

# Netzplantechnik

Udo Kelter

20.03.2003

## **Zusammenfassung dieses Lehrmoduls**

Die Netzplantechnik ist eine klassische Methode zur Projektplanung. Dieses Lehrmodul erklärt Aufbau, Struktur und Auswertung von Netzplänen. Netzpläne sind weiterhin eine Basis, auf der der Bedarf an Ressourcen, die zur Durchführung eines Projekts erforderlich sind, festgestellt werden kann. Durch eine geeignete Anordnung der Vorgänge muß man ggf. die erforderlichen Kapazitäten an die verfügbaren anpassen.

## **Vorausgesetzte Lehrmodule:**

obligatorisch: – Vorgehensmodelle  
– Projektmanagement

**Stoffumfang in Vorlesungsdoppelstunden:** 1.2

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Entwicklung von Netzplänen</b>	<b>4</b>
2.1	Zerlegung der Gesamtaufgabe in Vorgänge . . . . .	4
2.2	Zuordnung zu ausführenden Instanzen . . . . .	5
2.3	Identifizierung von Reihenfolgeabhängigkeiten . . . . .	6
2.4	Schätzung der Dauer einzelner Vorgänge . . . . .	7
2.5	Meilensteine . . . . .	7
<b>3</b>	<b>Auswertung des Netzplans</b>	<b>8</b>
<b>4</b>	<b>Alternative Reihenfolgebeziehungen</b>	<b>13</b>
<b>5</b>	<b>Strukturierung von größeren Netzplänen</b>	<b>17</b>
<b>6</b>	<b>Abgeleitete Dokumente</b>	<b>18</b>
<b>7</b>	<b>Einsatzplanung und Kapazitätsausgleich</b>	<b>18</b>
7.1	Verfügbare Ressourcen . . . . .	19
7.2	Ressourcenbedarf . . . . .	20
7.3	Kapazitätsausgleich . . . . .	21
	Literatur . . . . .	23
	Glossar . . . . .	23
	Index . . . . .	24

# 1 Einleitung

Die Netzplantechnik ist eine alte (oder sagen wir lieber: eine bewährte) Methode zur Termin- und Ressourcenplanung von Projekten (s. [PM]). Ihre Anwendung beschränkt sich nicht allein auf Software-Entwicklungsprojekte, sie kann auf nahezu alle Arten von Projekten angewandt werden, sei es die Entwicklung eines Toasters oder den Bau eines Hauses oder was auch immer. Das – aus Sicht der Informatik biblische – Alter von Netzplänen erkennt man u.a. daran, daß es bereits seit den 70er Jahren eine DIN-Norm (DIN 69900) für Netzpläne gibt.

Ein **Netzplan** ist ein (naheliegenderweise graphisch darzustellendes) Netzwerk, in dem die Knoten einzelne Vorgänge darstellen und die Kanten Reihenfolgeabhängigkeiten. Statt einer netzartigen Darstellung sind auch **Balkenpläne** (GANTT-Diagramme) üblich; Balkenpläne sind nur eine andere Darstellung der gleichen Informationen. In Balkenplänen wird jeder Vorgang durch eine eigene Zeile in einer Matrix dargestellt. Bild 1 zeigt einen Balkenplan und einen Netzplan, die die gleiche Information darstellen. Wir werden i.f. fast immer von Netzplänen reden; es sollte aber klar geworden sein, daß damit inhaltlich auch Balkenpläne abgedeckt werden.

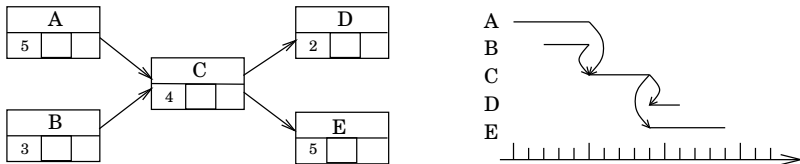


Abbildung 1: Vorgangsknotennetz vs. Balkenplan

Netzpläne dienen dazu, die folgenden Probleme bei der Projektplanung zu lösen:

- die Gesamtdauer eines Projekts einzuschätzen
- die Auswirkungen von Verzögerungen bei der Erledigung einzelner Aufgaben einzuschätzen

- den Einsatz von Ressourcen zu planen

## 2 Entwicklung von Netzplänen

Man kann vier Tätigkeiten unterscheiden, die bei der Entwicklung eines Netzplans anfallen:

1. Zerlegung der Gesamtaufgabe in Vorgänge
2. Zuordnung zu ausführenden Instanzen
3. Identifizierung von Reihenfolgeabhängigkeiten
4. Schätzung der Dauer einzelner Vorgänge

Wenn der Netzplan fertig erstellt ist, kann er durch Werkzeuge (oder von Hand) ausgewertet werden. Zunächst werden wir in den folgenden Abschnitten die genannten Tätigkeiten erläutern.

### 2.1 Zerlegung der Gesamtaufgabe in Vorgänge

Um mit Netzplänen arbeiten zu können, muß folgende Voraussetzung erfüllt sein: die Gesamtaufgabe muß sich in eine *feste* Menge von Teilaufgaben, die man auch als **Vorgänge** bezeichnet, zerlegen lassen. Jede Teilaufgabe wird genau einmal ausgeführt. Netzpläne sind nicht anwendbar bei iterativen Strukturen, in denen Teilaufgaben in Zyklen immer wieder erneut ausgeführt werden.

Als Beispiel betrachten wir die Aufgabe, auf einem frisch gekauften Rechner das Betriebssystem und die Anwendungssoftware zu installieren. Wir können grob folgende Teilaufgaben unterscheiden:

1. Hardware testen und ggf. einstellen
2. Platten partitionieren und formatieren
3. Betriebssystem installieren
4. Anwendungssoftware installieren

Die vorstehende Aufteilung ist qualitativ orientiert, da die einzelnen Teilaufgaben unterschiedliche Kenntnisse und Fähigkeiten seitens

der ausführenden Personen sowie ggf. andere Werkzeuge und Hilfsmittel erfordern. Motiviert ist eine solche Aufteilung insbesondere dadurch, daß ggf. verschiedene Personen benötigt werden, die entsprechende Qualifikationen aufweisen.

Die Qualifikationen, die für eine bestimmte Klasse von Aufgaben erforderlich sind, bezeichnet man oft als **Rolle**. Jemand, der die Platten unseres Rechners partitionieren und formatieren soll, muß die entsprechenden Systemprogramme kennen und einschlägige Erfahrung haben. Eine Person, die die Qualifikationen, die zu einer Rolle gehören, hat, kann diese Rolle einnehmen.

Angenommen, wir haben 2 Platten im Rechner und müssen 5 Anwendungspakete installieren; dann können wir die zweite und vierte Teilaufgabe entsprechend weiter verfeinern. Derartige Zerlegungen sind quantitativ orientiert. Sie könnten zwar bzgl. der Qualifikationen durch eine einzige Person ausgeführt werden, aufgrund des Arbeitsumfangs teilt man sie aber ggf. doch auf mehrere Personen auf.

Bei der Bildung eines Netzplans kann eine Aufgabe sowohl qualitativ als auch quantitativ orientiert zerlegt werden; für den Netzplan ist dieser Unterschied unerheblich.

Das vorstehende Beispiel zeigt ferner, daß eine Gesamtaufgabe ggf. in mehreren Stufen in immer kleinere Teilaufgaben zerlegt wird, so daß insgesamt ein Baum entsteht. Die Zahl der Verfeinerungsschritte ist im Prinzip beliebig. Normalerweise kommt man mit 2 - 3 Verfeinerungsstufen aus. Wenn man bei jedem Verfeinerungsschritt ca. 3 - 10 Teilaufgaben bildet, hat man bei 3 Baumebenen schon einige hundert Blätter.

Ein Blatt entspricht üblicherweise einem Arbeitsaufwand von ca. 1 - 10 Arbeitstagen.

## 2.2 Zuordnung zu ausführenden Instanzen

Jedem Vorgang und jeder Teilaufgabe muß eine ausführende bzw. verantwortliche Instanz zugewiesen werden. Die Instanz kann eine einzelne Person, eine Gruppe oder ein anderes Unternehmen sein.

Zusammen hiermit werden oft betriebswirtschaftliche Daten festgelegt, z.B. welche Stelle die anfallenden Kosten trägt, welche Art von Kosten vorliegt u.a.; hierauf gehen wir nicht näher ein.

## 2.3 Identifizierung von Reihenfolgeabhängigkeiten

Wir betrachten zunächst nur Reihenfolgeabhängigkeiten der Form, daß ein Vorgang erst begonnen werden kann, wenn davorliegende Vorgänge beendet worden sind.

Bei unserem oben erwähnten Rechner können die vier Teilaufgaben (Hardware einstellen, Platten partitionieren, ...) nur in dieser Reihenfolge ausgeführt werden. Die 5 Anwendungspakete könnten aber theoretisch parallel installiert werden. Im allgemeinen kann es zu einer Teilaufgabe beliebig viele **Vorgänger** bzw. **Nachfolger** geben.

Logisch gesehen liegt hier ein Graph vor, in dem die Vorgänge die Knoten sind und die Vorgänger-/Nachfolger-Beziehungen die Kanten. Dieser Graph muß *zyklusfrei* sein.

Die Vorgänger-/Nachfolger-Beziehungen können auf verschiedene Art und Weise dargestellt werden:

- tabellarisch: zu jedem Vorgang werden die Vorgänger und Nachfolger textuell z.B. durch eine Nummer angegeben.
- graphisch als Netzplan; es stehen zwei Arten von Netzplänen zur Auswahl:
  1. **Vorgangsknotennetze**: Bei diesen wird jeder Vorgang durch einen Knoten dargestellt. Kanten stellen die Vorgänger-/Nachfolger-Beziehungen dar. Die Ereignisse, daß ein Vorgang beginnt bzw. endet, werden hier nicht dargestellt.
  2. **Vorgangspfeilnetze**: Bei diesen wird jeder Vorgang durch einen Pfeil dargestellt (s. Bild 2). Knoten stellen Ereignisse (oder auch "Meilensteine" oder Synchronisationspunkte) dar; sie treten ein, wenn die "ankommenden" Vorgänge beendet worden sind<sup>1</sup>. Nur indirekt dargestellt werden die Reihenfolgebeziehungen zwischen den Vorgängen.

---

<sup>1</sup>In dieser Hinsicht sind sie vergleichbar mit Transitionen in einem Petri-Netz.

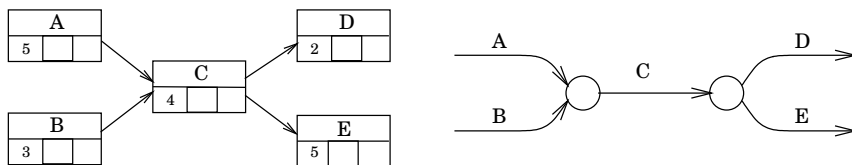


Abbildung 2: Vorgangsknotennetz vs. Vorgangspfeilnetz

- In Balkendiagrammen durch zusätzliche Pfeile (s. Bild 1)

Alle Varianten sind im Prinzip gleichwertig. Wir werden in diesem Text i.f. nur Vorgangsknotennetze verwenden.

## 2.4 Schätzung der Dauer einzelner Vorgänge

Nach der Zerlegung der Gesamtaufgabe ist die Zeitdauer der einzelnen Vorgänge zu schätzen. Mit Zeitdauer ist hier Kalenderzeit gemeint, nicht der Aufwand<sup>2</sup>. Wenn beispielsweise an einem Vorgang zwei Personen parallel arbeiten, die erste Person 6 Tage und die zweite 4 Tage, dann dauert der Vorgang 6 Tage, während der Aufwand 10 Personentage beträgt.

Die Einheit, in der die Zeitdauern geschätzt werden, sollte ca. 1 % der gesamten Projektdauer sein. Bei Projekten, die 1 - 2 Jahre dauern, wären also Wochen als Einheit angebracht. Vorgänge, die wesentlich kürzer als eine Einheit dauern, sind im jeweiligen Kontext eher zu kleinteilig. Bei größeren Einheiten entstehen durch die notwendigen Rundungen zu große Fehler.

Für einen Vorgang  $v$  bezeichnen wir i.f. dessen Dauer mit  $D(v)$ .

## 2.5 Meilensteine

Der Sinn von Meilensteinen liegt darin, definierte Punkte zu haben, an denen der Fortschritt des Projekts überprüft wird. Mit einem Meilenstein verbunden ist daher eine Bedingung, die erfüllt sein muß, damit

<sup>2</sup>Auf die Aufwandsschätzung gehen wir in Abschnitt 7 noch näher ein.

der Meilenstein als erreicht gilt. Diese Bedingung muß möglichst objektiv und mechanisch überprüfbar sein. Objektiv prüfbar ist z.B. die Bedingung, daß ein Programm ohne Compilerwarnungen übersetzbar ist. Die Bedingung “das Programm ist zu 50 % fertig” ist hingegen unbrauchbar.

Daß ein Meilenstein erreicht ist, ist eigentlich kein Vorgang, sondern ein Ereignis. Ereignisse können in Vorgangsknotennetzen nicht direkt dargestellt werden. Ersatzweise verwendet man spezielle Vorgänge mit Dauer 0. Sie werden optisch anders dargestellt als normale Vorgänge, z.B. als Raute.

### 3 Auswertung des Netzplans

Nachdem wir einen Netzplan erstellt haben, können wir ihn auswerten und erhalten zwei wesentliche Informationen:

1. die minimale *Gesamtdauer*, die zur Abarbeitung aller Vorgänge erforderlich ist
2. für jeden einzelnen Vorgang den frühesten und spätesten Startzeitpunkt innerhalb dieses zeitlichen Gesamtrahmens.

Ermittelt werden diese Daten durch eine sogenannte Vorwärts- und Rückwärtsrechnung. Für diese Algorithmen benötigen wir zwei Begriffe:

- Ein **Startvorgang** ist ein Vorgang, der keine Vorgänger hat.
- Ein **Zielvorgang** ist ein Vorgang, der keine Nachfolger hat.

Aus technischen Gründen nehmen wir an, daß genau ein Startvorgang und genau ein Zielvorgang vorhanden sind<sup>3</sup>. Sofern zunächst kein eindeutiger Startvorgang vorhanden ist, führt man einen künstlichen Vorgang  $s$  mit Dauer  $D(s)=0$  ein und macht ihn zum Vorgänger der Vorgänge, die bisher noch keinen Vorgänger haben. Analog führt man bei Bedarf einen künstlichen Zielvorgang ein.

---

<sup>3</sup>Dies vereinfacht die folgenden Definitionen und Algorithmen; inhaltlich ist es keine Einschränkung. In Netzplan-Werkzeugen existiert diese Restriktion i.a. nicht.



**Die Vorwärtsrechnung.** Bei der Vorwärtsrechnung wird der Netzplan ausgehend vom Startvorgang “vorwärts”, also in der normalen Zeitrichtung durchlaufen. Zu jedem Vorgang  $v$ , den wir beim Durchlauf bearbeiten, berechnen wir die beiden folgenden Werte:

**FAZ(v)**, der früheste Anfangszeitpunkt, und

**FEZ(v)**, der früheste Endzeitpunkt,

an dem der Vorgang  $v$  beginnen bzw. enden kann. Die beiden Werte berechnen sich wie folgt:

$$FAZ(v) := 0$$

falls  $v$  der Startvorgang ist.

$$FAZ(v) := \max \{ FEZ(d) \mid d \text{ Vorgänger von } v \}$$

falls  $v$  Vorgänger hat. Diese Formel besagt, daß ein Vorgang frühestens starten kann, wenn der am spätesten endende Vorgänger beendet worden ist.

$$FEZ(v) := FAZ(v) + D(v)$$

für alle Vorgänge  $v$ .

FAZ(v) kann nur berechnet werden, wenn für alle Vorgänger von  $v$  schon der früheste Endzeitpunkt berechnet ist. Ausgehend vom Startvorgang kann FAZ(v) zunächst also nur für die direkten Nachfolger des Startvorgangs berechnet werden. Sobald für einen dieser Vorgänge FAZ(v) und danach FEZ(v) berechnet ist, kommen auch dessen Nachfolger in Betracht. Die Reihenfolge, in der die in Betracht kommenden Vorgänge bearbeitet werden, hat keinen Einfluß auf das Endergebnis.

Zuletzt berechnen wir den frühesten Endzeitpunkt des Zielvorgangs; dies ist die **minimale Gesamtdauer** des gesamten Netzes.

Als Beispiel betrachten wir den Netzplan in Bild 3. Dort ist oben links angegeben, wie u.a. die Größen  $D$ ,  $FAZ$  und  $FEZ$  innerhalb der Knoten notiert werden. Für die 5 Knoten des Netzplans ist zunächst nur die Dauer angegeben. Offensichtlich ist:

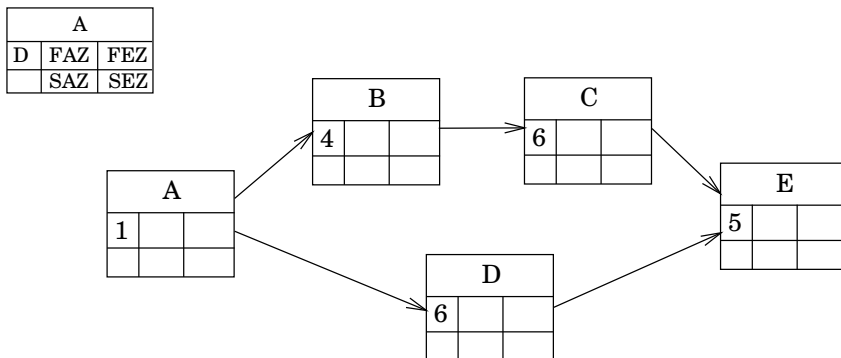


Abbildung 3: Netzplan

$$FAZ(A) = 0$$

$$FEZ(A) = FAZ(A) + D(A) = 1$$

$$FAZ(B) = \max \{ FEZ(u) \mid u \text{ Vorgänger von } B \}$$

$$= \max \{ FEZ(A) \} = \max \{ 1 \} = 1$$

$$FEZ(B) = FAZ(B) + D(B) = 1 + 4 = 5$$

Berechnen Sie nun als Übung anhand der vorstehenden Formeln für die Vorgänge C, D und E jeweils FAZ und FEZ.

**Die Rückwärtsrechnung.** Bei der Rückwärtsrechnung durchlaufen wir ausgehend vom Zielvorgang “rückwärts”, also entgegen der normalen Zeitrichtung, den Netzplan. Zu jedem Vorgang  $v$ , den wir beim Durchlauf bearbeiten, berechnen wir die beiden folgenden Werte:

**SAZ(v)**, der späteste Anfangszeitpunkt, und

**SEZ(v)**, der späteste Endzeitpunkt,

bis zu dem der Vorgang  $v$  beginnen bzw. enden muß, wenn die gegebene Projektgesamtdauer nicht verlängert werden soll. Jede Verzögerung eines Vorgangs über  $SEZ(v)$  hinaus führt zu einer entsprechenden Verlängerung der Projektgesamtdauer. Die beiden Werte berechnen sich wie folgt:

$$SEZ(v) := FEZ(v)$$

falls  $v$  der Zielvorgang ist.

$$SEZ(v) := \min \{ SAZ(d) \mid d \text{ Nachfolger von } v \}$$

falls  $v$  Nachfolger hat.  $v$  muß also spätestens enden, sobald der früheste Nachfolger beginnen muß.

$$SAZ(v) := SEZ(v) - D(v)$$

für alle Vorgänge  $v$ .

Als Übung berechnen Sie bitte anhand dieser Formeln  $SEZ(E)$ ,  $SAZ(E)$ ,  $SEZ(C)$ ,  $SAZ(C)$ ,  $SEZ(D)$ ,  $SAZ(D)$  usw. im Netzplan in Bild 3. Bild 4 zeigt das (hoffentlich entstandene) Endresultat.

**Pufferzeiten.** In Bild 4 fällt auf, daß bei den Vorgängen B und C der früheste und späteste Anfangszeitpunkt identisch sind, bei Vorgang D hingegen tritt eine Differenz von 4 Einheiten auf. B und C dauern zusammen 10 Einheiten, D nur 6. D kann gleichzeitig mit B anfangen, aber auch bis zu 4 Zeiteinheiten später; diese Zeiteinheiten sind die Pufferzeit von D. Allgemein definieren wir die (Gesamt-) **Pufferzeit** eines Vorgangs  $v$  als

$$GP(v) := SAZ(v) - FAZ(v) \quad ^4$$

Ein Vorgang  $v$  kann  $GP(v)$  Zeiteinheiten später als  $FAZ(v)$  anfangen, aber auch  $GP(v)$  Zeiteinheiten länger als geplant dauern, ohne daß deswegen das *gesamte Projekt* länger dauern müßte.

Ein Vorgang  $v$  heißt **kritisch**, sofern  $GP(v) = 0$ . Im Netzplan in Bild 4 sind A, B, C und E kritisch. Wenn ein kritischer Vorgang nicht sofort nach Abschluß seiner Vorgänger begonnen wird oder länger als geplant dauert, verlängert sich die Gesamtdauer entsprechend.

Als **kritischen Pfad** (bzw. kritisches Teilnetz) bezeichnet man die Menge aller kritischen Vorgänge. Diese Vorgänge bilden immer ein

---

<sup>4</sup>In der Definition von  $GP(v)$  könnten wir auch die Endzeitpunkte heranziehen. Zeigen Sie als Übung: für beliebiges  $v$  gilt:  $SAZ(v) - FAZ(v) = SEZ(v) - FEZ(v)$

zusammenhängendes Teilnetz des Netzplans, meist einen Pfad. Ein Teilnetz, das kein Pfad ist, erhalten wir z.B. in Bild 4, wenn wir die Dauer von C auf 2 reduzieren; dann wird auch Vorgang D kritisch.

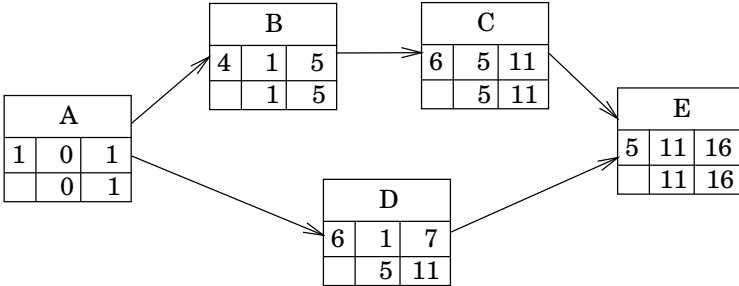


Abbildung 4: Netzplan

Man könnte nun der Meinung sein, daß *jeder* Vorgang um die oben berechnete Pufferzeit verlängert werden kann. Dies trifft leider nicht zu. Betrachten wir hierzu unser obiges Beispiel in einer Variante, bei der der Vorgang D durch zwei hintereinanderliegende Vorgänge D1 und D2 mit jeweils der halben Zeitdauer von D ersetzt worden ist.

Bild 5 zeigt das Ergebnis einer erneuten Vorwärts- und Rückwärtsrechnung. Man beachte, daß hier sowohl D1 als auch D2 eine Pufferzeit von 4 Einheiten haben - genausoviel wie D, die Pufferzeit ist nicht etwa anteilig reduziert worden. In der Tat könnte z.B. D1 4 Einheiten länger dauern, ohne die Gesamtdauer zu verändern; allerdings ist anschließend D2 kritisch geworden und hat keine Pufferzeit mehr. In einer "Reihenschaltung" von Vorgängen, wie sie bei D1 und D2 vorliegt, haben immer *alle* Vorgänge die gleiche Gesamtpufferzeit; sie kann aber nur von *einem* Vorgang ganz ausgenutzt werden bzw. von mehreren nur teilweise.

Diese Abhängigkeiten zwischen den Pufferzeiten sind natürlich unschön und können zu Fehlinterpretationen führen. Besser wäre es, unabhängig ausnutzbare Pufferzeiten zu kennen. Hierzu sind verschiedene Ansätze denkbar. Man kann z.B. die Gesamtpufferzeiten von Hand auf die Vorgänge aufteilen. Bequemer sind natürlich automa-

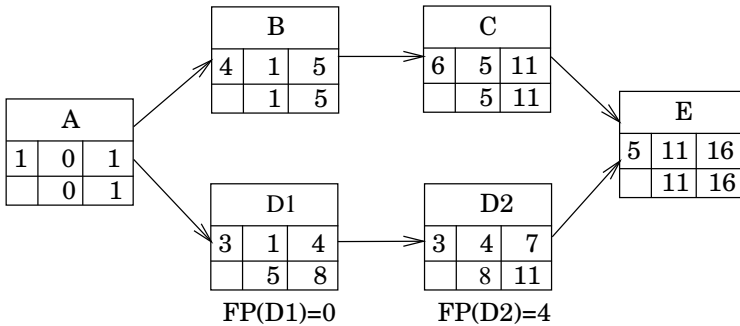


Abbildung 5: Pufferzeiten in einer “Reihenschaltung”

tisch durchführbare Verfahren.

Wir stellen hier nur ein automatisches Verfahren vor. Bei diesem Verfahren lassen wir uns von der Strategie leiten, möglichst vorsichtig zu sein und möglichst viel Zeitreserven zum Projektende hin vorzuhalten. Jeder Vorgang  $v$  soll daher bei  $FAZ(v)$  beginnen können. Die **freie Pufferzeit  $FP(v)$**  eines Vorgangs  $v$  definieren wir dann als die Zeit, die  $v$  länger dauern darf, ohne daß *irgendein anderer Vorgang* deshalb später anfangen muß. Formal ausgedrückt ist

$$FP(v) := \min \{ FAZ(u) \mid u \text{ Nachfolger von } v \} - FEZ(v)$$

Bei unserem Netzplan in Bild 5 ist  $FP(D1) = 0$  und  $FP(D2) = 4$ .

In einer Reihenschaltung wird immer dem *letzten* Vorgang die Gesamtpufferzeit als freie Pufferzeit zugewiesen; alle davorliegenden Vorgänge haben eine freie Pufferzeit von 0.

## 4 Alternative Reihenfolgebeziehungen

Wir haben in allen Beispielen bisher nur eine Art von Reihenfolgebeziehungen zwischen Vorgängen benutzt, und zwar **Ende-Anfang-Beziehungen**: Der Nachfolger kann erst beginnen, wenn der Vorgänger geendet hat. Tatsächlich kann man noch 3 weitere Arten von Reihenfolgebeziehungen unterscheiden.

Um sich die Bedeutung dieser Beziehungen klarzumachen, ist es hilfreich, sich jeden Vorgang als ein Zeitintervall vorzustellen, das bei einem Anfangszeitpunkt beginnt und bei einem End-Zeitpunkt endet. Eine Ende-Anfang-Beziehung zwischen den Vorgängen v1 und v2 besagt nun, daß das *End*-Ereignis des Vorgängers v1 *vor* dem *Anfangs*-Ereignis des Nachfolgers v2 liegen muß. Im rechten Teil von Bild 6 sind die Vorgänge v1 und v2 jeweils durch einen Zeitbalken dargestellt; der gestrichelte Pfeil stellt dar, für welche Ereignisse eine Reihenfolge explizit vorgegeben wird.

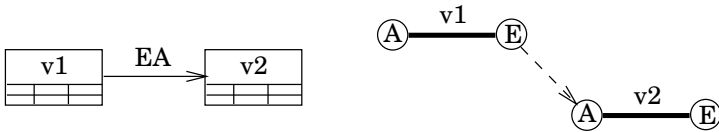


Abbildung 6: Ende-Anfang-Beziehung zwischen v1 und v2

Die anderen möglichen Reihenfolgebeziehungen legen nun für andere Paare von Ereignissen die Reihenfolge fest.

**Anfang-Anfang-Beziehungen:** Ist v1 Vorgänger von v2, dann kann v2 erst anfangen, wenn v1 angefangen hat (s. Bild 7).

Beispielsweise läuft die Erfassung und Auswertung von Aufwänden parallel zu einem Projekt, sie kann aber erst beginnen, wenn das Projekt selbst begonnen hat.

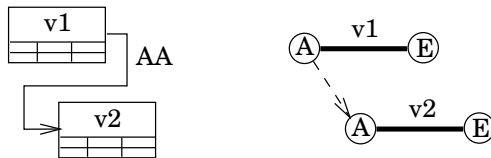


Abbildung 7: Anfang-Anfang-Beziehung zwischen v1 und v2

**Ende-Ende-Beziehungen:** Ist v1 Vorgänger von v2, dann kann v2 kann erst enden, wenn v1 geendet hat (s. Bild 8).

Beispielsweise können die Qualitätskontrollen erst enden, wenn auch die Systementwicklung beendet worden ist.

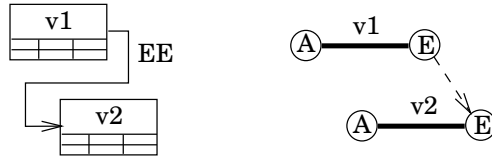


Abbildung 8: Ende-Ende-Beziehung zwischen v1 und v2

**Anfang-Ende-Beziehungen:** Ist v1 Vorgänger von v2, dann kann v2 erst enden, wenn v1 angefangen hat (s. Bild 9).

Beispielsweise kann der (Reserve-) Betrieb eines abzulösenden Systems erst beendet werden, wenn der Betrieb des neuen Systems begonnen hat.

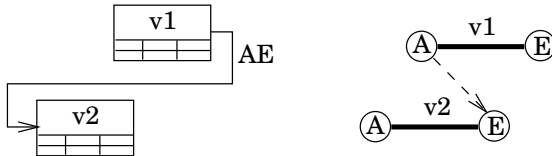


Abbildung 9: Anfang-Ende-Beziehung zwischen v1 und v2

In den Netzplänen wird die Art der Reihenfolgebeziehung durch die Kürzel EA, AA, EE bzw. AE als Beschriftung des Nachfolger-Pfeils angezeigt.

Wie aus den vorstehenden Diagrammen ersichtlich ist, können die beiden involvierten Vorgänge bei einer AA-, EE- oder AE-Beziehung teilweise *parallel* liegen; über die Reihenfolge der beiden übrigen Ereignisse (z.B. die beiden End-Ereignisse bei einer AA-Beziehung) wird keine Aussage gemacht. Für EA-Beziehungen trifft dies nicht zu, denn

hier wird indirekt eine lineare Reihenfolge aller 4 Ereignisse festgelegt.

**Abstände von Beziehungen.** Bisher legte eine Reihenfolgebeziehung nur fest, daß ein Vorgänger-Ereignis e1 vor einem Nachfolger-Ereignis e2 liegen muß. Interpretiert man die Ereignisse als Zeitpunkte, so gilt für diese folgende Ungleichung:

$$e1 \leq e2$$

Zwischen e1 und e2 kann demnach ein beliebig großer zeitlicher Abstand liegen. Oft will man jedoch einen Mindestabstand im Sinne einer (Mindest-) **Wartezeit** haben; als Beispiel nehmen wir an, daß die Zeiterfassung frühestens 5 Tage nach Beginn des Projekts beginnen soll. Mit e1 = Projektbeginn und e2 = Beginn Zeiterfassung würde folgende Ungleichung gelten:

$$e1 + 5 \leq e2$$

Bild 10 zeigt diesen Effekt graphisch.

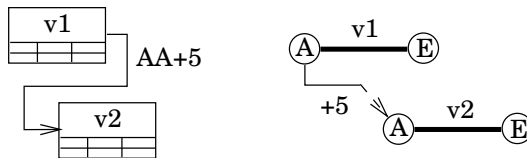


Abbildung 10: Anfang-Anfang-Beziehung zwischen v1 und v2 mit Wartezeit

Eine weitere Verallgemeinerung besteht darin, auch “negative Wartezeiten” zuzulassen, also eine Ungleichung

$$e1 + d \leq e2$$

für beliebige, also auch negative Werte für d zuzulassen. Bild 11 zeigt die Wirkung für d = -4 graphisch.

Alle bisherigen Abstände waren Mindestabstände. Eine zusätzliche Erweiterung könnte darin bestehen, auch Maximalabstände einzuführen, also letztlich Ungleichungen der Form



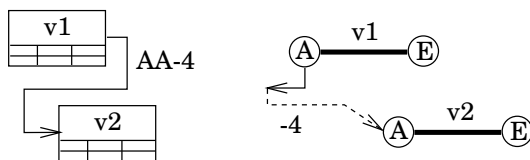


Abbildung 11: Anfang-Anfang-Beziehung zwischen v1 und v2 mit “negativer Wartezeit”

$$e1 + \min \leq e2 \leq e1 + \max$$

Allerdings stellt sich die Frage, ob derart komplexe Netzpläne noch praxisgerecht sind. Außerdem können nun Inkonsistenzen auftreten: zwei Ereignisse können durch zwei verschiedene Pfade verbunden sein, wobei der erste eine Gesamtzeit von  $< 10$  und der zweite eine von  $> 20$  hat.

## 5 Strukturierung von größeren Netzplänen

Bei größeren Projekten können Netzpläne viele hundert Vorgänge enthalten; diese sind auf einem Blatt Papier nicht mehr handhabbar. Zur Lösung dieses Problems sind mehrere Techniken anwendbar:

**Teildiagramme:** Hier wird aus dem Gesamtplan, der im Prinzip nach wie vor erforderlich ist, die Teilmenge der Daten herausgefiltert, die z.B. für eine Arbeitsgruppe oder ein Teilprojekt relevant sind. Irrelevante Teile müssen ggf. in geeigneter kondensierter Form, z.B. als Wartezeiten, dargestellt werden.

**Verfeinerungshierarchien:** Hier werden einzelne Vorgänge durch jeweils einen eigenen Netzplan verfeinert.

Verfeinerungshierarchien sind sehr naheliegend und erlauben es, zunächst nur eine Grobplanung vorzunehmen und danach eine Feinplanung. Die Feinplanung muß nicht gleich komplett ausgearbeitet

werden, sondern kann zunächst auf die kurzfristig anstehenden Teilaufgaben beschränkt bleiben.

Die Dauer eines Vorgangs, der eine Verfeinerung hat, kann automatisch als die Mindestdurchlaufzeit des verfeinernden Netzplans festgesetzt werden. Alternativ kann sie direkt geschätzt werden.

## 6 Abgeleitete Dokumente

Aus einem Netzplan bzw. dem entsprechenden Balkenplan können nach dessen Auswertung zusätzliche Dokumente abgeleitet werden, u.a.

- eine Vorgangsliste
- eine Terminliste, z.B. nach FAZ sortiert
- eine Rückmeldeliste

usw. Derartige Listen können ggf. selektiv nach Teilprojekten oder zuständigen Abteilungen gestaltet sein.

In den o.g. Listen müssen normalerweise konkrete Kalenderdaten verwendet werden. Netzpläne hingegen haben zunächst nur eine Zeit zum Projektbeginn. Diese relativen Zeitangaben müssen unter Berücksichtigung von Sonn- und Feiertagen und Ferien in konkrete Arbeitstage bzw. letztlich Datumsangaben umgesetzt werden.

## 7 Einsatzplanung und Kapazitätsausgleich

Netzpläne dienen, wie schon erwähnt, dazu, Projekte zu planen und vorab ihre Durchführbarkeit zu untersuchen. Bei unseren bisherigen Betrachtungen diente ein Netzplan primär dazu, die benötigte Gesamtzeit zur Durchführung eines Projekts zu berechnen. Diese Größe ist zwar wichtig, um zu entscheiden, ob ein Projekt in einem vorgegebenen Zeitrahmen überhaupt durchgeführt werden kann, ist aber natürlich nicht die einzige interessierende Größe.

Eine weitere wesentliche Frage ist, ob die Institution, die ein Projekt durchführen will, im vorgegebenen Zeitraum überhaupt über aus-

reichende Ressourcen verfügt. Die wichtigste Ressource bei Software-Entwicklungsprojekten ist normalerweise die Arbeitszeit geeignet qualifizierter Mitarbeiter. Wie schon im Kontext des Phasenmodells diskutiert können die erforderlichen Qualifikationen je nach Phase recht verschieden sein, es muß also der zeitgerechte Einsatz unterschiedlicher Spezialisten geplant werden. Netzpläne bilden hierfür eine Basis.

Das Vorstehende gilt analog für beliebige andere Ressourcen; wir können uns daher i.f. weiter auf die Arbeitszeit von Mitarbeitern beziehen.

Bei Vorgängen, die Pufferzeiten haben, muß ferner entschieden werden, wann sie nun definitiv beginnen sollen, also wie sie im durch Pufferzeit gegebenen Freiraum angeordnet werden.

## 7.1 Verfügbare Ressourcen

Für die Ressourcenplanung muß bekannt sein, welche Ressourcen verfügbar sind und welche benötigt werden.

Die verfügbaren Personalressourcen ergeben sich in erster Näherung aus den vorhandenen Mitarbeitern und der bei den einzelnen Personen an konkreten Tagen netto verfügbaren Arbeitszeit, wobei Urlaubs- und Feiertage und ähnliche Einflußfaktoren zu berücksichtigen sind. In verteilten Projekten oder Firmen muß ferner die örtliche Verfügbarkeit der Mitarbeiter und ggf. die organisatorische Zuordnung berücksichtigt werden.

Mitarbeiter sind außerdem keine Roboter, die man in beliebigen Formationen zusammenstellen kann und die dann brav zusammenarbeiten, sondern soziale Wesen mit eigenen Erfahrungen, Vorlieben und Abneigungen, Stärken und Schwächen. Wenn also 3 Java-Programmierer verfügbar sind, heißt das noch lange nicht, daß diese Personen beliebig austauschbar sind.

Bei jedem Wechsel des Arbeitsgegenstands muß sich ferner ein Mitarbeiter immer erst in die neue Aufgabenstellung und in den aktuellen Stand des Projekts einarbeiten, was viel Zeit kostet. In dieser Zeit ist der Mitarbeiter noch unproduktiv, weil er keine Projektergebnisse erstellt; im Gegenteil kostet er seine Kollegen, die ihn einweisen müssen,

zusätzlich Zeit. Ein zu häufiger Wechsel des Arbeitsgegenstands und hektisches Umdirigieren von Mitarbeitern ist also nicht sinnvoll.

## 7.2 Ressourcenbedarf

Die benötigten Personalressourcen für ein gesamtes Projekt können anhand der Aufgabenzerlegung, die dem Netzplan zugrunde liegt, zusammengestellt werden. Für jeden nicht weiter zerlegten Vorgang muß der Arbeitsaufwand und die erforderliche Qualifikation geschätzt werden.

Mit den üblichen Schätzverfahren erhält man zunächst nur den Aufwand in Personentagen (oder in einer anderen Einheit). Die Programmierung eines Moduls könnte bspw. 20 Personentage benötigen. Erbracht werden kann diese Arbeitsleistung auf ganz unterschiedliche Art:

- durch einen Mitarbeiter, der 20 Arbeitstage lang ununterbrochen an der Aufgabe arbeitet
- durch zwei parallel arbeitende Mitarbeiter an 10 Arbeitstagen
- durch zwei Mitarbeiter, von denen einer 15 Arbeitstage ganztags und der andere 10 Arbeitstage halbtags an der Aufgabe arbeitet

usw. Wenn Dauer und Aufwand für einen Vorgang gegeben sind, berechnet sich die durchschnittliche Anzahl der benötigten Mitarbeiter als

$$\frac{\text{Aufwand (in Personentagen)}}{\text{Dauer (Tagen)}}$$

Bei 30 Personentagen Aufwand und 5 Tagen Dauer müßten also durchschnittlich 6 Mitarbeiter an der Aufgabe arbeiten.

Für die Einsatzplanung nehmen wir nun zunächst vereinfachend an, daß jeder Vorgang  $v$  zum Zeitpunkt  $FAZ(v)$  beginnt und zum Zeitpunkt  $FEZ(v)$  endet und während dieser Zeit gleichviele Mitarbeiter daran arbeiten. Betrachten wir hierzu ein einfaches Beispiel mit 4 Vorgängen:

Vorgang	FAZ	D	FEZ	Aufwand
A	0	3	3	3
B	2	3	5	9
C	5	3	8	3
D	2	4	6	4

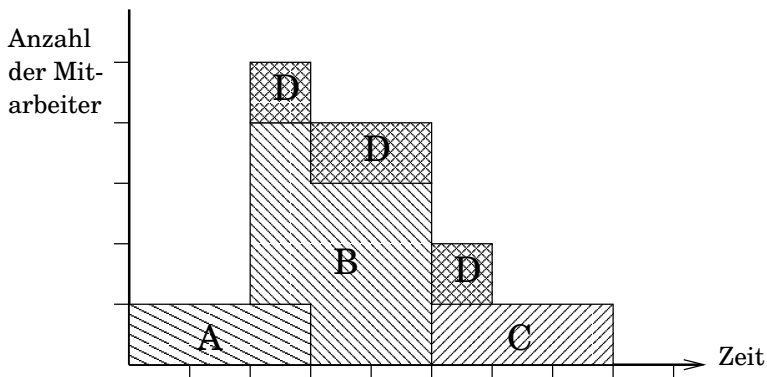


Abbildung 12: Aufwandsverteilung

Bild 12 zeigt die resultierende Zahl benötigter Mitarbeiter über die Zeit hinweg. Diese Zahl schwankt erheblich. Sind ausreichend Mitarbeiter vorhanden, wäre ein Teil oft unbeschäftigt. Sind weniger als die Maximalzahl Mitarbeiter vorhanden, ist die Einsatzplanung falsch und muß angepaßt werden.

### 7.3 Kapazitätsausgleich

Vorgehensweisen beim Kapazitätsausgleich sind:

- Verschiebung innerhalb der Pufferzeiten: wenn man im Beispiel in Bild 12 Vorgang D um eine Zeiteinheit nach hinten schiebt, reduziert sich die Maximalzahl der Mitarbeiter von 5 auf 4.
- Stauchung: Bei einer **Stauchung** eines Vorgangs wird dessen Dauer verringert, indem die Zahl der Mitarbeiter erhöht wird. In un-

serem Beispiel könnte Vorgang D auch in 2 Zeiteinheiten durch 2 Mitarbeiter bearbeitet werden.

- **Streckung:** Bei einer **Streckung** eines Vorgangs wird umgekehrt die Dauer erhöht, wodurch die Zahl der Mitarbeiter durchgehend oder streckenweise reduziert werden kann. Nehmen wir an, in unserem Beispiel wird Vorgang B durch 3 Mitarbeiter parallel bearbeitet. Wenn wir stattdessen einen Mitarbeiter einen Tag später anfangen lassen, dauert B 4 Zeiteinheiten, und es ergibt sich die in Bild 13 gezeigte Verteilung<sup>5</sup>. Alternativ könnten wir einen Mitarbeiter 4, einen 3 und einen 2 Zeiteinheiten lang für B einsetzen.

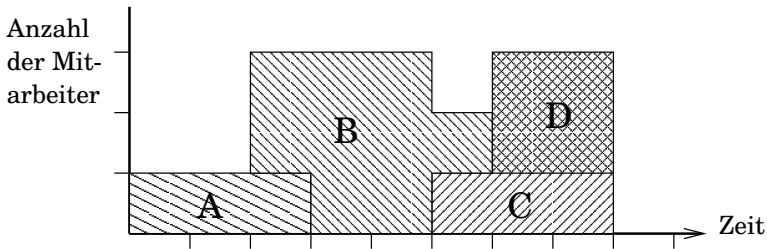


Abbildung 13: ausgeglichene Aufwandsverteilung

- **Aufteilung und Unterbrechung:** Hier wird eine Aufgabe mit Unterbrechungen bearbeitet. Dies ist nicht unproblematisch, da neue Einarbeitungsaufwände entstehen können und die Aufteilung dann nicht aufkommensneutral ist.
- **Verlängerung der Projektdauer:** Wenn die vorgenannten Verfahren nicht zum Erfolg führen, kann die Projektdauer verlängert werden.

Alle vorstehenden Verfahren gehen davon aus, daß die verfügbaren Ressourcen unverändert bleiben. Man kann natürlich auch auf dieser Seite ansetzen und durch Überstunden, Neueinstellung, Fremdvergabe von Teilaufgaben usw. die verfügbare Kapazität erhöhen.

<sup>5</sup>Hierbei ist unterstellt, daß C beginnen kann, bevor B komplett abgeschlossen ist.

Für die Anordnung der Vorgänge werden trotz allem oft Alternativen verbleiben. Bei der Wahl zwischen den Alternativen sind folgende **Optimierungskriterien** abwendbar:

- für besonders unsicher schätzbare Vorgänge möglichst hohe Pufferzeiten freilassen (d.h. deren Vorgänger nicht strecken und möglichst früh anordnen)
- Wenn man das Risiko für Verzögerungen des Gesamtprojekts minimieren möchte (weil eine saftige Konventionalstrafe droht), wird man alle Vorgänge möglichst früh anordnen. Nachteil hiervon ist bei langlaufenden Projekten, daß sehr früh viel Kapital eingesetzt wird, das finanziert werden muß, sofern der Kunde erst bei Abnahme zahlt.
- Wenn man umgekehrt den Kapitaleinsatz minimieren möchte, wird man Vorgänge möglichst spät anordnen.

## Literatur

[PM] Kelter, U.: Lehrmodul Projektmanagement“; 2001

## Glossar

**Balkenplan:** Darstellungsform eines Netzplans; Vorgänge werden als waagerechte Balken dargestellt, Reihenfolgeabhängigkeiten als Pfeile zwischen den Balken

**Kapazitätsausgleich:** Bildung eines Terminplans zu einem Netzplan, bei dem Spitzenwerte in den Ressourcenanforderungen vermieden werden, indem Voränge innerhalb von Pufferzeiten geeignet angeordnet und ggf. gestreckt oder gestaucht werden

**kritischer Pfad:** Menge der Vorgänge, deren Pufferzeit Null ist

**Meilenstein:** definierter Punkte zu haben, an dem bestimmte Projektziele erreicht werden (sollen) und der Fortschritt des Projekts überprüft werden kann

**Netzplan:** Strukturierung eines Projekts in einzelne Vorgänge und Reihenfolgeabhängigkeiten zwischen diesen; kann als Graph (als Vorgangs-

knotennetz oder Vorgangspfeilnetz) oder Balkenplan dargestellt werden

**Pufferzeit** (eines Vorgangs): Zeit, um den ein Vorgang zeitlich gestreckt oder später als am frühesten Anfangszeitpunkt gestartet werden kann, ohne daß die Gesamtdauer des Projekts dadurch verlängert wird

**Pufferzeit, freie** (eines Vorgangs): Zeit, die ein Vorgang länger dauern oder verspätet anfangen darf, ohne daß deswegen irgendein anderer Vorgang später anfangen muß

**Rückwärtsrechnung:** Verfahren, das für einen Netzplan die spätesten Anfangs- und Endzeitpunkte der Vorgänge sowie den kritischen Pfad berechnet; der Netzplan wird hierzu zeitlich "rückwärts" durchlaufen

**Vorgangsknotennetz:** Darstellungsform eines Netzplans, bei der jeder Vorgang durch einen Knoten und jede Reihenfolgeabhängigkeit durch eine Kanten dargestellt wird

**Vorgangspfeilnetz:** Darstellungsform eines Netzplans, bei der jeder Vorgang durch einen Pfeil dargestellt wird; Knoten stellen Ereignisse (oder "Meilensteine") dar

**Vorwärtsrechnung:** Verfahren, das für einen Netzplan die frühesten Anfangs- und Endzeitpunkte der Vorgänge berechnet; der Netzplan wird hierzu zeitlich "vorwärts" durchlaufen



# Index

- Balkenplan, 3, 23
- Einsatzplanung, 18
- FAZ(v), 9
- FEZ(v), 9
- FP(v), 13
- früherster Anfangszeitpunkt, 9
- früherster Endzeitpunkt, 9
- GP(v), 11
- Kapazitätsausgleich, 18, 21, 23
  - Optimierungskriterien, 23
- kritischer Pfad, 11, 23
- kritischer Vorgang, 11
- Meilenstein, 7, 23
- Nachfolger, 6
- Netzplan, 3, 23
  - Auswertung, 8
  - Bildung von Teilaufgaben, 4
  - Hierarchie, 17
  - minimale Gesamtdauer, 9
  - Startvorgang, 8
  - Strukturierung, 17
  - Teildiagramm, 17
  - Zielvorgang, 8
- Pufferzeit, 11, 24
  - freie, 13, 24
  - Gesamtpufferzeit, 11
- Qualifikation, 5
- Ressourcenplanung, 18
- Rolle, 5
- Rückmeldeliste, 18
- Rückwärtsrechnung, 10, 24
- SAZ(v), 10
- SEZ(v), 10
- Startvorgang, 8
- Stauchung, 21
- Streckung, 22
- Teilaufgabe, 4
- Teildiagramm, 17
- Terminliste, 18
- Vorgänger, 6
- Vorgang, 4
  - kritischer, 11
  - Reihenfolge
    - AA-Beziehung, 14
    - Abstände, 16
    - AE-Beziehung, 15
    - EA-Beziehung, 13
    - EE-Beziehung, 15
  - Verfeinerungshierarchie, 4
  - Wartezeiten, 16
- Vorgangsknotennetz, 6, 24
- Vorgangsliste, 18
- Vorgangspfeilnetz, 6, 24
- Vorwärtsrechnung, 9, 24
- Wartezeit, 16
- Zielvorgang, 8