

口伝と体伝

アブストラクト

学びには「知っている」、「わかる」、「できる」、「教えられる」という段階がある。

学校の授業では先生から生徒に口伝で知識や技術を伝える。一方で、武道や芸術の世界では師匠から弟子に体伝で知識や技術を伝える。

それぞれの知識トランスファーの情報伝達方法には一長一短がある。口伝は形式知化できない暗黙知を伝えることができない。一方で、体伝では暗黙知は伝えるのに適しているが、形式知を伝えるのが難しくなる。

実際に口伝と体伝、双方を知識トランスファーの際に用いることでより効率的に知識や技術を伝授できるのではないかと考え、実践したところ、より効率よく知識や技術の伝達が行えた。また、自分以外に口伝と体伝、双方を知識トランスファーの際に用いている実例も見つけることができた[1]。

1. 私自身が実践した具体的な内容

業務や業務外での学生への教育において、私自身の持つ知識や技術を伝える数多くの機会を得てきた。私自身の持つ知識や技術を伝える際、次の2つの手法で主に伝えていた。

ひとつは講義やテキスト、仕様書などを用い、主に言語化や図式化できる知識や技術を文章、図表、数式などによって伝える手法である、以後本手法を口伝と記載する。もうひとつはお手本を見せたり、ペア作業を用い、主に言葉に表せない知識や技術を振る舞いによって伝える手法である、以後本手法を体伝と記載する。これらの手法で実際に、Rubyを用いたプログラミングやテスト駆動開発、継続的インテグレーション、継続的デプロイメント、アジャイルソフトウェア開発、スクラム、DevOps、モブプログラミングなど、ソフトウェア開発現場で用いる手法や概念を伝えてきた。

口伝と体伝、どちらかの手法だけで伝えるとうまく知識や技術が伝わらないことに気がついた。実際、口伝でテスト駆動開発を伝えると、テスト駆動開発についての理論や体系立てた知識は相手に伝わる。口伝で説明した後、相手に「わかった」内容を説明してもらうことで理解しているかどうかを確認した。実際に相手に説明させ、その内容を確認すると、説明している内容に大きな間違いはないことから「わかっている（知識や経験が伝わっている）」ことが確認できた。しかし、実際にテスト駆動開発を実践させると実践「できない」のである。一方、体伝でテスト駆動開発を伝えると、テスト駆動開発がどのように振る舞うものなのかという知識は相手に伝わる。体伝で説明した後、相手に「わかった」内容を実践してもらうことで理解しているかどうかを確認した。実際に相手に実践させ、その内容を確認すると、実践している内容に大きな間違いはないことから「わかっている（知識や経験が伝わっている）」ことが確認できた。しかし、テスト駆動開発を私に説明させると説明「できない」のである。

このことから口伝と体伝それぞれで相手に伝わる知識や技術が異なるのではないかと考えた。

- 口伝
 - 伝わる知識や技術：文章、図表、数式などの理論や体系立てた知識や技術である形式知
 - 伝わらない知識や技術：言葉に表せず、言葉で説明できない振る舞いの知識や技術である暗黙知
- 体伝
 - 伝わる知識や技術：言葉に表せず、言葉で説明できない振る舞いの知識や技術である暗黙知
 - 伝わらない知識や技術：文章、図表、数式などの理論や体系立てた知識や技術である形式知

そこで、相手にその事柄に関する知識や技術を伝える際、文章、図表、数式などの理論や体系立てた知識や技術を口伝で伝え、体伝で言葉に表せず、言葉で説明できない振る舞いの知識や技術を伝えることでうまくいくのではないかと考え、組み合わせて実践することにした。口伝が先、体伝が先、口伝や体伝の手法などによって伝達に差が出ることも考慮し、下記の組み合わせで実践した。

- Ruby
 - テキストによる自習の後にペアプログラミング
 - 講義の後にペアプログラミング
 - ペアプログラミングの後にテキストによる自習
- モブプログラミング
 - 講義の後にモブプログラミング
 - モブプログラミングの後に講義
- テスト駆動開発
 - テキストによる自習の後にペアプログラミング
 - 講義の後にペアプログラミング
 - ペアプログラミングの後に講義
- 継続的インテグレーション
 - テキストによる自習の後にペア作業
 - 講義の後にペア作業
 - ペア作業の後に講義
- Python
 - モブプログラミングの後に講義
 - テキストによる自習の後にモブプログラミング
 - 講義の後にペアプログラミング
- スクラム
 - 講義の後にペア作業
- DevOps
 - 講義の後にツールを用いたグループ作業
 - ツールを用いたグループ作業の後に講義

上記の組み合わせで口伝と体伝による知識や技術を相手に伝えることを実践したところ、口伝のみ、体伝のみに比べ、口伝の後に体伝、体伝の後に口伝で伝えた場合、伝えた知識や技術を説明でき、実践できる結果を得られた。また、口伝や体伝の各手法（テキストによる自習と講義、ペア作業とペアプログラミングとグループ作業とモブプログラミング）で違いを認識できる程大きな差はなかった。

これらの成功している（＝うまく知識や技術が伝わっている）実践から繰り返し見られる「パターン」を抽出したものが本論文にて提案するパターンであ

る。

2. パターンの説明

2.1 口伝と体伝

2.1.1 要約

自分の持っている知識や技術（形式知、暗黙知）をその知識や技術（形式知、暗黙知）を知りたいと望んでいる相手に伝授する。

2.1.2 状況

自分の持っている知識や技術（形式知、暗黙知）をその知識や技術（形式知、暗黙知）を知りたいと望んでいる相手がいるため、伝授したいと考えている。

2.1.3 問題

口伝の場合、「やり方を相手に伝え、相手にやり方を“知って”もらい、“わかって”もらう」、「実際に相手に実践させ、“できる”ようになるまで繰り返す」というプロセスを経るが、口伝は口伝で形式知を形式知として伝える情報伝達方法であり、暗黙知を伝えるのは不可能である。

一方で、体伝の場合、「やり方を相手に伝え、相手にやり方を“知って”もらい、“わかって”もらう」、「実際に相手に実践させ、“できる”ようになるまで繰り返す」というプロセスを経るが、体伝は実践による共同体験で暗黙知を暗黙知として伝える情報伝達方法であり、形式知を伝えるのは不可能である。

2.1.4 フォース

同調現象[3]、模倣[4]、ミラーリング[5]によって、自分が受けた情報伝達方法（口伝・体伝）で相手に伝えなければならない心理が働いてしまう。

2.1.5 解決方法

口伝、体伝のどちらかだけでは、すべての知識や技術を相手に伝えることは不

可能であることを理解する。口伝と体伝を組み合わせ、口伝で形式知を形式知として、体伝で暗黙知を暗黙知として相手に伝える。

2.1.6 結果状況

「1. 私自身が実践した具体的な内容」参照。形式知を形式知として伝える口伝、暗黙知を暗黙知として伝える体伝を同じ対象に順不同で行うことで、自分の持っている知識や技術の形式知と暗黙知を相手に伝えることができる。

2.1.7 適用事例

本パターンを適用したソフトウェア開発の現場では、口伝による知識トランスファーをしていた以前よりも効率的（短時間・短期間）に知識や技術を伝授できた。また、ソフトウェア開発の現場以外にもゴールドジムというジムでは口伝と体伝を組み合わせ、ジムへ筋力トレーニングをしにきた人たちに正しい知識と技術を伝えている[1]。

Appendix

A.1 References

[1] トレーニング サポート | ゴールドジム 公式サイト (Official Site) フィットネスクラブ・スポーツクラブ ゴールドジム
<http://www.goldsgym.jp/training-support>

[2] ナレッジマネジメント - Wikipedia
<https://ja.wikipedia.org/wiki/%E3%83%8A%E3%83%AC%E3%83%83%E3%82%B8%E3%83%9E%E3%83%8D%E3%82%B8%E3%83%A1%E3%83%B3%E3%83%88#SECI%E3%83%A2%E3%83%87%E3%83%AB>

[3] 同調現象 - Wikipedia
<https://ja.wikipedia.org/wiki/%E5%90%8C%E8%AA%BF%E7%8F%BE%E8%B1%A1>

[4] 模倣 - Wikipedia
<https://ja.wikipedia.org/wiki/%E6%A8%A1%E5%80%A3>

[5] ミラーリング - Wikipedia

<https://ja.wikipedia.org/wiki/%E3%83%9F%E3%83%A9%E3%83%BC%E3%83%AA%E3%83%B3%E3%82%B0>