

## 先進製造業の生産マネジメント論から見た 事務用情報システム開発

戸田 忠良

(株)戸田ソフトウェアオフィス

[E-mail: fwhe4979@mb.infoweb.ne.jp](mailto:fwhe4979@mb.infoweb.ne.jp)

要約：情報システムにおける従来の開発法は「万能型 SE 一貫生産方式」であり、技術者の個人能力に依存しており、組織能力構築による生産性向上が困難なやり方である。そのため、生産革新を求め、自動車産業などの先進製造業に学ぶ動きが起り始めている。しかし、製造業と情報システム開発には違いがあるので、製造業における取り組みを、システム開発の特徴に応じて読み換える必要がある。本稿では、「製品開発期間の短縮」に関する汎用的ロジックを参考にし、現実のシステム開発でどの程度、そこで取り上げられた時間制御策が行われているかについて分析を行う。分析の結果、システム開発において上述の方法の多くは既に適用済みであるが、「工程オーバーラッピング」と「工程内分業」による期間短縮策は未適用であることが明らかとなった。これらの組織能力構築がテーマとなるアプローチが今後の生産革新の大きなヒントになると考える。

キーワード：情報システム開発ビジネス、万能型 SE 一貫生産方式、生産マネジメント論、工程オーバーラッピング、工程内分業、組織能力構築

### 1. はじめに

事務用情報システム開発ビジネスは、技術者単価の長期低落傾向や「3K 職種」という評

<sup>1</sup> 本稿は 2007 年 6 月 28 日開催のコンピュータ産業研究会での報告を福澤光啓（東京大学大学院）が記録し、本稿掲載のために報告者の加筆訂正を経て、GBRC 編集部が整理したものである。文責は GBRC に、著作権は報告者にある。

価の定着による就職不人気などから、慢性的な技術者不足を来し、中国、インドなどのオフショア開発を大規模に採用し始めるなど、大きな転換期を迎えている。従来の情報システムの開発は、「万能型 SE 一貫生産方式」による開発体制で行われており、技術者の個人能力依存が強く、組織能力構築による生産性向上が困難である。そのため、生産革新を求めて、自動車産業などの先進製造業に学ぶ動きが起り始めている。しかし、先進ノウハウの適用は簡単ではない。例えば、自動車組立工程は、情報システムではシステム運用に相当し、システム開発自体は、通常の製造業では、「新製品開発」に相当するからだ。そのため、情報システム開発の特徴に応じた読み換えが必要となる。そこで、どの産業にも共通な基本的なロジックを中核にした「生産マネジメント論」に学ぶべきと考える。本稿では、藤本（2001, 14章）で示された「製品開発期間の短縮」に関する汎用的ロジックを参考にして、現実のシステム開発においてどの程度、そこで取り上げられた時間制御策が実施されているかについて分析を行っていききたい。

## 2. システム開発の現状と構造

### (1) 最近のシステム開発を巡る動向

事務用情報システム開発は、相変わらず困難なプロジェクトであり、その生産性に目覚ましい向上は起っていない。日本におけるソフト開発の品質は、欧米、インドなどに比べ高いが、受託開発に偏重し、利益率の高い「基本ソフトのパッケージ販売」などは米国勢の独占を許している。

「SE 不足」にも関わらず、「SE 単価の上昇はごく僅か」であるという日本の現状は、IT 業界が「価格決定力」を喪失している証拠である。その原因として、技術革新の低迷（生産性向上によるコストダウンが実現できない）や、低コスト手段（オフショア開発やパッケージ・ソフト）との競合の激化、経営的效果が計算できる開発対象需要が一巡したことなどが挙げられる。特に問題なのは、製造業の常道である「生産性向上によるコストダウン」は困難であり、「技術者単価の削減によるコストダウン」以外の展望が見えない点である。

本質的・長期的には、「コンピュータや通信という商品」が消滅し、「（ネット経由）サービス」<sup>2</sup> に移動しつつあることが原因で、現在の手先 SI (System Integration) 企業を頂点

<sup>2</sup> 必要な機能やサービスにネットワーク経由でユーザーが自在にアクセスできる。例えば、Google

とする体制の転換を意味する。これがWeb 2.0のIT業界における本当のインパクトである。

上記の低迷を脱出する方法として、中国やインドのオフショア開発によるコストダウンを図る動きが本格化すると同時に、TPS（トヨタ生産システム）など先進製造業のノウハウの情報システム開発への適用を探る動きが出てきた。

## (2) 従来のシステム開発の特徴と問題点

従来のシステム開発の特徴として、システム要件主導（実現する要件を決めることから開発がスタート）と機能要件からのトップダウン開発が挙げられる。そして、これを実行する体制として「万能型SE一貫生産方式」による開発体制が行われてきた。この方法では、システム分割後の適当な大きさの作成単位を一人の担当者が設計または製造（あるいは双方）を行う仕組みで、この単位以下での分業は行われない、この方法は担当範囲内での要員間調整コストが一人で作業するため不要になるメリットがあり定着してきた、万能型SE（技術知識全般と業務知識の双方を兼備）が必須である。従って、システム開発の価格の決定は、万能型SEを何人、何ヶ月動員するかで決まる、いわゆる「人月方式」で行われてきた。しかし、この開発体制は、生産性向上の道に限られるなど問題が多い。すなわち、先輩SEによるバックアップ能力が劣化すると仕様の詳細を担当者以外が把握できず、トラブルが頻発する危険性が大きい、個人能力が向上する以外に生産性向上をすることが困難、つまり、先進製造業の得意とする組織能力構築（例えば、更なる分業化による作業の並行化で時間短縮を実現）による生産性向上が困難なことである。

システム開発の複雑さ・困難さの原因は、利用者による入出力要求や対象業務の業務ロジック（業務モデルや業務ルール）それらを実現する実装プラットフォームへの適応といった多様な要素を同時的に扱わねばならない点にある。これらの混合物である情報システムを開発するためには、本来、これらの変動要素を分離して変化を吸収する仕組みが必要であるが、それらの分離・独立化が実現していないため、現時点ではこれ以上の工程内分業の推進が困難で、すべてを理解できる万能型SEが不可欠なのである。

万能型SE一貫生産方式では、最初に共通インタフェース設計（データベース設計を含む）が行われ、続いて各機能単位の設計・製造が並行して行われて（各機能の開発については万能型SEがそれぞれ一人で担当）、最後に、結合・統合テストで接続を確認するとい

---

は、日本で組む相手は不要。それは、直接ホームページ上で検索などの機能を提供しており、ソフト製品を売らないので販売代理店も不要ということ。従来のソフトウェア・ビジネスの枠を超えた存在が登場している。

う順で行われる。この生産方式では、全ての担当者が計画通り作業を完了することが必要で、「プロジェクト完全成功確率 = 担当者全員が同時に成功する確率」ということになる。例えば、開発者がそれぞれ 90% の完全成功確率を持っているとして、そのような担当者 5 人で開発を行うとすると、当該プロジェクトの完全成功確率は、 $(0.9)^5 = 0.590$  となり、さらに 10 人で行う場合の成功確率は、 $(0.9)^{10} = 0.34$  となるので、たとえ一人あたりの成功確率が高い（つまり、優秀な SE を集めた）としても、プロジェクト全体の完全成功確率は要員が増えるほど低くなる。これが、システム開発プロジェクトを困難なものにしている。

また、ウォーターフォール型プロセスによる万能型 SE 一貫生産方式では、基本設計 詳細設計 製造というように、下流工程に行くほど技術者が増加する。ここでは、各工程の最も遅い担当作業が結果的にクリティカルパスとなってしまうので、開発期間の短縮は最も遅い担当者の時間短縮が必要となり容易ではない。万能型 SE 一貫生産方式の利害得失をまとめると表 1 のようになる。

システム開発のコスト管理の焦点は、「要員間調整コスト」( = 「コミュニケーション・コスト」、あるいは「すり合わせコスト」) をいかにして制御するかである。要員間調整コスト = (1 箇所当りの調整コスト) × (調整必要箇所数) と表される。採るべき対策は、

表 1 万能型 SE 一貫生産方式の利害得失

利点・前提条件	問題点・課題
<b>要員間調整を不要化</b> ：要員間の情報共有の必要性を限定したので、要員間調整コストが低く抑えられる。	<b>担当者への過大な負荷</b> ：詳細な仕様を担当者以外知らないで、万が一のカバーが困難→「指導役 SE」がうまく機能すれば、バックアップとしてカバーも可能。
<b>全体管理はインタフェース部に集中</b> ：プロジェクト管理の焦点は、インタフェース部分に限定。細部は担当者に任せる	<b>標準化や部品化の進展を阻害</b> ：常に、ゼロスタートの開発となり、いつまでも経験の蓄積が進まない。
<b>「高自由度・低制限の開発」に適合</b> ：自由度の高い、制限の少ない開発に極めて有効な対応策となる。	<b>個人依存の生産性向上</b> ：個人の能力向上以外に生産性向上の手段がなく、組織力が生産性向上に反映しない。
<b>万能型 SE が必要</b> ：担当者全員が、必要な作業は全部出来る万能型 SE である必要あり。	<b>技術の専門化に逆行</b> ：技術の複雑化により、どの役割もこなせる万能型 SE を養成することが困難になりつつある。

この二つの要素を共に減少させることで、作成プログラム内非分業（つまり「万能型SE一貫生産方式」により内部調整を不要とし調整必要箇所を削減）や、「モジュール化」（呼び出し方法の固定による調整必要箇所の削減）、グループ化（疎結合のグループ分割による調整必要箇所の削減）の三つが調整必要箇所数を減少させる手法であり、1箇所当たりの調整コストを低減する方法としては、経験を重ねてチームとして熟練し、少ない調整コストで済むようにすることが挙げられる。

### 3. 製造業のノウハウを導入する道筋

製造業のノウハウをシステム開発に導入するための道筋として、システム開発における「見える化」の意味の違いの理解、<sup>3</sup> 製造業と情報システム開発の比較、<sup>4</sup> システム開発とは「情報生産工場」を作ることであるという理解（情報システムは、「情報生産工場」であり、情報システム開発はその工場を作ることに対応する）、生産革新のテーマとノウハウの学び方<sup>5</sup> を取り上げて、それぞれについて考察する。

#### (1) 情報システム開発における「問題の見える化」の意味

トヨタ生産方式に代表される製造業の基本は「問題の見える化」である。現在は、システム開発業界でもこの言葉が強調され始めたが、これまでは十分に実現できていないのが現状である。製造業の組織的問題解決の考え方と従来の情報システム開発の考え方との違いを示すと表2のようになる。どの産業であれ、問題の発見が早いほど、問題解決コストは安くなる。組織的な行動として、早期に問題やリスクの存在を提起し、解決に導くことが企業の利益になるということを徹底的に理解することが、「問題の見える化」を推進する上で絶対に必要な企業体質の転換ポイントである。

<sup>3</sup> 情報システム開発では、SE個人の「属人的問題解決」が中心であり、「問題の見える化」が本当に効果を発揮する「組織的問題解決」が行われていない。

<sup>4</sup> 情報システム開発は通常の製造業の「製品開発・生産準備」が該当。実際の製品の組立は、「システムの運用」が相当する。

<sup>5</sup> 「短納期要求」がユーザーの根源的要求であり、これに挑戦するのがソフトウェア生産革新の最大のテーマである。また、汎用ロジックの抽出と別業種への演繹的適用こそが、ノウハウの利用の有効なやり方である。直接的な製造業ノウハウの適用は混乱を招く懸念がある。

表2 従来の情報システム開発と製造業的組織的問題解決との違い

従来のIT産業流思考法	項目	製造業的組織的問題解決の考え方
担当者全員が自分の担当分野を成功させれば、仕事は成功する。従って、 <b>頑張る担当者が必要</b> 。成功は担当者の能力次第。 →「 <b>属人的問題解決</b> 」	基本的考え方	問題が明確になれば、組織的な知識の集積を使って問題は解ける。そこで、早期に <b>問題を組織的に共有することが大切</b> 。それがトヨタの「問題の見える化」 →「 <b>組織的問題解決</b> 」
特に顧客の業務内容の詳細は担当者しか知らないことが多く、その <b>担当者が解決するしか実際には方法が無い</b> 。業務はそれぞれ違うので問題解決のための知識集積は困難。 「 <b>問題を個人が抱え込む体質</b> 」	知識の集積と問題の解決法 組織体質	部品が多く、問題の解決が多くのメンバーの協業で成り立つので、一人だけの力では問題解決にならない。そこで、 <b>多くのメンバーの知恵を集めることが不可欠</b> になる。 「 <b>問題を早期に共有することの奨励</b> 」

表3 製造業と情報システム開発の違い

	製造業	情報システム開発
生産	製品を製造することが生産プロセス	情報システムの生産するものは、画面の出力結果や帳票。生産プロセスは、情報システムの稼動・運用を意味する。
開発	新製品の企画・設計のことで、実際の生産は工場で行われ、製品が生まれる	最終的に欲しい帳票などを作成するプログラムの設計・製造。これは工場のラインに並ぶ工作機械の設計・製造に相当
部品表 (BOM: Bill Of Materials)	製品を構成する部品の構造を示す	データ・アーキテクチャ(データモデル): 出力結果のデータの構成部品とその関連を示す。
工作機械	各々の工作機械の部品構造	プログラム(ソフトウェア)・アーキテクチャ: プログラムの部品構造。部品化というとプログラムの部品化を一般に意味する
製造ライン	製品を製造するライン構造	システム・アーキテクチャ: データ処理の構造で、いわば情報処理工場の全体構造に該当。
製品開発	新製品開発のプロセス	工程アーキテクチャ: システム開発のプロセス。ウォーター・フォール型、イテレーション(Iteration: 反復)型などがある

(2) 製造業と情報システム開発の違い

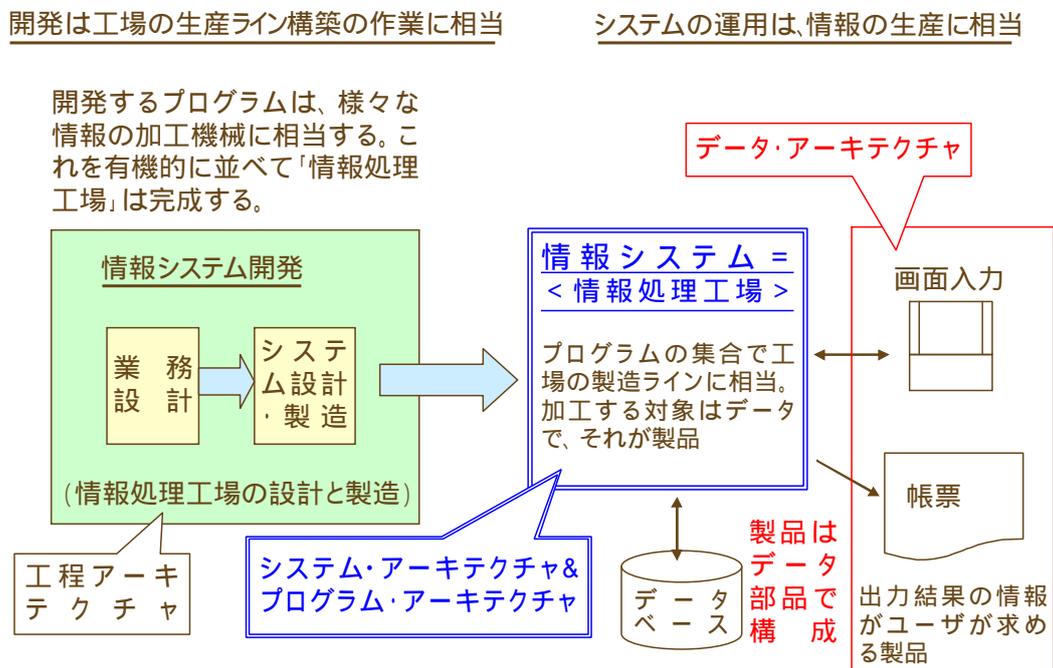
製造業と情報システム開発の違いを表したものが、表3である。例えば、製造業では、製品を製造することが生産プロセスであるのに対して、情報システム開発では、情報システムの生産するものは、画面の出力結果や帳票であり、生産プロセスとは情報システムの稼働・運用を意味する。

(3) 情報システム開発 = 「情報生産工場」を作ること

情報システム開発において必要な設計対象として、工程アーキテクチャ（開発プロセス自体の設計、一般には、標準的工程モデルを採用し、その都度設計はしない）、システム・アーキテクチャ（対象システムの構造設計）、プログラム・アーキテクチャ（各プログラムの構造設計）、データ・アーキテクチャ（データモデル設計）が挙げられる。

「情報生産工場」の特徴として、プログラムは工作機械に相当し、情報を加工するのが役目、データモデルは部品表<sup>6</sup>に相当、オンライン即時処理は「1個流し」生産

図1 製造業との比較で見る情報システム



<sup>6</sup> 情報製品（帳票・画面）の論理的部品構造を示す。しかし、一般には、部品はプログラム部品のことを指すことが多い。これは工作機械の部品に相当し、生産対象の製品を構成する部品ではない。

に相当、情報システムの開発工程は新製品開発工程であり、期間短縮が重要、情報生産工場の生産性<sup>7</sup>として処理能力・処理性能が重要、が挙げられる。以上を踏まえて、情報システムと製造業とを比較すると、図1のようになる。

#### 4. 「生産マネジメント論」に学ぶ製品開発期間短縮の手順

藤本(2001)で述べられている製造業における製品開発期間短縮の方法が、情報システム開発へどのような形でどの程度適用可能なのかということについて考察する。問題工程の検討論理とその手順としては、まず、どの工程に問題があるのかということをはっきりさせるために、PERT(Program Evaluation and Review Technique)を用いてクリティカルパスを発見し、その後、問題の工程について期間短縮を具体的にいう流れとなる。後者において採るべき方策は、個々の活動の短縮化やタスク分割とオーバーラップ型問題解決、フロントローディング型問題解決の三つである。ここで示されている汎用のマネジメント手法は、システム開発のプロジェクト管理の定石にも相通じるものである。

##### (1) 問題工程の発見

PERTあるいは、CPM(Critical Path Method)は、問題工程を発見する技法である。これらは、ネットワーク状に連結する工程のうち、最も時間のかかる経路を見つけ出す技法で、クリティカルパス上のプロセス以外の時間短縮は全体の時間短縮に全く貢献しないので、クリティカルパスの発見と当該プロセスの時間短縮が取り組むべきテーマとなる。

万能型SE一貫生産方式では、クリティカルパスは事前には判明することは不可能であり、動的に制御されるべき問題である。プロジェクトの進捗管理では、各工程の最も遅い担当作業が結果的にクリティカルパスとなるので、プロジェクトマネージャーは、最も遅くなりそうなパス(担当者)を発見して、遅延防止の対策を打つことが役割となる。

##### (2) 問題工程の期間短縮策

問題工程の遅延防止のための期間短縮策として、個々の活動の短縮化、タスク分

<sup>7</sup> 情報生産工場における情報生産方法として、バッチ(Batch:一括)処理法とオンライン即時(Online Real-time)処理法の二つがある。前者では、生産されたものは「情動的在庫」になり、後者では、生産物の製品在庫や仕掛かり在庫を持たない。また、前者では処理能力が重視されるが、後者では、応答時間(response time)すなわち処理性能が重視される。

割と工程オーバーラップ型問題解決、フロントローディング型問題解決の適用が挙げられる。

### 個々の活動を短縮するための方策

個々の活動を短縮するための方策として、人員の追加投入、<sup>8</sup> 製品開発のコンピュータ化、<sup>9</sup> 「開発に埋め込まれた製造活動」の短縮、<sup>10</sup> 「情動的在庫」の低減<sup>11</sup> がある。

### タスク分割と工程オーバーラップ型問題解決

あるサブシステム A を A1 と A2 の二つの互いに独立なタスクに分割可能であれば、別々に担当者を割り当て設計を進めることができる。そうすると、A1 と A2 の二つで並行作業が可能になり、開発期間全体が短縮される。例えば、「照会機能」の開発では、「受注内容照会」と「製品在庫照会」とにタスク分割することが可能であるとすると、これらのタスクを別々の担当者に割り当てることで、並行作業が可能になり、期間短縮が可能となる。しかし、これが可能なケースは、「二つのタスクが相互に独立」という条件が満たされる場合に限定される。この制限を越えて、作業の並行化による時間短縮を行うには、工程内分業の更なる進展が必要で、分業間の調整をスムーズに行う組織能力の構築が求められる。

また、オーバーラップ型問題解決に関しては、従来型の工程直列ルールがあり、要件定義 概要設計 詳細設計 製造・試験という流れにおいて、前工程が完全に終わるまで、次工程の作業は待たされており、実施されていない。一方、このような工程直列の制限を解除して工程のオーバーラッピングを行うとすると、例えば、概要設計工程での詳細な機能仕様の決定と並行して、詳細設計をオーバーラップさせることが出来れば、その分だけ期間短縮が可能になる。この時、概要設計担当者と詳細設計担当者間に最終結論ではない仕様を動的に調整できる組織能力構築が必要となる。自動車産業では、そのような前後

<sup>8</sup> 情報システム開発においては一般的な解決策。しかし、担当部分の知識が欠ける人員を追加投入しても容易には時間短縮に貢献しないことに注意が必要である。

<sup>9</sup> 自動車業界の例では、CAD (Computer Aided Design)、CAM (Computer Aided Manufacturing) あるいは CAE (Computer Aided Engineering) などの利用に相当する。システム開発では既に多くの支援ツールを使用中である。

<sup>10</sup> 自動車の開発では、「試作車の製作」という製造活動が該当する。システム開発では、テスト開発、試行開発と呼ばれる活動に当る。これは、新技術などを事前に試験する目的で行われる。

<sup>11</sup> 処理待ちの設計文書の待ち行列が、「情動的在庫」である。自動車の場合は、情動的在庫とは CAD 入力待ちの設計情報であったり、試作工程の前の設計図面などの製品設計情報ストックである。情報システム開発では、工程のオーバーラップが禁止されているので、次工程の前に情動的在庫が膨らんでいることが問題となる。

の工程のオーバーラッピングにより、製品開発の期間短縮が実現している。

### フロントローディングの情報システム開発への適用

「フロントローディング型問題解決」とは、後半の期間に特に大きな時間を要することになる設計変更の回数を減らすために、より多くの問題解決を初期の部分に集中的に行うことである。どの製造業でも、後になるほど仕様変更コストは大きくなる傾向にあるので、この手法は効果的である。IT業界においても、SI開発の成功の条件として、知識のフロントローディング<sup>12</sup>と、問題解決のフロントローディング<sup>13</sup>に相当する早期の問題解決の実施が業界のプロジェクト成功のための共通認識となっている。

## 5. まとめ

システム要件決定からシステムの稼働・運用開始までにかかる時間を短くすることが、根本的な顧客満足の実現であると同時に、開発コストの削減に直結する。しかし、無理に短縮を行うと、テスト工程が時間不足になるなど、品質劣化を引き起こす。そこで、論理的・合理的な開発期間短縮策が求められる。汎用の開発期間短縮策の情報システム開発への適用状況の分析から、開発期間短縮活動の阻害要因として、工程直列ルール(各工程に最低所要時間があり、全体の短縮が困難)と万能型SE一貫生産方式(この方式では更なる分業化・作業の並列化による時間短縮が不可能)を挙げることができる。

藤本(2001)における「製品開発期間の短縮」に関する汎用的ロジックを参考にし、現実の情報システム開発への期間短縮策の適用状況についてまとめると表4のようになる。そこから言えることは、システム開発において上述の方法の多くは優れたプロジェクト管理では既に適用済みであり、「工程オーバーラップ型問題解決」と「工程内分業(つまり、万能型SE一貫生産方式の見直し)による期間短縮策」については、未だ実施されていないということである。他製造業では効果のあったこれらの組織能力構築の必要なアプローチが、今後の情報システム開発の生産革新の大きなヒントになると考えられる。

<sup>12</sup> 設計工程の初期に問題を徹底的に見つけ出し解決してから設計の作業に入るのが、成功するプロジェクト運営のノウハウである。PMO(Project Management Office)は過去の問題解決の知識のフロントローディングを推進する組織として解釈できる。

<sup>13</sup> 利用者との要件のすり合わせを徹底的に行い、双方が同じイメージを共有するまでに議論を深めることが成功の鍵を握る。

表4 情報システム開発の期間短縮策の実施状況

汎用対応策		システム開発の実施
個々の活動の短縮	人員の追加投入	<b>実施済み</b> : IT 業界がこのような問題に対処する基本的な手段。効果を生む投入法の工夫がより効果的な結果に繋がる。
	製品開発のコンピュータ化	<b>実施済み</b> : 統合型開発支援環境の利用で、テストや自動生成の機能などが充実。(製造業におけるCAD/CAMの利用)
	「開発に埋め込まれた製造活動」の短縮	<b>実施済み</b> : 試行開発、プロトタイプ開発の終了を的確に制御。新技術採用時などに大きな管理テーマになる。
	「情動的在庫」の低減	<b>未実施</b> : 次工程の前の設計書の待ち行列は工程直列ルールで次工程開始まで作業は進められない。
タスク分割とオーバーラップ型問題解決	タスク分割	<b>実施済み</b> : 互いに独立なタスクの分割が可能な場合に限定。広範な作業並行化による時間短縮には、工程内分業が必要。(万能型 SE 一貫生産方式の見直し)
	オーバーラップ型問題解決	<b>未実施</b> : 工程のオーバーラッピング禁止のルールを見直しし、一部のプロジェクトでは効果が見込める工程間すり合わせを模索すべき。今後の大きな検討課題。
フロントローディング型問題解決	知識のフロントローディング	<b>実施済み</b> : 過去の経験からリスク項目を抽出し、初期に集中的に対策を打つ。リスク管理活動やPMOがこれに相当する。
	問題解決のフロントローディング	<b>実施済み</b> : プロトタイプ開発などにより、初期に問題を発見・解決する施策を行い、問題解決の時期を初期に持ってくるようにする。

参考文献

藤本隆宏 (2001) 『生産マネジメント入門 2 生産資源・技術管理編』日本経済新聞社.



**赤門マネジメント・レビュー編集委員会**

編集長 新宅 純二郎

編集委員 阿部 誠 粕谷 誠 高橋 伸夫 藤本 隆宏

編集担当 西田 麻希

**赤門マネジメント・レビュー 6巻8号** 2007年8月25日発行

編集 東京大学大学院経済学研究科 ABAS/AMR 編集委員会

発行 特定非営利活動法人グローバルビジネスリサーチセンター

理事長 高橋 伸夫

東京都千代田区丸の内

<http://www.gbrc.jp>