March 31, 2009
RT5299
Communications; Engineering Technology 6 pages

Research Report

An efficient progress management approach by subjective maturity using check list

Hirofumi Matsuzawa, Atsushi Fukuda, Takashi Nerome

IBM Research, Tokyo Research Laboratory
IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato
Kanagawa 242-8502, Japan

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

チェックリストに基づく客観的成熟度を用いたプロセス進捗管理の効率化手法

松澤 裕史　福田　敦史　根路銘　崇

日本アイ・ビー・エム株式会社

E-mail:　{ matuzawa, atfukuda, nerome} @jp.ibm.com,

キーワード　進捗管理, 成熟度，チェックリスト

Abstract

In this report, we describe an efficient method of work progress management using an objective maturity based on check list. In recent business activities of development project in industrial area and so forth, Work progress reports which are submitted by engineer depend on senses of engineer. This situation causes a problem that project manager can hardly recognize actual progress. Therefore we propose a new efficient method to use not only subjective value input by engineer but also objective value which is predicted using accumulated check list values.

1. はじめに

製造業などにおける開発プロセスの管理において，製品開発のプロセス進捗管理は非常に重要である．開発作業は，複数のグループに分かれ，並行して実施される．しかし，自分たちの作業がどこまで作業が進んでいるのか，他のグループに影響を及ぼす作業がどこまで進んでいるのか，他のグループの作業に影響があるような寸法等の設計情報がどこまで決まっているかということを管理することは非常に重要である．

例えば、自動車会社における開発プロセスなどでは，同時に作業可能なプロセスを並行的に実施する同時並行処理実行が進み，開発期間の短縮を行っている．一方で，同時並行的に進められている他のタスクと成果物の情報を共有する事を目的として，開発フェーズ毎でエンジニア同士が作業進捗会議などにより，全体の（相互の）進捗を確認する手段（大部屋方式）がとられている．

このような並行開発プロセスに対して, タスクや成果物の進捗管理が課題となっている. その方法として, WBS（Work Break-down Structure）や PBS　(Product Break-down Structure)　の 2 軸による管理方法などが用いられている．

関連手法として ODMM[1]では、開発中に，エンジニアによって入力した成果物の成熟度を用いて，必要な作業変更の影響に対する今後のスケジューリングへの影響度及びリソースの影響度の算定を行う．これにより，開発プロセス全体の管理を効率化するというようなアプローチが用いられている．しかしながら，エンジニアが成熟度として入力するタスク単位の進捗情報は，粒度が粗く，主観的な数値であるが故に，全く同じ成果物に対する進捗情報であっても，エンジニアの熟練度合いや後工程への影響度合い，さらには，エンジニアの性格が楽観的か悲観的によっても異なってくる性質がある．このように，主観的な数値で入力される成熟度については，人により，ばらついてしまうという性質がある．ゆえに，開発プロセスや成果物の管理ツールの中で扱う情報としては，成熟度の精度はそれほど高くないというのが課題である．

本稿では，プロセス管理や成果物管理で扱うべき成熟度の人によるばらつきを補正し，より客観的に取得する方法について述べる．

2. チェックリストを用いた開発プロセス

本研究では，開発方法論に従った開発手順のガイダンス情報や，入出力となる開発成果物を示す開発定義書を用いてエンジニアが開発を進めていく開発プロセスを前提とする．個々のエンジニアが開発を終了する際に，チェックリストをチェックし，進捗度として主観的な数値を入力する．チェックリストは，開発定義書に沿って作成されるが，開発定義書自体は複数の開発で利用されることとし，汎用性の高いものとする．本研究にてチェックリストを用いた開発プロセスで利用される成果物とチェック方法について以下に示していく．

2.1. 開発定義書の例

チェック項目が示された開発定義書の例を，図 1 に示す．開発定義書は，開発の標準化として開発に携わるエンジニアが全員参照するべきものとする．

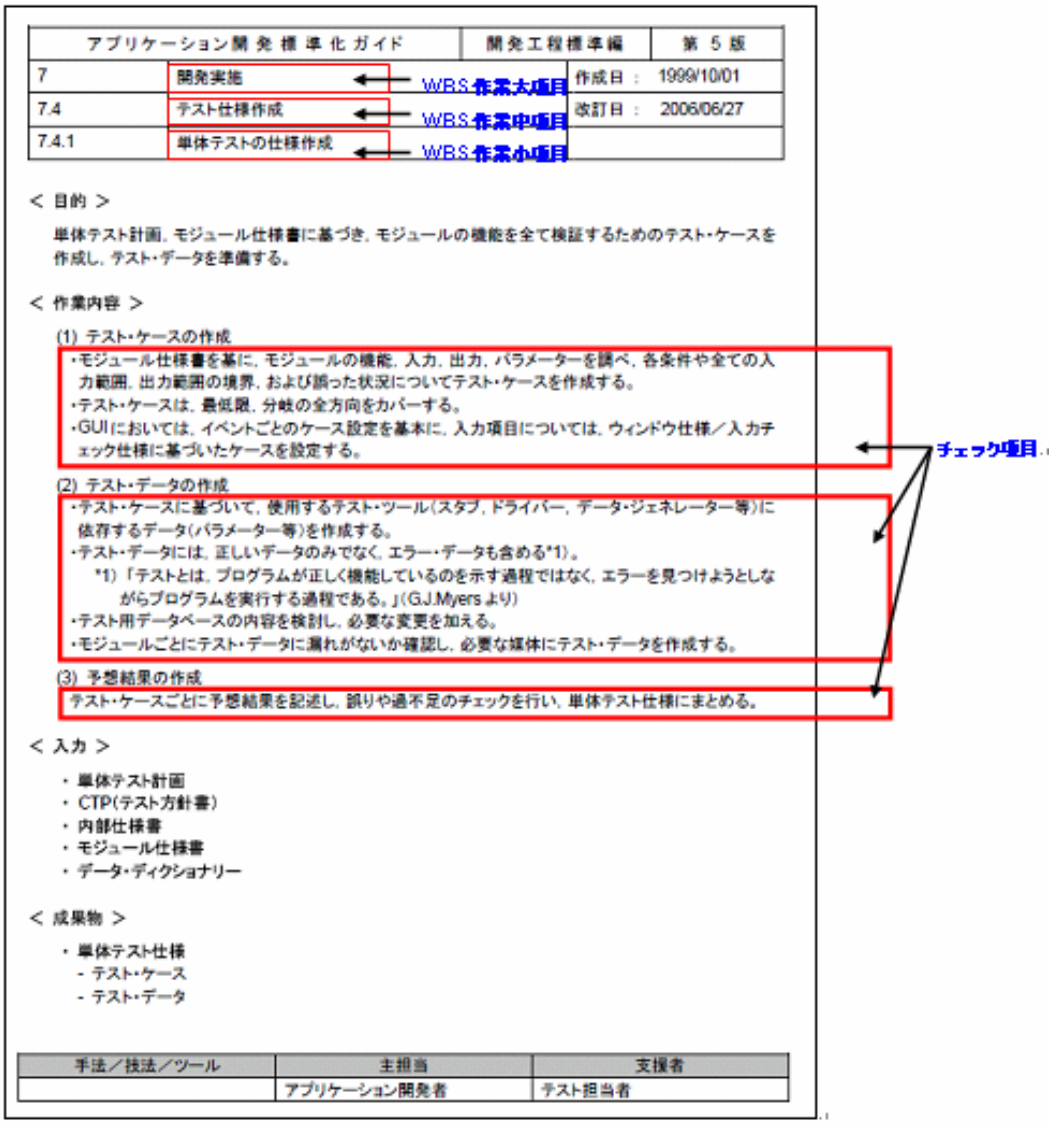

図 1　開発定義書の例

図 1 では，アプリケーション開発標準化ガイドの例を示しているが，開発プロセスとしての開発作業の位置づけを，大項目，中項目，小項目の 3 階層で示している．それぞれの例は，”開発実施”，”テスト仕様作成”，”単体テストの仕様作成”と記されている．また，<作業内容>として定義されている箇条書きの項目は，チェックリスト作成のためのチェック項目とみなす．

このように開発定義書から，WBS に関する分類情報とチェックリストの候補を取り出していくこととする．

2.2. 開発定義書に沿った WBS の例

開発定義書に従った WBS の例について図 2 に示す．

作業大項目	作業中項目	作業項目ID	作業小項目	開発対象モジュール	担当者	成熟度/チェック項目入力
開発実施	モジュール開発	1	モジュール仕様書の作成	ログイン画面表示機能	Aさん	
				ログイン情報検索機能	Bさん	
				顧客情報登録機能	Dさん	
				購入履歴情報取得機能	Fさん	
		2	モジュールのコーディング	ログイン画面表示機能	Aさん	
				ログイン情報検索機能	Bさん	
				顧客情報登録機能	Dさん	
				購入履歴情報取得機能	Fさん	
	テスト仕様作成	3	単体テストの仕様作成	ログイン画面表示機能	Gさん	
				ログイン情報検索機能	Gさん	
				顧客情報登録機能	Hさん	
				購入履歴情報取得機能	Hさん	
		4	統合テストaの仕様作成	ログイン機能	Iさん	
				顧客情報登録機能	Iさん	
				購入履歴検索機能	Iさん	

図 2　WBS の例

開発責任者により WBS は作成される．担当者の割り当てや開発作業の指示，開発作業の終了後には進捗を管理する．図 2 は，図 1 の開発定義書に沿って，作業大項目，作業中項目，作業小項目，開発対象モジュール，担当者が記入された例である．成熟度/チェック項目入力欄は開発作業の終了後に開発責任者によって入力される．開発責任者は，開発定義書以外の要件に合わせて，枠で示す”単体テストの仕様作成”のように開発作業を追加更新する．また，作業項目 ID 欄には，作業項目単位でユニークになる数字や記号などを入力する．今回の例では，開発作業を構成する作業項目の最小単位を開発対象モジュールとしており、これは例えば作業小項目が”単体テストの仕様作成”で，開発対象モジュールが”ログイン画面表示機能”の場合，ログイン画面機能についての単体テストの仕様作成を実作業として行うことを表している．

担当者の割り当てに関して，WBS の最小項目単位でタスクを実施するのに適当な担当者の名前を入力することになるが，ここで割り当てられた担当者が作業を実施の上で成熟度及びチェック項目の入力を行うことになる．

成熟度/チェック項目入力欄においては、この欄をクリックすると後に述べるチェック項目や成熟度を入力するフォームが表示されるようになる．

2.3. 開発定義書に沿ったチェック項目の例

開発定義書に基づき，更に開発作業の作業小項目に沿ったチェック項目を用意する（図 3 を参照）．チェック項目は，作業項目 ID，チェック項目内容で構成される． 作業項目 ID には，各 WBS 作業小項目の単位でユニークに決められた番号や記号を入力する．チェック項目内容に開発プロセス定義書の作業内容にある項目を入力する．

作業項目ID	チェック項目内容
1	内部モジュール分割を行い，内部モジュール構造図を作成した
1	内部モジュールごとに，その内容を構造化文，アクション・ダイアグラム，シュード・コード，デシジョン・テーブル等を用いて記述した
1	モジュール仕様書のウォークスルー／インスペクションを行った
2	内部仕様書／モジュール仕様書に従って，モジュールをコーディングを行った
2	モジュール単位に単体テストをインクリメンタルに実施し，テスト・ケースごとに予想結果と実際の結果とを対比させ，検証を行った
3	モジュール仕様書を基に，モジュールの機能，入力，出力，パラメーターを調べ，各条件や全ての入力範囲，出力範囲の境界，および誤った状況についてテスト・ケースを作成した
3	GUIにおいては，イベントごとのケース設定を基本に，入力項目については，ウィンドウ仕様／入力チェック仕様に基づいたケースを設定した。
3	テストケースは最低限、分岐の全方向を力バーした
3	テスト・ケースに基づいて，使用するテスト・ツール（スタブ，ドライバー，データ・ジェネレーター等）に依存するデータ（パラメーター等）を作成した
3	テスト・データには，正しいデータのみでなく，エラー・データも含めた
3	テスト用データベースの内容を検討し，必要な変更を加えた
3	テスト・ケースごとに予想結果を記述し，誤りや過不足のチェックを行い，単体テスト仕様にまとめた

図 3 チェック項目の例

2.4. 開発作業のチェック項目/成熟度入力フォーム

2.1 および 2.2 で述べた WBS 及びチェック項目を用いて，担当者は自分に割り当てられたタスクを実施し，その結果を基にチェック項目及び成熟度の入力を行う．その際の入力フォームは図 4 のようになる．

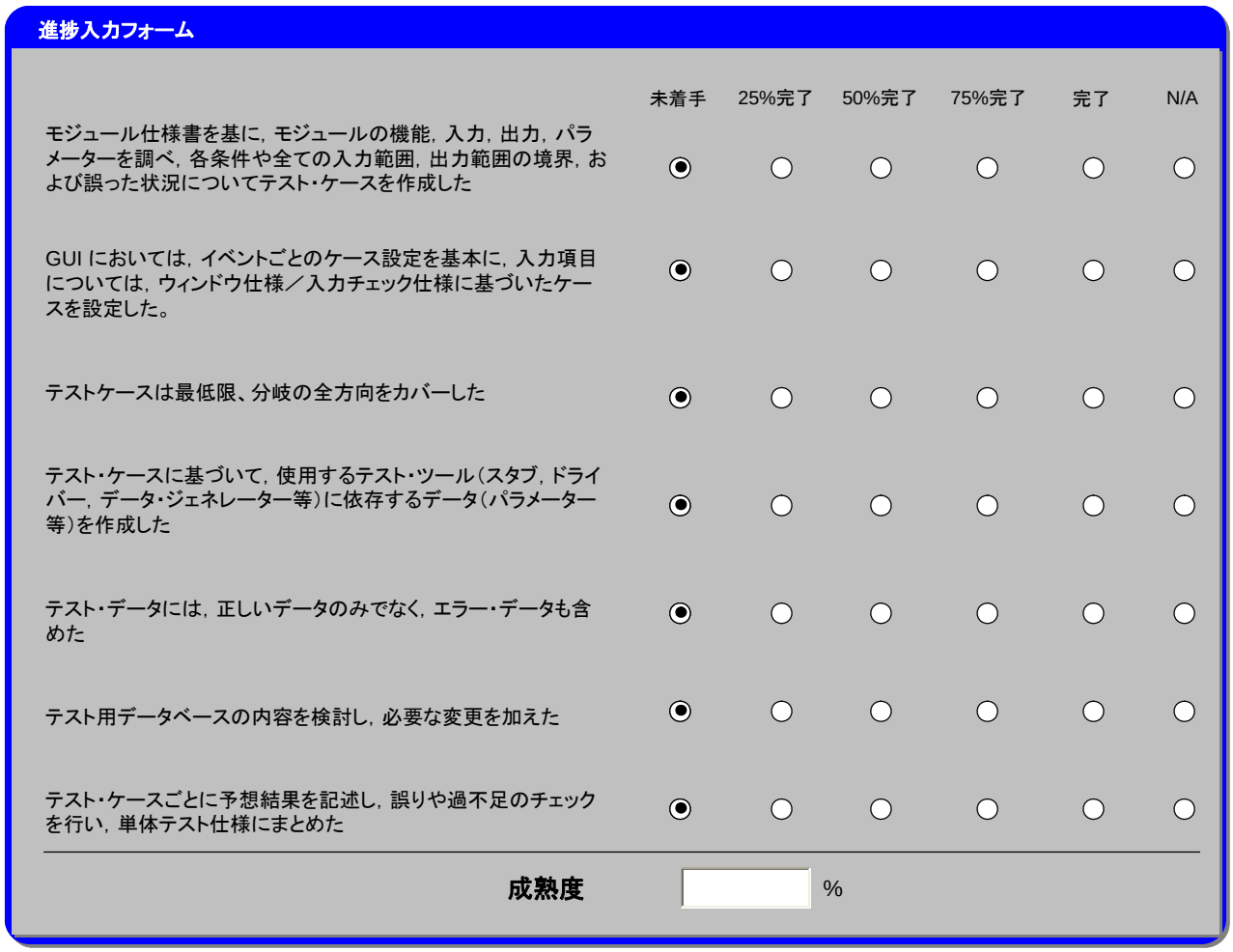

図 4　進捗入力フォームの例

図 4 のフォームには，各チェック項目と、各項目に対しての評価値の入力欄，成熟度入力欄が存在する．各項目に対しての評価値として，この例では未着手，25%完了，50%完了，75%完了，完了，N/A (適合しない)を入力させる．この際，次に示す手順により，フォーム上のチェック項目として各作業項目に適したものを取得し，画面上に表示する必要がある．

1. ユーザーによって選択された作業項目の作業項目 ID を取得
2. 取得した作業項目 ID を基に図 4 にあるチェック項目から同じ作業項目 ID をもつチェック項目群を取得
3. 取得した項目をチェック項目/成熟度入力フォーム上の項目として表示

これらの開発定義書，WBS，チェック項目に基づき開発担当者は開発を行い，進捗フォームにより，結果の進捗を記す．しかし，ここで得られる成熟度は，開発担当者の主観的な成熟度であると言える．故に，開発者のスキルや価値観により値が異なってくる可能性がある．次章では，客観的な成熟度に補正する方法について述べる．

3. 客観的成熟度へ補正する方法の検討

本研究では，開発プロセス進捗管理の効率化のために，２章で示したようにエンジニアが入力する成熟度のばらつきを補正するアプローチを検討した．全体の概要を図 5 に示している．手順は以下のとおりである．

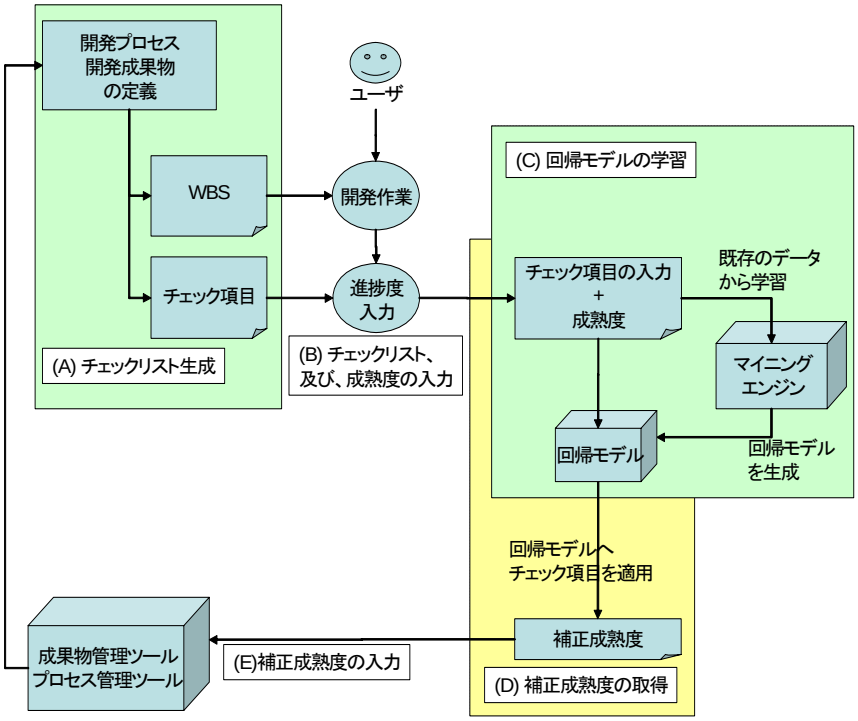

図 5　全体の概要

（A）　開発成果物や開発プロセスの定義からの　チェックリスト生成

2 章で示す手段で，WBS 及びチェック項目を生成する．チェック項目の性質として，単一開発プロセスでのみ利用可能な項目ではなく，多くのプロセスで共通で利用可能かつ，客観的に入力できる項目であることが前提として挙げられる．

（B）　チェックリスト，及び，成熟度の入力

ここでは，開発担当者が，実際に開発作業を行い，日々の（あるいは週毎の）進捗について，チェック項目への入力、(主観的でいいので）成熟度をシステムへ入力を行っていく．

（C）　回帰モデルの学習

開発担当者によって，チェックリストと成熟度が入力されるので，これらに対して，チェックリストを説明属性として，成熟度を目的属性とし，回帰モデルを生成する．本研究では，回帰木を想定しているが，実際にはロジステック回帰など他の学習アルゴリズムを用いることも可能である．

（D）　補正成熟度の取得

開発担当者が入力したチェックリストを（C）で生成された回帰木に適用する．回帰木からは成熟度が出力される．(開発担当者は，チェックリストと同時に成熟度の入力も行っているが，これは，回帰木の適用には用いない．ただし，出力された(補正)成熟度との差が大きい時にアラーとを上げたり，エンジニアに再入力を促すなどの使い方が，アプリケーションによってはあり得る．

（E）　補正成熟度の入力

成果物管理システム，または，プロセス管理システムに補正後の成熟度を入力する．入力された成熟度の利用方法については，各ツール（管理システム）に依存する．本稿では，利用方法については省略する．

運用上は，（A）⇒（B）⇒（D）⇒（E）という手順で，日々の作業が行われる．（C）の作業は，ある程度のチェックリスト＋成熟度のデータ件数が蓄積されてから実施される．また，エンジニアの熟練度も向上してくることも予想されるので，時折，（C）の作業を行って回帰木を更新することで，より精度の高い補正が行われるようになる．

4. まとめ

従来のように，成熟度のみを扱う場合では，各エンジニアが申告する成熟度の値がばらついてしまうため，精度の問題が発生する恐れがある．これに対して，チェックリストの導入を行うことにより，成果物の進捗を項目ごとに把握することができるようになる．しかしながら，チェックリストの内容は，成果物管理やプロセス管理ツールへの入力として，一般的には扱うことが出来ないことにより，ここから成熟度として，これを数値化する必要がある．

成熟度は主観的な入力であるため，チェックリストの入力が同じであっても，異なる成熟度が申告される恐れがあるが，客観的に入力できるようになっているチェックリストから回帰木を生成して，これを用いて成熟度を予測（補正）させることにより，同じチェックリストからは同じ成熟度が値として取り出せるようになる，また，エンジニアによるばらつきを解消が可能となる．結果として，後工程にあるプロセス管理システムや成果物管理システムの精度の向上が図られる．

今後は，客観的成熟度の補正による進捗管理の効果を狙い，実際に複数の開発プロジェクトに適用していきたい．

5. 参考文献

[1] “Work management system, work management system construction support service, control method and program “, USPTO Application #: 20090043618, US Patent