

---

---

## 当社における独自開発システムの課題と展開について

株式会社 ズコーシャ

---

### ■ 執筆者Profile ■



徳岡 竜太郎

2013年 株式会社 ズコーシャ 入社

2014年 IT事業部 システム1課 所属  
自治体業務担当

### ■ 論文要旨 ■

当社の独自開発システムの構築において、各開発者が独自の思考でシステム構築を行うことにより、システムの開発言語や規約が異なっている。このため、その独自システムを開発した者以外は、システム管理や運用保守が困難なのが当社の現状である。また、開発者によるシステムの管理や運用保守作業は、開発者の作業負担増加により、作業効率の低下が懸念されている。

本論文では、全ての業務システムに共通するログイン機能などを「共通基盤」として準備することにより、今後の独自開発システムの開発規模の縮小化（各開発者の作業負担軽減）と、開発規約の制定によるシステム保守性の向上を見込めることについて論述する。

## ■ 論文目次 ■

<u>1. はじめに</u> .....	《 3》
1. 1  当社の概要	
1. 2  背景	
<u>2. パッケージ製品と独自開発システムの特徴</u> .....	《 4》
2. 1  パッケージ製品の特徴	
2. 2  独自開発システムの特徴	
2. 3  比較	
<u>3. 現状と課題</u> .....	《 5》
3. 1  独自開発システム構築の現状	
3. 2  課題	
<u>4. 課題解決に向けて</u> .....	《 6》
4. 1  共通基盤の構想	
4. 2  共通基盤導入のメリット	
4. 3  共通基盤導入のデメリット	
4. 4  デメリットに対する打開策	
4. 5  実施に向けた計画	
<u>5. 今後の展開</u> .....	《 10》
<u>6. 終わりに</u> .....	《 10》

## ■ 図表一覧 ■

図1  当社の沿革 .....	《 3》
図2  システム開発の工程.....	《 4》
図3  共通基盤の概念図 .....	《 7》
表1  パッケージ製品と独自開発システムの比較.....	《 4》

# 1. はじめに

## 1. 1 当社の概要

当社は、「環境・農業・まちづくり」のテーマをもとに、土木・建築、環境調査・研究、測量、IT などの総合コンサルタントとして、社会のインフラを支える業界のリーディングカンパニーを目指し、1959年に設立された企業である。（図1）

当社におけるIT分野の事業では、主に顧客のシステム企画・設計・構築・運用のすべてのフェーズでサポートするトータルアウトソーシングや、当社データセンターにて、顧客システムのリソースを管理・監視し、顧客の負担を低減させるハウジングサービスなどを行っている。またITソリューションの1つとして、地方自治体、民間企業が利用する業務システムの構築や運用、保守管理などのサービスを提供している。

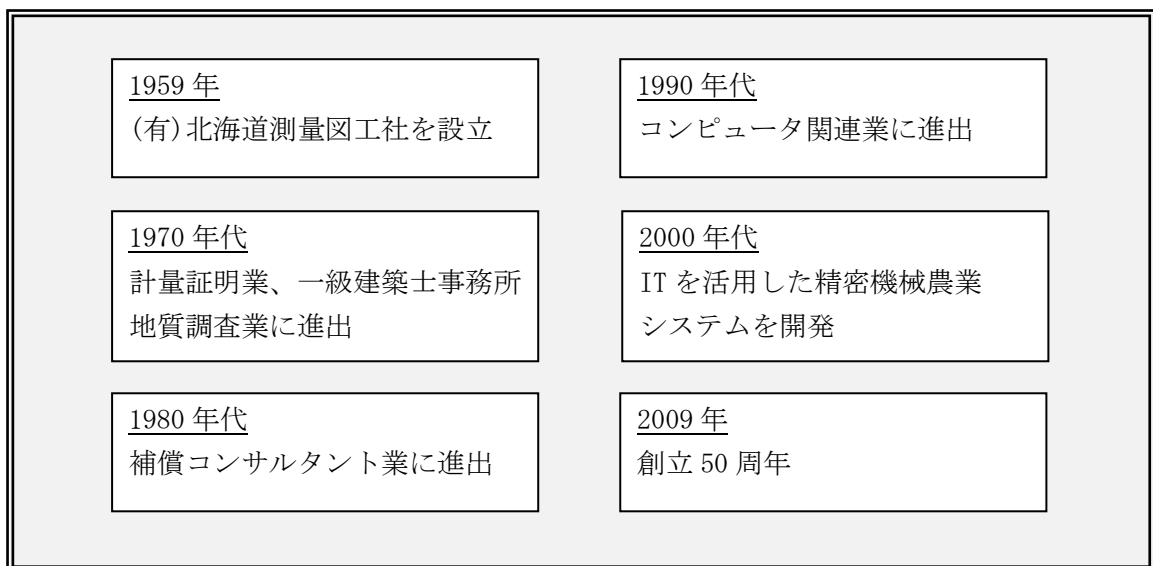


図1 当社の沿革

## 1. 2 背景

顧客が利用する業務システムの構築において、既製のパッケージ製品を用いて構築するものと、要件を取りまとめ、当社が独自に構築するものがある。両者を比較した際に、システム構築に掛かるコストや、導入までの期間に相違点がある。

両者の特徴をもとにして、各顧客の要件に合わせたものを選定し提供をしているが、当社における独自開発システムの構築方法については課題点があり、より質の高いシステムやサービスを顧客に提供できるようにするためには、効率的なシステム開発と運用ができるような仕組み、即ち独自開発システム構築における枠組みや、基礎となる部分が必要があると考えた。

## 2. パッケージ製品と独自開発システムの特徴

システムには、独自開発システムの他に、パッケージ製品と呼ばれるものがある。

### 2.1 パッケージ製品の特徴

パッケージ製品とは、既にシステムが完成している状態で、システム設計、開発などの工程を必要としないものである。

### 2.2 独自開発システムの特徴

独自開発システムは、図2のように要件定義、設計、開発、テストなどのシステム開発工程を必要とする。システムの開発規模が大きくなると、導入までの期間が長くなる傾向がある。また、システムの要件漏れによる手戻りのリスクがある。

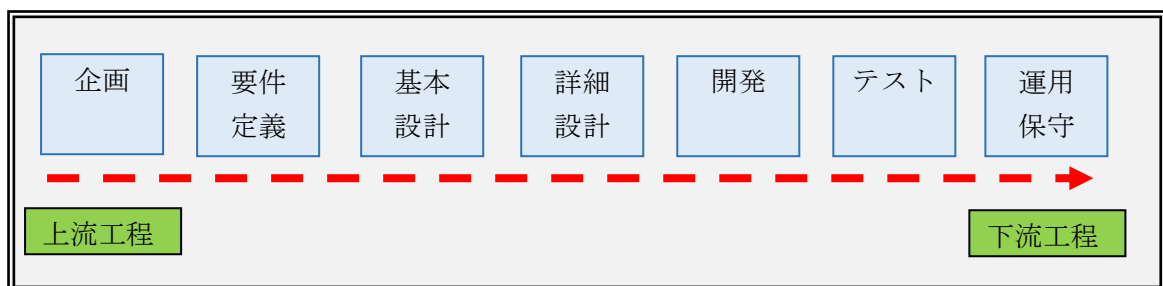


図2 システム開発の工程

### 2.3 比較

業務システムの構築において既製のパッケージ製品と、独自開発のシステムの相違点について、特徴を簡潔に比較したものが表1になる。表1をもとに考察すると、既製パッケージ製品は低コストであり、導入までの期間が短いことが大きな特徴である。業務システムの導入において既製のパッケージ製品は、独自開発システムよりも選択しやすいものとなっている。

対して独自開発システムは、柔軟性に優れており、顧客からの要件などを忠実に反映し、システムを構築できるのが特徴である。また、独自開発システムは一から構築を行うため、予算の折り合いを付けながら、顧客の環境、予算に合わせたものを構築することができる。

表1 パッケージ製品と独自開発システムの比較

	パッケージ製品	独自開発システム
コスト	低い	高い
導入期間	短い	長い
安定稼働までの期間	短い	長い
柔軟性	低い	高い

### **3. 現状と課題**

#### **3.1 独自開発システム構築の現状**

当社で開発した、独自開発のシステムは多種多様なものが存在するが、当社の現状としてどのシステムにおいても、運用、保守、障害対応、システム管理は、システム開発に携わった者に集中して行っている傾向がある。懸念される事項として、複数の独自システムを開発している技術者に対してより一層業務が集中し、業務の負担が増加することである。

また当社における独自システムの開発は、システム開発者の意向によって仕様が左右されやすいところがある。例として既存システムの活用、開発者が得意とする言語によってシステムの仕様が決まってしまうことが挙げられる。

さらにシステム開発の案件ごとに、個々の技術者が独自の組み方で構築をするため、作成者がその後の管理者になりがちである。万が一システム障害や、システムの改修作業があった場合、システム開発が終わった後も引き続き、システムの開発者が対応しなければならない。

#### **3.2 課題**

当社における、独自システム開発の現状課題を以下に示す。

##### **① システム開発者の作業負担が大きい**

作業負担の増加による他業務への影響と、肉体的・精神的負担により作業効率の低下が懸念される。

##### **② 開発言語やシステムの組み方に統一性が無い**

システム開発者以外の方がレビューする際に、システム構成の把握に時間が掛かってしまい、非効率であることが懸念される。

##### **③ システム開発者以外の技術者が、保守や障害などの対応ができない**

システム開発者が繁忙または不在の場合、対応に時間が掛かる。また、システム障害などのトラブルに対応できなかった場合、顧客側の業務にも多大な支障をきたすことが懸念される。

## 4. 課題解決に向けて

### 4.1 共通基盤の構想

「3.2 課題」で述べた3つの課題について、以下の構想によって解決すると考える。  
(図3)

業務システムとは、業務に関する機能だけではなく、ログイン機能や利用者権限管理が必ず備わっている。これらの機能は、業務システムの数だけ必ず存在するということになる。

仮に複数の業務システムのログイン機能や、利用者権限管理をひとつの基盤上で管理・利用することが可能になれば、今後新規の業務システムを構築する際に、改めてログイン機能や利用者権限管理を備える必要がなくなり、システムの開発規模を小さくすることが可能になる。このことにより、「3.2 課題」で述べた①（システム開発者の作業負担が大きい）についてシステム開発者の作業負担を軽減することができる。

要約すると、当社で構想しているシステム（これを本稿では「共通基盤」と呼ぶ）に、複数の業務システムのログイン機能や、利用者権限管理を一括管理をして、共通化できない個々の業務機能は、共通基盤の配下に設置することにより、各業務は1つのログイン機能と、利用者権限管理を利用して、アクセスできるようになるということである。

さらに開発規約を設けて、「3.2 課題」で述べた②（開発言語やシステムの組み方に統一性が無い）を解消する。開発言語や共通基盤と、各業務機能をつなぐインタフェースを統一化することにより、システム保守性が向上する。このことによって、「3.2 課題」で述べた③（システム開発者以外の技術者が、保守や障害などの対応ができない）について、システム開発者以外の技術者が保守できるようになる。

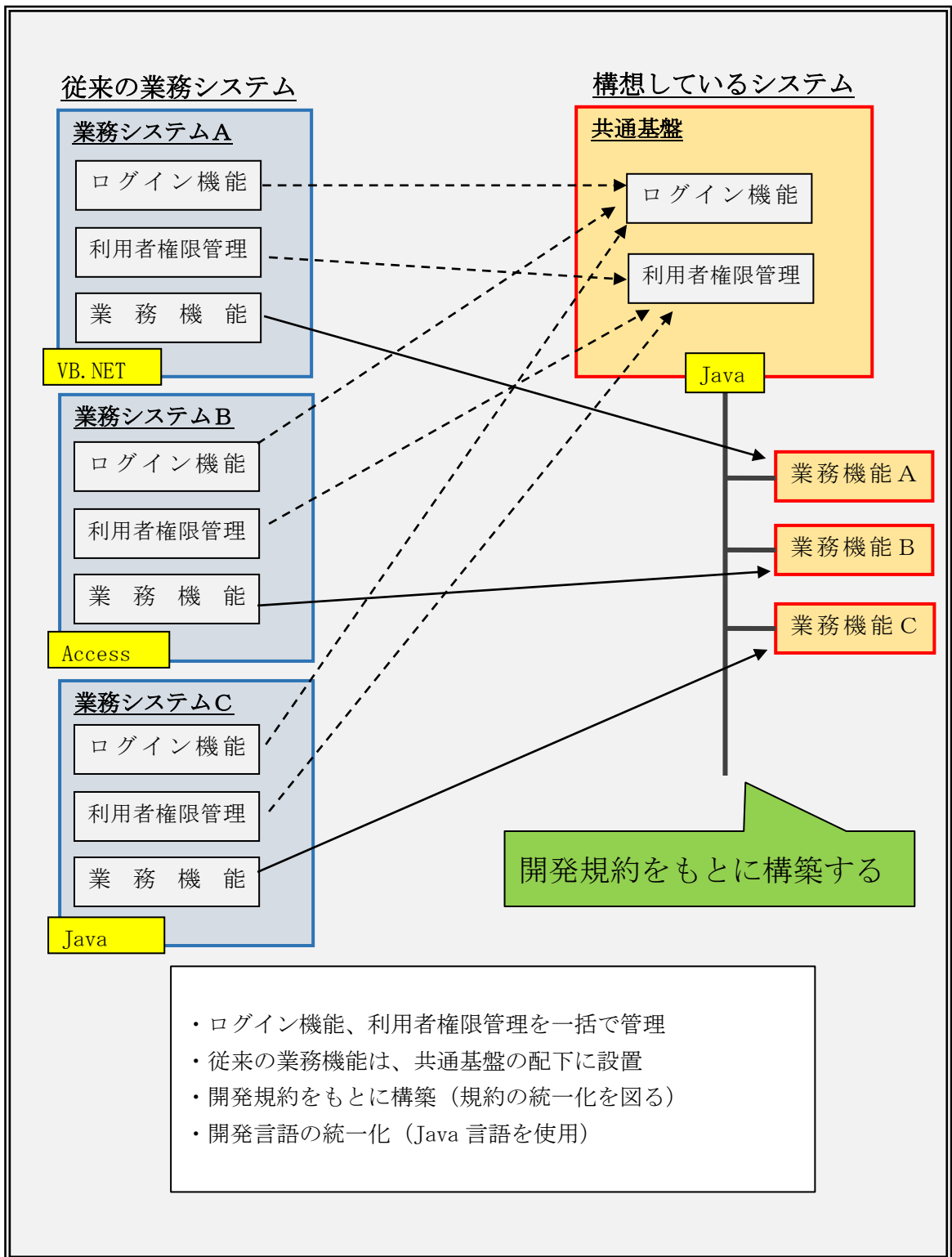


図 3 共通基盤の概念図

## 4. 2 共通基盤導入のメリット

共通基盤の構築により、「3.2 課題」に記載した3つの課題の低減もしくは解消が、①～③のように見込まれ、加えて④のメリットがある。

### ① システム開発規模の縮小による作業負担の低減

業務システムのログイン機能や利用者権限管理の部分を共通化することにより、業務機能の開発のみで構築することができるようになるため、当社の開発規模が縮小し、技術者の作業負担が低減する。

### ② 社内開発技術の統一化による業務の平準化

共通基盤の開発言語として Java 言語を採用し、Java 言語の研修・教育により知識を身につけ、技術者個々にある技術格差の緩和を図る。これにより、開発者以外の人でもシステムを管理できるように、社内における開発技術が統一され、システム開発者の業務を平準化させることが可能となる。

### ③ 開発規約によるシステム保守性の向上

当社の業務システムでは、各システムごとに設計書のフォーマットや開発言語が統一されていなかったが、開発規約によって統一性を図ることにより、設計書やプログラムの可読性が向上し、システム開発に携わっていない人でもシステムの保守が可能となる。

### ④ 作成したシステムを、多くの顧客に展開することが可能

共通基盤を導入することで、クライアントサーバモデルから Web 型システムに切り替えることができる。

システムを Web 型で提供することによって、各顧客先のクライアントごとに行う障害対応が不要となり、作業負担が低減し、少ない人数でより多くの顧客にシステムを展開することが可能となる。



#### 4.3 共通基盤導入のデメリット

共通基盤の構築によるデメリットを以下に示す。

##### ① 構築にかかるコスト・開発に必要なリソースの確保が困難

システム開発としては規模が大きいため、現状の業務と並行して行うのは困難である。また、共通基盤の開発に必要な人員の確保が必要になる。

##### ② 開発規約により作業効率の低下

新たに開発規約を設けることにより、システムの設計方法や、プログラムの記述方法などの細かな部分にも制約が発生し、開発者独自のルールによる構築方法から、開発規約に基づいた構築方法へと切り替えていく必要がある。そのため、システム開発者が、開発規約に習熟するまでの間、作業効率が低下する。

#### 4.4 デメリットに対する打開策

共通基盤の構築において、メリットだけではなくデメリットになる部分もある。「4.3 共通基盤導入のデメリット」に記載した①（構築にかかるコスト・開発に必要なリソースの確保が困難）については、打開策としてシステム部分を外注することによって対応できる。

しかし、「4.3 共通基盤導入のデメリット」に記載した②（開発規約により作業効率の低下）については、「3.2 課題」に記載した③の課題（システム開発者以外の技術者が、保守や障害などの対応ができない）が低減されるが、「3.2 課題」に記載した①の課題（システム開発者の作業負担が大きい）を増大させてしまうので、今後の検討課題とする。

#### 4.5 実施に向けた計画

共通基盤の実用化に向けて、以下の計画を進めている。

##### ① システム化対象範囲の定義

共通基盤に、どの機能を実装（共通化）するか定義する。「4.1 共通基盤の構想」では、ログイン機能、利用者権限管理について記述しているが、共通化できる機能は他にも存在し、ログ出力機能や、ライセンス管理など様々なものがある。そのため、共通化した場合のメリット・デメリットを各機能について比較検討し、共通基盤上に実装する機能を選定する必要がある。

##### ② 開発規約で規約化するものを明確にする

開発規約で規約化するものを定める。規約化するものとしてインタフェース設計や、プログラムの記述方法などを想定しているが、明確ではない。そのため、より詳細に開発規約で規約するものを定める必要がある。

##### ③ 共通基盤に関する社内研修の実施

各技術者に対して、共通基盤及び開発規約に関する研修を実施する。前述の「4.3 共通基盤導入のデメリット」の通り、システム開発者が開発規約に習熟するまでの間、作業効率が低下する。そのため、習熟するまでの期間を短縮する。

## 5. 今後の展開

今後の展開として、独自開発システムによる業務システムの導入から運用・保守などの作業を従来よりも効率良く低コストに行い、共通基盤を広く展開させることで、社内の開発における環境をより良いものに変化していけるようにしたいと考えている。また、将来的には低コストを新規顧客獲得の武器として、共通基盤の技術を活用していきたい。

## 6. 終わりに

当社の独自システム開発における課題は、共通基盤の仕組みを取り入れることによって解決が見込まれる。さらに、共通基盤の仕組みによって、より多くの顧客に対してシステムを展開することができるようになる結論づけた。

最後に、業務の効率化による社内環境の改善と、業務拡大による更なる会社の発展を目指して努めていきたい。