

# Technisches Briefing

KPI-Berechnung, PULS-Score & Datenmodell

<b>Dokument:</b>	Developer Briefing v1.0
<b>Produkt:</b>	Mitarbeiterpuls (mitarbeiterpuls.de)
<b>Herausgeber:</b>	Onyva Innovation GmbH
<b>Datum:</b>	04.03.2026
<b>Vertraulichkeit:</b>	Intern / Auftragnehmer

Dieses Dokument beschreibt die vollständige Berechnungslogik für alle Metriken der Mitarbeiterpuls-App: die 5 Einzel-KPIs und den zusammengefassten PULS-Wert. Es dient als verbindliches Briefing für die technische Umsetzung.

# 1. Architektur-Überblick

Mitarbeiterpuls basiert auf einer zweistufigen Metrik-Hierarchie:

## Informationshierarchie

PULS (Gesamtwert) beantwortet: "Wie geht es uns?" | 5 Einzel-KPIs beantworten: "Warum?"  
| KI-Empfehlungen beantworten: "Was tun?"

## 1.1 Datenfluss

Mitarbeiter geben alle 1-2 Wochen anonymes Feedback ab. Pro Feedback-Runde werden genau 4 Fragen gestellt, jeweils auf einer Skala von 1 bis 10. Aus diesen 4 Antworten werden 5 KPIs abgeleitet (der 5. KPI wird berechnet). Aus den 5 KPIs wird der PULS-Gesamtwert aggregiert.

Schicht	Elemente	Quelle	Sichtbarkeit
Rohdaten	4 Fragen (1-10)	Mitarbeiter-Input	Nur Backend
Einzel-KPIs	5 KPIs	Aggregation + KI	CEO Dashboard
<b>PULS</b>	1 Gesamtwert	Gewichtete Formel	<b>Hero-Metrik, Monitor, E-Mails</b>

## 2. Die 4 Feedback-Fragen

Jede Feedback-Runde stellt genau diese 4 Fragen. Die Reihenfolge ist fix. Jede Frage wird mit einem Wert von 1 (schlecht) bis 10 (sehr gut) beantwortet, plus einem optionalen anonymen Freitextfeld am Ende.

#	Frage (DE)	Ziel-KPI	Skala	Polarität
Q1	Wie war deine Woche?	<b>Engagement</b>	1-10	hoch = gut
Q2	Wie erlebst du deine Führungskraft?	<b>Führung</b>	1-10	hoch = gut
Q3	Wie ist deine Arbeitsbelastung?	<b>Belastung</b>	1-10	hoch = schlecht
Q4	Wie energiegeladen fühlst du dich?	<b>Energie</b>	1-10	hoch = gut

### WICHTIG: Invertierte Polarität bei Belastung

Q3 (Belastung) ist der einzige Wert mit invertierter Logik: Ein hoher Wert bedeutet HOHE Belastung, also etwas Negatives. Dies muss in allen Berechnungen, Visualisierungen und Ampelfarben berücksichtigt werden. Steigt der Belastungswert, wird er rot — nicht grün.

### 2.1 Datenbank-Schema (Supabase)

Jede Einzelantwort wird als Zeile in der Tabelle `feedback_responses` gespeichert:

```
CREATE TABLE feedback_responses (  
  id          UUID DEFAULT gen_random_uuid() PRIMARY KEY,  
  created_at  TIMESTAMPTZ DEFAULT now(),  
  team_id     UUID REFERENCES teams(id),  
  period_start DATE NOT NULL,           -- Beginn der Feedback-Periode  
  period_end  DATE NOT NULL,           -- Ende der Feedback-Periode  
  q_engagement SMALLINT CHECK (q_engagement BETWEEN 1 AND 10),  
  q_fuehrung  SMALLINT CHECK (q_fuehrung BETWEEN 1 AND 10),  
  q_belastung SMALLINT CHECK (q_belastung BETWEEN 1 AND 10),  
  q_energie   SMALLINT CHECK (q_energie BETWEEN 1 AND 10),  
  freitext    TEXT,                    -- optional, anonym  
  -- KEIN user_id! Vollständig anonym.  
);
```

## 3. Die 5 Einzel-KPIs

Aus den 4 Fragen werden 5 KPIs berechnet. Die ersten 4 KPIs entstehen direkt aus den Antworten (Durchschnittsbildung pro Periode). Der 5. KPI (Fluktuation) wird aus den Mustern der anderen 4 KPIs abgeleitet.

### 3.1 Engagement

Quelle: Q1 ("Wie war deine Woche?")

Berechnung: Arithmetischer Mittelwert aller Q1-Antworten der aktuellen Periode, gerundet auf 1 Nachkommastelle.

```
SELECT ROUND(AVG(q_engagement)::numeric, 1) AS kpi_engagement
FROM feedback_responses
WHERE period_start = :current_period_start
      AND team_id = :team_id; -- optional: ohne team_id = unternehmensweit
```

### 3.2 Führung

Quelle: Q2 ("Wie erlebst du deine Führungskraft?")

Berechnung: Arithmetischer Mittelwert aller Q2-Antworten der aktuellen Periode, gerundet auf 1 Nachkommastelle.

```
SELECT ROUND(AVG(q_fuehrung)::numeric, 1) AS kpi_fuehrung
FROM feedback_responses
WHERE period_start = :current_period_start;
```

### 3.3 Belastung

Quelle: Q3 ("Wie ist deine Arbeitsbelastung?")

Berechnung: Arithmetischer Mittelwert aller Q3-Antworten. ACHTUNG: Invertierte Polarität!  
Hoher Wert = hohe Belastung = negativ.

```
SELECT ROUND(AVG(q_belastung)::numeric, 1) AS kpi_belastung
FROM feedback_responses
WHERE period_start = :current_period_start;

-- Für den PULS-Score wird invertiert: (10 - kpi_belastung)
```

#### Ampelfarben Belastung (invertiert!)

Grün: Wert <= 5.5 (niedrige Belastung) | Gelb: Wert <= 6.5 (mittlere Belastung) | Rot: Wert > 6.5 (hohe Belastung). Bei allen anderen KPIs ist die Logik umgekehrt: >= 7 = grün, >= 6 = gelb, < 6 = rot.

### 3.4 Energie

Quelle: Q4 ("Wie energiegeladen fühlst du dich?")

Berechnung: Arithmetischer Mittelwert aller Q4-Antworten der aktuellen Periode, gerundet auf 1 Nachkommastelle.

```
SELECT ROUND(AVG(q_energie)::numeric, 1) AS kpi_energie
FROM feedback_responses
WHERE period_start = :current_period_start;
```

### 3.5 Fluktuation (KI-berechnet)

Quelle: Nicht direkt abgefragt. Wird aus den Mustern der 4 anderen KPIs berechnet.

Der Fluktuations-KPI wird als prognostizierte Fluktuationsrate in Prozent angezeigt (z.B. 12%). Er gibt an, wie hoch das geschätzte Risiko ist, dass Mitarbeitende das Unternehmen verlassen. Die Berechnung basiert auf folgender Heuristik als MVP-Lösung (kann später durch ein ML-Modell ersetzt werden):

```
// Fluktuation Risk - Heuristik v1
// Gibt eine prognostizierte Fluktuationsrate in % zurueck (0-50%)
function calculateFluktuationRisk(data) {
  // data = letzte 4 Perioden der aggregierten KPI-Werte
  let riskPct = 12.0; // Basis: deutscher Durchschnitt ~12%

  // --- Aktuelle Werte ---
  const current = data[data.length - 1];

  // Niedriges Engagement ist der staerkste Praedikator
  if (current.engagement < 5) riskPct += 15;
  else if (current.engagement < 6) riskPct += 8;
  else if (current.engagement < 7) riskPct += 3;
  else if (current.engagement >= 8) riskPct -= 4;

  // Hohe Belastung erhoht Risiko
  if (current.belastung > 7.5) riskPct += 8;
  else if (current.belastung > 6.5) riskPct += 4;
  else if (current.belastung <= 4) riskPct -= 3;

  // Schlechte Fuehrung erhoht Risiko stark
  if (current.fuehrung < 5) riskPct += 12;
  else if (current.fuehrung < 6) riskPct += 5;

  // Niedrige Energie erhoht Risiko
  if (current.energie < 5) riskPct += 8;
  else if (current.energie < 6) riskPct += 3;

  // --- Trend-Analyse (letzte 4 Perioden) ---
  if (data.length >= 4) {
    const engTrend = trendSlope(data.map(d => d.engagement));
    const eneTrend = trendSlope(data.map(d => d.energie));

    // Fallender Engagement-Trend ueber 4+ Wochen
```

```

if (engTrend < -0.3) riskPct += 8;
else if (engTrend < -0.1) riskPct += 3;

// Fallender Energie-Trend
if (eneTrend < -0.3) riskPct += 5;

// Toxische Kombination: sinkendes Engagement + steigende Belastung
const belTrend = trendSlope(data.map(d => d.belastung));
if (engTrend < -0.1 && belTrend > 0.1) riskPct += 8;
}

// Clamp auf 0-50%
return Math.round(Math.min(50, Math.max(0, riskPct)) * 10) / 10;
}

// Fuer den PULS: Prozent zu 1-10 Skala konvertieren
function fluktuationPctToScore(pct) {
  // 0% = 10 (kein Risiko), 10% = 7.5, 20% = 5.0, 40% = 0 -> clamped 1
  return Math.min(10, Math.max(1, 10 - (pct / 4)));
}

// Einfache lineare Trendsteigung
function trendSlope(values) {
  const n = values.length;
  const xMean = (n - 1) / 2;
  const yMean = values.reduce((a, b) => a + b, 0) / n;
  let num = 0, den = 0;
  for (let i = 0; i < n; i++) {
    num += (i - xMean) * (values[i] - yMean);
    den += (i - xMean) ** 2;
  }
  return den === 0 ? 0 : num / den;
}

```

### Hinweis: Fluktuation wird als Prozent angezeigt

Der Fluktuations-KPI wird im Dashboard als Prozentwert dargestellt (z.B. "12%"), nicht auf der 1-10 Skala. Ampelfarben: <= 10% grün (niedriges Risiko), <= 20% gelb (mittleres Risiko), > 20% rot (hohes Risiko). Für den PULS wird intern mit fluktuationPctToScore() auf eine 1-10 Skala konvertiert.

## 4. Der PULS (Gesamtwert)

Der PULS ist die zentrale Hero-Metrik des Produkts. Er beantwortet die Frage: "Wie geht es unserem Unternehmen?" auf einer Skala von 1 bis 10. Er wird aus den 5 Einzel-KPIs gewichtet berechnet.

### Namenskonvention

Der Gesamtwert heißt im gesamten Produkt, in der UI und in der Kommunikation immer nur "der PULS". Kein "Score", kein "Wert", kein "Index". Beispiel: "Euer PULS steht diese Woche bei 7.2". In Variablen und Datenbank: `puls_score`.

### 4.1 Gewichtung

KPI	Gewicht	Polarität	Begründung
Engagement	30%	direkt	Stärkster Prädiktor für Produktivität und Bindung (Gallup)
Energie	20%	direkt	Frühindikator für Motivation und Innovationskraft
Führung	20%	direkt	Stärkster Einzelfaktor für Fluktuation (McKinsey)
Belastung	15%	invertiert	Burnout-Prävention, wird als (10 - Wert) eingerechnet
Fluktuation	15%	% invertiert	Anzeige in %, wird intern zu 1-10 konvertiert

### 4.2 Berechnungsformel

```
function calculatePULS(kpis) {
  // kpis = { engagement, energie, fuehrung, belastung, fluktuation_pct }
  // engagement, energie, fuehrung, belastung: Skala 1-10
  // fluktuation_pct: Prozentwert (z.B. 12 fuer 12%)

  // Fluktuation von % zu 1-10 konvertieren
  const flukScore = fluktuationPctToScore(kpis.fluktuation_pct);

  const puls = (
    (kpis.engagement      * 0.30) + // direkt: hoch = gut
    (kpis.energie         * 0.20) + // direkt: hoch = gut
    (kpis.fuehrung        * 0.20) + // direkt: hoch = gut
    ((10 - kpis.belastung) * 0.15) + // invertiert: hoch = schlecht
    (flukScore            * 0.15)   // bereits invertiert durch
    Konvertierung
  );

  // Runden auf 1 Nachkommastelle, Clamp 1-10
```

```

return Math.round(
    Math.min(10, Math.max(1, puls)) * 10
) / 10;
}

```

### 4.3 Rechenbeispiel

KPI	Rohwert	Normalisiert	Gewicht	Beitrag
Engagement	7.4	7.4	0.30	$7.4 \times 0.30 = 2.22$
Energie	6.9	6.9	0.20	$6.9 \times 0.20 = 1.38$
Führung	7.1	7.1	0.20	$7.1 \times 0.20 = 1.42$
Belastung	5.8	$10 - 5.8 = 4.2$	0.15	$4.2 \times 0.15 = 0.63$
Fluktuation	12%	Score: $10 - (12/4) = 7.0$	0.15	$7.0 \times 0.15 = 1.05$
<b>PULS</b>			<b>1.00</b>	<b>= 6.7</b>

Ergebnis: Der PULS steht bei 6.7 — im gelben Bereich. Die Hauptursache ist die erhöhte Belastung (5.8 = gelb) und das moderate Fluktuationsrisiko (12% = gelb).

## 5. Ampelsystem (Traffic Light)

Alle KPIs und der PULS werden visuell als Ampel dargestellt. Die Schwellenwerte sind:

	Grün (gut)	Gelb (Achtung)	Rot (kritisch)
<b>PULS</b>	$\geq 7.0$	$\geq 5.5$ und $< 7.0$	$< 5.5$
<b>Engagement</b>	$\geq 7.0$	$\geq 6.0$ und $< 7.0$	$< 6.0$
<b>Energie</b>	$\geq 7.0$	$\geq 6.0$ und $< 7.0$	$< 6.0$
<b>Führung</b>	$\geq 7.0$	$\geq 6.0$ und $< 7.0$	$< 6.0$
<b>Belastung (invertiert!)</b>	$\leq 5.5$	$> 5.5$ und $\leq 6.5$	$> 6.5$
<b>Fluktuation (in %)</b>	$\leq 10\%$	$> 10\%$ und $\leq 20\%$	$> 20\%$

### 5.1 Farb-Hex-Codes

```
// Ampelfarben (UI)
const STATUS_COLORS = {
  green: { main: '#34D399', bg: '#ECFDF5', text: '#059669' },
  yellow: { main: '#FBBF24', bg: '#FFFBEF', text: '#D97706' },
  red: { main: '#F87171', bg: '#FEE2E2', text: '#DC2626' },
};

// Statusfarbe bestimmen
function getStatus(value, kpiType) {
  if (kpiType === 'belastung') {
    if (value <= 5.5) return 'green';
    if (value <= 6.5) return 'yellow';
    return 'red';
  }
  if (kpiType === 'fluktuation') {
    // Fluktuation wird in % angezeigt
    if (value <= 10) return 'green';
    if (value <= 20) return 'yellow';
    return 'red';
  }
  if (kpiType === 'puls') {
    if (value >= 7.0) return 'green';
    if (value >= 5.5) return 'yellow';
    return 'red';
  }
  // Standard (Engagement, Energie, Führung)
  if (value >= 7.0) return 'green';
  if (value >= 6.0) return 'yellow';
  return 'red';
}
```

## 6. PULS-Warnlogik

Ein grüner PULS kann täuschen, wenn ein Einzelwert kritisch ist. Deshalb gilt folgende Regel:

### Kritische Einzelwert-Warnung

Wenn der PULS  $\geq 7.0$  (grün) ist, aber mindestens ein Einzel-KPI im roten Bereich liegt, wird der PULS NICHT grün dargestellt, sondern mit einem Warnsymbol versehen. Beispiel-Anzeige:  
"PULS 7.1 — Belastung kritisch"

```
function getPulsDisplay(puls, kpis) {
  const pulsStatus = getStatus(puls, 'puls');
  const criticalKpis = [];

  if (getStatus(kpis.engagement, 'engagement') === 'red')
    criticalKpis.push('Engagement');
  if (getStatus(kpis.fuehrung, 'standard') === 'red')
    criticalKpis.push('Fuehrung');
  if (getStatus(kpis.belastung, 'belastung') === 'red')
    criticalKpis.push('Belastung');
  if (getStatus(kpis.energie, 'standard') === 'red')
    criticalKpis.push('Energie');
  if (getStatus(kpis.fluktuation_pct, 'fluktuation') === 'red')
    criticalKpis.push('Fluktuation');

  return {
    value: puls,
    status: pulsStatus,
    hasWarning: criticalKpis.length > 0,
    warningText: criticalKpis.length > 0
      ? criticalKpis.join(', ') + ' kritisch'
      : null,
  };
}
```

## 7. Aggregationsebenen

Alle KPIs und der PULS werden auf mehreren Ebenen berechnet:

Ebene	Beschreibung	Sichtbarkeit
Unternehmen	Alle Mitarbeiter, alle Teams	CEO Dashboard (Hero), Public Monitor, E-Mails
Team	Nur Mitarbeiter eines Teams	CEO Dashboard (Team-Heatmap, Detail-View)
Trend (12 Wochen)	Historische Werte pro Periode	Sparklines, Trend-Charts, Trendpfeil

### Anonymitätsregel: Mindestgröße 5

Teams mit weniger als 5 Feedback-Antworten in einer Periode werden NICHT als eigenes Team angezeigt. Stattdessen fließen ihre Werte nur in den Unternehmensdurchschnitt ein. Dies verhindert Rückschlüsse auf Einzelpersonen.

### 7.1 Trend-Berechnung

Für jeden KPI und den PULS wird der Trend als Differenz zur Vorperiode angezeigt:

```
const trend = currentValue - previousValue;
// Anzeige:
// trend > 0:  grüner Pfeil nach oben   "+ 0.3"
// trend < 0:  roter Pfeil nach unten   "- 0.5"
// trend == 0: grauer Strich           "+-0"

// ACHTUNG bei Belastung und Fluktuation (invertiert):
// Wenn Belastung STEIGT, ist der Trend-Pfeil ROT (nicht gruen!)
// Wenn Belastung SINKT, ist der Trend-Pfeil GRUEN
// Fluktuation: wird in % angezeigt, sinkend = gruen, steigend = rot
function getTrendColor(trend, kpiType) {
  if (kpiType === 'belastung' || kpiType === 'fluktuation') {
    return trend > 0 ? 'red' : trend < 0 ? 'green' : 'gray';
  }
  return trend > 0 ? 'green' : trend < 0 ? 'red' : 'gray';
}
```

## 8. Sichtbarkeitsregeln nach View

Das Produkt hat verschiedene Ansichten mit unterschiedlichen Datenfreigaben:

Datenpunkt	CEO Dashboard	Public Monitor	E-Mail Report	API
PULS (Gesamt)	Ja	Ja	Ja	Ja
Engagement	Ja	Ja	Ja	Ja
Energie	Ja	Ja	Ja	Ja
Führung	Ja	Ja	Ja	Ja
Belastung	Ja	Ja	Ja	Ja
Fluktuation	Ja	NEIN	Ja	Ja
Team-Breakdown	Ja	NEIN	Optional	Ja
KI-Empfehlungen	Ja	NEIN	Ja	Nein
Teilnahmequote	Ja	Ja	Ja	Ja

### Begründung der Einschränkungen beim Public Monitor

Der Public Monitor zeigt KEIN Fluktuationsrisiko (erzeugt Unsicherheit), KEINE KI-Empfehlungen (interne Führungshinweise) und KEINEN Team-Breakdown (kann einzelne Teams bloßstellen). Er zeigt nur positiv gerahmte, aggregierte Metriken.

— Ende des Briefings —