アジャイル開発による
ＳａａＳ型業界共通ＸＭＬ／ＥＤＩの構築

小島プレス工業（株）菅野　修一

■ 執筆者Ｐｒｏｆｉｌｅ ■

菅野　修一

1997 年　若宮工業（株）入社
　　品質保証業務担当
1999 年　営業業務担当
2000 年　システム業務担当
2010 年　小島プレス（株）転籍
　　ＩＴプロジェクト担当

■ 論文要旨 ■

企業間の情報授受は、インターネットの普及により独自Ｗｅｂ－ＥＤＩが拡大している。特に、自動車部品業界における課題は、業界標準ＥＤＩがなく、各社異なった方式により多端末・多画面現象となり、企業間のシステム連携ができていない。また、中小企業における課題は、システム費用が掛かるためコンピューター処理が殆どされておらず、紙・メール・ＦＡＸが主流となっており、ＥＤＩの整備ができていない。

そこで、自動車部品業界、及び中小企業において共通で利用が可能な「ＳａａＳ型業界共通ＸＭＬ／ＥＤＩ」を構築した。今回のシステム構築は、ＩＴにおける社会貢献として、「ＳａａＳ」を提供する側としてシステムを構築した。また、この「ＳａａＳ型業界共通ＸＭＬ／ＥＤＩ」構築では、従来のウォーターフォール開発では無く、アジャイル開発を適用した。このアジャイル開発における考慮事項、及び評価について述べる。

■ 論文目次 ■

■ 図表一覧 ■

１．はじめに

当社は「創立 72 周年」を迎えることができた。社是を「和」と定め、ものづくりはひとづくりとの考えで全員参加で取組んでいる。会社内における「和」だけではなく、家庭・地域・社会、更には国際社会へとつながり、当社の基本精神として、今も脈々と受け継がれている。

企業間の情報授受は、インターネットの普及により独自Ｗｅｂ－ＥＤＩ[1]が拡大している。特に、自動車部品業界における課題は、業界標準ＥＤＩ[2]がなく、各社異なった方式により多端末・多画面現象となり、企業間のシステム連携ができていない。また、中小企業における課題は、システム費用が掛かるためコンピューター処理が殆どされておらず、紙・メール・ＦＡＸが主流となっており、ＥＤＩの整備ができていない。

そこで、自動車部品業界、及び中小企業において共通で利用が可能な「ＳａａＳ[3]型業界共通ＸＭＬ[4]／ＥＤＩ」を構築した。今回のシステム構築は、ＩＴにおける社会貢献として、「ＳａａＳ」を提供する側としてシステムを構築した。また、この「ＳａａＳ型業界共通ＸＭＬ／ＥＤＩ」構築では、従来のウォーターフォール開発では無く、アジャイル開発を適用した。このアジャイル開発と経営改革、及び評価について述べる。

---

1 Ｗｅｂ－ＥＤＩ：インターネットを利用した企業間の電子取引をおこなうシステム。
2 ＥＤＩ：Electronic Data Interchange
3 ＳａａＳ：Software as a Service ソフトウェアーをサービスとして提供。
4 ＸＭＬ：Extensible Markup Language 情報を記述するための言語。

２．アジャイル開発の概要

２．１　ウォータフォール開発とアジャイル開発の特徴

アジャイル開発の特徴は、仕様の変更に柔軟に対応できオブジェクト開発を基本に、アプリケーションを構築していく手法である。

イテレーション単位で・要件定義・詳細設計・システム設計・実行テスト・確認・改良を繰り返す。全体を一括で行うウォーターフォール開発を、小規模の単位まで圧縮した形態となっている。そこで、ウォーターフォール開発とアジャイル開発の特徴を説明する。

＜ウォータフォール開発＞

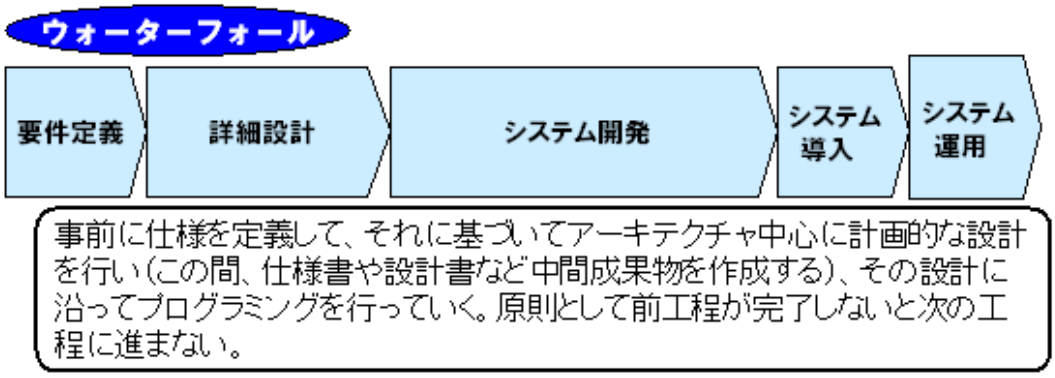

図１．ウォータフォール開発

ウォータホール開発は、プロジェクトによって工程の定義に差はあるが、開発プロジェクトを時系列に、「要求定義」「（詳細設計　外部設計を含む（外部・内部設計）」「システム開発　テストを含む（プログラミング）」「システム導入」「システム運用」などの作業工程（局面、フェーズ）に分割し、原則として前工程が完了しないと次工程に進まない（設計中にプログラミングを開始するなどの並行作業は行わない）事で、前工程の成果物の品質を確保し、前工程への後戻り（手戻り）を最小限にする。通常は「線表（ガントチャート）」を使用してスケジューリングする(図１．参照)。

プロジェクトの進捗は、一般的には、次のような納品可能な成果物をもとに計測される。

- 要件仕様
- 設計文書
- テスト計画
- コードレビュー

＜アジャイル開発＞

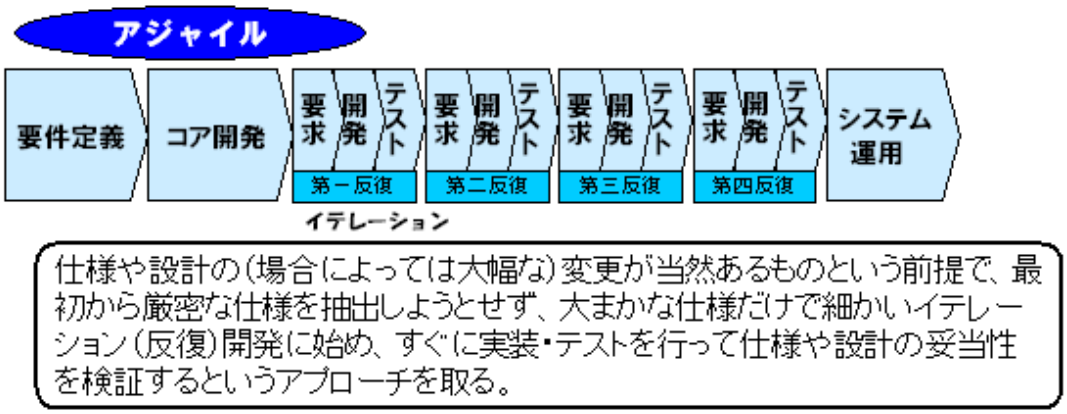

図２.アジャイル開発

アジャイル開発は、反復（イテレーション）と呼ばれる短い期間単位を採用することで、リスクを最小化している。 １つの反復の期間は、プロジェクトごとに異なるが、１週間から４週間くらいである（図２．参照）。

アジャイル開発は、開発対象を多数の小さな機能に分割し、１つの反復（イテレーション）で１機能を開発する（⇒反復型開発）。 この反復のサイクルを継続して行い、１つずつ機能を追加開発してゆく。各反復は、小規模なソフトウェア開発プロジェクトに似ており、そこでは、それまでに開発した成果物に１つの小さな機能を追加する。アジャイル開発手法では、各反復が終了するごとに、機能追加された新しいソフトウェア（ビルド）をリリースすることを目指す。 プロジェクトチームは、プロジェクトにおける優先度を評価し直す。

今までの開発では、業務の最適を考えて開発を行ってきた。そのため開発自体が単発となり他の業務との整合性・協調性を持たすための変更や修正に柔軟に対応できないアプリケーションになっている。（つぎはぎだらけである）
アジャイル開発ではイテレーションごとに問題確認・変更・修正を繰り返すため、コア部には全体業務を考え拡張性を持たせなければいけない。
今回の開発では、基本コア同士の連携に重点を置き、各コアの拡張性は柔軟に行えるようにし、コア同士の連携部分の変更不可能とした（コア連携部の変更はシステム全体の見直しにつながる）

２．２　アジャイル開発のモデル

アジャイル開発プロセスと呼ばれている開発手法には次のようなものがある。

・エクストリーム･プログラミング（Extreme Programming, XP）

アジャイル開発プロセスの代表として、その普及に一役買った。顧客の巻き込み、2 週間程度の繰り返し開発、テスト駆動などを特長とした、明確な技術的手法を持つ方法論である。

・スクラム (Scrum)

30 日ごとに動作可能な製品を作成するスプリント(Sprint)、毎日決まった時間に決まった場所で行われる短時間のミーティングなどを特長とした、プロジェクト管理を中心とした方法論である。

・クリスタル･ファミリー (Crystal Family)

プロジェクトの規模と影響の大きさに応じて数種類の方法論から成っている。
その中でももっとも小規模で影響の小さいプロジェクト向けの方法論クリア(Crystal Clear)は、XP ほどは厳格ではないし効率も高くない代わりに、多くのプロジェクトで採用しやすい方法論である。

・フィーチャー駆動型開発 《Feature－Driven Development, FDD》

フィーチャー(ユーザ機能)ごとに 2 週間の繰り返し開発を行う。この手法の提唱者であるピーター･コード(Peter Coad)が提案している UML を用いた設計手法とも密接に関連した方法論である。

・適応的ソフトウェア開発 (Adaptive Software Development, ASD)

ソフトウェア開発をカオス複雑系と見なし、それを前提に適応的なソフトウェア開発を行うことを特に考慮した方法論である。内容的には他の手法と共通する部分が多いが、Joint Application Development(JAD、ユーザや顧客が設計に参加する開発方法論)を採用しているのが少し変わっている点である。

・エクストリーム･モデリング (Extreme Modeling)

エクストリーム･モデリングは、UML を用いたモデリングを中心とした方法論である。ただし、現在よく行われている通常のモデリングとは異なり、常に実行可能/検証可能なモデルを作成する工程を繰り返し、最終的にはモデルから自動的に製品を生成する。

これらのアジャイル開発プロセスは、それぞれに異なった特長や守備範囲があり、ある面では相互に影響を与え合っている。アジャイル開発プロセスを採用している多くのプロジェクトでは、特定の手法だけを採用してマニュアルどおりに表面をまねるのではなく、複数の手法の中から自分たちのプロジェクトにあったプラクティスを取捨選択し、組み合わせて、さらには独自のプラクティスを追加して大きな効果を上げている。

２．３　アジャイル開発の必要性

最近のシステム開発の環境は、世の中の変化により要求に変化している。

・ビジネスの変化＝要求の変化
・スピードが求められる＝短期間開発
・従来の開発では解決できない＝要件の多様化

現在の経営環境は激しく変化をしており、経営と一体化している情報システムは経営環境の変化に対応する必要があるので、迅速なシステム開発が必須である。システム開発は便利なツールがたくさん登場して、開発自体はスピードアップできている。しかし、システム開発を成功させるのにコミュニケーションが最も重要になる。このコミュニケーションにはいろいろな意味が含まれている。直接対話のコミュニケーション、レビューによる１対多のコミュニケーション、そしてドキュメントを使った文字と図解によるコミュニケ

ーション･･･など。システム開発に限ったことではない。どんな業種のプロジェクトでも重要なことである。アジャイル開発はこのコミュニケーションを 重要視している点からも現在のシステム開発にもとめられている。

３．ＥＤＩの概要

３．１　ＥＤＩの動向

ＥＤＩとは電子データ交換のことで、本来はデータ交換のための各種標準規約を示すが、近年においては企業間の電子的な受発注取引を示す場合が多い。本論文ではＥＤＩを、企業間の電子データ交換とし、受発注取引はＥＤＩを利用する業務システムと位置づける。

初期のＥＤＩは、大手企業とその系列企業の間で専用線を結び、個別にデータ授受がすすめられていた。その後、ＶＡＮ[5]型のＥＤＩが誕生した。ここまでのＥＤＩは専用機を使用していた関係で受注側では取引先ごとの専用機を必要としたため「多端末現象」が起こり受注側にコスト・受注社内システムとの連携に問題を多く起こしていた

そして現在は、インターネットの普及に伴いＷｅｂ－ＥＤＩが主流となってきている。ただ、そのＷｅｂ－ＥＤＩは独自システムとなっているため、多画面現象等が発生し、仕入先の負担が増加している（図３．参照）。受注側では多端末から多画面現象へ移行しただけで業務の改善には寄与しなかった。

・専用線ＥＤＩ　　：発注先と受注先が直接専用線を結び、データを授受する方式
・ＶＡＮ型ＥＤＩ　：発注先はデータをＶＡＮ事業者に預け、受注先が受取る方式
・Ｗｅｂ－ＥＤＩ　：発注先がＷｅｂサーバーにデータを公開し、
　　受注先が内容を確認する方式

＜初期の
ＥＤＩ＞

＜１０年前の
ＥＤＩ＞

＜近年の
ＥＤＩ＞

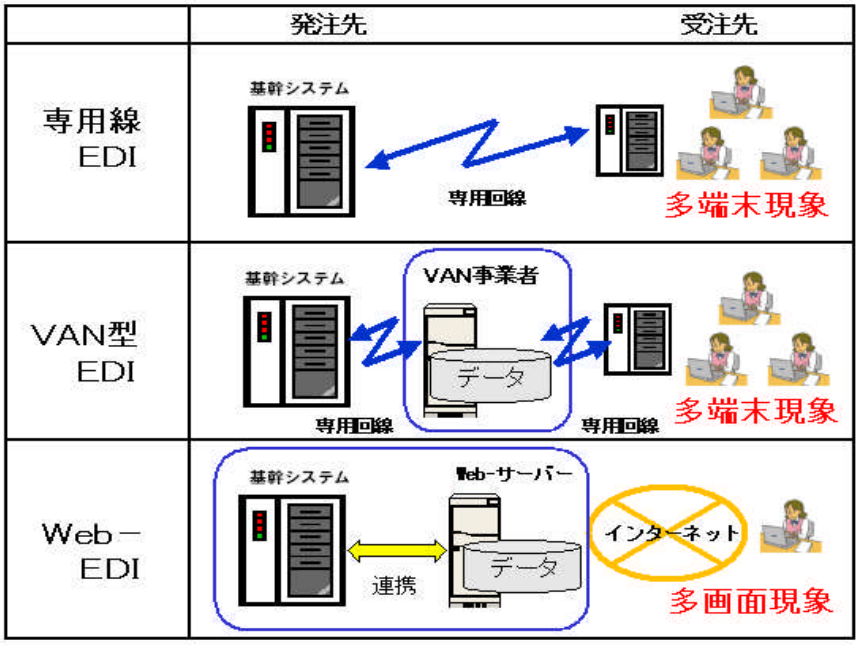

図３．今までのＥＤＩ

ＥＤＩ：Electronic Data Interchange（電子データ交換）

[5]ＶＡＮ：Value Added Network 付加価値通信網。

３．２　ＥＤＩにおける政府、業界の取組み

経済産業省は、２００８年１１月に業種や事業規模が異なる企業が、共通して利用できるＥＤＩの整備に着手した。特に中小企業にとっては、取引先が増えるほど個別フォーマットを変更する手間がかかるほか、コスト負担も大きく、大企業に比べると導入が進んでない。そこで経済産業省内に新たな仕組みを構築するために「ビジネスインフラ研究会[6]」を立ち上げた。

研究会では、取引先の業種が多岐にわたる自動車業界と電気・電子業界を対象に、データを簡易にやりとりできる標準的な仕組みの構築を、業界団体と一緒に検討をはじめた。

また、２００８年１２月に経済産業省中小企業庁は、「下請代金支払遅延等防止法[7]」に対する「素形材産業取引ガイドライン[8]」の改正をおこなった。その改正では、発注側が受注側に自社固有のＷｅｂ－ＥＤＩやＥＤＩ端末の導入を求めることを禁止した。また、規定に違反した場合、公正取引委員会から勧告され、最終的には排除命令措置となる。
今後、次世代ＥＤＩ推進協議会で認定したＥＤＩだけを、業界横断標準として認定するとしている（図４．参照）。

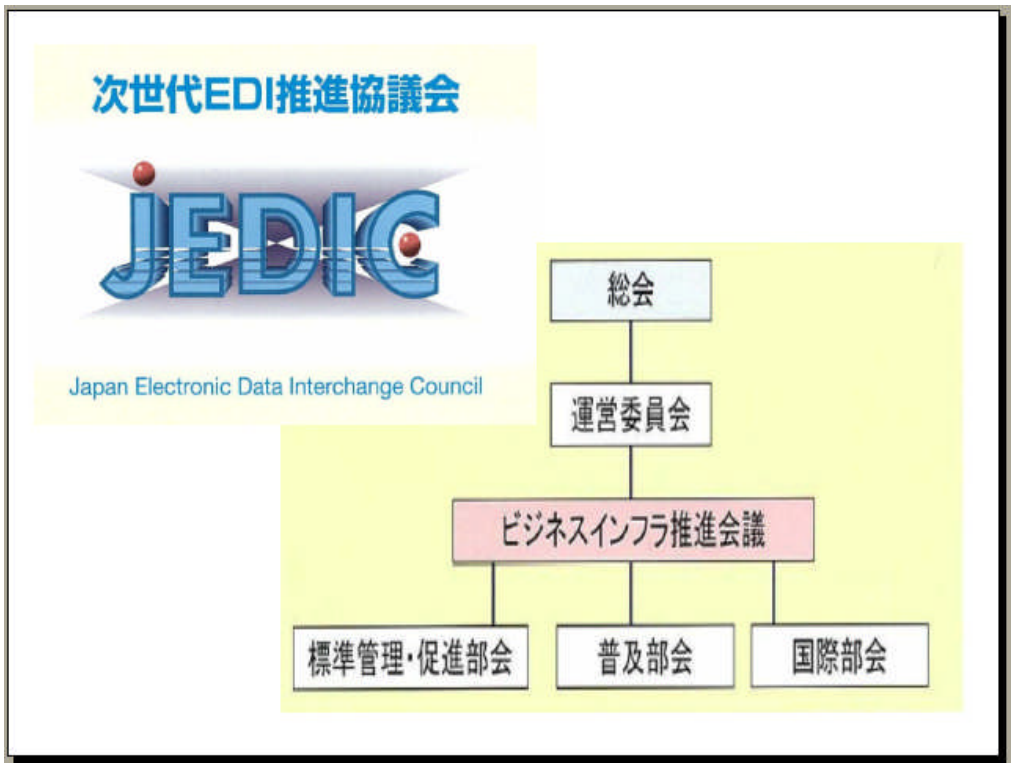

図４．ＪＥＤＩＣ組織図

6 ビジネスインフラ研究会：平成 20 年 11 月に経済産業省 商務情報政策局 情報経済課が中心になり発足。
7 下請代金支払遅延等防止法：下請代金の支払いを適切に確保するための法律。
8 素形材産業取引ガイドライン：平成 20 年 12 月の改定で EDI のソフトウェアーや端末の導入依頼の禁止を追加。

４．ＥＤＩにおける課題

４．１　自動車部品業界の課題

自動車部品業界は、自動車メーカーを中心としたピラミッド構造になっている。自動車メーカーから１次仕入先には、各自動車メーカーに特化したＥＤＩで受発注取引が行われている。しかし、１次仕入先から２次仕入先は、各社異なった独自方式になっている。

特に、最近はインターネットを利用した独自Ｗｅｂ－ＥＤＩでの受発注取引が拡大している。一方、２次、３次の仕入先は紙・メール・ＦＡＸが主流となっており、ＥＤＩの整備ができていない（図５．参照）。

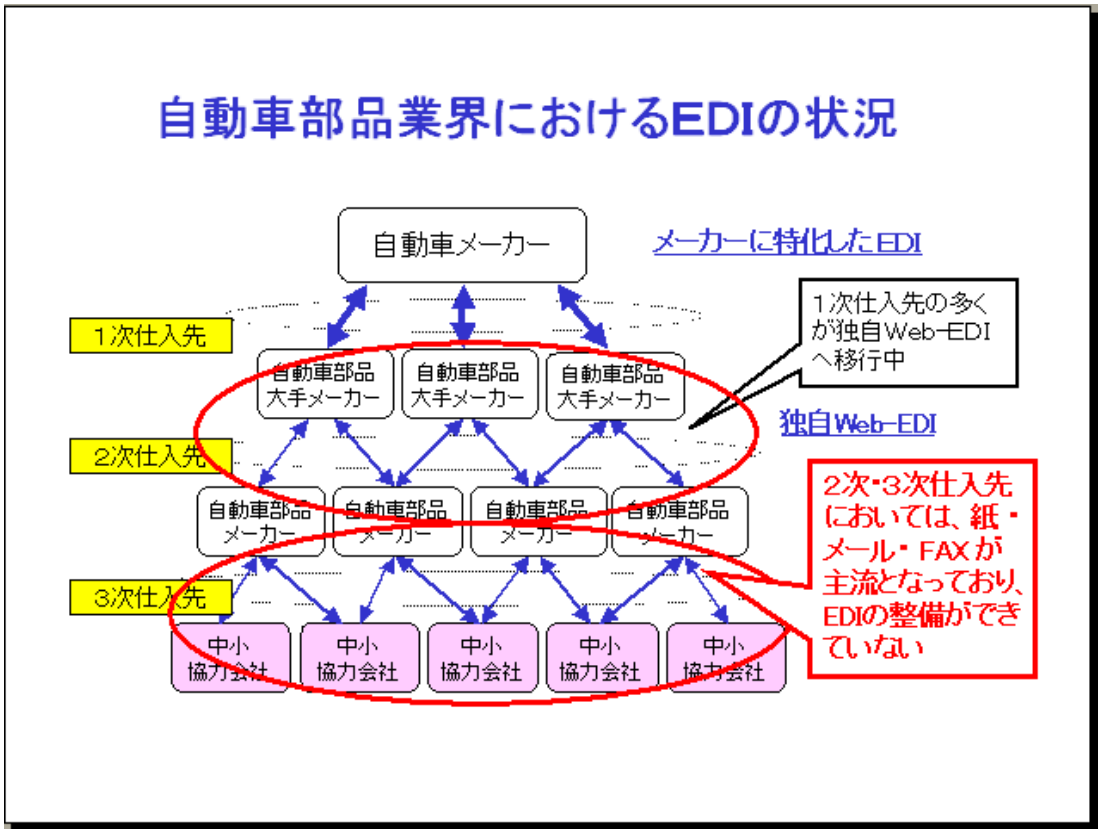

図５．ピラミッド構造（独自ＥＤＩ）

１次仕入先（発注先）から２次仕入先（受注先）で実施している独自Ｗｅｂ－ＥＤＩによる受発注取引は、インターネットを利用した非常に先進的なシステムのように見えるが、仕入先にとっては 非常に不効率なシステムとなっている。２次仕入先（受注先）は、インターネットを利用して１次仕入先（発注先）が指定したサイトに入り、各社異なったパスワード・画面・操作の処理を実施している。２次仕入先（受注先）では、場合によっては、１人の担当者が１０社～３０社の異なった画面の操作を行っている。

また、その独自Ｗｅｂ－ＥＤＩは、１次仕入先（発注先）の都合の良いように作られており、２次仕入先（受注先）には入力データの保管ができず、企業間のシステム連携ができない場合が多い。

＊自動車部品業界の課題まとめ＊

＜課題１＞ ＥＤＩの業界標準がない。

→ 対策１：業界標準のシステムを構築する。

＜課題２＞ 各社異なった方式で多端末・多画面となっている。

→ 対策２：共通画面・共通操作が可能なシステムを構築する。

＜課題３＞ 企業間のシステム連携ができていない。

→ 対策３：企業間のシステム連携が可能なシステムを構築する。

４．２ 中小企業の課題

中小企業においては、今も「手書伝票」を利用している（図６．参照）。自動車業界に限らず、すべての業界において同様となっており、ＥＤＩの整備ができていない。特に、従業員が５名～１０名の小規模企業においては、数十万円～数百万円のシステム導入は困難であり、コンピューター処理は殆どされていない。なぜなら、中小企業の多くは業界団体に所属しておらず、業界標準の動きから取り残されている。

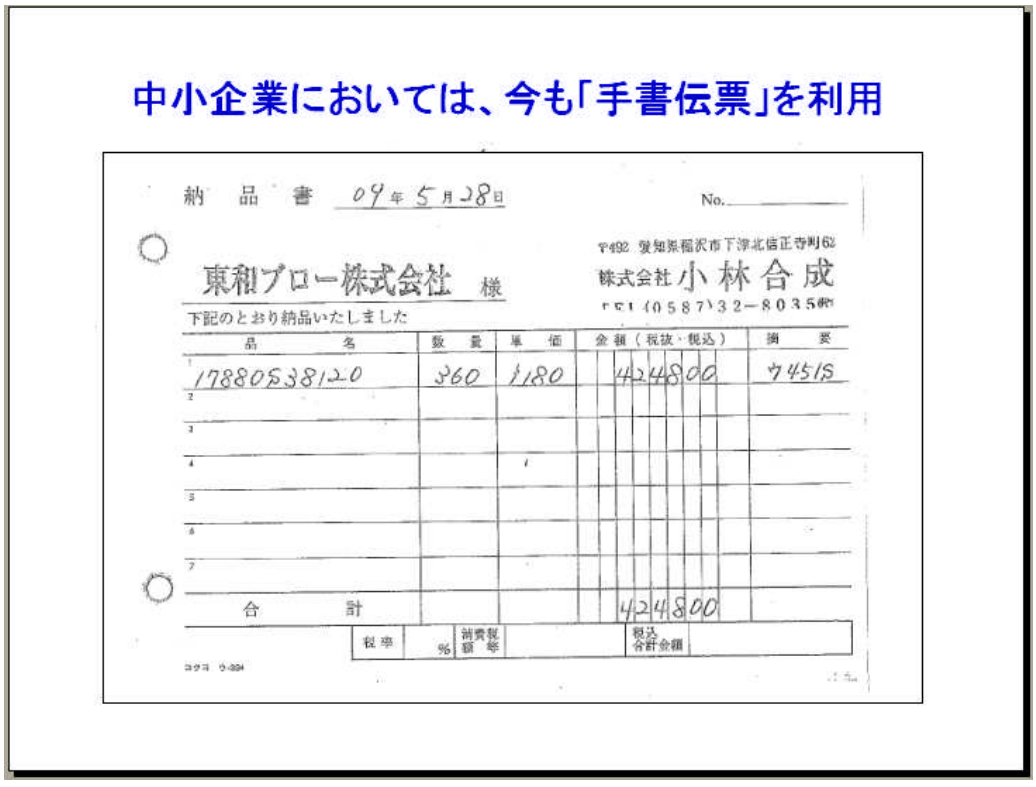

図６．手書伝票

４．３　共通ＥＤＩの実証実験

自動車部品業界、及び中小企業における各課題を解決するために、共通ＥＤＩの実証実験を行った。実証実験は、経済産業省中小企業庁の「戦略的ＩＴ促進事業」[9]に応募して、補助金事業として実施した。自動車部品業界、及び中小企業の各課題から、共通ＥＤＩの要件を下記の要件１から要件４とした。

要件１：業界標準、及び中小企業標準となるシステム
要件２：各社共通の画面・操作が可能なシステム
要件３：企業間のシステム連携が可能なシステム
要件４：基本機能はオープン（無料）となるシステム

自動車部品業界には、標準となるＥＤＩはないため、ＣＯＸＥＣ[10]が構築した電子部品業界用のＥＤＩを利用して、実証実験を行った。ＣＯＸＥＣとしても他業界での利用拡大を希望していた。

４．４　実証実験の結果と解決策

実証実験を行った結果、ＣＯＸＥＣのＥＤＩは自動車部品業界、及び中小企業では利用できないことが分かった。自動車部品業界では通常発生する複数回処理ができない点や、自動車部品業界用にプログラムを修正するのに予想以上に費用が掛かる点があり、自動車部品業界、及び中小企業では利用できない。

自動車部品業界では通常発生する複数回処理とは？

自動車業界で行う複数回処理とは１日のうちに同じ納入業務を行い納入する処理である

ＣＯＸＥＣ開発のＥＤＩは、発注単位ごとに単価・数量・納期など決めて発注を掛ける仕組みであったために、同じ日にちで複数回の発注を同じ場所から掛けたり同じ場所への複数回の納入が掛けれるようになっていなかった。

プログラムの修正内容

自動車業界独自のかんばん発注システムへ対応
複数回納入対応
内示注文対応
実証実験用修正カスタマイズ

通常修正費用　　１２０万円
カスタマイズ費用　２５０万円　　総額３７０万円

9 戦略的ＩＴ促進事業：平成 19 年度の EDI システムの普及促進を図ることを目的とした補助金事業。
10 ＣＯＸＥＣ：共通 XML/EDI 実用化推進協議会。

＜実証実験期間、及び実施規模＞

２００８年1月７日～２月２９日　発注先：１社（中規模企業）、受注先：１社（小規模企業）

＜実証実験結果＞

要件１：業界標準、及び中小企業標準となるシステム　→　×：
自動車部品業界では利用が不可

要件２：各社共通の画面・操作が可能なシステム　→　○

要件３：企業間のシステム連携が可能なシステム　→　○

要件４：基本機能はオープン（無料）となるシステム　→　×：
修正費用１２０万円発生

＜解決策＞　要件１：×、要件４：× から

要件１ → 解決策１：他業界でも利用が可能な「新たなＥＤＩ」の構築をする。

要件４ → 解決策２：中小企業でも利用が可能な安価なシステムの構築をする。

５．アジャイル開発の適用と考慮事項

５．１　業界共通システムの対応

自動車部品業界の専用ＥＤＩではなく、他業界、及び中小企業においても利用が可能な業界共通ＥＤＩを構築する。通常は、業界標準は各業界団体が決めてきている。ＥＤＩについても、過去に何度も業界標準は検討していた。しかし、各企業の要望を取り入れると、仕様が膨れ上がり真の標準化が進んでいないのが実態となっている。また、中小企業においては、業界団体に所属していないため、業界標準の動きから取り残されている。そこで、今回はユーザー企業である当社が、中小企業の標準を検討し、実際に運用して、業界標準、及び中小企業標準を提案する。

まず、他業界については業界共通ＥＤＩ基盤をＣＯＸＥＣと連携する。次に、自動車部品業界の団体であるＪＡＰＩＡ[11]（自動車部品工業会）へ提案をする。中小企業への展開[12]は、日本商工会議所、及びＩＴコーディネータ協会[13]に協力をお願いする。そして、日本の中小企業２４０万社を対象に展開して、日本におけるＩＴの底上げを実現する（図７．参照）。

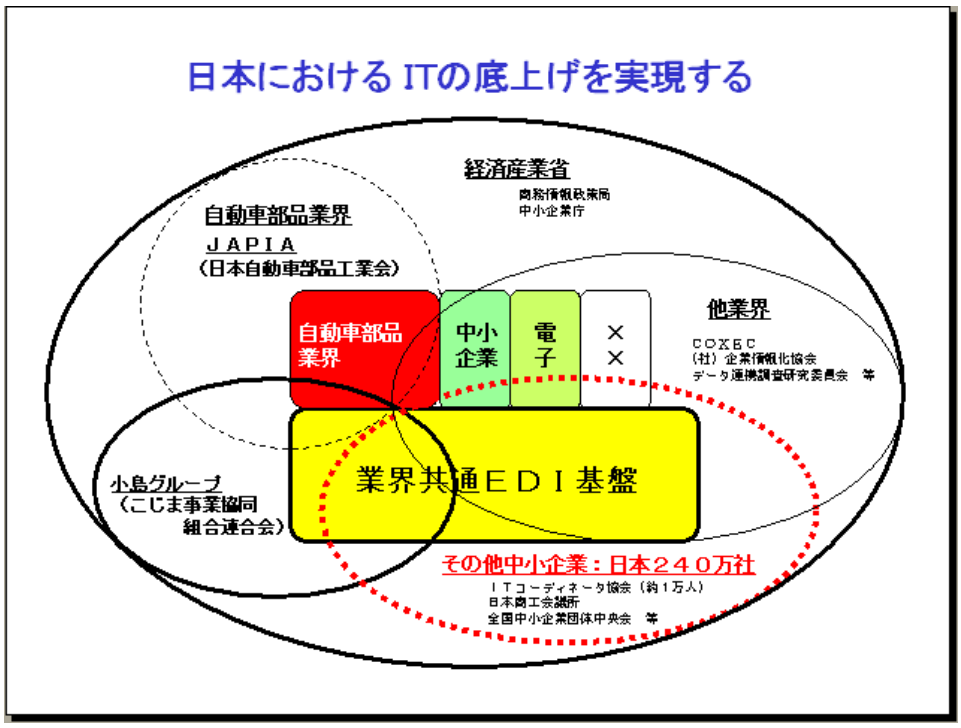

図７．全体構想図

---

[11] ＪＡＰＩＡ：Japan Auto Parts Industries Association 社団法人日本自動車部品工業会（部工会）。

[12] 中小企業への展開：2009 年 5 月 物流ＩＴソリューションフェア（東京ビックサイト）出展。
豊田商工会議所主催 2009 年 12 月とよたビジネスフェア（スカイホール豊田）出展。
岡山県中小企業団体中央会（会員：900 団体、5 万社）において 2010 年 2 月説明。

[13] ＩＴコーディネータ協会：特定非営利活動法人　http://www.itc.or.jp/index.html。

５．２　アジャイル開発における利用技術と進め方

５．２．１　アジャイル開発における利用技術

アジャイル開発における利用技術は、下記の技術を適用した（図８．参照）。

図８．アジャイル開発における利用技術

５．２．２　アジャイル開発の手法と進め方

今回採用したアジャイル開発プロセスは、「スクラム(Scrum)」を採用した。「スクラム(Scrum)」は、もともと竹中弘高氏と野中郁次郎氏の製品開発プロセスの研究に基づいている。

- スクラム（Scrum）

  決めた開発期間ごとに動作可能な製品を作成するスプリント(Sprint)、毎日行う短時間のミーティングなどを特長とした、プロジェクト管理を中心とした方法である。

２週間の集中開発期間(スプリント)と、スプリントを計画する 1 日のバックログ管理、レビュー、組織化作業を繰り返した。

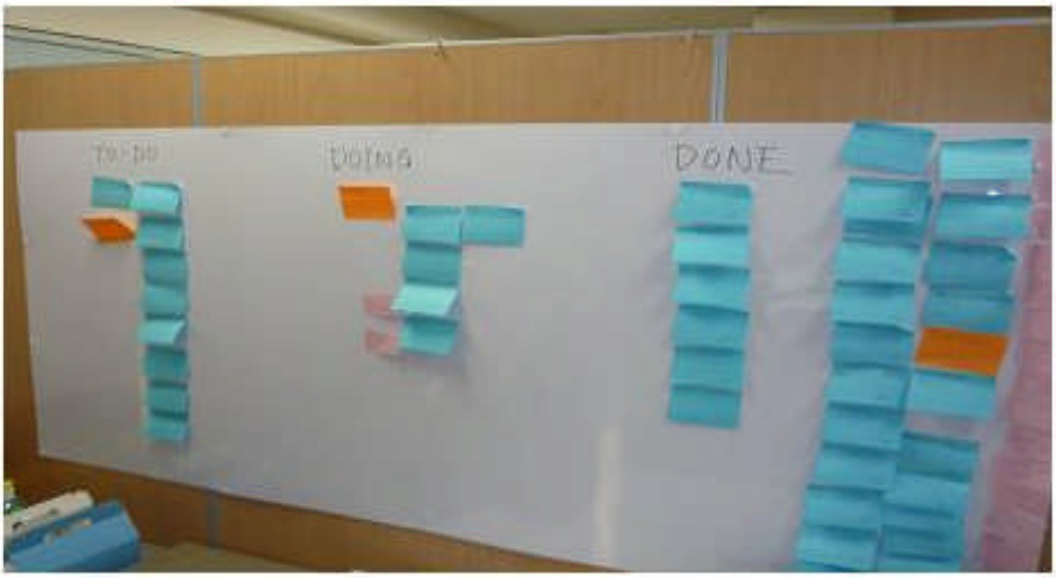

図９．タクトかんばん

アジャイル開発を進めるにおいて、開発チーム全員で問題点・慎重状況の見える化を実施した。タスクかんばん方式で、ＴＯ－ＤＯ，ＤＯＩＮＧ，ＤＯＮＥ にわけ、始めに自分が出来るものを選び、残ったものを全員で解決していく（図９．参照）。全員のレベルが近づいた所で，進捗トラッキングへ切り替え，ＷＥＢ上で進捗が確認できる方法へ切り替えた（図 10．参照）。

図１０．進捗トラッキング

５．３　アジャイル開発における考慮事項

アジャイル開発では，開発ベンダーとユーザーがお互いに、最良の業務プロセスへ近づけるために活動し続ける。そのために、詳細な仕様はプログラムの動作確認を行いながら進める。ＩＴベンダーは、此れまでとは異なる見積り方法や開発者のスキルが必要となる（図１１．参照）。

その中でも最も重要となるのがプロジェクトリーダーの資質である。プロジェクトリーダーには、下記の資質が必要となる。

・モチベーションを持ち続ける
・技術力（向上心）
・協調できる人
・プロジェクト全体の状況を見渡せる感受性

開発チームには、理想とするソースコードの共通認識が無いと、スキルのある人に偏った開発になり、開発工期を延ばす結果になる。（スプリント管理）

アジャイル開発では、たくさんの仕様書を書くことよりも、プロジェクト関係者間で必要な時に即座に直接顔を合わせて意思疎通を行うことが必要となる。

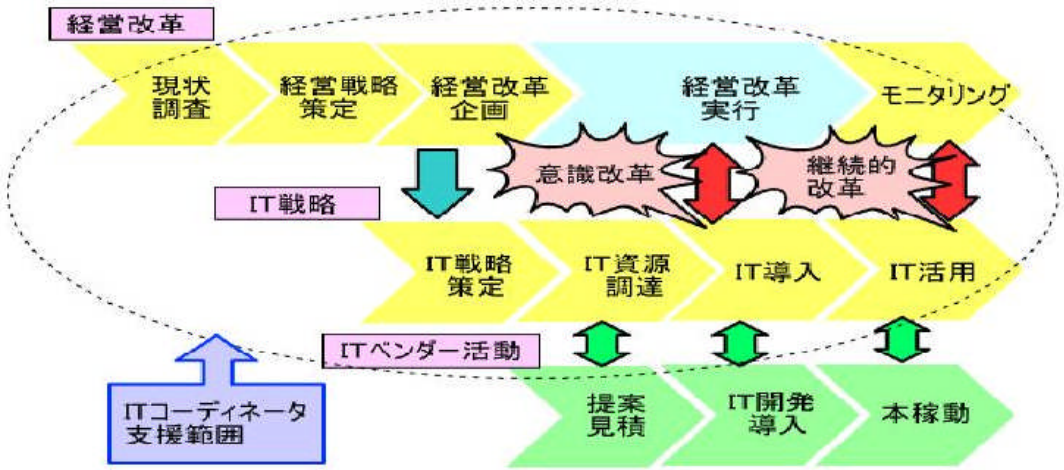

図１１．アジャイル開発と経営改革

＜アジャイル開発において発生した問題点　①＞　（開発メンバー同士の確執）

初めてのアジャイル開発を進める工夫として、開発者同士でソースコードのレビューを行いより良い開発につなげようとした所、感情論に発展しチーム全体に不協和音が広がってしまった。これは、チーム全体で理想のソースコードの認識が無かったために、おこったことと考えられる。

そこで取った解決策は、感情論を持ち込まない様に、ソースコードのレビューはテストツールを使い機械的に問題の在るソースコードを抽出し、開発リーダーが主導権を取り

コードのリファクタアリングの改善をチーム全体で行った。そのおかげでチーム全体で理想とするソースコードの共通認識が進んだ。

＜アジャイル開発において発生した問題点　②＞　（業務内容の理解不足）

アジャイル開発の特徴は、ユーザー部門とのコミュニケーションを密に行う。イテレーション単位ごとに、ユーザー部門に対しデモを行いフィードバックを受け、開発を進めていく。開発担当者が業務の内容を理解して進めてもらうため、ユーザー部門も業務内容の詳細を的確に説明できる人選をしないといけない。

業務内容の知識不足で、ユーザー部門への調整もうまくいかず短時間での開発にも支障をきたすことになってくる。

５．４　ＳａａＳ型業界共通ＸＭＬ／ＥＤＩの構築

５．４．１　システムの構造

「新たなＥＤＩ」の基本構造は、業界共通ＥＤＩ基盤をベースに、機能追加が可能な「モジュール構造」とする。この構造により、自動車部品業界だけではなく他の業界も利用が可能となる。したがって、自動車部品業界においても、特殊な発注方式はモジュールとして追加が可能となる。特に中小企業においては、最小な機能を基本部として、追加機能はオプションとする（図１２．参照）。

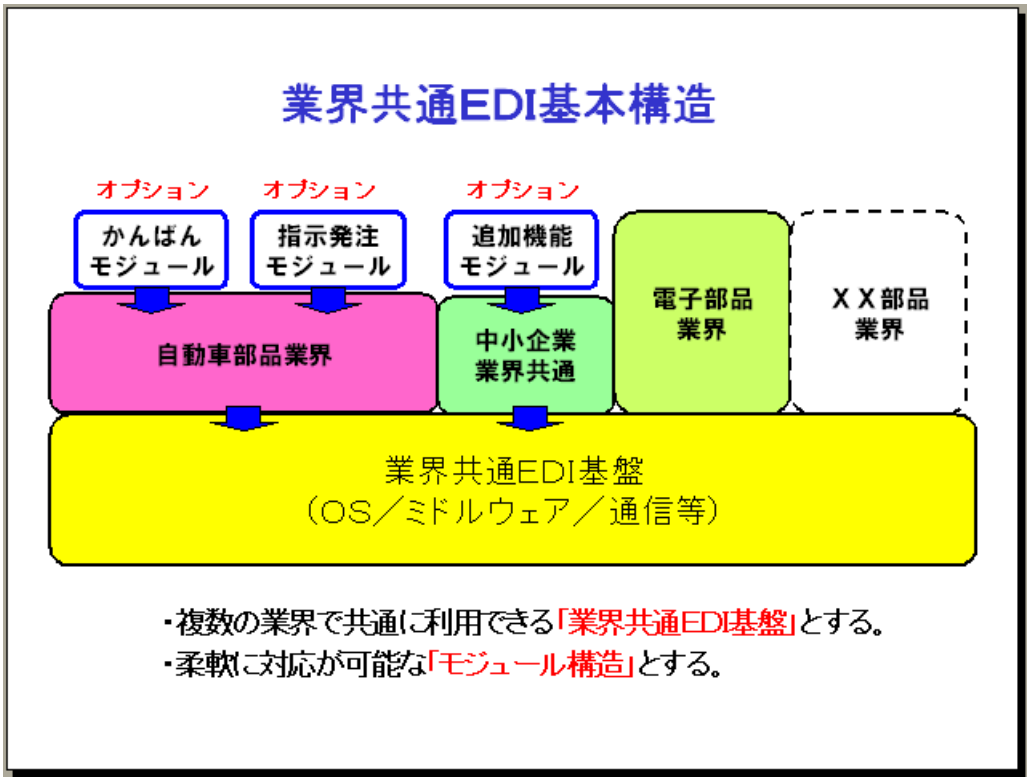

図１２．業界共通ＥＤＩ基本構造

５．４．２　システムの特徴

「新たなＥＤＩ」は、ＸＭＬ、及び「Ｓａａ Ｓ」に対応した企業間の連携が可能な「Ｓａａ Ｓ型業界共通ＸＭＬ／ＥＤＩ」とした（図１３．参照）。この「Ｓａａ Ｓ型業界共通ＸＭＬ／ＥＤＩ」は、他業界でも利用が可能で安価なシステムとなった。更に、「Ｓａａ Ｓ型業界共通ＸＭＬ／ＥＤＩ」をベースとした「受発注取引システム」を構築した。

このシステムは各企業の個別サーバーに導入しても良いが、中小企業が導入しやすいように「Ｓａａ Ｓ」とした。自社にどうしてもサーバーを設置したい場合は、それも可能となっている。基本は「Ｓａａ Ｓ」とし、外部のデータセンターを利用する。基本ソフトの利用料はオープン（無料）とする。ただし、データセンター利用料は、データ量により課金される。

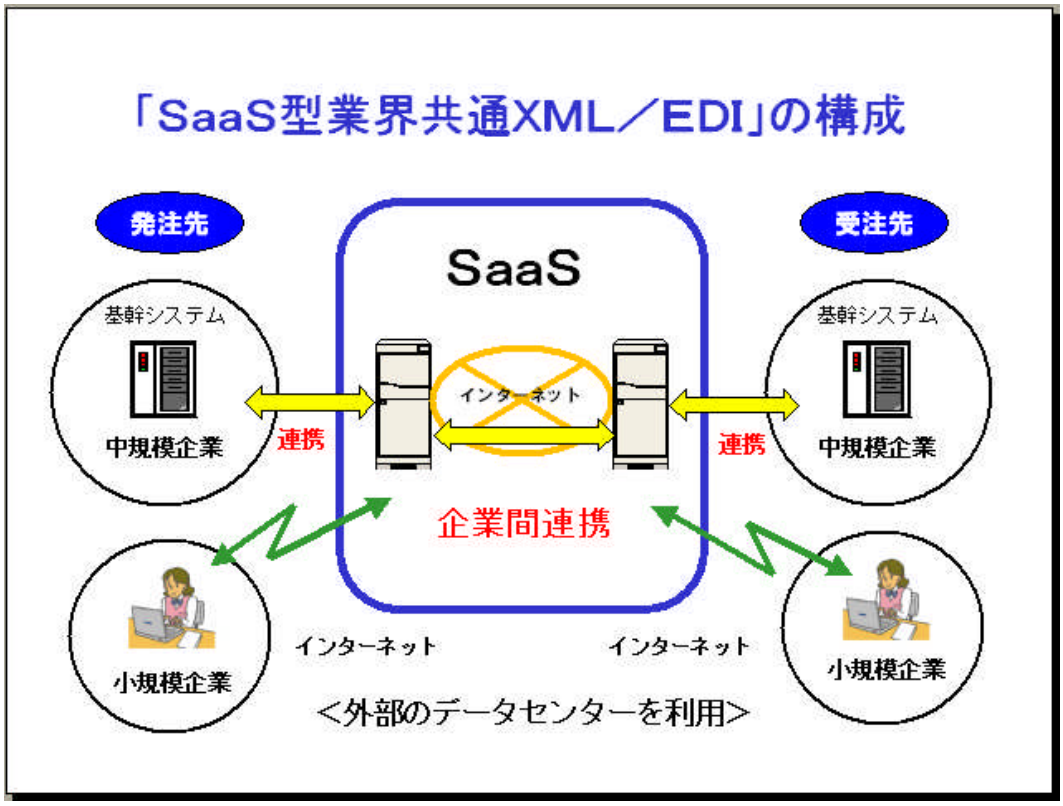

図１３．Ｓａａ Ｓ型業界共通ＥＤＩ構成

６．アジャイル開発の評価

＜効果＞

受発注取引システム稼動可能アプリケーション

開発期間　　３年　　→　　１年　　（１／３低減）

今回開発した「業界共通ＸＭＬ／ＥＤＩ」の開発は、アジャイル開発に向いていた。
特に、下記の条件の場合、ウォーターフォール開発ではなく、アジャイル開発が適している。

・開発要件が固まっていない
・納期が短い
・頻繁な要件変更がある

上記の条件の場合、ウォータホール開発のように要件をすべて洗い出し、その後要件定義を行い詳細設計を行う手順を踏んでいては何年かかっても要件定義すら完成できない。

今回は、アジャイル開発で要件定義を最小に作成し、業務内での要件とすり合せを行いながら開発を進め、多々の問題を一点ずつクリアーしていき、無事開発を終了した。
実証実験として、発注業務において台帳管理による手作業（発注者側）納入業務による納品書・受領書手書き伝票の作成（受注者側）など業務の改善に寄与した。

＜実績＞　　４０社稼動　　（図１４．参照）

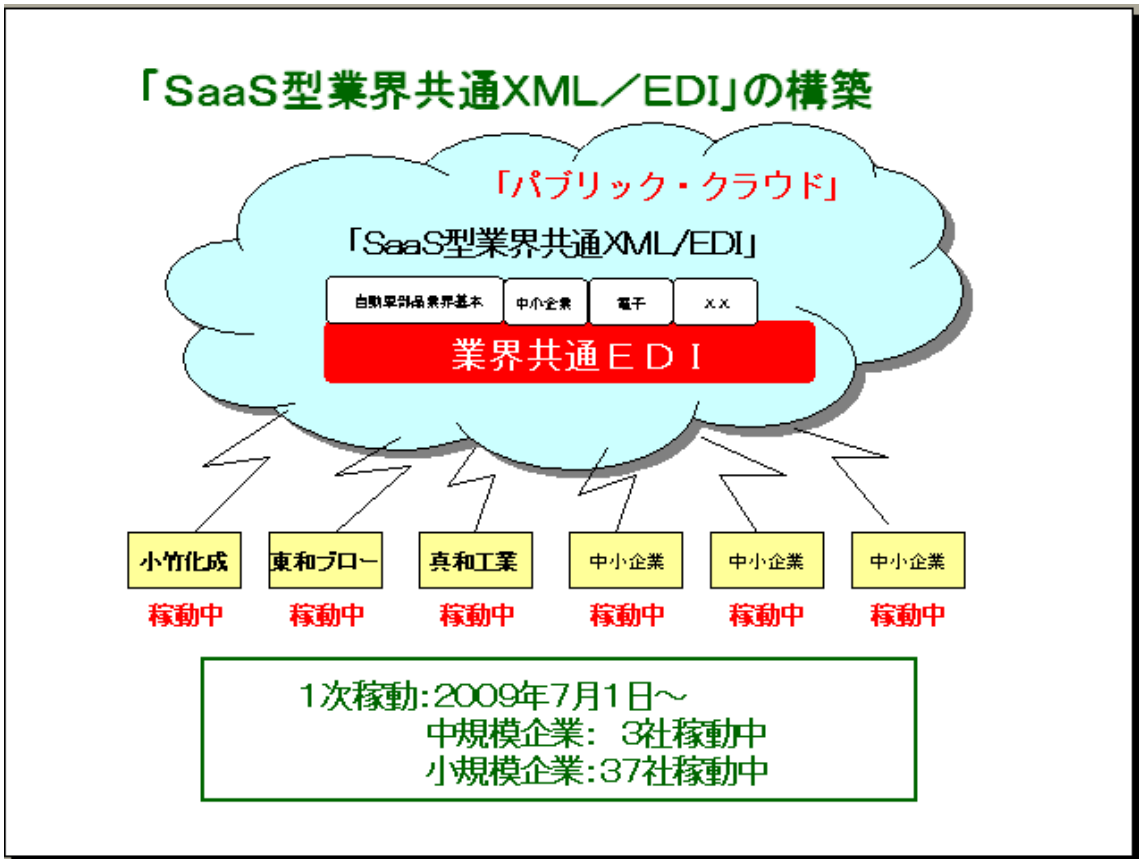

図１４．現状稼動構成

７．今後の課題

７．１　アジャイル開発

・契約の課題

アジャイル開発では開発途中で重要な要求が発見されると、そちらの要求を優先させる。開発を進める中で、要求がより効果的な方向に変更していく。したがって、初期要件と納品される内容が変わってしまう。たとえば最初の要求が固定された契約を行うと、アジャイル開発は実行できない。初期要件よりも良い要求機能が実現できても、納品物は要件を満たしていないことになってしまうためだ。したがって、アジャイル開発を行うときには、期間契約で短期更新いう形態が１つの方法となる。

・アジャイルの間違った認識

アジャイル開発での失敗を聞くことがあるが、企業のＩＴ戦略やビジネス戦略が十分に認知されていないことにある場合が多い。開発チームは、ビジネス戦略やＩＴ戦略を明確に理解し、それが開発チームにどのような影響を及ぼすかを理解することが必要となる。

アジャイル開発手法では、実際に動くソフトウェアこそが最重要なプロジェクト進行尺度であることを、強調する。 この実際に動くソフトウェアという進行尺度の採用と、直接顔を合わせた意思疎通の重視とがあいまって、アジャイル開発手法で作成する文書の量は、他の開発手法と比較すると、非常に少ない。この少ない文書化については、無統制で雑な作業（ハッキング、カウボーイコーディング）であるとして、アジャイル開発に対する批判材料の一つとなっている 。

７．２　ＳａａＳ型業界共通ＸＭＬ／ＥＤＩ

・ＳａａＳ型を広める上での課題

インターネットとサーバという新たな要素が加わることで、スタンドアロン型システムに比べてサービス中断の可能性が格段に高まる。まず、サーバがダウンすると、当然サービスを利用できなくなる。ダウンする原因として、次の要因が考えられる。

- メンテナンス
- 機器故障
- プログラムのバグや設定ミス
- サーバへの不正アクセスによる業務停止

メンテナンスは通常計画的に行われるもので、深夜や早朝など利用が少ない時間帯に行われることが多い。メンテナンス以外の事例は突然発生する上、解決までにどれくらいの時間がかかるか分からない。

さらに、ネットワークシステム特有の難しさがある。アクセスが集中してシステムに負荷がかかる場合や、複数のアクセスが微妙なタイミングで発生する場合、といったシステム設計上の問題だ。

このようにネットワーク型システムはスタンドアロン型システムよりサービスを利用できなくなる可能性が高い上に、原因特定が難しい。

８．おわりに

最近ではＩＴによる経営の効率化という言葉をよく耳にする。中小企業は今まで、ＩＴ投資を、ほとんどおこなっていなかったために、受発注業務でのコンピューター処理がない。したがって、紙・メール・ＦＡＸが主流となっている。しかし、最近になってコンピューター本体の低コスト化で中小企業にもＩＴインフラが整備進んできた。それでもＩＴ化が進まないのでしょうか？これは、コンピューター上で動かすアプリケーションが高額で中小企業での導入コストが高すぎることに問題がある。

アジャイル開発は、あたかもシステムを新入社員のように一から育てていき一人前にする事に良く似ている。失敗・改善を加えて完成させていく開発手法だといえる。

開発範囲を最小限にし、繰り返しユーザーとの対話の中で開発を進めるアジャイル開発は、今後のシステム開発の主な手法となる。最後に、今回の事例が皆様に役に立つ事例となることを願う。