

○渡邊浩一郎（IBMビジネスコンサルティングサービス），
桑嶋健一（筑波大ビジネス科学研）

1. はじめに

パッケージ・ソフトウェアの製品開発マネジメントについては、Cusumano(1995, 2004)をはじめとした一連の研究がある。これら既存研究は、Windows OS, Microsoft Office といった大規模アプリケーションや OS を分析対象としている。こうした製品群では、製品を構成するモジュールが多く、相互が複雑に機能するため製品を完成させること自体が難しい。このため、製品開発の成功の可否は、製品を完成させることであるととらえられてきた。しかし、パッケージ・ソフトウェアには、一般消費者を対象とした製品群がある。この製品群（一般向けパッケージ・ソフトウェア）は、OS や統合型ビジネスソフトに比べれば製品規模が小さく、製品のモジュール間の連携も単純であることが多いため、製品を完成させることは大規模アプリケーションほど難しくはない。一方、製品のリリースもしくはバージョンアップは一定期間毎（多くの製品では一年に一度）に繰り返されるため、限られた期間の中で、より多くの機能拡張をしつつ、より高品質な製品を開発することが、この製品群の成功の可否を左右する。

既存研究ではこの製品分野については十分検討されているとは言いがたい。そこで本研究では、日本における成功事例分析を通じて、一般向けパッケージ・ソフトウェア開発の成功要因を明らかにすることを目的とする。

2. 製品開発論におけるソフトウェア開発についての先行研究

ソフトウェア全般を主に経営学の製品開発論的な観点で取り扱う分野の一つとして、開発プロセスにおけるマネジメント、組織パターンとパフォーマンスとの関係を実際の企業活動の観点から分析し、効果的な製品開発パターンを明らかにするプロセス・アプローチがある。過去の研究でも、ソフトウェア分野においてもプロセス・アプローチによる分析が試みられており、今回の研究はこの視点を起点として研究を進めた。ソフトウェア開発は一般に、要件定義・概要設計・詳細設計・コーディング・結合テストの一連の流れにより

構成される。ソフトウェア開発におけるプロセス・アプローチの基本的な類型には、ウォーターフォール型およびその改良版であるスパイラル型(Boehm, 1988, Brooks, 1995)がある。また、Cusumano(1991)は、Microsoft 社におけるソフトウェア開発の事例を通じて、シンクアンドスタビライズ型が製品開発パフォーマンスに影響を与えると指摘している。

また、Cusumano(2004)では、過去のソフトウェア開発研究事例の分析経験から、ソフトウェア製品一般に当てはまるとされる成功要因として、①製品に求められる機能・品質にあった開発類型の選択、②イノベーションと設計戦略、③アーキテクチャ戦略、④組織・チーム管理、⑤プロジェクトリソースの管理、⑥テストおよび品質改善活動をあげている。これらの成功要因は既存のソフトウェア研究の総論でもあり、ソフトウェア製品開発の既存研究で言われている成功要因の多くを含んでいる。代表的なものとして③については Boldwin & Clark(2000)、Iansiti(1998)、④については妹尾(2001)がある。

これまでの研究では、分析の中心が基盤ソフトウェア(OS)や業務アプリケーションにおかれ、一般向けパッケージ・ソフトウェアについてはあまり言及されてこなかった。そこで、この研究では一般パッケージ・ソフトウェアの製品開発に着目した(表 1)。

表 1 本研究の位置づけ

カスタマイズソフトウェア	直接配布	業務用	基幹業務アプリケーション	Cusumano, Iansitiら の一連の研究
パッケージソフトウェア	パッケージ (箱売り)	基本ソフト	Windows 95/NT/2000	本研究
		業務用	Oracle, SQL Server等	
		一般向け	多くの市販アプリケーション	
脱パッケージソフトウェア	オンライン (ネットワーク)	基本ソフト	Linux	佐々木・北山(2000) 藤田・生稲(2005)
		業務用	Apache, BIND, Sendmail	
		一般向け	シェアウェア、フリーウェア	

3. 一般向けパッケージ・ソフトウェアの製品特性

製品開発の観点から、ソフトウェア全般の製品特性を分析すると、生産工程を持たず(生産技術に関する考慮が不

要)、多人数による労働集約型作業であるという特徴がある。実際の開発でも開発スケジュールやプロセス管理が重要視されてきた。

一般向けパッケージ・ソフトウェアが市場での成功を実現するためには、新バージョンの製品は、例えば年末商戦等の何らかの販売促進タイミングを開発目標日として定められるため、開発目標日までに予定した機能を実装し、製品としてリリース出来る品質で完成されなければならない。製品機能の面では、変化の激しい外的環境(インターネット技術、利用者の嗜好の変化)への順応のために多数の追加機能が必要である。そして、一般消費者向けの製品であることから、バージョン毎に増加する製品機能を統合された一つのソフトウェアとして実現することが要求される。

本研究では、これらの特徴を念頭に置き、プロセス・アプローチとソフトウェア製品一般に当てはまるとされる成功要因(Cusumano, 2004)の観点からケースを分析することにより一般向けパッケージ・ソフトウェア固有の成功要因を検討した。

4. 事例分析:A 社 Web 開発ツール 製品X

4-1 ケース選定の理由

今回取り上げるのは A 社の Web 開発ツール製品Xの 1995 年から 2002 年までの開発事例である。製品Xは主に国内市場向けに独自に開発された製品であるが、同時期、競合する海外製のソフトウェアが多数ある中、国内ではトップシェアを維持しており、A 社でも成功した製品であると位置づけられている。製品Xはこの期間、計8回にわたる定期的なバージョンアップを行うことに成功し、当時インターネットエクスプローラやネットスケープ等により飛躍的に発展したブラウザ技術の変化への対応、動画編集機能、携帯端末への対応等、利用者から求められる機能をいち早く製品に追加してきたという実績がある。以上の点から、製品Xは一般向けパッケージ・ソフトウェアとして成功した製品といえることができる。

ケースの分析に当たり、長期にわたる一連の製品開発プロジェクトをバージョン毎に個々のプロジェクトとして捉えた。この事例は製品開発時期により初期、中期、後期に分類することができた。

調査方法としては、各開発時期の開発マネージャへのインタビューと製品開発計画書、製品発表レターなどの資料調査を実施した。製品開発の成功要因としては、開発リード

タイムの短縮と製品の統合化がどのように実現されているかを中心に分析した。

4-2 製品Xの概要

① 初期における製品開発の概要

製品Xは当初はWebデザイナー向けのツールとして、UNIX WS 上で動作するソフトウェアとして開発された。開発が開始された 1995 年前後は、インターネットが一般家庭に普及する直前であり、Web ページ作成の需要が多く見込まれたが、当時はそれをテキストエディタとよばれる文書編集ソフトウェアを使用してHTMLと呼ばれる表記言語で直接記述するという方法が多く用いられていた。この方法で思い通りのWebページを作成するにはHTMLに対する高度な知識と熟練が要求されるため、技術者以外が本格的なWebページを作成することは困難であった。そこで製品Xは、技術にあまり強くないデザイナーでも使えるツールというコンセプトのもと、当時普及し始めたGUI(グラフィカル・ユーザー・インタフェース)を用い、Webページを視覚的に作成できるソフトウェアとして開発が進められた。

開発組織はUNIX WSのソフトウェア開発の経験があったため、製品の基本構成や開発のアプローチはUNIX WSのソフトウェア開発でものをベースとして始められた。具体的には開発に当たっては設計から開発までを順列的にながすという、A社が大規模アプリケーションの開発経験で築きあげた開発アプローチが採用された。また、プログラム内部ではC++言語の使用、モジュール化された内部構造など、当時の最新の開発技術が用いられた。これにより、特定OSに依存するモジュールと汎用的なモジュールを明確に分けることができ、別のOSで動作させるためには、特定OSに依存するモジュールを書き換えるだけですむ。初期の製品では、複数OS版の製品リリースを実現することが出来た。ただしビルド(ソースコードから実行プログラムを生成する作業)についてはOS毎別々に手作業で生成しなくてはならなかった。このためビルドは3日から1週間に一度といった間隔でしか実施出来なかった。

② 中期における製品開発の概要

中期では、一般向けOSとしてWindows OSの普及が進み、動画や音声機能といった技術的面で他のOSとの差が開いたことから、製品を単一プラットフォーム(Windows OS)に集約することが決定された。開発環境もMicrosoft社の提供する

製品に統一され、これにともないプログラム全体の再構築が行われた。具体的には、これまでのすべてのコードを破棄し、Microsoft社の言語製品であるVisual C++と開発ライブラリ(MFC)を用いた完全な書き直しが行われた。Windows OSのバージョンアップ毎に更新される新しいGUIデザインへの対応や、ドラッグアンドドロップ、OLEに代表されるWindows独自のテクノロジーをスムーズにサポートするにはMicrosoft社の開発技術が不可欠であったためである。ソースコード管理についても、Microsoft社の商用ソースコード管理ソフトウェアであるVisual SourceSafeが採用された。この製品には、ビルド作業の自動化を支援する機能があり、ほぼ毎日のペースでのビルドができるようになった。

③ 後期における製品開発の概要

後期では、ユーザビリティ研究部門の支援を得てUCD (User Centered Design)と呼ばれるユーザビリティ(使いやすさの評価)テストを製品開発プロセスに積極的に取り込むアプローチがおこなわれた。この製品はGUIを重視した製品であることから、製品企画当初から利用者の嗜好をどう製品に組み込むかに取り組んできた。初期・中期でもユーザビリティテストは開発者の身内等を用い最終工程で行われていたが、この工程では製品がすでにある程度完成しており、全体のデザイン設計に影響したり、モジュール構成を変更するような修正は次バージョンに持ち越しとなった。また、バージョンを経る毎に製品機能が増大したが、開発は機能単位で平行して進められるため、操作の統一性を保つことも困難であった。その他、ユーザー登録はがきからのフィードバックの積極的な取り込みも試みられてきたが、即時性に乏しく情報量も不足しがちで現行バージョンへの反映は難しかった。

そこで後期では、開発の設計段階からユーザビリティテストを積極的に取り込むアプローチが取られた。ユーザーはがき等から得られた製品利用者層を中心に、開発の設計段階では前バージョンを用いた課題抽出、コーディング段階ではプロトタイプ(試作品)を用いた設計評価、テスト段階では従来通りのテストという形で、開発全体にユーザビリティテストが取り込まれた。一方、後期では製品開発プロセスや開発組織についての変動はほとんど無かった。後期においても製品は一定レベルの成功を維持し続けていることから、後期では製品開発サイクルや組織構成については一種の完成型に至ったものと考えられる。

4-3 成功要因の分析

第一の成功要因は製品開発プロセスとしてシンクアンドスタビライズ型の開発類型を採用したことである。

先行研究では、ソフトウェア開発プロセスの理念形としてシンクアンドスタビライズ型とウォータフォール型が指摘されているが、それぞれの要件や開発類型の定義(何をもって〇〇型と呼ぶのか)については十分明確にされていない(Cusumano, 1995, 2004; 立本 2004)。そこで本研究では、開発プロセスに則して両型の特徴を5つの観点から整理した。そしてこの枠組みに基づいて製品Xの開発プロセスを分析したところ、初期はウォータフォール型に近く、後期はシンクアンドスタビライズ型に近いことがわかった(表2)。

表2 製品Xの各開発時期における開発類型

	初期	中期	後期
要件定義・概要設計			
①設計段階における全体設計の明確化度合い	明確	曖昧	明確
詳細設計			
②詳細設計とコーディングの順序	順列的	同時並行的	同時並行的
プロジェクトサイクルの繰り返し			
③サブプロジェクトの分割数	1	1	2~3
④プロセスの出発の有無	有り	有り	無し
⑤各工程における検証工程の厳密さ	厳密	厳密	厳密
類型	ウォータフォール型	ウォータフォールとシンクアンドスタビライズの中間	シンクアンドスタビライズ型

初期の製品は、コーディング作業で数週間~1ヶ月程度の遅れが発生するなど、製品開発は不確実であり、市場的にも他社製品と同程度のシェアであったという点ではうまくいったとはいえなかった。しかし、後期では開発モデルを変化させることによりこの遅れが解消した。また、製品機能やソースコードの量の面では、初期よりも後期のほうが変動が大きい(=開発量も多い)ことから、リードタイムの短縮の観点で、シンクアンドスタビライズ型開発が有効であったと考えられる(図1)。この点はCusumano & Selby(1995)でも指摘されていたが、今回の一般向けパッケージ・ソフトウェアでも当てはまった。

第二の要因として設計段階からのユーザビリティテスト(使い易さのテスト)の実施があげられる。初期・中期ではユーザー登録はがきを活用したが、現バージョンへの要望の反映は現在開発中の次バージョンの製品では間に合わず、2バージョン先に持ち越されることが多かった。後期では、従来安定化過程で行われていた一般被験者によるユーザビリティテストを設計段階から行った。これにより利用者の嗜好

好を現在開発中の次バージョンから組み込めると同時に、テスト工程の一部前倒し(リードタイムの短縮)ができた。また、早期から製品の統合性について検討されるため、コーディング工程では従来通りモジュール化により分散開発しつつ、ユーザー側からは統合性の高い一つの製品として完成させることができた。この要因について過去の研究ではあまりふれられてこなかったことである。

第三に、開発対象 OS を一つに絞り込んだことである。製品 X では当初、複数の OS 向けに開発されていたが、市場規模と作業負担の観点から、中期に切り替わる時点で開発対象 OS を一つに限定した。これにより、作業負担の低減など開発資源の効率的活用をもたらし、定期的な製品リリースが実現した。加えて、OS 固有の GUI や機能が利用できるため、使い易さの向上や製品機能の強化にも大きく貢献した。開発対象 OS の絞り込みは製品開発の成功には間接的な影響を与えるに過ぎないが、本製品においては正否を分けた一つの要因であった。

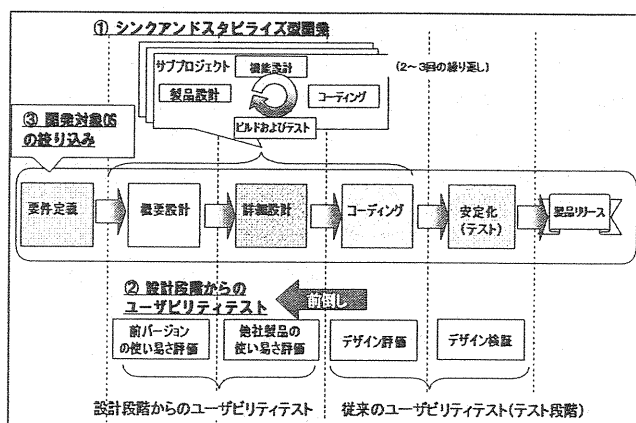


図1 製品Xの開発プロセスと成功要因

この製品では初期段階から、Cusumano (2004) 等で指摘されている、ソフトウェア製品一般に当てはまるとされる成功要因の多くも実践されていた。組織面では開発メンバーはバージョン毎に変わっていたが、Cusumano (2004)で指摘されたリーダーシップの重要性、頻繁なコミュニケーションについては、初期から一貫して行われていた。技術面では初期から Baldwin & Clark, Iansiti が指摘したモジュラー型アーキテクチャが採用されている。今回のケースでは、これらの要因は開発リードタイムや製品統合性の向上という点では製品開発の成否を分ける要因ではなかったといえる。

今回のケースは、バージョン毎に連続した複数のプロジ

ェクトが行われているとみることでもできる。本事例では、開発リーダーを開発経験のあるメンバーから選出することを通じて、青島(1998)で指摘される、複数プロジェクトにおける開発プロジェクトに関する経験則的な文書化が困難な知識の伝承が行われ、製品開発の定期的な継続に寄与していたと考えられる。

5. まとめ

今回の研究の結果、一般向けパッケージ・ソフトウェア開発においても、従来、業務向けアプリケーションで言われていたソフトウェア開発の成功要因のうち、シンクアンドスタビライズ型のような構成変更を前提とした開発タイプの採用、開発対象 OS の絞り込みが有効であることがわかった。これらは主に定期的な製品リリースと同時に高品質な製品の開発の実現に寄与していたものと考えられた。一般向けパッケージ・ソフトウェア製品固有の成功要因としては、設計段階からのユーザビリティテストの実施があげられ、リードタイムの短縮と製品の統合性向上の面で有効であることが明らかとなった。

定期的な製品リリースを通じ、製品開発の成功を持続させる仕組みとして、開発リーダーの持ち回りを通じた製品開発知識のスムーズな伝承も有効であると考えられた。

今回の研究は一般向けパッケージ・ソフトウェアの一つの成功事例をもとに成功要因を提示した。今後は、より多くの事例収集と実態調査、定量分析を通じて詳細な検証をする必要がある。また、近年は一般向けパッケージ・ソフトウェアの中にも、ネットワークを媒体として提供・販売される新しい形態のパッケージ・ソフトウェアが増えており、これらについては今後の研究課題である。

<主要参考文献>

- 青島 矢一 (1998)「製品アーキテクチャと製品開発知識の伝承」『ビジネスレビュー』46(1), 46-59.
- Cusumano, Michael A and Selby, Richard W, (1995), *The Microsoft Secrets*, NY: Free Press 邦訳 マイケル・A・クスmano (1996)『マイクロソフトシークレット』上・下 山岡 洋一 訳 日本経済新聞社
- Cusumano, Michael A (2004), *The Business of Software*, NY: The Free Press 邦訳 マイケル・A・クスmano (2004),『ソフトウェア企業の競争戦略』サイコム・インターナショナル監訳 ダイヤモンド社
- 藤本 隆宏 (2000)「効果的製品開発の論理」、『成功する製品開発』有斐閣所収