

## 2013年度 第2回特別講義 レポート

日時	2013年6月14日(金) 10:00~12:00
会場	日本科学技術連盟・東高円寺ビル 2階講堂
テーマ	「なぜなぜとカイゼン(人重視マネジメントの考え方)」
講師名・所属	黒岩 雅彦氏(日本電気株式会社)
司会	第2分科会主査 早川 勲氏(アズビル株式会社)
アジェンダ	1. 現在要求されるもの、過去の品質マネジメントにおける課題 2. なぜなぜアーキテクチャー 3. 現実的なカイゼンの考え方 4. 人重視マネジメント
アブストラクト	<p>現在のSI事業、SW開発においては、非常に短期間で高品質の要求が求められています。しかしながら、要求される機能は複雑化し、その体制もオフショア開発を含めた多くの組織、会社、要員による遂行が必要になります。また、要員確保のために、多くの新人、新規参画者による体制を構築せざるを得ません。</p> <p>今回の講義では十分な実践を繰り返し、チューニングを繰り返してきている人の行いに着目した、「人重視マネジメント」をご説明させていただきたいと思っております。非常にスピード感の高い、なぜなぜの具体的なやり方、組織の作り方、上司や組織のあるべき姿、開発風土の作り方等、PJや組織運営そのものに対しての具体的な考え方や施策をご説明させていただきます。</p> <p>本講義はあくまで実践型であり、すべてこれまでの実践経験に基づいた結果として、現実的な内容となっております。</p>
<b>&lt;講義の要約&gt;</b>	
<b>◆今事業に求められるもの</b>	
<p>これからの事業運営に最も求められるのは「スピード感」である。この命題に答えるためには、間接的な業務を最小限に抑える事が必要だが、実際はなぜなぜ分析に多くの時間をかけている現実があり、かつ問題の混入工程から以降の工程での分析では、例え対策を打てたとしてもそれまでに無駄なコストと時間を使い切っているわけであり本末転倒である。その対策として取り組んでいる内容を以下に示す。</p>	
<b>◆なぜなぜ(なぜ3)の種類と目的</b>	
1) 品質保証型なぜなぜ(プロセス実行型なぜなぜ)	

開発プロセス実行中におけるインシデント発生時にリアルタイムで実施するなぜなぜであり、数値分析を併用する事により、品質改善は元よりコストカットも可能とする。これが NEC グループにて展開中の取り組みである。

## 2) 振り返り型なぜなぜ(プロジェクト振り返り型なぜなぜ)

従来の一般的な取り組みであり、失敗プロジェクト等の大きな課題に対して要因分析と対策を検討するものである。振り返り型なぜなぜは、組織の問題の掘り下げや事実確認を目的として反省会形式で風土創りや教育のために行われる。そのため、個々のインシデントに対して行うと過去の現場作業の振り返りとなり、無駄な作業を招く。

### ◆基本的な品質管理について

本講演の具体的内容に触れる前に、一般的な品質管理方法を確認しておく。

#### 1) 品質管理とは

品質管理には「プロダクト品質の管理」と「プロセス品質の管理」があり、前者は結果の確認であり、重要なのは後者の手段としての開発手順の品質を管理する事により品質を保証することである。この手順重視の考え方が重要である。

#### 2) 一般的な品質保証の方法

一般的な品質保証の方法には「静的品質分析」と「動的品質分析」がある。

静的品質分析とは、ソフトウェアメトリクスによる定性的分析と指標との比較分析により工程終了判断に用いるものである。そもそも、指標の設定根拠自体が過去版の実績値に基づくものであるため、分析対象のプロジェクトに対し人的要素を考えた場合は曖昧であり、ある意味理想論である。

動的品質分析とは、障害発生数の収束状況を時系列的にグラフ化し、残障害数必要な試験件数等を予測するものである。信頼度成長曲線等を用いるが、そもそも試験項目内容により故障発生率が異なる事から、分析においては各試験実施時期の考慮が必要となる。

### ◆従来の品質管理手法の課題

ここまで述べたように、様々な分析理論はあるが、現実にはその理論通りには行かないのが実態である。

その理由として、前述した理論はハードウェアの開発における品質管理理論から来ているものであり、開発において人間の関与が大きなソフトウェア開発においては、これらの理論だけでは品質保証が難しいからである。

#### 1) 数値分析の課題

数値分析においては、工程判断基準値からの乖離を問題視するが、一方、乖離していない場合でも問題が潜在している場合がある。この分析理論においては、開発プロセス自体がきちんとやら

れている事が条件であり、「手抜き」「さぼり」等により作業プロセスに異常がある場合には意味を持たない。

さらに、一番品質に影響する「開発機能の難易度」「開発方式」「スキル」という要素は数値化が難しいジレンマがある。

また一般的に、開発規模とバグ摘出数は一次関数のグラフで表されるが、開発者のスキルを考慮した場合、特異値を分ける必要もある。つまり「だれが作ったか」が重要であり単純な式では表現できない。

## 2)バグ件数成長モデル(グラフ)の課題

グラフィカルで見やすいため関連者への説明に使われることが多いが、実は条件がむずかしい。各テスト項目の1件1件の内容が等価であることが大前提である。また、単体テストのように、次々と別の機能を試験する場合には収束はありえない。総テスト件数が少ないと試験対象によるぶれの影響が大きくなるため、試験目的を分けた試験は収束に意味を持たない。収束は視覚的な感覚なので、縦軸を縮小しただけでも見誤る場合がある。

### ◆ミッションクリティカルシステムでの考慮点

ここまで述べた課題事項を基に、ミッションクリティカルなシステムの開発における考慮点を整理する。

#### 1)体制として多くの人間で開発すること。

事例としては大規模プロジェクトだが小さなプロジェクトでも同じであり、「仕事を軽くして品質を向上させる」事が目標である。

手段として、全員が「やるべき事をサボらないようにする」必要がある。そのためには、「担当者の手順

にできない事を入れさせない」事が重要である。

その確認として「きちんとやったかをチェックできる仕組み」が必須である。これを実現するためには、随時できなかった場合のポイントを分析して、できる手順に改善するという地道な取り組みが必要である。精神論でやらせるのではなく、個々の人としてできる手順に落とし込む事が重要である。

#### 2)今まで経験の無い高品質の開発を行うこと。

今後の日本のソフトウェア開発の担うポジションは「超短期開発」「超高品質開発」である。その他はオフショアに出さざるを得ない。

この達成には、一般的には高いレベルのプロセス標準が必要と考えるが、経験の無い環境、目標に対し最初から標準を用意する事は不可能である。そのため、とにかくプロダクトを早く動かして確認しチューニングする事とし、全体を止めないという手段が良い。例として、NECが開発した惑星探査機「はやぶさ」は、未知の宇宙において想定外の事が起きても手直しを可能とする補助エンジンを搭載する事で補正により目標を達成した。

3)利益を出してビジネスとして成り立つこと。

この命題達成には後戻り工数を減らす事が最重要である。大規模開発でかつ短期開発の場合、手戻りが発生してからではビジネスそのものが成り立たなくなる。再レビューにより現場開発者は元より、運営の間接工数の増加は一番の無駄である。

#### ◆対策としての「プロセスベースの品質保証」の適用

以上の考慮すべきポイントに対し対策を以下に述べる。

##### 1)プロセスベースの品質保証

以下の取り組みを最上流工程から継続実施する事が大事である。問題発生時には、プロセス上の課題を追求し発生した当該工程の中で横展開までを完了する。そのためには、非常に短期間の分析とアクションが必要である。

##### 2)なぜなぜ分析時の注意事項

必ず混入工程を意識し、例え次工程以降に持ち越したバグでも作りこみ工程へ立ち返り分析が必要である。作り込み工程が明確になったら、その工程の問題を招いたプロセスを明確にする。この場合作り込み担当者本人にヒヤリングする事が必須である。プロセス上の真の原因が明確になれば、それを裏返しプロセスの改善と横展開を行う。

##### 3)うっかりミスの扱い

人間の作業にはうっかりミスもある。この場合数値分析により、多発ではなくかつ特定の人や機能に集中していない場合は次工程での様子見とする。うっかりミスに対してもプロセス改善をやろうとするとプロセスが動かなくなるため、「うっかりミスかどうかの見極め」が重要である。

##### 4)なぜ3ができる条件

- ・やるべき事から何が外れているかを追求することであり、これが明確でないとなぜなぜはできない。これを明確にするには担当者の目線に合わせる事が重要であり、目線合わせができない管理者には、正しいなぜなぜを実施できないのである。
- ・誤ったなぜなぜ、つまり、現象ベースのまま対策を実施する事は大きな無駄を招く。正しいなぜなぜにより、ピンポイントでの対策が実施でき最小の工数で対策が可能となる。

##### 5)なぜ3のコツ

- ・正しい原因追求があつて始めて対策が決まる。つまり、正しい原因追求ができる人がいる事が重要となる。よって重大な問題が発生時には自ら現場に出向きなぜなぜを実施している。
- ・なぜなぜの回数は多いほうが良いと言われるが、原因を裏返してピンポイントの対策になれば回数は関係ない。
- ・事後だからわかる事ではなく、事前に防げる対策でないと意味が無い。
- ・なぜなぜにおいて上司の意見に押し切られる場合が多いが、現場目線になれず無駄な対策になる場合が多い。

・本質の問題と偶然見つけた問題を分ける事が重要である。本来のなぜなぜができるのは一つの問題についてである。よって、二次的に派生した問題は別途整理して対策を打つ事が重要である。

・先入観を捨ててヒヤリングする事が重要であり、過去の経験とか常識論から誘導しそうな立場の人はヒヤリングを実施すべきではない。

#### ◆プロセスカイゼンの考え方

・なぜなぜをやるほど、仕事が楽になり生産性も上がるようにするには、安易にチェック項目を増やすのではなく、チェックを如何に簡単にさせるかを重視した対策にすることが重要である。また、問題を作り込まないことをも重視した対策にすることも重要である。

・標準やルールを事細かく規定してしまう事が多いが担当者の稼働を増やす事となり逆効果である。担当者が如何に必要な要所を押さえられるかが重要であり、そのためには担当者の立場を理解することが重要である。

・一つ一つの改善は妥当な内容でも、積み重なると重くなる。結果として担当者がサボる原因になる。よって、ほとんどの担当者が守れる「単純」「簡単」なプロセスとすることが重要である。

・全ての担当者が守れる事を目的に底上げする事は本来ではない。ほとんどの人ができるレベルとし、それに満たない人は個別に指導するのが良い。

・カイゼンとは、目的を理解しやり方を変える事である。カイゼン内容は具体的でやさしい言葉で表現し、相手に目的や価値を理解してもらうことが大事である。

・カイゼンはタイミングが重要であり、一部で試行してみて随時チューニングしていくことも良い。

#### ◆忙しい人を更に忙しくするジレンマ

・なぜなぜの原因追求結果で「工数が無い」は相応しくない。こういう人にいくら時間をあげても別のことをするので無駄である。「工数が無い」から「何をサボったのか」がプロセスの原因であり、工数が無いのが原因ではないからである。

・元々の作業が溢れている状態で発生した問題のカイゼン策として新たに作業を追加することは、新たにサボるものを増やすこととなり悪循環を招く。対策は今の仕事をカットするかトレードオフしないと解決に結びつかない。

#### ◆有識者の苦悩

「次回は有識者を入れてレビュー実施」という対策は無くすべきである。有識者とは業務や技術に長けた人ではなく、過去に一人で泥をかぶり苦勞して何かを少しだけ掴んだ人である。つまり、本来ではない事を記載して責任を擦り合うばかりであり価値が無い。

#### ◆良いチェックシートとは

セルフチェックにおいて失敗事例として「やってはいけないこと」を A4一枚で記載する。多くの過去事例は羅列しない。これであれば曖昧な回答は無く YES/NO で回答できる。セルフチェックをサボった要因、セルフチェックを行う際に根拠資料を確認していなかった理由、改造の場合は変更箇所

しかチェックしなかった理由を明記しチェックさせる。つまり、過去事象ではなく過去の混入要因を挙げる事がポイントである。

#### ◆なぜ3の適用範囲

なぜ3はソフトウェア開発以外でも使える。例えば、進捗管理では、通常頁数の出来高で管理するが、その場合仕様追加変更等で母数が変われば意味が無い。よって課題管理が重要である。残課題数、残課題内容の重要度の把握が大事だしこれを管理する事が進捗管理である。

#### ◆現実のスピード感を考慮しての対策制御

品質保証をするには分析工数がかかる事は当然である。しかしながら、これと「面倒くさい」ということは別である。現実のプロジェクトではスピード感が必要であるし、そのためには理想論による対策は不要である。今できてない事を全て把握した上でそれら全てを短期間で実施するのは困難である。よって、「今からベストなことをやって品質を保証できる自信がある事だけをやる。」その上で、更に問題が出た場合には原因の分析有りきでピンポイントで対策を打つ。

#### ◆大局観によるカイゼンの重要性

カイゼンを要するプロセスの流れの中の一箇所に対策を打ってもその前後のプロセスへの影響が考慮できていなければ効果が出ない。つまり大局観の感性を鍛え、周りとの関係を大切にすることが重要である。

#### ◆コミュニケーション能力の重要性

ソフトウェア開発は人間が行う以上、一番必要となるのはコミュニケーション能力だし、それにより「現場の痛みがわかること」が重要である。これにより分析をするまでもなく、日々のコミュニケーションで問題がわかることが多い。なお、コミュニケーション能力を鍛えるのはリーダークラスに成ってからではなく、担当者の時点から積み上げて行くものである。

#### ◆適材適所で、かつ助け合える要因配置

人は得意不得意があるのは当然である。管理者は技術的な視点からの人員配置ではなく、担当者個々人の得意不得意性格を把握した上で体制を組む事が重要だし、これにより個々人の能力の補完や助け合いにより単に足し算以上の能力が発揮される。

#### ◆権限委譲によるチーム力の発揮

管理者が全て指示により作業をやらせると、指示できる範囲のうちはいいがこれからあふれた途端に破綻する。大切なのは人をきちんと見て権限委譲をしてチームとしての力を破棄することである。つまり、「実行力として多くの人に動いてもらえる」事が重要である。ただし、投げっぱなしでなくやり方と結果の確認は必須である。

#### ◆当事者意識の重要性

第三者的な評論は不要である。カイゼンを推進する者は一人称で当事者とのコミュニケーションに

よって行うべきである。

#### ◆部下の育成

管理者は部下を「育成」しなければならない。最初から出来そうにもない事をやらせるのが「投げっぱなし」、最終的に出来なかったら責任を取る覚悟で渡すのが「育成」である。そのためには、やり方を教えるのではなく、「考え方」や「目的」、「繰り返し」で結果が出せるようにしてあげる根気と忍耐が必要である。

#### ◆マネジメントの役割とは

マネージャーは「教師」であり「刑事」であり「医者」でもある。何れも「相手の立場や気持ちの理解」が出来ないと勤まらない。例え、医者の腕で患者の生存率を上げて、最後は患者の治す気持ちが無いと助からない。つまり、医者は患者に対し十分な事前説明をして「同じ目標」をもって取り組む事が重要なのである。

#### ◆内部摩擦の無駄

技術者として正しいものを作るにはどうしたら良いかを目標に体制や役割を決めるべきであり、似非商人や似非法律家は不要であし、これに類似したメンバーが増えると内部摩擦が生じる。我々は技術者集団である事を肝に銘じて磨きをかけるべきである。

#### ◆楽しく仕事をするということ

楽しく仕事をするには、嫌なことを減らすということも大切である。例えば価値もわからない帳票を作成したり、時にこれが誤ったなぜなぜ分析票だったりするのだが、これらに目をつぶらずに、楽にする努力を惜しまない事が大切である。

#### ◆風土を作ることの重要性

カイゼンの実現には、プロセスを変える技術論のみではだめで、「やるべき事をちゃんとやる」風土を作ることが重要である。プロセスを変えても守らなければ意味が無いからである。そのためには、管理者は「何事にも逃げない姿勢を自ら実践」し、担当者は「自ら考えプライドと責任を持つ」必要がある。「風土が作れず現場が目をつぶったらおしまい」であるし、立場に関係なく「正しいことが正しいと言える風土作り」が大切である。

#### ◆質疑応答

Q.なぜなぜ分析で「偶然見つけた問題」とは、どんなものなのか？

A.例えば、バグレベルのなぜなぜをやっている時に、「これはもともと設計書の記載が不十分だからである」といった方向に脱線して見つかる問題である。つまり、本来コーディングレベルの分析であるのに「思考が日ごろの思いの方向に流れる」ことで見つかる別の問題のことである。

Q.本日説明いただいた内容の動きが出来れば本当に組織は良くなると思うが、黒岩様レベルの人を育成するのは現実には難しいと思う。なぜなぜ塾を開講して育成しているそうだが、具体的に

はどのようなやり方をしているのか教えてほしい。

A.例えば何千人もの戦略プロジェクトでなぜなぜを広げるために、実際どうしたかと言うと、最初は私しか出来なくて他のプロジェクトリーダーは「ど素人」だった。そこで、なぜなぜ分析結果を現場から直に私に持って来るように指導した。そしてグループリーダには、私が赤入れした結果を見て勉強するように指導した。これで作業を回しておき「この人は出来る」と私が判断した人には赤入れを権限移譲した。要はプロジェクトマネージャの仕事が権限委譲できる人を育てていくことが大切である。そして、権限委譲できるスタッフは専門職化して組織上の上司とも対等に戦えるようなスタッフとすべきである。

Q.「私は面倒くさがりやだから・・・は嫌だ」という言葉を多く聞かれたが、その反面、「人を育てないとみんなが幸せになるラインを決めて、そこから下の人は個別に指導すればいい」とも仰っておられる。そもそも「個別に指導」は外から見ると面倒くさいことだと思うが黒岩様はそのようには見えない。「何を面倒くさいと思うか」が大事なキーポイントのように思うがこの基準は何か？

A.「レビューが嫌い」とか「設計は嫌だ」とか言ってはいるが、私は逆にこれらが好きな人を見つけて組んでいる。つまり完全な権限委譲をしている。

逆に現実的ななぜなぜ分析とか対策の検討については非常に細かく実施している。実際、担当者が気づかないレベルまで実施しているが、これが出来るのは私が好きだからである。つまり私の個人的な感覚で面倒くささを判断している。これによる誤りもあるので他人の意見を聞く耳はいつも大切にしている。

大切なのは組織上の立場ではなく、「現場を良くすることに対し何をすべきか」という尺度である。そのために私は常に現場目線である。つまり面倒くささの定義はしにくいし、多くの人が私のように感覚論で動くのは組織的に難しい。各組織に数名で良いと考える。

#### ◆感想

黒岩氏が一貫して掲げるのは、「人中心のマネジメント」である。開発に携わる個々人のモチベーション高揚が個々人の仕事の楽しさにつながるし、結果的に短期間での高品質なソフトウェア開発につながる事が理解できました。その手段として、正しいなぜなぜに回帰する事が必須であると認識しました。

従来、誤ったなぜなぜにより不用意に対策を追加し、そもそもあふれた作業のために誤りを作り込んだのに、さらに追い討ちをかける対策で負のスパイラルに入り込むという経験は多く経験してきました。また、対策としてのチェックシート活用については、現場のためではなく報告書に記載するためのチェックシートが多く氾濫しており、今回紹介されたような、現場担当者の作業目線で真の原因まで落とし込みされた最小限のチェック項目は、どこの社でもそのまま使える事項ではないでしょうか。

ソフトウェア開発は大組織化するにつれ、上部と現場との距離感は増すばかりです。だからこそ、組織の層を跨いだ又は意識しないコミュニケーションが重要であるし、建前の理想論ではなくゴールに向かうためには、何をすべきかという橋渡しの取り組みをする黒岩氏のようなエバンジェリストの存在が重要であると感じました。