

# クリティカルチェーン (CCPM) プロジェクト管理 概要

当資料は、CCPM(クリティカルチェーン)の概要をご理解いただくプレゼンテーション・スライドです。  
弊社発売のCCPM+は、このCCPMの理論を実践するソフトウェアです。

MSI株式会社

# はじめに

---

- TOCのプロジェクト管理手法は、従来のPERTやクリティカルチェーンによるプロジェクトのスケジューリングや日程管理の領域をカバーしています。
- さらに従来の手法では対応できなかった不確実性、プロジェクト内の資源の競合、複数のプロジェクト間での資源の競合といった問題に対応しています。
- TOCプロジェクト管理手法は一般にプロジェクト管理と呼ばれている広い領域はカバーしていません。つまり、プロジェクト管理の組織運営、チーム作り、リーダーシップ、品質管理、コスト管理などは含まれていません。

# 目次

---

1. プロジェクト管理の基礎知識
2. TOCとプロジェクト管理
3. CCPMによるスケジューリング
4. 資源競合とクリティカルチェーン
5. バッファ管理の方法
6. 複数プロジェクト管理
7. まとめ

# 1. プロジェクト管理の基礎知識

---

- プロジェクト管理の問題点
- PERT

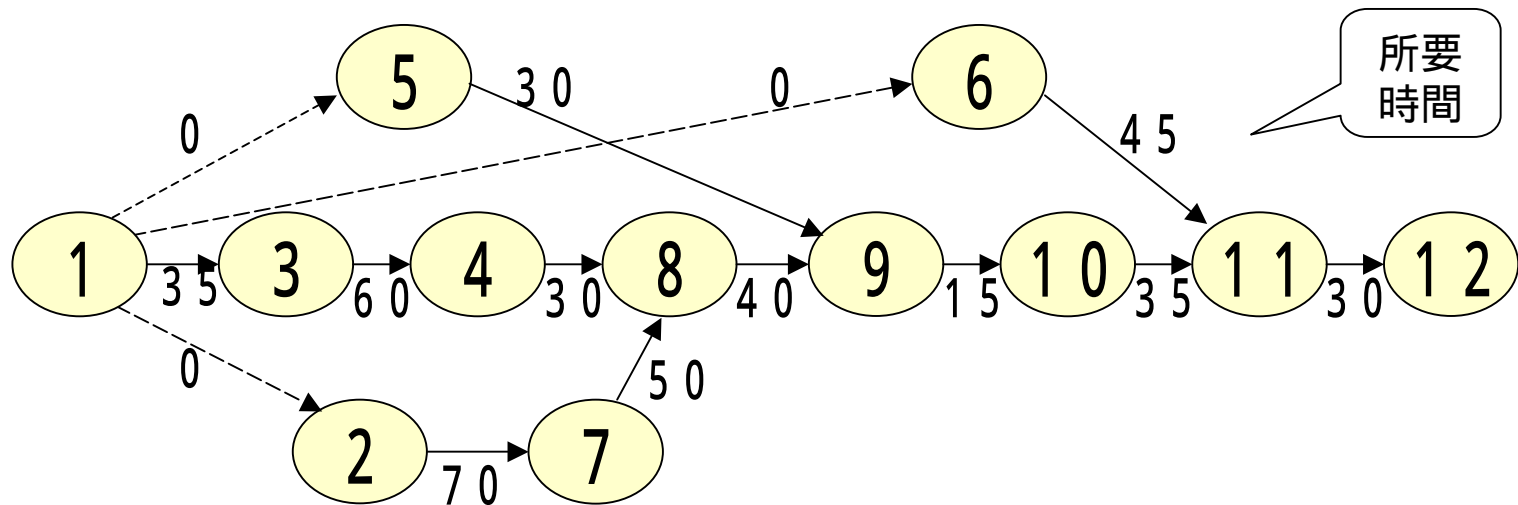
# プロジェクト管理の問題点

---

- ◆ **納期遅延**： 管理者に担当者の作業進捗が見えない。  
最後の10%問題。  
進捗把握ができず管理不能。
- ◆ **予算超過**： 納期超過分はコスト持ち出し。  
追加コスト(増員・設備・残業等)の発生。
- ◆ **品質/性能劣化**： 納期優先の場合に仕様・性能を削減。  
投資効果の無い結果となる。
- ◆ **設計変更多発**： 納期間際まで仕様変更が発生。

# PERT

- ◆ Program Evaluation and Review Technique
- ◆ PERTは1958年にコンピュータ処理用手法として開発。
- ◆ アローダイアグラムで表現される。
- ◆ 矢印(アクティビティ:実際の工程)と結合点(イベント)

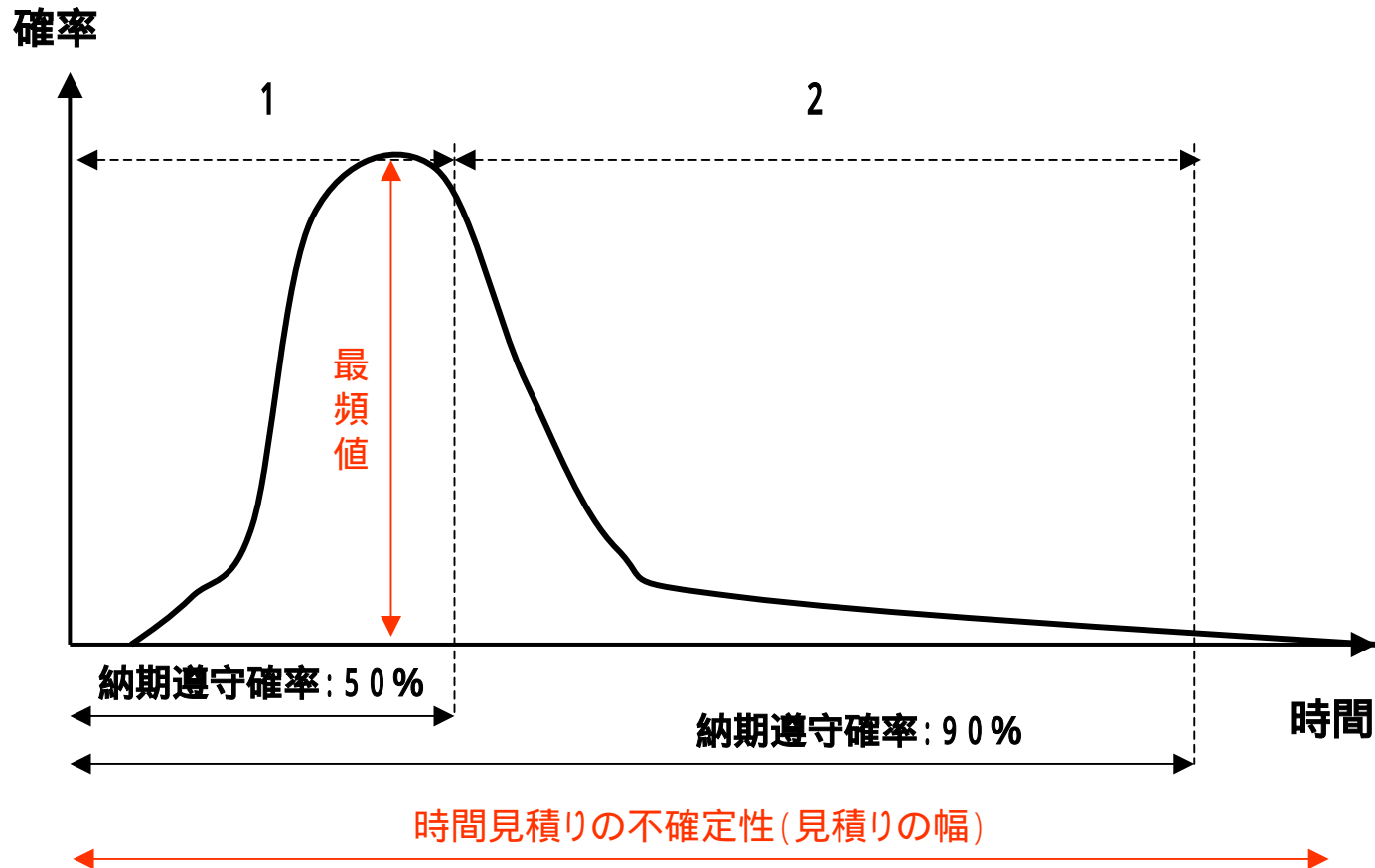


# PERTの時間見積りの特徴

---

- ◆ プロジェクトの時間見積りはベータ分布に従う。
- ◆ 予定とおりに完了する確率が90%になる期間は、50%になる期間の3倍である。
- ◆ つまり、全体の半分の工程では見積の1 / 3で完了していることになる。
- ◆ 見積時間の2 / 3は安全余裕時間であり、管理階層が多いほど加算されて、プロジェクトの見積時間は長期化する。

# PERTの時間見積りはベータ分布



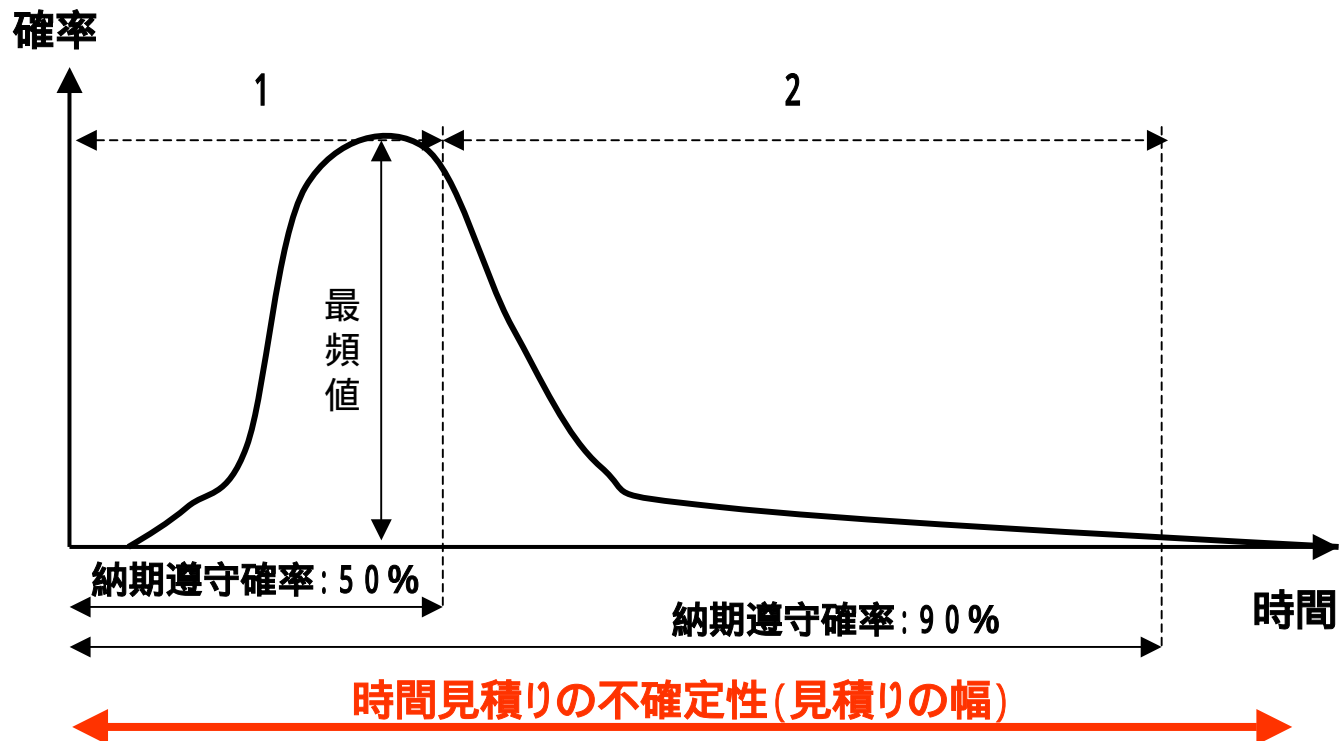
## 2. TOCとプロジェクト管理

---

- ◆ 時間見積りの不確定性
- ◆ プロジェクト遅延のインパクト
- ◆ 安全余裕の無意識内蔵メカニズム
- ◆ プロジェクト遅延メカニズム (安全余裕をムダにする)

# 時間見積りに内蔵された不確定性

- ◆ プロジェクト管理の諸問題の元凶
- ◆  $5 + 5 = 13$  の怪



# プロジェクトの必要時間見積りは増大する

---

- ◆ プロジェクト全体の安全余裕
- ◆ 個別工程毎の安全余裕
  
- ◆ 安全余裕を意識しない。入れていないと考えている。
- ◆ 現実的な時間見積りとは：
  - 過去の最悪の結果を基に見積もる
  - 80%以上の確率で時間を見積もる
  
- ◆ 現実的な時間見積りの結果として：
  - 実は各工程に**200%の安全余裕**が無意識のうちに組み込まれている。

# 安全余裕を追加するマネージャの行動

---

- ◆ マネージャ： 担当者の時間見積りに安全余裕を20%～30%を加算する。
- ◆ 上の階層のマネージャも安全余裕を加算する。
- ◆ 階層が多いほど時間見積りは増加する。

## 一律削減を見込んだ見積り行動

---

- ◆最上層のマネージャが納期遅延を発見する。
- ◆下位のマネージャに納期厳守を命令する。
- ◆その結果：一律に時間削減を通告する。
- ◆担当者は被害者意識を持つようになる。
- ◆賢い担当者/マネージャは一律削減を見越して、多めに時間見積りを行うようになる。

# プロジェクト遅延メカニズム (安全余裕をムダにする)

---

- ◆ 見積り Estimating: 実績は必ず計画をオーバーする
- ◆ 学生症候群 Student Syndrome
- ◆ パーキンソンの法則 Parkinson's Law
- ◆ マルチタスク Multi-tasking
- ◆ 遅れのみが伝播する No Early Finishes

# 見積り Estimating

## 実績は必ず計画をオーバーする

---

- ◆ 担当者の工数見積りには安全余裕が隠されている。
- ◆ プロジェクトの不確実性が安全余裕を要求する。
- ◆ 現実には、安全余裕が有っても工数(期間 + コスト)は超過する。なぜか？

# 学生症候群 Student Syndrome

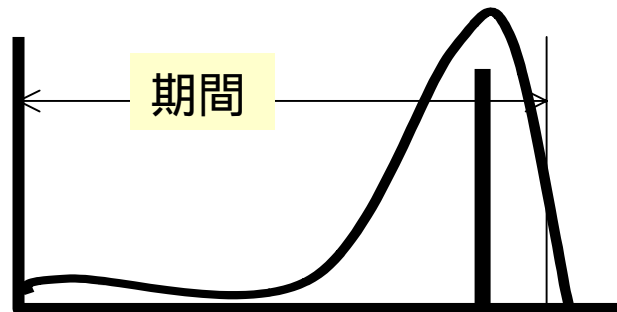
---

- ◆ 納期直前まで他の作業を行う「学生症候群」。
- ◆ 遅い開始は貴重な時間を浪費する。
- ◆ 問題は計画の80%進行時点で発生しがちである。

# パーキンソンの法則 Parkinson's Law

---

- ◆ 法則：仕事は予定工数を全て使い切るように拡大する。
- ◆ 早く終わりそうになると、手直しを行ったりして、納期まで仕事を抱える。
- ◆ なぜなら、従来のプロジェクト管理は「遅延しない」ように圧力をかける。
- ◆ しかし、予定よりも早く終わるようには促進しない。



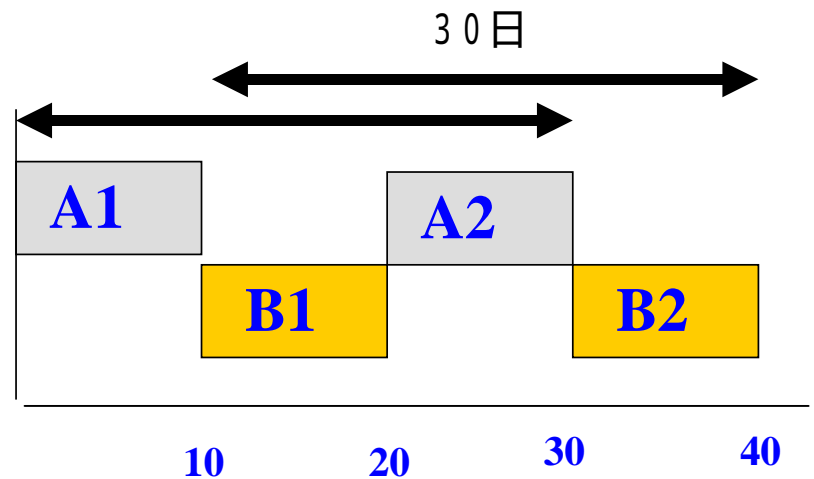
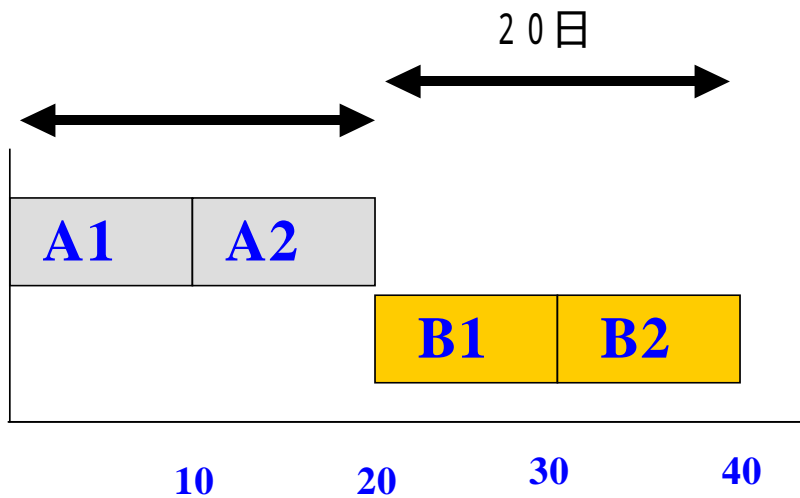
# マルチタスク Multi-tasking

---

- ◆ 現実にはマルチプロジェクト(複数のプロジェクトを同時に抱える)。
- ◆ 分業体制でのマルチタスク(1つのプロジェクト内でも複数のタスクを同時に抱える)。
- ◆ 原因は恒常的に圧力が担当者にかかっている。  
**真の原因は「優先順位的意思決定」が不在。**
- ◆ このためオーバーヘッド時間増大が必然的に発生する。
- ◆ 全てのプロジェクトの終了が遅くなる。
- ◆ プロジェクトの優先順位を決め、シングルタスクで行うと、プロジェクトのパフォーマンスは格段に向上する。

# マルチタスク (Multi-tasking)

- ◆ マルチタスクでは結局ほとんどが遅延する。
- ◆ マルチタスクの状態は他人には見えない。



# 遅れのみが伝播する

---

- ◆ 従来のプロジェクト管理は予定よりも早く終了することは期待されていない。
- ◆ 予定より早く仕事を完了しても上司は評価しない。
- ◆ 早く終ると、逆に計画能力が無いと叱責される。
- ◆ 次回には見積工数を削減される。
- ◆ 早く終わっても、納期まで仕事を抱える。
- ◆ このため遅延だけが後へ伝えられる。

## 3 . C C P Mによるスケジューリング

---

- ◆ ステップ1: プロジェクトに必要となるタスクを洗い出す。
- ◆ ステップ2: クリティカルパスを見つける。
- ◆ ステップ3: プロジェクトの含んでいる安全余裕をとる。
- ◆ ステップ4: プロジェクトバッファを設置する。
- ◆ ステップ5: 資源の競合を解消する。
- ◆ ステップ6: クリティカルチェーンを見つける。
- ◆ ステップ7: 合流バッファを設置する。

# Step1: 必要なタスクを洗い出す

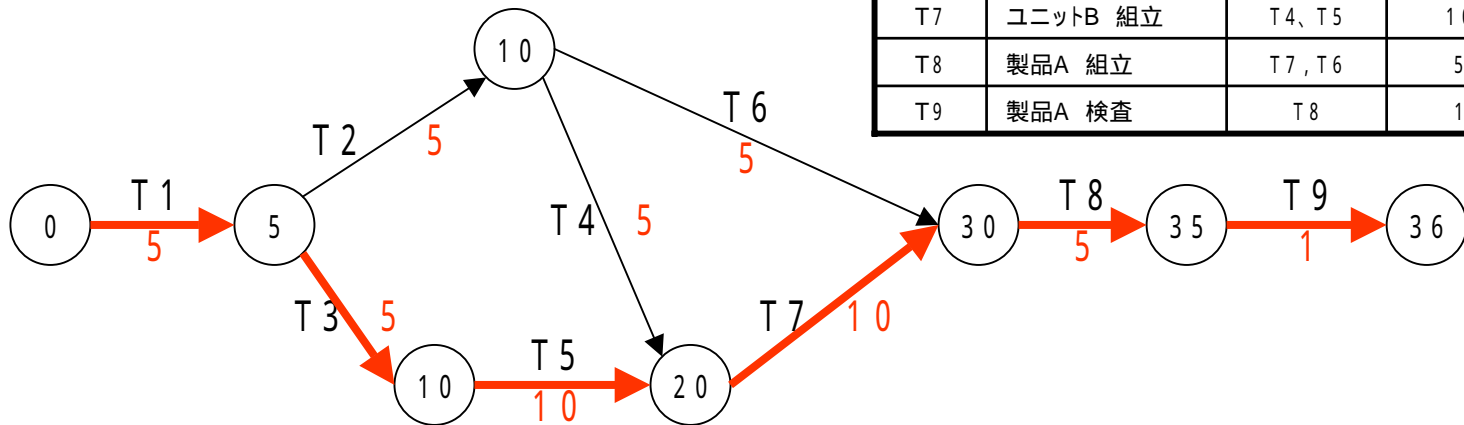
- ◆ 製品Aの製造計画をPERTで作成する。
- ◆ まず作業表を作成する。

タスク	作業内容	先行タスク	期間(日)
T1	設計	(なし)	5
T2	材料F 調達	T1	5
T3	材料G 調達	T1	5
T4	部品D 加工	T2	5
T5	部品E 加工	T3	10
T6	部品C 加工	T2	5
T7	ユニットB 組立	T4、T5	10
T8	製品A 組立	T7、T6	5
T9	製品A 検査	T8	1

## Step2: クリティカルパスを見つける

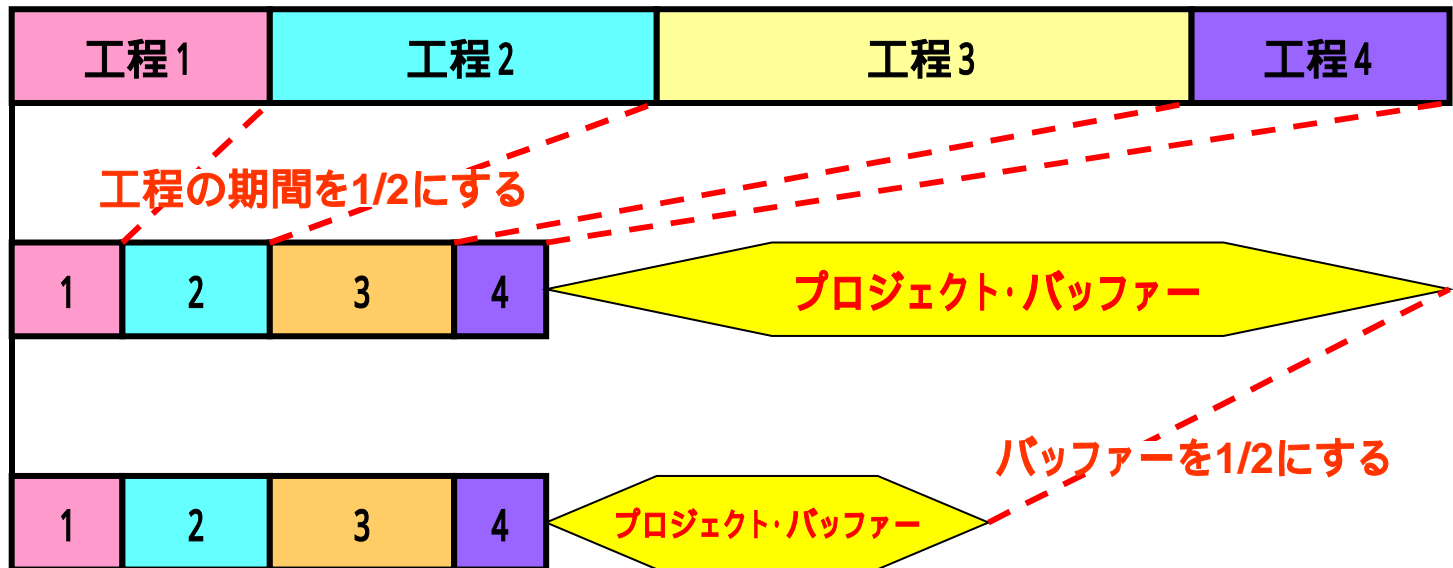
- ◆ 製品Aのアーロ・ダイアグラムを作成する。
- ◆ クリティカルパス = 最長経路
- ◆ クリティカルパスがプロジェクトの納期を決定する。
- ◆ ゆえに、クリテカルパスが制約。

タスク	作業内容	先行タスク	期間(日)
T1	設計	(なし)	5
T2	材料F 調達	T1	5
T3	材料G 調達	T1	5
T4	部品D 加工	T2	5
T5	部品E 加工	T3	10
T6	部品C 加工	T2	5
T7	ユニットB 組立	T4, T5	10
T8	製品A 組立	T7, T6	5
T9	製品A 検査	T8	1



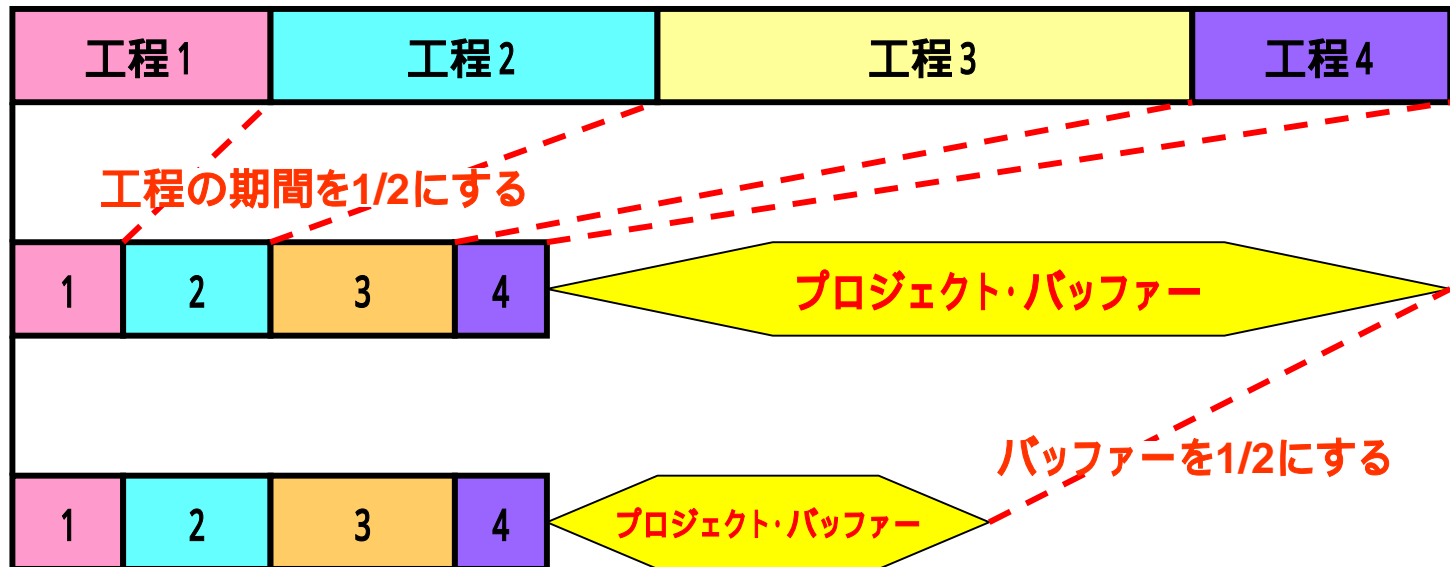
# Step3:プロジェクトが含んでいる安全余裕をとる

## ◆ 工程の期間を1 / 2にする



## Step4:プロジェクト・バッファ

- ◆ プロジェクト・バッファは、クリティカルパスの最後に置かれる。
- ◆ クリティカルパスの遅れからプロジェクト納期を守る。



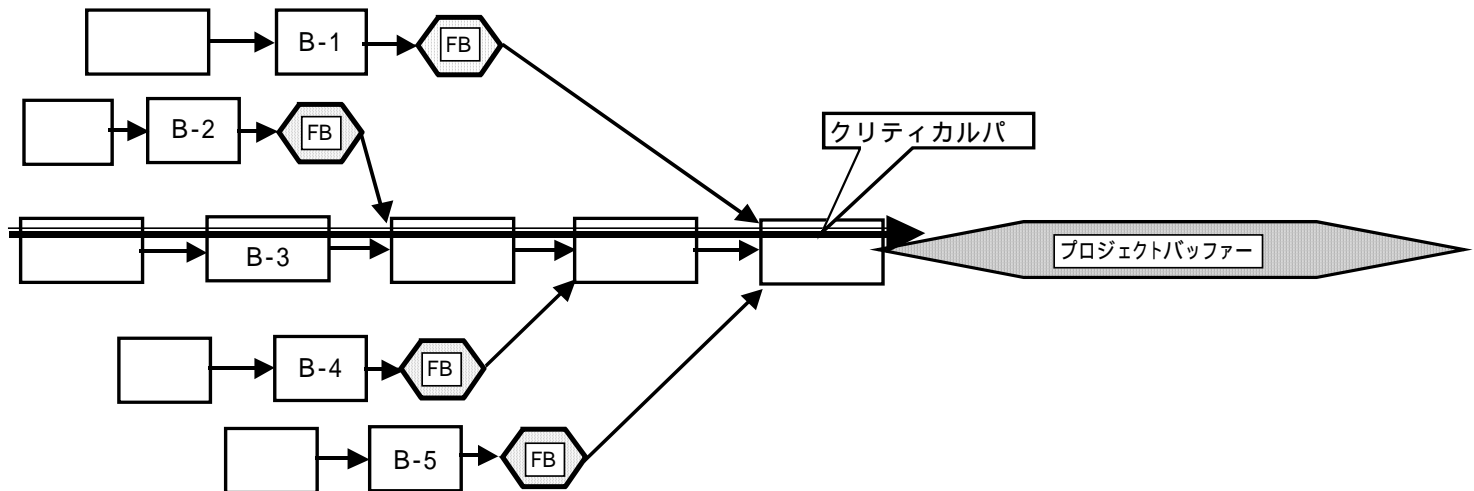
Step5: 資源の競合を解消する  
Step6: クリティカルチェーンを見つける

---

- 次章で説明します。

# Step7:合流バッファー Feeding Buffer

- ◆ 合流バッファーは、非クリティカルパスの遅れでクリティカルパスが遅れることを防ぐためのバッファーで、クリティカルパスへの合流点に設置する。
- ◆ さらに重要なことは、プロジェクトの制約がクリティカルパスからクリティカルチェーンへと変わることにより、合流バッファーの位置も変わる。



## 4 . 資源競合とクリティカルチェーン

---

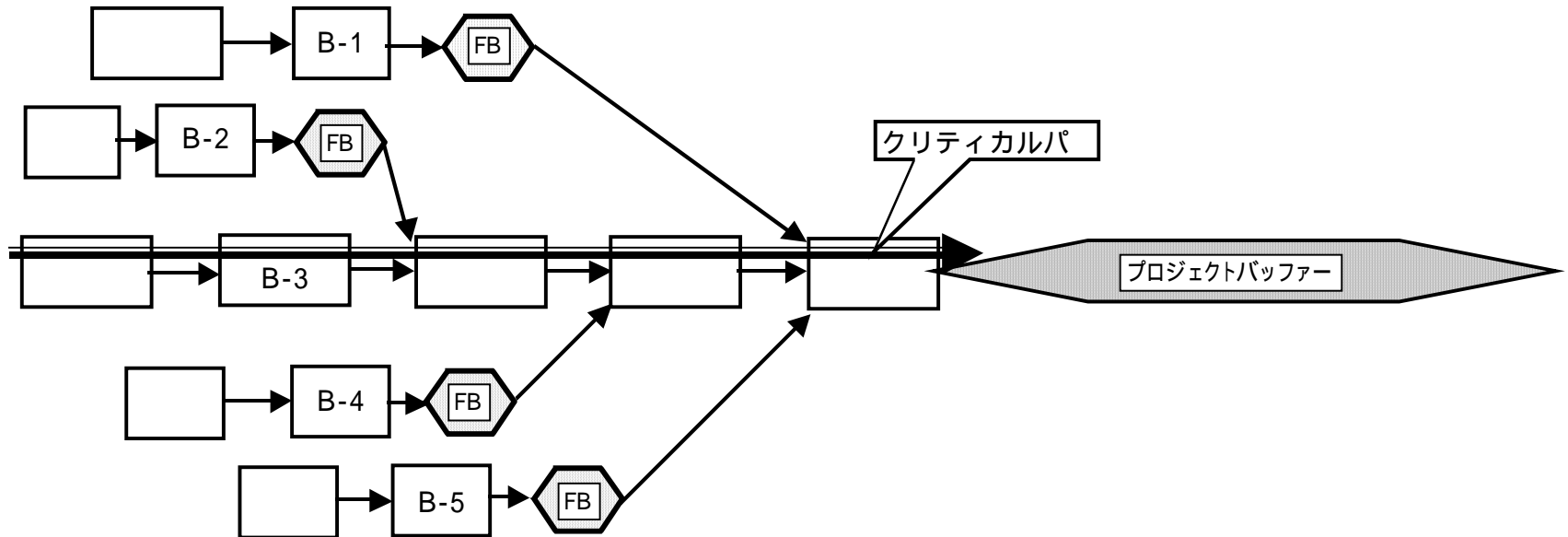
- ◆資源の競合
- ◆クリティカルパス
- ◆クリティカルチェーン

# 資源の競合

---

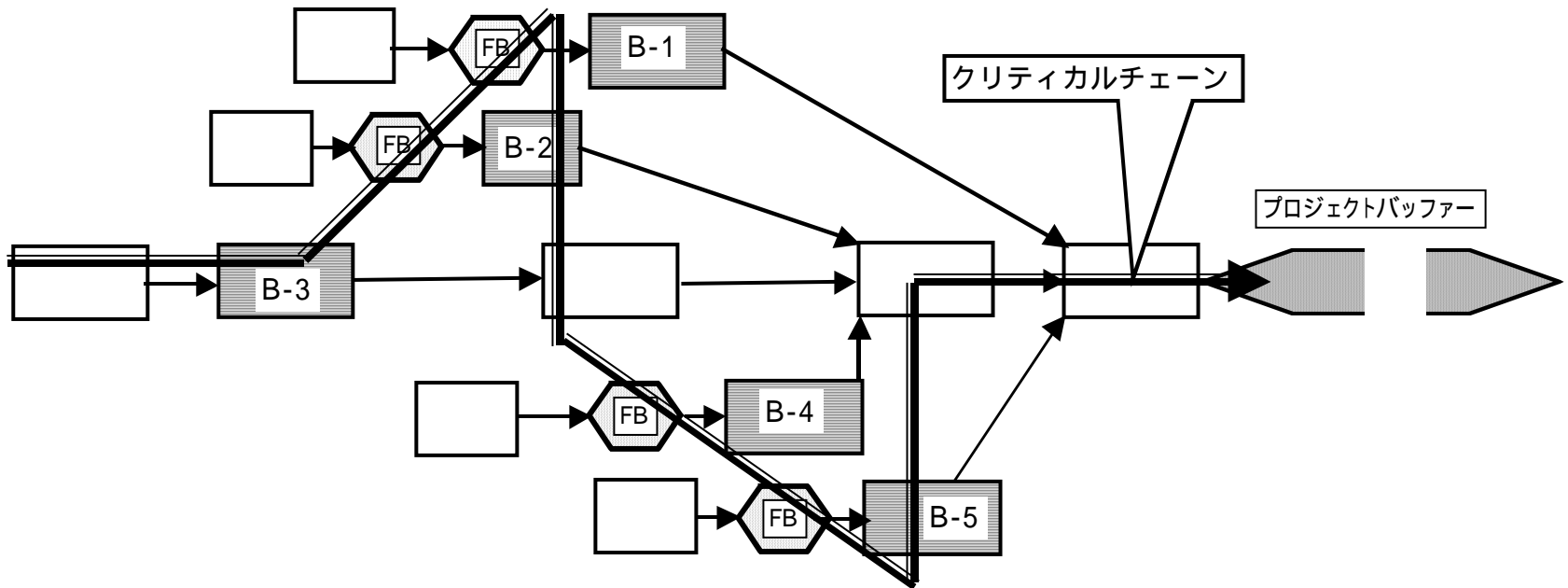
- ◆ 優先順位が不明確： 資源の競合を発生させる。
- ◆ 資源の競合の問題がない時は、クリティカルチェーンはクリティカルパスと同じになる。
- ◆ プロジェクトのボトルネック = 競合する資源
- ◆ 単一プロジェクト内の資源競合
- ◆ 複数プロジェクト間の資源競合
- ◆ 資源競合の結果： 工程間の従属性が発生。
- ◆ マルチタスクの状態になる。
- ◆ クリティカルパスが途中で変化する現象を発生させる。
- ◆ バッファ管理が不能になる。
- ◆ 結果： プロジェクトの大幅遅延。

# クリティカルパス(最長経路)



◆従来のPERTのクリティカルパスは、プロジェクトに使われる資源の競合といった動的な条件を考慮していない。

# クリティカルチェーン(資源競合を回避)



- ◆資源の競合があり、資源は一度に1つの作業しかできないという条件を考慮した一連の作業工程が、プロジェクトの本当の制約条件になる。
- ◆これをTOCプロジェクト管理ではクリティカルチェーンと呼ぶ。
- ◆つまり、プロジェクトのクリティカルパス上の要素と制約資源を使用する工程(制約工程)の組み合わせを通るパスである。

## 5 . バッファ管理の方法

---

- ◆ プロジェクト・バッファの残量管理
- ◆ 合流バッファの残量管理
- ◆ リソース・バッファ管理

# プロジェクト・バッファーの残量管理

---

- ◆ プロジェクトの持つ安全余裕を全て取り除いている。
- ◆ 各タスクの作業時間は遅れることを前提に見積もられている。
- ◆ クリティカルチェーン上のあるタスクが遅れると、その情報がプロジェクト・バッファーに送られる。
- ◆ プロジェクトの責任者は、プロジェクト・バッファーを常に監視して、プロジェクトの状況を常に把握していなければならない。
- ◆ もしも深刻な遅れ(プロジェクト・バッファーの残量が減少)が出たならば、要員を増やしたり、作業効率を上げるために設備投資を行うなどの対策を打つ。

# 合流バッファーの残量管理

---

- ◆ プロジェクトの責任者は合流バッファーも常に監視する。
- ◆ 非クリティカルチェーンに遅れが出て、クリティカルチェーンに影響が出そうであれば、必要に応じて対策を打つ。
- ◆ 合流バッファーが全て無くなっても、まだプロジェクト・バッファーが残っているので、プロジェクト・バッファーとの関係で対策を打つ余地がある。

# リソース・バッファ管理

---

- ◆ リソース・バッファ：各タスクの資源が使用開始日時になっても利用できないことがないように、資源を予約しておくこと。
- ◆ 特にクリティカルチェーン上のリソース・バッファを監視する。
  - 10日前に予告
  - 3日前に予告
  - 前日に最終確認
- ◆ CCPMでは担当者に開始日と納期を知らせないため、リソース・バッファ管理で開始を知らせる。

## 6 . 複数プロジェクト管理

---

- ◆ 複数プロジェクト間の資源競合
- ◆ 複数プロジェクト・スケジューリングの方法

# 複数プロジェクト間の資源競合

---

- ◆ PERT/ガントチャートは資源競合の意識無し。
- ◆ 当初のクリティカルチェーンはプロジェクト内の資源競合のみ意識した。  
クリティカルパスにボトルネック資源の経路を付加して、クリティカルチェーンと命名した。
- ◆ 現実には複数のプロジェクトが走り、ボトルネック資源の競合が起き、原因不明のプロジェクト遅延を引き起こす。

## 複数プロジェクト・スケジューリングの方法

---

- ◆ ステップ1： 計画されている全てのプロジェクトでそれぞれ各資源をどれだけ必要とするかを算定する。
- ◆ ステップ2： どの資源に対する負荷が最も高いかを調べ、全プロジェクトの制約資源を探し出す。これは工場の制約工程を探し出すのと同じこと。
- ◆ ステップ3： 制約資源上のスケジュールを立てる。これは制約資源がプロジェクトの出力の総和最大化の制約条件になるため。
- ◆ ステップ4： 制約資源の前にバッファを挿入する。これは制約資源だけは絶対遊ばないようにするため。

お問い合わせは下記まで。

[hiranabe@msi-jp.com](mailto:hiranabe@msi-jp.com)

# 補 足

---

- 1 . Projectは未来のことー不確定要素が多い
- 2 . Management
- 3 . 見積りの仕方
- 4 . 技術者の能力 能力の単位
- 5 . ProjectリーダーのWL
- 6 . 完了基準の設定
- 7 . 変更管理
- 8 . 契約と言う考え
- 9 . チェックシート
- 10 . リスク管理の考え