

# 海外の銀行業務パッケージシステム

## 導入プロジェクトのチェックポイント

(株) さくらケーシーエス

### ■ 執筆者Profile ■



黒田 晃弘

2002年 (株) さくらケーシーエス入社  
金融コンサルティング業務担当  
2014年 現在 金融システム三部所属  
金融コンサルティング業務担当

### ■ 論文要旨 ■

一般にパッケージシステムとは導入が簡単で、安く、早くできると考えられている。筆者は、銀行向けの海外パッケージの導入に複数関わった経験がある。その経験より、銀行海外パッケージ導入は安く、早く、簡単にできなかった事例を述べる。パッケージシステムで複雑な銀行業務を実現するためには、パラメータの設定やカスタマイズが必要になる。また、パッケージ機能の詳細な確認やシステム基盤面での確認も必要である。

また、海外の銀行業務パッケージ導入でうまく導入できた事例についても紹介する。これらの事例を振り返り、筆者の反省をもとに海外のパッケージ導入に対するチェックポイントを述べる。

## ■ 論文目次 ■

1.	はじめに	《 3》
2.	うまくいかなかった海外パッケージ導入プロジェクトの概要	《 3》
2. 1.	プロジェクト計画の工夫	《 4》
2. 2.	プロジェクト運営について	《 5》
2. 3.	プロジェクトで発生した問題	《 5》
2. 4.	プロジェクトの反省点	《 6》
3.	二つめの海外パッケージ導入プロジェクトの概要	《 7》
3. 1.	プロジェクト計画の工夫	《 8》
3. 2.	プロジェクト運営について	《 8》
3. 3.	プロジェクトの運営について	《 6》
3. 4.	プロジェクトの完了	《 10》
5.	おわりに	《 10》

## ■ 図表一覧 ■

図2.1	プロジェクトAの体制	《 4》
表2.1	海外パッケージ導入での反省点	《 5》
図3.1	プロジェクトBの体制	《 8》
表3.1	海外パッケージ導入での反省点	《 10》
表4.1	海外パッケージ導入時の顧慮点チェックリスト	《 11》

## 1. はじめに

日本の銀行のコンピュータシステムは、日本でシステム開発をして日本で利用することが大部分であった。なぜかという、日本の銀行の業務は、日本国内での業務が多くを占めていたからである。日本の銀行業務は特殊といわれ、日本の銀行には、個別に開発した銀行システムが適していると考えられてきた。過去には日本の銀行もバブル時代に海外への進出をしていたが、バブル崩壊後、一部の銀行を除き、ほとんどが撤退した。また、一部では海外パッケージの導入や国内で開発したものを日本で利用することもあった。しかし、それはインターネット専業の新規設立銀行や市場系業務等の一部のケースに限られていた。

しかし、現在状況は変化しつつある。従来日本でのみ活動していた銀行も本格的に海外進出を検討し初めている。一部の銀行はすでに預金、融資、送金等のあらゆる業務を銀行がコンピュータシステムを利用して海外での業務を開始している。今後、日本の銀行の海外業務拡大に伴い、コンピュータシステムの見直しが必要となる。銀行の業務拡大に伴い、急速なシステム導入が要求されるため、導入が容易であると考えられているパッケージシステムが検討されると想定される。

そこで、従来うまくいかないケースが多かった海外製パッケージシステムの導入について、筆者がコンサルタントとして反省した点と反省を生かしたプロジェクトでの工夫点を述べたい。

## 2. うまくいかなかった海外パッケージ導入プロジェクトの概要

このプロジェクト（以降、プロジェクトA）は、日本の銀行が国内の支店の基幹業務系システムに海外パッケージシステムを導入するものであった。このパッケージシステムは海外金融機関での導入実績は数多くあったが、日本の銀行が導入するのは初めてのパッケージであった。プロジェクトの体制は銀行の企画部門がオーナー部となりプロジェクトを立ち上げた。海外パッケージの要件定義とギャップ分析は、パッケージベンダーと銀行の業務部門が担当した。関連システムとパッケージシステムのインターフェースの開発は、銀行のシステム関連会社が担当した。パッケージへのカスタマイズは、パッケージベンダーが担当した。パッケージベンダーは東京にオフィスがあったが、本社はシンガポールにあり、開発拠点はインドであった。プロジェクト期間は約1年間半のプロジェクトであった。総工数はおよそ1000人月。プロジェクトのメインオフィスは東京であった。

筆者は銀行の企画部門へプロジェクト推進支援として参画し、システム化計画、業務要件定義支援、課題管理、進捗管理、パッケージベンダーとの交渉窓口を担当した。

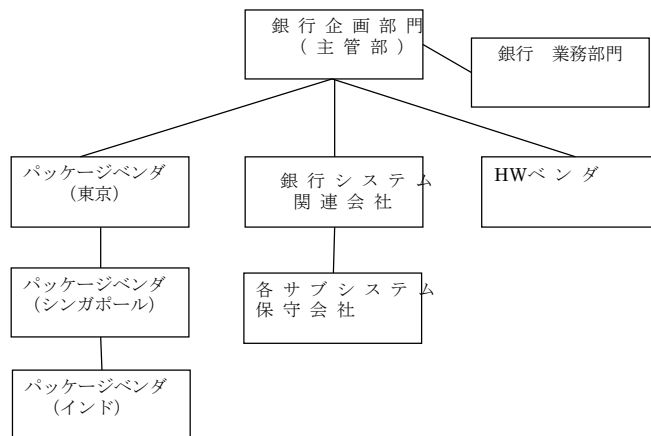


図 2. 1 プロジェクト A の体制

## 2. 1. プロジェクト計画の工夫

このプロジェクトは海外のパッケージシステムを国内に導入するものである。そこで、国内のパッケージシステム導入と同様の課題があると考えた。プロジェクトの計画策定時に国内のパッケージシステム導入の経験を生かし次のような工夫をおこなった。

### 2. 1. 1. フィット&ギャップの進め方

業務要件定義においては日本の銀行業務とパッケージ機能とのフィット&ギャップ分析より開始した。パッケージシステムは海外の実績が豊富にあるという安心感より、大まかな違いはないと判断した。銀行業務機能の一覧表に基づいて、パッケージベンダのコンサルタントと銀行の各業務担当者が一組となり、パッケージ機能で実現可能かをヒアリングを行った。

### 2. 1. 2. パッケージ導入期間の短縮

海外で実績のあるパッケージであり、フィット&ギャップでのギャップが少なかったため、通常の個別システム開発と比較して、3分の2程度の期間で導入可能と判断した。一般的に銀行の基幹業務システムのパッケージ導入では2年から2年半かかる。しかし、カスタマイズ部分をなるべく少なくし、銀行の業務をパッケージの基本仕様に合わせることを基本方針として、このプロジェクトでは1年半を導入期間としてスケジュールを作成した。

### 2. 1. 3. 現行システムのデータ移行チーム立ち上げ

銀行の基幹系業務パッケージのアーキテクチャが現行システムとかなり大きく違うことが判明していた。そのため要件定義当初より、システム移行が可能かどうかの検討を開始した。銀行の業務部門のメンバに加え、現行基幹業務システムのメンバ、パッケージベンダのメンバで検討を行った。

### 2. 1. 4. カスタマイズ開発について

ローンの利息計算方式の追加、マーケットレート取り込み機能、画面の日本語化、帳票

の日本語化をギャップとして抽出し、カスタマイズ項目とした。画面構成の変更や画面の項目は自由に追加可能とのことであった。そこで、要件定義の漏れがあったとしても、後で対応可能であると判断した。基本設計以降の実際の開発はインドを中心としたオフショアで行うこととした。

## **2. 2. プロジェクト運営について**

プロジェクトAは、プロジェクトの言語を日本語中心で行った。そのため、パッケージベンダーとの会議は銀行側が日本語で話し、通訳が英語に訳して行った。また、ドキュメントは日本語で作成し、翻訳業者に翻訳を依頼した。パッケージベンダーからのドキュメントは英語で作成されるので、逐次翻訳業者が翻訳を実施した。システム開発における用語には専門用語が多く、そのため通訳を介しての会話やドキュメントのやり取りはうまく伝わらないことも多く、その確認に多くの時間を費やした。

## **2. 3. プロジェクトで発生した問題**

### **2. 3. 1. カスタマイズ開発**

パッケージベンダーのカスタマイズ開発が完了し、日本で受け入れテストを開始した。すると要件定義で要求したカスタマイズ機能がうまく実装されていないことが判明した。

ローンの利息計算方式について、両端日数計算方式の追加を要求した。しかし、実際に追加された計算機能は実行当初の利息計算だけでなく、一部元本繰り上げ返済時にも両端計算が適用されていた。つまり、借入人が繰り上げ返済をすると、一日分余分に利息を返済したことになっていたのである。

また、マーケットレートの取り込み機能についても不具合があった。この機能はシステムにファイルで取り込む仕組み、マーケットレートをパッケージのデータベースに格納する機能は作りこんでいたが、画面に表示する機能やマーケットレートを取り込めなかった場合の機能は実装されていなかった。

画面や帳票の日本語化についても同様で、日本語を登録可能なようにデータベースの項目をシングルバイト定義からダブルバイトの定義に変更されただけであった。パッケージが実際の画面や帳票は仕組みを提供するので、銀行の導入メンバが対応する必要があると説明された。

さらに、カスタマイズしていない機能についてもシステムテストを開始してみると、多くの不具合や想定通りの動きをしないことがわかった。これは、パッケージにバージョンがいくつも存在し、現在提供されたパッケージは最新の提供予定のものであった。そこで、機能は追加されたのであるが、実際の銀行がユーザとして利用しておらず、パッケージシステムとしての機能確認が十分に実施されていないものであった。

### **2. 3. 2. 仕様の認識齟齬の発生**

要件定義でのパッケージベンダーの説明ではできると説明を受けていた機能でも、実はカスタマイズすればできるという説明であった。確かにマニュアルには機能があることは記載されているが、どのような機能かについては、明確な記載はなかった。

### 2. 3. 3. プロジェクトの中断

システムテストを開始後、カスタマイズ項目の不具合への対応、日本語化への対応等に想定以上の工数がかかった。さらに、カスタマイズしていない基本機能についての不具合の対応工数を見積もるとさらに工数は膨れあがった。最終的にプロジェクトの継続は難しいと判断して、銀行がプロジェクトAを中断することになった。

### 2. 4. プロジェクトの反省点

#### 2. 4. 1 基本機能の確認

パッケージシステムは、既に他社で利用実績がある場合が多い。導入銀行はすでにシステムとして利用実績があるため、一からの設計や開発が不要で導入に時間がかからないと考えがちである。プロジェクトAは、導入銀行がプロジェクト計画段階で稼働実績の多いパッケージであれば、品質も高く、短期間で導入できるのではないかと思いついてきた。そのため、プロジェクト計画時に、カスタマイズしない機能についての確認に十分な期間と工数を見込んでいなかった。

導入銀行はパッケージ機能を調査して、自行の業務プロセスとどこが不適合なのかを洗い出す。パッケージ機能のギャップ毎にカスタマイズ工数を見積もり、代替案がないかについても検討する。その後、予算と期限を考慮して、カスタマイズする機能を決定する。決定に基づき、パッケージシステムのカスタマイズを実施し、システム導入を行うのが一般的である。

しかし、実際は他銀行でパッケージシステムのすべての機能を利用しているわけではないケースが多い。新規に導入する銀行はどの機能が他銀行で実績があり、どの機能は実績がないとは知るべきがない。実際に銀行が業務を稼働させようとする、うまくいかないケースが多い。代表的な例は、パッケージシステム機能のうち、初めて利用する機能の場合である。パッケージシステムを利用する業務開始直前に判明したのでは、時間的に対応策を検討することは難しい。しかし、早期に不適合な点を見つけることにより、銀行の業務処理をパッケージシステムの機能に合わせることを検討できる。

#### 2. 4. 2. カスタマイズ機能について

カスタマイズ機能の開発は、要件を細かく提示し、お互いに理解の齟齬がないかを通常のシステム開発と同様に段階的なプロセスを実施していく必要がある。しかし、一般的に銀行がパッケージシステムを採用する理由は、コストが安い、導入期間が短い、他行実績があるため、細か要件定義や設計が省けると考えている。この点に、導入銀行の過度なパッケージ導入に対する期待と、パッケージ導入の実態との乖離があった。

プロジェクトAは、カスタマイズ開発の拠点をオフショアで実施した。要件定義以降の基本設計から請負契約としてオフショアを中心として開発を依頼した。そのため、各工程の成果物の進捗はカスタマイズ機能の受け入れまで現物を確認することができなかった。納品されてきたものを確認してみると、大きな認識の違いがあった。

### 2. 4. 3. テスト&機能確認

プロジェクトAでは、海外で多く導入実績のあるパッケージであると理解していたため、カスタマイズ開発の完了を待って、業務機能のテストを実施した。そのため、当初導入銀行が想定していた動きとパッケージが違う動きをした機能について、その対応に多くの時間が必要となった。導入銀行の業務フローはすでに決まっており、パッケージの機能が想定と違っていたからといって、簡単には変更できない。すでに関連部署と業務フローは合意されており、その変更には関連部署の合議が必要である。

たとえば、他システムから顧客属性の変更をシステム間インターフェースにより自動的に反映させるというフローが決まっていた。しかし、他システムからの顧客属性の変更はパッケージで受け付けるが、パッケージシステムの承認プロセスが必要であった。また、属性変更の反映処理がエラーとなった場合でも、そのエラーを検知する仕組みが備わっていなかった。このような場合、システム間インターフェースで属性変更を自動反映させるべきか、個別業務オペレーションで実施させるべきか、業務部門を含めて決定していく必要があった。当初より、導入銀行がこの機能についてテスト及び機能確認ができていれば、早く課題を解決できていたであろう。

要件定義などでフィット&ギャップを実施する場合は、パッケージのマニュアル等による説明を受け、銀行の要求しているものが本当にできるかを確認する必要があった。

### 2. 4. 3. システム基盤

パッケージを導入する場合、どのような基盤で稼働させるかも重要な検討項目である。Unix で稼働実績はあるが、Linux ではどうか。X社のハードウェアで稼働するがY社のハードウェアで稼働するか。この製品バージョンでは稼働した実績があるが、新しいバージョンでは未確認である、など様々な基盤面での検討も必要である。

以上の反省点をまとめると以下のようになった。

表 2.1 海外パッケージ導入時の反省点

プロジェクトでの確認	反省点
基本機能の確認	既存機能の確認不足
カスタマイズ機能の確認	詳細な実現機能の確認不足
マニュアルによる機能確認	ドキュメントによる機能の確認不足
システム基盤の確認	実績のあるシステム構成の確認不足

### 3. 二つめの海外パッケージ導入プロジェクトの概要

このプロジェクト（以降、プロジェクトB）は、日本の銀行が海外の拠点に対してパッケージシステムを基幹業務システムとして導入するものであった。この海外パッケージシステムはうまくいかなかったプロジェクトAと同様に海外銀行での導入実績は多くあったが、日本の銀行が海外拠点に導入するのは初めてであった。プロジェクトBの体制は導入する銀行の東京のシステム部門が導入計画の作成及び導入推進を担当した。業務要件の提示は海外拠点の担当者が提示した。パッケージシステムを販売している海外のベンダーがパッケージのフィット&ギャップ分析を行い、その結果にもとづきパッケージのカスタマイズ及び導入パラメータの設定を行った。東京本社のシステムとパッケージシステムのイ

インターフェース開発は東京本社のシステムを保守している日本のシステム会社が担当した。パッケージベンダーは東京にオフィスがあったが、本社はシンガポールにあり、開発拠点はインドであった。プロジェクトの中心は東京であったが、パッケージベンダーや海外拠点は日本語が話せるメンバがほとんどいなかったため、会議及びドキュメントは英語で実施した。プロジェクト期間は約2年半のプロジェクトであった。総工数はおよそ2000人月。プロジェクトのメインオフィスは東京であった。筆者はプロジェクトBの推進支援として参画した。

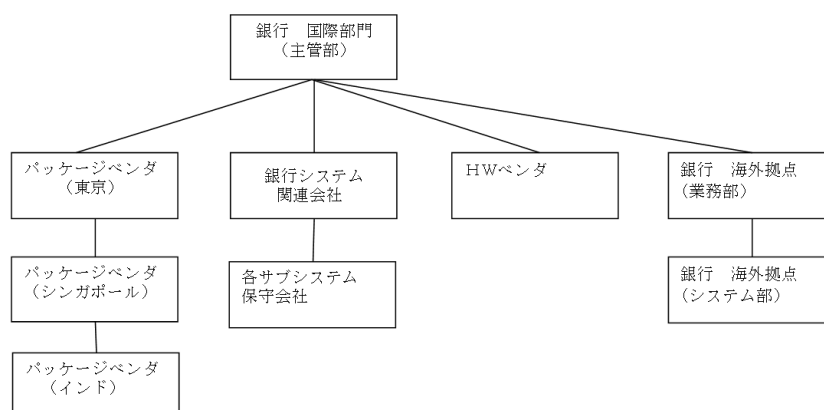


図 3. 1 プロジェクト B の体制

### 3. 1. プロジェクト計画の工夫

このプロジェクトも海外のパッケージシステムを導入するものであり、前述したプロジェクトと同様の課題があると考えた。そこで、プロジェクトの計画策定時に前述した反省点を生かし次のような工夫をおこなった。

プロジェクトBは、銀行内部開発にするか、パッケージ導入にするかプロジェクト計画時に作業項目の比較に多くの検討を実施した。そのため、パッケージ導入においても、基本機能についてのプログラム開発は工数の削減及び期間の短縮は可能であるが、要件定義、基本設計、システムテスト、運用テスト、ユーザ受け入れテストについては、ほぼ同じだけ期間や工数がかかると見込んでいた。

### 3. 2. プロジェクト運営について

プロジェクトBは、銀行の海外拠点向けシステムの導入であったため、銀行のプロジェクトメンバは英語によるコミュニケーションに問題はなかった。そのため、要件定義、基本設計や業務機能の確認が正確にできたのではないかと考えた。

### 3. 3. プロジェクトで発生した課題とその対応

#### 3. 3. 1 パッケージの基本機能の確認

プロジェクトは当初の目標に沿って、要件定義を開始した。筆者の経験より、海外パッケージで銀行業務処理を行うものは、カスタマイズを行わない部分についても十分なテストが必要であると認識していた。そこで、業務要件定義の開始と同時に、一般的な銀行業



務の基本機能のテストを計画した。まず、代表的な銀行業務である融資機能からテストを開始した。

パッケージシステムの基本機能の確認テストを実施してみると、依然経験したものと同様に、実績があるはずの機能についても、いくつかの不具合が判明した。

たとえば、負荷分散装置を介して複数の拠点の処理合算を行う処理において、トランザクションが大量発生すると、一部の処理が欠落してしまうことがわかった。

そこで、事前に不具合が判明した機能について、パッケージベンダーに不具合の確認を行い、プロジェクトの早い段階で解決することができた。

### **3. 3. 2. カスタマイズ開発について**

前述のプロジェクトAでは、カスタマイズの実施をオフショアで行った。そのため、どのような開発が途中で実施されているかを確認することが困難であった。そこで、プロジェクトBではオフショア開発の中心メンバである現地SEを東京に常駐させ、逐次開発の進捗や開発上のあいまいになっていた点をオンサイトで解決した。

パッケージベンダーのオフショア開発では、週次で進捗を確認する仕組みとなっていた。そのため、問題が発生しても1週間近く解決されない可能性があった。そこで、同じオフィスで開発することにより、開発メンバや開発リーダーからすぐに課題を挙げてもらい、解決できるようにした。

前回の反省を生かし、カスタマイズ開発完了後、すぐに詳細機能の確認が取れる体制とした。その結果、細かい仕様の認識相違は多数発生したが、いずれも短期間で解決することができた。

### **3. 3. 3. テスト&機能確認**

プロジェクトBでは、カスタマイズ機能の確認テストやシステムテストの前に、事前に基本機能の確認テストを実施した。パッケージの機能をテストしてみると、一部動かない機能や思った通りのユーザ権限の制御ができないことが判明した。調査してみると、一部機能が動作しない件に関しては、パッケージのバージョンが上がり、パラメータの初期設定を変える必要があったのだがその対応が漏れていた。ユーザ権限の制御が十分にできない件に関しては、追加のカスタマイズを実施して対応した。

このように、当初想定していたパッケージの機能とかい離があった場合でも、事前に対応策をとることができた。

### **3. 3. 4 マニュアルによる機能の確認**

英語による要件定義のフィット&ギャップでは、日本人が会話だけでうまく外国人と実施するのは難しい。そこで、要件定義時に実装する機能について、ギャップとして抽出したものは、ギャップリストに記載して管理した。要件定義期間中に、システムでの対策案、事務処理の変更による代替案などの検討を実施した。ギャップについては、業務処理への影響度、カスタマイズの規模、難易度、等も合わせて検討した。そして、要件定義終了時にカスタマイズを実施するか許容するかを判断した。

一方、既存の機能で実現できると説明されたものについて、マニュアルに記載があるか、

パラメータ等の設定変更で実現する場合はどのように実現可能かマニュアルをもとに確認した。記載がなく実現可能と説明をうけたものについては、逐次パッケージの設定を変更して実機で挙動を確認した。

### **3. 3. 5 システム基盤の確認**

このプロジェクトで利用したシステム基盤は、Red Hat 系 Linux 基盤であった。この基盤とパッケージの組み合わせは数多く導入されており、大きな不具合はなかった。

データベースはオラクルを用いたが、最新のバージョンを利用したため、SQLのチューニングやテーブル設定に一部修正が発生した。しかし、プロジェクト当初よりオラクルの最新バージョンを利用することを計画していたため、チューニングを実施して対応可能な範囲であった。

### **3. 4. プロジェクトの完了**

海外パッケージシステム導入時の経験を生かし、プロジェクトを進めた結果、予定期間を少し上回ったが、システムを稼働させることができた。ただし、機能確認やドキュメントの確認に多くの時間を費やしたため、プロジェクトの企画段階に想定していたコストを上回ることになった。

実際のシステム稼働時には、想定できていなかった不具合も数多く発生した。しかし、業務が止まったり、顧客に影響を与えるような事象は発生しなかった。

表 3.1 海外パッケージ導入時の注意点と対応

プロジェクトでの注意点	対応
基本機能の確認	プロジェクト初期での機能確認
カスタマイズ機能の確認	一部オンサイトでの開発・対面レビュー
マニュアルによる機能確認	ドキュメントの提出依頼とドキュメントの確認
システム基盤の確認	実績のあるシステム構成での構築

## **4. 終わりに**

導入実績の数多くある海外のパッケージシステムの導入であっても、銀行業務のように複雑で様々な機能があるパッケージでは、開発期間、コスト、機能、品質管理において内製でのシステム開発と大きな差はないといえる。要件定義では、銀行業務担当者及び銀行システム開発者と実施するが、パッケージ導入の場合はパッケージベンダーのシステム担当者と実施する。システム設計や開発は内製の場合は、既存のシステムをベースに新しいシステムを開発していく。パッケージ導入の場合は、パッケージ製品をベースにパラメータの設定やカスタマイズを実施していく。テスト実施については、内製開発であろうとも海外の実績あるパッケージ導入でもほぼ同じではないかと考える。

また、プロジェクト運営においては、日本語のわかるパッケージ技術者の調達、または、銀行側に英語でプロジェクト運営が可能な要員を十分に配置する必要がある。カスタマイズ開発については、チェックポイントを設けて、開発途中で十分な確認を実施していく必

要がある。これらは、内製システム開発であろうとパッケージ導入であろうと同様であるといえる。

日本の銀行は海外の拠点を設立、拡充を始めている。短期間で安く銀行のシステムを導入して、業務を開始したいと考えている。しかし、海外の銀行業務パッケージを導入する場合には、筆者の経験より日本のユーザでの導入実績、日本語でのプロジェクト運営可否、導入スケジュール、パッケージシステムのカスタマイズ方法などを十分に検討する必要があると考える。パッケージであるから簡単に短期で安く導入できるとは考えずに、内製システム開発と同様に要件定義、カスタマイズ開発、テスト等を十分に実施する必要があると考える。

最後に2つのプロジェクトを通じて筆者が感じた注意点を記載する。今後、海外パッケージを導入する上での手助けになれば幸いです。

表 4.1 海外パッケージ導入時の考慮観点チェックリスト

プロジェクトのフェーズ	観点
パッケージ選定	10件以上のユーザがいるか
	ユーザの規模は導入銀行と同じくらいの規模か
	複数国で導入されているか
	日本で導入実績があるか
	標準的導入期間はどれくらいか
	日本にパッケージ会社のオフィスはあるか
	パッケージに対するQAは日本語で可能か
	パッケージの機能説明書はあるか
	コア機能のカスタマイズは可能か
	顧客番号、口座番号などの桁数拡張は容易か
	営業店追加は容易か
	商品の追加は容易か
	カスタマイズの工数基準は明確か
	ユーザで画面・帳票のカスタマイズ可能か
	銀行業務とパッケージを理解したコンサルタントは多く参加可能か
導入フェーズ	カスタマイズ機能の成果物は工程毎にレビュー可能か
	日本語で開発レビュー可能か
	成果物の定義は明確か
	ドキュメントは日本語で作成されるか
	パッケージベンダーと導入銀行の役割分担は明確か
	導入するパッケージのバージョンは安定しているか
	導入するパッケージの保守期限は短期間で終了しないか