

定量的プロジェクト管理のための開発計画作成支援システム —要求品質決定ツールの作成—

牛尾 亘綱[†] 築場 豊[‡] 松本 健一[†] 鳥居 宏次[†]

大阪大学 基礎工学部[†] (株) SRA[‡]

あらまし

ソフトウェア開発のプロジェクト管理における開発計画の作成では、ユーザの要求品質を明確に把握して開発技術の選択を行うソフトウェア品質定量化の枠組が提案されている。この枠組に基づく開発計画作成支援システム SQUARE では、ユーザの要求する品質特性の強さを表す数値を入力としているが、用語の抽象度が高い、ユーザの階層による品質の捉え方の違いに対応できない、ユーザが品質特性の依存関係を考慮しにくい、等の問題点があった。その解決のため、SQUARE のサブシステムとして要求品質決定ツールを開発した。本ツールを用いると、どの階層のユーザでも、データベース化された具体的表現の 70 個の質問に選択枝で答えることによって要求品質を決定することができる。

Project Planning Supporting System for Quantitative Project Management —Development of Quality Requirements Decision Tool—

Watatsuna USHIO[†], Yutaka YANABA[‡], Ken-ichi MATSUMOTO[†], and Koji TORII[†]

Faculty of Engineering Science, Osaka University[†] SRA[‡]

Abstract

A New Framework for Software Quality Assurance has been proposed, which helps make a project plan by selecting the most suitable activities to realize user's quality requirements. The SQUARE system, based on this framework, requests users to set up each quality factor's value, which indicates its seriousness, however, some problems have been left unsolved. They are: abstractness of terms for the factors, ignored difference of viewpoints between users on quality factors, and incomprehensible dependencies between factors to users. For the solution, we developed a quality requirements decision tool, as a sub-system of SQUARE, with which any kind of users can decide definitely their quality requirements, just by answering 70 concrete, comprehensible alternative questions, that are databased.

1. まえがき

高度情報化社会の進展に伴い、近年開発されるソフトウェアはますます大規模化、複雑化、多様化してきている⁽⁵⁾。これに伴って、プロジェクト管理の必要性が高まっているが、効果的なプロジェクト管理方法は確立しておらず、プロジェクト管理をうまく行うことは一種の芸術である⁽²⁾、と言っていいほど困難な問題となっている。

プロジェクト管理の主要な項目の一つである「開発計画の作成」に限っても、(1)ソフトウェアに対するユーザの要求を正しく把握することが難しい、(2)開発計画の作成が不完全であることや、全くなされないことが多い、(3)開発に要するコストやスケジュールの見積りが難しい、(4)プロジェクトに最適な開発技術(例えば、設計技法、テスト技法)を選択する基準がない、等の問題点がある⁽³⁾。

通常、ソフトウェア開発プロジェクトは、ユーザの要求分析から始められる。この時、要求内容から実現すべきプロダクトの機能だけでなく品質(例えば、性能)も正確に把握することが必要である。一方、開発者(設計者、プログラマ等)は、ユーザに要求された機能と品質をプロダクトに実現しなくてはならない。そのためには、品質や機能の実現に有効な開発技術を明確に示した開発計画を作成する必要がある。

我々は、上述した考えに基づいて、品質特性を開発技術へ定量的に対応付け、開発計画に反映させるための枠組(ソフトウェア品質定量化の枠組)を提案している⁽¹²⁾。更に、枠組に基づく開発計画作成支援システム SQUARE を開発し、実際の開発過程に適用して評価実験を行ってきた⁽¹²⁾。提案した枠組は、ソフトウェア品質をユーザ、管理者、開発者の3つの視点で定量的に捉え、ユーザ視点で捉えたプロダクトの品質を、開発者視点で捉えたプロセスの品質に変換するという、体系的なプロジェクト管理のための枠組を提供する。しかし、実際のプロジェクトに適用した結果、次の(1)~(3)の問題点が明らかになってきた。

(1)プロダクトの要求品質特性(Factor 及び Sub factor)が、ISO の分類という専門的な用語で表現されているため、ソフトウェアの機能や性質を熟知しないユーザには、要求すべき品質がどの品質特性に属するのかがわかりにくい。(2)品質特性は互いに独立ではなく、ある特性の実

現が他の特性に影響を与えることがある(ユーザの一つの具体的な要求が、複数の Sub factor にまたがる要求であることが多い)。(3)ソフトウェアの用途によってユーザの品質に対する捉え方が異なり、要求品質を一つの基準で選択させることが困難である。

本研究では、以上の問題を解決するため、SQUARE のサブシステムとして要求品質決定ツールを開発した。これは、あらかじめ設定された、品質に関する具体的な表現の質問に対するユーザの回答から、要求品質特性を得るものである。質問は約 70 項目あり、従来の ISO の 21 種類の Sub factor に多対多の対応をしている。

ユーザは、品質に関する専門的な知識や分類用語を知らなくても、ツールの質問に順次答えることで、要求品質を明らかにできる。管理者はこの結果を直接 SQUARE システムに入力することで、その品質の実現に有効な開発技術と評価尺度の選択を行え、従来より質の高い開発計画を簡単に作成することができる。

以降、2章でプロジェクト管理の概略と主な用語について説明する。3章では SQUARE システムの従来の主な機能と構成について述べ、問題点と解決策を明らかにする。そして、4章で開発したツールの構成と機能を説明する。さらに5章において、ツールの使用例を示す。

2. 定義

2.1 プロジェクト管理

プロジェクト管理(Project Management)とは、工学におけるプロジェクトを成功させるための管理に必要な計画作成(planning)、組織化(organizing)、人員配置(staffing)、指揮(directing)、制御(controlling)を提供する手続き、実行、技術、ノウハウの体系である⁽¹⁾⁽⁴⁾(図1参照)。

特に、対象がソフトウェアの場合、ソフトウェア工学プロジェクト管理(Software Engineering Project Management)という(以下、特にことわらない限り、単にプロジェクト管理という)。

2.2 開発計画

プロジェクト管理における開発計画作成とは、開発組織の目的を達成するための一連の行為を予め決定することである⁽¹⁾。

開発計画には、次の5つの項目が含まれていなければならない⁽⁶⁾。

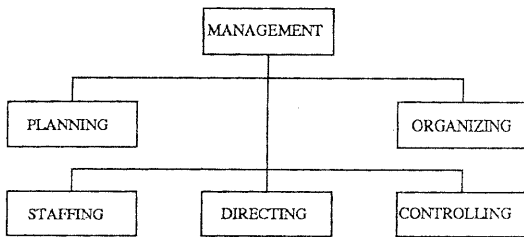
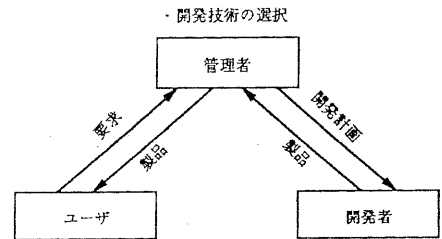


図1 プロジェクト管理のモデル⁽³⁾



・要求品質の明確化
・品質の定量的評価

・製品の作成
・品質の定量的予測

図2 ユーザ，開発者，管理者

- (1) プロジェクトの各作業と要求されている結果の定義
- (2) 作業量の見積り
- (3) 作業のスケジュール
- (4) 開発者やその他の資源の割当の詳細
- (5) 各資源に対する「負荷」や資金の見積り

3. 開発計画作成支援システム SQUARE

3.1 ソフトウェア品質定量化の枠組

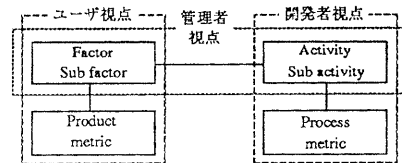
ソフトウェア品質定量化の枠組は、要求品質を示す立場のユーザと要求品質を実現する立場の開発者、及び両者の調整を行なう管理者の3つの視点で品質を捉え、効率良く品質を実現するためのものである⁽¹²⁾(図2参照)。

ユーザ視点では、ソフトウェアの特性(Factor)として品質を捉え、その評価は完成したソフトウェアに対する尺度(Product metric)を用いて行う。開発者視点では、自らの開発技術(Activity)の結果として品質を捉え、その評価はソフトウェアの開発過程に対する尺度(Process metric)を用いて行う。管理者は、ユーザと開発者の間の対話を助け、Factorをその実現に有効なActivityに変換する立場にある(図3参照)。

この枠組では、その適用に必要な情報を3つの視点に対応する3つの品質表 T1, T2, T3 で管理している(図4参照)。

ユーザ視点の品質表 T1 には、Factor とその実現の度合を評価する Product metric が示されている。管理者視点の品質表 T2 には、Activity j ($1 \leq j \leq |T3|$) が Factor i ($1 \leq i \leq |T1|$) の実現にどの程度有効であるかを表す値 y_{ij} ($0 \leq y_{ij} \leq 10$) が示されている。開発者視点の品質表 T3 には、Activity とその実施の度合を評価する Process metric が示されている。

品質表 T1 及び T3 の項目は、一般に知られた分類に基づいている。T1 における Factor は



Factor...ユーザから見たソフトウェアの品質
Activity...ソフトウェア開発技術または活動
Product metric...Factorの評価尺度
Process metric...Activityの評価尺度

図3 ソフトウェア品質定量化の枠組

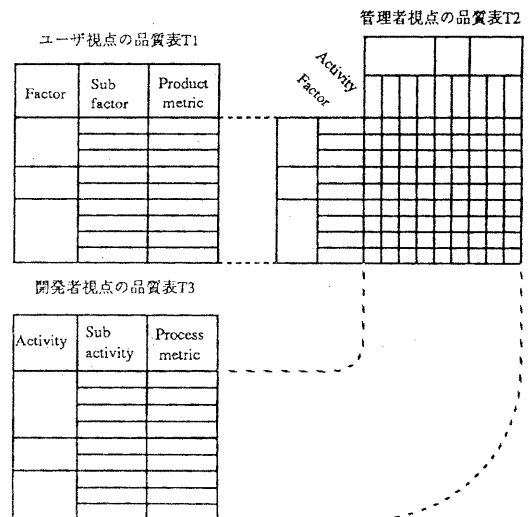


図4 SQUAREの品質表

Factor	Sub factor	Activity																					
		設計技術					コーディング技術			テスト技術													
Sub activity		集合設計	入出力データ設計	データ構造化	機能定義	故障回避機能機構	故障診断機能機構	コマンド形式標準化	自習機能機構	実行速度改善	利用容易機構	設計レビュー	コード標準化	コード自動生成	プログラム再利用	コードレビュー	テストデータ自動生成	テストデータ再利用	ホワイトボックステスト	ブラックボックステスト	実行速度テスト	エラーレポート作成	
機能性	合目的性	1										3			2								
	正確性	1	2									2			2				2	1			
	セキュリティ			4	4							1			1								
	互換性	1								2		2			2	1		2					
信頼性	接続性		2								2			1	1		2		2				
	無欠陥性										2			2	1		2	2				1	
	誤り許容性					7					1			1			1						
使用性	可用性					7					1			1			1						
	理解性					7					1			1			1						
	習得容易性					7					1			1			1						
	操作性					7					1			1			1						
効率性	対話性					7					1			1			1						
	実行効率性								4		2			2					1	1			
保守性	資源効率性									5	2			2						1			
	修正容易性	1											2	2	1	1						3	
	拡張性	1	2										2	2								3	
	テスト容易性	2									3	1	2										

図5 品質表T2の例(一部)

ISO 標準による分類 (23 項目)⁽⁹⁾ で表される。また、T3 における Activity は、品質機能展開における品質管理工程表の一般的な例⁽⁷⁾⁽⁸⁾ (106 項目) に基づいている。品質表 T2 の例 (一部) を図 5 に示す。

3.2 開発計画の作成手順

従来の SQUARE システムが出力する開発計画は、2 章において定義した開発計画の一部であり、各開発作業、作業の達成度を評価する尺度、及び、その目標値 (見積値) から構成される。

SQUARE システムによる開発計画の作成手順は次の通りである。

Step 1: Factor による要求品質の明確化

ユーザが、T1 の各 Factor i の実現目標値 c_i ($0 \leq c_i \leq 5$) を設定すると共に、各 Factor の実現の度合を評価する Product metric を T1 より選択する。

Step 2: Factor から Activity への変換

管理者が、Step 1 で設定された各 Factor の実現目標値 c_i と編の y_{ij} に基づいて各 Ac-

tivity j の実施目標値 a_j ($0 \leq a_j \leq 5$) を算出する。

Step 3: Process metric の選択

開発者が、各 Activity の実施の度合を評価する Process metric を T3 より選択する。

以上の各 Step において設定された Factor の実現目標値 $\{c_i\}$ と Activity の実施目標値 $\{a_j\}$ 、および、選択された Product metric と Process metric の集合が開発計画である。

開発計画の作成において最も複雑な処理を必要とするのは Step 2 の Factor から Activity への変換である。SQUARE システムは、 $\{c_i\}$ が与えられると次の条件を満たす $\{a_j\}$ を自動的に算出する機能を持つ。

$$\sum_{j=1}^{|T3|} cost(a_j) \rightarrow \min$$

$$c_i \leq \sum_{j=1}^{|T3|} \frac{y_{ij}}{10} \cdot a_j \quad (1 \leq i \leq |T1|)$$

但し、 $cost(a_j)$ は T3 中の Activity j を a_j だけ実施するために必要なコストを表す。

3.3 問題点

SQUARE システムを実際プロジェクトに適用した結果、次の (1)~(3) の問題点が明らかになった。

- (1) 品質特性 (Sub factor) を表す ISO の用語の抽象度が高い。
- (2) 各 Sub factor が互いに独立でないことを入力時に考慮しにくい。
- (3) 対象となるユーザによって品質の捉え方が違うことに対応していない。

(1) は、入力すべき Sub factor の意味がユーザに十分理解されず、ユーザの意図した品質を実現するような開発計画がたてられないという問題である。一般にユーザは、品質特性の意味や開発の詳細を知らず、品質を非常に具体的な表現で要求することが多い。一方、Sub factor の表現はこれに比べてかなり抽象的である。この対応を正確に把握することがユーザ側に大きな負担となっていた。

(2) は、ある品質を実現する際に他の品質が影響を受ける場合の問題である。例えば、資源効率性を上げることと実行効率性を上げることが両立させるのは困難である。すなわち、メモリの消費量を減らして、なおかつ実行速度を上

げること難しい。この場合、資源効率性を要求すればするほど、実行効率性がある程度、犠牲にせざるを得ない関係にあると言える。

(3) は、開発するソフトウェアの用途、ユーザの分野や階層によって、品質に対する認識に差があることを意味する。例えば、エンドユーザは、システム部門と比較して、保守に対する要求に厳密でない可能性がある。また、銀行系のソフトウェアの信頼性に対する要求は、他の場合に比べて非常に大きくなる。

3.4 解決策

前節の問題点の解決策として、次の (a)~(c) が考えられる。

- (a) Sub factor よりも具体的な言葉で表現された品質に対する要求を、ISO 標準の Sub factor に変換する。
- (b) 具体的な一つの品質表現を、複数の Sub factor に対応させる。
- (c) 品質の表現をユーザの分野別、階層別に用意する。

これらの解決策を機能として持つツール(要求品質決定ツール)を開発し、SQUAREのサブシステムとして入力部に接続する(図6参照)。

ユーザは品質に対する知識に乏しいことを前提にしているので、

- 最終的に全ての Sub factor を網羅した要求となること。
- 一つの Sub factor に対応する具体的な表現が、様々な角度からなされていること。
- 要求の強さの把握でも、できるだけ具体的な表現を用いて行なわれること。

などの点を考慮する必要がある。よって、ツールの開発にあたっては、以下の3点を加味した。

- (1) 品質に関する質問を予め用意して、入力時にユーザがそれに答える形式とする。
- (2) 質問は Sub factor の評価尺度の定義に基づいて可能な限り用意する。
- (3) 質問に対する答は、予め用意された選択肢から選択する形式とする。

ツールによってなされる質問と SQUARE の Sub factor, Factor の関係を図7に示す。質問と Sub factor は、多対多の対応関係にある。質

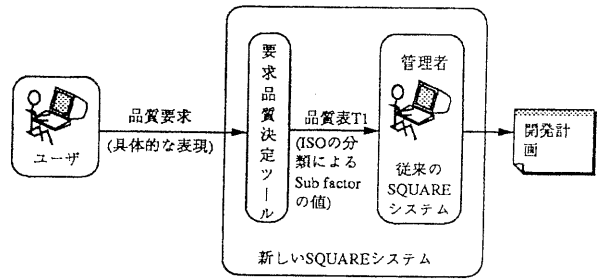


図6 要求品質決定ツールの位置付け

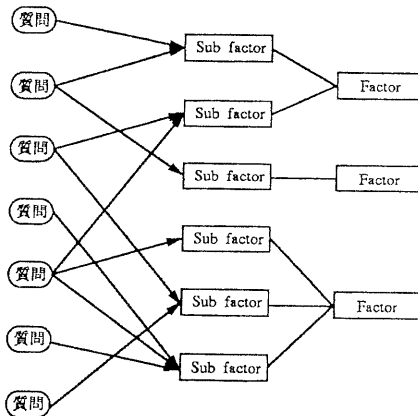


図7 質問とFactor, Sub factorの対応

質問テーブル

key1	質問	選択肢				
		1	2	3	4	5
a1		大	中	小		
a2		高	中	低		
a3		1	2	3	4	5

加点テーブル

keys		加点				
key1	key2	1	2	3	4	5
a1	01	2	1	0		
a1	02	4	2	0		
a2	02	5	3	1		
a2	03	2	1	0		
a2	04	2	1	0		
a3	05	4	3	2	1	0

Sub factorテーブル

key2	Sub factor	得点
01		0
02		0
03		0
04		0
05		0

図8 各テーブル間の関係とデータの例

...
35)処理の要求を行ってから結果が得られるまでの時間は？ [1]10秒[2]20秒[3]30秒[4]40秒[5]5秒
36)要求の終りから応答が始まるまでの時間は？ [1]5秒[2]10秒[3]15秒[4]20秒[5]25秒
37)プログラムの実行が開始されてから終了するまでの時間は？ [1]2秒[2]7秒[3]12秒[4]17秒[5]22秒
38)プログラムの実行が開始されてから終了するまでに要しても良いCPU時間は？ [1]1秒[2]2秒[3]3秒[4]4秒[5]5秒
39)外部記憶との入出力にかけても良い時間は？ [1]2秒[2]4秒[3]6秒[4]8秒[5]10秒
40)プログラムの実行がOSや実行資源からの割り込みによって待たされる時間は？ [1]0.5秒[2]1.0秒[3]1.5秒[4]2.0秒[5]2.5秒
41)センターと端末間で伝送の開始から終了までにかけても良い時間は？ [1]0.5秒[2]1.0秒[3]1.5秒[4]2.0秒[5]2.5秒
42)端末での処理の開始から終了までに要しても良い時間は？ [1]1秒[2]2秒[3]3秒[4]4秒[5]5秒
43)1日に最高何件の処理が必要ですか？ [1]100件[2]200件[3]300件[4]400件[5]500件
44)1時間に最高何件の処理が必要ですか？ [1]10件[2]20件[3]30件[4]40件[5]50件
...

図9 質問項目と選択枝の例(実行効率性)

合目的性	= 1.5
正確性	= 3.8
接続性	= 4.0
標準適合性	= 4.5
セキュリティ	= 0.0
成熟性	= 2.7
障害許容性	= 3.7
回復性	= 2.4
理解性	= 2.5
習得性	= 2.7
運用性	= 5.0
実行効率性	= 4.3
資源効率性	= 4.2
解析性	= 0.0
変更性	= 3.0
変更適用性	= 4.0
試験性	= 3.5
理解性	= 0.0
変更作業性	= 3.0
環境適応性	= 0.0
移植作業性	= 0.0
規格適合性	= 0.0
置換性	= 0.0

図10 出力例(品質表T1)

問は具体的な表現でなされるので、図において右にいくほど抽象度が高い。

4. 要求品質決定ツール

4.1 機能の概要

開発した要求品質決定ツールは、次のような機能を実現している。

- (a) ユーザが理解しやすいように、要求品質に関する具体的な質問 (70 個) を行い、要求の強さを具体的な表現の選択枝から選ばせる。
- (b) 一つの質問は、関係する一つ以上の Sub factor に対応しており、回答でえられる要求の度合に応じて、Sub factor の重みを加減する。
- (c) 様々な分野、階層のユーザに対応させるため、質問を複数種類用意している (現時点ではエンドユーザ、システム部門の 2 種類)。

本ツールの入力は、要求品質に関する具体的な質問 (データベースにより与えられる) に対するユーザからの回答 (選択肢) である。ツールは入力に基づいて、

- (1) ISO 品質特性 (Sub factor) の選定
- (2) 選定された Sub factor への加点 (重み付け)

を行う。ツールの出力は (1), (2) の処理の結果得られる重み付けされた Sub factor の集合である。この出力が直接 SQUARE の入力 (品質表 T1) となる。

4.2 構成

ツールは、データベース部とプログラム本体から構成される。

データベース部は、次の 3 つのテーブルから構成される。(図 8 参照)

- (1) 質問テーブル…要求品質に関する質問とその選択肢を保有する。
- (2) 加点テーブル…質問および Sub factor 番号 (Sub factor を識別するための番号)、選択肢毎に加点する数値を保有する。
- (3) Sub factor テーブル…Sub factor 番号と実際の Sub factor 名との対応テーブル。

プログラム本体は、ユーザインターフェース、データベースに対するアクセス部、及び、出力表示部から構成される。

4.3 処理手順

実際の処理手順を以下に示す (図 8 参照)。

Step 1: ユーザによるツールの起動

ユーザが自分に適したコマンドを 2 つのコマンド (rq1 と rq2) から選択し、実行する。



図11 ツール稼働中の画面イメージ

なお、コマンド `rq1` は、エンドユーザ向けの質問を、コマンド `rq2` は、システム部門ユーザ向けの質問を、それぞれ起動する。

Step 2: 質問の表示

ツールは質問テーブルの先頭から質問とその選択枝を検索し、画面に表示する。

Step 3: ユーザによる回答

ユーザは質問に対して回答を選択し入力する。

Step 4: 回答の処理

ツールは画面に表示された各質問に対し、以下の (a)~(d) の処理を行う。

- (a) 質問番号 (key1) に該当するデータを加算テーブルから検索する。
- (b) ユーザが選択した選択枝に該当する位置にある加算を「得点」とする。
- (c) Sub factor 番号 (key2) に該当するデータを、Sub factor テーブルから検索する。
- (d) 選択された Sub factor の得点欄に、「得点」を加える。

Step 5: 繰り返し (終了)

ツールは質問テーブル中に未表示データが

残っている場合、(2) 以降の処理を繰り返す。質問テーブル中に未表示データがない場合、Sub factor テーブルから Sub factor 名とその得点を出力、表示して、処理を終了する。

5. 例

要求品質決定ツールがユーザに行う質問の例として、Sub factor 実行効率性の重みを決定するための質問を図9に示す。また、最終的に得られる出力例を図10に示す。ツール稼働中の画面イメージの例は、図11である。

6. あとがき

本研究では、開発作成支援システム SQUARE のサブシステムとして、要求品質決定ツールを開発し、その使用例を示した。これにより、ソフトウェアの品質を熟知しないどのような分野のユーザによる要求も、容易にプロダクトの品質特性に変換が可能となり、管理者は SQUARE の入力を従来より正確かつ容易に得ることができるようになる。また、要求品質決定ツールの出力は SQUARE システムを通じて定量的な開発技術に変換され、開発計画として開発者に示される。

今後は、ツールの評価実験を行うとともに、質

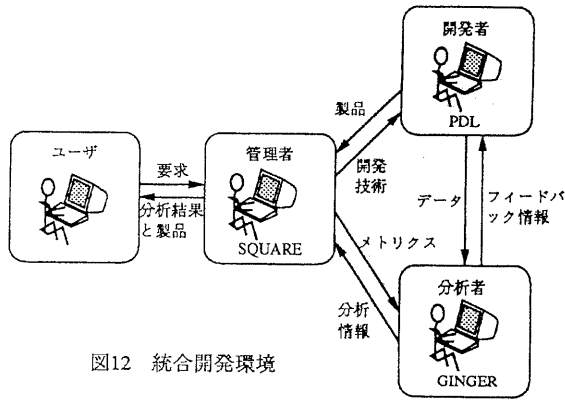


図12 統合開発環境

問項目の見直し、質問と品質特性の対応関係をより明確なものにしていく予定である。特に、ソフトウェアの用途、ユーザの階層に応じた質問項目の選定は、現段階の2種類からより多くの数に増やさなければならない。また、現在のSQUAREシステムにより作成される開発計画は、一般に言われる開発計画の核となる部分を構成しているにすぎず、今後、開発作業のスケジュールや資源の割当などの項目も出力できるように、さらに研究を進めて行く予定である。

要求品質決定ツールによって、SQUAREシステムは、ユーザの要求から開発計画の作成までを、一貫して管理できるものとなった。今後は、実際の開発過程における進捗管理とのインターフェースを見直し、開発過程の制御システムGINGER⁽¹⁰⁾や開発過程記述言語PDL⁽¹¹⁾による開発環境と接続し、統合されたプロジェクト管理システムの構築を目指す方針である(図12参照)。

謝 辞

ツールの開発にあたって御協力いただいた、(株)三菱電機東部コンピュータシステムの細谷氏に感謝致します。

文 献

- (1) H. Koontz, C. O'Donnell, and H. Weihrich: "Management," 8th ed., McGraw-Hill Book Co., NY(1984).
- (2) B. W. Boehm, and R. Ross: "Theory-W Software Project Management: Principles and Examples," IEEE Transaction

on Software Engineering, Vol. 15, No.7, pp.902-916(July 1989).

- (3) R. H. Thayer: "Software Engineering Project Management A Top-Down View," IEEE(1987).
- (4) R. A. MacKenzie: "The management Process in 3-D," Harvard Business Review, Vol.47, No.6, pp.80-87(November-December 1969).
- (5) 山田茂: "ソフトウェアマネジメントモデル入門," bit, Vol22, No.12, pp.1305-1311(1990).
- (6) W. B. Miller: "Fundamentals of Project Management," Journal of Systems Management, Vol.29, pp.22-29(November 1978).
- (7) 情報処理振興事業協会 技術センター: "品質機能展開による高品質ソフトウェアの開発手法 解説編," pp.84-92, コンピュータ・エージ社(1989).
- (8) 情報処理振興事業協会 技術センター: "品質機能展開による高品質ソフトウェアの開発手法 活用事例編," pp.15-28, コンピュータ・エージ社(1989).
- (9) "平成元年度 ソフトウェア開発・システムの文書化標準化調査研究報告書 STD-WG5(ソフトウェア品質特性) 分科会報告," 日本規格協会・情報技術標準化センター(1990).
- (10) S. Kusumoto, K. Matsumoto, T. Kikuno, and K. Torii: "GINGER: Data collection and analysis system", IEICE Technical Report, SS90-5, pp.39-48(1990).
- (11) K. Inoue, T. Ogihara, T. Kikuno, and K. Torii: "A Formal Adaptation Method for Process Description," Proc. of 11th International Conference on Software Engineering, pp.145-153(May 1989).
- (12) K. Matsumoto, Y. Takada, T. Kikuno, and K. Torii: "SQUARE: Software Quality Assurance Supporting System," IEICE Technical Report, SS90-4, pp.29-38(1990).