

CCPM ist kein ToC, es ist ToC ++

Vorwort

Mittlerweile habe ich das Buch "Die kritische Kette" gelesen und finde meine eigene These so nicht mehr haltbar. Interessanterweise hat niemand anders dieser These widersprochen. **Ich hätte mir gerne Widerspruch zu dieser These gewünscht**, daran könnte ich festmachen wie gelungen die These ist. Da ich es wichtig finde auch die Entwicklung aufzuzeigen, werde ich daher nicht einfach das inzwischen falsch erkannte entfernen, sondern hier entsprechend farblich kommentieren.

Ich (Wolfram) würde den Faden gerne aufnehmen und weiter texten. Im letzten Thread über [High-Speed-Projektmanagement](#) haben wir ausschließlich in den Kommentaren diskutiert, das führt dazu, dass das Wissen nie konsolidiert wurde, was schade ist. Daher würde ich einfach mal in "grün" weitere Ergänzungen machen. Und ja ich hab die These für gut befunden, daher auch die "Antwort" erst jetzt. Denn mir ging es lange genau so - was hat CCPM mit TOC zu tun? nicht wirklich viel. Je länger man es anwendet dann aber doch. Und die Bezeichnung im Titel **TOC++** ist einfach genial - da wollte ich nicht widersprechen 😊

Mit beiden Büchern habe ich so meine Probleme. "[Goldratt und die Theory of Constraints](#)" von Uwe Techt ist ungenau und unvollständig (würde ich nicht unterschreiben). Z.B. Differenz zwischen kritischer Kette und kritischem Pfad (ist aber drin in 4. Auflage auf S. 137). "Die kritische Kette" von E. Goldratt ist eine Art von Roman (gewöhnungsbedürftig). Es ist schwierig einen Aspekt nachzuschlagen. In welchem Kapitel wird die "Differenz zwischen kritischer Kette und kritischem Pfad" angesprochen? (aus Verzeichnis der Aussagen - Kapitel 22, Seite 224 aktuelle Auflage - ist unter der Idee: Arbeitspakete sind nicht nur über funktionale Abhängigkeiten verbunden sondern auch noch über die zu verwendende Ressource). Wichtig ist beide Grundlagenbücher von Goldratt zu lesen (die Kritische Kette ist eigentlich das zweite Buch) "das Ziel" das erste. Das eine macht eigentlich ohne das andere nur teilweise Sinn. Also wer sich von Romanen nicht abschrecken lässt - dann beide lesen.

Bei aller Kritik, ToC/CCPM sind ein wertvolle Ansätze. Dennoch sollte niemand außer acht lassen, dass es wie jedes andere Konzept seinen Gültigkeitsbereich hat. Man kann es falsch anwenden, oder überfordern. Und natürlich hat ToC/CCPM auch Schwächen.

These

CCPM (Critical Chain Project Management) basiert nicht auf ToC (Theory of Constraints). Es sind zwei völlig unterschiedliche Konzepte.

Diese These ist überraschend da beide Konzepte vom selben Author E. Goldratt stammen und auch im selben Buch beschrieben werden. Es wird auch der Eindruck erweckt CCPM sei ein Teil von ToC.

Die These ist grundsätzlich o.k. - die Mechanismen wie sie im Buch "Kritische Kette" beschrieben werden, sind zuerst einmal unabhängig, ob man einen Engpass (Constraint) hat oder nicht. Daher o.k. - je weiter man in die Materie einsteigt, desto mehr Aspekte aus der TOC werden dann aber sichtbar. Ich versuche ein paar davon weiter unten auszuführen. Hier nur zwei Richtungen: die TOC umfasst auch grundlegende systemische Betrachtungsweisen und Methoden - oft genannt sind die "fünf Fokusschritte" und die "Konfliktwolken" - wenn man diese (TOC) anwendet kommt man auf solche Ideen wie die kritische Kette - also basiert CCPM auf den Denkprozessen der TOC. Die andere Richtung ist, dass sich CCPM seit dem Buch vor allem auch vom Single-Projektmanagement (Buch) hin zu Multiprojektmanagement hin entwickelt hat. Multiprojektmanagement "sieht" wiederum mehr aus wie Produktion und daher kommen hier die Ideen aus der TOC wieder voll zum Zuge.

Es ist wichtig zu verstehen warum CCPM funktioniert. Nur so kann das Potential voll ausgeschöpft werden. Falsche Regeln/Bilder sorgen für minderwertige Ergebnisse. Absolut!

Hintergrund

ToC operiert mit den Begriffen

- *Ressource - als das wertvolle Element, dass aus Rohstoffen etwas von Wert schafft*
- *Engpass - als das Element an dem die Aufträge gestaffelt und gesteuert werden. Wenn hier etwas schief geht wird Durchsatz verloren!*
- *Durchsatz - als übergeordnetes Ziel*
- *Puffer - die die allgegenwärtigen Störungen/Störungen absichern um Zuverlässigkeit zu erreichen*

CCPM (Single-Project-Management) operiert mit den Begriffen

- *Ressource - im positiven Sinne auch Menschen - denn jeder Mensch hat Ressourcen*
- *Engpass - im Sinne von "wenn hier etwas schief geht wird Durchsatz verloren!" (s.o.)*
- *Kritische Kette (unter Berücksichtigung der aktuellen Ressourcensituation im Unternehmen)*
- *Kritischer Pfad (nur als Abgrenzung zur kritischen Kette)*
- *Zulieferpuffer - Puffer die Zulieferketten so absichern, dass sie die Kritische Kette nie stören*
- *Projektpuffer - Puffer die den Endtermin oder den Integrationspunkt absichern*

CCPM Multiprojektmanagement operiert mit den Begriffen (tief aus der TOC)

- *Ressource*
- *Engpass*

- Durchsatz
- Ressourcenpuffer (die dritte Art von Puffer, die Streuungen im Engpass absichert)
- Staffelnung - also die terminierung der Projekte anhand der kritischen Ressourcen und sicher stellung, dass nie zu viel Work-In-Progress im System ist. Was der strategischen Priorisierung entspricht.
- Projektstatus über Fortschritt zu Pufferverbrauch = Projekttampel
- operative Priorisierung der Ressourcen anhand der Ampel. Das "roteste" Projekt bekommt alle Aufmerksamkeit und Ressourcen

Durchsatz

Ziel von ToC ist es den Durchsatz zu erhöhen, also letztlich den Gewinn (korrekter Deckungsbeitrag 1)

Bei einer Fertigung ist es recht offensichtlich, was Durchsatz ist: transformiertes (bearbeitetes, zusammengesetztes) Material

Bei einer (Software-) Entwicklung ist das nicht mehr offensichtlich. Was soll Durchsatz sein?

- Lines of Code, Anzahl Klassen, Anzahl Methoden
- Anzahl implementierter Story points/Use cases
- Verbrauchte Zeit
- Abgeschlossene Entwicklungen

Wie sich das in Gewinn umrechnet, hängt auch stark vom Kontext ab

- Auftragsentwicklung Festpreis macht mit wenig verbrauchter Zeit Gewinn
- Auftragsentwicklung Aufwand macht mit viel verbrauchter Zeit Gewinn
- Eigenentwicklung macht gar keinen Gewinn

Letztlich führt CCPM und ToC dazu, den Engpass optimal zu nutzen. Bei klassischen Produktionsprozessen werden einfach mehr Stücke erstellt und der Zusammenhang mit dem Durchsatz ist naheliegend. Bei Projekten ist er es nicht, siehe die schon vorgetragenen Argumente. CCPM hebt als besonderen Nutzen die schnellere Fertigstellung hervor und erlaubt damit einen möglichen früheren Kapitalrückfluss. **Dazu muss allerdings auch das Projekt selbst einen Nutzen erzeugen, wofür CCPM keinerlei Garantie schafft.**

In der Tat lässt sich der Durchsatzbegriff aus der TOC/Produktion nicht direkt auf Projekte umsetzen. Ich bin aktuell gerade in einer CCPM-Vollimplementierung in der wir dies trotz dem machen. Es handelt sich hier um den Themenblock "Throughput Accounting" aus der TOC. Hierbei dreht sich alles um den Quotient aus Deckungsbeitrag 1 zu Aufwand im Engpass (auch gerne als Oktanzahl bezeichnet). Wenn man die Projekte nach diesem Quotient priorisiert und auswählt erreicht man den optimalen Durchsatz (in Geld). Der Trick ist einfach, dass man entweder den Deckungsbeitrag rechnen kann (Achtung Personalkosten sind in der TOC keine vollvariablen Kosten, wenn es nicht gerade Freelancer sind). Bei Abogeschäft oder Produktentwicklung nimmt man meist den erwarteten Deckungsbeitrag über zwei Jahre. Damit kann man wieder alle Werkzeuge der TOC und des Throughput-Accounting anwenden.

Kritischer Pfad und kritische Kette

Der kritische Pfad ist wohl den meisten bekannt. Es ist der längste Pfad von Aktivitäten, die (logisch) aufeinander folgen. Der kritische Pfad macht eine praxisferne, implizite Annahme: die Aktivitäten konkurrieren NICHT um Ressourcen.

Die kritische Kette berücksichtigt die Verfügbarkeit von Ressourcen sowie die maximal sinnvolle Parallelisierbarkeit und ist damit mindestens so lange wie der kritische Pfad. Ein gutes Bild für einen Projektplan nach Critical Chain wäre - ein "green Field Plan" - also ein Plan, wie wenn das Projekt, das man plant, das einzige Projekt im Unternehmen wäre. ~~Der kritische Pfad kann als minimale Länge verstanden werden, wenn beliebig viele Ressourcen vorhanden sind.~~

Zulieferpuffer

Die Zulieferpuffer (!Pural) schützen ~~den Engpass~~ die kritische Kette. Sie liegen ~~VOR dem Engpass~~ jeder Einmündung in die kritische Kette und entspricht daher dem (einen) Puffer von ToC. Auch in ToC könnten viele Puffer ~~VOR dem Engpass~~ auftreten, wenn der Engpass viele Materialströme zusammenführt ((End-) Montage). Daher spricht man heute nicht mehr so von Engpass sondern mehr von Integrationspunkt. Der Integrationspunkt kristallisierte sich über viele Jahre als der kritische Punkt in Projekten (und in der Produktion) heraus. Hier passieren "Sachen" die nicht planbar sind aber auf jeden Fall Experten oder schnelle Entscheidungen benötigen um das Projekt nicht zu verzögern. Beides Experten und Entscheidungen sind oft in Unternehmen ein Engpass. Allerdings kein realer sondern ein virtuelle - weil man letzt endlich nicht weiß welcher Experte oder welcher Entscheider gebraucht wird.

Projektpuffer

Der Projektpuffer liegt ~~NACH~~ (das ist wohl oft der Fall - aber nicht zwingend) dem Engpass. Innerhalb von ToC gibt es dazu keine Entsprechung. Ich bin mir sicher, dass auch bei ToC es in gewissen Fällen Sinn machen könnte, einen Puffer nachzuschalten. Aber bislang ist mir kein Beispiel eingefallen.

Wenn man TOC mit Produktion gleichsetzt gibt es für den Projektpuffer (daher auch der Name Projektpuffer) keine Entsprechung. Die Produktion ist gekennzeichnet durch viele kleine Aufgaben mit hoher Vorhersagbarkeit und wenig Störungen. Das ist in Projekten definitionsgemäß nicht gegeben - diesen Störungen/Streuungen trägt der Projektpuffer rechnerisch um den Termin oder den Eintritt in den Integrationspunkt abzusichern.

Engpass und Puffer

In ToC wird vor dem Engpass ein **Material**puffer gelegt, der dafür sorgt, dass es meistens etwas für den Engpass zu tun gibt.

In CCPM wird ein **Zeit**puffer NACH abgeschlossenen Arbeitspaketen gelegt. Nach ToC müsste sich also NACH dem Puffer der Engpass befinden. Was soll das sein?

Also hier liegt der Fehler. Der Projektpuffer entspricht nicht dem ToC-Puffer VOR dem Engpass. Die Zulieferpuffer bilden den Puffer vor dem Engpass. Genau s.o.

In ToC begrenzt der Engpass den Durchfluss von Material (weil nicht schneller transformiert werden kann). Was fließt überhaupt gemäß CCPM? Und was ist da begrenzt?

Der Fehler ist wohl einen Fluss zu unterstellen. Vielmehr sind es wohl abhängige Aktivitäten, die nicht schneller abschließbar sind (genau). Wenn in CCPM mehrere Aktivitäten sich vor einen Engpass stauen bilden, warum sollte es in ToC nicht anders sein? Also nicht eine Maschine ist der Engpass sondern ein ganzes System von Maschinen?

Der Fluss in CCPM sind die Arbeitspakete. Aus Sicht einer Ressource sieht es so aus, als ob Arbeitspakete auf sie ein- und durchströmen. Daher gilt gerade hier auch auf jeden Fall den Work-In-Progress zu Begrenzung um Little's Law nicht zu tragen zu bekommen.

Ressource

In ToC ist eine Ressource etwas was pro Zeiteinheit eine bestimmte Transformation vornehmen kann. Begrenzende Ressourcen sind in der Regel Maschinen, die eine klare Spezifikation ihrer Leistungsfähigkeit haben,

In CCPM ist eine Ressource ein oder mehrere Entwickler. Entwickler transformieren Anforderungen in technische Lösungen, soweit gilt die Analogie. Nur hängt der Zeitbedarf von extrem vielen, auch zufälligen Parametern ab.

Bestimmte Schritte werden als nur von bestimmten Entwicklern machbar eingeschätzt, das erzeugt zwar Zwänge und Abhängigkeiten, nachvollziehbar ist dies jedoch nicht.

Insgesamt bin ich mit dem Begriff Ressource nicht zufrieden, auch nicht mit dem Begriff Aktivität. Beide implizieren Sachverhalte, die nicht gefordert sind. Z.B. Innerhalb eines Projekts sei es notwendig den Prototyp 100 Stunden in absoluter Ruhe trocknen zu lassen. Der Prototyp passt nur in den größten Raum des Unternehmens. Dieser Raum ist sicher keine "Aktivität", vielleicht eine Ressource, aber gewiss kein Entwickler. Dennoch wirkt er im Beispiel als (Teil des) Engpass. Und wenn sonstige Projekte nie solch große Prototypen umsetzen, dann wird in der Regel niemand den Raum als verfügbare Ressource wahrnehmen.

Ressource wird in heutigen CCPM Implementierung immer Synonym zu "Skill" verwendet. Also wirklich die Fähigkeit aus irgendwas etwas wertvolleres anderes zu machen. Das es sich meist um Menschen handelt und diese wie wild streuen und vielen Störungen unterliegen - oder man kann es auch positiv formulieren: hochgradig kreativ sind 😊 braucht man bei CCPM eben den Projektpuffer um diese ganze Unsicherheit abzufangen. Hier wird in dem Sinne nicht versucht mit Analogien zu Produktion zu arbeiten. Gerechnet und geplant wird im ganzen CCPM oft nur über die Zeit (nicht die Kapazität) - das ist gewöhnungsbedürftig aber am Schluss konsequent.

In neueren Implementierung wird das Konzept noch viel weiter getrieben - man plant die Ressourcen praktisch gar nicht mehr! Man wählt eine virtuelle Engpassressource und baut das Unternehmen drum herum, so dass es in anderen Teams/Skills nicht zu einem Engpass kommt. Funktioniert interessanterweise wunderbar - ist aber noch viel viel gewöhnungsbedürftiger für die Top-Manager als CCPM an sich schon - aber extrem leistungsfähig! Das wird kaum in Büchern erwähnt - höchstens in Seminaren.

Pfade / Chains

Auffällig ist, dass Projektpläne völlig unpassend dargestellt werden (das ist eine recht moderne verdichtete Darstellung - verschränkte Swin-Lane-Diagramme). Ich beziehe mich auf Pläne wie auf Seite 124 des Buches (4. Auflage) "Goldratt und die Theory of Constraints" von Uwe Techt. Falls ich den vergleichbaren Link wiederfinde, binde ich ihn hier ein. (Man kann auch einfach mit mir/Wolfram Kontakt aufnehmen - ich hab das Buch im 10er-Pack bei mir liegen und schicke es gerne als Dauerleihgabe zu)

Unpassend, weil alle Blöcke Ressourcen genannt werden, aber eigentlich Arbeitspakete oder Phasen sind (richtig es sind Arbeitspakete, die von Ressourcen bearbeitet werden - daher ist das de facto das gleich - aber unglückliche Wortwahl)

Unpassend, weil jedes Arbeitspaket in diese Teile zerfällt

- Abstimmung der Schnittstellen, hier müssten mehrere Entwickler zusammenkommen --> oft auch Auftragsklärung, Grobplanung, Planung genannt
- Implementierung, die weitgehend unabhängig vom Nachfolger erstellt werden kann --> oft auch Stream, Teilprojekte genannt, der Aspekt hier ist oft die Parallelität
- Integration --> inklusive Endabnahme und dann der Roll-Out

Unpassend, weil Entwicklung von Zusammenarbeit lebt, aber sicher nicht von sequentieller Übergabe. Die Diagramme implizieren aber sequenziellen Fluss ("Material").

CCPM im Buch "Goldratt und die Theory of Constraints" ist unglücklich wenn nicht sogar falsch dargestellt. Engpässe sind nicht zwingend Ressourcen. Der Engpass in CCPM ist die kritische Kette. Im Buch "Die kritische Kette" ist das inhaltlich besser dargestellt.

Ich bin zuerst auch von "unpassend" ausgegangen - weil es nicht intuitiv so ist. Normalerweise sieht man nur die Ressourcenkonflikte an den realen Engpässen. Wenn man die realen Engpässe aber nacheinander auflöst verändert sich das Bild und das Muster dahinter. Ich hab ca. 540 Projekte auf dem Buckel und kann mittlerweile bestätigen, dass Projekte alle!!!! den grundsätzlich gleichen Aufbau haben. Es gibt eine Phase (Auftragsklärung, Planung), dann wird parallelisiert (in Teilprojekten) die dann in mehreren Integrationspunkten münden. Neu sind Ideen aus dem agilen - z.B. kontinuierliche Integration - das ist aber ein ganz anderes Kapitel und ganz ganz neu - das wird gerade erst geschrieben.

Was ist anders bei CCPM

In CCPM werden folgende Aspekte angesprochen

- Schädliches Multitasking
- Versteckte Sicherheiten

Schädliches Multitasking

In einer Fertigungsumgebung würde man von Rüstzeiten und Durchlaufzeitverlängerung aufgrund von Little's Law sprechen, die durch Multitasking hervorgerufen werden (da sind sie eher erwünscht um kleine Losgrößen fahren zu können). Vergleichbares gibt es zwar auch in der Entwicklung, ist aber nicht der dominante Aspekt.

Das zugrunde liegende, allgemeinere, umfassendere Konzept heißt:

Vermeide Konflikte

genauer, vermeide Konflikte die der Entwickler nicht konstruktiv im Rahmen seiner Kerntätigkeit entscheiden und auflösen kann. Es gibt eine Unmenge an Konfliktfeldern, die zu Minderleistungen führen können.

- andere offene Aufgaben in andere Projekten/Arbeitspaketen
- unklare Vorgaben/Anforderungen
- schlechtes Arbeitsklima
- nicht zuhörendes Management oder Toll-dass-wir-darüber-gesprochen-haben-Management oder Das-habe-ich-gemacht-Management oder 10%-Streichen-geht-immer-Management
- störende Nebenaufgaben, Bürokratie
- schlechte/fehlende Werkzeuge

Das ist IMHO übrigens der Motor/Kernpunkt/Knackpunkt an CCPM. Das ist eigentlich einfach setzt aber die Selbstregelung in Gang - ohne dieses würde CCPM nicht funktionieren. Es geht grob verkürzt so:

- die Projektpuffer dienen nicht nur der Absicherung des Termins oder Integrationspunkt
- wenn man einen Projektpuffer hat kann man den Fortschritt (zeitlich auf der kritischen Kette) dem Pufferverbrauch gegenüberstellen
- ist der Fortschritt größer als der Pufferverbrauch - alles o.k, wenn nicht muss man was tun.
- zum operativen steuern der Ressourcen/Arbeitspakete wird hiervon nur der Quotient aus Fortschritt zu Pufferverbrauch verwendet
- im Konfliktfall (s.o.) wird das Arbeitspaket des Projektes vorgezogen, das den schlechtesten Fortschritt zu Pufferverbrauch aufweist
- **Achtung jetzt kommts: wenn die strategischen Priorisierung im Multiprojekt richtig durchgeführt wurde und nicht zu viel Work-In-Progress im Gesamt-Projektsystem ist — und — wenn immer genau das Arbeitspaket mit dem schlechtesten Fortschritt zu Pufferverbrauch bevorzugt wird — DANN kann man davon ausgehen, dass 95% der Termine der Projekte eingehalten werden.**
- In der CCPM Praxis dreht man das rum - man stellt die Termine so ein, dass 95% gehalten werden und dann kann man nach Fortschritt und Pufferverbrauch die ganzen Konflikte lösen - ohne Diskussionen (sehr erleichternd)
- Man kann das übrighends leicht sehen, wenn im Portfolio wenige als 10% der Projekte rot sind dann funktioniert das.
- Achtung - das sind alles Daumenwerte - man startet mit diesen und lernt in der Regel schnell, wie man sie für das Unternehmen richtig einstellen muss
- Das ist die Steuerung von CCPM - die Regelung des Ganzen - CCPM ist ein geregeltes System und daher so extrem stabil! Die ganzen Fehler die wir alle machen, werden einfach ausgeregelt.

Die Regelung hat aber noch weitere sekundäre Aspekte - von denen ich heute Abend (es ist jetzt schon spät) nur wenige schnell nenne:

- Die Projektmanager müssen nicht mehr den Ressourcen nachjagen - werden ja über die Ampeln verteilt ... dadurch haben sie mehr Zeit für echte Projektmanager-tätigkeiten und Qualität ... z.B. Führung, Auftragsklärung, erstellen von steuerbaren Projektplänen
- Um an Ressourcen zu kommen braucht man die Ampel - um an die Ampel zu kommen braucht man einen Projektplan, um einen Projektplan zu bekommen braucht man Arbeitspakete, Abhängigkeiten und Dauern, um das zu bekommen muss man die den Ressourcenverantwortlichen sprechen, wenn man mit den Ressourcenverantwortlichen sprechen will sollte man wissen was man will, wenn man nicht weiß was man will sollte man mit den Stakeholdern reden u.s.w. was wir immer wieder sehen ist — CCPM wirkt wie ein Katalysator der die ganzen guten Ideen aus dem Projektmanagement aktiviert - wie ein Turbo!
- Wenn man diese Ampeln hat kann man für jedes Arbeitspaket messen, wie viel Puffer es verbraucht hat. Wenn man die Arbeitspakete nach Team und Verbrauch aggregiert bekommt man das Team in dem der meiste Puffer verbraucht wird. Keine Angst - genau dieses Team ist nie Schuld - es muss es nur ausbaden. Die Probleme kommen immer von Upstream (also vorgelagerten Teams) — aber man weiß endlich wo man anfangen muss. Das ist wie Turbo-Kaizen - also nicht mit der Gieskanne sondern genau an der Stelle wo es was bringt = FOKUS TOC wird oft auch als Lean mit Fokus bezeichnet.

Versteckte Sicherheiten

Sicherheiten werden aus Sicht des Entwickler immer notwendig wenn kein Vertrauen gegeben ist. Nur wenn Offenheit, auch bei Fehlern nicht zu Strafen führt, kann dieses Vertrauen entstehen. Nur dann wird aus Entwicklern und Management ein Team wo jeder in seiner respektierten Rolle das Optimum einbringt.

Wird Vertrauen missbraucht, wird trotz offiziellem CCPM wieder der private Puffer draufgepackt.

genau! Am Schluss führt CCPM zu viel Vertrauen im System - dadurch werden viel viel viel versteckte Puffer mit der Zeit freigegeben - ehrlich!

Fazit

(das Beste Fazit was ich zu CCPM je gelesen habe - Hut ab!)

Der wesentliche Beitrag/Nutzen ist die Änderung/Verbesserung der **Firmenkultur** und der **Ethik**

Dann kann sich ein Team ausbilden, stark und leistungsfähig werden.

Die verwendete Projektsteuerung ist dann eigentlich nebensächlich.

Ja die Steuerung wird nebensächlich. Die Betonung liegt auf "wird" - sie ist existenziell aber um den Prozess in Gang zu bringen. Wenn man "begrenzt ist" muss man sich "fokussieren" und dann spielt die Reihenfolge, in der man Maßnahmen ergreift, die entscheidende Rolle. Daher haben die TOC'ler die Strategie und Taktikbäume erfunden in denen stehen die bewährten Reihenfolgen drin - aber auch das ist ein recht neues Kapitel der TOC ... und es werden noch viele folgen 😊

Weiterführende Informationen

Wikipedia Critical-Chain-Projektmanagement.

Wikipedia Theory_of_Constraints

Buch "Goldratt und die Theory of Constraints" von U.Techt

Buch "Die kritische Kette" von E. Goldratt

p.s.: es gibt auch Seminare - weil das meiste gar nicht in den Büchern steht - und leider nur ganz wenige Seminare