Alle Item-Werte zwischen 1 und 10 (nicht 0).
2. Dimensionswerte nur aus vollstaendigen Item-Paaren.
3. KPI-Ausweisung nur ab  $n \geq 5$  gueltigen Personenwerten.
4. PULS nur berechnen wenn alle fuenf KPIs vorliegen.
5. Geometrischen Mittelwert verwenden (nicht arithmetisch).
6. Alle Werte auf 1 Nachkommastelle runden.
7. `visible = false` wenn  $n < 5$  fuer irgendeine Dimension.
8. `comment_text` niemals mit identifizierenden Metadaten ausgeben.
9. Freitext-Ausgabe erst ab  $\geq 5$  Freitextantworten.

## 8. Rechenbeispiel zur Verifikation

---

```
// Eingabe:
engagement=8.2, fuehrung=7.8, belastungsbalance=6.4, energie=7.1,
bindung=6.6

// Berechnung:
logSum = ln(8.2)+ln(7.8)+ln(6.4)+ln(7.1)+ln(6.6)
        = 2.1041+2.0541+1.8563+1.9601+1.8871 = 9.8617
PULS   = exp(9.8617/5) = exp(1.9723) = 7.19 -> gerundet: 7.2

// Erwartetes Ergebnis: PULS = 7.2, Status: green
// Gelbe Einzelwerte: belastungsbalance=6.4 (yellow), bindung=6.6 (yellow)
```

## 9. Was aus v1 vollstaendig entfernt werden muss

---

### ➔ Vollstaendig entfernen, nicht nur deaktivieren

- Gewichtete Summe 0.30/0.20/0.20/0.15/0.15
- Invertierungslogik (10 - belastung) oder (10 - fluktuation)
- Fluktuation als Item oder KPI im PULS-Berechnungspfad
- Arithmetischer Mittelwert als PULS-Aggregation
- Skala 0-10 (v2 verwendet 1-10)

