

チーム開発支援モデルと TeamFactory

Team Development Support Model and TeamFactory

橋 本 篤

要 約 チーム開発支援モデルは LUCINA による開発を支援する方法を定義し、開発支援チーム主導の下でツールによる支援環境を構築し、成果物の履歴、構成情報、依存性情報を管理することで、無駄な手戻りを防ぎ、工程間の引き継ぎや変更発生時の適切な対応を可能にする。そして、要求追跡や変更要求対応などのプロセスを標準化し、プロセスと成果物を関連付けることで、要求や変更要求への漏れ・誤りのない対応と、成果物の品質維持を可能にする。

TeamFactory は、大規模システム開発の支援を目指した構成管理ツールであり、最大の特徴はライン管理によるチーム開発のサポートにある。今回、LUCINA による開発の支援ツールとして実際に TeamFactory を適用し、チーム開発支援モデルのツールとしての有効性を評価した。

チーム開発支援モデルは、プロジェクトによって異なることが多かった開発の支援方法について、具体的で一貫性のある支援体制の構築に貢献できるだろうし、TeamFactory は、チーム開発を前提とした構成管理作業を強力に支援でき、チーム開発支援モデルに適用するツールとして十分な素質を持っている。

Abstract The team development support model defines how to support a system development using LUCINA. It builds the development support environment using the tools under the development support team leadership and manages the history, configuration information and dependency information of the work products, which enables to prevent the development processes from unnecessary redoing, and cope effectively with to the smooth movement to the next development activity and the changes encountered during a development processes. And then, it enables to cope completely and correctly with the requirements and/or the changes to the requirements, and maintain the quality the work product, by standardizing the processes such as the trace of requirements and the action to be taken for the changes to the requirements, and associating the development process with the corresponding work product.

TeamFactory is a configuration management tool aimed at the support of large-scale system development, and its biggest characteristics exist in the support of team development under the baseline management. Recently, the author has actually applied TeamFactory to the development using LUCINA as support tools and evaluated the validity as a tool of team development support model.

The team development support model will be able to contribute the building of a concrete and consistent supporting system in the development support which depends on each project and then TeamFactory might have all the makings of the tool by means of which powerfully support the configuration management work based on development by several teams and apply to the team development support model.

1. はじめに

ユニシスの TEAMmethod^{*1} 方法論では、システム開発を成功に導くために必要なプロジェクト管理要件について示している。LUCINA では、様々なプロジェクト管

理要件の中から、構成管理・要求管理・変更管理に着目し、これらの管理を「チーム開発支援モデル」として定義した。チーム開発支援モデルでは、LUCINA 開発プロセスを円滑に進めるために必要な管理要件やツールによる支援内容を具体的に示している。

一方、日本ユニシス（以下、当社）では、平成 12 年 12 月、チーム開発支援ツール TeamFactory^{*2} を商品化した。TeamFactory は、複数メンバ・複数チームによる構成管理を徹底するために、並行開発、成果物の照査、複数開発者間の連携などを特徴とする。

筆者は、TeamFactory のチーム開発支援モデルへの有効性を評価するために、実際に LUCINA for Java/WLS¹ 開発プロセスを採用したシステム開発への適用を行った。本稿では、チーム開発支援モデルの必要性とチーム開発支援モデルへの TeamFactory の有効性について、実際の適用経験を踏まえながら示していく。

2. チーム開発支援モデル

2.1 プロジェクト管理の現状

プロジェクト管理の主な要件としては、品質管理・進捗管理・財務管理・リスク管理・外注管理・構成管理・要求管理・変更管理などがあげられる。近年は、プロジェクト管理の重要性が認識され、大規模なプロジェクトでは、プロジェクト運営チームを設立して専任の要員がこれらの管理を進めていく傾向が強くなっている。

しかし、構成管理・要求管理・変更管理などの管理要件については、成果物の内容を把握する必要がある。開発者が管理作業の主体になるため、未だに開発者まかせにされがちである。そして、実際の開発現場では、手作業や、紙ベースによる旧態依然の管理が行われ、開発が佳境になるほど漏れや誤りが多発し、納期の遅れや品質の低下を招き、プロジェクトを失敗に至らしめることになる。

2.2 チーム開発支援モデルの焦点

チーム開発支援モデルは、構成管理・要求管理・変更管理に焦点をあてている。図 1「チーム開発支援モデル」で示すように、チーム開発支援モデルでは、システム開発のライフサイクル全体を通して発生する「成果物」と「プロセス」の面から管理する。

成果物とは、各種設計文書やコード、報告書などの電子ファイルを示す。成果物管理では、成果物ごとに過去に有効だった内容を保存しておくことで、無駄な手戻りの発生を回避したり、保守の際には過去の経緯を参照できるようにする（リビジョン管理）。また、システム全体で同期性を保ちながら成果物を整理・統合することで、ある時点における有効な成果物群を構成情報として把握し（バージョン管理）、この構成情報を必要なタイミングで確定することで次工程への引継ぎやリリース、保守への対応を容易にする（ベースライン管理）。

プロセス管理では、システムに対する要求が適切に成果物に反映されているかどうかを追跡するプロセス、および成果物に対して変更要求が発生した場合に関係者がとるべきプロセスをあらかじめ規定する。これらのプロセスを標準化することで、対応の漏れや誤りを防止し、要求の影響を受ける成果物に対して一定の品質を維持できる。

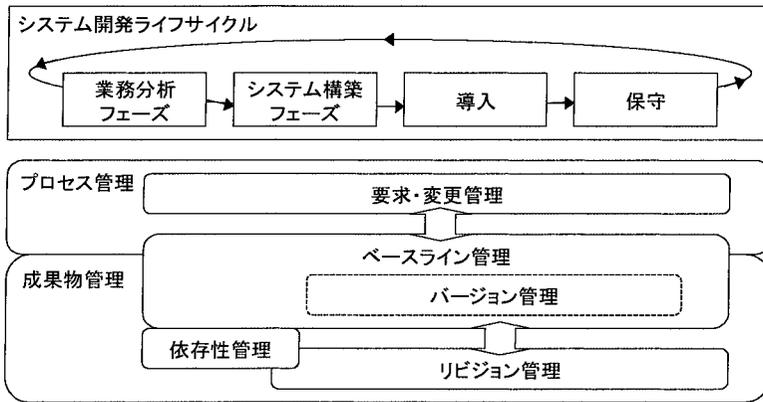


図 1 チーム開発支援モデル

さらに、変更要求に対応する過程で成果物が作成・変更されたら、変更要求と対応する成果物の関連付けが必要になる。成果物管理とプロセス管理は相互に関連しながら運用されることによって、成果物が適切に作成・変更・参照され、システムを効率的に成長させることができる。

なお、チーム開発支援モデルでは、これらの管理に構成管理ツールや変更管理ツールを適用することを前提としている。ツールを効果的に利用することで、管理作業が、効率化、標準化され、開発者を直接的に支援し、開発現場の混乱を回避する。

2.3 チーム開発支援モデルの支援内容

2.3.1 開発支援体制

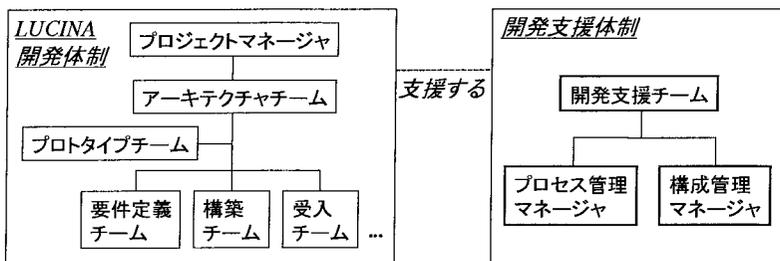


図 2 開発支援体制

チーム開発支援モデルでは、まず、モデルを円滑に運営するために、図2「開発支援体制」で示すような開発支援チームを設立する。

開発支援チームの役割は、ツールの機能に基づいてモデルの運用基準を立案し、それに則って支援を推進することである。そのため、適用するツールにある程度精通した要員を構成管理マネージャやプロセス管理マネージャに任命する必要がある。そして、開発支援チームが、必要な環境構築や教育を実施し、開発者に運用基準の遵守を徹底させて行く。

なお、これらの役割は、小規模な開発プロジェクトの場合は開発者が兼任できるか

もしないし、規模が大きければプロジェクト内に専任の担当者が必要になるだろう。Brown^[2]は、企業内で独立した開発支援組織を設立し、複数の開発プロジェクトを並行して支援する専任体制の重要性を述べている。いずれにしても、役割と担当を明確にすることが重要と考える。

2.3.2 成果物管理

プロジェクトにおける成果物は全て一元管理され、参照要求や変更要求に対応できるように、どこに何があるのか、担当者は誰なのか、最新のシステム全体を構成する有効な成果物はどれなのか、変更の影響を受ける成果物はどれとどれか、なぜ変更されたのか、変更された箇所はどこか、といった点を明確にする必要がある。

これらの要件を満たすために、成果物管理を「リビジョン管理」と「ベースライン管理」、および「依存性管理」により実施する。

1) リビジョン管理

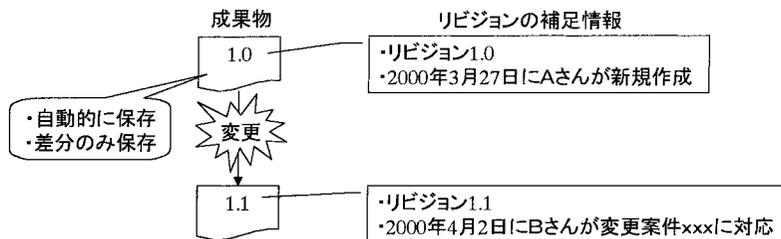


図3 リビジョン管理のイメージ

成果物個々の変更履歴を管理する。図3「リビジョン管理のイメージ」で表すように、成果物を変更する際には、必ず変更前の内容を過去のリビジョンとして保存し、リビジョンごとに、変更した担当者、日時、理由などの補足情報を保存する。こうすることにより、万一過去の内容に戻さなければならなくなった場合に、容易に過去の適切な時点の内容に戻せるようになる。

なお、ツールを利用することで、変更前の内容を自動的に保存し、差分・圧縮保存により、ディスクスペースが節約され、異なるリビジョン間での内容の比較やマージを機械的に行うことができる。

2) ベースライン管理

① ベースライン管理のイメージ

必要なタイミングで、ソフトウェア全体を構成する有効な成果物の情報を管理する。図4「ベースライン管理のイメージ」では、要件定義完了時点で要件定義書のバージョン1.1と論理設計書のバージョン1.1によって要件ベースライン1.0を確定している。このように、特定の時点でシステム全体で同期的に有効な成果物とその有効なリビジョンはどれなのかを串刺しして管理するイメージになる。

② ベースライン確定のタイミング

図5「ベースライン確定のタイミング」で表すように、ベースラインは、工

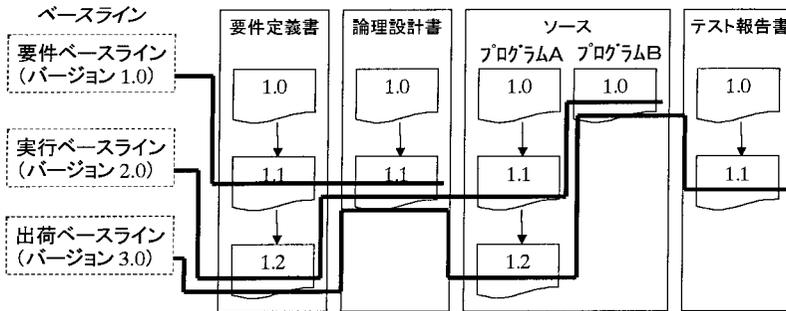


図 4 ベースライン管理のイメージ

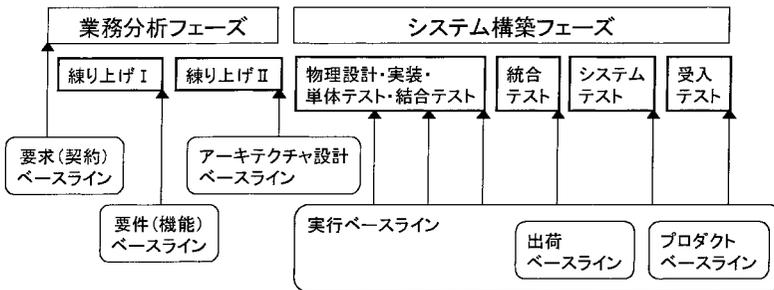


図 5 ベースライン確定のタイミング

程や開発プロセス，または担当チームが切り替わるタイミングで確定する．こうすることで，前後の工程や担当チーム間でベースラインという共通の見方ができ，前工程の成果物を整理・統合した状態で確実に次工程に引き継ぐことができる．例えば，実装工程の終了時に実行ベースラインを確定して統合テスト工程に引き継げば，ビルドがやり易くなるであろうし，統合テスト工程の終了時に出荷ベースラインを確定すれば，出荷ベースラインを構成する成果物をそのまま出荷媒体にコピーすれば良い．

チーム開発支援モデルでは，要求(契約)ベースライン，要件(機能)ベースライン，アーキテクチャ設計ベースライン，実行ベースライン，出荷ベースライン，プロダクトベースラインなどを想定しているが，どのタイミングでベースラインを確定するかは，プロジェクトによって検討すべきである．

なお，ツールを利用することで，ベースラインの確定にあたって，有効な成果物を容易に組み込むことができ，ベースラインから，構成成果物の有効なリビジョンを特定でき，適切に保守作業を行うことができる．また，確定したベースラインを不用意に更新しないように，適切なセキュリティを設定できる．

③ ベースライン確定時のレビュー

ベースラインを確定する際には，必ずレビューを実施する(図6)．レビューを確実に実施することによって，個々の成果物の品質が保たれ，最終的にベースラインの品質が保たれる．

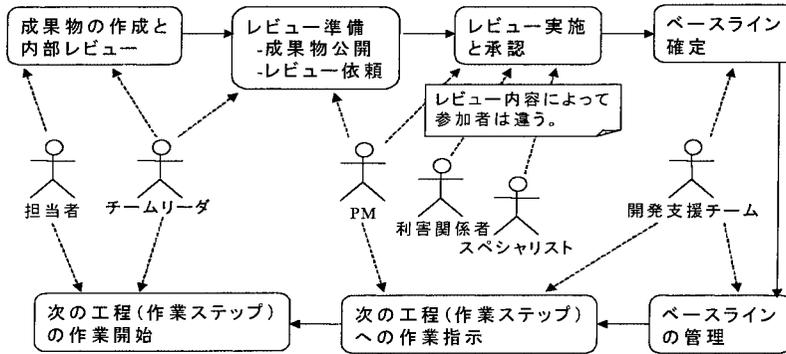


図 6 レビューの制御

- ・ 成果物の作成と内部レビュー
 担当者は、成果物を作成したらチーム内部でレビューを行う。プログラムを開発した場合は、担当者レベルでの単体テスト、およびチームレベルでの結合テストを行い、テスト文書もレビュー対象とする。
 - ・ レビュー準備
 チームリーダーは、レビューアが成果物を参照できるように成果物を公開してから、然るべき人物にレビューを依頼する。依頼の際には、成果物の公開場所、名称、リビジョン番号、およびレビューの期限などを明確にする必要がある。
 - ・ レビュー実施と承認
 レビューアは、成果物をレビューし、承認するかどうかを回答する。承認しない場合は理由を明記し、チームリーダー、または担当者の対応を要求する。
 - ・ ベースライン確定
 構成管理マネージャは、レビューで承認された成果物をベースラインとして確定する。
 - ・ ベースラインの管理
 構成管理マネージャは、一度確定したベースラインが不用意に変更されないように厳密なセキュリティを設定する。万一変更の必要が発生した場合は、変更要求とし、後述する変更プロセスに基づいて変更する。
 - ・ 次の工程への作業指示
 構成管理マネージャは、ベースラインを確定したら、即座に次工程のための構成管理作業領域を準備し、その旨をプロジェクトマネージャに伝える。プロジェクトマネージャは、次の工程や作業ステップを担当するチームに対してベースラインの参照方法を周知し作業指示を行う。
 - ・ 次の工程の作業開始
 次工程のチーム、および担当者は準備された構成管理作業領域で、ベースラインの成果物を参照しながら作業を開始する。
- 3) 依存性管理
 成果物には必ず他の成果物との依存関係がある。依存関係を管理することによ

り、ある成果物に変更が必要になった場合に、その変更の影響を受ける可能性がある成果物を即座に把握できる。そして、影響を受ける可能性のある成果物について、具体的な影響内容を調査すれば、変更要求に対する正確な工数把握や要員準備を行うことができる。

また、ヘッダファイルとソースファイル、ソースとバイナリ、ライブラリとライブラリアクセスモジュールなどのプログラムコード間の依存関係は、ビルドスクリプトを作成する際の入力情報としても活用できる。

2.3.3 プロセス管理

1) 要求の管理（追跡）

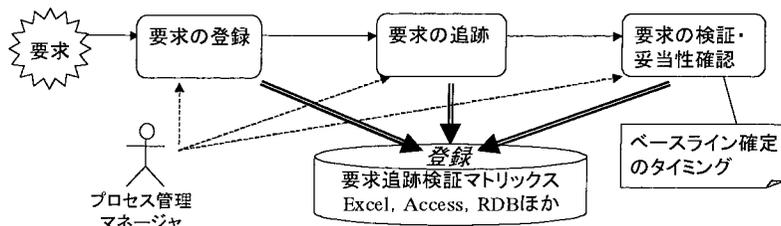


図7 要求追跡プロセス

要求の管理（追跡）は、要求一件ごとに、ベースライン確定の都度、要求への対応漏れが無いかが、要求への対応が妥当かどうかを確認し、品質の低下をできるだけ早い時期に防ぐようにする（図7）。

以下のような内容で、プロセス管理マネージャが表計算ソフトなどを使用し、マトリクス管理する。

- ・ 要求の登録
要求ごとに、要求番号、記載文書、要求の重要度（必須、検討要、できれば、任意等）などの属性を付与して登録する。
- ・ 要求の追跡
登録した要求ごとに、対応する成果物はどれなのかをマトリクスに追記する。
- ・ 要求の検証・妥当性確認
ベースライン確定時には、マトリクスから成果物を辿り、要求への対応内容を検証し、妥当性をマトリクスに追記する。

2) 変更管理

変更要求は、図8「変更管理プロセス」で表すような手順で追跡・対応する。

変更が発生したら、まず、構築チームや開発支援チーム、ユーザなどが変更依頼書を起票する。以降は次のようなプロセスを経る必要がある。

- ・ 依存性分析
開発支援チームは、開発者と共に変更の影響を直接受ける成果物を特定し、その成果物を変更した場合に影響を受ける可能性がある成果物を依存性情報から洗い出す。そして、洗い出した成果物の担当チームリーダーに対して、影響分析を指示する。

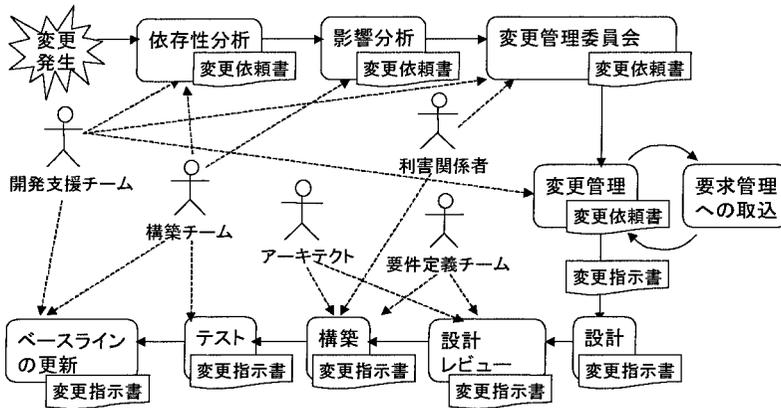


図 8 変更管理プロセス

- ・影響分析
影響分析を指示されたチームリーダーは、チームメンバーと影響内容を詳細に調査し、修正・追加に必要なコスト・期間・要員などを割り出す。
- ・変更管理委員会
開発支援チームは、変更管理委員会を招集し、影響分析の結果を元に要求の受け入れを検討する。変更管理委員会は、利害関係者であるユーザ、プロジェクトマネージャ、開発担当者などで構成する。なお、明らかに故障対応のための変更要求であれば、この部分はスキップしてもよい。
- ・変更管理
変更管理委員会で変更要求の実施が決定された（または故障対応の）場合、開発支援チームは、変更依頼書から変更指示書を起票し、変更を担当するチームのリーダーに変更作業の開始を指示する。
- ・要求管理への取り込み
変更要求が要求ベースラインにも影響を及ぼす場合、プロセス管理マネージャは、新しい要求として追跡管理の対象とする。
- ・設計
ベースラインの設計文書に対して、変更指示に基づいて設計内容を変更する。
- ・設計レビュー
設計内容をレビューして承認が得られたら、設計文書と変更指示書との関連付けを行う。
- ・構築
変更された設計文書に基づいて、プログラムの変更、単体テストを行い、テストに合格したら、成果物と変更指示書との関連付けを行う。
- ・テスト
総合テストを行って合格したら、テスト報告書と変更指示書との関連付けを行う。
- ・ベースラインの更新

変更指示書と関連付けられてきた成果物（設計文書・プログラム・テスト報告書の有効なリビジョン）を、同期性を保ちながらベースラインに組み込み、新しいベースラインとして確定する。

なお、対応の過程で成果物を変更したならば、変更案件と変更された成果物のリビジョンを関連付ける。ある変更案件によってどの成果物が更新されたのか、逆に、ある成果物のリビジョンはどの変更案件に対応したもののなのかを双方向から認識できるようにし、ベースラインの更新に役立てる。

また、ツールを利用することで、担当者をアサインしていない変更案件や進捗の滞っている変更案件をチェックして変更作業の漏れや遅れを防止できる。さらに、関係者に自動的に案件の最新の対応状況が通知されるので、案件関係者間の連携をスムーズに行える。さらに、案件同士の依存関係を把握し、全ての依存案件が完了してから元の案件を完了させるようにルール化できる。

2.4 チーム開発支援モデルへの期待

開発者は、4 GL・CASE・IDE などの開発に直接関係するツールには注目するが、構成管理・変更管理などの補助的な分野やツールには関心が低い。ツール適用を想定した開発支援方法を示したことにより、どのような開発支援が必要なのか、具体的にイメージできるようになるだろう。

また、開発者は、一旦開発に入ると、納期迄に作業を終えることに集中しがちで、補助的な管理はどうしても場当たり的になりかねない。これを、開発支援チームが推進することで、プロジェクトを通して一貫性のある開発支援を実践でき、ゆくゆくは組織や企業レベルで共通の開発支援体制の構築へと期待が持てる。

今後は、開発者だけの支援に留まらず、成果物の作成状況から進捗管理に、レビューの状況から品質管理に結びつくような、総合的なプロジェクト支援が望まれる。これらの内容は、ツールによる統合的な支援が実現されていない分野であるが、チーム開発支援モデルには先行的に組み込み、より理想的な支援体系をイメージした上で、ツールの成長を見据えるべきと考える。

3. TeamFactory によるチーム開発支援モデルの実践

前述したように、チーム開発支援モデルは、構成管理や変更管理などのツールの適用を前提としている。そこで、開発中であった TeamFactory を、LUCINA for Java/WLS による開発に適用し、チーム開発支援モデルを実践するツールとしての有効性を評価することにした。

ここでは、TeamFactory の主な特徴と適用のポイント、および実践で得られたチーム開発支援モデルの支援ツールとしての適性について示す。

3.1 TeamFactory と従来の構成管理ツールとの違い

TeamFactory は、複数チームによる開発が前提となる中・大規模システムの開発を、成果物管理を中心に支援するチーム開発支援ツールである。

では、従来の構成管理ツールとどこが違うのだろうか。もちろん、既存のツールでもリビジョン管理やベースライン管理を行う機能は提供されている。しかし、そこには、構成管理作業も本来の開発と同様に、複数のチーム、さらには各チームに属する

複数のメンバによって行われるという配慮が欠けている。

例えば、数十人、数百人が作成した膨大な成果物をいきなり一つのベースラインとしてまとめるのは非常に困難で負荷が高い。また、ベースラインに取り込む前には、成果物の品質が保証されていることを確認しなければならない。さらには、ベースラインを確定した後、次の工程を引き継ぐチームへの作業指示が必要になる。これらの点が考慮されていない既存の構成管理ツールでは、開発規模が大きくなると、たとえ専任の構成管理マネージャが運用したとしても、プロジェクト全体の構成管理を効率良く、正確に実施することは困難になってくる。

そこで、複数チーム・複数メンバによる並行開発と同じイメージで構成管理を実践すべく開発されたのが TeamFactory である。TeamFactory は構成管理の作業負荷を複数のチームやメンバへ適切に分散し、逆に、分散する成果物を効率的にベースラインへ取り込むことができる。また、成果物をベースラインに取り込む際には、必ず品質をチェックさせるプロセスを介在させる。さらには、関係者間に必要な連絡を自動化するなど、大規模プロジェクトにおける開発作業を円滑に進めるための配慮がなされている。

3.2 TeamFactory の特徴

3.2.1 ライン管理

TeamFactory では、メインライン・サイドライン・個人ラインという3種類のラインを使い分けて構成管理の作業領域を設定する(図9)。メインラインの下位に複数のサイドライン、サイドラインの下位に複数の個人ライン(サイドラインも可)というように階層的に分岐(以降ブランチと称す)し、メインラインでは全体、サイドラインではチーム担当部分、個人ラインでは個人担当部分というように作業範囲を絞り込みながら割り当てる。そして、メインラインは構成管理マネージャ、サイドラインはチームリーダー、個人ラインはチームメンバというように各ラインの管理者を割り当て、ユーザ(管理者)はライン上で割り当てられた作業範囲、つまり担当する成果物を登録・更新できるようになる。

開発が進み、チームメンバの作業が完了する都度、個人ラインを上位のサイドラインに統合(以降マージと称す)し、個人ラインの成果物がサイドラインに取り込まれる。したがって、チーム内の全メンバの作業が完了し、ブランチした個人ラインを全てマージすれば、サイドラインにはチームで担当する全ての成果物が存在することになる。こうして、チーム作業の完了時、サイドラインを上位のメインラインにマージすることにより、サイドラインの成果物がメインラインに取り込まれる。ブランチした全てのサイドラインをマージすれば全チームの成果物がメインラインに存在することになり、開発全体が完了する。

3.2.2 照査機能

マージの実行は、必ずマージ先(上位)ラインの管理者が行う。マージの実行者は、下位ラインで作成された成果物の品質が不十分と判断したらマージを拒否することもできる。

また、ラインをマージする前には、照査作業が義務づけられている。照査作業を開発者側ではない品質管理担当者などに依頼し、承認を得ることによって、第三者によ

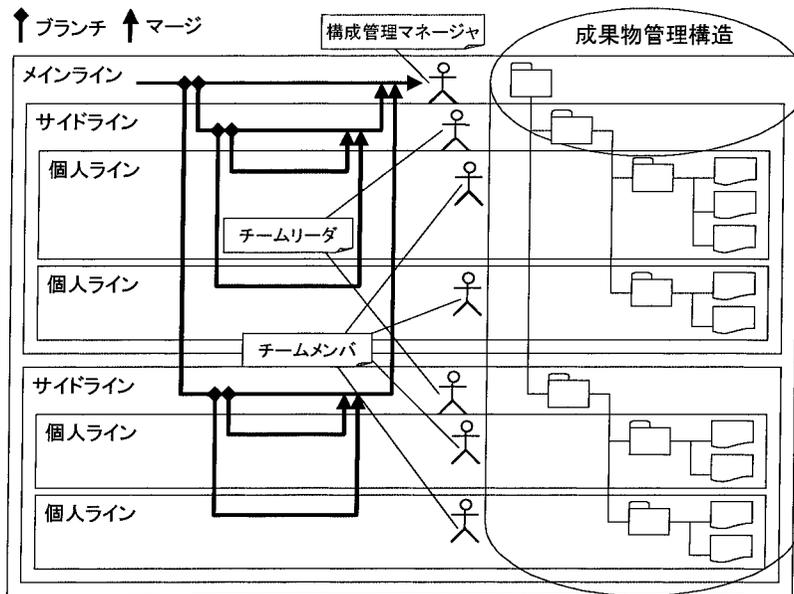


図 9 作業領域のイメージ

って品質が保証された成果物だけが上位に取り込まれていくことになる。

3.2.3 電子メール連携/アクション管理

ライン管理のコンセプトによると、作業の主体は、構成管理マネージャがサイドラインをブランチした時点でチームリーダーに移り、チームリーダーが個人ラインをブランチした時点でチームメンバに移る。逆に、作業の主体は、チームメンバが作業を完了し個人ラインのマージを要求した時点でチームリーダーに戻り、チームリーダーがサイドラインのマージを要求した時点で構成管理マネージャに戻ることになる。

TeamFactory は、このような制御が移るタイミングで、作業の関係者に自動的にメール通知を行い、必要な作業を行うように促してくれる。したがって、たとえメインラインやサイドラインの管理者であっても、下位の構成管理状況を常時監視する必要は無く、メール通知を受けた時だけ必要な作業を行えば、構成管理作業は円滑に進むことになる。また、メールでの通知内容は同時にアクション管理で記録される。したがって、アクション管理画面を常時開いておけば、メールを自由に使えない開発環境であっても必要な構成管理作業を認識できる。

3.3 TeamFactory 適用のポイント

チーム開発支援モデルで開発支援チームを設立し構成管理マネージャの任命を提唱しているように、TeamFactory でも構成管理マネージャの役割を担当する要員が必要になる（あくまでも役割であり他の役割と兼任してもよい）。そして、構成管理マネージャが TeamFactory 適用の基本方針を立案し、具体的な作業内容を決定する。決定した事項は、構成管理基準書（以降基準書と称す）として記述し、開発チームの合意を得て確定、開発者全員に周知・徹底する。本節では、TeamFactory ベースの基準書で規定すべき主な項目と記述内容の概略を述べる。

3.3.1 構成管理の基本方針

1) 構成管理における役割

TeamFactory では、「構成管理マネージャ」の他に、照査を実施する「照査者」を想定している。また、開発者側では、「チームリーダー」と「チームメンバ」という二つの役割を想定し、作業内容も異なってくる。それぞれにどのような機能・作業が求められるのかは、後述する方針・構成管理作業の中で述べる。

2) ライン管理の方針

まず、ラインの使い分け方について、方針を決める。ポイントとしては、メインラインとチーム用サイドラインの間に構成管理用サイドラインを入れるかどうか、またベースラインをどのようなタイミングで確定して行くかという2点になる。

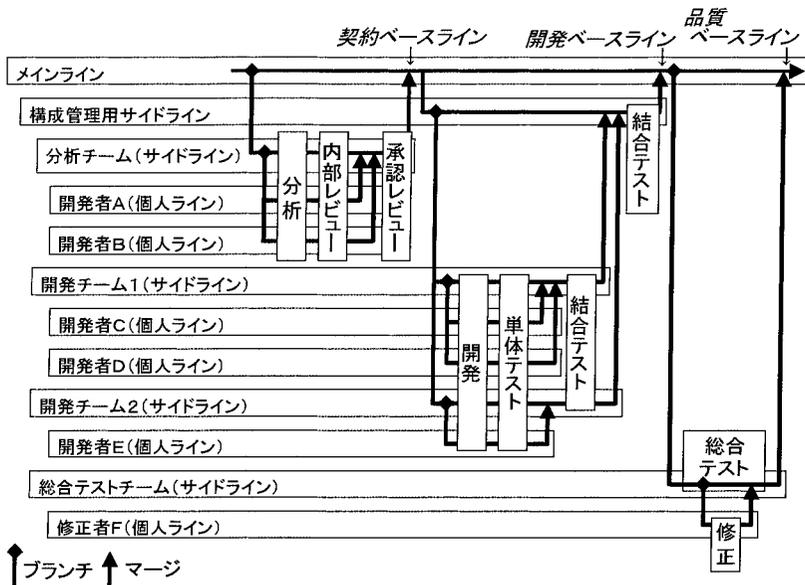


図 10 ライン管理の方針

図 10「ライン管理の方針」ではラインの使い分け方、およびベースライン確定のタイミングを例示している。この例では、実装フェーズを複数チームで作業するため、メインラインの直下に構成管理用サイドラインをブランチしている。この場合の各ラインの意味は、次のようになる。

- ・メインライン
構成管理マネージャが管理者となる。ベースライン確定専用のラインと位置付け、ベースラインを確定する時だけ下位のラインをマージして更新する。
- ・構成管理用サイドライン
構成管理マネージャが管理者となる。開発側が複数チームになる場合に、メインラインの下位に存在させ、複数チームの成果物をいきなりメインラインにマージするのを回避する。そして、全てのチームの成果物が構成管理用サイドラ

インに集まった状態で全体的なレビューやテストを行い、成果物の品質を確認してから、メインラインへのマージを実行する。

- ・チームリーダー用サイドライン

チームリーダーが管理者となる。チームで担当する成果物のみを作業範囲として割り当てる。やがてチームメンバの開発が完了し、当ラインにチーム内の全ての成果物が集まった段階で、レビューやチーム内結合テストを実施する。必ずチームレベルで成果物の品質を確認してから上位のラインにマージを要求する。

- ・チームメンバ用個人ライン

チームメンバが管理者となる。チームリーダー用サイドラインの下位ラインとしてブランチし、メンバが担当する成果物のみを作業範囲として割り当てる。各成果物に対してレビューや単体テストを実施し、必ずメンバレベルで品質を確認してからチームリーダー用サイドラインにマージを要求する。

また、ベースラインは、「契約ベースライン」、「開発ベースライン」、「品質ベースライン」を確定する。この場合の各ベースラインの意味は、次のようになる。

- ・契約ベースライン

分析チーム用サイドライン下で分析作業を進め、必要なレビュー、顧客の承認を受けて終了したら、メインラインへのマージを行って契約ベースライン確定する。要件定義書・論理設計書などの文書が成果物となり、ここでの確定内容で実装フェーズ以降の契約を交わし、実装に入ることになる。

- ・開発ベースライン

各開発チーム用サイドライン下で実装を進め、必要なレビュー、チーム内テストを完了したら、構成管理用サイドラインへのマージを行う。そこで、全体的な結合テストを実施し、完了したら、メインラインへのマージを行って開発ベースラインを確定する。この段階で文書に加え、プログラムやテスト報告書など、全ての成果物がメインラインに取り込まれる。ここでの確定内容で顧客の総合（受け入れ）テストを進めることになる。

- ・品質ベースライン

総合テストチーム用サイドライン上で顧客が総合テストを進め、バグが検出されたら修正用ラインをブランチして開発者側に修正させる。適宜修正内容をマージして全てのテストに合格したらメインラインにマージして品質ベースラインを確定する。

3) 成果物管理構造

プロジェクトの特性に応じて、成果物の格納場所がわかり易くなるように格納構造を定義する。図 11 の「成果物管理構造」は、LUCINA for Java/WLS による開発で、開発プロセス・作業ステップに準じた構造による定義例を示している。

4) ベースライン確定時の作業

ベースライン確定時に必要な作業を決めておく。次のような作業が、いかなるプロジェクトにおいても必要になると考える。

- ・メインラインへのマージ前の照査

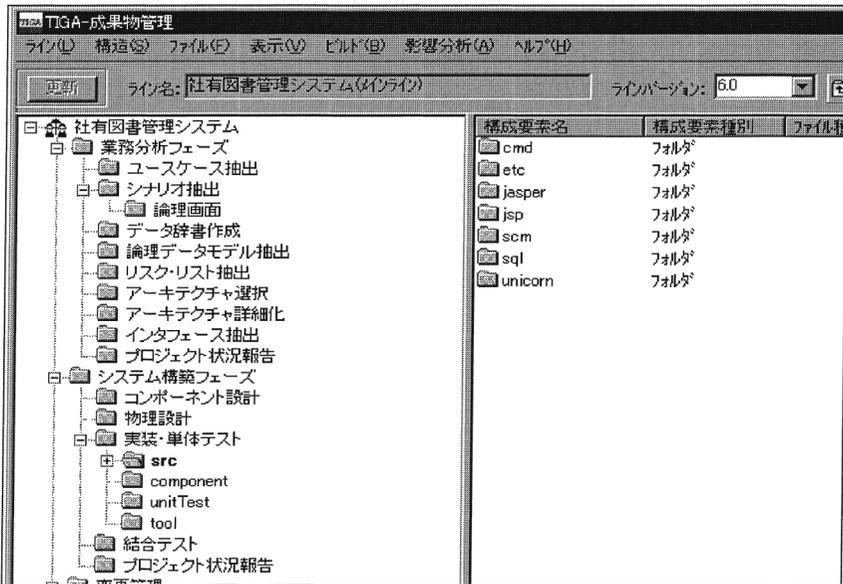


図 11 成果物管理構造

構成管理マネージャは、メインラインへのマージを要求された場合、マージを実行する前に、下位のラインの成果物が適切なレビューを受けているかを確認する。

- ・リリースバージョンの付加

構成管理マネージャは、マージを実行した後、TeamFactory の [リリースバージョンの付加] 機能を使用してベースライン名を付加する。そして、その旨を全チームリーダーに通知する。

3.3.2 構成管理作業

ここでは、これまで述べてきた方針を元に、役割ごとに実施すべき主な構成管理作業 (= TeamFactory の操作事項) と作業の流れを示す (図 12)。

図 12 では、構成管理マネージャ、チームリーダー、チームメンバ、照査者という役割ごとに必要な構成管理作業と、自動メール通知 (およびアクション管理機能) によって、作業の主体が移り変わることを表している。以下では、それぞれの作業内容について簡単に補足する。なお後述する各作業の番号は、各役割 (アクタ) ごとに派生する矢印の連番と対応している。

- 1) 構成管理マネージャの作業

構成管理マネージャは、以下の作業と、前述したベースライン確定時の作業を行う。

- ① 構成管理用サイドラインのランチ

複数チームで開発を進める場合は、メインラインの直下に構成管理用サイドラインをランチする。

- ② チームリーダー用サイドラインのランチ

チームリーダー一人一人に対して、サイドラインをランチし、各リーダーを管

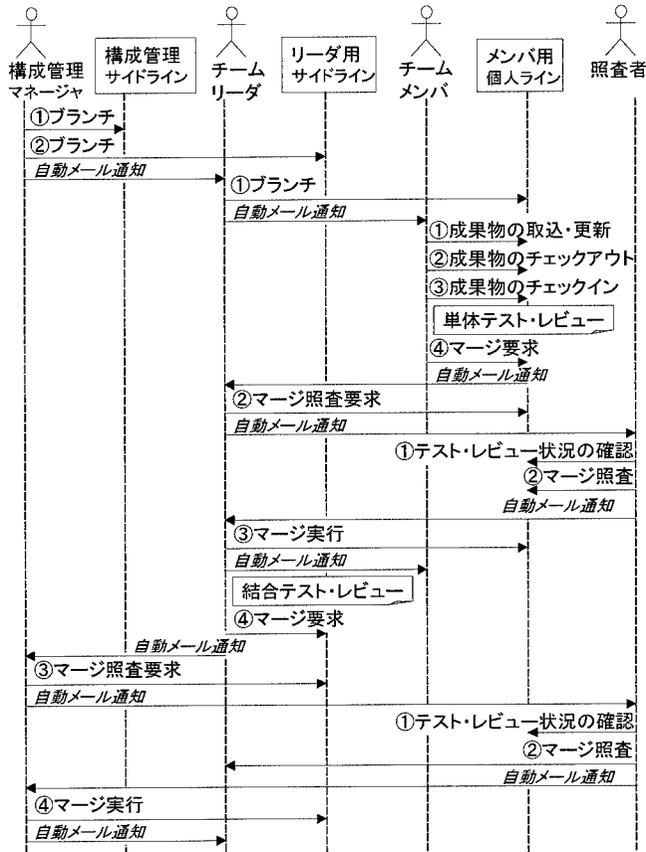


図 12 構成管理作業の流れ

理者に設定する。次に各チームが担当する作業範囲の割り当てと、チームリーダーに対して、作業開始を指示するメール/アクション通知が行われる。

(この後、各チームの開発が行われるので、各チームリーダーからマージ要求が依頼されるまで待機していればよい。マージ要求を受けたら次の作業が必要になる)

③ マージ照査要求

照査者に対して、マージ照査を依頼する。この作業を実行すると、照査者に対して、照査要求を依頼するメール/アクション通知が行われる。

(この後、照査者のマージ照査が行われ、完了すると次の作業が必要になる)

④ チームリーダー用サイドラインのマージ実行

照査者からのマージ照査承認通知を受けたら、実際にマージを実行する。(照査結果が却下の場合マージは実行できずマージ要求の取り消しが必要になる) マージが完了すると、チームリーダーに対してマージが完了した旨のメール/アクション通知が行われる。

2) チームリーダーの作業

チームリーダーは、チームリーダー用サイドラインがブランチされると、その旨の

メール/アクション通知を受けるので、その後以下の作業を実行する。

① チームメンバ用個人ラインのブランチ

チームメンバー一人一人に対して、個人ラインをブランチし、各メンバを管理者に設定する。次に各メンバが担当する作業範囲を割り当てると、チームメンバに対して、作業開始を指示するメール/アクション通知が行われる。

(この後、各チームメンバの開発作業が行われるので、各チームメンバからマージ要求が依頼されるまで待機していればよい。マージ要求を受けたら、次の作業が必要になる)

② マージ照査要求

照査者に対してマージ照査を依頼する。この作業を実行すると、照査者に対して、照査要求を依頼するメール/アクション通知が行われる。

(この後、照査者の作業が行われ、完了すると以降の作業が必要になる)

③ チームメンバ用個人ラインのマージ実行

照査者からのマージ照査承認通知を受けたら、実際にマージを実行する。(照査結果が却下の場合マージは実行できずマージ要求の取り消しが必要になる) マージが完了すると、チームメンバに対してマージが完了した旨のメール/アクション通知が行われる。

④ 上位ラインへのマージ要求

チーム内結合テスト/レビューを実施した後で実行する。マージ要求を実行すると、構成管理マネージャに対して、マージ要求を依頼するメール/アクション通知が行われる。

3) チームメンバの作業

チームメンバは、チームメンバ用個人ラインがブランチされると、その旨のメール/アクション通知を受けるので、その後以下の作業を実行する。

① 成果物の取り込み・更新

成果物を作成した後、個人ラインに成果物を取り込み、ラインを更新する。

② 成果物のチェックアウト

一旦取り込んだ成果物の変更が必要になったら、成果物を個人ラインからチェックアウトして変更する。

③ 成果物のチェックイン

成果物の変更が完了したら、成果物をチェックインしてラインに反映する。

④ チームリーダー用サイドラインへのマージ要求

単体テスト/レビューを実施した後で実行する。マージ要求を実行すると、チームリーダーに対して、マージ要求を依頼するメール/アクション通知が行われる。

4) 照査者の作業

照査者は、チームリーダーや構成管理マネージャからマージ照査要求が依頼されると、その旨のメール/アクション通知を受けるので、その後以下の作業を実行する。

① テスト・レビュー状況の確認

レビュー報告書やテスト報告書などを確認し、マージ要求しているライン上の成果物の品質を確認する。

② マージ照査

同一の成果物を、マージ元とマージ先双方で重複して更新していないかどうかを、TeamFactory が自動的にチェックしてくれるので、この内容を確認して承認すると、マージ照査の要求者（構成管理マネージャ、またはチームリーダー）に対して、照査を承認した旨のメール/アクション通知が行われる。

3.4 TeamFactory の強み

ライン管理により、チーム主導でチームの作業特性に合わせて構成管理を進めることができ、構成管理マネージャは、ベースライン確定に注力すればよい。構成管理マネージャ・チームリーダー・チームメンバといった構成管理上の役割が明確になる。

また、照査機能により、必ず上位の管理者や照査者により成果物を有効なものとするかどうかの判断が行われるので、品質の低下が起こりにくい。

さらに、メール連携/アクション通知機能により必要な通知が自動化されるので、構成管理作業の停滞を回避してくれる。

これらの基本的な能力に加え、TeamFactory が大規模システムの開発をより包括的に支援していくために、今後、次のような機能拡張を計画している。

大規模な開発では、複数のベンダーで開発を分担することが多く、開発場所が分散する可能性がある。構成管理環境を容易に分散/統合できる機能を追加する。また、チーム開発支援モデルで述べているように、成果物管理とプロセス管理を連動するために、他の変更管理ツールとの連携を図る。さらに将来的には、テスト・出荷管理・進捗管理などの機能を TeamFactory 自身に組み込むか、既存ツールと連携することを視野に入れている。

TeamFactory は、複数チーム・複数メンバによる開発を考慮したコンセプトをベースに、今後さらなる能力の成長を図りながら、チーム開発支援モデルの支援を目指していく。

4. おわりに

チーム開発支援モデルによって、プロジェクトによって異なるが多かった開発支援方法を具体的にイメージできるようになる。そして、チーム開発支援モデルに構成管理・変更管理等のツールを適用することで、支援作業の効率化・標準化を図り、プロジェクトを通じて一貫性のある支援体制の構築を実現できる。

TeamFactory は、ライン管理・照査機能・メール/アクション通知など、大規模なプロジェクトにおいても円滑な構成管理を実践するための能力を備えている。元々、中・大規模を想定する LUCINA 開発の支援ツールとして、TeamFactory は十分な素質を持っている。

近年 企業や組織のソフトウェア開発能力を明確にする動きが活発になりつつある。既に米国では、CMM (Capability Maturity Model)³⁾という指標が存在し、ここには能力をレベル付けするための様々な管理要件が定義されている。(米国防総省は CMM レベル 3 以上の認定がない企業の開発案件への入札を許可していない) また、

ISO/IEC では、開発プロセス評価の国際規格⁴が策定され、現在評価中である。今後は、単に技術力を磨けばよいということではなく、プロジェクト管理能力も含めた総合的なソフトウェア開発能力が、より重要なものとして位置付けられてくる。

このような状況に対応するためにも、LUCINA でのシステム開発が、自然にソフトウェア開発能力の高いレベルを維持していることが理想的である。チーム開発支援モデルや TeamFactory も、今後は CMM などをより強く意識した充実・発展が必要になるだろう。

-
- * 1 TEAMmethod: 情報化計画, ビジネスプロセスの再構築 (BPR), システム設計, システム構築およびプロジェクト管理といった、情報化のあらゆる局面を効果的に推進するための、手法、技法およびツールの体系。
 - * 2 TeamFactory: 成果物のバージョン管理, ベースライン管理を中心に、並行開発, 工程, 情報共有, ビルド連携といった、複数チーム, 複数メンバによるシステム開発を円滑に進めるための支援環境を提供するツール。

- 参考文献**
- [1] 杉野, LUCINA for Java/WebLogic Server の構造”, 技報 68 号, 2001.
 - [2] William J. Brown, and Hays W.“ Skip ” McCormick III, and Scott W. Thomas, “ AntiPatterns and Patterns in Software Configuration Management ” [岩谷宏訳, ソフトウェア構成管理の悪夢, ソフトバンクパブリッシング, 1999]
 - [3] Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber, Software Engineering Institute, Carnegie Mellon University (CMU) ” Capability Maturity Model Version 1.1 ”; SEA SPIN/CMM 研究会訳, ソフトウェア能力成熟度モデル 1.1 版, 1993.
 - [4] ISO/IEC, “ TR 15504 Software Process Assessment ”, 1998 (この日本語版が標準情報 TR X 0021)

執筆者紹介 橋 本 篤 (Atsushi Hashimoto)

1986 年中央大学卒業。同年日本ユニシス(株)入社。主に第四世代言語や CASE 等の開発ツールの適用・評価業務に従事。現在、E ビジネス技術部コンポーネント技術室に所属。