

1. 本論文の構成

本論文の構成は以下の通りである。

はじめに
本論文の目的
本論文の意義
本論文の特徴
調査の概要
中心的主張
本稿の構成

第1章 本論文の問題意識と課題

- 1.1. 問題意識
- 1.2. 研究課題の設定

第2章 既存研究の検討

- 2.1. 既存の製品開発論
- 2.2. ソフトウェア産業
- 2.3. 知識創造理論と状況論
- 2.4. 再設定した研究課題

第3章 概念枠組みと調査対象

- 3.1 概念枠組み
- 3.2 調査対象
- 3.3 調査方法

第4章 事例研究

- 4.1. インタビューの方法
- 4.2. 事例整理の方法
- 4.3. 事例記述の概略
- 事例01～事例09 (9つの事例の紹介)

第5章 分析と考察

- 5.1 定量的分析
- 5.2 定性的分析
- 5.3 類型化に関する考察
- 5.4 変化のダイナミズムに関する考察
- 5.5 その他の副次的な発見事項に関する考察

第6章 結論と含意

- 6.1. 目的は達成されたか
- 6.2. 理論への含意
- 6.3. 実務への含意
- 6.4. おわりに(ナレッジ・エンジニアリングの今後)

付録 インタビュー・フォーム
追加調査の質問票(電子メール)
参考文献

2. 本論文の目的

この論文は、製品開発における部門内相互作用について考察した研究をまとめたものである。この論文では、コンピュータ・ソフトウェア(以後「ソフトウェア」と略す)の製品開発プロジェクトを題材として取り上げ、以下の研究目的を達成しようという試みがなされる。

目的1: 製品開発を「関係者の知識が相互作用を経て対象製品の中に再構成されていくプロセス」としてとらえ、部門内相互作用のパターンを類型化する。

目的2: 部門内相互作用のパターンが変化していくダイナミズムを明らかにする。

目的3: プロジェクト・リーダーが部門内相互作用を適正化しようとしておこなう行為の性質を明らかにする。

目的1に示した「関係者の知識が相互作用を経て対象製品の中に再構成されていくプロセス」という観点
は、組織論における「知識創造パラダイム」に立脚している。この論文では、「知の分解と合成」という観点から、部門内での分業や調整、学習について考察していく。この観点を成立させているのは、製品を「機能のシステム」とする前提である。この論文で題材として取り上げるソフトウェア製品は、知の形式化の程度が高いので、「サブ・システムとしての機能が特定の構造で結合されているシステム」という製品観を持ってその開発プロセスを眺めると、知識が組み合わさっていくプロセスが浮かび上がってくる。このような観点でとらえた製品開発プロセスを、この論文では「ナレッジ・エンジニアリング・プロセス」と呼ぶ。

この論文においては、「部門内相互作用」を、「分業」、「調整」、「学習」の3つの点からとらえる。この部門内相互作用のパターンが変化していく、あるいは違うパターンへ移行する際の、動的過程と原動力を明らかにするというのが目的2の文の主旨である。製品開発に関する既存研究の多くにおいては、何らかの次元で製品開発のありようを類型化し、それらを開発成果に与える影響という点で静態的に比較することに焦点が置かれてきたため、パターンの変化や移行といった動的過程はあまり取り上げられてこなかった。

目的3は、部門内相互作用を適正化しようとしてプロジェクト・リーダーがおこなうマネジメント行為について記述し、その行為の性質を明らかにしようとするものである。ソフトウェア開発プロジェクトの成功と失敗を分ける要因のなかで、最大のものは、「経験豊富な管理者」を持つか否かであるといわれている。この論文では、経験豊富なリーダーと経験の浅いリーダーを分けるものは、変化と異常に対応することのできる技能であると考え、開発プロジェクトの現場におけるリーダーの行為を、「状況的行為」という概念を頭に入れながら、観察・記述する。「状況的行為」とは、「すべての行為のコースは本質的なあり方で、物質的・社会的な周辺環境に依存したものだ」という見方を強調する概念である。この概念を提唱する「状況論」では、人間の行為はその場の周辺環境とその変化に即興的に対応するものであり、次に取るべき行為の可能性が明らかになるのも、ある行為をとることによって周辺環境と相互作用しているときのことであって、決して前もって予期できるものではない、と考える。したがって、人間がとる一連の行為は、前もって立てられた「プラン(計画)」に制御されているのではなく、即興的に作りだされていく、と考えるのである。リーダーのマネジメント行為を、前もってプログラム化されていた手順の実行であったり、プランに制御されているものであるとみなすのではなく、状況的行為とみなして接近し、理解しようと、この論文は試みた。

製品開発を「ナレッジ・エンジニアリング・プロセス」と見る観点、そしてプロジェクト・リーダーの行為を「状況的行為」と見る観点で、この論文全体を貫く2つの観点である。そして、この論文の主たる特徴として挙げることができるのは、(1)製品開発を「知識創造活動の部門内分業」という観点から調査分析した(おそらく数少ない)研究であること、(2)ケースヒストリーによる知識の伝承と、将来の研究への資料提供を意識した記述をしていること、そして(3)被観察者それぞれの間にある認識の違い、および被観察者と観察者(研究者)の間にある認識の違いを前提としていること、の3点である。

以降は各章ごとに概要を述べる。

3. 第1章 本論文の問題意識と課題

第1章では、研究に取り組む際の問題意識を紹介し、この論文で設定する課題を明らかにする。問題意識は、次の3点である。

問題意識1: 製品開発プロセスのデザインを類型化することが試みられているが、見過ごされている次元があるのではないか。

問題意識2: 製品開発プロセスのデザインに、「正解」はあるのか。なぜ多数のモデルが存在しているのか。

問題意識3: プロジェクト・リーダーは「経験」によって何を心得るのか。得たものはどのような行為にあらわれるのか。

製品開発研究の主流は、産業横断的アプローチから、産業を絞ったフィールド調査へと移行してきた。後者において、多くの研究者が着目し、知見の蓄積が特にすすんでいると思われるのが、自動車産業や家電産業である。このように産業が限定されたことで、見過ごされている次元、あるいは十分に議論されていない論点が存在しているのではないか、というのがこの論文の第1の問題意識である。

第2の問題意識は、製品開発プロセスのデザインに唯一無二の正解、ないし最適な方法というものは存在するのだろうか、という疑問である。唯一無二の正解を否定したコンティンジェンシー理論においても、特定条件下での「最適な組織状態」の存在は想定されていた。この考え方からすれば、多様なデザインは、その均衡解にむけて収斂していくことになる。ところが現状をみると、開発プロセスのデザインについては、実に様々なモデル(以後「プロセス・モデル」と呼ぶ)が提唱されつづけており、収斂というよりはむしろ多様化の方向に向かっていくように思える。そして、多数のモデルが存在していることを説明するには、最適デザインへの収斂過程という枠組みから離れなければならない。そして、これに代替する研究枠組みとして考えられるのは、開発プロセスのデザインがどのように変化するか、という動的過程を把握するような枠組みである。

第3の問題意識は、プロジェクト・リーダーの熟練に関するものである。これまでの熟練に関する研究は、主に職人や技術者を対象とするものであり、人とモノの関係を中心に考察されてきた。しかし、人と人の関係においても、経験によって得られる「何か」が存在するはずである。しかし、その「何か」はどのような場面で発揮され、その促進要因、阻害要因にはどのようなものがあるのかというような問いに対しては、まだ明確な答えが得られていないばかりか、それを調べるための方法も、まだ確立されていない。

これらの問題意識と研究目的から、この論文で遂行しようと試みる研究課題が設定された。それは以下の5つである。

研究課題1: ソフトウェア製品開発プロセスを、部門内相互作用に着目して記述する。

研究課題2: 記述から特徴的なパターンを見出し、いくつかの次元を抽出して吟味する。(以上の2つは研究目的1に対応)

研究課題3: 部門内相互作用のパターンが変化する動的過程を明らかにする。

研究課題4: 変化の原動力となっているのは何であるのかを推測する。

(以上の2つは研究目的2に対応)

研究課題5:ベテラン・リーダーたちが開発途中で即興的におこなったマネジメント行為のうち、部門内相互作用に影響を与えたものを記述する。(この研究課題は研究目的3に対応)

4. 第2章 既存研究の検討

第2章第1節では、製品開発プロセスのデザインを対象とする既存研究が、どのような図式に基づいて議論をしてきたのか、その図式の背後にある論理は何であるのか、について整理した。ここでは、開発段階間の重複を指す「ステージ重複」の次元において、「順次プロセス vs. 重複プロセス」という図式で議論してきた諸研究が、議論の背景として「調整」の論理を持っていることを示した。そして、「調整の緊密度」から「調整のタイミング」に論理の焦点が移るにつれて、「初期調整」が「初期計画」と読み替えられ、「初期計画の強調 vs. 試行錯誤的反復」という、背景論理に「学習」を持つ新たな議論の図式が誕生したことを示した。さらにこの節では、自動車産業を対象とする研究でも、最新のものは、分業、調整、学習という3つの観点を重視していることも明らかにした。

第2章第2節では、この論文が題材として取り上げるソフトウェア産業を概観し、論考のための準備を整えた。そして、ソフトウェア工学における議論の流れを、プロセス・モデルを中心としてフォローし、一時期支配的であったウォーター・フォール・モデル(以後「WFモデル」と略す)からの離脱を示す現象が観察されるようになってきていることを示した。さらに、従来型の工業製品に対するソフトウェア製品開発の特殊性を、製品開発プロセスのデザインと絡めて論じ、(1)製造業ほど機能別の分業と専門化が進んでいないこと、(2)「設計工程」と「製造工程」の連携をほとんど必要としないこと、(3)最終製品が物理的構造物ではなくデジタル・データであるため、出荷後の修正や機能追加が容易であること、の3点を指摘した。この節の最後では、開発プロセス管理の難しさは、技術的なアプローチだけではなかなか解決できず、人間的・組織的なアプローチが重要であることを述べた。

第2章第3節では、この論文全体を貫く2つの観点を提供する「知識創造理論」と「状況論」について、両者をそれぞれ情報処理パラダイムと対比しながらレビューした。

第2章第4節では、第1章で設定した研究課題を、既存研究の知見をふまえながら再設定したものを提示した。それは以下の5つの研究課題である。なお、ここでは、以降の調査分析においてWFモデル以外のモデルを「anti-WFモデル」と総称することを定めた。

(再)研究課題1:ソフトウェア製品開発プロジェクトを題材として、ナレッジ・エンジニアリングの集団プロセスを、分業、調整、学習という観点から記述する

(再)研究課題2:anti-WFモデルを特徴づける次元は何であるかを特定する。(既存研究で指摘されている「ステージ重複」と「サイクル反復」に限らない)

(再)研究課題3:anti-WFモデルの採用理由と影響を、採用者たちから聞き取り、知見を得る。

(再)研究課題4:知見を総合し、anti-WFモデルへの移行を説明する論理を導く。

(再)研究課題5:ベテラン・リーダーたちが、どのような視点からプロジェクトの文脈をとらえ、どのような状況的行為をおこなったか、ということ記述する

5. 第3章 概念枠組みと調査対象

第3章では、本論文の調査と分析に用いる概念枠組みを示した。この概念枠組みは、「事前モデル」、「事後モデル」、「集団プロセス」、「マネジメント」、「環境認識」の5つの要素から構成されている。この枠組みの最大の特徴は、WFモデルやanti-WFモデルを人々の頭の中にある「プロセス・モデル」としてとらえ、既に終了した開発プロセスを理解・省察するために事後的に用いられる場合には「事後モデル」、これから開始される開発プロセスの方向付けをする戦略や構想として用いられる場合には「事前モデル」というように概念的に区別している点である。これ以降で「WFモデル」または「anti-WFモデル」と言うとき、特に断りのない限り、それは事前モデルのことである。続いてこの章では、調査対象となるソフトウェア開発会社の概要を述べ、その事業と組織について説明した。最後に、調査の方法について説明した。この研究で中心としたのは、プロジェクト・リーダー24人を対象とするインタビュー調査である。

6. 第4章 事例研究

第4章では、WFモデルに準拠したソフトウェア製品開発プロジェクトと、anti-WFモデルに準拠したソフトウェア製品開発プロジェクトの両方でリーダーを務めた経験のある人たちに語ってもらった事例9編を収録している。調査実施順に事例は収録されている。それぞれの事例のリーダーとanti-WFプロジェクトでの開発製品名は次のとおりである。

事例01:リーダーA氏(C++仕様書工房2)

事例02:リーダーB氏(SIMPLIA/ensemble YPSオプション)

事例03:リーダーC氏(ICAD受託)

事例04:リーダーD氏(NetShop V1.0)

事例05:リーダーE氏(NT版ListWORKS V5)

事例06:リーダーF氏(GkitOCX)

事例07:リーダーG氏(Digitalbook Creator)

事例08:リーダーH氏(HPF V20L10)

事例09:リーダーJ氏(Contact Scope)

それぞれの事例は、次のような順序に沿って記述されている。

(1)リーダーのキャリア

- (2)比較対象としてのWFモデル・プロジェクト
- (3)製品の概要
- (4)プロジェクト開始にいたる経緯
- (5)開発メンバーと分業方式
- (6)プロジェクト開始から終了までの経緯
- (7)リーダーとして気づいたこと(感想)

7. 第5章 分析と考察

第5章第1節では、第4章で紹介した9事例について、「垂直的分権度」と「仕様凍結時期」という指標を用いて定量的分析をおこなった。「垂直的分権度」とは、リーダーとメンバーの間の責任と権限の分担を調べるための指標である。垂直的分権は、「問題発見における垂直的分権」と「選択解決における垂直的分権」に分けて調査がおこなわれた。「問題発見」とは、顧客要求を発見することである。ここでは、開発途中での問題発見や機能提案も含む「要求仕様の明確化」に、メンバーをどれだけ参加させようとしたかを尋ねた。一方の「選択解決」とは、発見された問題に対処することである。機能を作りこむか積み残すかを決める取捨選択をおこない、必要に応じて要求をソフトウェアに作りこみ機能を実現していく、問題解決のことである。こちらでは、リーダーが中心となって選択解決するようにしたのか、それともメンバーそれぞれが選択解決をするようにしたのかを尋ねた。そして、この垂直的分権度の分析結果から推測できた、リーダーとメンバー間の分業パターンは、次のようにまとめられた。

- (1) WFモデルに準拠するプロジェクトでは、問題発見はリーダーを含む少数のメンバーがおこなう。そして、メンバーが発見した問題は、ほぼ必ずといっていいほどリーダーが選択解決の指揮をとる。
- (2) anti-WFモデルに準拠するプロジェクトでは、問題発見は過半数のメンバーがおこなう。選択解決は、WFモデルほど絶対的ではないが、依然リーダーの指揮のもとにあることが多い。

次に、「仕様凍結時期」という指標を用いた分析をおこなった。「仕様凍結時期」とは、開発期間のどのタイミングで最終仕様が固定されたのかをあらわす指標であり、調整のタイミングや学習のありようを推測するための指標である。仕様凍結時期の測定には、次の式を用いた。(式のなかの「入試」とは、製品が出荷前の最終品質検査のために品質検査部門に渡されるというイベントのことを指す。通常はこのイベントによって、製品が開発プロジェクト・チームの手から離れることになる)

$$(\text{仕様凍結時期}) = (\text{開始から仕様凍結までの期間}) \div (\text{開始から入試までの期間})$$

探索的分析によると、WFプロジェクトとanti-WFプロジェクトの仕様凍結時期の間には差がありそうだったので、ノンパラメトリック検定(Mann-Whitney test)をおこなった。その結果、有意水準 $\alpha=0.01$ で、有意性が認められた。このような、「事前モデル(WFとanti-WF)」と「仕様凍結時期」の関係が疑似相関である可能性も考え、「プロジェクト人員数」、「開発リードタイム」、「プロジェクトの年代」という3つの変数についても、WFプロジェクトとanti-WFプロジェクトとの間で差の検定をおこなった結果、これらの変数が第3の変数で、事前モデルと「仕様凍結時期」の関係が疑似相関であるという可能性は低いことがわかった。以上の分析から、WFプロジェクトとanti-WFプロジェクトについて次のようなことが明らかになった。

- (1) WFモデルに準拠するプロジェクトでは、開発の前半で仕様が凍結されることが多い。
- (2) anti-WFモデルに準拠するプロジェクトでは、開発期間の8割から9割がた終わったところで仕様が凍結されることが多い。

第5章第2節では、事例の記述を分析することで、分業、調整、学習というそれぞれの側面において、WFプロジェクトとanti-WFプロジェクトを対比する次元が抽出された。まず、分業の側面から抽出されたのは、(1)分業の単位、(2)権限委譲、(3)熟練の割合、の3つであり、これらを総合して、WFプロジェクトを「工程別分業」、anti-WFプロジェクトを「モジュール別分業」というように対比した。次に、調整の側面から抽出されたのは、(1)公式化、(2)調整の頻度、(3)人格依存度、の3つであり、これらを総合して、WFプロジェクトを「公式的調整」、anti-WFプロジェクトを「非公式的調整」というように対比した。最後に、学習の側面から抽出されたのは、(1)行動性、(2)現場性、(3)自発性、(4)迅速性、(5)意外性、の5つであり、これらを総合して、WFプロジェクトを「計画型学習」、anti-WFプロジェクトを「試行錯誤型学習」というように対比した。

第5章第3節では、部門内相互作用を類型化するための考察に入った。定量的分析と定性的分析での発見事項を整理するために、考察のための枠組みが新たに構築された。その枠組みとは、機能のシステムとしてのソフトウェア製品を成り立たせている「構造」と「機能」を縦軸にとり、それらが明確化していくプロセスのスタート段階である「発見」とゴール段階である「実現」を横軸にとったマトリクスである。

この枠組みを用いて、リーダーとメンバーの間の分業パターンを、WFプロジェクトとanti-WFプロジェクトで比較して、anti-WFモデルにおける分業の構図を相対的に形成した。WFプロジェクトとの比較でみると、anti-WFモデルに準拠するプロジェクトでは、垂直的分権度が高くなっていた。なかでも特に問題発見の垂直的分権度が高く、選択解決の垂直的分権度は、問題発見ほどの大きな違いはなかった。問題発見の垂直的分権度が高いということから、anti-WFプロジェクトではメンバーが「機能の発見」に関与していることが推測できた。また、選択解決の垂直的分権度に大きな違いがなかったということから、WFプロジェクトでもanti-WFプロジェクトでも、機能を選択してその諸機能の間の関係性を決定する「構造の実現」は、リーダーが担当する領域であることが推測できた。そして、anti-WFプロジェクトでは、リーダーが特定の機能モジュールを担当して持ち、自らコーディングもおこなう、という例が多かったことから、リーダーが「機能の実現」に関与する傾向があらわれていることが推測できた。

さらに、同じ枠組みを用いて、製品がどのような順序で形作られていくかということも考察した。WFモデルでは、まず(1)製品についての大まかなイメージが形成され(「構造の発見」)、次に(2)製品システム全体の構造

が定義される(「構造の実現」)、(3)これとほぼ同時に製品を構成する単位(コンポーネント、モジュール、機能)が定められ(「機能の発見」)、最後に(4)諸機能が作り込まれる(「機能の実現」)という経路をたどっていると推測した。これに対してanti-WFモデルでは、まず(1)製品についての大まかなイメージが形成され(「構造の発見」)、次に(2)製品を構成する単位が暫定的に定められ(「機能の発見」)、(3)暫定的な単位が動作可能になるように作り込み(「機能の実現」)、最後に(4)機能の取捨選択を通して製品システム全体の構造が決定する(「構造の実現」)という経路をたどっていると推測した。以上の製品開発順序の相違を、考察のための枠組みを用いて図示し、WFプロジェクトが「Z字型」の軌跡を残す進行であるのに対して、anti-WFプロジェクトは「U字型」の軌跡を残す進行であることを示した。そして、模式図を用いて開発の流れをあらわし、さらにリーダーとメンバーの間の分業についての所見も考慮して、次のような一般化を試みた。

(1) WFモデルに準拠するプロジェクトでは、初期段階でリーダーが創造した製品全体に関する知識(イメージ、設計)を分解していき、メンバーはその分解された知識の具体化を遂行する。

(2) anti-WFモデルに準拠するプロジェクトでは、メンバーもそれぞれの担当部分で知識を創造し、それらの知識が合成されることで最終的な製品全体が姿を現す。

以上の分析と考察の結果導き出された両者の特徴を総合して、WFモデルに準拠するナレッジ・エンジニアリングの部門内相互作用パターンを「知識分解型」、そしてanti-WFモデルのそれを「知識合成型」と類型化した。両者の差異を、ひとことで際立たせるためのメタファーを挙げるなら、WFモデルからくる部門内相互作用パターンは「ジグソーパズル」であり、anti-WFモデルからくるそれは「粘土細工」である。

第5章第4節では、以上にみたような部門内相互作用パターンの違いを生み出す原因についての考察に入り、プロジェクト・リーダーの環境認識とマネジメント行為に目を向けた。環境認識についての考察では、anti-WFモデル準拠の理由として不確実性の高さが挙げられ、マネジメント行為としては「計画による調整」よりも「フィードバックによる調整」が指向されていることが示された。そして、不確実性の高さという環境認識と、フィードバックによる調整を指向するマネジメント行為の間の関係は、情報処理パラダイムが想定するような「情報処理能力拡充戦略」によってのみ説明されるのではなく、知識創造パラダイムに立脚した「知識創造の促進」という解釈も可能であることを、調査の分析と考察の結果を考慮しながら明らかにした。

ここから、知識合成型の集団プロセスが、プロジェクト・リーダーのマネジメント行為と相互に影響を与えあっているのではないか、という仮説を立てて調査結果を再検討した。そして、WFモデルに準拠した際のマネジメント行為が定型的で予定調和的な色彩が強いのにに対して、anti-WFに準拠した際のマネジメント行為は非定型的で即興的であるようにみえること、そしてこれは「状況的行為」が、事前モデルという「プラン」に邪魔されることなく発現している姿であるように解釈できるということを示した。

anti-WFプロジェクトにおけるマネジメント行為の本質が「状況的行為の発現」にあるとするならば、これに影響を与える次元こそが、ダイナミズムを説明する上で、事前モデルを特徴づける最も重要な次元となるはずである。そこで、「拘束力」という次元を導入した。これは、「どの程度、モデルに基づく行為を強要するものであるか」という次元である。事前モデルの拘束力が強いというのは、事前モデルが「プラン」として扱われ、開発プロセスにおけるリーダーやメンバーの行為を強く規定するようなときである。一方、事前モデルの拘束力が弱いというのは、事前モデルが「行為のリソース」として扱われ、開発プロセスにおける状況的行為が許容・是認されるようなときである。anti-WFモデルに準拠するプロジェクトで状況的行為が発現しているという観察から、WFモデルは拘束力が強いモデルであり、anti-WFモデルは拘束力の弱いモデルである、と解釈する。

「拘束力」という次元の導入によって、WFモデルからの離脱という現象、すなわち事前モデルの変化は、「拘束力の強い事前モデルから、拘束力の弱い事前モデルへ」という動きとしてとらえなおすことができる。このようにとらえなおすことで、「不確実性の増大を感じたリーダーは、自らの状況的行為を解放できるような状態を整えるために、事前モデルの拘束力を弱めようとする」がゆえにWFモデルから離脱する、という説明論理を構築した。

この論理は、事前モデルと状況的行為の適合度、つまりは事前モデルの「拘束力」を重視する。したがって、拘束力さえ弱まれば、リーダーは状況的行為を容易に発現できるので、モデルの「内容」がWFモデルから変わっていても十分に満足すると考えられる。事例の中からも、anti-WFモデルは、まったく新しいモデルではなく、WFモデルの修正版、改良版である、とする記述を見いだすことができた。仮にどんなに高い不確実性にも適合するような、「優れた」モデルが登場しようとも、その拘束力が高ければ、あるいは拘束力が何らかの理由で高まれば、結局は他のモデルにその地位を明け渡すか、もしくはリーダーによって修正が加えられる、ということが論理的に導かれる。

状況的行為の本質とは、認知と行為が分けがたく結合していることである。WFモデルは、リーダーの認知と行為を無理やり切り離してしまう効果を持っていたと考えれば、WFモデルからの離脱は、切り離されたリーダーの認知と行為が再結合しようとしていることの現れであるといえるだろう。これが、この論文の中心的主張であり、結論である。そして、この主張の一般化を試み、次のような命題を提出した。

命題1: 製品開発における部門内相互作用パターンの変化の原動力となっているのは、「状況的行為発現への要請」である。

命題2: リーダーの状況的行為の発現を許容する事前モデルは、より大きな不確実性に耐えることができる。

第5章第5節では、事例分析から副次的に得られた発見事項から、メンバー個々人のなわばり意識が強まっているとする「NIM症候群」という概念と、非公式型の調整を促進すると考えられる「リム型座席配置」という概念を提案した。前者は、最新のソフトウェア開発者たちの、「他人がつくったプログラムは修正したくないし、自分がつくったプログラムは他人に修正されたくない」という傾向である。後者は、多くのリーダーたちが採用している、「プロジェクト・メンバー同士が背中合わせに位置し、振り向いたらすぐに相手のモニター画面が見え、そこでモニターを見ながら議論できる」ような座席配置である。

第6章では、研究目的がどのように達成されたかを確認することで、以上の諸章で積み上げてきた論考の要約と、この論文の結論に代えた。そして、事前モデルの拘束力という次元を用いて、今後、開発プロセスのデザインに関する研究をする際には、分析単位を個別のモデルに置くのではなく、複数モデルのコンビネーションに置くべきではないか、と理論への含意を述べた。さらに、実務への含意としては次の3つを提言の形であらわした。

提言1:「新モデル」の探索や開発にかかるエネルギーを、事例ベースの構築に向けるべきである。

提言2:リーダーの暗黙的技能を有効に用いようとするならば、これを形式化(文書化、一般化、普遍化)しようとするよりも、状況的行為として発現しやすいように環境を整えるべきである。

提言3:リーダーの疲労に目を向け、これに配慮するべきである。

そして最後に、ナレッジ・エンジニアリングの今後の方向性についての筆者の私見を示して論文を締めくくった。