

プロジェクトマネジメント講座

6.6 タイムマネジメント スケジュール作成

6.6.2 スケジュール作成【ツールと技法】

● クリティカル・パス法

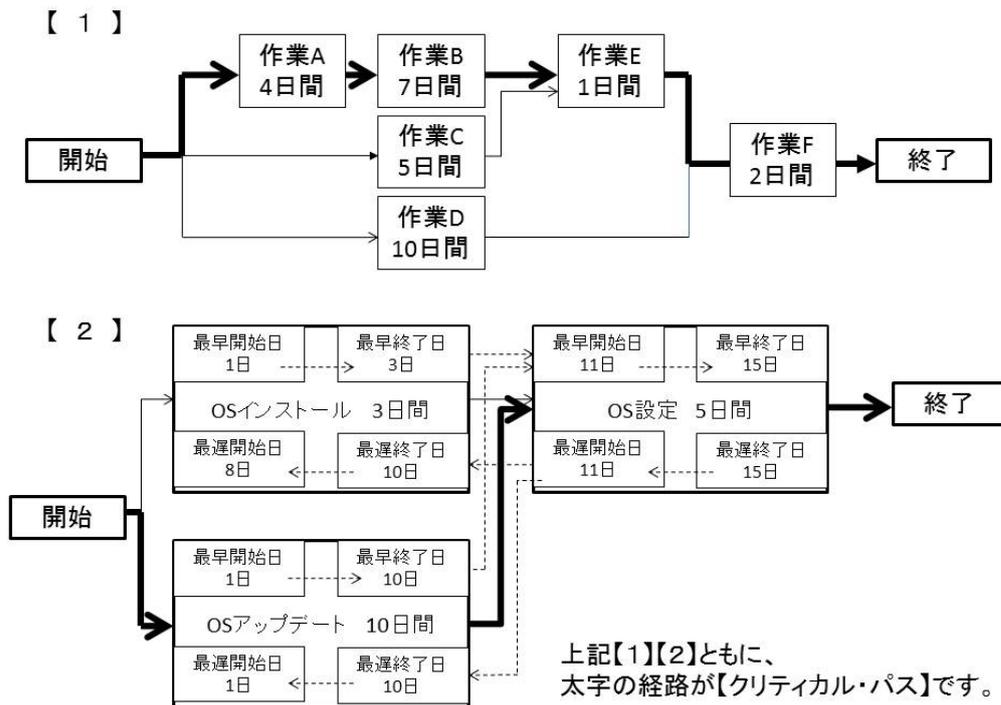
スケジュールアクティビティの理論的な最早開始日 (ES)・最遅開始日 (LS)・最早終了日 (EF)・最遅終了日 (LF) を計算する方法。またクリティカル・パスとは、プロジェクトを実行していく上で、スケジュール面での最長経路のこと。つまりクリティカル・パスとは、複数の作業のうち、開始から終了までをつなぐ時間的余裕のない一連の作業の集まりのこと。

ではクリティカル・パスとは何か？ より具体的に言うと以下のとおりです。

【クリティカル・パスの重要性】

進捗管理において、どの作業ルートが最長であるのかを認識して、このクリティカル・パスを形成する作業の管理には注意が必要になります。その理由としては、クリティカル・パスを形成する作業が、万が一遅延してしまった場合は、その時点で、納期に間に合わない等の事態になるため、何かしらの手段を講じる必要が出てくるためです。またクリティカル・パスでない作業をいくら短縮しても、プロジェクト全体の短縮にはつながりません。

ではそのクリティカル・パスを確認してみましょう！



上記【1】では、作業Aー作業Bー作業Eー作業Fが最長の作業経路となります。また上記【2】では、OSアップデートーOS設定が最長の作業経路となります。どちらの経路でも、作業での遅れが発生した場合は、プロジェクト全体に影響を与えることになるため、スケジュールを管理する上で注視する必要があります。また上記【2】には、最早開始日 (ES)・最早終了日 (EF)・最遅開始日 (LS)・最遅終了日 (LF) の4つ名称が記載されています。それらは、計画上、各作業をいつから開始すれば、当初の予定に間に合うのかを示しています。たとえば、作業【OSインストール 3日間】は、8月1日から開始しても、また8月8日から開始しても、作業【OSアップデート 10日間】の完成を待たないといけないという

事が分かります。つまり作業【OS インストール 3 日間】は、8 月 1 日から開始しても、8 月 8 日から開始しても、計画上全く問題がありません。従いまして、その【OS インストール】という作業には、7 日間の余裕（フロート）【別名：スラック】があるという事になります。

【フロートについて】

フロートとは時間的余裕のこと（フロートの求め方： $LS-ES$ もしくは $LF-EF$ ）。フロートには 2 つの種類がある。

- ・ **トータルフロート**とは プロジェクト全体の終了時期を遅らせない範囲で、タスクの開始を遅らせることができる期間
- ・ **フリーフロート**とは 後続のタスク開始 (ES) を遅らせない範囲で、あるタスクの開始を遅らせることができる期間

＜その他の注意事項＞

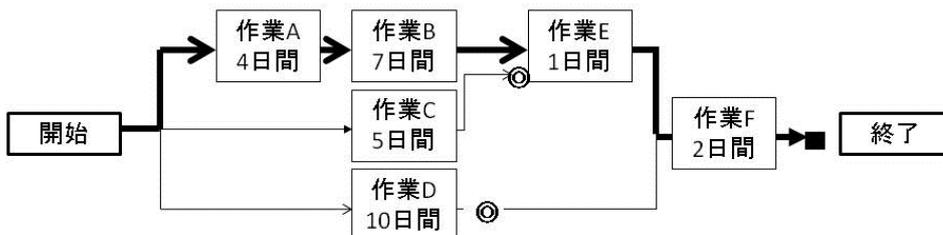
- ・ 複数のクリティカル・パスが存在する場合は、リスクが高くなる
→フロート 0 が複数存在している為、何かトラブルが発生した時に、作業が時間通り終わらない可能性が高まる
- ・ スケジュールの短縮は早めのアクティビティで行う方が良い（遅めのアクティビティであると、リスクが伴う）
- ・ 人を増員しても、スケジュールの短縮にならないケースもある

● クリティカル・チェーン法

要員の制約がある場合に、余裕を一切考えない厳しいスケジュールを計画して、実態に合わせ、弱点を補強するため必要なバッファ（予備時間）を追加して、スケジュールを修正する方法。

プロジェクトスケジュールを作成する際は、本来は、各工程が計画どおり確実に終了することを前提にしています。このため現場の作業者は自身の作業が計画に対して遅れが生じないように、あらかじめ余裕を確保しようとします。しかし、こうした余裕あるスケジュールになっていると、**学生症候群**（学生の宿題、テスト日までには時間の余裕がありすぎると、別のことに取り掛かって、ぎりぎりまで着手しない。この結果、最後に問題が発生し、間に合わなくなる。）や、**パーキンソンの法則**（早期に完了しても計画上の終了日まで次工程に引き渡さない）が発生する可能性があります。

【3】クリティカルチェーン法について



クリティカルチェーン法において設定するバッファは以下のものです。

◎合流バッファ

クリティカル・パス上にない工程がクリティカル・パスに合流する部分に設置される予備時間で、これに余裕のあるうちはクリティカル・パス（すなわちプロジェクト全体の完了期日）に影響を与える可能性はない。

■プロジェクトバッファ

各工程の予備時間を一まとめにしてプロジェクト全体で管理するもので、ネットワーク図の上ではクリティカルパスの後ろに置かれる。

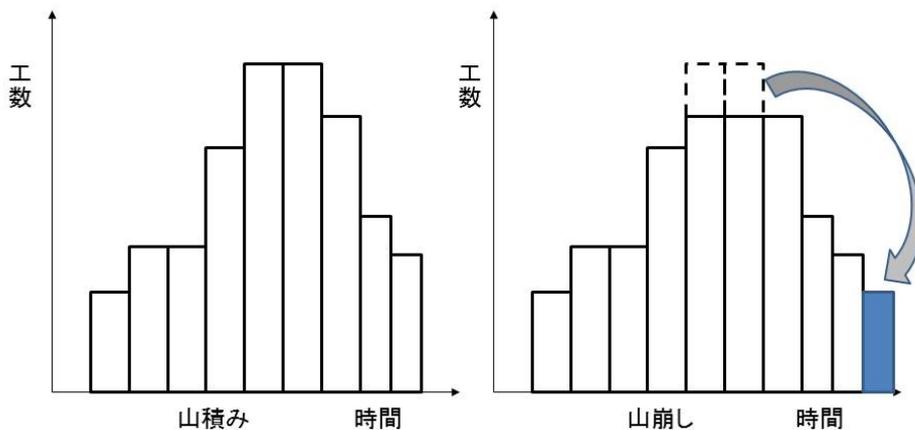
そこで、あえて各作業に余裕を一切考えない厳しい日数（一般的には、作業が完了する確率が50%である日数）を与え、もしその日数で作業を進めることが難しい場合は、各作業に**バッファ**を投下します。**バッファ**とは個々の作業ではなく、PMが管理する予備時間です。そのため、PMとしては、各工程の締め切りを管理するのではなく、予備時間がどれだけ消費されたかを管理するもので、「バッファが減るペースが速い」「バッファが残り少ない」という点に注意を払う必要があります。バッファの管理方法としては、3段階で考えておくのが良いと言われています。その3段階とは、グリーンゾーン（安全）、イエローゾーン（注意）、レッドゾーン（危険）です。それぞれの段階で許容値を決定して、バッファ全体の内、30%までの消費量であれば、グリーンゾーン（安全）、60%までの消費量であれば、イエローゾーン（注意）、90%までの消費量であれば、レッドゾーン（危険）と定めておくのが一般的です。

資源最適化技法

● 資源平準化 (Resource Leveling)

資源平準化とは、特定の時期に作業の必要工数が偏っている場合、特定の資源に合わせて、作業を均等化することです。並行作業が多く行われるプロジェクトにおいては、各作業の必要工数を時間軸に対応して集約し、必要人数を算定します。この方法を【山積み】といいます。しかし【山積み】の結果、要員に一定の限界がある場合は、新たな作業計画を考え、要員のピークを変更します。この事を【山崩し】といいます。では、以下の図で、この資源平準化について確認をしましょう！

資源平準化について



【山積みと山崩し】

図表で考えると、上記のとおりになります。上記のような方法を利用すれば無理のない作業工数を計画立てることができます。

確かに上記の方法を利用すれば、無理のない作業工数を計画立てることができますが、作業工数が減少する訳ではないため、**作業スケジュールが長期化する（クリティカル・パスが変更になる）ことが前提**となります。

● 資源円滑化 (Resource Smoothing)

資源平準化とは異なり、クリティカル・パスを変更させることがなく、フロートの中でアクティビティを調整する方法です。資源円滑化の実施が難しい場合は、資源平準化を検討する必要があります。

モデリング技法

- **What-If シナリオ分析** (別名：課題分析)

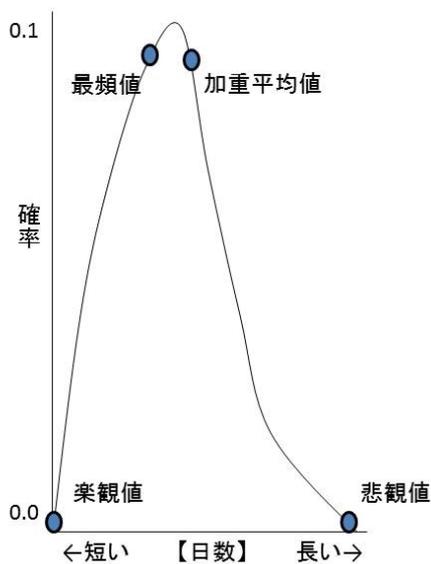
シミュレーション分析の1つです。現時点で考えられるシナリオに合わせて、いくつかのパターンでスケジューリングを行う方法です。そのシナリオには、ベンダーの納入の遅れ・ストライキなどの外的要因を含む場合もあります。またプロジェクトの所要期間を何通りも計算することになります。

- **モンテカルロ分析**

シミュレーション分析の1つです。コンピュータを利用して、ベータ分布・三角分布などの確率分布に従って乱数を発生させ、その値を積み上げることを繰り返すことで、最も妥当な日数でスケジューリングをする方法です。確率分布の図例は以下のとおりです。

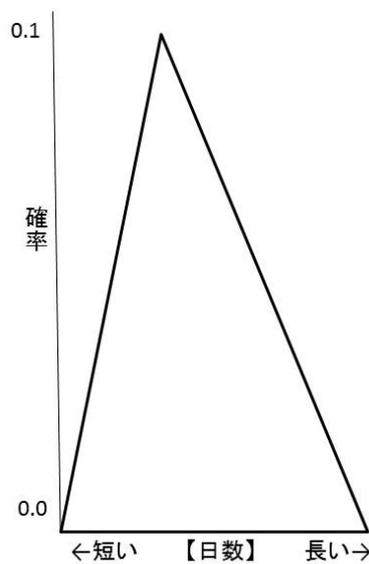
ベータ分布

(楽観値 + 最頻値 × 4 + 悲観値) ÷ 6 で算出



三角分布

(楽観値 + 最頻値 + 悲観値) ÷ 3 で算出



スケジュール短縮

- **クラッシング**

短縮したい作業に、最小の資源を追加して、該当する作業の予定所要期間を短縮化する方法です。例えば、クリティカル・パス上にある作業に追加要員を投入して、作業の完了を早める方法です。当然のことですが、**コストが増加**するため、**コストとスケジュールのトレードオフ**が求められます。

- **ファスト・トラッキング**

本来であれば、順番にすすめることが必要な作業を、前の作業が終了していない内に、開始する方法です。従いまして、前作業の結果（成果物）が、次の作業に使用されるようなケースの場合は、未完成の結果を利用して、次作業を並行して実施することになってしまいます。そのため、前作業が完全に終了した段階で、既に作業を開始している次作業では、何かしらの修正が必要になる場合があります。よってデメリットは、**リスクの増大・手直し**といわれています。