

Kurzbeschreibung des Algorithmus (Critical Path Method, CPM)

1. **Modell:** Der Netzplan wird als gerichteter, azyklischer Graph (DAG) mit Aktivitäten (Knoten) und Abhängigkeiten (Kanten) modelliert.
2. **Topologische Sortierung:** Mit Kahn's Algorithmus wird eine Reihenfolge ohne Regelverletzung berechnet. Dabei werden Zyklen bzw. fehlende Knoten implizit erkannt (Robustheit).
3. **Vorwärtsrechnung (Forward Pass):** Für jede Aktivität werden Frühester Start (ES) und Frühestes Ende (EF) bestimmt. Der ES einer Aktivität ist das Maximum der EF aller Vorgänger; es gilt $EF = ES + \text{Dauer}$. Die Projektdauer ergibt sich als Maximum aller EF.
4. **Rückwärtsrechnung (Backward Pass):** Ausgehend vom Projektende werden Spätestes Ende (LF) und Spätester Start (LS) bestimmt. Ohne Nachfolger gilt $LF = \text{Projektdauer}$; sonst $LF = \min(\text{LS der Nachfolger})$; anschließend $LS = LF - \text{Dauer}$.
5. **Puffer (Slack) und kritische Aktivitäten:** Der Puffer ergibt sich als $Slack = LS - ES$. Aktivitäten mit $Slack = 0$ sind kritisch. Ein kritischer Pfad lässt sich deterministisch als geordnete Liste aller Aktivitäten mit $Slack = 0$ angeben (z. B. sortiert nach ES).

Komplexität: Alle Schritte laufen in $\mathcal{O}(V + E)$, wobei V die Anzahl der Aktivitäten und E die Anzahl der Abhängigkeiten bezeichnet.

Zentrale Implementierungsentscheidungen und Auswirkungen auf die Softwarequalität

A) Verwendung von Kahn's Algorithmus (Topologische Sortierung) statt rekursiver DFS

Vorteile: Explizite Zyklenerkennung und stabile Reihenfolge der Verarbeitung (Determinismus).

Qualitätswirkung: Erhöht Robustheit (Fehler früh erkennbar), verbessert Reproduzierbarkeit der Ergebnisse und damit Testbarkeit.

B) Klare Domänenmodellierung über eine Activity-Klasse

Details: Felder für ID, Dauer, Abhängigkeiten, ES/EF/LS/LF und Slack; plus getrennte Berechnungsmethoden im Analyzer.

Qualitätswirkung: Gute Lesbarkeit und Wartbarkeit; geringe Kopplung (Berechnungslogik zentral im Analyzer), hohe Testbarkeit einzelner Komponenten.

C) Deterministische Ausgabe des kritischen Pfads

Details: Reihenfolge der kritischen Aktivitäten wird stabil über Einfüge- bzw. Topo-Reihenfolge (bzw. ES) erzeugt.

Qualitätswirkung: Reproduzierbare Testergebnisse und zuverlässige Vergleichbarkeit in CI-Pipelines.

D) **Convenience-Methode** `project_duration()`

Details: Einfache, zentrale Ermittlung der Projektdauer (max. EF) ohne redundante Berechnungen in Aufrufern.

Qualitätswirkung: Reduziert Duplikation, verringert Fehleranfälligkeit und erleichtert Wiederverwendung (API-Klarheit).