

第2分科会

プロジェクト描写のための情報構造 ～正確なプロジェクト把握のために～ An information structure for describing a project

主査:	板倉 稔	(株式会社 ビズモ)
副主査:	森本 勝美	(中国電子商務協会ジャパン)
	早川 勲	(株式会社 山武)
研究員:	高橋 敏浩	(株式会社 日立システムアンドサービス)
	古長谷 貢一	(双日システムズ株式会社)
	新井 章寿	(株式会社 NTTデータ三洋システム)
	越野 僚太	(三菱電機コントロールソフトウェア株式会社)
	谷野 靖雄	(三菱電機コントロールソフトウェア株式会社)
	長谷川 俊之	(伊藤忠テクノソリューションズ株式会社)
	張田 文邦	(伊藤忠テクノソリューションズ株式会社)

概要

プロジェクトにはそれぞれの役割を持った人がいて、様々な意志決定をする。意志決定の元になるのは、プロジェクトの状況である。プロジェクトの状況は、モノ作りしている担当者から発信され、まとめる役割を持つ人へ順々に報告されていく。報告を受け取った人は、それぞれの役割で意志決定を下す。意志決定はプロジェクトの成功を左右する。プロジェクトを成功させるためには、意志決定の元となる正確な状況報告は欠かせない。

本研究では、それぞれの役割を持つ人々に適切な情報を与えるため、プロジェクト情報を構造化し、正確な状況把握と報告する方法を提案する。

Abstract

People who are concerning a project make various decisions. The decisions may decide the success or failure of the project. For better decision making, it is necessary to get the correct states of the project and to inform correctly the appropriate people.

We discovered a method for getting correct value of project states and a structure for abstracting the value to the appropriate people.

1. はじめに

ソフトウェア開発時、開発実態を把握し、把握された情報が関係者の間で共有されている必要がある。この情報が、様々な人の判断の基礎になるからである。

現場の状況を正しく把握し、必要な情報を関係者に正しくかつ適切に伝える必要があるが、必ずしも上手くいっていない。また、報告の研究はなかなか進んでいないのが実情である。

開発現場の実態を如何に伝えるか、プロジェクト・マネージャ（以下PM）は悩んでいる。

2. 研究背景と研究目的

(1) 研究背景

現場のPMが、遭遇した問題を具体的に挙げてもらった。その結果を表1に示す。

表1：今プロジェクト現場で起きている問題点

項番	具体的に起きていること	問題点	進捗報告の問題	仕様凍結の問題
1	外注先からの報告が実態を表していない	プロジェクトの実態が掴めないで、対策が打てない	○	
2	週報などの進捗記述精度にムラがある		○	
3	100%完了と言われたが、実は完了していなかった		○	
4	次工程で品質問題が起きる		○	
5	進捗報告をその都度、報告を受ける側の都合で作っている	進捗報告資料作成に時間がかかる	○	
6	要件未決のまま次工程へ進む	次々に変更が発生し、進捗管理できない		○

我々は問題が現れる都度、対策を打っているつもりであるが、プロジェクトが変わるたびにこの種の事象は起こる。列举した問題点を分類すると進捗報告の問題と仕様凍結の問題の2つになる。表1の項番1から5までが進捗把握と報告の問題であるため、それを採り上げる。進捗報告の問題に対する我々が打ってきた対策が表層的なため、毎回同じことが繰り返されていると考えている。本研究では、進捗報告の根本を明らかにして、問題の根を絶ちたい。

(2) 研究目的

プロジェクトの進捗報告には、正確な実態把握が必要である。実態把握と状況報告をするために考えなくてはいけないことや、やらなくてはいけないことの本質を我々が理解することを目的とする。プロジェクト情報の構造を整理し、進捗の実態把握と状況報告を対象として問題が解決できる具体的な施策を考える。

3. プロジェクト情報の構造

3.1 コミュニケーション構造

プロジェクトには受注元のPM、発注元の役員等、様々な人が関係している。このプロジェクトに関係している人達を本論文では、関係者と呼ぶ。彼らは、それぞれのコミュニケーションチャネルを持っている。この関係者間のコミュニケーションチャネルは、図1で示すとおりネットワーク構造を構成している。このコミュニケーションチャネルを通して、プロジェクトで把握された実態と状況が飛び交っている。

この飛び交っている実態と状況は、プロジェクトに存在する予算やプログラムなどに基づいて作られる。プロジェクトに存在する予算やプログラムなどの“モノ”を要素と呼ぶ。

関係者によって、知りたい観点が異なるので、どの要素を見るかも異なる。つまり、誰かに情報を報告するという事は、相手が必要とする観点からプロジェクトを描写することである。例えば、プロジェクトの損益に対する意思決定が必要な人には、損益の観点でプロジェクトの描写が必要である。

加えて、関係者によって、必要となる情報の抽象度も異なる。例えば、役員もPMも、プロジェクトの進捗を知りたいが、必要とする情報の詳細度（抽象度）は異なる。

このように、コミュニケーションは、ネットワーク構造と観点と抽象度の掛け算であるので複雑である。このうち観点と抽象度を整理すると、情報の構造が単純になるはずである。

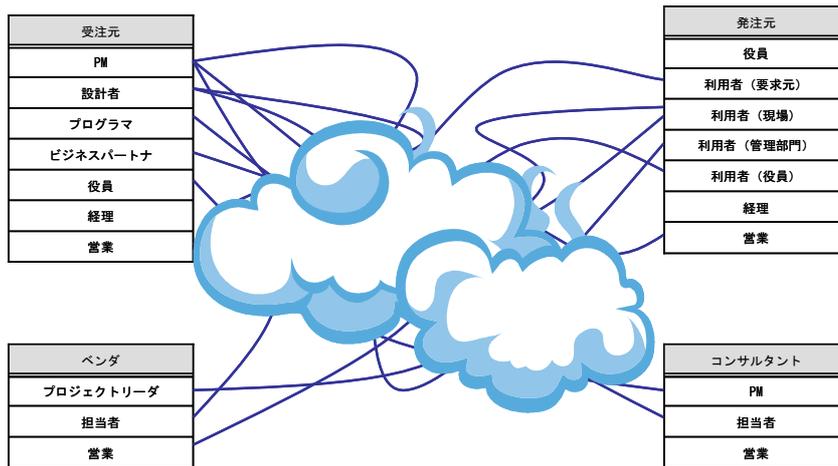


図1：関係者間の情報

3.2 プロジェクト情報の構造

観点と抽象度を整理した結果を、図2に示す。図2の横軸は観点、縦軸は抽象度を示す。横軸の観点は、例えば予算、費用、進捗などが挙げられる。また、抽象度の例として、役員、PMなどが挙げられている。役員やPMはプロジェクトでの損益の決断や品質向上策の策定などの役割を果たしている。つまり、縦軸にはPMなど人が記されているが、ここで表されている人は役割を意味している。役割を短い言葉で表現するために、人を用いた。

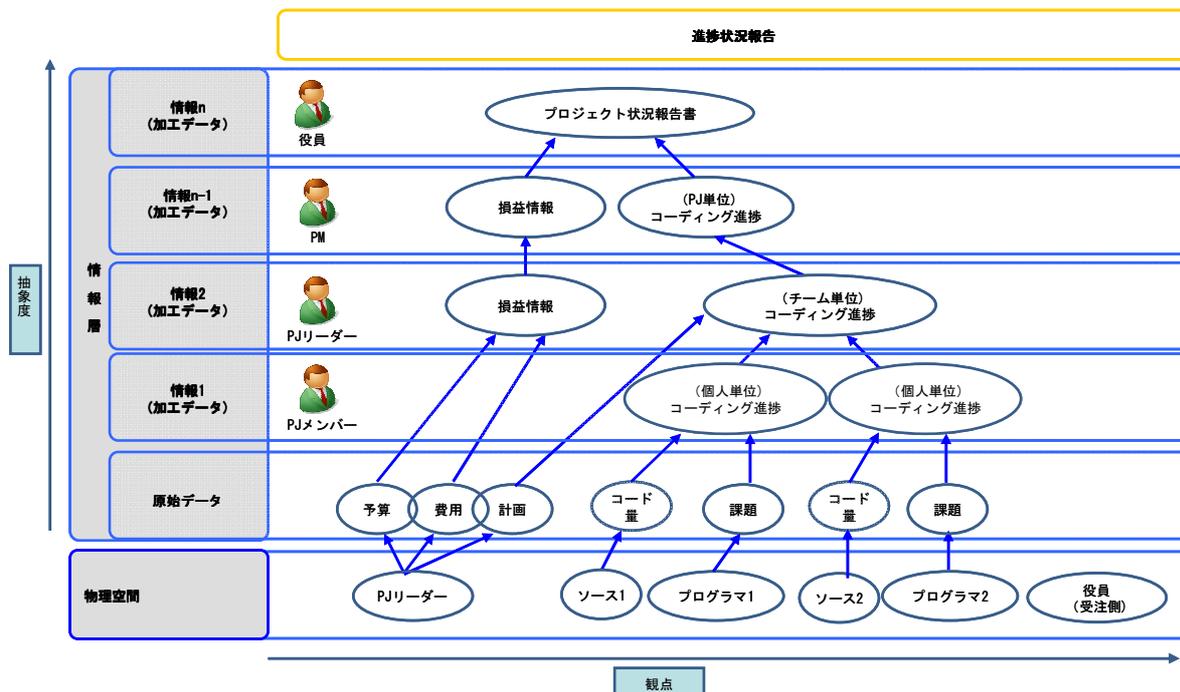


図2：プロジェクト情報の構造

図2のプロジェクト情報の構造を、PMを例に説明する。PMには、損益情報と進捗情報が集っている。PMは損益情報と進捗情報を把握し、プロジェクトの方向付けをする役割である。その役割を果たすため、損益情報の観点と進捗情報の観点で実態把握する。損益情報は、プロジェクトリーダーが持っている予算と費用から得ることができる。進捗情報は、物理空間にあるソースのコード

量から掴むことができる。ソースのコード量を個人単位に集計すれば個人単位の進捗情報を、チーム単位に集計すればチーム単位の進捗情報を掴むことができる。

プロジェクト情報を構造化したことで、観点により必要な情報が検出され、役割に合わせて情報が加工されていく過程が明らかになった。各々の役割に必要な情報は、観点と抽象度が異なる。メンバから上位に行くに従って、情報は合計され、些々情報はカットされ、抽象度があがる。

図2で表記されている4つの用語について説明する。

(1) 物理空間

プロジェクトは、様々な要素で構成されている。要素には、設計者、開発者、役員のように人とプログラムソースなどのモノがある。この要素が存在している空間を、物理空間と呼ぶ。

(2) 情報層

我々はプロジェクトにおいて様々な情報を扱っている。情報は関係者間、例えばプログラマからリーダーに、リーダーからPMに伝達される。このように情報が存在している空間を情報層と呼ぶ。

(3) 原始データ

物理空間に存在している要素は、それぞれ属性を持っている。例えば、人は、労働時間や単金などの属性を持っている。プログラムソースはコードであり、そのコードには、コード量、品質などの属性を持っている。この属性を原始データと呼ぶ。図2の予算、費用、計画や、ソースのコード量といった情報が、原始データである。

(4) 加工データ

ある役割の観点でプロジェクトを見ることにより、必要とされる原始データが見える。また、観点によっては、複数の原始データを結合したり、集計したりすることがある。結合や集計されたデータを加工データと呼ぶ。加工が進むにつれ、抽象度が上がり、詳細な情報は削ぎ落とされる。

3.3 代用特性

現場のPMは、プロジェクトの出来栄を計る際、定性的な評価を定量化する工夫、あるいは定量的な指標の選択をしてきた。設計工程の進捗度や、品質の良し悪しなどを直接計測する指標はないので、作成したページ数とかバグ件数を指標の一つとして定量化する。そういう指標を代用特性と呼ぶ。

例えば、損益という観点でプロジェクトを見てみる。プロジェクトには予算という原始データがある。PMは、役員に予算とコストなどの情報を使用し、プロジェクトの損益を報告する。役割は、PMから役員に変わったが、予算という原始データは、そのまま使用されている。つまり、役割が変わっても、原始データの情報はそのまま使用されたということである。この例は、図3のA(予算)とA'(プロジェクトの損益)の関係であり、A'の観点で見たとき、Aが存在するため、代用特性は不要である。

一方、品質という観点で状況報告する場合、損益報告時の予算と異なり、直接品質を計る定量的な指標はない。品質を計るには、残存するバグ数を証明すればよいが、そのような指標は存在しない。そのため、発生したバグ数を用いて、これを推定する。

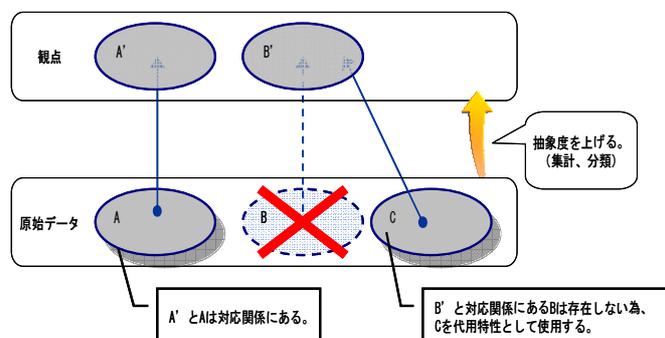


図3：代用特性

このように直接測る指標がないので、代わりに測定できる数値を代用特性として使用する。この例は、図3のC（バグ数）とB'（品質）の関係である。B'（品質）の観点で状況を描写したいが、直接対応できるB（品質を計る指標）が存在しないため、代わりとなるC（バグ数）を代用特性として使用する。

3. 4 報告の観点

我々は、都度の進捗報告（図4のA～Eクロスセクション）を積み重ねることで、プロジェクトの変化を把握している。クロスセクションのデータを時系列に積み重ねる事により、変化を捉える事ができる。

このように、時系列な変化はクロスセクションのデータの積み重ねである。したがって、報告の観点は、クロスセクションのデータの精度が重要となる。

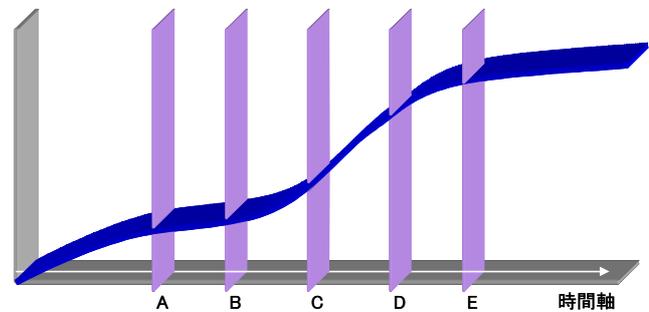


図4：クロスセクション

4. プロジェクト情報の構造を具体的に当てはめると

前章でプロジェクト情報の構造を明らかにした。プロジェクト情報の構造は、プロジェクトの管理で使われるメタ情報であるから、プロジェクト管理とは何かを始めに解き明かしておきたい。

プロジェクト管理は、「一所懸命考えたいい加減な仮の計画値」に「実際の値」を近づけていくことである^[1]。一所懸命考える仮の計画値の観点は、納期、コスト、品質である。

物理空間にある仕様書やソースは、メンバがそれぞれ作り出す。従って、仕様書やソースを持つメンバが、どれくらいできたかという「実際の値」を最も正確につかんでいる。「実際の値」の把握を難しくしている一つ要素は、メンバが持っている仕様書やソースから、情報層へ加工するための項目（抽出したり、選択したり）を決めていないからである。仕様書やソースから原始データとなる「実際の値」を掴むことが重要である。この掴み方を4.2節で述べる。

このように「実際の値」である原始データを掴めば、プロジェクトの状態が正しく把握できるようになる。それを基に、それぞれの役割にあわせて加工すれば、PMが持っている一所懸命考えた「計画値」と比較することにより差異が明確になり、プロジェクトで起こっていることが把握できる。

4. 1 プロジェクト情報の構造に報告を当てはめる（原始データ、加工データ）

プロジェクト情報の構造を、実際のプロジェクトに当てはめてみる。プロジェクトには、メンバやPM、スポンサなどそれぞれの役割を持った人たちが存在する。その役割ごとにプロジェクトの状態を報告する会議が行われている。

(1) 実際の会議体をプロジェクト情報の構造に当てはめる

実際の会議体を図2のプロジェクト情報の構造に当てはめる上で、進捗報告を例にとる。図5に某プロジェクトの「会議体と進捗報告（例）」を示す。

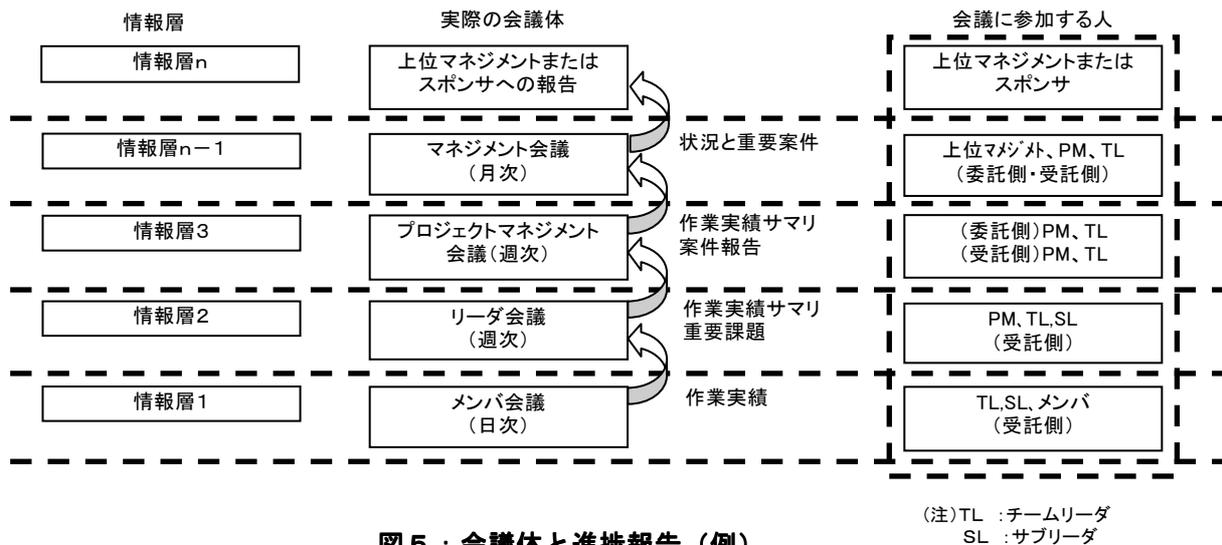


図5：会議体と進捗報告（例）

メンバとサブリーダーやチームリーダーとの日々の打合せが情報層の最下層で行われる。この会議では、メンバが個々に持っている作業実績をサブリーダーやチームリーダーに報告する。これが情報層1である。次にその作業実績をチーム単位に加工してPMに報告するのが情報層2である。情報層3は、PMが委託側のPMに報告する空間であり、ここでは情報層2の作業実績サマ리를更にまとめたり、委託側と調整しなくては解決できない案件について報告する。

このようにプロジェクトの状況報告は、報告を受ける側に合わせて抽象度が変わる。

(2) 進捗報告の抽象化の実際

各々の情報層での抽象度の変化を明らかにするために、進捗報告で使う報告書を図2のプロジェクト情報の構造に当てはめる。図6に外部設計工程の進捗報告書の抽象度変化（例）を示す。前述の図5では、受託側と委託側が存在するが、ここでは、受託側の進捗報告書のみ考える。

項番	情報層	情報保持者	情報の移動	進捗報告書	課題・問題点報告書
1	情報3	PM	PM⇒スポンサ	(PJ単位) 進捗状況報告書	(PJ単位) 問題点一覧
2				(PJ単位) アローダイアグラム	
3	情報2	TL, SL	TL, SL⇒PM	(チーム単位) 仕様書進捗	(チーム単位) 問題点一覧
4	情報1	メンバ	メンバ⇒TL, SL	(個人単位) 仕様書進捗	(個人単位) 問題点
5	原始データ	メンバ	メンバ	仕様書 ページ数(例)	課題・問題点自体
6	物理空間	メンバ、TL(チームリーダー)、SL(サブリーダー)、PM、スポンサ(課長/部長/役員) / 仕様書、課題・問題点			

↑ 情報の抽象化

図6：外部設計工程の進捗報告書の抽象度変化（例）

外部設計工程では、メンバが仕様書を持っている。仕様書が持っている原始データのページ数(例)が、仕様書の進捗になる。この個々の仕様書の進捗を、管理単位、例えば個人、チーム、プロジェクトでまとめると、おのおのの情報層に対応した報告データになる。最後はPMの(PJ単位)の進捗状況報告書や、アローダイアグラムになる。

次に、図6の課題・問題点報告書の抽象度の変化について述べる。課題・問題点自体は、物理空間の要素(人)が持っている原始データである。これを個人単位にまとめれば、メンバの問題点となり、順にチーム、プロジェクトの問題点となっていく。

PMは、(チーム単位)に作成される加工データを(PJ単位)に加工する。正確に(チーム単位)

の情報を掴むためには、(個人単位)が正確でなければならない。抽象度の変化を逆に辿れば、進捗報告や、課題・問題点報告は、メンバが加工する情報1のデータ、そして原始データに行き着くはずだ。

以上のことから4章の冒頭で述べた「実際の値」を正確に掴むためには、メンバ単位の進捗状況の把握が大切であることが解った。つまり物理空間にある要素から原始データ、加工データへの抽象化が正確にできれば、プロジェクト状況を正確に把握できるということである。

物理空間から情報層へのデータ加工は、「抽出」と「選択」の2つのやり方がある。

抽出は、物理空間にある要素から原始データを見つけ出すことである。物理空間にある仕様書自体は進捗を表していないので、そのままでは進捗情報は見えない。見えていないものの中から出来栄を計れる代用特性を探す。

選択は、原始データから情報層nまでのデータ加工で使う。物理空間にある要素から代用特性を抽出して進捗が見えるようになった。見えている状態から報告を受ける側のニーズに合わせて原始データ、あるいは加工データを選択する。

進捗情報を掴むために必要な代用特性は何かを考えて、適切な項目を抽出したり、選択したりする。抽出も選択も、各情報層で変化を捉えるために必要となる情報が漏れないようにする。また、報告を受ける人に誤解を与えたり、人によって捉え方がぶれたりしないようにする注意を払うことも忘れてはいけない。

4.2 進捗の観点での原始データの抽出例

外部設計書の進捗を測ることを例として、適切な代用特性の抽出はなにか、報告を受ける側が誤解しない管理の仕方はなにかを考える。図7に外部設計書から原始データの抽出例を示す。

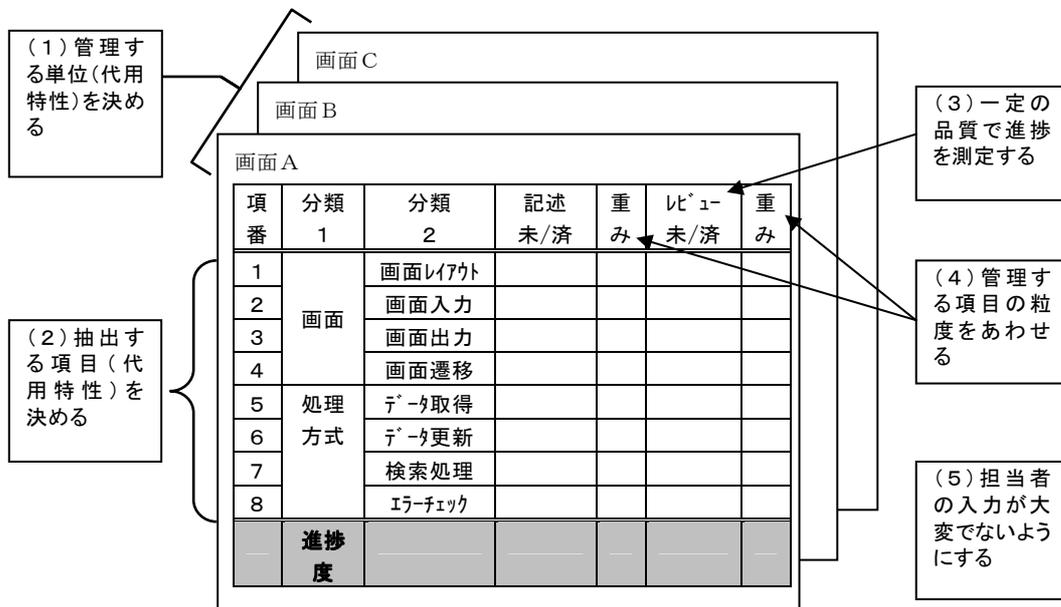


図7：外部設計書から原始データの抽出例

(1) 管理する単位(代用特性)を決める

外部設計で画面を設計する場合を考えてみる。作成したページ数を計画したページ数で割り算して、進捗を把握する方法がある。また、画面数で管理する方法もある。画面数で管理できる場合は、ページ数での管理より画面数で管理することを勧める。

画面が変更になると十中八九ページ数は変更になる。逆にページ数が変更になっても画面数は変わらない場合がある。また、外部設計書の工程では、既に画面数は決まっている場合が多い。ページ数より画面数のほうが、分母が変わらないので進捗率全体が振れることが少ない。

このように、さまざまな代用特性が考えられるが、長短を比較の上、適切な代用特性を選択する必要がある。また、代用特性は構築するシステムの特長（画面系かトランザクション系か）や、開発形態（新規か改造か）などによっても変わってくる。

（２）抽出する項目（代用特性）を決める

管理する単位が決まったら、どの項目（情報層（図 2）で必要となる項目）で、何をするか（どのように項目を掴むか）決める。図 7 の分類 1、2 は外部設計書の目次である。目次は、何を定義すべきかが示されている。図 7 の横軸では、記述とレビューの済み/未済み、つまり何をするか表した。これで定義すべきものが、どこまで進んでいるかを測ることができる。

（３）一定の品質で進捗を測定する

品質が担保されない仕様書は、何ページ書こうと、担当者が完了したと宣言しようと完成したとは言えない。そのため進捗を測定する上では、一定の品質が確保されていなくてはならない。ここでは、一定の品質確保する代用特性として、レビューの有無を挙げている。レビューが済みであることは、一定の品質を確保していることを意味している。

（４）管理する項目の粒度をあわせる

抽出した管理項目ごとに作業時間の長さが異なる場合には、できる限り大きさの同じ作業単位に分割する。その分割がうまく同じ大きさにできない場合、それぞれの管理項目に重みをつけることで、管理する項目の粒度をあわせる。こうすることで簡単な項目ばかり作業を終わらせていて、見かけ上の進捗だけが上がっているといった、実態を表していない進捗把握を防ぐ。

（５）担当者の入力が大変でないようにする

この管理表に入力するのは、仕様書を持つメンバである。入力は簡単に、実態に沿ったものを考え、管理表への記入がメンバの負担にならないようする。また、メンバ自身に対しても遅れ進みが分かるようにすることで、モチベーションの向上や、危機意識を持つことにつながるようにする。ここでは、記述の済み/未済みや、レビューの済み/未済みを記入するだけとした。

付録 1 の付表 1 に外部設計工程の原始データの抽出（例）、付表 2 に単体テスト工程の原始データの抽出（例）を示す。

4. 3 チームとプロジェクトへ（加工データの選択）

4.2 の通り、物理空間にある要素から原始データが抽出でき、個人単位の進捗が掴めた。個人単位の加工データをチーム単位や、プロジェクト単位へ抽象化する上位情報層への加工方法を考えていく。図8に個人単位の加工データからチーム単位の加工データ作成例を示す。

チームZ						
aさん				bさん		
画面A				画面X		
項番	分類1	分類2	記述未/済	重み	レビュー未/済	重み
1	画面	画面レイアウト	1	-	1	-
2		画面入力	1	-	1	-
3		画面出力		-		-
4		画面遷移		-		-
5	処理方式	データ取得	1	-	1	-
6		データ更新	1	-	1	-
7		検索処理		-		-
8		エラーチェック		-		-
	進捗度		4	-	4	-
	進捗度		1	-	1	-

チームZ（aさんとbさん）の進捗測定例

1. 個人の進捗度
 - （個人の進捗）

$$= \text{総和（記述済} \times \text{重み} + \text{レビュー済} \times \text{重み）} / \text{（計画した記述} \times \text{重み} + \text{計画したレビュー} \times \text{重み）} \times 100$$
 - aさんの進捗（ここでは説明の簡略化のため、重みは考えない）

$$= (4+4) / (8+8) \times 100 = 50\%$$
2. チームの進捗
 - （チームの進捗）

$$= \text{総和（個人の実績）} / \text{総和（個人の計画）} \times 100$$
 - チームZの進捗

$$= (\text{aさんの実績} + \text{bさんの実績}) / (\text{aさんの計画} + \text{bさんの計画})$$

$$= (8+2) / (16+16) \times 100 = 31\%$$

図8：個人単位の加工データからチーム単位の加工データ作成例

情報層1のメンバ単位の進捗データが決まるので、情報層2のチーム単位の進捗は加算だけで容易に算出できる。原始データから代用特性を抽出した後、情報1から情報nをつくることは、データをどのように使うかである。例えば、進捗は先に述べた通り加算することで得られる。課題は報告させる人に合わせ、選択することで得られる。報告される側が欲する情報によって抽象度を（サブシステム毎、機能毎、画面毎）を変えれば良い。

物理空間にモノ（仕様書など）があれば、情報層1への原始データの値を正確に掴むことができるので、それ以降の情報層への進捗報告の信頼性が保障される。また、物理空間にモノがなければ、進捗の値はゼロであり、正確に掴むことができる。

5. 結論

プロジェクトの情報は、情報層の最下層にある原始データから、上方向に抽象度を上げながら伝達されていく。従って、物理空間の要素の状況を適切に表現する原始データを抽出しなくてはならない。本論文では、図 7 および図 8 を提案した。この方法は、WBS の完了度による進捗状況を数値化する方法である。

これによって、進捗の測り方のブレが小さくなるはずであるので、今まで長年繰り返された実態を正しく反映しない測定値から逃れることができると考える。

また、プロジェクトの関係者は、それぞれの立場で、異なる抽象度と観点の情報を必要としている。観点と抽象度を整理すると、報告情報のモデル化が可能であることを示唆している。本論文の観点と抽象度を実プロジェクトに当てはめて、報告ルールモデルを作り上げてほしい。これにより、報告の精度があがるとともに、省力化が実現できる。

なお、プロジェクトにおける管理情報は、「一所懸命考えたいい加減な仮の計画値」との比較で用いられるので、過度に精緻である必要はない。

以上の如く、PMの悩みが軽減されることを期待している。

参考文献

- [1]板倉稔, 植野俊雄, 齋藤謙二郎, 佐藤光彦: 「スーパーSEによるプロジェクトの解明ープロジェクトマネジメントの本質が見える、わかる、きわめられるー」, 株式会社日科技連出版社, 2003

付録 1

付表 1 外部設計工程の原始データの抽出（例）

項番	分類 1	分類 2	実施	重要度	レビュー	レビュー
			済/未済	重み	済/未済	重み
1	画面	画面表示				
2		画面入力				
3		画面出力				
4		画面制御				
5		画面遷移				
6		帳票出力				
7	機能	データ取得				
8		データ更新				
9		検索処理				
10		計算処理				
11	エラー処理	エラーチェック				
12		例外処理				
13	操作性	操作性				
14		入力補助				
15	インタフェース	接続仕様				
16		ファイル入出力				

付表2 単体テスト工程の原始データの抽出（例）

項番	分類1	分類2	実施	重要度	レビュー	レビュー
			済/未済	重み	済/未済	重み
1	画面表示	大きさ				
2		共通表示項目				
3		初期表示項目（新規モード）				
4		初期表示項目（修正モード）				
5		初期表示項目（削除モード）				
6		初期表示項目（照会モード）				
7		項目表示形式				
8	カーソル	Tab キーカーソル移動				
9		Enter キーカーソル移動				
10		特定項目フォーカス時の制御				
11	ボタン	各種ボタン配置				
12		各種ボタン押下時の制御				
13		ファンクションキーの制御				
14	値	入力時チェック				
15		最大値、境界値チェック				
16		必須チェック				
17		エラーメッセージ				
18	関連	関連チェック				
19		エラーメッセージ				
20	検索	初期検索				
21		検索機能				
22		ソート順				
23		検索結果の件数による制御				
24	帳票出力	出力結果				
25		改ページ				
26		ソート順				
27	DB	更新結果				
28		更新確認ダイアログ				
29		更新時のエラーによるデータ保全				
30		削除結果				
31		削除確認ダイアログ				
32		削除時のエラーによるデータ保全				
33	ファイル	インポート機能				
34		エクスポート機能				
35	計算	ロジックの妥当性				
36		端数処理				
37	バッチ	起動確認				
38		結果確認				
39	その他	排他処理				
40		コーディング規約チェック				