

企業の内部・外部環境に基づく情報システム開発方針の選択

河村 竜幸

Tatsuyuki Kawamura

1. はじめに

本稿では、企業の内部・外部環境に基づいた情報システム開発方針の選択に注目する。情報システムの開発において、そのシステムの開発環境や人員、言語等の実現手段を無数に検討することが可能である。システム開発方針の選択時にあらゆる制約にとらわれない場合、人間の指向性選択が保守的であるとするならば、意思決定者の成功体験に基づいたものとなりえる。つまり、システム開発責任者は自身のこれまでのシステム開発経験に基づいた開発方針を選択するものと考えられる。しかしながら実際は、事業のターゲット業界や市場成長余地だけでなく企業の規模、人材の技能、既開発済みシステムや開発ノウハウ等の諸条件によって、企業にとって少しでも利益率や売上高が大きくなるシステム開発方針はある程度限定される。例えば新規市場への参入を検討する時、システム開発責任者の成功体験に基づいた開発方針は最適な選択となるとは限らない。そこで企業の内部及び外部環境の違いによりシステムの開発方針がどのように選択されているのかを実例に基づいて比較分析することは、中小企業、大企業に関わらずある一定の示唆を得られるという点において有効であると考えられる。

以降では、最初に企業規模、事業内容、製品構成が異なる3つのケースについて取り上げる。各々のケースでは簡単な企業内構成、本研究で対象とするシステムの基本構成、製品構成のパターン、開発チーム構成を紹介した後、当該ケースで採用された開発方針を説明する。その後3ケースを総合的に分析し、内部環境・外部環境の違いによるシステム開発方針の選択肢について考察する。

2. ケース1：少人数の開発環境、少品種の情報システム

ケース1では、非ソフトウェアメーカー小企業における少人数かつ技能者不在状況下での情報システムの開発方針について論じる。また本ケースでは、同一目的の製品構成が3パターンと少品種の製品開発を対象としている。

2.1 システムの基本構成

対象製品は、主にエンジン内の燃焼現象を観測することを目的としていた。本製品は本ケース企業が独自に開発した特殊な光学センサを用いて観測される時系列データを収集するものであった。また収集されたデータをリアルタイムで解析し、解析結果をモニタ上で表示することが可能であった。計測されたデータはPCメモリ上に一定時間蓄積することが可能であり、さらに蓄積されたデータを保存し、オフラインデータとして後から読込・再解析したりすることが可能であった。

製品構成のパターンは計測する点数の違いによるものであった。最初の製品は2点の同時計測を可能とするものであり、次の製品は4点の同時計測を可能とするものであった。最新の製品は小型化対応品でありハードにてデー

タをロギングし、オフラインにてデータを解析する機能を有するものであった。

2.2 製品構成パターン

本製品は計測システムであることから、必要となるデータ収集を実施できることが重要であった。また本製品の差別化ポイントは本ケース企業開発の光学センサにあり、ソフトウェアは製品において従の存在であった。これらのことから、ソフトウェアに対して顧客から要望を出されることは無かった。むしろ機能要望は社内研究員や営業員との打合せによって提案されることがしばしばであった。

2.3 開発チーム構成

本ケースでは、主たる開発者（プロジェクトリーダー）が1名、開発補助者が1名の2名体制であった。開発補助者のプログラミング歴は2～3年程度と初心者に近いものであった。そのため、社内での仕様決めや機能設計、ソフト品質保証等、ほとんどの業務についてプロジェクトリーダーが責任者となっていた。プロジェクトリーダーはC/C++やC#等の経験があり本ケースの作業に支障は無かった。一方、開発補助者の経験はC/C++のみであったため、プログラミング歴の浅さに追加してC#に関する教育も必要であった。

2.4 採用した開発方針

本ケース製品の特徴からソフトウェア開発を顧客要望に応じてカスタマイズされることはなかった。むしろ製品の品質向上と機能強化の観点からソフトウェアを修正する時間が大半であった。製品のソフトウェア修正に関しては、プロジェクトリーダーが実施した。開発補助者は、ソフトウェア修正に関連した付属ツール等、周辺ソフトウェアの開発を分担していた。周辺ソフトウェアであっても新規に開発する場合は、最初にプロジェクトリーダーが雛型を作成し、開発補助者に雛型を提供した。開発補助者は雛型の設計方針に関するレクチャを聞き、残る未実装機能について作業を進めていった。

ソフトウェア開発の効率を上げるため、両者の机は横に並ぶ配置となっていた。お互いにモニタを参照しあえる距離にすることで、開発補助者の作業進捗を即時に確認することができた。また開発補助者は、わからないことがあるとすぐプロジェクトリーダーに質問することができた。質問内容によっては、プロジェクトリーダーは自身のモニタ上で質問内容の回答となるコーディングをその場でしてみせた。いわば、業務時間全体でOJTを実施している状況と考えることが可能であった。

3. ケース2：少企業、高カスタマイズの情報システム

ケース2では、非ソフトウェアメーカー小企業における中人数状況下での情報システムの開発方針について論じる。本ケースで対象とする製品は、特定目的の1品種製品でありながら顧客の要望により機能の追加・修正を実施する高カスタマイズ品である。

3.1 システムの基本構成

対象システムは全て、医療施設で使用されるPACS（Picture Archiving and Communication System）に関するものであった。さらにここではDICOM（Digital Imaging and Communication in Medicine）と呼ばれる医用画像の画像規格及び通信プロトコルに関する統一規格に準拠することが重要であった。全製品はDICOM規格に準拠し

ていた。各製品は DICOM サーバシステム、画像管理システム、マンモグラフィ用画像診断システム、マルチモダリティ医療用画像表示システム、放射線レポートシステム、循環器系動画表示システム、心臓カテーテル検査情報管理システム、DICOM 画像 Web 配信サーバシステム、CD/DVD 自動作成システム等、多岐に渡っていた。これらがネットワーク上や同一 PC 内で連携することで高付加価値のサービスを顧客に提供していた。当時、既に全システムは上市済みであった。

3.2 製品構成パターン

本ケースにおける製品構成パターン上の特徴は、①複数のシステムがネットワーク上に個々で存在すること、②システム間連携の部分が重要であること、③顧客要望や導入環境によってシステムをカスタマイズする必要性が発生すること、の3点にある。一部の高付加価値製品を除いて、ほとんどの製品は機能の多少などによる製品シリーズ化はされていなかった。各製品は必須機能と製品差別化のための特殊機能により実現されていた。そのため、当該製品に興味を示す顧客から製品機能を減らす方向での話はなかった。むしろ各顧客（各病院）の PACS システムの導入状況や、システムを利用する医療従事者等によって製品機能の一部変更や機能追加の要望が出されることがたびたび発生した。要望は導入前・後に関わらず出されることがあった。機能の追加は有償で実施していた。追加された機能が一般的に受け入れられると考えられるものは、製品のマイナーチェンジとしてそのまま公開された。

3.3 開発チーム構成

本ケースでは、システム開発部員として7名が所属していた。各部員のプログラミング歴は10年以上と全員の経験は豊富であった。そのため、社内業務は各人で責任をもって分担することが可能であった。社内での使用言語はC/C++とC#が主であり、ウェブ利用のシステムではPHPを利用することもあった。これら言語の差はシステム機能によって使い分けられていた。C/C++は主に画像の表示高速性が必要なビューア系システムで採用されていた。一方、C#はデータ検索やレポート作成等、ユーザが頻繁に利用するGUI上の高ユーザビリティ性が必要なシステムで採用されていた。教育面に関しては、各人が主体的に自主学習を実施していたため、社内教育として特に実施されることはなかった。

3.4 採用した開発方針

全員の能力水準が一定であったことから、情報ネットワーク内の個々のシステムを各人が分担していた。開発対象となるシステムは全て上市済みであるため、システム内の機能強化、システム間の連携強化という顧客カスタマイズ作業がほとんどであった。システム間連携機能の開発では、関係システムの担当者同士ですり合わせ的に設計を実施していた。システム連携の大半は外部モジュールの開発によって実現させ、既実装部分への影響が最小となるように調整していた。

座席配置は、システム機能やシステム間連携を考慮したものとなっていた。本ケースではシステム間の連携が非常に重要であったことから、社内で連携動作に関するテストを実施する際、座席移動がほぼ無い配置となるよう考慮した。

4. ケース 3：中人数の開発環境、多品種の情報システム

ケース 3 では、非ソフトウェアメーカ大企業における中人数（5～10 名）かつ上級から初級までの技能者が幅広く所属する状況下での情報システムの開発方針について論じる。また本ケースでは、同一目的でありながら、客先での対象品ごとに個別設計が必要な多品種少量生産の製品構成を対象としている。

4.1 システムの基本構成

本ケースでは対象システムを、製造工程における検査装置に限定する。この検査装置は製造部品（以降、ワークと呼ぶ）の良品 / 不用品を自動的に検査し判定する機能を有する。検査装置が他の装置と独立した運用、いわゆるスタンドアローン型である場合、装置機能は①ワークの供給部、②ワークの排出部、③検査部の 3 箇所の配置に分類することができる。本ケース企業ではシーケンス制御による製造装置の製作に関するノウハウは豊富であり、装置制御を実現するためにシーケンス制御専用の PLC（Programmable Logic Controller）を①と②の実装において採用している。一方③の検査部では、画像撮影に付帯する各種デバイス制御や画像処理等が複雑にからみ、また非常に高速な CPU、大規模メモリ、大容量ストレージを必要とする。そのため、検査部を他部と独立させて PC 上でシステムを実装することがシステム構成上望ましい。

4.2 製品構成パターン

最大の特徴は顧客ごとに製造するワークが異なる部分にある。顧客が検査する対象ワークそのもの、ワークのサイズ・形状、検出したい欠陥内容、1 ワークの検査にかかる時間（以降、タクトと呼ぶ）等の仕様が大きく影響する。この仕様の違いによって装置のサイズ、ワークの搬送方法、検査手順等が大きく変化する。よって本ケースではハード面だけでなく、ソフト面に関する部分もすり合わせによって設計・製作されることが多い。

4.3 開発チーム構成

本ケースでは、システム開発グループとして 9 人が所属・関係していた。内 7 名が専任であり、2 名は業務負荷の状況により応援という立場であった。2 名（グループ A）は、ソフトウェア設計・プログラミング能力に関して一定水準以上を超え十分な域に達していた。2 名（グループ B）は装置製作に関する経験は 10 年以上と一定水準以上を超え十分な域に達していたが、単独で一定規模のソフトウェアを設計・製作する能力を有してはいなかった。1 名（グループ C）は装置製作に関する経験は無く、また単独で一定規模のソフトウェアを設計・製作する能力を有してはいなかった。2 名（グループ D）はソフトウェア設計・プログラミング経験は 10 年以上と豊富であったが、単独でソフトウェアを設計・製作することが可能なレベルに達していなかった。残る 2 名（グループ E）の内、1 名は経験年数が 3 年程度であり、残る 1 名は中途入社直後という状況であった。社内での使用言語は主に C# であった。一部処理の高速性が必要なモジュールについては C/C++ で実装する部分が存在した。教育面ではグループ D とグループ E に対して、個々の業務能力の違いを勘案し実施していた。教育内容はソフトウェア開発に限らず装置の知識増強も含め、広く浅いものであった。その他の特記事項として、本ケースに関わる事業は、直近で大きく拡大してゆく予想・計画が立っているため、現状組織への拡大は短期間に実施された。当初は 2 名で開発されていた規模のものであった。

4.4 採用された開発方針

本ケースでは、製造装置に関する広い経験・知識が必要である一方で、情報システムに関する深い経験・知識が必要であった。しかしながら、両方を同時に満たす人材は存在しなかった。個々の知識・経験獲得に10年以上を必要とすると、最新技術に関する領域も含めて網羅的に全体を把握している人材を育成することは非常に困難であることは明確である。そこで本ケースでは、グループAとグループBから個々に1名をリーダーとして選出し、マトリックス型のシステム開発を推進した。グループAリーダーは、ソフトウェア要素技術開発を責任範囲とした。またグループBリーダーは、装置全体設計を責任範囲とした。

本ケースの対象企業は他ケース企業と異なり、企業文化として標準化を重視していた。装置内部のPLCソフト設計に関しては、各種設計図面等の記述ルールだけでなく、ソフトコーディングルールや、品質確認のためのチェックシート等が充実し、全図面に関して実作者と評価者によるダブル・チェックが実施されていた。PCソフト設計ではこれまでPLCソフト設計と同程度の体制で実施されていなかった。しかしながら品質向上を実現するために、PCソフト設計でも各種ルールの拡充が推進されている。

本ケースでは事業の拡大による検査装置の受注増が事前に予想・計画されているため、その受注増への対応を人員増強以外の面でも考慮する必要があった。そこで、装置検査部のシステム開発に関しては次の方式を採用することとした。①案件間で共通性が非常に高い機能に関しては共通機能としての開発を別途実施する。②複数人での並行開発を実現するためにシステム内の個別機能をモジュール化する。③案件毎にすり合わせが必要な部分については、ソフト設計の柔軟性を残しておく。

5. ケース間の比較分析

ここでは3つのケースについて総合的に分析し、企業の内部環境・外部環境の違いによる開発方針の選択肢について考察してゆく。3ケースのデータを一覧にしたものを表1に示す。ここでは特にビジネス・アーキテクチャの考え¹⁾を重視した方針の選択について考察する。

表1：ケース間データ比較

| | ケース1 | ケース2 | ケース3 |
|-----------|----------|----------|----------|
| 対象分野 | 燃焼解析 | 医療画像 | 検査装置 |
| 企業規模（社員数） | 40名程度 | 10名程度 | 900名程度 |
| 対象技術者数 | 2名 | 7名 | 9名 |
| 技能範囲 | 初級・上級 | 中級～上級 | 初級～上級 |
| カスタマイズ | 非カスタマイズ品 | 高カスタマイズ品 | 高カスタマイズ品 |
| 業務標準化 | なし | なし | あり |
| 開発方針 | 主従 | 個別担当 | マトリックス |

ケース1のように対象技術者の絶対数が少ない場合、案件毎に顧客からの要望が多い高カスタマイズ製品を対象とすると労働集約型の形態となるため、対応可能な業務総量の上限に達しやすくなってしまう。そのため、企業として売上や利益の確保を最大限に活かすことが困難となる。よって、主たる技術者の能力を製品内部のすり合わせに集中させ、客先でモジュール的に利用可能な製品を顧客に販売できることが望ましい。例えばケース1の場合は、1計測システムの機能を小型軽量化することで、特殊光学センサーモジュールとして新たな製品化を実施すること

が考えられる。一方で、あえて高カスタマイズ製品の取り扱いを採用する場合は、新規のシステム開発であれば顧客の高カスタマイズ要求に耐えうるようシステム内部をモジュール型で開発する方針を当初から採用すれば良い。既に高カスタマイズ製品が存在する場合は、当面は主たる技術者の能力に頼るしかない。その間は主従関係をもつて続く技術者を育成するか、主たる技術者と同等レベルかそれ以上の技術者を着実に採用してゆく必要がある。

ケース2のように対象技術者数が一定数在籍しかつ十分な技能を有する場合、ケース1の時と異なり高カスタマイズ製品を対象とした戦略を選択することも可能となる。しかしながら対象製品がさらに増加することでシステム間連携の複雑性が増大してゆく。そのような状況では、全システム間の連携も含めた部分ですり合わせ型だけの製品構成では、システム開発にかかる工数の増加を抑えることは困難となる。よって、システム間連携部分やシステム内部の開発方針をモジュール型へとシフトさせてゆく必要がある。システム間連携部分等の非常に複雑な箇所に関しての高い機能レベルや安定性を示すためのノウハウを長期間継続して蓄積させてゆくことで、特にケース3のような開発方針を選択する企業に対して模倣困難性として現れ、企業の差別化を図ってゆくことが可能である。またケース2のような企業が新規に事業を拡大してゆく時はケース2への人員増強だけでなく、ケース1のようにシステム内部をすり合わせ的に開発する選択肢も存在する。

ケース3のように対象技術者は一定数在籍するが技能にばらつきを有する場合、ケース2のように各システムに1人ずつ技術者を分担させることは、開発速度やシステム品質保証の面から考えても妥当ではない。そのため高カスタマイズ製品であるか否かに関わらず、最初からモジュール型&業務標準化によるシステム開発方針を選択する必要がある。モジュール型&業務標準化のシステム開発方針を選択することのメリットには、すり合わせ型のシステム開発方針と比較して、人的リソースを増強させてゆくことで品質を一定に保ったままより多くの案件を受注することが可能になることである。案件の増加によって、規模の経済や経験曲線といったコスト抑制戦略を有効に活用することが可能となる。一方で、業務標準化により技術者技能に対して一定以上の技能が身に付かないという成長の壁が生じてしまうというデメリットが生じる。この問題を少しでも解消するためには、企業内の他事業にて一定のすり合わせ型のシステム開発方針を採用し業務ローテーションを実施することで、技術者技能の深化を推進させることも必要であると考え。ただし前川製作所²⁾のようにすり合わせ型の製品開発を特徴としている大企業も存在するため、企業規模が大きいからといって必ずしも業務分担化・標準化の開発方針だけが選択となるとは限らない。

6. おわりに

本稿では、企業の規模や取扱製品が異なる3つのケースを取り上げた。また情報システムの開発方針の選択について比較分析した。中小企業では人的リソースにより事業の拡大は制限されるが、開発方針の選択肢は残されている。人的リソースが乏しい間は一定の開発方針を継続して選択し続けてゆく必要がある。企業規模・技術者数が増加してゆくにつれて、様々な開発方針を逐次選択してゆくことが可能となる。企業成長のある時期から企業は自身の規模により業務の分業化や業務の標準化が推進される。しかし、それ自身が制約となるため、適宜最適な開発方針を選択することは困難となってゆく。このような企業の内部・外部環境に基づいたシステム開発方針の選択は、単独のシステム開発プロジェクトに限定せず、システム開発方針のポートフォリオとして全体を俯瞰する分析とマネージメントも重要ではないかと考える。

参考文献

- 1) 藤本隆宏、武石彰、青島矢一（2001）『ビジネス・アーキテクチャ』有斐閣
- 2) 「冷凍の巨人、無競争に生きる」（2012）NIKKEI BUSINESS