



IT人材育成を担う人材を育て 自ら学ぶ組織を「OJL」で推進

加藤正彦氏

一般社団法人 IT人材育成協会 会長

次代を担うIT人材の不足は、95万人ともいわれている。IT人材が情報処理の専門家だった時代は終わり、いまやIT人材はビジネスの担い手である。こうした課題に対して「IT人材を育成、指導できる人材の育成」というアプローチから取り組む、IT人材育成協会の加藤正彦会長から話を聞いた。



育成の支援を行う専門職 ラーニングファシリテータ

私が会長を務めるIT人材育成協会（ITHRD＝アイサード）は、企業内でIT人材の育成を担う専門職や現場における指導者の育成を目的として活動しています。

ITHRDでは、IT人材自身が主体的に学ぶことを支援する指導の人材を「ラーニングファシリテータ」（LF）と名づけ、その育成のための研修プログラムを開発し、実施しています。また、経済産業省が策定したITスキル標準（ITSS）レベル3～5相当の「ラーニングプロフェッショナル」という資格認定制度も運営しており、LF基礎の研修修了者は検定試験に合格することで資格を取得できます。

現在、LF基礎修了者は100人ほどになり、IT教育のベンダーやユーザー企業の社員などが受講しています。

私は大学卒業後、1968年に日本ユニバック（現・日本ユニシス）に勤めはじめましたが、そこでコンピュータシステムのサービスを担当しているときには、まだIT人材という言葉はありませんでした。その頃は、システムを開発する人材は顧客企

業におり、われわれメーカーはそのサポートをするという役割だったのです。

しかしその後、システム開発部門が子会社として独立するなど、独立系のシステム開発会社が数多く生まれ、開発業界は下請け企業が多重化した構造となりました。そのなかで親会社から独立した開発子会社は親会社以外の仕事の比率が増え、システムを開発することが主目的となったため、次第に親会社の業務知識が薄くなりました。一方、親会社にはシステム開発のノウハウが蓄積されなくなっていました。システム開発会社が情報処理だけで仕事ができる時代はそれでもよかったのですが、ITシステムそのものがビジネスにかかわるようになった現在は、業務知識に乏しいシステム開発会社とIT知識に乏しいユーザー企業が、システムを開発しなければならなくなり、開発にあたり難題を抱えるようになっていくのです。システムの開発には、ビジネスや業務に関する知識が必要です。しかも、ビジネスを取り巻く環境が大きく変わり、短期間に開発、改善を繰り返すことが求められています。

ユーザー企業は社内で優れたIT人材を育成する必要に迫られていますが、その人材を育てる講師や指導者が不足しているのです。

私は日本ユニバックから独立したシステム開発会社に移って取締役になり、開発事業や自社商品開発販売などに携わった後、子会社として、2002年にIT教育会社を設立しました。

新会社ではIT人材のための研修プログラムの開発・販売に取り組み、主な顧客先はITベンダー

でした。そのお客さまの人材育成の悩みは何か聞き取りをしようと、人事や育成の担当者を集めて議論したことがあります。そのとき驚いたのは、担当でありながらも育成に関する知識がほとんどないという現実でした。

アメリカでは教育のプロがいて、優れたIT人材を育てているわけですが、日本ではいろいろな部署を回って育成や研修の担当になることが多く、教育や人材育成に関する知識もスキルも不足している場合があつて、いわゆる教育のプロではありません。育成する側のプロを育てないと日本のIT人材は人数だけでなく、質的にも問題になると痛感し、2002年にIT教育ベンダーの有志30社を募って設立された、IT人材育成事業者協議会(ITTVC)に参加しました。

しかし、その後、日本の教育研修市場が縮小してきて、IT教育ベンダー各社とも経営状況が悪化、協議会の参加者も減って、立て直すために2014年に一般社団法人化し、ITHRDに移行しました。IT教育ベンダーだけでなく、ユーザー企業も巻き込み、現在、ベンダー10社、ユーザー企業6社、個人会員10人程度が会員となっています。ユーザー企業が加わって活動も活発になっています。

日米のSEの能力差に愕然 新しいIT人材の育成が急務

IT人材が不足しているといわれていますが、今後は大型のシステム開発案件は業界全体で減少することが予想され、ITベンダーも従来のビジネス形態を大きく変えていく必要があります。顧客へのSE派遣型から請負型に変わってきていますが、これを提案型ビジネスに転換するためには従来型のIT人材では対応できません。

日本のシステム業界は多重下請け構造で、最初に時間をかけて設計し、プログラミングは下請け企業のプログラマーに切り分けて作業させるウォーターフォール型の開発方法が主流です。そのため、設計と実際のプログラミングに違いが生じることもあり、各プログラマーはシステムの全体を理解せず、いわれたことだけをやっている。そのため、能力が伸びず、得意分野を生み出すことができないのです。これからはシステム開発とビジネスを組み合わせる力がないとイノベーションを起こすことができず、現在のIT人材は、さらに新たな知識やノウハウを身につける必要が



「人材育成の比率で70%を占めるのが、職場や仕事を通じてです。ですから、仕事を通じて学ぶことを習慣化してもらうために、専門職としてのラーニングファシリテータ育成と、職場で育成の支援を担うオンザジョブラーニングのファシリテータ育成とその普及を進めています」と語るITHRD会長の加藤正彦氏

あります。

たとえば、ウォーターフォール型に対して、スピーディーに開発を行うアジャイル型の開発手法があります。アジャイルでは、システムを小さな機能に分割し、短期間の単位で開発し、1つずつ機能を追加しながら完成させます。これによってリスクを最小化するわけです。今後のIT人材は当然、アジャイル型の開発手法も知らなければなりません。そのために、効果的に短期間で教育できる人材が必要なのです。

かつて、SEの能力差を強く実感したことがあります。よく、「できるSEとできないSEのパフォーマンス差は10~100倍ある」などといわれますが、アメリカでシステム開発を学んだ日本人SEを採用したことがあり、ある仕事を任せたことがありました。仕事の中に本を読むなど、日本人の感覚からすると不真面目で、仕事ができるのだろうかと思っていたら、あっけなく1人で3週間で完成させたのです。

驚いて、もっと大きな仕事を任せたら、日本人SEが束になってもできそうもない案件を、やはり1カ月程度でつくりあげました。通常は誰でもつくるフローチャートを彼はつくらず、実に美しくわかりやすいプログラミングを書くのです。

おそらくアメリカでオブジェクト指向や抽象化の概念を教わってきたのだと思いますが、アメリカのプロは、皆それくらいの力をもっています。この日米差に愕然とし、育成方法の違いが大きい

のだろうと思いました。

日本の企業内の育成担当者は、とにかく研修さえしていればいいと、その成果にはあまり関心をもっていません。しかし、経営者は投資に対する効果を当然考えます。あまり効果がないと考えれば、研修費を削ろうということになります。

ウェストミシガン大学のロバート・ブリンカーホフ教授によると、研修の効果が出ない原因の40%はその準備段階にあるといいます。受講者が何のためにその研修を受けるのか理解しておらず、心構えがないということです。さらに、原因の40%は研修後、職場に戻ってからサポートやフォローがなく、受講した知識を上手く活用できていないことです。研修プログラムに問題があるのは20%といわれています。しかし、近年さらにこの比率は減少し、研修前の準備、研修後のサポート不足の割合が増えているといわれています。

また、一般的に人材育成において、研修の効果は10%程度といわれています。残り20%は先輩などによる薫陶・指導、そして70%は現場における経験です。研修会社は、この10%だけでビジネスをしているわけですが、本当に人を育てることを考えたら、70%の領域に入らないといけなのです。つまり、仕事をしながら学べる仕組みを提供することです。

残念ながら、現状ではこうした仕組みはほとんどありません。そこで、われわれは、これからお話しする「オンザジョブラーニング」(OJL)を考案し、OJLを推進するOJLファシリテータ(OJLF)育成のプログラムを開発しました。

仕事の現場で学びを導く OJLFの研修プログラムを開発

LFは専門職としての講師であり育成担当者なので、実際の職場や現場での人材育成を推進する役目として、ファシリテータが必要になります。これがOJLFです。

現状のオンザジョブトレーニング(OJT)は新入社員がメインであり、長いスパンでIT人材を育成するには、現場での育成支援であるOJLが大事であることから、それを推進するOJLFの育成研修体系も開発しました。

LFは基礎からエキスパート、プロフェッショナルと3ランクになっていますが、OJLFも基礎、リーダー、マネジャーと3ランクに分かれていま

す。現状のリーダーやマネジャークラスにOJLF研修を受けてもらいたいと思っています(図表)。

1990年代以降生まれのZ世代やミレニアル世代には、昔のように上からの一方的な指示や指令は通用しません。それよりも自発的に参加意欲をもつように指導するファシリテーションのほうがふさわしいでしょう。

OJLを推進するスキームもつくっていますが、基本は職場単位でチームに分けて、そこにOJLFが入り、お互いに学び合うスタイルです。ミーティングで体験を語り合い、情報や思いをシェアします。学習素材は経験であり、毎日毎週繰り返すことで経験が学びに昇華します。人は他者からの刺激で啓発されるものです。研修は年に1、2回しかありませんが、毎日の仕事のなかで学びが積み重なれば大きな進歩になります。スポーツ選手は練習の積み重ねで試合に勝てるわけです。ビジネスパーソンもこうした練習の「場」が必要です。

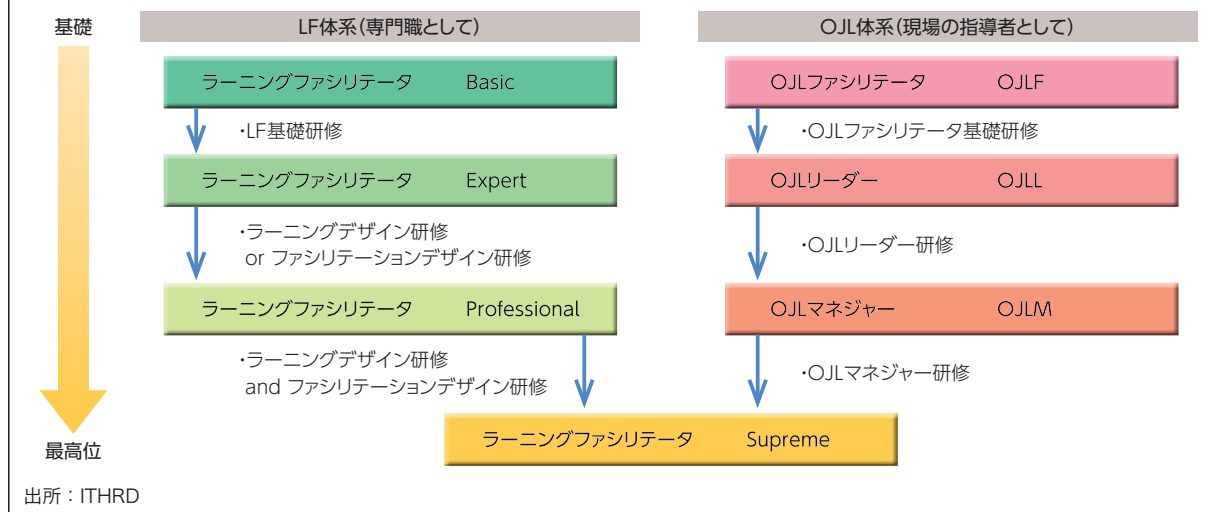
OJLでは仕事を振り返る「リフレクション」が重要な練習の場になります。良かった点、悪かった点、経験をチームで共有し、新しい方法を話し合い、次の仕事に活かし、また振り返る。その繰り返しです。顧客からクレームがあればチームで話し合う。OJLFはチームの仕事と学びを支援します。そのなかから、イノベーションも起きやすくなるでしょう。

実はこのスキームは、アジャイル型のスクラムという開発手法と似ています。スクラムは開発チームが自発的に行動することを前提としており、「スプリント」と呼ばれる工程、すなわち計画、実行、リフレクション、改善、そして新しい方法の計画とサイクルを回していきます。常に進捗がわかるように「見える化」することも特徴です。こうした活動を通じて、実践力や変化対応力が磨かれていきます。そうしたしっかりとした土台の上に、技術力を蓄えていくことができ、技術者としての専門性を高めていくことが可能になります。

OJLを導入した企業で 予想以上に高い効果

ある大手事務機器メーカーで、モデルケースとして2017年2月からOJLを導入し、検証していただきました。ラインの技術者をサポートする150人の部隊のなかから15人のマネジャーに対して、まず、2日間のOJLF研修を受講してもらい、その

図表 ラーニングファシリテータ育成研修体系 (育成認定部会)



うえて、さらに10日後に社内でフォロー研修を行いました。そして、研修を受けたOJLFは現場に戻り、メンバー10人前後でチームを結成しました。年間OJL実施計画を立て、4月からOJLを実施し、9月までの半期でレビューしました。途中結果ですが、予想以上に高い効果を得ることができました。

「チームが活性化したか」という質問ではファシリテータの92%が評価(「そう思う」「かなり」「少し」の合計)し、メンバーは94%に達しました。「学習意欲が向上した」と思うファシリテータは83%、メンバーは88%。「OJTに比べて有効」と思うファシリテータは92%、メンバーは82%となりました。「変化への対応力が増えた」と思うファシリテータは58%と少し低めですが、メンバーは79%が評価しています。

この結果に導入した会社の研修責任者も驚いたほどです。

アメリカのギャロップ社が2017年に従業員のエンゲージメント(仕事への熱意)を調査したところ、日本では「熱意あふれる社員」が6%しかおらず、調査した139カ国中132位というひどいありさまでした。OJLはエンゲージメントも高める効果があると思います。

OJLを導入するには、まず社内に検討委員会を設置し、トップからメンバーまでOJLの必要性を理解し、対象部門を選定し、めざすべき状態と現状とのギャップを明らかにします。

その分析から改善策を立案し、OJLを実践する組織に即した実施計画を策定します。その後、実践し、経過と結果をレビューして改善を加えて活動を継続します。

OJLでめざすべきは仕組みの構築ですが、そのポイントは以下の5つになります。

- ①業績と学習の同期化…業績を生み出す場でリアルタイムに新たな能力開発を伴う学習の仕組み
- ②応用力と適応力の強化…現場の経験を活かした職業的実践力を養成する仕組み
- ③経験学習の推進…常に経験したことを振り返り、新しい適応の方法を学び、実践する仕組み
- ④協働学習の場づくり…チーム、グループ活動を通して振り返り、他者の経験を共有できる真の協働学習となる仕組み
- ⑤変化への対応力…繰り返し押し寄せる経験から学びを通して、激しい変化への対応力を磨くことができる仕組み

企業の人材育成は業績につながらないと意味がありません。組織学習を通じて、業績や生産性を向上させることが重要になります。OJLはその点を重視した仕組みになっています。

企業のトップや幹部、育成および人事担当者には、新しいやり方に果敢にチャレンジしてほしいと願っています。新しいIT人材を育てるには育成の仕組みも変えなければならないからです。しかし、人事や育成部門は従来からのやり方を変えない傾向があります。人材の重要性を認識しているのも、人事や教育を担当する方がたのはずです。そうであるのならば、人事や育成の担当者が経営幹部にもの申すくらいの覚悟がないと、これからのIT人材を育てることはできないと考えて、取り組んでいただけることを切望します。 ㊦