

超高速開発に関する当社の取り組み ～超高速開発プラットフォームAZAREA(アザレア)～

ビジネス統括本部
デジタルリノベーションビジネスユニット
デジタルITプロダクト部
サービスプロデューサー



中川 達夫

1. 超高速開発が必要とされる背景

「超高速開発」という言葉が聞かれる様になって久しい。当社も参加している超高速開発コミュニティ(2013年設立)は、現在正会員のユーザー企業103社、ツールベンダー・SIer44社、準会員の企業が36社と、計183社を数えるまでの規模になっている。

企業経営におけるITの役割がますます加速している中、経営環境の変化に応じて情報システムを素早く対応させることは、多くの企業にとって重要な課題である。従来型の属人的で労働集約的なシステム開発のやり方を見直さなければ、企業は競争力を高めることはできない。加えて、システム開発者の人材不足は常態化しており、より効率的で生産性の高いシステム構築手法を導入する必要がある。

それでは、どのようなシステムにおいて超高速開発が必要とされるのだろうか？

ガートナー社では、以前から「ベース・レイヤー戦略」というコンセプトで、企業の情報システムを変更の頻度(ベース)で次の3つのレイヤーに分ける考え方を提唱している。

- ・記録システム(Systems of Record)
 - －変更の頻度が低く、一般的な機能を持ちビジネスの土台となる。
 - －法的規制への対応などもあり、成熟したアプリケーションパッケージ(ERP)が向く。
 - －他社と同じでも構わない。
- ・差別化システム(Systems of Differentiation)
 - －そこそこの変更頻度を持ち、他社との違いを出すために自社に最適なパッケージソフトやSaaSなどを利用する。
 - －他社とは異なるアイデアを実現する。
- ・革新システム(Systems of Innovation)

- －変更の頻度が最も高く、新しいアイデアをもとに、試しに作ってみてはすぐに改修し、トライ&エラーを繰り返す。
- －全くの新しいアイデアを実現する。

超高速開発の適用対象は、この「革新システム」、あるいは自社に最適なパッケージやSaaSが存在しない「差別化システム」の一部である、と筆者は考えている。もちろん、当社のお客様の中には、記録システムの領域までも超高速開発プラットフォームで開発し、自社の業務にフィットさせつつ他社との差別化を図っているお客様も存在するため、一概に断定することはできないが、ITによって自社ならではの差別化を図る部分において、超高速開発が必要とされるケースが多い、ということ間違いはない。

2. 超高速開発に必要な要素

超高速開発を進めるにあたり、重要な要素となりうるのは「人材」と「開発プロセス」である。

ビジネスの観点から、どのような目的を持ってシステムを素早く構築し、トライ&エラーを繰り返すのか。ビジネス的な思考ができ、開発技術を駆使してシステムとして具現化できる人材が必要となる。

また、システム開発の技術は年々進歩を続ける一方で、Webシステムを1つ構築するにしても、多様な技術要素が必要とされるようになってきており、システムの品質を保ちながら開発プロジェクトを成功させる難易度は増している。

このような中、システム開発の納期短縮と高品質の実現という、一見矛盾するような成果を出すことのできる「超高速開発」というカテゴリーのツールやプラットフォームが登場し、様々な企業で取り組みが進んでいる。

超高速開発ツールの多くは、ツール上で設定を行うとすぐに動作するアプリケーションを自動生成するなどして実行することができ、動作するアプリケーションを確認しながら開発を

進めることができる。

システムを一から開発できる技術を持たなくとも、ツールの使い方を習得していれば動作するシステムを構築することができるため、開発者に必要とされる技術要素が多岐にわたらなくても済む。

また、セキュリティの面においてもクロスサイト・スクリプティングやSQLインジェクションといった脆弱性対策が予め施されており、プログラマーが一から実装した時に比べて脆弱性が埋め込まれにくい仕組みとなっている。様々なデバイスから情報システムが利用されるようになった今日では、意識せずとも、様々なWebブラウザやモバイル機器で動作可能なアプリケーションを構築することも従来型の開発と比較した場合のメリットであるといえる。

従来型の開発工程を大幅に短縮するだけでなく、ユーザーと要件を詰めながら開発を進めることにも向いており、ウォーターフォール型とアジャイル型のどちらの開発プロセスを採用した場合でも、大きなメリットとなる。

ビジネス環境の変化が早い今日では、ウォーターフォール型で時間をかけて開発するよりも、ひとまず短期間でシステムを構築して運用を開始し、走りながら修正を加えてビジネスにフィットさせていくという開発スタイルの方が向いている場合があり、超高速開発はそのようなビジネス要求にも応えることのできる開発手法ともいえるであろう。

3. 超高速開発ツール・プラットフォームのカテゴリー

さて、ここからは超高速開発ツール・プラットフォームには、どのような種類があるのかを述べたいと思う。超高速開発ツールには、大きく分けて「設計・コード生成型」、「業務アプリ実行エンジン型」、「Non-programming PaaS型」が存在し、それぞれに特徴がある。

(ミック経済研究所のレポートからの引用)

<p>・設計・コード生成型 業務仕様情報に基づき、システムの設計からソースコードを自動で生成するツール。</p> <p>・業務アプリ実行エンジン型 あらかじめ用意されたテンプレートを元に、アプリケーションを生成するツール。</p> <p>・Non-programming PaaS型 プログラミングなしで、Webブラウザ上からの操作だけで開発が可能なツール。</p>

(ミック経済研究所:アジャイル開発で注目される超高速開発ツールの市場動向2018年度版より)

この3種類のツールについて、利用に必要とされる技術スキルと開発の柔軟性の関係を図1に、市場の規模を図2にそれ

図1 必要とされる技術スキルと柔軟性のポジショニング

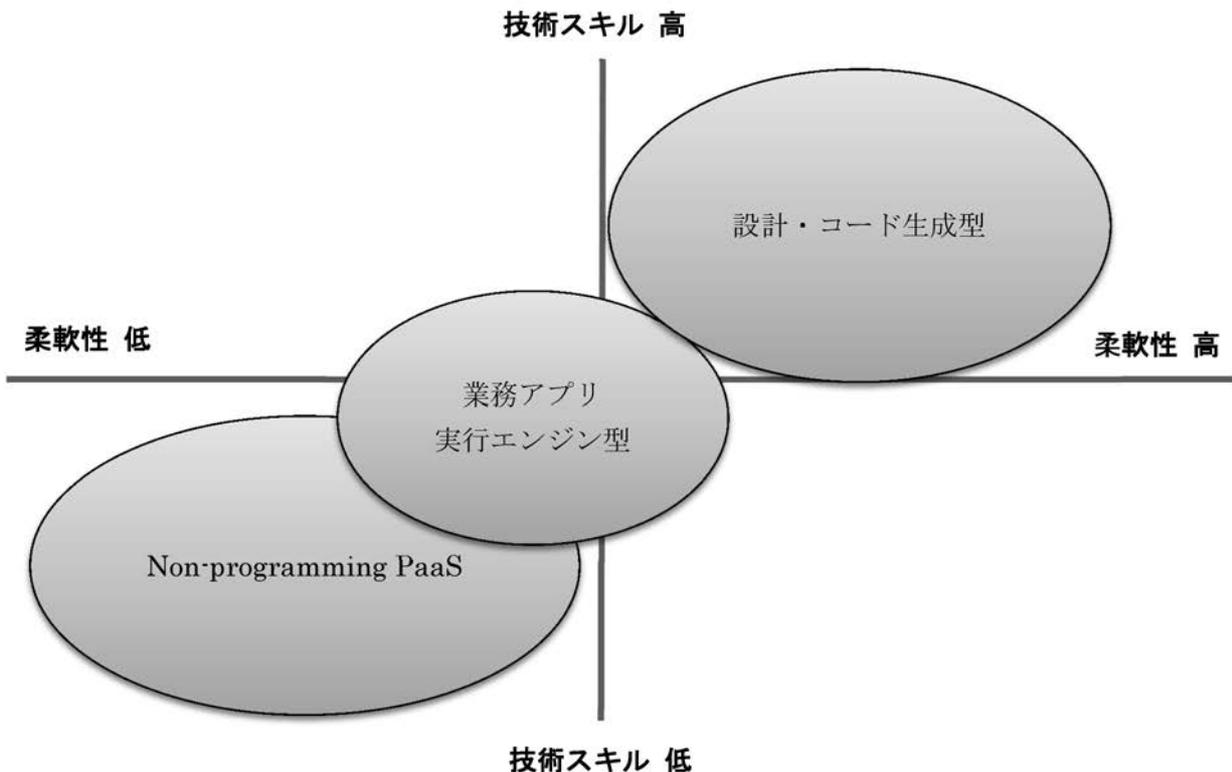
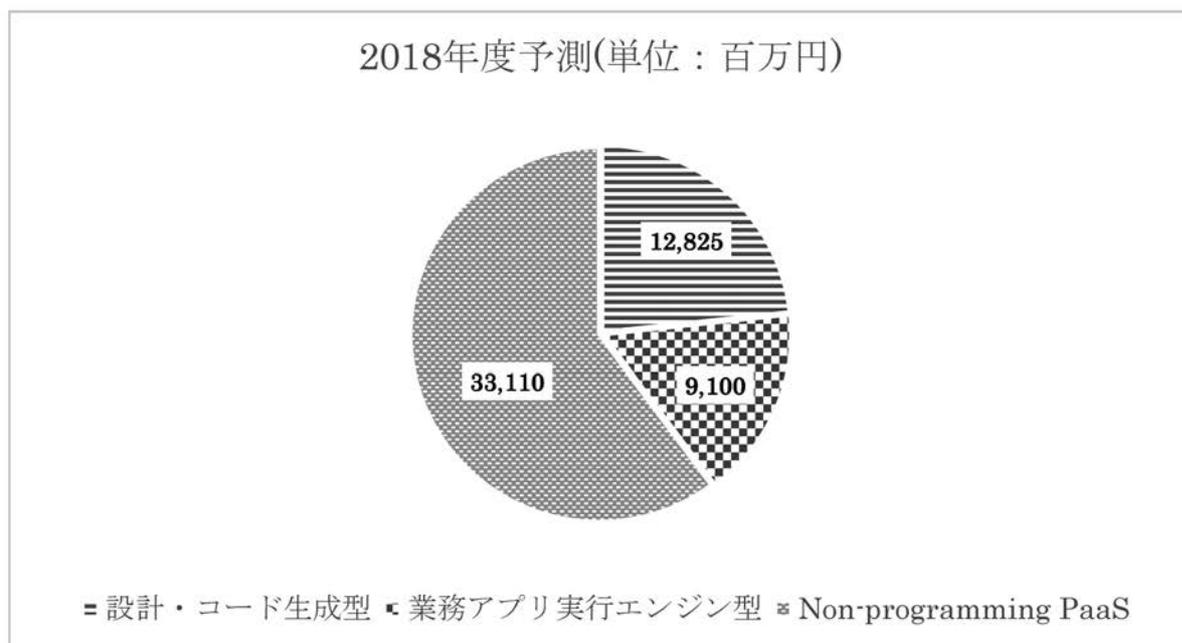


図2 超高速開発ツール市場動向:全体推計



(ミック経済研究所:アジャイル開発で注目される超高速開発ツールの市場動向 2018年度版より)

ぞれ示す。

3.1 設計・コード生成型

開発ツール上で設定された設計情報に基づき、プログラミング言語のソースコード(プログラム)を生成する。Webアプリケーションを開発するためのツールである場合、生成するプログラミング言語はJavaに対応していることが多く、多数のプログラミング言語に対応するツールも存在する。生成されたソースコードをカスタマイズしてもよいかどうかはツールによって異なる。

多数のプログラミング言語の生成に対応するツールでは、カスタマイズを前提としたものではなく、実行環境に合わせてソースコードを生成することを目的としている場合があり、生成されたソースコードのカスタマイズが禁止されていることもある。

設計・コード生成型のツールの中でも、ソースコードのカスタマイズが禁止されているものに関しては、どちらかといえば業務アプリ実行エンジン型に近いといえる。

筆者は、設計・コード生成型の開発ツールで重要なポイントは、以下だと考えている。

- ・システム開発者によるカスタマイズを前提として可読性の高いソースコードが生成されているか
- ・ソースコードをカスタマイズした後に開発ツールでの設定変更、再生成が行えるか
- ・開発ツールや統合開発環境上でデバッグを行えるか

理由は、ソースコードが目に見える以上は、システム開発者にとって読みやすいソースコードであり、システム開発者がカスタマイズしやすい必要があるからだ。

カスタマイズ実施後であっても、開発ツールによって効率的に保守・メンテナンスできる箇所はツールの設定変更のみで対応可能(=ソースコードが再生成でき、再生成してもカスタマイズ部分が残る)であると望ましい。また、システム開発者がソースコードをカスタマイズ実装した部分に問題があったとしても、統合開発環境などでブレークポイントを設定してステップ実行し、変数などを参照しながら素早く問題点を見つけ、不具合やエラーを解消できる必要がある。

設計・コード生成型の開発ツールの用途としては、本来であればソースコードの実装を伴う開発(従来型の開発≒カスタム開発)において、システム開発者が実装するソースコードの量を最小限に抑えることであると筆者は考える。開発ツールによって構造が統一されたソースコードが生成され、システム開発者のスキルに頼る属人的な開発の割合を極力減らすことで、不具合の発生を抑え、システムの品質を向上させることができる。

3.2 業務アプリ実行エンジン型

設計・コード生成型とは異なり、ソースコードを生成せずに開発ツールの設定情報をもとにアプリケーションを動作させるため、ノンプログラミングでアプリケーションを開発できるタイプのツールとなっている。テンプレートや部品を組み立てる感覚で開発ができる。開発ツールによってはシステム開発者が実

装したソースコードを、プラグイン方式で組み込むことができるツールも存在する。

業務アプリ実行エンジン型は、ソースコードが生成されていないため、バージョンアップの際にクラスライブラリ等の差し替えのみで対応が完了できることがメリットである。

一方でソースコードをシステム開発者が調べるができないため、ソースコードレベルでのアプリケーションの動作仕様の把握や、カスタマイズ実装を行う際の柔軟性の面で設計・コード生成型にやや劣るところがあるのではないかと考えている。

用途としては、基本的にはソースコードのカスタマイズを伴わない範囲、つまり開発ツールに用意された範囲(制約)の中でできることで実現可能なアプリケーションの構築に向いていると考えている。

システム開発者に要求されるスキルレベルが、設計・コード生成型と比較してそれほど高くないこともメリットであるといえる。

3.3 Non-programming PaaS型

Webブラウザ上での設定だけでアプリケーションを構築することができ、かつ構築したアプリケーションの運用環境までがセットで提供されているプラットフォームが、このNon-programming PaaS型である。システム開発者や情報システム部門でない、ユーザー部門であっても、使い方を覚えればアプリケーションを構築できる場合がある。

PaaS上にアプリケーションや部品のマーケットプレイスが用意されていることもあり、他社が構築したアプリケーションを追加する形で利用できる点がメリットである。また、他のSaaSとのAPIによる連携手段などが予め用意されていることもあるため、SaaSなどと組み合わせてシステムを構築したい場合には有力な選択肢となる。

反面、複雑な処理ロジック等のカスタマイズは苦手としている場合が多く、簡易さがあるかわりに柔軟性に劣る部分があるため、構築したいシステムが実現可能かどうかを検討する必要がある。

4. 当社の超高速開発プラットフォームAZAREA(アザレア)とそのターゲット領域

AZAREA(アザレア)は、主に企業におけるカスタム開発のWebシステムの開発ツールを主軸として、ビジネスルールエンジン、ワークフロー、バッチ処理の開発基盤を兼ね備えた開発プラットフォームである。

当社では、主要顧客である銀行、証券、保険などの金融機関様や製薬企業様向けのシステム受託開発・運用を主要な事業の1つとして展開しており、小規模から大規模まで、お客様の業務に合わせて、きめ細やかな要件のシステムを設計・

実装し、ビジネス要件に応えられるシステムを構築している。

SaaSやパッケージソフトで実現することが困難な要件のシステムを構築することが求められているため、当社ではシステム開発の柔軟性を保つことを優先に、その中で如何に開発生産性を高めるか、という取り組みを長年行ってきた。大幅な制約を課して、限られた用途で開発生産性を飛躍的に高めるのではなく、従来型のカスタム開発の領域をターゲットとし、超高速開発ツールを導入することで、お客様のシステム開発期間やコストの圧縮、より高品質なシステムを利用していただけるようになることを目指している。

当社が超高速開発プラットフォームAZAREAを「設計・コード生成型」にしたのは、小規模から大規模まで、カスタム開発での複雑な要件を実現するためというのが、一番の理由である。

超高速開発プラットフォームには、開発を簡単にする分、実現できるシステムに制約を課すことで、エンドユーザーに近い方にも開発可能としているものも存在するが、AZAREAは普段はプログラムを書いているような技術者の開発生産性を高めることを狙っている。

5. 超高速開発プラットフォームAZAREAが生まれた背景と特長

現在、当社で超高速開発プラットフォームAZAREAの開発を担当しているチームは、以前は社内ではアプリケーション・アーキテクトグループ(略称AAG)と呼ばれる技術者集団として存在していた。

AAGに所属するアーキテクトは、当社の主要顧客である金融機関様や製薬企業様をはじめとする様々なお客様向けのシステム開発プロジェクトに対して、要件定義段階から参画し、お客さまのビジネスを支援するための機能要件や、非機能要件を満たすシステムのアーキテクトを策定し、技術面からプロジェクトを成功に導く役割を担ってきた。

参画したプロジェクトでは、システム開発の基盤となるフレームワークや共通機能の設計・実装を担当し、システム開発の後工程において、個々の機能の設計者やプログラマーが高い生産性でスムーズに開発を進められるための基盤を整備。プロジェクトの後工程での技術的なトラブルが発生することを未然に防ぐ役割を果たしていた。

しかし、アーキテクトによるシステム開発プロジェクトへの人的な支援では、支援可能なプロジェクトの数がアーキテクトの人数により制約を受けることや、アーキテクトの力量にも左右されるという属人的な問題が生じていたため、より支援可能なプロジェクトをスケールさせるために、方針を転換することにした。

個別のシステム開発プロジェクトに対してアーキテクトを参画させることを控えて、アーキテクトが参画したときと同様の効

果が得られることを狙い、小規模から大規模まで様々なシステム開発プロジェクトに耐えうる共通のフレームワークを整備することにした。このとき開発されたWebシステムのフレームワークが現在のAZAREAの前身となるもので、フレームワークの利用が進むに連れて、プログラマが皆似たような処理のソースコードを実装していることに着目し、それらのソースコードを自動生成する仕組みの整備を進めることで、今のAZAREAの形となった。

6. 超高速開発ツールは普及するか？

突然だが、少しでもシステム開発に携わったことのある方は、超高速開発という言葉に懐疑的なのではないだろうか？ というのも、現在でもプログラマが必要とされるシステム開発プロジェクトは数多く存在し、プログラマの需要は減っていないからである。過去にも、RAD、CASEツール、ビジュアルプログラミング言語などというような名称で存在しており、どれもがプログラマからテキストエディタや統合開発環境を使った開発から仕事を奪うほどには至っておらず、今でも多くの労働集約型のシステム開発は根強く残っている。であるならば超高速開発ツールとは、ツールを開発するベンダーが吹聴する幻想なのだろうか？

超高速開発が、従来型の開発でよしとしている現場に導入が進まない背景には、もしかしたら作業量が増えれば増えるほど工数が増えて売上が膨らむという人月ベースによるビ

ジネスモデルが影響している可能性があるのではないかと、筆者は考えている。

ベンダーによっては、工数が減る=売上が減る超高速開発ツールの導入には気が引けるかもしれない。

システム開発の現場では、ある一定の複雑度を越えるロジックはプログラマがプログラムを組む方がよい。ただし、全てをプログラマが組む必要はないのではないかと、というのがAZAREAの出発点である。

超高速開発というと、従来型のカスタム開発と呼ばれる開発スタイルと比較され、超高速開発VS従来型開発という構図にされがちだが、我々はそうは考えていない。

当社は従来型の開発の領域の生産性を高めるために、様々な企業に超高速開発ツールを導入していただき、システム構築のスピードや生産性を高めることによって、より戦略的なIT投資に振り向けることができるように支援していきたいと考えている。当社の超高速開発プラットフォームAZAREAが、その一助になれば幸いである。

◆参考文献

- ・ミック経済研究所 「アジャイル開発で注目される超高速開発ツールの市場動向2018年度版」
- ・片山治利 「競争優位を得るためには、アプリケーションに応じた開発方法を変えよう」
<https://tech.nikkeibp.co.jp/it/atclact/active/14/369470/020200035/>