

UNICEF – Kinder in Not: Überblick

- UNICEF wurde 1946 gegründet, ist die Kinderhilfsorganisation der Vereinten Nationen und in 192 Ländern aktiv (32 Komitees weltweit)
- Schwerpunkt und Hauptauftrag:
 - Weltweites Sicherstellen der Kinderrechte für alle Kinder, unabhängig von Herkunft, Religion oder Hautfarbe
 - Jedem Kind eine Kindheit zu ermöglichen (Gesundheit, Bildung, Schutz und Entwicklungschancen), insbesondere bei Konflikten, Naturkatastrophen und Pandemien
- Unterstützt langfristige Entwicklung und Stabilität
- Zusammenarbeit mit Kindern, Gemeinden, zivilen Organisationen, Wirtschaft und Regierungen
- Programme kombinieren praktische Hilfe (Brunnen, Impfungen, Schulen) mit politischer Arbeit und Beratung der Regierungen
- Entwickelt keine Insellösungen, sondern übertragbare Leuchtturmprojekte (Orientierung/Vorbild)
- Unterstützt die Agenda 2030, mit dem Ziel, die Welt gerechter, nachhaltiger und lebenswerter zu machen, da Kinderrechte essenziell für nachhaltige Entwicklung sind
- Informiert, sammelt Spenden und sensibilisiert für Kinderrechte – durch Öffentlichkeits- und Bildungsarbeit mit Ehrenamtlichen
- Finanzierung erfolgt überwiegend durch freiwillige Finanzierungen von Regierungen, Privatpersonen und Unternehmen
- Gemeinnütziger Verein (nicht staatlich), Erfolg wird nicht in Gewinn gemessen, sondern Anzahl geimpfter Kinder, Zugang zu sauberem Wasser, Bildungschancen verbessert etc.
- Unterliegt internationalen Ethikstandards: Kinderschutz, Transparenz, Verantwortung etc.

Deutsches Komitee für UNICEF

- 1953 gegründet, 7000 Ehrenamtler in 184 UNICEF Gruppen, 200 Mitarbeitende (bürgerliches Engagement, Digital & Technologie, Finanzen und Verwaltung, Kommunikation, Kinderrechte und Marketing)
- Arbeit für Kinder bekanntmachen und Menschen zum Spenden bewegen
- Förderung des Engagements von Erwachsenen, Kindern und Jugendlichen für das Wohl der Kinder und ihrer Rechte

Aktuelle Einsatzländer und Programme (Stand: 2025)

- Ukraine, Libanon, Syrien und Nachbarländer
- Afghanistan: Gesundheitsleistungen, Ernährung, sauberes Wasser und psychosoziale Betreuung
- Sudan: Wasser, Hygiene, Schutz vor Ausbeutung, akuter Mangelernährung
- Palästina / Gaza: Behandlung mangelernährter Kinder
- Kongo: Einsatz gegen Cholera, Kinderarbeit und sexualisierte Gewalt in bewaffneten Konflikten

Künftige strategische Schwerpunkte von UNICEF (ab 2025)

- Ausbau von Frühwarnsystemen und datenbasierter Risikoplanung in Krisenregionen
- Resilienzförderung = bessere Krisenbewältigung und nachhaltiges Erholen
- Verbesserung digitaler Lernangebote für Kinder in abgelegenen oder gefährdeten Regionen
- Aufbau lokaler Kapazitäten, um Abhängigkeiten von Hilfslieferungen zu reduzieren
- Innovation bei der Nutzung von Daten, Satellitenbildern und KI zur besseren Zielgruppenansprache und Bedarfsplanung

Aktuelle Projekte im Data Engineering (BI & Controlling)

DWH Entwicklung

- Entwicklung und Optimierung von ETL-Prozessen zur Datenintegration aus verschiedensten Quellen (z.B. Partnerorganisationen, Gesundheitsdaten) für BI- und Controllingzwecke
- Aufbau und Pflege eines DWHs zur Speicherung von Monitoring- und Krisendaten für Dashboard-Anwendungen
- Umsetzung von Dashboards für Echtzeitvisualisierung von Projektfortschritten, Hilfsleistungen und KPIs zur Unterstützung der Steuerung in Krisenregionen

Supply Chain Intelligence & Dashboarding

- DWH Optimierung durch Integration von Supply-Chain- und Demand-Planungsdaten
- Ziel: Bereitstellung konsistenter Supply-Master-Daten zur besseren BI-Nutzung und zum Monitoring kritischer Versorgungssituationen

Relevante Aufgabenbereiche für BI & Controlling

- Konzeption, Entwicklung und Wartung robuster ETL-Prozesse für Datenintegration, -qualität und Automatisierung
- Aufbau, Administration und Weiterentwicklung von DWHs und BI-Plattformen zur zentralen Datenanalyse
- Erstellung und Pflege von Dashboards und Reports für Controlling und Monitoring

Beispiele für datenorientierte Entscheidungen

- Analyse von Konflikt- und Fluchtdaten, um Krisengebiete zu identifizieren, in denen Kinder akut gefährdet sind
- Verknüpfung von Erhebungsdaten mit Monitoring-Systemen zur Bewertung der Wirksamkeit früherer Notfallmaßnahmen

Werdegang

- Studium der Physik
- Anschließend Berufliche Weiterbildung zum ML-Entwickler
- Einstieg als klassischer Software-Engineer bei Capgemini
- Wechsel in die Einheit I&D als Data Scientist / Data Engineer
- Erstes Projekt: Luftfahrt und Aircraft Health Management
 - Verantwortlich für ganzheitliche Betreuung von Use Cases
 - Exploratorische Datenanalyse, Bereinigung und Modellierung der Daten (ETL)
 - Deskriptive und präskriptive Analytics
 - Erstellung von klassischen Algorithmen mittels PySpark
 - Visualisierung der Datenprodukte in Dashboards
- Weiteres Projekt: Automobilbranche, Migrationsprojekt in Palantir und Dashboard Erstellung
 - Backend Lead eines kleinen Teams für Erstellung eines Management-Dashboards im Bereich Qualität und Versorgung
 - Verantwortlich für Datenmodellierung, -qualität, Erstellung der Datenprodukte für Dashboard
- Letztes Projekt: Automobilbranche
 - Analyse von digitalen Nachkäufen und Bereitstellung von Potentialen für digitale Produkte
 - Markt- und Konsumentenanalysen
 - Erstellung von Dashboards für KPI-Analyse

Wechselgründe

- Mir fehlt seit einiger Zeit die fachliche/technische Weiterentwicklung in meiner Rolle
- Ich möchte meine berufliche Entwicklung aktiv mitgestalten, meine Stärken gezielt einsetzen und vor allem weiterentwickeln
- In der Beratung werden Projekte oft von externen Trends bestimmt → Ich suche Themen, die meinen Interessen und Kompetenzen entsprechen
- Ich strebe eine Aufgabe an, die nicht nur für mich sinnstiftend ist, sondern idealerweise auch einen positiven, greifbaren Beitrag für die Gesellschaft leistet

Fragen

- Wie läuft die Einarbeitung?
- Wie groß ist das Team?
- Wie wird gearbeitet (SCRUM etc.)?
- Wer sind übliche Stakeholder?
- Klassische Projekte und Use Cases?
- Wie oft Feedbackgespräche?
- Wie läuft der gezielte Einsatz in Projekten ab?
- Interne Weiterbildungen → Wer schlägt vor oder kann man wählen?
- Wie lange dauern Projekte für gewöhnlich?
- Wie sehen die langfristigen Aufstiegsmöglichkeiten aus?
- Wie ist die HO-Regelung, wie oft sind die Kollegen vor Ort?
- Weiteres Vorgehen?
- Zu wann ist die Stelle zu besetzen?

Data Warehouse

Grundlegendes

- Zentrales, strukturiertes System zur langfristigen Speicherung und Analyse großer Datenmengen aus verschiedenen Quellen
- Optimierte für komplexe Abfragen und BI-Anwendungen
- Fokus liegt auf strukturierten, sauberen und konsistenten Daten, die für BI, Reporting und Data Analytics genutzt werden

Eigenschaften

- Themenorientierung: Daten werden nach Themen (Kunden, Produkte, Verkäufe) organisiert, nicht nach einzelnen Transaktionen
- Integration: Daten aus verschiedenen Quellen werden vereinheitlicht
- Zeitbezug: Speicherung historischer Daten für Analyse von Trends und Entwicklungen

Architektur

- ETL-Prozess: Daten werden extrahiert, aufbereitet und ins DWH geladen
- Storage: Relationale oder spaltenbasierte Datenbanken, optimiert für große Datenmengen und schnelle Abfragen
- Metadatenmanagement: Beschreibung der Datenquellen, Transformationen und Datenqualität
- Front-End Tools: BI-Tools, Dashboards, Reporting-Systeme, die auf das DWH zugreifen

Betrieb

- Monitoring und Performanceoptimierung für Abfragen, Speicherplatz und Ladeprozesse
- Datensicherheit und -schutz: Zugriffsrechte, Verschlüsselung und Compliance sind zentral
- Datenqualität: Daten werden validiert, Duplikate bereinigt und Fehlerquellen minimiert
- Wartung und Updates: Regelmäßige Wartung der Datenbanksoftware, Anpassungen an neue Datenquellen und Business-Anforderungen
- Backup und Recovery: Strategien zur Sicherung und Wiederherstellung der Daten

Datenstrukturen

- Methoden, wie Daten intern dargestellt und organisiert werden, damit man effizient darauf zugreifen, sie verarbeiten und speichern kann
- Beispiele:
 - Tabellen und relationale Strukturen
 - Arrays und Listen (JSON, Parquet)
- Wichtigkeit:
 - Effizienz: Daten müssen schnell geladen, transformiert und abgefragt werden können
 - Speicherplatz: Optimale Datenstrukturen sparen Speicherplatz
 - Kompatibilität: Unterschiedliche Systeme nutzen unterschiedliche Formate
 - Skalierbarkeit: Große Datenmengen erfordern effiziente Strukturen

Datenqualität

- Vollständigkeit: Sind alle erforderlichen Datenfelder vorhanden und ausgefüllt?
- Korrektheit: Entsprechen die Daten den realen, erwarteten Werten (PLZ $\hat{=}$ Stadt)
- Konsistenz: Sind die Daten in verschiedenen Systemen oder Tabellen widerspruchsfrei (z.B. gleicher Kundennamen in allen Datensätzen)
- Aktualität: Sind die Daten aktuell (z.B. keine veralteten Transaktionen im Reporting)
- Eindeutigkeit: Gibt es doppelte Datensätze, wo es keine geben sollte (z.B. doppelte Kunden-IDs)
- Validität: Entsprechen die Daten den erwarteten Formaten oder Regeln (z.B. E-Mail-Adressen im gültigen Format, Zahlen in numerischen Feldern)

Entwicklung von Datenladestrecken

Wichtiges

- UNICEF arbeitet mit sehr unterschiedlichen Quellsystemen → Unterschiedliche Formate und Strukturen:
 - Administrative Daten (z.B. Personal, Finanzen)
 - Feldberichte (z.B. Gesundheitsdaten, Bildungsstatistiken)
 - Externe Quellen (Regierungsdaten, Partnerorganisationen)
- Datenintegration & Harmonisierung: Vereinheitlichung unterschiedlicher Datenformate
- Datenqualität sicherstellen: Fehlerhafte, unvollständige oder doppelte Daten sind häufig → Validierung und Bereinigung sind zentral
- Automatisierung & Wiederholbarkeit:
 - Datenladestrecken müssen regelmäßig laufen
 - Automatisierte Jobs, die bei Fehlern Alarm schlagen und selbstständig neu starten
- Skalierbarkeit & Performance: Datenmengen können schnell wachsen, vor allem bei Feldberichten → Effiziente Ladeprozesse, z.B. inkrementelles Laden (nur neue/geänderte Daten)
- Sicherheit & Datenschutz: Schutz sensibler Daten (z.B. persönliche Gesundheitsdaten)
- Transparenz & Dokumentation: Klare Dokumentation der ETL-Prozesse, Datenherkunft (Lineage) und Transformationsregeln → Besonders wichtig bei internationalen Organisationen für Audits und Reporting

Probleme

- Heterogenität der Quellsysteme: Unterschiedliche Formate und fehlende Standardisierung
 - Entwicklung modularer Pipelines, die Daten aus verschiedenen Formaten aufnehmen und in ein einheitliches Zwischenformat (z.B. Parquet, JSON) transformieren
- Datenqualität: Unvollständige oder fehlerhafte Daten führen zu falschen Analysen
 - Automatisierte Validierung auf Vollständigkeit, Konsistenz, Wertebereiche
 - Alerts bei Datenanomalien, damit Teams schnell reagieren können
- Komplexe Transformationen: Oft müssen Daten stark angepasst werden, was Fehlerquellen erhöht
- Fehlende Automatisierung: Manuelle Prozesse sind fehleranfällig und nicht skalierbar
- Performance-Engpässe: Große Datenmengen können Ladeprozesse verlangsamen
 - Inkrementelles Laden von Daten, um nur Änderungen zu verarbeiten
 - Nutzung von spaltenorientierten Speicherformaten für schnellere Analysen
- Sicherheitsanforderungen: Schützen sensibler Daten, was zusätzliche Komplexität bringt
 - Datenklassifikation und Anonymisierung oder Pseudonymisierung sensibler Daten

Kontext UNICEF

- UNICEF arbeitet oft in schwer zugänglichen Regionen, wo Daten nicht standardisiert oder unvollständig sind
- Daten kommen aus verschiedenen Ländern mit unterschiedlichen IT-Systemen und Sprachen
- Rechtliche und ethische Anforderungen im Umgang mit sensiblen Daten (z.B. von Kindern) sind besonders streng
- Häufig ist eine enge Zusammenarbeit mit lokalen Partnern nötig, um Datenquellen zu verstehen und zu verbessern

API Anbindung

- Schnittstelle, über die verschiedene Software-Systeme Daten austauschen
- API-Anbindung: Kommunikation von Systemen über definierte Protokolle (z.B. REST), die Daten senden oder empfangen
- UNICEF integriert Daten aus vielen internationalen Partnern, Regierungen und eigenen Systemen. APIs sind zentral für:
 - Feld-Datenerfassung: Mobile Apps in Krisengebieten senden Gesundheits- oder Bildungsdaten per API an zentrale Datenbanken
 - Partner-Daten: Automatisierter Austausch mit NGOs, WHO, UN-Systemen
 - Monitoring & Reporting: Echtzeit-Updates für Fortschrittsberichte und Entscheidungsfindung

Plattform Monitoring

- Kontinuierliche Überwachung von IT-Systemen, Infrastruktur und Anwendungen, um deren Verfügbarkeit, Performance und Sicherheit sicherzustellen
- Ziel: Probleme frühzeitig erkennen, Ausfälle vermeiden und schnelle Fehlerbehebung ermöglichen

Wichtige Bereiche beim Plattform-Monitoring

- Infrastruktur-Monitoring: Überwachung von Hardware-Komponenten (CPU-, Netzwerkauslastung, Latenz)
- Anwendungs-Monitoring: Überwachung der Verfügbarkeit und Leistung von Software-Anwendungen und Services (Antwortzeiten, Fehlerquoten, Transaktionsvolumen)
- Log-Management: Sammlung und Analyse von System- und Anwendungs-Logs
- Sicherheits-Monitoring: Überwachung von Zugriffen, verdächtigen Aktivitäten und Angriffen
- Benutzer- und Zugriffsmonitoring: Überwachung von Nutzeraktivitäten und Berechtigungen

Wichtige Funktionen und Methoden

- Alerting: Automatische Benachrichtigung bei Überschreitung von Schwellenwerten
- Dashboards: Übersichtliche Visualisierung der Systemzustände in Echtzeit
- Trendanalysen: Historische Daten zur Kapazitätsplanung und Problemprevention
- Self-Healing: Automatisierte Reaktionen auf bestimmte Probleme (z.B. Neustart eines Dienstes)

Tools für Plattform-Monitoring (Beispiele)

- Prometheus & Grafana: Open-Source-Monitoring und Visualisierung

- Visualisierung von Metriken und Zeitreihendaten aus verschiedenen Datenquellen
- Erstellung interaktiver Dashboards zur Überwachung von Systemen und Anwendungen
- Alarmierung und Benachrichtigungen bei Grenzwertüberschreitungen

Herausforderungen im Plattform-Monitoring

- Integration und Überwachung heterogener Systeme
- Skalierbarkeit: Monitoring-Lösung muss mitwachsen, z.B. bei Cloud-Plattformen
- Sicherheit: Monitoring-Daten müssen geschützt und nur autorisierten Personen zugänglich sein

Plattform-Monitoring im Kontext von UNICEF

- UNICEF betreibt oft verteilte Systeme und globale Datenplattformen. Monitoring ist entscheidend, um Ausfälle in Krisengebieten schnell zu erkennen und Datenverluste zu vermeiden
- Monitoring unterstützt die Einhaltung von Datenschutz und Sicherheit, vor allem bei sensiblen Daten von Kindern
- Echtzeit-Überwachung ermöglicht schnelles Reagieren auf technische/sicherheitsrelevante Probleme

MLOps

Grundlegendes zu MLOps

- Praxis, ML-Modelle effizient, reproduzierbar und skalierbar zu entwickeln, zu deployen und in Betrieb zu halten
- Automatisierung und Überwachung des gesamten ML-Lifecycles
- Ziel: Schnelle und sichere Bereitstellung von ML-Modellen in produktiven Umgebungen sowie kontinuierliche Verbesserung und Wartung

Prinzipien von MLOps

- Automatisierung: Datenaufbereitung, Modelltraining, Testing, Deployment
- Reproduzierbarkeit: Sicherstellen, dass Modelle und Ergebnisse nachvollziehbar und reproduzierbar sind
- CI/CD: Automatisierte Pipelines, die Code, Daten und Modelle integrieren und aktualisieren
- Skalierbarkeit: Modelle und Infrastruktur sollen einfach skalierbar sein, z.B. bei steigender Datenmenge oder Nutzerzahlen
- Monitoring und Feedback: Überwachung der Modelle im Betrieb, um Leistungseinbußen frühzeitig zu erkennen

Konzepte in MLOps

- Versionierung: Versionierung von Code, Daten und Modellen (z.B. mit Git, MLflow)
- Pipeline-Orchestrierung: Automatisierte Abläufe für Datenvorverarbeitung, Training, Evaluation und Deployment (z.B. mit Apache Airflow)
- Testen: Unit Tests, Integrationstests, Modelltests (z.B. Validierung der Genauigkeit)
- Feature Store: Zentralisierte Speicherung und Verwaltung von Features zur Wiederverwendung
- Governance und Compliance: Dokumentation und Einhaltung von Datenschutz und Richtlinien

Betrieb und Monitoring

- Modell-Monitoring: Überwachung von Modellmetriken wie Genauigkeit, Latenz, Antwortzeiten
- Daten-Monitoring: Überwachung der Eingabedaten auf Drift, Anomalien oder veränderte Verteilungen
- Alerting: Automatische Benachrichtigung bei Leistungsabfall oder Fehlern
- Automatisiertes Retraining z.B. bei Datenverschiebung
- Logging und Auditing: Nachvollziehbarkeit aller Aktionen und Entscheidungen im ML-System

CI/CD Pipelines

- CI: Automatisches Testen und Bauen bei jeder Codeänderung
- CDelivery: Automatisches Ausliefern in eine Staging-Umgebung
- CDeployment: Automatisches Ausliefern bis in die Produktion (ohne manuelle Eingriffe)

Automatisierte Jobs mit MLflow

- Plattform zur Verwaltung des ML-Lifecycles mit Fokus auf Automatisierung und Reproduzierbarkeit
- Es unterstützt vier Kernkomponenten: MLflow-
 - Tracking: Protokollierung von Experimenten, Parametern, Metriken und Artefakten
 - Projects: Strukturierung und Verpackung von ML-Code zur reproduzierbaren Ausführung
 - Models: Verwaltung und Bereitstellung von ML-Modellen in verschiedenen Formaten und Umgebungen
 - Registry: Zentralisierte Modell-Repository mit Versionierung, Genehmigung und Lifecycle-Management
- Automatisierung wird durch standardisierte Pipelines erreicht, in denen Training, Evaluation und Deployment orchestriert werden
- Integration mit CI/CD-Tools ermöglicht automatische Ausführung von Experimenten und Updates
- MLflow erleichtert die Nachverfolgbarkeit und Reproduzierbarkeit
- Unterstützt verschiedene ML-Frameworks und Deployment-Optionen (z.B. REST APIs, Batch)