要求工学における品質特性の測定について

情報処理学会ソフトウェア工学研究会要求工学ワーキンググループ †
(発表者: 海谷 治彦 †‡)

†http://www.selab.cs.ritsumei.ac.jp/
~ohnishi/RE/rewg.html

‡信州大学 工学部 情報工学科
kaiya@cs.shinshu-u.ac.jp

あらまし　情報処理学会 ソフトウェア工学研究会要求工学ワーキンググループにおける研究成果の 1 つとして，要求工学における品質特性の測定に関する報告を行う．本稿では特に，要求仕様書を測定対象として，要求仕様書自体の特性と，開発されるソフトウェア製品の特性を定量的に評価・予測するための方法についての考察をまとめた．仕様書自体の特性は IEEE の標準規格の記述に基づき，製品の特性は非機能要求に基づいた．

キーワード　要求工学，標準規格，尺度，非機能要求

Preliminary Study for Measuring Quality Characteristics in Requirements Engineering

Requirements Engineering Working Group, SIGSE of IPSJ †
and Haruhiko Kaiya †‡

†http://www.selab.cs.ritsumei.ac.jp/~ohnishi/RE/rewg.html

‡*Faculty of Engineering, Shinshu University*
http://www.cs.shinshu-u.ac.jp/~kaiya/

Abstract　*We report the results of the discussion for measuring the quality characteristics of requirements engineering. We especially focus on both the characteristics of a software requirements specification itself, and the characteristics of a software product developed based on the specification. We explore what to measure a software requirements specification for specifying these characteristics.*

Key words　*Requirements Engineering, Standard, Measure, Non-functional Requirements*

1 はじめに

要求工学は，高品質のソフトウェア要求を開発するためにある [1]．そこで，要求工学実践者は，要求獲得法，要求記述法，要求検証法，要求管理法などの技法を用いて，ソフトウェア要求を高品質なものにしてゆかなければならない．要求工学に限らず，ソフトウェア工学における多くの技法は，その技法の効果を定性的に示すのみである場合が多いが，定量的に効果を示すことも有効である局面がある．

要求工学ワーキンググループでは，図 1 に示すようなスパイラルな形で要求工学プロセスは進められると考えている [2]．ソフトウェア要求の品質を一度に高度なものとする方法がない以上，我々は，1 つ，もしくはいくつかの技法を繰り返し適用することにより，その品質を高めてゆくしかないからである．しかし，品質が高まっていることを認識するためには，定性的な評価だけでなく，ある程度，定量的なデータに基づく評価が必要となる．

DeMarco 著書 [3] における「測定できない事柄を，コントロールするわけにはいかない」というくだりは大変有名であり，説得力のあるものである．そこで，要求工学が工学足りうるのであるためには，要求工学の諸元を反映する尺度を選び，その尺度を得るためのデータを測定する必要がある．

本稿では，要求工学ワーキンググループのワークショップにおいて，ソフトウェア要求の尺度として何が適切で，その尺度を得るためにはどのような諸元を測定すべきかについて検討した結果を報告する．特に，要求仕様書の記述内容をもとに，要求仕様書自体と作成されるソフトウェア製品を定量的に評価する可能性を模索する．尚，本稿では特に断りがない限り，要求仕様書を単に仕様書と呼ぶ場合がある．

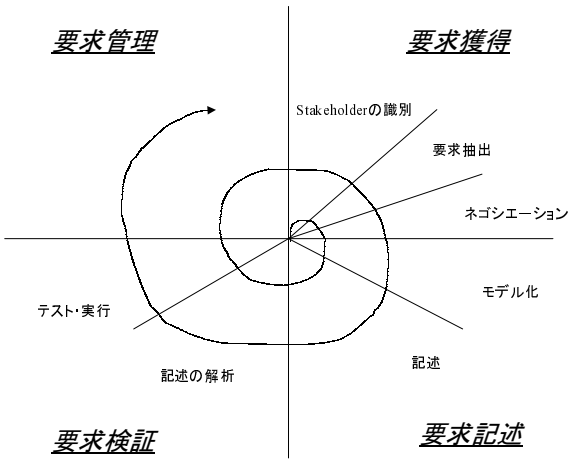

図 1: 要求工学プロセス (文献 [2] より)

2 節では，要求工学プロセスの何が測定できるかについて概観する．次に 3 節では，IEEE の要求仕様書に対する企画・標準の概要を紹介する．この標準に記述されている要求仕様書の持つべき 8 つの特性は，要求工学の尺度として有望である．4 節では，特に最終製品であるソフトウェア製品やプログラム自体の品質を予測するのに有望な性質である非機能要求について概観する．立場の違いによって，ソフトウェア要求に対する関心は異なる．5 節では，Stakeholder 毎に，ソフトウェア要求のどのような特性に関心を抱いているかを概観する．6 節では，3 節や 4 節で議論した特性を，要求仕様書をもとに具体的にどのように定量化するかについて述べる．

2 要求工学プロセスの何が測定できるか？

ソフトウェアプロセスにおける基礎的な尺度は，プロダクト尺度，プロセス尺度，リソース尺度の 3 つのカテゴリに分かれている [4]．それぞれの代表例としては，プロダクト尺度では LOC(行数)，プロセス尺度ではサイクル時間 (1 つのプロジェクトが完成するまでに必要な時間)，そしてリソース尺度では作業者の労働時間などがある．

全ての尺度が要求工学の評価のために有用なわけではないので，まず先に，ソフトウェア要求の何を知りたいかを明確にする必要がある．しかし，ここでは，ソフトウェア要求を反映するであろう測定対象の概要をつかむため，要求工学プロセスにおける製品，プロセス，リソースとは何かを簡単に考察する．要求工学では，それぞれのカテゴリにおける測定対象になるものは，以下であると考えられる．

- 製品: 要求仕様書
- プロセス: 要求工学プロセス (図 1)
- リソース: Stakeholder

リソースについては，他のソフトウェアプロセスのように消耗される労力に注目するよりも，Stakeholder に蓄積される知識や理解度に注目したほうが，より適切な尺度が得られるように思われる．

要求工学プロセスでは，開発者はもちろん，顧客さえも明確に認識していなかった要求そのものが共通理解を伴い段階的に確定してゆくと思われる．この要求そのものの段階的確定は，要求自体の品質向上を反映しているとみなすことができ，要求仕様書の品質とは明らかに異なる．このような要求自体の品質を反映している尺度は，開発者や顧客を含む Stakeholder 自体から測定可能かと思われる．

3 ソフトウェア要求仕様に対する規格・標準

IEEE の要求仕様書に対する企画・標準 [5] は，高品質のソフトウェア要求を議論する適切な開始点である．本節では，この標準書「ソフトウェア要求仕様書のための推奨実践」の概要を説明する．本規格では，何を記述すべきかを推薦しているのは勿論，何を書くべきでないかも随所に言及している．例えば，設計や実装の詳細に立ち入ることや，品質保証計画については要求仕様書ではなく，他の文書に記述べきであるとしている．

本文書は，本文 5 節，付録 2 節から成っており，それぞれの概要は以下のとおりである．

1. 概要
2. 参考文献
3. 定義: 用語の定義等．
4. 良い要求仕様書を生産するための考察: 後述．
5. 要求仕様書に含まれる部品の解説: 仕様書の記述項目が具体的な例とともに説明されている．

付録には，要求仕様書のテンプレートと，IEEE/EIA 12207.1-1997 と呼ばれる別の標準規格との関連についての説明がある．本標準規格はそのタイトルおよび目次から，プロダクト指向で記述されているのがわかる．ちなみに，IEEE/EIA 12207.1-1997 はシステム開発全体をプロセス指向で説明した標準規格のようである．

本稿での議論で最も重要な部分は 4 節である．そこでは要求仕様書を記述する際，考慮しなければならない背景を以下の 8 項目に分けて説明している．

1. 要求仕様書の本質
2. 要求仕様書の環境
3. 良い要求仕様書がもつべき特性
4. 要求仕様書の合同準備
5. 要求仕様書の進化
6. プロトタイピング
7. 要求仕様書内の設計事項
8. 要求仕様書内のプロジェクト要求

最初の部分「4.1 要求仕様書の本質」には，そもそも要求仕様書とは何かが説明されている．要求仕様書とは，

> 特定の環境下で，特定の機能を発揮する特定のソフトウェア製品やプログラム (群) の仕様書

であるとしている．よって，要求仕様書は以下の点を扱わなければならないとしている．

- 機能性 (Functionality): どのような仕事を支援してくれるのかの説明．
- 外部インタフェース (External Interface): 人間，ハードウェア，他のソフトウェアとのインタフェース．
- 効率 (Performance): スピード，可用性，レスポンス時間，障害復帰時間等．
- 属性 (Attribute): ポータビリティ，プログラムの妥当性，保守性，セキュリティ．
- 実装を制約する設計制約: 開発言語や OS の指定など．

4 節 3 番目の部分「4.3 良い要求仕様書がもつべき特性」の中に，要求仕様書の持つべき特性として以下の 8 つが直接的に言及されている．

- 妥当である (Correct): 仕様書にある全ての要求が開発されるソフトウェアが満たすべき事柄と一致している場合，仕様書が妥当であるとする．これは，顧客や利用者の真のニーズに合致しているか否かを示す性質であり，これを自動的，系統的に確かめる術はない．追跡可能性は，この性質を確認する助けになる．

  また，妥当であるか否かは，要求仕様書より上位の仕様書 (例えば，システム要求仕様書 [6] など) や，プロジェクトに関する文書，適用すべき標準規格などとの整合性も含んでいる．
- 非曖昧である (Unambiguous): 仕様書内の全ての要求が唯一の解釈を持つ場合，その仕様書は非曖昧であるとする．例えば「大きい，小さい」などの表現ではなく，具体的な数値による指定などにより，この性質は充足される．
- 完全である (Complete): 標準規格の 4.1 節にある基本的記述内容と，5 節にある記述項目が仕様書に含まれている場合，仕様書は完全であるとする．
- 無矛盾である (Consistent): 要求仕様書内部に矛盾がない場合，要求仕様書は無矛盾であるとする．しかし，外部，例えば上位のシステム仕様書との矛盾がある場合には，その仕様書は「妥当でない」とする．
- 重要度と安定性がランク付けられている (Ranked for importance and/or stability): 記述内容が変更され易いものか否か，および，記述内容は絶対充足しなければならないのか，条件付でいいのか，それともオプションなのかを明らかにしている場合，その仕様書は重要度と安定性がランク付けられているとする．
- 検証可能である (Verifiable): 開発されるソフトウェアが要求仕様書の記述内容を満たすか否かをチェックするための方法があり，チェック作業は妥当なコスト内，妥当な時間内に終わる場合，そ

の要求仕様書は検証可能であるとする．例えば，「良い」,「快適」などの曖昧な記述を含む場合，検証可能とは言えず，数値などで指定された記述は検証しやすい．

- 変更可能である (Modifiable): 仕様書が容易かつ完全かつ矛盾を発生させずに変更可能である場合，仕様書が変更可能であるとする．
- 追跡可能である (Traceable): 将来の開発で改良された要求仕様書との対応や，他の開発段階における文書との対応がとれている場合，要求仕様書は追跡可能であるとする．

これらは，要求仕様書がもっていなければならない性質であるため，持っているか否かの判定しか意味がないようにも思われる．しかし，要求工学プロセスの途中で，これら性質が段階的に充足されていくことが鑑みれば，それぞれに，どの程度の達成度であるかを判定することは有効であると考えられる．

4 非機能要求

IEEE の標準 [5] は要求仕様書という文書に関する勧告である．よって，仕様書で扱うべき内容 (標準内の 4.1 節) と，その妥当性 (4.3 節) については言及しても，どのような内容が高品質なソフトウェア要求であるかについては深く言及していない．要求を測定するという観点からは，機能に関する要求内容は，その機能の有無ということで自明な場合が多い．しかし，非機能要求と呼ばれる機能以外の要求は，たとえ非曖昧かつ検証可能性であっても，妥当性を保障していない危険をはらんでいる．また，非機能要求は，最終成果物 (ソフトウェア製品やプログラム) の品質につながる．よって，仕様化された非機能要求の妥当性を確認するための手段が必要となる．本節では，非機能要求とは何かについて概観する．

非機能的要求は文献 [7] で詳しく検討されている．同文献では，プロダクト指向ではなく，プロセス指向の考えて方で，以下のような非機能要求について定性的な分析を行っている．

- Performance
  - Time
    - Response Time
    - Throughput
    - Process Management
  - Space
    - Main Memory
    - Secondary Storage
- Cost
- User-Friendliness
- Security
  - Confidentiality: 情報の認定されていない公開を防止
  - Integrity: 情報の認定されていない修正を防止
  - Avaliablility: 情報や資源の認定されていない抑制を防止

尚，上記の箇条書きの階層は is-a 関係 (サブクラス) を示している．

5 Stakeholder 毎のソフトウェア要求への関心

立場の違いによって高品質なソフトウェア要求とは何かが異なる．Stakeholder とは，システムの変更によって自己の利害が影響を受け，その変更による損失や利益に固執する者を指す．そこで，ソフトウェア要求に関与する代表的な 4 種類の Stakeholder が何をゴールとしているかを考慮して，品質項目の洗い出しを行う．尚，標準規格 [5] では，Customer, Supplier, User の三分類に注目しているが，本稿ではより詳細な議論をするため，文献 [8] の分類を利用する．

5.1 設計・開発者

プロジェクト管理者，ソフトウェア設計者，通信の専門家，テクニカルライター等に相当する．彼らにとってのソフトウェア要求との接点は，要求仕様書であり，そこに何を作るかが明確になっていることが重要な関心事である．

よって，標準規格 [5] の「4.3 良い要求仕様書がもつべき特性」の，非曖昧性，完全性，無矛盾性，検証可能性などがこの Stakeholder の関心事を反映すると思われる．スケジュールに関する妥当性も彼らにとっては大きな関心事ではあるが，これは要求仕様書で扱う範囲ではない．

5.2 経理・販売・購入者

ビジネスアナリスト，市場管理者，バイヤーなどに相当する．彼らにとって重要なのは，妥当な時期に妥当な価格で製品を販売することである．

よって，非機能要求の中でも，コストに抵触する部分，そして，製品のサイクルタイムの速さ，すなわち要求プロセスの回転の速さなどが関心事だと思われる．プロセス管理は直接は要求工学の範疇とはいえないが，変更可能性，追跡可能性はプロセスサイクルに大きな影響を与える．

5.3 導入・保守者

トレーニグスタッフや保守スタッフなどに相当する．彼らにとって重要なことは，ソフトウェアを導入しやすいこと，および，保守・変更しやすいことである．

その観点からは標準規格 [5] の「4.3 良い要求仕様

書がもつべき特性」の，変更可能性，追跡可能性などが重要である．

5.4 利用者

ユーザー管理者，およびユーザーそのものに相当する．ユーザーにとって最も関心があるのは，製品が妥当である (Correct)，すなわち期待通りに作成されていることである．

ソフトウェア開発が要求仕様書をもとに行われる限り，少なくとも仕様書が妥当でなければ，利用者の希望は果たされない．この妥当性は，必要な機能が提供されていることは勿論，製品の非機能的な特性が，利用者の思い描いているものと一致していることも含まれる．

6 ソフトウェア要求の具体的な測定法

5 節で議論したように，3 節で紹介した仕様書の特性，および 4 節で紹介した非機能要求に関する特性は，Stakeholder がソフトウェア要求について知っておきたい情報に関連している．しかし，これらの特性を具体的な要求工学プロセス内のどのような現象をもとに定量化すればよいのかは明らかではない．

本節では要求仕様書を測定対象とした場合，前節にあげた特性群のうち何が測定可能か，可能ならばどのようにして測定できるかを考察する．3 節でのべた 8 つの特性は，要求仕様書自身の品質特性であるのに対し，4 節で述べた非機能要求の特性は要求仕様書どおりにシステムが実現された場合のシステムの品質特性とみなすことができる．本節では，(1) 要求仕様書自身の品質と，(2) 要求仕様書から測定したシステムの品質 (ただし，システムはまだ実現されていない) とに分けて議論する．

6.1 要求仕様書自身の品質

要求仕様書がどのような形式で書かれているかによって，具体的な測定項目も異なってくる．しかし，IEEE の要求仕様書に対する企画・標準 [5] には，どのような内容を含むべきかについては明確な定義がされている．よって，ある程度は形式に依存しない議論が可能である．

ここでは，まず要求仕様書単独で定量的な測定が可能と思われる特性について述べる．

- 非曖昧性，無矛盾性: 仕様書中に含まれている曖昧な文，矛盾しあう文のペアの割合を測定する．仕様書が形式言語で記述されている場合，曖昧な文はありえないが，矛盾しあう文は自動的に検出できる可能性がある．
- 完全性: 記述項目に漏れを測定する．IEEE 標準 [5] などのように仕様書のフォーマットが決まっている場合，各項目がすべて記載されているかどうかを調べることによって漏れの度合いを推定することができる．また，未定義語の出現頻度なども測定手段のひとつになりうる．
- 変更可能性 仕様書中に冗長な文が含まれていたり，互いに意味的に依存しあう文が含まれていたりする場合，要求変更があったときにこれら複数の文の変更を考慮する必要がある．冗長性のある文や依存する文の割合を測ることにより，変更がしやすいかどうかの指標に用いることができる．
- 重要度と安定性のランクづけ: 仕様書中で，必須要求とオプショナル要求の区別があるときはそれらの優先順位が明記されているかどうかを調べる．単純に仕様書が英文で書かれている場合，助動詞 should と may に注目することにより，それが必須要求であるかオプショナルかを区別することができる．
- 検証可能性: 検証可能性は，仕様書がどのような言語で書かれているかに大きく依存する．まず記述言語が検証系を持った形式言語などの検証可能な言語か，自然言語かで分かれる．自然言語で書かれていた場合，曖昧語 (「ときどき」など) が含まれている割合を調べるのも一つの手法である．

次に仕様書単独では測定が困難な特性について述べる．

- 妥当性: 妥当性は仕様書外の要素との関連で決まるため，その測定は，仕様書単独では測定不能である．

  まず，妥当性は，顧客やユーザといった要求を出す人の真のニーズが書かれているかどうかであるため，なんらかの形で要求を出す人の真のニーズを表現するか，要求を出す人に測ってもらうしかない．顧客やユーザがチェックできる作業が含まれているような要求獲得の方法論やプロセスを採用するのが一つの方策である．

  例えば，要求仕様を実行可能な言語で書き，プロトタイプとして顧客・ユーザに妥当性を評価してもらったり，シナリオ分析法で書かれたシナリオを読んで評価してもらったりするなどの方式が考えられる．この場合，プロトタイプやシナリオの評価方法を確立する必要がある．また，後で述べるように，要求獲得の方法論やプロセスを決め，その最初の作業への入力物が顧客・ユーザの真のニーズに近いと仮定し，最後の作業で作成されたプロダクトの差異を測定する手法も考えられる．

  また，妥当性は関連する他の文書との整合性も含んでいるため，それらの文書で指定される開発

コスト，開発期間，稼働環境との整合性ととらねばならない．これは後述の見積り技術によって得た見積り値と，他の文書で指定された計画値の比較によって定量化できる．

- 追跡可能性: 追跡可能性も，要求仕様書単独では測定不可能で，設計仕様書など他の文書との関連をもとに測定する必要がある．例えば，設計手法に合った要求記述になっているか（オブジェクト指向設計法が採用されているのであれば，要求仕様書の章立て，章名がオブジェクトごとになっているなど）に通目する．一般的には，要求仕様書のまとまった部分部分に ID が振ってあり，それが種々の文書を通して使用されているか（記述の識別可能性）を見るのも一つの手法である．

6.2 要求仕様から測定する最終成果物の品質

- **Cost**: 製品のコストは，開発プロセスのコストに影響を受ける．要求仕様書から開発プロセスの費用や期間を見積もる手法はいくつか提案されており，ファンクションポイント法 [9, 10] がその代表的な例である．ファンクションポイント法では，要求仕様書に書かれている機能の複雑度や数をもとに推定する．
- **Performance**: Performance はどのようなアルゴリズムを用いてシステムが実現されるかや，実際のシステムの運用環境に依存する．要求分析段階で，アルゴリズムが指定される場合は，そのアルゴリズムを解析し計算量などを分析することよって Performance が推定できる．運用環境が仕様書に明記されている場合も，同様にその分析を行えばよい．
- **Security**: 要求仕様書にセキュリティポリシィの記述が明確にあるかどうかで判断する．セキュリティポリシィとは，計算機システムにおいて，どのアクションの遂行が許されていて，どのアクションの遂行が禁止されているかのルールの集合である [11]．

  また，オレンジブック [12] 等にあるセキュリティ評価のガイドライン (D: 最小プロテクション ~ A: 検証されたプロテクションなどのレベル分け) も利用できるが，高レベルの場合，開発手法・プロセスに関する審査も行うため，要求仕様レベルのみでは測定できない場合がある．
- **User-Friendliness**: User-Friendliness の測定法は，ヒューマンインターフェース分野で提案されつつある．これらの手法を要求仕様書に適用する．例えば，ユースケース記述やシナリオ記述を読ませてどれだけ人間の記憶に残っているか（長期記憶性）かどうかで判断する．プロトタイプを用いて評価するのも一般的な手法である．
- **Reliability**: 要求仕様書から統計モデルを組み立て，シミュレーションなどの手法によって，どれぐらいの確率で処理が正常終了するかを測定する．

7 WG メンバーの成果

本節では，要求工学ワーキンググループの各メンバーによる要求工学における測定と定量的評価にかかわる成果を紹介する．

大西らは，要求仕様の consistency を扱った研究を行っており，今後の定量的評価の足がかりとなると思われる [13, 14]．

佐伯らは，ゴール分析法とユースケース記述法の方法論のもとで，要求仕様の品質測定法を提案している．ゴール分析法で，顧客・ユーザの抽象的な初期ゴール（真のニーズを表していると考える）がどれぐらい具体化されたかというカバレージを測定し，顧客の要求達成度をもとに妥当性を測定する手法や，ユースケース間の依存度をもとに変更可能性を評価する手法を述べている．

中谷は，特定分野で一般的に使用されるユースケースを集め，定型ユースケースとし，定型ユースケースの再利用によってユースケース記述を支援する手法を考案している．また，定型ユースケースは「一般的に使用される」ため長期記憶性があるという理由から User-Friendliness が高いと考え，定型ユースケースの含有率で評価する手法も述べている．

橋本らは，仕様記述言語で記述したソフトウェアの再利用性について，作業時間の測定による評価を行った [15]．また，廣田らは，建築構造設計のドメインに特化した仕様記述言語と，データ処理用の非手続き型言語、C++の記述量を比較・評価した [16]．

友枝らは，品質特性要求の明確化とその達成するためのアーキテクチャ指向による要求分析法を提案している [17]．そこでは，ドメインにはそこで問題とすべき固有の非機能的要件があり，当該ドメインのアーキテクチャは各々それに対応した品質特性を持つことを明らかにした．このことを活用し，顧客が望む非機能的要件の明確化とその要件を満たす解（アーキテクチャ）の選定へと導けることを示した．

富田らは協調オブジェクトの連結性に着目した IDEF3 モデルの分析についての成果を発表予定である [18]．生産や業務のプロセスでは，様々なデータ，原材料，図面などがやりとりされている．それらは，複数のもの，人、設備が連携するプロセスにおける協調関係を具現化したもの (協調オブジェクト) と考えることができる．富田らが提案するシンセティックバ

インディング法は生産プロセスや業務プロセスに存在する協調関係を明示的に獲得する手法である．獲得された協調関係は IDEF3(Process Description Capture) を用いてシナリオとして表現できる． IDEF3 は，直観的に理解しやすいモデリング手法なので、業務エキスパートにとって取り組みやすい．この発表予定稿では，IDEF3 によりモデル化されたシナリオに対して，シンセテックバインディング法により獲得される協調オブジェクトに着目した OSTD の定量評価尺度を提案する．この尺度を用いることで、協調関係に着目した生産プロセスや業務プロセスのシナリオの獲得，評価が可能となる．

妻木らは，再利用性と拡張性のためのソフトウェア構築技法と評価指標についての研究成果を発表している [19]．

要求工学のリソース尺度として，Stakeholder そのものを測定することについては 2 節で既に述べた．海谷は，要求獲得段階ではなく，設計段階ではあるが，設計者間の意図の違いを測定する研究を行っており [20]，ソフトウェア要求定量化の可能性の 1 つを示している．

また，海谷らは，非機能要求を Stakeholder 間の共通評価尺度として，Stakeholder 間の利害のトレードオフをとりながら要求変更を行う手法も提案している [21]．同手法によって，変更に関して複数の可能性がある場合，非機能要求をもとにした評価点をもとに，より全体の評価が高くなるような選択肢を選ぶことができる．これは，要求仕様の定量的評価の応用例の 1 つである．

8 おわりに

本稿では，情報処理学会 ソフトウェア工学研究会 要求工学ワーキンググループにおける研究成果の 1 つとして，要求工学における定量的評価に関する報告を行った．現時点では，プロダクトに注目した測定のみを考慮しており，プロセスやリソースについては今後引き続き検討する必要がある．特に，プロセスを測定対象とした場合，開発途中での評価が可能な場合があり，要求プロセスの進捗管理に貢献できると思われる．

文献 [22] では，要求工学はソフトウェアプロジェクト管理に貢献する可能性について述べており，実際の要求工学プロセスに関するフィールドスタディを活発に行うべきであることも主張している．本ワーキンググループの活動を通して，そのような調査研究を行う可能性についても模索したい．

品質や定量的な評価の研究は，その歴史や量も膨大ではあるが，それらも適宜調査しつつ検討を進める必要がある．例えば，文献 [23] などでは，本稿の 6.2 節で議論した品質要求に関する仕様化について述べられている．

参考文献

[1] 要求工学ワーキンググループ　ホームページ． http://www.selab.cs.ritsumei.ac.jp/~ohnishi/RE/rewg.html.

[2] 海谷治彦, 中谷多哉子, 佐伯元司, 大西淳. ウィンターワークショップ・イン・金沢報告 要求工学. 情報処理学会研究報告, Vol. 2001, No. 56, pp. 87–89, Jun. 2001. ソフトウェア工学 131-14.

[3] Tom DeMarco. *Controlling Software Projects, Management, Measurement & Estimation*. Prentice-Hall, Inc. Yourdon Inc., 1982.

[4] N. Fenton. *Software Metrics: A Rigorous Approach*. Chanpman and Hall, 1991.

[5] IEEE Recommended Practice for Softare Requirements Specification, Oct. 1998. IEEE Std 830-1998, ISBN 0-7381-0332-2 SH94654(Print).

[6] IEEE Guide for Developing System Requirements Specifications, Dec. 1998. IEEE Std 1233-1998, ISBN 0-7381-0337-3 SH94654(Print).

[7] Lawrence Chung, Brian A. Nixon, Eric Yu, and John Mylopoulos. *Non-functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000.

[8] Linda A. Macaulay. *Requirements Engineering*. Applied Computing. Springer, 1996.

[9] A. J. Albrecht and Gaffney. Software Function, Source Lines of Code, and Development Effort Predicition: A Software Science Validation. *IEEE Trans. on Software Engineering*, Vol. 9, No. 6, pp. 639–648, Nov. 1983.

[10] 児玉公信. 実践 ファンクションポイント法. 日本能率協会マネジメントセンター, Jul. 1999.

[11] Dieter Gollmann. *Computer Security*. John Wiley & Sons, Feb. 1999.

[12] US Department of Defence. *DoD Trusted Computer System Evaluation Criteria*, 1985. DOD 5200.28-STD, (The Orange Book).

[13] Hideaki Sugimoto and Atsushi Ohnishi. A Supporting Method of Making a Consistent Software Requirements Specification Based on the Dempster and Shafer's Theory. *IEICE Trans. Information and Systems*, Vol. E83-D, No. 4, pp. 659–668, 2000.

[14] 大西淳. UML におけるモデル整合性検証支援システム. 電子情報通信学会論文誌, Vol. J84-D-I, No. 6, pp. 671–681, 2001.

[15] 橋本正明, 廣田豊彦, 横田和久. ドメインモデル

に基づくソフトウェア再利用に関する一考察. 情報処理学会論文誌, Vol. 36, No. 5, pp. 1040–1049, 1995.

[16] 廣田豊彦, 橋本正明, 長澤勲. 応用ドメインに特化した概念モデル記述言語に関する一考察. 情報処理学会論文誌, Vol. 36, No. 5, pp. 1151–1162, 1995.

[17] 友枝敦. アーキテクチャ指向による要求分析− 品質特性要求の明確化とその達成− . ソフトウェア科学会第 7 回ソフトウェア工学の基礎ワークショップ (FOSE2000) , 2000.

[18] 富田芳孝, 熊谷敏. 協調オブジェクトの連結性に着目した IDEF3 モデルの分析-シンセティツクバインディングによるシナリオの獲得と評価-. 経営情報学会誌, Vol. 10, No. 4, Mar. 2002. 掲載予定.

[19] 妻木俊彦, 森澤好臣. 再利用性と拡張性のためのソフトウェア構築技法と評価指標について. オブジェクト指向 1998 シンポジウム論文集, pp. 112–115, 1998.

[20] 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, Nov. 1999.

[21] Haruhiko Kaiya and Kenji Kaijiri. Lesson and Learned from a Requirements Elicitation for Conference Support System. In *Proceedings of Workshop On Web Intelligence and Software Engineering*, pp. 15–20, Beijing Polytechnic University, China, Aug. 2001.

[22] Hubert F. Hofmann and Franz Lehner. Requirements Engineering as a Success Factor in Software Projects. *IEEE Software*, Vol. 18, No. 4, pp. 58–66, Jul./Aug. 2001.

[23] Steven E. Keller and Laurence G. Kahn. Specifying Software Quality Requirements with Metrics. In Richard H. Thayer and Merilin Dorfman, editors, *System and Software Requirements Engineering, IEEE Computer Society Press Tutorial*, pp. 145–163. IEEE Computer Society Press, 1990.