

**【DRニュース・005】：アジャイル方式の開発と超高速開発のツールについて**

2015年12月14日発信

最近では、ソフトウェアの開発を迅速に、安く、良いものを提供するには、どうしたら良いのかの議論が盛んに行われています。

そんな中から、ソフトウェア開発の進め方を探求した「アジャイル方式の開発」と「超高速開発のツール」を調べて見ます。

**(1) アジャイルソフトウェア開発 ( agile software development)**

アジャイル開発プロセスとは、ソフトウェアを迅速に、また、状況の変化に対して柔軟に対応できるように開発する手法の総称を指しています。

「アジャイル」(agile)とは、俊敏(しゅんびん)な、しなやかな、素早い、などの意味で、短いプロセスを何度も反復して次第に全体を組み立てていくアプローチの手法が多く用いられています。

従来から主流であるウォーターフォール型などの開発プロセスでは、要件定義、設計、実装、テストなどの各工程を順番に一度だけ行なうことを前提にしていますが、

・・・**アジャイル開発プロセスでは一度ですべてを作ろうとせず、当初は最低限の機能だけを持ったソフトウェアの完成を目指し、各工程を迅速に進める方式をとっています。**

とりあえず動作するソフトウェアを元に、開発チーム内あるいは顧客とチームが密接に議論を交わし、変更する箇所や追加する機能を決め、もう一度各工程を反復します。

・・・**このサイクルを短い間隔で何度も繰り返すことで、徐々にソフトウェアの完成度を高めていく。**

アジャイル開発プロセスはどのような開発プロジェクトにも適しているわけではなく、一カ所に集まった10人程度までの少人数の開発チームが、ネット関連など変化の速い分野や、最初からはっきりとした要件を定義するのが難しい分野のソフトウェアを開発する際に最も適しているとされています。


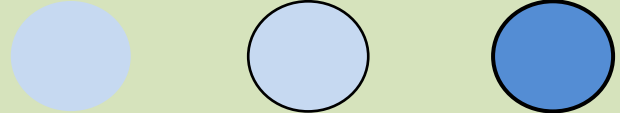
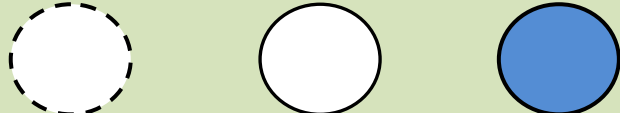
最初に考えた機能を全て完璧に作るうとしていないからです。それで本当に必要なもの、コアになるもの、大事なものから順に作っていけるので、結果として変化に対応できます。

人を重視することも、軽量であることも、動くソフトウェアを中心にする 것도、変化に対応することも、顧客との協調すらも、アジャイルの特徴を表したものではありません。その目指すところとまでは至っていません。

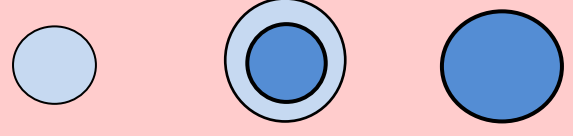
・・・**「アジャイル開発では当初に想定した機能を”全部” つくらない」**

## (2) インクリメンタル開発とイテレーション開発 (繰り返しの中で~完成度高める)

- ▶ **インクリメンタル開発**は、文字通り追加して行くということです。  
繰り返しのなかで、第1フェーズはここまで、第2フェーズはここまで作る、と規模を拡張して行く動きがあります。
- ▶ **イテレーション開発**は、何度も薄く色を塗る事で完成に持ち込む動きがあります。  
とりあえず、動く状態を作り、細かな処理を追加して行くニュアンスです。

<p>◆ <b>インクリメンタル開発とは？</b> 繰り返しの中で新しい機能を追加して行きます。</p>	
<p>◆ <b>イテレーション開発とは、</b> 繰り返しの中で大まかに作成済みの機能を追加して行きます。</p>	
<p>◆ <b>ウォーターフォール開発とは、</b> 枠を確実に作成し、後半に一気に機能を充実して行きます</p>	

- ・インクリメンタルな開発は、顧客が満足しないものを作ってしまうリスクが軽減されていない。
- ・イテレーティブな開発は、開始時点からの絵の変化を見ることができる機会を提供している。  
・・・そして一歩ずつ全体の絵の完成に向けて進んでいくのをガイドしてくれる。

<p>◆ <b>インクリメント+イテレーション開発</b> コア機能をともかく動かし、コア機能を完成させる。その後、追加の機能を作成させる。</p>	
--	--

- ・・・繰り返して、完成度を高める方法においては、折衷案を進めて行くのが、ベターか！？

### (3) 超高速開発のツール (日経コンピュータの 2015. 10. 1 版から一部抜粋)

ツールがプログラムを自動生成する「超高速開発」、2～3年前から動きは拡大しつつあります。

みずほ銀行が次世代基幹システムの開発に採用、パナソニックやソフトバンクといった大手企業も保守までを含めた「効率の高さを評価」して、新規開発に選んでいます。

..... この背景にあるのが、従来手法の限界があります。

**超高速開発とは**、プログラムを自動的に生成するツールを用いた開発手法のことで、業務に関するルールを自然言語や数式、独自の開発言語で設計情報として入力すれば、当該業務を実行するプログラムをクリックするだけで完成します。

メリット①は、これによりプログラミングと単体テストの工程をほぼゼロに短縮できる。開発、さらには、運用後の保守にかかる時間を大幅に短縮ができる。

メリット②は、開発の速さではなく、業務ルールをツールに入力するプロセスによって、システムが統制しやすくなる点を評価する。  
(従来手法に対する危機感から、30年にわたり使い続けてきたシステムは、保守を重ね、構造が複雑化、肥大化している→システム障害の遠因となる)

メリット③は、速さ、統制のしやすさだけがメリットではない。ベンダー（製品のメーカー、ハードウェアベンダー・ソフトウェアベンダー・システムベンダー）依存から脱却するために取り組める。

日本語で業務ルールを入力するのが特徴で、

「合計金額＝振込金額＋手数料」と入力すると COBOL や JAVA のプログラムを自動生成する。入力する日本語は「要件定義で洗い出したツールの辞書に登録した用語のみ」使用。

..... ここに来て、急速に普及し始めたのには理由がある。

理由その①は、ハードの処理能力が格段に向上したこと。

理由その②は、プログラムの自動変換やリポジトリ（何らかのデータや情報、プログラムなどが体系立てて保管されている場所）での大量の情報管理が実用レベルとなったこと。

..... リポジトリを持つツールで開発、スピードの秘密はリポジトリの構造にある。

#### (4) GeneXus (ジェネクス : ウルグアイ国・Artech (アルテッチ) 社製)

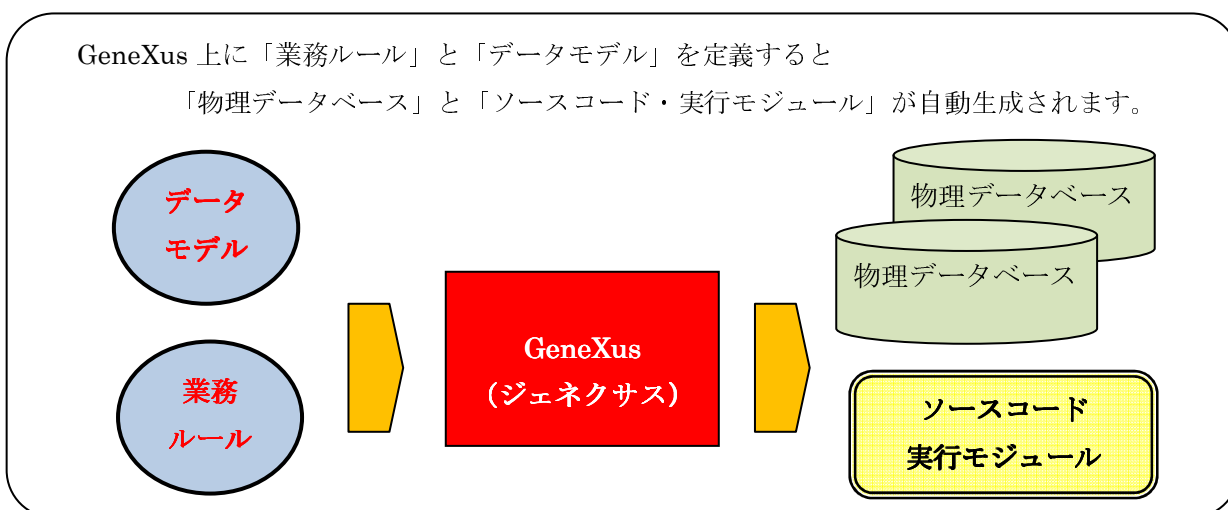
**ツールの選択技は膨大、見極めが大切となる。ウルグアイで開発されたツールを調査してみます。**

GeneXus の開発では、ユーザビューを核とした業務要件 (GeneXus に記述したこの業務要件をナレッジといい、このナレッジ群を蓄積した DB をナレッジベースという、以下**ナレッジベース**) を記述することで、様々な IT 環境に対してシステムを自動生成します。

業務要件を決めると生成後即時に動き始めるため、「**アジャイル開発とも非常に親和性が高い**」です。

また、GeneXus で構築したシステムは、**業務要件の記述だけで生成されている**ため、プログラムのブラックボックス化や属人化が原因で保守ができなくなるという事態に陥ることはありません。さらに、業務要件はそのまま、新しい IT 環境上に再構築することができるアーキテクチャも備えているため、初期の開発はもちろんのこと、長期にわたって TCO (コンピュータシステムの導入、維持・管理などにかかる費用の総額) の削減が実現できます。

大規模な業務システム構築において、生産性・保守性を最大限に引き出していただけるよう、プロセスよりもデータを先に開発するというデータ中心開発のコンセプトを核にした情報システムの全体アーキテクチャの企画、業務設計、要件定義などの超上流や上流工程の領域で活用する独自の技法と方法論 (PLAN シリーズノウハウパッケージ) を“データ総研”と開発を行っています。



- ..... **業務ルールを分析し、ツールを使いこなせる人材を育成・確保が問題となります。**
- ..... **自動生成への挑戦は何十年も前から生まれては、消えている夢でした。**  
→ツール生成できないミドルウェアの領域や業務共通の仕組みなど、重要なシステムを構築する構造化などは、別途構築が必要なのでツールの適応範囲など見極めが必要！！