

The Dynamic Progress Method: A New Alternative to the Critical Path Method

ダイナミック・プログレス・メソッド: クリティカル・パス・メソッドに代わる新しい選択肢

翻訳：東なおふみ PMP

Overview 概要

クリティカル・パス・メソッド（CPM）が 1950 年代に導入されたので、プロジェクト見積もりと計画における最新のイノベーションについて学びましょう。その当時 CPM は大幅に改善されましたが、しかし今では現在の単純化された CPM のアプローチではとても不十分です。今日のプロジェクトはより大規模で複雑化しており、より厳密なアプローチが求められています。CPM を使うことにより、リソースはタスク期間と切り離され、現実的なプロジェクト計画を立案することを困難または不可能にします。CPM では「期間」はインプットです。しかしながら、現実の社会では「期間」はアウトプットです。リソースが適用され、仕事が完了するまで、我々は実際にタスクまたはプロジェクトにどのくらいの期間がかかるのか知りません。現在の期間ベースの CPM アプローチを使うことで、今日のプロジェクトプランニングツールは、プロジェクトに対して、実際のプロジェクトスケジュールの約 50% 下回る見積もりをし、プロジェクトの実際のコストの約 25% 下回る見積もりを出します。もちろん、このような非現実的な計画では、のちに多くの問題が発生する可能性があります。最終的に、全てのプロジェクトの作業はリソースの割り当てによって行われます。そこで、ダイナミック・プログレス・メソッド（DPM）は、リソースを土台とし、そこから上向きに構築します。DPM のようなリソースベースのアプローチでは、プロジェクト・マネジャーはプロジェクトで何を行う必要があるかを正確に指定する柔軟性と制御性を備えています。これにより、より現実的で、より信頼性のある（最も重要なことに）より成功したプロジェクトプランに導きます。

NOTE: システムが視聴記録を正確に記録するために、ビデオは最後まで通して視聴されなくてはならない。PDU についての詳細は ProjectManagement.com を参照すること。

司会: 本日は J・クリス・ホワイト氏による「ダイナミック・プログレス・メソッド：クリティカル・パス・メソッドに対する新しい選択肢」のウェビナーに参加いただきましてありがとうございます。ホワイト氏は ViaSim ソリューションでシステムダイナミクス、その他のモデリング、シミュレーションの取り組みを指揮されています。彼は、数冊の本とトレーニングマニュアルそして多数の記事を発表し、プロジェクトマネジメント、戦略プランニング、TQM、シックスシグマ、リーダーシップ、モデリング、シミュレーションの分野において数多くの会議で発表されています。そして、リーンに関する師であり、シックスシグマ マスターブラックベルト、プロジェクトマネジメント・プロフェッショナル、認定スクラムマスターです。それでは早速 J・クリス・ホワイト氏をお迎えしましょう。

講演者: デレクさんありがとうございます。また、皆さんがダイナミック・プログレス・メソッドについて学習するために忙しい時間を割いてくださりありがとうございます。「ダイナミック・プログレス・メソッド：クリティカル・パス・メソッドに対する新しい選択肢」がなにを意味するか理解されるでしょうが、少しだけ明確にしたいと思います。ここでの「クリティカル・パス・メソッド」は、期間ベースであるアプローチについてを指しています。そして私たちはそれが意味することをもう少し掘り下げ、クリティカル・パス・メソッドが一般的に使用されているアプローチであることを理解していきます。ですので私からクリティカル・パス・メソッドまたはCPMという言葉を開くでしょう。ですが実際には、期間ベースもしくは、期間のインプットに基づく種類のツールについて話します。もう一度、参加して下さりありがとうございます。私はとても恐縮しています。私はこのチャットボックスを見していますが、こんにちはこのメッセージいただいています、40ヶ国目で数えることを辞めました。大変素晴らしいことですね。私の故郷のテキサス、ダラスから参加してくれている人もいます。ようこそ、皆さん。それでは早速、始めましょう。

ダイナミック・プログレス・メソッドは仕事の中で、10年以上前に私たちが抱えていた問題から始まりました。多くのプロジェクトはたくさんの努力と誠意をもっていても、依然として遅延したり、予算超過しています。どれだけのプロジェクトが失敗しているかが、議論の対象になっています。PMIの Pulse of the Professionを見ると、約3分の2は成功し、3分の1は失敗していると言われています。Standish Group Chaos Reportsを見ると、それはまた違って、約3分の1が成功し、3分の2が失敗していると言われています。そこで、3分の1と3分の2のプロジェクトの間どこかで、遅延、予算オーバー、もしくは両方による失敗が見られます。私たちは長い間このことを見てきています。そして、ウェビナーに参加されている皆さんもそれについてよくご存じだと確信しております。

これは珍しい問題ではありません。しかし、ここでの仕事は、アメリカの国防総省(DoD)内の国防総省国防高等研究事業局(Defense Advanced Research Projects Agency)であるDARPAから始まり、そして複雑なDoD問題に高度な数学的手法を適用する解決策を出しました。もちろん私たちは、国防総省の中でよく発生している遅延もしくは、予算オーバーの問題を提案しています。私は皆さんも、国防プロジェクトの多くが遅延や予算オーバーしたという話を聞いたことがあると確信しています。私たちのDARPAへの提案の焦点は、プロジェクト計画と、マネジメントについてでした。そして、最初の目標は私たちがこれらを実世界に取り込むことができることを証明することでした。それらは、大規模で複雑なプロジェクトを計画、管理するのに役立ちました。

そして、私たちの2番目の目標は、もしツールの中でそれを取り込んだ場合、そのツールは現在のアプローチより精度の高い見積もりを出すことができるということを示すことでした。繰り返しますが、クリティカル・パス・メソッドまたは、期間ベースのアプローチについて話します。これが、ダイナミック・プログレス・メソッドアプローチを実際に作り始めた背景です。そのリサーチで得られた幾つかの結果は、私たちはポートフォリオプロジェクトを説明するための特定の基準を満たしていなければならず、平均的な期間またはスケジュールは約448日、すなわち1年と少しであり、コストはおよそ197万ドル、およそ200万ドルで、18,000労働時間を少し超えるといえます。LHとは、Labor Hour(労働時間)の略です。

ダイナミック・プログレス・メソッドでは、もっと現実的に、これらの期間はより平均的な691日、コストも平均的な約250万ドル、そして大体同じか少し多い労働時間をお見せできました。それを数字にして私たちがお見せしたものはもっと現実的で、スケジュールは期間ベースのツール、もしくはクリティカル・パス・メソッドベースのツールが示したものより平均的に54%長く、コストは約26%高くなり、労働時間に関しては、たった2%の変更もしくは差異でした。私たちにとっては、このことが意味することはプロジェクトに関して何をすべきか知っており、労働時間はとても近いということです。私たちはどのくらいの仕事をする必要があるかを知っています。それは問題ではありません。しかし、現在利用可能なツールは、現実的にこの情報を使っていません。それが、私たちが目指すところで、私たちの研究はそれを開始点として構築されました。

クリティカル・パス・メソッドに関して皆さんにお教えようとしているわけではありませんが、私は意図的にこの1950年代に作成されたこの古いダイアグラムをお見せしています。これは単なる基本的な概念図です。この場合、全てのメモと矢印はアクティビティを表しており、そのアクティビティには継続時間があることがわかります。ここで重要なことは、それが非常に継続時間に基づいているということです。私たちがこれらのアクティビティを見るのは、それらにどのくらいの期間かかるかということを見るのです。

しかし現実の世界では、ほとんどのプロジェクトはとても複雑で、以前よりもより多くの不確定要素を持っているように見えます。労働生産性の問題は現実のものです。これらの問題は、私たちが使うことになるかもしれないどのようなツールにも組み込む必要がある実際的なものです。なぜなら、繰り返しますが、それらは現実だからです。そこにはプロジェクト・マネジャーが対応し、是正措置を取らなくてはならない遅延や混乱が常に存在します。そして、マネジメントの是正措置、私たちがプロジェクトに対して行うこれらの変更、要員に対する取り組みは常にプロジェクトの方向を変えます。そして勿論、是正措置の目的は、正しい方向に進み続けること、スケジュールおよび予算を守り続けることです。見積もりは、大抵の場合リスクをカバーするために水増しになっています。

極端な例を挙げます。あなたが時間通り、予算取りに物事を確実に完了する方法の1つは、期間とコストを10倍多く言うのです。そうすれば確実に、あなたはそれより少ないものでプロジェクトを完了することができます。したがって、私たちがプロジェクトに関与している場合、何が起こるか分からないため、これらの水増しされた衝撃を和らげるような見積もりにより、リスクをカバーしていることを見ることができます。しかし、最終的には、私たちはプロジェクトプランニングツールに依存して仕事をしています。これらは単に紙と鉛筆を使って行うことができないほど複雑で、実際に紙と鉛筆を使う必要があった50年代にクリティカル・パス・メソッドは開発されましたが、現在はプロジェクトはもっと大きく複雑になっているので、コンピューターがそれらすべての仕事を私たちの代わりにしてくれます。ですが、私たちを支援するためのツールが必要であり、ツールを利用して正しい答えを得、ツールから得た情報をもとに意思決定することができます。

しかしながら、CPMベースのツールはかなり間違えることがあります。例えば私が使ったツールの幾つかでは、要員を2倍にして期間を半分にすることができます。ですので、もし私が10日かかるタスクをもっており、それに1人割り当てたとします。もし、そこにもう1人追加したら、ツールは自動的に5日間とってくるでしょう。それは期間の半分です。もう10日でなく5日です。場合によっては、それは非現実的になり得ます。

「9人の女性が集まっても1か月間で赤ちゃんを産むことができない」という例えがありますが、スタッフを投入しても必ずしも仕事を早く進めることができるということではありません。もしあなたが、ソフトウェア開発の世界にいるなら、おそらく「架空の人月」をよく知っているでしょう。それは本質的に、もしあなたが遅延しているソフトウェアプロジェクトがある時、それに対し人員を追加することは、それをさらに遅延させることとなります。なぜなら、追加投入された人員は、プロジェクトで起こっていることを新たに学ぶ必要があり、しばしば修正しないといけないバグを作り出し、プロジェクトに熟練した要員は新規参加した要員の訓練、バグの修正等に時間を取られるからです。というわけで、架空の人月がこれをとても良く表しています。

CPMツール内の他の誤解を招く恐れのある仮定は、社員の労働時間を2倍にすることで、期間を半分にすることができるというものです。例えば、社員が1日8時間働いて10日かかる仕事があるとします。もし私が社員に1日16時間働いてもらったら、その仕事を5日で完了することができます。と言うのは、繰り返しますが全く非現実的です。私たちは、ロボットではありません。人間はロボットではなく、疲れと疲労に影響されます。人間は、1日24時間働くことはできません。もちろん、ツールへの入力ならば1日24時間働かせることにできるでしょう。それにより期間は3分の1になるでしょう。とにかく、誤解を招く恐れのある仮定です。あるツールでは、要求された期間に社員が働けない場合、その社員は仕事をしないでさぼっているということになります。そこで、例えば私たちがある社員をある期間100%利用したい場合、その社員が1日、または2日間の間50%の時間しか活動できないならば、アルゴリズムは、実際に対象となる日をスキップし、50%の時間に対してその社員を使いません。ですので、ある場合には現実の世界の期間より長くかかると示すことがあります。

そして、もう一つは期間です。ここではタスク期間を意味しますが、もちろんそれらをまとめて、完全なプロジェクト計画となります。仮定の検証を全くせず、入力することが可能です。それは非常に簡単で、私がまだ仕事を始めたばかりの頃、上司が私に、この日に開始しこの日に終了することが必要であると言ってきたので、要求された日を開始日と終了日にした素晴らしいガントチャートを作成しました。ですが、私はそれがとても非現実的であることを知っていました。どのようにして完了すのだろうと思っていましたが、私はまだその仕事に対して駆け出しでしたので、上司が言ったことに従いました。しかし、重要なのは、それらの期間を持つことができ、それに対して必ずしもサポートする必要はないということです。これがしばしば引き起こすことはとても複雑な分析で、モンテカルロシミュレーションやこれら全ての複雑なもの、疑わしい変数などから始まります。そして、私が使いたいと思っているアナロジーは、マイクロメーター、このとても細かいマイクロメーターですが、振り向いて斧で細断しますが、正確ではありません。私たちが実際にマイクロメーターで測定し、斧で切り刻むというのは比喻です。

これをまとめると、クリティカル・パス・メソッドと、この期間ベースメソッドの重大な欠陥は、タスクの継続時間がツールへインプットされることが1つ目の要因です。現実の世界では、持続時間はアウトプットです。私たちは、人員を使い、それが実際起こるまで何が起こるかわかりません。そこで、期間がインプットであるという事実は、現実社会で見ていることの反対です。多くのこのようなツールにおいて、すべてではないですが多くのツールで、生産性の影響は考慮されていません。例えば、スーザンとジョーは全く等しいと考えられていますが、スーザンは、ジョーよりもあるタイプの仕事において優れているかもしれませんし、またその逆かもしれません。ジョーは、スーザンよりもあるタイプの仕事において優れているかもしれませんが、それは考慮されていません。それは先ほど言ったように、プロジェクト・マネジャーたちによって調整されることが常にあります。もしあなたにプロジェクトを崩壊させるという概念があるなら、クリティカルパス上で物事を促進しようと試みたり、また、時間を追加したり、要員を追加したりしますが、これは是正措置で、実際現在のツールでは考慮されていません。現在のツールでは手動でそれを行うことができますが、ですが、ツール自体はそのような機能がないことをお話ししています。

結果としておこる問題は、第一の仮定が何度もチャレンジされないということがあります。プランを見て、その額面価格でただ受け入れるのは簡単です。繰り返しますが、すべての人員は同等とみなされています。私は、国防総省に関する多くの経験があり、プロジェクトにも参画しています。私たちは、プロポーサルを開始し、コストとスケジュールを作成しました。私たちは、このプロジェクトに最高のシニア要員がいて、競争力のあるコストとスケジュールを持たせてくれると想定しました。

しかし、契約が実際に決まってから9か月以内に、これらの要員は参加不可能になっていたため、私たちは、他の誰かを参画させなければならず、シニア要員の代わりに大学を出たばかりの要員を採用しました。ジュニアレベル要員と、シニアレベル要員の間には、大きな差がありました。したがって問題の1つは、全ての要員は等しいとみなされていますが、実際には違うということが分かっているということです。別の問題は、これらのツールでの要員の標準化が、しばしば一貫していないということです。私が、言おうとしているのは、人員の標準化の中で、あなたが知らないとしたらですが、要員に関する時間の幅を広げようとして、誰もが特定の日において割り当てられないようにしようとしていることです。1日あたり8時間です。8時間働くことになっていますが、8時間以上働かないようにしますので、そのために仕事を広げます。

それは人員の標準化と呼ばれています。いくつかのツールでは、人員の標準化を適用することができ、いつ、どのように作業するかについてさまざまな回答を得ることができます。そして、ある時いくつかのツールは、人員の標準化機能を完了できませんでした。それは5人中3人の標準化をすることができたといいましたが、そのうち2人に関しては割り振りが超過しており、アルゴリズムでは問題が解決されません。したがって、問題は現状のアルゴリズムでは必ずしも解決されないということです。プロジェクト計画自体は、プロジェクトの一度限りの見方であるという点では変化はありません。ですので、もし私が何かを変えたいのであれば、プロジェクト・マネジャーとして変更し、新しいプランを公表します。各プラン自体は変化はありません。時間の経過とともに変化するもの、ダイナミックなものは何もキャプチャしません。例えば、支払いが遅れたので、このマネジメントは是正措置を実行してそれを修正し、スケジュール通りに戻しましょう、といったような。

ですので、そのようなダイナミクスは含まれていません。そして、これらの全てが、プロジェクトプランはしばしば守ることが難しいということにつながります。そして実際に何度も計画を立案し、仕事の契約がされてから数か月後もまだそのような感じかもしれませんが、それでもこの計画を使って仕事を進めます。さもなくば、どのように人員が使われるかを正確に言うことは難しく、私たちが最後に仕事した時に完了することができたので、今回も今年中に完了できると想定して行うのです。これらのプランは、しばしば、いつもではありませんが、守ることが難しいです。それはもちろん、コストとスケジュールに対するリスクを増加させます。もし、私たちが堅実な計画を持っておらず、その計画のすべての仮定を明確にしていなければ、コストやスケジュールの目標を達成するという面でリスクが発生します。そして、このすべてが成功の確率をより低下させます。私たちが作成したコストとスケジュールに見合う可能性は実際低く、Chaos Reports と PMI Pulse of the Professionの数値によると、100%を達成していないということがわかります。そして、繰り返しますが多くのツールを見てみると、この持続時間ベース、またはクリティカル・パス・メソッドエンジンが問題の一部となっています。

そこで、実世界の例を覗いてみたいと思います。これは、防衛契約者用で飛行機センサー開発プロジェクトです。そこで、私は実世界の例を覗くことで、幾つかの差異をお見せしたいと思います。私はたまたまこの為にMicrosoft Projectを使っていますが、使えるツールは他にたくさんあります。皆さんにお見せするのは、同じプロジェクトの3つの異なったバージョンです。まず1番目。私たちは1番目の計画を立て、39ヵ月で、3年を少し超えていて、1310万ドルです。私たちは、それが楽観的過ぎることを知っています。何故ならこの中に、超過に割り当てられている要員がいることを知っているからです。楽観的過ぎることを知っているのも、これが必ずしも達成しなくてはならないことではないことも知っています。Microsoft ProjectはOKだといっているので、これを週間ベースで平準化することを推奨します。それをすると、プロジェクトを実際に50年、599ヵ月に引き延ばすことができ、そして驚いたことに同じ1310万ドルでできるのです。

これは、悲観的過ぎるということを私たちはわかっています。私たちは、3年から50年になることはないわかっているのも、これも答えでないということも知っています。実際、あなたが進めるにつれ、30年以上何もしないで待機している人を見ることがあります。ですが、私たちがたとえ何を変更しても、それが変わることはありません。週間で標準化するときそれはとても広く広がります。繰り返しますが、私たちが知っているこの楽観的なものは現実的ではなく、悲観的なものも現実的ではありません。そこで、私たちは、月単位で標準化します。そして、中間の数を得ます。それは77ヵ月で、6年を少し超えます。そして、繰り返しになりますが、驚くことに依然として1310万ドルです。そしてその値が他の2つの値の間にあるからと言って、それが正答であったり現実的なものであるということではありません。それは実際私たちにとって予想不可能です。全く同じプロジェクトに3つのバージョンがありますが、どれでしょう？そこで、少しそれらを覗いてみたいと思います。

ですが、私は一歩さがってプロジェクト・マネジャーが使うかもしれない典型的なアプローチについて話したいと思います。これは数個のプロジェクトに参画している人で、これは彼らの初めてのプロジェクトです。初めてのものではなく何回か経験を積んでいて、彼らは何をすべきかを知っています。ですので、彼らが行ったことは、行う必要がある全てのものを把握することでした。そしてタスクを見ると、しなくてはならない期待されている仕事の一部があります。ジョーが、この仕事をする事になっています。何故ならジョーと私は、ジョーが効果的、もしくは効率的なのを知っているからです。そして、私たちがオフィスにいて1日8時間働いていて8時間の時間を利用できることも知っています。そのことから、期間のようなものと、そのタスクの期待される期間を得ます。さて、ここで数字を入れると、計算すべき数字が出てきます。

80時間の仕事。もしジョーが担当していて、彼がオフィスにいる1時間ごとが100%生産的であれば、実際その仕事を1日8時間働くことで完了させることができます。80割る8は10ですので、この作業の期間は10日間であるということがわかります。CPMツールがPERTを行い、それはCPMのもう一つの特徴です。そしてそれらはしばしば、そのような相互性もしくはPERT/CPMを使用しますが、この期間を使います。そこで、タスク1が10日間で、タスク2が5日間である例をお見せしています。ダイナミック・プログレス・メソッド(DPM)は方程式の反対側を見て、これらをインプットとして使います。DPMは、人、生産性、どのくらい働くのか、などを見ます。それについてもう少し詳細を見ていきましょう。

ですが、ここで注意すべき面白いことは、前提されている仮定がすべて正しいと証明された場合、これらのメソッドは全く同じ答えを出すということです。したがって、全ての過程が真実であると証明されれば、これらは同等なのです。それでは、ダイナミック・プログレス・メソッドが何をするか少し見てみましょう。それは、異なっています。あなたがここで目にしているものは、全てのタスクでシュミレーションされる基本的なエンジン、モデルです。先ほどの画面で見たものは、このようなものでした。すべてのタスクに本質的に作業すべき、あるペースで完了させるべき多くの作業項目があり、それが、この完了した仕事になります。この場合の作業は左から右に移動するので、左に80時間、右に0時間があり、タスクが完了したときに、左側が0時間になり、右側が80時間になります。

そして、この仕事が行われる速度によって、どれだけ早く達成されるかが決まります。その完了率は、基本的に、効果的な要員の時間に基づいています。それは、何人の人が働いたか、人員は何人か、そして彼らは何時間働いたかです。以前の例では、人員であるジョーがいて、1日8時間働き、毎日8時間の効果的な作業が行われ、1日目終了した後、8時間が左から右に移動することを見ることができ、そして、私の仕事量は80から72に減り、完了した作業は、0から8にはねあがります。さて、このグラフで黒く見えているものは、今日ツールの中に存在しているものです。

全てのツールではありませんが、いくつかのツールはその機能を持っているので、この機能が存在すると言っています。そして、Microsoft Projectについて話しましたが、Effort Driven モードでそれを使うと、このようなアプローチを使用します。完了率や、仕事を先に進めていかどうかを決定するために使用されるのは、他のタスクに関係しています。ですので、タスク2を開始する前にタスク1は完了していなければならないので

すが、タスク1が完了していないのでそれは見るからに窮屈なものになります。DPMが少し違うのは、一般的な完了ー開始、開始ー開始、リード、ラグの一番重要な関係、そしてパーセント完了を表すことです。そこで100%完了の意味は、完了ー開始と同意になります。タスク1は、タスク2が開始される前に100%完了しなければなりません。0%完了は、開始ー開始と同じです。

それは、タスク1が0%完了か、もしくは開始時であるときにタスク2が同時に開始できることを意味します。そして、これはリードとラグを処理する良い方法であるのと対照的に2のリードを持つ10日間の作業は8日間になります。代わりに、それを80%の完了として表現してみましょう。そうすることによって、それが少しだけうまくいき、それはそれほど複雑ではないことがわかりました。しかし、繰り返しますが、黒色のものは、ツールとして取り込まれています。DPMがすることの1つは、スーザンはあるタスクにおいてジョーと異なっている、もしくは、ジュニアレベルの要員はあるタスクにおいてシニアレベルの要員と違っているということを見せることです。そしてそれが広がると同時に、物質的な依存関係の概念も広がっていきます。言い換えると、タスク1が完了するのを待つ必要がないということです。

タスク1がエンジニアリングの資料、または成果物を作成する代わりに、そしてタスク2がその成果物を要求するか、またはその作業の一部としてそれを消費すると完了率依存とは対照的に実際の材料依存があるが、そこにはほぼ同等であるが、少し違ったメソッドがあります。そして、繰り返しますが、それはここで取り込まれるものです。どの時点においても、このタスクやプロジェクトのステータスを得ることができ、そのステータス計算としてアードバリューを使うことになりませんが、これは防御のために多く使われます。それは、コストとスケジュールパフォーマンスもしくは、プレッシャーを設定するためのとても一般的なメソッドであります。どの時点においても、何をするか知りたいということをステータスと考えてください。私たちは予算をオーバしているのか、予算が足りないのか、過度のスケジュールなのか、スケジュールより遅れているのか、スケジュールより進んでいるのでしょうか？

そしてそれがステータスと考えてください。繰り返しますが、現在のツールはその機能を持っているのか、またはサードパーティーのアドインツールでより良い仕事をするのか。私は実際既存のツールに機能追加またはレイヤーを追加するサードパーティーのツールがあることを知っています。とにかく重要なのは、このステータスです。DPMが追加したものはそのステータスに基づいては是正措置が取られている可能性があり、そのことは、先ほどお話ししました。例えば、私たちは予定より遅れているかもしれないので、プロジェクト要員に残業をしてもらいます。これは、私がこの仕事が予定より遅れていると気づいたので、是正アクションで仕事時間を追加したのです。しかし、現実世界の疲労による影響は、人員の生産性に影響を与えるため、必ずしも1から1を得ることはできません。そこで私は、短期間の間このチームに残業させ、生産性への影響は実際にそれほどありませんので、私はそこから利益を得ます。

しかし、これが長期間続き、要員に何週間も続けて6時間から8時間の残業をさせようとする、彼らは疲労し生産性は落ち、そのためある種の調整が必要になります。ここでこれが示しているのは是正措置ですが、その是正措置の実世界の負の結果です。要員の人数を変更できるということは言いました。そこで、繰り返しますが、私はスケジュールが遅延していることに気づき、その仕事に何人かの人員を追加し、有能な人員を追加して、有効時間数を上昇させようとする。しかしここでも、過剰の人員投入で、生産性に実世界の影響がでます。台所に多すぎる料理人、仕事に対し要員が多すぎるということは、調整を難しくするか、またはソフトウェア開発における架空の人月について議論したように、彼らは別の問題を発生させたり、生産性を低下させたりします。

それは物理制約であるかもしれません。私は、国防総省の中で飛行機に関連する、ホイールの中の着陸のギアの仕事をしていたが、例えばそこには2人分のスペースしかありませんでしたので、10人を投入して仕事を完了させることはできません。そして、もちろんこれらの是正処置は、私たちの予想より遅かったり、何らかの混乱が発生したり、何かが遅れている時に下請け業者からの成果物が来るのを待っている時に起こります。それが、これらのすべてのことです。これはすべてプロジェクトを計画に戻そうとして実施するものなので、DPMでキャプチャしたいと思うものです。そこで、ダイナミック・プログレス・メソッドは、クリティカル・パス・メソッドでお話した3つの問題に対処しており、期間はこのリソースのインプットをベースとしたアウトプットで、生産性が考慮され、管理是正措置も考慮されます。この運用シミュレーションを作成するために、今日のコンピューターのパワーを使いますので、今日の多くのツールで使用されているとても簡易的な期間ベースのクリティカル・パス・メソッドエンジンとは異なったエンジンです。そして、クリティカル・パス・メソッドは紙と鉛筆でこれを行うことができる50年代に行われていました。それは、当時必要であった簡易的なアプローチの中核であり、その当時は先進的なものでしたが、現在はその有用性が失われています。

そしてもちろん、私たちが学んだことの他にも、それらを取り込もうとします。それでは、先ほどの例に戻ってみましょう。これは私たちが先ほど見たのと同じ3つのプロジェクトです。3つの全く異なるバージョンです。私たちが最初に行ったことは、これらの3つのバージョンのそれぞれをDPMシミュレーションに取り入れることでした。3つのバージョンの各々において、全く同じ結果になりました。私たちがより現実的なシミュレーションエンジンをCPMに当てはめ使用した時、その3つがチャート上のどこにあっても、3つ全てが正確に同じ答えを出したためより一貫性があり、より現実的なものを見つけました。そして、全てのリソースが100%生産性があると仮定していなく、実際オフィスのリフレッシュルームで話したり、別のミーティングに参加したり、トイレ休憩、喫煙休憩など他のことをしたりすることを考慮した85%の業界の標準を取り込むなら、1日8時間の労働時間に対して、全ての8時間を得ていません。そこでもっと現実的なものがあります。「少し時間がかかります。約7年くらいかかります。」と言うと、そのコストが増加することも知っているのも、コストの違いも見えてきます。

その利益は、DPMがすぐに私たちが問題が起こる前に決定できる実行可能な情報を提供します。何故なら、私たちが実際にシミュレーションしていることは、全てのダイナミックな変更を伴う反応であり、私たちが実施していることが最善なものであるかどうかを見ることができるからです。そこで、それが、リソース割り当ての間

題を解決することに役立ちます。何故ならそれは、文字通り1つのタスクで作業していて、他のタスクに変更したり仕事を行ったり来たりする人をシミュレートするからです。なので、実際にはすべてがキャプチャされ、現実の人員が実際にシミュレートされます。どの是正策がどの状況に最も効果的であるかを示すこともできます。場合によっては、少し時間外労働をしてスケジュール通りに戻すことがとても有用で役立つでしょう。

しかし、他のケースでは、時間外労働の代わりに人を追加することがより良い方法かもしれません。もしくは、両方の組み合わせもあるかもしれませんが、それが実際私たちが考えていたより生産性を低くするかもしれません。ですので、これらのすべてのメソッドのどれもがよくない場合もあり、何もせずにそのままにすることがより良いかもしれません。とにかく重要なことは、私たちの管理アプローチが、良いか悪いか、効果的かそうでないか、を私たちが実際に見ることができるので、私たちが取り組んでいるプロジェクトを実際どのように管理するかという良いアイデアを持てるということです。また、人員が複数のタスクで使用されている場所や、人員が上手く配置されておらずより長くかかっているタスクを確認できるため、リスクの高い領域も見つけることができます。最後に、この異なったエンジンがあるので、プロジェクトを加速するためのオプションを定量化することができます。そして、このプロジェクトに対して理論的に可能な最短時間がありますが、私はあまりにも積極的にコミットはしませんし、適切なコストとその調整もしません。

そして繰り返しますが、これらがDPMが提供するすべてのことです。加速の最後の例ですがみなさんのためにさっと見ていきたいと思います。私がおここに持っているチャートには、X軸、水平軸、期間があり、170日から260日の間の私たちが見ているどこかを拡大したものです。そして、Y軸、すなわち、垂直軸では、コストがドルで書かれていて25万ドルから始まり、最大50万ドルです。それでは、幾つかのシナリオを見ていきましょう。これが私たちのベースラインであるスケジュールです。おおよそ285日、29万ドルとしましょう。それが、私たちがここで開始するベースラインスケジュールとしましょう。もし、私たちが同じスケジュールで、少しだけ人員を是正するかどうかを確認するために管理アクションを適用すると、実際わずかに短い期間、275日、わずかに増加したコスト30万ドルとなります。

そこで、残業があり、それに対する費用が必要な場合の生産性の損失を考慮しているため、コストもしくは別の人員を持ってくるときの差異を見ることができますので、私たちはそれを短くしようとするので、使うことのできる変数を入力し、スケジュールを短くしようとします。そのスケジュールの80%を目指します。例えば、スケジュールが10か月だったとしましょう。私たちは、8か月でその仕事を完了しようとしていて、それが可能かを見ようとしています。そこで、私たちは管理アクションを使って、これが実際に実現できるかどうか確認します。そして、私たちは、スケジュールを例えば235日、最小限のコスト増加で大体30万ドルとなることがわかるので、大丈夫です。それで推し進めましょう。初めにできると言っていた3分の2の時間で完了しようとしており、管理措置が私たちがそれに向かうのに役立ってくれます。

ここで私たちが見ているのは、少しのコスト変更を伴う期間の短縮です。ここに、幾つか他の物を示そうと思います。私たちがスケジュールの半分、時間の半分で仕事を完了しようすると、実際にはほんの数日しか改善されませんでした。急速にコストが上昇しました。そこで私たちは、少しの期間を短縮するために多く

のコストがかかるという結論に到達しようとしています。そして、もし、スケジュールを4分の1にさらに短縮し続け、10カ月の代わりに2か月半でその仕事を完了することが可能かを確認しても、2か月半に短縮するのは無理だということが明確にわかります。私たちは実際にどの程度スケジュールを短縮できるかの限界が分かり始めていますが、私たちが努力し続けると、それに関係するコストがますます高くなります。そして最後の例は、10カ月のプロジェクトを1か月で完了しようとしているものです。そしてみなさんお気づきになると思いますが、実際には25%のシナリオより長くかかり、再びコストも増加します。

そして、曲線を引くと、ここに理論的な最短期間があることを見ることができます。それは、例えば190～195日で、かなりの費用がかかり、これについて2つのことに気づきます。1つ目は、多くのクリティカル・パス・ツールがあなたに信じさせようとしている線形でないことです。そして、繰り返しますが、もしあなたが人員を2倍にしたら時間を半分にすることができ、そしてGは、半分のコストになるということです。それは、必ずしもこのような線形である必要はありません。2番目ですが、激しく圧力をかけ、長い時間を費やし、さらにコストがかかる神秘の人月のような状況を作り出すことはできますが、曲線の最適点を通過してしまいます。とにかくこれは私たちがプロジェクトの1つとして作業した非常に興味深い例ですが、別のいくつかのプロジェクト上でも見ることができます。そこで、ちょっと現在市場で使えるツールを見ましょう。このうちのどれがダイナミック・プログレス・メソッドの要素を具体化しているか簡単に見て、それらを「認証」に入れます。

そこにはレベル1があり、これらはお互いの上に築かれています。レベル1は、単に人員の割り当てを使用して実際にそれをシミュレートしています。レベル2は次のステップですが、人員の生産性を取り入れているか確認しましょう。レベル3は管理アクション、または是正措置です。レベル4は、そうした管理アクションを生産性に反映しようという現実的な結果をもたらします。これらの4つのレベルをみてください。それぞれ他の上に構成されています。市場には多くのツールがあり、これらは全てではありません。とてもたくさんのツールがあります。私は、ちょっと主要なツールを取り上げてみました。多くのツール（スライド上：JIRA, Wrike, mavenlink, Basecamp）は、レベル1にさえ達していません。例えば、Basecampでは、サードパーティのツールでガントチャートを作成する必要があります。それでもそれは、開始日と、終了日を教えてくれるだけで、人員とのつながりはありません。

そして、これらのツールのいくつかは、それらを修正しようとしています。それでも未だにその機能を持っていません。人員の割り当ての機能を持つツールには、Premavera、Project、Projects Simulatorがあり、ProModel製です。そして、人員の生産性をもたらしてくれ始めているものはSpider Projectです。Premaveraの幾つかのハイバージョンは、そのようなことができます。そして、レベル4のpmBLOXですが、これはプロジェクト上でキャプチャしてクリティカル・パス・メソッドと比較できるようなツールで、私たちがたまたま作り出したものです。これは、そこにある風景をあなたに与えるようなものです。CRC出版により丁度発行された本の中で公表されているすべてのことについて語っているスライドで終了しようと思います。これは4月1日に出版された本の情報です。

そこにはより長い説明があり、先ほどのスライド上にウェブサイト、dynamicprogressmethod.comがありました。そこからはさらに多くの情報を得ることができます。これらすべてのことがさらに詳しく本とウェブサイトに記録されています。さて、これらはアプローチに対する簡単な紹介でしたが、幾つかの質問が来ていますので紹介させてください。アジャイルとウォーターフォールに関するいくつかの質問をいただいております、これはどのようにアジャイルと機能するのか、またはアジャイルとどのように違うかです。

そしてこの上ではタスクはタスクであり、ウォーターフォール対アジャイルは実際にどのような方法でも実際に機能します。それは問題ではありません。それは、あなたがプロジェクト・マネジャーとしてどのようにそのタスクを構成するかです。そこで、もしあなたがウォーターフォール アプローチを使いたいならば、1つのフェーズが次のフェーズに続くような場合、それは問題ありません。典型的なことです。ですが、アジャイルでは、ストーリー、ポイント、もしくはあなたのバックログがなんでもあれ、実際に作成される資料であり、次のステップに必要です。ですので、あなたはそれを持つことができ、各フェーズはスプリントの1つのようなものです。ですので、それに関しても問題ではありません。繰り返しますが、それをどのように機能するように構成するかです。アードバリュー、アードスケジュールに関する質問も来ています。私たちは、アードスケジュールに精通しています。実際、本の中でもそのことに関して言及しています。少なくともこのバージョンではアードスケジュールは使用していません。

それがそこにあり、一部の人がそれを使用していて、それから利益を得ている一方で、正式のアードバリューのアプローチではありません。そして、防衛のパートナーによって使われ、もしくは防衛省により確立されたものではありません。その一部ではありませんので、私たちはそれを取り入れていません。私たちは、典型的なアードバリューの一部であるCPIとSPIを使い続ける傾向があります。このスライドを使って、この本の目的は今日存在する多くのツールを駆動する基本的なエンジンに挑戦することです。そして、これは最初のスタートです。長時間にわたる議論を期待しています。そして、それはダイナミック・プログレス・メソッド最初のバージョンです。話をするための名前をつけるだけでした。私たちは本当に、みなさんがこれに加わり、このコンセプトがもう少し広がることを願っています。そして、この本はスタート地点です。本当にそれがそこから構築されることを望んでいます。そして、アードスケジュールや他に組み込むことができるようなものは、簡単に追加されるでしょう。

他の質問を引き続き見ていきましょう。生産性に関する幾つかの質問があります。生産性とそれをどのように測定するのか、そしてそれをどのように予測するのかについていくつかの質問が来ています。本ではもう少し詳しく説明していますが、それについて触れてみます。それは、生産性は、仕事量と同じくとても単純なものだということです。この人は、1時間当たり10フィートのケーブルを敷設し、他の人は1時間当たり20フィートのケーブルを敷設します。そこで、そのような仕事率になります。10本のコード対20本のコードもしくは一種のベンチマークとして、ある人を使ってもいいかもしれませんが、それは、良い、悪いということではなく、それらは一種のベンチワーク量で、ある条件においてある人の生産性が良いか、劣っているか言うだけで、もし、ベンチマークの人が1人で、他の人が2倍の生産性があるなら、生産性は、1に対して2です。または、彼らが半分の量を行うと、それは0.5です。あなたはそのいくつかを持っています。調節可能な疲労による影響の初

期曲線ができます。ですので、もし私たちがある人をロボットのように扱いたい、または、疲れの影響を受けない人員がいたら、それをツール内で設定して調整することは可能です。これら全てのことは、調節の余地があります。

引き続き他の質問を見させてください。いくつかは、少し域を超えていますが、マイクロソフトプロジェクトに関する詳細な質問をみっていますが、このウェビナーで紹介する内容を少し超えています。私は、実際に幾つかの例を持っていますので、オフラインで私に問い合わせいただければと思います。実際、私たちが使ったツールは、マイクロソフトプロジェクトをインポートでき、そのファイルのすべてがインポートされるので、再構築する必要がありません。私は実際、2つのガントチャートをお見せしました。1つは、DPMに基づいたガントチャートで、もう1つはクリティカル・パス・メソッドまたはマイクロソフトプロジェクトで表示されたものです。この2つの異なるガントチャートを比較してみて、どの点で長い時間がかかるか、短い時間で済むかなどを見てください。繰り返しますが、私は詳細について触れることができません。そのためのスライドを持っていませんし、そのために準備してきませんでした。ですが、それに関して、喜んでオフラインで話したいと思います。

いろいろなスケジュール分析とモンテカルロや、一種の確率的なスケジュールリスク分析に関する質問を見えています。違いは、それらのアプローチのエンジンは依然として期間ベースのエンジンであり、それに関係する不確実性と確立の中を通り抜けます。この仕事は通常10日間かかりますが、12日間かかるかもしれないし、7日間程度であるかもしれないということを取り込みますが、繰り返しますが、それは単に期間です。ですので、私たちがDPMで行っているのは実際にはリソースレベルのモンテカルロです。利用可能な人員は不確定なものであるか、もしくは人員の生産性は期間とは対照的に不確実なものでありますが、我々はそれを最後まで実行します。そして、本の第1章では、期間レベルでの不確実性とリソースレベルでの不確実性を示し、それらがどのように異なる結果をもたらすのかを比較します。

いただいている質問を見てもみると、管理アクションと、どのように管理アクションを定量化するかについていただいています。多くのアプローチがありますが、私たちがしたことは、表関数と言うもので、繰り返しますが、アード・バリュースを使います。ですので、それはスケジュール・パフォーマンス指標、SPI、もしくはコスト・パフォーマンス指標に基づいています。ですので、あなたが多量の変更は、もしSPIが1以下に落ち込んだ場合のグラフを作成することです。例えば、0.9になったら、私は少しスケジュールが遅れているので、全員に1時間残業してもらいます。ですが、もしそれが0.8または0.7に落ちた場合スケジュールよりさらに遅れていることを意味します。そこでポリシーとして、2時間残業しましょう、または3時間もあり得るかもしれません。そしてもしSPIが0.5以下のようにはるかに遅れていれば、誰もが5~6時間働いてもらうことになるかもしれません。

ですので、皆さんは、もしスケジュールより少し遅れていたらみんなに少し残業してもらうか、もしくは数人の人員を追加することができますが、ひどくスケジュールより遅延している場合は多くの時間もしくはたくさんの人員を追加します、と言うような提案のアプローチのようなものを適用することができます。しかし反対に、あなたは積極的になり、少しスケジュールを下回ったり、スケジュールが遅延するやいなやスケジュールを予定に戻すために全ての人を参加させ、全ての人を即座に6時間の残業をしてもらいたいということもできるでしょう。こ

れらは2つの異なったアプローチの例で、どのように、それらがとらえられるかです。繰り返しますが、これは、スライドにはありません。もう少し詳細が知りたければ、私に連絡してもらえれば喜んでそれらを共有します。いろいろな例をご紹介したいと思いますが、全ての質問に対するスライドを用意するのは難しいです。ですので、それらのいくつかに答えていたらうれしいです。

これらを見続けてみると、もう少し質問にお答えする時間がありそうです。いくつかお答えできるか確認させてください。何社がこれを使っているか、または、何人がこれを採用しているかについての質問をみています。一握りの、実行中の評価プロジェクトを持っています。このツールを実際に使っている幾つかのプロジェクトがあります。使っている規模は依然として小さく、私たちはまさに開始点にいます。ですが、それに関する本が発行されたので、世に広めもう少し皆様に知ってもらいたいと思っています。

質問を繰り返し見ていると、“証明（certification）”についての幾つかの質問がありました。スライドを見ていきましょう。少し戻って、そのスライドについて説明します。そして、これが意味していることを明確にさせてください。各レベルは、ある種、その上に構築しています。このケースで私が言おうとしていることは、レベル2にいるためには、レベル1を持っていなければならないということです。人員の生産性を高めるために人員を使用します。レベル3を行うために、他の2つを整えなければなりません。人員を使わなければなりません。何故なら、そうしなければ、管理アクションを始めることができなく、人員を使わないと、人員を変更できないからです。そしてレベル4では、管理アクションと生産性を持つことで、それによってこの2つを連結することができます。そして、それらはお互いの上に構築するのです。例えば、レベル2のスパイダープロジェクトチームは、私がお話ししたレベル1とレベル2で実際に行うことができ、そして、pmBLOXでは、全体のスペクトルをカバーできます。そしてなぜそれができるのかと言うと、DARPAの研究でこれを具体化するために私たちが開発したものなので、DPMのすべての要素が含まれているからです。しかし、他の幾つかのものがそれに関連しているところをお見せしたいと思います。

すでに触れられたと思う不確定に関する幾つかの質問を見ました。クリティカルチェーンに関する幾つかの質問があります。この本の中で議論しているクリティカルチェーンは、正しい方向への第一歩です。私たちは、それが人員に少し焦点を当てて、しかもプロジェクトにおけるクリティカルとは何かによりますが、基本的にはクリティカル・パス・メソッドに基づく期間ベースのアプローチであることが好きです。しかし、それは依然として期間ベースのアプローチであるか、または期間ベースのアプローチ上に構築されているため、私たちの観点から見ると、人員または生産性に起因した種類のエンジンユニットとは対照的に、期間ベースエンジンと呼ばれます。それに関してで、終わりにしようと思います。DPMとプロジェクトマネジメント体系の知識を取り入れるにはどのように努力したらいいか、という最後の質問にお答えします。

私たちは最終的にはそれに到達します。それが私たちの目標です。繰り返しますが、この本の目的は、会話を開始し、この期間ベースのアプローチが持つ全てを壊し始めることです。電気自動車会社、テスラについて考えていきたいと思っています。テスラは電気自動車ができると思うものに革命を起こしました。テスラ以前は、それはとても遅くなるだろう、範囲は限られ、退屈なものになると予想されていました。そして、テスラが現れまし

た。早くすることができるし、カッコよく、スポーティにすることができ、実際ガス駆動車に非常に近い範囲で作ることができると言いました。そして、テスラは私たちが電気自動車ができること、そうであるべきだと思っているものを打ち破りました。そして、それが私たちがDPMでしようとしていることであり、このようなより現実的なアプローチを構築しようと話しています。それが私たちのここでの目標です。

本日はご参加いただきありがとうございます、みなさんお気軽にご連絡ください。これが、Kindleで使えるか、ダイナミック・プログレス・メソッドに関する本はアマゾンで購入できるのかに関して質問いただいております。皆さんが見ているページは出版社のページですが、アマゾンにアクセスしてみると、Kindle版もあります。皆さんのフィードバックを聞くことを願っております。是非聞かせてください。そして会話を始めたいと思います。会話は既に始まっています。私たちは、これらのことを話す最初の人ではありません。わかっています。今日の複雑なプロジェクトに必要な現実感をキャプチャするために必要なプロジェクトを攻撃する根本的に異なる方法として、プロジェクトマネジメント界の中で真剣に検討するために、より公式的なパッケージを取り入れようとしています。皆様本日はご参加下さり、ありがとうございました。繰り返しますが、もし質問がありましたらご連絡ください。デレック、マイクをお返しします。

HOST:皆様、今日のウェビナーに参加いただきまして、ありがとうございます。J・クリス・ホワイトさん今日の素晴らしいプレゼンテーションをありがとうございます。お知らせですが、このウェビナーは、24時間以内にprojectmanagement.com上で閲覧可能になります。また次回お会いしましょう。