



Title	システムエンジニアの問題解決を支援するQ&A方式に基づく情報提供の枠組みとその効果に関する研究
Author(s)	池田, 文人
Degree Grantor	奈良先端科学技術大学院大学
Degree Name	博士(工学)
Issue Date	2000-09-13
Doc URL	https://hdl.handle.net/2115/20095
Type	doctoral thesis
File Information	thesis.pdf



NAIST-IS-DT9861001

博士論文

システムエンジニアの問題解決を支援する
Q&A方式に基づく情報提供の枠組みと
その効果に関する研究

池田 文人

2000年9月13日

奈良先端科学技術大学院大学
情報科学研究科 情報処理学専攻

システムエンジニアの問題解決を支援する Q&A 方式に基づく情報提供の枠組みと その効果に関する研究¹

池田 文人

論文梗概

システムエンジニアがソフトウェア開発において生じる様々な問題を解決するために必要とする情報は膨大であり、その複雑性や多様性は増大する一方である。これらの情報は実際の問題状況に即してドキュメント化されているわけではなく、実際の問題状況に即した情報は個々のシステムエンジニアのノウハウや経験として蓄積されている。このためシステムエンジニアは他のシステムエンジニアに対面で質問することにより必要な情報を獲得しているのが現状である。その結果、質問される側のみならず質問する側の作業効率も低下し、組織全体のコミュニケーション効率も低下する。他者への質問そのものは、問題状況に即した情報を迅速に獲得できるとともに、他者からの回答および自分で記述した質問により問題状況が再認識されるため、問題解決の質と効率を向上させるために有効である。本研究では Q&A 方式によりシステムエンジニアの問題解決を支援するとともに Q&A の組織知としての蓄積を可能とするコンピュータを用いた枠組みを構築・実装し、その評価をおこなった。その結果、(1) 問題状況に即した情報の迅速な獲得、(2) 回答による問題状況の再認識、(3) 質問による問題状況の再認識、(4) Q&A の再利用性の向上、という四つの効果を確認することができた。1 章では本論文の主要題目について説明し、2 章ではシステムエンジニアに要求される情報に着目してソフトウェア開発支援を行う研究の課題と対面での他者への質問により必要な情報を獲得しているという現状の三つの弊害について論じる。3 章では他者への質問による問題解決における三つの効果を示し、これらをもつ現状の三つの弊害を解消するための枠組みと再利用性向上を加えた四つの効果について 4 章で論じる。5 章では実際の事例に基づく効果の検証について論じ、この検証結果に基づいた本枠組みを改善するための枠組み「コミュニティ知識ベース環境」を 6 章で提案する。今後のソフトウェア開発という観点からの考察を 7 章で論じ、最後に 8 章で総括する。

キーワード

ソフトウェア開発, 問題解決, Q&A, ネットワークコミュニティ, 知識管理

¹奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文, NAIST-IS-DT-9861001, 2000 年 7 月 21 日

NAIST-IS-DT9861001

博士論文

システムエンジニアの問題解決を支援する
Q&A方式に基づく情報提供の枠組みと
その効果に関する研究

池田 文人

2000年9月13日

奈良先端科学技術大学院大学
情報科学研究科 情報処理学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
博士（工学）授与の要件として提出した博士論文である

池田 文人

審査委員：井上 克郎 教授
 小山 正樹 教授
 関 浩之 教授
 松本 健一 助教授
 中小路 久美代 助教授

Assessing Q&A-Based Support Systems for System Engineers²

Fumihito Ikeda

Abstract

In software development, system engineers need a vast amount of information with increasing complexity and variety to solve their problems. Information available for them is currently not stored in such a way as to meet the demands of solving real-world problems, and may only be accessible through other system engineers. As a result, system engineers often resort to asking questions to other system engineers face-to-face, causing a decline in work efficiency by disturbing both those who answer and those who have to wait for answers. The question-and-answer (Q&A) paradigm itself is a commonly-used helpful way to obtain information necessary to solve a problem. The questioner can promptly acquire problem-related information by formulating a question. The research explored in this thesis focuses on a Q&A-based framework for supporting system engineers by providing information in a Q&A form useful for their problem solving and by storing such Q&As as a part of organizational knowledge. This thesis describes four effects of the approach on helping system engineers: (1) timely acquisition of helpful information, (2) awareness of a new perspective of a problem implied by an answer provided by their peers, (3) reformulation of a problem situation by expressing it as a communicable question, and (4) accumulation of reusable Q&As. This thesis starts with Chapter 1, which explains the main theme of my dissertation research. Chapter 2 describes related works in terms of information system engineers need in their software development practice, and discusses three issues in the existing support. Chapter 3 explains three effects of using Q&A as a paradigm to solve their information needs. Chapter 4 of this thesis describes my Q&A-based framework that amplifies the three effects and at the same time enabling the accumulation of organizational knowledge by storing Q&A's. The framework has been applied as case study, which is presented in Chapter 5. Chapter 6 describes a community knowledge-based environment as an extension of this framework. Chapter 7 envisions future software development styles based on the approach. Chapter 8 concludes the thesis.

Keywords

software development, problem solving, Q&A, network community, knowledge management

²Doctor's thesis, Department of information processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT-9861001, July 21, 2000

目次

第1章	序論	1
1.1	本研究の背景	1
1.2	本研究の目的	3
1.3	本論文の構成	4
第2章	ソフトウェア開発に必要な情報に着目した研究の課題と現状	7
2.1	ソフトウェア開発に必要な知識や情報の削減	8
2.2	ソフトウェア開発におけるコミュニケーションの支援	9
2.3	ソフトウェア開発における設計理由の記録	11
2.4	ソフトウェア開発におけるノウハウの蓄積と再利用	12
2.5	対面での他者への質問による情報獲得の三つの弊害	13
2.6	まとめ	14
第3章	問題解決としてのソフトウェア開発における他者との Q&A による効果	17
3.1	問題解決に必要な情報を迅速に獲得することの困難さ	18
3.2	他者への質問による問題状況に即した情報の迅速な獲得	19
3.3	他者への質問による問題状況の明確化における二つの効果	20
3.3.1	回答による問題状況の再認識	21
3.3.2	質問による問題状況の再認識	21
3.4	まとめ	22
第4章	システムエンジニアの問題解決を支援する回答作業の専門化と電子メールの利用による Q&A 方式に基づく情報提供の枠組み	25
4.1	他者との Q&A における電子メールの効果	26
4.2	他者との Q&A における電子メールの問題	27
4.3	問題状況の共有における電子メールの効果	28
4.4	回答作業の専門化による効果	29
4.5	まとめ:本アプローチにより期待できる四つの効果	29
第5章	事例に基づく効果の検証	31
5.1	検証対象となる事例	31
5.2	検証のための枠組み	33
5.3	第一の効果の検証:問題状況に即した情報の迅速な獲得の検証	37
5.3.1	検証のための仮説	37
5.3.2	仮説の検証方法	38
5.3.3	検証結果	38
5.4	第二の効果の検証:回答による問題状況の再認識の検証	42
5.4.1	検証のための仮説	42
5.4.2	仮説の検証方法	43

5.4.3	検証結果	43
5.5	第三の効果の検証：質問の記述構造に関する調査	44
5.5.1	検証のための仮説	45
5.5.2	仮説の検証方法	47
5.5.3	検証結果	48
5.6	第四の効果の検証：Q&Aの再利用性の検証	51
5.6.1	検索結果として得られたQ&Aの有益性に関する調査	51
5.6.2	有益なQ&Aを獲得するまでの迅速性に関する調査	52
5.7	まとめ	53
第6章	コミュニティ知識ベース環境：検証結果に基づく改善	55
6.1	Q&Aの形成と再利用の統合的支援のための調査	55
6.1.1	形成と再利用とにおけるQ&Aの質の相関関係に関する調査	56
6.1.2	質の高いQ&Aの迅速な獲得に関する調査	56
6.2	Q&Aの形成と再利用の統合的支援のための条件	59
6.3	コミュニティ知識ベース環境	60
6.3.1	コミュニティ知識ベース環境の枠組み	60
6.3.2	プロトタイプシステム	62
6.4	まとめ	64
第7章	考察	67
7.1	ソフトウェア開発における学習する組織の実現の必要性	67
7.2	コミュニティ知識ベース環境による学習する組織の実現の支援	69
7.3	今後の課題	71
7.4	まとめ	72
第8章	結論	73

目次

1.1 システム規模の推移	2
2.1 Experience Factory の枠組み	13
3.1 外部に存在する情報に対する認識	19
3.2 他者からの回答による問題状況の再認識	22
3.3 質問による問題状況の再認識	23
5.1 技術支援システムの枠組み	32
5.2 調査対象となる Q&A の例	34
5.3 Q&A に含まれるドメインタームの例	34
5.4 アンケートの例	35
5.5 効果検証のための枠組み	36
5.6 回答期限に関する調査結果	39
5.7 回答期限と回答に対する認識との関係	39
5.8 期限超過 5 日までを期限内とした場合の回答期限と回答に対する認識との関係	41
5.9 回答に対する認識に関する調査結果	44
5.10 回答に対する認識と Q&A の分類との関係	45
5.11 質問におけるドメインタームの分布と回答に対する認識 (1)	49
5.12 質問におけるドメインタームの分布と回答に対する認識 (2)	50
5.13 検索結果に対する認識の分類	52
5.14 検索結果として得られた Q&A の有益性に関する調査結果	53
5.15 有益な Q&A を獲得するまでの迅速性に関する調査結果	54
6.1 形成時と再利用時の Q&A の有益さの相関関係	57
6.2 質問の記述構造を反映した検索手法による適合率に関する調査結果	58
6.3 コミュニティ知識ベース環境の枠組み	61
6.4 QA&履歴の参照画面例	62
6.5 質問入力画面例	63
6.6 回答入力画面例	64
6.7 再利用時の評価入力画面例	65
7.1 コミュニティ知識ベース環境における学習過程	69
7.2 コミュニティ知識ベース環境におけるアイデンティティ確立の過程	71

表 目 次

1.1	本論文の主要題目	5
2.1	COTS に基づく開発プロセス	8
5.1	技術支援システムの特徴	33
5.2	回答に対する評価と回答期限との関連性 1	40
5.3	回答に対する評価と回答期限との関連性 2	42
5.4	回答に対する評価と質問分類との関連性	46
5.5	質問の種類と記述構造との関係	47
5.6	ドメインタームを抽出した質問数	48
5.7	抽出したドメインタームの平均個数	49

第1章 序論

本章では、本研究の背景と目的、および本論文の構成について説明する。システムエンジニアに要求される知識や情報は膨大であり、ますます増加する傾向にある。個々のシステムエンジニアに要求される知識や情報が増加すれば、作業効率は低下することから [2]、ソフトウェア開発に必要となる知識や情報に着目してソフトウェア開発の質や効率を向上させる様々な研究が行われている。しかし、いずれの研究も課題を抱えており、現状では他のシステムエンジニアに対面で質問することにより問題解決に必要な知識や情報を獲得している。他者への質問には問題状況に即した知識や情報を迅速に獲得できるという利点がある反面、開発プロジェクト全体の作業効率や質を低下させるという欠点がある。そこで、本研究では、システムエンジニアがソフトウェア開発において生じる問題を解決するために必要となる知識や情報を、回答作業の専門化と電子メールの利用による質問と回答 (Q&A) に基づいて効率的に提供するアプローチを提案し、ソフトウェア開発の質と効率を高めることを目的とする。本論文ではこうした研究の背景や目的について論じるとともに、本アプローチにおける効果とその検証結果、そして検証結果に基づく本アプローチの改善策について論じる。

1.1 本研究の背景

システムエンジニアがソフトウェア開発において生じる様々な問題を解決するために必要となる知識や情報は膨大である。ソフトウェア開発は要求分析 → 設計 → コーディング → 試験 → 運用保守といった一連のプロセスを経て行われ [20]、これらのプロセスはライフサイクルとして互いに密接に関係し合っている [9]。これらのプロセスを経てソフトウェア構成に加えらるる変更や修正を管理し [13]、プロジェクト自体の管理や品質管理もそれぞれのプロセスにおいて実行される [14, 33]。こうした様々な作業 (問題解決) はそれを支援するツールを用いて実行される [20]。したがって、個々のシステムエンジニアがそれぞれの問題解決を行うためには、一連のソフトウェア開発プロセスに関する知識や情報が要求されるとともに、ソフトウェア構成や品質、プロジェクトなどの管理、ソフトウェア開発で利用される様々なツール、そして開発プロセスや各プロセスで生成される仕様書や設計書などの生産物に関する知識や情報も要求される [5]。

システムエンジニアに要求されるこうした知識や情報はますます増加する傾向にある。一つの原因は、コンピュータシステムに対する顧客の要求の複雑化や情報通信技術の急速な進展によりソフトウェア開発の規模が増大してきていることである [81, 86] (表 1.1 を参照 [86])。個々のシステムエンジニアは複雑化する顧客の要求を理解し技術的な変化に対応する必要がある。また、開発規模の増大に伴い開発プロジェクト全体で必要となる知識や情報、および生成される仕様書や設計書などの生産物に関する知識や情報も増加するため、これらも把握する必要がある。

これらの膨大な知識や情報を一人のシステムエンジニアがすべて把握することは不可能であり [61]、自分の頭の中以外の外界に存在する知識や情報を自分の置かれている問

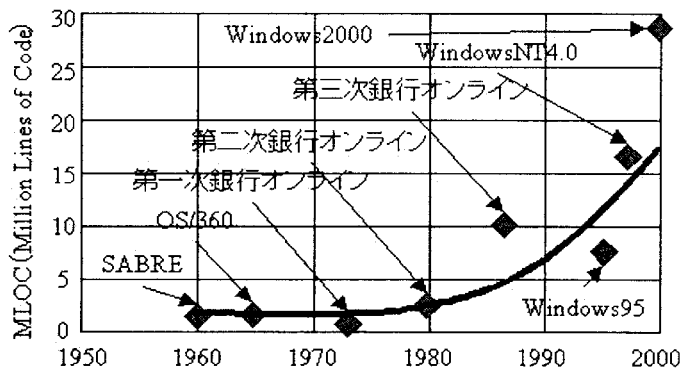


図 1.1: システム規模の推移 (文献 [86] より)

題状況に応じて獲得する必要がある。これらの膨大な知識や情報は様々なマニュアル、開発手順書、設計書、仕様書といったドキュメントとして記録され、膨大なマニュアルや何百冊ものキングファイルに綴じられたドキュメント、および膨大なデータベースの中から個々のシステムエンジニアは随時参照することが可能である。しかし、これらのドキュメントはそれが参照される際の実際に個々のシステムエンジニアが置かれている問題状況に即して体系化されているわけではなく、その問題状況とは無関係な知識や情報を多量に含んでいる。このため問題解決に必要な知識や情報を、そうした外界に存在するリソースから迅速に獲得することは困難である [26]。したがって、ソフトウェア開発に必要な知識や情報が増大すれば、必要な知識や情報を獲得するためのシステムエンジニアのオーバーヘッドは増大し、作業の質も効率も低下する [2]。

このような背景から、ソフトウェア開発に必要な知識や情報に着目してソフトウェア開発の質と効率を向上させることを目的とした様々な取り組みが行われている。それらをまとめると以下の四つのアプローチになる。

- (a) ソフトウェア開発に必要な知識や情報の削減
- (b) ソフトウェア開発におけるコミュニケーションの支援
- (c) ソフトウェア開発における設計理由の記録
- (d) ソフトウェア開発におけるノウハウの蓄積と再利用

(a) のアプローチでは、従来のソフトウェア開発とは異なる新たな知識や情報が必要となるため、個々の問題状況に応じた知識や情報が必要となることに変わりはない。こうした個々の問題状況に応じた知識や情報は、個々のシステムエンジニアの頭の中にノウハウや経験として分散して存在するケースが多いため、(b) のアプローチのように他者とのコミュニケーションを支援することが重要となる。しかし、(b) のアプローチではコミュニケーションで発生した知識や情報をいかに再利用するかという観点からの考慮が不足しているとともに、コミュニケーション支援ツールを導入すること自体にも課題を抱えている [30, 40]。ソフトウェア開発におけるコミュニケーションの内容を記録し再利用することによりソフトウェア開発の質や効率を向上させるというアプローチが (c) である。しかし (c) のアプローチでは設計理由の再利用性を高めるとその記録の負荷が高くなり、逆に記録性を高めると再利用性が低下するという問題を抱えている。これらいずれのアプローチにおいても、個々の問題状況に応じてシステムエンジニアが必要な

知識や情報を迅速に獲得できることと、そこで生じた知識や情報を効率的に再利用することが統合的に支援されている必要がある。そうしたアプローチが (d) である。しかし、(d) のアプローチでは経験やノウハウを実際に収集するための方法や、効率的な蓄積形態、再利用の方法などが具体化されていない。本研究におけるアプローチはシステムエンジニアの個々の問題状況に応じた情報提供とその再利用とを統合的に支援するための具体的な枠組みを提案するものである。

現状のソフトウェア開発では、実際の問題解決に即した知識や情報は個々のシステムエンジニアのノウハウや経験として部分的かつ分散して蓄積されている場合が多いため、自分の置かれている問題状況に応じて必要な知識や情報を持っていそうな他のシステムエンジニアに対面で質問することにより問題解決に必要な知識や情報を獲得している場合が多い [36]。しかし、他者に対面で質問することには以下の三つの弊害がある [18]。

第一の弊害: 質問された人は質問した人とは異なる問題状況に置かれているため、他者から質問を受けることにより作業の中断を強制され作業効率は低下する。

第二の弊害: 質問する人も必要な知識や情報を持っていそうな人を見つけ出し、直接コミュニケーションを行う機会を獲得するための交渉や調整を行うといった、実際の問題解決とは無関係な作業を行う必要が生じるため、作業効率は低下する。

第三の弊害: こうした質問と回答の対 (Q&A) は実際の問題解決事例として有益なノウハウであるが、他のシステムエンジニアに直接質問を行った場合にはこうしたノウハウは個人の頭の中に蓄積されるだけで他のシステムエンジニアが再利用できるように明示化され蓄積・共有されることがない。このため同じような Q&A が別のシステムエンジニアらにより繰り返されることとなり、ソフトウェア開発を行う組織全体の作業効率が低下する。

このような弊害がある一方で、他者へ対面で質問することには、問題解決の質と効率を高める上で以下の三つの効果がある。

第一の効果: 問題状況の共有が迅速に行えることにより、問題状況に即した知識や情報を迅速に獲得することができる。

第二の効果: 回答として問題状況に関連した意識していなかった知識や情報が獲得できることにより問題状況をより明確化することができる。

第三の効果: 問題状況を質問として明示化することにより問題状況を再認識でき、問題状況をより明確化することができる。

本研究のアプローチは他者への質問によるこれらの効果に着目し、前述した弊害を解消することにより、Q&A 方式に基づいてシステムエンジニアの問題解決に必要な知識や情報を迅速に提供する。

1.2 本研究の目的

本論文の主要題目をまとめると表 1.1 のようになる。本論文は他者への質問による三つの弊害を解消し、かつ他者への質問による問題解決における三つの効果を高めることにより、システムエンジニアが必要とする知識や情報を効率的に提供することを目的とする。そのために、回答作業の専門化と電子メールの利用による Q&A 方式に基づく情報提供のアプローチを提案する。

電子メールを利用することにより、質問された側は自分の作業状況に合わせて回答を行うことができ [1, 6]、質問する側は相手の場所や作業状況を考慮する必要がなく [1, 6]、

質問と回答は電子メールとして記録されるため [87], 1.1 節で述べた他者への質問による三つの弊害を緩和することができる。また, 相手への回答や質問を記述する必要が生じることから, 記述したものを客観的に認識することが可能となり, 問題状況に関するより多くの知識や情報を相手に伝えることができる。これにより, 1.1 節で述べた他者への質問による三つの効果のうち第二と第三の効果が高められる。しかし, 相手の作業状況によっては十分な回答が得られなかったり, 必要とする期間内に得られなかったりするため, 第一の効果は期待できない。このような欠点を補うために回答作業を専門化し, 開発プロジェクトから分離する。これにより質問したシステムエンジニアに対する迅速な回答の提供を保証する。また, どこに質問したらよいかが明確になるため質問を行うためのオーバーヘッドが削減されるとともに, 形成された質問と回答 (Q&A) の管理を一括して行うことが可能となるため, システムエンジニアが再利用のために Q&A を管理する負荷が削減される。

以上のことから, 本アプローチにより期待できる効果は他者への質問による第三の弊害, すなわち他者との質問と回答の再利用性における課題を解消した結果として, 他者への質問による三つの効果に,

第四の効果: 問題解決事例としての Q&A の再利用性向上

という効果を加えた四つである。

これら四つの効果を検証するために, 株式会社エヌ・ティ・ティ・データで開発・運用している本アプローチに基づく情報提供の枠組みである「技術支援システム」において, 質問を行った延べ約 13,000 人のシステムエンジニア, および蓄積されている約 13,000 件の Q&A を対象に表 1.1 に示すような分析調査を行った。本論文では, これらの検証の結果, 本アプローチにより四つの効果が得られることが実際に確認されたことについて論じるとともに, これらの検証結果に基づいて本アプローチを改善するために開発したプロトタイプシステム「コミュニティ知識ベース環境」の概要と有用性について論じる。

1.3 本論文の構成

2 章では, まず, ソフトウェア開発に必要な知識や情報に着目することによりソフトウェア開発の質や効率を向上させることを目的とした四つのアプローチに基づく研究に関して説明し, それらの課題について論じる。次に, システムエンジニアは他のシステムエンジニアに対面で質問を行うことにより問題解決に必要な知識や情報を獲得している現状における三つの弊害について論じる。

3 章では, まず膨大なドキュメントを検索するのではなく, 他者に直接質問することにより必要な知識や情報を獲得してしまう原因について認知科学的な観点から論じる。次に他者に質問を行うことの問題解決における三つの効果について, 問題状況に関する知識や情報の迅速な獲得と問題状況の明確化という観点から論じる。問題状況に関する知識や情報を迅速に獲得できるだけでは問題解決の質や迅速性が向上できるわけではないことから, 他者への質問により問題状況に関して得られた知識や情報の問題解決の質的な向上における効果について, 他者からの回答と自分で問題状況を質問として明示化することの二つの観点から論じる。

4 章では, 他者に質問を行うことの三つの効果を高めつつ, 2 章で論じる三つの弊害を解消するために, 回答作業の専門化と電子メールの利用による Q&A 方式に基づくシステムエンジニアへの情報提供というアプローチを提案し, 他者への質問による三つの効果に再利用性の向上という効果を加えた四つの効果が本アプローチにより期待できることについて論じる。

表 1.1: 本論文の主要題目

研究の目的	問題解決の質と効率の向上には問題状況の迅速かつ適切な明確化が重要であるため、そのために必要な情報を効率的にシステムエンジニアに提供すること			
ソフトウェア開発の特徴	問題解決に必要な情報を迅速に獲得するために他のシステムエンジニアに対面で質問を行うことが多いこと			
現状の課題 (他者への質問による弊害)	質問する側とされる側との問題状況の相違による質問される側の作業効率の低下	対面で質問するためのオーバーヘッドが大きいことによる質問する側の作業効率の低下		やり取りされた質問と回答が記録され再利用されることは困難であることによる開発組織全体のコミュニケーション効率の低下
現状の利点 (他者への質問による効果)	問題状況に迅速な共有が可能であることによる必要な情報の迅速な獲得が可能	回答に含まれる他者の異なる観点からの情報を獲得することによる問題状況の再認識が可能となり問題状況をより明確化することが可能	質問として問題状況を記述することによる問題状況の再認識が可能となり問題状況をより明確化することが可能	
アプローチ	回答作業の専門化と電子メールの利用による Q&A 方式に基づくシステムエンジニアへの情報提供			
期待される効果	問題状況に即した情報の迅速な獲得（問題解決の迅速化）	回答に含まれる他者の異なる観点からの情報を獲得することによる問題状況の再認識（問題状況の明確化）	質問として問題状況を記述することによる問題状況の再認識（問題状況の明確化）	実際の問題解決事例としてやり取りされた質問と回答が記録・管理されることによる再利用性向上（問題解決の効率化）
効果の検証	質問したシステムエンジニアが指定した回答期日と実際の回答日時との差に関する調査	質問したシステムエンジニアの獲得した回答に対する認識に関する調査	回答に対する認識と質問の記述構造との関係に関する調査	検索結果として得られた Q&A に対する認識に関する調査、および質問構造を反映した検索手法に関する評価
改善策	問題状況の明確化のための能力と知識の習得を支援する「コミュニティ知識ベース環境」			

5章では、まず、本アプローチに基づいて株式会社エヌ・ティ・ティ・データで開発運用している「技術支援システム」の枠組みについて論じる。この技術支援システムを対象に4章で論じる四つの効果を検証するための調査の枠組みについて論じ、個々の調査における仮説・方法・結果についてそれぞれ論じる。

6章では、知識管理においては知識の形成と再利用を統合的に支援する必要があるという観点から、技術支援システムに蓄積されているQ&Aに対して行った調査について論じる。この調査結果、および5章で論じる調査結果に基づいて、質問形成および検索において質問の記述構造を意識させることを支援するために開発したプロトタイプシステム「コミュニティ知識ベース環境」の仕組みについて論じる。

7章では、6章で論じたコミュニティ知識ベース環境が、今後のソフトウェア開発を行う組織において求められる「学習する組織」の実現に有効であることについて、コミュニティ知識ベース環境における学習過程とアイデンティティの確立過程という二つの観点から論じる。

8章では、本論文をまとめるとともに、本研究により得られた知見が実際にどのように活用されているかについて論じる。また、ソフトウェア開発における知識管理の方法としては、現在主流であるトップダウン型の知識収集方法よりも本アプローチのようなボトムアップ型の知識収集方法の方が優れていることについて論じる。最後に、本研究の社会的な意義について、人に優しいコンピュータ社会の実現に対する貢献、および人の創造的活動に対するコンピュータシステムによる支援の可能性という観点から論じる。

第2章 ソフトウェア開発に必要な情報に着目した研究の課題と現状

本章では、ソフトウェア開発に必要な知識や情報に着目してソフトウェア開発の質や効率を向上させることを目的とした四つのアプローチに基づく研究の課題について論じるとともに、システムエンジニアが問題解決に必要な知識や情報を迅速に獲得するために他のシステムエンジニアに対面で質問を行う場合が多いという現状とその三つの弊害について論じる。

ソフトウェア開発に必要な知識や情報が増大すれば開発の質や効率も低下する [2]。そこで、ソフトウェア開発に必要な知識や情報を削減するアプローチとして、COTS (Commerce Off The Shelf) [37, 47] とデザインパターン [28, 60] を取り上げる。COTS では膨大な COTS に関する知識や情報、および COTS のベンダーに関する知識や情報などが新たに必要となってきたという問題について論じる。デザインパターンに関しては、実際の問題状況に応じてデザインパターンを適用するためのノウハウが新たに要求されるという問題について論じる。

そうした実際の問題を解決するノウハウは個々のシステムエンジニアの頭の中に部分的に分散して存在する機会が多いことから、ソフトウェア開発における他者とのコミュニケーションを支援することはソフトウェア開発の質や効率を向上させる上で重要である。このようなアプローチとして、ソフトウェア開発におけるグループウェア [11, 42] を取り上げ、コミュニケーションの結果として生成された知識や情報の再利用を考慮していないこと、およびグループウェアの導入自体に課題を抱えていることについて論じる。

ソフトウェア開発において生成された知識や情報の再利用性を高めるという観点から設計理由の記録 [59, 39] を取り上げる。仕様書や設計書などの生産物が生成されるまでに行われた議論やコミュニケーションの内容を記録することにより、生産物を他者が利用する際の理解を助けるというアプローチである。このようなアプローチによるソフトウェア開発の質や効率を高める効果について論じるとともに、設計理由の記録と再利用における問題、およびソフトウェア開発に必要な知識や情報を増加させてしまっているという問題について論じる。

以上のことから、実際の問題解決に必要な知識や情報を迅速にシステムエンジニアに提供するとともに、そうした問題解決のノウハウや経験を蓄積し再利用することを統合的に支援することがソフトウェア開発の効率や質を高める上で重要である [7]。このようなアプローチとして“Experience Factory [7, 8]”がある。Experience Factory ではノウハウを収集し蓄積する組織を開発プロジェクトから独立させており、本アプローチに類似する。しかし、Experience Factory ではノウハウや経験を効率的に収集、蓄積し、再利用する方法などが具体化されていないという課題を抱えていることについて論じる。

このようにソフトウェア開発に必要な知識や情報に着目して様々なアプローチによる取り組みがなされているが、いずれのアプローチも課題を抱えており、個々のシステムエンジニアは実際の問題状況に応じて他のシステムエンジニアに対面で質問することにより必要な知識や情報を獲得している場合が多いというのが現状である [36, 18]. これは問題状況に応じて必要な知識や情報をその場で獲得できるためである。しかし、対面で他のシステムエンジニアに質問を行うことにより、(1) 相手の作業効率を低下させるだけでなく、(2) 質問する人の作業効率も低下し、(3) さらに質問と回答が記録されずらいために再利用が困難である、という三つの弊害を生じさせることについて論じる。

2.1 ソフトウェア開発に必要な知識や情報の削減

ソフトウェア開発に必要な知識や情報の増加を抑制するために、COTS (Commercial Off The Shelf) と呼ばれる市販のアプリケーションソフトウェアやコンポーネントをより大きなシステムの要素として組み込むという開発形態が主流になってきている [37]. COTS を利用することにより効率的なソフトウェア開発が実現されるだけでなく、実際の利用を通して信頼性を証明された市販製品を部品として利用するため信頼性の高いソフトウェアを開発することができる [47].

COTS に基づく一般的な開発プロセスは表 2.1 [78] のようになる。COTS を利用した場合の利点は、第三フェーズの統合と試験において COTS はブラックボックスとして扱われるということである。COTS を用いた部分の内部設計やモジュールのコーディング、単体試験などの工数は削減され、これらの作業に必要な知識や情報も削減される。

表 2.1: COTS に基づく開発プロセス (文献 [78] より)

Phase	Major Activities	Products	Management Check-points
Requirements analysis and COTS identification	Requirements analysis, COTS survey, preliminary evaluation	Requirements, High-level architecture, Candidate COTS	Systeme requirements review
Architecture definition and COTS selection	COTS evaluation, requirements modification to use existing COTS, prototyping	Modified requirements, System architecture, Final COTS	Systeme design review
System integration and test	Use-case implementation, independent test	Delivered system	Use demonstrations, Operational readiness review
Technology update and system management	Evaluation of new products and technology	Enhanced system	Use demonstrations

しかし、第一フェーズの要求分析においては膨大な COTS に関するそれぞれの機能や性能評価、他の COTS との統合性などに関する知識や情報が求められる。第二フェー

ズのアーキテクチャー設計においても膨大な COTS に関する評価情報が求められる。第四フェーズの運用保守では COTS 部分で問題が生じた場合にそのベンダーとの契約に基づく交渉などに関する知識や情報が要求される。

このように、COTS に基づくソフトウェア開発は従来のソフトウェア開発に比べるとコーディングやデバッグ、結合試験やコードインスペクションといった作業が削減され、そうした作業を遂行するために必要となる知識や情報が削減される。しかし、膨大な市販ソフトウェア製品の機能などに関する知識や情報、製品間での品質や性能における違いなどに関する知識や情報、製品間の互換性に関する知識や情報など、膨大な知識や情報が新たに要求される [47]。

こうした知識や情報は実際の問題状況に応じて異なる。しかし、COTS に関する知識や情報は Web やマニュアル、ベンダーやサードパーティが提供する製品評価データなどのドキュメントに記録されており、実際の問題状況に即して体系化されているわけではなく、無関係な知識や情報を多量に含んでいる。したがって、COTS を用いる場合でも個々の問題状況に応じて必要な知識や情報を効率的にシステムエンジニアに提供することが求められる。

一方、問題状況とそれに対する設計パターンを記録し再利用することによりシステムエンジニアの問題解決に必要な知識や情報を削減する取り組みがデザインパターンである。COTS がコーディングや試験のプロセスで必要となる知識や情報を削減することを目的としているのに対して、デザインパターンはより上流工程での知識や情報を削減することを目的としている。

ソフトウェア開発におけるデザインパターンとは、主にオブジェクト指向に基づいて開発されたソフトウェアシステムに典型的に現れる構造や機能を表現した抽象的なソフトウェア部品として再利用することである [28, 60]。COTS と同様に、このようなソフトウェア部品は繰り返し使われてきたものであるため信頼性が高く、問題状況が明確化されればそれを解決するための方法を考え出す設計プロセスに必要な知識や情報が削減される。

しかし、あるデザインパターンを現実の問題状況に適用すれば必ず新しい問題が発生する。このため、新しい問題状況に対してデザインパターンを修正することを繰り返していく必要がある [3]。そのためにはどのような問題状況においてどのようなパターンを利用すればよいか、どのようにデザインパターンを修正すればよいかという知識や情報が必要となる [26]。したがって、COTS の場合と同様に、デザインパターンを用いる場合でも個々の問題状況に応じて必要な知識や情報を効率的にシステムエンジニアに提供することが求められる。

2.2 ソフトウェア開発におけるコミュニケーションの支援

個々の問題状況に応じて必要となる知識や情報は、個々のシステムエンジニアの頭の中に部分的に分散して存在する 경우가多く、ソフトウェア開発においてシステムエンジニアが対面で他者へ質問を行ったり情報提供したりする時間は全作業時間の 50% 以上に及ぶという指摘もある [36]。したがって、ソフトウェア開発の質や効率を向上させるためには、他者とのコミュニケーションを支援することが重要である [84]。実際、今後の CASE (Computer Aided Software Engineering) ツールに期待される機能としては、Coordination, Navigation, Communication, Control, Adaptation であると言われており [92]、最初の四つの機能は他者とのコミュニケーションに関わる機能である。

従来のソフトウェア開発におけるコミュニケーション支援は、レビュー方法論 [27]

などに頼ってきた。しかし、他者とのコミュニケーションや情報共有におけるグループウェア [22] の効果が論じられるようになると、ソフトウェア開発へのグループウェアの導入が進められてきた。

Bellcore で開発された ICICLE [11] はソースコード検査を支援する同期型のグループウェアであり、X Window システムで動作し、ソースコードを表示するウィンドウ、コードに対するコメントを表示するウィンドウなど複数のウィンドウを利用したものである。ICICLE には以下のような特徴がある。

- コード検査会議における司会者・検査者・記録係などの議論における役割を支援する。
- ペーパーレスでの検査作業を支援する。すなわち、ソースリストのハードコピーや紙への記録などは一切行わない。
- 検索者は自分のワークステーション上で作業を行うことができる。
- コードに対するコメントは検査会議に参加したメンバー間で共有される。

ICICLE の適用実験の結果、検索作業の効率化に有効であったと報告されている。また、手作業によるコード検査と比較した場合、例えば記録係が決定権を持ち始めるなど、参加者の役割に変化が生じることが報告されている。

ソフトウェア開発で生成された情報を共有することを中心に他者とのコミュニケーションを支援するグループウェアとして KyotoDB がある [42]。KyotoDB はデータベースを中心とした総合的な CASE 環境であり、作業員間の対話支援機能（協調活動支援ツール）を含んでいるところに特徴がある。KyotoDB ではソフトウェア構成要素とそれらの相互関係をデータベースに蓄積しており、仕様変更などが生じた際に協調活動支援ツールが関係する作業員間での話し合いを仲介する。変更が影響する部分の担当者リストを自動的に作成し、話し合いの仲立ちをするという機能を持つ。実際の話し合いは UNIX の phone 機能などを用いて行われる。

こうしたグループウェアでは実際に問題解決を行う場合のコミュニケーションは支援されているが、そこでやり取りされた内容をいかに記録し再利用するかという観点欠缺している。問題解決の事例や経験を蓄積し再利用することは、効率的で質の高いソフトウェア開発を行うために重要である [7] ことから、コミュニケーションの支援だけでなく、そこでやり取りされた情報を記録し再利用することも含めて検討を行う必要がある。

また、グループウェアはグループを構成するメンバー全員がそれを利用しなければ導入の効果が得られないことから [22]、その導入に際して以下のような課題を抱えている。

- グループウェアの導入により利益を得る人と仕事が増えるなどの不利益を被る人との不一致がある場合、グループウェアの導入による効果は期待できない [30]。
- 電子メールのようにユーザの利益と不利益が対称的な場合でも、グループを構成する特定の人々（クリティカルマス）がグループウェアを受け入れない場合、グループ全体として効果が得られない [40]。

さらに、グループウェアを導入した効果を評価するにあたっては、以下のような問題があることが指摘されている [83]。

効果の測定における障害：非同期型グループウェアはコミュニケーション効率を高めることを目的とする場合が多い。このようなコミュニケーション効率を計測するためには、現場での作業とは直接関係のないアンケート等の調査を行う必要があり、効果の測定にあたって現場から協力を受けることが困難である。

実用評価における障害：グループウェアはコラボレーションという定量的計測が困難な人間活動を支援するため、現場での運用評価により効果を説明するしか現在のと

ころ評価の手立てがない。しかし、実験的なシステムを一定数以上のユーザに現場で利用してもらうためには既存のシステムとの置き換え等が必要となり、それが許されない場合がほとんどである。

人間行動の予測における障害: Grudin の指摘を回避するためには、グループの各メンバーが通常どのような行動をとっているかを把握するとともに、グループウェアの導入によりどのような行動をとるようになるかを予測する必要がある。しかし、人間の行動のすべてを把握し予測することは不可能であるとともに、メンバーのすべてが利益を得るようにグループウェアを設計するのは不可能である。

グループウェアを導入する場合には上述した二つの課題を考慮する必要があるとともに、導入した場合に期待される効果を明確にしておき、その効果を実際にどのように検証するかを十分検討しておく必要がある。このため、実際のソフトウェア開発におけるコミュニケーションはまだ対面で行われる場合が多い [36, 18]。

2.3 ソフトウェア開発における設計理由の記録

ソフトウェア開発における議論やコミュニケーションの内容を記録することにも課題がある。ソフトウェア開発において形成される知識や情報は、各プロセスの生産物として定義された仕様書や設計書、プログラムコード、試験項目表、運用設計書などがある。しかし、このような生産物の記述を標準化したり、ある生産物の記述を支援したりすることよりも、これらの生産物を得るまでの過程がソフトウェア開発の効率を向上させる上で重要であることが指摘されてきた [16]。このような考えに基づき、ソフトウェア開発の各プロセスにおける生産物を得る過程で議論された設計理由をその生産物とともに記録する支援が行われている [49, 59, 39]。ソフトウェア開発における仕様書等の生産物は、そこには明示化されていない目的や要求、他の生産物やドキュメントなどに基づいて形成されるものであり、ある生産物が得られるまでの過程においては様々な技術的・非技術的な議論や検討が行われている。このような議論や検討を記録しておくことは、以下のような効果がある [84, 77]。

- 仕様変更や設計変更などにおいて、それらが決定された経緯を把握することができれば、変更可能な箇所や変更による影響部分などを容易に同定することが可能となる。
- 過去のシステム事例や他者の構築したモジュールを再利用する場合など、カスタマイズを行うための資料となる。
- 仕様や設計が決定された理由を忘れて、過去の議論や検討を蒸し返すといった無駄が削減される。
- 特に明確な理由がないまま決定された仕様や設計を必要以上に絶対視し、見直しを怠ることがなくなる。
- 議論や検討を記録するという行為を行うことにより、議論や検討の内容を明確に意識することができ、問題を積み残したまま次のプロセスに進んでしまうことを回避できる。

設計理由の記録に関しては、議論を構成する要素とその関係を定義したモデルに基づいて記録するという方法が一般的である。このような議論モデルには大きく二種類のものがある。一つは IBIS (Issue Based Information Systems) [15] という意志決定の際の議論モデルを用いたものであり [59]、他方は議論内容を IS - A 階層や論理表現と

いった知識処理技術を用いて記述したものである [39]。いずれの手法においても議論中の発言を分類し、それらを関係付けることにより議論を記録する。

これらのモデルを用いた設計理由の記録方法には、議論中に記録する方法と議論の後に記録する方法とがある [39]。発言と同時に記録する方法の場合は時間的な制約を伴うため、比較的簡単な議論モデルを用いてフリーテキストにより記録される。自動推論や再利用時の編集などは困難であるが、記録にかかる負担は少ない。一方、議論の終了後に設計理由を記録する方法では、時間的な余裕があるため、表現力に富んだ言語を用いて記録することが可能であり、自動推論や編集等が容易である。しかし、記録に伴う負担は大きくなる。

いずれの記録方法においても、議論内容を記録することが目的ではないという点に問題がある。議論中においては、自分の意見を他者に伝えるとともに他者の意見を検討し、今後のよりよい解決策を考案していく必要がある。また議論後においては、今後の方針等が決定し、個々のシステムエンジニアはそれぞれの作業を進める必要がある。いずれの場合も記録係が専門にしていることは稀であるため、記録を取りそれを管理することは本来の目的とは異なる作業である。したがって、問題解決の事例として議論内容を再利用するためには、それを記録し管理することが動機付けられている必要がある。

2.4 ソフトウェア開発におけるノウハウの蓄積と再利用

ソフトウェア開発を行う組織は一般的に最も低いコストでより品質の高いシステムを開発しようとしている。しかし、ソフトウェア開発を行う組織では過去の経験が組織内に蓄積されていないため、コストパフォーマンスの高いシステムを開発するのは困難な状況である。すなわち、経験したことは忘れ去られ、新しいプロジェクトが立ち上がるたびに過去の過ちを繰り返すこととなる。ソフトウェア開発で得られた生産物やプロセス、そして経験を再利用できるような環境を整えることはソフトウェア開発の質と効率を高める上で重要である [7]。

このような背景に基づいて、ソフトウェア開発で得られる経験を生成し蓄積し再利用するための組織的な基盤環境として Experience Factory が提案されている [8]。この Experience Factory の考えでは、ソフトウェア開発を行う組織を二つに分割する。すなわち、ソフトウェアを実際に開発するプロジェクトの組織と Experience Factory とである。プロジェクトの組織は各プロセスでの生産物を生成することを目的とし、Experience Factory は過去の経験に基づいてソフトウェア開発における実践的な作業を改善する方法を学習することに焦点が置かれる。このように二つの組織は分離されているが、お互いの目的を達成するために相互作用を行う。一方から他方へのフィードバックはある特定の目的に対して十分に明確化された経路を通じて行われる。その様子を図 2.1 に示す [8]。

Experience Factory は、ソフトウェア開発のプロセスや生産物を向上させるためには以下の三つの要素が必要であることを意識している。

- 経験は常に experience packages の中で評価され統合された上で蓄積される
- experience packages の中で評価され統合された経験は experience base に蓄積され、いずれの組織からも参照することができる
- 異なる組織が異なる方法で同じ experience base に蓄積された経験を参照することにより、それぞれの視点が形成される

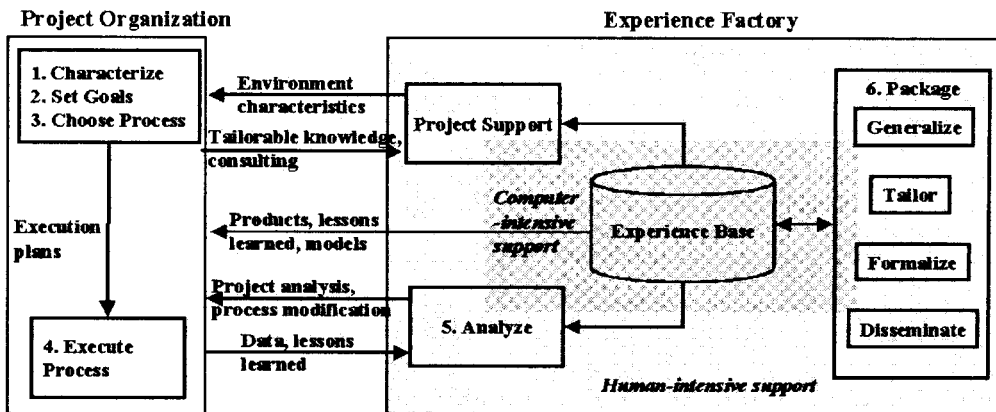


図 2.1: Experience Factory の枠組み (文献 [8] より)

Experience Factory の仕組み (図 2.1) は経験やノウハウを蓄積するために開発プロジェクトとは異なる組織を設置しているという点で本アプローチに類似している。しかし、Experience Factory では何を行うかは提案されているが、実際に経験を収集する方法や経験を蓄積する形態などは具体的に示されていない。経験やノウハウは実際の問題状況に即して体系化された知識や情報として収集され蓄積されていなければ意味がない [54]。そうした知識や情報を効率的に収集し再利用するための方法を検討する必要がある。

2.5 対面での他者への質問による情報獲得の三つの弊害

前節までで論じてきたことをまとめると、必要となる情報に着目してソフトウェア開発の質と効率を向上させるためには、個々の問題状況に即した知識や情報を迅速にシステムエンジニアに提供する必要があり、そのためには他者とのコミュニケーションとやり取りされた情報の再利用とをいかに効率的に支援するかが重要である。

ソフトウェア開発におけるコミュニケーションの形態としては、他のシステムエンジニアに対面で質問することにより問題解決に必要な知識や情報を獲得している場合が多い [36]。個々のシステムエンジニアが置かれている問題状況はそれぞれ異なるため、質問する側とされる側とが置かれている問題状況が異なる。質問される側は他のシステムエンジニアから質問を受けることにより自分の問題解決作業を中断され、質問してきたシステムエンジニアの置かれている問題状況を理解し自分の持っている知識や情報を提供する必要が生じる。このような作業が中断される時間に加えて、中断した作業に戻るためにも時間を要する。中断される前までの自分が置かれていた問題状況や作業状況を思い出し、作業への集中を取り戻す必要があるからである [18]。したがって、他のシステムエンジニアに対面等で直接質問を行うことは、質問した相手の作業効率を著しく低下させることとなる。

他のシステムエンジニアに直接質問を行うことは質問する側の作業効率をも低下させる。他のシステムエンジニアに直接質問を行うためには、まず自分が置かれている問題

状況を解決するために有益な知識や情報を持った人を見つけ出す必要がある。次にその人と会うための調整や電話番号等を調べる必要がある。電話をかけて相手が不在であれば再度電話し直すことが必要となる。このように他のシステムエンジニアへの質問により必要な知識や情報を獲得するためには直接自分の作業とは関係のない作業（オーバーヘッド）を行う必要が生じるため、質問する側の作業効率も低下する [2].

以上に述べたように、他のシステムエンジニアに直接質問することは質問する側にとってもされる側にとっても作業効率が低下するという問題がある。このような作業効率の低下を抑制するためには、一度行われた質問と回答（Q&A）を記録して蓄積しておき、再利用することが有効である [7]. 同じ Q&A が他のシステムエンジニアの間で繰り返されることがなくなり、作業の阻害もある Q&A に関しては一度ですむからである。しかし、他のシステムエンジニアと対面で Q&A を行う場合、それを記録しておくことは困難である [6]. したがって、繰り返し同じ Q&A が別のシステムエンジニア間で行われることとなり、組織におけるコミュニケーション効率の低下、ひいてはソフトウェア開発全体の作業効率の低下につながる [2]. 実際、ソフトウェア開発におけるコミュニケーションと情報共有との関係を調査した結果から、作業中に他者との間で行われたコミュニケーションの内容は明示化され記録されることが少なく、また記録されていたとしても再利用しづらく効率的に情報共有されていないことが判明している [32, 64].

2.6 まとめ

本章では本アプローチに関連する研究として以下の四つのアプローチを取り上げ、いずれのアプローチにおいても実際の問題状況に即した知識や情報を迅速にシステムエンジニアに提供するとともに、そうした問題解決のノウハウや経験を蓄積し再利用することを統合的に支援する具体的かつ効率的な方法が必要とされていることについて論じた。

- (a) ソフトウェア開発に必要な知識や情報の削減
- (b) ソフトウェア開発における議論内容の記録
- (c) ソフトウェア開発におけるコミュニケーションの支援
- (d) ソフトウェア開発におけるノウハウの蓄積と再利用

現状では、問題状況に応じた知識や情報をその場で獲得できることから他のシステムエンジニアに対面で質問する場合が多く、それにより以下の三つの弊害が生じることについて論じた。

第一の弊害: 質問された人は質問した人とは異なる問題状況に置かれているため、他者から質問を受けることにより作業の中断を強制され作業効率は低下する。

第二の弊害: 質問する人も必要な知識や情報を持っていない人を見つけ出し、直接コミュニケーションを行う機会を獲得するための交渉や調整を行うといった、実際の問題解決とは無関係な作業を行う必要が生じるため、作業効率は低下する。

第三の弊害: こうした質問と回答の対（Q&A）は実際の問題解決事例として有益なノウハウであるが、他のシステムエンジニアに直接質問を行った場合にはこうしたノウハウは個人の頭の中に蓄積されるだけで他のシステムエンジニアが再利用できるように明示化され蓄積・共有されることがない。このため同じような Q&A が別のシステムエンジニアらにより繰り返されることとなり、ソフトウェア開発を行う組織全体の作業効率を低下させることとなる。

次章では、他のシステムエンジニアに質問を行わなければならない理由について論じるとともに、他者へ質問することが問題解決においては重要であることについて論じる。そして、問題状況に応じた知識や情報を迅速にシステムエンジニアに提供するとともに、それらを効率的に再利用することを統合的に支援するためには、他者への質問による弊害を解消しながら他者への質問を支援する必要があることについて論じる。

第3章 問題解決としてのソフトウェア開発における他者とのQ&Aによる効果

本章では、2.5節で論じたような三つの弊害を引き起こす他者への質問がなぜ行われるのかについて認知科学的な観点から論じるとともに、他者への質問による問題解決における三つの効果について論じる。

ソフトウェア開発に必要な膨大な知識や情報はドキュメントとして記録されておりシステムエンジニアは随時参照することが可能である。しかし、これらのドキュメントは個々の実際に生じた問題状況に応じて体系化されているわけではなく [26]、個々の問題状況に応じてどのような知識や情報をどのように利用するかというメタな知識や情報は個々のシステムエンジニアの経験やノウハウとして分散して存在するケースが多い [65]。このため必要な知識や情報を獲得するにあたっては、そうしたドキュメントを検索するのではなく、他のシステムエンジニアに対面で質問を行わざるを得ないのが現状である [36]。

このようなメタな知識や情報ではない知識や情報が必要な場合も多い。そのような場合にも膨大なドキュメントから必要な知識や情報だけを迅速に獲得することは困難であることについて論じる。問題解決に必要な知識や情報を自分の頭の外にある情報源から獲得するためには、どのような知識や情報が必要かを明確にする必要がある。必要な知識や情報を特定するためには自分が置かれている問題状況を明確にする必要がある。しかし、問題状況を明確にするにあたっては、問題状況の非列挙性 [76]、暗黙知の存在 [58]、外部に存在する知識や情報の認識 [24] などの問題がある。これらの問題が存在することにより、自分の持っていない知識や情報を獲得するためには他者に対面で質問せざるを得ないことについて論じる。

他者に質問することは、これらの問題を解消する上で効果があり、問題解決に必要な知識や情報を迅速に獲得することができるという効果があることについて論じる。また、問題解決においては問題状況をより明確化することがその質と効率を向上させるために重要であり [61, 48]、他者からの回答を獲得することや他者へ質問することにより問題状況をより明確化できるという効果があることについて論じる。すなわち、他者への質問による問題解決における効果として以下の三つであり、それぞれの効果について論じる。

第一の効果: 問題状況に即した知識や情報の迅速な獲得

第二の効果: 問題状況に即した意識していなかった知識や情報を他者からの回答として獲得することによる問題状況の再認識

第三の効果: 質問として明示化した問題状況を客観視することによる問題状況の再認識

3.1 問題解決に必要な情報を迅速に獲得することの困難さ

問題解決に必要な知識や情報が自分の頭の中になく、外界に存在する知識や情報を利用しなければならぬ場合、まず自分の置かれている問題状況を明らかにし、どのような知識や情報が必要なかを明確化する必要がある [72]。しかし、問題状況を明確化することは、問題状況の非列挙性 [76]、暗黙知の存在 [58]、外部に存在する知識や情報に対する認識のギャップ [24] という三つの理由から困難である。

問題状況の非列挙性: 人は問題解決における一般的な知識をルールやノウハウとして記述できる一方で、個々の様々な問題状況に対して柔軟に対応することができる [74]。つまり、人は個別の問題状況に応じてルールやノウハウと矛盾するような問題解決を人は行うことができる。このようなルールやノウハウと矛盾するような問題解決を行った理由をその人に問うことにより、ルールやノウハウに例外条件として追加することができる。このような例外条件は常に発生し、問題状況を何らかの表現によって明示化することにより必ず明示化されない部分が残される [76]。したがって、問題の条件を完全に数え上げることは不可能であり、個々の問題条件に必要な知識や情報を完全に特定することも不可能である [21, 48]。

暗黙知の存在: 問題状況を把握し解決策を構築する際に利用された知識の一部は暗黙的なものである [48]。なぜなら、人は話すことのできる以上のことを知っているからである [58]。自分の知っていることを正確に説明することはできず、何かを知っていることにすら気付かない場合すらある [68]。したがって、問題状況のすべてを明確化し明示化することは不可能であり、そのような暗黙的に知っている問題状況に対して必要となる知識や情報を明確化することも不可能である。

外界の情報に対する認識のギャップ: 問題状況を明確化するためには、その問題状況に関する知識や情報が必要である。このような知識や情報は問題状況が明確化され解決策が構築されなければ特定することができない [21, 48]。たとえ問題状況の明確化に必要な知識や情報が明確化されたとしても、それらが自分の頭に存在しない場合はそれらを獲得することは困難である。自分の頭の中以外に存在する知識や情報に対する人の認識には、図 3.1 に示すような四つのレベルが存在するからである [24]。

- D1:** 熟知しており、容易に入手することが可能な定常的に利用している外部に存在する知識や情報の領域。
- D2:** どこにあるかは知っているが、その内容については曖昧にしか記憶していない使用頻度の低い外部に存在する知識の領域。
- D3:** 外部に存在すると思いこんでいるが、どこにどのような知識や情報があるかは把握していない知識や情報の領域。
- D4:** 実際に外部に存在する知識や情報の領域。

これらの外部の知識に対する認識のレベルをプログラミングで利用する関数ライブラリを例にとって説明する。まず、よく使っている関数で、リファレンス等を参照しなくてもその関数がどのような機能を持ち、どのような条件で利用できるかを熟知しているような関数は D1 に該当する。また、時々使う関数で、そのライブラリの中に登録されていることは知っているが、実際にその関数を使う場合には、その関数がどのような機能を持ち、どのような条件で利用できるかに関し

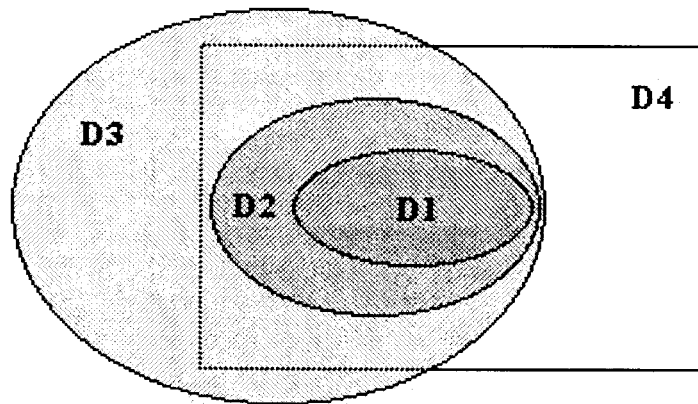


図 3.1: 外部に存在する情報に対する認識 (文献 [24] より)

てはリファレンス等を参照する必要があるような関数は D2 に該当する。また、このような関数は一般的に検索機能などを用いて見つけ出す必要がある。一方、よく利用している関数や時々利用する関数などから類推して、実際に利用したことはないがライブラリの中に登録されていると思っている関数は D3 に該当する。そして実際のライブラリ自体が D4 である。ここで問題となるのは D3 の領域と D4 の領域のずれである [23]。D3 に含まれているが D4 には含まれていない領域の知識とは、外部に存在すると思いつているが実際には存在しない知識であり、どんなに検索をしても入手することが不可能である。このような知識としては、記録されることのないシステムエンジニアの経験や直感といった暗黙知が挙げられる。一方、D4 に含まれるが D3 には含まれない知識とは、システムエンジニアが置かれている問題状況を解決するために有益な知識が実際に存在するにもかかわらず、システムエンジニアは存在しないと思いつている知識であり、このような知識は検索により見つけ出させる努力さえされないため、獲得することは困難である。

以上の理由から、自分の置かれている問題状況を明確化し必要な知識や情報を特定することは困難であり、例えそれらを特定できたとしても、実際にそうした知識や情報を外界から見つけ出し利用することは困難である。このような認知的な原因により、システムエンジニアが参照することのできる膨大なドキュメントから必要な知識や情報を見つけ出すことには大変な労力が必要である。このため、他のシステムエンジニアに対面で質問することにより必要な知識や情報を獲得していると考えられる。次節では他者に対面で質問することがこうした認知的な負荷をどのように解消できるのか、また問題解決にどのような効果があるのかについて論じる。

3.2 他者への質問による問題状況に即した情報の迅速な獲得

問題状況をより明確化する方法として分散認知の利用が効果的である [65]。分散認知とは自分の頭の中にある以外の知識を指し、他者が持つ知識やデータベースに蓄積され

ている知識、ある問題解決を行うための知識の具現化としての道具などを指す [89]. こうした分散認知を利用することにより、問題状況に関する様々な視点や知識を獲得することが可能となるとともに、個人に要求される知識や情報の量が削減される [65].

問題状況に即した知識や情報を獲得するためには、分散認知の中でも他者の存在が重要であることが指摘されている [56, 75]. 実際の問題状況において自分の知識や認知的スキルに基づいて他者と相互作用を行いながら、他者からのフィードバックにより自分の知識や認知的スキルを内化することによって有益な知識は獲得されるからである [19]. つまり、他者と問題状況を共有しつつ、かつその問題状況に対する理解や視点が異なることにより、問題状況に関する様々な知識や観点を獲得することができ、問題状況をより明確化することが可能となる [45]. このような効果により問題状況の非列挙性や問題状況に対する暗黙知の存在という問題を解消することが可能となる.

他者が持っている知識や情報を利用する場合には以下のような利点がある.

- 相手の行動や状況からの推測などにより、人は問題状況の暗黙的な部分も理解することができる [53]. したがって、他者の置かれている問題状況をコンピュータよりも正確に共有することができる.
- 相手の認識している問題状況と自分の認識している問題状況に食い違いがある場合でも、相手の言動からの推測により問題状況に対する認識を修正し、より正確に相手の問題状況を共有することができる [17].

他者とのインタラクションによるこれらの利点により、問題状況に対する共通の理解が迅速に形成されるため、質問された側は相手の問題状況に即して自分の持っている知識や情報を迅速に提供することができる [71].

一方、自分が把握した問題状況やそれに対する解決策を他者に伝えることにより、その問題状況や解決策に関して他者が知っている知識や情報を得ることができ、逆に他者が知らない知識や情報は何かを把握することが可能となる. 問題状況を共有していることから、他者も独自の問題状況に対する認識や解決策を持っているため、自分が意識していなかった問題状況の側面について他者が知識や情報を持っているれば、他者との相互作用を通じてそうした知識を得ることができる. これにより相手が持っていると思っていなかった知識や情報を獲得することが可能となる. したがって、D4 理論における D3 と D4 の領域における知識や情報の相違、すなわち外界に存在すると思いでいた知識や情報が実際にはない場合や、外界に存在することを意識していなかった知識や情報が実際には存在する場合における知識や情報の獲得における問題が解消される.

3.3 他者への質問による問題状況の明確化における二つの効果

問題解決の質と効率を向上させるためには問題状況をより明確化することが重要である [61]. 実際、要求分析から運用保守までのプロセスを順次実行し後戻りしないウォーターフォール型のソフトウェア開発手法では、間違いを修正するためにかかるコストはその間違いの発見がより下流のプロセスになるほど指数関数的に増加する傾向があると指摘されている [66]. つまり、最初の要求分析、問題解決では問題状況の明確化をより正確に行うことがソフトウェア開発の質と効率を向上させる上で重要である.

このように問題状況を迅速にできる限り明確化することは問題解決の質と効率を向上させることにつながる. しかし、”Problem framing and solving cannot be separated.(文献 [48], pp.2)”であるため、解決策を構築する前に問題状況を完全に明確化することは不

可能である [72]. したがって, "Starting with a vague incomplete problem requirement, they sketch out a partial solution. By seeing the partial solution, designers identify portions of the problem that have not yet been understood, gain an understanding of the problem and then refine the solution.(文献 [48], pp.13 より)".

実際, ソフトウェア開発ではウォーターフォール型の欠点を解消するために, プロトタイプ手法やスパイラルモデルなどが採り入れられている. これらの開発手法では顧客の主要な要求だけを満たす機能を持ったプロトタイプシステムを開発し, それを顧客に見せることにより新たな要求を抽出し, 徐々に顧客の要求との乖離の少ないシステムを開発していくという手法である [10]. つまり, 問題状況の明確化と解決策の構築とを繰り返すことにより問題状況をより明確にしていくという手法である.

他者への質問にもこのような問題状況の明確化と解決策との構築を繰り返すことにより問題状況がより明確化される効果がある. すなわち, 他者からの回答による問題状況の再認識と, 質問として問題状況を明示化することによる問題状況の再認識とである. 本節ではこれら二つの効果について論じる.

3.3.1 回答による問題状況の再認識

明示化されたある表現によって意味するものは, 人やその対象が置かれている状況により異なる [35]. つまり, ある問題状況を明示化した表現としての質問に対して, その質問を行った人とその質問を受けた人とはその質問によって意味される問題状況に対する理解が異なる.

したがって, 他者に質問することにより回答として獲得した知識や情報は, 置かれている問題状況に対する自分の理解とは異なる観点からのものである. このような自分とは異なる観点からの知識や情報を獲得することができることにより, 問題状況を様々な観点から捉えることが可能となり, 問題状況をより正確に把握することが可能となる [46].

このような効果を図示すると図 3.2 のようになる. 質問する側は自分が他者への質問として形式知化することのできた問題状況に対応するだけの知識や情報が他者からの回答として獲得できると期待する. しかし, 回答する側は形式知化された質問の解釈が質問する側とは異なるため, 実際に回答に含まれる知識や情報は問題状況に関するものであるが, 質問する側が期待していたものとは異なる. このような他者の異なる観点からの問題状況に関する知識や情報を獲得することにより, 自分が問題状況に関してどこまで把握していて何を把握していなかったか (自己の認知状態の監視), および, 自分の構築した解決策が実際の問題状況に十分に適合しなかった場合に他者の異なる観点を採用入れてどのように新たに解決策を構築するか (自己の認知状態の制御) といった能力を向上させることが可能となる [12]. そして, 自分一人では意識しなかった問題状況に関する知識 (領域知識) を習得することも可能となる [19] ため, 自分の置かれている問題状況を別の観点から再認識することが可能となり, より問題状況を明確化することが可能となる

3.3.2 質問による問題状況の再認識

我々が文章を理解する場合, "...we have initial anticipations and expectations of what the meaning of the whole sentence will be and interpret each word accordingly. Based on the interpretations of each word, we refine the anticipations of the meaning of the whole. The redefined whole functions in a dialectical fashion to refine and redefine the

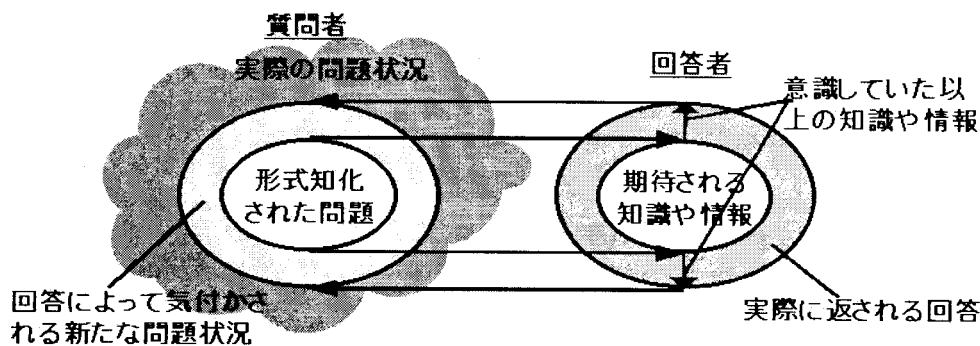


図 3.2: 他者からの回答による問題状況の再認識

parts.(文献 [48], pp.13 より)”, というように部分と全体を行きつ戻りつしながら行っている [73].

モノを作る場合にも同じようなやり方が採られる. すなわち, まず”Starting with a vague incomplete problem requirement, they sketch out a partial solution.(文献 [48], pp.13 より)”. このような行動を起こすことにより, ”By seeing the partial solution, designers identify portions of the problem that have not yet been understood, gain an understanding of the problem, and then refine the solution.(文献 [48], pp.13 より)”といった問題状況の明確化が可能になる. このような問題解決の特徴は”reflective conversation [67]”と呼ばれる.

このような方法はソフトウェア開発でも取り入れられている. すなわち, 問題状況の明確化 (要求分析), 解決策の構築 (設計, コーディング, 試験), 解決策の問題状況へのフィードバック (顧客レビュー) という一連のプロセスを短期間に繰り返し実行するという開発手法である [9]. このような開発手法の利点は, 顧客の要求と実際に構築されたシステムとの乖離を最小限に抑えることができるということである. すなわち, 問題解決の質を向上させることが可能となる. さらに, 個々の開発プロセスにおける議論等の内容を記録し, 次回の議論で参照できるようにすることにより, 議論が不足している部分の把握が容易になりソフトウェア開発の効率が向上する [77]. このようなサイクルによる効果は問題状況を他者への質問として明示化し表現することによっても得られる. 他者に質問を行うためには, 自分の置かれている問題状況を明示化し客観的に認識することのできる対象として言語等を用いて表現する必要がある. このように外的に表現されたものを認識することにより, 明示化していなかった問題状況の部分に気付くことができる. このようなプロセスを繰り返すことにより, 問題状況について自分が把握していることをできる限り明示化し表現することが可能となる. このようなサイクルを図 3.3 に示す [50].

3.4 まとめ

本章では, 問題状況を迅速にできる限り明確化することが問題解決の質と効率を向上させるために重要であるが, それが非常に困難であること, および他者への質問によりその困難さを解消することが可能であること, について論じた.

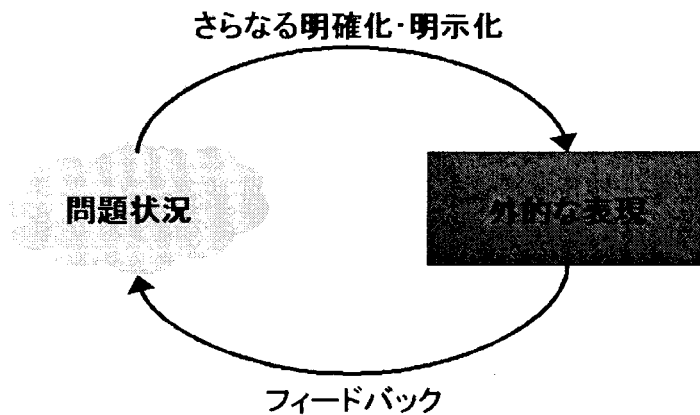


図 3.3: 質問による問題状況の再認識 (文献 [50] より)

問題状況の明確化の重要性については、それが解決策の構築と密接に依存し合っているため、問題状況の明確化の迅速性と質とが問題解決の迅速性と質と結び付くことを論じた。

問題状況の明確化が困難である理由については、問題状況を明確化し定義すれば必ず明確化・定義できない部分が残されること、問題状況に関して表現できないことも知っているため完全に問題状況を明確化することは不可能であること、問題状況を明確化するために必要な自分の頭の中にはない知識や情報を獲得することは困難であること、という観点から論じた。

他者と問題状況を共有し協調して問題解決を行うことにより、問題状況を多角的な観点から捉えることができるため、問題状況の明確化に有効であることについて論じた。他者に質問することには以下の三つの効果があることについて論じた。これらの効果はいずれも問題解決の質と効率を向上させるために有効である。

- (1) 問題状況に即した知識や情報の迅速な獲得
- (2) 問題状況に即した意識していなかった知識や情報の獲得による問題状況の再認識
- (3) 質問として明示化した問題状況を客観視することによる問題状況の再認識

次章では、これらの効果を維持しつつ、2.5 節で述べた他のシステムエンジニアに質問することの弊害を回避するための枠組みについて論じるとともに、この枠組みにより上述した三つの効果に加えて、質問と回答 (Q&A) の再利用性の向上という第四の効果が期待できることについても論じる。

第4章 システムエンジニアの問題解決を支援する回答作業の専門化と電子メールの利用によるQ&A方式に基づく情報提供の枠組み

本章では、3.4節で論じた他者への質問による問題解決における三つの効果を維持しつつ、2.5節で論じた他者への質問による三つの弊害を解消するための枠組みを提案する。3.4節で論じた他者への質問による三つの効果とは以下の三つである。

第一の効果: 問題状況に即した知識や情報の迅速な獲得

第二の効果: 問題状況に即した意識していなかった知識や情報の獲得による問題状況の再認識

第三の効果: 質問として明示化した問題状況を客観視することによる問題状況の再認識

2.5節で論じた他者への質問による三つの弊害とは以下のものである

第一の弊害: 作業状況の相違による回答者の作業効率の低下

第二の弊害: 作業オーバーヘッドの増加による質問者の作業効率の低下

第三の弊害: Q&Aの再利用が困難であることによる組織全体のコミュニケーション効率の低下

第三の弊害を主に解消するために、グループウェアの一種である電子メールの効果について論じる。電子メールが非同期分散型のグループウェア [87] であることにより、第一や第二の弊害を緩和する効果もあることについて論じる。

電子メールには二つの問題があることが一般的に指摘されている。意図を伝えることが対面などに比べて困難であること、および、相手からの迅速な反応を得ることが困難であること、である [6]。電子メールが対面に比べて意図を伝えることが困難であるという欠点は、他者への質問による三つの効果を損なう。問題状況を伝達できなければ問題状況に即した知識や情報を得ることはできないからである。相手から迅速な反応を得られないことは、他者への質問による第一の効果である問題状況に即した知識や情報を迅速に獲得できるという効果を損なう。

電子メールでは問題状況を伝達することが困難であるという欠点について、ソフトウェア開発におけるほとんどの作業がコンピュータ上で行われていることから、そこで発生した問題の状況を伝達するには対面よりも電子メールの方が有効であることについて論じる。相手からの迅速な反応を得ることが困難であるという欠点を解消するために、回答作業をソフトウェア開発プロジェクトから分離し、回答作業を専門に行う組織を設置することの効果について論じる。このような組織の設置により、他者への質問による第一、第二の弊害も解消することができることについて論じる。

電子メールの利用と回答作業の専門化という二つの仕組みを他者との Q&A のやり取りに適用することにより、3.4 節で論じた三つの効果に加えて、Q&A の再利用を促進することによる組織全体のコミュニケーション効率の向上という効果が得られる。したがって、このような枠組みにより期待できる効果は以下の四つである。

第一の効果: 問題状況に即した知識や情報の迅速な獲得

第二の効果: 問題状況に即した意識していなかった知識や情報の獲得による問題状況の再認識

第三の効果: 質問として明示化した問題状況を客観視することによる問題状況の再認識

第四の効果: Q&A/の再利用の促進による組織全体のコミュニケーション効率の向上
それぞれの効果について論じる。

4.1 他者との Q&A における電子メールの効果

本節では、他者との Q&A のやり取りに電子メールを用いることによる効果について、Q&A の再利用性の向上、作業状況の相違による回答者側の作業効率の低下や質問者の作業オーバーヘッドの緩和といった観点から論じる。

電子メールは、利用者に時間や場所の制約を課さない非同期分散型のグループウェアであり [34, 87]、利用する人の負荷や役割が対称的であるため [30]、最も広く利用されているグループウェアの一つである [30, 41]。電子メールが非同期分散型という特徴を持つことにより、以下の三つの効果が得られる [1, 93]。

- 電子メールが送付された相手は都合のよい時に返答を行うことができる
- 電子メールを送付する際に相手がどこにいるかを考慮する必要がない
- コミュニケーションの内容が明示化され記録されるため再利用ができる

電子メールによる第一の効果により、質問された側は自分の置かれている問題状況や作業状況に合わせて作業の中断等による作業効率の低下が最も少ない時に回答を行うことができる。したがって、他者への質問による第一の弊害を緩和することが可能である。

電子メールによる第二の効果により、必要な知識や情報を他者から獲得するために質問する相手がいる場所や時間を考慮する必要がなくなる。したがって、他者への質問による第二の弊害を緩和することが可能である。

電子メールによる第三の効果により、Q&A の内容を随時確認することができるとともに、問題解決事例として別の他者に提供することが可能となる。記録された Q&A をデータベースに蓄積し組織内で共有することが可能となることにより、同じような Q&A のやり取りが繰り返し組織内で行われることが抑制される。

電子メールを利用する人の負荷や役割が対称的であるということは、電子メールを送る側もそれに対して回答する側も、テキストを記述する負荷や相手がどこにいるかを考

慮する必要がないことなどは同じであり、また、回答する側にとっても Q&A を記録しておくことで別の他者から同じような質問を受けた場合に再利用することが可能となる。

電子メールを利用することは、他者との Q&A による三つの弊害を解消もしくは緩和するばかりではない。他者への質問による第二や第三の効果を高める上でも有効である。

電子メールにより他者に質問や回答を行うためには、自分の置かれている問題状況や相手の問題状況に関して知っている知識や情報をテキスト等により記述する必要がある。記述されたテキストは、対面や電話等で話された言葉よりもより客観的に認識することができる [17, 43]。自分の記述した回答をより客観視できることにより、記述し忘れた知識や情報に気付くことが可能となり、質問した人は問題状況に即したより多くの知識や情報を回答者から獲得することが可能となる。自分の記述した質問をより客観視できることにより、問題状況について記述し忘れた事柄に気付くことが可能となり、より正確に問題状況を相手に伝えることが可能となる。

このような効果は Q&A の再利用性を高める上でも有益である。電子メールを用いて Q&A を行うことにより、問題状況を定義する情報がより多く明示化され記録されるとともに、その問題状況を解決するために有益な知識や情報がより多く明示化され記録される。こうした問題状況に関するより多くの知識や情報は、他者がそれを理解する場合により正確に元の意味を汲み取ることを可能とするからである [38]。

4.2 他者との Q&A における電子メールの問題

Daft と Lengel によれば、対面や電話といった他者との直接的なコミュニケーション手段の利点は以下の二つである [17]。

- 身振りや声の調子など相手に意図を伝達する手段が多様であること
- 相手から即座に反応が得られること

これらの利点により、対面や電話等でのコミュニケーションでは、コミュニケーションの豊かさを定義する以下二つの可能性がともに高くなる [17, 93]。

- 話者が持っている知識や情報に対する理解がある時間内で変化する可能性
- ある時間内で話者が持っている知識や情報に対する理解を変化させるために、知識や情報の相互関係や曖昧さを明確にするような情報が多く得られる可能性

問題状況に関連した情報が多く得られることにより話者の持っている知識や情報の不確実さは減少し、それによって持っている知識や情報に対する理解が変化することにより、それらの知識や情報の曖昧さ、すなわち多義性が減少する [6]。対面や電話といった直接的なコミュニケーション手段では、より正確に問題状況を相手に伝達することができるとともに、その問題状況に即した知識や情報を迅速に獲得することができる。

電子メールを用いた場合、前節で述べたように一度に相手に伝えられる情報の質や量は高くなる。しかし単位時間で見た場合には、前述したコミュニケーションの豊かさを定義する二つの可能性は低くなる [6, 93]。電子メールでは主にテキストベースで意図が伝えられ、相手からの反応を即座に得ることは困難なためである。電子メールの送り手が持っている知識や情報に対する理解が変化するまでに要する時間が長く、知識や情報の相互関係や曖昧さを明確にするような相互作用も少ないため [17]、問題状況が不確実もしくは曖昧なまま相手に理解され、問題状況に対する理解にずれが生じる可能性が高い [6, 93]。問題状況に対する共通の理解が質問する側とされる側との間で確立され

ないため、回答が得られても、どの質問に対する回答なのかを理解することが困難になる [6].

電子メールを用いて質問する側とされる側との間で問題状況に対する共通の理解を確立するためには何度かお互いの記述内容に関するやり取りを行う必要がある。しかし、電子メールでは相手から即座に反応が得られることは保証されないため、コミュニケーション密度は低くなり、お互いに必要な知識や情報を獲得できるまでに時間がかかる。質問する側にとっては相手からの回答が得られるまでの間は問題解決が行えず、作業が停滞することとなる。回答する側は作業の中断が断続的に生じるため、相手の問題状況を思い出すまでに時間がかかり、自分の作業へ集中することが妨げられる。

電子メールを用いて効率的に他者との Q&A を行うためには、質問する側とされる側との間で問題状況に関する共通の理解を迅速に確立すること、および質問する側が迅速に回答を獲得できること、を支援する必要がある。

4.3 問題状況の共有における電子メールの効果

本節では、電子メールの欠点である問題状況に関する共通の理解を迅速に確立することの困難さについて、ソフトウェア開発ではむしろ電子メールを用いる方が相手に問題状況を正確かつ迅速に伝達することが可能であり、相手からも正確かつ迅速に問題状況により即した知識や情報を迅速に獲得することが可能であることについて論じる。

コストパフォーマンスの優れたワークステーションやパーソナルコンピュータの普及と高速ネットワークの発展により、システムエンジニアは一人一台以上のワークステーションやパーソナルコンピュータを利用できるようになってきている [90, 82]。これらのコンピュータ端末上では、ソフトウェア開発における様々な作業を支援する CASE (Computer Aided Software Engineering) ツールをはじめ、GUI (Graphical User Interface) やグループウェアなどが動作しており、ソフトウェア開発におけるほとんどの作業はこうしたアプリケーションを利用してコンピュータ端末上で実行される [85]。

ソフトウェア開発においてシステムエンジニアが解決しなければならない問題のほとんどはこのようなコンピュータ端末上で生じたものである。その問題状況もコンピュータ端末上に存在する。これらの問題状況はコンピュータ端末上に存在するために、スクリーンショットやプログラムコードのカット&ペーストなどにより電子メールに容易に添付することが可能である。このような方法により、電子メールでは言語的な表現に加えて問題状況そのものを視覚的に相手に伝達することができる。プログラムコードなどは実際に回答する側のコンピュータ端末で試験を行うことが可能になることにより、言語でエラー状況を相手に伝達するよりもより正確に伝達することが可能となる。回答する側もスクリーンショットに解決策を示したり、修正したプログラムコードをカット&ペーストしたりすることにより、より正確かつ迅速に解決策を質問者に伝達することが可能となる。

このようにコンピュータ端末上でほとんどの作業を実行するソフトウェア開発においては、相手のところに行って対面で質問したり電話で質問したりするよりも、電子メールを用いる方がより正確かつ迅速に問題状況を相手に伝達することで、相手から問題状況に即した知識や情報をより正確かつ迅速に獲得することができる。対面や電話などのように相手の作業状況や相手の時間や場所を考慮する必要もない。さらに、スクリーンショットやプログラムコードといった問題状況そのものが電子メールに添付され蓄積されることにより、同じような問題状況において必要な知識や情報を求めている他のシステムエンジニアがその電子メールを見ても問題状況が同じものかどうかを判断しやすい。つまり、電子メールによる Q&A は再利用性が高い。

4.4 回答作業の専門化による効果

本節では、電子メールを用いる場合の二つ目の欠点である、質問する側が迅速に回答を得られる保証がないことを解消するための方法として、回答作業を開発プロジェクトから分離し回答作業を専門に行う組織を設置するというアプローチの効果について論じる。

4.1 節で論じたように、電子メールを用いることにより、質問された側は自分の作業状況に合わせて回答を行うことができ、質問する人は相手がどこにいるかどのような状況を考慮することなく質問することができる。これにより他者への質問による第二と第三の弊害を緩和することができる。

質問された側が自分の作業状況に合わせて回答を行うことができるということは、質問した側は相手からいつ回答が得られるか分からないということを意味する。回答が得られるまでの間は質問した人は問題解決が進まないため作業効率は低下する。

質問する人は誰から適切な回答を得られそうかを知っていれば、相手の居場所や時間を気にすることなく質問を行うことができるため、必要な知識や情報を獲得するためオーバーヘッドは緩和される。誰に質問すればよいか分からない場合には質問する相手を見つけなければならないためオーバーヘッドはそれほど緩和されない。

回答する側も、回答を行うためには自分の作業状況とは異なる問題状況を理解し、知識や情報を提供するという労力は電子メールを用いる場合でも変わることはない。そのために自分の作業とは別に回答を作成しなければならないため、回答する側の全体的な作業時間は延びてしまい、結果として作業効率は低下することとなる。

このように、電子メールを用いた場合でも、システムエンジニア間で Q&A のやり取りを行う限り、作業効率が低下することを回避することはできない。ある作業を行う上で現在の組織体制においてコスト的に明らかなデメリットがある場合、その作業を分離し、その作業を専門に行う組織を新たに設置もしくはアウトソーシングするという対処方法は有効である [57, 55]。そこで、システムエンジニア間での Q&A のやり取りによる作業効率の低下を抑制するために、回答を専門に行う組織を設置することは有効であると考えられる。

この組織に属する人々は回答を行うことが仕事となるため、質問を行うシステムエンジニアは相手の作業状況を考慮する必要がなく、質問する相手が他の作業を行っている場合よりも迅速に問題状況に即した知識や情報を回答として獲得することが可能となる。また、誰に質問を送付すればよいかも明確になるため、必要な知識や情報を持っていそうな人を探すという、必要な知識や情報を得る上でのオーバーヘッドも減少する。

さらに、一人の人が様々なソフトウェア開発プロセスからの質問に回答を行うことを経験することにより、ある問題状況に対して様々な観点から分析を行う能力が習得され、質問を行ったシステムエンジニアは自分の意識していなかった問題状況の側面に関する知らなかった知識や情報を回答において獲得する可能性が高くなる。このような知識や情報は問題状況をより明確化するのに有効であり、問題解決の質の向上につながる。

4.5 まとめ:本アプローチにより期待できる四つの効果

本章では、2.3 節で論じた他者への質問による三つの弊害を解消し、3.4 節で論じた他者への質問による三つの効果を維持もしくは高めるための仕組みとして、電子メールを用いることによる効果と、電子メールの欠点を補うための回答作業を専門化するというアプローチの効果とについて論じた。

電子メールを用いて他者との Q&A のやり取りを行うことにより Q&A の再利用が可能となるため、他者への質問による三つ目の弊害を解消できることについて論じた。電子メールを用いることにより問題状況に即した知識や情報をより多く記録することができるため、再利用性がさらに向上することについても論じた。

電子メールを用いた Q&A では、相手と問題状況に関する共通の理解を確立することが困難であり、相手から迅速に回答を獲得することも困難であることについて、対面や電話等のコミュニケーションの利点と欠点とを比較することにより論じた。

相手との問題状況に関する共通の理解を確立することの困難さについて、ほとんど作業がコンピュータ端末上で行われるソフトウェア開発においては、対面などよりもむしろ電子メールの方が相手に正確かつ迅速に問題状況を伝達することができ、回答する側も質問者に正確かつ迅速に問題状況に即した知識や情報を理解してもらうことが可能であることについて論じた。

相手から迅速に回答を獲得することの困難さを解消するために、回答作業を開発プロジェクトから分離し回答作業を専門に行う組織を設置することの効果について論じた。非同期分散型で利用者の役割や負荷が対象的なグループウェアである電子メールを利用することによる、(1) 相手の作業状況の考慮、(2) 必要な情報を獲得するまでのオーバーヘッドの削減、(3) 記述されることによる Q&A の再利用性の向上、という三つの効果により、2.5 節で論じた他者への質問による三つの弊害がいずれも緩和されることについて論じた。また、他者への質問による第二と第三の効果が期待できる一方で、他者への質問による第一の効果である問題状況に即した知識や情報の迅速な獲得という効果が減じられること、および相手との問題状況の共有が困難であるという二つの問題があることについて論じた。必要な知識や情報の迅速な獲得のために回答作業を専門化することが効果的であり、さらに他者への質問による三つの弊害が解消され、Q&A の管理が集中的に行われるため再利用のためのシステムエンジニアの負荷が削減され Q&A の再利用性が向上することについて論じた。相手との問題状況の共有が困難であることに対しては、ほとんどの作業がコンピュータ上で行われているソフトウェア開発においてはスクリーンショットやプログラムコードの電子メールへの添付といったコンピュータツールにしかできない機能のおかげで対面よりもむしろ相手との問題状況の共有が行いやすいこと、およびこのような実際の問題状況そのものが蓄積されることにより再利用性が向上するという効果があることについて論じた。したがって、本枠組みにより期待できる効果は以下の四つである。

第一の効果: 問題状況に即した知識や情報の迅速な獲得

第二の効果: 問題状況に即した意識していなかった知識や情報の獲得による問題状況の再認識

第三の効果: 質問として明示化した問題状況を客観視することによる問題状況の再認識

第四の効果: 再利用される時に他者が理解するのに十分な情報が Q&A として記録されることによる再利用性の向上

次章では、株式会社エヌ・ティ・ティ・データで開発・運用している、回答作業の専門化と電子メールを利用することにより Q&A を支援する仕組み、「技術支援システム」の概要を説明し、上述した四つの効果を検証するための枠組みについて論じ、それぞれの効果を検証するために行った調査内容と結果について論じる。

第5章 事例に基づく効果の検証

前章では、他者との Q&A のやり取りを効率的に行うための枠組みとして、電子メールの利用と回答作業の専門化を相互補完的に採り入れることの四つの効果について論じた。本章では、このような枠組みを実際のシステムエンジニアの問題解決支援に導入した事例として、株式会社エヌ・ティ・ティ・データで開発・運用している「技術支援システム」の仕組みを説明する。この技術支援システムを対象に 4.5 節でまとめた四つの効果を検証するための枠組み、および個々の検証内容について論じる。

技術支援システムは、システムエンジニアからの電子メールによる質問に対して回答を形成する技術支援チームと呼ばれる組織と、形成された Q&A を蓄積し検索を通じてシステムエンジニアに再利用してもらうためのデータベースとから構成されている。技術支援チームおよびデータベースにどのような情報がどのような時点で入出力されるかについて論じる。

質問もしくは再利用を通じて技術支援システムを利用したシステムエンジニア、およびデータベースに蓄積された Q&A を対象に、4 章で論じた以下の四つの効果について検証を行う。

第一の効果：問題状況に即した知識や情報の迅速な獲得

第二の効果：問題状況に即した意識していなかった知識や情報の獲得による問題状況の再認識

第三の効果：質問として明示化した問題状況を客観視することによる問題状況の再認識

第四の効果：再利用される時に他者が理解するのに十分な情報が Q&A として記録されることによる再利用性の向上

第一から第三までの効果については、Q&A が形成される時に技術支援チームに質問を行ったシステムエンジニアを対象としたアンケート調査、およびアンケート調査に基づいた Q&A の分析を通じて検証した。第四の効果については、再利用性を問題解決に有益な知識や情報が迅速に獲得できたかどうかという観点から捉え、検索を行ったシステムエンジニアに対するアンケート調査を通じて検証した。どのような調査がどのような時点で行われたかについて論じる。

個々の効果の検証を行うにあたり、効果を検証するための仮説、仮説の検証方法、検証結果のそれぞれについて論じる。

5.1 検証対象となる事例

本節では、回答作業を専門化・組織化し、Q&A によりシステムエンジニアが問題状況に即した知識や情報を迅速に獲得することを支援する仕組みとして、株式会社エヌ・ティ・ティ・データで開発・運用しているシステムエンジニアのための技術支援システム（以下、技術支援システムと呼ぶ）の枠組みについて論じる。

技術支援システムでは、システムエンジニアからの電子メールによる質問に対して、技術分野ごとのエキスパートが回答を作成して電子メールで質問したシステムエンジニアに送付し、形成された Q&A を技術支援システムのデータベースに蓄積し、ネットワークを通じて社内のシステムエンジニアが蓄積された Q&A を検索により再利用できる仕組みを開発・運用している [88].

この技術支援システムにおいて、システムエンジニアからの質問により回答が形成され、蓄積された質問と回答が検索を通じて他のシステムエンジニアに再利用されるまでの流れは以下ようになる (図 5.1 参照).

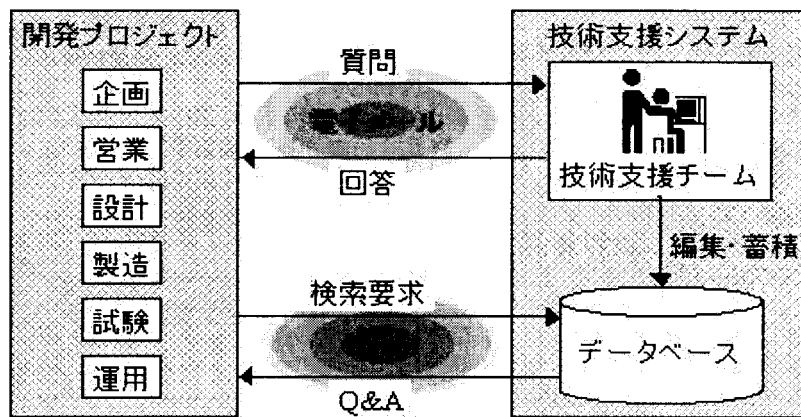


図 5.1: 技術支援システムの仕組み

1. システムエンジニアから電子メールにより問い合わせのあった質問に対して、ネットワークやデータベースといった各ドメインの専門家約 20 名から構成される「技術支援チーム」が回答を電子メールで送付する。個々の質問に対し、技術支援チームではチーム内の各メンバーの専門性や能力等に応じて回答作成担当者が割り振られる。担当者は、まず回答案をチーム内のメーリングリストに投稿し他のメンバーからのコメントを反映して回答の質を高めた上で、質問したシステムエンジニアに対して回答を電子メールで送付する。
2. 上記 1. の手順で形成された Q&A は、技術支援チームにおいて編集され、質問内容ごとに (1)「トラブルシューティング依頼」(あるトラブルを解決する方法を知りたい)、(2)「実現機能調査依頼」(ある機能を実現するために考案した方法が妥当かどうかを知りたい)、(3)「製品等紹介依頼」(ある機能を持った製品等を知りたい)、(4)「比較評価依頼」(ある機能に関してどの製品等が優れているかを知りたい)、(5)「その他」、のいずれかに分類されデータベースに蓄積される。
3. Q&A がある程度蓄積された時点から、このデータベースはシステムエンジニアに対して公開されている。何らかの Q&A を必要とするシステムエンジニアは、まず Web を介してこの技術支援システムのデータベースに対してキーワードを用いた全文検索を行うことによって必要な Q&A の再利用を試みる。必要とする Q&A を検索によって得られなかった場合は、技術支援チームに電子メールによる質問を行う。この場合、上記 1. と同じ手順で回答が作成され、質問と対になった Q&A が上記 2. と同じ手順で技術支援システムのデータベースに蓄積される。

表 5.1: 技術支援システムの特徴

運用実績	約 6 年間
Q&A 登録件数	約 13,000 件
質問数	約 200~300 件/月
参照数	約 7,000~8,000 件/月
総単語数	約 2,000 万語
ドメインターム数	約 600~700 万語

このような枠組みを備えた技術支援システムに蓄積されている Q&A は表 5.1 に示すような特徴を備えている。Q&A の登録数が約 13,000 件であることから、述べ約 13,000 人のシステムエンジニアが技術支援チームに質問を行ってきており、Q&A の形成に関する第 1 と第 2 の効果の検証はこの約 13,000 人のシステムエンジニアを対象としたものである。また、参照数が毎月述べ約 7,000~8,000 件であることから、毎月述べ約 7,000 人~8,000 人のシステムエンジニアが技術支援システムに蓄積されている Q&A を検索し参照していることになる。再利用に関する第 4 の効果における再利用時の Q&A の有益性に関する調査ではこれらのシステムエンジニアを対象としている。

実際の Q&A の例を図 5.2 に示す。電子メールを用いて技術支援チームとシステムエンジニアとの間でやり取りされた Q&A は図 5.2 に示すように一つのドキュメントとして編集されて蓄積されている。表 5.1 におけるドメインタームとはこのような Q&A に含まれるソフトウェア開発に特有の専門用語、すなわち情報通信関連用語のことである。ドメインタームの例を図 5.3 に示す。また、技術支援チームへの質問もしくは Q&A の検索により技術支援システムを利用したシステムエンジニアを対象に行ったアンケート調査は図 5.4 のように、質問により回答を獲得した場合も Q&A の検索を行った場合も Web ページとして表示された Q&A の下部にアンケート入力項目が組み込まれており、該当する項目を選択しアンケート送信ボタンが押されることにより調査回答を収集している。図 5.4 は検索結果に対する評価を行う場合の入力画面例である。図 5.4 では、Q&A に添付資料が貼り付けられていることもわかる。

5.2 検証のための枠組み

本節では、技術支援システムにおける他者との Q&A のやり取りによる四つの効果を検証するための枠組みについて論じる。四つの効果を検証するための枠組みをまとめると図 5.5 となる。

第一の効果である、問題状況に即した知識や情報が迅速に得られるかどうかを検証するにあたり、質問する際にシステムエンジニアに回答期限を設定してもらうとともに、実際に回答がシステムエンジニアに送付された日時ログを取得し、その差を計測した。

第二の効果である、他者からの回答により、意識していなかった問題状況に関する知らなかった知識や情報が得られるかどうかを検証するために、質問により技術支援チームから回答を得たシステムエンジニアに対して、得た回答をどう認識しているかに関してアンケート調査を行った。回答に対する意識として、(1) 期待以上の知識や情報が得られた、(2) 期待通りの知識や情報が得られた、(3) 期待以下の知識や情報しか得ら

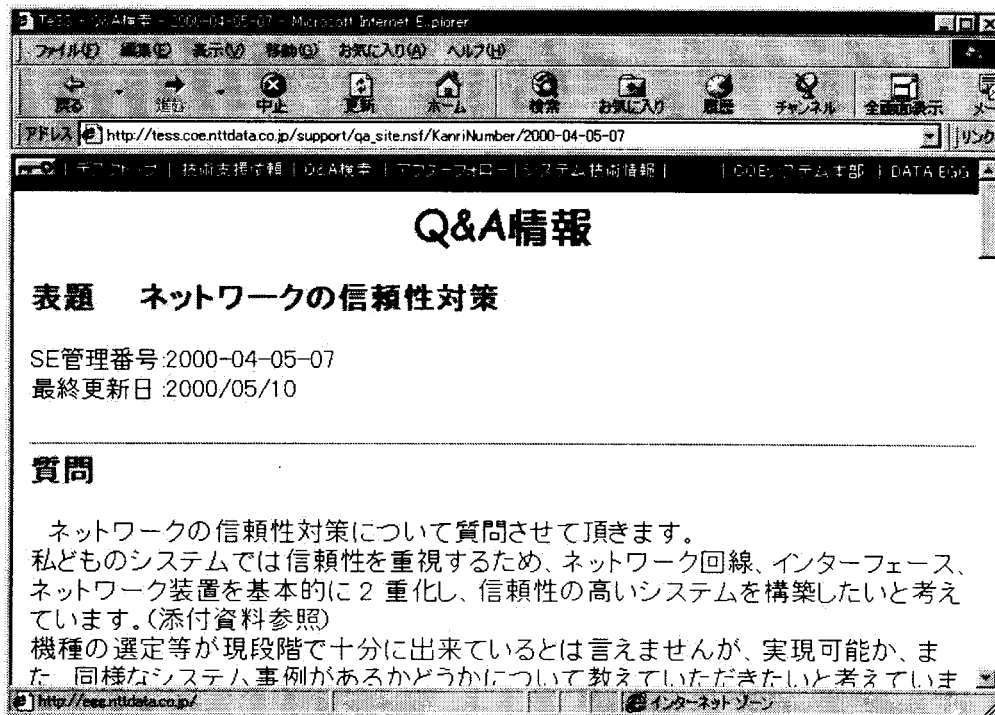


図 5.2: 調査対象となる Q&A の例

Telnet, 自動化, ネットワーク監視装置, ログ,
通信機器, スクリプト, DXP / DS, 自動運転...

図 5.3: Q&A に含まれるドメインタームの例

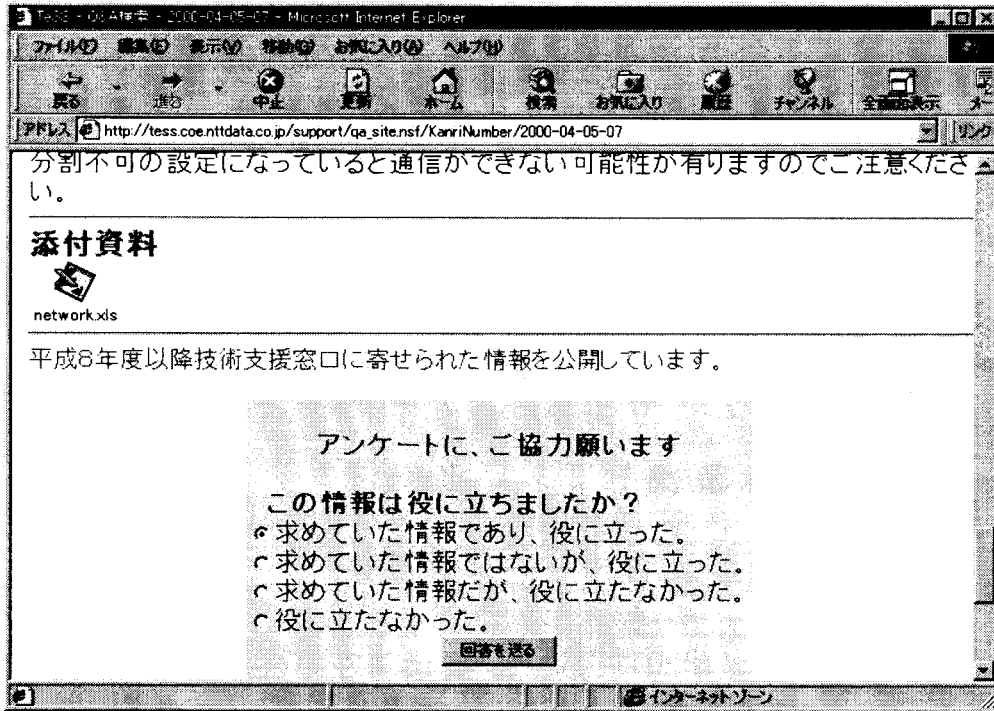


図 5.4: アンケートの例

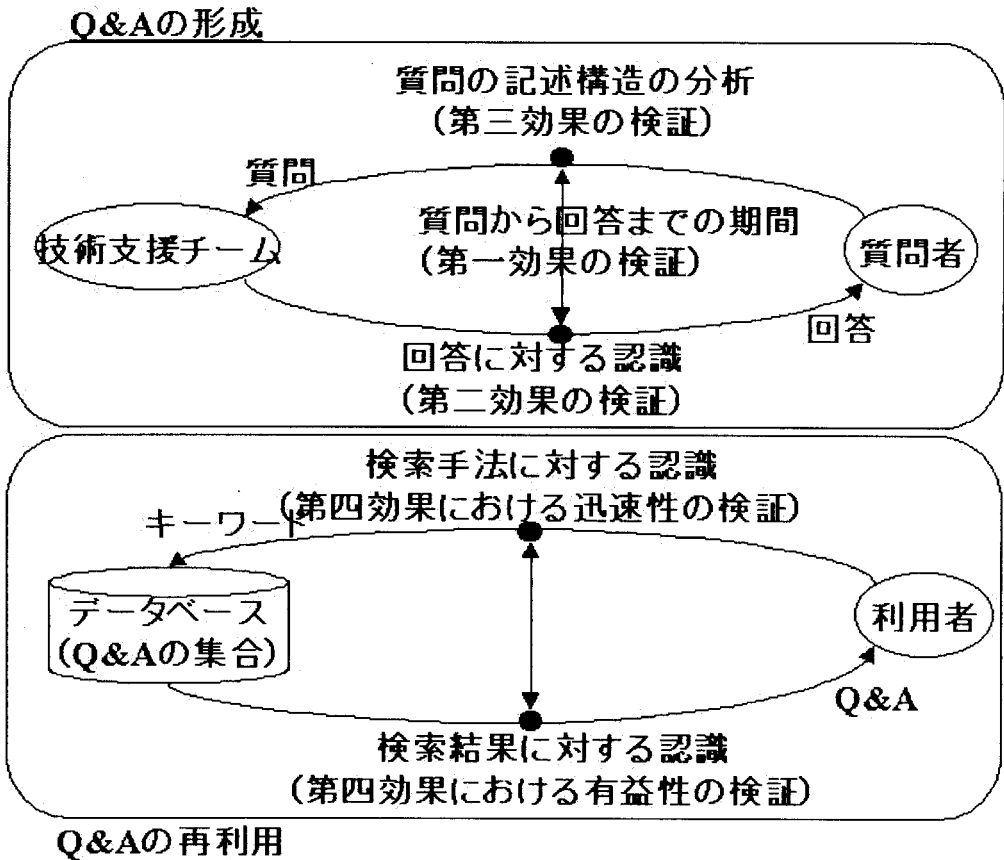


図 5.5: 効果検証のための枠組み

れなかった、の三通りの判断基準を設定した。期待以上の知識や情報が獲得できた場合が多いほど、第二の効果が得られていると判断する。また、第三の効果もあったと判断する。回答は一人のエキスパートが担当して行うが、作成した回答は質問を行ったシステムエンジニアに送付する前に、技術支援チームの約20名のエキスパート全員がチェックを行っているため、質問に即していない回答は基本的に作成されないと考えられ、回答の良し悪しはその質問の良し悪しに依存すると考えるためである。

第三の効果である、他者への質問により問題状況が明確になったかどうかを検証するにあたり、質問により満足できる回答を得るための質問の特徴について分析を行う。回答の良し悪しはその質問の良し悪しに依存すると考えられるが、期待以上の回答を獲得することのできた質問と期待以下の回答しか獲得できなかった質問との間に相違がなければ、期待以上の回答が獲得できる場合が多くとも、質問を記述することにより問題状況が明確化されているとは判断できないからである。

技術支援システムに蓄積された Q&A は、実際的な問題状況に即した問題解決に必要な知識や情報の事例である。こうした問題解決に有益な事例を迅速に再利用できることにより、類似した問題状況において迅速かつ適切に問題解決を行うことが可能となる。そこで、第四の効果を検証するにあたり以下の二つの観点から調査を行った。

- (1) 検索結果として得られた Q&A が問題解決に有益かどうか
- (2) 問題解決に有益な Q&A を検索により迅速に獲得することができているかどうか

一つ目の調査は蓄積された Q&A が再利用においても有効かどうかを検証するためのものであり、二つ目の調査はシステムエンジニアからの質問に対して直接回答を行う場合とそれらを間接的に再利用する場合の両方において効果のある知識や情報の提供を検討するためのものである。

5.3 第一の効果の検証：問題状況に即した情報の迅速な獲得の検証

本節では、技術支援システムによりどれだけ迅速に問題状況に即した知識や情報をシステムエンジニアが獲得できているかを検証するための方法と検証結果について論じる。

まず、問題状況に即した知識や情報を獲得する迅速性はシステムエンジニアが属しているプロジェクトの状況に依存すること、したがって、システムエンジニアが判断した期限内に必要な知識や情報を質問したシステムエンジニアが得られるかどうかを迅速性の判断基準となることについて論じる。

次に、このような迅速性を検証するために、システムエンジニアが質問を行う際に回答期限を設定してもらうとともに、実際に回答がシステムエンジニアに送付された日時に関するログを取得し、その差分を検証するという方法について論じる。

そして、その検証結果について、回答に対する三通りの認識、すなわち「期待以上」「期待通り」「期待以下」のそれぞれの場合が、回答期限と回答が送付された日時との差分にどのような影響があるかという観点から論じる。

5.3.1 検証のための仮説

問題状況に即した知識や情報を技術支援チームからどれだけ迅速に質問を行ったシステムエンジニアが得られるかを考えた場合、その迅速性はシステムエンジニアが置かれ

ている問題状況に依存する。したがって、質問を行ったシステムエンジニアがいつまでに問題解決に必要な知識や情報が得られればよいと考えているかに基づいて、実際に技術支援チームの回答の迅速性を検証する必要がある。

そこで、質問を行うシステムエンジニアがいつまでに回答を得たいかという基準に基づき、実際に回答が送付された日時との差分を検証する。つまり、システムエンジニアが回答を欲しい期限内に実際に回答が送付されているかどうかを検証する。期限内に回答が得られた比率が高いほど、技術支援システムを利用することにより質問を行ったシステムエンジニアが満足できる迅速さにおいて回答が得られたことを意味する。

5.3.2 仮説の検証方法

システムエンジニアが技術チームの Web ページから質問を入力する際に、「回答期限」という入力欄を設ける。回答期限には、1日とか10日というように回答が得られるまでの希望の日数を入力してもらう。一方、質問が入力された日時と、実際に回答が質問を行ったシステムエンジニアに送付された日時に関するログを取得する。

質問が入力された日時から実際に回答が送付された日時までの、土曜日と日曜日および祝日などの日数を除いた値を日単位で計算し、回答期限として設定された日数と比較した。

また、回答期限内に回答が得られたとしても、その回答が問題解決に必要な知識や情報を含んでいなければ意味がない。そこで、回答期限内および回答期限外の回答のそれぞれについて、個々の回答が質問を行ったシステムエンジニアにどのように認識されているかの比率を調査した。

対象としたデータは、平成9年1月1日から平成12年3月末までの8377件のQ&Aとした。この理由は、平成8年度以降は一年間にほぼ25,000件程度のQ&Aが蓄積されており、技術支援チームが回答作業に対して慣れてきている時期と考えられるからである。

5.3.3 検証結果

8377件のQ&Aのうち、回答期限が入力されていない場合が18件あった。この18件を除く8359件について、回答期限内および回答期限外の比率を図5.6に示す。この図から約80%のシステムエンジニアが回答期限内に回答を得ることができていることが分かる。

一方、回答期限内および回答期限外の回答に対して、質問を行ったシステムエンジニアがどのような評価を行っているかという比率を調査した。対象となるデータは平成9年1月1日から平成12年3月末までに回答に対する評価が得られた995件のQ&Aであり、これら995件のQ&Aはすべて回答期限が入力されていた。これら995件の回答がどのように認識されているかという比率を、回答期限内のものと回答期限外のものとで比較した結果が図5.7である(5.4節を参照)。

図5.7において、回答に対する認識の差異と回答期限に対する超過という二つの指標間に関連性があるかどうかを検定する。帰無仮説はこの二つの指標が独立であるというもので、相関が5%以下の場合にこの帰無仮説を棄却する。まず、帰無仮説に基づいて現在得られている数値の期待値を算出する必要がある。しかし、これらの数値を理論的に算出することは不可能である。これら二つの指標間の数値的な関係に関する理論的根拠がないからである。そこで、同じ調査を何度繰り返しても995件のデータが取得され、期限内の場合が746件で期限外の場合が249件、期待以上の場合が511件で期待通りが382件で期待以下が102件という比率であると仮定する。この仮定に基づくそれぞれの

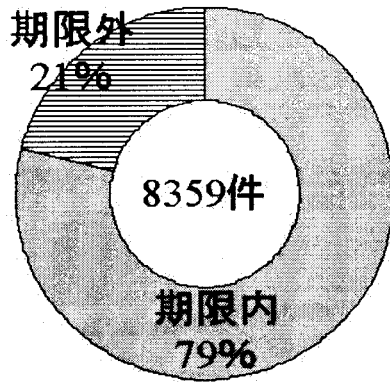


図 5.6: 回答期限に関する調査結果

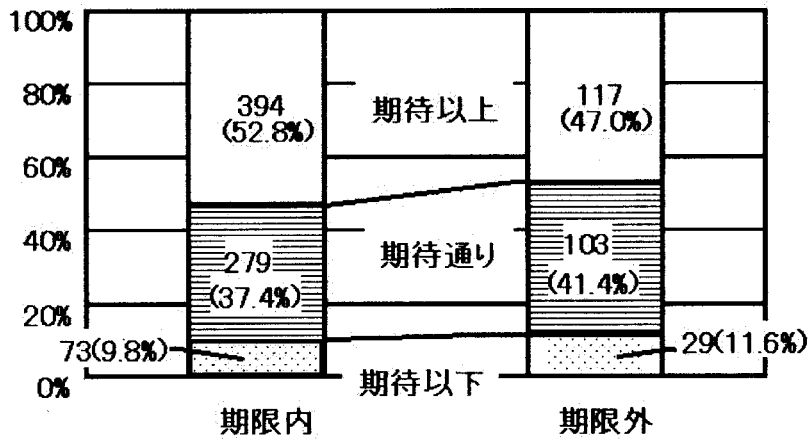


図 5.7: 回答期限と回答に対する認識との関係

期待値は表 5.2 に示す通りである。すなわち、期待以上の場合は 995 件中 511 件を占めることから、期待以上でかつ期限内の期待値は、

$$746 * 511/995 \simeq 383$$

となる。他の期待値も同様に計算される。

表 5.2: 回答に対する評価と回答期限との関連性 1

	期限内	期限外	合計
期待以上	394 ($746 * \frac{511}{995} \simeq 383$)	117 ($249 * \frac{511}{995} \simeq 128$)	511 ($\frac{511}{995} \simeq 51\%$)
期待通り	279 ($746 * \frac{382}{995} \simeq 286$)	103 ($249 * \frac{382}{995} \simeq 96$)	382 ($\frac{382}{995} \simeq 39\%$)
期待以下	73 ($746 * \frac{102}{995} \simeq 77$)	29 ($249 * \frac{102}{995} \simeq 25$)	102 ($\frac{102}{995} = 10\%$)
合計	746	249	995

これらの期待値に基づいて χ^2 検定を行うと、

$$\chi^2 = \frac{(394-383)^2}{383} + \frac{(117-128)^2}{128} + \frac{(279-286)^2}{286} + \frac{(103-96)^2}{96} + \frac{(73-77)^2}{77} + \frac{(29-25)^2}{25} \simeq 3.01 (\geq 5.99)$$

となる。回答に対する認識が三通り、回答期限が二通りであるから、この場合の自由度は $(3-1) * (2-1) = 2$ である。自由度 2 の χ^2 分布における有意差 5% に相当する値は 5.99 であり、この値よりも大きければ回答に対する認識と質問の分類とは関連性があると判断できる。計算した χ^2 の値は 5.99 を下回り (22.2%)、帰無仮説は採択される。すなわち、回答に対する認識と回答期限という二つの指標間の相関関係はないと判断される。

この理由はシステムエンジニアが設定する回答期限はシステムエンジニアが置かれている問題状況に依存するためであると考えられる。すなわち、回答期限はシステムエンジニアが置かれている問題状況の以下の二つの要因に依存すると考える。

緊急性: システムエンジニアが所属するプロジェクトのスケジュールに基づいて、いつまでに自分の抱えている問題を解決しなければいけないのかが決定される。その問題解決の期限により、回答が必要な期限が決定される。

複雑性: 質問内容がどれだけ複雑でどれだけ調査が必要なのかにより、回答が得られるまでにどの程度の日数がかかるかが判断される。

この二つのバランスにより、質問を行うシステムエンジニアは回答期限を設定すると考えられる。したがって、例えば期待以上の回答が得られた場合でも、緊急性が高かったり、質問が複雑であった場合には回答期限を超過してしまう場合が増える。そこで、回答期限から 5 日 (土日を除く 1 週間) まで超過した回答を期限内として、対象となる

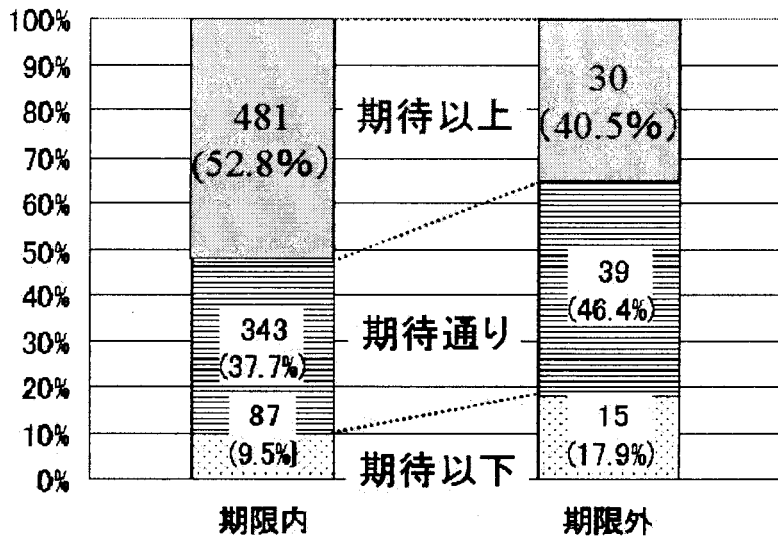


図 5.8: 期限超過 5 日までを期限内とした場合の回答期限と回答に対する認識との関係

995 件の回答がどのように認識されているかという比率を、回答期限内のものと回答期限外のものとして比較した結果が図 5.8 である。

図 5.8 において、回答に対する認識の差異と回答期限に対する超過という二つの指標間に関連性があるかどうかを検定する。先ほどと同様に帰無仮説はこの二つの指標が独立であるというもので、相関が 5% 以下の場合にこの帰無仮説を棄却する。それぞれの期待値は表 5.3 に示す通りである。

これらの期待値に基づいて χ^2 検定を行うと、

$$\chi^2 = \frac{(481-468)^2}{468} + \frac{(30-43)^2}{43} + \frac{(343-350)^2}{350} + \frac{(39-32)^2}{32} + \frac{(87-93)^2}{93} + \frac{(15-9)^2}{9} \approx 10.35$$

となる。回答に対する認識が三通り、回答期限が二通りであるから、この場合の自由度は $(3-1) * (2-1) = 2$ である。自由度 2 の χ^2 分布における有意差 5% に相当する値は 5.99 であり、この値よりも大きければ回答に対する認識と質問の分類とは関連性があると判断できる。計算した χ^2 の値は 5.99 を上回り (0.6%)、帰無仮説は棄却される。すなわち、回答に対する認識と回答期限という二つの指標間に相関関係があると判断される。

表 5.3: 回答に対する評価と回答期限との関連性 2

	期限内	期限外	合計
期待以上	481 ($911 * \frac{511}{995} \approx 468$)	30 ($84 * \frac{511}{995} \approx 43$)	511 ($\frac{511}{995} \approx 51\%$)
期待通り	343 ($911 * \frac{382}{995} \approx 350$)	39 ($84 * \frac{382}{995} \approx 32$)	382 ($\frac{382}{995} \approx 38\%$)
期待以下	87 ($911 * \frac{102}{995} \approx 93$)	15 ($84 * \frac{102}{995} \approx 9$)	102 ($\frac{102}{995} \approx 10\%$)
合計	911	84	995

5.4 第二の効果の検証：回答による問題状況の再認識の検証

本節では、他者に質問することにより意識していなかった問題状況に関する知らなかった知識や情報が得られるかどうかという、本アプローチにより期待できる二つ目の効果を検証するために、質問を行ったシステムエンジニアがそれによって得た回答に対してどのような認識をしているかを分類し、検証する上での判断の仮説について論じる。次に実際にアンケート調査を行った方法について論じる。そして、アンケート調査の結果について論じる。

5.4.1 検証のための仮説

アンケートを実施するために、回答に対して質問したシステムエンジニアがどのように認識するかを、意識していた知識や情報が回答に含まれていたかどうかという観点から、以下の三つに分類した。

期待以上：システムエンジニアが質問に対して意識していた以上の知識や情報が回答に含まれ、かつそうした知識や情報が問題解決に有益であった場合を指す。

期待通り：システムエンジニアが質問に対して意識していた通りの知識や情報が回答の中でほぼ過不足なく含まれ、そうした知識や情報が問題解決に有益であった場合を指す。

期待以下：システムエンジニアが質問に対して意識していた知識や情報よりも少ない知識や情報しか回答の中に含まれておらず、問題解決に不充分であった場合、および、意識していたものとは異なる知識や情報が回答に含まれ、そうした知識や情報が問題解決に役に立たなかった場合を指す。すなわち、意識していた知識や情報であるか否かに関わらず、回答が問題解決に十分に役に立たなかった場合である。

システムエンジニアにアンケートを行う際に、回答に対する認識が（１）期待以上、（２）期待通り、（３）期待以下、の三つの選択肢を与えた場合、システムエンジニアのこの三つの選択肢の意味に対する理解は、前述した内容の通りになると考える。なぜなら、この三つの選択肢の中で負の意味を持つのは「期待以下」だけであり、「期待以上」と「期待通り」の差を考えた場合、「期待通り」は問題解決に役に立つ知識や情報が満足いく程度に得られた場合と判断し、期待以上という言葉には予想していた以上というニュアンスがあることから、予想していた以上の知識や情報が得られ、かつそれらの知識や情報が問題解決に役に立った場合であると判断されると考えられる。すなわち、回答に対する認識が「期待以上」とは、システムエンジニアが意識していなかった問題状況の側面に関する知識や情報が回答に含まれ、かつそうした知識や情報がその問題状況を解決するために役に立つと認識された場合である。

したがって、この三つの選択肢により質問を行ったシステムエンジニアにそれによって得た回答に対する認識を調査した場合の検証の仮説は、回答に対して「期待以上」と認識するシステムエンジニアが多ければ多いほど、技術支援システムがソフトウェア開発における問題解決の質を高める上で有効であると判断する。

5.4.2 仮説の検証方法

アンケートは技術支援チームの Web ページに用意した。そして、質問をしたシステムエンジニアに電子メールで回答を送付する際に、記述した回答の下部にそのアンケートページへのリンクとアンケートへの回答を促す旨の記述を自動的に付加するようにした。

このような方法をとることにより、ヒアリングやアンケート用紙への記述といった調査方法に比べてコストがかからず、また、アンケートへ回答する負担も少なくなる。したがって、アンケートの回収率を上げることが期待できる。また、アンケートに回答することを強制しないことにより、アンケートに回答する意志を持ったシステムエンジニアだけがアンケートに答えることになり、アンケートに対する回答の精度を向上させることが期待できる。アンケートに回答する意志を持つ場合とは、有益な知識や情報が得られ、その感謝を伝えようとする場合か、もしくは、有益な知識や情報が得られず、その不満を伝えることで今後の改善を望む場合であると考えられる。これにより、「期待通り」という中間的な回答が減少することが期待できる。しかし、「期待以下」の回答しか得られず、かつ、今後の改善を諦めてしまったシステムエンジニアからはアンケートに対する回答を得られない可能性が高くなる。

対象としたデータは、平成 9 年 1 月 1 日から平成 12 年 3 月末までの 8377 件の Q&A とした。この理由は、平成 8 年度以降は一年間にほぼ 25,000 件程度の Q&A が蓄積されており、技術支援チームが回答作業に対して慣れてきている時期と考えられるからである。

5.4.3 検証結果

平成 9 年 1 月 1 日から平成 12 年 3 月末までに回答に対する評価が得られた件数は 995 であった。その内訳は、「期待以上」511 件、「期待通り」382 件、「期待以下」102 件であった（図 5.9 を参照）。

「期待以下」が極端に低い理由は二つ考えられる。一つは 5.1.2 で述べたように、「期待以下」の回答しか得られなかったが、技術支援システムに対する今後の改善を諦めてしまったシステムエンジニアからはアンケートに対する回答を得られない可能性が高いからである。もう一つの理由は、実際に回答の質が高く、そもそも「期待以下」の回答

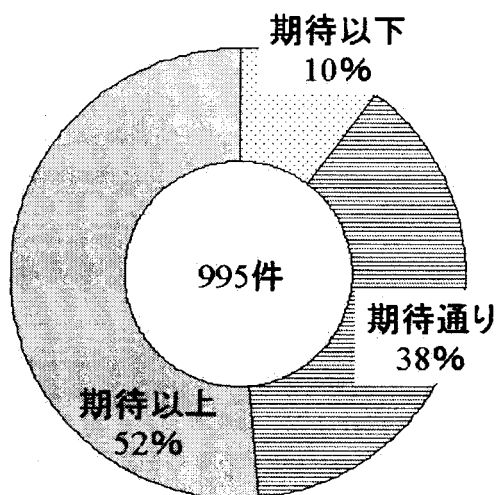


図 5.9: 回答に対する認識に関する調査結果

が形成されている比率が低いということである。この根拠は、ある質問に対して回答を作成する担当者は一人であるが、他の約 20 名のメンバが質問と回答とを審査しているため、質問に対する誤った認識や不適切な回答等は基本的には生じないと考えられるからである。「期待以上」が少ないことに対してこのような理解に基づくと、「期待以上」は「期待以下」の約 5 倍であり、技術支援チームに質問を行うことにより、システムエンジニアは他者に質問することにより意識していなかった問題状況に関する知らなかった知識や情報を得ることが十分に支援されていると判断する。

5.5 第三の効果の検証：質問の記述構造に関する調査

本節では、前節で述べた調査結果に基づいて、システムエンジニアが技術支援チームに質問を行うことにより、自分の置かれている問題状況を明確にすることができるかどうかという三つ目の効果を検証するための方法と結果について論じる。

まず、質問において問題状況が明確になっているかどうかを判断する基準として、質問を行うことにより得られた回答に対して質問を行ったシステムエンジニアがどのように認識したかという前節の調査結果を用いることの意味について論じる。

どのような質問が問題状況を明確にしている、すなわち、質問により他者から問題状況に即した知らない知識や情報を得るために有効であるかを調査するための方法について論じる。そのような質問とは質問内容に応じた記述構造とその構成部分における情報通信関連用語（ドメインターム）の比率に特徴があるという観察結果が得られたことについて論じる。

5.5.1 検証のための仮説

3.3節で述べたように、質問によって問題状況が明確になるという効果は、明らかになった問題状況を他者に伝えることにより、他者からその問題状況に即した知識や情報を得るために必要とされる。そして、問題状況に即した知識や情報の中でも、意識していなかった問題状況に関する知らない知識や情報が他者から得られることが、問題状況のさらなる明確化、ひいては問題解決の質を高める上で重要である。

したがって、5.3節で分類した回答に対する三つの認識のうち、「期待以上」と評価された回答の比率が質問により問題状況が明確化されているかどうかの判断基準となる。

このような考えに基づいた場合、5.3.3節の調査結果から、約50%のシステムエンジニアが「期待以上」の回答、すなわち、意識していなかった問題状況に関する知らなかった知識や情報を獲得している。すなわち、約50%のシステムエンジニアが技術支援チームに質問を行うことにより問題状況を効果的に明確化することができている。しかし、「期待以上」と認識された回答と「期待以下」と認識された回答とが形成される元となった質問の記述の仕方に差異がなければ、「期待以上」の回答を約50%のシステムエンジニアが獲得しているからといって、その質問が問題状況を明確にしているとは判断できない。もし、「期待以上」と認識された回答と「期待以下」と認識された回答とが形成される元となった質問の記述の仕方に差異がないならば、それは回答の仕方に問題があることを意味する。

そこで、「期待以上」および「期待以下」と認識された各回答と対になった質問が、技術支援システムのデータベースに蓄積される際に技術支援チームによってどのように分類（5.1節を参照）されたかを図5.10に示す。

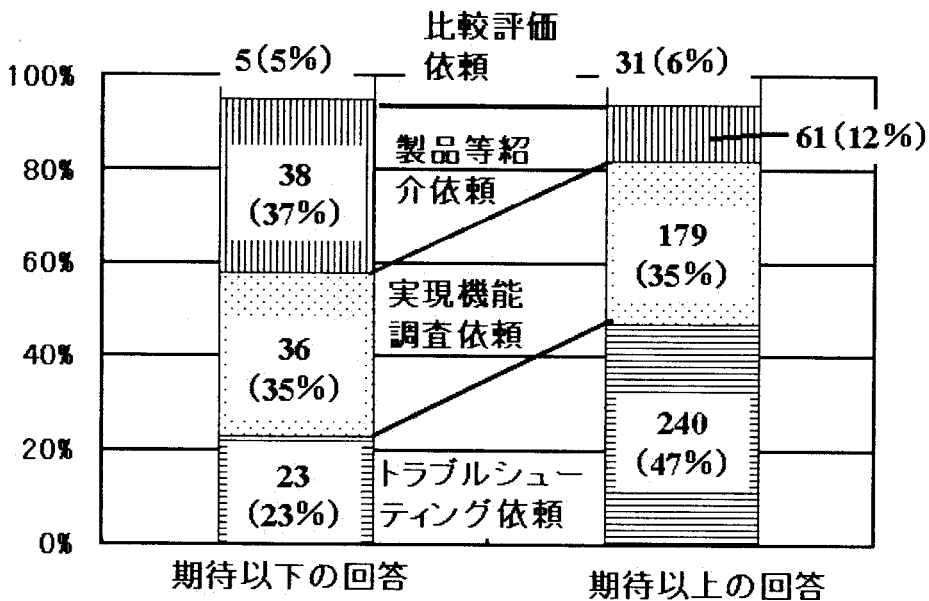


図 5.10: 回答に対する認識と Q&A の分類との関係

図 5.10 において、回答に対する認識の差異と質問の分類とが関連するかどうかを検定する。一つの質問に対して回答を作成する担当者は一人であるが、他の約 20 名のメンバーが質問と回答とを審査することを考えれば、回答の質のレベルはどのような質問の内

容であってもほぼ均一であると考えられ、期待以上の場合と期待以下の場合とにおける質問の分布は同じになると考えられる。そこで、期待以上と期待以下とが合わせて 613 である標本を複数回にわたって収集した場合に、今回と同様、期待以上が 511、トラブルシューティング依頼が 263、実現機能調査依頼が 215、製品等紹介依頼が 99、比較評価依頼が 36 であるとした場合のそれぞれの期待値は表 5.4 に示す通りである。すなわち、トラブルシューティングは 613 件中 263 件を占めることから、トラブルシューティング依頼でかつ期待以上の場合の期待値は、

$$511 * 263 / 613 = 511 * 0.43 = 219$$

となる。他の期待値も同様に計算される。

表 5.4: 回答に対する評価と質問分類との関連性

	期待以上	期待以下	合計
トラブルシューティング	240 ($511 * \frac{263}{613} \approx 219$)	23 ($102 * \frac{263}{613} \approx 44$)	263 ($\frac{263}{613} \approx 43\%$)
実現機能調査	179 ($511 * \frac{215}{613} \approx 179$)	36 ($102 * \frac{215}{613} \approx 36$)	215 ($\frac{215}{613} \approx 35\%$)
製品等紹介	61 ($511 * \frac{99}{613} \approx 53$)	38 ($102 * \frac{99}{613} \approx 16$)	99 ($\frac{99}{613} \approx 16\%$)
比較評価	31 ($511 * \frac{36}{613} \approx 30$)	5 ($102 * \frac{36}{613} \approx 6$)	36 ($\frac{36}{613} \approx 6\%$)
合計	511	102	613

したがって、回答に対する認識と質問の分類とに関する χ^2 検定を行うと、

$$\chi^2 = \frac{(240-219)^2}{219} + \frac{(23-44)^2}{44} + \frac{(179-179)^2}{179} + \frac{(36-35)^2}{35} + \frac{(61-53)^2}{53} + \frac{(38-16)^2}{16} + \frac{(31-30)^2}{30} + \frac{(5-6)^2}{6} \approx 43.69 (\geq 7.81)$$

となる。回答に対する認識が期待以上か期待以下かの二通り、質問の分類が四通りであるから、この場合の自由度は $(2-1) * (4-1) = 3$ である。自由度 3 の χ^2 分布における有意差 5% に相当する値は 7.81 であり、この値よりも大きければ回答に対する認識と質問の分類とは関連性があると判断できる。計算した χ^2 の値は 7.81 を大幅に上回ることから、回答に対する認識と質問の分類とは強い関連性があると判断できる。すなわち、回答に対する認識が異なれば質問の分類も異なる傾向が強いと判断できる。

したがって、トラブルシューティング依頼に関する質問は、「期待以上」の回答を得る確率が高く、逆に、製品等紹介依頼に関する質問では「期待以下」の回答しか得られない確率が高い傾向があると判断できる。すなわち、質問の内容によってそれに対する回答の認識に差異が生じていることとなり、質問の記述の差異により、それにより得られる回答に対する認識に差異が生じていることを裏付けていると判断できる。

5.5.2 仮説の検証方法

「期待以上」と認識された回答と「期待以下」と認識された回答とが形成される元となった質問の記述の仕方の差異を検証する。

まず、質問の内容ごとに「期待以上」の回答が形成される元となった質問と「期待以下」の回答が形成される元となった質問の構造に差異が見られるかどうかを調査した。まず質問の内容ごとに「期待以上」の回答が形成される元となった質問と「期待以下」の回答が形成される元となった質問の構造を手作業により分析した結果、いずれの場合の質問も共通して表 5.3 のような構造が抽出された。すなわち、質問を構成する要素として「システム要求事項」「機能要求事項」「機能実現方法」「トラブル状況」「比較対象」の五つが抽出され、それぞれの質問の内容に応じて質問に含まれる構成要素は異なる。「システム要求事項」とは、ある機能をシステムにより実現するために必要なネットワークやオペレーションシステムなどに関する特徴を記述した部分であり、「機能要求事項」とは、システムによりどのような機能を実現したいのかに関して記述された部分である。また、「機能実現方法」とは、「システム要求事項」と「機能要求事項」とに基づいて、どのように機能を実現していくかという方法に関して記述された部分であり、「トラブル状況」とは、機能を実現するにあたって生じた問題に関して記述された部分である。そして、「比較対象」とは「システム要求事項」と「機能要求事項」とを満足させる製品やコンポーネント、設計方法等において複数の候補が挙げられており、そのどれが適切かを質問によって尋ねる場合に、挙げられた候補に関する情報が記述された部分である。

表 5.5: 質問の種類と記述構造との関係

質問構成要素	主な質問分類			
	トラブル対処	実現機能調査	製品等紹介	比較評価
システム要求事項	○	○	○	○
機能要求事項	○	○	○	○
機能実現方法	○	○		
トラブル状況	○			
比較対象				○

各質問内容に分類される質問は、構成要素のうちの最も関連が深い要素だけでなく他の要素も含んでいる。例えば「トラブルシューティング依頼」の場合、質問者であるシステムエンジニアは「トラブル状況」を伝え、その解決方法を技術支援チームから引き出すことが目的であると考えられる。そして、「トラブル状況」を伝えるためのコンテキストとして、トラブルが生じることとなったある機能を実現するための方法（「機能実現方法」）、および、どのような機能の実現が要求されているか（「機能要求事項」）や機能実現するためにネットワークやオペレーションシステムなどの特徴（「システム要求事項」）などの記述がなされていると考えられる。

また、実現機能調査依頼の場合、システムエンジニアは「機能実現方法」を説明し、その方法が妥当なものかどうかという判断を技術支援チームから獲得するのが目的であると考えられる。そして、「機能実現方法」を説明するためのコンテキストとして、「機能要求事項」や「システム要求事項」が記述される。製品等調査依頼や比較評価依頼の場合も同様である。

このように「期待以上」の回答を得た場合も「期待以下」の回答を得た場合にも質問が記述される構造は共通していることが観察された。そこで、質問の記述の仕方に差異

表 5.6: ドメインタームを抽出した質問数

	期待以上	期待以下
トラブルシューティング	23	23
実現機能調査	36	36
製品等紹介	38	38
比較評価	5	5

があるとするれば質問を構成する要素がどのように記述しているかが異なると考える。そこで、質問の構成要素の記述の仕方を検証するために、各構成要素に含まれる情報通信関連用語（ドメインターム）の比率を、比較する。このようなドメインタームの数量が多いほど、記述内容の曖昧性は減少する [17] と考えるからである。

5.5.3 検証結果

ドメインタームを抽出した質問数は、質問内容が四つの分類のいずれであったか、および回答に対する認識が「期待以上」か「期待以下」という分類に対して、それぞれ表 5.4 に示した値であった。これらの数字は、表 5.10 において期待以下の場合にそれぞれの質問分類に分けられた Q&A の数と同じであり、この数だけ「期待以上」の場合に対しても無作為抽出を行った。無作為抽出の方法はくじ引き方式による単純任意標本抽出に基づく。

質問の各構成要素に含まれるドメインタームを抽出するにあたっては、Windows95 などの製品マニュアルや、以下の辞書を参照しながら人手で行った。ある質問部分から抽出されたドメインタームの例は図 5.3 に示したとおりである。そして、抽出したドメインタームの平均個数と分散 σ^2 は表 5.5 に示す通りである。

- member.nifty.or.jp/~toyoki/pcdic/
- www.ascii.co.jp/ghelp/
- www.alles.or.jp/~cherub/abbreasy/
- www.asuka.net/wavedic/
- www.big.or.jp/~mio/ga/cd/cddic/cd_dic.htm
- www.fmworld.ne.jp/internet/p1400000/p1400000.html
- www.lares.dti.ne.jp/~yomi/

「期待以上」および「期待以下」の回答を獲得する元となった各質問内容ごとの平均のドメインターム数は表のようであった。

図 5.11 と図 5.12 に、「期待以上」の回答を得た質問と「期待以下」の回答を得た質問とにおいて、質問の各構成部分にどれだけのドメインタームが分布しているかを比較した結果を示す。図 5.11 と図 5.12 とが示しているように、例えばトラブルシューティング依頼の場合、システムエンジニアが、質問を発する際の目的である「トラブル状況」の

表 5.7: 抽出したドメインタームの平均個数

	期待以上	期待以下
トラブルシューティング	113 $\sigma^2 = 111.8$	109 $\sigma^2 = 132.2$
実現機能調査	87 $\sigma^2 = 49.3$	85 $\sigma^2 = 60.0$
製品等紹介	56 $\sigma^2 = 132.9$	61 $\sigma^2 = 149.9$
比較評価	63 $\sigma^2 = 98.7$	58 $\sigma^2 = 125.6$

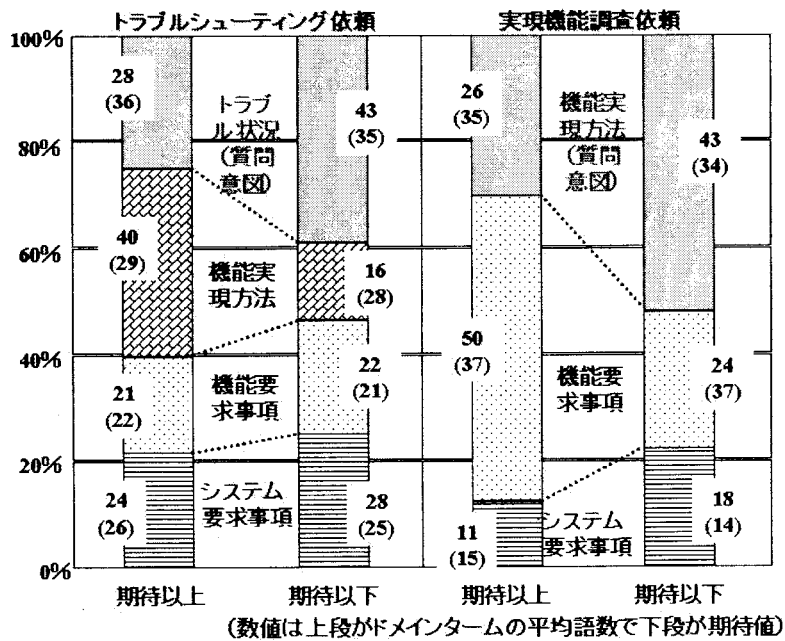


図 5.11: 質問におけるドメインタームの分布と回答に対する認識 (1)

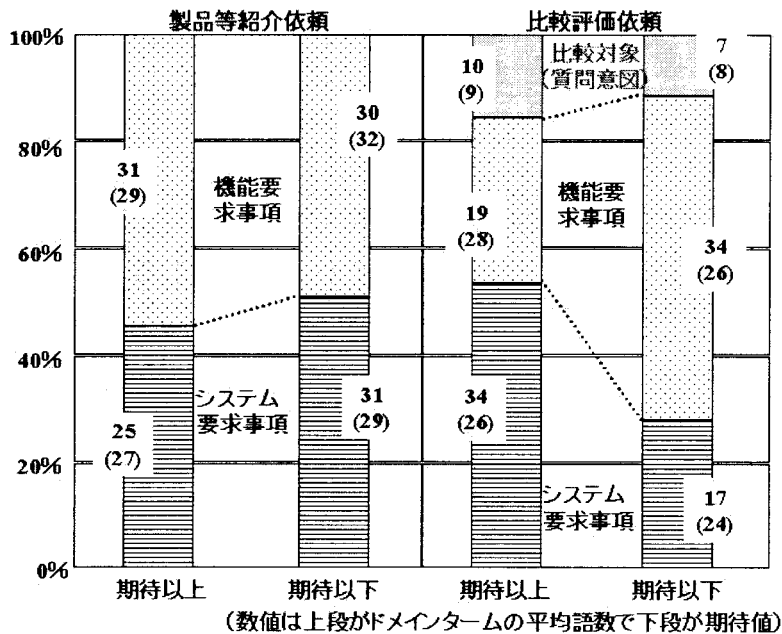


図 5.12: 質問におけるドメインタームの分布と回答に対する認識 (2)

情報よりも、そのコンテキストである「機能実現方法」に関する情報をより詳細に（ドメインタームを多く用いて）記述した方が、「期待以上」の回答を得られる確率が高いことが分かる。逆に、システムエンジニアが、質問を発する際の目的である「トラブル状況」に関する情報を十分に伝えることだけを意識し、「機能実現方法」に関する情報が不十分であると、「期待以下」の回答しか得られないことが多い。このような傾向は製品等紹介依頼に属する質問以外で観察された。表 5.4 と同様に、回答に対する認識と構成部分に含まれるドメインタームの分布との関連性について各質問分類ごとに χ^2 検定を行ったところ、

- トラブルシューティング $\Rightarrow \chi^2 \approx 13.53$
- 実現機能調査 $\Rightarrow \chi^2 \approx 16.04$
- 製品等紹介 $\Rightarrow \chi^2 \approx 0.44$
- 比較評価 $\Rightarrow \chi^2 \approx 11.03$

という値がそれぞれ得られた。トラブルシューティング依頼の場合、回答に対する認識が二通り、構成要素が四通りなので自由度は 3 である。自由度 3 の χ^2 分布における有意差 5% に相当する値は 7.81 なので、トラブルシューティング依頼に関しては回答に対する認識と構成部分に含まれるドメインタームの分布とは関連性があると判断できる。実現機能調査依頼と比較評価依頼の場合、回答に対する認識が二通り、構成要素が三通りなので自由度は 2 である。自由度 2 の χ^2 分布における有意差 5% に相当する値は 5.99 なので、いずれの場合も回答に対する認識と構成部分に含まれるドメインタームの分布とは関連性があると判断できる。製品等紹介依頼に関しては、回答に対する認識が二通り、構成要素が二通りなので自由度は 1 である。自由度 1 の χ^2 分布における有意差

5%に相当する値は 3.84 であるから、製品等紹介依頼に関しては回答に対する認識と構成部分に含まれるドメインタームの分布とは関連性がないと判断できる。

したがって、技術支援チームから「期待以上」の回答、つまり問題状況に即した意識していた以上の知識や情報をシステムエンジニアが得るためには、技術支援チームがそのような回答を作成するために必要とする質問の構成要素に関してドメインタームを十分に含んで記述することが重要である。そのような質問が問題状況を適切に記述している質問であると言える。製品等紹介依頼の質問に関しては、その構成部分はシステム要求事項と機能要求事項の二つと少ないため、回答者にとっては質問の意図を限定するコンテキスト情報が少ないことが期待以上の回答を得ずらい原因ではないかと考えられる。

5.6 第四の効果の検証：Q&Aの再利用性の検証

本節では、第四の効果である再利用性の向上を検証するための仮説と検証方法、検証結果について論じる。技術支援システムに蓄積された Q&A の再利用性を以下の二つの観点から検証する。

- 検索結果として得られた Q&A が役に立ったかどうか、そのうち期待以上の知識や情報が含まれていた割合はどのくらいか
- 役に立つ Q&A を検索により迅速に獲得できたかどうか

まず、検索結果として得られた Q&A に問題状況に即した意識していた以上の知識や情報がどの程度含まれているかを検証するにあたり、検索結果に対してシステムエンジニアがどのように認識するかを「役に立つ」ものかどうか、および「要求していたものかどうか」という二つの観点から分類した理由およびその結果について論じる。

次に、「役に立つ」Q&A を迅速に獲得できているかどうかを調べるために、現在、技術支援システムで利用している全文検索の方法に関するアンケート方法とその結果、および「役に立つ」Q&A を迅速得るための検索手法について論じる。

5.6.1 検索結果として得られた Q&A の有益性に関する調査

データベースを検索した SE が、検索結果として提供された Q&A をどのように認識しているかについて、以下の二つの観点から調査した。

- 検索結果として提供された知識 Q&A が「役に立つ」ものであったかどうか
- 検索結果として提供された知識 Q&A が「要求していた」ものであったかどうか

この二つの観点を考慮した理由は、提供された Q&A がシステムエンジニアの要求していないものであっても、検索を行う時点では意識されていなかった別の問題等に役に立つ可能性があるからである [80, 91]。このような Q&Aこそ、意識していなかった問題状況に関する知らなかった知識や情報であり、問題状況をより正確に明確化すること、ひいては問題解決の質を高めることにつながる。そこで、検索結果として得られた Q&A の有益性に関するアンケート調査を行うにあたり、Q&A に対する認識を図 5.13 に示すように四つに分類した。「役に立つ」と認識される場合が多ければ多いほど再利用時における Q&A の有益性は高く、さらにそのうち「要求していないが役に立った」と認識される場合が多いほど Q&A の問題解決における再利用性は高いと判断する。

	要求していた		
	要求していたが 役に立たなかった	要求しており 役に立った	
役に立た なかった			役に立った
	要求しておらず 役にも立たなかった	要求していないが 役に立った	
	要求していなかった		

図 5.13: 検索結果に対する認識の分類

技術支援システムのデータベースに対して検索を行ったシステムエンジニアを対象に、検索結果として提供された Q&A が「役に立ったかどうか」「要求していたものかどうか」についてアンケート調査を実施した。平成 9 年 1 月 1 日から平成 12 年 3 月末までに 3635 名からアンケートに対する回答が得られた。その結果を図 5.14 に示す。

この結果から、「要求していたものであり役に立った」と「要求していたものではないが役に立った」とがそれぞれ全体の約四割を占めることが観察された。したがって、技術支援システムは、そのデータベースに蓄積された Q&A が再利用される際にも、問題解決に有効であると判断する。

また、この結果は従来の検索技術に新しい指針を与えるものである。すなわち、従来の検索に関する研究では、再現率（すべての正解データのどれだけが検索結果に含まれているか）と適合率（検索結果にどれだけ正解データが含まれているか）という、検索者が検索意図を明示化できることを暗黙的に前提とした検索手法の評価を行ってきた。しかし、本調査の結果から、検索結果が問題解決に役立つものであるためには、検索意図が明示的に示されない場合すなわち「要求していたものではない」場合にも「役に立つ」情報を提供することが重要である。

5.6.2 有益な Q&A を獲得するまでの迅速性に関する調査

検索結果に対する認識に関するアンケートを行った際に、その検索結果を得るまでの過程についてどのように認識しているかも合わせて調査した。アンケートの選択肢として、(1) 比較的容易に役に立つ Q&A が得られた、(2) 何度も検索条件を入力し直した上で役に立つ Q&A が得られた、(3) 役に立つ Q&A が得られなかった、の三つである。この調査は平成 9 年 1 月 1 日から平成 10 年 12 月末にわたって実施され、511 件のアンケー

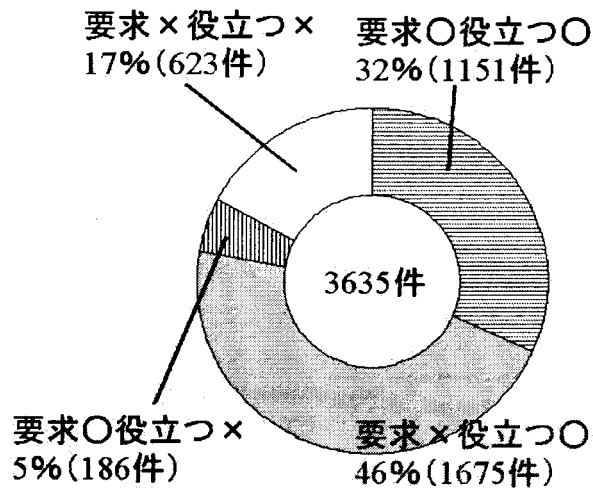


図 5.14: 検索結果として得られた Q&A の有益性に関する調査結果

ト結果が得られた。その内訳を図 5.15 に示す。

「比較的容易に役に立つ Q&A が得られた」場合が約 30%であり、これは迅速に問題解決に役に立つ知識や情報が得られた場合と考えられる。しかし、「何度も検索条件を入力し直した上で役に立つ Q&A が得られた」場合が約 40%に及び、これは迅速に問題解決に必要な知識や情報が得られなかった場合である。役に立つ Q&A が得られなかった場合も約 30%に及ぶ。役に立つ Q&A が得られない原因は、検索の仕方に問題がある場合と、そもそも必要な Q&A が技術支援システムのデータベースに蓄積されていない場合とが考えられる。前者の場合は検索手法を検討しなおす必要がある。役に立つ Q&A が迅速に獲得できないことが繰り返されると検索するよりも直接技術支援チームや他のシステムエンジニアに質問した方が早いと判断され、同じような質問が繰り返され、組織全体のコミュニケーション効率が低下する。後者の場合、再利用時に役に立つと認識される Q&A をより多く蓄積しておくとともに、どのような Q&A がすでに存在しどのような Q&A が存在しないのかを把握できるような仕組みが必要である。

5.7 まとめ

本章では、回答作業の専門化による電子メールを用いた Q&A 方式に基づくアプローチに期待される以下四つの効果が実際に確認されたことについて論じた。

第一の効果: 問題状況に即した知識や情報の迅速な獲得

第二の効果: 問題状況に即した意識していなかった知識や情報の獲得による問題状況の再認識

第三の効果: 質問として明示化した問題状況を客観視することによる問題状況の再認識

第四の効果: 再利用される時に他者が理解するのに十分な情報が Q&A として記録されることによる再利用性の向上

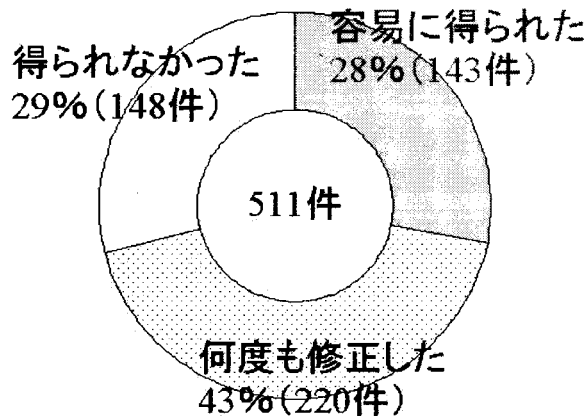


図 5.15: 有益な Q&A を獲得するまでの迅速性に関する調査結果

まず、本アプローチに基づく実例として、株式会社エヌ・ティ・ティ・データで開発・運用している技術支援システムの枠組みと、質問もしくは検索により技術支援システムを利用したシステムエンジニアの人数や形成・蓄積された Q&A の数量などを説明した。

第一の効果を検証するために、質問を送付する際に開発者に回答期限を設定してもらい、回答が送付された日時に関するログとの差を集計した。約 80% の開発者が期限内に回答を入手できており、期限内に回答を入手できた場合の回答に対する開発者の認識は「期待以上」が、「期待以下」の比率より高く、逆に、期限内に回答を得られなかった場合の回答に対する開発者の認識は「期待以下」が「期待以上」の比率を上回ることが観察された。

第二の効果を検証するために、質問を行った開発者がその回答に対してどのように認識しているかを調査した。995 件の調査回答が得られ、うち約 50% が「期待以上の情報が得られた」と認識していることが観察された。

第三の効果を検証するために、質問内容に応じてその構造を分析し、質問の構成部分にどれだけのドメインターム（情報通信関連用語）が分布しているかを調査した。「期待以上」と認識された回答を入手することができた質問と、「期待以下」と認識された回答しか入手することができなかった質問とでは、ドメインタームの分布に違いが観察され、期待以上の回答を獲得するためには問題そのものの記述よりもそのコンテキストとなる部分をより詳細に記述する必要があることが観察された。

第四の効果を検証するために、Q&A の再利用性について、検索結果として得られた Q&A の有益と有益な Q&A を得るまでの迅速性という観点から調査を行った。有益性に関しては要求していたか否かに関わらず役に立ったと認識される場合が約 80% を占め、さらに「要求していないが役に立った」場合が約 50% を占める。迅速性に関しては容易に有益な Q&A を獲得できた場合は全体の 30% に留まる。

次章では、有益な Q&A の迅速な形成と再利用とを統合的に支援することを中心として技術支援システムを改善するため開発したプロトタイプシステム「コミュニティ知識ベース環境」の枠組みと仕組みについて論じる。

第6章 コミュニティ知識ベース 環境：検証結果に基づく 改善

本章では、質問形成と Q&A の検索の支援を中心に本アプローチを改善するための枠組みと、開発したプロトタイプシステム「コミュニティ知識ベース環境」の仕組みについて論じる。本アプローチは電子メールを用いることにより質の高い Q&A を迅速に形成・蓄積し、質の高い Q&A を迅速に再利用することを支援する枠組みである。このような知識管理においては、Q&A の形成と再利用とを統合的に考慮する必要があることについて論じる。

本アプローチにおけるこれら二つの側面を統合的に考慮するために、5.3.3 節で得られた回答に対する質問者の認識に関する調査結果と、5.6.1 節で得られた検索結果として得られた Q&A に対する検索者の認識に関する調査結果との相関関係を調査した結果について論じる。

本アプローチにおける検索を改善するにあたり、5.5 節で抽出された質問の記述構造が検索にも応用できるのではないかと考え、質問の記述構造を反映したキーワードセットとそうではないキーワードセットとを用いた場合とで Q&A の全文検索による適合率にどのような違いが見られるかを検証した結果について論じる。

以上の結果から、質の高い Q&A を迅速に形成・蓄積し、質の高い Q&A を迅速に再利用することをより効率的に支援するためには、質問を行う場合にも検索を行う場合にも質問の記述構造を意識することが重要であることが判明した。質問を行う場合、あるいは検索を行う場合に質問構造を意識させるために開発したプロトタイプシステム「コミュニティ知識ベース環境」の仕組みについて論じる。

6.1 Q&A の形成と再利用の統合的支援のための 調査

本アプローチは電子メールを用いることにより質の高い Q&A を迅速に形成・蓄積し、質の高い Q&A を迅速に再利用することを支援する枠組みである。このような知識管理においては、以下の二つの観点を統合的に考慮する必要がある。

知識の形成: いかに質の高い知識をより多く [52, 79], 実務を通じて漸次的に [53] 形成し蓄積していくか

知識の再利用: 形成・蓄積された知識を再利用する際に、「検索・理解・フィードバック」というサイクルをいかに円滑に支援するか [25]

本アプローチもしくはその具現化である技術支援システムを改善するにあたっては、質問形成（知識の形成）と Q&A の検索（知識の再利用）を個別に支援するのではなく、統合的に支援する必要がある。

そこで本節では、質の高い Q&A を迅速に形成・蓄積し、質の高い Q&A を迅速に再利用することを統合的に支援するために、技術支援システムにおいて行った以下の二つの観点からの調査について論じる。

- (1) 形成時においても再利用時においても質の高い Q&A とはどのような特徴を持っているか
- (2) 形成時においても再利用時においても質の高い Q&A を迅速に形成・再利用するためにはどのような手法が有効か

6.1.1 形成と再利用とにおける Q&A の質の相関関係に関する調査

5.6.1 節で示した結果（図 5.14）から、技術支援システムのデータベースが「役に立つ」Q&A を利用者であるシステムエンジニアに提供できるのであれば、検索手法の良否（利用者が要求していたかどうか）に関わらず、利用者は技術支援システムにおけるデータベースの検索により問題解決に有益な Q&A を得ることができる。

データベースが提供した Q&A が利用者にとって「役に立つ」かどうかは、それが形成されるときに「質問と回答」の対として十分な情報を持っているかどうか依存すると考えられ、以下のような分析を行った。

データベースから提供された Q&A を再利用するシステムエンジニアが認識した Q&A の有益さ（5.6.1 節の図 5.14）と、Q&A が形成され蓄積されるときにの有益さ、すなわち質問を行ったシステムエンジニアの回答に対する認識（5.4.3 節の図 5.9）との相関関係について分析を行った。その結果を図 6.1 に示す。

図 6.1 において、期待以上の場合と期待以下の場合とにおける検索結果に対する認識との関係について、5.5 節と同様に χ^2 検定を行ったところ、70.80 という値が得られた。この場合、回答に対する認識は二通り、検索結果に対する認識は四通りであるので、自由度は $(2-1) * (4-1) = 3$ である。自由度 3 の χ^2 分布における有意差 5% に相当する値は 7.81 であり、計算した結果はこの値を大幅に上回ることから、期待以上の場合と期待以下の場合とで検索結果に対する認識には強い関連があると判断される。すなわち、システムエンジニアが再利用時に「役に立つ」と認識した Q&A は、その Q&A が形成される際に質問者が「期待以上」と評価した回答とその元となった質問である傾向がある。逆に、形成される時点で質問者により「期待以下」と認識された回答とその質問は、それが再利用されるときに「役に立たない」と利用者が認識する傾向がある。

5.5 節で論じた調査結果から、「期待以上」と認識される回答を獲得するためには、質問において質問の内容に応じた記述構造を意識する必要がある。つまり、形成時と再利用時においてともに有益であると認識される Q&A を形成・蓄積するためには、形成時に質問の記述構造を意識して質問を形成することが重要である。

6.1.2 質の高い Q&A の迅速な獲得に関する調査

5.3.3 節で論じたように、技術支援チームに質問することにより「期待以上」の回答を迅速に獲得するためには、質問内容に応じた記述構造を意識して質問を行うことが有

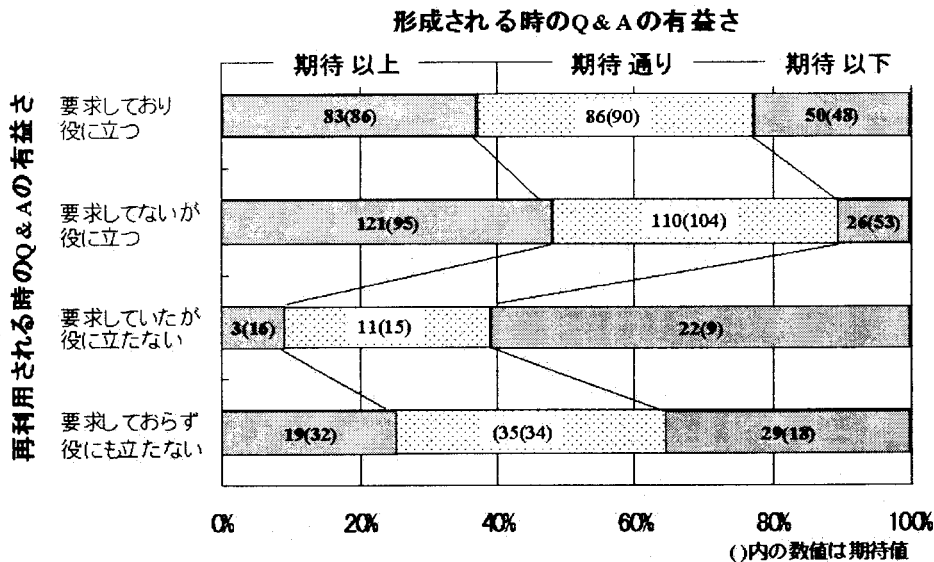


図 6.1: 形成時と再利用時の Q&A の有益さの相関関係

効である。一方、5.6.2 節で論じたように、役に立つ Q&A を迅速に獲得するためには現状の全文検索の手法では不充分である。そこで、本節では役に立つ Q&A を迅速に獲得するための検索手法について行った検証実験について論じる。

Q&A の検索を行うシステムエンジニアが置かれている問題状況と、技術支援チームに質問を行うシステムエンジニアの状況とは、ソフトウェア開発において生じた問題を解決するために問題状況に即した知識や情報を必要としているという点において類似している。したがって、「期待以上」の回答を得るために質問を発するシステムエンジニアが意識しなければならない質問の記述構造が検索においても利用できるのではないかと考え、以下のような実験を行った。実験対象は、5.4 節における調査結果から、「期待以上」の回答とその質問とが技術支援システムのデータベースに最も数多く蓄積されていると考えられる平成 9 年 1 月 1 日から平成 12 年 3 月末までに形成された「トラブルシューティング依頼」に関する Q&A とした。

【実験の準備】 質問を発した SE から「期待以上」と評価された回答とその質問の対 (Q&A) から 1 件を任意に抽出し、「要求していた」検索結果の「正解データ」として設定した。「正解データ」として設定した対の質問から、5.5 節において論じたものと同様の方法でドメインタームを切り出し、コンテキスト付加構造のどの構成部分（「システム要求事項」、「機能要求事項」、「機能実現方法」、「トラブル状況」）に属するかを調査した。

【実験 1】 これら四つの各構成部分に属するドメインタームからそれぞれ一つずつを抽出してキーワードセットとし、この中からキーワードとして一つ、二つ、三つ、四つと組み合わせ、それぞれの組み合わせについて、技術支援システムのデータベースに蓄積されている約 13,000 件の Q&A に対して AND 条件による全文検索を行い適合率を調査した。

【実験 2】 5.4 節で述べた結果から、トラブルシューティング依頼の場合に「期待以上」の回答を獲得するためには、質問を発したシステムエンジニアはその質問の中で

特に詳細に「機能実現方法」に関して記述（ドメインタームをより多く用いて）しなければならない。そこで、この「機能実現方法」に属するドメインタームから四つを任意に抽出してキーワードセットとし、この中からキーワードとして一つ、二つ、三つ、四つと組み合わせ、それぞれの組み合わせについて、技術支援システムのデータベースに蓄積されている約 13,000 件の Q&A に対して AND 条件による全文検索を行った場合の適合率を調査した。

適合率は検索結果の中にどれだけ正解が含まれるかという比率を示したものである。検索結果の中に役に立つ Q&A が多く含まれているほど問題解決に有益な Q&A を迅速に獲得できるため、適合率が高いほど有益な Q&A を迅速に獲得できることを示す。以上のような実験準備と実験 1, 2 を一つのセットとし計 10 セットを行った場合に計測された適合率の平均を図 6.2 に示す。

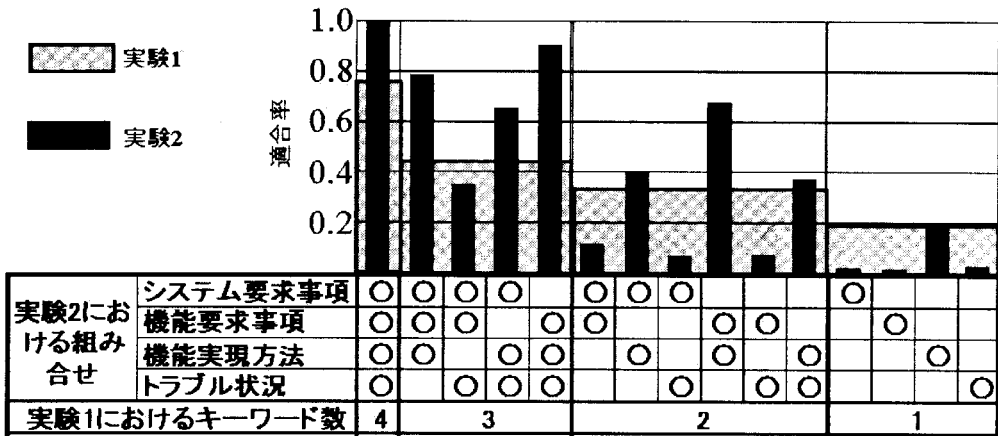


図 6.2: 質問の記述構造を反映した検索手法による適合率に関する調査結果

この結果から、コンテキスト付加構造の各構成部分に属するドメインタームからそれぞれ一つずつを抽出してキーワードセットとした場合（実験 1）、「要求していた」Q&A を確実に獲得できることが観察された。また、検索に用いるキーワード数が少ない場合でも、「機能要求事項」と「機能実現方法」に属するドメインタームを組み合わせることにより、適合率が飛躍的に向上することが観察された。一方、トラブルシューティングに関する Q&A を検索する場合、「トラブル状況」に属するドメインタームをキーワードとしなくても適合率はあまり変化しないことも分かった。このことから、通常システムエンジニアが重要であると感じる「トラブル状況」の説明は、検索の際にはあまり重要でなく、むしろそのコンテキストを説明するようなキーワードで検索するほうが有効であるといえる。これは Q&A の形成の際と同じ結果である。

6.2 Q&A の形成と再利用の統合的支援のための条件

6.1 節で論じた二つの調査結果から、質の高い Q&A を迅速に形成・蓄積し、質の高い Q&A を迅速に再利用することを統合的に支援するためには、他者へ質問する際にもデータベースを検索する際にも、質問内容に応じた記述構造を意識することが重要であることが分かった。本節では、こうした質問内容に応じた記述構造をシステムエンジニアに意識してもらうための枠組みについて論じる。

質問の記述構造を意識するためには、自分が置かれている問題状況について十分な知識や情報を持っており、質問構造のどの構成部分にどの知識や情報が該当するかを把握できていなければならない。つまり、自分の置かれている問題状況について質問構造に則して十分に明確化・明示化することができなければ質問の記述構造を意識して質問や検索を行うことはできない。

自分の置かれている問題状況を明示化するという事は、その問題状況に関してどこまで把握しており、何を把握していないのかを明確にする能力が要求される [63]。そして、問題状況をより正確に把握するためには、その問題状況に関する知識や情報（領域知識）を習得する必要がある [19]。このような問題状況を把握するための領域知識や能力を習得するために効果的な方法は、他者との相互作用である [12, 4, 46]。

現在の技術支援システムでも、Q&A という他者との相互作用により、問題状況のより正確な把握を支援している。しかし、質問形成という観点から見ると、システムエンジニアが質問を記述する際に、記述した質問が適切かどうかを意識することは支援されていない。つまり、他者への質問によって獲得した回答に対して、「期待以上」「期待以下」などといった認識はするが、なぜそのような認識をする回答が得られたか、「期待以上」の回答を他者から得るためにはどのように質問を記述すればよいのかといったことを意識することはない。このような意識をシステムエンジニアが持たない限り、領域知識や問題状況を把握する能力を習得しても質問形成に効率的に反映させることは困難である。問題状況に即した意識していなかった知識や情報を他者から得るために、自分の記述した質問が適切かどうかを意識した上で、他者との相互作用により領域知識や問題状況を把握する能力を習得していく必要がある。

現在の技術支援システムの枠組みにおいて、自分の記述した質問が適切かどうかを意識することは困難である。回答を行うことが技術支援チームの仕事となっているため、質問を通じた回答の獲得という、他者との協調的な相互作用に主体的に関わっていることが意識されないからである [29]。また、回答を行う技術支援チームも、質問に対して必要な知識や情報を教授することが目的となり [54]、どのような質問であれば十分な回答を形成できるかということまで意識されないため、Q&A が他者との協調的な相互作用であることを認識づらい。Q&A の形成がこのような他者との協調的な相互作用であるという意識を質問する側も回答する側もともに持たなければ、より適切な質問とそれに対する十分な回答とを形成していくことは困難である。

このような意識を他者との相互作用において持つためには、ある問題状況に関して専門家ではない人同士が教え合うという方法が有効である [29]。この方法を現在の技術支援システムに適用すると、持っている知識や情報の質や量が様々なシステムエンジニア同士が質問と回答を行い合うことになる。しかし、システムエンジニア同士が質問と回答を行い合うことは、3.4 節で述べたように、ソフトウェア開発全体の効率を低下させる。そこで、ある問題状況に関して関心を共有する人々が都合のよい時間と場所において質問と回答を行うことのできる環境、すなわちシステムエンジニア間におけるネットワーク上でのコミュニティの形成が Q&A を通じて支援されている環境が必要である。コミュニティの定義は様々であるが、コミュニティの概念を最初に提唱した McIver [44] や、

様々なコミュニティに関する研究におけるコミュニティの概念を比較した Hillery [31]によれば、一般的に以下の三つの条件を満たすことである。

- (1) 他者との関心の共有に基づき、
- (2) 他者との相互作用を行うことにより、
- (3) ある集団の中でアイデンティティを確立すること

このようなコミュニティでは個々人は主体的にコミュニティでの活動に参加しているため、他者との相互作用の質を意識することが促進される。すなわち Q&A という相互作用においては他者への質問の質や他者への回答の質が意識されるため、より問題状況が明確化された質問とそれに関するより多くの知識や情報を含む回答の形成が促進される。このようなコミュニティの効果により、問題状況を適切かつ迅速に明確化する能力や知識を習得することが促進される。そこで、改善のためのアプローチである Q&A を通じた学習するコミュニティの形成の条件では、上述したコミュニティの三つの条件に以下の四つ目の条件を加えて検討していく。

- (4) 問題状況を迅速かつ適切に明確化する能力や領域知識を習得する

6.3 コミュニティ知識ベース環境

本節では、6.2 節で論じた質問の記述構造を意識するための条件を支援することにより技術支援システムを改善するための枠組みと、その枠組みを実装したプロトタイプシステム「コミュニティ知識ベース環境」の仕組みについて論じる。

6.3.1 コミュニティ知識ベース環境の枠組み

6.2 で論じた条件を満足させることにより、質の高い Q&A の迅速な形成・再利用の統合的支援という観点から技術支援システムを改善するためのプロトタイプシステムとして「コミュニティ知識ベース環境」を開発した。ここでのコミュニティとは、他のシステムエンジニアとの Q&A という相互作用を通じて、問題状況を迅速かつ適切に明確にするための能力や知識を習得していくことを目的に主体的に参加する人々の集合体である。本プロトタイプシステムの枠組みを図 6.3 に示す。

まず Q&A が形成される時に取得される質問者や回答者の属性、形成された日時、Q&A のカテゴリ、および Q&A が再利用される時に取得される再利用者の属性、再利用された日時、再利用の際の評価情報などを Q&A 履歴として蓄積し、対応する Q&A とのリンクを形成する。この Q&A 履歴はコミュニティに公開され、コミュニティの参加者は様々な Q&A 履歴の様々な属性により Q&A 履歴をソートすることが可能である。

この Q&A 履歴は関心の共有や検索による再利用、他者への質問などを行う際に参照される。関心を共有する場合には参加者は様々な属性で Q&A 履歴を参照し自分の興味のあるカテゴリが存在するかどうか、どのようなレベルの Q&A が誰によっていつそこで形成および再利用されているかといったことを把握することにより、自分の関心とコミュニティとの活動との関係の有無を判断する。Q&A 履歴を検索によって再利用する場合は、主に Q&A 履歴のカテゴリにより必要な Q&A 履歴を見つけ出し、その Q&A 履歴の過去の再利用評価を見ることにより、誰がいつどのような評価をその Q&A に対して行っているかを把握し、その Q&A が自分にとって役に立ちそうかどうかを判断し、役に立つと判

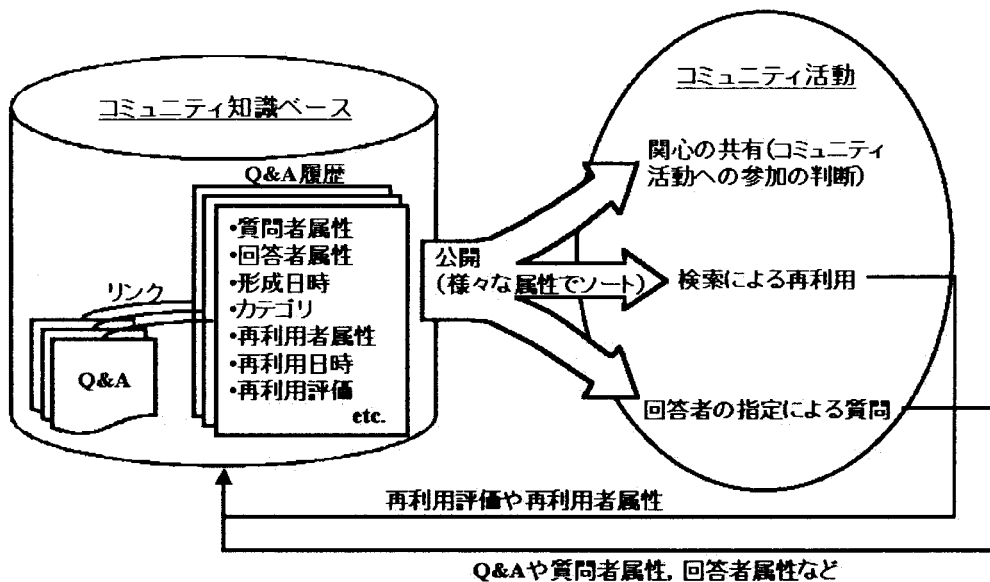


図 6.3: コミュニティ知識ベース環境の枠組み

断した場合はリンクをたどって実際の Q&A を参照する。参照した Q&A には評価を入力する機能が埋め込まれており、再利用評価が入力されるとその Q&A の履歴に追加される。他者への質問の際には Q&A 履歴の主にカテゴリと質問者、回答者、再利用者の属性でソートして Q&A 履歴を参照し、適切な回答が得られそうな人を見つけ出し、Web 上でその人の氏名等と質問内容とを入力すると、指定された回答者に電子メールにより回答依頼の通知がなされ、その電子メールには質問内容が登録された Web ページへのリンクが添付されている。指定された回答者がこのリンクをたどって回答を入力すると、再度電子メールにより質問者に回答が登録された通知がなされる。

Q&A 履歴が公開されていることにより、コミュニティに参加しようとしている人はコミュニティで行われている Q&A が自分の関心と共通するかどうか、すなわち他の参加者との関心の共有が可能かどうかを判断することが支援される。

Q&A 履歴を参照して回答者を指定することにより、他者との Q&A を通じた相互作用が支援され、Q&A のやり取りの質を意識することが可能となるとともに、お互いのアイデンティティ、すなわちお互いがどのようなことに関心がありどのくらいの能力や知識を持っているかを認識することが支援される。

Q&A 履歴に過去にその Q&A が再利用された際の評価が含まれていることにより、Q&A 履歴を参照することによりどの Q&A が自分にとって役に立ちそうかを判断する基準が与えられ、迅速に役に立つ情報を獲得することが支援される [62]。この評価を参照された Q&A を形成した質問者や回答者が参照することにより、今後の Q&A 形成の質に対する認識とコミュニティにおいて他者に自分の関心や能力などが意識されているというコミュニティにおける自分のアイデンティティに対する認識とが支援される。

6.3.2 プロトタイプシステム

コミュニティ知識ベース環境の開発は Lotus Notes 上でを行い、Domino Server を介した Web アプリケーションとして開発した。Lotus Notes を利用した理由としては、データベース機能や電子メール等のメッセージング機能、そしてコンポーネント間の情報のやり取りを規定するエージェント機能など、本研究で提案するコミュニティ知識ベース環境のようなグループウェアに必要な機能をすべて備えているためである。また、Web アプリケーションとした理由は、将来的に SOHO やテレワーク等の多様な働き方に対応できるからである。開発に使用した Lotus Notes および Domino Server のバージョンは R4.61 である。

図 6-4 はコミュニティ知識ベース環境において Q&A の履歴を参照する画面の例である。質問者名や回答者名、Q&A の形成日時（回答が登録された日時）、Q&A のカテゴリ、再利用者名などの基準により履歴をソートすることが可能である。そして、履歴をダブルクリックすることにより、その履歴が付加された Q&A を参照することが可能である。

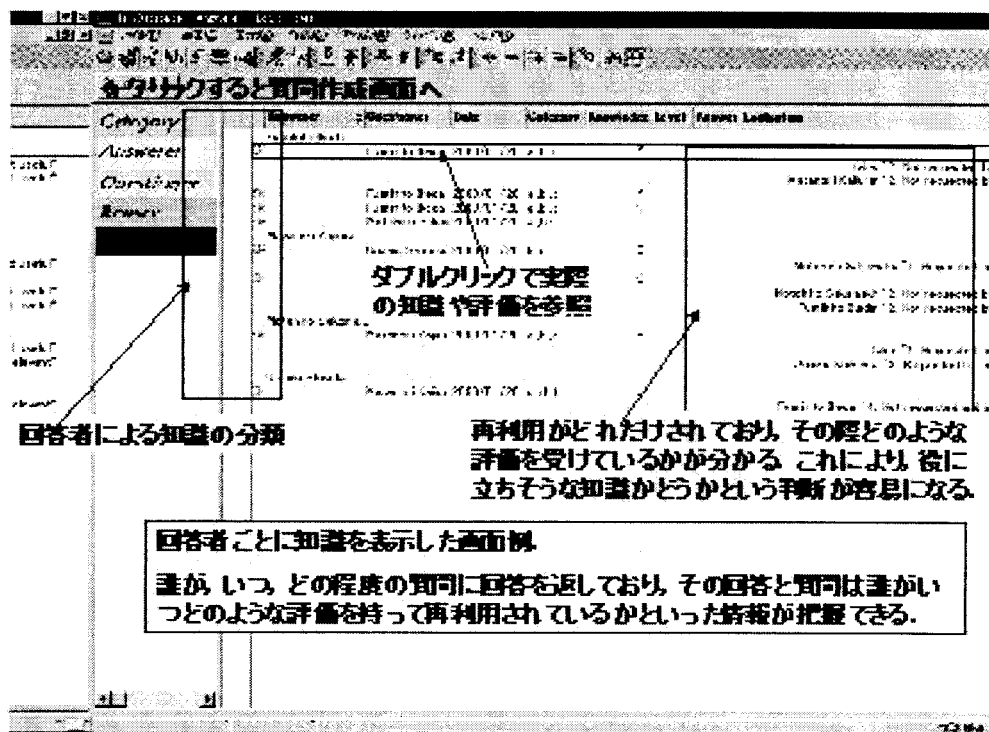


図 6.4: QA&履歴の参照画面例

図 6-5 は質問を入力する画面の例である。ここでは自分の氏名や所属、指定する回答者名、質問内容を記入する。質問した日時は、質問が登録された時にシステムが自動的に付与する。そして、必要事項の記入が終了し質問者が登録ボタンをクリックすると、コミュニティ知識ベースに質問が登録されると同時に、電子メールにより指定した回答者に質問が送付される。

図 6-6 は回答を入力する画面の例である。ここでは回答を入力するとともに、質問がどのようなカテゴリに属するかを入力する。また、質問が適切かどうかを評価し、その評

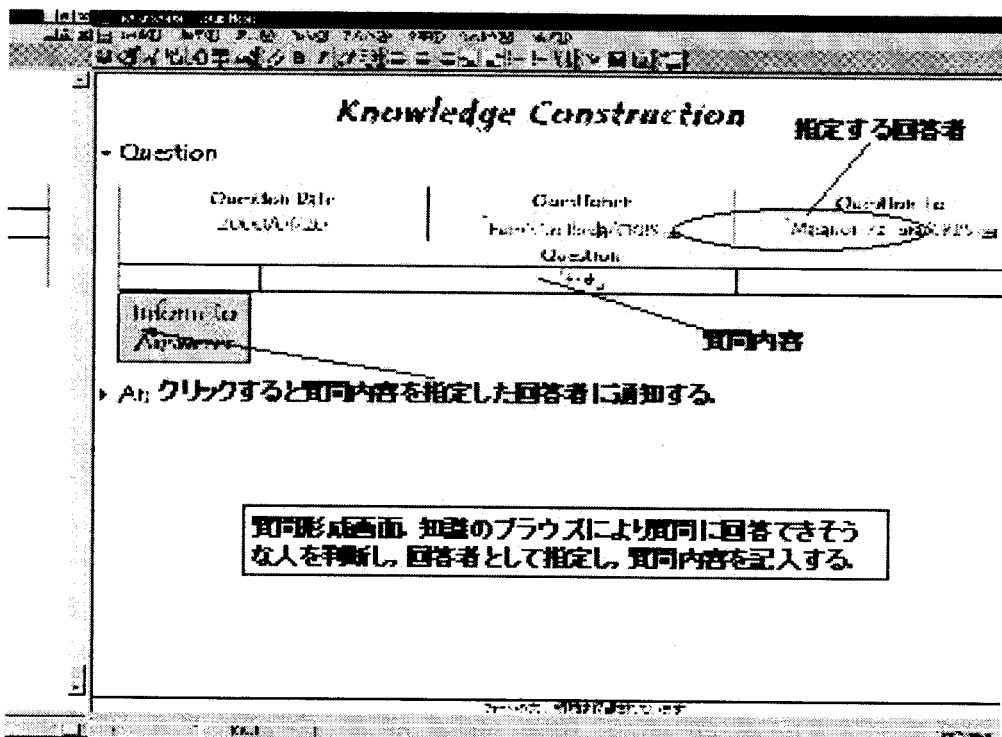


図 6.5: 質問入力画面例

価情報を入力する。必要事項の入力が終了し、登録ボタンをクリックすると、元となった質問の対として回答がコミュニティ知識ベースに登録される同時に、電子メールにより質問者に回答が送付される。

Knowledge Construction

Question: 質問内容から判断し、カテゴリとして3つのキーワードを入力する

Answer: 回答を入力

Answer Date	Answerer	Category
2003/06/20	Q1-Answerer, Q2-Answerer, Q3-Answerer	4, 5, 6

Evaluation of questions

Question	1	2	3	4	5
Hypothesis formation	C-1 Very good	C-2 Good	C-3 Normal	C-4 Bad	C-5 Very bad
Hypothesis Evaluation	C-1 Very good	C-2 Good	C-3 Normal	C-4 Bad	C-5 Very bad
Thoughtful thinking	C-1 Very good	C-2 Good	C-3 Normal	C-4 Bad	C-5 Very bad
Conceptual thinking	C-1 Very good	C-2 Good	C-3 Normal	C-4 Bad	C-5 Very bad
Practical problem solution	C-1 Very good	C-2 Good	C-3 Normal	C-4 Bad	C-5 Very bad

6つの能力を評価

クリックすると回答を質問者にメールで送

図 6.6: 回答入力画面例

図 6-7 は Q&A を参照し、再利用時の評価を入力する画面の例である。ここでは、参照した人の氏名や所属を入力するとともに、参照した結果が「要求していて役に立った」「要求していないが役に立った」「要求していたが役に立たなかった」「要求しておらず役にも立たなかった」の四つの選択肢から評価を選択する。入力が終了し、登録ボタンをクリックすると、コミュニティ知識ベースに登録され、参照した Q&A にリンクが貼られると同時に、参照した Q&A を形成した質問者と回答者に再利用時の評価情報が電子メールにより送付される

6.4 まとめ

本章では、質の高い Q&A の迅速な形成と質の高い Q&A の迅速な再利用とを統合的に支援するにあたり、技術支援システムを対象に行った二つの調査とその結果に基づく改善のための枠組み、およびその枠組みを実装したプロトタイプシステムの仕組みについて論じた。

技術支援システムを対象に行った二つの調査とは、形成時と再利用時の両時点においてともに有益であると認識される Q&A に関するものと、再利用時に迅速に有益な Q&A

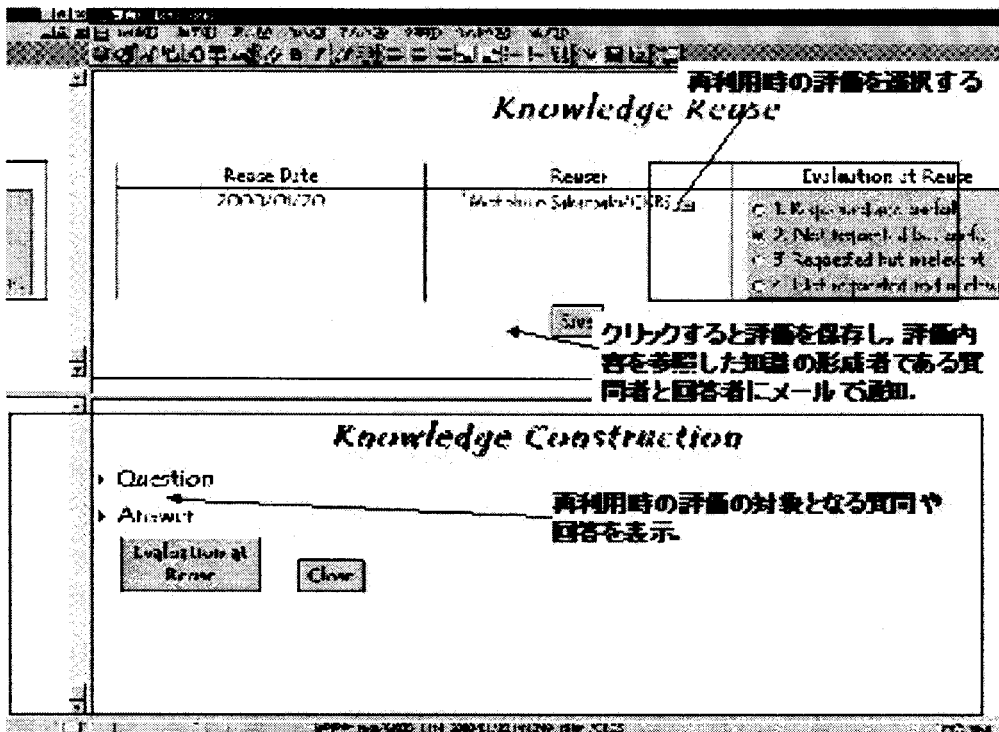


図 6.7: 再利用時の評価入力画面例

を獲得するための検索手法に関するものである。形成時と再利用時の両時点において有益であると認識される Q&A とは、形成時に質問者から「期待以上」と認識された回答とその元となった質問であることが観察された。また、質問の記述構造を反映したキーワードセットを用いた全文検索により適合率を向上させることができることが観察された。質の高い Q&A の迅速な形成と質の高い Q&A の迅速な再利用とを統合的に支援するためには、質問形成および検索の際に質問の記述構造を意識させる必要がある。

そのための仕組みとして「コミュニティ知識ベース環境」を提案した。質問の記述構造を意識するためには、自分の置かれている問題状況を明確化・明示化するだけの知識と能力が必要であるため、コミュニティ知識ベース環境では Q&A の再利用から他者への質問、そして他者への回答へと立場を変えていくことにより徐々に問題状況を明確化・明示化するための知識や能力を習得していくことを支援することについて論じた。そのための仕組みとして、過去にどのような Q&A が誰によっていつ形成され、それが誰によっていつ再利用されたかという履歴をシステムエンジニアに公開し、この履歴を調べて自ら回答者を指定して質問を行うという方法をコミュニティ知識ベースでは取り入れていることについて論じた。

次章では、本論文において提案したコミュニティ知識ベース環境の今後のソフトウェア開発における有用性について、学習する組織の実現およびそのための学習環境の提供という観点から論じる。

第7章 考察

本章では、今後のソフトウェア開発においては学習する組織の実現が求められていることについて論じるとともに、ソフトウェア開発における学習する組織の実現に6章で論じたコミュニティ知識ベース環境がどのように有効であるかについて論じる。

学習する組織は Senge によって提唱された考えであり、その実現にあたっては組織の構成員の一人一人がシステム思考を習得する必要がある [69]。Senge が提唱するこのシステム思考と本論文で論じてきた問題状況を迅速かつ適切に明確化するための能力や知識との類似性について論じる。

学習する組織の考えの中では実際にシステム思考をどのように習得していくかは論じられていない。システム思考は問題状況を明確化するために必要な能力や知識と類似することから、本論文で提案しているコミュニティ知識ベース環境がシステムエンジニアのシステム思考の習得に有効であり、ひいてはソフトウェア開発における学習する組織の実現に有効であることについて論じる。

7.1 ソフトウェア開発における学習する組織の実現の必要性

本章ではソフトウェア開発において学習する組織 [69] の実現が求められる理由について論じるとともに、学習する組織の実現に必要な条件であるシステム思考の習得と本論文で論じてきた問題状況の迅速かつ適切な明確化のために必要な能力や知識との類似性について論じる。

近年の情報通信技術の発達にともない、組織を取り巻く状況はめまぐるしく変化している。このめまぐるしく変化する状況にすばやく対応し組織活動を維持していくために、「学習する組織 (Learning Organization)」の実現が求められている。

従来からの組織における学習方法である OJT や研修といった教授方法は、過去のノウハウや知識がそのまま現在の状況に適応できるような場合に有効である。しかし、組織を取り巻く状況がめまぐるしく変化する現代においては不適切な学習形態である。なぜなら前例のない状況に対処しなければならないからである。さらに、企業のような営利組織においては、前例のない状況に対処するだけでなく、自ら状況を変化させていかなければ営利活動を維持することはできない。したがって、状況を適切かつ迅速に把握し、その対処方法や状況を変化させる方法を新たに見出せる能力や知識を、組織を構成する個々人が学習していくことが要求される [69]。

「学習する組織 (Learning Organization)」という考えを提唱した Senge によれば、学習する組織へと組織を変革していくためには、組織における個々人およびチームは以下に示す五つの基本的なスキルを習得する必要があると述べている [69]。

systems thinking: 状況を構成する様々な物事をそれぞれ関連付けて考えること。こうした思考法を習得することにより、最小限の努力でもっとも効果的に状況を変

化させる方法を導き出すことが可能となる。

personal mastery: 個々人のヴィジョンを明確にし深め、何にエネルギーを注ぐべきかを見極め、困難にぶつかった場合の冷静さや寛容さを身につけ、現実を客観的に見る能力を習得する。こうした能力を習得するためにはまず我々の組織にとって真の問題を見極め、組織における最も高い目標を達成するためにどのような貢献ができるか、すなわち組織における自己実現の仕方を個々人が考える必要がある。

mental models: 我々を取り巻く状況をどのように理解し、どのような行動をおこすかを決定する時の拠り所となる、様々な仮定や一般法則、イメージなどに深く関係したもの。Mental modelを獲得するためにはまず自己を内省する必要がある。つまり、世界に対する内的なイメージを掘り起こし、そのイメージをしっかりと意識し続ける必要がある。

building shared vision: 個々人が持つ共通の将来に対するイメージを見つけ出すことにより、組織に服従するのではない、真の組織への参加が可能になる。

team learning: チームとしての学習の基本は対話であり、個々人の仮説を一つ一つ検証していくだけの寛容さがチームメンバに求められる。現代の組織においてはチームが基本的な学習単位であるため、学習するチームなくして学習する組織はありえない。

学習する組織にとって必要な五つの条件の中でもっとも重要なスキルは systems thinking である [70]。なぜなら、世界を構成する物事を体系的に捉えることにより、より深く世界を理解した mental models を個々人が形成することができ、それらの mental models に基づいて個々人のヴィジョンが明確化・深化され、personal mastery が可能になるからである。さらに、個々人の mental model と personal mastery に基づいて共通のヴィジョンを見つけ出し、その過程において team learning が可能となる。

この systems thinking は組織における意思決定においても重要な役割を果たす。すなわち、systems thinking によって mental models を形成することは、個々人の仕事に対する要求や動機を明らかにするとともに、組織活動における意思決定に必要な適切な状況の把握にも必要なスキルである。そして building shared vision は決定された組織活動の目的と個々人の欲求や動機との擦り合せであり、その擦り合せの過程で必要なスキルが team learning である。

学習する組織という考え方と従来の組織論との違いは、従来の組織論ではリーダーシップを持った人が組織活動のヴィジョンを決定し、そのヴィジョンと個々人のヴィジョンとを擦り合わせるという経営者もしくは管理者の立場から意思決定を捉えていたのに対して、学習する組織の考え方では組織を構成するメンバは個人というレベルで対等であり、その対等な個々人のビジョンから共通の将来像を見つけ出すという真の意味での個人としての組織への参加という観点から意思決定を捉えていることに特徴がある。

学習する組織にとって個々人が習得すべきもっとも重要なスキルは systems thinking、であり [69]、上述したように状況を構成する様々な物事を相互に関連付けて考えられる能力である。これは問題状況を明確化することと同じである。本論文で論じたコミュニティ知識ベース環境は、問題状況を明示化すればかならず抜け落ちてしまう部分 [76] を極力少なくするための能力や知識の習得を支援し、実際の問題状況との乖離の少ない解決策を構築することを支援している。問題状況の明示化により抜け落ちる部分を少なくするためには様々な観点から問題状況を把握する必要がある [46]。すなわち問題状況を構成する様々な物事（部分）を様々な観点から認識することで他の物事（部分）との関係を見出し、明確化された物事（部分）とそれらの関係から解決策を構築することをコミュニティ知識ベース環境は支援しているからである。

7.2 コミュニティ知識ベース環境による学習する組織の実現の支援

コミュニティ知識ベース環境において学習する組織の実現が支援される過程を、参加者の学習およびアイデンティティの確立という二つの観点から論じる。

コミュニティ知識ベース環境における学習過程は図 7.1 のようになる。

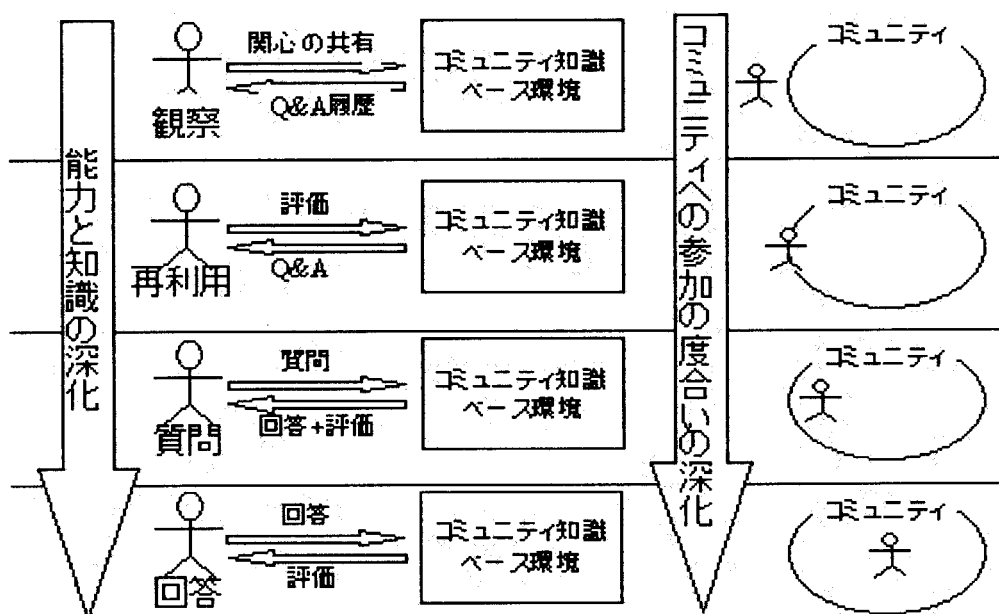


図 7.1: コミュニティ知識ベース環境における学習過程

何か知りたいことがあり、その事柄について知識を持っていない人は、コミュニティで公開されている Q&A 履歴を参照し、自分の関心のあるテーマについてコミュニティ内で学習活動（知識の需要と供給に基づく社会的インタラクション）が行われているかどうかを「観察」する。この段階ではまだコミュニティに参加していない状態である。

観察により自分の関心事についての学習活動がコミュニティで行われていると判断した場合、Q&A 履歴を参照しながら必要な Q&A を参照することを通じて必要な知識を習得し、質問や回答の記述の方法を習得していく。この際、再利用評価を行うことによりその評価が他の参加者がその Q&A を再利用する際に役に立つかどうかを判断する基準となるとともに、その Q&A を形成した質問者や回答者への評価となることから、間接的にコミュニティの他の参加者と相互作用を行うこととなる。この段階は、評価を付与することによりコミュニティに貢献しているが、コミュニティにおいて形成された Q&A を単に参照しているだけで自ら Q&A を形成することに関与していないため、コミュニティにおける位置付けは周辺的なものである。

現在蓄積されている Q&A だけでは必要な知識や情報が獲得できなくなると、Q&A の再利用を通じて習得してきた知識や質問の記述の仕方に基づいて実際に他者へ質問を行うこととなる。これにより必要な知識や情報が他者からの回答として獲得できるとともに、コミュニティに蓄積される Q&A を増加させることに直接貢献することとなる。また、形成した Q&A が他者に再利用され評価がフィードバックされることにより、コミュニティ

におけるアイデンティティ、すなわち誰がどのようなことに関心がありどのくらいの能力や知識を持っているか、が認識される。この段階ではコミュニティに蓄積される Q&A を増加させ、他者がより多くの Q&A を再利用できることに貢献しているが、直接他者の学習を支援しているわけではないため、コミュニティにおける位置付けは再利用の場合よりもやや中心に移動していると考えられる。

Q&A の再利用や他者への質問を通じて問題状況を迅速かつ適切に明確化する能力や知識を習得していくと、その能力や知識の深化が他の参加者から認識されるようになり、回答依頼がなされる。回答を行うためには他者の置かれている問題状況を客観的に理解し、他者が問題解決を行うために有益な知識や情報を提供しなければならない。逆にいえば、回答者として指定されるということはそれだけの能力や知識を自分が持っていることを他者が認識しているということであり、コミュニティにおけるアイデンティティの認識につながる。この段階はコミュニティに蓄積される Q&A を増加させることにより他者がより多くの Q&A を再利用できることに貢献しているとともに、直接他者の学習を支援しているため、コミュニティにおける中心的な位置付けにある。

以上のような学習過程を経ることにより、問題状況を迅速かつ適切に明確化するための能力や知識、すなわちシステム思考を習得することが支援される。

一方、上述した学習過程を主体的に実行するためには、コミュニティから参加者への動機付けがなされる必要がある。学習する組織においては、personal mastery において組織における自己実現、すなわち組織の真の目標の実現において自分がどのような貢献ができるかを考える必要が論じられている [70]。この組織における自己実現こそが組織活動に参加する動機付けと考えられる。コミュニティ知識ベース環境における動機付けはコミュニティにおけるアイデンティティの確立である。コミュニティにおけるアイデンティティとは、コミュニティにおいて共有された関心に基づいて他者との Q&A のやり取りや再利用を行うことを通じて、お互いがどのようなことに関心がありどのくらいの能力や知識を持っているかを認識し合うことである。このような認識の度合いが高いほどアイデンティティは高く、したがってコミュニティへの貢献度も高い。このような Q&A に基づく学習によりコミュニティにおけるアイデンティティを確立していく過程は図 7.2 のようになる。

習得期: 観察を経て、Q&A の再利用を始めた時期であり、Q&A 履歴が他者に参照されても自分がどのような知識や情報に関心があるかを認識してもらうにとどまり、自分がどれだけの能力や知識を持っているかまでははっきりと認識してもらえないため、コミュニティにおけるアイデンティティがほとんど確立されていない時期である。

アイデンティティ確立期: 質問や再利用を繰り返すことでコミュニティからアイデンティティを徐々に認識してもらう時期。アイデンティティが認識されると、質問や再利用によりある分野に関する知識を習得している程度が推測されるため、その程度に応じて質問を受ける機会も増えてくる。最終的には、ある分野に関する第一人者となり、コミュニティの中で絶対的なアイデンティティを確立する。

アイデンティティ喪失期: 同じ分野に関して知識の習得を行ってきた人のアイデンティティが確立されることにより、相対的にアイデンティティが失われる時期。

高度習得期: 自己のアイデンティティが失われるのを防ぐために、それまでの知識に加え、異なる分野に関する知識を習得し、複合的なより高度な知識を習得していく時期。他分野の知識を習得し直すため、最初はアイデンティティがさらに失われるが、複合的な知識がある程度習得されると、二つの分野においてアイデンティティが認識されるようになるため、アイデンティティは徐々に回復していく。

アイデンティティ回復期: 複数の分野の知識を習得し、複合的な知識を習得していくと、複数分野にまたがる複雑な質問に回答できるようになる。複数分野の知識を習得

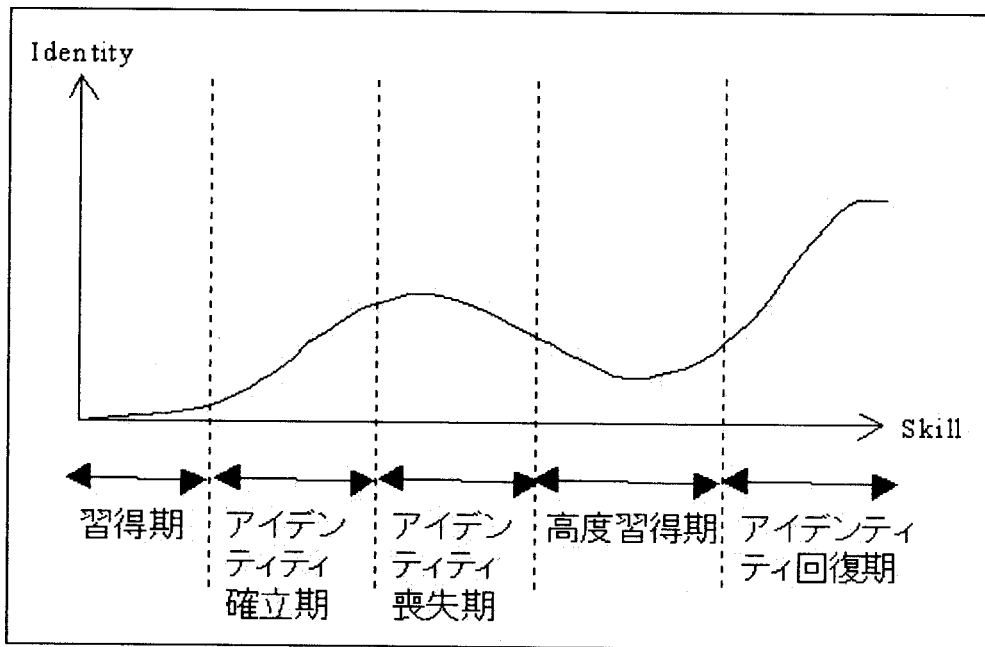


図 7.2: コミュニティ知識ベース環境におけるアイデンティティ確立の過程

することは困難であるため、最初に確立したアイデンティティ以上に高いアイデンティティが確立される。

このような過程の繰り返しを通じて、コミュニティにおいてアイデンティティを確立していく。また、その過程において他者に回答という形で知識を提供することにより、他者のスキルが伸びるため、自分がアイデンティティを確立していったのと同様に他者のアイデンティティの確立を助ける。すなわち、個々人が他者とのインタラクションを通じた学習によりアイデンティティを確立していく過程が他者のアイデンティティの確立をも促すため、コミュニティの形成につながる。

7.3 今後の課題

コミュニティ知識ベース環境では、(1) 電子メールによるシステムエンジニア間で Q&A のやり取りを行うこと、(2) 質問者が回答者を指定すること、(3) Q&A とその履歴をコミュニティに公開すること、の三つを前提条件としている。このため、実際のソフトウェア開発において生じた問題をシステムエンジニアが解決するために必要となる知識や情報を提供する枠組みとしては以下のような問題を解決する必要がある。まず、(1) の前提により、質問した人はいつ回答が得られるか分からないという問題が生じる。これはコミュニティに参加する人は実務の範囲外において、あくまでもコミュニティの活動に関心を持った人々が参加するためである。すなわち、指定された回答者がいつまでに回答を行わなければいけないのかという強制力がない。この問題を解決するためには、回答が形成される迅速性を回答者の評価としてフィードバックすることが有効であると考えられる。例えば、質問を受けてから相手に回答を送付するまでの期間に応じてポイント

を回答者に与え、累積したポイントをコミュニティに公開するといった仕組みを導入することにより解決することが可能である。

(2)の前提により、一人に質問が集中する可能性があるという問題が生じる。この問題を解決するためには、質問が集中した人と同じような知識領域に関心がある人々に、それぞれの知的スキルの向上を考慮して、質問を振り分けるような仕組みを導入することにより、コミュニティ全体の能力の向上を図りつつ、問題を解決することが可能となる。(3)の前提により、蓄積されるQ&Aやその履歴が増加した場合にどのようにそれらの情報を参加者に見せることにより参加者が必要な情報を迅速に獲得することを支援するかが問題となる。コミュニティの個々の参加者の関心を可視化する技術 [51] などを取り入れ、学習過程の遷移やアイデンティティの確立度合いなどを可視化することにより、このような問題を解決することが可能である。

7.4 まとめ

本章では、今後のソフトウェア開発においては学習する組織の実現が求められることについて論じるとともに、本論文で提案したコミュニティ知識ベース環境が学習する組織の実現に有効であること、および本研究の課題について論じた。

ソフトウェア開発に要求される知識や情報はますます多様化・複雑化してきており、従来のようにOJTや研修といった過去の事例に学ぶという教育方法では組織活動の存続・発展は望めないことについて論じた。このような状況に対処するためには、組織に属する個々人が周囲の状況を迅速かつ適切に把握し、その状況を自分や組織の目的に合わせて変化させていく方法を学習していくような組織が必要であることについて論じた。このような学習する組織の実現において最も重要な能力がシステム思考であり、これは本論文で論じてきた問題状況を迅速かつ適切に明確化するための能力と知識であることについて論じた。

このような類似性があることから、学習する組織の実現における本論文で提案したコミュニティ知識ベース環境の有効性について、コミュニティ知識ベース環境における学習過程およびアイデンティティ確立過程という二つの観点から論じた。コミュニティ知識ベース環境における学習過程では、コミュニティ活動の観察を通じた関心の共有、Q&Aの再利用を通じた学習と間接的な他者との相互作用、他者への質問を通じたQ&Aの形成とそれによるアイデンティティの認識、そして他者への回答による直接的な他者の学習支援とそれによるアイデンティティの認識というようにコミュニティにおける立場を変えながらより深くコミュニティに貢献していくことにより、問題状況を迅速かつ適切に明確化する能力と知識を習得することが支援されていることについて論じた。アイデンティティ確立過程に関しては、個々人の能力や知識のレベルに応じてアイデンティティの確立がなされていくことについて論じた。

コミュニティ知識ベース環境では、(1)電子メールによるシステムエンジニア間でQ&Aのやり取りを行うことにより回答が迅速に獲得できることが保証されないこと、(2)質問者が回答者を指定することにより一人に質問が集中する可能性が高いこと、(3)Q&Aとその履歴をコミュニティに公開することにより必要な情報を見つけ出すことが困難になること、といった問題が生じることについて論じた。それぞれの問題は(1)回答者へのインセンティブを検討すること、(2)能力や知識のレベルに応じた質問の振り分け、(3)情報の可視化技術の導入、により解決することが可能であることについて論じた。

次章では、実際のソフトウェア開発において問題解決のノウハウや経験を収集し蓄積・共有することの困難さについて論じるとともに、本研究の意義について論じる。

第8章 結論

本論文ではシステムエンジニアの問題解決に必要な知識や情報の迅速な提供を行うための方法として、回答作業を専門化しコンピュータ上の問題状況をそのまま添付可能な電子メールを用いることによる Q&A 方式に基づく枠組みを提案し、株式会社エヌ・ティ・ティ・データにおける事例に基づく本枠組みの効果検証について論じるとともに、検証結果に基づく本枠組みの改善のためのプロトタイプシステムである「コミュニティ知識ベース環境」の仕組みについても論じた。そして、このような改善を含めた本枠組みの有効性について、関連研究および今後のソフトウェア開発に求められる「学習する組織」の実現という観点から論じた。

本研究の背景として、ソフトウェア開発に要求される知識や情報が本来的に膨大なものであるとともに、近年の情報通信技術の発達や開発形態の変化などによりますます増加しつつあり、こうした傾向によりソフトウェア開発の質や効率が低下することについて論じた。

本研究はシステムエンジニアの問題解決に必要な知識や情報を Q&A を通じて効率的に提供するアプローチを提案することにより、ソフトウェア開発の質や効率を向上させることを目的とすることについて論じた。そして、ソフトウェア開発に必要なとされる知識や情報に着目した関連研究として、

- (a) ソフトウェア開発に必要な知識や情報に対するニーズを削減する
- (b) ソフトウェア開発において生成される知識や情報をより多く明示化し記録する
- (c) ソフトウェア開発において生成された知識や情報を蓄積し効率的に再利用する
- (d) システムエンジニア間のコミュニケーションを支援する

という四つのアプローチに基づいてソフトウェア開発の質と効率を向上させることを目的とした研究についてその概要を説明し、それぞれの研究の欠点を本論文で提案した枠組みにより補うことが可能であることについて論じた。

現状のソフトウェア開発では、個々のシステムエンジニアは必要な知識や情報を獲得するために他のシステムエンジニアに対面で質問するケースが多く、これにより以下の三つの弊害が生じることについて論じた。

第一の弊害: 他のシステムエンジニアに直接質問することにより質問された側の作業は強制的に中断されるために生じる質問された側の作業効率の低下

第二の弊害: 他のシステムエンジニアに直接質問するためには必要な情報を持っていそうな人を見つけ出すというオーバーヘッドが増大することによる質問する側自身の作業効率の低下

第三の弊害: 対面等で質問と回答がやり取りされてもそうした知識が記録されないために同じような質問と回答が他のシステムエンジニア間で繰り返されることによる組織全体におけるコミュニケーション効率の低下

一方で、他者への質問には問題状況の迅速かつ適切な明確化における以下の三つの効果があることについて論じた。

第一の効果: 問題状況に即した知識や情報を迅速に獲得できること

第二の効果: 他者の異なる観点を含む回答を獲得することにより問題状況を再認識できること

第三の効果: 問題状況を質問として表現することにより問題状況を再認識できること

そこで、本研究では、他者への質問による三つの弊害を回避しつつ、こうした三つの効果を高めるための枠組みとして、電子メールの利用と回答作業の専門化とを相互補完的に組み合わせるアプローチを提案した。そして、この枠組みにより前述した三つの効果が高められるとともに、

第四の効果: Q&A の再利用性改善による組織全体のコミュニケーション効率の向上

という第四の効果が期待できることについて論じた。

本枠組みによる四つの効果を検証するにあたり、株式会社エヌ・ティ・ティ・データで開発・運用している技術支援システムを対象に行った調査について論じた。その結果、四つの効果はいずれも検証されたこと、および、これらの効果をさらに高めるためには本研究において明らかにした質問の記述構造を Q&A を形成する場合、すなわち他者へ質問する場合にも、再利用する場合、すなわちデータベースを検索する場合にも、意識することを支援する必要があることについて論じた。

このような検証結果に基づいて本枠組みを Q&A の形成と再利用の統合的支援という観点から改善するために二つの調査を行った結果、質問を行う場合にも検索を行う場合にも質問の記述構造を意識する必要があるということが判明したことについて論じた。そして、それを支援するためのプロトタイプシステムである「コミュニティ知識ベース環境」の枠組みについて論じた。

また、今後のソフトウェア開発において学習する組織の実現が求められていることについて論じるとともに、「コミュニティ知識ベース環境」がネットワーク上での Q&A を通じた学習するコミュニティ形成を支援していることにより、システムエンジニアの学習する組織の実現に有効であることについて論じた。

本研究は Q&A を用いてソフトウェア開発において生じた実際の問題解決の経験やノウハウを収集し再利用することによりソフトウェア開発の質や効率を高めることを目的としている。本論文で紹介した株式会社エヌ・ティ・ティ・データで開発・運用している技術支援システムでは個々のプロセスにおける個々のシステムエンジニアが行った問題解決の経験やノウハウの事例として Q&A が蓄積されている。いわばボトムアップで経験やノウハウといった実践的な知識を収集し再利用を促進する試みである。

現在の知識管理の主流は CKO (Chief Knowledge Officer) など知識管理を専門に行う役職を設置し、その役職の人が収集し管理する知識を決定して現場から知識を収集してくるというトップダウン型の方法である。この方法では効率的に知識を収集し管理することができる。しかし、ソフトウェア開発のように用いられる技術が常に変化するとともに開発手法や開発形態が絶えず見直されている産業においては、よほど先見性のある知識管理の責任者でなければこのようなトップダウン型の知識管理では対応することは困難である。

また、トップダウン型の知識管理方法では実際の問題解決に有益な知識を収集することは困難である。問題解決に有益な知識とは類似した問題状況における多数の失敗事例である。いくら優れた解決策があっても問題状況が少しでも違えば同じような成功が期待できない。一方、類似した問題状況における様々な失敗事例を見ることにより、失敗しそうな選択肢をつぶしていくことができ、結果的により有効な問題解決が行える。しかし、このような現場の失敗事例が組織の上部に上げられることは期待できない。自分

を評価する権限を持っている人に自分の評価にマイナスとなる情報を提供する人はいないからである。

本研究でのアプローチのように個々のシステムエンジニアが抱える問題解決を支援しながらボトムアップ的にソフトウェア開発における個別の知識を収集し管理することは、このような変化の激しい産業では有効な知識管理の方法である。また、現場の個々の問題解決の失敗事例をトラブルシューティングという形で集めることができ、人名等を隠蔽することでその人の評価に影響ないようにそうした失敗事例を共有し再利用することも可能である。さらに、このような個別の知識を収集し、再利用の少ないものは削除していくことにより、より大きな単位でのソフトウェア開発における問題に対処できる解決策としてまとめあげていくことにより、一人の人が意思決定をする従来のトップダウン型の知識管理とは違い、リスクの少ないトップダウン型の知識管理も可能になる。

本研究に基づき、ソフトウェア開発におけるより広範囲な知識や情報を収集する仕組みの導入も進められている。技術支援システムの改善のためのフレームワークであるコミュニティ知識ベース環境では、形成した Q&A に対する他者の評価をその Q&A を形成した質問者や回答者にフィードバックすることにより、より適切な質問や回答が形成できることについて論じた。このような仕組みは、現在、株式会社エヌ・ティ・ティ・データにおける組織活動において生成されたあらゆる知識や情報を収集し共有するシステムに取り入れられている。イントラネット上に全社員が参照可能な Web ページを設置し、そこに自分の形成した電子ドキュメント等を登録してもらう。そして、登録 1 件あたりにつき 1 ポイントが付加され、参照されるたびに 3 ポイントが付加される。そして、トータルのポイントで誰がもっとも情報提供を行っているか、またそれが再利用されているかを順位付けし、Web ページ上で公開される。このような仕組みが及ぼす形成される知識や情報の質的な効果に関してはまだ検証が行われていない。しかし、それまで一人の人に閉じていた、もしくはその人の属する小さなプロジェクト内に閉じて共有されていた知識や情報が、企業全体の知識や情報として共有することが可能になったこと、およびそれまで個人を中心とした狭い範囲に閉じていた知識や情報が企業全体としても参照される価値があるということ個人個人が認識することが可能となり、登録される件数が増加してきていることなど、企業における情報共有に与える効果は大きいと考える。しかし、共有される知識や情報がどのような問題状況において形成されたのかを知ることは困難であり、今後は Q&A のようにそれが形成された問題状況とその解決策の事例として知識や情報を蓄積し共有できる仕組みを検討していく必要がある。

本研究の社会的意義は次のとおりである。従来のソフトウェア開発の支援に関する研究は技術駆動型であった。ワークステーションやパーソナルコンピュータの性能向上と普及に伴う CASE(Computer Aided Software Engineering) ツールやグループウェアなどが大量に導入され、技術の急速な進歩に伴い導入されるツールも日々刻々と変化してきた。生産性は向上してきているのかもしれないが、その結果はシステムエンジニアの平均寿命はあらゆる職種の中で最も低いものの一つといわれるに至る。

本研究はソフトウェア工学的な観点から現状のソフトウェア開発における課題について考察し、その課題の原因に関して、個々のシステムエンジニアが他のシステムエンジニアを助けることにより自らの首を絞めないように、また組織全体の効率を低下させることがないように、人と人との質問と回答のやり取りにどのような効果があるか、どのような技術を用いればよいかを認知科学的な観点から考察したことに意義がある。このようにコンピュータツールを利用する人の観点からその効果を明確にしているため、利用する人の観点からコンピュータツールを評価することが可能となり、より人に優しいコンピュータツールへの改善が可能になる。

このような可能性はソフトウェア開発に携わるシステムエンジニアだけのものではない。システムエンジニアが構築したシステムを利用するユーザに対する可能性でもある。単にユーザの望む要求を正確にシステムとして実現するのではなく、そのシステムを利用するユーザにとって本当に優しいシステムなのかどうかを検討した上でユーザの

要求を実現することが真にユーザ満足度の高いソフトウェア開発であると考え、そのようなソフトウェア開発を行うためにも、本研究のように認知科学的な観点からシステムの利便性を検討していくことは有意義である。

今後ますますコンピュータシステムは社会に普及していく、そうした中で人に優しいコンピュータシステムを構築していくことはソフトウェア開発に携わるシステムエンジニアや企業の責任である。また、人に優しいコンピュータシステムを要求することはユーザの権利である。このような人に優しいコンピュータ社会を築き上げていくための枠組みとして本研究のような視点は重要であると考え。

一方で、本研究はコンピュータシステムの可能性を示唆するものでもある。5.6.1節で論じたように、人がデータベースを検索する場合、得られた検索結果が検索条件を入力した時に意図していたものでなくても、それが偶然役に立つと認識される確率は非常に高い。人間対人間の場合、このように相手は意図していないが相手の潜在的な問題にとって有益である情報を提供することは大変な労力を要する。なぜなら、人間同士では相手の持っている暗黙知を共有することができ、より正確に相手の置かれている状況を共有することが可能だからである。このため質問された側は自分の持っている膨大な知識や情報から相手の置かれている状況に即したものだけを選択して迅速に提供することが可能となる。しかし、まったく相手が意図していない有益な知識や情報を提供することは困難である。逆にコンピュータシステムは入力された検索条件をあるアルゴリズムにしたがって計算し、該当する情報だけをユーザに検索結果として提示しており、ユーザの置かれている状況やユーザが抱えている問題を考慮しているわけではない。このことにより、偶発的に有益な情報を得ることのできる可能性が生じる。このような偶発的に獲得できた情報は人間の創造性を高めるために有効であると考えられ、本研究はコンピュータシステムのこうした不器用さを利用することにより人のより創造的な活動の支援が可能となることを示していると言える。

謝辞

本研究を遂行するにあたり多くの方々に御助力をいただきました。御助力いただきましたすべての方々に深く感謝いたします。

会社での業務と併行して本研究を行う機会と資金的な援助を賜りました株式会社エヌ・ティ・ティ・データ代表取締役会長である神林留雄氏、同社代表取締役社長である青木利晴氏、同社代表取締役副社長である河合輝欣氏、同社取締役で技術開発本部長である荒川弘熙氏に深く感謝いたします。小生が同社において所属します技術開発本部情報科学研究所の所長である中村太一氏、同部長である松田栄之氏、同課長である吉谷文徳氏には業務上の指導とともに本研究を遂行するためのご指導もいただきました。深く感謝いたします。本研究を実施する実質的な機会を賜りました同社地域事業本部事業部長である菊田道夫氏、同社 COE システム本部長である秋山雅俊氏と青木弘之氏、同課長代理である橋爪宗信氏には、入社以来公私において多大なる御指導御鞭撻を賜りました。深く感謝いたします。本研究において同社で開発・運用している技術支援システムの評価検証を行うにあたっては、同社 COE システム本部長である西部俊憲氏、同課長である荒木勉氏と油川健氏、そして牧嘉登氏には快く Q&A データを提供していただきましたとともに、分析にあたっては多大なる御助言御助力を賜りました。深く感謝いたします。同社技術開発本部企画部部長の高橋淳一氏、同社技術開発本部情報科学研究所課長代理である梶浦正規氏と山岡正輝氏、同僚の佐藤慎一氏と坂巻資浩氏には、研究内容の検討等において貴重な御意見をいただくことができました。皆様の忌憚なき御意見がなければ本研究の遂行は困難であったと思います。深く感謝いたします。ここには書き切れなかった株式会社エヌ・ティ・ティ・データの先輩や同僚、後輩の皆様に深く感謝いたします。

本論文の審査を快くお引き受けいただくとともに様々な観点から貴重な御助言を賜りました。奈良先端科学技術大学院大学情報科学研究科教授である井上克郎氏、小山正樹氏、関浩之氏、同大学助教授である松本健一氏、中小路久美代氏に深く感謝いたします。中小路久美代氏には、同大学博士前期課程に在学以来貴重な御意見と御指導を賜りました。同氏との出会いがなければ認知科学という曖昧模糊とした、しかしこれからのコンピュータ社会において最も重要な学問に足を踏み入れることはなかったと思います。このような研究を企業で行うことは荊の道を歩むようなものですが、荊の中に美しいバラがあることを信じて一步一步前進していきたいと思います。同講座の助手であられました現慶應義塾大学工学部専任講師である高田真吾氏には貴重な御意見をいただきましたとともに、様々な御助力をいただきました。深く感謝いたします。同講座の現助手である蔵川圭氏には貴重な御意見をいただくとともに、手料理までいただくことができました。深く感謝いたします。同講座に博士後期課程の社会人学生として入学以来懇意にさせていただくとともに、数々の貴重な御意見御助力をいただきました同講座博士後期課程三年の山本恭裕氏に深く感謝いたします。同氏とは 1999 年 5 月に行われた米国での研究発表において起床をともにし、夜が更けるのも忘れるほどの煙草の煙の中で、深遠なる哲学的議論を交えることができました。卒業生を含めた同講座の皆様には、時々しか姿を見せず、あまり講座の役にも立たない小生に親切にいただきましたとともに

に、数々の御助力をいただきました。深く感謝いたします。

1999年6月に盛岡で行われた Software Symposium'99 において、数少ない同年代ということで懇意にさせていただきました北陸先端科学技術大学院大学情報科学研究科情報システム学専攻ソフトウェア計画構成学講座の助手である村越広亨氏には、本研究を協調的学習という観点から捉えなおす貴重な御助言をいただきました。深く感謝いたします。現在 NTT コミュニケーション科学基礎研究に勤務する平尾努氏とは、大学院で同じ研究室であり会社の部署も同じであったこともあり、公私にわたって忌憚ない議論を行うことができました。数値的な評価が困難な小生の研究に対して、同氏の研究成果を数値的に評価するという姿勢には触発されることも多々ありました。深く感謝いたします。

奈良先端科学技術大学院大学に近いこともあり、本研究の遂行において同大学を訪れた際にいろいろお世話になりました義父の吉田邦男氏と義母の吉田靖子氏に感謝いたします。孫ともども今後ともどうかよろしくお願い申し上げます。

博士号の取得を志すに至ったのも、小生が生まれたときから学問の道を志すよう育ててくれました父・池田純二と母・池田惇子のお陰です。深く感謝いたします。

本研究を遂行するにあたっては、1998年2月3日に生まれた長男の貴希、そして2000年1月22日に生まれた長女の瑞希、この二人の笑顔がどれほど励みになったか知れません。また、貴希と瑞希は人間が本来的に持っている能力の素晴らしさに気付かせてくれました。認知科学の重要性を主張しつづけてこられたのもこの二人のお陰です。二人ともありがとう。そしてこれからもよろしく。

最後に、二人の子育てで最も辛い時期に、朝早くから夜遅くまで、そして週末も、本研究の遂行のために家を空けることの多かった小生を、不平一つこぼさずにいつも笑顔で励まし送り出してくれた妻・理佐子に本論文を捧げます。

関連図書

- [1] Abigail, J.S., Remote conversations: the effects of mediating talk with technology, *Journal of Human-Computer Interaction*, Vol.10, pp.401-444, 1995
- [2] Adams, E.W., Honda, M. and Miller, T.C., Object management in a CASE environment, *Proceedings of the 11th ICSE*, IEEE Computer Society Press, pp.154-163, 1989
- [3] Alexander, C., *The Synthesis of Form*, Harvard University Press, 1964
- [4] Artzt, F. and Amour-Thomas, E., Development of a cognitive-metacognitive framework for protocol analysis of mathematical problem solving in small groups, *Cognition and Instruction*, Vol.9, No.2, pp.137-175, 1992
- [5] Bagert, D.J., Hilburn, T., Hislop, G., Lutz, M., McCracken, M. and Mengel, S., *Guidelines for Software Engineering Education, Version 1.0*, CMU/SEI-99-TR-32, ESC-TR-99-002, 1999 (available at <http://www.sei.cmu.edu/collaborating/ed/workgroup-ed.html>)
- [6] Barbara, L.C., Fish, S. and Kraut, R.E., Expressive richness: a comparison of speech and text as media for revision, in *CHI'91 Proceedings*, pp.21-26, 1991
- [7] Basili, V.R., Caldiera, G. and Cantone, G., A reference architecture for the component factory, *ACM Transactions on Software Engineering and Methodology*, Vol.1, No.1, pp.53-80, 1992
- [8] Basili, V.R., Caldier, G. and Rombach, H.D., The experience factory, in *Encyclopedia of Software Engineering*, John-Wiley & Sons, pp.470-476, 1994
- [9] Boehm, B.W., A spiral model of software development and enhancement, *IEEE Computer*, Vol.21, No.5, May, pp.61-72, 1988
- [10] Boehm, B.W., A spiral model of software development and enhancement, in Dorfman, M. and Thayer, R.H. (eds.), *Software Engineering*, IEEE Computer Society, pp.415-426, 1997
- [11] Brothers, L. et al., ICICLE: groupware for code inspection, *CSCW'90, Proceedings of ACM*, pp.169-181, 1990
- [12] Brown, A., Metacognition, executive control, self-regulation, and other more mysterious mechanisms, in Weinert, F.E. and Kluwe, R.H. (eds.), *Metacognition, Motivation, and Understanding*, Lawrence Erlbaum Associates Publishers, pp.65-264, 1987
- [13] Buckley, F.J., *Implementing Configuration Management: Hardware, Software, and Firmware*, Second Edition, IEEE Computer Society Press, 1996

- [14] Clarke, E.M., Wing, J.M., et al, Formal methods: state of the art and future directions, *ACM Computing Surveys*, Vol.28, No.4, pp.626-643, 1996
- [15] Conklin, J. and Begeman, M.L., gIBIS: a hypertext tool for exploratory policy discussion, *Proceedings of CSCW'88*, ACM, pp.140-152, 1988
- [16] Curtis, B., et.al, On building software process models under the lamppost, *Proceedings of the 9th ICSE*, IEEE Computer Society Press, pp.96-103, 1987
- [17] Daft, R.L. and Lengel, R.H., Organizational information requirements, media richness and structural design, *Management Science*, Vol.32, No.5, pp.554-571, 1986
- [18] Demarco, T. and Lister, T., *Peopleware : Productive Projects and Teams*, 2nd Edition, Dorset House, 1999
- [19] Dillenbourg, P. and Self, J., Designing human-computer collaborative learning, in O'Malley, C. (eds.), *Computer Supported Collaborative Learning*, Springer-Verlag, pp.245-264, 1995
- [20] Dorfman, M. and Thayer, R.H., *Software Engineering*, IEEE Computer Society Press, 1997
- [21] Draper, S.W., The nature of expertise in UNIX, *Proceedings of INTERACT'84*, IFIP Conference on Human-Computer Interaction, Elsevier Science Publishers, pp.182-186, 1984
- [22] Ellis, C.A., Gibbs, S.J. and Rein, G.L, GROPUWARE - some issues and experiences, *Communication of ACM*, Vol.34, No.1, pp.39-55, 1991
- [23] Fischer, G. and Girgensohn, A., End-user modifiability in design environment, *Human Factors in Computing Systems, CHI'90*, ACM, pp.183-191, 1990
- [24] Fischer, G., User modeling in human-computer interaction, Contribution to the 10th Anniversary Issue of *Journal of User Modeling and User-Adapted Interaction (UMUAI)*, 2000 (in preparation)
- [25] Fischer, G., Henninger, S. R., and Redmiles, D. F., Cognitive tools for locating and comprehending software objects for reuse, *Proceedings of the 13th ICSE*, pp.318-328, IEEE Computer Society Press, 1991
- [26] Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G. and Sumner, T., Embedding critics in design environments, in Maybury, M.T. and Wahlster, W. (eds.), *Readings in Intelligent User Interfaces*, Morgan Kaufmann, pp.537-561, 1998
- [27] Freedman, D.P., and Weinberg, G.M., *Handbook of Walkthroughs, Inspections, and Technical Reviews*, Little, Brown and Company, 1982
- [28] Gamma, E., Helm, R., Johnson, R. and Vissides, J., *Design Patterns - Elements of Reusable Object-Oriented Software*, Professional Computing Series, Addison-Wesley, 1995
- [29] Goodlad, S. and Hirst, B., *Peer Tutoring: A Guide to Learning by Teaching*, Kogan Page, 1989
- [30] Grudin, J.: Why CSCW applications fail: problems in the design and evaluation of organizational interfaces, *CSCW '88*, *Proceedings of ACM*, pp.85-93, 1988

- [31] Hillery, G.A., Definitions of Community: Areas of Agreement, *Rural Sociology*, Vol.20, 1995
- [32] Ikeda, F., Sakamaki, M., Aoki, H., Takada, S. and Nakakoji, K., Improving software quality from the viewpoints of knowledge production and the work environment, *The 6th European Conference on Software Quality*, pp.306-317, April, 1999
- [33] Jazabek, S. and Huang, R., The case for user-centered CASE tools, *Communications of the ACM*, Vol.41, No.8, pp.93-99, August, 1998
- [34] Johansen R., *Computer Support for Business Teams*, New York, 1988
- [35] Johnson, M., *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*, The University of Chicago Press, 1987
- [36] Kedzirski, B.J., Communication and management support in system development environments, in Greif, L. (eds.), *Computer Supported Cooperative Work: A Book of Reading*, Morgan Kaufman, 1988
- [37] Kontio, J.A., A case study in applying a systematic method for COTS selection, *Proceedings of the 18th ICSE*, IEEE Computer Society Press, 1996
- [38] Lakoff, G., *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*, The University of Chicago Press, 1987
- [39] Lee, J., Extending the Potts and Bruns model for recording design rationale, *Proceedings of the 13th ICSE*, IEEE Computer Society Press, pp.114-125, 1991
- [40] Markus, M.L. and Connolly, T., Why CSCW applications fail: problems in the adoption of interdependent work tools, *CSCW'90 Proceedings*, ACM, pp.371-380, 1990
- [41] Markus, M.L., Electronic mail as the medium of manager choice, *Organization Science*, Vol.5, No.4, pp.502-527, 1994
- [42] Matsumoto, Y. and Ajisaka, T., A data model in the software project database KyotoDB, *Advances in Software Science and Technology*, 日本ソフトウェア科学会編, Vol.2, pp.103-121, 1990
- [43] McCarthy, J.C. and Monk, A.F., Measuring the quality of computer-mediated communication, *BEHAVIOR & INFORMATION TECHNOLOGY*, Vol.13, No.5, pp.311-319, 1994
- [44] Melver, R.M., *Community: A Sociological Study Being an Attempt to Set Out the Fundamental Laws of Social Life*, International Specialized Book Service, pp.124, 1970
- [45] Miyake, H. and Norman, D.A., To ask question, one must know enough to know what is not known, *Journal of Verbal Learning and Verbal Behavior*, Vol.18, pp.357-364, 1979
- [46] Miyake, N., Constructive interaction and the iterative process of understanding, *Cognitive Science*, Vol.10, pp.151-177, 1986
- [47] Morisio, M., Seaman, C.B., Parra, A.T., Basili, V.R., Kraft, S.E. and Condon, S.E., Investigating and improving a COTS-based software development process, *Proceedings of ICSE2000*, IEEE Computer Society Press, pp. 2000

- [48] K. Nakakoji, Increasing Shared Understanding of a Design Task between Designers and Design Environments: The Role of a Specification Component, PhD dissertation, Department of Computer Science, University of Colorado, 1993 (Also available as TechReport CU-CS-651-93)
- [49] Nakakoji, K. and Takada, S., Computer support for externalizing design rationale, Proceedings of Design & Systems Conference'98, JSME, Tokyo Japan, November, pp.596-599, 1998 (in Japanese)
- [50] Nakakoji, K., Yamamoto, Y., Suzuki, Y., Takada, S. and Gross, M.D., Beyond critiquing: using representational talkback to elicit design intention, Knowledge-Based Systems Journal, Vol.11, No.7-8, Elsevier Science, pp.457-468, 1998
- [51] Nakakoji, K., Yamamoto, Y., Sugiyama, K. and Takada, S., Finding the "right" image: visualizing relationships among persons, images and impressions, in A. Sutcliffe, J Ziegler, P. Johnson (Eds.), Designing Effective and Usable Multimedia Systems, Kluwer Academic Publishers, pp.91-102, 1998
- [52] Nakayama, Y., Manabe, T. and Takebayashi, Y., Development of knowledge/information sharing system, Transaction of Information Processing Society of Japan, Vol.39, No.5, pp.1186-1194, 1998
- [53] Nonaka, I. and Takeuchi, Y. The Knowledge Creating Company, Oxford University Press, 1995
- [54] Papert, S., Mindstorms: Children, Computers, Powerful Ideas, Harvester Wheatsheaf, 1980
- [55] Parikh, G., Handbook of Software Maintenance, John-Wiley & Sons, 1986
- [56] Piaget, J., Judgement and Reasoning in the Child, Harcourt Brace, New York, 1928
- [57] Pigoski, T.M., Practical Software Maintenance: Best Practices for Managing your Software Investment, John-Wiley & Sons, 1997
- [58] Polanyi, M., The Tacit Dimension, Doubleday, Garden City, NY., 1966
- [59] Potts, C., A generic model for representing design methods, Proceedings of the 11th ICSE, IEEE Computer Society Press, pp.217-226, 1989
- [60] Pree, W., DesignPatternsfor Object Oriented Software Development, Addison-Wesley, 1995 (佐藤 啓太, 金澤 典子訳, デザインパターンプログラミング, トップアン, 1996)
- [61] Rittel, H.W.J, Second-generation design methods, in Cross, N. (eds.), Developments in Design Methodology, John-Wiley & Sons, pp.317-327, 1984
- [62] Rocchio, J.J., Relevance feedback in information retrieval, The SMART Retrieval System, Prentice-Hall, pp.313-323, 1971
- [63] Roschell, J. and Teasley, S.D., The construction of shared knowledge in collaborative problem solving, in O'Malley, C. (eds.), Computer Supported Collaborative Learning, NATO ASI series, Vol.F-128, Springer-Verlag, pp.69-97, 1994

- [64] Sakamaki, M., Ikeda, F., Takada, S. and Nakakoji, K., Measuring work factors: a case study to identify relationships among work, working style and work environment, Second International Workshop on Cooperative Buildings: Integrating Information, Organizations and Architecture (CoBuild'99), Springer-Verlag, pp.164-176, 1999
- [65] Salomon, G., No distribution without individuals' cognition: a dynamic interactional view, in Salomon, G. (eds.), Distributed Cognition: Psychological and Educational Considerations, Cambridge University Press, pp.111-138, 1993
- [66] SWEBOK Knowledge Area Description for Software Requirement, Version 0.6, 2000 (available at <http://www.swebok.org/>)
- [67] Schoen, D.A., The Reflective Practitioner: How Professionals Think in Action, Basic Books, 1983
- [68] Schoen, D.A., Designing as reflective conversation with the materials of a design situation, Knowledge-Based Systems Journal, Vol.5, No.1, pp.3-14, 1992
- [69] Senge, Peter M. The Fifth Discipline: The Art and Practice of the Learning Organization, Doubleday, 1990
- [70] Senge, Peter M, et al. The Fifth Discipline Fieldbook: Strategies and Tools for Building a Learning Organization, Doubleday, 1994
- [71] Shingaki, N. and Nojima, H., Social aspects of dependency in navigation: route guidance using mobile phone with location information, CogSci98 proceedings, the 20th Annual Conference of the Cognitive Science Society, pp 1264, 1998.
- [72] Simon, H.A., The Sciences of the Artificial, The MIT Press, 1981
- [73] Snodgrass, A.S. and Coyne, R., Is designing hermenewtical?, Technical report, Department of Architectural and Design Science, University of Sydney, 1990
- [74] Suchman, L.A., Plans and Situated Actions, Cambridge University Press, Cambridge, 1987
- [75] Vygotsky, L.S., Mind in Society: The Development of the Higher Psychological Processes, Harvard University Press, 1930 (re-published in 1978)
- [76] Winograd, T. and Flores, F., Understanding Computers and Cognition: A New Foundation for Design, Ablex Publishing Corporation, 1986
- [77] Yakemovic, K.C.B and Conklin, E.J., Report on a development project use of an issu-based information system, CSCW'90, Proceedings of ACM, pp.105-118, 1990
- [78] Yakimovich, D., Bieman, J.M. and Basili, V.R., Software architecture classification for estimating the cost of COTS integration, Proceedings of the 21st ICSE, IEEE Computer Society Press, pp.296-302, 1999
- [79] Yamakami, T., Multi-user shared failure analysis of organizational information system, Transaction of Information Processing Society of Japan, Vol.39, No.10, pp.2899-2906, 1998
- [80] 稲垣佳代子, 「認知への動機付け」, 認知心理学講座 4 「学習と発達」, 東京大学出版会, 1962

- [81] 亀田靖浩, 大規模ソフトウェア開発の現状と課題, 電子情報通信学会誌, Vol.74, No.5, pp.457-460, 1991
- [82] 斎藤信男 (編), ワークステーションと分散処理, 「コンピュートロール」, Vol.33, コロナ社, 1991
- [83] 垂水浩幸, CS と CW の統合に関する考察～CW から CW へ, そしてまた CS へ～, 情報処理学会, グループウェア研究会, Vol.27, No.7, pp.37-42, 1998
- [84] 垂水浩幸, 分散開発環境: グループウェアのソフトウェア開発への応用, 情報処理, Vol.33, No.1, pp.22-31, 1992
- [85] 青山幹雄, 分散開発環境: 新しい開発環境像を求めて, 情報処理, Vol.33, No.1, pp.2-13, 1992
- [86] 青山幹雄, コンピュータが消える日: インターネット時代のソフトウェア, 情報処理, Vol.41, No.5, pp.523-527, 2000
- [87] 石井裕, 「CSCW とグループウェア」, オーム社, 1994
- [88] 池田文人, 山本恭裕, 高田眞吾, 中小路久美代, コミュニティ知識ベース環境の構築へ向けての知識の形成と利用に関する調査と分析, 情報処理学会論文誌, Vol.40, No.11, pp.3887-3895, 1999
- [89] 中島秀之, 認知の状況依存性と分散認知, 「認知科学の発展」, Vol.7, 日本認知科学会編, 講談社, pp.1-4, 1994
- [90] 長野宏宣, 分散開発環境の基盤技術, 情報処理, Vol.33, No.1, pp.14-21, 1992
- [91] 波多野誼余夫, 稲垣佳代子, 「知的好奇心」, 中公新書, 1973
- [92] 落水浩一郎, CASE とは, 日本ソフトウェア科学会, サマーチュートリアル「90年代の CASE」資料, 日本ソフトウェア科学会, pp.1-13, 1991
- [93] 村越広享, 電子メールを利用したコミュニケーションにおける会話の円滑性の評価尺度とその計算法について, 北陸先端科学技術大学院大学情報科学研究科, 博士論文, 1999

研究業績

査読付きフルペーパー

- 池田文人, 山本恭裕, 高田眞吾, 中小路久美代, コミュニティ知識ベース環境の構築へ向けての知識の形成と利用に関する調査と分析, 情報処理学会論文誌, Vol.40, No.11, pp.3887-3895, November, 1999

査読付き国際会議

- Ikeda, F., Sakamaki, M., Aoki, H., Takada, S. and Nakakoji, K., Improving software quality from the viewpoints of knowledge production and the work environment, The 6th European Conference on Software Quality, Vienna, Austria, pp.306-317, April, 1999
- Ikeda, F., Takada, S. and Nakakoji, K., Examining shared knowledge at production time and use time, Proceedings of the International Workshop on Collaboration and Mobile Computing (CMC'99), Panda, D. and Takizawa, M. (Eds.), IEEE Computing Society, Aizu-Wakamatsu, Japan, pp.94-99, September, 1999
- Sakamaki, M., Ikeda, F., Takada, S. and Nakakoji, K., Measuring work factors: A Case Study to Identify Relationships among Work, Working Style and Work Environment, Second International Workshop on Co-operative Buildings: Integrating Information, Organizations and Architecture (CoBuild'99), Springer, Pittsburgh, PA., pp. 164-176, October, 1999
- Ikeda, F., Yamamoto, Y., Takada, S. and Nakakoji, K., A study on the effect of a Q&A system on software development, 2nd World Conference on Software Engineering (2WCSE), Yokohama, Japan, September, 2000 (to appear)

査読付き国内会議

- 池田文人, 高田眞吾, 中小路久美代, 構築時と検索時という二つの視点からみたシステムエンジニア支援知識の「質」に関する考察, ソフトウェアシンポジウム 99, 盛岡, ソフトウェア技術者協会, pp.119-126, June, 1999

研究会／全国大会

- 池田文人, 稲餅正幸, 永瀬孝, 青木弘之, 協調分散オフィス構築手法, 第 55 回情報処理学会全国大会講演論文集, Vol.4, pp.131-132, 福岡, 1997
- 池田文人, 坂巻資浩, 稲餅正幸, 永瀬孝, 高田眞吾, 中小路久美代, 情報・空間・運用の統合による協調作業支援環境に関する考察, 情報処理学会グループウェア研究会, Vol.97, No.91, pp.55-60, 1997
- 坂巻資浩, 池田文人, 青木弘之, 情報・空間・運用の統合による協調作業支援環境評価について, 情報処理学会グループウェア研究会, Vol.98, No.80, pp.31-36, 1998
- 池田文人, 山本恭裕, 蔵川圭, 中小路久美代, コミュニティ形成を目的とする知識管理の枠組みに関する提案, 人工知能学会, 第 40 回人工知能基礎論研究会, SIG-FAI-9904, pp.91-96, 2000