

特集1 ● Agileなソフトウェア開発

第3章

Agile Modeling (AM)

「身軽な旅」の勧め

Agile Modelingとは

注1) Scott W. Ambler, 『Process Patterns』, Cambridge University Press, ISBN0-521-64568-9

注2) Scott W. Ambler, 『More Process Patterns』, Cambridge University Press, ISBN0-521-65262-6

注3) Key Process Areas, CMMでは合計18エリアが存在します。KPAとは、組織がソフトウェアプロセスを改善するために、焦点を当てるべき「鍵となる」プロセス領域のことです（CMMについては第7章で解説しているので、そちらを参照してください）。

Agile Modeling（以下AMと記述）の提唱者であるScott.W.Amblerは、XPのKent Beckと並んで、現在のAgile開発の中心的な人物の1人となっています。

AMは他のAgile開発方法論とは異なり、開発方法論や管理プロセスではなく、モデリング技法です。

AMは、現代のシステム開発に課せられている制約の中で、「費用対効果」の高い、本当に必要で求められている作業を行うことを重視します。そして、可能な限り開発作業やプロジェクト管理を効果的に行う、ユニークな技法（技術）を紹介しています。

AMは技法なので、図1に示すように別途開発方法論（開発者の職場で定めている独自のプロセスでもOK）と併用するこ

とを前提としています。

ところで、AM提唱者のScott.W.Amblerは、過去に『Process Patterns』^{注1} 『More Process Patterns』^{注2}で、Rational Unified Processに似たHeavyweightプロセスを定義し発表しており、彼のプロセスではSEI-SW-CMMのKPA^{注3}との対応関係も盛り込まれています。

このように、Amblerは開発プロセスや管理プロセスに対して、きわめて造詣の深い人です。そのAmblerが、ある意味Heavyweightプロセスと対極に位置するAMを提唱しているのは、興味深い事実です。

Amblerによれば、現代のソフトウェア開発は、規模と種類が多岐に渡っており、その中には作業手順や成果物がキッチリ定義されたHeavyweight開発方法論のような開発プロセス、管理プロセスが必要なプロジェクトも多い反面、Heavyweight開発方法論よりもっと効率の良い開発の進め

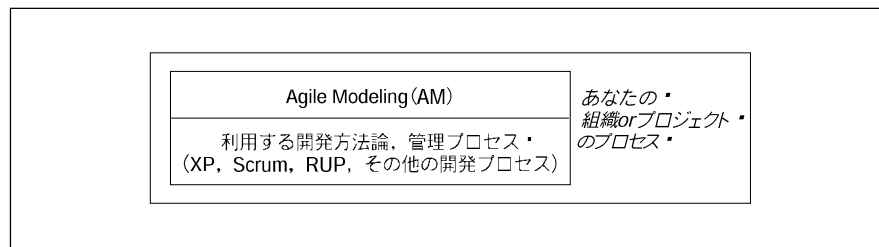


図1 AMとプロセスの関係
（『Agile Modeling』より）

方を必要とするプロジェクトも数多く存在すると述べています。

AMは、後者のようなプロジェクトに最適化されたモデリング技法ですが、Heavy weight開発方法論と併用することも可能であると、彼の著書『Agile Modeling』^{注4)}の中で紹介されています。

AMの「価値」「原則」「プラクティス」

AMが目指すのは、意味のない形式的な作業、利用効果の低いドキュメント作成、開発本来の目的に沿わないCASE Toolの利用等、無駄な作業をソフトウェア開発から排除することです。

つまり、高価なCASE Toolの利用や、“読まれることのない”分厚いドキュメントを作成することが本来の目的ではないにもかかわらず、実際には多くの開発プロジェクトでこうした無意味な作業が多く行われ、膨大な時間と労力が費やされていることを見直しています。

AMは、効率的であればToolもCheap(ポストイットや模造紙)のようなもので十分と指摘し、効率が良く、安上がりなモデリング技法やToolを数多く紹介しています。たとえば、ホワイトボードに描いたクラス図やシーケンス図をデジカメで撮影することや、模造紙にポストイットでモデリングを行うことで十分ならば、わざわざCASE Toolを使って美しい図を描く必要はないということです。

AMは、効果的な作業を進めるために、「価値」「原則」「プラクティス」を定めています。「価値」と「原則」はプラクティスを実践するための条件であり、逆に「プラクティス」を実践するためには、前提条

件となる「価値」「原則」の認識が重要になります。この「価値」と「原則」の2つがAMの基本理念であり、価値体系であるといえるでしょう。

AM自体はプロセスや方法論ではなく、各種開発方法論やプロセスと併せて利用する技法です。この点についてAmblerは、AMはXPのようなAgile方法論だけでなく、RUPのようなHeavyweightプロセスとも併用が可能と述べていますが、利用の際にはAMとプロセスの価値体系に整合性があるか、十分注意する必要があります。

逆に、AMの価値観や基本理念と、開発方法論やプロセスの価値観や基本理念が合致すれば、AMの技法を活用できる可能性が大きいといえます。

ここからは、AMの「価値」「原則」「プラクティス」を紹介しながら、AMのユニークな技法をいくつか紹介します。ただし、誌面の関係上すべてを掲載するのは無理ですので、一部をピックアップして紹介したいと思います^{注5)}。

● 価値

AMの価値は、XPと同様の位置づけにあります。つまり、AMがとらえているソフトウェア開発における価値観を表しています。

● コミュニケーション

開発者同士や関係者間の“直接会話”によるコミュニケーションの重視です。直接会話によって、多くのドキュメント作成作業が不要になり、より多くの情報共有が可能になります。

● 単純さ

単純な開発を心がけます。過剰な技術

注4) Scott W. Ambler, 『Agile Modeling』, John Wiley & Sons, ISBN0-471-20282-7(参考文献5)

注5) 詳細な解説は、書籍『Agile Modeling』(参考文献5)か、あるいはAgile Modelingの日本語公式サイトがオージス総研の藤井拓さんの尽力によって公開されているので、そちらを参照してください(<http://www.ogjs-ri.co.jp/otc/hiroba/agilemodeling/index.html>)。なお、本特集の単語も、なるべくこのサイトの訳に合わせることにしました。

特集1 ● Agileなソフトウェア開発

(必要以上のデザインパターンやアルゴリズムetc)は避け、求められている機能だけを盛り込みます。拡張性は考えません。必要になったときに、機能を追加すればよいのです。

●フィードバック

設計のピアレビュー、ソフトウェアのテストなど、評価から得られる情報を重視します。

●勇気

ソフトウェア開発では、重要な決断が要求されます。決断する勇気が重要です。

●素直

自分以外の開発関係者に対する尊敬の念を持つことや、他の人からのアドバイスを謙虚に受けとめることは大切です。

○ 原則

AMにおける原則は、実際のソフトウェア開発でAMのプラクティス(後述)を使えるようにするための、利用者側の基本姿勢を表現しています。以下、原則の一部を抜粋して紹介します。

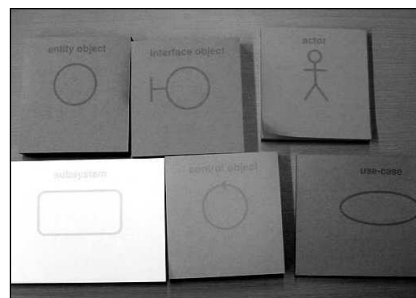


図2 便利で利用価値が大きいUMLポストイット

●単純なものの採用

開発では、「シンプルイズベスト」が重要です。要求されていないことは、開発しないようにします。拡張性や再利用性は考えず、必要になったときに行います。

●変化を受け入れる

ビジネス状況や、システムに課せられている制約の変化などで、システム要求は変化するものと考えることが重要です。そして、変化に合わせて柔軟に対応することです。

●インクリメンタル開発

すべてを1度に開発するのは困難です。システム要求の変更への対応や、システムアーキテクチャの技術的検討のために、少しずつ実現していくようにします。こうすることで、変化に適応し、顧客からのフィードバックを得ることが可能になります。

●目的に沿ったモデリング

作業に求められることを常に考え、目的を果たす成果物を作成します。余分な作業をしないようにします。

●素早いフィードバック

ホワイトボード、付箋、模造紙など、簡単ですぐ使える利用価値の高い道具を使い、チームメイトから設計やアイデアに対するフィードバックを素早く得るようにします(図2)。

●ソフトウェアが第1のゴール

第1のゴールは、要求を実現したプログラムであって、設計書やモデルなどのドキュメントではありません。



●見栄えより中身

成果物は、ソースコードのための中間成果物が多いものです。それならば、必要以上に美しく記述したり、印刷したりする必要はありません。重要なのは中身です。

●道具を知る

CASEツールなどのToolは、よく機能を理解して、効果的な使い方や使用するタイミングを理解するようにします。

●オープンで正直なコミュニケーション

オープンで正直なコミュニケーションを行うことで、ドキュメントが不要になり、より正確な情報が得られ、より適切な判断を下すことができるようになります。

●身軽な旅

Agile（機敏）に動き回るためには、余分な荷物は持たないほうがよいです。

必要以上のドキュメントや複雑すぎるモデルは、役に立つ場合もあるかもしれませ

んが、結局その保守が重荷になり、Agility（機敏さ）が失われることになります。

必要最小限のものだけを装備すれば、身軽な旅をすることができます。

●プラクティス

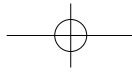
プラクティスとは、開発作業を行うときに利用する、AMの技法のことです。以下、プラクティスの一部を抜粋して紹介します。

●モデルの掲示

モデルをプリントアウトして、ホワイトボードにテープで貼り付けます。また、Webで情報を共有します。これは、コミュニケーションの促進につながります。

●他の成果物への移動

作業で行き詰まったら、別の作業をすることよい場合があります。視点を変えることで、最初に行き詰まった原因に対処できることもあります。



特集1 ● Agileなソフトウェア開発

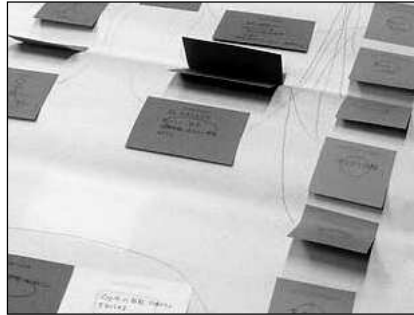


図3 模造紙とポストイットによるモデリング

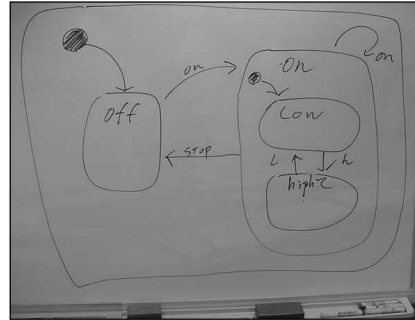


図4 ホワイトボードをデジカメで撮影

●最も簡単な道具の使用

多くのモデルは中間成果物で、使い捨てがほとんどです。美しい清書や印刷を行う必要がある場合は少ないものです。ホワイトボードの板書は、デジカメで撮影することや、紙に書き取ることで十分役に立ちます(図3, 図4)。高価なツールは必要ありません。

●パターンの緩やかな適用

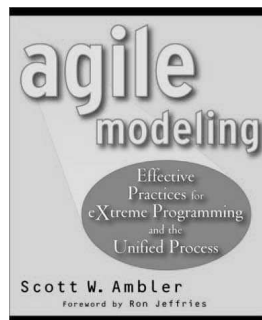
アナリシスパターン、デザインパターンにとらわれ過ぎてはいけません。パターン

の使用は、本当に必要な場合だけ、必要最低限にとどめるようにします。

●一時的なモデルの廃棄

多くのモデルは中間成果物で、使い捨てが多いものです。モデルはソースコードと一致しなくなる運命にあります。ソースコードとモデルを更新するための費用対効果を判断して作業するようにします。

コラム：AM関連書籍の紹介



本書は、Agile Modelingの提唱者Scott W. Amblerによる原典ともいべき書です。

大きくは、概論編(Part One)、理論編(Part Two)、実践編(Part Three-Four)、まとめ(Part Five)という構成をとっています。

ボリュームがあるため、てっとり早く読んでしまいたいという人には、概論編に相当するPart Oneを読んだ上で、Part Five

のchapter 29「Agile Modelingの適用もしくは逆境の克服」を読みつつ、要所要所で言及・引用される章(chapter)を適宜参照するようにすると、効率的な斜め読みができます。

何はともあれ原則(chapter 3)、価値(chapter 3-4)、プラクティス(chapter 5)には目を通しておきましょう。 ■

