R&D Process improvement / innovation / intelligence

製品開発コンサルティング・レポート

勘と経験に頼らないプロジェクト管理

■ 目 次 ■

第１回　プロジェクトを可視化する重要性
正しい姿を把握することで改善策も見えてくる

第２回　メトリクスが活用できる仕組み作り
必要最小限のデータを収集し基本となるモデルを作成

第３回　メトリクスの活用範囲の拡大
二つの軸でマトリクス組織を管理、個人レベルまで落とし込む

出典：掲載記事は、日経BP社『日経ものづくり』
2005年7月号 p106～109、2005年8月号 p128～131、2005年9月号 p122～125
「勘や経験に頼らないプロジェクト管理　第1回～第3回」から転載。

株式会社 ＲＤＰｉ

連載

勘や経験に頼らない

プロジェクト管理

石橋良造 アジレント・テクノロジー R&Dプロセスコンサルティング

第1回 プロジェクトを可視化する重要性

正しい姿を把握することで改善策も見えてくる

製品の大規模化，複雑化が進展する中，製品開発におけるプロジェクト管理が大きな課題となってきている。今号からは，ますます重要さを増しているプロジェクト管理を，メトリクスを利用することで効果的に進め，そして成功に導く方法について解説してもらう。第1回では，メトリクスを利用する上で大前提となるプロジェクトの可視化の重要性を，事例を用いて考える。（本誌）

製品開発プロジェクトにおける問題は，品質の劣化や生産性の低下，納期の遅れなどとして表面化し，これまでは開発プロセスや組織の問題ととらえ解決を図ってきた。しかし，プロジェクトの問題は多岐にわたっているため解決が難しく，最近はプロジェクト管理の問題としてとらえることが多くなっている。

ただ，多くの組織で，プロジェクト管理の問題をプロジェクト・リーダーの資質や能力の問題として片付けてしまっているのも現実。これではプロジェクト管理の仕組みを改善するには至らない。

このような事態を避けるには，プロジェクトにおける問題のほとんどがプロジェクトが見えていないことに起因していることに気付く必要がある。実態が見えないため，問題を正しく把握することも，適切な対応を取ることもできないのである。

本連載では，プロジェクト管理を効率的に実施する方法として，プロジェクトの本当の姿を把握でき，問題を解決に導くことができるメトリクスを利用した仕組みを，3号にわたって紹介する[*1]。

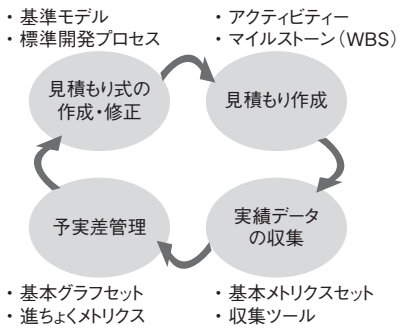

図1●メトリクスを利用したプロジェクトの管理サイクル
すべてのプロジェクトでこのサイクルを実行することが重要。さらに，それぞれのステップで適切な手法・技法を導入することも重要である。

管理のために加工した指標

メトリクスとは，プロジェクト活動（開発活動）を定量化し，そのデータを適切にプロジェクト管理ができるように加工した指標のことである。データを収集しただけではメトリクスではない。メトリクスというためには，プロジェクトをどのようにマネジメントするのか，そのためにどのようにデータを収集・加工するのか，加工したデータをどのように利用するのかが明確になってないといけない。

図1は，メトリクスを利用したプロジェクト管理のサイクルを示したもの。①見積もり式（基準とするメトリクス）で見積もりを作成②開発過程で実績データを収集・加工（実績のメトリクス作成）③見積もりと実績で予実差を管理（進ちょくメトリクス）④開発が終わった段階で見積もり式を

＊1　本連載の著者は，2005年に「ザ・チェンジ」（日経BP社）を執筆している。同書は，日本ヒューレット・パッカード（当時）の業務改革の軌跡を紹介したもので，その中でメトリクスを利用した管理にも触れられている。同書の読者から，メトリクス管理を詳細に解説してほしい，という要望が寄せられたため，本連載の執筆を依頼した。

筆者プロフィール　日本ヒューレット・パッカードに入社し，会社分割後アジレント・テクノロジーに所属。この間，半導体計測システムの研究・開発に従事した後，社内改革プロジェクトを実施した。現在，これらの経験を基にコンサルティングを行っており，これまでに家電，通信，電子機器，自動車業界に数十社の実績を持つ。

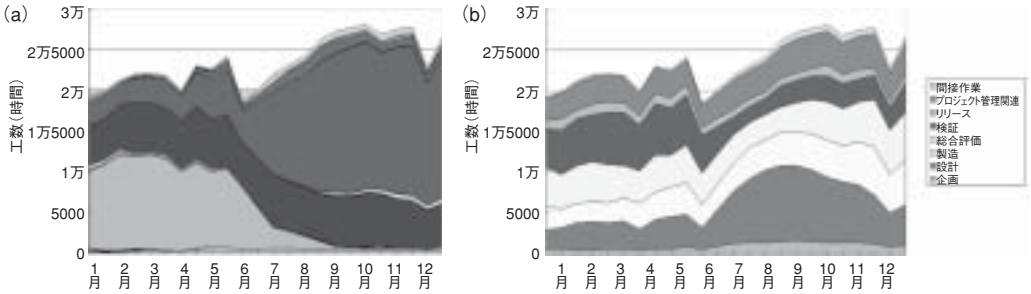

図2●プロジェクト別および開発工程別の工数推移（その1）
大規模・少量のクルマ関連製品開発を行っているメーカーにおける1年間の開発工数の推移。プロジェクト別の開発工数推移では，プロジェクトごとに色を変えて表示している（a）。作業工程別の開発工数推移では，作業工程ごとに色を変えて表示するとともに，それぞれの作業工程の名称も記入した（b）。大規模プロジェクト中心の組織である。

改善――といった流れとなる。

これらの各ステップの詳細については次号でも述べるが，今号では，見積もり作成と予実差管理のステップを例にとり，メトリクスがどういうものかを紹介する。まずは，メトリクスを作る際の基本の考え方である，プロジェクトの可視化について解説したい。

可視化で浮かび上がる問題

プロジェクト管理の問題を解消するためには「見えないものをコントロールすることは不可能」という点を，認識する必要がある。プロジェクトのマネジメントには「見える化」が必須であり，単なるデータは見える化によってメトリクスになるのだ。

では実際に，幾つかのプロジェクトを見える化する過程を紹介しよう。

図2はクルマ関連の製品を開発しているあるメーカーにおける，1年間の週ごとの開発工数*2（単位は時間）を表したもの。

図2（a）ではプロジェクト別に工数推移をまとめた。このメーカーでは，ベースプロダクトを開発する2～3の大規模プロジェクトに開発工数のほとんどが投入されていることが分かる。

一方で**図3**は，同じくクルマ関連の製品開発を行っている別のメーカーの，開発工数推移である。**図3**（a）を見ると，先ほどのメーカーとは違い，非常に多くの小規模プロジェクトが並行して存在していることが分かる。ここまでは全体のプロジェクトの様子が分かるものの，プロジェクト管理には使えない単なるデータである。

これを頭に入れた上で，次に**図2**（b），**図3**（b）を見てもらいたい。各メーカーの開発工数推移を，今度は見方を変えて設計や製造など開発工程別にまとめた結果だ。同様の傾向が多くの繁忙な開発組織で見られるのだが，これらのグラフを見て，何か新しい発見はないだろうか。

多くの人は，企画，設計，製造，評価など各工程が逐次的に進み，その流れをコントロールするのがプロジェクト管理や開発プロセスだと考えているだろう。しかし，**図2**（b），**図3**（b）を見て分かるように，現実は大規模プロジェクト中心だろうと小規模プロジェクト中心だろうと，開発工程は同時並行に実施されている。企画，設計，試作，製造，評価などの作業が常に並行しているわけだ*3。

このような現状にもかかわらず，ど

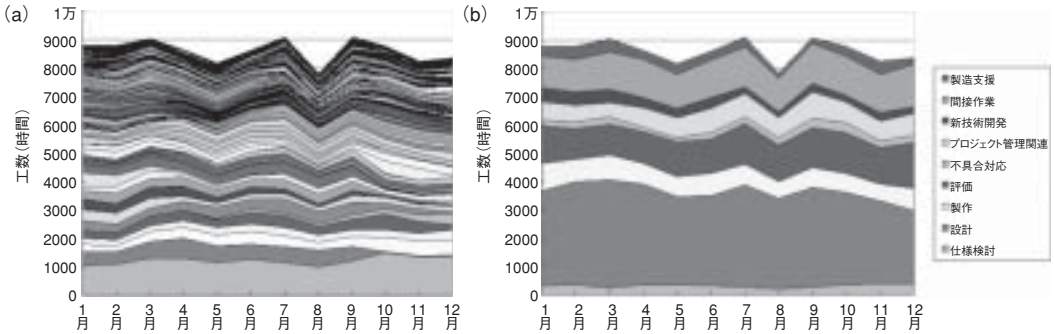

図3●プロジェクト別および開発工程別の工数推移（その2）
小規模・大量のクルマ関連製品を開発しているメーカーの1年間の開発工数の推移。プロジェクト別の開発工数推移を見ると，多くのプロジェクトが並行で実施されている（a）。一方で，作業工程別を見ると，図2の大規模・少量プロジェクトと比較して大きな差がない（b）。

＊2　本連載において「開発工数」というときには，製品仕様検討から製造移管（工場引き継ぎ）までの工数を指す。多くの組織で開発や設計の部署に所属している技術者の工数である。製造技術や工場技術などと呼ばれる技術者の工数は含んでいない。

＊3　厳密には，ここに示したデータだけでは，開発組織全体で個々の開発工程が並行実施されていることしか観測できない。しかし別のデータで，大規模プロジェクトではプロジェクトごとに，そして小規模プロジェクトでは技術要素グループごとに同様の工数推移となっていることが分かるので，開発の基本単位で個々の開発工程が並行実施されていると記述した。

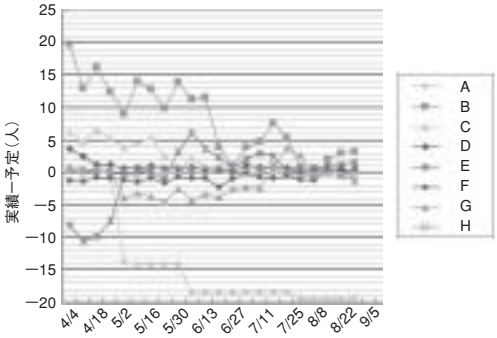

図4●要素技術グループにおけるプロジェクト別工数予実差

開発のある特定部分を受け持っているグループにおいて，プロジェクトごとに実績工数から予定していた工数を引いた予実差を表したグラフ。プロジェクトBやプロジェクトCに予定以上に工数を割いてしまったため，プロジェクトAに予定通りに工数を投入できていないことが分かる。

ちらのメーカーも企画，設計，試作，製造，評価などの各工程が逐次的に進む開発プロセスが規定されているだけであり，すべての開発工程が並行に実施される現実を無視している*4。

大規模プロジェクトでは同時に設計，試作，製造，評価が行われているにもかかわらず，工程移行確認中心の進ちょく会議やデザインレビューが管理の基本となっており，重複した開発工程を同時にコントロールするための手順や方法を規定化できていない。

一方，小規模プロジェクトでは，プロジェクトごとに開発プロセスを規定してもほとんど意味はなく，技術者が複数のプロジェクト向けに同時進行している設計，試作，製造，評価作業を兼務している状態をコントロールする仕組みが求められている。

マネジメントできる姿に変更

ここまでで，見える化により現状が把握でき，どのようなマネジメントが必要なのかが見えてきた。最後の一歩は，マネジメントに結びつける仕組みだ。

実際どのような仕組みを作ればよいか。ここでは，その一つの例を紹介する。

図4はある技術要素グループ*5について，そのグループが関与しているプロジェクトごとに，実績工数と予定工数との差分をグラフ化したもの。プロジェクトの進ちょくを可視化することに加え，回路やソフトといった技術要素グループごとの進ちょくも可視化した。両者をどう活用するのかを示した開発プロセスは別途必要だが，これで，マネジメントがとるべき行動が明らかになった。

以上，同時並行に進行している開発作業の実態を見える化し，さらに，技術要素軸での技術者の過不足という視点を与えることで問題への対応方法を議論できるようにした。これでマネジメントに活かせる指標，メトリクスとなったわけだ。メトリクスにするための基本は，実態を把握でき，マネジメントできるようにするための見える化であることも分かっていただけたと思う。

モデル化できれば精度が高まる

次にメトリクスの評価基準的な側面を紹介しよう。プロジェクトに共通するパターンを抽出し，見積もり基準を与えるメトリクスである。このプロジェクト共通パターンを「基準モデル」と呼ぶ。基準モデルを作ることは，プロジェクトをモデル化することだ。

実例を見てみよう。**図5**は通信関連機器メーカーにおける，いくつかの開発プロジェクトの開発工程別工数比率である。このメーカーでは国内向け汎用機，海外向け汎用機，国内向け特注機の，大きく分けて三つのカテゴリーの製品を開発している。

図5を見ると，カテゴリーごとに開発工程別工数比率に共通の傾向があることが分かる。例えば国内向け汎用機カテゴリーのプロジェクトは，設計が全体の30％程度の工数，評価が全体の35％程度，プロジェクト管理関連が15％程度という共通の特徴を持

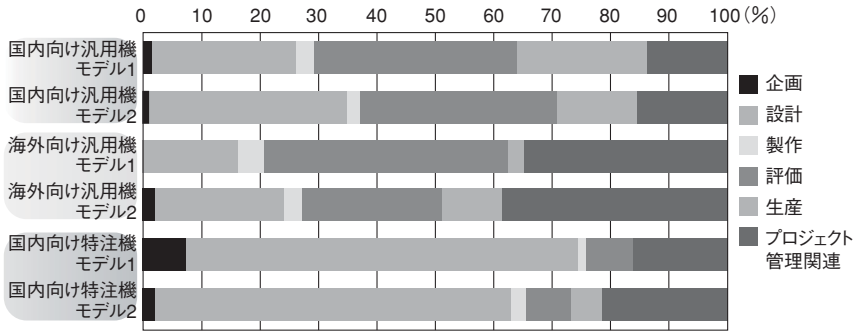

図5●プロジェクトの開発工程別工数の比率

製品カテゴリーは大きく分けて三つあり，プロジェクトの開発工数比率にはカテゴリーごとに共通の傾向がある。

＊4　開発プロセスあるいは標準開発プロセスは，すべての開発プロジェクトが従うべき基本の仕組みで，従うことで効率良い開発を約束するはずのもの。しかし，ISO9001やCMMなどの活動で，開発プロセスが確立している組織でも，このように開発現場の実態とは乖離し，本来の目的を果たしていないケースが多い。実際，この両社ともISO9001などにより確立した開発プロセスを持っているにもかかわらず，プロジェクトは納期に追われ，不具合による手戻り作業に忙殺されている。

＊5　ここでの技術要素グループというのは，開発組織がプロジェクト軸と技術軸のマトリクス組織的な構成となっている場合の，技術軸の一つのグループ（組織）を指す。多くの開発組織では，国内向けとか北米向けといった製品系列によるプロジェクト軸というつながりと，回路，メカ，ソフトといった技術要素系列によるつながりとが存在している。

っている。個々のプロジェクト関係者は，開発メンバーの特殊性や仕様・技術の独自性を強調するにもかかわらずである。

海外向け汎用機カテゴリーのプロジェクトにも共通の傾向が見て取れる。特徴的なのは，プロジェクト管理関連が全体の40％程度と，他のカテゴリーと比較すると多いことである。そして，国内向け特注機カテゴリーの場合は，設計が全体の60％程度と，非常に多くの時間を占めるのが共通の特徴だ。同じカテゴリーに属しているプロジェクトは工数の使い方に共通の傾向を持っていることが分かるだろう。

ミクロ的には個々のプロジェクトはメンバーの経験年数やスキル，技術開発の難易度など，固有の事情を抱えているのだが，マクロ的に見た場合は共通のパターンが存在するのである。マクロ的に見れば，メンバートータルのスキルや開発スタイルなどは大幅に変わることがないからだ。

従って，新しいプロジェクトもこのパターンを踏襲した開発作業となる可能性が高い。つまり，基準モデルを作ることができれば，かなり精度の高い見積もりができるのである。

以上述べたように，データを加工して，見積もりに利用する共通パターンとした基準モデルもまた，メトリクスの一つである。

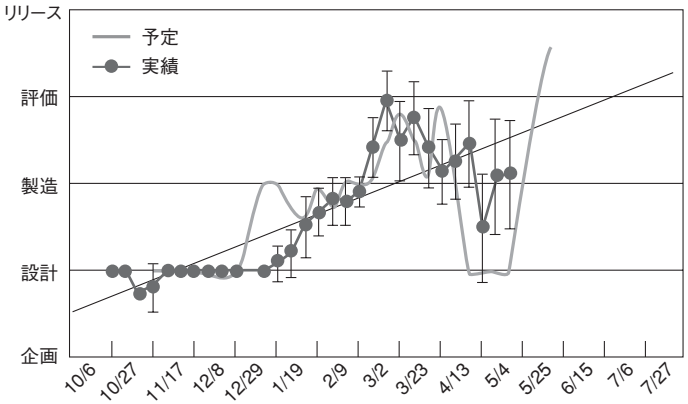

図6●アクティビティセンター
進ちょくを一目で把握するための表現の例。開発作業の中心がどのように推移してきたのかが分かる。予定との比較により，遅れや手戻りも把握できる。

進ちょくは伝える相手を明確に

最後に，進ちょくを見える化したメトリクスを紹介しよう。進ちょくについてはほとんどの組織で何らかの定量化が行われているが，定量化したデータの活用はなかなか進んでいない。

その原因は，進ちょくを伝える工夫が十分ではないことにある。とくに，誰に分かってもらうのかという視点が不足している。

進ちょくを可視化する際に最も重要なことは，トップマネジメント（上級管理者）にプロジェクトの進ちょくを分かりやすく（分かってもらえるように）伝えることである。進ちょくの可視化は，プロジェクト内では解決できない課題に対してトップマネジメントに動いてもらうためのツールなのである。同様にプロジェクトに直接かかわっていないキーパーソンに協力を仰ぐためのツールでもある。

図6は，一つの指標で進ちょくを表現した例である。「アクティビティセンター」とよぶ指標で，トップマネジメントにも一目で進ちょくを把握してもらうことを狙っている。

開発工程を縦軸に取り，横軸には時間軸（**図6**では1週間単位）を取る。その週で実施している作業の広がりを上下の線で表し，その工数重心をプロットすれば，時間経過とともに開発作業の進み具合が分かる。全体の開発工程の中で，プロジェクトの過去と現在が分かるわけだ。

さらに，予定線を加えて予定と実績とを比較できるようにすれば，作業が計画通りに進んでいるのかどうかも一目で把握できるようになる。**図6**を見ると，12月末ぐらいからプロジェクトが遅れたこと，そして，予定も実績もグラフが蛇行していることから，計画段階から手戻りがあることを想定しており，実際にも手戻りが発生していることが分かる。計画段階から非効率な進め方になっていたわけだ。

＊　＊　＊

今回は，メトリクスとは何か，そしてプロジェクト管理にメトリクス*6を利用することがどういうことなのかを，実例を使って紹介した。次回は，このようなメトリクス管理の仕組みをシステマチックに構築するための方法を紹介したい。今回同様，なるべく実例を紹介しながら解説する予定だ。ⓜ

＊6　プロジェクト管理のメトリクスは，プロジェクトを一つのエンティティとしてみたときの，インプットとアウトプットを定量化したものをそろえるのが基本である。これを基本メトリクスセットという（詳細は次号）。今回はその中でも，プロジェクトにとってインプットである開発工数を中心に紹介した。

連載

勘や経験に頼らない

プロジェクト管理

石橋良造 アジレント・テクノロジー R&Dプロセスコンサルティング

第2回　メトリクスが活用できる仕組み作り

必要最低限のデータを収集し基本形となるモデルを作成

メトリクスは，マネジメントがアクションを取ることができるような管理指標であり，うまく活用することでプロジェクトの質を大幅に向上できる。今号は，メトリクスを利用したプロジェクト管理の仕組みをどのように構築するかを解説する。その仕組みの土台となるのは五つのフレームワーク。これらの中身を正確に把握した上で，プロジェクトの実行時に取り込む必要がある。（本誌）

前号ではメトリクスの概要，プロジェクトの可視化の重要性を説明した。本号は，メトリクスを利用したプロジェクト管理の仕組みを構築する方法を解説する。①プロジェクト管理サイクル②WBS[*1]とアクティビティー③基本メトリクスセット④基準モデル⑤予実差管理――の五つ（基本フレームワークと呼ぶ）を，プロジェクト管理の仕組みに組み込むのだ。

プロジェクト管理サイクル

メトリクスを使ったプロジェクト管理サイクルは①基準モデルを使って見積もりを作成する②可視化のための実績データを収集する③予定と実績の差を進ちょくメトリクスによって管理する④開発終了後，基準モデルを改善する――という流れだ。

大切なのは，このサイクルをすべてのプロジェクトで実行すること，サイクルの各ステップで適切な方法論，技法・手法，ツールを使うことだ[*2]。徒手空拳で仕組み作りをすることは可能だが，大変な時間がかかる。まずは，技法・手法を導入し，確実に仕組みを運用できるようにするべきだ。

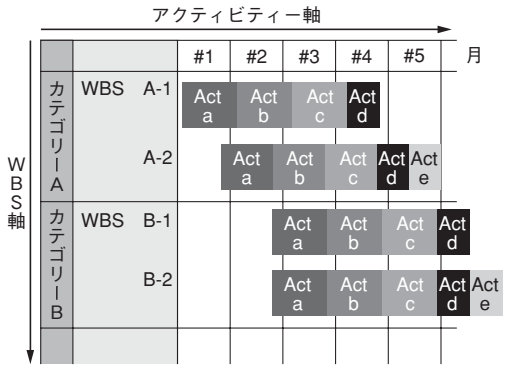

図1●アクティビティーとWBSによる2次元管理
開発作業は「アクティビティー」と「WBS」の2軸で特定できるように分解する。見積もりや進ちょく管理に関してもこの2軸で実施する。

WBSとアクティビティー

実績データを収集するのは手間がかかる。このため，収集するデータを特定するための軸を適切に設定することが重要となる。

では，手間を最小限にして最大限にデータを活用できる軸とは何か。それが，WBSとアクティビティーである。WBSとアクティビティーの2軸により開発作業を特定するわけだ（図1）。WBSは開発作業の範囲を特定する軸で，進ちょく管理単位となる。アクティビティーは開発作業工程，つまり作業の流れを特定する軸である。

最初にアクティビティーについて解説しよう。開発作業の流れは大きく，企画，設計，製造，評価などの開発工程で表される。それを分解したものがアクティ

＊1　WBS　Work Breakdown Structure。プロジェクト管理において，遂行に必要なあらゆる作業を洗い出し，プロセスや責任範囲・管理範囲といった管理しやすい単位で階層化して表現したものを指す。

＊2　後述するが，①のステップではアクティビティーやWBS，②のステップでは基本メトリクスセットや収集ツール，③のステップでは基本グラフセットや進ちょくメトリクス，④のステップでは基準モデルや標準開発プロセスといった技法・手法がある〔前号の『日経ものづくり』2005年7月号図1（p.106）参照〕。

＊3　本来，開発工程は共通の作業をひとかたまりにしたものであり，時間の要素を含まない。しかし，一般的には開発は工程ごとに流れ作業で実施されることを前提としているため，開発工程は作業の流れを表現することになる。

筆者プロフィール　日本ヒューレット・パッカードに入社，会社分割後アジレント・テクノロジーに所属。この間，半導体計測システムの研究・開発に従事した後，社内改革プロジェクトを実施した。現在，これらの経験を基にコンサルティングを行っており，数十社の実績を持つ。http://www.agilent.co.jp/find/kaihatsu/を参照。

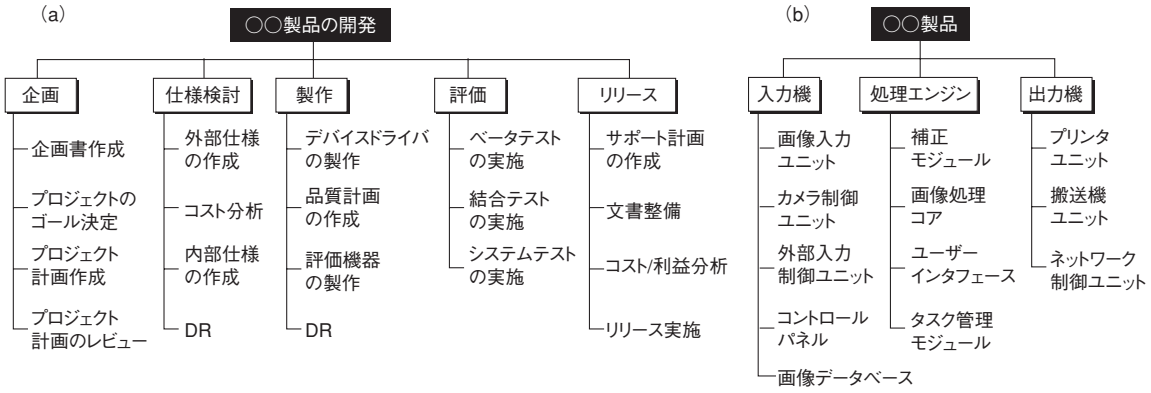

図2●WBSの考え方
一般のWBSの場合，開発対象と開発行為を同じレベルで扱っているため，進ちょく管理には向いていない（a）。一方，メトリクスの仕組みでは，WBSは開発対象のみに限定して作成し，製品の内部構造と一致させる（b）。

ビティーである[*3]。通常は3レベルくらいまで分解する[*4]。

アクティビティー軸を明確にするには，開発工程をしっかりと定義しておく必要がある。開発対象と独立して，共通に定義できるものを抽出する。開発工程の定義には，個々の作業の説明のほか，作業の目的，開始条件，終了条件，入力物，出力物，責任者などを含めておく。

次にWBSとは，開発作業全体を階層的に詳細化した構造のことである。分解した一つひとつの要素をWBS要素，または単にWBSと呼ぶ。WBS要素を積み重ねると開発作業全体になる。開発対象の分解の仕方で，進ちょくを管理する単位が決まるため，WBSの作成には十分に注意が必要だ。

PMBOK[*5]などの普及で，WBSの重要性は広く知られてきた。しかし，どういう単位で進ちょくを管理するかが明確になっていないプロジェクトは多い。

WBSは進ちょくの単位なのだが，運用するには注意が必要だ。メトリクス管理でのWBSは，一般的なWBSと考え方が少し違う。一般的なWBSでは，作業の分解の方法が明確でなく，**図2（a）**に示すようなものが多い。「デバイスドライバの製作」というWBS要素があるが，製作（コーディング）という作業行為とデバイスドライバという作業対象とが一緒で，進ちょく管理単位としては使いづらい。この場合，デバイスドライバの開発進ちょくや，製作作業の進ちょく度合いを知ることは困難だ。

メトリクスの仕組みでは，作業行為を含まず作業対象だけにしてWBSを作成する。前述の例では「デバイスドライバ」となる。開発対象だけに注目してWBSを作成したのが**図2（b）**。WBSは製品の内部構造と一致する[*6]。

このように，WBSとアクティビティーという相互に独立した軸でデータを特定することで，データが最大限に活用できる。例えば開発工数なら，製品構成上の任意の範囲でかかった工数が分かる。**図2（b）**に示す製品全体はもちろん，処理エンジン部分だけの工数も分かるし，処理エンジンの一部である画像処理コア部分だけの工数も分かる。また，入力機などの任意の範囲で，発生している工数が設計なのか製造なのかの把握も可能だ。さらに，評価に時間が掛かっている部分（WBS

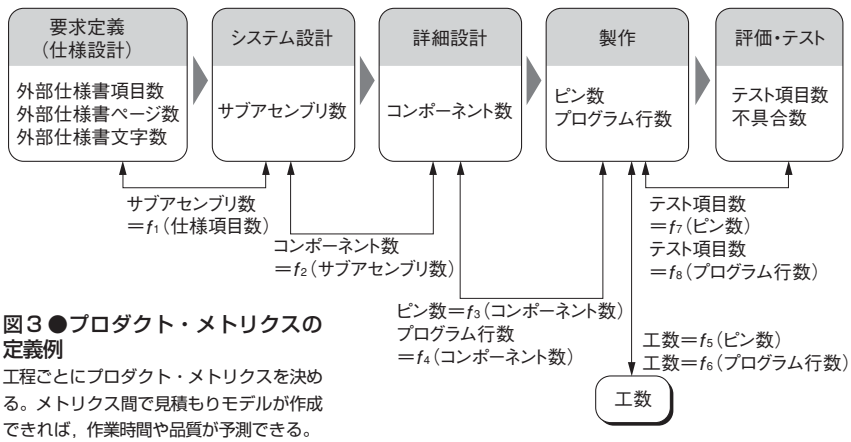

図3●プロダクト・メトリクスの定義例
工程ごとにプロダクト・メトリクスを決める。メトリクス間で見積もりモデルが作成できれば，作業時間や品質が予測できる。

＊4　例えば，設計工程（第1レベル）であれば，第2レベルは方式設計，基本設計，詳細設計などとなる。そして，詳細設計を例に取れば，第3レベルは，回路設計，回路レビュー，機構部品設計，実装設計などとなる。開発工程をこのように定義することで，「設計－詳細設計－回路設計」というアクティビティーで開発作業を特定できるようになるわけだ。

＊5　PMBOK　Project Management Body of Knowledge。米Project Management Instituteが作ったプロジェクト・マネジメントに関する知識体系。

＊6　WBSは幾つかのカテゴリーごとに作成する。内部構造と一致させるWBSはその中の一つである。これ以外に，外部イベント，内部テーマ，開発環境，アクティビティー依存のカテゴリーがある。

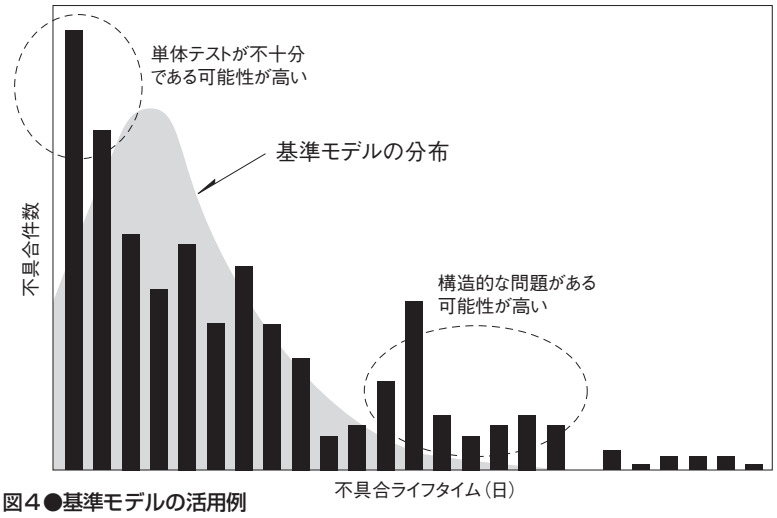

図4●基準モデルの活用例
不具合ライフタイムに対する不具合の度数分布を取ると，成功しているプロジェクトでは基準モデルのような分布となる。この基準モデルと比較することで，現状分析ができる。

要素）はどこか，手戻りが多いチーム（WBS要素）はどこかなども分かる。

基本メトリクスセット

メトリクス管理の仕組みづくりの際に陥りがちな過ちの一つが，取れるデータは何でも取ってしまい，手間がかかること。そうならないためには，プロジェクトの開発活動を特定する必要最小限なデータを知っておく必要がある。適切なプロジェクトのインプットとアウトプットを選ぶことで，プロジェクトの振る舞いが把握できる。

このようなデータ一式を基本メトリクスセットという。プロジェクトに対するインプットとしては**開発工数**，アウトプットとしては**WBS要素**，**作業成果物**，**不具合**の三つ，合計4種類のデータ（メトリクス）がある。

①開発工数メトリクス

開発工数を収集することにより，開発のどの作業にどれだけの時間をかけているのかが分かる。WBSとアクティビティーの2軸を使うことで，作業進ちょくなどプロジェクトの振る舞いをさまざまな視点から把握できる重要なメトリクスとなる。

②WBSメトリクス[*7]

WBS要素を集計することで，作業項目数（WBS 要素数）で進ちょくを把握できる。WBS要素ごとに作業の着手予定日や完了予定日なども管理しておけば，いろいろな見方ができる。例えば，遅れを生じている作業項目（WBS要素）の個数により，遅れの度合いを把握できるなどだ。

③プロダクト・メトリクス

開発の各工程における作業成果物のサイズ（量）を計測すれば，各工程の作業進ちょくが分かる。ただし，計測対象を決めるのは簡単ではない。

作業成果物は開発工程で変わる。従って，開発工程が明確に定義されている必要がある。その上で，①開発工程ごとに作業進ちょくに合わせて作成される作業成果物を洗い出す②その作業成果物が一定のフォーマットやルールに基づいて作成されることを検討する③その作業成果物のカウント方法を検討し実現する――という手順を通じて実際のメトリクスを決める。

図3は工程ごとの管理指標の例である。開発工程定義によって，各工程での作業や入力物，開始条件，終了条件，出力物などが明確なら，各工程での成果物を特定するのは難しくない。ただ，作業成果物をカウントする際のバラつきをなくすため，それぞれの作業成果物のフォーマットや作成手順が標準化されている必要がある[*8]。

「サブアセンブリ」が作業成果物となっているが，サブアセンブリは回路ブロックやソフトウエアモジュールの

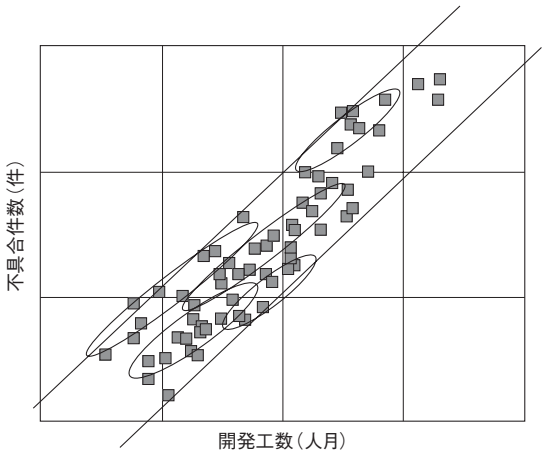

図5●開発工数と不具合件数
両者には正の相関があることが分かる。この関係から不具合対応時間の予定が立てられる。

＊7　WBS要素ごとに作業完了予定日を定義すれば，WBS要素は作業完了の予定や実績を表すことになる。従って，WBS要素は開発のマイルストーンと考えることができ，WBSメトリクスはマイルストーン・メトリクスとも呼んでいる。

＊8　さらに，決めた管理指標（プロダクト・メトリクス）が，プロジェクトの入力である開発工数と相関を持つことを検証する必要がある。例えば，ソフトウエア開発の世界ではさまざまな議論があるが，ステップ数を代表的なプロダクト・メトリクスとしているのは，工数との相関が高いためである。

ことで，ほかとのインタフェースが明確になっていることや一定の表記方法があることなどが前提条件だ。「コンポーネント」なども同様である[*9]。

開発工程ごとの管理指標が決まれば，管理指標間の関係式ができる。管理指標間は一定の相関を持つことが多い。そして**図3**のように，すべての工程で管理指標間の関係式が作れれば，ある工程の作業成果物を計測することで，それ以降の工程の作業成果物を見積もれる。最終的には必要工数まで見積もれる。

④不具合メトリクス

評価やテストにより検出した不具合を計測したもの。不具合として記録する属性の中に，起因工程，障害部位などを含んでいることが大切である。不具合は前述のプロダクト・メトリクスの一つだが，品質などの非常に重要なデータとなるため別扱いしている。

基準モデル

基準モデルについては前号で，開発工程別の工数比率が製品カテゴリーによって共通のパターンになることを紹介した。共通パターンを抽出できれば，見積もりの際に参照でき，経験と勘による属人的な見積もりの仕組みを組織的なものへと変えられる。

個々のプロジェクトには，メンバーの経験年数やスキルなど固有の事情，特徴があるが，マクロ的に見れば，開発グループのスキルや開発スタイルなどは大きく変わらない。現実の開発現場では，開発する製品も開発者も短期間に大幅に変わることはないからだ。従って，基準モデルを作るには，組織の中で成功したプロジェクトを選別し，そのデータを分析して共通パターンを見つければよい。

成功したプロジェクトのパターンを作ることで，その組織における成功パターンを定量化したことになる。個々のプロジェクトはそれを参考に計画を作成し，また，それを基準にプロジェクト進ちょくを比較すればプロジェクトの悪さも分かりやすい。

図4は不具合について，検査などで発見した日から修正される日までの期間を調べ，その期間ごとの不具合件数分布を表したものである[*10]。開発工程を適切に管理できているプロジェクトでは，基準モデルに示す分布となる。この基準モデルとの差異を生じている部分についてその原因を推測することが可能であり，プロジェクトの進め方にフィードバックできる。

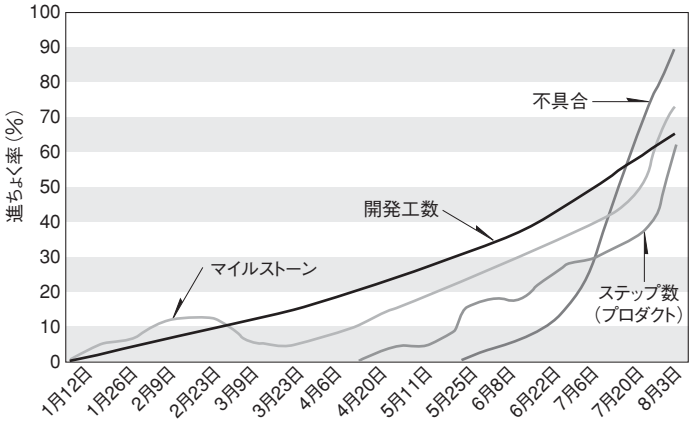

図6●進ちょく率の比較

予定差による進ちょく率を基本メトリクスごとに相互比較すると，プロジェクトの進ちょく状況を総合的に把握できる。ここでは，それぞれの最終予定値に対する実績値の割合を進ちょく率としてグラフ化した。開発中盤のかなり長い期間で，投入している工数と比較して，作業（WBS）は遅れていたことや，開発終了間際になって作業（WBS）の遅れをキャッチアップしているが，品質が急激に悪化していることなどが分かる。

図5は開発工数に対する不具合件数である。これは複数の組織でのプロジェクトをまとめてプロットしたものだが，これから，開発工数と不具合件数とは正の相関があることが分かる。楕円（だえん）で囲んでいる部分は同一組織におけるプロジェクトである。開発工数を見積もることで，不具合時間を見積もることができるのが分かる。

予実差管理

進ちょく管理の基本は予定と実績との乖離（かいり）を見ること，すなわち予実差管理である。これまでに述べたメトリクスを，予定と実績の乖離が分かるようにして可視化するということだ。

図6は基本メトリクスセットの一つひとつに対して予実差をグラフ化したもの。一つのメトリクスだけを見るよりも，プロジェクトの振る舞いがより把握できる。プロジェクトに対する複眼的な視点が持てる。

次回はスコープを少し広げて，メンバー管理など進ちょく管理以外でのメトリクスの活用について紹介する予定である。

＊9　要求定義工程における代表的な作業成果物が「外部仕様書項目数」などとなっているが，これは適切な作業成果物とはいえない。通常，「外部仕様書」は，内容の表記方法は決まっておらず，文章や図を使ったフリーフォーマットで計測対象があいまいだったり，計測方法が標準化できていなかったりするからである。

＊10　不具合を発見してから修正するまでの期間を「不具合ライフタイム」と呼んでいる。不具合が生きている期間である。

連載

勘や経験に頼らない

プロジェクト管理

石橋良造 アジレント・テクノロジー R&Dプロセスコンサルティング

最終回 メトリクスの活用範囲の拡大

二つの軸でマトリクス組織を管理 個人レベルまで落とし込む

メトリクスを利用したプロジェクト管理の仕組みを解説してきた本講座も最終回。メトリクスはプロジェクト活動を定量化した上で，加工した指標のこと。マトリクス組織における開発進ちょく管理や，個人の行動を改善するためにも活用できる。マトリクス組織では複雑な管理を強いられるため，メトリクスを使用する効果が高い。また，個々の役割を明確にするためにも，メトリクスによる可視化は重要だ。（本誌）

前号では，メトリクスを利用したプロジェクトの進ちょく管理の仕組みを構築する方法を解説した。最終回の今号では，ここから少し範囲を広げたメトリクスの仕組みとして①マトリクス組織における開発進ちょく管理②個人の行動（振る舞い）へのフィードバック―― の二つを紹介したい。

マトリクス組織での仕組み

多くの開発組織では，プロジェクト軸と技術要素軸によるマトリクスの組織構成を取っている（**図1**）。プロジェクト軸は，高機能型と普及機，あるいは国内向けと北米向けというような製品系列による分類。一方，技術要素軸は電子回路，機構，ソフトといった，開発に必要となる技術による分類である。マトリクス組織では，この両方の軸に責任者を置き，技術者は両方の軸の責任者から指示を受け，そして進ちょくを報告する。

マトリクス組織にする理由は，プロジェクトごとに技術者を専任化したのでは製品開発をこなせないからである。技術者に同時に複数プロジェクトを担当してもらうことで，従来の技術者数で今まで以上の開発プロジェクト数をこなしたいのである。

従って，マトリクス組織での開発は，通常よりも精密で複雑なプロジェクト管理が要求され，そのためメトリクスによる進ちょく管理の仕組みが非常に有効なのだ。以下に，マトリクス組織におけるメトリクスを利用した管理の仕組みを紹介する。

①役割分担の明確化

最初にすべきことは，プロジェクト

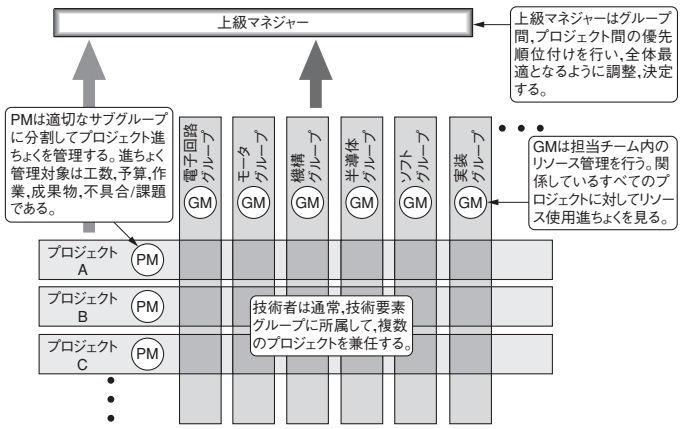

図1●プロジェクト軸と技術要素軸のマトリクス組織 プロジェクト単位の責任者（PMもしくはPL）と技術要素単位の責任者（GMもしくはGL）が存在し，技術者は両方の管理者の下で開発作業を実施する。組織上はGM/GLが直接上級マネジャーにつながっていることが多い。

＊1 マトリクス組織は，プロジェクト軸と技術要素軸のどちらが組織的に強いのかで，プロジェクト主導マトリクス組織と技術要素主導マトリクス組織に分類される。プロジェクト主導と技術要素主導とでは，同じマトリクス組織であっても各人の役割や責任は違うが，この違いを深く考慮せずマトリクス組織にしてしまうことが多い。そのため，プロジェクト・マネジャー（PM）制を導入してPMに多くの責任を持たせているにもかかわらず，技術要素主導マトリクス組織のため

筆者プロフィール　日本ヒューレット・パッカードに入社，会社分割後アジレント・テクノロジーに所属。この間，半導体計測システムの研究・開発に従事した後，社内改革プロジェクトを実施した。現在，これらの経験を基にコンサルティングを行っており，数十社の実績を持つ。http://www.agilent.co.jp/find/kaihatsu/を参照。

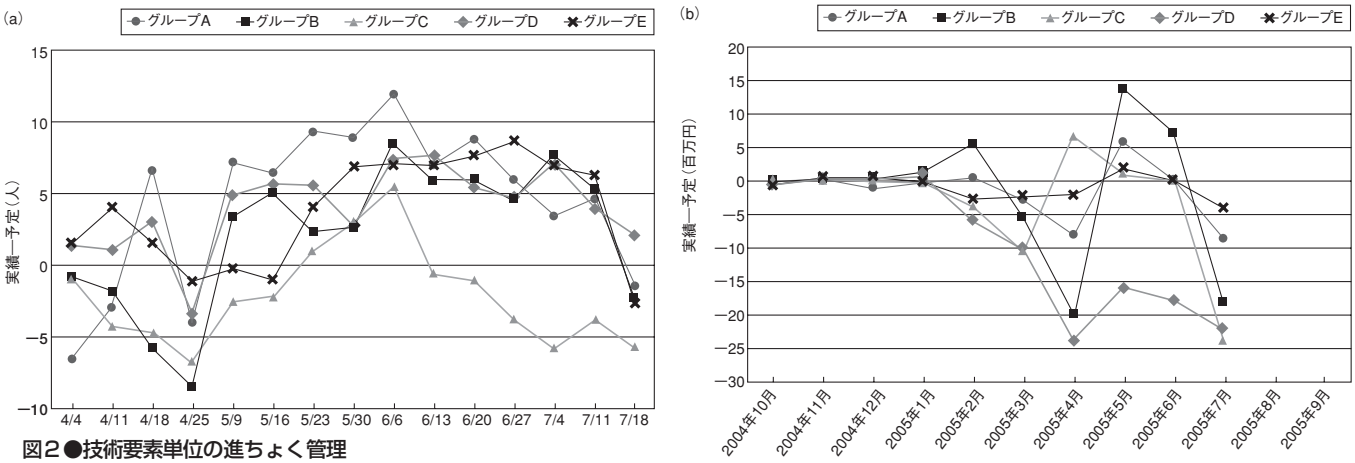

図2●技術要素単位の進ちょく管理

グループごとに，毎週のメンバー全員の工数（a）と毎月の開発費執行状況（b）を予実差として表現した。工数予実差では，ほとんどのグループで実績が予定を上回っていることからオーバーワークな状態であることが分かる。ただし，グループCだけが定常的に予定工数を投入できておらず，異常な状態である。原因を特定した上で，今後の対策を計画することが必要である。開発費予実差では，全体的に予定通りに開発費を使えていない傾向が分かる。このような場合，上位方針などさまざまな理由から予算執行を止められ，今後開発遅れとなって問題が顕在化するケースが多い。

軸と技術要素軸それぞれにおける役割分担の明確化である[*1]。プロジェクト単位にはプロジェクト・マネジャー（PM）もしくはプロジェクト・リーダー（PL）という責任者，技術要素単位にはグループごとに責任者〔グループマネジャー（GM）もしくはグループリーダー（GL）〕がいる。PM/PLとGM/GLの役割を明確にすることが重要だ。

プロジェクトの進ちょく管理やプロジェクトのQCD（品質・コスト・納期）はPM/PLの責任，技術者などのリソースのアサインと調整，担当している技術要素の開発効率と品質の改善はGM/GLの責任と考えるのが基本である。

②進ちょく管理

前号まで解説したプロジェクト単位の進ちょく管理に加えて，技術要素単位の進ちょく管理を実施する。具体的には，技術要素グループごとのリソースについて管理することになる。

ある開発組織における予実差グラフを示したのが**図2**。このように技術要素グループごとに抱えている技術者の工数や開発予算の投入について予実差を見るのは重要だ。予実差は予定と実績の比（除算）ではなく差（減算）にしている。これは，リソース管理では工数にしてもコストにしても絶対値が重要だからである。組織全体で見たときにどのグループに問題があるのかが明白になる。

図3には，あるグループにおけるプロジェクト別の工数と開発費の予実差を表示した。技術者の1人月当たりの費用を100万円とすることで，工数と開発費の合計を同じスケールで見ている。GM/GLが適切に対応できていれば，グラフ上のすべてのプロジェクトについて，その予実差が減少する[*2]。

③予定の更新

マトリクス組織での進ちょく管理では，プロジェクト軸でも技術要素軸でも予定を常に更新することが重要となる。開発の最中は，日々のトラブル対応に時間も意識も取られがちで，日々のトラブルを解決できていたとしても，これはリアクティブ（受け身）な管理であり，管理レベルは低い。

常に現状をベースに最新の見通しを立て，それを関係者と共有し，事前調整するスタイルに変えることが大切だ。これがプロアクティブ（前向き）な進ちょく管理であり，トラブルによる手戻りや調整作業を減らし開発効率向上につながるマネジメントである。事前調整するための具体的なアクションの一つが，可視化された予定を常に更新することである。

あるプロジェクトについて予定工数が四半期ごとに更新されている例を**図4**に示す。工数以外についても同様である。また，技術要素軸でも同様に予定を更新する。予定を常に更新するこ

PMに権限がほとんどないような組織も多い。このような組織ではPMは苦労の連続でボロボロになってしまい，やりたいと思う人はいなくなる。「うちには人材がいない」というが，人材の問題ではなく仕組みの問題なのである。

＊2　基本的には，予定に合うように技術者のアサインを変更したり購入品や外注の契約を進めたりするか，現実に合うように予定を変更するかのどちらかである。さらに，技術要素単位で予実差が解消されても，プロジェクト単位で見たときに予実差を生じているのであれば，PM/PLが計画変更や技術者の変更をGM/GLに要求するはずである。このように，PM/PLとGM/GLの役割分担は相互にチェック＆バランスが利くようにするのが基本である。

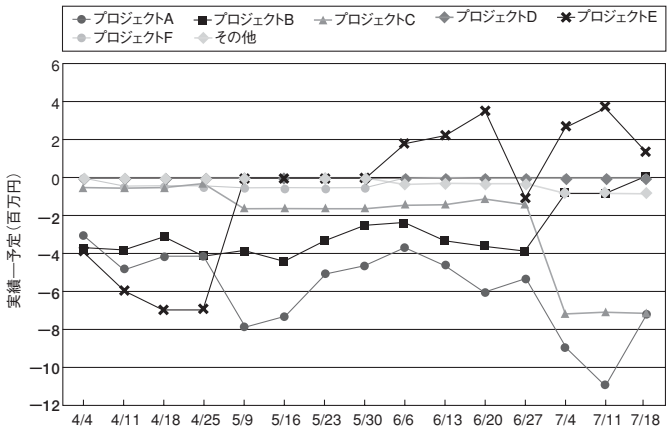

図3●グループに着目した開発コストの予実差
問題を抱えているグループCに着目して，開発工数と開発費の合計予実差を表示した。プロジェクトAは，継続してリソースがアサインできていないこと，プロジェクトCも同じ状態になりつつあることが分かる。原因の一つはプロジェクトEの予定超過だが，その影響は限定的で，構造的な問題ととらえるべきである。

とで，今後の見通しのギャップが客観的，定量的に把握でき，議論できるようになる。今後の見通しについて議論できれば，プロアクティブなプロジェクト管理実現の第一歩だ。

④進ちょく管理スキルの評価

最終的に予実差が小さいとは，予定と実績とをグラフ化してみて，二つのグラフで囲まれる面積が小さいということである。進ちょく管理のスキルはこの面積の大きさで評価する[*3]。

進ちょく管理のスキルにはさまざまな要素が含まれるが，見積もりの質も含めて予実差の規模で評価するのが分かりやすい。実際，この予実差を小さくできる開発組織では計画が信頼できるため，製造や営業などの他部門との交渉や調整を優位に進められ，開発マネジメントの好循環を生んでいる。

個人の行動を改善

これまでも何度か役割分担の重要性を説いた。プロジェクトを可視化してコントロールできる仕組みを作ったとしても，開発プロジェクトの中で個人の役割定義が明らかになっていないと，自立的に問題解決のためのアクションが取れないからだ。

ここでは，役割に応じた個人の振る舞いに注目したメトリクスの活用方法として，アクティビティー・プロファイルを紹介する。

アクティビティーは，開発における必要な役割を分類・定義し，その役割ごとに必要な開発作業を定義したもの[*4]。開発関係者一人ひとりが，自分の役割は何で，どういうアクティビティーを要求されているのかが分かるものになっている。

従って，アクティビティーを使って個人が役割に合った仕事を本当に実施できているかどうかを評価し，フィードバックできる。それをメトリクスとしたのが，アクティビティー・プロファイルである。**図5**はあるPLを対象に，半年の工数を集計し，アクティビティーごとにかけている工数の割合をグラフ化した例である。PLなどの役割ごとに担当すべきアクティビティーが明確になっていれば，担当作業と本来担当すべきではない作業とが識別できる。

開発プロジェクトでは日々の問題を解決することや開発作業そのものに忙しく，マネジャーなどの第三者はもちろん，自分自身でもどのような時間の使い方をしたのかは分からないことが多い。実際，同じPLという役割でも時間の使い方には大きな差があることが分かる。ここでも重要なのは，基準（予定）と実績の差に注目し，次のアクションにつなげることだ。

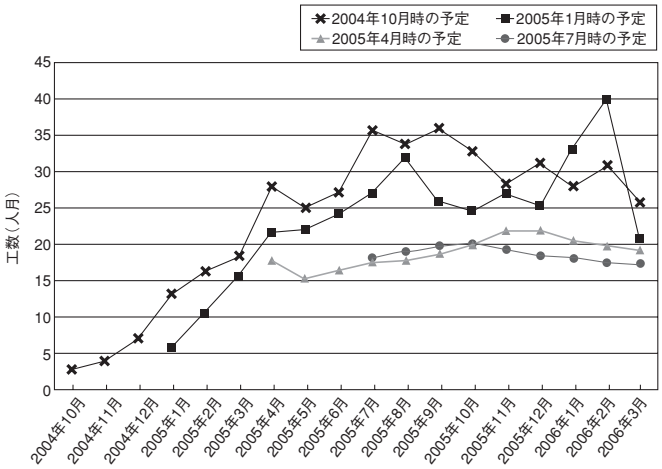

図4●予定の変化
あるプロジェクトを例にして予定工数の変化を表示したもの。3カ月ごとに見直すことで，予定工数は少なくなり，変更の幅が小さくなっていることが分かる。見積もりを繰り返すことで，精度や確度が上がり誤差が小さくなる。この例では，2004年10月時および2005年1月時に予定した技術者の人数よりも，2005年7月時に予定した人数が大幅に減ったため，確保していた技術者を他のプロジェクトへアサインできることが分かる。

＊3　見積もりの正確さを表す指標としてEQF(Estimating Quality Factor) と呼ばれるものがある。これは，時間軸上のその時々でプロジェクトの総工数（総コスト）を見積もり，その見積もりの軌跡と最終的な実績値との間の面積の逆数をとったものである。ここでは，EQFとは違う指標となっているが考え方は同じである。

＊4　アクティビティーについては，前号の「WBSとアクティビティー」を参照。

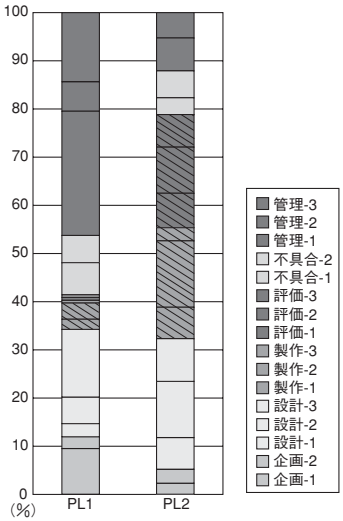

図5●アクティビティー・プロファイル
ある二人のPLについて，アクティビティーごとに半年の工数を集計したもの。本来の業務を実施しているかどうかをかを評価する。斜線を引いた部分は，本来PLが担当すべきでない作業。PL1は10%以下しか実施していないが，PL2は50% 近くも実施しているのが確認できる。

そのほかにも，前述のマトリクス組織において，プロジェクト軸と技術要素軸の交点に属するリーダーのアクティビティー・プロファイルを並べてみても新たな知見が得られるであろう。リーダーによって，業務がプロジェクト管理中心なのか，設計・製作・評価中心なのか，バラついていることが分かるはずだ。

実際，役割/アクティビティーを定義するのが一番難しく，そして，一番トレーニングを必要としているのはこのクラスである。業務定義とトレーニングにより，このようなバラつきを減らす必要がある。

重要なミクロの改善

開発プロセスの構築や改善は個人レベルの行動を変えることなしに効果に結びつけることは難しい。そのため，個人の行動に注目したのが，アクティビティー・プロファイルである。最後に，開発プロセス改善の考え方について解説する。

採用する方法論や技法はISO9000などさまざまだが，開発プロセスの構築や改善は開発業務に対してマクロ的な視点で行うことがほとんど。組織間，あるいは，グループ間の仕事の流れとインタフェースを決め，さらに，組織（グループ）内のアウトプットや作業完了条件などを定義することが主だ。

この場合，対象としているのは組織レベルの業務である。これをマクロプロセスの改善活動と呼んでいる。必要なことには間違いないが，実際に開発効率向上などの成果を上げるのが難しいのも現実だ。図6に示すように，組織やその業務という形のないものを改善対象としているため，実施責任がどこに（誰に）あるのかが希薄で，業務規定などの文書が主な成果となってしまうことが多い。

開発に対して本当の成果を出すには，別の視点でプロセス改善を実施する必要がある。これをミクロプロセスの視点でのプロセス改善と呼ぶ。ミクロプロセスは，技術者の実際の行動に注目してその行動を変えることが狙いであり，活動対象，そして活動主体は技術者個人である。開発者個人の行動を変えることで，開発効率向上などの実のある成果を上げる。アクティビティー・プロファイルのように，メトリクスはミクロプロセスを改善するためのツールの一つになる。

＊　＊　＊

3回にわたって，メトリクスを利用したプロジェクト管理について解説してきた。メトリクスの仕組みは，考え方や技法は共通だが実装は固有である。実際にデータを取り，分析することを繰り返すことで，使える仕組みとなるので，ぜひ自身でアクションを起こしてほしいと願っている。

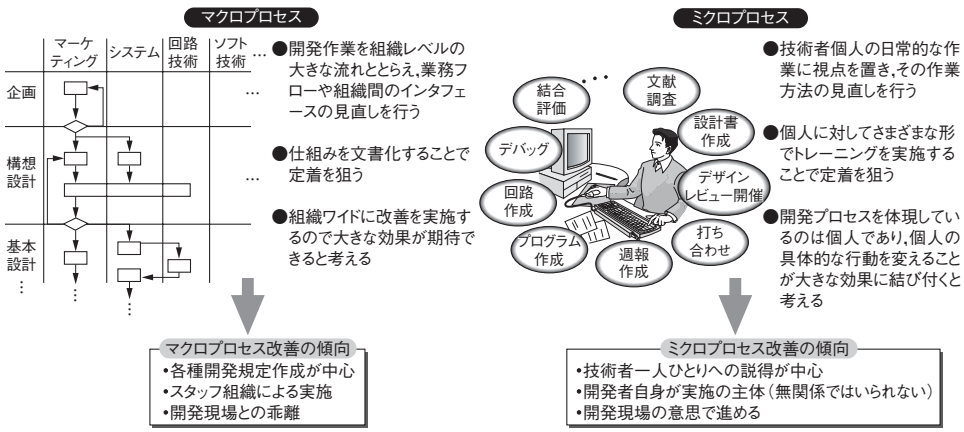

図6●マクロプロセスとミクロプロセス
技術者の「振舞い」を変えるのが，開発プロセス改善の本当のゴール。必要となるのはミクロプロセスの改善である。