

新入社員プログラミング教育の蘇生 - 日本語能力とプログラミング技能 -

4E-3

田口 昭仁

日立製作所 ソフトウェア開発本部

1. はじめに

ここ数年来、正規のプログラミング教育を経験していない新入社員の即戦力化のために、各企業ともその教育に努力してきたことと思う。

しかし、新入社員に対する初級プログラミング技能の教育は、どこでも実効が上がっていないのが現実である。『プログラミング言語仕様 + コーディング』教育に甘んじてしまっている。その結果、プログラミング演習での課題をどうにか完成しても、受講者には次のような不満が残る(自分で「ヤッター」という達成感がない!)。これは、各人のその後のモラル/メンタルヘルスの面でも問題だろう。

- 「問題を与えられても何から手をつけていいのか、どうPADで書けばいいのかわからない。」
- 「いきなり作れといわれても、……という感じで、ちっとも満足にやれなかったと思う。」
- 「仕様をよく理解できないまま演習にとり組んでしまいました。そのため、それぞれの関数のつながりが見えないままプログラミングをしていて、…」

受講者に達成感を与え実効を上げるには、受講者自身の『考える』働きを活性化させることが重要だと考える。『考える』働きの強化には、考えるツールとしての母国語能力を活用することが不可欠である。

本稿では、現状の教育での与えすぎ(教えすぎ)が『考える』働きを阻害している面を指摘する。

2. プログラミング教育の本質と日本語能力

初級プログラミング教育の受講者の不満への対処策として、よく以下が挙げられる。

- 小さな課題でプログラミングの初歩を順次教える
- アルゴリズムを教える
- プログラム解析(読み/理解)の訓練をさせる

しかし、筆者は、これ以上何かを教え込むのではなく、受講者自身の『考える』働きの活性化が第一歩だと考える。プログラミングという知的作業のために必要な『考える』とは次のようなものであろう。

- 物事や手順等(例えば、無意識的にできる手順)を意識化して把えて言語化する働き
- 物事や手順等(特に、複雑な手順)をイメージ的(抽象化)に把える働き
- 上記の働きをもとに、ボトムアップアプローチとトップダウンアプローチの使い分け/融合のセンス

このような『考える』基本的働きは、新入社員は既に実生活の中で体得済みとも言える。日常体験から得た『考える』働きを活性化することで、【菊池 87】の指摘する「日常の体験や既習事項を設計に役立てられない」という問題点も克服できる。プログラムが人間や人間組織の行為をシミュレーションするものとする、日常体験を意識化し参考にできることは、プログラミングにおいて大切な能力である。

したがって、教育の主眼は、これらの基本的働きをプログラミング作業に沿って活性化することであろう。この点から第一に留意すべきは、受講者自らが能動的に『考える』ようにしむけることである。特に、受身的受講態度の原因になる過剰な教示は避けるようにしたい。第二に、慣れないプログラミング言語の枝葉末節で受講者の『考えようとする』働きを金縛りにしない教え方が必要という点である。

それには、日本語能力を発揮させる。不慣れたプログラミング言語ではなく、慣れた日本語での物事や手順の表出を指導する。『日本人に日本語を教える』のは動機付けの面で難しいが、プログラミング教育をとおして明確な日本語文章を書く能力を養うことは、より上級のエンジニアに育つ土台にもなる。

筆者は、プログラミング方法論の本質的向上のためにも、類似性のある文章作成プロセスの考察が有益だろうと考える(【田口 84】、【菊池 87】、【竹内 93】等参照)。また、具体的なケースとし、ワープロでの文章作成(【木村 93】参照)とPAD/PAMプログラミング法(【二村 84】参照)とも類似性を見出すことができる。前者で『心の地底湖』、『魚』と比喩するものは、後者の『モヤモヤ状態』、『スレッド』にあたる。それぞれ、無自覚状態の『思・考』、少しずつ意識され始めた『断片的アイディアの結晶』を意味する。文章やプログラムを作るということは、これらの結晶を体系的に積み上げ(PAMではマージ)ていくことである。筆者も、【田口 84】で、非母国語による文章作成とワーニエ・プログラミング法との類似を示した。ワーニエ法では、トップダウン的にプログラムの構造を決めることと、行われるべき単位処理(操作)の採集との二つの異質の仕事に分ける。この二者を融合させる過程をマッピングと呼ぶ。

ここで挙げた文章作成やプログラミングのプロセスは、いずれもボトムアップとトップダウンの自然な融合と見ることができる。【川合 85】は、PADではボトムアップでもトップダウンでもない自然な順序で考えられるし、それが問題解決の自然の心理的順序であろうと述べている。

Towards Essential Improvement of Fundamental Programming-Courses in Japanese Computer-Makers
Akihito Taguchi; Hitachi, Ltd.

5030 Totsuka-cho, Totsuka-ku, Yokohama, Kanagawa, Japan

3. 現状の阻外要因の克服

現状の初級プログラミング教育の教材や指導内容が『考える』働きを阻害していると思える点として下記の①や②等がある。このような要因を除去することで受講者の『考える』働きが活性化でき、新人プログラミング教育の質的向上につながる。

①与える演習課題の文章説明が拙劣

②上記の①を補足するためか、図示や例示等の不必要(過剰)な提示

①の文章表現の欠陥は必ずしも直に阻外要因というわけではない。ただし、不用意な用語不統一を正したりすること等は必要だ。受講者には、与えられた課題を自身の日本語で完成することから始めるように指導すべきである。実際のプログラムの仕事も、曖昧な要求仕様を明確な機能仕様として把握し記述することから始まる。筆者の経験から、『自分の言葉』はイメージ把握をも促進すると思える。この過程が、プログラミングに入るための『助走』になる。この段階は、慣れた日本語を活用すべきものであり、不慣れたプログラミング言語で進めるものではない。日本語を補佐するPAD等のツールの教え方も、清書用としてばかりでなしに、『考える』ツールとしての活用にもっと注目すべきである。

次に②は、受講者がイメージ思考を発動する契機をうばってしまう。受講者自身が図に表現したり入出力サンプル例を考えたりすることで、課題の理解が深まる。例示化も理解のプロセスとしての精緻化にあたり(【池田/村田 91】参照)、プログラミングに入るために有効な『助走』になる。

なお、上述の過程での演習課題等に対する疑問・質問には、インストラクタが対処すればよい。このQ&Aの過程こそ、真にプログラミングレベルの指導への第一歩である。

プログラミング演習指導で筆者自身も経験したことであるが、過剰な教示で受身になった受講者は不十分な『助走』で『プログラミングの峡谷』に飛び出し、その結末は『谷底』を迷走ということになってしまう。一方、①や②の阻害要因の克服で『考える』プログラミング演習への『助走』の機会を与えられた受講者は、見通しよく自主的にプログラミング演習が進められると期待できる。

4. おわりに

【菊池 87】は、企業内でのプログラマ育成経験をもとに、以下のようなプログラム設計業務上の問題指摘をしている。本稿での主張は、これらの問題点に対処するための基盤であると思う。

■文章題をイメージ化できない

■選択式の解答に馴れ、論理的な思考や記述式に弱い、または嫌やがる

■日常の体験や既習事項を設計に役立てられない

■論理の表記法(PADを使用)は知っているが使えない

■WHILEとUNTILの区別ができない。

■アルゴリズムがわからない

■トップダウン思考、詳細化の過程不明

例えば、UNTIL文(～まで)/WHILE文(～の間)の区別にとまどう原因も、単に前判定か/後判定かだけではなく、条件記述の相違[UNTIL(A=B) ⇨ WHILE(A≠B)]の認識しにくさにあると思う。これを単に言語仕様上の指導ですますのは、真のプログラミング教育の機会を見逃すことになる。

なお、プログラミング言語仕様教育と、知的スキル教育であるプログラミング教育との違いに対する認識不足の弊害は、次のレベルの技術教育にも現われている。例えば、ソフトウェア構造化設計技術の受講者の「実務の場で、どこでどのように使うのかピンとこない」というような意見がそれだ。この場合も、『考える』ことを介しての技術の体得が欠けているのである。必然的に、習った技術を実務で活用してやろうとするモラルも低い、ということだ。日本語能力を蘇生させ、かつ主体的に『考える』教育を早目に体験させておくことは、この意味でも不可欠である。

受講者の色々な疑問・質問に臨機応変に答えるのは、インストラクタにとっても大変なことだ。しかし、これなくしては、【木村/大野 93】が指摘するように「～実は全部の答えを知っている教師は、たちまち学生を小手先で操ろうとする不誠実さを見破られることになる。」ということだ。大変ではあるが、インストラクタにとって、力量アップのチャンスでもあると確信する。

参考文献

池田謙一, 村田完二: こころと社会 -- 社会認知心理学への招待, 東京大学出版会(1991)

二村良彦: プログラム設計法PAD/PAM, 情報処理 Vol.21 No.11(1984)

川合敏雄: PADプログラミング, 岩波書店(1985)

菊池尚弘, 他: 日立SK 企業内教育におけるプログラマの養成, 日本科学教育学会(1987)

木村 泉: ワープロ作文技術, 岩波書店(1993)

木村 泉, 大野浩之: ソフトウェア専門家を志す人のための基礎訓練科目, 情報処理学会論文誌 Vol.34 No.12(1993)

田口昭仁: A Study of Processes of Writing in a Non-Native Language

—Analogy to Warnier's Programming Methodology—,

情報処理学会 第29回全国大会(1984)

竹内郁雄: 翻訳 作文とプログラミング, コンピュータソフトウェア Vol.10 No.4(1993)