

アジャイルの振り返りとシステム・シンキングの有効性について ～ロジカル・シンキングは万能ではない～

日山 敦生
緑ビジネスコーチ研究所
hiyama@k04.itscom.net

要旨

アジャイル開発の振り返りには、フレームワークの KPT がよく使われる。問題の分析手法であるロジカル・シンキングは、技術的な問題には非常に有効であるが、万能ではない。人や組織の問題には、システム・シンキングを基本とする解決志向アプローチが、有効である。原因を究明しない解決志向アプローチを用いることにより、チームの強味の再利用や、問題の迅速な解決が容易となる。アジャイル開発の振り返りに、解決志向アプローチを用いることで、設計品質のさらなる向上に、貢献できる。

1. はじめに

近年、益々、巨大化、複雑化するソフトウェアに反比例して、今まで以上に短納期が求められる。複雑化するなかで、要件定義を開発初期段階で、明確に定義することが困難な状況がある。そこで、反復開発を行うアジャイル開発が注目されている。アジャイル開発は、反復開発のサイクル毎に、振り返りを行うことにより、設計品質を高めていくことができる。この振り返りで使われているのが、振り返りのフレームワークである KPT である。振り返りでは、技術的な問題だけではなく、人や組織の問題についても、振り返りが行われている。

技術的な問題については、原因究明を基本とするロジカル・シンキングが有効であるが、人や組織の問題については、原因究明を行わないシステム・シンキングが有効である。日本では、幼いころから社会人に至るまで、問題を解決するためには、原因を究明し、原因を取り除くことによって、問題は解決できると学んできた。バグのような技術的な問題には、原因究明を基本とするロジカル・シンキングが有効であるが、人や組織の問題は、複雑で、原因と結果が相互に悪循環になっているとするシステム・シンキングが、問題解決に有効である。

心理療法から生まれたシステム・シンキングを基本とする、解決志向アプローチがある。解決志向アプローチを、ビジネスの人や組織の問題解決に用いると、迅速に、問

題解決ができる。

また、良かったことは、項目としてあげるだけではなく、深堀して、再利用することが重要と考える。苦勞して上手くできたことを、再利用しないで使い捨てにしてしまうのは、もったいないと考える。

アジャイルの振り返りのフレームワークである KPT に、システム・シンキングを基本とする解決志向アプローチを適用した振り返りのフレームワーク ROTT を考案し、ワークショップを開催しているので、ご紹介する。

2. KPT と ROTT

2.1. KPT とは

KPT (図 1) は、アジャイルの振り返りでよく使われるフレームワークであり、Keep: 良かったこと、Problem: 悪かったこと、Try: 次回、試してみることで、構成されている。

KPT は、悪かったことだけではなく、良かったことにも注目する振り返りの優れたフレームワークである。

KPT は、ロジカル・シンキングの視点から見ると優れたフレームワークであるが、システム・シンキングを基本とする解決志向アプローチの視点からみると、残念な点が2つある。1つは、良かったこと(Keep)の深堀が、不明確であること。もう1つは、悪かったこと(Problem)が、物や技術の問題と、人や組織の問題に分かれていないことである。そこで、KPT の改良版として ROTT を提案したい。

Keep	Try
Problem	

図 1 振り返りのフレームワークの KPT

2.2. ROTT とは

KPT に、解決志向アプローチのキーワードである「成功の責任追及」を用いたフレームワークとして、ROTT(図2)を考案した。ROTT は、Recycle:良かったこと(成功究明, 目的究明), Organizational Problem:組織的問題(解決究明), Technical Problem:技術的問題(原因究明), Try:次回, 試みることで、構成されている。

良かったことは、Keep ではなく、Recycle とする。Keep には、維持するという意味合いが強く、深掘して再利用するという意味合いが乏しい。苦勞して出来たことを深掘せずに、次回も、ゼロから始めるのは、もったいないと考える。Problem は、解決究明が重要な人や組織の問題(Organizational Problem)と、原因究明が重要な技術的問題(Technical Problem)に分ける必要がある。解決究明と原因究明では、問題解決のアプローチが大きく異なる。

Recycle	Organizational Problem
Try	Technical Problem

図 2 振り返りのフレームワークの ROTT

ROTT の作業手順(図3)は、一度に、③組織的問題と④技術的問題に分けることが難しいので、②悪かったこととして、一度、項目出しを行ってから、分離作業を行う。

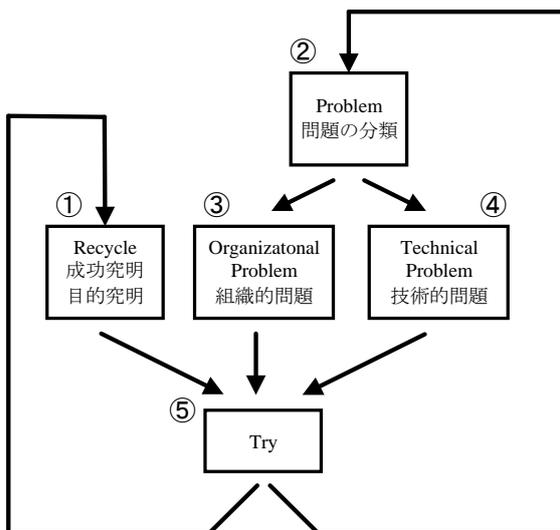


図 3 ROTT の作業手順

3. ROTT

3.1. Recycle

良いことは維持するのではなく、良かったことは深掘を行い、良い循環を強化することである。良い循環を強化するキーワードとして、「成功の責任追及」がある。失敗の責任追及があるように、成功の責任追及があつて当然と考える。失敗の責任を追及されることは辛い、成功の責任追及は、自慢話を聞かせてくださいということで、いくら追及されても辛くはない。ヒヤリ、ハットしたことも、最終的にカバーできたということであり、ヒヤリ、ハットしたことをどのようにカバーできたかの、貴重な対策資料となる。

良かったことは、2 種類に分類できる。抽象的なことと具体的なことである。抽象的な例として、「顧客と信頼関係が築けた」、具体的な例として、「議事録を書くことができた」を取り上げる。

抽象的なことである「顧客と信頼関係が築けた」(図4)の成功究明は、どのような自分たち行動が、顧客の信頼関係を築くことに、貢献したのかの究明を行うことである。顧客の信頼関係を築くために、有効であった行動が明確になれば、既に自分たちが出来ていることなので、次回も、有効であった行動を容易に繰り返すことができる。

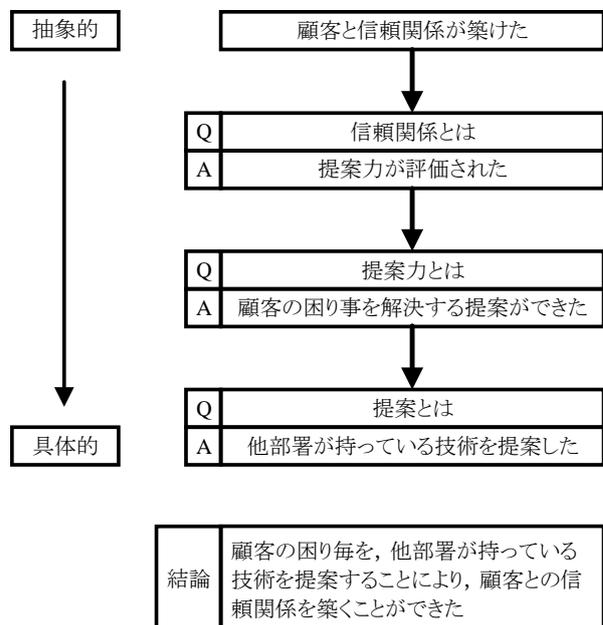


図 4 顧客と信頼関係が築けた

具体的なことである「議事録を書いた」(図5)であれば、

目的究明を行い、仕様に関して顧客と相違が生じた場合に、自分たちが正しいことを証明する目的が、今回、有効に機能した。さらに、深堀すると、顧客に勝つことが目的ではなく、仕様に関して顧客と相違が生じないようにするために、何が必要かという問題が浮かび上がる。

判りにくい所には、具体例を入れた仕様書にするという対策が生まれる。

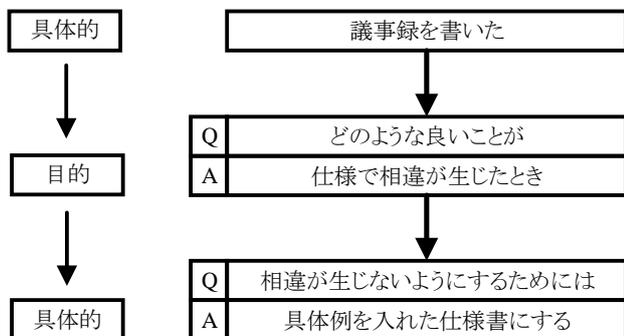


図 5 議事録を書いた

このように、良かったことは、項目としてあげるだけではなく、再利用することが重要だと考える。成功究明は、すでに出来ていることであり、目的究明は新たな問題に気づく可能性があるが、簡単に再利用できるにも関わらず、再利用ができていない。

3.2. Problem

問題は 2 種類に分類できる。ソフトのバグのような技術的問題と、チームワークが悪いというような人や組織の問題である。

問題の原因と結果に注目すると、技術的な問題は、原因と結果が直線的に結びつく、直線的因果律である。人や組織の問題は、原因と結果が悪循環になっているとする円環的因果律がある。

直線的因果律の例として、パソコンから資料が印刷できないという問題(図 6)を取り上げる。この問題では、原因の切り分けを行っていくことになる。第1段階として、パソコンとプリンタで、どちらに原因があるかを分析し、パソコンではなく、プリンタに原因があると判明したとする。次に、プリンタのどこに原因があるかを分析するとインクがなかったと判明する。したがって、インクを交換することにより、この問題は解決できる。

直線的因果律の特徴は、原因と結果に時間遅れがないこと、原因究明を行い、原因を取り除かない限り、問題が解決できないことである。真の原因は何かということが、

重要となり、ロジカル・シンキングが有効である。

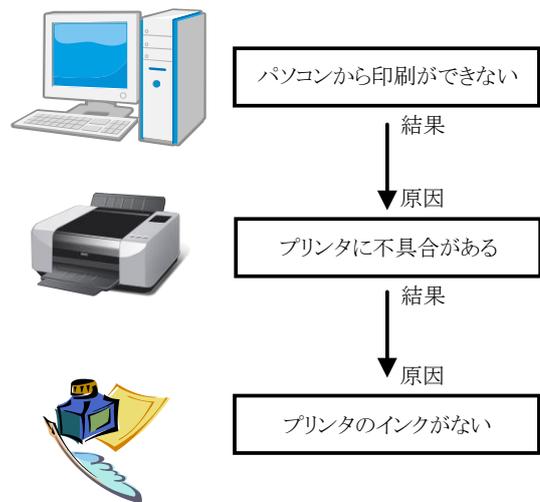


図 6 直線的因果律

円環的因果律の例として、2008 年に起きたリーマン・ショックを契機とした世界的金融危機によって、不況になり商品が売れない問題(図 7)を取り上げる。

この問題は、図 7 のように、アメリカで金融危機による不況が起きると、日本の消費者の消費マインドが冷え込んで、商品を買わなくなる。消費者が商品を買わなくなると、企業では売上げが下がり、経営者がすぐに行う行動は、研究開発費と研修費の削減となる。研修部門では、研修費を削減されて、社員の研修が出来なくなる。技術部門では、社員の研修が行われなくなるので、やがて技術力が低下して、魅力的な商品の開発ができなくなる。消費者は、魅力的な商品が提供されないの、物を買わなくなるという悪循環になる。

円環的因果律の特徴は、原因と結果に時間遅れがあり、関係者全員が原因であり、結果でもある。原因と結果に時間遅れがあるとは、経営者が研修費を削減して、社員研修を止めても、すぐには、技術部門の技術力低下には、繋がらないということである。関係者全員が原因であり、結果でもあるとは、例えば、図 7 の経営者の視点から原因を追究していくと、消費者、技術部門、研修部門、経営者の順に原因をさかのぼることができ、原因が自分に戻ってくる。同様に、経営者の視点から、結果をさかのぼっていくと、自分に戻ってくる。人や組織の問題で、原因究明を行うと、原因が判らなかつたり、犯人探しになりがちで、問題解決には有効とは限らない。したがって、原因究明しないシステム・シンキングが有効である。

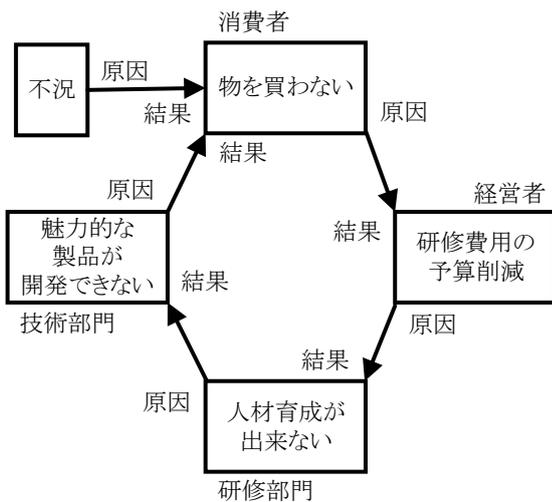


図 7 円環的因果律(その1)

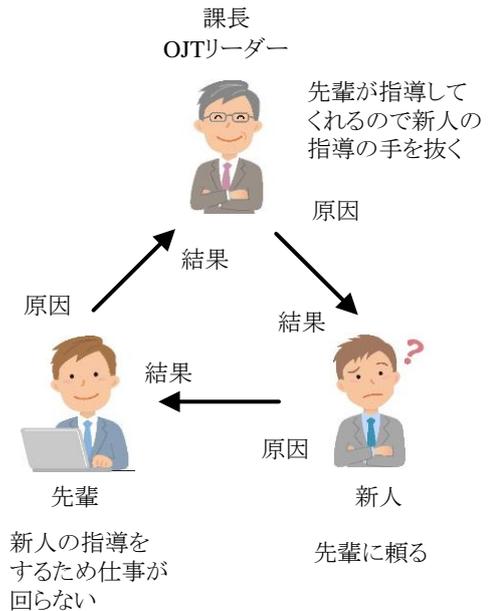


図 8 円環的因果律(その2)

もう1つ円環的因果律の具体例として、OJT の問題(図8)を取り上げる。OJT リーダーである課長が、忙しくて新人の指導ができない。先輩は、指導してもらえない新人を可哀そうに思い、代わりに新人を指導しているため、先輩の本来の仕事が回らなくなっているという問題を取り上げる。

先輩が新人の OJT を指導すればするほど、課長は、新人の OJT の指導をしなくて済んでしまう。課長が OJT の指導をしてくれないので、益々、新人は先輩に頼ることになる。この関係者(課長、新人、先輩)のそれぞれの行動が、問題を維持する要因になっている。

この問題では、それぞれの責任の度合いは異なるが、このように関係者全員(課長、新人、先輩)が原因でもあり、結果でもある。

円環的因果律の問題を解決する方法は、2つある。1つは、悪循環を断つ「MRI アプローチ」であり、もう1つは、この悪循環に注目するのではなく、稀に起こる良い循環に注目する「解決志向アプローチ」である。

MRI アプローチの具体例として、この悪循環を断ち切る方法の1つは、別の課長のもとに、新人を人事異動することである。MRI アプローチを用いる場合は、犯人捜しとならないように、関係者への配慮が必要となる。

通信プロトコルの OSI 参照モデルを参考に、人や組織の問題解決レイヤーを考えると図9のようになる。

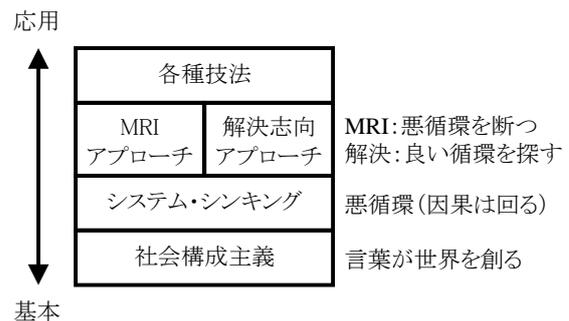


図 9 問題解決レイヤー

3.2.1. Organizational Problem (組織的問題)

人や組織の問題を組織的問題(Organizational Problem)と名付けた。人や組織の問題は、原因と結果が円環的因果律となっていて、原因究明が有効とは限らないことを説明した。システム・シンキングを基本とした解決志向アプローチを用いた問題解決法について、説明する。

解決志向アプローチは、原因究明ではなく、解決究明に焦点をあてる。出来ていないところではなく、出来ているところに焦点をあてる問題解決法である。

解決志向アプローチの問題解決方法の例として、遅刻防止対策を考えてみる。会社に遅刻しないように出社したいと思っているが、よく遅刻してしまう。

原因究明型の問題解決法を問題志向アプローチと呼ぶ。この問題志向では、遅刻した日に焦点をあて、遅刻した原因を究明して、原因を取り除く問題解決法である。出来ていないことを出来るようにするためには、時間も費用も掛かる方法である。

解決志向アプローチでは、遅刻した日ではなく、遅刻しなかった日に焦点をあてる。どのような条件が揃うと、遅刻しないかを明確にすることである。遅刻しない条件が明確になれば、既に出来ていることなので、遅刻の大幅な減少が期待できる。

人や組織の問題を解決する解決志向アプローチの質問技法には、4つの質問技法があり、2種類に分類できる。過去のリソースを引き出す質問と未来を創造する質問である。

過去のリソースを引き出す質問は、

- ・例外探しの質問
 - ・サバイバル・クエスチョン
- である。

未来を創造する質問は、

- ・スケーリング・クエスチョン
 - ・ミラクル・クエスチョン
- である。

例外探しの質問は、問題の中にある稀に上手くいった時に焦点をあて、上手くいった時の対処法を繰り返すことにより、問題の解決を行う。

サバイバル・クエスチョンは、問題を抱えている当事者を勇気づける質問である。

スケーリング・クエスチョンは、目標を10点満点としたときに、現状は何点で、10点満点にするためには、何が必要かを考える質問である。

ミラクル・クエスチョンは、夜、寝て、朝起きるまでに、奇跡が起こって、目標が達成できていたら、どのような一日を過ごすかを聞く質問である。

4つの質問の中で、例外探しの質問が、人や組織の問題を迅速に解決するためには、一番有効である。

例外探しの質問は、上手く行っていることに注目する質問で、上手く行っていない状況の中でも、上手く行っている事はある。いつも上手く行かないと思っている場合(図10, 図11)でも、気づかないだけで、上手く行ったときがある。気づかないのは、人は、問題であることを認識すると、問題が起こった時に関心が集中し、上手くできなかった時は、よく気づくようになる。ところが、稀に、上手く対

処できた時があると、問題が起きないので、気づかないことになる。

例外探しの質問手順は、図12のようになる。

- ・1回ぐらい上手く行った時は、ありませんか
- ・1回毎に、上手く行った事があるか、確認し見つかるか
- ・上手く行ったことが見つからなかったら、1回毎に点数化してみて、一番点数の高い時はいつか
- ・上手く行った時は、何が違ったのか

1回毎に、上手く行った事がないか、確認する質問では、現在から過去の方に、確認していくことが重要である。現在から過去の方に、確認していくことにより、上手くいった時が見つからないときに、記憶が怪しくなったところで、質問をやめることができる。最近の方が、記憶が明確に残っているのだから、上手く対処するための行動が、明確になる。

例外探しの質問の特徴は、現状把握する時間を最小にして、短時間で問題解決できる可能性がある。解決志向アプローチでは、いくら問題を詳しく聞いても、問題は解決できない。上手く対処できた時を、丁寧に探すことが重要となる。私の経験では、1対1のコンサルティングでは、現状を詳しく聞くことなく、10分程度で、上手く対処できた時を見つけ、解決できている。

例外探しの質問は、短時間に人や組織の問題を解決できるよい方法であるが、ビジネスの現場で用いる場合には、大きな問題を抱えている。

解決志向アプローチの問題解決法は、原因を究明しない問題解決法である。解決志向アプローチのワークショップをビジネス向けに行っているが、参加者の感想として、「解決志向アプローチの有効性は理解できるが、原因を究明しない問題解決法は、職場では理解されないので使えない」という感想を、いただくことがある。解決志向アプローチをビジネスで用いる場合は、ステークホルダーに解決策を説明する必要がある。日本では、幼いころから社会人にいたるまで、問題を解決するためには、原因を究明し、原因を取り除くことによって、問題は解決できると学んできた。したがって、原因究明しない問題解決法を、受け入れることが難しい。対策として、解決志向アプローチでは、解決策が先に出てくるので、解決策に相応しい原因を後から考えて、ステークホルダーに伝えることで、問題を解決できる(図13)。解決策に相応しい原因は、多くの場合、解決策の否定を原因とすればよい。注意点として、原因は人ではなく、事によることである。

当事者の認識(いつもだめ)						
日	月	火	水	木	金	土
×	×	×	×	×	×	×

図 10 当事者の認識

実際(出来ている時がある)						
日	月	火	水	木	金	土
×	×	×	○	×	×	×

図 11 実際

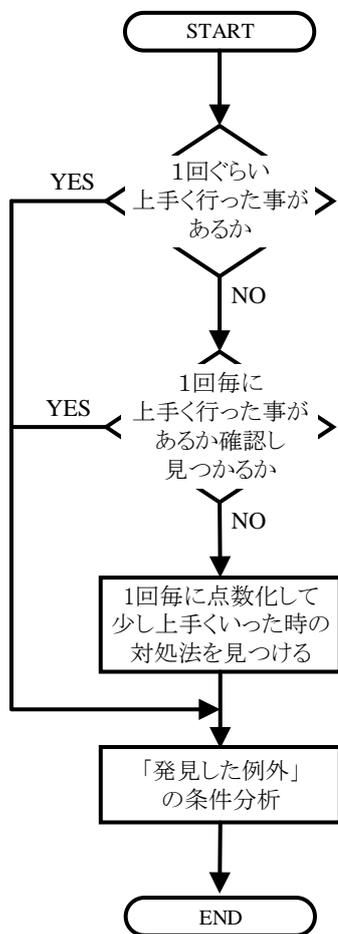


図 12 例外探しの質問手順

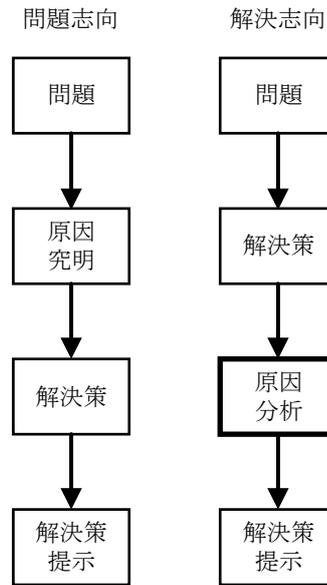


図 13 ビジネス向け解決志向アプローチ

3.2.2. Technical Problem(技術的問題)

技術的問題は、原因究明が重要となるので、なぜなぜ分析をはじめとするロジカル・シンキングが、有効である。

3.3. Try

Recycle, Organizational Problem, Technical Problem で行った分析結果から得られた対処方法を記述し、良かったことも、悪かったことも、無駄なく、次回に活かすことができる。日本では、特に、良かったことを深掘して、再利用するということが少ないように思う。

4. 結果と考察

アジャイル開発の振り返りのフレームワークである KPT の進化系として、ROTT をビジネス向けワークショップで、紹介してきた。参加者から、ワークショップの満足度に関するアンケートで、高い評価をいただいている。

日本のビジネスは、弱点克服に偏っていると感じている。弱点克服と同様に、強味も活かすことが重要だと考えている。特に、ソフトウェア業界では、問題といえば、バグ対処である。バグは、原因究明が必要な直線の因果律で、原因を取り除かない限り、問題は解決できない。そのため、

人や組織の問題においても、真の原因は何かという方向に、なりがちである。失敗から学ぶと同様に、成功からも学ぶことが重要だと考える。

今回、問題を2種類に分類したが、両方の問題解決法が必要となるようなバグもある。例えば、ウォーターフォールモデルのシステムテストで、発生したバグの原因分析をした結果、上流工程のシステム要件定義に、問題があることが判明した。システム要件定義からの対応となり、バグ修正の工数が大きくなり、開発体制の強化が必要となった場合である。ステークホルダーに働きかけて、組織を動かすことが必要となる。この場合、原因究明型のロジカル・シンキングでバグ分析を行い、次に、組織を動かすために、システム・シンキングを基本とする解決志向アプローチを用いると効果的である。

解決志向アプローチは、人や組織の問題であれば、振り返り以外にも、幅広く使うことができる。問題を起している関係者自身が、解決志向アプローチを使って問題解決することが、容易ではない場合がある。問題の関係者は、問題の悪循環に取り込まれているので、客観的に悪循環を分析して、悪循環から抜け出す事は容易ではない。第3者である上司、同僚、コンサルタントからの、解決志向アプローチを用いたサポートが効果的である。

人をまとめることが仕事であるリーダーやマネジャーは、システム・シンキングは必須のスキルだと考える。ところが、日本では、ロジカル・シンキングと比較するとシステム・シンキングは、あまり知られていない。google 検索エンジンを使って、2018年4月18日現在の知名度を比較した。システム・シンキングは、システム思考と呼ばれることも多いため、「システムシンキングとシステム思考」、「ロジカルシンキングと論理思考」で検索した。検索結果は、ロジカル・シンキングが、463,000件となり、システム・シンキングが、94,200件となった。百分率で表すと、ロジカル・シンキングは、83.1%、システム・シンキングは、16.9%となった。

日本のビジネスに、ロジカル・シンキングと同様に、システム・シンキングや解決志向アプローチが、広まることを願っている。

参考文献

- [1] 天野勝, これだけ! KPT, すばる舎
- [2] 中原淳, 長岡健, ダイアログ 対話する組織, ダイヤモンド社
- [3] 西村行功, システム・シンキング入門 (日経文庫), 日本経済新聞社
- [4] 森俊夫, 黒沢幸子, 森・黒沢のワークショップで学ぶ 解決志向ブリーフセラピー, ほんの森出版