

訳者あとがき

本書はAlan Cooper, *The Immates are Running the Asylum* (SAMMS, 1999) の全訳である。タイトルを直訳すると、文中の節のタイトルにも使われている「キチガイの仕切る精神病院」だ。ソフト開発、そしてソフトを使う各種の製品開発すべてが、そう例えられても仕方のない状況になっているのだ、というのが著者の主張である。そして著者は、それをどう解決したらいいのかについて、自分なりの解決案を出している。

本書のおもしろさ：診断編

ソフトウェアに使いにくいものが多い、というのはだれもが思っていることだ。さらにソフトがバージョンアップされるたびに、どうでもいいよけいな機能が増えるばかりで、ひたすらでかく、遅く、不安定になるばかり、というのもよく経験することだ。なぜそういうことが起こるのだろうか。

さらに、ソフトだけでなく、ソフトを使った各種の製品も同じ傾向を示すことは、われわれだれしも知っている。本書でも例としてあげられている、多機能テレビとか情報端末なんかとか。あるいは携帯電話の機種交換をして、えらく面食らったおぼえのある人は多いだろう。携帯に力二力二ならゲームがついて、どうしろっての。携帯を電卓として使おうなんて考えるやついないって。こんなものいらないから、そ

このファイルは、株式会社翔泳社が発行する書籍『コンピュータは、むずかしすぎて使えない!』（アラン・クーバー著、山形浩生訳、定価：本体2,200円＋税、ISBN 4-88135-826-X）の内容見本です。商業的利用を目的とした複製・再配付を禁じます。この本のお買い求めはお近くの書店、または翔泳社のインターネットショップ SEshop.com (<http://www.seshop.com/>) へご注文ください。この本の内容に関するお問い合わせは、826-x@shoeisha.co.jpまで電子メールでお送りください。

の分安くしてくれればいいのに。つい先日、ぼくの勤め先の部で携帯電話を一斉に交換したときに、みんな「バー」文句を言っていた。

さて、文句を言いつつもわれわれはなんかするか？ もちろんしない。仕方ない、そういうものだと思うている。それと一方でわれわれは、自分がわかってしまったことについては、結構得意だったりする一方で、「こんなことがわからんヤツはバカだ、努力が足りない」と急に威張ってしまったりする。わかんない人はあきらめて、わかる人はわかれて得意だから文句を言わない。結局だれもはつきりとは文句を言わない、不健全な状況ができていく。

本書はその前半を使って、そういう状況についてかなり克明に描き出している。そしてその原因についても、非常に説得力のある議論を展開してくれる。「機能はとにかく多いほうがいい」「いらなきゃ使わなければいい」という信念で、その場の生煮えの思いつきがどかどか詰め込まれる状況。できあいのソフトをいい加減に切り張りして作られるために質の低下を起しているソフト。もともとおたくで人付き合いが苦手だったうえに、コンピュータと同化しすぎていて、普通の人の普通のニーズがまったく見当もつかなくなっているプログラマたち。マイクロソフトやジェネラル・マジックを含め、豊富な事例を挙げつつ次々に繰り出される議論は、読んでいて（訳していて）ニヤニヤさせられるところだ。途中で出てくる、ソニーの開発したインフライト用システムのオリジナル案（悪い見本ね）なんか、あまりにいかにもすぎて大笑いさせてくれる。ここらへんは、ソフト業界の重鎮の一人たる著者の面目躍如というところだ。

また本書のおもしろさは、プログラミングだけでなく、産業としてのソフトウェア業界全体を見ている点にもある。確かにプログラマが好き放題やっているのも問題だ。でも、それをきちんと評価して律する力を失っている管理職も困ったものだ。なんでもいいからあわてて市場に出そうとする企業、市場の求めているものを理解しようとさえしない経営者、とにかく機能一覽を増やすことしか考えていないマーケティング。そしてもちろん、さっき挙げたような、それに対して文句を言わない顧客。それらが全部あわさって、いまの惨状を作りだしている。

このままじゃダメだ。なんとかしなきゃいけない。というわけで、著者の議論は後半へと進む。

本書のおもしろさ：療法編

じゃあそれに対してどうすればいいのか。著者の答は、実に簡単だ。ソフトを書きはじめる前に、まずデザインしろ、というのが著者の基本的な答えだ。ソフトとして必要な仕様をきちんと事前に決めて、それからコーディングにかかれ。

……と書くと、「ああなんだ、仕様書をきちんと書けということか、それなら弊社はバッチリ」と安心してしまふ人がいるかもしれない。でも、そうじゃない。本書のデザイン仕様書というのは一般のソフト会社でやられているような仕様書ではない。というのも、通常のソフト開発での「仕様書」というのは、ユーザ

に対しては、本書に挙げられている買い物リストに毛が生えたような代物でしかない場合が多いからだ。

一方で一部の大ソフト会社などでは、実際のコード書きはすべて下請けませといところが多い。で、かれらが言う仕様書というのは、その下請けに対する指示書のことだったりする場合が多い。ちなみにそういう大ソフト会社の人たちは、実はまともな意味でのコードが書けない場合も多々あって、しかも「われわれはそんな下流の仕事はやらない、上流の付加価値の高いところをやる」とかほざいて胸を張ったりする例さえ散見される。

で、結局のところ実際にかれらがやっているのは、どっちつかずの中途半端な代物で、中身はすべて下請けませ（というの、本書で指摘されている通り、コードを書くやつが実はかなりの実権を掌握してしまふからだ）、でもユーザーの側に対しては買い物リストみたいなものだけで話を進めて、できたシステムの説明をするからといので言ってみると、データベースがどうなっていてそれをこの時点でどのルーチンがいじって、だの、このところでマスターに書き出して、だの、そんなことをこのオレに聞かせてどうするんだ、というふうなことをまくしたてて平然としているケースが多々ある。らしい。

でもって、できたシステムのレスポンスが遅くてつかいものにならない、という場合に、こいつらはさっきの買い物リストをふりまわして、レスポンス時間は仕様書にない、といって逃げを打とうとするだけ。らしい。

ここで言っているデザインというのは、そんなものじゃない。機能一覽表でもなければ、コーディングの指示書でもない。ユーザーの操作とそれに対するレスポンスを明確化したもの、とても言おうか。具体的にそれを使うユーザー経験の記述のようなものだと思うってほしい。

そしてそういうデザインをするための手法をかれはいくつか紹介している。それを使う人の目標と作業を仕分けすること、それを使う人の顔をペルソナの設定で明らかにすること、さらにそれが使われる場面に具体性を持たせるシナリオだ。この、シナリオに基づいたソフトデザインという技法は、それなりに勢力を持つてきているようで最近になって方々で見かける。

で、これで万事解決、とクーバーは言うんだが……

それがこの本の弱いところなのだ。確かに事例を見ると、ペルソナやシナリオを活用して利用者や利用場面をはっきりさせることで、ソフト製品がかなり改善されたのがわかる。そこでのデザイナーの役割というのがイマイチ見えにくいのだ。優れたデザインは、あとから見ればあたりまえにしか見えない、とクーバーも本書で書いているから、こういう書き方はたぶん不公平なんだろう。でも、こうして見る限り、デザイナーをわざわざつれてこなくても、どれもプログラマたちが自分で十分にできそうな話に見えてきちゃうのだ。

さらに、デザインしてうまくいった例はうまくいっただろう。でも、どんなデザインでもいいのか？ そんなはずはないだろう。デザイナーのデザインがそれでいいのだ、というのをどうやって見分けなければいっだろう。デザインだって、いいデザインもあれば悪いデザインもある。有能なデザイナーもいれば、有能でないデザイナーもいるだろう。でも本書は「デザイナーができればそこらへんは仕切る」「デザイナーはそういうプログラマをおさえられる」「デザイナーはユーザーのニーズをきちんと把握できる」と述べて、デザイナーがいればなんでもオッケーと言わんばかりの書き方をする。

じゃあ、その何でも解決してくれるすごいデザイナーさんってのは、どこにいるんだろうか。

はじめのうちは、要するに著者の会社を雇え、と言っているようにしか本書は読めない。これだけの技能をそなえた人間がそうそう転がっているとは思えない。そして確かにクーバーはデザイナーはだれでもなれるものじゃない、というようなことも言う。プログラマはデザインはできない、と著者は主張する。そりゃそうだろうなあ、単に観察力や分析力だけでなく、すさまじい対人能力、弁舌、交渉力と調整能力、政治力がないとこんな仕事は動まるまい。たぶんウチの小淵総理なんかじゃ無理だぞ、これ。でもそんな技能保持者を養成しないとダメなら、ソフト製品って永久に絶望なんじゃないの？

ところが本書は最後になって、「権限をやると言えば、だれか手を挙げるからそいつにデザインさせればいいんだ」と言わんばかりの主張をする。ええっ、そんな手を挙げたやつにホイホイやらせていいのあ？ その人がこんな仕事をこなせるかどうか、どうしてわかるの？ その人がやった仕事が優れているかどうか、どうしてわかるの？

その意味で、本書がユーザビリティ試験をかなりこきおろしているのは興味深い。このユーザビリティ試験については、ランダワー『そのコンピュータシステムが使えない理由』（アスキー出版局）に詳しい。この本の主張は、ソフトウェアというのはつくりっぱなしで、ユーザーテストをして使い勝手を確かめないし、まともな意味でユーザーの声を反映しない、だからあんな使いにくいものが次々に出てくるんだ、とい

うことである。いまのソフトはひどいけれど、作る側はそれを聞いていない。それを数字で見せてやって、さらに作るにもなんども定量テストをして、その結果を作る側にフィードバックしなきゃいけない。それでソフトを改善するんだ、というのがユーザーリティ試験の考え方だ。

これに対して本書の批判というのは、定量テストであるインターフェイスが悪いのがわかってても、それをどうなおせばいいかはユーザーにきいてもわからない、ということだ。デザイナーには、どこをどうなおせばいいか提案ができる。でも、ユーザーリティの専門家は、「これは六〇点」と言えても、それをどうなおせばいいか提案できないじゃないか、とクーパーは主張する。

確かにそうなんだが……でも一方では、クーパーはなぜデザイナーにそういう改善提案ができるのか、きちんと説明できていない。だれもひどいインターフェイスを作ろうとは思っていないのだ。かれが批判するプログラマだって、いいインターフェイスをつくらうとは思っているはずだ。でも、ろくでもないものを作ってしまう例が多々あるのも事実。デザイナーだって、ろくでもないものを作ってしまうこともあるだろう。それについて、クーパーはどういう手をうつんだらうか。本書ではそれについては触れないばかりか、それをテストしようというユーザーリティ試験をかれは否定してしまう。

たぶん実際には、両者のバランスが必要になるのだろう。どっちもいるんだらう。デザインもいるけれど、それに対して定量的な評価もいるんだらう。これまでの製品開発ってのはそういうものだった。ソフトだつてそのはずだ。なぜクーパーが定量評価を否定したがるのかについては、本書ではよくわからないのだ。

著者について

著者、アラン・クーパーについては別のところに詳しい著者紹介があるのでそちらを参照してほしい。ピジュアルベースシックスの父、というのがいちばんポピュラーな紹介方法だろうか。

ちなみに、文中で安易に作られたソフトや、できあいのパーツを寄せ集めてデザインなしに「既存のものがそうになっているから」というだけの理由で作られているソフトに対し、著者は痛烈な批判を浴びせている。しかし、そうしたソフト蔓延の理由の一つは、まさにピジュアルベースシックスの存在だったりする。そういうくだりを最初に読んだときは、「おまえが言うかあっ！ あんたが諸悪の根源だろうが！」とつっこみそうになったものだ。でも、創始者が問題点をきちんと指摘して、できれば改善してくれないと、後の人たちがそれをやるのはとてもむずかしいというも事実。クーパーとしても、なんとか事態を改善したいとは思っているのだろう。と、ここでは善意に解釈しておく。

本書の(意図せぬ)おまじなれ

さて、この本の主張が正しければ(かなり正しいだろう)、これはもう一つとても重大な意味をもつてくる。最近はやりの、Linuxをはじめとするオープンソース・フリーソフトに対して、本書の議論は大きな疑問をなげかけるものだからだ。

ソフトウェア関係者で、いまオープンソース・フリーソフトというのがなかなか知らない人は、観察力と情報収集力の著しく劣る、かなり問題のある人物だろう。詳しくは本書でも引用されているエリック・レイモンドの『伽藍とバザール』（光文社 ISBN:485421886）を参照してほしいのだけれど、基本的には、ソースコードを完全に複製・改変自由な形で公開し、興味のある人がみんなでそれを改善・拡張することで優れたフリーなソフトをつくらう、という動きだと思えばいい。むかしからUnix系の多くのソフトはそうやって発展してきたし、インターネットの基幹部分にはずっと使われ続けているけれど、特に近年になってLinux、FreeBSD、Apacheなどかなり目に見える形でそうしたソフトが躍進してきて、一部では商業ソフトのシェアを喰い、さらにネットスケープ社のブラウザなど、商業ソフトだったものを公開に切り替える動きまで近年は見られるようになってきて、大きな注目を集めている。

さてみんながよってたかってコードをいじるし、方向性は定まらず、ハッカーたちが自分のやりたいように勝手な変更を加えるというフリーソフトは、いわばクーパーの言う傷跡だらけなわけだ。他人の書いたコードを持ってきて、自分に必要なところだけ変えて使うというのは、クーパーが批判するコード再利用の最たるものといえる。この点についてクーパーが正しければ、オープンソース・フリーソフトが大規模ソフト開発でまともに使え見込みは皆無で、LinuxやApacheはガタガタのゴミクズになっていなければならないだろう。

しかしこの点について、クーパーの批判はあたっていない。フリーソフトのある程度の部分（すべて、と

はいわない）は、類似の商用ソフトよりも安定性が高く、高品質だったりする。したがって、かれの批判で見落とされている何かがあるわけだ。おそらくクーパーはそれが、Linuxの場合にはもともとのUnixというデザインだ、と主張するだろう。すぐれたデザインに従ってみんなが分担して作っているから優秀なのだ、と。それがどこまで妥当性があるかは、ちょっとまだ検討の余地があるはずだし、その他のソフトの場合にはいろいろ疑問の余地はあられるだけだ。

しかしもう一つ、クーパーの批判の射程に入っている部分がある。現在オープンソース・フリーソフト系のOSはかなり普及を見せしており、クーパーの分野ではきわめて有力な勢力となっている。だがそれがデスクトップでも優位性を発揮できるか、というのがいまでは議論的的となっている。LinuxにせよFreeBSDにせよ、あるいはフリーソフトとして出回っている各種のソフトにせよ、みんなコンピュータの知識があるハッカーは喜んで使っている。そして多少は一般ユーザーよりも知識のある（つまり比較的ハッカーに近い）システム管理者が主にいじっているサーバー部門では、そうしたソフトは大きな成功を収められた。ハッカーは、ハッカーのためのソフトは作れるのだ。でも、今後デスクトップで成功するためには、非ハッカーでも使えるようなもの、それもコマンドラインのインターフェイスなんか絶対無理だから（岩谷宏がなんと言おうと）、まともなGUIをもったデスクトップとアプリケーションをそろえなくてはならない。

前出のエリック・レイモンドも、これをオープンソース・フリーソフトにとつての大きな挑戦だと指摘している。でも、その見通しについてはきわめて楽観的だ。「オープンソース・フリーソフトは、これまで不可能とよばれてきたハードルをすべて突破してきた。GUIだけできないと考えるべき理由はない！」とかれは述べる。まあかれは、フリーソフトの宣伝係でもあるので、そこらへんは割り引いてきいたほうがいい

だろう。しかしクーパーはそれをきっぱり否定している。GUIだけではできないと考えるべき理由はある、オープンソース・フリーソフトは一般ユーザーに使いやすいシステムは作れない、と実質的に宣言しているに等しい。さつきも書いたが、そもそもソースコードをいじってプログラミングをしようなどというハッカー連中は、ふつうの人間とはちがっていて、一般人が何を望むかわからないからだ。

これは『コンピュータシステムが使えない理由』でも指摘されていた。そこで挙がっていたのはハッカーたちは平均よりかなり頭がよく、また空想的な記憶力や暗算能力が常人とくらべて明らかに優れている、ということだった。わかっちゃう人には、わからない人がなぜわからないのか、理解できないのだ。コンピュータの仕組みがちよっとわかっている人にとって、セーブしてないファイルは消える、なんてことは火を見るより明らかで、「セーブ忘れてファイル消えちゃったんだけど、なんとかならないかなあ」なんて言うてくるやつは思考能力皆無のバカ以前の存在にしか見えない。

あるいはUnix系のハッカーの多くは、E m a c sという異様なエディタを愛している。知らない人にはまったく使えない、無愛想きわまる代物で、みんながそれを好むのは、半分以上はそれがむずかしいからで、投入した勉強投資を否定したくないからだったりする。E m a c sを嫌う人もいるけれど、その人たちはもっとひどくて、v i tという一般人には石器時代の遺物としか思えないものをあがめ奉る始末（それすら否定する人も昔はいたが、最近はずがにいない）。そんなハッカーたちに、非ハッカーたちの気分がわかるだろうか。おたくに、非おたくのためのソフトがつくれるだろうか。自分たちが喜んで使っているような代物ではダメなのだ、ということがわかるだろうか。そこらのおじさんにE m a c sを押しつけるのは無理だ、そしてそれはおじさんたちがバカなのではなく、そもそも無理難題なのだ、ということを理解してくれるだろうか。

そしてもしハッカーたちにそれができないとしたら、いちばん優れたやり方というのはアップル社がやっているような、コアとなるカーネルの部分だけはオープンソース・フリーソフト化して、その上のGUIの部分については自分たちがおさえる、というやりかたなのかもしれない。おりしもいま、いろいろな動きが一斉に立ちあがってきている。今後十年くらいで、たぶん何らかの見通しは出るんじゃないだろうか。本書で指摘されているようなポイントも、その中でかなり大きくクローズアップされてくるはずだ。そこでの見通しを得るためにも、本書は有益だと思う。

本書の編集は、榎本統太氏が担当された。また、訳者の質問に答えてくれた著者と、ローラ・リンドグレン、ケン・スウィージーの諸氏にも感謝したい。ありがとう。

二〇〇〇年正月
山形浩生