システム開発プロジェクトにおけるEVMの具体的実行手順

大橋 新悟

Concrete procedure of EVM in IT systems development project

Shingo Ohhashi

情報処理振興事業協会による「EVM（Earned Value Management）活用型プロジェクトマネジメント導入ガイドライン」の公表など日本国内においてもＩＴプロジェクトに対するEVMの活用が注目されてきている。しかし、考え方としてその有効性を理解できても、ＩＴプロジェクトにおける設計、プログラム製造・テスト作業などに対するEVMの具体的な実行手順が未確立であること、管理工数のオーバーヘッドなどにより、実プロジェクトに適用している例が少ないのが実情である。本論文ではEVM実務を検討しているプロジェクトマネジャーを対象とし、日常、システムエンジニアが携わっているプロジェクトに近い具体的なモデルプロジェクトを設定し、現実的なEVMの実行手順を検討すると共に、実行時に発生する課題に対するガイドラインを提示する。

We can find the utilization of the EVM to IT project to be drawing attention in Japan, for example, from the publication of "Guide line for introduction of project management that utilize EVM(Earned Value Management)" by the Information-technology Promotion Agency, Japan (IPA). However it is very much a situation in which there are only few cases which utilize EVM, even though the effectiveness of the idea is understandable, due to the lack of detailed procedure for the utilization of EVM to design, development and testing in IT project and the additional amount of man-hour required. In this article, targeting the project manager who may be considering the utilization of EVM, we will set up a model project comparable to the actual project handled by engineers and investigate the EVM utilization procedure and any action assignment that the engineers may confront.

Key Words & Phases: I Tプロジェクト、請負契約、EVM、WBS、コストベースライン　
IT Project, Undertake contract, EVM, WBS, Cost baseline

1. はじめに

EVM（Earned Value Management）はスケジュールとコストを同一尺度で扱うことにより、プロジェクトの進捗管理、および完了時予測が可能なプロジェクトマネジメント手法として注目されており、日本国内においてもＩＴプロジェクトへの適用に関する研究1]が開始されている。一方、ＩＴプロジェクトで発生する個々の作業項目に対するEVMの具体的な実行手順が未確立であること、および管理工数のオーバーヘッドなどにより実プロジェクトに適用している例が少ないのが実情である。

本論文ではEVM実務を検討しているプロジェクトマネジャーを対象とし、日常、システムエンジニアが携わっているプロジェクトに近い具体的なモデルプロジェクトを設定し、現実的なEVMの実行手順を検討すると共に、実行時に発生する課題に対するガイドラインを提示する。

2. システム開発モデルプロジェクト

本論文で前提とするシステム開発モデルプロジェクト（以降、モデルＰＪ）について説明する。

2.1 プロジェクトタイプとフェーズ

モデルＰＪのプロジェクトタイプとしては、システム開発プロジェクトにおいて一般的に認知されているウォーターフォール型システム開発とし、プロジェクトフェーズに関しては、請負契約となる可能性が高いプログラム設計、プログラム製造、単体テストを対象とする。

2.2 WBSと見積

モデルＰＪ全体のWBS (Work Breakdown Structure) を表1に示す。作業分類としては、「開発作業」、「開発管理」、「標
準化・共通作業」、「システム環境構築」、「プロジェクトマネジメント」の5分類を設定し、作業項目毎に見積もりを算出している。尚、各作業項目の見積もり単位に関しては、工数は人日、単金、費用は単位なしとする。「開発作業」の詳細に関しては表2に示すが、サブシステムA、サブシステムBの2サブシステムを対象とし、各サブシステムでそれぞれ5本のプログラムを開発する。また、プログラムの難易度にバリエーションを持たせるため、Easy、Medium、Complexのランクを設定し、ランクに応じた工数、費用を設定する。更に、プログラム1本あたりの工数、費用は、プログラム設計30%、プログラム製作50%、単体テスト20%の比率で分割する。

尚、本論文ではモデルP Jの前提としてWBSを定義したが、現実のプロジェクトでは適切なWBS分解がEVM実践の必須条件となる。

表1 モデル P JのWBSと見積もり

<table>
<thead>
<tr>
<th>作業分類</th>
<th>作業項目</th>
<th>工数 (人日)</th>
<th>費用 (単位なし)</th>
<th>人員人数</th>
</tr>
</thead>
<tbody>
<tr>
<td>開発作業</td>
<td>サブシステムA</td>
<td>70</td>
<td>70</td>
<td>2</td>
</tr>
<tr>
<td>サブシステムB</td>
<td>80</td>
<td>80</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>開発作業</td>
<td>計</td>
<td>150</td>
<td>150</td>
<td>7</td>
</tr>
<tr>
<td>開発管理</td>
<td>サブシステムA</td>
<td>7</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>サブシステムB</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>開発管理</td>
<td>計</td>
<td>15</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>環境化・共通作業</td>
<td>環境設定</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>システム環境構築</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>プロジェクトマネジメント</td>
<td>計</td>
<td>225</td>
<td>225</td>
<td>12</td>
</tr>
</tbody>
</table>

2．3 プロジェクトスケジュール

2．2節で説明したWBS、見積もりに対して、プロジェクト期間6週間を前提とし、日次のプロジェクトスケジュールを作成した。（図1参照）

表2 開発作業の詳細見積もり

<table>
<thead>
<tr>
<th>サブシステム</th>
<th>プログラム名</th>
<th>ランク</th>
<th>全合計</th>
<th>プログラム設計 (単位の倍数)</th>
<th>プログラム製作 (単位の倍数)</th>
<th>備考</th>
</tr>
</thead>
<tbody>
<tr>
<td>サブシステムA</td>
<td>PG-A2</td>
<td>Easy</td>
<td>15</td>
<td>10</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>PG-A3</td>
<td>Medium</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>PG-A4</td>
<td>Complete</td>
<td>20</td>
<td>10</td>
<td>6</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>PG-A5</td>
<td>Easy</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>サブシステムB</td>
<td>PG-B3</td>
<td>Medium</td>
<td>20</td>
<td>20</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>PG-B4</td>
<td>Complete</td>
<td>15</td>
<td>15</td>
<td>4</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>PG-B5</td>
<td>Complete</td>
<td>20</td>
<td>20</td>
<td>6</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>サブシステムC</td>
<td>PG-C3</td>
<td>Medium</td>
<td>20</td>
<td>20</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>全合計</td>
<td>150</td>
<td>150</td>
<td>45</td>
<td>45</td>
<td>75</td>
<td>75</td>
</tr>
</tbody>
</table>

3．コストベースラインの作成

本章では、コストベースライン作成時の課題を検討すると共に、2章で説明したモデルP Jに対してコストベースラインの作成を行う。

3．1 コストベースライン作成時の課題

コストベースラインは、見積もり、プロジェクトスケジュールをもとに作成するが、作成時の主要課題として次の2点を検討した。

●各作業項目毎の進捗測定メトリクスの設定
●プログラム1本毎の費用分解

1点目の進捗測定メトリクスの設定については、固定比率配分法（Fixed Ratio法）、重み付け配分法（Weighted Milestone法）、出来高比率法（Percent-complete法）、基準達成法（Criteria Achievement法）などが主要な基準として提示されている[2],[3]が、これらの基準がシステム開発プロジェクトにおける作業項目毎の作業属性を考慮して適用する必要がある。また進捗測定メトリクスは、コストベースラインの作成時だけでなく、プロジェクト遂行中の進捗実績把握においても同一の基準として使用するため、実績データ収集の効率性も考慮する必要がある。

ここでは、モデルP Jで設定したシステム開発プロジェクトの主要作業について、作業属性を考慮した進捗測定メトリクスを定義する。

プログラム設計、プログラム製作に関しては、プログラム設計書やプログラムなどの成果物を作成した後に仕様書レビューやコードレビュー等で
修正が発生するため、進捗管理の観点からは、作業着手、成果物の初期作成完了、レビュー完了の3タイミングによる重み付け配分法を推奨する。品質管理の観点からも進捗測定メトリクスにレビュー実施を反映させるのが望ましい。重みの配分はプロジェクト毎に任意に設定すれば良いが、1例として、作業着手20％、成果物の初期作成完了40％、レビュー完了40％などが考えられる。一方、本論文のモデルP Jのように開発期間が短い場合には、簡略化した固定比率配分法でも機能すると考えることができる。作業の着手／未着手を識別可能とするため、実績計上の比率は0～100ルールではなく20～80ルールを推奨する。

単体テストについては、基本的にはプログラム設計、プログラム製造と同様に、作業着手、単体テスト完了（発生バグ対処完了）、単体テスト結果レビュー完了の3タイミングによる重み付け配分法が考えられるが、テスト項目の消去率を考慮した出来高比率法、または品質目標を考慮した基準達成法が実施できるべきである。ただし、進捗測定メトリクスの要素としてテスト項目数や品質目標を採用する場合、テスト項目の追加、品質目標の見直し時の対応を考慮する必要がある。

その他「標準化・共通作業」、「システム環境構築」の標準規定作成、システム環境設計などはプログラム設計と同様に扱えば良い。一方、システム間調整などの明確な成果物が定義されていない作業に関しては、プロジェクト期間完了日に100％計上する固定比率配分法の0～100ルールで問題無いと考える。尚、「開発管理」、「プロジェクトマネジメント」については、コストベースラインに含めず別管理となるため、特に進捗測定メトリクスは定義しない。モデルP Jで採用する進捗測定メトリクスを表3に示す。

次に2点目のプログラム1本毎の費用分解についてであるが、2章で説明したモデルP Jの場合、2サブシステムで1本のプログラム開発であり、プログラム難易度のバリエーションについてもEasy、Medium、Complexの3ランクだけしか設定していないが、実際の大規模プロジェクトでは、数百本のプログラムに対してプログラム1本毎の見積もりが存在する。また、表2ではプログラム1本あたりの費用をプログラム設計30％、プログラム製作50％、単体テスト20％の比率で分割したが、実際のプロジェクトにおいても何らかの基準を設定する必要がある。プロジェクトトリーダのコストベースラインを作成するための作業量軽減のためには、プロジェクトとしての基準値を設定しサブチームリーダに適切な作業移管を行うことやP M S（Project Management System）ツールの活用が必要となる。

### 表3 モデルP Jの進捗測定メトリクス

<table>
<thead>
<tr>
<th>作業項目</th>
<th>成果物の定義</th>
<th>実了率</th>
<th>検査対象メトリクス</th>
</tr>
</thead>
<tbody>
<tr>
<td>プログラム設計</td>
<td>プログラム設計のレビュー</td>
<td>80%</td>
<td>80% (80%)</td>
</tr>
<tr>
<td>プログラム製作</td>
<td>プログラム製作のレビュー</td>
<td>80%</td>
<td>80% (80%)</td>
</tr>
<tr>
<td>単体テスト</td>
<td>単体テストのレビュー</td>
<td>80%</td>
<td>80% (80%)</td>
</tr>
<tr>
<td>プロジェクトマネジメント</td>
<td>プロジェクトマネジメントのレビュー</td>
<td>80%</td>
<td>80% (80%)</td>
</tr>
</tbody>
</table>

### 3. 2 モデルP Jのコストベースライン作成

表3の進捗測定メトリクスをもとに作成したモデルP Jのコストベース一覧を表4、コストベースライングラフを図2に示す。

### 表4 モデルP Jのコストベース一覧

<table>
<thead>
<tr>
<th>作業項目</th>
<th>成果物の定義</th>
<th>実了率</th>
<th>検査対象メトリクス</th>
</tr>
</thead>
<tbody>
<tr>
<td>プログラム設計</td>
<td>プログラム設計のレビュー</td>
<td>80%</td>
<td>80% (80%)</td>
</tr>
<tr>
<td>プログラム製作</td>
<td>プログラム製作のレビュー</td>
<td>80%</td>
<td>80% (80%)</td>
</tr>
<tr>
<td>単体テスト</td>
<td>単体テストのレビュー</td>
<td>80%</td>
<td>80% (80%)</td>
</tr>
<tr>
<td>プロジェクトマネジメント</td>
<td>プロジェクトマネジメントのレビュー</td>
<td>80%</td>
<td>80% (80%)</td>
</tr>
</tbody>
</table>

### 4. EVMによるプロジェクト評価

本章では、プロジェクト評価時のEVM適用課
題を検討すると共に、モデルPJの擬似進捗に対してEVMによるプロジェクト評価を行う。

4. 1 プロジェクト評価時のEVM適用課題

EVMはプロジェクト実施前にコストベースラインとして作成した「出来高計画値：PV (Planned Value)」に対して、プロジェクト遂行中に「出来高実績値：E V (Earned Value)」、「実コスト：AC (Actual Cost)」を測定し、「スケジュール差異：SV (Schedule Variance)」、「コスト差異：CV (Cost Variance)」等の状態、および「スケジュール効率：SPI (Schedule Performance Index)」、「コスト効率：CPI (Cost Performance Index)」等の指標を把握することにより、プロジェクトの進捗管理や完了時予測が可能なマネジメント手法である。評価時の主課題として次の2点を検討した。

●請負契約におけるACの扱い
●効果的な評価指標

1点目は請負契約におけるACの扱いであるが、国内のシステム開発プロジェクトにおいて頻繁に適用されている請負契約の場合、基本的には実コストの発生は無し、受注者は請負金額の中で予定した成果物を納品することになる。また、情報処理振興事業協会発行のEVMガイドライン[4]においても「請負部分の作業についてはACにE Vの値を用いる」と提示されており、この場合、EVMにおけるCV (= E V－AC) は常に0となるため意味の無い項目となる。CPI (= EV/AC) についても常に1となるため同様である。よって、請負契約の場合の発注者に対する説明責任の観点ではSV (= E V－PV)、およびSPI (= EV/PV)を用いることになる。

一方、発注者に対する説明責任ではなく、受注者側の内部コスト管理のためにEVMを適用する場合、管理する金額自体を発注者側のコストから受注者側のコストに変換する必要があるが、プロジェクトの原価構造がプロジェクトメンバー全員に見なければなどの問題があるため、金額ではなく工数（人月、人日）で管理するのが望ましい。尚、主要なコスト増加要因としては仕様変更の発生、技術採用時の生産性見込み相違、品質の確保できていないようによる後退、プロジェクト変更のスケルトンマッチなどが挙げられるが、これらについては変更管理、リスク管理、品質管理、人的資源管理などを含め、統合的にマネジメントし早期検出する工夫が必要となる。

次に2点目の効果的な評価指標についてであるが、E VMでは現時点の状態を表すSV、CVの他に、「プロジェクト総予算：B A C (Budget At Completion)」をベースとした「残コスト予測：ET C (Estimate To Completion)」を算出するため。「残置予測：B A C －E V」、「完了時合計予測：E A C (Estimate At Completion)」を算出するため、E VはE A C＝A C＋ET C）、E C T（算出式）が提示されている。

ここで請負契約の場合には、前述の通りCPI＝1となり、ETCの算出において効率実績が考慮されないため、CPIではなくSPIを代用することとする。

一方、SPI（＝E V/PV）は作業属性、作業担当チーム（会社）毎に異なると考えるのが現実的であるため、プロジェクト全体のETC、E A Cを算出する場合には、作業属性、作業担当チーム（会社）毎にSPI、ETC、E A Cを算出し、その累計値を使用すべきである。本手順により問題発生時の原因究明ブレークダウンも容易になる。

4. 2 モデルPJの擬似進捗

モデルPJの擬似進捗として、プロジェクト開始20日時点においてシステムのBのPG－B2、PG－B3、PG－B5の製造が完了していなく達成状態を設定する。擬似進捗のE V推移表を表5、E V推移グラフを図3に示す。

<table>
<thead>
<tr>
<th>表5 E V推移表</th>
</tr>
</thead>
<tbody>
<tr>
<td>日付</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>33</td>
</tr>
</tbody>
</table>

図3 E V推移グラフ

プロジェクトマネジメント学会2004年度春季研究発表大会予稿集 91
4. 3 モデルPJに対するプロジェクト評価

4. 2で説明したプロジェクト開始20日時点の擬似進捗に対するEVMによるプロジェクト分析結果を表6に示す。

表6 EVMによるプロジェクト分析結果

<table>
<thead>
<tr>
<th></th>
<th>EVMによるプロジェクト分析結果</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BAC</td>
</tr>
<tr>
<td>プロジェクト全体①</td>
<td>15</td>
</tr>
<tr>
<td>プロジェクト全体②</td>
<td>15</td>
</tr>
<tr>
<td>プロジェクト全体③</td>
<td>15</td>
</tr>
<tr>
<td>プロジェクト全体④</td>
<td>20</td>
</tr>
<tr>
<td>プロジェクト全体⑤</td>
<td>10</td>
</tr>
<tr>
<td>プロジェクト全体⑥</td>
<td>10</td>
</tr>
<tr>
<td>プロジェクト全体⑦</td>
<td>10</td>
</tr>
<tr>
<td>プロジェクト全体⑧</td>
<td>10</td>
</tr>
<tr>
<td>プロジェクト全体⑨</td>
<td>10</td>
</tr>
<tr>
<td>プロジェクト全体⑩</td>
<td>120</td>
</tr>
</tbody>
</table>

実際のプロジェクトでは、プロジェクトリーダーは各合計欄（表6の網掛け部分）を時系列にてマネジメントしていくことになるが、今回は作業項目レベルまでブレーカダウンした20日時点の分析結果をもとに評価する。尚、負債契約を想定し、前述の通りAC=EVとし、ETCにはSPIを代用したETC=(BAC-EV)/SPIの算出式を採用している。また「完了時コスト差異」CVAC（Cost Variance At Completion）算出式はCVAC=BAC-EACを追加する。

プロジェクトの評価とは表6を見れば自明であるが、「プロジェクト全体①」欄のSV=-23,
CVAC=-34の原因は、サブシステムBであり、更には、PG-B2, PG-B3, PG-B5が原因であることがわかり、この原因明示ブレーカダウンを即時に行うためには、表6に示す様な作業項目毎のEVM管理が必須となる。また、CVAC等の予測値に関しては、「プロジェクト全体①」欄では各作業項目の合計値として算出し、「プロジェクト全体②」欄ではプロジェクト全体のPV合計値、EV合計値から算出している。ここで注目したいのは、「プロジェクト全体①」と「プロジェクト全体②」の間にCVAC等の予測値に14（BACの約7％）の差異が発生しており、高い精度を求めるためにも「プロジェクト全体全体」欄で採用した各作業項目の合計値としての算出を推奨する。

5. まとめと今後の課題

本論文ではシステムエンジニアが日常携わっているプロジェクトに近い具体的なモデルPJを使用し、EVMのコストベースライン作成、プロジェクト評価時の課題、具体的手順を検討した。
プロジェクトリーダーにとって、EVMはプロジェクト全体の状態を大別的に把握する際の基盤として有効な手法であるが、その恩恵を得るためには、適切なWBS分解が不可欠である。また現実のプロジェクトマネジメントの側面では、進捗遅延またはコストオーバーの兆候をEVMにより検知した場合、作業項目レベルにブレーカダウンできることが重要となるため、PMSツールの活用が必須となる。

今後は今回の一覧を基に実際のプロジェクトに適用し有効性を検証すると共に、PMSツールによる効率的な管理方法を検討したい。

本論文がEVM実践を検討しているプロジェクトマネジャーの手助けになれば幸いである。

謝辞 この研究は、日本電気株式会社によるMPMアカデミー講座の中で行いました。MPMアカデミーに多大な支援を頂いた弊社、佐藤執行役員に深謝いたします。

参考文献

92 プロジェクトマネジメント学会2004年度春季研究発表大会予稿集