第 2 分科会

今まで普通に行われていたコミュニケーション対策を疑う

Effective Solutions Come from Reconsideration the Ordinary Measures

主査:	板倉 稔	(株式会社 ビズモ)
副主査:	森本 勝美	(CECA ジャパン)
研究員:	森田 智幸	(ソニー株式会社)
	阿部 圭一	(富士ゼロックス株式会社)
	江田 友彦	(株式会社 大和コンピューター)
	正田 行雄	(株式会社 NTT データ三洋システム)
	高橋 敏浩	(株式会社 日立システムアンドサービス)
	中村 史郎	(株式会社 富士通北陸システムズ)
	二宮 孝之	(三菱電機コントロールソフトウェア株式会社)
	早川 勲	(株式会社 山武)
	藤中 孝康	(富士通マイクロエレクトロニクス株式会社)

概要

当研究会では、プロジェクトを円滑に進めるために、異文化コミュニケーション問題を研究した。しかし、異文化やコミュニケーションに起因する問題はほとんどなく、設計問題や正しくない方法論を使うことによって生じる問題が大半であった。

いくつかの問題は、ドキュメントに記載することで解決されている。しかし、ドキュメントの内容は、読まれないことや見落とされることが起きる。その結果、問題が生じる。

我々は、実プロジェクトでのコミュニケーションに関係すると思われていた問題を研究し、その結果、問題を考えるためのフレームを発見した。このフレームに従って、問題を考えた結果、今まで普通に行われていた行動にメスを入れると、本当に意味のある対策が立てられることが明らかになった。

Abstract

We studied cross-culture communication problems to perform projects in a stable manner. In our study, we found that there are few cross-culture problems but a lot of problems caused of design, incorrect methodology and so on.

Some of the measures for these problems are written in documents. But people sometimes overlook the writings so that problems will happen.

We studied problems in real project that concerned with communication. We found a framework to get measure for the problems. Also, we found that it is efficient to think according to the frame about what we ordinarily do to solve the problems.

1. はじめに

プロジェクトマネジメントにおいて、一番重要な要因はコミュニケーションであるという。元々日本では阿吽の呼吸よろしく、他人であっても似たような考え方をするもの、で通じているところが多々あった。しかしながら、国内においても多様な考え方をする人が多くなるに従い、コミュニケーションをとることが難しくなっている。それに加えてオフショア開発など、もともと全く考え方の異なる人たちとの異文化コミュニケーションを取りながら、プロジェクトを進める必要が出てきた。

このようなプロジェクトを進めるにあたり、プロジェクト実行時の問題を抽出した。共通に抱えている問題で多かったものは、「当たり前」として通用していたことが異文化には通用しないために、手戻りが発生していることである。各研究員の会社では、様々な再発防止策を講じてきているが、異文化コミュニケーション問題をなくすまでには至っていない。

そこで、『メンバに異文化を持った人がいるプロジェクトを円滑に実行する方法』を研究のテーマとした。

2. 現状把握

表1の「立場」・「環境」・「思考」が自分または自分の属する文化と差異がある場合に、「異文化」を認識する。

表 1: 「異文化」の層別と例

層別	具体例
立場	上司と部下、親会社と子会社、発注元と受注先、メーカーと顧客
環境	関東と関西、暑い地域と寒い地域
思考(考え方)	日本人と中国人、男と女、ハード開発者とソフト開発者

情報の発信者は立場、環境、思考に依存した情報を発信している。一方、情報の受信者も立場、環境、思考に依存して情報を理解している。このように、コミュニケーションには、少なからず「伝える」側に「受ける」側への暗黙の期待がある。両者間の期待のずれがコミュニケーション問題を引き起こしていると推測する。この推測に基づいて、各社は対策をとっている。しかし、暗黙の期待のずれを埋めきれていない。

3. 「暗黙の期待のずれを埋めるのに必要な要件」の検討

各研究員が今までに携わったプロジェクトで、「コミュニケーションに問題があった」と認識している事例とその問題点を表 2 に示す。表 2 の下線部分が、「コミュニケーション問題」と考えた理由である。

表 2: 事例概要とプロジェクトの問題認識

No	事例概要	プロジェクトの問題認識
1	設計工程で、顧客から要件定義の内容に不足があると指摘された	設計者側と顧客とで成果物の認識に隔たりがあることに気づかなかった **(認識の相違)**
2	システム利用開始後、設計時に合意していた表示色への、変更要求が発生した	クライアントの担当が変わったが、相手担当者間での引き継ぎミスがあり、合意内容が覆った **(伝達ミス)**
3	外注先が不要な情報を報告書に記載して顧客へ提出したため、顧客が混乱し苦情を受けた	不要な情報を記載しないように外注先の管理者に連絡していたが、実担当者には正確に伝わっていなかった**(伝達ミス)**

No	事例概要	プロジェクトの問題認識
4	発注先に対して不良箇所を指摘したが、類似不良が多発した	発注元は発注先が 1 つの不良から類似展開すると思っていたが、発注先は指摘箇所しか対応しなかった（**認識の相違**）
5	通信メッセージのタイミング仕様が変更され、障害が発生した	通信メッセージのタイミング仕様が変更されたことを周囲が知らなかった（**伝達ミス**）

表 2 の問題点を解決するために何が必要か検討した。この結果を付録1の付表 1 に示す。

付表1から、暗黙の期待のずれを埋めるのに必要な要件を伝える側、受ける側にまとめると、次のようになる。

表 3: 暗黙の期待のずれを埋めるために必要なこと

立場	暗黙の期待のずれを埋めるために必要なこと
伝える側	分かりやすく伝える。 内容を正しく伝える。 感情を正しく伝える。 確認する。 記録に残す。
受ける側	相手の思いを聞く。
伝える側、受ける側共通	日ごろから密にコミュニケーションをとる。 ダイレクトに話す。

しかし、表 2 の問題点の解決策に、表 3 のような「心がけ」的な方法は、決定打にはならない。なぜならば、「心がけ」的な方法は、人に依存する部分が大きく、継続して徹底することは難しいからである。また、このことは、各社ともに同様の失敗を繰り返していることからも明らかである。

4. 有効な対策の立案に向けて

4.1 「なぜなぜ 5 回」の手法を使った原因分析及び分類

前章で挙げた「心がけ」的な対策では、恒久的に有効な再発防止策は導き出せない。そこで、第 23 年度の第 2 分科会で有効性が検証できている「なぜなぜ 5 回」の手法[1]を利用し、コミュニケーション問題が発生した原因を深く掘り下げることで、コミュニケーション以外の原因を探り、対策を検討する。

各研究員が持ち寄った事例を分析した結果、我々が表面的に「コミュニケーションの問題」と認識していた事例の中には、別の原因も含まれていることが判明した。

そこで、各事例について、どのような種類の原因があるかを分析した。その結果を表4に示す。

表4：原因の分析と分類

事例 No	問題	原因	原因の要約	原因の分類
1	システムからのメール送信を、バッチ方式で作ってくれていると思っていたが、実際は即時送信で作られていた	• バッチ方式で実現するように、設計資料に追記したので、見てくれると思っていた • 変更履歴を文書につけていなかった • Excel シート追加のときの、変更箇所の表現方法を決めていなかった	情報の伝達が正しく行われていなかった	**伝達ミス**
2	設計変更が海外のテスト部隊に伝わっていないのが、テストの 2 週間前に発覚して、手戻りが発生した	• 当初テスト項目が未決定になっており、そのままであれば、当然、海外の部隊から確認が来ると思っていた • リリース時期が迫っていたので、議事録作成を省略した		
3	設計工程で、顧客から要件定義の内容に不足があると指摘された。	• 新規開発と派生開発のどちらの開発手法を使うかの基準を明確にしていなかった • 設計者は派生開発のつもりで、充分な要件定義が不要と考えた	要求を 100%定義するのは困難	**自然の摂理**

事例 No	問題	原因	原因の要約	原因の分類
4	システム利用開始後、設計時に合意していた表示色への変更要求が発生した	• 他社製の同一目的のシステムが使っている表示色と違っていたから • 他社製システムが使っている、表示色の RGB 値を入手できなかったから • 利用段階になって、他社製システムのユーザが多数使うようになったから	要求を 100%定義するのは困難	**自然の摂理**
5	顧客の望むテスト内容と、計画したテスト内容に食い違いがあった	• 以前に同システムをテストしたときと同じテスト範囲でよいと判断していた	人は書かれていないことを見つけるのは苦手である	
6	書き込んだはずのデータが消えてしまった	• OS の違いによる挙動の違いを認識していなかった	共通の仕組みで解決すべき問題	**その場しのぎ**
7	外注先が不要な情報を報告書に記載して顧客へ提出したため、顧客が混乱し苦情を受けた	• 報告書に書く内容は正確に伝えたが、そのとおりに実行されなかった • 外注先の報告書のチェックを省略して、直接顧客へ渡すようにした	時間に追われて省略してはいけないことを省略した	
8	類似不良が多発した	• バグが発生したときに、類似箇所があるのは分かっていたが、多忙のため、修正の作業範囲を明確に指示していなかった	とっさの判断で、暗黙の期待の違いに気づかずに判断した	
9	コマンドの書き込みが完了する前に読み込み開始された為、古いコマンドを読み込み誤動作した	• 共有メモリでのコマンド受け渡しタイミングが明確にされていなかった	設計での考慮不足	**設計の問題**
10	通信メッセージのタイミングが変更され、障害が発生した	• 別件の仕様変更で通信タイミングが変化してしまったことに気がつかなかった		
11	テスト結果のリストから、バグが発生しているのを見落とした	• テスト結果の確認を人手でやるように指示したが、人手でやれるような量ではなかった	人間に向いてない作業を、人間がやった	**方法論**
12	ソースファイルの更新手順が守られず、ソフトウェアが先祖がえりしてしまった	• ソースファイルの更新方法が暗黙知になっていた	道具で解決すべき問題を、ルールで解決しようとした	
13	構成管理ルールが守られず、システムがデグレードした	• ビルド担当者の手順に漏れがあった • 担当者が手順の意味を理解していなかった • テストの範囲が限定的であった		

原因を層別した結果、当初はコミュニケーションが原因とされていた問題が、次の 5 つに分類できることがわかった。

コミュニケーション問題

- 伝達ミス (2 件)

  コミュニケーションミスにより発生した問題

コミュニケーション以外の問題

- 自然の摂理 (3 件)

  発生するのが当たり前なことを、考慮しなかったために発生した問題

- その場しのぎ (3 件)

  とっさにとった対策が考慮不十分だったために発生した問題

- 設計の問題 (2 件)

  設計ミスにより発生した問題

- 方法論の欠如 (3 件)

  問題が発生しないような方法をとらなかったために発生した問題

上記の分類から、有効な対策をとる上で、コミュニケーション問題以外の対策を考える必要がある。

5. 対策案と考察

この章では、4 章の分類ごとに、各事例の対策案を説明する。

対策を一覧表にしたものを表 5 に示す。また、この表の使い方は、付録 2 で説明する。

表 5: 原因となる考え方や行動と対策

対策 / 原因となる考え方や行動	原因の分類									
	伝達ミス		自然の摂理		その場しのぎ		設計で解決すべき問題		方法論の欠如	
	【対策5.1-1】変更箇所は確実に分かるようにする	【対策5.1-2】相手に伝わったことを確認する	【対策5.2-1】要求は変化するのが当たり前として扱う	【対策5.2-2】対象と対象外を明確にする	【対策5.3-1】個別対応よりも共通部分で解決すべき	【対策5.3-2】どんな状況でも必要な作業は省略しない	【対策5.4-1】正しい設計ができるように支援する	【対策5.4-2】変化・変動を考慮した設計をする	【対策5.5-1】単純作業は、人ではなく、ソフトウェアで処理する	【対策5.5-2】人はルールを守れないから、ツールを使ってルールを守らせる
情報は文書を渡せば伝わる	事例1	事例1								
相手が「わかった」と言ったから、伝わっただろう		事例2								
要件は最初に100%定義できる			事例3 事例4							
やらない作業は伝える必要がない				事例5						
障害が発生したら、そこだけをすぐに修正する					事例6		事例6			
今まで問題なかったから、レビューは不要						事例7				
1つ伝えておけば、水平展開してくれるだろう					事例8	事例8				
設計者に任せておけば大丈夫							事例9			
この仕様が変わることはないだろう							事例10	事例10		
ルールは守られる						事例7				事例12
注意して作業すれば、ミスは発生しない						事例7			事例11 事例13	

5.1 伝達ミス

相手に情報を確実に伝える場合、正確に伝えるだけでなく、情報を受けた側が理解した内容を、自分の言葉でオウム返しさせる必要がある。

【対策 5.1－1】 変更箇所は確実に分かるようにする

No1 の事例では、設計を記述した Excel ファイルに、後日、新しいシートを追加して、そこに設計情報を追加した。既に存在しているシートに書かれている情報に、変更や追記をしたときは、その部分を赤文字で表記するようにしていた。しかし、シートを追加した場合は変更箇所を明示していなかった。そのため、追加されたシートに気づかれず、意図した設計どおりプログラムが作られなかった。

例えば、シート全体を赤くする等、目立たせることで、変更履歴を読まなくても、変更箇所を一目でわかるようにしておけば良かった。

【対策 5.1－2】 相手に伝わったことを確認する

No2 の事例では、口頭で指示を出したが、相手に伝わったかを確認していなかった。指示を出した側は、相手に指示が伝わっていると思っていた。しかし、相手は聞いていなかったのか、忘れてしまったのかは不明だが、指示した事項は実行されなかった。

この事例では議事録を相手に作成させて、指示が伝わったことを確認すべきだった。

5.2 自然の摂理

プロジェクトには発生して当たり前な問題がある。発生して当たり前の問題には、事前の対策で発生確率を低くできる問題と、どうしても発生を防ぐのが難しい問題がある。発生を防ぐのが難しい問題の場合は、問題が発生しても影響が少なくなる事前策を実行すべきである。

【対策 5.2－1】 要求は変化するのが当たり前として扱う

事前に要求を 100%定義するのは困難である。よって、要求は常に変化するものとしてプロジェクトを進める必要がある。

(理由 1) 常識は要件として出てきにくく、後で抜けが発覚する

No3 の事例は新規開発であったが、派生開発での記述レベルで要件定義を実施した。新規開発なので、顧客は、より詳細な記述レベルの要件定義書を提示されるのが当たり前だと思っていた。そのため、事前にどの程度の記述レベルにするかという要件を提示しなかったと推測する。

この事例では、それまでのプロジェクトとの特性の違いを捕らえ、開発手法選択の基準と照らし合わせて、要件定義の詳細度を決めるべきであった。

(理由 2) 利用者が変われば要求も変化する

No4 の事例では、納入後に、他社製システムの利用経験を持った多数のユーザが、新たにシステムを利用し始めた。そこで、ある状態を示す表示色が、他社製のシステムと納入したシステムで微妙に異なることから、変更要求が発生した。色は好みや環境によって変えたくなるのは自然の摂理である。したがって、色の変更を可能にしておくべきである。

【対策 5.2－2】 対象と対象外を明確にする

人間は文書に書かれていることは理解し、間違いや矛盾を検出できるが、書かれていないことを見つけるのは苦手である。実施対象外を明確にすることで、お互いの暗黙の期待に起因するトラブルを防ぐことができる。

No5 の事例では、テストする内容は明記されていた。しかし、顧客は期待しているテストが含まれていないことを、見つけることができなかった。

実施しないテストを明記しておけば、この事例の問題は防げた。

5.3 その場しのぎ

納期が迫っているなどの事情から、どうすべきかを深く考えずに、その場しのぎでとっさの判断を下すことがある。この場合、間違った判断をする可能性が高くなり、問題の発生につながりやすい。

【対策 5.3－1】 個別対応よりも共通部分で解決すべき

問題が発生したとき、複数の部分に個別に対処する場合がある。特に、納期が迫っているときは、共通部分で解決すべきかどうかを考えることなく、個別対応するという判断を下しやすい。

(理由) 共通部分で解決すれば、同種のトラブルが発生しにくくなる

No6 の事例では、OS の違いにより、プログラムの振る舞いが変わり、障害が発生した。この事例では障害が発生した箇所を個別に修正した。これは、目の前の問題に対処するために、とっさに判断した結果だった。

このプロジェクトの状況では、この方法を採らなくては納期に間に合わなかったかもしれない。しかし、個別に修正する以外の選択肢を考慮しなかったことに問題がある。

OS の違いによる振る舞いの違いを、共通部分で吸収すれば、複数の箇所で同じ修正をする手間を省ける。また、同一プロジェクトだけでなく、複数のプロジェクトで共通部分を共有すれば、組織全体の効率が上がる。

緊急対策をせざるをえない場合でも、緊急対策実施後に、共通対策がなされるよう管理すべきである。

【対策 5.3－2】 どんな状況でも、必要な作業は省略しない

(理由 1) 今まで大丈夫だったからといって、今後も大丈夫とは限らない

No7 の事例では、プロジェクトの途中までは、外注先が作成した報告書の内容をPM(Project Manager:プロジェクト・マネージャ)が確認して、顧客へ提出していた。しかし、納期が迫って、報告書を確認するための時間を惜しみ、外注先から顧客へ直接報告書を提出するように変更した。そのため、不適切な内容の報告書が外注先から顧客へ渡ってしまった。

この事例では、報告書に記載する内容は、外注先に正確に伝わっていた。しかし、外注先が間違った記載をしたまま顧客に提出してしまった。時間の短縮を目指す場合でも、省略するプロセスの意味を検討して、問題が発生しないように考慮する必要がある。プロジェクトの終盤では、外注先も時間の余裕がなくなるので、間違いも混入しやすくなる。

(理由 2) 担当者が水平展開してくれるとは限らない

No8 の事例では、障害が発生した箇所と類似した部分がシステムに多数あった。PMは、障害が報告された箇所の修正のみを外注先に指示した。このときPMは他の類似箇所も、当然修正してくれるだろうと思ってい

た。しかし、外注先は指摘された箇所のみを修正し、後日、他の類似箇所での障害が多数報告されることとなった。

PMは多忙であったため、特に指示をださなくても、外注先が修正を他の部分にも水平展開してくれると、暗黙の期待をした。しかし、特に日本と文化が異なる海外の会社には、そのような期待をするのは無理だと考えるべきである。この事例では、発注者側は、類似した部分の1つに障害が発生した場合は、他の類似部分も修正するように指示する必要がある。

5.4 設計で解決すべき問題

5.3 節までで説明した対策を実行しても、設計自体が悪いと、問題は発生する。設計者個人と組織は、設計ミスを防止するように取り組まなくてはならない。

【対策 5.4－1】 正しい設計ができるように支援する

No9 の事例では、共有メモリを使ったコマンドの受け渡しで、読み込みと書き込みが競合したときに障害が発生した。共有メモリの読み書き時には、排他制御が不可欠である。

設計者が設計技術を自ら向上させて、設計ミスを減らすのが最良の策である。しかし、設計は人に依存する部分がなくせない。よって、組織は設計常識の教育、設計品質の評価ツールの導入、設計レビューを怠ってはいけない。

【対策 5.4－2】 変化・変動を考慮した設計をする

No10 は通信タイミング仕様の変更が伝わってなかった事例である。この事例では通信タイミングは文書に明記されていたが、別件で仕様変更が発生し、通信タイミングがずれることに気づかなかった。これを防ぐには、タイミングの変更に影響を受けない設計をすべきである。

5.5 方法論の欠如

人間は忘れたり、手を抜いたり、勘違いしたりする。これを無視して、忘れない、手を抜かない、勘違いをしないことを人に要求すると、問題が発生する。人間は忘れ、手を抜き、勘違いすることを前提に、問題が発生しない方法を採用しなくてはならない。

【対策 5.5－1】 単純作業は、人ではなく、ソフトウェアで処理する

No11 の事例では、テストツールが生成した数千行のテスト結果を、人間が目視で調べて、不合格項目の有無を確認していた。目視で調べたときに、不合格の項目を見逃し、欠陥が流出してしまった。

ファイルに記録されている大量の情報から、不合格を示す情報を探すのは、人間よりもコンピュータの方が得意である。ファイルを走査して、合格か不合格かを判定するツールを使うべきである。

【対策 5.5－2】 人はルールを守れないから、ツールを使ってルールを守らせる

人にルールを 100%徹底させるのは難しい。人間が実行することなので、手順の実施もれや手抜き、誤りを完全に防ぐのは困難である。

ルールを道具で実現して、人為的な間違いを防ぐ必要がある。

(理由 1) 手順の実施もれは防げない

No12 の事例では暗黙のルールを文書化していなかったので、手順の実施もれが発生した。たとえ文書化していたとしても、不注意や手抜きによる手順の実施もれは防げない。

この事例では、構成管理システム等のツールを使うことで、手順の実施もれが発生しないようにするべきだった。

(理由 2) 人間が間違えるのは当たり前

No13 の事例は、ビルドの際に、ファイルのバージョンの一覧を見てファイルを集める作業で、間違えたバージョンのファイルが混入した問題である。これは、人手でファイルを集めるという作業方法が問題発生の原因である。ビルドに含めるファイルのバージョン一覧から該当するファイルのみを収集するツールを使うことで、人為的なミスを防ぐべきである。

6. 結論

我々は、個々の事象から問題の根本原因を抽出し、問題点を洗い直した。その結果、次の5つのカテゴリに分類できた。

- 伝達ミスによるもの
- 自然の摂理によるもの
- その場しのぎで対応したもの
- 設計上の問題によるもの
- 方法論の欠如によるもの

以上の分類で実際に起きた問題を整理した結果、5 章の表 5 の結果を得た。

この表の「原因となる考え方や行動」は、今まで普通に行われていた行動が並んでいる。そこにメスを入れると、本当に意味のある対策が立てられる。このような、疑ってもみなかったことが、他にもあるはずである。

また、オフショア開発やパートナー企業との問題に対し、一口に「異なる文化」のためのコミュニケーション不足と短絡的に問題を特定して、その対策を講じても恒久的な手が打たれたことにはならない。対策を考えるときは、問題の分解、局所化が肝要である。

今後、今まで普通に行われていた行動を疑うことにより、実効のある解を見つけ、プロジェクトの問題を撲滅することに繋げていくことを期待する。

参考文献

[1] 板倉, 森本, 石井, 古谷野他:「ソフトウェア品質管理の研究－第 23 年度ソフトウェア品質管理研究会分科会報告書」, 財団法人日本科学技術連盟, pp33-45「ソフトウェア開発へのなぜなぜ 5 回の適用」, 2008

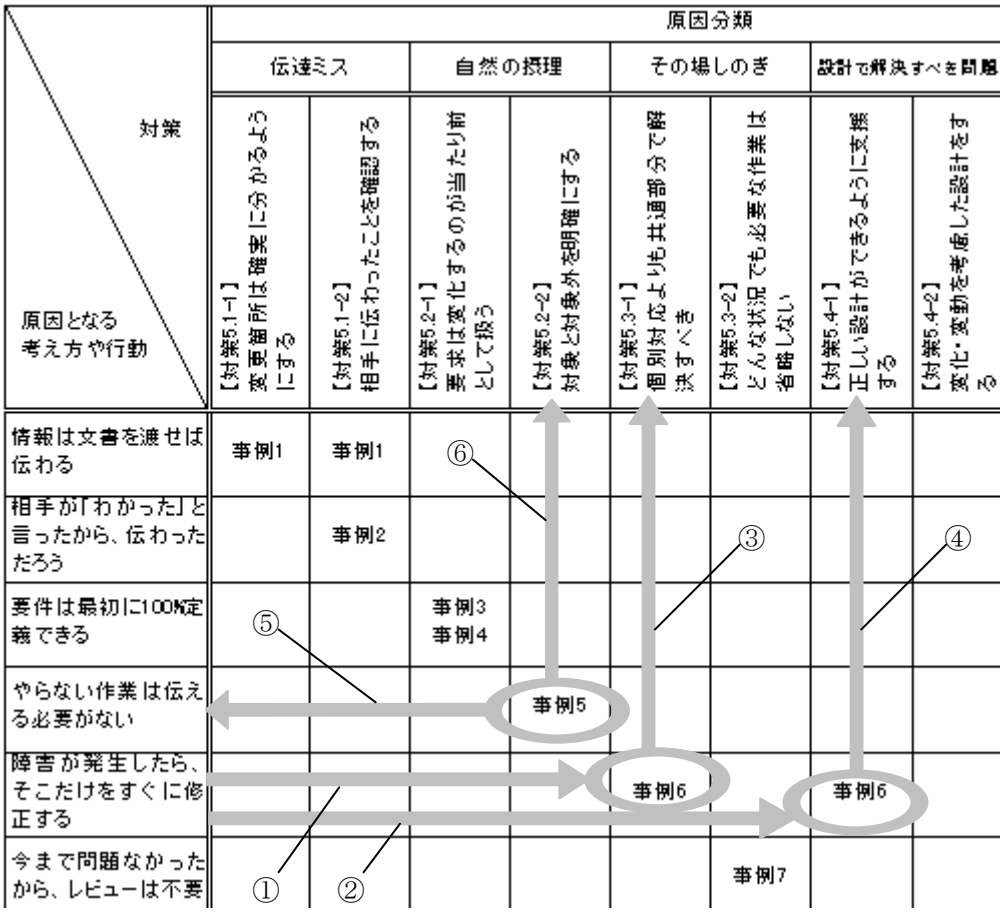
付録 1

付表 1: コミュニケーション問題を解決するために必要な内容の抽出結果

ベースとしてやりたいこと	直接	相手を想い聞く	相手を想い語る	わかり易く伝える	内容を正しく伝える	感情を正しく伝える	確認①	確認②	その他
日ごろから会話をする	面と向かって話す	まずは傾聴	相手の立場を少し思って言う	一言で言う端的に言う	違う解釈が出来ないかを確認する	相手の受け方感じ方を考慮する	確認する	議事	キーマンは誰？
普段から仲を良くする	誤解があったとき直接話す	相手の話を良く聞く	相手の立場・目線で考えて語る	POINT 重点	専門用語／社内用語をさける	伝え方を工夫する(ツールの使い分け)	確認する	相手に議事録を書かせる	背景を伝える
信頼する／してもらう	具体例を示す	相手の話をきく		要点を明確にする	相手が理解できる言葉で書く	絵文字･メール	フィードバック	メール＋電話	何故そういうのか誠心誠意説明する
相手のことを好きになる	体当たりで行う	相手のことを理解する	相手の自尊心を尊重する	文章は短く	第三者が聞いてもわかる言い方にする		相手に要約してもらう（理解を深める）		重要度は
回数を増やす（密にする）	最低限必要な人数で行う		相手によって話し方を変える	文の長さ	ストレートな表現		相手に理解した内容を伝えてもらう		興味を持たせる
考えを共有する	会話（双方向）				（中国）ゆっくり話す		差を明確にする		周りにやる気を起こさせる
	他人事ではなく当事者として熱意を持って語る						回答は？様子見		見える化
									図表を使う
									定量的に
									タイミング（定期／不定期）
									打ち合わせの時間の長さ(短時間)

付録 2　表 5 の使い方

表 5 には、縦に問題が発生する「原因となる考え方や行動」を、横に問題を発生させないための「対策」を並べている。「原因となる考え方や行動」と「対策」の交点に、表 4 で紹介した事例 No を記入した。表 5 の一部を付図 1 に示す。



付図 1：　表 5 の使い方

この表は次の 2 通りの使い方ができる。

(1)　原因となる考え方や行動から、とるべき対策を見つける場合

表の 5 行目にある、「障害が発生したら、そこだけをすぐに修正する」という「考え方や行動」を例として説明する。この「考え方や行動」から発生する問題への対策を調べるには、「障害が発生したら、そこだけをすぐに修正する」の行を右方向にたどる。すると、図中の①で示した矢印と、②で示した矢印が、それぞれ「事例 6」にたどり着く。これらの場所から、それぞれ③と④の矢印で上方向にたどると、③は「【対策 5.3-1】個別対応よりも共通部分で解決すべき」、④は「【対策 5.4-1】正しい設計ができるように支援する」が、対策として該当することがわかる。

すなわち、「障害が発生したら、そこだけをすぐに修正する」という考え方の人がいる場合、「個別対応より

も共通部分で解決すべき」ことを教えるとともに、「正しい設計ができるように支援する」必要があることがわかる。

(2) 表 4 の事例の問題と対策を調べる

表 4 に挙げた、それぞれの事例の「原因となる考え方や行動」と「対策」を調べる方法を説明する。事例 5 の「原因となる考え方や行動」と「対策」を知りたい場合、表 5 の中から「事例 5」と記入されている場所を探す。見つかったら、その場所から⑤の矢印のとおり、左方向にたどれば、「原因となる考え方や行動」は「やらない作業は伝える必要がない」であることがわかる。「対策」は⑥の矢印のとおり、上方向にたどることで、「対象と対象外を明確にする」であることがわかる。

複数の場所に同じ事例 No が記入されている場合は、全ての場所から、上記と同じ作業をする必要がある。