大規模開発のプロジェクトマネージメント

Project Management for Large-Scale Development

坪　内　　淳

要　約　プロジェクト管理には様々な方法論が存在し，管理体系も確立している．大規模プロジェクトでも管理体系と方法論については変わりないが，大規模ゆえに遂行上の障壁が多く，これを乗り越えるためにプロジェクト管理にいくつか加味しなければならい点がある．

まずプロジェクトの計画段階では，作業の大きさを考えてプロジェクトを管理できる実行単位へ細分化することが必要であるが，この際実行単位ごとにプロジェクト目的を細分化することも重要である．

次に実行段階では，進行上の問題と課題の把握と分析，対策検討に極力傾注することが重要であり，プロジェクトの状況に応じて柔軟に管理のやり方を見直すことも求められる．これらの柔軟な運営を実現するためにはプロジェクト管理を顧客と一体となった体制で実施することが効果的である．一体となって運営することで，課題解決のスピードアップや品質の向上を期待することができる．

Abstract　There are various methodologies regarding the project management, and its management systems are established also. These methodologies and management systems also can be applied to a Large-Scale project without any changes. But various barriers reside in a Large-Scale project. In order to overcome these barriers, there are points which should be taken into consideration in project management.

In the planning phase of a project, it is required to divide the project into executable and manageable units by considering the size of work in a project. At the same time, it is also important to divide the project purpose into units which correspond to executable units divided formerly.

In the execution phase of a project, it is important to concentrate on the analysis and grasp of the ongoing problems and challenges as well as the study on the countermeasures for the problems and challenges. And a flexible review for the manner of project management is also required depending on the situation of the project. In order to realize the flexible management, project management must be carried out by joint operation with the customer. By performing co-management with customer, speedup of problem solution and improvement in quality can be expected.

1. は　じ　め　に

何をして大規模プロジェクトとするかは色々意見があるが，一般には「人が多い・工数が多い・期間が長い・対象とするシステム規模が大きい」などを挙げることができる．これらを満たすプロジェクトを“大規模である”としたとき，そのマネージメントは概ね以下のような状況に苦悶する．

声が届かない（コミュニケーションが希薄になる）

関連者が多く声が届かない，また関連者の声も聞こえない．一同に会することもできない．組織階層も深い

cf.　意思疎通の組み合わせ　要員数×（要員数－1）/2

管理が複雑で難しい

管理者に集まってくる数字は膨大，状況は入り組んで相互に影響する．自ずと管理の仕組みも複雑になる

ex. 実行中の WBS ≧要員数，膨大なテスト

時間が掛かる（“状況の把握”“意思の決定”“方向性の周知”に時間を要する）

コミュニケーションが難しく，管理が複雑となれば，プロジェクトが常に動いている中で状況の把握に手間を要するのは必然

ex. 舵を切っても曲がらず，アクセルを踏んでも動かず，ブレーキを引いても止まらない，速度計の表示は 10 日前

情勢が変化する

長期に及ぶ作業の過程で，外的な情勢や内的な情勢に変化が生じることにより，プロジェクトの状況が変化する

ex. 社会環境（政治・経済），会社情勢，個人事情…

これらの状況を避けることはできないが，緩和する方法はいくつかある．

本稿は大規模なプロジェクトの経験を通して，実際にプロジェクトで実践したマネージメントの技法や，工夫した点について記述している．逆に通常のプロジェクトにおいて一般的に実践している範囲と同等の管理の項目については特に記述していないため，プロジェクトマネージメントとしてみた場合は記載に網羅性がないことは留意されたい．

2 章ではまず大規模プロジェクトにおける計画の立案段階での工夫と取り組みについて記述する．続いて 3 章では計画の実行準備，4 章ではプロジェクト実行段階における管理について記述する．

2.　計画の立案

2. 1　プロジェクトのプロファイル

ProjectAI は大規模ミッションクリティカルな国内旅客システムを，オープンアーキテクチャ（java）をベースとして次世代システム基盤に再構築するプロジェクトである．プロジェクトの規模として，ピーク時のプロジェクト要員数は約 800 名，総工数 2 万人月超，対象システムのテストは複合ケースで約 40 万ケース，プロジェクトの開始から終了まで 5 ヵ年を要するという紛れもない大規模なプロジェクトである．構築対象のシステムは，重要な社会的基盤の信頼を担うものであり，このプロジェクトの第一の使命は，業務・機能品質，性能品質を再構築前より劣化させることなく安定稼働させることであった．

本章では，プロジェクトのタスクから見た計画の立案（2. 2 節）と，その中でも特に管理する側面から見た管理計画の立案（2. 3 節）についてそれぞれ記載している．

2.2　プロジェクト計画の立案

2.2.1　全体計画の考え方

大規模プロジェクトで計画段階に意識すべきは，管理できる単位にプロジェクトを細分化することである．ProjectAIは長大かつ広大な計画となってしまうため，最上位計画としての「プロジェクト計画」を作成し，その下位計画に，「フェーズ計画」を策定した．フェーズは重複することなく，全体プロジェクトの該当時期における主な実行目的に即した時系列方向に一定期間で分割した単位である．ProjectAIではプロジェクトを目的別に大きく表1の四つのフェーズ（詳細は七つのフェーズ）に分割することで，管理の単位を細分化した．

表1　フェーズ一覧

フェーズ名	期間	フェーズの目的・概要
方向づけフェーズ	9ヶ月	システム全体の初期要件を取り纏める
推敲フェーズ	12ヶ月	共通部品の先行開発を行う．開発方式を整え検証する
作成フェーズ	21ヶ月	初期要件を本格実装し，提供する機能を構築する
現行改修取込開発[*1]フェーズ		現行改修分の差分開発を取り込み，全機能を完成させる
移行フェーズ	17ヶ月	最終確認，移行，切替え作業を実施し本稼働を迎える
現行比較テストフェーズ		機能面の現行比較テストを行い，機能性を確保する
システムテストフェーズ		本格的運用を想定した最終的なシステム確認テスト

プロジェクトの細分化は時系列にフェーズ分割することに併せ，作業領域の分割も必要となってくる．各フェーズの作業を考察し，適切なチームを構築することで，プロジェクトを時系列と作業領域の両方向へ分割し，管理可能な単位を定義することも全体のプロジェクト計画段階で考えておく必要がある．

2.2.2　作業計画の考え方

フェーズの実行段階では，さらに3ヵ月ごとにWBSを作成・更新していく運用を定める．WBSの一つのタスクは1名1週間で実施する作業ボリュームを基本的な考えとしてブレークダウンする（図1）．タスクを1名1週間とすることによりタスク内の進捗率ではなく，タスクの着手と完了を追うことで，週次の進捗を測ることができ，管理の単純化に繋がる．

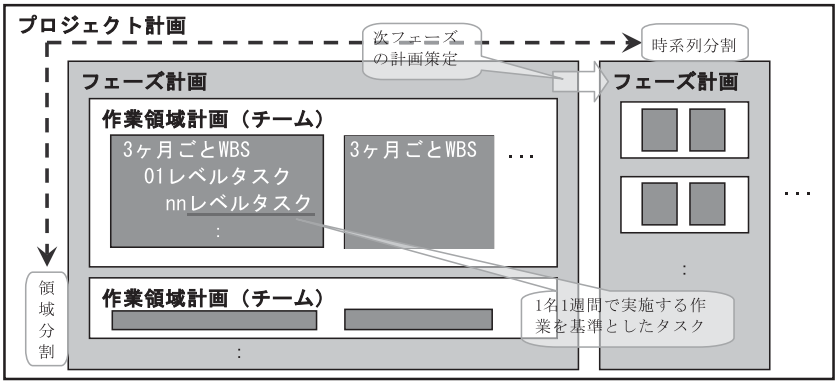

図1　計画の細分化

また，プロジェクトの分割では作業としての分割だけではなくそれに併せたプロジェクトの目的の分割も重要である．目的の積み上げにより，全体プロジェクトの目的が達成できるよう各フェーズのフェーズ計画，あるいはチームごとの作業計画の目的に掲げる必要がある．プロジェクトが迷子にならないよう参加者全員での目的とゴール共有が必要であり，何のためのプロジェクトか，一番大きな目的の理解により行間が読める活動ができるようになり，自律できるチームを目指す．

2.2.3　計画を立案する際に考えておくこと

プロジェクト計画やフェーズ計画，あるいは3ヶ月ごとのWBSなどの計画を立案する段階では，作業ごとにその作業の総量と生産性，投入要員数を見極める必要がある．その関係は以下の通りで，これらの係数を定めた上で計画化しなければならない．

総量　＝　生産性　×　要員数

ただし，この段階では，認識している事実と過去の類似作業の実績を基本として係数を決めることとなり，以下のようなリスクが内在することは免れない．

・想定した総量以上に作業が発生する（作業増加要因は様々）
・想定した生産性が出ない（予定要員が入らない，要員スキルが低いなど）

従って，計画立案時には少なくとも作業ごとに基礎情報（見積り前提としての総量，想定生産性，要員計画）を定めておく必要があり，プロジェクトの実行段階においてこれらの実績を随時追跡する必要がある．

フェーズの計画立案にあたっては上位であるプロジェクト計画と，前フェーズまでに発生した課題などの近日状況を踏まえ，見直しを加えた計画とすることで，開始時点のリスクを最小化する必要がある．前フェーズの完了時にはフェーズ内の課題と対応，および積み残した課題を洗い出し，次フェーズでの計画に繋げることができる．

2.2.4　テスト工程における作業計画の考え方

テストについては，様々な目的のテストがフェーズを超えて散在するため，テスト戦略書を作成し，テスト全体の組み立てと方向づけ，および各テストの目的の明確化を行い，テスト間の矛盾や重複が発生しないようにせねばならない．テスト間の矛盾とは，テストの抜けにより品質が劣化することや，テストの重複，あるいはテストの順序間違いにより前提とすべき条件が揃っていないなど作業の無駄が発生することである．

さらにテスト工程の作業計画は，品質の状況，すなわち障害の検出状況に応じて作業規模に影響を与えるリスクが他の作業に比べて高いため，臨機応変に計画の見直し（障害の修復ボリュームや追加テストによる見直し）を行う必要が出てくる．単体テストレベルでは影響の範囲も局所化されるが，機能テスト以降のフェーズでは品質の状況による計画への影響は大きい．

このため，テストの実行段階において各テストの計画書を作成し，フェーズ計画・完了報告と同様に前テストの完了報告において課題と対応の整理を実施して次の計画での精度を高める必要がある．また，テスト工程は管理の粒度も他の工程よりも高め，WBSではなくテストケースごとの完了予定と完了実績が日次で把握できる仕組みが必要となってくる．

2.3　管理計画の立案

プロジェクトにおける管理活動の目的は，「プロジェクト活動の透明性を高め，必要な対処を早期に行える」ようにすることにある．基本的なプロジェクト活動のポイントは以下のように整理でき，そのための管理の仕組みを計画する必要がある．

1)　課題やリスクを認識した場合は即時に共有し状況を確認する
2)　すべての作業は，その見積り前提を共有する
3)　計画に基づき作業を遂行し，完了状況については，実績を反映する
4)　実績と見積り前提を比較しプロジェクト状況を数量で表現し，乖離の傾向と原因究明を行い，その是正策を検討の上，作業化していく
5)　障害内容から傾向（機能や組織での偏り）を分析し，ウィークポイントを洗い出すとともに，その是正策を検討の上，作業化していく
6)　共有していない課題がないか，計画化していない作業に従事していないか，生産性を低下させる事象（例えばメンバのモチベーション低下）がないか，などプロジェクトの健全な進行を阻害する要因の有無についてプロジェクト活動に入り込んだ監査を行う
7)　成果物の品質（資料の体裁と妥当性，資料間の整合性など）を検査する

表2　管理項目一覧

管理項目	管理のポイント	管理対象 (管理主体)
課題管理	・課題対応は進んでいるか ・新たな課題，気になる点（課題の卵）はないか ・問題はなぜ起こったか？深層原因は何か	課題管理台帳
品質管理（Q）	・品質管理プロセスを守っているか ・障害検出数・内容に偏りはないか	障害管理ツール
工数管理（C）	・計画工数と実績工数との間に乖離がないか	工数管理台帳
進捗管理（D）	・どれくらい遅れているか，進んでいるか（計数化） ・計画規模が増大していないか ・計画した生産性はでているか	WBS，KPI
変更管理	・計画追加の妥当性の確認 ・計画は平準化できているか	変更管理台帳
構成管理	・ソフトウェアのバージョン管理をする ・開発ラインの管理をする	Rational ClearCase[*2] Rational ClearQuest[*2]
リスク管理	・期間の長いリスクと短期で検討しなければならないリスクに分けて管理する ・短期リスクについては課題管理の一環	リスク管理台帳
内部監査	・課題管理の一環として課題の検出に力点をおいた活動を限定時期で実施．内在課題や無計画作業の洗い出しを行う	第三者による活動
障害運営	・障害対応を円滑に進めるための仕組みづくりと運営フロー	障害管理ツール （品質管理）
質問表運営	・質問表（TQ）対応を円滑に進めるための仕組みづくりと運営フロー	質問表（TQ） （障害対応）
計画変更ルール	・手続きに則って適正に計画を変更する ・変更の理由と対処，影響を対面で確認	(課題管理)
開発環境	・テストスケジュールに合わせて開発環境を整備する	開発環境
リリース／ライブラリ運営	・リリース／ライブラリ管理を適切に行う	

表 2 に示す管理項目は管理活動目的達成のために必要な管理体系として ProjectAI で定義した主なものである．

3. 管理の実行準備

3. 1 管理準備作業

大規模プロジェクトにおいては，フェーズ計画やテスト計画などの計画立案と計画実行を繰り返すことになるが，その際管理の視点において以下のような準備作業が必要であることが多い．準備作業はどのレベルの計画であっても実行段階に移る前に行う必要がある．この作業がうまく処理できていないと，計画の立ち上げ遅延という状況に陥る．一旦動き出してからでは，方向修正が利きにくくなるのも大規模プロジェクトの特徴であり，準備作業を確実に行う必要がある．

1） 運営要領とルールの準備（誰が何をどのような手順で行うのか）
2） 管理ツールの準備（進捗，課題，品質などの状況が把握できる仕組み）
3） 実行検証（実際にやってみる，管理の仕組みの検証）
4） 計画の周知徹底（特に切替えが必要となる項目の変更のタイミング．例えば，体制が変わる，進捗管理の方法が変わる，といった場合のタイミングの周知）

3. 2 コミュニケーションの準備

実行段階に移ると“コミュニケーション問題”が課題となることは多い．ProjectAI においては幸いにもほぼ全員が一拠点に集まって開発をすることができたことはこの問題を考える上で大きなプラス要素であるが，それでもコミュニケーションに関連する課題は数多く存在する．その際解決に際して講じた有効策を表 3 に挙げる．特に大規模プロジェクトでなくてもよく使う対策だが，この問題に関しては総じて奇策はなく，地道な活動が効果的である．また，コミュニケーションの課題を解決するとその先には直接の課題解決以上の相乗効果が期待できる場合もあり，プロジェクトに与える効果は大きい．

表 3 コミュニケーション対策一覧

対策	概要
朝会などの日次定例	多数の朝会が多数の場所で開催．業務的な単位，目的別（テストなど）の単位，マネージメント単位など様々
目標とゴールの掲示	テストや障害修復の件数目標と実績の掲示
全体朝礼（／月）	フロアでマイクを使用しての朝礼．表彰制度などの啓発も兼ねて実施
活動体制の融合	日本ユニシスと顧客の活動体制の一体化．合同チーム制

定量的なデータではなく，経験則に基づくが一般に“良い話”は「より都合良く，比較的素早く」伝達され，“悪い話”は「より悪く，中々伝わらない」．また，伝達の方向（報告と指示）においても伝達の速度や情報の精度に違いが見られる（図 2）．

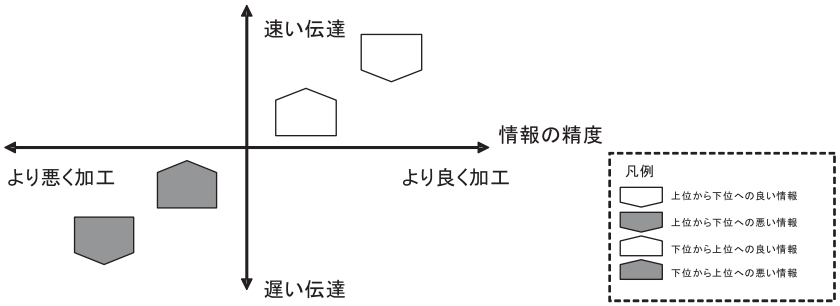

図 2　情報の伝達速度と精度

4.　管理の実行

4. 1　課題管理

課題管理はプロジェクトの状況を把握するダッシュボードとして位置づける．週次報告会（進捗報告が中心）とは別に課題解決推進会（週次）を設け，他の管理体系の上位に位置づけて課題に関わる情報の吸い上げと解決に注力できる運用体系とする．そのために体制上も顧客と一体となった課題解決推進チームを作り，そのメンバを各統括に配置し，統括内の活動や週次報告会からプロアクティブに課題を吸い上げる運用が必要である（図 3）．ここでいう“課題管理”とは，計画作業を妨げるような要因以外にも，計画外の作業や，単に“気になること”，それに管理期間の短いリスクも含めて，一旦共有し吟味し追跡することで，課題の発生前の抑止効果も期待した活動である．

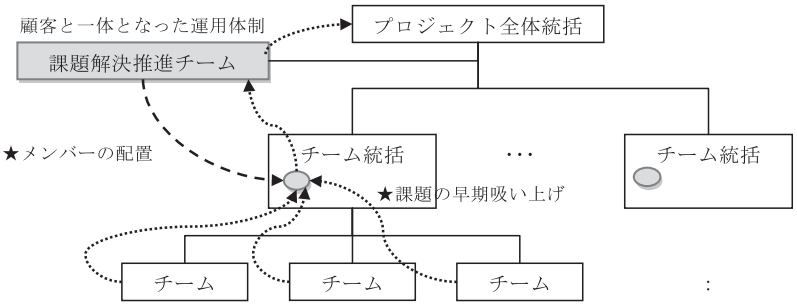

図 3　課題の伝達ルート

4. 1. 1　課題管理運営要領

発生した課題について，プロジェクト内で早期に共有し解決することを目的とした運営とする．課題の解決責任の明確化，正確な状況把握と上位へのエスカレーションを的確に行うために課題のレベル分けとレベルに応じた運用を実施する．

一般的にプロジェクト体制の階層に応じて課題管理の階層は異なるが，ProjectAI では，「チームレベルのチーム課題」，「各統括レベルの統括課題」，「プロジェクトレベルのプロジェクト課題」の三つのレベルで管理した．このうち，チーム課題は課題解決推進担当が確認し統括課題へのエスカレーションの必要性を吟味する．統括課題とプロジェクト課題は課題解決推進会においてレビュの対象であり，統括課題は課題解決推進リーダがプロジェクト課題へのエスカレーションの必要性を吟味する（図 4）．

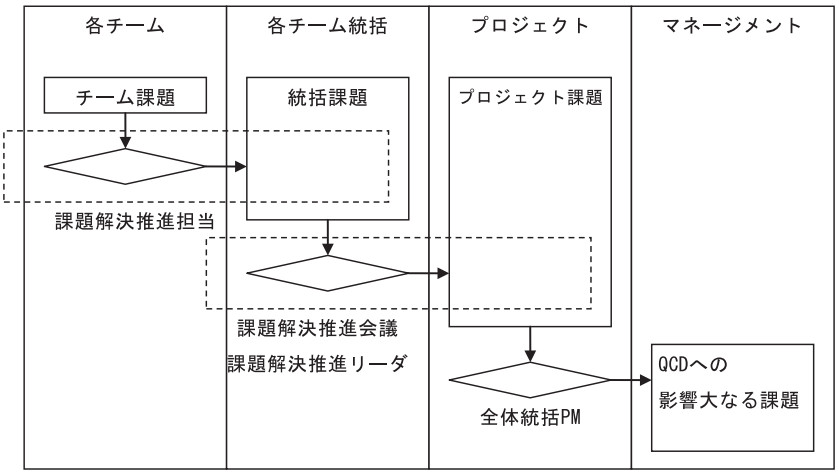

図 4　課題管理フロー

4. 1. 2　課題解決推進チームの役割

課題解決推進チームの役割は，各統括をはじめとするプロジェクト全体に対して，活動状況の把握，および進捗管理・課題管理・障害管理など各種管理・推進状況の評価を行い，問題事象に対する管理と対策の推進や支援（問題提起・改善策検討・是正支援）を行うこと，である．具体的には，以下の活動である．

1）　プロジェクト全般に対する運営面の状況把握
2）　各統括（各チーム）の進捗面での定量指標のトレンド監視
3）　潜在課題の発掘や顕在化した課題への対応に向けた推進支援
4）　上記活動からの定性評価の上位マネージメント向け報告

これらを実施するための具体的な個別のアクションは，以下の 2 点である．

1）　課題管理台帳がプロジェクトのダッシュボードたるよう課題情報を吸い上げる
2）　課題解決が滞らないよう解決をファシリテートする

そのため課題解決推進チームでは課題解決推進チーム会を開催し，課題解決推進担当が集めた情報を整理（課題の解決策の検討，課題の管理レベル決め，短期リスク追跡，“気になる”事項の対応）して課題解決推進会（課題解決推進リーダ，および各統括が参加）を主催することとなる．課題解決推進チームの担当者は，担当する各チームのチーム運営に参加し，上述した役割を全うするため主に以下の活動を実施する．

1）　課題解決推進のため，課題の解決担当者，解決目標期日，解決ステップが定まっていて作業が進捗していることを定期的に確認する
2）　いずれ課題となる可能性はないか，広い視野でチーム運営を見定め，気になる事項を課題解決推進チーム会へエスカレーションする

4. 2　品質管理

4. 2. 1　品質管理の考え方

品質管理は品質管理計画に準じて全工程を通じ体系立てた管理の中で運営する必要がある．品質管理計画では作業の工程ごとに異なる管理項目を定めて管理と評価を行う．アプリケー

ションの開発部分は工程ごとに品質管理のプロセスを定め定量的管理と定性的管理を行う．品質管理計画で定義される管理項目の範囲は広いが，本稿ではその中でも業務（アプリケーション）の機能性の品質確保に絞って言及する．

実行工程においては，計画において定めた品質管理の対象とするデータの採取・分析を行い，評価する．評価の結果，課題を検出した場合には，原因を特定して対応策を検討・実施し，結果データを採取するというサイクルを回す．

4.2.2　品質管理プロセス

品質確保は，まずはプロセスが重要である．品質を担保するための適切な活動を定義することがスタートであり，機能性品質におけるテスト密度や障害検出率といった管理指標を満たすことは“あって良き（必要だが充分ではない）もの”と考える．大規模プロジェクトは，品質を担保する“適切な活動”が普通のプロジェクトより多岐に渡り，フェーズやシーン，成果物ごとに色々な手立てを考えねばならない．しかも常にプロセスを評価し是正し次の管理計画（運営）を立ち上げる，というサイクルを回す必要がある．プロセス以外では，すべての成果物を一つずつ個体で品質を見ることが重要であり，これを省略することもできない．大規模であっても同じである．平均点ではわからない．

すなわちプロジェクトの実行段階における品質管理は，個体ごとの品質を担保する（見て是正するを繰り返す）ためのプロセスを作ってプロセス自体も評価・是正する活動である．

ProjectAI においても品質を担保するための活動は何度も改め，結果，機能テスト～システムテストの段階において実施したプロセスは表 4 の通りである．

表 4　テスト工程における品質管理プロセス一覧

プロセス	目的	概要
障害対応プロセス	テストで検出した障害の対応	テスト担当で障害を検出し，修正担当で修復．さらに結合テストを終え，テスト担当で再テストする．これらの流れをプロセス化し，効率よく，確実にテストを実行し，品質確認が組み込まれているプロセス．
障害対応プロセス（修正妥当性確認プロセス）	テストで検出した障害の対応で，新たな障害を作りこむことなく，確実にプログラム品質を向上させる	障害対応に携わらない第三者でソースコード，設計書の改訂内容をレビュし「正しくプログラム修正がされていること，要件書や設計書の修正内容と矛盾がないこと，性能・機能を劣化させる可能性がないこと等」を確認する．
障害対応プロセス（適正確認プロセス）	テストで検出した障害の対応で，定められたプロセスを実施した証跡を確認する	ツールを使って機械的な各種チェックを実施する．および，プロセスの実行で残されている証跡を収集しプロセスが適正に実行されていることを検証する．
作業成果物維持管理プロセス	テストと直接関係のないドキュメント成果物の品質確保プロセス	計画書，完了報告書／評価書／結果報告書，などのドキュメント類の内容確認．
全体プロセス	個々の品質管理プロセスが連携して全体の品質を担保するための仕掛け	品質維持活動に対する各担当者の使命（何の責任を負うか）と役割，およびフェーズ終了段階における品質管理チームの評価を定義している．

4.2.3　定量的な管理

あらかじめ定めた管理項目ごとにメトリクス管理目標値を設定し，管理許容範囲に入っているか否かの管理・評価を行う．管理許容範囲の上限あるいは下限を超えた場合は詳細な調査を行い，対策が必要と判断した場合は，担当するチームと協議し対策を策定の上，対策の実施を

管理する．

例）不具合検出率が上限を超えたため，詳細を確認したところ，特定の人物が設計した部分について不具合が多いことが判明した．そこで当該人物が設計した部分について，すべて再レビュを実施して不具合を是正した結果，不具合検出率が下がった．

4. 2. 4　定性的な管理

不具合や障害がメトリクス管理目標値に入っていても，検出内容に偏りがないかについて監視する．偏りを検出した場合は原因を明確にし，対策が必要であれば，担当するチームと協議し対策を策定の上，対策の実施を管理する．

例）障害数は管理許容範囲内であるが，特定のファンクションに障害が集中していることを検出した．詳細を確認したところ，このファンクションの設計品質が悪いことが判明し，この部分について設計まで遡って再レビュを実施して品質を向上させた結果，当該ファンクションの障害が霧消した．

4. 2. 5　品質管理の実行段階における見直し

大規模プロジェクトではプロジェクト開始段階に品質管理計画書を作成し，それに準じて管理していても，想定されていなかったプロジェクト固有の問題が発生することは多い．従って評価項目も品質の状況に応じて随時改善する必要がある．表 5 は ProjectAI にて見直し改善した評価項目である．

4. 2. 6　品質評価

本番稼働を迎えることを踏まえた品質の最終的な評価に関しては，顧客と一体となった評価が必要である．“最終的な評価”ではあるが，一時点で評価できるものではなく，時間軸に沿って弱点や劣化したポイントを克服しながら，品質が徐々に向上していく状況を共有する活動が必要である．品質が“達した”とか“達していない”という“点”で語れるものではない．大規模であればなおさらである．

このためには，有識者による業務的なテスト期間を通じた定期的な評価が効果的である．ProjectAI ではこの定期的な評価について，個々の機能単位に定期的確認を実施することは機能の数から現実的ではないため，定点観測として実施されているテストに対して第三者視点での一定の確認を行った．

4. 3　工数管理

管理の対象は以下の 2 要素である．ここでいうスキルとは，スキルレベルの問題ではなく役割では超えることのできないスキルの種類であり，業務領域の違いや，作業領域の違いも考慮すべきである．

1）　要員計画（チーム別・スキル別，月単位，要員数の計画）　　⇒　計画工数

2）　要員実績（チーム別・スキル別の要員数実績）　　⇒　実績工数

進捗管理の観点において「計画作業規模÷計画生産性＝計画工数」と「作業実績規模 ÷ 投下要員数（＝実績工数）＝実績生産性」の根拠から，「計画規模と実績規模」・「計画生産性と実績生産性」の比較を行う．

表5　品質管理の見直しと追加

項目	説明	効果，改善項目
複雑度の測定	プログラムの複雑度を業務機能別に測定した．複雑度が高ければ障害も多く発生すると予測し，メトリクスの補正を実施．	メトリクス評価時に数字が高めな理由，低めな理由を客観的に示せる．
障害発生部位	機能テストにおいて，障害検出時の障害発生部位を特定し分類した．自モジュールI/F誤り，他モジュールI/F誤り，自モジュールDB誤り，他モジュールDB誤り，プログラム等アルゴリズム誤り．	ファンクションの組合せバリエーションは無数にあることから，障害発生部位を特定することにより，弱点を定性的にとらえやすくなった．
見逃し率 すり抜け率	障害を埋め込んでしまった工程を特定し集計することにより前工程以前の品質を確認する．	後続工程で検出された不具合や障害からそれ以前の工程の品質が推し量れるようになった．障害が多発した場合に，どの工程が弱点であったのか容易に推定できるようになった．
障害率	現行改修取込フェーズから新規開発と改修が混在したことから一律的な障害検出率の測定ができなくなった． テストケースあたりの障害割合である障害率を採用した．システムテストにおいても障害率を採用し連続性をもたせた．	過去からの一貫した流れで品質状況を把握することができるようになった．
機能別評価	機能テストにおいては下記の方法により，すべての機能について評価を実施した． ログから実行された機能を集計 →実行実績があり，障害が検出されていなければOK 障害が検出されたものは機能ごとに障害ポイントを付与し，重障害については重点監視を実施 障害を検出した機能についても，修正後にログで継続して障害なく稼働していることが確認された時点でOK	機能は単独でテストされることはなく，テスト用のデータ作成や結果の参照など，テスト対象の機能以外も組合せてテストされているが，テスト対象の機能のみの観点から，全てのテストで利用されている機能について機能単位に評価を実施したことにより，評価の精度向上が図れた．すべての障害について定性的分類を行ったことにより，弱点が明確になった．
品質確認	過去にメトリクス偏重による誤った評価をした反省を踏まえ，品質チームによる確認と定性的な評価を行った．	メトリクス値には表れない弱点を客観的に明確にすることができた．当確認を定期的に行うことで,プロジェクトとして弱点の追い込みと解消を可視化して進めることができた．

基礎数字である「実績工数」を，工数管理から進捗管理に連携することにより，進捗管理レポートには工数管理の要素も含んでいることとなる．

直接業務に携わる要員の工数管理については，月次で人別・タスク別の月間投入時間を採取することにより，実績工数を把握する．

4.4　進捗管理

進捗管理は「テスト以外工程の作業」と「テスト工程の作業」で管理の仕組み，粒度を変えて運用する（表6）．理由は2.2.4項で述べた通り，テスト工程における進捗遅延のリスクを極小化するよう，よりきめ細かく管理するためである．

表 6　テスト以外工程とテスト工程の進捗管理比較

管理の要素	テスト以外工程	テスト工程
管理粒度	WBS（１名で１週間が目安）	１テストケースごと １障害ごと
管理周期	週次状況確認	日次状況確認
共有会	週次進捗確認会議	日次朝・夕会
生産性管理	月次で計画作業と工数より算出	日次KPIで管理
管理ツール	PD（Process Director）*3	テスト実施記録（Office自前ツール） 障害管理ツール（Office自前ツール）

4. 4. 1　テスト以外工程の WBS による進捗管理

WBS は作業期間が 1 週間で単独名による実行タスクにブレークダウンすることを基本的な考えとする．これにより WBS の進捗は，WBS 内の進捗率（このタスクは何パーセント進捗したか）を管理する必要がなく，WBS の「未着手」→「作業中」→「完了」というステータスを押さえることで全体の状況把握が単純化できると同時に，担当者の主観的な進捗認識を排除する．また，作成物については作成物管理台帳で作成物ごとに作成フェーズ，工程，担当者，承認レベルを管理する．承認レベルは作成物の重要度・影響範囲から 4 段階設定している．

4. 4. 2　テスト工程の進捗管理

テスト工程における作業は,「テストの実行」と「検出した障害の対応」の二つに大別できる．他の工程作業との作業特質の違いは，テスト実行も障害対応も障害の検出状況に応じて作業ボリュームが変更される可能性があることである．とはいえ，要員や期間が際限なく利用できるテストというものはあり得ない．従って，テストの進行に併せて仔細に状況を把握し，計画を適切に見直す必要がある．そのために ProjectAI ではテストにおける関連計数を KPI（Key Performance Indicator）として管理し日次で追跡する運用とした．

テストにおける KPI は，テスト工程の実行時に主に生産性（テストの実行生産性と障害の対応生産性）を追跡することを目的に設定した指標である．以下は ProjectAI で設定した主な KPI である．

1）　テスト実施生産性

　　算出式：テスト件数　/　テスト実施人数　/　日数

2）　検出障害率

　　算出式：検出障害件数　/　テスト件数

3）　解析生産性

　　算出式：解析件数　/　解析者人数　/　日数

4）　実障害率

　　解析した結果「障害の起票誤り」や「過去の検出障害と同件」といった実際には対応が不要な障害を除いた，実際に対応が必要となった障害の検出率.

　　算出式：実障害件数　/　検出障害件数

5）　障害対応生産性

　　算出式：実障害件数　/　障害対応人数　/　日数

上記の通り，KPI は見積り（計画）前提そのものである．計画時に設定した作業ごとの生産性と想定の作業発生率である．これらの計数に定めた値を是としてテストの期間を決め，計画

を立てている．日々の活動において，これらの計数の実績値を取得し，計画時点の値と比較・評価することで，進捗状況の判断に使用する指標の一つとする．日次の KPI モニタリングで生産性が計画時の想定を下回ったり，あるいは障害検出率が上回ったりした場合は，いずれも進捗に影響することが判断できる．

5. お　わ　り　に

「大規模開発のプロジェクトマネージメント」は，管理の項目自体は中小規模のプロジェクトマネージメントと大差はない．ただ，プロジェクトを立ち上げる段階で手法ややりたいことがすべて見えていなくても，プロジェクトの進行状況に併せて適切な管理手段を考えて適用すること，これが大規模プロジェクトにおいて重要な要素の一つである．

もう一つのポイントは，顧客と一体となったプロジェクト管理である．顧客がプロジェクト推進に占める役割と効果は大きく，今回記載の範囲でも，課題管理や品質管理は顧客と一体となった推進が効果的であった主な分野である．ProjectAI では終盤のテスト工程において，体制上も一体となったチームを作り，プロジェクトのゴールだけではなく，すべての状況や課題を共有し一丸となって解決してきたことは，プロジェクトを本番稼働へと導いた大きな推進力となった．

最後に ProjectAI をともに進めてくださった全日本空輸株式会社関係各位，並びに ANA システムズ株式会社関係各位，プロジェクトメンバの皆様，関連パートナ各位，および日本ユニシス関連部署の各員に感謝の意を表したい．

＊ 1　現行改修取込開発　ProjectAI は長期の開発であるため，ProjectAI 開始後に現行にて追加開発・改修した機能についての開発を現行改修取込開発と定義した
＊ 2　IBM 社製品　ProjectAI にて構成管理ツールとして使用
　　IBM Rational ClearCase：ソフトウェア資産管理ツール
　　IBM Rational ClearQuest：ソフトウェア変更管理ツール
＊ 3　日本電気株式会社製品　プロジェクト管理システム「ProcessDirector」．現在は販売停止となっている．

執筆者紹介　坪　内　　淳（Atsushi Tsubouchi）
1989 年日本ユニシス(株)入社．金融部門で勘定系システムの大規模開発・保守を担当．2000 年より CRM 担当として多業種の CRM/コールセンターのコンサルティング，開発・構築を手がけ，2008 年より公共部門エアライン事業部で ProjectAI に従事．2013 年より製造流通部門で流通システムの開発を務め現在に至る．