

アジャイル型 SD：迅速で有効で信頼できるやり方 訳注1)

Agile SD: fast, effective, reliable.

キム・ウォーレン (Kim Warren) 著
Strategy Dynamics Ltd,
Two Farthings, Aylesbury Rd, Princes Risborough,
BUCKS HP27 0JS, U.K.
+44 1844 274061
kim@strategydynamics.com

末武 透 訳
日本未来研究センター研究員
ts178051@yahoo.co.jp

私自身は、新しい問題に出会うたびに、最初に何がストックかを明らかにし、そのストックが量的にどのように変化するかという観点でその問題を考えている。因果関係図が有名であることは知っているが、私自身はモデル開発で因果関係図を使っていない。

—Forrester, 2013

キーワード：モデル構築、方法論、アジャイル型開発法、ウォーターフォール型開発法、システム・ダイナミックス

要旨：システム・ダイナミックスの進歩やインパクトについての最近の関心は、方法が展開されることによる意味について疑問を投げかけることである。書籍や教育コース、あるいは出版されている事例では、関心のある問題が時間と共にどのように変化するかを定義することによって開始し、次に利害関係者と一緒に定性的な因果関係図を構築する。この作業の結果となる、関係者と共に共有されるメンタルモデルには、取り扱う問題の範囲と関係する要素の因果関係の仕組みを含んでいる。次に、ストック・フロー構造がこの因果関係図のモデルに追加され、データを勘案してこの構造が数式定義され、数学的な動くモデルとして完成する。このモデル構築プロセスは難しく、時間もかかり、しかもあまり信頼できない。間違いを犯し、何かを見逃すリスクがあり、類似した対象でも違ったモデルを作成してしまいかねない。方法の科学は、より簡易なプロセスを示唆している。この方法は、実績値の変化から直接的にストックとフローが数量的にどのように変化するかをマッピングすることである。このことを手掛かりに、やはり数量化した変化をもとに相互関係を追跡し、重要なフィードバックの構造を明確化していく。この方法をいろいろ試した結果、この方法の方が簡単で、早くモデルを構築でき、時間や労力を大幅に短縮でき、モデル開発の開始時から作業品質を保てることが分かった。この方法からも貴重な洞察を得ることができる。この方法は、今やソフトウェア開発では主流になっている「アジャイル（俊敏）」アプローチを連想させる。この方式による開発を行ってみて十分メリットがあることが分かっているので、もっと広められ、多くのモデル開発者によりその便益性を確認されるべきと考える。この方法も補完的アプローチとして整合性があり、多くの先駆的な SD 実践家の中で一般的であり、類似した事例に渡って繰り返して証明された構造をより活用するものである。

近年、環境、保健、経済、経営、安全保障など広くさまざまな分野での、その分野の一人者による政策決定で、システム・ダイナミックスが認知され、採用され、現実の世界での影響が高まっていることに伴い、懸念事項も増えている(Forrester 2007; Homer 2013)。特に、国際システム・ダイナミックス学会の年次国際会議に提出された論文から鑑みると、この問題は、不適切なモデル構築方法と相まって、高い品質の作業がなされていないことに原因があると思われる。しかし、基本的に高品質のモデルを作成することは可能であり、高品質のモデルも事実構築されている。スキルの高い専門家が作成した高品質のモデルは、それを広めるチャネルが無く、また、優れた仕事を出版によって広く知らせるインセンティブもないことから、そういった高品質の仕事のごく一部のみが知られているに過ぎない。言うまでもなく、システム・ダイナミックス関係の専門家と話し合ってみると、「システム・ダイナミクスを売り込む」ことの難しさをよく聞かされる。最初は売り込みに成功しても、やがて、ユーザーは、懷疑的になり、幻想を感じるようになり、拒絶するようになるという。

他の改善方法論は、影響のあるユーザーからのこのような反応を避けているように見える。従って、導入も急

速であり、利用者も急速に広がっている。これらには、バランスト・スコアカードなどの比較的単純なものもあるが、もっと複雑な、付加価値経営、システム工学、6 シグマ、ビジネス・プロセス・エンジニアリングなどのようなものもあるが、これらは、このわずか 10~20 年間で急速に広まり、経営改善の主流を成す方法論になっている。それなのに、なぜ、システム・ダイナミックスは、これらと同じように歓迎されなく、これらと同じように適用され、影響を与えることができないのであろうか？

方法が採用されるには、明確で表現可能な利便性があること、費用対効果の点で、妥当な費用をかけて、妥当な成果を得られること、実施方法が信頼できることの 3 つの基準があると考えられる。実施方法が信頼できるという「信頼性」は、その方法が自信をもって同様の状況に展開され、似たような効果が得られるということである。認知が低く、適用もそれほど急速には広がらず、効果発揮にも限界があることから、システム・ダイナミックスは、この 3 つの基準に照らし、方法として疑問を持たれるのであろう。

成熟した専門分野で活躍する専門家は、いわゆるベスト・プラクティスと呼ばれている、確実に、適切な費用で十分な成果を上げられるようにするための標準的な作業手順を導入している。プロフェッショナルな協学組織は、そのようなベスト・プラクティスに関する代表的なものを出版している。(例えば、サプライ・チェーン委員会は SCOR Framework を、バランスト・スコアカード協会は Nine Step to Success を、INCOSE は、システム・エンジニア・ハンドブックを出版している。)

システム・ダイナミックスのモデルをどのように構築したら良いかについての綿密に定義された手続きは存在しないが、主要な文献に示されるガイドラインは、比較的類似性を示している (Lane 1994, Vennix 1996, Sterman 2000 p.86, Maani and Cavana 2000, Morecroft 2007, Pruyt 2013)。また、モジューラ・アプローチのような代替的方法 (Wolstenholme 1994) も存在するが、学会発表や出版された論文などから推測するに、概ね先のガイドラインのようなものに従ってモデルを開発していると思われる。しかしながら、重要な点は、かなりの SD 専門家は少し異なる方法を用いてモデルを構築していることである。彼らはしばしば、以前に証明された構造を、それと似たようなケースに対して採用している。後でこの点について述べるが、このことには、SD モデルの構築や適用に関する重要な洞察が含まれている。

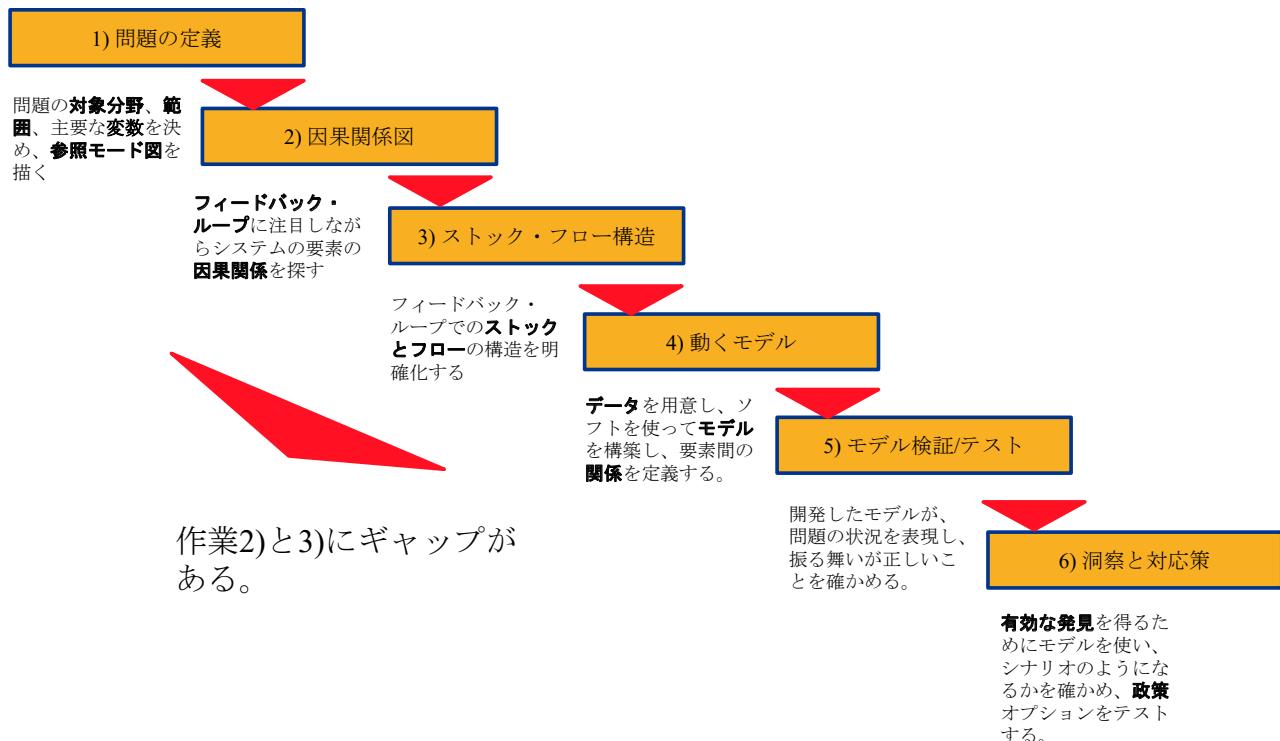


図 1 単純化した一般的なシステム・ダイナミックスのモデル構築手順

図 1 に概要を示しているように、一般的に採択されている方法では、最初に、問題の対象範囲を決め、そうなるであろうという出力結果を明確化する。この作業では、出力が時間によってどのように変化し、将来はどのようにになるかをグラフにプロットしていく。この作業で数量化を行っている。出力となる指標が数値的にどのように

な値を取るかを決め、時間経過でどれだけの変化量が変化するかを決め、その結果を「参照モード(Reference Mood)」として定義する。この参照モードによってモデルの結果を評価し、シナリオを展開し、政策オプションの検証を行う。この作業では、複数の出力値を考慮する。例えば、何か悪影響を及ぼす要因を改善するようなことを考え、その要因の改善費用を考えるので、参照モードを示す時間挙動図は一枚では済まなく、数枚記載しなければならない。註注2)

ステップ2では、問題の関係者に対して、対象としている要因の原因になっているものが他にないかどうか、因果関係がその2つの要因間に存在するかどうかを尋ねる。SD専門家は、このステップでは特に、因果構造の中からフィードバック・ループを発見することが重要であると強調している。これらの作業結果や参加者の見解は、一つの因果関係図(CLD:Causal Loop Diagram)として統合され、これが、対象としている出力がどのようにシステム構造の影響を受けるかを表現した、関係者の間で共有されるメンタル・モデルとされる。

ステップ3では、何かを累積するようなストックを先の因果関係図の中から発見し、それを明示化した図として表現する。SDモデルの構築者によつては、最初に因果関係図を描く際に、すでにストックとフローを認識している。しかし、ストックとフローをどのように認識するかをうまく説明することは難しい。また、場合によれば、ステップ2とステップ3の作業を合わせて1つのステップとして実施していく、ファシリテーターによつては、ストックとフローを含んだ因果関係図を最初から描かせようとしている。この方法にはいくらかメリットがあるものの、作業が大変であることには変わりがない(Lane D.C., 2008, Fisher 2010)。

いざれにしろ、ストックとフローを含んだ因果関係図が、ステップ4で行う作業の元になる。ステップ4では、実データを探し、数式で要素間の関係を定義し、これによって定量的に動くモデルが開発される。簡単にできる部分もあるが、このステップは、モデル開発プロセスの中で一番技術が要求される部分である。

次に、参照モードとモデルの振る舞いを比べ、モデルが参照モードのような振る舞いを示すかどうかを確認し、次に、代替シナリオや代替政策のオプションをさまざまに試してみるというモデル・テストも含め、さまざまなやり方でモデルの検証を行う。図1では、モデル構築プロセスが一方通行の線形プロセスであることが示唆されるが、Sterman (2000) は、モデル構築は繰返しプロセスであり、ある局面で得られた教訓から、再び初期段階のステップからの結果を訂正することもあることを強調している。しかし、これまで発表されている研究作業からは、このような相互作用が一般的に実行されているか否かは明らかにされていない。

標準的な構築方法の問題点

図1にまとめたモデル構築プロセスは、実際にSDモデルを開発している関係者が原則的に採択している方法である。しかし、注意深く見てみると、このようなやり方を効果的に信頼できるやり方で進めていくことはそう簡単ではないことが示唆されている。つまり、初期の段階では純粹に定性的な方法で作業進めていくにも係らず、このモデル構築の開始時点で問題の全体像をマッピングし、それを元に、段階的な方法でモデル構築を進めようとしている点で、(繰返しプロセスが追加されるか否かに関わらず) 最初に全部を完全に把握するような「海を沸騰させようとする(boil the ocean)」試みという、この分野でよく知られて受け入れられている重要な問題を無視していることによる。

ステップ1では、対象としている変数出力を直接的に数量化し、グラフで表すことを行っている。しかしながら、この段階では、グラフ表示の対象としている変数がストック(例えば魚の生息数)であるのか、あるいは数個のストック等によって影響されるが、それ自身はストックではない変数(例えば、漁獲高は、漁獲量に販売価格を掛けたもので、変数ではあるがストックではない)なのかを区別しているわけではない。経験を積んだSDのファシリテーターであれば、対象としたものがどちらかなのかはすぐに分かることは言うまでもないのだが、そのやり方を知らない初心者には区別がつかなく、さらに、この違いの説明が難しく感じられるので、概念が複雑すぎるという拒否感を持つてしまいがちである。

ステップ2の作業にはもっと問題がある。注目している変数の関係を探すというのは完全に定性的なやり方で行われるにも係らず、その関係を把握するための方法として、注目している変数の時間変化をグラフに示すという定量化作業を行っている。関係者は、そのような関係があることを示す証拠があろうとなかろうと、要素Bは要素Aに依存して決まるといった、彼らの見解を述べる。言うまでもなく、関係者の役割は、要素の関係も含めて、システムにおいてそれが妥当かどうかの判断を下すことである。しかし、システム・ダイナミックスの研究で分

かつた公理のようなこととして、人間の認知能力ではフィードバックが存在するシステムがどのように動くかを理解できないという点であった(Sterman 1989, 1994, Paich and Sterman 1993, Moxnes 2000)。さらには、人間の認知では認知対象を極めて主観的に選び、さらにバイアス（偏見）を持って認知してしまう(Kahneman, Slovic and Tversky 1982, Haselton, Nettle and Andrews 2005)。従って、個々の持つメンタル・モデルを合成したものが表しているものは、実際のシステムの動きを信頼できるレベルで表現しているとはとても言えない。

この定性化作業では、参加者間の社会的、権力的な関係が因果関係図の構造やモデルの結果に影響を及ぼすというリスクをもたらしかねない。重要な要素や重要な関係が、それに関する知識についてだれも尋ねなかつたからとか、参加者が、自分の持っている知識や見解を発言することを躊躇したとか、発言しても無視されてしまったといった単純な理由から取り込まれないこともある。機能障害を引き起こすだけのしろものであるような事項にも係らず、会議の席で認められたという理由だけで、モデルに組み込まれる事項もある。

ステップ2の作業での2番目の問題は、どのようなシステムであれ、ストックとして累積されるような要素があり、累積するという振る舞いが見られるにも係らず、この段階では、ストックとフローの区別をしないという点である。(もちろん、この代替的な方法として、最初から、因果関係図を構築する際に、ストックとフローを明確化するというやり方も提唱されているが、この方法を採択している人はほとんどいない。) これも、SD研究の成果として知られている公理のようなものだが、人々は、ストック・フロー構造の振る舞いを信頼できる水準で予測できない(Cronin, Gnozalez and Sterman 2009, Cronin and Gonzalez 2007, Sterman 2010)。しかし、どのようなフィードバック・ループであれ、最低1個は累積ストックを含まなければならないので、そのようなループでは最低1個は、人々が理解できない因果関係を含んでいることになる。因果関係図ではエージング・チェーンをあまり表記しないのだが、因果関係図の作成プロセスでストックを考えないということは、ストックにエージング・チェーンの要素が含まれている場合、そのようなストック要素がシステムの振る舞いに悪影響を及ぼすので特に危険である。過去10年間のSDR: System Dynamics Review誌上に掲載された研究論文には、エージング・チェーンのことを区別して取り扱っている因果関係図は全く見られない。エージング・チェーンのことを因果関係図に全く示していないにも係らず、その因果関係図を基に後で開発されたストック・フロー図には、エージング・チェーンが見られる。この点も含め、推奨されている、そして広く採択されているモデル構築方法(Morecroft 1982, Richardson 1986)には、現場で長く使われ、基礎的な知識となっている事項が欠落している。

3番目に、ステップ2の作業は、フィードバック・ループとそれらの振る舞いとの間の関係、つまり増強フィードバック・ループはさらなる増加あるいは減少を生成し、均衡フィードバック・ループは目標収斂型(goal-seeking)の振る舞いを生成するという関係の確かさに依存している。しかしながら、どちらのタイプのフィードバック・ループも、他の因果メカニズムの存在のため、そのような振る舞いを生まないこともある。また、他のメカニズムで、あたかも先のフィードバック・ループが生成するような振る舞いが生成されることもある。例えば、成長が加速されるという振る舞いは、要素が閾値を越えて線型的に変化することで起きる。この例として、製品の性能や価格が潜在顧客に魅力的に感じられ製品普及が進んでいく状況や、仕事がだんだんきつくなり、従業員がどんどん辞めていくといった状況がある。このような状況は、フィードバック・ループを使って表現できるが、単独に、フィードバック・ループを使わず、製品の価格を継続的に変えるとか、授業員の労働負荷の高まりを放置するといったことだけでもこのような状態が起きる。

問題の当事者は、システムのフィードバック構造を調べ、システムの振る舞いを直観的に推測できるわけでもなければ、振る舞いの結果、すなわちステップ2のデータの欠如によって悪化した問題を調べ、システムのフィードバック構造を直観的に推測できるわけでもない。

4番目に、このステップ2の作業結果の妥当性の判断は、記録や数値情報を使わない、参加者の「メンタルなデータベース(記憶にある情報)」を信用して行われることである。仮に数値情報が存在したとしても、それを人間が完全に記憶できるわけではなく、人間の暗記情報だけでは現場作業には十分でない。仮に重要な情報を人間が頭脳に蓄えていたとしても、情報から抽出するという作業は難しい。ステップ2の作業で上げた3つの問題は極めて類似していて、われわれは、全ての関連情報を探し難いし、情報に関連性を持たせられないし、仮に正しい情報を得たとしても、その情報に対し偏見を持ち、認識を間違う。

もし、非常に重要な情報が、メンタルなデータベースとして人間の頭脳の中に記憶されていたとしても、同じ問題が浮かび上がる。例えば、関係者のだれかが、ある要素が重要であると考えているとする。彼は、その事項に関してものすごい知識を有しているとする。彼はシステム全体がどのように動くかということが理解できない

ので、だれかのそのことに関する意見を間違っていると考える。こうして、参加者は他の重要な要素を見逃してしまう。実際には、関係者のだれもが信頼できる情報を持っているのだが、問題は、その関係者が個人的に係っている部分的なシステムについての知識であり、全体ではないことである。そして、われわれは、他人が何を信じているかについては気にしない。例えば、企業の顧客の振る舞いを他人がどう思っていようと気にしない。しかも、顧客の振る舞いに関する信頼できる情報というのは、顧客そのものからしか、あるいは顧客の行動を観測した観測結果からしか得られない。そして、そうだとしても、そうした顧客から得られた情報も信頼できない。

ステップ2で問題領域の全てを包含し定義しようとしている。全てのステップが、データが無いまま作業が進められるので、このステップ2での作業で問題の全体像を本当に把握できるのか、何か限界のようなものを越えてもっと違った風に全体像を捉えられるのかが明確ではない。参加者のフィードバックに関する見解に焦点を当てて分析していく、いわゆる、「閉鎖ループで考える」作業に焦点を当てていこうとして、重要な外部要因を無視してしまうリスクを抱えている。システム・ダイナミックスのモデルを構築する上で、内生的なシステムの振る舞いに焦点を当てるようというアドバイスがよく行われるが、どの関係者がシステムにはあまり影響を及ぼさないのかということを見極めるような事項には触れられていない。

関係者の中でメンタル・モデルを共有するということに同意したとしても、ステップ3の作業での問題が解決できるわけではない。ストックとして累積されるものは、人口、あるいは人間以外の動物などの生息数、現金、物理的な資産などいくらでも存在する。言うまでもなく、初心者が一般的につまずくのがこのステップで、ストックではないものをストックと考えてしまったり、ストックにすべきものをストックとは考えなかつたりする。特に、エージング・チェーン型のストックが関係していると、この間違いを起こしやすい。このことはステップ2の作業で、エージング・チェーンが認識されていても起きてしまう。このことから、経験を積んだSDモデル開発者だけが、定性的なフィードバック図をから信頼できる品質で、ストックを認識できるといえる。

ステップ4では、モデル構築手順に沿って数値シミュレーション可能なモデルを構築する。しかし、この作業も、経験豊かなシステム・ダイナミックスのモデル開発者であり、ステップ1から3までを指導したファシリテーターの指導の下で行われる必要がある。通常、この段階になって初めて重要な数値情報が集められる。しかし、先の作業ステップで作業に参加した参加者は、すでに因果関係図に束縛されてしまっていて、因果関係図に記されているデータを探すということしか念頭にない。この状態は、モデル構築者が経験豊かであり、見逃しているが重要なデータが存在するということを知っていて、参加者にそのことを指摘し、参加者とモデルにその要素を含めるかどうかを議論した場合にのみ打破できる。従って、因果関係図に記されていなかったというだけで、重要なデータが見逃され、あまり重要でもない要素に関するデータがモデルに無理やり取り込まれ、関連性を希薄化してしまうという重大な問題が発生するリスクがこのステップ4の作業に存在する。

ステップ5は、広く認められている検証とテストのやり方に従って実施される。この広く認められている検証とテストのやり方に関しては、さまざまな文献が存在する(Forester and Senge 1980, Graham 1980, Barlas 1989, Coyle and Exelby 2000, Peterson and Eberline 1994, Sterman 2000 の 21 章, Schwaninger and Groesser 2009, Yucel and Barlas 2011)。図1に示したような、標準的とされる方法でモデル構築が行われたとすれば、この重要な作業に行きつく前に、かなりの労力と時間をかけてモデルが構築され、モデルの構造の妥当性が十分議論の中で確かめられているはずである。従って、根本的な問題のようなものが初期段階で除去されていなければ、再び、元のステップに戻って作業を大幅にやり直すということが起きてしまう。スキルを持ったモデル構築者であれば、こんな危険なことが起きないように、ステップ2とステップ3の作業で目を光らせてモデル構造の妥当性を確認しているはずであるし、ステップ4の作業で定義した数式の妥当性もきちんと見ているはずである。

ステップ6は、ステップ1の作業で明確化した問題に対し、完成したシミュレーション可能なモデルを使って質問に答える作業である。シミュレーション結果が実際の組織の実績と違っているとすれば、だれかがモデルに何か追加、あるいは変更しなければならないだろうし、そういうことを行えば、どこかの時点でかなりの規模の影響が起きる。数値的なものなので数値的な解析が必要になる。定性的な洞察は、すでにステップ2での作業で行われているのだが、参加者は、ステップ3からステップ5までの作業が終わるまでは確信を持って何かをすることができない。

逆の結果となるモデル開発プロジェクトと現場適応

もちろん、先の標準的に受け入れられているモデル開発手法でも十分洞察が得られ、シミュレーション可能な

モデルの開発も成功しているが、先に述べたような問題から、特定のプロジェクトにおいても、そしてその方法が広く採用されインパクトを与えるという点でも、実際のモデル開発においていろいろ深刻な問題が起きている。

メンタル・モデルを共有することは良いモデルを構築する上で重要な要件であることが広く認められていて、このことに関し、いろんな研究が行われている。個人の研究やグループによる研究の評価や採択した方法から得られた成果は、概ね3つのカテゴリーに分けられる。最初は参加者の満足度と許容に関するものであり (McCartt and Rohrbaugh 1995, Vennix, Scheper W and Willems 1993, Huz, Andersen, Richardson and Boothroyd 1997)、2番目は参加者やグループの考え方に関連するもの (Franco and Rouwette 2011)、そして最後が、参加者の能力が改善されたかどうかに関するもの (Scot, Cavana and Cameron 2014) である。

しかしながら、インパクトの観点から見てみると、これらの問題は必ずしも主要な関心となっていない。メンタル・モデルの共有で形成される価値を評価する基準に対して2つの事項が抜け落ちている。最初の事項は、メンタル・モデルの共有が実際に検証されたのか、あるいはメンタル・モデルに重要な何か欠陥やエラーのようなものがあったかどうかという疑問である。2番目に、メンタル・モデルを共有するというプロセスの結果、だれかが実際に何か対策を講じたのか、例えば、もっとお金を使うようになったとか、もっとお金の消費が減ったとか、確約するようになったとか、しなくなったとかなどが起き、そのことでどのように成果が大きく改善されたかという疑問である。この方法が広く採用されるようになるために最も重要な点は、この方法を使ったことで価値ある違いが生み出されたかどうかで、このことが抜け落ちている点が特に懸念される。

こういった事項が研究から抜け落ちているということは、何か代替的なプロセスが試みられ、参加者の関与している状況に対し実施すべき事項や実績の変化に関する意思決定に関して、メンタル・モデルの共有化でなされる改善と同じような、あるいはそれ以上に状況を改善するようなことが行われ、彼らの理解を変えたのではないかという疑問が浮かび上がる。

これは、ステップ2の作業で実際に何か価値あるものを産み出したかのかが不明確であるということだけではなく、この作業自体が時間も費用もかかるものであることからも、費用対効果があるのかどうかが疑われる。広く参加者を集め、共有するためのメンタル・モデルを開発し、参加者にそのモデルに同意させるためにはものすごい時間がかかり、また、モデルを開発し、改善していくために何回ものイベントを開催する必要がある。これとは別に、必要なデータを取集するという極めて時間がかかる作業も含め、実際に稼働するモデルの仕様を決め、モデルを開発するという作業がある。従って、モデル開発から、実行可能な解決策を得られるまでに長い時間がかかることになる。

データ無しで開発作業を実施していれば、あるいはどんなに検証を実施しても、因果関係図やSDモデルで重要な要素（特にストック）や重要な因果関係（特に複層的なストック・フロー構造）が抜け落ちていることや、無関係なものを含めてしまうこと、さらには間違った状態やリンク（結び付き）を含めてしまうことを防げない。SD分野で長い期間共有されているジョークがある。これは、ある有名な経済学者が牛乳の生産に関する計量経済モデルを作成したが、そのモデルには1行も牛についての記述が無かったというものである。しかし、発表されているSD関係のモデルでも似たような驚くべき欠陥が見られる。例えば、SD専門家が構築した保険会社のモデルにおいて、政策意思決定者が行う政策意思決定関係のストックがそのモデルではなく、漁船団に関するモデルでは魚の販売価格という要素が抜けていた。

このような、モデルに欠落した重要と思われる要素が存在することは、どんなに状況が似通っていても、違った問題や違った環境に対するモデル構築にそのモデルを流用できなくしている。モデルが流用できないのであれば、因果関係図を作成しても、メンタル・モデルとして共有化が難しいことになる。従って、全く同じ、あるいは似たような対象を取り扱っていても、他のグループにとって、この因果関係図は最初から受け入れられない。この点は、他の経営関係の方法論とは大きく違っている。他の方法論では、現場での適用を考え、同じ種類の問題に対して同じやり方を適用してソリューションを導くようにしている。またその手順や方法をテストし、精錬し、文書化し、多くの類似案件に対し、適切な調整を加え適用されている。SDのモデル開発プロジェクトの「システムをモデルで表現するのではなく、問題をモデルで表現する」傾向は、その部分を厳しく見るとすれば、この分野では、他の経営関係の方法論のように、同じやり方で解決策を繰り返し提供できないままに放置することになる。フォレスター(2013)が述べているように、「個人が興味を持った特殊な状況をモデルにせずに、代わりに、その特殊な状況が存在しているシステムの共通事項をモデルにすべきである」と考えるし、またホーマー(Homer 2013)も述べているように、我々の方法は「他のモデリングの分野が取り扱ってきたすべてを包含する『類

似性』と同様に取り扱える」べきだという考えにも同意する。

彼らの意見を含め、先の一般に受け入れられている方法に見られる問題が、なぜ SD のモデル開発プロジェクトは費用も時間もかかり、それだけの価値があるのか不明確な理由を説明しているように思える。また、もし、このことが正しいのであれば、潜在的な顧客に、最初のプロジェクトをトライアルとして薦める、あるいは他の問題へ努力を繰り返し薦めることは困難になるであろう。このことで、なぜこの方法が採用されにくく、この方法のインパクトが限定的なのかの妥当な説明を持つことができる。

この問題は、現場でモデルを開発する専門家の能力も困難にしている。複雑で不確定な方法論しか存在しないということは、例え単純なシステムであっても、初心者が、うまく機能するモデルを開発するような根本的な原理を基にモデルを構築できる機会が無いということを示している。構築手順でのリスクからプロジェクトを守るすべは完全にファシリテーターの経験と才能にかかっていて、経験が不十分な SD モデル構築者ではプロジェクトを成功に導くのはかなりチャレンジングなことである。また、本当に価値が現れるのは、長い時間と多大な費用をかけた労力の後なので、まだ経験の無い若い専門家が最初の段階からプロジェクトで主導権を握り、プロジェクトでの作業の価値を納得させることは難しいであろう。

アジャイル型開発法による代替

図 1 に示したモデル構築方法は 1990 年代まで、ソフトウェア開発で主流を占めていた、「ウォーターフォール」型開発法に極めて類似している。この方法でも同じく、ソリューションとなるアプリケーションをプログラム・コーディングする前に、まずは、望ましいソリューションが適用される範囲を明確にすることから作業を開始し、開発すべきアプリケーションの全体の構造を定義する。このソフトウェアは、インストールされ、ユーザーによって運用される前に、テストが実施され、デバッkingがかけられる。

多くの SD モデル構築プロジェクトと同じく、ソフトウェアをウォーターフォール型で開発するのは、費用も時間もかかり、しかも結果は不確実であることが広く認められている(Fowler 2001)。その後、この方法に代わって、斬新で適応的な「アジャイル」型開発法が適用され、その適用が増えている。この方法は、ユーザーが開発に参加し、ユーザーを満足させることが至上とされ、また常に機能するソフトウェアがユーザーに示される。また、この方法では、関係者を大勢集めての型どおりのレビューはあまり重要視されなく、むしろ豊富な知識を有する個人との密接な共同作業が重視され、学習しソフトウェアを進化させるために仕様変更を行なうことも歓迎され、(未完成の作業を最小化するという意味で) 作業の単純化が歓迎され、技術的に優れていることが歓迎される。このやり方では、作業の最後にテストを行い、デバッグを行うのではなく、最初から品質が高い成果を産み出す (Abrahamsson, Solo, Ronkainen and Warsta 2002)。

この IT の開発で行われるようになったアジャイル型の開発法の特徴や利便性を SD モデル開発にも使えないものであろうか。このようなプロセスも技術的卓越性を必要とするが、4 つの原則から構成される方法が頼りとする観察を伴う厳密科学によって確実化されるかもしれない。

1. 関心を持っている対象の結果(outcome)に関する時間的な変化を説明し、予測し、改善することを試みる。
2. それらの結果は、各時点で、数量的に累積するようなストックの影響を直接的に受けている。ただし、対象が初めからストックであることが分かっている場合はこの限りではない。従って、その対象の結果の変化は、それに影響を及ぼすストックから直接的な影響を受ける。さらに、外部要因やシステムからの変化がここに加わる。
3. ストックに累積していく量は、関連するフローを介してのみ、先に述べた他のストックや変数の影響を受ける。従って、他のストックや変数から直接変化を受けるのではなく、それらからは独立している。
4. フローの変化量(flow rate)は、それに関連するストックの量、関係者の意思決定、あるいは、外部要因の影響によって決まる。

項目 3 と項目 4 はそれぞれ、フィードバックも含めて関連性を持っている。しかし、そのフィードバックは、これらの原則のもとで結果として導き出されるものであり、フィードバックそれ自体を発見することは原則の一つではない。

この論理的な原則を作業の流れにして、SD モデル構築のアジャイル型プロセスとすることができる (図 2)。

最初の反復事項は、問題を抱えている当人や、そのことに最も関心を持っている関係者に直接的に関係する事項を、時間的な変化として把握することである。たとえ正確な数値が不明で、推測値でしか示せないとしても、注目している対象の全て、そして、それと因果関係を持つ全ての要素について、その時間的定量変化をグラフで表示したものの関係をダイアグラムする。モデルの部分に関する特別な知識を持った個人、あるいは小グループが、モデルを詳細に検討することで、それに続く反復作業を推し進めていくであろう。全ての作業ステップは、同様に、シミュレーション可能な動くモデルによって、発見された因果関係の分析結果が妥当であり、現実的であるかをチェックされる。この方法は、他のグループ・モデリング・プロセスを代替するものではなく、ラジカルな応用と考えるべきであろう (Vennix 1996)。

一般的に推奨されているモデル開発手順の問題点を長く認識している実践家であれば、同様の手順を採択してモデルを開発していると思う。このような手順が利用されていることが広く知られていないだけである。ただし、この方法が有用であり、実際に役立つとしても、文書化し、将来の検証に耐えるようにしておく必要がある。

この手順を具体的に示すために、次の節で、中規模のITサポート企業を診断したものを取り上げてみた。この例は極めて単純であり、従って先のアジャイル型開発法の原則を明確に示している例として取り上げた。



図2 アジャイル型開発法

事例：中規模のITサポート企業

この企業は、小型小売店グループ、会計事務所や法律事務所、小規模の建設会社や輸送会社などの中規模のビジネスを行っている顧客に対してITサポートを提供している会社である。この会社は、顧客が必要としている、コンピュータのハードウェア、ソフトウェア、通信機器などを提供し、若手のITスキルを有する技術スタッフを従業員として雇用し、顧客への技術支援を行っている。この会社は、何年かうまく経営した後、企業成長のための努力を行い、それは当初成功したが、およそ2年前から顧客からの苦情が多くなり、顧客をどんどん失うようになった。

ステップ1は、標準的な方法と同じで、懸念事項である問題を時間挙動図で示し、これまでの変化と、今後の望ましい変化を示す。この時間挙動図では、最低限、実際に業績指標（ここではAとする）の値がどう変化してきたかは、およそであっても記載しなければならない。ここでは、次の作業としてステップ2に進むのか、それともステップ2を省いていきなりステップ3に進むのかの選択が分かれるので、さらに、その変化が、ストックの変化を示しているのか、それとも、ストックによって影響を受けている変数の変化を示しているのかを明確にしなければならない。

ステップ1：時間経過でどのように業績が変化するのか

コンサルタント：貴方の話では、顧客からの苦情が増え、そして顧客数が減ってきていて、この問題は2年前から続いているということですね？

何が起きていて、その起きていることをそれぞれ図に表すことができるでしょうか？

社長：2年前、顧客数は約90社でした。そして、当時は、ビジネスが好調でした。そこで、営業部長はもっと顧客を獲得することにしました。

彼はうまくやり、顧客数は135社に増えました。しかし、同時に、このころから顧客から寄せられる苦情数が増え、最大で、全ての顧客から、月次で3件近くに苦情が寄せられるようになりました。

技術担当は通常の顧客サポートの仕事よりも苦情処理を優先せざるを得ませんでした。

その後、長年付き合ってきた顧客が離れ始め、現時点では、顧客は115社に減っています。

どのようにこの問題を取り組んでいけばいいのか分からなく、手をこまねいている間に、顧客がどんどん離れています（図3）。

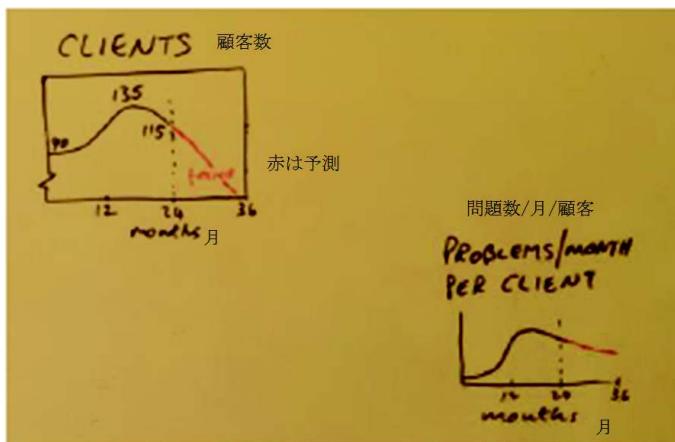


図3 ITサポート企業の懸念事項を時間挙動図で表したもの

ステップ2では、変数の変化を直接的に説明するような起因となるストックが無いかを見る。そのようなストックが1個、あるいは数個あるはずである。（ただし、対象としてステップ1で取り上げた変数がストックである場合は、この作業は省略し、ステップ3に進む。）通常は2個、あるいは数個であるが、時には複雑な関係になっていて、もっと多くのストックが関係してくることもある。（ここでは2個として、それぞれB、Cと表わすことにする。）Aを、B及びCから計算することが可能であるはずなので、因果関係を確認するために、このB及びCについての時間挙動図を描き、先のものに追加する。この作業はB及びCについても、それぞれB及びCの変化を直接的に説明するような起因となるストックを探し、時間挙動図を描く。この作業を、因果関係が、ストック、意思決定、外部要因を含めて完結するまで継続する。（このプロセスでは、システムの振る舞いが内生的なしくみで全部説明できるようにするまで因果関係を詳細化することを求めてはいない。実際の世界でも、適切に管理できる範囲を超え、外部から大きな影響がシステムに及ぼされることは良くある。）

通常、因果関係となるものは2個から4個であるので、このステップの作業はそれほど時間がかかるない。このステップの作業では、少し関係の定義で不確実性があるかも知れないが、因果関係を確認できる数量データがあればその関係の不確実性は解消できるであろうし、単位のチェックや、別途、並行作業でのシステムの動きを確認するためのシミュレーション可能なモデル作成からも確認できる。（ホワイトボードで簡略的に書いた予想の部分のグラフを修正して）モデルとして示したものが図5であり、この顧客の実績値に加えて、赤線で示した部分に、経営者が恐れている、将来の顧客数の落ち込みの傾向を示している。（モデルは最新のブラウザがあれば、<http://sdl.re/m203>から参照できる。）

ステップ2：どのようにストックは変化を受けるか

コンサルタント：何が原因で、顧客からの苦情が増加したのでしょうか。

社長：技術担当スタッフの作業負荷が増えたせいと考えられます。彼らの作業内容は、装置を導入し、ソフトウェアをインストールし、システムのアップ・グレードを行い、顧客のIT担当者にトレーニングを実施するといったことです。そういう技術スタッフの作業の記録を付けていて、その情報から、作業負荷が、彼らの業務処理能力を超えたものとなっています。

そのため、技術担当スタッフは深夜残業や休日出勤を行わざるを得なくなっています。技術担当スタッフは最大で、標準勤務時間よりも60%残業していると推測されます。

コンサルタント：各顧客の要望に応じて、例えば、新しいソフトウェアが発売された、あるいは新しく装置が必要になったなどの変化で業務量が変わったのでしょうか。
 業務変化量は通常よりもかなり多いのでしょうか。
 スキルが十分ない技術担当スタッフや生産性が悪い技術担当スタッフがいないのでしょうか。
 社長：当社の従業員のスキルは十分で、生産性も悪くないと考えています。
 コンサルタント：それでは、技術担当スタッフ数はどうですか。
 社長：技術担当スタッフ数は、今年の前半期は若干増えています。その後減っていて、現在は37名です（図4）。

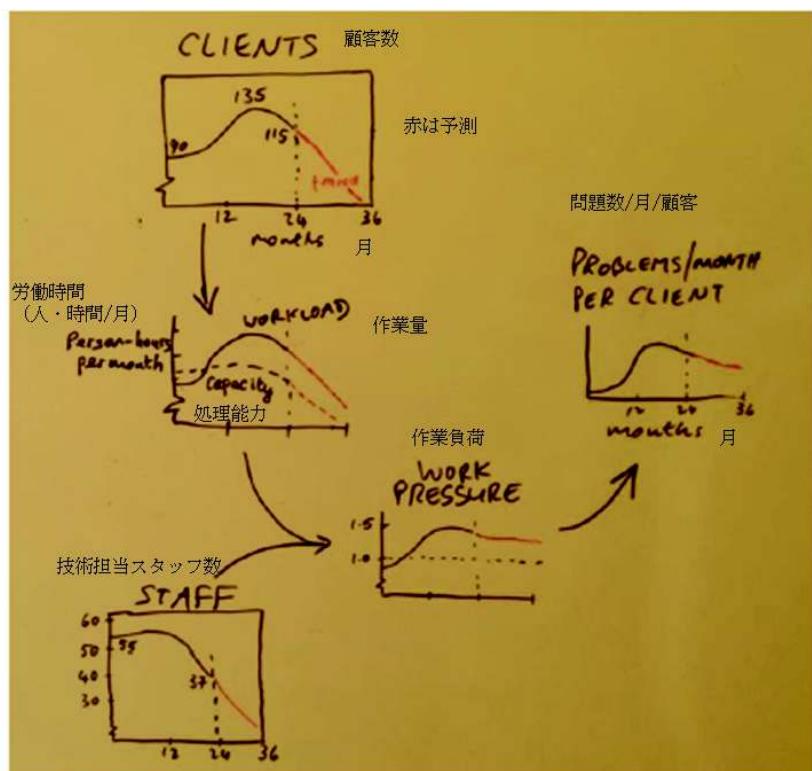


図4 ITサポート企業のストックの業績を起因させるもの

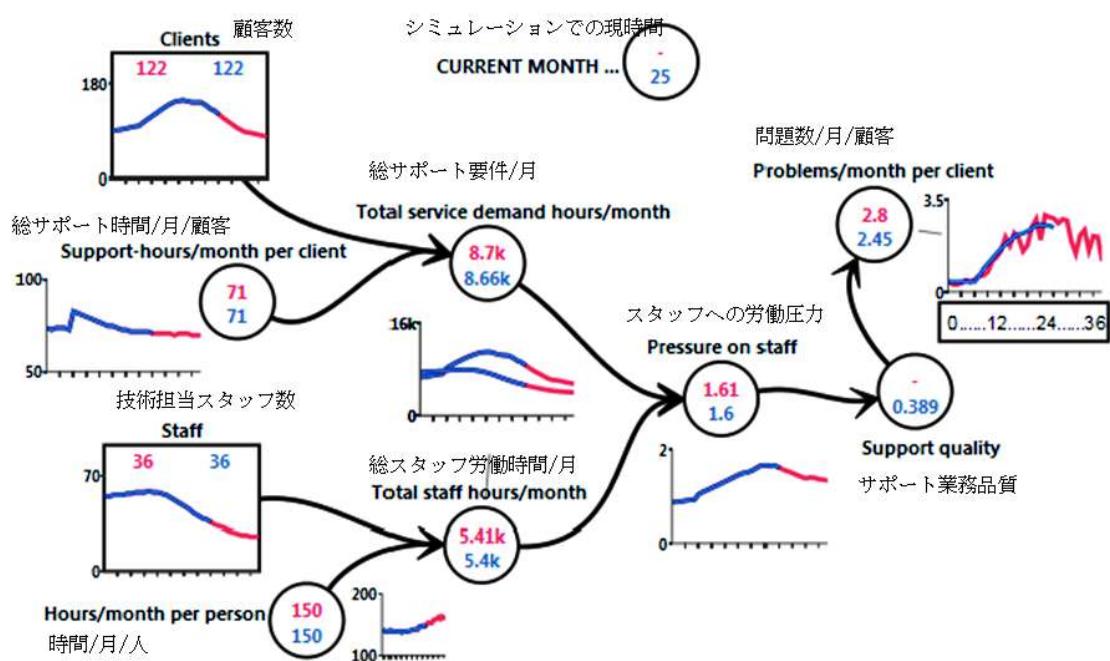


図5 ITサポート企業のストックの業績を起因させるものをモデル化したもの

もちろん、このストックから、対象としている出力変数への因果関係を検証できるようなデータを見つけるのはそう簡単ではなく、時にはそのようなデータが存在しないこともある。しかしながら、このプロセスでは、参加者が、因果関係をそうであろうと認めていればそれで良く、もし、AがB及びCによって影響を受ける関係であり、データAとデータBがあればCの値も自己検証できる。

このプロセスで重要な点は、グラフで変化を示すという点と、それと並行して動くモデルを作成するという点である。最近のSDソフトは十分ユーザー・フレンドリーなので、グラフで要素の変化を最初に表わしておき、それとは別途、シミュレーション可能なモデルを開発し、グラフで表した要素の変化をそのモデルに取り込むというやり方が可能である。モデルを素早く作成し、関係者のチェックを行ってもらい、そのチェックをベースにモデルを改善していくことは必要だが、モデル開発を関係者に見せながら、それと並行して、ホワイトボードなどを使って関係者と議論を進めていく必要がある。

図示とモデル開発を並行的に進めていくやり方には大きな利点がある。参加者は、自分達の目の前で、自分達が行ったシステムがどのように動くかという説明がシミュレーション可能なモデルとして表されるのを目にすることで、直ちにそのモデルに対する信頼を持つ。2番目に、そのモデルには実際のデータが使われているので、モデルの全ての要素を見ることができる。一部は推測が含まれるが、抽象的な項目や間違った項目は、モデルにされる際に発見される。従って、最初の段階から品質が確保され、品質は作業を進めて行っても確保される (Barlas 2014)。

ステップ3では、ストックの時間に対する挙動の変化を説明する。ストックに対して、入力フローと出力フローだけが時間的に変化を与えることができるので、従って、ここでも、同じくフローの時間挙動図で変化を示す。もし、ステップ1の作業で、関心があり対象としている要素がストックである場合、先に述べたステップ2の作業は不要である。このステップ3の作業では、全ての入力と出力を合計したものがストックに累積されるので、数学的には全く曖昧さが無い。もしあるとすれば、フローが複数の場合で、その場合は数量的なデータによる裏付けが必要になるであろう。

ステップ3：フローがどのようにストックの値を変えるかの検討

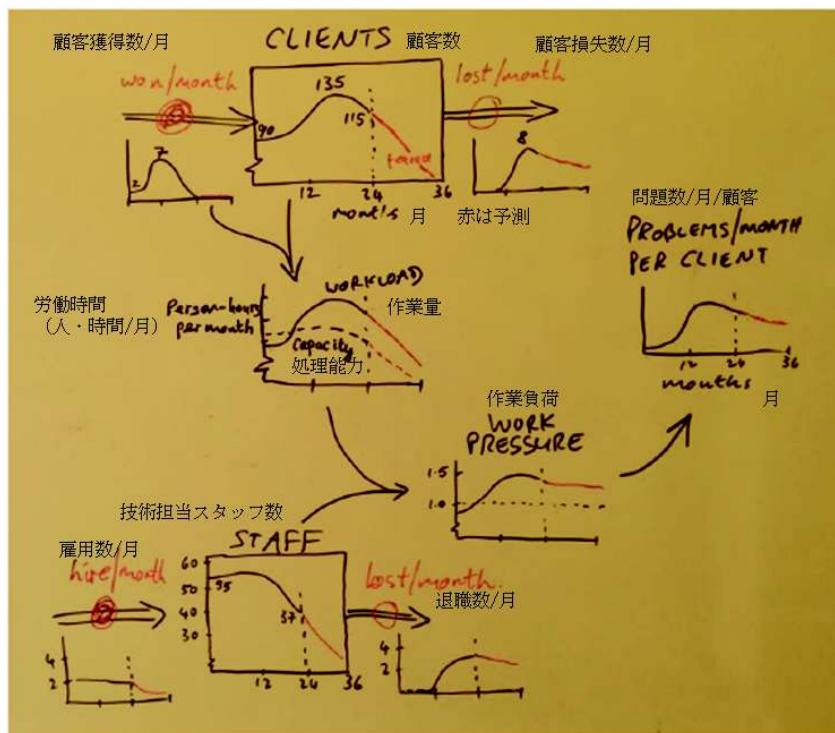


図6 ITサポート企業の顧客と技術担当スタッフに関するフローの変化

コンサルタント：どのくらいの速さで顧客を獲得、あるいは喪失しているのですか。

社長：営業は最初うまくいき、新規顧客を2ヶ月に約7社の割合で増やしていくのですが、だんだん新規顧客を増やすことが難しくなってきていて、現在では新規顧客は全く獲得できていません。

以前は、一旦獲得した顧客を離すということはありませんでした。しかし、1年前から顧客を失うようになり、数ヶ月前には、これまでで最大の、月に8社も顧客を失うようになりました。

現時点では、これほど多量の顧客を失うことは無くなっています。

コンサルタント：技術担当スタッフの雇用や退職の状況はどうですか？

社長：以前は、月約2名の割合で新規雇用を行っていました。そして、以前は、それほど退職者はいませんでした。しかしこの数ヶ月で大勢の技術担当スタッフが退職していました（図6）

グラフによるホワイトボードでの議論と並行して、モデルを作成し、そのモデルの挙動がホワイトボード上で検討したものと一致しているかを見ている。（図7. ここではホワイトボードで検討した予測を修正している。モデルは <http://sdl.re/m302> から参照できる。）

この時点では、関心を持っている結果は数量的にストック量の変化として、完全に、かつ信頼できる水準で説明できている。ただし、この事例では、意思決定や外部要因による直接的な対象への関与が無いが、他の事例ではそのようなことが起きことがある。そして、ストックの変化は、それに関係するフローの変化量として説明できている。ここで残されているのは、フローの時間変化を引き起こす原因が何であるかということだけである。このことは、ステップ2で行った方法と同じやり方をステップ4で行い解決する。ここでも、再度、因果関係のチェーンを考え、そこで1個、あるいは数個のストック、意思決定及び外部要因を考える。そして、ここでも、その起因関係を、実際のデータで、データが無い場合は推測値で検証し、（新しく加えた要素について）変数の時間挙動図を描き、チェーンとしての繋がりの妥当性を検討する。

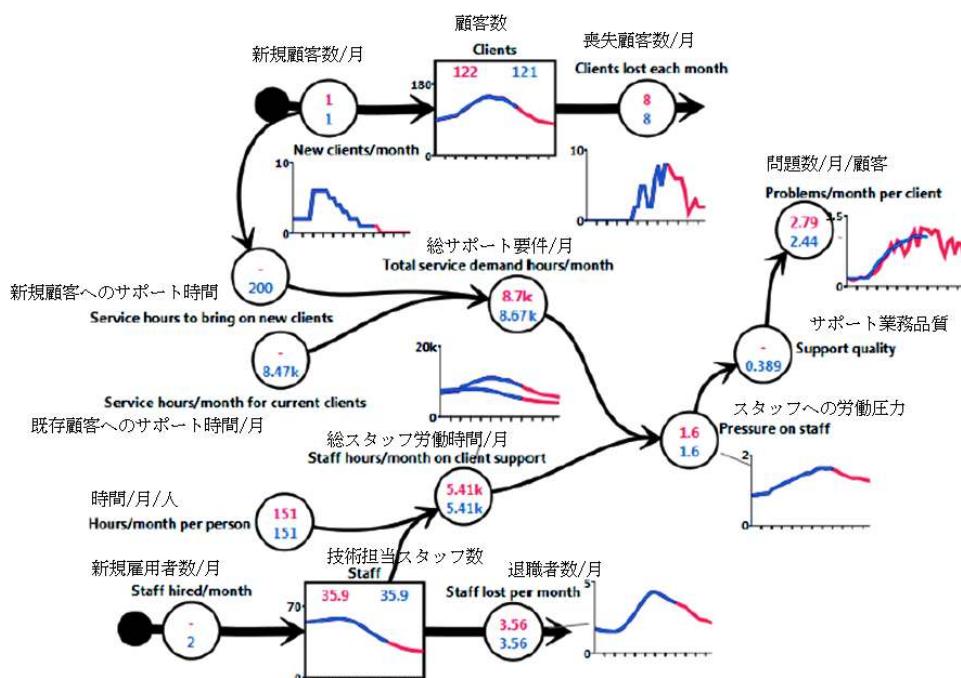


図7 ITサポート企業のモデルで、顧客と技術担当スタッフのストックにフローの変化を追加

ステップ4：何がフローの変化量を変化させているのか

コンサルタント：顧客が増えたのは営業部長が営業活動を行い、営業がうまくいったからということですが、何か他の要因はないのですか？

社長：営業がうまくいったというのはその通りで、当社は、競合入札に勝ったわけでもなく、既存顧客が当社を強く推薦してくれたといった話も聞いてはいません。

コンサルタント：それでは、なぜ新規顧客が増えているまさにその時に、既存顧客が離れて行っているのですか？

社長：私の個人的意見ですが、当社とのサービス契約を解消する顧客が出てくることはある程度は避けがたいことです。当社のスタッフが行った作業のせいで、後で問題が発生し、顧客から苦情を受けることもあります。問題への対処が遅れ、どんどん状況が悪くなることもあります。

しかし、顧客にとっては、多少当社のサービスに問題があつても、他のプロバイダーを探すのは面倒なものです。ただし、問題が多発し、それが何ヶ月も続くとなると別の話で、そうなると、当社との契約を止めるという話になるでしょう。

コンサルタント：スタッフの雇用と退職の状況はどうですか？

社長：問題があるということを認識した時には、もっとスタッフを増やそうとしても手遅れでした。（いずれにしろ、雇用には時間がかかります。）従業員がどんどん辞めていくのを止めることは避けがたいことです。

採用者は、最初は残業もうまくこなしてくれます。しかし、やがて、残業はもう十分と感じるようになります。スキルがある技術者ならば、別の就職先を見つけることも簡単です。

このステップ4では、挙げられた全てのストックに影響を与える全てのフローの変化量の因果関係を考える。これはストックに付随するフローについても同じで、同じように評価する。そして、このステップ4で、相互関係とフィードバックを明らかにする（図8）。従来型開発手法と違って、フィードバック・ループの存在は証拠に基づいて行い、参加者が主張したからといった理由だけで取り上げることはしない。

ステップ4（つづき）：相互関係とフィードバック

コンサルタント：それでは、顧客が離れていったのは、サービス品質が悪くなつたからで、サービス品質が悪くなつたのは顧客が多くなりすぎ、しかしスタッフの数が少なすぎたので作業負荷が増えたからということですね？

社長：そんな感じです。しかし、顧客の数が減ったので、仕事に対するプレッシャーは少し弱くなってきていると思います。

コンサルタント：スタッフがどんどん退職していくのは、顧客が多すぎ、それに対してスタッフが少なすぎるところで起きたのですね？

社長：そう思います。現在の問題は、多量にスタッフが辞めていったので、以前よりももっと少人数で、しかも仕事量が増えていることです。そのため、残ったスタッフも仕事がきつくなつたので辞めていくことです。

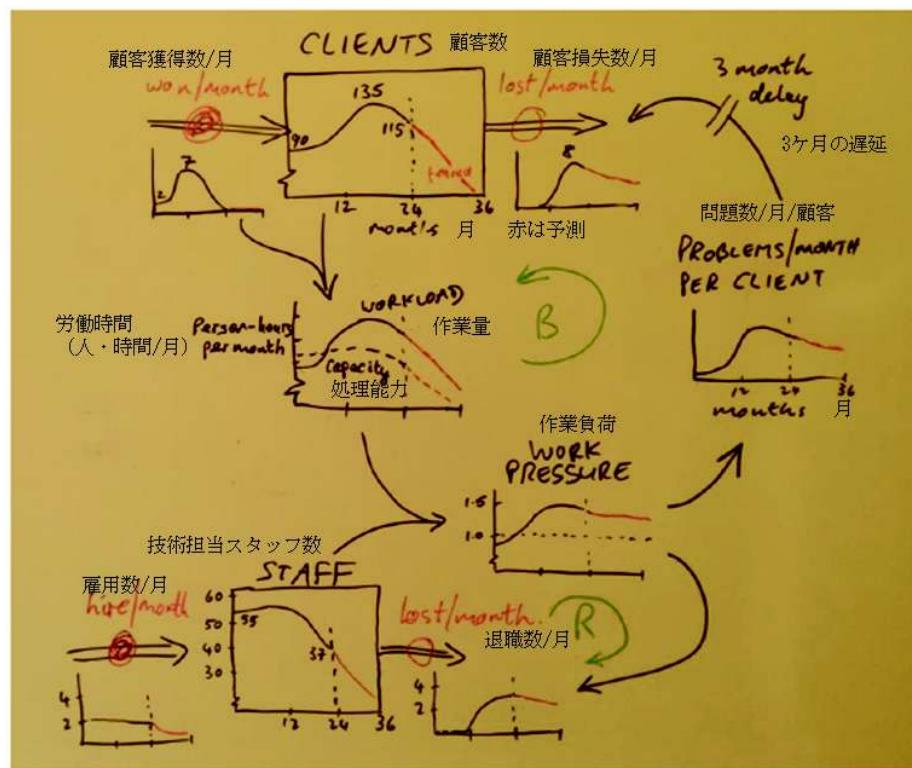


図8 ITサポート企業のストックとフローの相互関係とフィードバック

ステップ4では、この相互関係とフィードバックをモデルに追加し、因果関係の検証とシステム全体の振る舞いの検証を行う。この段階になれば、何か意思決定のようなものを持ち込んで業績を改善できないか、あるいは問題を解決できないかを検討し、そのための実行計画を策定する。このITサポート企業のソリューションと実行

計画は、支援が大変な顧客は他のプロバイダーを紹介し、契約を止め、顧客から外すことで、顧客数を適切な水準にまで引き下げるというものであった。この対処でスタッフの作業負荷が減り、サポート品質を回復させることができるようにになった。新規雇用も行い、この企業は再び成長するようになったが、今度は十分気を付けて、むやみに顧客を増やさないようにしている。この「望ましい」未来をグラフに追加し、モデルでそれが可能かどうかを確かめた。(図9及び図10。これはある一つのシナリオに基づいた結果である。モデルは <http://sdl.re/m4a02> から参照できる。)

このステップ4では、ステップ2の作業で取り上げられなかった新しいストックを追加することがある。(ただし、この事例に取り上げた、IT サポート企業では、このようなことは起きていない。) このような、このステップで追加することになったストックは、直接的に、最初の方で取り上げたストックに対し関係するものではないが、ストックに関連するフローに関係を及ぼし、ストックの振る舞いに間接的に影響を及ぼす。例えば、企業が取り扱っている製品の種類はその企業の売上を直接的に説明するような要素ではないが、顧客獲得率を説明する一つの要素である

これらの新しく追加されたストックは、それまでに列挙されているストックや関連するストックと合わせ、エージング・チェーン構造になっていないか、例えば、魚が幼魚から成体になるようなエージング・チェーンが存在し、幼魚のストックを追加しなければならなかどうか、会社の新人採用では、採用可能な人間のストックを経てでないと新卒採用ができないような構造になっていないか特に注意する必要がある。このような場合、新しくストックを追加しなければならないかどうかは、「そもそも、そのフローの源泉はどこで、その源泉が重要かどうか？」あるいは、「そのフローはどこに向かっていて、その向かっている先にあるストックは重要かどうか？」という質問を行ってみれば明確になる。ここで事例として取り上げたIT サポート企業の場合、新規採用者は数人なので、完全に一人前になる前に数ヶ月の研修を実施し、経験を積ませるような変更を追加する必要があつただけであった。(モデルは <http://sdl.re/m602> から参照できる。)

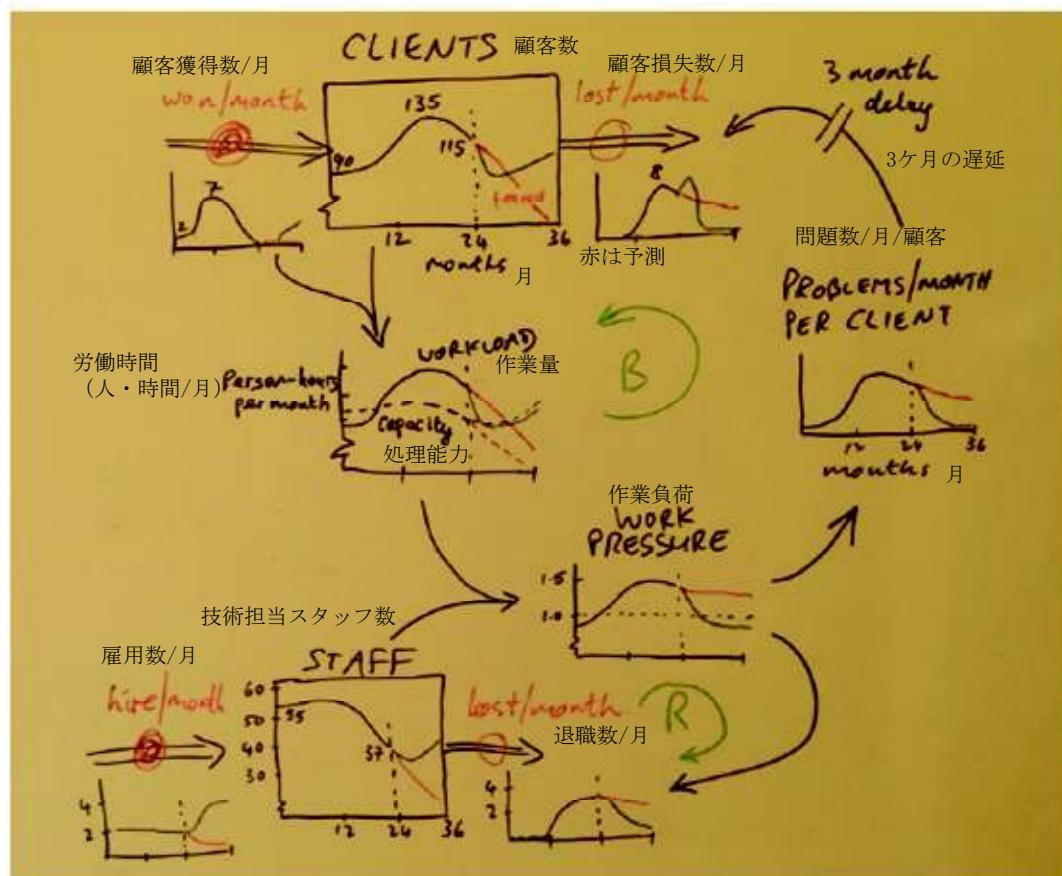


図9 IT サポート企業のソリューションをグラフに記入したもの (24ヶ月目からの上向きの青の線)

もっと多くのストックが関連するような例を図11に示した。この企業はハイテク関係のハードウェアを製造している会社で、この企業でこのモデル構築を行った時は、会計年度開始時期から9ヶ月過ぎた時で、マーケティ

ング、営業、人事に関する今後2年間の経営計画を策定しようとしていた。この会社の売上は、省エネ型のデバイスを顧客に販売し、インストールする時点で計上される。この会社の製品の主要顧客であるスーパーマーケット（そして、その他の顧客セグメントである、データー・センターやある特定の業界のユーザー）を獲得しようと営業担当が売り込みをかけていて、この営業担当の労働時間は、フルタイム当量(FTE: Full Time Equivalent)に換算した数字で把握している。見込みがあると思われる顧客候補は、「未接触」というストックに記録され、次いで、「見込み顧客」のストックに移動し、最後に、契約済「顧客」のストックに移動する。顧客が獲得されると見込み顧客を顧客にするための顧客への電話数が減るので、まだ未接触の顧客の数や見込販売金額を減らすことでバランスを取るため、そこにフィードバックが生成される。

次に、この会社の技術担当が顧客を訪問し、どのデバイスにどのようなこの会社の省エネ・デバイスを接続するかを審査する。次に、別の技術者が顧客を訪問し、この会社の省エネ・デバイスを接続する。この時点で、インストールしたデバイスは、顧客の「導入見込資産」として計上され、次に、技術者が再度訪問し、「導入済資産」となる。デバイスのインストールが早く行われれば行われるほど、まだ(ストックの箱の中の緑の点線で示した)インストールが終わっていない、資産が少なくなっていく。

営業秘密なので財務的な数値を示していないが、この企業は、顧客が得られた省エネ・コストのうちの何パーセントかを「売上」として得る(右側の紫で記載した部分)。大幅に費用削減が可能な顧客に最初に機器を導入するので、「顧客当たりの収益」は、最初はかなり高い。しかし、やがて、あまり省エネ効果がない顧客に順次機器を導入していくことになるので、収益が落ちていく。しかし、技術者が効率的に作業を進めていくようになると、「一人当たりの月次機器設置台数」が増えていくというフィードバックが働く。規模の経済や経験学習の効果で、「省エネ機器当たりの設置費用」も下がっていく。(ここでは利益は数値で示していないが)、「省エネ機器の設置コスト」と「人件費」が下がるので、その効果を考慮して利益が計算される。

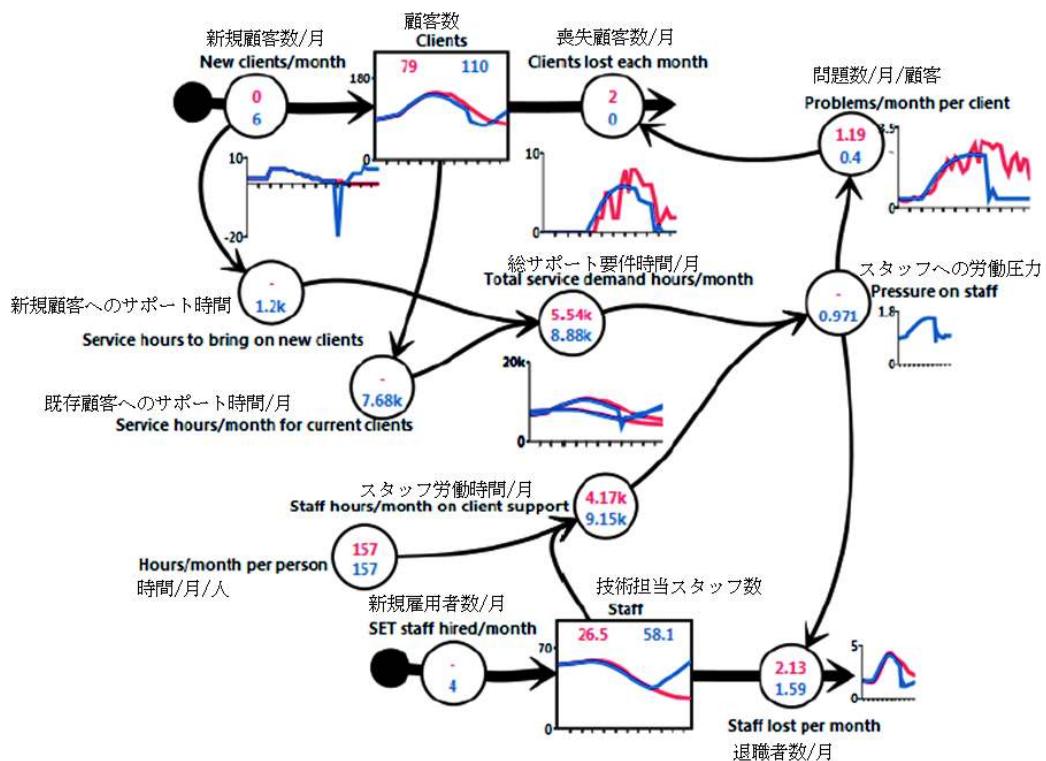


図10 ITサポート企業のソリューションを、モデルを使って検証したもの

このダイアグラムは、データや図の予測も含め半日の作業で作成したものであるが、すでに見込み客になっている顧客を本当の金銭を支払ってくれる顧客にする代わりに、新しい見込み客を探すという営業活動に多くの努力を払い過ぎていることが分かる。また、技術スタッフの時間も、顧客に省エネ・デバイスを導入して、収入を上げるよりも、見込み客の機器の評価の方にもっと時間と労力を払っていることが分かる。ただ、この段階では、見込み顧客はほぼ省エネ・デバイスの導入に同意しているので、技術スタッフがそれほど時間と労力をかける必要があるとは思えない。シミュレーション可能なモデルは、あまりこういったモデル構築に慣れていない

新人が、先の図を完成した後、3週間かけて作った。彼は3つの主なセグメントに分かれたモデルを作り、完成したモデルを使って、どの市場セグメントをターゲットにするのがいいのか、セグメント内でどの顧客をターゲットにするのがいいのかを検討し、営業部門と技術部門のスタッフの適切な数を算出した。

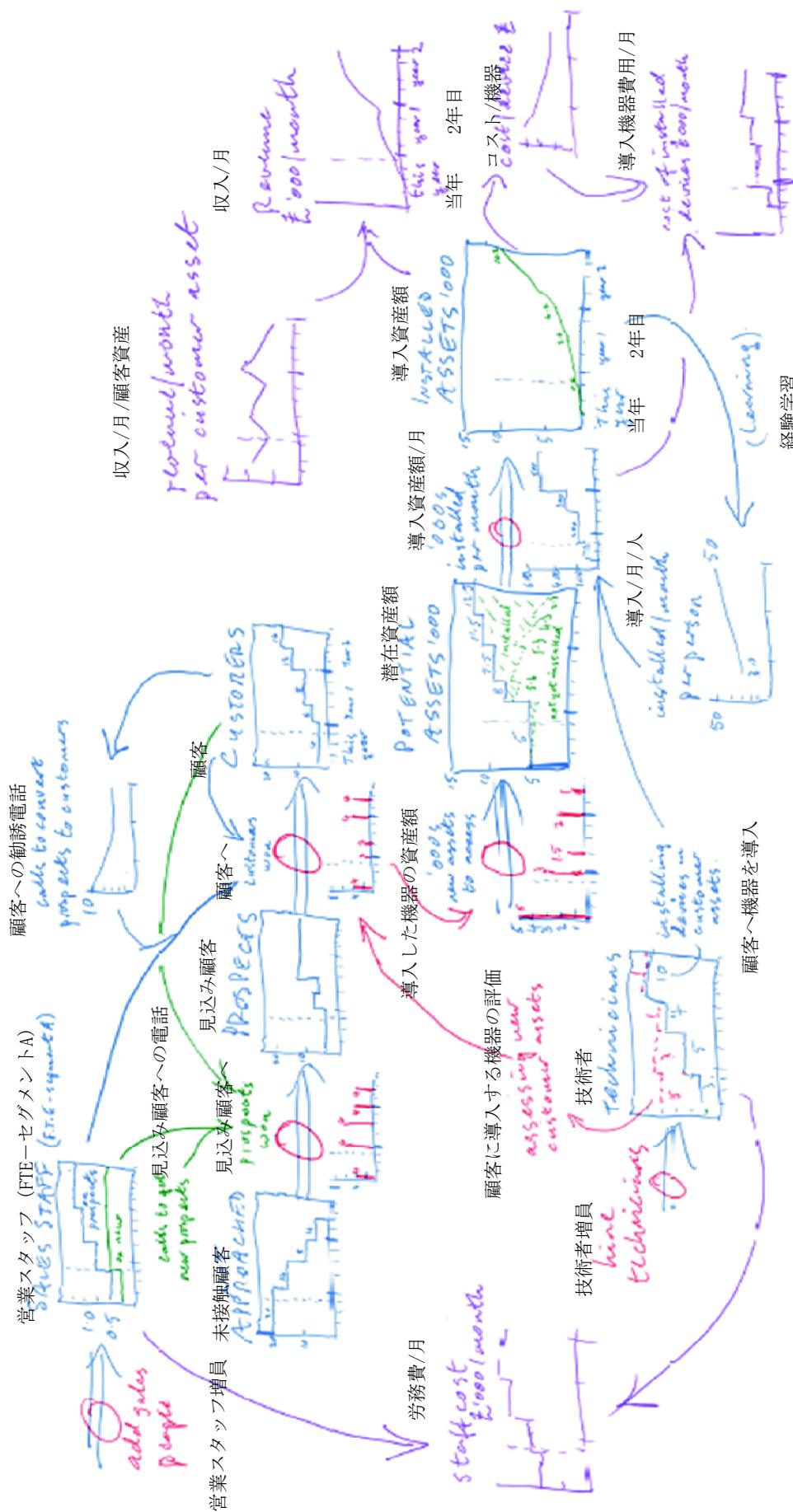


図 11 ハイテク企業の顧客のストック、フローと装置資産

システムの主な「物の流れの」構造、政策フィードバック及びモデルの範囲

この時点では、アジャイルで進めてきた、システムでの中心となるストックとその相互関係が明確化され、数量データで裏付けされる。つまり、システムの「物の流れの」構造とでも呼ぶべきものが完成し、フィードバックが暗示的に含まれる。そこで、最終的な作業として、意思決定などの政策を表わす情報のフィードバックを完成させ、どの情報が使われ、その情報をどのように意思決定のために使うかということを明確化する。先のITサポート企業では、この意思決定は、どれだけ顧客を増やすかどうか、そしてどれだけスタッフの雇用を増やすかどうかということであった。営業担当を採用し、顧客を増やすことは、顧客への品質の高いサポートを達成しつつ収益性も確保できたことと相まって、かつてはこの企業の成功要因であった。収益性については図9及び図10には示していないが、顧客が払う料金からスタッフの給与やその他諸経費を引くことで簡単に計算することができる。政策フィードバックを含めたこの完成したモデルは <http://sdl.re/m912> から参照できる。

このアジャイル型開発方でのプロセスの包括的な構造を図12にまとめた。ここでは、最初に、業績の時間変化をストックの時間変化として説明し、その時間変化は、そのストックに直接流入、あるいは流出するフロー量の変化でもたらされるとし、さらに、外部変数、意思決定、あるいは既存のストックの値によってフローの変化がもたらされたとした。この最後の、他のストックに影響されるような依存性は、システムの物の流れのフィードバックから発生し、さらにここに、意思決定などの政策的なフィードバックを追加することになる（図12で、点線で示した部分）。

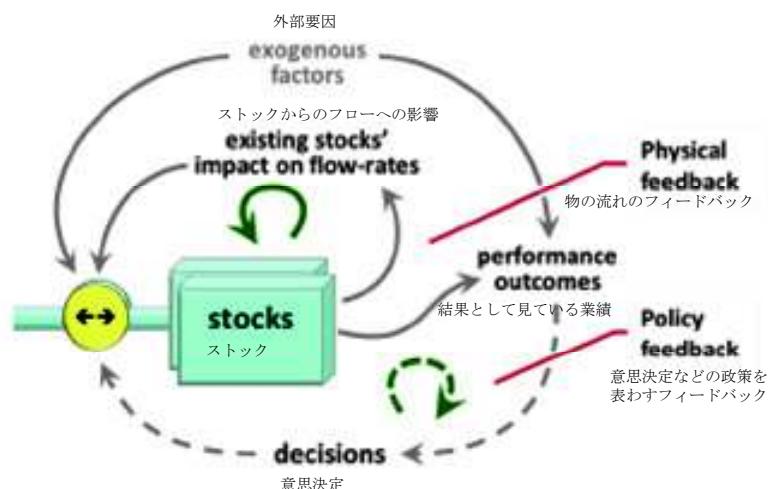


図12 機能、政策的フィードバック、意思決定、外部変数

アジャイル型の開発手法の長所

先に述べたプロセスを、繰り返してモデルを構築していくやり方によって2つの一般的かつ重要な洞察が得られる。最初に、単純なことだが、業績を改善するためには、システムの中のフローの量の変化に全面的に集中すればいい。もし、システムのパフォーマンスが、注目しているストックによるもので、外部要因が一定であり、フローの値がゼロであれば、出力は変わらないはずである。従って、フローの値がゼロではないことで出力が生成されなければならない。

2番目の洞察も、最初に述べた関係から直接的に導き出される。意思決定では、フローの量を変化させることによって、業績を変えなければならない。顧客を獲得する、従業員を増やす、といったITサポート企業の意思決定は、フローの量に関するもので、多くの意思決定は、直接的に、そういうフローの量がどれだけであるべきかを考えている。他の事項、例えば、価格変更、営業費用、研修といったものは、それほど直接的にフローの値を変化させない。因果関係図はフローの量の変化にも意思決定要因にもあまり重要性を置いていない、このことがモデル開発プロセスで良く見られるさらなる問題を示している。

図2に示したアジャイル型開発のプロセスは、図1に示した従来型のモデル開発の作業で挙がってきた問題点を解決するものである。参加者は常に数値的な証拠の必要性を分かっているので、たとえただもっともらしい想定でしかないとしても、それが見知ったものかどうか、だれの定義や値が一番理解できるかを実態として明らか

にする。彼らはまた、挙げられたエンティティの数量変化は、そのエンティティが依存していると彼らが信じている別のエンティティの数量変化によって変わるということを理解しているので、直ちにその因果関係が妥当かどうか、その変化が正しいかどうかをすぐにチェックする。違った意見や違った解釈は簡単にはこのプロセスを生き延びられないで、従って、問題の当事者やその関係者は、メンタルモデルを共有できたというだけではなく、そのメンタル・モデルが妥当で使えるものであることを示すこともできることになる。

実行可能な洞察がこうして、初期の段階からモデル完成まで、どんどん提示されていく。ここで取り上げたITサポート企業の事例は、数時間で枠組みを決め、モデルを構築したので、これらの発見事項の重要性を示すには単純すぎたかもしれない。もっとも、モデルは数時間で作成したが、実データと経営者の推測が合っているかどうかをチェックするのにもう1日必要だった。もっと複雑で大きなモデルの場合、モデル構築やデータ収集に時間がかかる。そのような場合は、例えば、売上が落ちたのは、顧客数が減ったのが原因であり、原材料購入が減少したからだとか、採用に問題があったわけではなく、採用者よりも退職者の方が多かっただけといったことを発見するのに有用である。

このアジャイル・プロセスは、モデル構築時間を大幅に短縮できるが、これは、顧客との議論を定量的なグラフで進められることと、モデル構築を議論と並行して行えるからということだけでなく、データの必要性が初期段階で顧客側に認識され、作業が継続されている間にデータ収集が手配され、早期に必要なデータ入手できる点も大きい。(なお、モデル構築は、顧客との議論が行われている同じ部屋で、同時に実施することが望ましい。)さらに、関係者が、因果関係で重要だと思ったデータは、すぐに議論やモデルに反映され、構造上の問題はこのプロセスで直ちに解決さる。

アジャイル・プロセスでは、単純に数量的ダイアグラムを作成する場合に(同じ部屋で同じ時間帯で)並行作業で作業を実施できるからというだけではなく、データの必要性が最初から理解されているので、他の作業と並行して収集作業を進めていくので、モデル構築に必要な時間が大幅に短縮される。さらには、関係者が因果関係ありと信じていることに対して相反するようなデータがあれば、それは直ちに反映されるので、構造的な問題が、このプロセスを通過するのは難しい。

この並行作業プロセスによって、大規模に関係者を集めて議論する必要性を抑えることができ、従って、忙しい人から時間を割いてもらうことや、またそのような議論のための時間が発生することも少なくすることができます。最初の診断の際は、システムの一部について知識があるだけの人も含め全員に参加してもらうのではなく、問題の当事者だけが作業に参加すればいい。そういうシステムの一部について知識があるだけの人たちは、彼らの知識がある問題領域のモデル構築が実行可能となった際に参画してもらい、モデルをよりよい、もっと妥当なものに改善していく際に、彼らの個人的見解やアドバイスを述べてもらう。しかしながら各段階における、このアジャイル型のモデル構築プロセスの単純だが論理的かつデータに裏付けられた手順は、因果関係図の作成作業に必要となる時間制限のないコンサルテーションよりも、はるかに少ない時間で遂行できる。

問題の当事者は、何が必要であり、なぜそれが必要なのかをすぐに理解できるので、モデル構築者がモデル構築で対象にしている特定ポイントについて特殊な知識を持ち、それらのポイントを追加したり数量化したりした結果を確認するために再招集する必要とする人との共同作業を行うことを信頼することができる。関係者全員を集め、全員で関係の理解を共有することは役に立つかもしれないでの、モデル構築作業の最後の方でそれを行っている。このモデル構築作業の最後の作業の段階であっても、何か間違いがあれば修正し、今まで見過ごされていた重要な構造を追加することは完全に可能である。

明確にしておきたいことは、アジャイル・プロセスでは、定性的な因果関係ループや影響関係のダイアグラムは、どの作業段階でも全く使っていないことである。ここで行われる手順で、データによって明らかなフィードバックのメカニズムが明らかにされる。そして、関心のある振る舞いが分析によって十分に説明できる限り、付加的フィードバックメカニズムの重要性を期待する理由はない。言うまでもなく、フィードバックを追加したり、マネジメントの改善のためにフィードバックが設計されたりするかどうかを確認することは妥当なことである。ただ、このような確認作業は、それまでの作業での分析作業、つまり、システムの中でどのストックが時間変化で増加、あるいは減少していく、どのような割合でそれが起きているか、その変化を引き起こすフローの変化量、以前には挙がっていなかった相互関係する要素があり、それがその変化を引き起こしているのではないかという検討を行う作業の中で行われる。この確認作業は、例えば、政策改善の検索や、ストックを追加あるいは削除するといったシステムの再設計の際に有益となるかもしれない。

アジャイル・プロセスでは、問題としている空間に關係する対象範囲も明確化する。これらの対象範囲は最初にある範囲の推測を試みるというよりも、答えられていない疑問がもう残っていないとなった時点で分かる。従って、モデルの対象範囲は、広く受け入れられているやり方と比べ、狭く設定されるかもしれない。しかし、多くの通常プロセスがデータに対する注意不足を通じて見逃す重要なメカニズムから発生する追加的な範囲も含むことができるだろう。

標準的な場合：標準的なシステム

SD 専門家が常に、あるいは、常ではなくとも多くの場合、図 1 に単純化して示したような作業を行っているかどうかは実はよく分からぬ。しかし、2 つの事実から、そうではないかということが証拠づけられる。第 1 に、SD の第一人者や成功しているコンサルタント達は、特定分野に絞ったモデル開発を行っている点である。Ford (2009) は水と水力発電に絞っているし、Moxnes (2004) は再生資源に、Paich and Peck (2009) は製薬に、Cooper, Lyneis, Els, Ford and others (Lyneis and Ford, 2007) はプロジェクト・マネジメントにというように、対象分野を絞り、その分野でのモデル開発に集中している。(これは、これらの人たちが、先の、専門としている分野以外の SD モデル開発をしていないとか、SD 以外のモデル開発の仕事をしていないという意味ではなく、専門分野に絞り、その専門分野で経験や成果を累積しているという意味である。) こういった専門家たちは、全く白紙の状態から作業を開始し、定性的なものからスタートするわけではなく、また、それ以前の経験で得られた構造とか現象とかを全く無視し、新しい関係者と毎回メンタル・モデルの共有化を行っているわけではない。

2 番目の証拠は、同じ題材で、一貫した、似たような複数のストックとフローの繋がりの構造が何度も見られることである。多くのケースは単純なチェーン構造で、例えば、魚の場合は小魚のストックが成魚のストックになるとか、社員を訓練し熟練工にすると、感染症の流行で、未感染者が感染し、回復するといったものがあるが、単純ではない構造のものも存在する(例えば、Carter and Moizer, 2011; Paich, Peck and Valant, 2011 を参照)。そういうチェーン構造の定性的性格はかなり特徴的で、複数の均衡ループが関係する。しかし、共有化されたメンタルモデルとして公表されている因果関係図にはあまりそのことの説明がない。多分、共有化するメンタル・モデルを作成する際に、定性的な説明が必要になり、自発的にこのような構造が浮かび上がってきたのではないかということが推測される。従って、どのような構造は、関係者のメンタル・モデルを反映したものというよりは、SD 専門家自身が生成したものであろう。

専門家はさまざまな経験をしているので、新しいモデル構築プロジェクトが始まると、まっさらな状態から新しい因果関係図を作成するのではなく、それぞれのプロジェクトの状況に合わせ、そこにケース・バイ・ケースで自分が知っている構造や知識を移す。これは、どのような状況であれ、同じ分野であれば、システムの政策的な要素は違ってはいても、その物理的なシステムの運用の方法は極めて似たようなものとなるからである。どのような方法であれ、最初に主要となるモデルを開発してしまえば、その後のプロジェクトでは、認識され開発された標準構造を広めていくだけである。この原則から、今まで対象としていた範囲を広げ、もっとドメインを拡張していくこともできる。例えば、魚のモデルであれば、漁業対象区域の魚全部だけではなく、漁船、漁民などをモデルに取り込んでいけるし、法律事務所のモデルであれば、顧客、弁護士だけではなく扱う事件の分野まで広げていける。保健医療モデルであれば、保健医療システムだけではなく、患者、医者、治療、病院まで広げていける。都市モデルであれば、どのような都市であれ、犯罪者、犯罪の被害者、警察が存在する。似たような状況では、因果関係の多くも共有化できる。漁船で魚を獲るのだし、医者が患者を治療する。対象が違えば、数値や関係の強さは違ってくるが物理的な部分は共有化できる。

この論理的、自己検証的なアジャイル・プロセスでモデル開発を進めていけば、あまり経験がない SD 専門家も、まだ初期段階でも十分インパクトがある経験を積むことができるであろう。もっと複雑なモデル開発を行う場合は、経験豊かな同僚が開発した標準テンプレートを使って仕事を始めることができるであろう。複数の関係者との定性的な議論をファシリテートするための才能や経験はもはや重要な要素ではなくなる。また、成果が十分確認された事例を使うことは、SD 利用者にとって、実際にどのようなことがそこから得られ、結果として作成されたモデルは実際に稼働し、有用であることに確信を持つことができる。

SD モデルのプレゼンテーション

アジャイル・プロセスを支援する原則は、SD のプレゼンテーションともかかわっている。通常のモデリング

プロセスの最後のステップでは、モデルの分析結果や政策提言を関係者に発表し、また、公表する場合には、関係者以外にも説明することになっている。国際会議での発表や、多くの SD プロジェクトの報告書や書籍、記事、発表スライドでは、データがきちんと記載されていないものが多い。これは、因果関係図においてばかりでなく、スロック・フロー図においても数値データについては記載されていないことが多い。数量的結果は、ほとんどの場合因果構造と完全に切り離されて説明されるので、プロジェクト関係者の定性的な考えに基づくと、専門家が説明しているそれらの図から、そのような結果が導き出されたのかが信じられないことがある。図 13 に、事例として取り上げた IT サポート企業の因果関係図を使ってこのことを説明した。

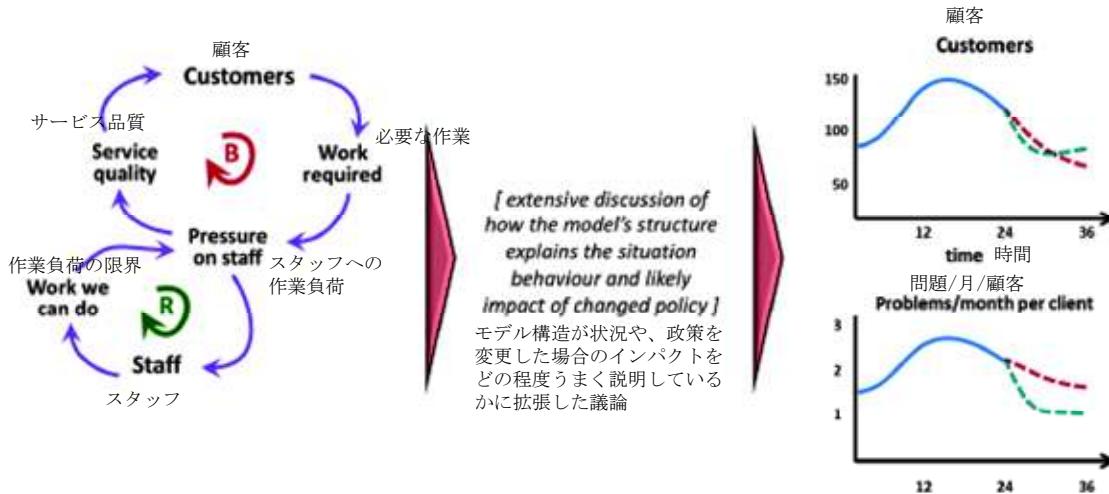


図 13 別途実施したモデル構造と業績改善との関係の説明

このプレゼンテーションのやり方自体が問題を含んでいる。第 1 に、例えこのようなプレゼンテーションの主目的が、どのようにシステムの構造が、結果の振る舞いを引き起こしていることを説明するものであっても、構造を示すスライドと振る舞いを示すスライドが分かれ、しかもその間に何枚もの説明用のスライドが挟まれる場合、目的を妨害しているだけである。繋がりがあるということを理解させるためには、最低限、構造と振る舞いを同じスライドに表示すべきである。

第 2 に、たとえ単純なフィードバック構造であっても、人々は感覚的にその振る舞いを捉えられないので、定性的な構造とその振る舞いと一緒に示しても、多くの読者には感覚的に、その 2 つがどう関係しているのかを理解できないであろう。

例え、多くの人々が論理を理解できた、あるいは、構造が大まかな振る舞いのパターンを結果として生成するのだということを何となく把握したとしても、説明資料には定性的な情報しか示されていないので、因果関係のメカニズムの関与の規模が分からぬ。例に示した図 13 では、左側に示した因果関係図からは、この会社はもう 5 社顧客を増やせるのか、それとも 5 社ではなく 50 社なのか、あるいは、スタッフの不足は 2 人なのか 20 人ののかが分からぬ。雇用率を 2 倍にすべきなのか、3 倍にすべきなのか、いつ増やせばいいのかが分からぬ。

知見はモデル構築プロセス全体から見出すべきものであり、かつ、その結果である改善実施計画は数量化されたもので策定されるべきものであることから、顧客との対話において説明で使われる全ての図もまた全て数量的であるべきである。さらに、われわれの関心は、業績が時間経過でどのように変化するかにあるので、説明に使うグラフでは因果関係の道筋に沿って、要素の動的変化を示すべきである。どんなものであれ、だれにとっても、定性的な因果関係の構造を基に感覚的に振る舞いを予測できないという、広く認められた事実を考慮するならば、時間挙動図抜きには SD を図で説明できないであろう。もちろん、時間挙動図を全ての要素について示したらただ煩雑なだけになるが、構造上主要な要素、例えばストックとそれに関連するフローについては最低限、時間挙動図を付けて示すべきである。

第 3 に、分析でも積み重ね的に実施され、対処すべき問題も、核となる部分から初めて、その周辺に徐々に広げ、複雑していくという方法を探査していることから、図 2 で示したように、ストックに関係する別の要素を追加する、さらには別のストックを追加していく、その結果が、図 3 から図 8 までのように変化するという説明

の仕方を、発見したことの説明に関しても、積み重ねるようなやり方を採択すべきであろう。因果関係図でも似たようなやり方で完成していくのだが、SD プロジェクトでは、積み重ね的に説明するのではなく、最後の完成したものだけを説明し、また説明も口頭あるいは記述的なものであり、SD に馴染みがない人には決して総合的に理解できるようなものとなっていない。

最後に、メンタル・モデルでは、どの関係者が意思決定を行い、あるいはどの関係者が実行するのかという要素に関してはほとんど焦点を当てていない。例えば、システム・ダイナミックス・レビュー誌の過去 10 年の投稿論文をレビューしてみたが、因果関係図の中で意思決定に焦点を当てたものは皆無であった。因果関係図を引用し、いかに意思決定に使われたかとか、どのような政策がなされたかという一般的な説明はあるのだが、しかし、モデルの中に意思決定のことが記載され、問題の当事者が、システムの振る舞いにどのように影響を及ぼしたかという点では全く不明確だった。

結論

システム・ダイナミックス関係者の間に、潜在的に SD は問題に適切に対処でき、その価値が高いにも係らず、その他の方法に比べ、適用が落ちていること、この方法のインパクトが限定的であることへの懸念が広がっている。このことは、SD の実践での適用がどのような方法でなされているのか、どのように関係者とコミュニケーションしているかという問題が問われているように思う。このような疑問は、現在行われている、若手の SD 専門家に対し重要な文献を与え、研修を施すというやり方に問題があり、根本的にこのやり方を変えるべきであり、それには、今のプロセスを基本的に再構築し、方法の最も基本原則から始めるべきであることを示唆している。

実践でのモデル構築で従うべき原則は、ソフトウェア開発プロジェクトで今や広く採択されている「アジャイル型開発手法」と似たものなる。ソリューションはユーザーと一緒にになって形成され、実際にシミュレーション可能なモデルと、実行可能な洞察が継続して産み出される。何年もこの論文に説明したアジャイル型開発法でモデルを開発してきた経験から、アジャイル型開発法でモデルを開発すれば、早く、効果的に実行可能な提言を提供でき、似たような事例で、以前に開発されたモデル構造と似たような構造のモデルが構築されるので、そのモデルは信頼性があるものになる。言うまでもなく、この方法は、新しい方法ではなく、今までのやり方を大きく変え、時間を大幅に短縮し、必要な労力を可能な限り少なくしながら効果の大きさを可能な限り最大化するやり方ではなく、むしろ、グループ・モデル・ビルディング手法の極端な修正である(Vennix, 1996)と言うことができる。

この方法で行ってきた多くのモデル構築プロジェクトでは、比較的単純なものを基に、それを完成させていくという手法を採択しているので、この方法も、もっと大規模でもっと複雑なもので適用性や有効性を検証する必要がある。幸いなことに、さまざまな関連分野での、大規模で複雑な対象への適応が各分野の専門家によって行われ、その結果も良く知られてきていて、繰り返される構造があるような関係する多くのドメインでの事例に修正によって適用可能ということが分かってきている。

アジャイル・プロセスで新しい問題のモデル化に取り組むことと、すでに良く知られているそして、妥当性が確認されているモデル構造を使うことの 2 つを合わせることで、より早く、より信頼できる価値ある SD モデルを提供することができるかもしれない。そして、現実世界でのこの手法のより広範囲な適用とインパクトに貢献することができよう。この 2 つの手順は付随的なものであるかもしれないが、これによって、まだ経験が少ない若い SD 専門家が科学的にかつ実践的に SD をマスターし、さらにシステム・ダイナミックスの適用を大きく推し進めていけるようになるという大きな利点を持っている。

訳注：

1) これは、Kim Warren の“Agile SD: Fast, effective and reliable.” International System Dynamics Conference, 2014 の翻訳である。

<http://www.systemdynamics.org/conferences/2014/proceed/papers/P1126.pdf>

この論文はいくつかのソースから入手可能であり、著者の経営する Strategy Dynamics 社の HP からも入手可能である。

<http://www.systemdynamics.org/conferences/2015/papers/P1143.pdf>

<http://www.strategydynamics.com/agilesd> (いずれも 2016 年 12 月 1 日現在)。

また、原文では段落ごとに空行が挿入されていることから、本稿もそれにしたがっている。

2) この論文では従来型の SD のモデル開発法とアジャイル型開発法という 2 つのモデル開発法について論じている。しかし、どちらの方法でも、参加型モデル開発法という、関係者を集め、関係者と議論しながらモデルを開発するやり方を前提としている。

3) これは、翻訳者のこの論文全体に関する注釈である。

システムの状態は、通常、モデルの中にある主要なストックの状態で把握する。このようなストックをシステム状態変数と呼ぶ。アジャイル型開発法では、このシステム状態変数に集中し、システム状態変数に主要な変化を与える要素は何かだけを考える。ただし、ストックはそれ自体で変化しなく、必ずフローによって変化する。フローの値がゼロの時は、ストックは変化しないので、ストックが変化するということは、フローに(他のストックも含め)何か値を変化させる変数が関係していることになる。従って、フローに変化を及ぼす変数(つまり、フローをゼロにしないような変数)をどんどん追跡していくばモデルが完成することになる。この方法だと、従来型モデル構築方法で行っていた作業である、システムが対象とする範囲の定義とか、システムをまずサブ・システム構造で設計するといった余分な作業を行う必要がない。ただ単純に、注目しているシステム状態変数に関連するフローに変化を及ぼすような変数を追いかけていくだけいい。この関係を追及していく作業の中で挙がってきた要素も、ストックだとかフローだとかいろいろ細かく気にする必要はなく、その要素がシステム状態変数でありかつストックであるかどうかだけを識別するだけでいい。その要素が例えストックになるものであっても、システム状態変数でないのであれば、とりあえずはストックとして取り扱う必要はなく、当面、単なる変数として取り扱つかっておけばいい。このような意味で、モデル開発の単純化が図れると著者は主張している。

この訳文中にも記載があるが、筆者は、ここで紹介したアジャイル型開発法で SD モデル構築を行うに関して品質管理の観点から、モデルに標準型があり、それを基にすることの重要性を、Kim Warren, "Standard Cases: Standard Structures: Standard Models," International System Dynamics Conference 2014 でさらに詳しく述べている。

4) 翻訳に際しての謝辞

この Kim Warren のアジャイル型開発法による SD モデル構築に関する論文の翻訳に関し、何度も翻訳原稿に細かく目を通してコメントを頂いた内野明教授、及び細かく原文と訳文原稿を突き合わせ、厳密な翻訳チェックを行っていただいた木村誠教授に感謝する。ただ、翻訳の誤りや解釈の違いは全て訳者の責任である。

参考文献

- Abrahamsson P, Salo O, Ronkainen J and Warsta J. 2002. Agile development Methods: Review and Analysis. VTT Publications 478.
- Barlas Y. 1989. Multiple tests for validation of system dynamics type of simulation-models. European Journal of Operational Research 42(1): 59–87.
- Barlas Y. 2014. Unpublished comment, noted in a presentation to the Asia-Pacific Conference of the System Dynamics Society in Tokyo, February 2014.
- Carter D, Moizer JD. 2011. Simulating the impact of policy on patrol policing: introducing the emergency service incident model. System Dynamics Review 27(4): 331–357.
- Coyle RG, Exelby DR. 2000. The validation of commercial system dynamics models. System Dynamics Review 16(1): 27–41.
- Cronin M, Gonzalez C, Sterman JD. 2009. Why don't well-educated adults understand accumulation? A challenge to researchers, educators and citizens. Organizational Behavior and Human Decision Processes 108: 116–130.
- Cronin M, Gonzalez C. 2007. Understanding the building blocks of system dynamics. System Dynamics Review 23(1): 1–17.
- Fisher D. 2010. Modeling Dynamic Systems. ISEE Systems: Lebanon NH.
- Ford A. 2009. Modeling the Environment: An Introduction to System Dynamics Models of Environmental Systems. Island Press: Washington DC.
- Forrester J, 2013, Economic Theory for the New Millennium, System Dynamics Review 29(1): 29-41.
- Forrester JW, Senge PM. 1980. Tests for building confidence in system dynamics models. In System Dynamics: TIMS Studies in the Management Sciences, Legasto AA, Forrester JW, Lyneis JM (eds). North-Holland: Amsterdam: 209–228.
- Forrester JW. 2007. System dynamics—the next fifty years. System Dynamics Review 23(2–3): 359–370.
- Fowler M. 2001. Is Design Dead? in Succi G and Marchesi M (eds). Extreme Programming Explained. Addison-Wesley: Boston.
- Franco LA, Rouquette EAJA. 2011. Decision development in facilitated modelling workshops. European Journal of Operational Research 212: 164–178.
- Graham AK. 1980. Parameter estimation in system dynamics modeling. In Elements of the System Dynamics Method, Randers J (ed). Productivity Press: Cambridge, MA: 143–161.
- Haselton MG, Nettle D, Andrews PW. 2005. The evolution of cognitive bias. In DM Buss (Ed.), The Handbook of Evolutionary Psychology. Wiley: Hoboken NJ.
- Homer J, 2013, The aimless plateau, revisited: why the field of system dynamics needs to establish a more coherent identity, System Dynamics Review 29(2): 124-127.
- Huz S, Andersen DF, Richardson GP, Boothroyd R. 1997. A framework for evaluating systems thinking interventions: an experimental approach to mental health system change. System Dynamics Review 13(2): 149–169.
- Kahneman D, Slovic P, Tversky, A. 1982. Judgment under uncertainty: Heuristics and biases. Cambridge University Press: Cambridge MA.
- Lane DC. 1994. Modeling as Learning: A Consultancy Methodology for Enhancing Learning in Management Teams. In Modeling for Learning Organisations. Morecroft JDW, Sterman JD (eds). Productivity Press: Portland OR.
- Lane DC. 2008. The emergence and use of diagramming in system dynamics: a critical account. Systems Research and Behavioral Science 25(1): 3–23.
- Lyneis JM, Ford DN. 2007. System dynamics applied to project management: a survey, assessment and directions for future research. System Dynamics Review 23(2): 157–189.
- Maani KE, Cavana RY. 2000. Systems Thinking and Modelling. Pearson Education New Zealand: Aukland.
- McCartt AI, Rohrbaugh J. 1995. Managerial openness to change and the introduction of GDSS. Organizational Science 6: 569–584.
- Milstein B, Homer J, Hirsch G. 2010. Analyzing National Health Reform Strategies With a Dynamic Simulation Model. American Journal of Public Health 100(5): 811–819.
- Morecroft JDW. 1982. A critical review of diagramming tools for conceptualizing system dynamics models. Dynamica 8(1): 20-29.
- Morecroft JDW. 2007. Strategic Modelling and Business Dynamics. Wiley: Chichester UK.
- Moxnes E. 2000. Not only the tragedy of the commons: misperceptions of feedback and policies for sustainable development. System Dynamics Review 16(4): 325-348.
- Moxnes E. 2004. Misperceptions of basic dynamics: the case of renewable resource management. System Dynamics Review 20(2): 139-162.
- Paich M, Peck C, Valant J. 2009. Pharmaceutical Product Branding Strategies: Simulating Patient Flow and Portfolio Dynamics. Informa Healthcare: New York NY.

- Paich M, Peck C, Valant J. 2011. Pharmaceutical market dynamics and strategic planning: a system dynamics perspective. *System Dynamics Review* 27(1): 47-63.
- Paich M, Sterman JD. 1993. Boom, bust, and failures to learn in experimental markets. *Management Science* 39(12): 1439-1458.
- Peterson DW, Eberlein RL. 1994. Reality checks: a bridge between systems thinking and system dynamics. *System Dynamics Review* 10(2-3): 159-174.
- Professor Jay W Forrester, 2013, Fireside Chat, Conference of the International System Dynamics Society, July 24th.
- Pruyt E. 2013. Small System Dynamics Models for Big Issues. TU Delft Library: Delft, Netherlands.
- Richardson GP. 1986. Problems with causal-loop diagrams. *System Dynamics Review* 2(2): 158-170.
- Schwaninger M, Groesser SN. 2009. System dynamics modeling: validation for quality assurance. In *Encyclopedia of Complexity and System Science*. Meyers RA (ed). Springer: Berlin: 9000-9014.
- Scott RJ, Cavana RY and Cameron D, 2014, Evaluating immediate and long-term impacts of qualitative group model building workshops on participants' mental models, *System Dynamics Review* (forthcoming).
- Sterman J. 2000. People Express Management Flight Simulator (2000 edition). Strategy Dynamics Ltd: Princes Risborough: UK.
- Sterman JD. 1989. Modeling managerial behavior: misperceptions of feedback in a dynamic decision making experiment. *Management Science* 35(3): 321-339.
- Sterman JD. 1994. Learning in and about complex systems. *System Dynamics Review* 10(2): 291-330.
- Sterman JD. 2000. *Business Dynamics*. McGraw-Hill: Boston MA.
- Sterman JD. 2010. Does formal system dynamics training improve people's understanding of accumulation? *System Dynamics Review* 26(4): 316-334.
- The Balanced Scorecard Institute. 2013. Building & Implementing a Balanced Scorecard: Nine Steps to Success.
<https://balancedscorecard.org/Resources/TheNineStepstoSuccess/tbid/58/Default.aspx>; Retrieved 16-3-2014.
- The International Council on Systems Engineering (INCOSE). 2011. *Systems Engineering Handbook v3.2.2*. INCOSE: Los Angeles CA.
- The Supply Chain Council. 2013. The SCOR Framework. <https://supply-chain.org/scor>; Retrieved 16-3-2014.
- Vennix JAM, Schepers W, Willems R. 1993. Group model building: what does the client think of it? In Proceedings of the 1993 International System Dynamics Conference. System Dynamics Society: Chestnut Hill, MA.
- Vennix JAM. 1996. *Group Model Building*. Wiley: Chichester UK.
- Wolstenholme EF. 1994. A Systematic Approach to Model Creation. In *Modeling for Learning Organisations*. Morecroft JDW, Sterman JD (eds). Productivity Press: Portland OR.
- Yücel G, Barlas Y. 2011. Automated parameter specification in dynamic feedback models based on behavior pattern features. *System Dynamics Review* 27(2): 195-215.