



商品設計ではなぜ同じような失敗を繰り返すのか Why do we repeat identical mistakes in product design?

福島史郎 (Fukushima, Shiro)
オムロン株式会社
shiro_fukushima@omron.co.jp

Abstract : The electronics components manufacturers are promoting various kinds of design process improvement activities, because quality in the designing process is important when designing products to satisfy the customer's individual requirements. In some cases the quality is not improved in spite of such efforts. The goal of this paper is to clarify the cause of the problem and propose the solution. In this paper two hypotheses are derived in analyzing the problems in the designing process. (1) Uneven distribution of design knowledge causes a bottleneck in the design process capability. (2) The standardization of the design process causes a state of "Being used to designing" to an engineer and it causes a failure. A virtual business model is developed to simulate the business performance by analyzing the mechanism causing the failures. As a result, these 3 conclusions are provided. (1) In the long run, process improvement activities increase the value of the business because the establishment of the process is effective in decreasing failures. (2) In the short run, improvement is not effective in suppressing failures due to insufficient definitions of specifications and incomplete reviews of the designs. As a solution, the following three actions are suggested: (1) train engineers to become experts who are familiar with the business, (2) training for managers to perform reasonable resource distribution, and (3) a consciousness reform of the engineer in hope of breeding professionalism.

キーワード： システムダイナミクス、開発、設計、経営モデル、プロセス、品質、生産性、コスト、SPI、CMMI

要旨：顧客の求める仕様に合わせて個別に設計を行なうタイプの部品メーカーでは、設計品質が重要であり、商品開発現場はさまざまな改善活動に取り組んでいる。このような改善努力にも関わらず設計品質が向上しないことがある。本論文では、その原因を明らかにするとともに解決策を明らかにした。過去の重要な失敗事例について因果ループ図を用いて原因分析することにより、①組織内の設計ナレッジの偏在を背景とする設計活能力 (Capability) のボトルネック化、②標準化の背後で進行する技術者の慣れを仮説した。要因別に失敗発生メカニズムをモデル化するとともに、それを組み込んだ仮想経営モデルを開発しシミュレーションを実施した。その結果、改善活動には、①長期的には制度手順起因欠陥の減少が優勢となり事業の現在価値を向上させる効果があるものの、②短期的に優勢となる仕様定義能力起因欠陥、デザインレビュー起因欠陥の増加を抑制する効果は無く、③長期的には慣れ起因の欠陥の増加も課題となることがわかった。解決策として、①ドメインエンジニアの早期育成、②適正なりソース配分を期する管理者教育、③プロ意識の醸成を期する技術者の意識変革を提案する。

1. はじめに

1.1 問題意識

メーカーにとって設計活能力は重要である。とりわけ顧客の求める仕様に合わせて個別に設計を行なうタイプの部品メーカー (以下、都度設計型部品メーカー) では、最終製品の品質 Quality・コスト Cost・納期 Delivery (QCD) に加えて、商品設計プロセスが生み出す「設計情報」のQCDが重要である。その理由は次の3つである。

- ① 誤った設計情報は、製造プロセスにおいて誤った情報のまま材料に転写され不良品が製造される。顧客からの信用失墜や商品在庫、仕掛り在庫の廃棄にもつながる。(品質)

¹ 活能力: Capability の意味である。Kim Warren [1] の第9章に Building the Capability to Perform としてその概念が紹介されている。個人が保有する様々な能力 (Skill) と区別して、組織が一定の目的の遂行のために保有する能力を Capability と呼ぶ。本論文では、Capability を目的遂行に向けて活性化された能力との意味で活能力と称する。

② 都度設計型部品メーカーでは、顧客毎に商品設計するので、汎用品メーカーに比べて商品設計コストの割合は高くなる。従って商品設計コストの低減は経営上重要な課題である。(コスト)

③ 都度設計型部品メーカーの商品設計は、顧客が行う商品開発と同時並行で実施されることが多い。顧客が行う商品開発のスケジュール管理の側面からも納期厳守は勿論のこと、その前倒しが強く望まれる。(納期)

都度設計型部品メーカーでは、先ず商品開発により商品設計情報を創出し、その後顧客毎あるいは顧客の商品毎に、数十種類から数百種類の商品設計を繰り返して行われる。それらの設計情報にもとづいて、それぞれの顧客専用の製品を製造し販売する。一連の商品設計を高いQCDで遂行できる設計活能力は事業成果を左右する重要な要素である。

そこでこの設計活能力に関する問題解決や記述のために実在する企業(以下、A社)の都度設計型部品ビジネスを参考にして仮想経営モデルの作成を試みた。

1. 2 商品設計部門における失敗と課題

商品設計における失敗とは、商品設計部門が生み出した設計情報に瑕疵があり、顧客クレームを起すことを行う。経営に対する失敗の影響は重大である。経営に対する失敗のインパクトは次の2つに分類できる。一つは、失敗の対応に要するさまざまなコストである。もう一つは、顧客のメンタルモデルにおける品質視点でのポジショニングの低下である。近年、企業の社会的責任に対する関心の高まりに対応して、後者の観点がより重要になりつつある。特に社会的影響の大きい商品やそれに組込まれる部品の場合、失敗への対応を誤れば事業の継続が困難になる。

A社では、商品設計における失敗低減の重要性に鑑み、さまざまな打ち手を行ってきた。例えば、技術部門ではソフトウェア設計を中心としたベストプラクティスであるCMMI²にもとづくソフトウェアプロセス改善(SPI)を推進してきた。そのようななか、商品設計における失敗の件数は漸減傾向を示しているものの、その努力に見合う程度には十分な効果を生んでいるといえない状況である。このような状況を打開し、より効果的・効率的に商品設計における失敗の低減を図ることが課題である。

2. 分析と仮説

2. 1 失敗の根本原因

A社では一定の基準を定めて重要な失敗を定義したうえで、重要な失敗については本社スタッフ部門に報告することとなっている。本社スタッフ部門に報告された重要な失敗について、2000年度から2004年度までの5年間の報告書を調査した。報告書から失敗の原因として明示されているキーワードを抜き出して特性要因図を用いて整理した。そこから、失敗の根本原因として次の6点が明らかになった。

- ① 縦割り組織の伝言ミス：設計プロセスが組織によって分断されていることから生じたコミュニケーションミスに起因する失敗である。例えば、SEと設計部門の間で設計すべき商品の要件に関して十分に内容が伝わらなかったような場合である。
- ② プロ意識の欠如：技術者が自らに課せられた責任を狭く理解して業務を遂行したことに起因する失敗である。例えば、プログラマーが仕様書に書かれた事項のみをプログラムし、仕様の背景を考慮しなかったか、疑問について確認しなかったような場合である。
- ③ 技術の伝承：事業分野固有の知識(以下、ドメイン知識)の伝承不足に起因する失敗である。例えば、当業者の間では常識的な事項なので仕様書には明示的に記載されない事項がある。これらの知識が不足したために、仕様書の内容を誤って理解したような場合である。
- ④ ドメインエンジニア³の不足：ドメイン知識を習得し、それを活用できる熟練技術者が不足しデザインレビュー等、失敗を回避するための活動が充分でなかったような場合である。
- ⑤ 慣れによるチェック不足：良好な結果が長期間続いたことにより、「今回も多分大丈夫だろう」という心が支配的になり、その結果注意力が低下したことに起因する失敗である。例えば、優秀な設計委託先の成果物として納入された設計結果を十分に検査せずに採用したような場合である。
- ⑥ 制度・手順の形骸化：デザインレビュー等の制度・手順に関して、その本来の目的に対する意識が希薄に

² CMMI : Capability Maturity Model Integration の略称、カーネギーメロン大学ソフトウェア工学研究所が開発したソフトウェア開発組織のベストプラクティスモデルである。組織能力成熟度モデルと訳される。CMMI については参考文献[2]を参照されたい。

³ ドメインエンジニア : ドメインとはソフトウェアのアプリケーション領域のこと。ドメイン知識とは当業者の間で暗黙的に共有されている知識のことである。ソフトウェア開発では要求仕様書に明示的に記載された事項とドメイン知識を合せて始めてソフトウェアは開発できる。A社では、ドメイン知識を習得したレベルの熟練したエンジニアをドメインエンジニアと呼んでいる。

なり、手続きとして会議体を開催することのみに注意が向いてしまうことにより、実質的な内容のレビューが不十分になったような場合である。

以上の6つの原因から、更に本質的な原因を発見するために、因果ループ図を用いた分析を実施した。その結果、不具合の本質的な原因はドメインエンジニアのボトルネック化と標準化の副作用である「慣れ」であることがわかった。

2.2 ドメインエンジニアのボトルネック化

A社の商品設計部門において、ドメインエンジニアは希少なリソースである。ドメインエンジニアは「失敗」の低減に関連して次の3つの役割を担うことが期待されている。

- ① CMMIに基づく標準化の推進メンバ
- ② デザインレビューのレビューア
- ③ 顧客との要求仕様定義作業や要件レビューア

これらの因果ループを図1に示す。「失敗が増加するとCMMIによる標準化によって作業プロセスのパラツキをなくし失敗を減少させようとし、その結果失敗が減る」というバランスループがある。同様に、「失敗が増加するとデザインレビューの実施で設計ミス減らし失敗を減少させようとし、その結果失敗が減る」というバランスループ、「失敗が増加すると要求仕様定義を充分に実施し、顧客との間の仕様に関する認識の不一致をなくし失敗を減らそうとし、その結果失敗が減る」というバランスループが存在する。

「これらの失敗を減らす努力の増加の結果、ドメインエンジニアの需要が高まり、ドメインエンジニアの供給の制限により、必要なドメインエンジニアが割当てできなくなり、失敗を低減するための活動に対するドメインエンジニアの充足率が低下し、その結果失敗が増える」という加速ループが存在する。要するに、「失敗が増加した結果、CMMIにもとづくSPIによる標準化、デザインレビューの充実、要求仕様定義作業の充実等でドメインエンジニアのリソースを消費する活動レベルが高まり、その結果ドメインエンジニアのリソース不足が発生し、却って失敗を増やす」という加速的な因果ループが見出される。

2.3 標準化の副作用としての「慣れ」

CMMIにもとづくSPI活動を一言でいえば、「商品設計プロセスを定義し、それを組織で共有し、全員がその定義したプロセスを全員が遵守するように、科学的かつシステムティックなアプローチで現場を変えてゆく」というものである。その結果、作業のパラツキがなくなるとともに、共有したプロセスを全員で改善することにより失敗を低減できる。これら一連のSPI活動をエンジニアの立場で見れば、個々のエンジニアの果たすべき役割が組織により明確に定義されたうえで、決められたプロセスを確実に遂行することが求められる。当初は改善余地が多くあるから、エンジニアもプロセスの定義やその改善のためにその注意力は適切に維持される。しかしながら、改善項目は徐々に減少する⁴。最後には改善の余地がなくなり遵守すべきプロセスは固定化され、エンジニアのメンタルモデルのなかに業務に対する「慣れ」が発生する。「慣れ」は注意力散漫を招き、失敗を増加させる。この因果ループを図2に示す。

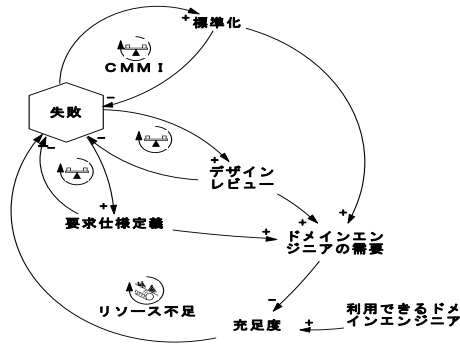


図1 ドメインエンジニアリソースと失敗の因果ループ図

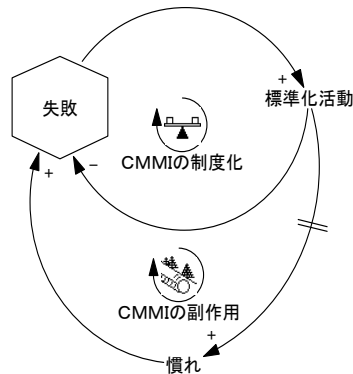


図2 標準化による慣れと失敗の因果ループ図

⁴改善項目の低減とそれによる慣れの発生は、事業環境の変化速度によっても度合いが変化する。

3. 仮想経営モデル

前節で設定した仮説を検証するために、システムダイナミクスを用いて仮想経営モデルを構築する。以下構築したモデルについて順次説明する。モデルは本節末にまとめて図示する。

3.1 市場モデル

商品設計プロセスのインプット量(新規発生テーマ数)は、「顧客毎の平均テーマ数」と「顧客数」で規定される。「顧客平均テーマ数」は、A社の顧客がA社の製品に対して抱く「魅力度」で決まる。「魅力度」は、A社の製品のQCDの状況が競合と比較してどの程度であるかで決まる。A社は、顧客の商品開発フェーズから顧客と協力して商談を進めることにより、顧客との間に長期的パートナーシップを確立している。そのため、製品のQCDの状況が直ちに顧客の獲得や喪失に結びつくことは稀である。しかし、QCDの状況は「魅力度」に影響を与え、「魅力度」の状態は長期的視点で「顧客数」に反映される。モデルの対象領域は、「潜在顧客数」はA社が認識している潜在顧客であり、顧客の業界への新規参入等で発生する可能性のある潜在顧客は含まない。また、「魅力度」も品質と納期のみで決まるものと簡略化している。このモデルを図3に示す。

3.2 商品設計と製造モデル

製造業では原材料を購入、製造し付加価値をつけて製品として販売する「製造プロセス」が存在する。都度設計型のビジネスモデルでは、製造プロセスへのインプット量(受注量)は、「開発済みテーマ数」で決まる。つまり、「商品設計プロセス」のアウトプットである「開発済みテーマ数」が組織のパフォーマンスを決定する重要なドライバになっている。このモデルを図4に示す。

3.3 商品設計マネジメントモデル

商品設計プロセスがどの程度のパフォーマンスを生むかは、商品設計部門のマネジメントに影響される。商品設計部門のマネジメントのメンタルモデルを支配するのは、次の4つのプレッシャである。

- ① 仕事へのプレッシャ・・・アウトプットの量が目標値に達していないとき、「エンジニアに残業させて、もっとたくさん働かせよう」という心理⁵
- ② 改善へのプレッシャ・・・組織の中長期的なことを考えると「リソースの一定量を改善活動に当たらせ、もっと生産性をあげよう」という心理⁶
- ③ 品質へのプレッシャ・・・「品質に対する危機意識からデザインレビュー活動へのアテンションをもっと当てよう」とする心理
- ④ 顧客引き合いからのプレッシャ・・・「顧客からの新規テーマの引き合いに応えるべく要求仕様定義活動へのアテンションを当てよう」とする心理

商品設計部門のマネジメントは、これら4つのプレッシャのバランスの中で、要員の割当てを行っている。モデルの対象領域は、考慮すべきプレッシャをアウトプットの量と失敗の量に関係する項目に限定している。このモデルを図5に示す。

3.4 スタッフと活能力モデル

商品設計部門には2種類の重要な活能力がある。一つは開発量を出力する「エンジニアリング活能力」である。もう一つは失敗を減らす活能力であり、具体的には「デザインレビュー活能力」と「要求仕様定義活能力」である。

- ① 「エンジニアリング活能力」とは、商品設計組織が、設計情報を創出する能力であり、単位工数(1人月)が生み出すことができる設計情報の量をテーマ数として測ったものである。単位は、テーマ/人月である。ただし、実際に完了できるテーマ数は、「エンジニアリング活能力」に対して一定量の「ドメインエンジニアリング活能力」が不可欠であるという制約がある。これは、エンジニアの業務を指導・監督するのはドメインエンジニアであり、ドメインエンジニアをエンジニアで代替することは不可能だからである。
- ② 「デザインレビュー活能力」とは、デザインレビューを実施し、失敗が発生しない程度に設計エラーを取り除くために必要な活能力である。単位工数(1人月)が適切に設計エラーを除去できる設計情報の量をテーマ数で測ったものである。単位は、テーマ/人月である。
- ③ 「要求仕様定義活能力」とは、将来失敗の原因にならない程度にまで顧客との仕様上の認識の差異を除去す

⁵ Nelson P. Repenning and John D. Sterman [3] らはこの心理を“[Work harder](#)”と紹介している。

⁶ Nelson P. Repenning and John D. Sterman [3] らはこの心理を“[Work smarter](#)”と紹介している。

⁷人月(にんげつ)とはソフトウェアの開発現場で良く使われる仕事の量を表す単位である。ソフトウェアの開発コストを見積もる際に使われる。当該カテゴリに属する平均的な能力を持った要員が1か月間(実労20日程度)従事したときの仕事量である。「このソフトウェア開発にはシステムエンジニアが1人月、プログラマーが6人月必要」といった使われ方をする。

る活能力である。単位工数 (1 人月) が適切に顧客との差異を除去できる設計情報の量をテーマ数で測ったものである。単位は、テーマ/人月である。

モデルの対象領域は、失敗の量に関する要素に限定している。このモデルを図 6 に示す。

3. 5 「制度化活能力」と「慣れ」モデル

「制度化活能力」とは、組織が設計業務の標準化を進め、設計プロセスをマニュアル等の形で共有させ、組織内のメンバにマニュアル等に従って業務遂行させる活能力である。標準化が進みマニュアルがあると、無い場合に比べて業務遂行が円滑化しアウトプットされる設計情報の量が増える。これはマニュアルが無い場合に発生する試行錯誤が減るからである。このように、「制度化活能力」はアウトプットされる設計情報を増減するという意味では「エンジニアリング活能力」と等価である。また、失敗の原因となる設計エラーを効果的・効率的に除去できるという意味では「デザインレビュー活能力」と等価である。また、顧客との要件に関する共通認識を効果的・効率的に得ることができるという意味では「要求仕様定義活能力」と等価である。一般的に、制度化の裏では「慣れ」が発生する。「慣れ」は失敗を増やし、組織としてアウトプットできる有効な設計情報の量を減らすという意味で、「デザインレビュー活能力」や「要求仕様定義活能力」と逆の作用をするから負の活能力と考えられる。モデルの対象領域は、失敗の量に係る制度化ならびに慣れに限定している。このモデルを図 7 に示す。

3. 6 失敗の要因モデル

2.1 節で述べた根本原因分析のなかで明らかになった、失敗の要因を次に掲げる 4 つに絞り込んでモデル化した。

- ① 仕様定義能力起因欠陥 設計に先立って、顧客の要求仕様を十分に定義しなかったことが原因となる欠陥
- ② デザインレビュー起因欠陥 設計途中で実施するべきデザインレビューが十分になされなかったことが原因となる欠陥
- ③ 制度手順起因欠陥 設計プロセスが十分に定義されていない、あるいは徹底されていなかったことが原因となる欠陥
- ④ 慣れ起因の欠陥 設計プロセスの標準化が進んだことにより、技術者のメンタルモデルに「慣れ」が発生し注意力を低下させたことが原因となる欠陥

これらのうち、①、②については、要員充足率と欠陥の期待値の間の相関関係を定義することによりモデル化した。充足率とは、要求定義作業やレビュー作業に必要なドメインエンジニアの工数に対して、実際に割当てられたドメインエンジニアの工数の比率である。③は標準化された制度手順の量と欠陥の期待度との相関を定義することによりモデル化した。また、④は「慣れ」の量と欠陥の期待値との相関を定義することによりモデル化した。このモデルを図 8 に示す。

3. 7 財務モデル

Kaoru Yamaguchi [4] の提案したシステムダイナミクスによる会計システムのモデルにもとづいている。今回構築した仮想経営モデルの中で定義された全ての活動の財務的データを財務諸表に反映したものである。モデルの対象領域は、一般的な財務諸表に記載される項目を網羅している。このモデルを図 9 に示す。

3. 8 パラメータの調整

パラメータの調整は、A社に対するコンサルティング活動の中で経験によって得られた値にもとづくものである。重要なものについては、同僚のコンサルタントとのディスカッションの結果、コンセンサスに至ったものを採用している。

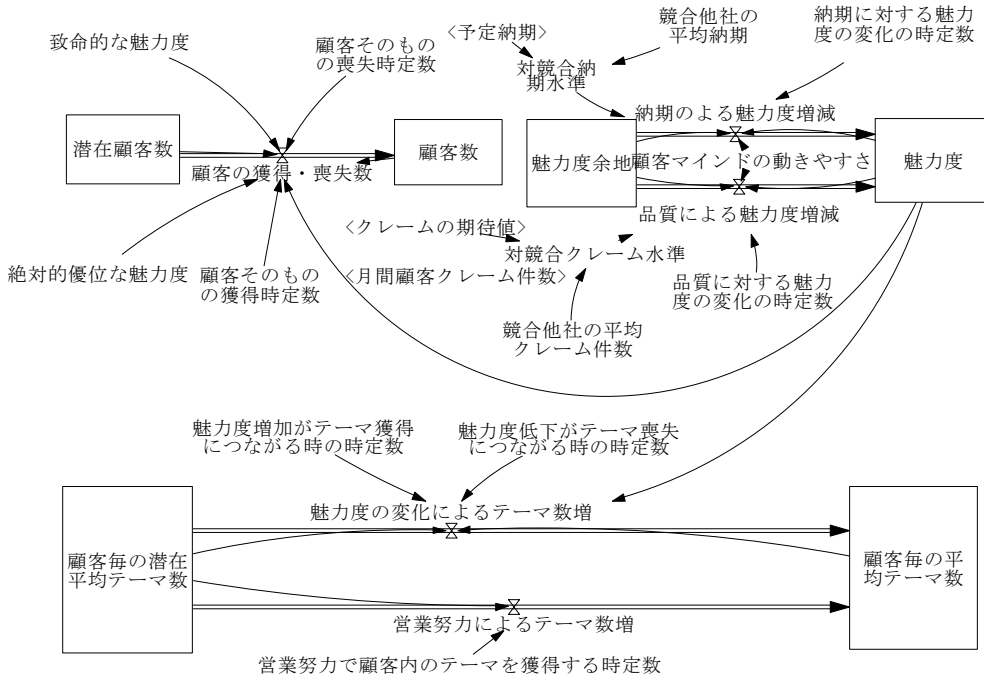


図3 市場モデル

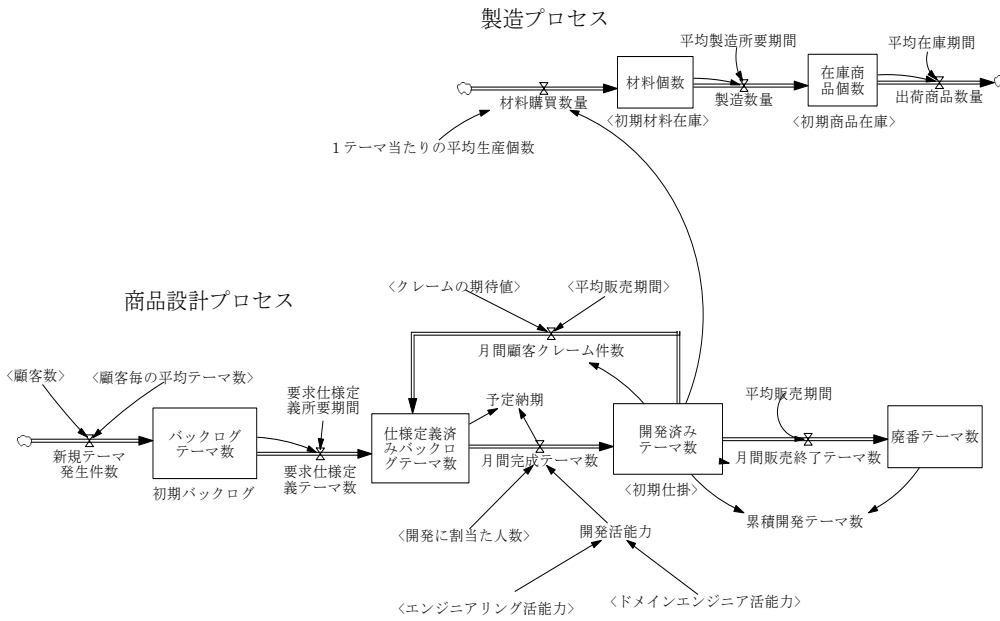


図4 商品設計と製造モデル

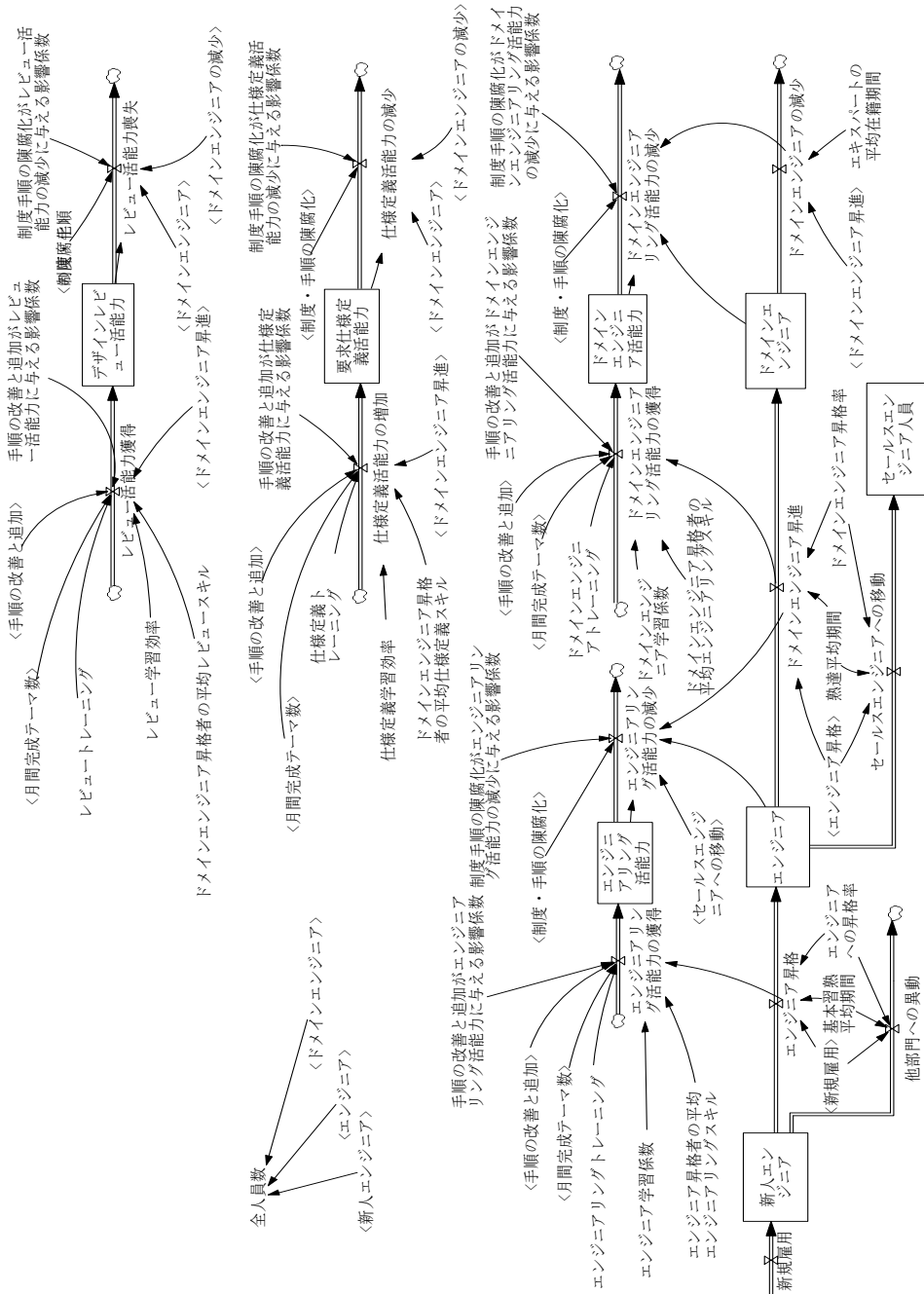


図6 スタッフと活能力モデル

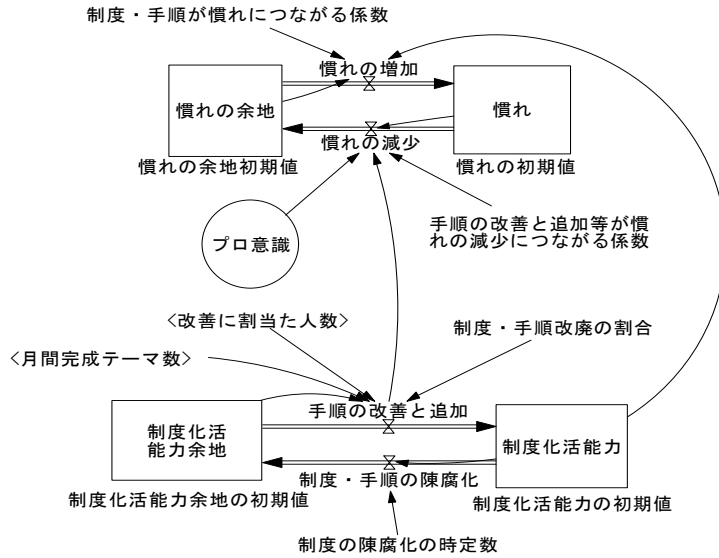


図7 制度化活能力と「慣れ」モデル

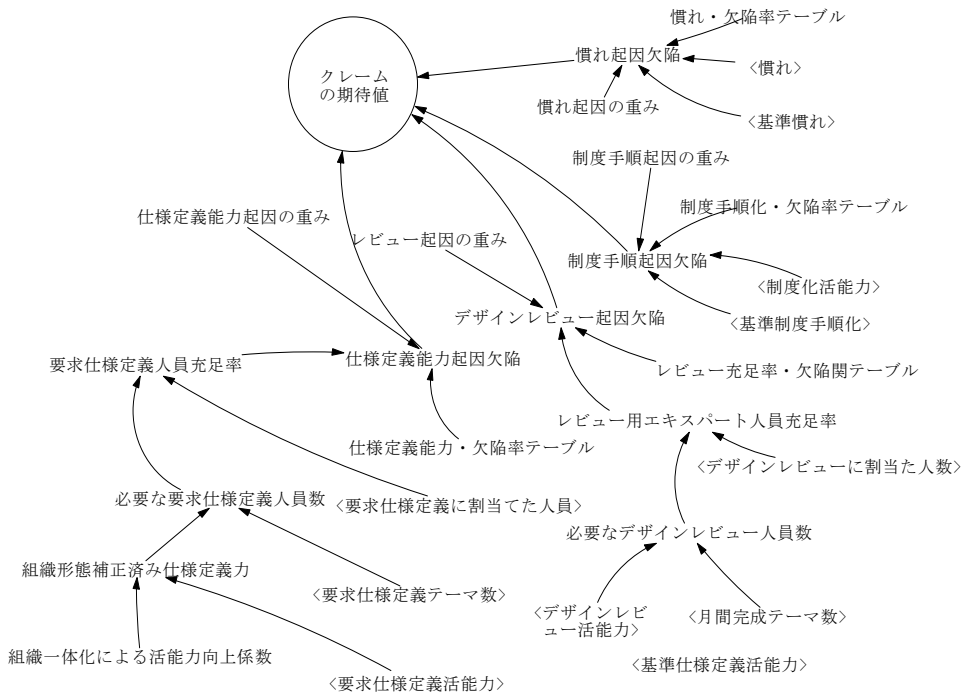


図8 失敗の要因モデル

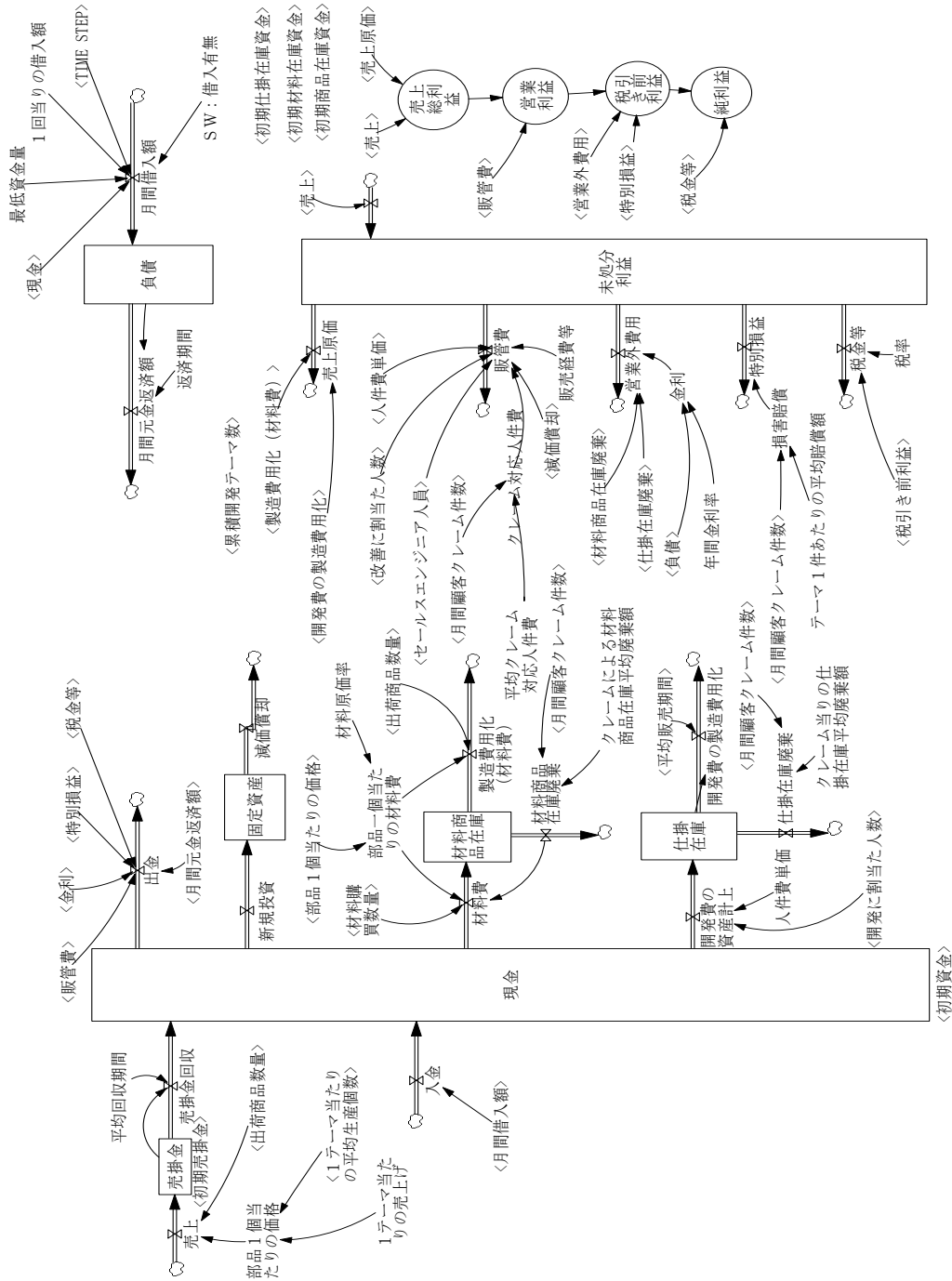


図9 財務モデル

エンジニアのジョブ・ローテーションにかかわる人事政策の見直し等で実現できる。エンジニアからドメインエンジニアへの昇格率を16%にした場合のシミュレーションの結果を図10、図11において Run2 として示す。

Run2における事業体の正味現在価値は16,891百万円となりRun1に比べ大幅に改善する。クレームの期待値を更に低減させるための方策を考察するため、Run2のクレームの起因別グラフを図13に示す。

4.3 マネジメントの見直し

図13によると、「デザインレビュー起因欠陥」が24ヶ月付近から100ヶ月付近にかけて高い状態で継続している。また、「仕様定義能力起因欠陥」が100ヶ月以降で急増している。原因は、デザインレビュー要員や要求仕様定義要員の充足率が低いことである。その原因は商品設計部門において、品質プレッシャが強いときは十分にデザインレビュー要員を割当てるものの、品質プレッシャが低くなったときは、デザインレビューへの要員割当てを減らすというマネジメントを行っているからである。「喉もと過ぎれば熱さ忘れる」マネジメントポリシーを改め、品質プレッシャが低いときもデザインレビュー等のための要員を必要量割当てに続けるという、「当たり前のことを当たり前」にやる」マネジメントポリシーに変えれば、「デザインレビュー起因欠陥」によるクレームの期待値の高原状態は改善する。Run2に対して、デザインレビューへの要員割当てに加えて要求仕様定義への要員割当て、改善活動要員割当てに関するマネジメントポリシーを改善した場合のシミュレーションの結果を図10、図11のRun3として示す。Run3における事業体の正味現在価値は26,115百万円となる。このことから、クレームの低減をドライブする業務に対して継続的にドメインエンジニアを必要量割当てに続けることは、たとえそれが事業量のボトルネックになっているような希少リソースであり、目先の事業量の減少というデメリットを伴うとしても、事業体の正味現在価値という長期的視点では十分に合理性を説明できる判断であるということがわかる。ボトルネックを考察するため、Run3におけるクレームの起因別グラフを図14として示す。

4.4 「慣れ」の抑制

図14によると、クレームの期待値の全体水準を下げ止まらせている原因は、「慣れ起因欠陥」である。標準化により業務の制度手順が整った結果、「制度手順起因欠陥」は減少している。しかしながらその副作用として、エンジニアのメンタルモデルのなかに「慣れ」が発生し「慣れ起因欠陥」を生んでいる。対策として、エンジニアに「プロ意識」を植え付けることにより、制度手順化が進

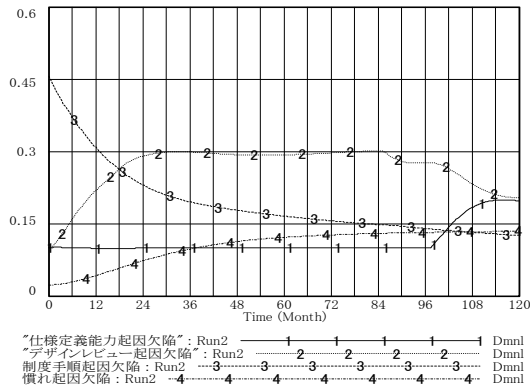


図13 クレームの起因別分析 (Run2)

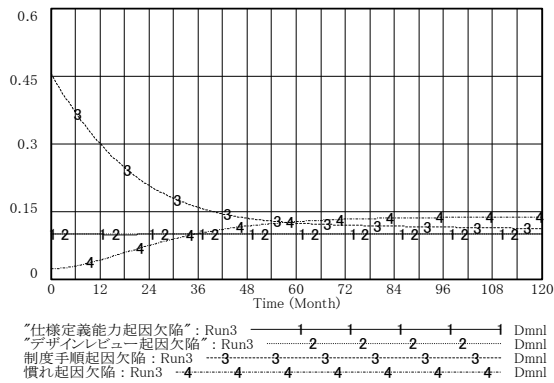


図14 クレームの起因別分析 (Run3)

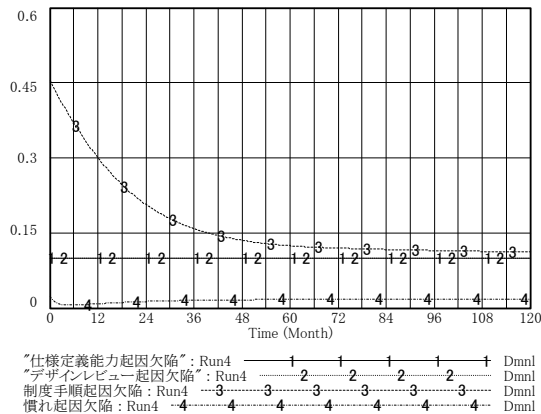
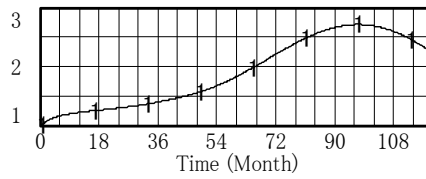


図15 クレームの起因別分析 (Run4)

行しても「慣れ起因欠陥」を低減させることができる。「プロ意識」とは、「制度手順」に従ってさえいれば良いというエンジニアの心理に対抗し、業務の本来目的を追求することにより、「制度手順の改善と追加」を仮想的に行う精神の働きであると考えられる。従って、モデルでは「制度手順の改善と追加」と等価なものと考え、「制度手順の改善と追加」と「プロ意識」の和に対して「慣れ」のアウトフローを発生させている。エンジニアに「プロ意識」を植えつけた場合のシミュレーションの結果を図 10、図 11 の Run4 として示す。Run4 における事業体の正味現在価値は 31,783 百万円となる。ボトルネックを考察するため、Run4 におけるクレームの起因別グラフを図 15 として示す。

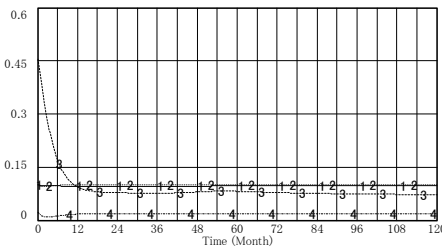
4.5 人員の増加

図 15 におけるクレームの期待値の推移にはシステムダイナミクス上の課題を想起させるような所見はない。一方で、市場モデルにおける対競合納期水準（他社の納期を 1 としたときの自社の納期）を図 16 として示す。Run4 では対競合納期水準が 2.7 倍となり、組織は人員不足に陥っていることがわかる。そこで要員の増員をシミュレーションする。採用計画で最大値を考える方法の 1 つとして、バックログを発生させないだけのリソースを投入することを考えてみる。実際にシミュレーションにおいて、新規雇用を増加させて要求仕様定義済バックログのストックがマイナスにならない限界値を求めると 2.7 人/月となる。エンジニアの新規雇用のレートを現状の 1.5 人/月から 2.7 人/月まで増加させてみる。それにより組織は 300 人体制から 554 人体制となる。この場合のシミュレーションの結果を図 10、図 11 の Run5 として示す。Run5 における事業体の正味現在価値は 61,289 百万円となる。Run5 におけるクレームの要因別グラフを図 17 として示す。



対競合納期(Run4) ———— 1 ———— 1 ———— 1 ———— 1

図 16 対競合納期水準 (Run4)



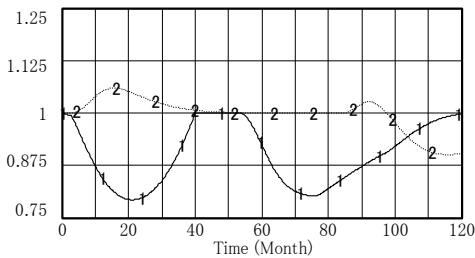
“仕様定義能力起因欠陥”: Run5 ———— 1 ———— 1 ———— 1 ———— 1 Dmnl
 “デザインレビュー起因欠陥”: Run5 ———— 2 ———— 2 ———— 2 ———— 2 Dmnl
 “制度手順起因欠陥”: Run5 ———— 3 ———— 3 ———— 3 ———— 3 Dmnl
 “慣れ起因欠陥”: Run5 ———— 4 ———— 4 ———— 4 ———— 4 Dmnl

図 17 クレームの起因別分析 (Run5)

5. 考察

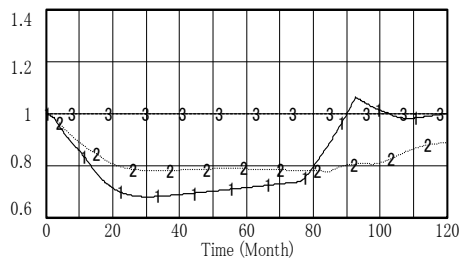
5.1 ドメインエンジニアのボトルネック化

Run1 における不具合の発生要因を探るため、Run1~Run2 における「要求仕様定義人員充足率」を図 18 に、Run1~Run3 における「デザインレビュー人員充足率」を図 19 に示す。



“要求仕様定義人員充足率”: Run1 ———— 1 ———— 1 ———— 1 ———— 1
 “要求仕様定義人員充足率”: Run2 ———— 2 ———— 2 ———— 2 ———— 2

図 18 要求仕様定義人員充足率



“デザインレビュー人員充足率”: Run1 ———— 1 ———— 1 ———— 1 ———— 1
 “デザインレビュー人員充足率”: Run2 ———— 2 ———— 2 ———— 2 ———— 2
 “デザインレビュー人員充足率”: Run3 ———— 3 ———— 3 ———— 3 ———— 3

図 19 デザインレビュー人員充足率

要求仕様定義人員充足率は、Run2 においてほぼ充足状態にまで回復していることがわかる。デザインレビュー

一人員充足率は、Run2 において Run1 より部分的に改善がみられるものの悪化した部分も見られる。そして Run3 においてほぼ充足状態に至っている。このことから、改善活動の効果を出すためには、①ドメインエンジニアの早期育成に加えて、②適正なリソース配分を期する管理者教育が必要であることがわかる。

これらのことを、因果ループ図で表現すると図20で示すとおりとなる。ドメインエンジニアの「リソース不足」因果ループに「計画的人材育成」ループを付加することにより、「失敗」をバランスさせる作用を生じさせている。

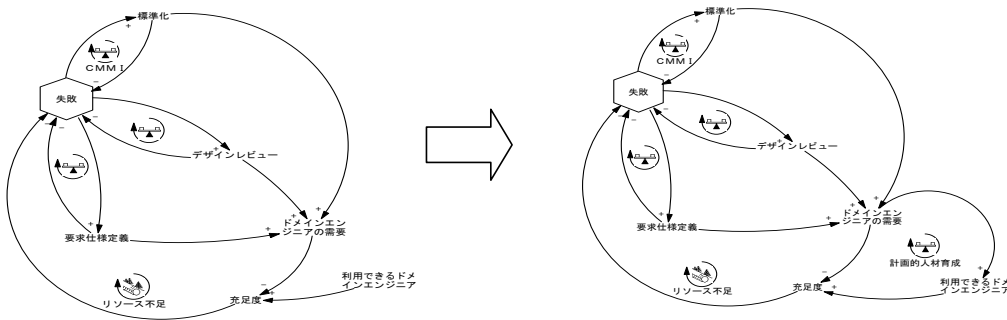


図20 ドメインエンジニアのボトルネックの解消

5.2 ドメインエンジニアの育成が制度遵守の徹底か？

4節のシミュレーションでは、ドメインエンジニアの育成を行った後に制度遵守のマネジメントの徹底を実施した。制度遵守のマネジメントの徹底を先に実施し、ドメインエンジニアの育成はその後に行うという考え方もある。図21は、4節で示したとおりドメインエンジニアの育成(Run2)を実施し、そののち、制度遵守のマネジメントの徹底(Run3)を行ったケースである。それとは逆に、図22は制度遵守のマネジメントの徹底(Run2)を行った後ドメインエンジニアの育成(Run3)を行ったケースである。

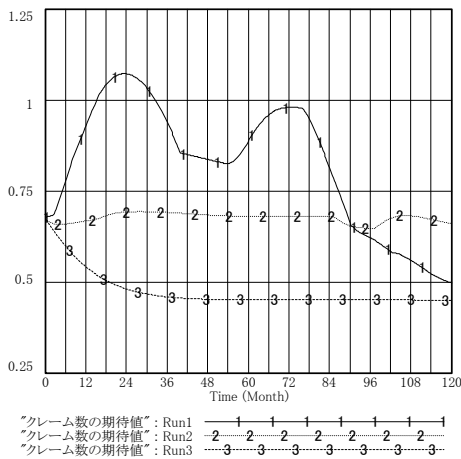


図21 ドメインエンジニアの育成を先行時

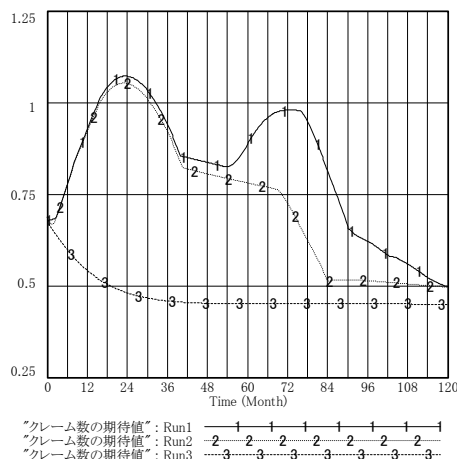


図22 制度遵守のマネジメントを先行時

ここから分かることは、ドメインエンジニアが不足する中で、制度手順の遵守を徹底しようとする短期的に失敗の増加が避けられない。現実的には、ドメインエンジニアの育成と制度遵守のマネジメントも同時進行的に行われることが多い。従って、ここから得られる知見は、ドメインエンジニアが十分な量が確保できていない段階では、制度手順のマネジメントの徹底は慎重に行うべきであるということである。マネジメントが示す「あるべき姿」に対して、現場が追従しきれずに不具合を増加させてしまうことを防ぐことが重要である。

5. 3 慣れの進行

「失敗」の発生要因の1つである「慣れ」の発生メカニズムは、図2に示したとおりである。シミュレーション結果は、Run3における「慣れ起因欠陥」の増加が、Run4において増加が見られなくなったことから、プロ意識の強化が有効であることがわかる。これらのことを、因果ループ図で表現すると図23で示すとおりとなる。図2に示した「標準化活動」の副作用として生じる「慣れ」による「欠陥」を起こす「CMMIの制度化」ループに対して、「失敗」に対しては「ペナルティ」を科すことにより「失敗」を減少させる「信賞必罰」ループを付加する。このことは「信賞必罰」ループによって「慣れ」を抑制し、「失敗」をバランスさせる作用を生じさせることを意味する。

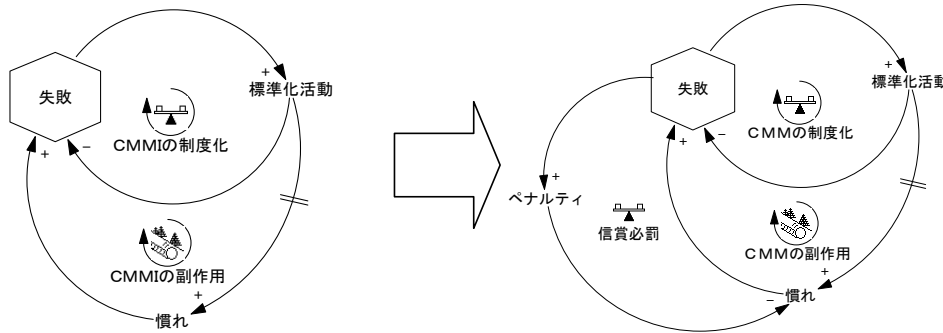


図23 慣れの副作用の解消

6. 結言

因果ループ図ならびにシステムダイナミクスモデルによる仮想経営シミュレーションを用いて、失敗が十分には減らない根本原因はドメインエンジニアの不足にあること、CMMIが強調する制度化の副作用として発生する「慣れ」が品質に悪影響を与え得ることを説明するとともに、解決策としてドメインエンジニアの計画的早期育成、設計活動と改善活動の適正なりソース配分を維持し続けることを期する管理者教育、プロ意識の醸成を期する技術者の意識変革が有効であることを示すことができた。

謝辞

本研究を遂行するにあたり、モデル構築を基礎から親身にご指導戴き、また議論をとおして課題の深堀をご指導戴いた同志社大学大学院ビジネス研究科の山口薫教授に心よりの感謝の意を表する。

参考文献

- [1] Kim Warren, Competitive Strategy Dynamics, JOHN WILEY & SONS, LTD, 2002
- [2] Mary Beth Chrysis, Mike Konrad, Sandy Shrum, JASPIC CMMI V1.1 翻訳研究会訳, CMMI 標準教本, 2005
- [3] Nelson P. Repenning and John D. Sterman, "Nobody Ever Gets Credit for Fixing Problems that Never Happened: Creating and Sustaining Process Improvement", CALIFORNIA MANAGEMENT REVIEW Vol.43, No.4, pp.64-88, 2001
- [4] Kaoru Yamaguchi, Principle of Accounting System Dynamics – Modeling Corporate Financial Statements, International System Dynamics Conference 2003, Proceedings, New York, 2003

付録

主要パラメータ一覧表

パラメータ	Run1	Run2	Run3	Run4	Run5
ドメインエンジニア昇格率	8%	16%	16%	16%	16%
改善人員テーブル 改善活動にアサインする内部工数の割合を示す。					
デザインレビュー人員テーブル 適正値を1としたときのデザインレビューに投入するドメインエンジニアの工数を示す。					
要求仕様定義人員テーブル 適正値を1としたときの顧客との要求仕様定義に投入するドメインエンジニアの工数を示す。					
プロ意識	0.04	0.04	0.04	1.00	1.00
	改善開始時点における「制度手順の改善と増加」と同程度の値を採用			「慣れ」のストックを発生させないために最大値を採用	
新規採用	1.5 人/月	1.5 人/月	1.5 人/月	1.5 人/月	2.7 人/月