

連載「プロマネの現場から」

第 217 回 経験だけでは人は成長しない

ープロジェクトが人を育てる本当の理由ー

蒼海憲治 (大手 SI 企業・グループ会社・事業部長)

1. プロジェクトの振り返りの場で

プロジェクトが一段落したとき、チームで振り返りのミーティングを行うことがあります。

「今回のプロジェクトは大変でしたね」

「学びの多い案件でした」

「今回の経験は、次に活かしましょう」

そんな言葉が交わされることは珍しくありません。

確かに、プロジェクトの現場には多くの経験があります。トラブルもあり、試行錯誤もあり、成功と失敗が入り混じっています。プロジェクトに関わったメンバーの多くは、その経験を通じて何かを学んだと感じているでしょう。

しかし、しばらくすると、その経験がどのように活かされたのかは、あまり語られなくなります。そして次のプロジェクトでは、どこかで見たような問題がまた現れます。

長年プロジェクトの現場に関わっていると、ある疑問が浮かびます。

経験を積みば、はたして人は本当に成長するのでしょうか。

2. 経験だけでは人は成長しない

多くの組織では、「経験を積みば人は成長する」と考えられています。確かに、経験は重要です。若手エンジニアもプロジェクトマネージャーも、実際のプロジェクトを通じて多くのことを学びます。

しかし現場を見ていると、少し違う現実があります。

同じように多くのプロジェクトを経験しているにもかかわらず、着実に成長していく人もいれば、何年経っても同じような問題を繰り返してしまう人もいます。

もし経験そのものが人を成長させるのであれば、このような差は生まれないはずです。

では、この違いはどこから生まれるのでしょうか。

この問いに対して示唆を与えてくれるのが、組織心理学者・デービット・コルブ（David A. Kolb）が提唱した、いわゆる「経験学習のサイクル」（Experiential Learning Theory）という考え方です。

3. 経験学習のサイクル

経験学習では、人が学習するプロセスを次の四つの段階で説明します。

- ・ 具体的経験
- ・ 内省（振り返り）
- ・ 概念化
- ・ 実験

つまり、

「経験」 → 「振り返り」 → 「教訓化」 → 「次の行動」

という循環です。

ここで重要なのは、「学習の核心は経験ではなく振り返りにある」という点です。

経験そのものが人を成長させるものではありません。経験を振り返り、その意味を理解し、次の行動を変えることで初めて学習が成立します。

この考え方は、プロジェクトの現場に非常によく当てはまります。

4. プロジェクトは経験学習の宝庫

プロジェクトという仕事は、本質的に「一度として同じものが存在しない仕事」です。

- ・ 顧客が違う
- ・ 業務が違う
- ・ システムが違う
- ・ メンバーが違う

つまり、同じ条件のプロジェクトは存在しません。

そのため、マニュアルだけでは対応できない状況が必ず生まれます。

だからこそ、プロジェクトマネージャーにとって重要なのは

「経験から学ぶ力」になります。

優秀な PM を観察すると、ある共通点があります。

それは、「経験を知恵に変換する速度が速い」ということです。

同じトラブルを経験しても、

ある人は「大変だった」で終わり、

別の人は「次はこうする」と教訓に変える。

この差が、数年後には大きな実力差になります。

5. ソフトウェアライフサイクルに潜む学習ポイント

プロジェクトの経験を振り返るとき、私はよく ソフトウェアライフサイクル の観点で整理します。

ソフトウェア開発は一般的に

- (1) 企画
- (2) 要件定義
- (3) 設計
- (4) 開発
- (5) テスト
- (6) 運用

という工程を経て進みます。

興味深いことに、プロジェクトの問題はこの各工程で典型的な形をとって現れます。しかし、その原因を深く探ると、多くの場合、問題は別の工程に存在しています。

プロジェクトを振り返ると、各工程にはそれぞれ 重要な学習ポイント が潜んでいることに気づきます。

(1) 企画：目的が曖昧なままプロジェクトは始まる

あるシステム刷新プロジェクトでは、経営層は「DX 推進」を掲げていました。一方で現場部門は、既存業務を大きく変えることには慎重で、「まずは現行業務の効率化を進めたい」という考えでした。

つまり、「プロジェクトの目的が組織内で一致していなかった」のです。

この違いは当初は表面化せず、プロジェクトは予定通りスタートしました。しかし要件定義に入ると、

「業務を変えるのか」

「既存業務を踏襲するのか」

という議論が何度も繰り返されることになりました。

振り返ってみると、このプロジェクトの問題はシステムの設計や開発ではなく、「目的の共有が不十分だったこと」にありました。

この経験から得られた教訓は、

「プロジェクトの成功は、目的の共有から始まる」

ということでした。

(2) 要件定義：合意とは理解である

別のプロジェクトでは、要件定義書は予定通り完成しました。レビューも実施され、関係者の承認も得られていました。

しかし設計工程に入ると、ユーザー部門から、「そんな仕様は聞いていない」という声が上がりました。

調べてみると、要件定義書は確かに存在していました。しかし、その内容を実際に深く理解していた人はほとんどいませんでした。

「要件は文書としては合意されていましたが、理解としては共有されていなかった」のです。

この経験から得られた教訓は、「合意とは文書ではなく理解である」ということでした。

要件定義とは、文書を作る工程ではなく、「関係者の認識を揃えるプロセス」なのです。

(3) 設計：理解されて初めて設計になる

あるプロジェクトでは、非常に優れたアーキテクトが詳細な設計書を作成しました。アーキテクチャも整理され、設計書としての完成度は高いものでした。

ところが開発工程に入ると、現場の開発メンバーから次のような声が上がりました。

「設計書の意図がよく分からない」

「この構造で実装する理由が理解できない」

設計書は確かに存在していました。しかし、その内容が実装者に十分理解されていなかったのです。その結果、開発チームは設計を十分に咀嚼できないまま実装を進めることになり、実装のばらつきや後工程での手戻りが発生しました。

振り返ってみると、この問題は設計の品質そのものというより、「設計が共有されていなかった」ことにありました。

この経験から得られた教訓は、「設計とは、理解されて初めて意味を持つ」ということでした。

（４）開発：プロジェクトは統合の仕事

開発工程では、各チームが順調に作業を進めているように見えても、統合の段階で問題が発生することがあります。あるプロジェクトでは、各開発チームの進捗は順調でした。進捗報告でも大きな問題はなく、スケジュールも守られていました。

しかし統合テストに入ると、機能間の整合性の問題やインタフェースの不一致が次々に見つかりました。

原因を調べると、各チームはそれぞれの担当範囲では最適な設計と実装を行っていましたが、「システム全体としての整合性を確認する視点が不足していた」のです。

つまり、「部分最適の積み重ねが、全体最適を崩していた」と言えるでしょう。

この経験から得られた教訓は、「プロジェクトとは統合の仕事である」ということでした。

（５）テスト：品質はテストで作られない

テスト工程では、しばしば大量の不具合が見つかることがあります。

あるプロジェクトでも、テスト工程に入った途端に不具合が急増し、チームは対応に追われることになりました。

当初は「テストが厳しすぎるのではないか」といった声もありました。しかし不具合の原因を分析すると、その多くは

「設計段階の曖昧さ」に起因していました。

つまり、テスト工程で発見された問題の多くは、「テスト工程の問題ではなかった」のです。

この経験から得られた教訓は、「品質はテストで作られるのではない」ということでした。

品質は、「設計と開発の段階で作り込まれるもの」なのです。

(6) 運用：システムの成功は稼働ではなく定着で決まる

システム開発のプロジェクトでは、本番稼働が一つの大きな節目になります。

あるプロジェクトでも、システムは予定通り稼働しました。性能にも問題はなく、技術的には成功と言えるものでした。

しかし運用が始まると、現場からは

「使いにくい」

「業務に合わない」

という声が多く聞かれるようになりました。

システムは仕様通りに動いていました。しかし、実際の業務の流れとは微妙にずれていたのです。

この経験から得られた教訓は、「システムの成功は、稼働ではなく定着で決まる」ということでした。

システム開発のゴールは、「リリースではなく業務改善」なのです。

(7) 問題は一工程前で生まれる

ソフトウェアライフサイクルに沿ってプロジェクトを振り返ってみると、ある共通したパターンが見えてきます。

それは、「問題は常に一工程前で生まれる」ということです。

テスト工程で発見された不具合の多くは、テスト工程そのものの問題ではありません。その原因は設計の曖昧さや仕様の不整合にあります。

設計工程で生じた混乱を辿っていくと、その多くは要件定義の段階での認識のずれに行き着きます。

さらに要件定義の問題を深掘りしていくと、そもそもプロジェクトの目的や方向性が十分に共有されていなかった、という企画段階の問題に行き着くことも少なくありません。

つまり、

- ・テストの問題は設計に原因があり
- ・設計の問題は要件定義に原因があり
- ・要件定義の問題は企画に原因があります。

プロジェクトのトラブルはしばしば後工程で顕在化します。しかし、その多くは前工程で既に作られているのです。

プロジェクトマネジメントの視点から見ると、ここに重要な洞察があります。

それは、「問題が発見された場所と、問題が生まれた場所は違う」ということです。

テスト工程で問題が見つかったからといって、テスト工程を改善すればよいとは限りません。むしろ、その問題が生まれた工程まで遡って考える必要があります。

言い換えれば、プロジェクトマネジメントとは、「問題を『前工程』に遡って考える仕事」とも言えるでしょう。

この視点を持つと、プロジェクトの見え方が大きく変わります。

たとえば、テスト工程の品質問題を見たときに、

「テストをもっと厳しくする」

と考えるのではなく、

「設計の段階で何が曖昧だったのか」

と問い直すことができるようになります。

さらに、

「要件定義の段階で何が共有されていなかったのか」

と考えることもできます。

このように問題を前工程に遡って考えることができるようになると、プロジェクトの改善はより本質的なものになります。

プロジェクトの世界ではよく、

「後工程になるほど手戻りのコストは高くなる」

と言われます。

だからこそ、問題が顕在化した場所だけを見るのではなく、その原因が生まれた場所まで遡って考えることが重要になります。

長年プロジェクトに関わっていると、次のような感覚を持つようになります。

「プロジェクトの問題は、突然起きるのではない。
それは前工程で、静かに作られている。」

6. 生成 AI 時代の経験学習

近年、生成 AI の進化によって、この経験学習のプロセスも大きく変わりつつあります。
これまでプロジェクトの振り返りは、人の記憶や経験に頼る部分が大きいものでした。プロジェクトが終わった後、関係者が集まり、

「今回の問題は何だったのか」

「次はどうすべきか」

と議論する形で振り返りを行います。

しかし実際には、プロジェクトが進むにつれて多くの出来事が積み重なり、誰が何をどのように判断したのかを正確に思い出すことは容易ではありません。振り返りはどうしても個人の記憶や印象に依存しがちになります。

生成 AI の登場によって、この状況は変わり始めています。

現在のプロジェクトには、

- ・ 会議ログ
- ・ チャット履歴
- ・ 議事録
- ・ チケット管理システム
- ・ ソースコード履歴

など、多くのデジタル記録が残されています。

生成 AI はこれらの情報を横断的に分析することで、プロジェクトの進行過程を客観的に振り返ることが可能になります。

たとえば、ソフトウェアライフサイクルの各工程においても、AI は次のような形で振り返りを支援できるようになっています。

「企画」フェーズでは、会議ログや意思決定の記録を分析することで、プロジェクトの目的がどのように議論され、どの段階で認識のずれが生まれたのかを可視化することができます。

「要件定義」フェーズでは、議事録やレビューコメントを分析することで、要件の曖昧さや関係者間の理解の差を早期に発見することが可能になります。

「設計」フェーズでは、設計書と実装コードの関係を分析することで、設計意図がどこまで実装に反映されているかを確認することができます。

「開発」フェーズでは、チームごとの開発履歴やコミュニケーションのパターンを分析することで、部分最適が進みすぎていないか、統合リスクが高まっていないかを把握することができます。

「テスト」フェーズでは、不具合の発生パターンを分析することで、問題がどの工程で作られた可能性が高いのかを推定することもできます。

「運用」フェーズでは、ユーザーの利用状況や問い合わせ内容を分析することで、システムが実際の業務にどのように定着しているのかを把握することができます。

このように、生成 AI はプロジェクトの各工程に残された情報を分析することで、

「プロジェクトの経験を可視化する」役割を果たすようになりつつあります。

経験学習のサイクルで言えば、AI は特に、『内省』と『概念化』のプロセスを強力に支援する存在です。

人間が経験を振り返る際には、どうしても主観や記憶の偏りが入り込みます。しかし AI は、大量のログや記録を分析することで、プロジェクトの構造をより客観的に示すことができます。

その結果、プロジェクトの振り返りは、

「なんとなく大変だった」

という印象的な議論から、

「どこで何が起きていたのか」

という構造的な議論へと変わっていきます。

もっとも、AI が示すのはあくまでデータのパターンです。

そこから何を学び、次にどのような行動を取るべきかを判断するのは、人間の役割です。
言い換えれば、AI はプロジェクトを分析することはできますが、プロジェクトから意味を見出すことは、人間にしかできません。

生成 AI の時代において、プロジェクトマネージャーの役割は変わります。

進捗を管理するだけの存在ではなく、「経験から学ぶ組織をつくる存在」へと進化していくのではないのでしょうか。

7. 経験から学ぶ組織をつくる

ここまで見てきたように、プロジェクトには多くの経験が蓄積されています。しかし、その経験が組織の知恵として活かされるとは限りません。

プロジェクトが終わると、メンバーは次の案件に移り、経験はそれぞれの個人の中に残ったままになります。結果として、同じような問題が別のプロジェクトで再び起こることになります。

多くの組織では、「経験はあるが、学習は起きていない」という状態になりがちです。

経験学習の観点から見ると、ここには大きな断絶があります。

プロジェクトの経験は本来、
「個人の学び」 → 「組織の知恵」
へと変換されるべきものです。

しかし、その橋渡しをする仕組みがなければ、経験は単なる出来事で終わってしまいます。

この点で重要になるのが、プロジェクトマネージャーの役割です。

プロジェクトマネージャーは、単にスケジュールやコストを管理する存在ではありません。

もう一つの重要な役割は、「プロジェクトの経験を学習へと変えること」です。

たとえば、

- ・プロジェクトの節目で振り返りを行う
- ・成功と失敗の要因を整理する
- ・得られた教訓を次のプロジェクトに共有する

こうした活動は、一見すると地味な作業に見えるかもしれません。

しかし、この積み重ねこそが、組織の学習能力を高めていきます。

生成AIの登場によって、プロジェクトのログや記録を分析することは容易になりました。

しかし、それだけでは組織は学習しません。

データを分析することと、組織が学ぶことは別の問題だからです。

重要なのは、そこから得られた気づきを、「次の行動に結びつけること」です。

つまり、

- ・経験を振り返り
- ・学びを言語化し
- ・次のプロジェクトに反映する

という循環を組織として回すことが必要になります。

プロジェクトが多い組織ほど、この循環が重要になります。

プロジェクトは、単に成果物を生み出す場ではありません。

それは、「組織が学習する場」でもあるのです。

プロジェクトマネージャーは、その学習を促す存在とも言えるでしょう。

言い換えれば、プロジェクトマネージャーとは、「経験から学ぶ組織をつくる存在」なのです。

8. 経験は変身資産になる

ここまで見てきたように、プロジェクトの現場には多くの経験が存在します。

しかし、その経験がそのまま価値を持つわけではありません。振り返り、意味づけられ、次の行動に結びついたとき、初めてその経験は学びへと変わります。

この視点から見ると、経験学習とは単にプロジェクトを改善するための手法ではありません。それは、「個人のキャリアを形づくるプロセス」でもあります。

近年、『ライフ・シフト』（リンダ・グラットン）などで語られているように、これからのキャリアは一つの職務や専門に固定されるものではなく、環境の変化に応じて自らを変化させながら形成していくものになると言われています。いわゆる「プロティアン・キャリア」

(神プロテウスのように、社会の変化に応じて、自分の意思で自由に姿を変え、形成していくキャリアのこと) の考え方です。

このとき重要になるのが、「変身資産」という考え方です。

変身資産とは、環境が変化したときに自分を変えることができる力のことです。知識やスキルだけではなく、人とのつながりや経験から得た洞察も含まれます。

プロジェクトの経験もまた、この変身資産の重要な源泉になります。

プロジェクトには不確実性があります。想定外の出来事が起き、関係者の思惑が交錯し、計画通りには進まないことも少なくありません。

しかし、そうした経験を振り返り、

- ・何が起きたのか
- ・なぜ起きたのか
- ・次はどうすべきか

を考えることで、その経験は単なる出来事ではなく、

「次の状況に対応するための知恵」へと変わっていきます。

このようにして蓄積された経験は、個人の中に残り、次のプロジェクト、さらには次のキャリアの場面で活かされていきます。

言い換えれば、「経験学習とは、変身資産を蓄積するプロセス」でもあるのです。

プロジェクトの世界では、同じ案件は二度と存在しません。

しかし、そこで得られた経験は決して消えるものではありません。振り返りを通じて意味づけられた経験は、次の挑戦に向かうための資産になります。

だからこそ、経験を単なる出来事で終わらせるのではなく、「学びとして蓄積していくこと」が重要になります。

プロジェクトは一度きりの仕事です。

しかし、そこから得られる経験は、個人の中に残り、やがてその人の働き方や意思決定のあり方を形づくっていきます。

その意味で、プロジェクトとは、

「変身資産を育てる場」でもあるのかもしれません。

9. おわりに

本稿では、経験学習のサイクルという考え方を手がかりに、プロジェクトの現場における学びについて考えてきました。

プロジェクトには多くの経験があります。成功も失敗も、試行錯誤も意思決定も、すべてが経験として積み重なっていきます。

しかし、その経験が自動的に学びになるわけではありません。

振り返られなかった経験は、やがて忘れられていきます。そして組織の中では、同じような問題が別のプロジェクトで繰り返されることになります。

経験学習の観点から見ると、ここに大きな分岐点があります。

経験を単なる出来事として終わらせるのか。

それとも振り返りを通じて、次の行動を変える学びへと変えるのか。

この違いが、

- ・個人の成長を分け、
- ・組織の学習能力を分け、

さらにはキャリアの可能性をも分けていくのではないのでしょうか。

プロジェクトの世界では、同じ案件は二度と存在しません。しかし、その中で得られた経験は、振り返りによって意味づけられることで、次の挑戦に活かされる知恵へと変わります。

そして、その経験はやがて、「個人の変身資産」として蓄積されていきます。

プロジェクトマネージャーの仕事は、単にプロジェクトを完遂することだけではありません。

プロジェクトの経験を振り返り、そこから学びを引き出し、次のプロジェクトへとつなげていくこと。言い換えれば、

「経験から学ぶ組織をつくること」もまた、重要な役割の一つです。

プロジェクトは成果物を生み出す場であると同時に、「人と組織が学習する場」でもあります。

だからこそ、プロジェクトを通じて得られた経験を、単なる出来事として終わらせるのではなく、学びとして蓄積していくことが重要になります。

最後に、今回のテーマを一行でまとめるとすれば、こう言えるかもしれません。

「経験は人を成長させない。

振り返られた経験だけが、人を変える。」

そして、その積み重ねが、やがて「経験から学ぶ組織」をつくっていくのだと思います。