



# 優秀なエンジニアになる方法

ロバート・E・ケリー (著)

デービッド・K・アサノ (訳)

ロバート・E・ケリー  
デービッド・K・アサノ 正員 信州大学工学部情報  
工学科

**How to be a Star Engineer.** By Robert E. KELLEY, Nonmember, Translated by David K. ASANO, Member (Faculty of Engineering, Shinshu University, Nagano-shi, 380-8553 Japan).

## 要旨 (訳者より)

これはケリー氏自作の「How to be a star at work」<sup>(注1)</sup>から改稿し、IEEE Spectrum 1999 年 10 月号<sup>(注2)</sup>に掲載された話である。主にエンジニアを対象にしてあるが、仕事で優秀になりたい人なら役に立つ内容であろう。

## 1. 背 景

私は 1985 年にベル研究所からある一連の質問を受けた。以来、それらの答えをずっと探し続けてきた。ベル研究所(当時は AT&T の一部で、現在はほとんどが Lucent Technologies に所属)が当惑していたのは、世界の有名大学から優秀な人材を採用したにもかかわらず、期待にこたえたのは数名しかおらず、ほとんどは平均的な成果を出すように成長したが、AT&T の利益に余り貢献をしなかった。

ベル研究所が知りたかったのは、「優秀なエンジニアと普通のエンジニアとどこが違うか」、「優秀なエンジニアになる素質は生まれつきなのか、教わるものなのか」、「普通のエンジニアを優秀なエンジニアに変身させることを手助けするような生産力向上プログラムは可能か」である。

こういった質問を聞いているのは会社だけではないし、自分の生産力を向上させたくないと思うエンジニアはいない。ほとんどのエンジニアは自分が優秀な人材になれると考えている。だれでも同僚に先を越されることは嫌だし、自分の仕事をよくこなすために常に努力をしている。職場では最低限のものと時間を使って最高、最多なものを作れといわれている。グローバルな競争、合併、リストラの影響を受けて、

彼らの責任は増加し、資金、人材等は減っている。5 年前に比べて、勤務時間や負担が増えていない人が何人いるか? たまっている仕事や返事を必要とする電子メール、電話のリストが増えていない人がだれかいるか? 我々の中でだれか、生産力を向上しなければ首になると恐れていない人がいるか? 自分の生き方、仕事と私生活のバランスをより良くしたいと思わない人がだれかいるか? 「もっと賢く働け」と言われてもやり方が分からない。

以来、私たちはこういった企業と個人的な生産力問題について以来調査し続けてきた。ベル研究所、3M、ヒューレット・パカードから千人以上のエンジニアがこの研究に協力してくれた。優秀なエンジニアへの秘訣を探るために、筆記試験、観察、業務日誌、グループ討論、個人面接の方法を利用し、統計分析、内容分析、反復モデル作成を必要に応じて行った。

ほかにも、電気・電子関係のエンジニアに頼る Analog Devices, Fore Systems, Air Touch などから、また異なる分野のエンジニアが関係する Shell Oil, Kimberly Clark のような企業まで、たくさんの企業の協力を得た。このような企業はこの研究から生まれた生産力向上プログラムを利用し、技術系社員の生産力を高めただけでなく、優秀なエンジニアに関する知識を高めるのに貢献した。

## 2. 優 秀 さ へ の 道

ライとヘンリは同等な資質を持ってベル研究所に採用された。優秀な成績で上位ランクの大学を卒業し、コンピュータ関連会社でインターンシップの経験もあり、大学の先生から立派な推薦ももらったことまで同じであった。しかし、最初の 6 か月の新人研修期間における仕事の取りかかり方はまるで違っていた。午前中は電話通信技術やベル研究所での研究や営業に関する研修を受けた。午後は初心者向けプロジェク

(注1): Adapted from "How to Be a Star at Work" by Robert E. Kelley, Times Books: New York, 1998, 1999.

(注2): Translated from the original English version and reprinted with permission, from IEEE Spectrum; vol. 36, no. 10, pp. 51-58; Oct. 1999; ©1999 IEEE.

ト (失敗しても重要なプロジェクトに大きな影響を及ぼさないもの) に参加した。

ヘンリは卒業論文をまとめたり、資格でも取るかのごとく自分の部屋にこもりきりだった。最新の技術に関する知識を身につけるために資料を集め、熱心に読んだ。部屋から出たのはトイレに行くときと欠席できない会議のときのみであった。当時、ヘンリは自分が技術的に優れていると同僚に証明することが一番大切だと考えていた。

ライは毎日、午後に自分の分担業務をこなしたり、技術力を高めるため、3時間を費した。その後の残り時間を利用して、同僚に話しかけ、彼らが抱えているプロジェクトについて尋ねた。だれか助けが必要であればライは自分から進んで手伝った。ライは新人だったけれど、自分の仕事と関係のない仕事でも快く手伝ってくれることが同僚にとってありがたかった。

ある日の午後、ライの同僚は次週が締切となるソフトウェアプロジェクトの難しいプログラムと格闘していた。ライは専門講義で覚えた新しいプログラミング手法がこの問題を解決できると考えたので、同僚がもっと大きな別の問題に専念している間にライがこのプログラムに取りかかるかと申し出た。またあるときは、全員のパソコンに高性能のソフトウェアをインストールする必要がある。こんな場合、全員が試行錯誤で自分のパソコンにインストールするのが普通である。インターンシップのときに同じような経験をしていたライは、一人が全部のパソコンにソフトウェアをインストールする方が能率が良いと考え、自分がやると進んで取りかかった。インストールは考えたより難航し、結局予定していた4日間ではなく、2週間かかった。ライは途中でやめることもできたが、研修や仕事に影響しないように数日間早めに出勤したり、残業しなければならなかったにもかかわらず最後までやり通した。

6か月後、ヘンリとライは研修と初めてのプロジェクトを終えた。彼らのプロジェクトは成功し、技術的に問題ないと判断された。ヘンリの方が技術的にライより上だったかもしれない。

しかし職場において、ヘンリには明らかに足りないところがあった。良い人だと評判はあったが一匹狼であるという印象が強かった。能力があることは認められたけれど、その能力を同僚との共同作業に生かせるかどうかは疑問だった。ヘンリは大学にいるときのように個人の成績を第一に考えていた。

それに対して、ライは自発的に仕事を進めるといった印象を与え、自分に責任のない問題でも見つけたら解決するために手を挙げた。職場にいるのが6か月よりはるかに長い期間だという印象も与えた。上司はライが優秀なエンジニアの素質

を持っていると考え、出世コースへの候補者と考えた。

ヘンリを含めて多くの人は「優秀さ」について先入観を持っており、その先入観はとんでもないほど間違っている。私たちは「優秀さ」に関する研究を始めて以来、よくある迷信が間違っていると証明し、優秀なエンジニアに関する驚くべき事実を発見した。最初に発見した事実の一つは「優秀なエンジニアはだれなのか」という質問に対し、エンジニアとその上司の意見が違うということである。管理職にまず「大事なプロジェクトがあればだれを選ぶか」「自分で会社を作るならだれを誘うか」などの質問をした。管理職が選んだ人をエンジニアたちに見せたところ、「なんでジョーがいるんだ? 数年間何もやってないじゃないか」「メアリーはどこだ? 問題があれば彼女が一番頼りになる。」などと驚いて言った。だれが優秀なのか、意見が違っていたのである。

この食い違いに私たちはとまどった。管理職の人間たちと優れたエンジニアたちにそれぞれだれが優秀なのか、特に優れた成果を優れた方法でやったのはだれかと聞いた。優れていたが途中で問題を起し、マイナスの成果の方が大きかったエンジニアを除外したかった。この調査の結果、両方の立場に共通して優秀と判断されていたのは半分しかいなかった。

私たちはベル研究所で行った最初の研究のため、優秀なエンジニアの定義を「上記の二つのグループに共通していること」(後で行った3Mでの研究のときには、取引先の意見も入れた)とした。ほかに、賞の数、特別昇給の回数、特許、論文も考慮した。このプロセスから得た優秀なエンジニアたちがこの研究の礎となった。

優秀なエンジニアと良好なエンジニアの違いを見つけるため、私たちは役員、マネージャ、エンジニアと他の研究者の意見を聞いた。優秀さにつながる45要素を集め、分類した結果、

- (1) 知能指数、論理思考、理性、創造性などの知的要素
- (2) 自信、野心、勇気、自分の運命を自分で変えようとする考えなどの性格要素
- (3) 他人との接し方、指導力などの社会要素
- (4) 上司との付き合い方、仕事に対する満足感給料とその他の報酬に対する姿勢などの組織要素

の4種類に大別した。

次に、この45要素と優秀さとの関係を調べるために優秀なエンジニアと平均的なエンジニアを全国から数百人集め、2日間のテストを実施した。その他に調査をしたり、何人かについて詳しい事例を調べたり、エンジニアと彼らを採用したマネージャに面接したりもしたし、個人情報も提供してもらった。

2年の研究の結果、私たちが当惑したのは優秀なエンジニアとそうでないエンジニアを見分けるためと考えていた知的要素、性格要素、社会要素、組織要素がなかったことである。ありとあらゆる分析をコンピュータを使って行ったが、定量化できる違いを見つかることができなかった。

しかし、これは大事な発見ではないだろうか？ 優秀さにつながるであろう4種類の要素は、実は関係ないということが分かった。

私たちの努力の価値は長い目で見れば、優秀さに関する迷信の真偽をはっきりさせたことであろう。その後の研究から、他の要素が関係してくることが分かった。ほとんどのエンジニアが立派に成功するために十分な能力を持っているが、普通のエンジニアになってしまう。優秀なエンジニアは能力があるだけでそうなのではなく、その能力を発揮できたから成功したのである。自分の能力を生産力にどうやって生かせるかが問題であり、これはポテンシャルエネルギーをどのように運動エネルギーに変えるのかに似ている。優秀なエンジニアは生まれつきのものではなく、作られるものである。

### 3. 九つの作戦

「自分の頭脳パワーをどのように生産力に生かすか」という質問に対する答えは、私たちの研究が始まる前にはなかった。優秀なエンジニアになる方法は学校でも職場でも教えてくれない。ほとんどは試行錯誤によるものであった。しかし、技術力があっても、生産性の向上の点では失敗して結局普通のエンジニアになってしまう人はたくさんいる。例えば、自発的に、何かを始めようとはしないし、組織にとって価値のないことを始めようとする。

私たちはエンジニアが自分の仕事への取り組み方や、同僚と一緒に働く方法を変えねばならないと分かった。優秀なエンジニアは他の人たちとかなり違うやり方をする。彼らは毎日の業務の中で常にこうしたきらめきをみせる。しかし、普通のエンジニアでも、必要な知識と動機を持っていれば、優秀なエンジニアの能力を自分のものにできる。

それにしても、優れた生産力は突然現れるわけではない。トップまですぐに育て上げる魔法もない。優秀な仕事の遂行は、互いに密接に結びついた九つの戦略に基づいている。これらの戦略を重要な順番に述べていき、全体として優秀なエンジニアのモデルに作り上げる。

#### 3.1 基礎を作る

ヘンリとライの事例はどう思いましたか。ヘンリが技術を第一に考えたことは評価されていないと思いましたが。ライが胡麻すりをし、気に入られたと思いましたが。

ヘンリのような普通のエンジニアは自発性というものが、仕事をより効率的にするための工夫とか、「コンパの幹事などのボランティア」のようなものだと考えてしまう。ヘンリ

は自分が自発的だと考えていた。彼は仕事を完璧にこなすため、最新の技術情報を集め、最新のソフトウェアについて勉強した。「だれかに言われたわけではない」と彼は言った。

ライが分かっているヘンリが分かっていたのは、ある行動しか「自発的」だと認められないことである。「優秀な自発性」というのは

- 本来の仕事をこなしながら、それ以外の仕事を探すこと (ライがパソコンのソフトウェアをインストールしたように)
- 同僚のため、または組織のため、手伝ってあげること (ライが同僚のプログラムを直すために手伝ってあげたように)
- だれの責任でもない、突然表れる仕事を自分から進んでやり、立派にこなすこと
- プロジェクトを始めたら、実現するまで根気よくやり遂げること (ライが残業をして新しいソフトウェアをインストールしたように)

多くの平均的なエンジニアは、オブジェクト指向 Java 言語のような商業的に成功する新しい商品を発明することが価値ある行動と思っている。The Wall Street Journal の表紙で利益が急激に上昇していると掲載されるくらいのものであれば、やる価値がないと思ってしまう。

調査した「優秀なエンジニア」は、画期的なアイデアを常に探しているが、日々の小さな努力も長い目で見れば一つの画期的なことと同じ効果があるという。その上、画期的なアイデアは、それまでの小さな努力の積み重ねによることが多いと言った。小さい努力を重視しない仕事環境を作ってしまうえば、画期的なアイデアは芽生えないであろう。例えば、ライが仕事を手伝ってくれたおかげで、同僚は自分の仕事を画期的に進ませることができたのかもしれない。

優秀なエンジニアは、自分の仕事の経験年数によって自発性が要求されると信じている。新入社員のライは大きい仕事を自発的に始めると期待されていなかったが、小さい仕事を自発的にしたことは、同僚をびっくりさせ、生産力のあるエンジニアであるという評判を確立した。ライは経験を積み積むほどもっと難しくリスクのある仕事を自発的にすると期待されるであろう。

ヘンリ、ライ、その他数百人のエンジニアを観察した結果、能力のある、競争心を持つ人々の集団に入る新人は自発性を示さなければならない。そういった行動は上司ばかりでなく同僚や顧客にまでも良い印象を与える。同僚は自分の与えられた仕事以外のことをやってくれる人がいるとありがたい。新人が一人前の仕事をしないと同僚の仕事が増えると分かっ

ているから、自発的に他の仕事をやるライのような人が欲しい。同僚や顧客に援助の手を差し伸べる人、市場の変化に対応できる人が必要である。顧客も取り引き先の社員に同じような人材を求める。このような期待にこたえられない新人はリーダーとしては見てもらえず、たぶんヘンリのように、有能であるが会社の利益につながるような生産力を持っていないと見られてしまう。

### 3.2 知っている人を知る

ほとんどのエンジニアは同僚と付き合うことの意味をうわさ話を聞くため、または同業者や将来転職したときに世話をしてくれそうな管理職の人と交際するためだと考えている。優秀なエンジニアは更にもっと大切な付き合い方をする。彼らは、現代社会における情報超過の中でも仕事をこなすために十分な知識を持っている人が少ないことが分かっている。大体の人は50%~80%の知識を持っているが残りの知識を得るまで前進できない。優秀なエンジニアは「付き合い方」をうまく利用して仕事を進める。

優秀なエンジニアは、利益に大きく影響する仕事を完成させるために手伝ってくれる専門家と、信頼できるつながりを事前に作っておくことが必要であると心得ている。それにより新しい仕事に取り組むときに生じる知識不足を最小限に抑えることができる。

優秀なエンジニアが作る人的ネットワークは普通のエンジニアのものとの二つの重要な点で異なる。適切な人間が入っている点と、連絡が速い点である。優秀なエンジニアのネットワークに入っている人は、正しい情報を常に提供してくれる。普通のエンジニアが間違った情報をもらうのは、適切でない人に聞いてしまったが、正しい情報を持っている人が自分のネットワークに入っていなかったからである。結果的にはプロジェクトが前進しないか、違う方向に行ってしまうかである。

速いネットワークを持っているおかげで、優秀なエンジニアは、つまったときにほかより速くプロジェクトを前進させることができる。優秀なエンジニアが半日で情報を手に入れることができたとしたら、普通のエンジニアは一、二日がかかった上に、その情報がたいてい間違っている。長い目で見ればその余分な時間は貴重である。つながりがよく速いネットワークは、優秀なエンジニアの生産力を格段に向上させ、同じような能力を持ちながら一人でやろうとする普通のエンジニアより多くの成果を出すことができる。

国際的なコンサルタント会社である Andersen Consulting 社に勤務するクロディオは、締切が迫っている契約案を書くように命じられた。内容は「バイオテクノロジー会社で使われている生物分析評価法のための情報技術支援」を提供することで、その費用は50万ドルであった。

クロディオは業界のトップである Genentech 社に就職した大学時代の友だちを思い出し、彼女に電話した。彼女は更に分析評価法を開発した科学者を紹介してくれた。クロディオは電話をたった2回かけただけで契約を書くための重要な情報を手に入れることができた。

それに対し、クロディオの同僚であるヌットが同じ情報を手に入れるために選んだ手段を比べてみよう。人的ネットワークを利用せずに、ヌットは会社が推奨する方法に従い、社内の電子掲示板に質問を載せた。次の日に返事が40通来ていたが、全部確認が必要であった。返事の多くは他の返事と矛盾していたし、返事をしてくれた人をだれも知らなかったのだとの返事が信用できるものか判断できなかった。ヌットは実質的には何の情報も得られていなく、40もの手がかりを調べなければならぬ状況にあった。結局ヌットがあふれた情報と戦っているうちに、クロディオは優秀な人的ネットワークを利用し、より速くプロジェクトを前進させることができた。

管理職の間で、コンピュータイントラネットを利用し、知識不足を解消する方法が今のはやりである。マネージャはヌットのように社員が電子メールで質問を解決できると信じていて、コンピュータハードウェアとソフトウェアをかうために数百万ドルを使う。しかし、人的ネットワークのような一対一のやり取りの方がコンピュータ技術による一対多数のやりとりより成功することが多い。優秀なネットワークを利用するにはお互いの利益のために知識を提供しあう専門家のグループを形成し、維持したり、グループの中で話し合ったりすることが必要である。技術とはほとんど関係がない。

### 3.3 率先的な自己管理

普通のエンジニアは自己管理を時間とプロジェクトをよく管理することだと信じている。仕事を締め切りまでに終らせ、予算内に収め、要求されたスペックを満足させれば自分は自己管理ができていると考える。

優秀なエンジニアは自己管理が時間やプロジェクト管理だけではないと分かっている。これらは当たり前のことである。優秀なエンジニアは業務戦略のおかげで、率先的にチャンスを作ったり、仕事上の選択を正しくしたり、より優れた仕事をしたり、出世コースにのることができる。それにより、才能を伸ばし、経験を積むことで会社に対する自分の価値を高めることができる。

エレナは自動車産業にその材料を供給している先端セラミックス会社の研究開発部に勤務していた。彼女は生産力と品質管理に関する学会に参加するための旅費の支給を会社に申請した。直接彼女の仕事と関係がなかったので上司は行っても意味がないと考え、旅費の予算もないので認めなかった。エレナは諦めなかった。この学会に参加することにより、

自分の価値を高めることができると信じていたので、休暇を取り、自費で参加した。学会に参加中、ヨーロッパでの品質に関する次期標準である ISO9000 を知った。この標準化の目的は原材料、製品、工程の品質を高め、世界市場におけるヨーロッパ製品をより有利にすることであった。彼女が勤めているような供給会社はこの標準をクリアできなければ、ヨーロッパのプロジェクトに入札できなくなってしまう。

エレナはやる気満々で帰ってきた。仕事外の時間に ISO9000 について勉強し、昼食の間に同僚に説明した。すると同僚も ISO9000 を勉強した方が良くと上司に説得しようとするほどやる気になった。同僚を説得するより、幹部を説得する方が難しかった。幹部はヨーロッパで標準化が決まるとは思わなかったし、決まったとしても強要されることはないと考えていた。しかしエレナはメモや記事を送って、一番乗りになる利点について説得し続けた。やっと幹部は具体的な利点が見えてきて支持するようになった。ヨーロッパは今やこの会社の一番大きい市場であり、品質が良くなったおかげでアメリカの企業からも仕事が増えるようになった。

会社の業績が上がったのはエレナの自己管理から生まれた。上司が協力的でないにもかかわらず、エレナは自分で自分の価値を高めるように努力した。また会社の地位を高めるのにも貢献した。更に、エレナの行動は仕事上の作戦が密接に関係していることを示している。彼女の自己管理能力には自発性も含まれている。皆が利益を得られるように上司を無視しても自分の仕事以外のことも積極的にやり、その上、途中であきらめなかった。

### 3.4 視野を広く

普通のエンジニアの視野は狭い。世界を自分の視点からしか見ず、同じことを何度も言う。それに対し、優秀なエンジニアは自分の意見を置いておき、様々な視点から見るようにする。「これについて顧客はどう思うか」「競争相手はどう思うか」「同僚はどうだろうか」「幹部や株主はどうだろうか」様々な視点から見ると、相対的な重要度を評価できるから、製品を改良できたり、問題のより良い解決法を見付けたりできる。

優秀な視野はパターン認識を身につけるための経験を積んだときに生まれる。サラはコンピュータサイエンス分野の修士学位を修得してからシリコンバレーでソフトウェア開発の仕事に就いた。学生時代や仕事によくある問題の解決法をノートにメモした。毎晩そのノートを読み直し、シャーロック・ホームズのようにヒントやパターンを探した。

経験も実務もあるおかげで他の新人に楽について行けたが、最終的に他の人との差をつけたのは彼女がソフトウェアとコンピュータ論理に関する知識を完全に頭に入れておいたことである。同僚はすぐサラの優れた洞察力を見だし、行

き詰まったときに彼女の助けを求めた。そういった相談を受けたことにより、自分の仕事で目にするのではない問題に触れる貴重な経験を得た。1 年間の過ぎたころ、サラがソフトウェア検査部に移動を希望したことに同僚は驚いた。ソフトウェア検査部は落ちこぼれ社員が集まる二流部所だとしてよく間違っているところである。検査部では他の人のプログラムが正しく動作するかどうかがチェックする。新しい製品を作るときのように自己満足感が極めて少ない。バグを指摘したり品質をチェックして検査部は、悪い知らせを持ってきたりするから、ソフトウェア開発者はそれらに対し半ばあきらめながらも、言い訳しながら我慢しなくてはならない。

しかし、サラは検査部の仕事が、新しく大事な視点から自分の仕事を見つめる機会だと考えた。ソフトウェアが正しく動作しないいろいろなケースについてよく知ることができて、サラは何年分もの経験を 1 年か 2 年の間に得ることになる。彼女は大事な顧客と一緒に、その顧客に合ったテストプログラムを作ることになる。その間に自分が将来にソフトウェアを設計するときに、狭い視点から問題を見るために起きる間違いや、基本的な間違いを避けるようになる。検査は同僚の視点を見る機会も与えてくれる。テストをやっている間にサラはソフトウェアを作るときの同僚のテクニックを覚え、欠点を直した。

2 年後サラがソフトウェア開発に戻ったとき、検査部の経験がすぐに効果となって表れた。彼女の同僚は直ちにサラをソフトウェアの女王と呼ぶようになり、業界でもソフトウェアの神様と思われるようになった。その結果、自分の会社をシリコンバレーのリーダーに持ち上げることに貢献した。サラのような視点のおき方の微妙な違いをマスターした優秀なエンジニアは生まれつきそうではない。視点について研さんを重ねて初めて体得できるのである。

### 3.5 正しい仕え方

普通のエンジニアは「仕え方」、つまり上司や幹部の人とうまくやっていくということは、彼らに自分は規則を守り、彼らの命令に従い、決してリーダーの地位を脅かさないと分かってもらうことだと信じている。

優秀なエンジニアはもっと有利な仕え方を早くから覚える。それは良い右腕になることである。つまり自分で点を取るより、アシストをする方が重要なことが多い。組織やリーダーを成功させるように手伝いながら、やるべき事とそのやり方について独立的で、かつ鋭い判断力を発揮する。そのような優秀な仕え方をする人は、性格や職場の違いがあっても組織の目標を達成するためリーダーと一緒に協力的に働く。

たいていの人は、優秀な人がリーダーか、中心的な存在だと考える。しかし、実際にその考えが間違っていたのは驚くべきことであった。多くの場合、優秀な仕え方をする人は、

リーダーに問題を指摘したり、話を聞いてあげたり、リーダーの判断に疑問を持ったりすることにより、リーダーを支援する。

多くの技術関係会社では、顧客が欲しいと考えているものと、エンジニアが一番良いと考えているものと、どちらを提案するかを慎重に決める必要がある。顧客はただの大衆車が欲しいのに、うちのエンジニアはロールスロイスを作っていると上司がばやくことをよく耳にする。エンジニアは最高の物を作るための能力を持っているので、期日の遅れや予算オーバーの可能性があっても余計な物を付けたがる。

このような場合の一つとして、ベル研究所の優秀なエンジニアは、この余計な物について口うるさい上司と対立した。上司は顧客に喜んでもらうため、電話交換機に簡単な通話ルーチング機能をつけて顧客に予定より早く納入して欲しかった。「付加機能を忘れなさい。顧客は一か月後の最高な製品より、基本的なものを今すぐ欲しがっている。」と上司は言った。優秀なエンジニアはそうとは限らないと言って、上司と一緒にこの顧客と市場内の他の顧客のために製品の持つべき短期的と長期的な目標をもう一回見直した。

優秀なエンジニアは「確かにこの顧客となら短期的な利益があるかもしれない。が、危険もある。高級市場をねらっているのに低級と見なされてしまうかもしれない。今もう少しの努力をすれば、他の顧客のための開発時間を減らせる。顧客にどちらが良いか聞いてみよう。」とやった。

この優秀なエンジニアは上司の心配が分かっていた。同時に皆が共に描いている、もっと大きい目標に上司の目を向けようとした。可能な限り優秀な仕方をする人は会社の目標に合うように自分の努力を合わせる。さもなば、自分に合う組織を探す。

### 3.6 チームワークは共同所有

普通のエンジニアは「チームワーク」がプロジェクトを成功させるために他の人と協力し、自分の役割を果たすことだと考える。

優秀なエンジニアはもっと高度に考える。チームワークはゴールの決定から、グループの責任、活動・スケジュールやグループの仕事の完成までを「共有」する複雑なスキルの流れだと考える。グループが円滑に機能するように明確な貢献をすることもチームワークの一つである。つまり、皆がチームの一員であると感じさせたり、チーム内で対立が起きたときに対処したり、問題を解決するために他のメンバーを手助けしたりする。

ある医療器機メーカーは、納入したばかりの最新の集中治療モニタの相次ぐ故障に病院側が激怒していたので、緊急対策チームを結成した。機器はでたらめに緊急警告を出したため、患者にも、急いで対応に駆けつけた医療スタッフにも、

この誤動作は大きなストレスとなった。

対策チームのメンバーは製造、研究、サービスを含む五つの部署からの専門家で構成された。この7人の中で優秀なのはアイデンだけだった。彼は顧客に対するサービスを勉強するためにサービス部に移動していた。最初の会議が始まって3時間たったころ、すぐやらなければならない対応について激しい討論になった。入社25年間の経験を持つ53歳の製造エンジニアであるユウイングは、修理担当を病院に派遣し、現場で修理を行う方が良いと主張した。しかし、研究部の新人であるジュリは、鎮痛剤タイラノール事件のときのJohnson & Johnson社の対応事例を基にすべての機器をリコールした方が良いと主張した。

議論は続いた。ユウイングとジュリの議論は更に白熱し、けんかになりかけていた。アイデンは皆がいらしていることに気が付き、何人かの同意をとりつけた上で「10分の休憩をとってから解決方法を探したら？」と提案した。

会議が再開したときにアイデンはある提案をした。ジュリにユウイングの立場から、ユウイングにはジュリの立場から議論するように頼んだ。ジュリとユウイングはそれをためらったけれど、これによりますます大きくなる緊張と怒りは鎮めることができた。その後、他のメンバーも意見を交換し始めた。経験はあるが内気な設計者のエロイーズはそのときまで一言も言わず、大人しく隅の方に座っていたが、小さい声で「すべての病院が苦情を言っているわけではないので、最初になぜこの病院の機器が誤動作するのか調べよう。たまたま初期不良なのか、納入先で何かが起っているのかもしれない。すべての機器をリコールするより問題のあるものだけをリコールし、病院で例えば、高磁場が何か原因があるかどうか情報を集めたらどうでしょうか」と言った。

エロイーズの意見にだれもコメントしないまま議論が続いた。数分後、アイデンは「皆にエロイーズの提案が聞こえなかったかもしれない。彼女の提案はこの問題を解決へ導いてくれるかもしれない。もう一度説明してくれる？」とやった。

それでエロイーズはもう一度提案をした。アイデンはこの提案は顧客の意になかった対応であるし、すべての機器をリコールするよりコストがかからないと言った。他のメンバーも妥協し、エロイーズの提案に賛成し、他の議題に移った。

アイデンの仲裁がなければ、ユウイングとジュリの激論が続いたかもしれないし、エロイーズは提案を言えなかったかもしれない。またチームが長期間にわたり議論を続けたかもしれない。サービス部の代表という立場を越えたことにより、アイデンはチームの効果を上げることができた。

### 3.7 「小さい」指導力

普通のエンジニアは「大きい」指導力に魅了させられる。「大きい夢」、「大きいカリスマ」、「大きい成功」を持ってい

る人は生まれつきそうであり、指導をとること、重要な決断を下す力を持つこと、興味のないことを他人にやらすことでエゴを見せびらかそうとする。

それに対し、優秀なエンジニアは指導力というのは何人かに力を合わせてもらい、大きな仕事に取り組むように説得する戦略だと考える。その指導力は様々な努力のたまものである。例えばグループにはっきりとした目的を作らせ、その目的を達成するための信頼と責任を持たせること、達成するための資源を見つけ出すことや、プロジェクトを成功に導く努力が必要である。

とても賢くともが、指導力を全く発揮できない人もいる。知能以外に欠かせない能力が「小さい」指導力である。「小さい」指導者は他人との結びつきをよく理解している。「大きい」指導者は自分の考え、自分の仕事ぶり、自分のゴールにピントを合わせすぎである。「小さい」指導者はチームメンバーの必要とするもの、能力、希望、力を考慮する必要があることが分かっている。

自分以外のところにもピントを合わせることは生産力の向上につながる。それは、「小さい」指導者が指導したい人間の直接の上司ではない場合が多いからである。「大きい」指導者はこの現場の実情をよく見落とす。同僚が指導権を取った人に従うには、その人が自分のためだけでなく、皆のために働いていると示す必要がある。このためには「大きい」指導者が時間の無駄だと考えているような交流が必要なこともある。カリスマがある「大きい」指導者より、毎日一緒に仕事を頑張ってやり、締切に間に合うために夜遅くまで働いてくれる「小さい」指導者の方が信頼性があり、忠実になってもらえる。

優秀な指導者から教わった、「小さい」指導者と、「大きい」指導者や普通のエンジニアとの差を付けるための秘けつは、他の人のことがすべて分かっていると決して思わないことである。それに対して「大きい」指導者を「神」と考える人がいる。つまり、「大きい」指導者は神のように部下や今の状況に何が一番良いかが分かるというイメージがある。

優秀な指導者は、既に事実が分かっていると思っけていても、必ず確認すると習慣を持つ。在米のドイツのソフトウェア会社に勤務するアニシアは、同僚のことが分かるまでプロジェクトを始めない。6人の同僚を指導し、新しいインターネットソフトウェアを作るプロジェクトが任命されたとき、最初のミーティングで時間をかけて仕事の分担についてプロジェクトメンバーに聞いた。「ジョン、前に一緒に仕事をしたとき、ハードウェアに関する仕事をもっとしたいと言っていたよね。今でもそうですか。このプロジェクトの大きな部分はハードウェアだよ。」

鋭い心理学者のように自分の先入観を入れないで、どう答

えても良いようなあいまいな質問をした。それにより、皆が自分の能力やこのプロジェクトでやりたいことを話しやすいようにした。結果として、分担を個々の能力や興味により近くに合わせる事ができた。同僚を決まった仕事に閉じ込めたくなかった。もちろん従業員は欲しい物がいつでも手に入るわけではない。しかし聞く耳を真剣に持ち要望に答える努力をすれば、正式な指導権を持たない「小さい」指導者にも影響力が得られる。そうした努力はプロジェクトが行き詰まり、ストレスがたまったときに役立つ。技術的に優秀な実績を持てば、当面の指導者をやらせてもらえるかもしれない。しかし「小さい」指導者は、権力の影響力が個人的な人間関係にまで及ばないと分かっている。そこで「皆が力を合わせて頑張ろう」という姿勢を持つとする。

ソフトウェアデザイナーであるアニシアが「小さい」指導者を務めたインターネットプロジェクトは大成功だった。毎年行われる功績賞のパーティーの席で、北米支社長が「アニシアらしい、素晴らしいもの。」とほめてくれた。アニシアが過去に成功まで導いたプロジェクトと比較した。それからアニシアのような社員が500人いたら、アメリカ市場独占は確実だと言ひ、アニシアにスピーチをしてもらうためにステージに呼んだ。

アカデミー賞をもらった自己中心な俳優のように、アニシアは上司やプロジェクトメンバーについて典型的なほめ言葉が言えたがそうはしなかった。その代り、プロジェクトメンバー全員をステージに呼び、メンバーの一人にメンバー全員の紹介を頼んだ。それからアニシアはマイクをとり「このプロジェクトは我々の努力の結果です。各メンバーの貢献がなければこんなに成功はしなかったでしょう。我々は誇りに思っている。皆さんもそう思っていることがとても嬉しいのです。」と言って全員でおじぎをした。

### 3.8 組織を知る

普通のエンジニアは出世するために「ゴマすり」が一番良いと信じ込んでいる。その上、社内政治を気にしすぎるか、全く無視する。

優秀なエンジニアは、大きな組織の中には、ある程度の競争意識が必要であることが分かっている。協調性を持っているおかげで、そういった競争の中で方向を定め、協力を推進し、時には衝突と立ち向かいながら、仕事を進めることができる。協調性に含まれるのは個人やグループの管理能力、争いを避けた方が良いときとぶつけていった方が良いときの区別をする能力や、敵になりそうな人を味方にする能力である。「視野を広く」の戦略で紹介されたソフトウェア開発のサラを思い出して欲しい。サラは検査部で働きたいと自分から志願し、同僚からは彼女はおかしくなったのではないかと思われた。サラは将来交流をもつであろう人間を見つけ、仕

事上の関係を作り始めた。このつながりは相手に対しサラの評判を良くし、将来の交流がやりやすくなった。

「率先的な自己管理」の戦略で紹介したエレナは協調性を利用し、彼女の会社を ISO9000 とヨーロッパ市場の可能性に目を向けさせた。エレナはまず食事のときに学会で聞いたことを同僚に説明した。知識が増えたときに詳しい説明会を行った。上司に会社にとっての ISO9000 の利点を説明したり、幹部にも関連した記事や、売上と利益の可能性に関するメモを流して説得し続けた。もちろん幹部とコンタクトを取る了解を上司から得たことである。それからエレナは他の社員にヨーロッパの入札方法を教えた。自分の考えを推進すると同時に会社のやり方を守りながら、その考えを会社の将来に結び付けた。

### 3.9 適切な発表

普通のエンジニアは、巧みなプレゼンをして管理職に良い印象を与えることや、長いメモや自分の仕事を皆に自慢することが「発表」であると考えている。聞く側ではなく自分のイメージと言いたいことを中心に考えている。

優秀なエンジニアは、「どの情報をだれに伝えるか」を決め、説得しようとする相手に合わせた効果的で分かりやすい形で情報を伝えるという一連の能力を利用できる。発表において最も難しいのは特定の人にその人に合った内容を選ぶこと、または特定の発表内容のために正しい聴衆を選ぶことである。

適切な発表は絶対に必要である。現在の社会は、自分のアイデアを伝えることが苦手な人にとって厳しい社会である。大半のエンジニアにとって、発表といってもそれは Bill Gates や Billy Graham が大きな講堂でマルチメディア効果を駆使に数千人の前でやるものではない。大体の発表は小さい会議室で 5~20 人のグループですることが多く、たまに講堂でやるくらいである。聴衆は同僚、管理職または顧客で、内容はたいがい商品についての技術的な話である。

優秀なエンジニアにとって、発表はもっと高度なものである。私たちの研究から、優れた発表は、情報をただ伝えることではなく、情報を理想な形に整えて伝えていくと分かった。私たちが見た優秀なエンジニアはその特定の聴衆に合ったメッセージを流し、自分の考えを受け入れてもらい、それから批判をうまくはね返す能力をマスターしていた。

普通のエンジニアがよく失敗するのは、ただ情報を伝えることだけを考えて、その後の反響をまでも考えたメッセージの伝達ができないことである。聴衆が変わっても発表のやり方を変えようとしない。

有望 500 社に入るある企業の労使担当は、組合と契約を更新する際、医療費削減を交渉しなければならなかった。しかし、やり方がうまくいった。契約は会社の役員にも組合にも認

めてもらわなければならなかった。彼がとった方法は同じ情報を言葉巧みに変えて伝えることだった。最初は組合の中間管理職数人に現場で一週間をかけて少しずつ情報を伝えた。簡潔で読みやすい資料を配った。資料はコピーを一般の組合員に配ればすぐに理解できる可能性が高いものを作った。一番重要なポイントは新しい医療プログラムに切り換える代わりに、会社が節約できたお金を使い、古い工場を改良することである。その結果、会社がより競争的になることによりリストラの可能性を減らすことができる。

彼が代表取締役と副社長らのために行ったプレゼンには基本的に同じ情報が含まれていたが伝え方は全然違っていた。まず組合の場合ほど時間をかけることができなかった。言いたいことのほとんどすべてを整然とそのまま資料に書き出した。その中に契約を促す説得力のある推薦も入っていた。この資料を詳しく説明するため、代表取締役と副社長と 3 人で社用飛行機の中で一時間のミーティングの時間が与えられた。

代りのものを何も与えないで管理職が医療プログラムの変更を要求したら組合が反発し、新しい契約はほぼ不可能だと彼は主張した。会社も急成長の時期に入ったばかりの今、長いストライキに入ったら株主は喜ばないと彼は指摘した。

双方から彼の提案を非難する声が挙がったけれど、この優秀な交渉人は交渉の土台をうまく築きあげた。最終的には管理側と組合側双方は少しだけの修正で提案を承認した。

この例から学べることは、優秀なものと普通のものとの差をつける最も重要なことは聞き手を知り、メッセージをその人たちに合わせることである。

ミアラは画像伝送のためのソフトウェアを設計する仕事に就いている。具体的には X 線写真、心電図、監視カメラの生中継などの映像を、電話回線を通し、病院の救急手術室との間に送受信するシステムである。最新バージョンを救急病棟の医師や病院の役員に紹介するために短いビデオを使った。ビデオの内容は車が急ブレーキをかけている場面、救急車がサイレンを鳴らしている場面、小さい子供が救急治療室に運び込まれている場面、そしてミアラの会社の機器を利用して小さい命を助けるために数分しかないと言っている場面で構成されていた。

ミアラは「私たちの機器はこの子供やあなたたちの子供の命を助けるきっかけになる可能性がある。」と言った。「システムを作っている間にこのビデオを何回も見て、最善を尽くすことの大切さを思い出した。あなた方にも見せてあげたい。」

前のバージョンと比較するため、ミアラは心電図と一緒に救急治療室で聞こえるような心音を使った。まず前のバージョンを実行した。画像が画面に現われる前に時間切れに



なってしまう、心音が止まり、救急治療室の警報が鳴った。新しいバージョンを実行したとき、画像がもっと早く現れ、時間切れにならなかった。

それからミアラはプロジェクトの経過を説明した。ソフトウェアを速くするために成功した方法と失敗した方法を紹介し、その理由も説明した。人の命を助けるために働いている医療専門家の仕事と技術的な点を結び付けた。自分の子供が救急室に運ばれたときの恐さを実感させ、ミアラの会社の製品を正しく動作する必要性を知らせた上でミアラは新製品の価値を余すところなく伝えた。

#### 4. 優秀になる

私たちが開発した生産力向上プログラムを受けたエンジニアを対象に、長期の調査を行った。プログラムを通し、優秀な戦略を習う前と習った後の生産力の違いを調べた。現在このプログラムはアメリカだけでなく、ヨーロッパを含めた会社員 1,000 人以上に教えられた。研修会社も採用しているし、大学で学生にも教職員にも教えられている。

評価の基準として、マネージャ、優秀な社員と普通の社員に自分の部署で生産力向上の要因を書いてもらった。何回か繰り返すうちに、この要因で高い評価を得たものは本当に高い生産力を持っていることが明らかになった。それからこの要因を基に参加者 300 人と非参加者 300 人を彼らの直接の上司に 2 回（研修の前と研修が終了した 8 か月後）評価してもらった。

この評価によると、プログラムの参加者は明らかに生産力が向上していると分かった。プログラムを修了したエンジニアはより速く問題を解決するようになったし、より品質の高い仕事をして顧客をよく喜ばせた。

このプログラムはできそこないのための補習授業ではない。プログラムを受けた人の約 30 % は既に優秀だと認められていた。もともと優秀な人でも同じように生産力が向上してきた。

上司の研修前後の評価からすると、一番大きな変化があったのは女性と白人以外の人たちであった。平均して生産力は 4 倍向上した。この人たちの成功は私たちの研究から出た重要な結果に基づいている。

優秀になるために魔法はいらない。エンジニアが普通の仕事しかできないのは能力がないからではなく、高い生産力につながる戦略を習わなかったからである。この戦略を身につければ生産力は上がる一方である。

#### 5. おさらいクイズ

優秀さに関する下記の文章を読んで正しいものに「○」をつけ、間違ってるものに「×」をつけて下さい。

1. 優秀なエンジニアは生まれながらであり、なるうとして

なれるものではない。

2. 優秀なエンジニアは普通のエンジニアより賢い。
3. 優秀なエンジニアは他の人より努力や意欲がある。
4. 優秀なエンジニアは他の人より指導力がある。
5. 机をきれいにして仕事をする人は、机が汚いまま仕事をする人より優秀になる可能性が高い。
6. 詳しい時間管理と組織力は高い生産力へのカギである。
7. 優秀なエンジニアは普通のエンジニアより長く、かつ一生懸命に働く。
8. 優秀なエンジニアは普通のエンジニアより自分の仕事に満足している。
9. 優秀なエンジニアが成功するのは主に社内政治に関心を持ち、幹部に格好良い発表をするからである。
10. 白人以外の人たちや女性は、OB ネットワーク等がよくつながっている人間ほど生産力を上げることがなかなかできない。
11. 人間は長期間にわたって 2 倍の生産力向上率を維持することができない。
12. チームを作るなら優秀なエンジニアを一人とその人を支援する普通のエンジニアを 4 人選ぶより、上位 5% に入るエンジニアを 5 人選んだ方が良い。

答え

- 1~11 は「×」
- 12 の正解はない。成功に導くのは一人の優秀なエンジニアなのか、それとも優秀なエンジニアとなりうる良いエンジニアのグループなのか、あなたがどちらを信じるかによる。現在の非常に競争的な世の中では後者の方が良い可能性が高い。