

解説

学習する組織—ソフトウェア・プロセス改善に基づく組織の継続学習—

Learning Organizations : Continuous Learning of Software Organizations based on Process Improvements by Mitsuru OHBA (Computer Science Department, Hiroshima City University).

大場 充¹

¹ 広島市立大学情報科学部

1. 組織として学習する

学習は、本来生物、とくに動物にみられる現象である。その現象は、特定の刺激に対する特定反応の形成という行動パターンの変化として観測される。学習によって獲得された行動パターンは、多くの場合その個体または種に対して好ましい結果を保証する。

従来、そのような学習の自然な解釈から、学習という言葉は生物の行動に対してのみ使われてきた。学習自体が議論され、研究されるようになったのは、コンピュータを含めて歴史的には最近のことである。学習の本質を考えれば、その主体が生物、とくに動物に限定される理由はない。学習に焦点をあてて考えれば、その主体が生物であるかどうかは本質的な問題ではない。

以上の議論から「組織として学習する」ことが、決して特異な現象でないことが理解できる。動物が学習するように、人間の集合体である組織も学習できるはずである。しかし、動物と違ってそれ自身が主体性をもたない、人工的な機構である組織の学習とはどのようなものであろうか。これは、必ずしも自明な問題ではない。

組織学習の概念は、1990年にマサチューセッツ工科大学、スローン・スクールのP. Senge教授によって提案され、米国の経営者に広く支持されたものである¹⁾。スローン・スクールの夏季経営者セミナーに参加した多くの米国大企業の経営者が、彼の問題提起から多くのことを学び、それこそが将来の米国企業の経営課題と確信した。そして、多くの企業でこの問題に対する取組みが開始された。1991年のことである。

Sengeの主張は、彼の日本企業に関する研究が基になっている。彼は、当時の米国を代表する企

業の文化と、日本を代表する企業の文化を比較して、次のように説明した。米国では、技術者や管理者はプロジェクトに参画した経験から学び、成長してゆく。彼らは、プロジェクトが終われば次のプロジェクトを求めて職場を変えてゆく。次の職場でも、また新しい経験から何かを学び成長してゆく。こうして、従業員である技術者や管理者は継続的に学習し、成長し続けてゆく。

ここで重要なことは、経験に基づいて獲得される知識(ノウ・ハウ)は、プロジェクトの実施主体である組織にではなく、プロジェクトの実施のために集められた専門家達だけに蓄積されるという結果である。専門家達はその組織を離れば、蓄積された知識も彼らとともに別の組織へ移転してしまう。このことから歴史的に労働力の流動性の高い米国社会では、個人の競争力は高くなっても、組織自身の競争力はグローバルにみれば決して高いと結論できるものではないことが理解できる。

ここで我が国の状況について考えてみる。

我が国の労働力は、とくに終身雇用制度が定着している大企業では、きわめて流動性が低く、米国とは正反対の極にある。少なくとも、最近まではそうであった。したがって、たとえ知識の蓄積が属人的であったとしても、組織の競争力に直接的な影響は与えなかったといえる。これが、グローバルにみたときの日本企業の優位性の1つである。少なくとも、米国企業が直面したような問題は表面化しにくい。その反面、我が国ではプロジェクトの実施に関する知識は、特定企業を前提としたきわめて個別的なノウ・ハウとして蓄積される結果を生んでいる。このため、技術者においても管理者においても、企業横断的に応用できる専門性が育ちにくい環境を生み出している。

組織に属している専門家だけに、本来企業の競争力の源泉となるべき知識が蓄積されることによって、米国企業には次のような問題が観察される。プロジェクトの実施に関係して、似たような失敗がほとんど定期的に繰り返されるという問題である。似たような設計の製品を開発して、失敗するという例は枚挙にいとまがない。これは、組織における過去の失敗が、しっかりと伝承されていないからである。技術者や管理者は彼らが最善と判断したやり方でプロジェクトを実施する。したがって、彼らの中に同じ組織における過去の失敗の記憶をもつ者がいれば、そのような選択はしない。しかし、誰も過去の失敗を知らなければ、組織の文脈とは独立な選択が行われる結果となる。

極端な場合には、今問題になっているプロジェクトで採用されたアイデアと同じようなアイデアを別のプロジェクトでも採用していて、そのことに誰も気づかないというような事態が、大きな企業では散見されるのである。そのようなことから、組織がその組織の過去の経験から学び、同じような失敗を繰り返さず、さらに確実に成功に結びつくような要因を把握することが、組織の競争力の強化に重要になる。単にプロジェクトの成功や失敗という定性的な問題だけでなく、プロジェクトの総工数や納期の精確な予測などの定量的な問題も、経験に基づく学習が必要な問題である。

2. ソフトウェア・プロセスと組織学習

Senge は、我が国の企業の多くが取り入れている全社的品質管理を、組織による学習の1形態と考えている。組織における過去の事例に学び、似たような失敗を未然に防止するという考え方は、組織学習も全社的品質管理も同じである。さらに、全社的品質管理では、似たような失敗を繰り返すことのないように組織的な歯止め策を要求する。これによって、プロジェクトに直接参画しなかった技術者や管理者も、ある程度同じ経験を共有でき、他人の経験から学ぶことが可能になる。この点において、全社的品質管理は組織学習の実践であるとみることができる。

ある意味で組織学習は、組織の学習に焦点をあてて、全社的品質管理を普遍化し理論化したものであるといえる。全社的品質管理は、必ずしも狭

い意味での品質管理だけに注目しているわけではなく、より効率的効果的な生産方法の開発とその組織的展開をもその視野に入れている。組織学習は、組織的な価値の生産に関する経験的な知識を効率的、組織的に獲得することを目標とする。したがって、その目標に着目すれば、両者に大きな違いはない。

では、そのような組織学習はどのように実現されるのだろうか。人間のように意志をもたない組織が学習するためには、そのための機構が明確に定義されなければならない。そのような組織学習のための基本的な機構として、前述のように全社的品質管理では類似の失敗を未然に防止する歯止めを重要視する。歯止めの具体的な例としては、作業結果のチェック・リストがある。特定の工程の特定の作業でチェック・リストを活用することで、類似の失敗が再発することを防止する。さらに、チェック・リストの内容を改善(たとえば詳細化)することによって、類似問題の再発をより確実に防止することも可能となる。

チェック・リストを活用した歯止めは、作業結果確認活動の一種のマニュアル化である。ほかの歯止めの実施法も基本的には、特定作業における特定活動のマニュアル化とみることができる。歯止めにおける活動の記述は詳細かつ具体的でなければならない。すなわち歯止めは、特定の活動の実施に際して作業者の多くが犯しやすい誤りを、その活動の実施手順を詳細に規定(マニュアル化)することで未然に防止しようとするものである。その記述が十分に詳細でなければ、適用時に解釈の幅が生じるため、その効果にバラツキが生じ、効果的でなくなる。

ここで以後の議論のために、プロジェクトの実施過程をプロセスと呼ぶこととする。プロセスは、一連の作業から構成される工程の組合せで表現されるものとする(図-1参照)。すなわち、プロセスは工程に分解でき、さらに工程は作業に分解できるものとする。作業は、作業者に割り当てられる仕事の単位でもある。作業者が、割り当てられた作業を実施する場合、それは一連の活動に分解される。たとえば、ソフトウェア開発のプロセスには、設計レビューがその工程の1つとして組み込まれている。さらに設計レビューは、準備、精査、レビュー会議、修正、再精査、再レビ

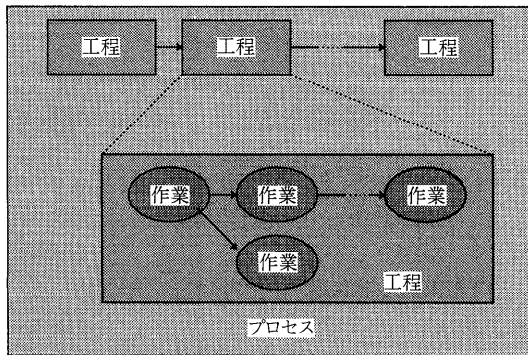


図-1 プロセスと工程

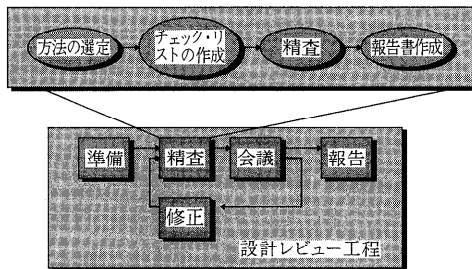


図-2 工程、作業、活動

ユー会議、報告書作成などの作業から構成される。さらに、個々のレビューによる精査は、精査方法の選定、精査チェック・リスト作成、精査(読む)、精査報告書作成などの詳細な活動に分解される(図-2参照)。

全社品質管理の歯止めの概念を普遍化することによって、組織学習の機構が決定できることは以上の議論で明らかであろう。マニュアル化するのは、特定な工程の特定な作業を実施するための詳細な活動について、その標準的な手順を明確に記述することである。すなわち、活動のレベルに至るまで詳細にプロセスを記述することこそが歯止めの本質である。一般には、プロセスの記述をするとき、作業のレベルまでしか記述しないことが多い。組織学習で求められているのは、歯止めと同じレベルでのプロセスの記述である。その意味で、従来のプロセス記述は組織学習に利用するには記述のレベルが抽象的で、詳細さが十分でない。

プロセス記述の詳細さが十分でないと、記述されている作業が記述されたとおり実施されたかどうかを判定することは困難である。作業をどのような手順で実施するかは、作業者の解釈に依存す

るからである。そうだとすれば、実際の作業結果は記述されたプロセスの完成度よりも、作業者の記述内容の理解度を含めた作業員個人の能力により大きく依存する結果となる。記述されたプロセスと実際に実施されるプロセスを一致させるには、プロセスを詳細に記述することが前提となる。

実施すべきプロセスがその活動のレベルに至るまで詳細に記述されると、実施されるプロセスと記述されているプロセスとの一致度は高まる。したがって、記述の少し違った2つのプロセスを似たような条件で実施し、結果が大きく違えばその差は2つのプロセスの違いによって生じたものと結論できる。つまり、理論的にプロセスの実験が可能になる。もし、プロセスAが多くの場合に、プロセスBよりもよい結果を生み出すとすれば、それはプロセスAがプロセスBよりも優れていることが経験的に結論できる。

ISO 9000でも強調されているように、記述されたプロセスが実施されるプロセスと一致していることは、上述のように組織での経験を形式化する観点から重要である。とくに、知識集約的な作業を中心としたソフトウェア開発では、そのことが組織全体の効率化に大きく影響する。したがって、測定・製造技術もプロセスと同じくらい重要なハードウェアの開発よりも、手作業の占める割合の大きなソフトウェアでは、プロセスの完成度がより大きな問題となる。つまり、ソフトウェア開発組織においては、プロセスの完成度が組織の競争力の源泉となる。

3. 経験工場と組織学習

ソフトウェア開発組織においてはプロセスの完成度が問題である。そのプロセスの完成度は、その組織における過去のプロジェクトの経験に基づいて、プロセスを少しずつ改善することによって高められる。ソフトウェア開発プロジェクトでは、要求に合ったソフトウェアを開発すると同時に、ある意味でプロセスに関する実験を実施し、長期的にはよりよいプロセスを見出す試みを行っている」と解釈できる。

そのような認識に立つと、ソフトウェア開発プロジェクトから組織として積極的に学ぼうとする米メリーランド大学の V. Basili の経験工場は、

ソフトウェア開発組織における組織学習についての具体的な方法論を提案した理論的枠組みと捉えることができる^{2),3)}。経験工場は、段階的プロセス改善の考え方と、それを経験に基づいて実践するためのG/Q/M法を柱にする。段階的プロセス改善は、以下のようなステップを繰り返し実施することで、プロセスを最適化しようとする考えである。

- ①最初に、現状のプロセスをできるだけ詳細に記述する。
- ②次に、現状のプロセスのボトル・ネック（最も大きな問題を抱えている部分）を分析し、その改善案を新しいプロセス記述としてまとめる。
- ③現状の改善案としてまとめられた新しいプロセス記述を実際のプロジェクトで実験する。実験結果から新しいプロセスが有効なものか否かを判定する。
- ④最後に、現状のプロセスを改善する新しいプロセス記述をその組織のプロセスとして展開する。

このように、プロジェクトを新しいソフトウェアの開発の場であると同時に、経験的知識の獲得の場であると考え、さらに、経験的知識を効率よく獲得するために、G/Q/M法が提案されている。この方法はまず、現状で何が分かっているかを整理し、次に獲得したい知識をどのように実験から抽出すべきかを整理する方法である。それは、名前のG (Goal) が目標、Q (Question) が設問、M (Metric) が尺度を表しているように、目標（何が知りたいか？何を達成したいか？）から出発して、それを設問（何が分かっているか？どのような状態になれば目標が達成されたといえるか？）として分解し、さらに各設問に明確な答えを得るための実験で何を観測（測定）すべきかを定義することを提案する（図-3）。

当然のことであるが、「新しく提案されたプロセスが従来のプロセスよりもよい」ものであることを実証するためには、その評価の基準（ベース・ライン）となる知識（データ）が必要である。そのために測定が必要不可欠である。また、一般的には新しく提案されるプロセスは万能なプロセスではなく、ある問題に焦点をあてたとき、従来のプロセスよりも望ましい結果を与えてくれ

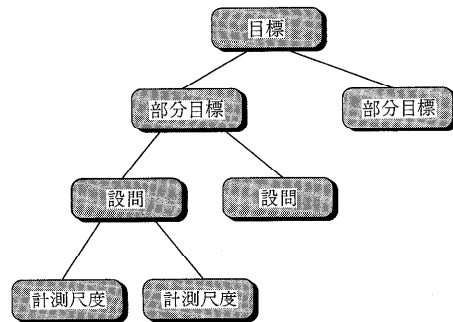


図-3 G/Q/M法の構成

るだけのことである。したがって、「何が問題か？」を目標という形で明確にしておくことが重要になる。目標は、より具体的で詳細な部分目標に分解されるが、ある程度まで分解されると、「その（部分）目標が達成されたということは、どのような状態をいうのか？」という設問で形式化の方が合理的である場合が多い。G/Q/M法はこのような思想に基づいてまとめられたものである。

4. 実践の事例

前述したように、1991年からの数年間、米国のさまざまな企業が組織学習を全社的な課題として取り組んだ。それは、1980年代後半からの日本企業との競争力の比較研究（ベンチマーク調査と呼ばれる）、1990年頃からのISO 9000への取組みと、ほぼ同時進行した。結果として、現在の米国企業の競争力の基礎体力を構成したといえよう。以下に、IBM社で著者自身が関係した組織学習への取組みについて紹介する。

相対的に高い信頼性を要求される従来のメインフレームを中心とするサーバ用のオペレーティング・システム（OS）の開発では、そのテストが従来から問題であった。これは、特定の企業が仕様を規定したOSだけに特有な問題ではなく、標準仕様に基づいて開発されるOSやほかの基本ソフトウェアにも共通した問題である。開発企業は、ユーザの暗黙の信頼性に関する品質要求を満足させるために多大なコストを投入する結果になっていた。それは、極端に単純化して言えば、「企業のソフトウェア開発組織が過去のプロジェクトの経験から学んでいなかった」ことの証拠であった。確かに、技術者や管理者個人個人は学ん

できたし、開発技術も進歩した。しかし、経験豊かな技術者や管理者が去った現場では、彼らが去ると同時に過去のプロジェクトの経験もなくなってしまった。

そのような背景から、1991年半ばに同社内のテスト・プロセスの改善をテーマに組織学習のパイロット・プロジェクトが開始された。このプロジェクトの全体の目的は、IBM社のさまざまな基本ソフトウェア開発部門に蓄積された経験的知識を整理して、企業内での標準的なプロセスを提案することであった⁶⁾。このため、プロジェクトの最初の段階では、現状のプロセス（複数）の調査を実施した。この調査と同時並行的に、我が国の類似した組織での実態調査も実施した⁹⁾。その調査で明らかになったことは、一般にテストのプロセスは、従来のような比較的高いレベルの記述では、組織間にそう違いがみられないものの、実際に「個々の作業をどう実施しているか？」まで調べると、組織間で違いがあることが分かった。また、「テストを何のためにやっているのか？」に関する認識が、個々の技術者や管理者によって違っていることも明らかになった。

この実態調査では当時、各組織において「どのようなデータが収集され、どう活用されているか？」についても調査した。その結果、「テスト・プロセスが有効に働いているかどうか」を判断するための測定尺度や基準がないことが判明した。これは、「現状のプロセスがどの程度有効なものか？」を評価する基準すらないことを意味している。そこで、調査後の活動は、G/Q/M法に基づいて、テストの各工程別に標準的な測定尺度を定義し、その尺度に基づいて現状のプロセスを測定し、現状のさまざまなプロセスを評価することに集中した。

たとえば、単体テストからシステム・テストへの橋渡しとして、統合テストとか機能テストとか呼ばれる、機能仕様と実現された機能に差がないことを確認する大規模なテストが実施されている。テスト対象はこのテストで初めて複数のコンポーネントが結合され、仕様書と同じレベルでテストされる。この段階で漏れがあれば、ソフトウェアは仕様書どおりには動かない。したがって、テストは網羅的になる。しかし、テストに投入される資源と時間には限界があるので、真に網羅的な

テストは実現不可能である。つまり、現場の技術者は、「できるだけ少ないサンプルで、複雑に絡み合った機能が正しく実現されていることを確実に確認する」という矛盾した問題に直面する。これは、経験豊かで有能な技術者にしか解決できない問題である。実際に、IBM社でもそのような使命をもった組織には、設計などのほかの使命をもった組織と比較して長期間その組織で経験を積んでいる技術者が多いことに驚かされた。

この問題に関して、著者らのグループはいくつかの網羅度を測定することを提案した。その中でもとくに著者らが注目したのが、従来はそのような機能レベルのテストの評価尺度とは考えられていなかったステートメント・カバレッジを利用することであった⁶⁾⁷⁾。実態調査中に何人かの優秀な技術者が、「機能レベルのテストでも常にステートメント・カバレッジを測定している」と証言したことに気づいた。彼らにその感想を聞くと、その効果を理論的ではないが感覚的に肯定する者が多かった。そこで、この尺度を中心に機能レベルのテスト工程での作業の有効性を評価する尺度を定義し、いくつかのプロジェクトで実験した。結果は、予想どおりであった。もっと驚いたことは、測定値が技術者の事前予想よりもかなり低かったことであった。これには実験に参加した技術者達も驚いていた。

現状のプロセスの有効性を客観的に測定する尺度の集合が決まり、いくつかのプロジェクトから実測データが収集された時点で、現状のプロセスの分析を行った。実際にはデータは少しずつ違った複数のプロセスから収集されるので、いくつかの仮説に基づいてそれらを集約したモデルを構築した。現状のプロセスを抽象化した標準プロセスは、そのようなモデルを記述化したものであった。ベースラインと呼ばれる、各工程の作業が有効に実施されたかどうかを評価する基準値も、実測値からではなくモデルから逆算して決定した。

最後に実験したことは、エキスパートと呼ばれている優秀な技術者が、「どのようにテストを設計しているか？」を詳細に分析し、それを記述することであった。この作業は、エキスパートの実施している作業の詳細な観察とヒアリングによって行われた。ヒアリングではたとえば、「どのよ

うな点に注意して機能仕様書を読むのですか?」
 というような質問がなされ、観察の結果と突き合わされた。そのような現状分析と既知のテストの理論を対比することで、現状のベスト・プラクティスに最も近いと想定される作業手順をテストの理論に基づいて詳細に定義した。そのような手順の最も重要な部分は、ツールとして実現した。そして、ツールを利用した実験によって定義された手順をエキスパートによる作業と比較した。この場合、ツールは詳細で厳密な手順の記述と等価である。ツールを利用した結果とエキスパートによる作業の結果に大きな差がみられなければ、記述された手順とツールに組み込まれた知識によって、エキスパートのもっていた知識を外外部化できたといえるからである。

そのようなツールとして実現された手順の詳細な記述の例として、機能仕様書からテスト項目を設計する作業手順がある⁹⁾。その例で適用されたテストの理論は、原因結果グラフ、決定表(デシジョン・テーブル)、同値分割法であった。そのテスト項目設計活動は、機能仕様書の原因結果グラフへの変換(人手で実施)、原因結果グラフから決定表の生成(ツールを利用)、決定表に基づく同値類の指定(人手で実施)、各同値類からのテスト項目の選定(ツールを利用)という4つのステップから構成された。実際にツールを用いてテスト項目を生成してみると、エキスパートが実際に生成した項目の集合とは部分的に違うものの、エキスパートでも合理的と思う項目が選択されていた。

5. おわりに：継続学習と組織学習

米国や日本などの先進国では、これまでの労働集約的で資本集約的なモノの生産を中心とした経済から、急速に知識が価値を生み出す源泉となるサービスを中心とした経済に移行することが予想される。そのような知識を中心とした経済社会では、個人においても、企業においても、地域においても、知識の蓄積が大きな課題となる。ここで問題となる知識とは、教科書などに説明されているような、整理され記述された理論ではなく、多くの場合過去の問題解決の経験に基づくノウ・ハウであることが多い。

そのような知識の蓄積のためには、専門化と継

続学習が重要である。専門化は知識の効率的な蓄積のためであり、継続学習は常に問題解決に必要な最新の知識を活用できるように蓄積された知識の陳腐化を防ぐためである。組織においても個人においても、専門化と継続学習は21世紀のグローバルな競争社会を生き抜くための基本的な条件になると予想される。組織学習は、「組織が継続的に学習するとはどのようなことであるか」を明確に説明した理論である。また、ソフトウェア・プロセス改善を形式化しようとした経験工場は、「ソフトウェア組織における組織としての継続学習をどう実践するか」の指針をまとめたものである。

現在、労働集約的なモノの生産は急速に先進国から労働コストの安い開発途上国へと移転され、先進諸国では空洞化による失業率の増加が社会的な課題となっている。さらに今後は、相対的にはより知識集約度の高いソフトウェア開発などの生産活動も労働コストの安い開発途上国に移転する可能性が高い。そのような経済環境において、先進諸国の知的産業は市場に直結している優位性を活かし、より知識集約度の高い生産活動にシフトしてゆけなければ、産業そのものが衰退する結果となる。そのような認識に立つと、先進諸国では専門家だけでなく、企業、地域、さらに国家として継続的に学習し、先端分野でのグローバルな競争力を維持することが命題となる。

参考文献

- 1) Senge, P.: The Fifth Discipline: the Art and Practice of the Learning Organization, Doubleday (1990).
 ピーター・センゲ：最強組織の法則，徳間書店(1995).
- 2) Basili, V., Rombach, D.: The TAME Project: Towards Improvement-Oriented Software Environments, IEEE Trans, On Software Engineering, SE-14, No. 6 (June 1988).
- 3) Basili, V. et al.: The Software Engineering Laboratory—An Operational Software Experience Factory, Proc. 14th International Conference on Software Engineering, Melbourne, Australia (May 1992).
- 4) Ohba, M.: The Impact of Knowledge on Quality, American Programmer, Yourdon Press (June 1993).
- 5) Ohba, M. and Basili, V.: Study of Large Scale Software Testing.
- 6) 大場 充：ソフトウェア・プロジェクトの実

- 績データ収集・分析技法, SRC (1993).
- 7) Ohba, M., Piwowarski, P. and Caruso, J.: Coverage Measurement Experience during Function Test, Proc. 15th ICSE, Baltimore, MD (May 1993).
- 8) Yokoi, S. and Ohba, M.: TCG: CEG-Based Tool and its Experience, 第13回ソフトウェア信頼性シンポジウム論文集 (Nov. 1992).
(平成8年11月27日受付)



大場 充(正会員)

1949年生, 1971年青山学院大学工学部経営工学科卒業, 1973年同大学院理工学研究科修士課程修了(経営工学専攻)。広島市立大学情報科学部情報数理学科ソフトウェア工学講座教授。1974年日本アイ・ビー・エム(株)入社。1982年同東京基礎研究所, 1990年米国IBM, 1993年同SE研究所を経て, 1994年より現職。この間, ソフトウェア・メトリクス, ソフトウェア・テスト, 抽象データ型の応用, オブジェクト指向技術の評価, 分散環境における協調的問題解決の研究に従事。