

ソフトウェア開発プロジェクト内での情報流通インフラの開発と評価

村田 大二郎[†] 工藤 裕[†] 平井 千秋[†] 伊野谷 祐二^{††} 三富 篤^{†††}

[†](株)日立製作所 システム開発研究所 ^{††}同 システムソリューショングループ
^{†††}同 ソフトウェア事業部

E-mail: {d-murata, yutaka, c-hirai}@sdl.hitachi.co.jp

本論文では、複数のサブグループが協調しつつ進行するプロジェクトの支援環境を提案する。具体的には大規模ソフトウェアの開発を対象とし、開発を複数サブグループで行った場合の成果物共有および全体進捗管理を支援する環境である。支援対象とした日立製作所では、組織内のソフトウェア開発プロセスの標準化が行なわれており、各サブグループは基本的に標準プロセスに従う。この点に着目し、組織内標準開発プロセスを、共有と管理の軸とするアプローチを取った。我々の方法では、開発プロセスをWBS (Work Breakdown Structure)で表し、各サブグループのWBSを標準WBSにマッピングするビューを提供する。このビューにより、他のサブグループとの情報共有および、全体プロジェクトの管理を支援することを狙う。開発した支援システムを紹介し、その評価について述べる。

Development and Evaluation of Information Sharing Infrastructure for Software Development Projects

Daijiro Murata[†], Yutaka Kudo[†], Chiaki Hirai[†], Yuji Inoya^{††}, Atushi Mitomi^{†††}

[†]Systems Development Laboratory, Hitachi, Ltd. ^{††}Solution Systems Group, Hitachi, Ltd.

^{†††}Software Division, Hitachi, Ltd.

E-mail: {d-murata, yutaka, c-hirai}@sdl.hitachi.co.jp

We developed an information sharing system for large-scale software development projects in which more than one sub-project works cooperatively. This system provides two scopes to share artifacts: the scope restricted to a single sub-project's artifacts, and that for the entire project. Arranging artifacts in a sub-project in order of the sub-project's Work Breakdown Structure (WBS), this system supports project members to focus on the sub-project's artifacts. The system also presents a view to show entire artifacts across sub-projects. This view is realized by linking the artifacts to the WBS that is generated by unifying all sub-groups' WBSs. In this paper we describe our system and discuss the evaluation of the system.

1 はじめに

近年、ソフトウェア開発企業では、組織内での開発プロセス標準化が進められている^[1]。開発プロセスを標準化して組織的に改善することで、ソフトウェアプロダクトの品質向上と開発コスト削減を狙っている^{[2][3]}。

標準プロセスを規定する為には、プロセスを明示的に書き表すことが必要になる。プロセスの表現手段として Work Breakdown Structure (WBS)がある。WBSは作業項目を製品の構造あるいは作業プロセスに沿って階層的に細分化したツリーである。WBSをプロジェクトのメンバが共有することで、メンバはどの仕事をすべきかを理解し、作業の

漏れをなくすることができる。

日立製作所においても、各ソフトウェア製品ドメインを担当する事業組織毎に標準プロセスを規定し、標準WBSを用いて開発プロジェクトの管理を行っている。

しかしWBSが規定しているのは成果物の種類であり、成果物の仕様に関する規定ではないため、作業間の仕様の整合性までは管理できない。ところが、ソフトウェア開発では、メンバー間で仕様に対する認識不一致が起こりやすいことが報告されている^[4]。

仕様に対する認識不一致の問題を解決するため、我々は、成果物を見つけやすくすることで、仕様を容易に参照可能にするアプローチをとった。メンバーが標準WBSを共有していることに着目し、標準WBSに成果物を関連付けて蓄積していく^[6]。プロジェクト内で成果物の情報を共有させることにより、仕様に対する認識の不一致を起こしにくくすることを狙ったものである。

WBSはまた、組織全体で標準化されているため、プロジェクトを超えて成果物を探すことも容易になる。他プロジェクトのノウハウを共有することで、プロジェクトの生産効率を向上できる^[5]。さらに、プロジェクト管理者が、成果物の完成度を参照することで、プロジェクト全体の進捗を把握できる。

我々は先にこの方法に基づき、WBSに沿ってプロジェクトの成果物を管理するシステムを構築し、ソフトウェア開発プロジェクトの情報共有を実現した^[6]。しかし大規模システムの開発にシステムを適用したところ問題点が判明したので、本稿ではその改良と適用について述べる。

2 我々のアプローチとその問題

2.1 我々のアプローチ

まず、当初の我々のアプローチを述べ、次節でその問題点を述べる。

我々は、WBSにプロジェクトで作成された成果物、参照される情報を関連付けて表示し、プロジェクトの情報を整理する方法をとった。

システムの画面を図1に示す。図1(1)にはプロジェクトのWBSが表示される。(1)においてワーク

を選択すると、(2)にそのワークで作成すべき成果物が、また、(3)に成果物を作成する為に参照すべき規則や、作業に関する情報が表示される。プロジェクトのメンバーはこの画面上でなすべき仕事を認識し、参照すべき情報を参照したり、成果物を登録したりできる^[6]。



図1 プロジェクト情報共有システム画面

2.2 我々のアプローチの問題

我々のアプローチでは、一つのプロジェクトに対して一つのWBSを用意し情報共有を行った。しかし、このアプローチでは大規模プロジェクトに対しては問題が起こった。

大規模システムの開発では、一つのプロジェクトを複数のサブプロジェクトに分け、それぞれのサブプロジェクトが協調することによりシステム全体を開発していく。典型的には、プロジェクトはサブシステム毎にサブプロジェクトに分けられていた。

このような大規模プロジェクトに2.1節で述べたアプローチを適用したところ、次の5つの問題が発生した。

- (1) プロジェクト全体を統括する管理者が、全成果物の完成度を一覧できない。その為に、プロジェクト全体の進捗を把握できない。
- (2) プロジェクト全体を統括する管理者が、サブプロジェクトの全仕様書を一覧できない為、サブシステム間の仕様不整合を発見し難い。
- (3) サブプロジェクトの管理者が、自分の管理するサブプロジェクトの作業範囲とその進捗を

把握できない。

- (4) サブプロジェクトの管理者が、全サブプロジェクトの仕様書を一覧できない為、自分の担当するサブシステムの仕様が他サブシステムの仕様と整合しているかを判断し難い。
- (5) サブプロジェクトのメンバがサブプロジェクトの作業範囲を把握できず、どの作業をすべきかがわかりにくくなる。

例として、図2に示すように、システムをサブシステムAとサブシステムBに分割し、それぞれ別のサブプロジェクトで設計、製造する場合を考える。一つのWBSでこのプロジェクトのプロセスを表す場合、WBSは図3(a)または(b)のようになる。

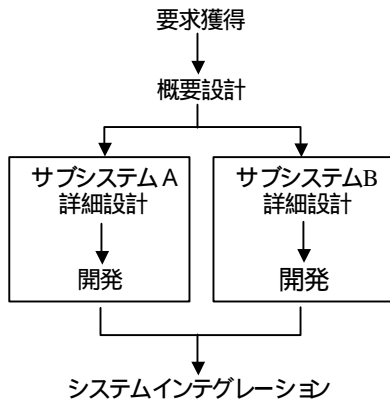
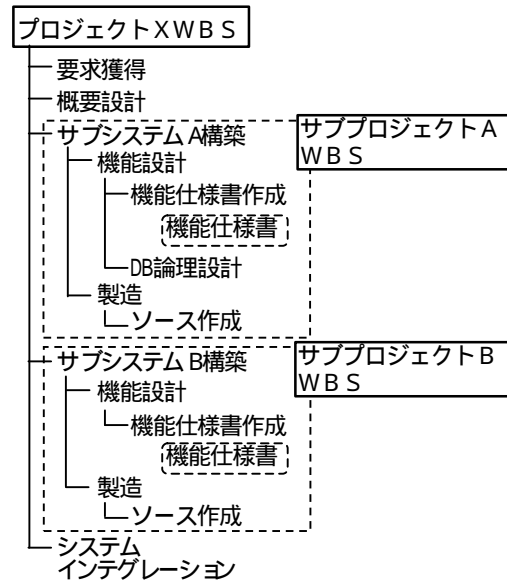


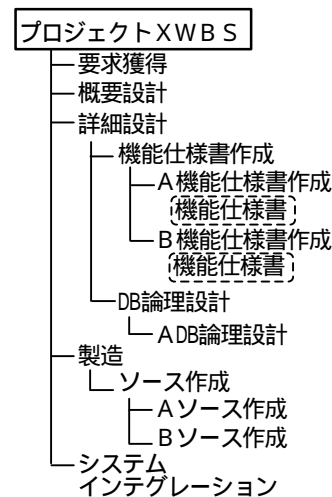
図2 大規模プロジェクトのプロセス例

図3(a)のWBSは、作業項目を、第一にシステムの構成に従って細分化し、次に作業手順の分類によって細分化したものである。「サブシステムA構築」と「サブシステムB構築」の各作業項目以下がそれぞれのサブプロジェクトのWBSとなる。この場合、上記(1),(2),(4)の問題が起こる。すなわち、各サブシステムの機能仕様書がWBS内で一個所にまとまらない為、システムXプロジェクト全体を統括する管理者は全機能仕様書の完成度を一覧できない。プロジェクト全体の進捗を把握し難く(問題(1))、また、仕様書の一覧性が損なわれているため、仕様の不整合が発見し難くなる(問題(2),(4))。

また、図3(b)のWBSはまず作業手順に従い細分化し、最後にシステムの構成に従い細分化したものである。この場合、上記(3),(5)の問題が起こる。すな



(a) システム構成により細分化



(b) 作業手順の分類により細分化

図3 単一WBSでの大規模プロジェクトのプロセスの表現

わち、サブプロジェクトAの作業項目がWBS内で一個所にまとまらない為、サブプロジェクトAの管理者はサブプロジェクトAの作業範囲と、サブプロジェクトAの進捗を把握できない(問題(3))。また、サブプロジェクトAの作業員は作業すべき項目をWBSから見つけ難い(問題(5))。

3 合成WBSによるサブプロジェクト情報共有方法

3.1 方法概要

我々は、上記(1)~(5)の問題を解決するため、複数サブプロジェクトのWBSからプロジェクト全体のWBSを合成する新たなアプローチを考案した。以下、合成WBSによるサブプロジェクト情報共有方法について述べる。

1章に述べた様にソフトウェア開発組織内でプロジェクトのプロセスは標準化されており、各サブプロジェクトは雛形とする標準WBSからコピーしたWBSを使用する。このため、サブプロジェクトのWBSの作業項目は、雛形のWBSの作業項目に一対一に対応する。

この対応関係を利用して各サブプロジェクトのWBSからプロジェクト全体のWBSを合成できる。すなわち、サブプロジェクトのWBSの各作業項目に関連する成果物を、雛形のWBSの対応する作業項目に関連付けることで、雛形のWBSをプロジェ

クト全体のWBSとできる。このWBSの作業項目を参照すると、全サブプロジェクトの対応する作業項目の成果物を一覧できるようにした。

図4の例では、サブプロジェクトA、BのWBSから合成したWBSを、プロジェクトX全体のWBSとする。プロジェクトXのWBSの作業項目「機能仕様書設計」を参照すると、サブシステムA、Bの機能仕様書を一覧できる。

合成WBSと、サブプロジェクトのWBSの二つにより、同じ情報に対し異なった見方ができることが本方法の特徴である。

実際の運用では、サブシステムの構築を別会社に発注することがあり、この場合、サブプロジェクト管理者がプロジェクト全体を参照できてはいけなことがある。サブプロジェクト管理者にサブプロジェクトWBSのみへのアクセスを許可し、合成WBSへのアクセスを許可しないセキュリティポリシーを定義できるようにし、このような場合にも対応できるようにした。

3.2 利点

本サブプロジェクト情報共有方法の利点は、一つのWBSで大規模プロジェクトを管理した場合の問題(1)~(5)を解決できることである。

合成WBSを参照することにより、各サブプロジェクトの対応する作業項目の成果物を一覧できる。したがってプロジェクトとサブプロジェクトの管理者は進捗の把握、仕様の比較が可能となり、問題(1),(2),(4)を解決できる。

また、各サブプロジェクトのWBSを参照することにより、サブプロジェクトの作業範囲の把握が可能になり、問題(3),(5)を解決できる。

4 サブプロジェクト情報共有支援システム

我々はプロジェクト情報共有支援システムを改良し、本方法に基づくサブプロジェクト情報の共有を可能にした。

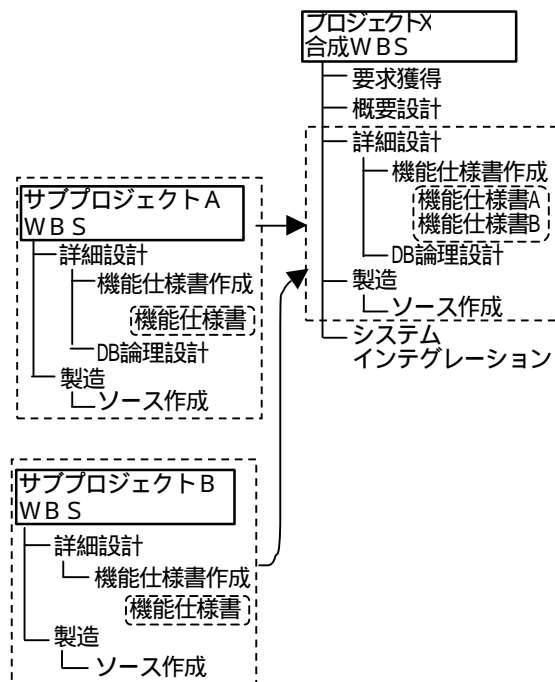


図4 WBSの合成

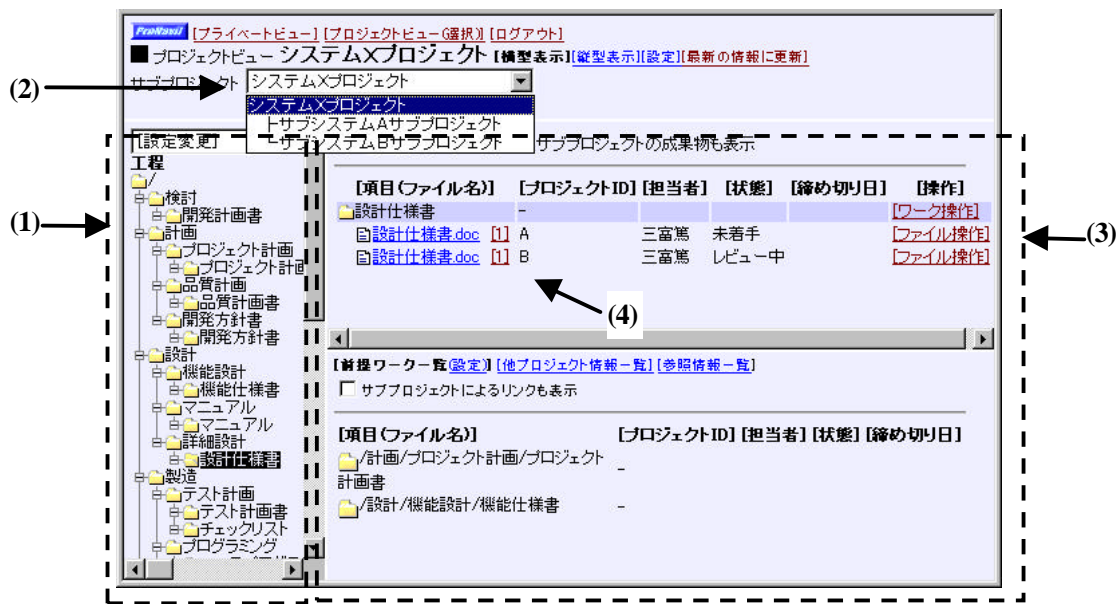


図5 サブプロジェクト情報共有システム画面

図5に改良したサブプロジェクト情報共有支援システムのメイン画面を示す。(2)に示すメニューでプロジェクト全体を選択し、プロジェクト全体の情報を表示するよう指定した場合は、左フレーム(図5(1))にプロジェクト全体の合成WBSが表示される。左フレームで作業項目を選択すると、右フレーム内にその作業項目に関連付けられた全サブプロジェクトの成果物が集約して表示される。成果物名をクリックすることによりファイルを参照できる。図5の例では、サブプロジェクトA、Bの設計仕様書ファイル名と、その完成度が表示されている。ファイル名をクリックすると、それぞれの設計仕様書が表示される。

プロジェクト全体の成果物を表示する場合、それぞれの成果物がどのサブプロジェクトに属するかを判別できる必要がある。そのため、図5(4)に示すように、成果物名の横にサブプロジェクトを示すIDを表示する。図5の例では、上から二番目の設計仕様書がサブプロジェクトBで作成されたものであることがわかる。

一つのサブプロジェクトの作業範囲を把握する場合は、図5(2)に示すメニューでサブプロジェクト名を選択する。サブプロジェクトのWBSが左フレ

ーム(図5(2))に表示され、左フレームで作業項目を選択すると、右フレーム(図5(3))に指定したサブプロジェクトのみの成果物が表示される。

セキュリティポリシーを実現する為、プロジェクト管理者はユーザのアクセス権限をサブプロジェクト単位で設定できる。

5 システム適用及び結果

5.1 適用状況

上記のシステムを、日立製作所社内の約100のソフトウェア開発プロジェクトに対し適用している。

5.2 ユーザビリティ評価

本システムはプロジェクトメンバが日常的に使用するポータルであり使い勝手が重要となる為、システムのユーザビリティの評価を行った。

サブプロジェクトの作業者、サブプロジェクトの管理者、全体を統括するプロジェクト管理者の各役割で作業する8人のユーザを選び、それぞれに対しアンケートと、一人2時間のインタビューを実施した。

その結果、システムの必要性については多くの賛

同を得た。また、開発現場でのシステムの利用のされ方と、開発現場のシステムへの要望が明らかになった。いくつかの要望については既に対策を完了した。以下に要望の一部を記す。

- ・システムからのログアウト時に最後に参照した作業項目を、次にログインした時に最初に表示するようにした。これにより、システムを使用する都度、WBSの上位からたどって作業中の項目を見つけるのではなく、ログイン後にすべき作業項目を直接参照できるようにした。
- ・メイン画面内にユーザが真に必要と思っている項目を残し、そうでない項目を除去することで一覧性を向上させた。

5.3 適用効果の定量的評価の方針

現在は効果についての定量的な評価はまだ実施していないが、評価方法としては、以下の方法を考えている。

- 1) 本システムを使用するプロジェクトと使用しないプロジェクトにおいて、工程遅延に有意な差があるかを調査する。このことにより、プロジェクト全体を統括する管理者とサブプロジェクトの管理者が、それぞれが担当するプロジェクトの進捗を参照し問題に早期に対処できたかを検証する。
- 2) 本システムを使用するプロジェクトと使用しないプロジェクトについて、仕様の不整合に基づく不良件数を比較する。このことにより、プロジェクト全体を統括する管理者とサブプロジェクトの管理者が、仕様の不整合を把握できたかを検証する。
- 3) 本システムと、本方法適用前のシステムのログを比較し、作業者が作業終了後次の作業に取り掛かるまでの時間に有意な差があるかを調査する。このことにより、作業者が次にすべき作業を的確に判断できたかを検証する。

6 おわりに

本論文では、複数のサブプロジェクトからなる大

規模プロジェクトにおいて、メンバがプロジェクト全体の情報とサブプロジェクト毎の情報の共有を行う為に、各サブプロジェクトのWBSとそれを一つに合成したプロジェクト全体のWBSの2種類のWBSを利用する方法を提案した。本方法により、プロジェクトとサブプロジェクトの進捗、また、サブシステム間の仕様の不整合が把握できると考える。

また、上記方法を支援するシステムを提案し、現在の適用状況を報告した。今後、開発現場での効果の定量的評価を行っていく。

参考文献

- [1] Rada, R., et al.: Standardizing Software Projects, Communications of the ACM, vol.43, no.12, pp.21-25 (2000).
- [2] Humphrey, W. S.: Managing the Software Process, Addison-Wesley (1989) 邦訳 藤野監訳:ソフトウェアプロセスの成熟度の改善, 日科技連 (1991).
- [3] 福山峻一, 高木英雄, 田中僚史, 渡辺道広, 中林效: ソフトウェアプロセスの持続的な改善を誘導するチェックリストの実装手順, 情報処理学会論文誌, vol.42, no.3, pp.529-541 (2001).
- [4] 海谷治彦: 代案発生に注目したソフトウェアの共同設計作業における認識不一致の分析, 情報処理学会論文誌, vol.40, no.11, pp.3878-3886 (1999).
- [5] 平井千秋, 工藤裕, 降旗由香理: ナレッジマネジメントのソフトウェア開発への適用, 人工知能学会誌, vol.16, no.1, pp.59-63 (2001).
- [6] 三富篤, 谷田耕救, 平井千秋, 山形尚美, 村田大二郎: ソフトウェアプロセス成熟度の改善を目指したプロジェクトポータルの開発, 情報処理学会第 62 回全国大会論文集 分冊(特1), pp.275-278 (2001).