NUPP

Nearly Universal Principles of Projects

This is a downloadable version of the online manual (https://nupp.guide), generated on 2025-06-02. Please check the website for newer versions.

This manual can be used and distributed freely under the Creative Commons Attribution 4.0 International license.

翻訳者: Takashi Izumoto

NUP1

所属よりも結果と真実を優先すること

私たちは皆、集団に属そうとする自然な傾向を持っています。その傾向はしばしば本来あるべき姿を超えて、強い所属関係を生み出し、そして問題を引き起こします。所属のせいで、得るものよりも失うものの方が多いこともあります。もし自分のアイデンティティや好みを特定のグループに限定しなければ、私たちはよりプロフェッショナルで能力の高い専門家になることができるでしょう。

例: アジャイル対ウォーターフォール

予測型アプローチが主流だった当時、IT開発において適応型開発アプローチを試みる勇気を持った、非常に熱心な人々が集まり、自分たちのそのアプローチを「アジャイル」と呼びました。これは、選択肢をいかにも必要そうに見えるものだけに限定しない、素晴らしい取り組みでした。アジャイルコミュニティには、今でも熱心で結果を重視する人たちがたくさん集まっていますが、残念なことに、このコミュニティの中には、アジャイルをカルト化させ、部外者をすべて敵とみなすような人たちも存在します。これは、以下のようないくつかの点で問題を引き起こします：

- 自分たちのグループ外の誰からも学ぼうとしない
- グループ外の人が彼らから学ぼうとするのを妨げる
- グループに属している自体を、その本来の目的よりも重要視するようになり、その結果、メンバーの多くが機敏さ（Agility）の本当の意味を学べなくなっている

この問題を解消するには、「アジャイル」を、メンバーの集まったコミュニティとしてではなく、開発アプローチを指す1つのラベルとしてのみ使用し、自分はクリエイター、問題解決者、リーダーであると考えている人たちが、アジャイルを自分たちのアイデンティティとしてではなく、単に自分が活用

できるイナブラー（道具・手段）の1つとして捉えるようになることが必要です。

真のプロフェッショナルにとっては、アジャイル対ウォーターフォールの戦いなどは存在しない。

例: PRINCE2® vs PMBOK®ガイド

コミュニティの中には、PRINCE2®あるいはPMBOK®ガイドのどちらかに属し（たいていは地理的な理由から）、もう一方のことをよく知らないプロフェッショナルがたくさんいます。私たちは皆、特定のリソースの方をより好むということはあっても、それを自分のアイデンティティとするのはよくありません。重要なことは、視野と選択肢を広げるために、すべてのリソースに精通しておくことです。

真のプロフェッショナルとは、あらゆるアイデアにオープンであり、それらを探し、学び、必要なときに必要なだけ、所属にこだわらずに利用するものです。

例: 継続的な学習

所属は、そのことが作り出す帰属意識によって人を満足させはしますが、学ぶことを後押しすることはなく、逆にその帰属意識の強さによって新たなことを学ぶことの妨げとなることさえあります。特別なグループに強い所属をしないエキスパートであるならば、自分に不足しているものを継続的な学習によって埋める続けることでしょう。

私たちが今日信じていることは真実とは限りません。それは、私たちの現時点での最良の理解に過ぎず、先へ進むにつれて改良しつづける必要があります。自分の考えが数年前とまったく同じというのは、何かがおかしいと考えた方がいいでしょう。このNUPPの場合でも同じです。数年後に戻ってきて、まったく同じものを目にしたら、疑ったほうがいいでしょう。

例: オープンネス（積極的に受け入れる姿勢）

誰かに異議を唱えるときは、その人に対してではなく、そのアイデアに異議

を唱えていることが重要です。こうすることで多くの緊張を防ぐことができるでしょう。同じように、誰かがあなたに対して、あるいはあなたのことについて異論を唱えているときは、それをあなたに対しての戦いと解釈せずに、あなたの考えについての議論であると解釈して、それに対してオープンであるよう心がけましょう。相手の話を聞く時は、返答するために聞くのではなく、相手を理解するために聞き、その相手とアイデアをより良くするために協業すると考えましょう。

時には、意図的にあなたのアイデアでなくあなた自身をターゲットにしてくる人もいるでしょう。その場合は、話を進める前に、相手が自分ではなくアイデアに集中できるように促し、会話中もその状態を保つように心がけましょう。

NUP2

エネルギーと資源を維持・最適化すること

リソースには限りがあります。プロジェクトで利用できるリソースには限りがあり、あなたが適切な決断を下すために必要な精神的エネルギーにも同じく限りがあります。だからこそ、自分とプロジェクトのために、あなたはそのリソースを温存しながら、最適に活用しなければならないですし、チームメンバーにもそのように促すべきです。

例: 80/20の法則

プロジェクトマネジメントから得られるベネフィットの大部分は、わずかな労力を費やすことで得ることができます。ほとんどの場合、可能性のあるベネフィットの100%を得ることを目標とすることは、非常に高くつき、正しいことではないでしょう。あなたが行うすべての行動においてこのルールを適用できないかを考え、他の人にも同じようにするよう勧めるとよいでしょう。

例: 決断疲れ

私たちはあらゆる種類の決断を下すために、また意志の強さを表明するために、ある種の精神的エネルギー源を使います。もし必要ではない、あるいは重要でない決断をするためにこのエネルギー源をたくさん使い切ってしまうと、本当に重要な決断をするためのエネルギーが枯渇し、悪い結果を招く可能性があります。これが、マイクロマネジメントを避けるべき理由のひとつです。（PRINCE2®の「例外による管理」の原則）

アイデアにおける対立というのはプロジェクトの為になり得ますが、人間関

係に関する対立はたいていの場合プロジェクトにとって有害となります。そのような対立を目にしたら、すばやく根本的な原因を見つけ、解決するために最善を尽くすべきです。

例: 自分を大切にする!

あなたが下す決断や意志の強さは、あなたの精神的エネルギーの源を多く使います。一方、睡眠や食事がきちんと摂ることで、このエネルギー源は満たされます。だから、十分な睡眠と休息をとり、よく食べるなど、自分自身を大切にする必要があります。もしあなたが有害な習慣や睡眠に問題を抱えている場合、一人で対処する必要はありません。そのような問題を解決してくれる専門家はたくさんいます。運動することが助けになるかもしれませんが、これに関しては研究でこれといった結論は未だ出ていません。

チームのメンバーにも、あなたと同じことをするように勧めてみてください。まず、彼らが持続可能なペースで、あまり残業せずに働くようにし、もし可能なのであれば、勤務時間中にエネルギッシュな健康によい食べ物、スナック、飲み物を提供するとよいでしょう

例: ワークライフバランス

私たちの多くは自分の仕事を好きかもしれませんが、仕事が人生で必要なもの全てではありません。仕事を最適化するのと同じように、私生活においても、楽しく幸せなものになるようなアイデアを計画し、実行すべきです。あなたがより幸せであれば、仕事でもより成功することができるのです。できれば、チームのメンバーにも同じようにするよう勧めてみてください。

例: モチベーション

モチベーションとは複雑な概念です。このトピックに関する興味深く有用な文献等もあれば、非科学的な情報もたくさんあります。とはいえ、モチベーションについては、継続的に学び、活用していくすべきです。なぜなら、モチベーションはチームの精神的エネルギーを高め、プロジェクトにより良い結果をもたらす可能性を高めるからです。

微笑みかけたり「ありがとう」とたった一言声をかけたり、ちょっと相手の

良い仕事ぶりを認めることで、モチベーションを高めることができます。ただし、少額の金銭的報酬といった、よくあるモチベーション向上策に関しては、マイナスの効果もあるので、注意が必要です。

例: 共同作業とチームワーク

共同作業することで、人はより良い結果を生み出す力を持つのかもしれませんが、それ以上に重要なのは、そもそも人間は社交的な生き物であり、グループの一員であることを楽しむものだということです。チームワークのマイナス面を取り除き、快適な環境を作ることができれば、プロジェクトに参加するチームメンバーはよりハッピーになるでしょう。

とはいえ、人はそれぞれ異なるものです。よりリラックスし、集中して、独りの時間を必要とする人もいれば、そうでない人もいるでしょう。これについてはバランスを取ることが求められます。

例: ワンチームカルチャー

様々な利害関係者たちが小さなグループを作り他のグループと緊張を引き起こすことがあります。例えば、プロジェクトのビジネス面を重視する人たちと、製品を作る人たちとの対立。このような緊張は、双方のエネルギーを大量に消費してしまいます。こういったことを避けようという試みの1つが、全員が同じ目標に向かって協力し合う、ワンチームカルチャーを築こうという活動です。

例: 群衆の叡智

多様性のある多くの人々が集まり、円滑な環境で仕事をすれば、非常に優れた結果やアイデア、解決策を得られる可能性があり、それは一人の専門家のそれよりずっと優れているかもしれません。

もしそのような環境を選べるのであれば、プロジェクトの難しい問題を解決するために、その多様性のあるチームメンバーたちに助けを求めることができるでしょう。これには、良い答え・結果が得られるかもしれないというだけではなく、チームメンバーは自分の意見を聞かれ、感謝をされ、プロジェクトで重要な役割を果たしていると感じることができるという良い効果もあ

ります。このことはチームのエネルギーレベルを高めることに繋がります。P3.expressのアクティビティE02は、プロジェクトにおける群衆の智慧の活用例の1つです。

例: チーフ・プロジェクト・ファシリテーター

もしあなたがプロジェクトマネージャーなら、あなたが行うことのほとんどはファシリテーション的な性質を持っているでしょう(少なくとも、そうであるべきです)。その一方で、チームメンバーが過去にプロジェクトマネジャーに嫌な思いをさせられた経験があり、その経験があなたとの関係に影響を与えていることを知る場合があるかもしれません。その時、チームメンバーのエネルギーの一部は、あなたを信頼する代わりに、潜在的な脅威がないかどうかあなたの行動を分析することに費やされているのです。こういった場合、肩書きをプロジェクトマネージャーからチーフ・プロジェクト・ファシリテーターに変えてもいいでしょう。結局のところ、それがプロジェクトにおけるあなたの本当の仕事なのだから。

NUP3

常にプロアクティブであること

私たちは生来、問題が起きてから対応する（＝リアクティブな）傾向があります。それは、重要でない事に対処する際にエネルギーをあまり使わないのに役立つし、自分がまったく能力を持ち合わしていない事に対処している場合には、この方がより良い結果をもたらすかもしれません。ただし、それはプロジェクトには当てはまりません。プロジェクトにおいては先を見越して積極的（＝プロアクティブ）に行動することでより良い結果を得ることができます。

例: 計画

もし新しい場所に車で行きたいのに遅刻している場合、「時間を節約する」ためにすぐさま車に乗り込み出発し、何か問題が発生したときにそれに対処するでしょう。一方、プロアクティブなアプローチでは、まず最初に時間をかけ、交通状況や起こりうる事故や通行止めを考慮した上で、最短ルートで行けるようにナビゲーション・システムをセットし、それから車を出発させます。そうすることで、後に起こり得る問題を避けて、結果として時間を節約することができます。

一部の人がアジャイルプロジェクトについて抱く印象とは対照的に、計画は常に必要であり、アジャイルとは計画の種類と詳細の度合いのことを言っているに過ぎません。プロアクティブアプローチとは、実行する前に計画を立てることです。

この諺を胸に刻んでおきましょう。「木を切り倒すのに6時間くれたら、最初の4時間は斧を研ぐのに費やすだろう」

予測型のプロジェクトであるならば、もしあなたが木を切り倒すことに確信を持っているならば、斧を研ぐのに4時間を費やすかもしれない。アジャイル

プロジェクトであるならば、木を切り倒すのか、折れた枝を集めるのか、芝を刈り取るのか、石炭を採掘するのか、それとも他のことをするのか、まだよくわからない。とはいえ、それら全てについて大まかな準備（最寄りの金物店の場所を知ること）をしておくこと必要であり、特定の解決策に集中する段階になった時には、具体的な準備（斧を研ぐ）が必要となります。これが計画を立てるということなのです

例: 計画の計画を立てる

プロジェクトをどのように実行するかを計画することは、プロアクティブなアプローチとなります。さらには、実行の計画をどのように計画するかで、よりプロアクティブとなります。PMBOK®ガイドのマネジメントプラン、PRINCE2®のマネジメントストラテジー、DSDM®のアプローチなどが、まさにこれにあたります。

例: 継続的な計画

現実が計画と一致することは稀であり、それは構わないことです。ただし、計画が現実的かつ実践的であり続けるように、継続的に計画を適応させていかなければなりません。それは、問題に直面してからではなく、その修正が必要と分かった段階ですぐに行われるすべきです。それがプロアクティブなアプローチです。

例: リスク管理

不確実な出来事に直面したとき、何が起こるか様子を見てから対応するのではなく、発生の可能性と影響度合いを考え、対応を検討し、可能なものはそれが起こる前に何かしらの策を講じる。このリスクマネジメントの概念こそが、プロアクティブであることに基づいているのです。

私たちがプロジェクトで行うことは重大なものであり、時として人の命に関わることがあることに留意しましょう。

例: 役割と責任を明確にする

役割と責任を明確にしないままプロジェクトチームのメンバーに仕事を任せても、遅かれ早かれそれぞれの役割と責任の形づいてきます。ただし、このやり方は高くつき、結局うまくいかなくなることが多いでしょう。プロアクティブなアプローチでは、早い段階でそれら役割と責任を定義し、必要に応じて変化させていきます。そうすることで、誰もが働きやすくなり、誰が何をするかを決めるために時間とエネルギーを浪費する代わりに、何かを生み出すことに集中できるようになります。

役割の数や種類は、プロジェクトのタイプや規模によります。スクラムにおけるシンプルな定義もあれば、P3.expressのようなほどほどの定義、あるいはDSDM®やPRINCE2®のような包括的な定義もあります。ただし、これらの手法における役割の説明は、あくまでもマネジメント活動に関するものです。よって、テクニカルな側面についても、常に役割の説明を加える必要があることを忘れないでください。

例: 利用可能な選択肢

プロジェクトを早期に終了すべきか、継続すべきか。

たとえそのような質問であっても、選択肢が本当に2つしかないというケースはほとんどありません。プロアクティブなアプローチをとり、決断を下す前にはすべての選択肢を検討する必要があります。もしかしたら、プロジェクトのスコープを調整できるかもしれないし、何か別のことが明らかになるまで一時中断するのがよいかもしれない、あるいはプロジェクトのアプローチを変えられるかもしれない（アウトソーシングなど）、など。

例: クリティカルシンキング

私たちは皆、たくさんの偏見を持っていて、それは一方では生き残るために役に立ち、また一方では私たちを惑わし誤った決断を引き起こします。プロジェクトに関する重要な決断をするときは、しばらく立ち止まって、問題を引き起こす前に、決断に影響を与え得るあらゆるバイアスの注意を払うのが最善です。

参考文献として、ウィキペディアの認知バイアスのリスト(英語のみ)
https://en.wikipedia.org/wiki/List_of_cognitive_biases

より良い決断をするために使える意思決定のフレームワークというものがあります。最初のうちは、それらを使うことに気を取られ、煩わしささえ感じるかもしれませんが、すぐに慣れ、あまり意識することなく、うまく活用できるようになるでしょう。

例: 透明性

私たちは、プロジェクトが遅れたり、その他の問題を抱えたりすることを好まないでしょう。だからといって、それを隠すべきではありません。透明性を保つべきです。なぜなら、ステークホルダーの中にはあなたを助けてくれる人もいるかもしれません。また、遅かれ早かれステークホルダーはその問題とその結果について知ることになり、中には早期の行動（例えば、ネガティブな結果を受け入れること）を取る必要がある人もいるからです。

例: 効果的なコミュニケーション

ステークホルダーにレポートを送っても、何もフィードバックが返ってこないというケースが多くあるかもしれません。あなたは、否定的なフィードバックが返ってこないから、全てがうまくいっていると思うかもしれませんが、そうとも限りません。ステークホルダーが本当のレポートを読んでいるかをプロアクティブに確認し、必要であればコミュニケーションのやり方を調整する必要があるでしょう。さもなければ、隠れた問題が後々になって深刻な問題を引き起こし、もうその時点では解決が困難となる可能性があります。

例: 責任を取る

悪い結果を他人のせいにするのは簡単でしょう。例えば、組織から全権を与えられてプロジェクトのすべてを自由に変更し、完璧にやりたい。しかし、そのような権限は与えられず、その結果プロジェクトが失敗してしまったとします。これはプロアクティブなアプローチではありません。

プロアクティブなアプローチとは、責任を負い、制約の中でできる限りの全てをすることです。組織があなたを全面的に信頼し、良い結果を期待して、あなたにすべてを委ねるということは期待できないでしょう。特にその組織

がこれまで多くのプロジェクトの失敗を見てきたならば尚更です。あなたがやらなければならないのは、与えられた制約の中でまずは小さな改善を行い、その結果少しの信頼を掴み取り、リソースを少し増やし、制約に対する寛容さも少し増やし、そこからもう少し大きな改善を行い、最適な目標に達するまでそれを続けることなのです。

NUP4

鎖の強さは一番弱いつなぎ目で決まることを忘れないこと

プロジェクトにはさまざまな領域があり、すべてに注意を払う必要があります。私たちはプロジェクトの全体的視点を持たなければなりません。一見重要そうに見えるドメイン（例えば時間）にだけ注意を払うのでは十分ではありません。なぜなら、すべての領域が相互に影響し合い、すべてに十分な注意が払われない限り、適切に機能しないからです。

例: 締め切りがすべて!

オリンピックに向けて何かを作っているとしましょう。非常に重大な締め切りがあるため、時間管理が非常に重要になります。しかし、それだけでいいのでしょうか？　品質が低すぎて、しばらくしたらまた作業を繰り返さなければならなくなったらどうなるでしょう。きっと時間にも影響が出ます。そうなると、時間と品質の問題になります。当初のプロジェクト定義には派手な庭が含まれていたとしても、もし時間が足りなくなった場合、それを省略して芝生で覆うこともできるでしょう。もし事前にその選択肢の可能性を予見し準備をしておいたならば。こうなると、スコープ管理も重要になります。結果、私たちはスコープ、時間、品質の3つの領域に注意を向けて考えることになります。

ケネディ大統領がNASAの清掃員に会い、何をしているのかと尋ねると、彼は「月に人を乗せるのを手伝っているんだ」と答えたという有名な逸話を耳にしたことがあるでしょうか。プロジェクトにそのような人材がいれば、期限を守ることに大いに寄与するのではないでしょうか。

突き詰めると、プロジェクトのあらゆる領域が時間管理に寄与していることに気づくでしょう。ただし、すべての領域に注意を払わない限り、許容可能

なレベルで確実に期限を守ることはできません。

例: チェリーピッキング

私たちは、さまざまな手法に目の前にしたとき、時にチェリーピッキングを始め、様々なシステムから良さそうなものを取り出しすべてをミックスしてしまうことがあります。これは、大抵うまくいきません。なぜなら、要素は単独では機能せず、互いに相性が合わなければなければならないからです。ですので、システムにおける追加や変更を行う場合は、全体的な視点から行われるべきなのです。

私たちは時々異なる手法において相反する要素を目撃します。あるシステムではうまく機能するのに、別のシステムではその正反対のものがうまく機能する。そういった要素は、それ自体が正しいとか間違っているとかいうものではありません。

最も安全なアプローチは、プロジェクトのために方法論を選択し、それを調整し、そしてシステム全体の一貫性を考慮しながら、慎重に新しい要素をそれに加えることです。

例: アンチプロセスアプローチ

アジャイルメソッドが行ってきた最も優れたことの1つは、人間的側面に注意を引いたことです。これは公平な比較ではないかもしれないかもしれませんが、アジャイルマニフェストは、プロセスやツールなどに比べると、個人と意思疎通に重きをおいています。ほとんどすべての手法も、人間的側面が重要であるとは言ってはいますが、アジャイルメソッドが他と本当に違う点は、人間的側面をそれ単体の話としてではなく、プロセスに一部に組み込まれているところです。つまり、人間的側面なのか、それともプロセスなのかと対峙するものではなく、システムの中における人間的側面をどう捉えるかという問題なのです。

より洗練されたプロセスを持つことで人間的側面を置き換えようとする人は間違いなくいるでしょうが、それは間違った使い方です。その逆を主張する人さえいます。プロセスを人間的側面に置き換えようとする人。こちらも上手くはいきません。

例: 必要な全ての領域

領域について考えるとき、どの領域も見逃さないように注意する必要がある。とはいえ、ではそれらは何なのか？いろいろと調べてみると、さまざまな異なる答えが返ってきます。そしてどれも真実ではありません。

PRINCE2®ではテーマが領域ですが、それはあくまでこの方法論において重要な役割を果たす領域に過ぎません。他の領域は、暗に示されているくらいです。

PMBOK®ガイドは方法論ではないですが、10の知識エリアを使ってもっとうまく系統化することができます。しかし、これらは中立的な視点ではなく、プロジェクトに対するPMBOK®ガイドの視点に基づいた全領域の解釈です（必ずしも中立的な視点が必要というわけではありませんが）。例えば、人的側面はPMBOK®ガイドではあまり注目されていません。

領域に関する良い情報源の1つはICB (Individual Competence Baseline)です。（※ICBは、IPMA：International Project Management Associationの提唱している基準です）しかし、それは領域というよりは、プロジェクトで必要とされるコンピテンシーに関するものです。そのコンピテンシーは領域と1対1の関係ではありませんが、ドメインを特定する上では大いに役立ちます。

NUPPに領域のリストがないのは、それがシステムというよりメタシステムであることが主な理由であり、また領域の分類がプロジェクトのタイプや環境によって異なるからです。（たとえば、ありふれた建設プロジェクトは、創造的なリサーチプロジェクトとは、異なる視点が必要となるでしょう）

NUP5

明確な目的のない限り、何もしないこと

明確な目的がない限り、何もすべきではありません。2つのパラレルワールド（平行世界）をイメージしてみてください。そこでは、これからあなたがやろうと考えていること以外はすべて同じです。あなたがそれを行うことで、その2つの世界の間には、どのくらいの違いが生まれるだろうか？生まれるその違いは、その努力に見合うだけの価値があるだろうか？

もしあなたが明確な目的を持っておらず、ただみんながやっているから、あるいはみんなが重要だと言うからという理由だけでやっているとしたら、それは、

- あなたにとっては本当の利益・恩恵はなく、それゆえに何も得るものはないかもしれない；あるいは、
- あなたにとって潜在的な利益があるかもしれないが、あなたがその目的を念頭に置いていないため、あなたの今のやり方では利益・恩恵を受けることができないかもしれない

例: ポートフォリオとプログラム

プロジェクトの選定や始動に携わるのであれば、プロダクトではなく、そのプロダクトが生み出すベネフィット（利益・恩恵）や解決すべき問題に注力すべきです。よく言われる例として、エレベーターの製造会社で、エレベーターのスピードに関する苦情を受けていたため、スピードを上げるためのプロジェクトを何度も実施したが、成功には至らなかった。その試みは、“すぐに思いつく“解決策（エレベーターの高速化）ではなく、問題（人々の退屈や不快感）に焦点を当てるべきだと気づくまで続いた。結果として、エレベーターに鏡を追加し、シンプルかつ安価に問題を解決したのでした。

プロジェクトマネジメントとは主に物事を「正しく行うこと」であり、ポートフォリオマネジメントとは「正しいことを行うこと」というのを忘れないでください。プロジェクトをいくらうまく運営しても、間違ったプロジェクトをやっていてはうまくいきません。すべては目的を持っているかどうかなのです。

例: プロジェクト全体

プロダクトの柔軟性はプロジェクトによって異なります。あるIT開発プロジェクトでは、プロダクトにとても柔軟性をもたせ、その最終形はプロジェクトを進めながら段階的に作っていくプロダクトに対するフィードバックに依存するため、適応型（アジャイル）のアプローチが必要となるでしょう。これは実務的な言い方をすればポートフォリオ、プログラム、プロジェクトのレイヤーの組み合わせであり、最終ゴールに最大限の注意を払う必要があります。目的を文書化し、みながアクセスできるようにしておくのは良いアイデアです。これは、スクラムプロジェクトで使用されている「プロダクトビジョン」の目的の1つです。アジャイルプロジェクトでビジネス価値に注目するというのは、目的との整合性を確保するための方法なのです。

他のタイプのプロジェクトでは、プロダクトが比較的固定化されていて、そのプロダクトが目的を満たすことを確かなものにするメカニズムがあります。そこでは、プロジェクトチームのメンバーのフォーカスの多くを目的からプロダクトに移すことができます（これが、PRINCE2®の「プロダクトにフォーカスする」原則）。しかし、最小限の注意は目的にも向けられ、作っているプロダクトが目的を満たすことを確かめることは求められます。これは、期待されるベネフィットと予想されるベネフィットとを比較することで行うことができます（これが、PRINCE2®の「ビジネス正当性の継続」原則と、「ビジネスケース」テーマ）

プロジェクトが社外のクライアントのために実行される場合、クライアントにはクライアントのプロジェクトの目的があり、あなたの会社にはあなたの会社の目的があります。本当の意味で成功するためには、この2つを理解し、使い分ける必要があります。

例: プロジェクトのモニタリング

最終的な目的に焦点を当てることは必要ですが、日々の活動レベルで使うには抽象的すぎるかもしれない。そのため、より実用的にするためには階層が必要となります。まず、最終的な目的に基づいてプロジェクトの正当性とベネフィットを定義し、正当性を満たすためのプロジェクト変数（時間、コスト、品質など）のターゲットを明示的、暗黙的に設定します。これらは抽象度が低く、日々のタスクレベルで有用なものとなります。

モニタリングに関しては、最終的な目的に対してのトラッキングは難しいため、プロジェクトの最も抽象度の低い変数を用いて行います。この場合でも、その目的を念頭に置くべきです：モニタリングすることの目的は何だろうか？

通常の受け入れられる答えとしては、計画通り進んでいるかどうかを確認し、もしそうでなければ、計画に戻すためのあるアクションをとるか、あるいは目標を調整しつつ、それでも最終的な目的を満たすことができるかどうかを確認するといったことでしょう。したがって、私たちの測定することは、この低いレベルの目的において役に立つべきであり、適切な測定というものは、完了した時点での変数の予測値を示すものです。例えば、いつプロジェクトを終えることができるだろうか？プロジェクトを完了させるために必要な資金は？

これまでの計画と実績の今日までの値 など、その他の測定値はすべて、予測を計算するために必要な中間値に過ぎず、経営的な目的で使用する最終値ではありません。

例: ドキュメント

プロジェクトでどのような開発手法を使うにせよ、計画は常に必要です。重要なのは、それぞれの計画の詳細レベルです。計画に十分な詳細さがなければ、その計画は十分な役割を果たすことができず、プロジェクトの実行は、統合された全体的なものではなく、場当たり的な決定に基づいて行われることになります。その一方で、多くの人は慎重になろうとするあまり、計画に細部すぎるものを加えしまいます。そうなると、計画の作成とその維持が難しくなり、プロジェクトの不確実性に対して硬直化しすぎてしまいます。その結果、計画は非現実的で役に立たないものになってしまいます。

詳細レベルを決める最善の方法は、計画の1つ1つの要素についての目的を念頭に置くことである。例えば、計画にリソースの追加を検討しているのであ

れば、その目的があるはずです。そのリソースをどのように使うのか？プロジェクトにどう役立つのか？どれだけの労力がかかるのか、そして、その価値はあるのか、など。

ある要素を計画に盛り込むかどうかを決めることもあれば、何かを計画したり準備したりする方法について決めることもあります。ビジネスケースであれ、プロジェクト憲章であれ、報告書であれ、なぜその文書を作成するのか、そしてそれがプロジェクトにどのように役立つのかを自問すべきなのです。

「テンプレート」を探すという行為は、目的に基づいて何かをする行為とは正反対のものです。

例: ステータス報告

多くのプロジェクトで、本当に長いステータスレポートを作成するのをよく見かけます。このNUPに基づき、他の多くの人が何をしているかに関係なく、私たちは、「その報告の目的は何か」、「その目的を達成するにはどうすればよいか」を自問すべきでしょう。

その結果、多くの場合、長い報告書ではなく、ステークホルダーが本当に読んで理解できるような、非常にシンプルな1ページの報告書を作成することにつながるでしょう。これこそが、目的に注意を払うということです。

とはいえ、1ページの報告書を作成すると、それに反対し「適切な」モニタリング・システムを導入できていないと考える人も出てくるでしょう。現実的には、こうなると（ステーホルダーにプロジェクトの状況を理解してもらうという第一の目的以外に）あなたにとって第二の目的が生まれることになり、その目的を満たすためには、長い第二のタイプの報告書を作成すればよいのです。しかし、この2つの目的は同じではないので、最初のレポートとミックスさせることはありません。

例: ビジネスケースとプロジェクト憲章

ビジネスケースやプロジェクト憲章のような文書を作成することは、ほとんどの人にとって退屈で、実りのない、官僚的な作業でしょう。ただし、これらの文書には、ほとんどのプロジェクトに通じる正当な目的があります。も

し「テンプレート」を見つけてそれを埋めようというなら、その作業は時間の無駄です。その一方で、それらの文章の真の目的を知り、あなたのプロジェクトに、どのように役立つかどうかを確認し、その目的を満たすというだけのために、好きなやり方で文書を準備するならば、それこそが正しい文書となります。

目的を満たすために、どのような文書を作成すればよいかを考えているうちに、すべてのシナリオを思い浮かべることができず、何かを見逃してしまうかもしれません。それを避けるためには、PRINCE2®、PMBOK®ガイド、P3.express、DSDM®などのリソースが示しているものを、目的を念頭におきながら参考にしてもよいでしょう。

例: ポストプロジェクト

プロジェクトには具体的な目的があり、プロジェクトを通じてその目的を考慮しますが、目的の実現は主にプロジェクト中の予測に基づいて評価されます。しかし、プロジェクトが終了したときも、目的を忘れてはならなりません。プロジェクト終了後にゴールを達成したかを確認することは重要である。なぜならば、

- 最終的なゴールがどのように達成されたかを確認することで、今後のプロジェクトに活かすことができる; また、
- 時々プロジェクト完了後に追加作業を行うことで初めて目標が達成されることもあり、その追加作業の必要性を理解するためにもモニタリングは必要です

ポストプロジェクトのモニタリングは、目的志向型であるためには必要なステップです。これは主にプログラムやポートフォリオマネジメントシステムの責任であり、ほとんどの企業にはそのようなマネジメント階層がないため、このステップはたいてい無視されます。そのため、P3.expressやDSDM®のような手法では、このステップをプロジェクトマネジメントの方法論の中に追加しているのです。

例: シンプルさ

世界は複雑で混沌としていますが、私たちのモデルは世界の一部を反映した

抽象的な近似であるため、単純であることができます。一方、シンプルなシステムの方が、中心的な考えに集中できるため、たいていはうまくいきます

私たちの多くは、システムに複雑さを加えることでより良い結果を得ようとし、それが世界の複雑さと一致することを望んでいます。実際には、これはシステムがうまく機能することを妨げ、たいてい目的を達成することを遠ざけてしまうことが多いです。

例: テーラリング

音楽をストリーミングするためのソフトウェアと、人の命がかかっているような病院や飛行機の設備に使われるソフトウェア、何年もクラッシュすることなく動くことが前提とされる人工衛星に使われるソフトウェアとでは、その状況はまったく異なります。また、別荘や消防署、製造工場を建設するのとも異なります。

目的を常に念頭に置いていれば、さまざまな異なるプロジェクトにシステムや作成物を合わせる方法をよく理解できるようになるでしょう。

NUP6

反復可能な要素を活用すること

プロジェクトに対して場当たり的なアプローチをとると、それにエネルギーとリソースを多く費やしてしまい、本当に必要な要素が欠けてしまう危険性が常にあります。やるべきことを単純化する最善の方法は、反復可能な要素を使うことであり、できれば反復可能なサイクルで行うことです。

例: 品質チェックリスト

チェックリストは、反復可能な要素のシンプルな例であり、多くの人が私生活や仕事で使用している。例えば、成果物の品質基準を考えてみましょう：

- まず、すべての基準のチェックリストを作成する
- NUP6が推奨するのは、それを一般化してみることです。プロジェクトにおいて類似した成果物は他にないだろうか。ある場合、そのカテゴリーの成果物の一般的な品質チェックリストを作成し、すべての成果物に使用する。いくつかのバリエーションがある場合は、一般的なリストはそのままにして、個々の成果物用にいくつかの項目を追加します。これで、繰り返し使えるチェックリストができました
- さまざまな種類の成果物について汎用的なチェックリストを作成したら、それらの間で繰り返される要素が見つかるかもしれません。それは実質的にそれらの要素の親カテゴリーがあることを示しています。その場合、すべての汎用チェックリストの項目を繰り返すのではなく、それらを抽出して親チェックリストに入れることができます。最終的には、プロジェクト全体に対して1つの汎用チェックリストができるでしょう。スクラムの“完了の定義 “は、プロジェクトレベルのとりわけ品質に対するチェックリストを使う例です。こうすることで、各成果物はカテゴリの階層に属し、その連鎖の中ですべてのカテゴリのチェックリストに現れる項目を満たす必要があります

こうすることで、親チェックリストの項目は、その下にあるすべての成果物に対して繰り返し使用できるようになり、計画と実行の時間とエネルギーを節約できます。

さらに重要なことは、1つのプロジェクトでこれを行えば、将来、類似したプロジェクトすべてにこれを調整しながら利用することができ、これは複数のプロジェクトで繰り返し使える計画のフォームとなります。

例: プロセスとワークフロー

成果物や、それに関連する目標の中には、標準化され、反復可能な特定のステップが必要なものがあります。例えば、成果物を個別に設計し、承認する必要がある場合、すべてのステップ、関係者、おおよその期間を明確にするシンプルなワークフローを準備することで、多くの面倒なことを避けることができる。ただし、ワークフローやプロセスを複雑にしすぎたり、文書化しすぎたりすると、かえってマイナスになるので注意が必要です。プロジェクトに関わるすべての人にとって、ワークフローやプロセスは、彼らの実際の仕事を妨げる官僚的な文書ではなく、彼らの仕事をサポートし、すべてを簡単にするものと見なされるべきです。

アジャイルプロジェクトでは、反復型開発アプローチで繰り返される要素があり、あるタイプの開発活動は、それぞれの特性（フィーチャー）で繰り返されます。例えば、XP（エクストリーム・プログラミング）の一般的な日課である、ペアを組み、アイテムを選び、ホワイトボードで設計し、テストスクリプトとコードを構築し、コードを統合する、などがそうです。

技術的な活動に使用できる繰り返しのワークフローの他に、プロジェクトマネジメント活動にも繰り返し可能な要素を持つことができます。PMBOK®ガイド、PRINCE2®、DSDM®のプロセス、P3.expressのアクティビティ、スクラムのイベントは、このコンセプトの例です。

例: サイクル

プロジェクトを管理するための繰り返し可能な要素を持つことは有用です。反復可能なサイクルに入れることで、さらに簡単になります。これらのサイクルは、プロジェクトのマネジメントやリーダーシップに携わる人々の日々

の活動を大幅にシンプルにします。複数のフェーズがあるプロジェクトで使用されるPMBOK®ガイドのプロセスグループのサイクル、PRINCE2®のステージ、P3.expressの日次、週次、月次のサイクル、DSDM®のイテレーションとタイムボックス、スクラムのスプリントは、すべてこのコンセプトの例です。

短いサイクルは長いものより理解しやすく使いやすく、例えば、スクラムのスプリントは、PMBOK®ガイドのフェーズとは対照的です。しかし、短すぎるサイクルは、ある種のプロジェクトをサポートするのに十分ではないかもしれません。その解決策として、DSDM®の長い反復サイクルを伴う短いタイムボックスサイクルの活用や、P3.expressの日次、週次、月次サイクルの使用のように、複数のサイクルを使用することが考えられるでしょう。

例: 方法論

プロジェクトを実行するための方法論やフレームワークを使用することも、繰り返し可能な要素を使用する例といえます。これは、PRINCE2®、P3.express、DSDM®、スクラムなどの既存のシステムでもよいし、自分でカスタマイズしたり構築したりしたものでもよいです。ただし、通常は、ゼロから構築するよりも、既存の手法の1つから始めて、ニーズに合わせて適応させる方がよいでしょう。

どれも反復可能な要素というのは抽象的なものなので、現実の世界で使うにはカスタマイズするが必要です。しかし、抽象度合いとカスタマイズの必要な度合いには幅があります。小規模で比較的具体的な品質チェックリストは、抽象度とカスタマイズの必要性が最も低い側の端に位置し、一方、方法論のようなものはカスタマイズの必要性が最も高い側の端にあります。反復可能な要素が適切にあなたのニーズに合わなくならないように、いつもカスタマイズの必要性については留意しておくことが重要です。

NUPPは、プロジェクトの”ほぼ普遍的な原則”を集めたもので、方法論やアプローチにかかわらず、プロジェクトの成功率を最大化するために、あらゆるプロジェクトで従うとよい原則です。

プロジェクトを運営するために利用できるリソースや手法というのは、どれもこのNUP（ほぼ普遍的な原則）のいくつかの原則に基づいているといえます。ただし、以下の点には留意する必要があります：

- 全てがいつも当てはまるという訳ではないが、プロジェクト実践者は、まずいくつかではなく全てのNUPを考慮してみるとよい。
- 物事の根底にある原則というものは、リソースや手法の中に明確には表れていないことが多く、そのためほとんどのプロジェクト実践者は実務の細部に没頭するがあまりに、原則を忘れ、原則に適合しないことを行ってしまう

NUPPは、PRINCE2®、PMBOK®ガイド、P3.express、PM²、DSDM®、XP、スクラムなどの主要な手法、システム、リソース、フレームワークにも適合します。これらのシステムのある部分の解釈とは相容れない時もあるかもしれませんが、その場合は他に解釈がないかと考えてみるとよいかもしれません。

NUPPは以下のNUPの集合体です：

- NUP1: 所属よりも結果と真実を優先すること
- NUP2: エネルギーと資源を維持・最適化すること
- NUP3: 常にプロアクティブであること
- NUP4: 鎖の強さは一番弱いつなぎ目で決まることを忘れないこと
- NUP5: 明確な目的のない限り、何もしないこと
- NUP6: 反復可能な要素を活用すること