

氏名	平井 直樹 <small>ひらい なおき</small>
学位の種類	博士 (経営管理学)
報告番号	甲第484号
学位授与年月日	2018年3月31日
学位授与の要件	学位規則(昭和28年4月1日 文部省令第9号) 第4条第1項該当
学位論文題目	ソフトウェア開発プロセスにおける分業構造と知識労働 —日本の受託ソフトウェア開発の組織問題—
審査委員	主査) 亀川 雅人 (立教大学大学院ビジネスデザイン 研究科教授) 品川 啓介 (立教大学大学院ビジネスデザイン 研究科教授) 木村 剛 (立教大学大学院ビジネスデザイン 研究科教授)

I. 論文の内容の要旨

(1) 論文の構成

1. 論文構成 (章と節の全体)

第1章 研究の目的と方法

- (1) 本研究の背景
- (2) 本研究の目的
- (3) 本研究の方法
- (4) 本研究の構成

第2章 ソフトウェア産業の現状

- (1) 日本のソフトウェア産業の現状
- (2)日本のカスタムソフトウェアへの偏重
- (3) 小括

第3章 先行研究と問題設定

- (1) これまでの先行研究との関係
- (2) 日本のソフトウェア産業に関する研究
- (3) 知識労働に関する研究
- (4) 問題設定・残された分析課題

第4章 ソフトウェアの製品設計思想

- (1) アーキテクチャーの分類
- (2) アーキテクチャーの選択
- (3) 小括

第5章 ソフトウェア開発の特徴と手法

- (1) ソフトウェアの開発プロセス
- (2) 開発手法の移り変わり
- (3) 開発手法の違い
- (4) 小括

第6章 ソフトウェア産業の下請け構造と製造工場化

- (1) ソフトウェア産業の成り立ち
- (2) ソフトウェア産業の下請け構造

- (3) ソフトウェア・ファクトリー
- (4) 小括

第7章 ソフトウェア開発の知識労働

- (1) ソフトウェア工学
- (2) ソフトウェア開発の知的作業とその軽視
- (3) 小括

第8章 知識労働の軽視と分業の問題－開発事例研究－

- (1) 調査概要
- (2) 事例1：要件定義、設計、プログラム作成の分業
- (3) 事例2：上流工程と下流工程の企業間分業
- (4) 事例3：要件定義と設計以降の分業
- (5) 小括

第9章 知識労働の軽視と分業の問題－考察－

- (1) 開発事例から明らかになった問題
- (2) ソフトウェア開発の分業構造と問題解決
- (3) ソフトウェア開発の知識労働の見直し
- (4) 小括

第10章 結論と今後の課題

- (1) 結論
- (2) 残された課題

参考文献

(2) 論文の内容要旨

平井氏の研究は、論文タイトルにあるように、わが国の受託ソフトウェア産業における開発プロセスの問題点を指摘する。ソフトウェア産業は、ハードの急速な技術的發展に伴い開発手法を変化させねばならない状況にある。しかしながら、この技術変化のスピードに対応できず、結果として革新的なイノベーションも期待することができない状況に陥っている。その問題点として、ハードとソフトの初期段階に普及した画一的なソフトウェア開発手法を採用していることに原因を求めている。

この開発手法は、設計と製造を分離する画一的な製品製造の工場モデルと類似しており、製造過程の労働が知的労働として位置付けられず、標準化した単純労働に置き換えられる。ソフトウェア開発プロセスにおいて設計工程とプログラム作成工程を分離することで、下流工程の作成工程の労働は、製品製造現場の不熟練労働と同じように労働の質ではなく、マニュアル化した労働の時間により評価されることになる。マニュアル化は、一定期間にわたる労働内容を固定化し、環境変化に柔軟な変動的な対応を困難にする。それは、固定的コストと認識される。その結果、ソフトウェアの開発プロセスは、新興国に輸出されるようなコスト競争に晒されることになる。

本研究では、まず、日本のソフトウェア産業の先行研究を分析し、**Waterfall Model** と呼ばれるこれまでのソフトウェア開発の管理を主体とした手法に問題があることを論じる。ソフトウェア・ファクトリーと呼ばれるソフトウェア開発の製造工場化は、画一化した大量生産モデルであり、差別化したプログラムや新しい事業領域に取り組むような革新的なソフトウェアの開発には向かない。新たな製品の製造が熟練技術者の知的創造プロセスであるように、革新的なソフトウェアの開発プロセスは、ニーズとシーズを結合する試行錯誤的なプロセスである。こうした開発プロセスが完成すれば、それは一般化し、標準化したモデルになるが、その時点では新たな価値の創造は無くなり、価格競争のみのプロセスになる。開発プロセスを見直すことがなければ、ソフトウェアの価値は価格競争により低下し続けることになる。

本研究では、ソフトウェアを含め製品を作り上げるための社内外の役割分担と連携方法、その全体の構成を決める製品設計思想であるアーキテクチャーを考察する。ソフトウェアは機能と性能がハードの中に組み込まれた構造であり、求められる機能や性能、ハードにより変化させねばならない。経営戦略や組織が変化すれば、これに応じたソフトウェアを開発しなければならず、画一的なソフトウェアは顧客の競争優位に繋がらない。一般的な工業製品のように、個々のモジュールが完全に独立した段階では、ソフトウェア開発企業の価値も単なる組み立て工場の価値にしかならない。単純に組み合わせるだけでは、新たな価値を生むことが難しく、顧客自身が発見できないニーズの発見や顧客のニーズの変化に応じた対応が要求される。この対応は、設計と作業工程を分離では実現しない。

新たなニーズは現場で探索され、問題の発見と解決の方法が考察される。開発プロセスにおける下流工程とは、顧客企業との接点であり、問題探索プロセスである。設計と作業

工程が分離され、作業工程における顧客との接点が遮断されれば、作業に従事する開発者は思考が止まり、マニュアルに従うしかなくなる。知的労働の機会は失われ、単純労働に特化することになる。しかし、開発プロセスにおいて、現場との接点を有し、問題探索行動が行われれば、開発プロセスに従事する活動は知的労働になる。開発プロセスの変更は、単純労働を知的労働に変化させ、下流工程の従事者を価値創造に参加させることになる。それは、モジュールそのものの開発やモジュール同士の組み合わせを試行錯誤的に擦り合わせる設計と作業の知的往復運動を意味している。

本研究は、こうした議論を展開するため、ソフトウェアの開発手法として現在も日本で主流となっている **Waterfall Model** の長所と短所を分析し、**Waterfall Model** に代わる **Agile** と呼ばれる海外の開発手法と比較考察する。前者は、コスト競争のための大量生産的な標準化であり、後者は差別化のための開発手法である。

Waterfall Model が支配的な手法となっている日本の現状は、その産業構造にも問題がある。日本のソフトウェア産業の下請け構造を確認するとともに、1970～1990年代にかけてソフトウェア開発手法の中心にあったソフトウェア・ファクトリーの貢献と欠点を論じる。日本のソフトウェア産業が、ハード技術の急速な発展のためにソフトウェア開発の人材を十分に育成することができず、設計部門に知的人材を囲い込み、作業工程は下請け構造の下、労働集約型産業となって、作業技術者の人数を確保することで売上を伸ばす施策が進められる。それは、下流工程を軽視する要因の一つになっている。

しかし、既述のように、こうした開発工程では、ソフトウェア開発産業のみならず、顧客企業の価値創造にも貢献できない。そのため、ソフトウェア開発を知識労働という点から整理する。知的労働とは、マニュアル化した開発手法に従うのではなく、顧客企業との情報交換を通じて、新たな問題の発見、問題の探索を行い、個々の企業の価値創造に繋がる解決手法を開発することである。それは試行錯誤的な開発過程となる。

ソフトウェア開発の問題整理に関わる作業は、技術者の知的な部分に依存しており、そのような知的な部分、つまり創造力を強化することが重要であること、設計段階ですべての問題が解決していることはなく、プログラムを作成しつつ問題解決を行っていくため、設計とプログラム作成の担当者の綿密な意思疎通が必要なことを整理する。特に、ソフトウェアを開発するには、専門性を持った人たちがチームを組んで行う必要があり、作業や工程を分割し、協業していく部分が多く発生するため、そのようなチームや組織では、特定の工程に対する専門性のみならず、ソフトウェア開発の目的を理解し、全体の流れを各開発者が認識する情報共有とコミュニケーションが必要がある。それは、企業の境界を超えた諸機能の相互関連性を理解する幅広い関連知識を必要とする。

本研究では、このような議論を展開するため、ソフトウェア開発の事例研究として、工程間が分業された3つの事例をあげ、その作業プロセスや知的部分の結合において、ソフトウェア開発が成功した事例と失敗事例を分析し、その問題点と解決手法を知識マネジメントの視点より考察している。

ここから明らかにされたことは、価値創造に導くソフトウェア開発は、細分化した作業工程の下流工程が上流工程のタスク処理型工程ではなく、またライン生産方式のような工場における単純な製造作業に類するものでもなく、開発技術や企業の業務など幅広い知識と高い専門性を有する試行錯誤的な研究開発的組織であるということである。

ソフトウェア開発は、現実世界の業務、習慣、文化、風習といった個々の企業、人間で異なるものを対象とするため、そこに現れてくる不確実で複雑な問題に対して、試行錯誤を繰り返していく知識労働であり、研究開発組織である。そして、製造業の製品設計や製品開発のように、Agile 型の開発プロセスでは、度重なる変更と繰り返し性のある組織的な問題発見と問題解決活動が必要とされ、蓄積された知識や熟練といった良質な経験による類推に基づいて対応する能力と、新しく難しい状況でも創造的な問題解決ができる能力の双方が重要とされる。

このような能力は、今までにない問題や新しい領域に取り組んでいくような革新的なソフトウェアの開発に繋がるものであり、それゆえ、特定の作業工程や機能に対する狭い専門性の高い人材だけでは対応できず、チームメンバーや企業の境界を超えるような幅広い共通知識が必要となり、Agile 型開発組織が要求される。Waterfall Model は、キャッチアップ型経済におけるソフトウェア開発の初期モデルに適しているとしても、イノベーティブな企業の創造に対応する現在は、創造的作業者を構成メンバーとする Agile 型開発組織への転換が必要とであると結論づけている。もちろん、新たに差別化するソフトウェアは、ニッチなニーズに対応したものであり、少数メンバーによる小規模な開発に限定される。開発されたソフトが標準化されると、この基本的なフレームワークに基づく開発手法 Waterfall Model となる。標準化のプロセスは、知的労働を単純労働に変換する一般的プロセスである。

II. 論文審査の結果の要旨

(1) 論文の特徴

本論文の特徴は、ソフトウェア開発のプロセスを製造業における大量生産工程と差別化製品の生産工程と類似の視点で分類する。大量生産工程は、マニュアル化した不熟練労働によるコスト競争力を追求する。他方、差別化製品の工程は、特定の顧客にターゲットを絞り、問題探索とその解決を試行錯誤的に繰り返すオーダーメイド型開発である。後者は、それが価値創造につながるとき、他社に模倣されるソフトとなり大量生産工程へと移行する。成熟した経済では、成熟した事業で価格競争をしつつ、新たな成長のための差別化が求められる。成熟事業のソフトと差別化する新しい事業のソフトが質的に異なるとすれば、その開発プロセスも異なることになる。

平井氏の論文は、この点を指摘し、差別化したソフト開発プロセスに知的労働を導入する意義を理論的に整理する。ソフト開発のプロセスは、日本企業のキャッチアップ型成長には大量生産型ソフトの導入が合理的であり、優先されてきた。この開発プロセスは、**Waterfall Model** と呼ばれ、一般的に普及している開発の作業工程である。しかし、成熟した経済では、新しい問題の発見やその解決工法を考案する試行錯誤的な開発が必要になる。これは欧米でみられる **Agile** 型開発組織であり、これを知的労働と位置づける。

Waterfall Model は、日本のカスタムソフトウェア開発における一般的モデルであり、このモデルにおける開発では、ソフトは機能の効率性を追求することができるが、それは価格競争に陥るモデルとなる。製造業における大量生産工程と同じく、分業構造は細分化され、マニュアル化される。ソフト開発は効率化するが、トップダウン型の設計とプログラムが分離され、下流の作業工程は単純作業となり新興国にも輸出可能なモデルとなる。このモデルが構造化されると、下流工程の作業員は、常に賃金低下の圧力に晒されることになり、時間により測定される単純労働としての価値しか見出せなくなる。

上流と下流の開発業務を分離することで、設計段階は上流の親会社が占有し、下請け会社は専ら指示に従う作業となり、顧客との情報交換は親企業が独占することとなる。下請けである下流工程には顧客情報が伝わらなくなり、顧客の問題やニーズを認識することなく作業を行うことになる。平井氏は、この構造的な関係に着目し、下流工程の作業者の知的労働を阻害する **Waterfall Model** の限界を指摘する。

Agile 型開発組織は、問題発見の機会を有する下流工程の情報が設計作業に生かされることで、開発プロセスは価値創造プロセスとなる。それは、設計とプログラム作成が結合することにより実現する価値であり、これらの諸問題が事例研究を加味して論証されている。

(2) 論文の評価

日本のソフトウェア開発の研究は工学系に偏っており、開発プロセスに関する研究は少ない。経営学の視点より分業構造を分析し、労働のあり方を考察した研究は稀有である。特に、開発プロセスの構造化により、設計段階と作業段階に企業の境界が生じ、開発プロセスの下請け構造が、顧客との情報交換を阻害し、顧客のニーズを生かせない開発につながっていることを論じている点は説得力があり、高く評価できる。実際の経験と先行研究に基づく緻密な論理構成は、博士学位論文として十分な価値がある。

今後の課題としては、Agile 型開発組織の事例、とりわけ、欧米で成功している開発プロセスを検証し、成熟した経済における開発プロセスの一般モデルを構築することとなる。