

The Dark Side of Forcing

Vol. 15



株式会社ペあのしすても倒産特集特別号

継続企業前提に捧げる歌

淡中 圏

おまえたち自由業には信じられないようなものを私は見てきた
フォルダの奥深くで燃える決算書
スプレッドシートの彼方で暗闇に瞬くガントチャート
そんな思い出も時間と共にやがて消える
雨の中の涙のように
清算する時が来た

目次

継続企業前提に捧げる歌	i
第1章 ペあのしすてむ衰亡史	1
1.1 この文書について	1
1.2 ペあのしすてむとはどんな会社だったか	2
1.3 大雑把なことの顛末	3
1.4 第一歩でつまづく	4
1.5 終わり	11
1.6 教訓1：仮定に気をつけ、事実に基づくこと	13
1.7 教訓2：チームとコミュニティ	14
1.8 連想と展望1：反知性主義と反反知性主義の泥喧嘩	16
1.9 連想と展望2：暗黙知と形式知	22
1.10 結び	25
1.11 2022-08-14 追記	26
参考文献	27
ペあのしすてむ跡地訪問記	29

第 1 章

ぺあのしすてむ衰亡史

淡中 圏

1.1 この文書について

かつて「株式会社ぺあのしすてむ」と名乗った不思議な会社があった。数学を会社の大きな柱とし、小さな会社の中で研究活動と企業活動を両立させようとした意欲的な姿勢が、Twitter 内の数学クラスターなどで話題になったりもした。それが 2020 年 1 月にひっそりと幕を下ろしていた。残念がる声も Twitter などで散見した。

ぺあのしすてむの当事者であった私が、それらの意見を見て危惧したことがあった。

今回の倒産劇を遠くから見ていたら、

「やはり、数学で会社をやっていくなんてことは（小さな会社には）無理だったのだろうか」

というような結論に至ってしまうのではないか、という懸念だ。

私もぺあのしすてむに夢を託した人間だ。そして、その夢は今も大事に抱いている。

ぺあのしすてむは結果的にうまくいかなかった。しかしそれは、決して「数学で会社をやっていくのは無理」などのような単純な話ではなかった。それはあの会社の中心近くにいた私だからわかることだ。

そこでその場にいた人間の責任として、なぜぺあのしすてむがうまく行かなかったのか、得られた情報をまとめて分析したものを公表したいと思う。

ただしこれは「犯人探し」ではない。

うまく行かなかったときに、「誰に責任があるのか」を探そうとするのは、長い進化の歴史の中で人間の社会に染み渡った習慣であるが、複雑なシステムでの失敗を調べるときはこれはうまく行かない場合が多い。そもそも一人の責任に帰せるような単純な話ではないのだ。

航空業界での事故の原因調査は、決して個人に責任を被せず、仕組みの側に欠陥を探すという。それによって、航空業界では事故率を継続的に減らすことができたのだ [24]。

そこで今回もできるだけ航空業界のやり方で考えたいと思う（私は好きな番組にちなんで「メーデースタイル [30]」と呼んでいる）。

また、個人の責任に帰さないためにも、個人名などは伏せておく（内情を知ってる人にはどうしても分かってしまうだろうが）。それによって正確性が減ったとしても、重要な

情報はそこにはあまりないと考えているからだ。

私自身はできるだけ客観的に物事をみたいと思っているが、ことの性質上どうしても「私からの見え方」になる部分は仕方ないと考えている。それを乗り越えて、本当に客観的な調査報告を作るほどのリソースは残念ながらない。あくまで私からの見え方であることを言い添えた上で、その見え方の上に構築した分析や見解などには、人が読んで参考にできる部分があるのではないかと自負している。

1.2 ペあのしすてむとはどんな会社だったか

ペあのしすてむの社長（以下単に社長と呼ぶ）は名古屋大学の多元数理科学研究科（以下単に多元と呼ぶ）を卒業したあと、プログラマーとして様々な仕事に携わっていた。そしてペあのしすてむは、実は元々社長の個人事業であったのだ。

そんな個人事業にメンバーが増え始めたのは、多元の在学生や卒業生同士の縁であった。

多元には、数学を応用して仕事を働いている卒業生を呼んで、様々な数学の応用について講義をしてもらう連携大学院制度を利用した「応用数理」と言う授業がある（私も講義をしたことがある）。

社長は別の卒業生から、その講義の講師にならないかと誘われた。それが縁で彼を役員として招いたのだ（彼を役員 A と呼ぶ）。

役員として招くと言っても、実際に運営に関わってもらったと言うわけでもなく（この時点でもペあのしすてむは単なる社長の個人事業だ）、役員報酬もなしで、時々会って何か面白いことがないか話し合っているだけだったと聞く（それが役員会議な訳だ）。

そこにもう一人加わったのが役員 B である。

役員 B は多元の博士課程に所属して研究活動をしていたのだが、当時は多元を休学していた（休学しながら研究活動をしている、というよくある状態である）。そしてそんな中役員 B はとある研究集会に出ようとしたのだが、そのためにはどうやら所属が必要らしいとわかった。困った、所属が欲しい、となっても所属は道端に転がってはいらない。どうしようかと考えていたところ、「ペあのしすてむの役員になったらどうか」と言う話になったようだ。

結局、その所属ではその研究集会には出られなかったらしく、役員になったのはそういう意味では無意味だったのだが、役員 B はその後も役員として残り続け、役員報酬なしで役員会議で時々あって話し合うようになる。

これが会社が 2018 年に本格的に動き始めるまでの 10 年ほどの間のペあのしすてむの姿である。

なんとも変な会社だな、そもそもこれは会社なのか、と言う印象を持つ人もいるかと思う。

その印象は間違っていないと思う。

この時代のペあのしすてむを私も実はよく知らない。私に関わる前の時代だからだ。

だが私はこの話が結構好きで、このような変な会社のままだうにか存続できなかったのか、と今でも惜しく思うことがある。

この時代、どうして会社がこんな感じだったのか、という話も全体を分析する際に少しする。

それでは、こんな牧歌的な会社がどうして、（一部の）人々の話題になるまで急成長し、

そしてシャボン玉のように破裂して消えてしまったのか。その話に次は移ろう。

1.3 大雑把なことの顛末

ぺあのしすてむの形が変わり始めたのは、二つのきっかけがある。

一つは社長が健康を害し始めたこと。これによって社長は自分が技術者としてバリバリ働く、という意識をあまり持たなくなった。

そしてそんな社長に、大きな仕事が舞い込んだ。たとえ社長の健康が良好でも、一人で扱える仕事ではない。

そこで社長は考えた。それならば、従業員を雇おう。そして、どうせ従業員を雇うなら、働きやすい会社を作ろう、と。

社長は、発達障害を持っていて、今までの仕事でもさまざまな苦勞をしてきた。だからこそ自分が働きやすいような会社を作りたいと考えたのだ。

そこで、多元のネットワークで心当たりのある人たちに、声をかけた。

その中に私もいた。

私は、多元の博士課程に5年ほどいて、博士号を取らずに多元を出てしまい、その後3年多元の卒業生のコネで高校の数学教師をしたあと、プログラミングが楽しそうなので教師を辞めて会社を転々としていた。

その間、数学同人誌を出していたせいで一部で知られていたりしたのだ。そういうわけで社長も私に声をかけてきた。2018年の9月ごろだ。

「月給40万で働きませんか？」

というような話だったはずである。

私は今に至るまでそんな月給もらったことない（ここは笑うところ）ので、とても信じられなかったが、なんか面白そうだったので、会って話を聞いてみることにしたのだ。

社長は、私が入ったら参加するプロジェクトの資料を真面目に説明してくれた。それほど「数学」という感じの仕事ではなかったが、将来的にはもっと数学を生かした仕事をしたいと言っていた。

その時点では役員Aや役員Bが何を考えていたのか、聞いた覚えはあまりない。

確か二回目に会ったときであったと思うが、もう一人の人物に引き合わされた。

彼はこの時点でまだ名大の学部生である（多元ではなく情報科学化）。彼の同級生がぺあのしすてむでバイトをしていたが、その同級生はすでに大学院生であった。なんでもIT業界のコンサルタントとして働いていて、その仕事が忙しくて学部が卒業できないのだという（時々こういう人がいる）。

会って、すぐに才能溢れる人物という印象を受けた。頭の良さのオーラを振りまいているような人物だった。

彼は名古屋大学でプログラミングコンテストなどの主催をしていたようで、それで役員Bと知り合ったという。その役員Bが彼を社長に引き合わせた。

彼と喋っていて、彼がClojureなどLispプログラミングに造詣が深いことはわかった。ClojureとF#を会社の主要言語にするという構想も聞かされた。Coqなどの形式手法を会社の主要な武器にしたい、という計画もかなり初めの方に聞いたはずだ。ゆくゆくは独自の言語で開発する構想を聞いたのはいつだっただろうか。

それらがどれくらい現実味があるかを評価するのは難しい（実現していない構想の現実

味なんてどれも曖昧だ)。ただどれも面白そうで一考に値すると私は考えた。今も私はそれらは一考に値すると考えているし、一考どころか幾考もし続けている。

もちろん、それを実現するためには、地道な仕事をこなさなくてはいけない。彼もそれはわかっていて、それがペあのしすてむに舞い込んだ先程の大口の仕事なのだ。

これを以降単に「初仕事」と呼ぼう。

そして彼を「役員 C」と呼ぶ。実は彼はこの時点では役員ではないのだが、のちに役員になるのでそう呼ぶことにする。

役員 C は最初、この初仕事のプロジェクトリーダーとして雇われた。そして私が実際の作業、というわけだ。

いつ頃だったかは正確な記憶がないが、実際に仕事が始まる前に役員 B と話をする機会があったはずだ。そのときに聞いた話では、役員 C の野望を実現する場所としてペあのしすてむを使えないか、と役員 B は考えていたようだ。「彼のためのサンドボックスを作りたい」という言葉で表現していた。

そして、私が彼の手足になって働く、というわけだ。

というか私しかできる人がいなかった、とも言える。役員 A と私はペあのしすてむ内で会話をする機会は結局全くと言っていいほどなかった。役員 A は自分の仕事で忙しかったのだ。役員 B は博士号を取るために論文の追い込み時期だったし、そもそも作業員としてペあのしすてむに関わるつもりがあったわけではない。社長は健康上の理由で無理だった。役員 C の友人でペあのしすてむでバイトをしている人もいたが、正直主戦力として働ける人はいなかった。

だから私が働くしかなかった。

私には文句はなかった。それは、面白そうだと素直に思ったからだ。

実際役員 C の話は面白かった。形式手法を大々的に使ってビジネスで成功するならば、数学の IT での応用例としてこの上ないと考えた。それが実現できたら、どれほど良かったことか。

役員 B は、大学にいて、日本の大学の現状について、思うところがあるようだった。そこで「半分研究で半分ビジネス」のような形の会社ができないか、と考え、それをペあのしすてむで実現できないか考えていたようだ。

それが彼が役員 C をペあのしすてむに引き入れる理由だった。

社長のビジョン、役員 B のビジョン、役員 C のビジョン、それらが合わさってペあのしすてむは動き出そうとしていた（役員 A が何を考えていたかは私には正直わからない）。そして私はそれらのビジョンを私なりに面白がっていて、だからこそこの会社に賭けてみようと思ったのだ。

だが、その前途はいきなり荒れ模様だったのだ。

1.4 第一歩でつまづく

私がバイトを辞めて、初仕事の作業に入ろうとした瞬間、待ったがかかってしまった。

待ったをかけたのは社長だった。

発注書がまだ出ていなかったのだ。

仕事を仲介してきた人は、発注書がまだないが作業を初めてくれ、と言い始めたが、社長は、そんなことはできない、と突っぱねたのだ。この人物をこれ以降、仲介者、と呼ぶ。

役員 C は仲介者と話して、作業を進めて良い、と進言したが、社長は決して首を縦に降らなかった。

これに関しては、IT 業界を始めとして正式な書類がないのに口約束で作業が始まってしまう業界が多いことは言うておかなくてはいけない。そして別に口約束も法的な根拠になりうるので、「口約束では絶対にダメ」とも言い切れないのも確かだ。

だが、生来の四角四面な性格とリスク回避への意識が相まって、社長は「絶対に発注書なしで仕事をしたりしない」というポリシーを持っていた。そしてこれはとても社長らしくて私はとても好きだ（正しいかどうかは少し難しい話になる）。

ただこのおかげで、私は入社してするはずだった作業がいきなりなくなって困ってしまった。

とりあえず代わりに社長が以前やっていた仕事の改修を 1 ヶ月間することになった。

それは色々な検索結果を API で取得してきてマッシュアップすることができる単純な web アプリだったのだが、PHP コードがなかなかすごくて笑ってしまった。

どうなっていたかを記憶から再構成する、例えば `TwitterSearchFunct.php` というファイルがあって中身を見ると、`main` という関数をつだけ持つ、`TwitterSearchFunct` クラスが定義されていて、`main` の中ではそれがコマンドライン引数を取得して、その文字列を解釈して `WebApi` に渡しているのだ。

そしてファイルの末尾では、`TwitterSearchFunct` のインスタンスを作成して、`main` 関数を実行している。そんなファイルがいくつもあり、そして Web アプリケーション部分からそれらのファイルが `exec` 関数（外部コマンドを実行する関数）を使って呼ばれ、場合によっては関数の中でも `exec` を使って他のファイルが呼び出され、という形でさまざまな処理がされていた。

当たり前だが、呼び出し関係がどうなっているのか、IDE では一切分からず、関数とか設定とか、意味ではなく機能で分類されたディレクトリの迷路をかき分けて呼び出し関係を探さなくてはいけなくて、とても苦労させられた。

はっきり言ってもものすごいクソコードだ。書籍で勉強してただけでは、なかなか知ることのできない世界がそこに広がっていて、私は正直興奮した。

私は 1 ヶ月かけてそれらをまともな Laravel の web アプリケーションに書き換えた。MeCab と CaboCha を使った、共起語分析のコードはあまりに読みにくくて、一から実装し直した。

ずいぶん後になって社長に聞いたところ、予想に違わずそれは社長のコードで、なんでもプロトタイプを作ったら、そのコードを使ってあとは自分たちで実装する、という話になってしまった結果があのだという。

プロトタイプは捨てるもの、という意識を周知させていきたいものである。

さて、私が呑気に仕事をしている間に、ようやく発注書が来たらしく、初仕事がどうにかこうにか始まった。

私は完全リモートで役員 C の指導のもと、研修的なことをし始めた。実際のところ、金をもらってコードを書くのが半年ほど前からバイトとしてし始めたばかりだったので、助かった作ったのは、のちに初仕事に生かす予定のスマホアプリ作成とデスクトップアプリ。でしばらくして、実際に使う予定のスマホアプリを作成した。

で、ここから話がまた拗れ始める。

嫌な予感はしていた。私が一度仲介者と会ったとき、彼はべあのしすてむについて「名

ばかりで、ただ単に集まってるだけの会社なんでしょ」と発言した。悪意は感じなかったが、どうやら何か認識のズレがあることは気にかかった。そしてそのズレは時と共にどんどん大きくなる。

例えば、私と社長が元請けの会社に訪問して話を聞いたりしたときに、社長が名刺を出したら仲介者が怒ってしまった事件が発生した。どうも仲介者的には、ペあのしすてむという会社の名前は出さず、全部自分がやっていることにしたかったらしい。

そういうスタンスの仲介業者がいることを今は理解しているが、ここに関しては社長の行動は間違っていないと私は今でも考えている。窓口を一本化した方が混乱が少ないのは分かるが、それなら窓口の整理にそれなりの作業をしてもらわなくては困る。^{*1}

結果的に社長と仲介者の関係は悪くなっていき、仲介者は主に役員 C と連絡を取り合うことになる。役員 C と仲介者はなぜか良好の関係を保ち続けていたらしい。しかし役員 C も仲介者も忙しく、話が早く進むわけではない。そこで結局私と社長が元請けに行行ってデモをしたり話をしたりしなくてはいけない。そしてますます捻れが大きくなる。次に仲介者は私が元請けに行くことにも難色を示し始めた。デモをするときに、私が立ち上がって説明しようとしたことが気に食わないというのだ。立ち上がるとお客さんを上から見下ろす形になるから失礼だという。あと私は緊張を和らげるために時々深いため息をつくくせがあるが、それもお客さんに不快感を与えと言われた。だから、私をお客さんの前に出すのはやめろ、と主張してきた。

だんだん我々は何をどうすればいいのかわからなくなってきた。

後から聞いた話で分かったのは、社長はもうこのプロジェクトは何かおかしいと思い止めたがっていたが、役員 C は「もう納品直前だ」と主張して、プロジェクトを前に進ませたがっていたことだ。

この「納品直前」という言葉の意味は今に至ってもよくわからない。作業者として「納品直前の何か」ができてるとはとても思えなかったのだ。私が存在を確信できるのは、デモ用のシステムだけで、それを納品するためにはまだまだ大量の作業が必要なものに思えた。

社長は直接作業に関わっていなかったため、役員 C の言っていることを信じたようだ。なので、問題は仲介者のみにある、と考えた。実際社長と仲介者の間でのコミュニケーションがうまくいっていなかったことは前にも書いた。そこで社長は直接元請け企業に会いに行って、仲介者に関する苦情を述べた。元請け企業でも仲介者の行動に関するフラストレーションは溜まっていたようで、「仲介者をなんとかしないと納品作業に進めない」という同意に至ったのだ（社長はこの時点では「納品直前」という話は信じている）。結局このプロジェクトはペあのしすてむ VS 仲介者、という泥試合になっていった。

私は今では、もっと根源的な部分でプロジェクトはおかしかった、と考えている。

仲介者だったのか、役員 C だったのかはわからないが、社長が作業前に欲しがっていた「プロジェクトのための確かな礎」がなし崩しにされてしまっていたのだろう。

お客さんと実際に話した感触から言って、お客さんがデモ用の実装を納品物として受け取るつもりだったようには私には思えない。渡しても、まだお客さんが自分で試すことすら難しい代物だった。

^{*1} この辺りはプログラミングにおけるカプセル化、より一般的には抽象化の議論がわかる人には、同じ話だと理解してもらえらるだろう。変数を `private` 宣言するだけでカプセル化や抽象化が実現できるわけではない。元請けにとってアクセスすべき情報があるのにそれを遮るのは間違っているのだ。

詳細を書くことはできないが、実際これは技術的にも難しいプロジェクトであった。だから、そんなにすぐに使えるシステムを作ることは最初から無理だったのだ。ならば、その中からとりあえず使える小さな納品物を切り出す必要がある。そしてこれはプログラミングよりずっと難しい。何回も何回も話し合いをしなくてはならない。

これがアジャイル開発と呼ばれる開発手法の中心部分だ。アジャイル開発では、大きな計画を立てるよりも、小さなサイクルでお客さんに見せられるもの、お客さんが試せるものを作ることを優先する [1]。そしてお客さんからのフィードバックを得て、次の開発サイクルに反映させる。そのために、計画よりもお客さんとのコミュニケーションに力点をおく。

今回も作業をする前にもっともっと話し合いをしなくてはいけなかった案件だったと今では考えている。実際の開発に入る前に、プロトタイプを作ることとその予算に同意しなくてはいけなかったはずだ。一見予算が増えるように見えても、それで将来の無駄な作業を減らせたはずなのだ。

実際に何をやるべきかが定まる前に仲介者と役員 C が作業を急がせてしまったので、このプロジェクトはセオリー通りに空中分解してしまった。

それ以降もいくつかのプロジェクトに携わってきたが、難しいプロジェクトで「納品物」というものが曖昧にされる事態には何度も出会った。難しいプロジェクトではそもそも「作るもの」のイメージをはっきりさせるのが難しい。だから「これが納品物ですよ」と持っていても、お客さんにとって求めているものと全く違う可能性が高い。

今回もおそらく納品物という言葉で意味したいるものが、一人一人違うという事態が発生してしまっている。だから、そもそも話が進もうがなくなってしまうのだ。

で、私がその頃何をしていたかというと、Clojure の `test.check` ライブラリーを元にした、自作の自動テストライブラリーを呑気に構築していた。

これは Haskell の QuickCheck ライブラリーを元にしてているが、このライブラリーは圏論的な匂いが強くとても面白い [31]。

それにしても、なぜこんなことをしていたかというと、実はこれが役員 C の構想に関わっていた。役員 C はこの初仕事で単なるこのプロジェクトのためのプロダクトではなく、今後も再利用可能なかなり大きなフレームワークを構築する計画だったのだ。

元々役員 C には大きな計画があることを先ほど書いた。そしてそのために役員 C は大胆な会社の膨張計画を考えていたのだ。形式手法を主要な武器にした会社を成立させるために、そこそこの人数の数学ができる従業員を抱える必要がある。そのためには短時間で会社を大きくしなくてはいけない、というわけだ。

そのためのスタートダッシュとして役員 C は初仕事を見ていたのであろう。プロジェクトでお金をもらいながら、会社の礎を築くのは、実際悪くない考えのように思える。ただ、そのプロジェクト自身が、大きなリスクを抱えていなければだが。

後知恵で言えば、この計画は全くアジャイルではない（ちなみに、アジャイル開発は開発手法だが、それを会社の戦略レベルで応用する考え方もある。『リーンスタートアップ』などが有名な例だ [5]）。役員 C はスクラムなどのアジャイル手法について詳しいと自負していたようだが、この件に関してはプロジェクトマネージングについても会社のマネージングにおいても、全くアジャイルではない。

結局入るはずだったお金は入らなかった。役員 C は「納品すればお金が入る」と考えていたようだが、社長はプロジェクトが整理整頓されるまで納品作業に入ろうとはしない

(私は「これが納品物です」と言って渡したらもっと話がこじれていたと思うので、社長の判断はよかったと思っている)。

結局、さまざまな名前がすでにつけられていたフレームワークは宙に浮いてしまった。

私はその間に、スマホやデスクトップのアプリ開発をしながら、憧れのフリーランサーのコードを読ませてもらったりして本当に勉強になったののだが、いきなりスタートダッシュを挫かれて会社は大変な状態になっていたのだ。

これが大体 2018 年の年末から 2019 年の新年に渡っての出来事だ。

そこからしばらくは、小さなことはいろいろあったはずだが、大きく語るようなことはなかった。

企業の機械学習の研修の手伝いとして TA をやって Python を教えたりしていたことが唯一記憶に残っている。^{*2}

確か 5 月ごろに、現在お世話になっている株式会社 RAKUDO の仕事の手伝いが始まっている。これはとある企業が App Store で配布しているアプリの機能追加だったのだが、これまた地獄のようなコードでびっくりした。

使われている Objective-C は参照カウントによるガベージコレクションが導入される前の、自分でメモリーを開放しているコードで、使っているライブラリーは、CocoaPods などのライブラリー管理ツールを使ってインストールされたものではなく、GitHub にあったであろう何年も前のコードを直接コードに組み込んで、所々書き換えてある代物だった。そして、ソースコードは ftp で管理されていて、全くバージョン管理されていなかった。

C 言語の do while 構文をループとして使わず、break のためだけに使う奇妙なイディオムを私はここで初めて見た。関数に切り分ければいいだろうに。

iOS の ViewController には元々 parent というプロパティがあるが、テーブルの行の親がテーブルで、テーブルの親が ViewController で、ViewController の親がページ遷移の前のページの ViewController で、と手動で設定するコードが書いてあり、この部分をページ遷移時に自分で設定しないとアプリが動かなくなってしまった。イベント処理を親に伝搬させていく仕組みだったのだ。

そして全ての親の ViewController は「なんたらサービス」という名前の、多数の(実質)グローバル変数を持つ神オブジェクトのメソッドを呼び出すことで、イベントを最終的に処理していた。

http のリクエストするときには、header や bodyなどを文字列操作で自分で作る構築していく必要があった。multipart/form-data の区切り文字列が決まった文字列になっていた。これは脆弱性だと考えてランダムな文字列にしようとしたら、サーバー (PHP で書かれていることは url から分かるけど、中身は全くわからない) からエラーが帰ってきて諦めたりもした。

http のレスポンスのハンドリングなどの非同期処理は、全て iOS の Notification の仕組みで先ほど出てきた「なんたらサービス」に送られ、「なんたらサービス」からまた

^{*2} このときの講師の先生は非常に優秀で感心したが、これは最初に希少な例に出会ってしまっただけで、その後企業の研修でプログラミングを教えている人たちに、なぜかプログラマーとして仕事をするのを諦めたような人たちが多数含まれていることを発見してびっくりしたものだ。数学が苦手なのに「自分は算数は教えられる」と考える人がなぜか結構いるように、ソフトウェアを作ることは苦手なのに「自分はプログラミングは教えられる」と考える人たちも結構いるようだ。謎は深い。

Notification の仕組みでいろいろな場所に送られていた。いわば goto の再発明である。IDE の仕組みを封じられてスパゲティを解くのは非常に大変だった。どうして皆、IDE の仕組みを殺そうとするのか。

全体的な感想は「仕事でアプリを作る前に入門書でも読んで欲しい」という感じだ。

結局私は、全てガベージコレクションを使ったコードに書き換え、警告の出る古いライブラリーを新しいものに置き換え、魔改造された可哀想なライブラリーも、ほぼ完全に私が作ったものに置き換えてしまった。

機能追加部分は、ローカルに置いてあるデータと、SNS などの外部サービスにあるデータなどを同等に扱う必要があったので、抽象化層を作って、プロトコル（Objective-C での Java のインターフェース）を実装したクラスを定義すれば新しいデータソースが簡単に扱えるようにした。それなりのフレームワークを構築できたのは楽しかった。プログラミングの勉強がしたい人は一度ちょっとしたフレームワークを自分で作ってみると自信が着くと思う。

同時に「IT 業界の既存のコードってこんな低レベルなものばかりなの？」と心配になったりもした。最近は心配していない。確信しているから。

その他にも細々とした仕事はあったが、会社に大きなプロジェクトが舞い込むことは最後までなかった。

この時期に私はボーナスをもらっているが、半額は株である。全額か半額か選べと言われて、半額にしておいて本当によかった。残りの半額は会社が潰れて紙切れにもならなかった（単なるデータだったので）。

結局、べあのしすてむの長期プランは「初仕事」に完全に頼っていたことになる。それが崩れてしまったので、べあのしすてむは会社を存続させるための基礎を欠いたまま、縋り付く藁を探して迷走していたことになる。

この時期にかつての多元の仲間が二人、別の大学で数学の研究をしている一人が正社員として加入した。みな数学については優秀だったが、IT 業界でいきなり即戦力になるような人材ではない。

Twitter などでもべあのしすてむに興味を持ってくれた優秀なプログラマーをバイトとして参加させてはいたが、彼らに振る仕事がたくさんあった覚えはない。

今、他の会社で営業からサポートまでさまざまな仕事に関わっているが、思い出してみればべあのしすてむにおいて、仕事をとってくるためにできることはまだまだあったはずだと思える。その仕事ができるのはバイトではなく、フルタイムに関われる正社員が妥当だ。しかし、あそこ正社員としてとっていたのは、そのような営業活動ができる人材ではなかった。

ここでも役員 C の計画が現実とずれてしまっていたと思われる。

ところがこの時期に役員 C が正式に役員に就任している。一体どういうことであろうか。

この時期、資金繰りに困って存続の危機に陥った会社に救いの手を差し伸べた人がいたのだ。彼を仮に出資者と呼ぼう。その出資者を連れてきたのは役員 C であった。彼の友人だったのだ。

出資者は社長に会って、「社員の働きやすい会社を作ろう」という社長の考えに感銘を受け、この会社に出資することを決めたのだ。

この少なくない出資によってべあのしすてむは首の皮一枚でこの危機を乗り越えること

ができたのだ*3。

社長はこれを「出資者は役員 C を支援しているのだ」と理解した。そこで役員 C を正式に役員として、会社の方向づけを本格的に役員 C に任せたのだ。

ここで大きな勘違いが生じてしまっていたようだ*4。

その後、会社の実質的な舵取りは役員 C に一任されるようになってしまう。

社長は出資者との連絡も役員 C を通して行っていたようで、社長が出資者の思惑を確認したのは、会社がなくなってからである。

2019 年の後半、流石にこのままではいけないとなって、私も含めて正社員全員で仕事をとるための営業活動に回ることになった。

主に営業資料の内容を考えたり、デザインを考えたりしていた。

今考えると他に色々やりようがあった*5とは思うが、この時点では私にも経験がなさすぎて代案があったわけではない。

社長はこの頃には本格的に体調が悪くなりはじめ、ほとんど口を出すこともできなくなっていた。

ただ役員 C が高額な事務所を借りようとしたりと、不必要にお金を使おうとしたときのみ止めていたようである。

確かこの時期、役員 A が役員をやめている。そして入れ替わるように仲間になったのが役員 D である。

彼はかつて大きな会社を辞めて、自分でビジネスを初めて、それを畳んだ経験があった。ひょんなことから役員 C と知り合っていて、それが縁で仕事を手伝ってくれないかと声をかけられたのだ。彼に任されたのは主に営業の仕事だった。

そしてその頃、出資者も本格的に日本でビジネスを始めようと、名古屋の新幹線のレールの下にある貸しオフィスを借りていた。5 分ごとに頭上を新幹線が通り過ぎて会話が不可能になるすごいオフィスであった。

今考えると出資者はお金を持っているのに、そんな安い貸しオフィスを借りていることに感心する。

ペあのしすてむは元々は完全リモートを歌っていた会社であったのだが、この時期に名古屋近郊に住んでいる正社員はこの事務所（ペあのしすてむの事務所ではないのだが）に出社することになった。

すでに仕事のわかっている人間ばかりならリモートも成立するのだが、正社員の多くは数学はできるがプログラミングも営業もできないので、そのメンバーにリモートで仕事をさせてもあまりに効率が悪かったからだ。

また役員 C がスクラム*6を使ってメンバーを動かすためにも、社員をできるだけ一堂に

*3 ちなみに出資者は最初数百万のお金を「自由に使って良い」と社長にポンと渡したらしい。律儀な社長は「ちゃんとした手続きをとって融資してもらわないと困る」とそのお金に一切手を付けずにしばらくとっておいたらしい。社長らしくていい話だ。

*4 ちなみにこの時期に私を役員に入れようかどうかという話も出たらしい。

それに反対したのも役員 C だという。なんでも「田中さんには楽しいことだけ考えていればいいと思います」と発言したらしい。この話は当時間かされてずっと喉の奥の魚の小骨のように引っかかっていたのだが、後から、なんとも失礼な話だと思うようになった。確かに私はいつも楽しいことだけ考えたいとは考えているが、それを視界から楽しくないことを除去することによって実現しようとしているわけではないのである。

*5 もっと足を使うべきだった。

*6 先ほど少し説明したアジャイル手法の一種

集めることが都合良かったのだ。

私がべあのしすてむで最後にしていた仕事は、Jekyll という ruby の静的サイトジェネレーターを使って、会社の web ページを作り替えることであった。つまり直前まで会社の長期的な戦略のために動いていたのだ。直近に必要なお金を得るための仕事が回ってくることはなかった。その点からいっても、私の直接の上司である役員 C に資金繰りの心配はほとんどなかったはずだ（その理由は後で明らかにされる）。

その時期に実は会社の奥では別の思惑が動いていた。

1.5 終わり

会社に入ってからすぐに役員 D はこの会社がどうもおかしいと気づいたのだ。お金の流れが不透明すぎる、と考えたのだ。役員 D は同じオフィスにいる出資者に質問した。そして出資者もお金の流れについて把握できないことを知ったのだ。

出資者は当然の権利として収支を確認させてくれるように役員 C に頼んでいた。しかしなかなか見せてもらえないまま時間が経ってしまったのだ。

そこで役員 D は営業の仕事と並行して、調査を始める。

週に一回社長の家まで赴いて、聞き取りなどを始めたのだ。

実際社長の健康悪化により、さまざまな手続きが滞っていたので、彼が社長の家に通うことには誰もなんの疑問も感じなかった。

その結果として、役員 C の金遣いがかなりあらく、それに対してなんの収入も得られていないし、得られる見込みもついていないことが明らかになったのだ。

役員 C はこのまま突き進む計画だった。そのために、現状で即戦力になっていないメンバーを解雇して、支出を減らそうと考えていた。役員も自分以外はほとんど残らない計画だったと言う話も聞いている。ギャンブルに勝つための方法は勝つまでやり続けることだ。もし資金が無限にあるならば、であるが。そして役員 C は出資者がお金を出すことを確信していた（彼がそう発言するのを私も聞いた）ので、彼にとってこの前提条件は成り立っていたのだ。

それと別の場所で役員 D と出資者も、どうやって会社を存続させられるか考えていた。彼らにとっても答えはほとんど同じだったのだ。即戦力になっているメンバーを除いて解雇する。役員も社長を含めてほとんどやめてもらう。そうすることによって会社を立て直すことは、原理的には不可能ではなかった。

前提が同じなので議論の筋道は結局同じになっていってしまうのだ。だが、そこからが違った。

役員 D と出資者は、そうして社長も含めてほとんどの人をやめさせてべあのしすてむを存続させたとして、それがべあのしすてむと言えるかどうかを考えたのだ。

言えない、と彼らは考えた。

役員 D は社長の家に赴いて、そもそもべあのしすてむという会社がどのようにして産まれたのかを聞いていた。社長と役員 A と役員 B が「何か面白いことできないか」と特に目的もなく時々集まって話している場としてのべあのしすてむの話のだ。彼はそれを素直にいい話だと思ったのだ。ビジネスと言うにはあまりに牧歌的だが、それはそれで一つの間として確かに機能していたのだ。

べあのしすてむを存続させたとして、そのような素敵な場としての歴史を完全に捨てて

しまつて良いわけではない、と役員 D と出資者は考えた。

彼らは何度も何度も顔を突き合わせて、会社を存続させるもっと良い方法がないか考えた。ペあのしすてむの良いところをなんとか保ったまま生き残らせるすべはないものか、と。

なかった。

だから答えは一つだった。

こうしてペあのしすてむを潰す決定が下された。

私がこの顛末で何より好きなのは、役員 D と出資者がペあのしすてむを潰す決断をしたのは、彼らがペあのしすてむの在り方についてしっかり考えて、尊敬の念を絶やさなかったからであることだ。

2019 年の 12 月 30 日に緊急の役員会議が行われ、ここで会社が存続しないことが決定された。会社の存続は出資者の出資にかかっていたので、それが為されないことが決定されたら、どうしようもなく会社は潰れてしまうのだ。

役員 C だけでなく、社長にとっても役員 B にとってもこれはショックな報せであった。

役員 C が「この事業には将来性があるから、資金を出したほうがいい」と説得しようとする、出資者は「それは素晴らしい。だったら他に投資家を探せばいい」と答えたという。

その知らせはその日のうちに私のところにも届いた。私は次の日にコミケで売るための同人誌をキンコーズで印刷した帰りであった。私は非常にぐらぐらした気持ちで次の日、同人誌を売っていた。

実に奇妙な年末であった。

ただ、私の年末年始はペあのしすてむのメンバーではかなりましな部類であったと思う。

私はペあのしすてむ在籍中に仕事を手伝った株式会社 RAKUDO に移籍し、今も在籍している*7 2022-08-14 追記：これについては最後に追加のオチがある。。まさかの無職期間 0 日である。

そして何より私はペあのしすてむで良い経験をさせてもらった。ほぼ未経験の状態、初めて本格的なコンピュータシステム構築やアプリ開発に関わらせてもらった。また優秀なフリーランサーの良いコードを読ませてもらった（同時にひどいコードをいくつも読ませてもらい、IT 業界でプログラマーとして働く自信をつけさせてもらった）。

私がアジャイル開発などの開発手法を勉強し始めたのはペあのしすてむ時代の後半であり、その時の勉強は今でも役に立っている（それまでプログラミングには興味があったが、チームマネジメントなんて全く興味を持っていなかったののに、会社が潰れるちょっと前に急に勉強を始めたのは、もしかしたら本能的に危機を感じていたのかもしれない）。

それ以外にも色々考える機会になった。

これは独学とはいえ長いことプログラミングを勉強していたからだ。博士課程で研究者を志した経験も生きたし、三年間の高校での非常勤講師の経験も生きたし、ペあのしすてむに入る前に経験したいくつかのベンチャー企業での経験も生きた（とんでもない企業や経営者をいくつも見てきた）。

経験を良い経験にするためには、ある程度の経験が必要なのだ。だから、教育が必要な

*7

のだ。ぺあのしすてむの他のメンバーには、そもそもこの会社で経験を積むために必要な経験を積んでいなかった人も多かったように見えた。そしてその人たちは、与えるべき教育も与えられないままだ振り回されただけだったように見えた。その人たちにとっては、本当にとんだ災難だったと思う。

この文章を書かなくてはいけないと思った理由の一つは、それがぺあのしすてむで良い経験をした数少ない社員の勤めに思えたからだ。

会社がなくなって以来会っていない人もいる。基本リモートの会社だったので、あの時の仲間でもっとも一度も会ったことのない人が実はまだいる。

できればあって話がしたい、と今でも思っている。

1.6 教訓1：仮定に気をつけ、事実に基づくこと

私が大きく感じたのは、会社や会社のプロジェクトというものがどれだけのフィクションの上に成り立っているかということだ。

1000円札というものを我々がなんの心配もなく欲しがるのは、他の人が1000円札を欲しがると思っているからである。

同様に、我々が会社で仕事をし、会社に仕事を頼むのは、会社が明日もあると信じているからであり、会社が収益を出し続けると信じているからである。

ところがそれには多くの仮定が含まれていて、仮定の割合があまりに大きくなりすぎると、会社が潰れたり、バブルが弾けて市場の大暴落に繋がったりする。

理屈としては分かっているけど、実際に我々が寄って立つ社会というものが本当にあやふやな仮定で成り立っていることを実感するのはなかなか難しい*8。

プロジェクトもそうで、多分こうだろう、という仮定の話が大量に含まれてしまう（多分、こうやればできる。多分、この日くらいにできる）。特にプログラミングのように、形のないものを扱っていると、すぐにプロジェクトの大部分が仮定の話で、はっきり断言できる部分はほとんどないようになってしまう。

すると、会社やプロジェクトのマネジメントをする人自身が、自分たちの言っていることがどれくらいフィクションでどれくらい確かな事実なのか分からなくなってしまうのだ。

プロジェクトマネージャーが「もうすぐできます」というとき、彼は確かにそう思っているのかもしれない。しかし実際には全然そうではないこともあるのだ。

社長が「うちの会社は大丈夫」というとき、彼は確かにそう思っているのかもしれない。しかし実際には全然そうではないこともあるのだ。

これは仮定がいけないという意味ではない。会社もプロジェクトも仮定が大量に含まれているからこそ、意味がある。会社やプロジェクトによってそれら仮定やフィクションに過ぎなかったものが、現実になるのだから。

だからこそ、意識的に「事実はなんなのか。どこまでが仮定の話なのか」を理解しようとしなくてはならない。アジャイル開発はそのための手法なのだ。

結局ぺあのしすてむ自体も、ぺあのしすてむの初仕事も、全て仮定の話に費やされ、仮定の話のまま終わってしまった。

*8 実感がなくても理屈でわかることは素晴らしいことなので、皆ぜひ経済学や会計学を学ぶべきだと思う

何か事実に基づくものをもっと早く提示すべきだった。

それが出資者への当然の義務だ。

出資されたお金は自分たちのお金ではなく、出資者のお金だ。だから、出資者への説明責任が生じ、それは事実に基づく必要がある。

この当然のことが見過ごされていた。

これがとりあえずのペあのしすてむの教訓だった。

これについては役員 C だけの責任だけでなく、そもそも会社の方針を決める役員全体の責任だったと思う。

1.7 教訓2：チームとコミュニティ

とはいえそれでも上記の話を読んで、「概ね役員 C が悪いのでは？」と思う方が多いかもしれない。

はっきり言うておくが、私はそうは思っていない。もしそうなら、そもそも私が最初に書いた「メーデースタイル」と矛盾してしまう。

そこでもう少し深掘りして、どんな会社の形を作るべきだったのか、もう少し無籍したいと思う。

確かに役員 C は暴走していたと思う。そしてそれを抑えるべき社長は体調不良で、ほとんど口を出せず、本当におかしい出費以外却下することができなかった。

役員 D と出資者が何かがおかしいことに気づいたときにはあまりに遅かった。

何かができるとしたら、「会社を役員 C のためのサンドボックスにする」と発言した役員 B だったはずだ。確かに会社は役員 C のためのサンドボックスと化していた。そのおかげで会社は潰れたけど役員 C は大きなダメージを受けずに済んだ（未来ある若者にとって非常に幸いなことに）。ただそのサンドボックスは社長の金でできていたわけだが。役員 B が言っていることはそう言う意味ではなかったはずで、もし会社の中にもう一枠サンドボックスを作るつもりだったのなら、真面目に作らなくてはいけなかったはずだ。

そもそも、私はペあのしすてむは会社としての体を成し損ねていたと思っている。

それを説明するために、「チーム」という概念と「コミュニティ」という概念を対照しよう。

「チーム」とはなんらかのプロジェクトを実行するために集められた人員だ。プロジェクトとは、(PMBOK の定義によるならば)「何かを実現するための時間的に区切られた仕事」だ。

ビジネスにおいてその時間的な区切りはどんどん短くなっている。すると、チームにおいては迅速な意思決定が求められる。チーム全体が合意を形成するのをゆっくり待ってられない場合は、何らかのプロセスにしたがってチームとしての意思決定を行わなくてはいけない。そのためには、リーダーシップも必要とされる場合も多い（単なる独裁ではなく、何らかのプロセスも必要であるが）。

かつてチームは同じ職種の人員で構成されていたが、プロジェクトを実行するために必要なさまざまな職種（IT ならば、営業とプログラマーとデザイナー、など）を集めた少人数のチームを運営するほうが効率がいいと言われている。これをクロスファンクショナルなチームと呼ぶ人もいる。

なぜクロスファンクショナルな方が良いかというと、そちらの方がさまざまな視点から

の考え方を集約させるのに効率がいいからだ。例えばクライスラーでは元々設計・エンジニアリング・製造・販売などが別々のチームに分かれていた。これによって競業他社が3年で新商品を市場に送り込んでいるところを5年もかかってしまっていた。これをクロスファンクショナルなチームにすることで2年半まで短くすることができた。

アジャイル開発でも必ず（開発者だけでなく顧客までチームに含めた）クロスファンクショナルなチームを構築することが推奨される。アジャイルな開発に必要な迅速なコミュニケーションにはそれが必要だからだ。

それに対して「コミュニティ」は、同じ興味関心を持つものの集まりである。コミュニティが向き合う課題も、プロジェクトのような時間的に区切られたものというよりは、もっと長期的で、そもそも締め切りがなかったり、問題自体が曖昧で定義のしにくいものだったりする。

コミュニティにおける意思決定は、チームの場合のような迅速性は求められない。

その代わり、コミュニティではチームよりも長期的な視点で動くことができる。

『コミュニティ・オブ・プラクティス』によれば、アジャイル開発の主導した「クロスファンクショナルチーム」によって、新商品の開発スピードは確かに上がった。しかしその代わり、複数のプロジェクトで似たような問題が起きていても、それに関する情報が共有されないことによる、別種の非効率が生じるようになった。

その結果、いくつかの会社では職種や興味関心を同じくするものが、情報共有するために自然発生的に定期的な会合を開くようになった。中には、それを半公式の組織として取り入れる会社も出てきた。

それが「実践コミュニティ」である。

実際、チームや会社というものは、人間よりも寿命が短いものがほとんどである。それに対して、コミュニティは、うまく運営すれば、チームや会社よりもずっと長く生き延びるものである。そしてチームは、迅速な意思決定が求められるが故に形式知（文章などで伝えられる知識）を伝えていくための媒体としては、あまり適していない。長期的視点を持つコミュニティこそが、形式知の生まれ育っていく場なのだ（それに対してチームは暗黙知が生まれる場所である。これに関しては後でまとめる）。

アカデミズムというものの、その出自としても現状としても、コミュニティとしての特徴が色濃い組織である^{*9}。

そう考えると、初期のペあのしすてむの様態についても納得がいけるものになりはしないだろうか。ペあのしすてむはそもそも、チームとしてではなく、アカデミズムから派生したコミュニティとして誕生したのだ。

私が目撃したのは、コミュニティとして誕生したペあのしすてむがビジネスのためのチームに変身しようとして、失敗した光景だったのだ。

コミュニティの運営には意思決定の迅速さは求められない。むしろ、できる限りメンバーに納得感が生じることが良しとされる。

三隅二不二が提唱したPM理論は、リーダーシップには目標を設定する「P（Performance）型」と組織を円滑に運営する「M（Maintenance）型」の2種類があると主張するが、コミュニティのリーダーには主にM型が求められ、P型はそれほど必要とされな

^{*9} ビジネスにおけるチームとして運営方法を無理矢理当てはめたことが、大学改革の失敗の一因、という見方もしかしただけでできるかもしれない。

いことになる（目標が各自で勝手に設定すれば良い）。

それに対して、さまざまに異なる出自と視点を持ったメンバーをかき集めたクロスファンクショナルなチームで、スピード感のあるプロセスで意思決定するためにはそこにはP型リーダーシップが求められる。

思い出してみれば、ペあのしすてむでの話し合いは、何かを迅速に決定することよりも、それぞれの意見や立場のの違いを整理することに重きを置いていたように思える。これはコミュニティにおける議論としては正しい姿である。相手の言っていることを整理して、持ち帰り、自分の考えに反映させることが議論の目的だからだ。しかし、それは何かに結論を出すことが永遠に起こり得ないタイプの議論の仕方である。コミュニティにおいてはそれで良い。アカデミズムの目的は、少しでも知識を深めることであり、今日中に結論を出すことではない。

しかし、チームではそれではうまくいかない。特にビジネスの現場では。

ペあのしすてむがチームの素早い意思決定を実現しようとしてしたことは、意思決定を一人の人間に任せてしまうことだった。しかしこれはチームのあるべき姿ではない。アジャイルなどが、階層型のトップダウンな意思決定ではなく、フラットなボトムアップの意思決定を重視するのは、意思決定時にさまざまな異なる視点からの情報を効率よく集めるためである。どんなに優秀だろうと、一人の人間からの視点では集められる情報は限られているし、偏る。いわゆる「心理的安全性」の実現が最近ビジネスの現場において求められているのも、これが理由だ。

この結果ペあのしすてむでは、全ての決定を一人の人物に押し付け、予想される通り暴走して散ってしまった。それは意思決定者の責任であるというよりは、彼に全てを押し付けたその他の役員の責任放棄であったとも考えられる。

またメンバー的にも数学科出身者が多く、似たような考えを持つ人が集まって、少しもクロスファンクショナルになっていなかった。これはリクルーティングの問題でもあるが。出資者や役員Dのような違う視点を持つ人物をもっともっと早く会社の中枢部に入れていれば、何か違ったのではないかと悔やまれる。

これがペあのしすてむの失敗の、一つ大きな視点からの分析だ。

もし読者がアカデミズムから企業する時には、気をつけて欲しいと思う。

1.8 連想と展望1：反知性主義と反反知性主義の泥喧嘩

ここからは、さらに連想の翼を広げて、少しペあのしすてむと関係ない話もしたい。

これまでの話は、私にとってはもっと大きな話と関係があるような気がしている。ただこれはまだ気がしているという段階で、それほど整理された話ではないことも注意しておく。

それは、例えばペあのしすてむがアカデミズムとビジネスを繋げようと画策したことに関連している。

産学協同なども実際行われているので、わざわざアカデミズムとビジネスを繋げることを大袈裟に語る必要はない気も私自身しているのだが、それでもアカデミズムとそれ以外、ビジネスや国や公教育などには、どうも大きなズレがあるような気がしている。これは私の杞憂に過ぎないのかもしれないが、一応語っておきたい。

それは「反知性主義」の問題だ。

これは「Anti-Intellectualism」の訳語で、知性に反対しているというよりは、「知的権威」とか「知的エリート」、つまりいわゆる「知識人」に反対する主義主張である。つまり知識に関する反権威主義だ。

「反知性主義」と言う言葉は否定的な意味で使われることが多いが、本来の Anti-Intellectualism は必ずしも否定的とばかりは言えない意味を持つ。そもそも最初にヨーロッパからアメリカに渡ってきた人の多くは、宗教的権威から逃れてきた人々と一攫千金を夢見た山師であり、彼らの考え方は反知性主義の傾向を強く持っていた。反知性主義はアメリカの魂のようなものだ。

そして、アメリカが 20 世紀の勝利者になったのは、まさに彼らの反知性主義のおかげの側面も多い。この辺りのことは『ファンタジーランド』という本に詳しく書いてある。

その本によると、起業家（アントレプレナーと格好つけて呼んだりもする）精神と呼ばれるもの、失敗してもとにかく試しにやってみるという精神は、反知性主義から生まれてきたものであ流。

だからこそ「引き寄せの法則」「ポジティブシンキングをすれば全てがうまくいく」などのような「ニューソート」と呼ばれる呪術的思考と非常に相性が良いのだ。^{*10}

上記『ファンタジーランド』[18] や『反逆の神話 反権威は金になる』[20] にあるように、現代世界はあらゆる場所に反権威主義が蔓延し、新しい権威となってしまった妙な時代になってしまった。それは一つにはアメリカの起業家精神が世界を席卷したからだ。そしてそれは反知性主義が実際に大きな効力を発したからだろうとも思う。

ただ私見を言わせて貰えば反知性主義的な考え方が成功したのは、実は反知性主義が無条件に正しかったからなわけではない。反知性主義がアメリカにもたらしたのは、健全な反権威主義であり、健全な現場主義である。

権威を盲信せず、自分の頭で考え、自分で確かめてみる。これは多くの場合、実際に有効な考え方である。

ただ、それは権威が必ずしも正しくない、という当然の話からであり、権威が必ず間違っているという話ではない。そして多くの場合権威は正しく、私たちに必要なのは健全な権威主義と健全な反権威主義のミックスという、非常にトリビアルな代物ということになる。ただしそのバランスを撮るのは簡単ではない。バランスを欠いた反知性主義は、既存の知識を否定するあまり、反合理主義やオカルティズムなどに傾倒しやすい。その例はすでにあげたし、日本の企業文化におけるそのような例は『カルト資本主義』を読めば出てくる。トランプ支持者などのオルタナ右翼と呼ばれる人たちの中に、とても正気とは思えない陰謀論を信じる人たちが出てきたことは記憶に新しいのではないかと思う。

^{*10} 日本でも起業家向けの本にはこのような授受的思考・ニューエイジ的思考が何気なく出てきてびっくりさせられる例えば最近だと『心理的安全性の作り方』[29] という本に、「真理の社会構成主義」の考え方が肯定的に紹介されていた。「真理の社会構成主義」は「真理は社会的に構成されている」という考え方で、「我々が真理としているものは全て社会的に構成されている」という意味だとすれば自明な主張になる。自明でない主張だとすれば、それは哲学的にかなり微妙な主張になるので、ここで専門的な議論を紹介することはやめておきたい。ただここで書いておきたいことは、この「真理の社会構成主義」が反知性主義に組み合わせられると、科学的知識や価値の相対主義の根拠として持ち出されやすいという話だ。ちなみに『心理的安全性の作り方』においては「真理の社会構成主義」は「実際にどうであるかよりも、何を信じているかが重要」という形の主張の根拠となっている。ここにも反知性主義の影響が見られる。私も「何を信じているか」の重要性を否定するつもりはないが、それは「実際にどうであるか」の重要性を否定しないし、そして、このような相対主義をもたらしかねない形でこの議論を援用するのはあまり勧められたものではないと思っている。

最近多くの場所で「反知性主義」が否定的なニュアンスで語られるようになってきたのは、バランスを欠いた反権威主義の弊害が色々なところで見られるようになってきたからだ。この辺りの事情はジョセフ・キースの『反逆の神話 反権威は金になる』[20]を参照してほしい。権威をチェックするのではなく、機械的に反権威を標榜することのお手軽さが、真面目にやれば解決できるはずの政治的問題を等閑にしているという指摘だ。

日本の大学改革において、企業的なやり方を導入することが金科玉条のように扱われたりするのも、もしかしたら反知性主義の表れの一つかもしれない。また、小学校教育における掛け算の順序論争において、教師側の過剰とも言える反応も、もしかしたら反知性主義なのかもしれない。そして、それに対する数学関係者の強い反感は、反知性主義に対する反感、いわば「反反知性主義」なのかもしれない。知識人に関して意味のある擁護をせず機械的に「反知性主義」を攻撃する「反反知性主義」にもあまり意味はないと考えられる。

私は最近そんなことを考えていた。

これについては、そもそも反知性主義自体の問題というよりは、すでに書いたように、反知性主義に含まれる反権威主義についての問題だと考えている。

権威を機械的に否定しようとする反権威主義には議論をそもそも不可能にする成分がある。だから反権威主義に対して、反論してもなんの解決にもならない。権威が攻撃してきたから自分は正しいとさらに信じ込ませるだけである。

私は、反知性主義と反反知性主義の間に分断を解決する方法について、何か意味のある提言ができると思っているわけではない。

そもそも議論で解決できる状況ではなくなってしまっている。

私がしたいのは、反知性主義が成功した原因である、その健全な現場主義をそこから抽出し、それを現場から離れた場所で仕事をする知識人の役割との関係を、うまく説明したいだけだ。

そうすれば、そもそも反知性主義と反反知性主義の喧嘩を（個人的には）無視してしまえるからだ。

ここで私が心配するのは、さまざまな考え方の中に、反知性主義が知らない間に混入しているのではないか、という問題である。

反知性主義の中の健全な現場主義が有益であるが故に、実は必要のない反知性主義まで取り入れてしまっていることがあるのではないか、ということだ。

もしそういうものがあったら、私はそこから現場主義だけを残して、反知性主義を取り除きたいと思う。

例えば、ハッカー文化をはじめとしたコンピュータ文化は、その初期にヒッピー文化の影響を強く受けているが故に、強い反権威主義的傾向を持っていた。ハッカー文化の初期にヒッピーのドラッグ文化と交流があったことは『パソコン創世「第3の神話」—カウンターカルチャーが育んだ夢』[19]に詳しい^{*11}。

ハッカー文化のもう一つの源流はアカデミズムであるが、そこにおいてもその反権威主義の傾向は持っていた。反権威主義がアカデミズムに反抗するのは、それが制度化されているからであり、元々のアカデミズムの発生を見れば、それはアマチュアの集団であっ

^{*11} マウスやドラッグ&ドロップなどの GUI の生みの親であるダグラス・エンゲルバートは、人間の思考能力の拡張を目的に GUI の研究などをしていて、同じ好奇心から LSD にも手を出していたことが書かれている。この話は深掘りすると本が何冊も書けてしまうので、語りたくて仕方ないが我慢しておく。

た。そういう意味では、いつだってアカデミズムは反知性主義の攻撃対象であり、根城でもあったのだ。

そしてアマチュアリズムこそ反知性主義の魂だ^{*12}。もちろんハッカー文化もアマチュアリズムを中心にしている^{*13}。

そして、アジャイル開発や DDD（ドメイン駆動設計）などの現代的なソフトウェア開発の手法も反知性主義の影響を受けているかもしれない、と私が本気で考えていると言ったら、妄想に聞こえるだろうか。

例えば DDD では、顧客の仕事の領域（ドメイン）を分析してドメインモデルを抽出して、顧客（ドメインエキスパート）と開発者が同じ言語（ユビキタス言語と DDD では呼ばれている）を話し、ソースコード上でもできる限りドメインエキスパートが理解できる単語で書くことが推奨される [3]。

なぜこんなことをわざわざ主張するのかというと、技術者というものが技術をよく知っているがゆえに、どうしても顧客の話を技術の側のみから見ようとしがちであるからである。その結果、顧客の本当の要求を見逃しがちなのだ。

また本来、そのドメインに対する知識をもっとも持っているのはドメインエキスパートである顧客であるはずなのに、彼らが技術に関する知識を持たないがゆえに、人間の持つ自然な本能ゆえか、技術者は顧客のことを見下しがちなのだ。これでは関係がうまくいくはずがない。

そこで、ドメインエキスパートの言うことを真摯に聞くこと、そしてドメインエキスパートの言葉遣いを学んで同じ言語を話すべきだと主張する DDD は技術者にとってとても大事なアドバイスであり、ドメインエキスパートの言葉の中の概念をドメインモデルとして抽出するテクニック集としてもとても役に立つものでもある。

これはアジャイル開発にも含まれる、は健全な現場主義である。

しかし、これらの現場主義はそれを支える「抽象化された知識」の基礎がなくては有効には働かない。それこそが『コミュニティ・オブ・プラクティス』[6] などが補完しようとした部分である。

これに関してはアジャイル開発も同じだ。大きな計画を立てるのが現実離れが大き過ぎてリスクがあるのは真実だが、それは計画を全く立てないわけではない。リスクを避けるためには、その時点で分析できることはできる限り分析する必要がある。これはバートランド・メイヤーが『アジャイルイントロダクション：Agile 開発の光と影』[2] でアカデミズムの側からアジャイル手法を建設的に批判した内容とほぼ同じである。

ここで私自身に象徴的に思えた例を挙げさせてもらおう。『エリック・エヴァンスのドメイン駆動設計』[3] には、DDD を実践するためのいくつかのデザイン・パターンが挙げられている。その第 10 章の 6 番目の節で「Closure of Operations」と言うパターンが説明されている。これを数学を知っている人間が素直に訳せば「操作の閉包」になると思う

^{*12} 反権威主義の成分を多く含んだ反知性主義なら制度化されたアカデミズムを非難するであろうが、必要なアマチュアリズムを存続させるために制度化したのがアカデミズムなのだと考えれば、別にアカデミズムを非難する必要は全くないことがわかる。ここでも、反知性主義はアマチュアリズムという大切なものと、必要のない反権威主義を混ぜることで問題を複雑にしている。

^{*13} このようなハッカー文化・アマチュア文化はコンピュータハッカーが初めてだったわけではなく、自転車や無線などで産業時代の初期から何度も発生している。それらは巨大産業がなし得ないスピード感のある技術革新を作り出したあと、産業に組み込まれて消えている。このような歴史については永瀬唯の『疾走のメトロポリス—速度の都市、メディアの都市』[28] を読んで欲しい。

が、日本語訳ではなぜか「閉じた操作」となっている。これでは意味が変わってしまう。「操作の閉包」ならば、ドメインの中で使われている操作を抽出し、その操作で閉じた集合なりクラスなりを構成しようと数学を知っている人間なら思うところだが、「閉じた操作」では、今考えているドメインの中で閉じている演算を探すことになってしまう。前者と比べて後者は考えている領域を広げようとする方向性に欠けている。

しかし実は本の中でエリック・エヴァンスが説明しているのは後者なのである。「操作の閉包」ではなく「閉じた操作」について説明しているのだ。

私はこれがDDDの限界を表しているんじゃないかと思っている。なぜなら「操作の閉包」をとるということは、問題のドメイン自体を加工してしまうことになるので、本来のドメイン、現場のドメインとズレていってしまう。これはDDDの中にはうまく収まらない行為だ。

もちろんDDDは「DDDは銀の弾丸ではない」と正しく主張している^{*14}。DDDだけで全ての問題に片付くとは提唱者自身が思っていない。ただ、DDDの本を読んでも、DDDを補完する手法が書いてあるわけではない。

そこでアジャイル開発やDDDが、従来の手法の補完ではなく、従来の手法の置換として振る舞おうとしてしまうこと、従来の手法を否定しようとしてしまったこと。これらはその出自にある反知性主義の影響だったのではないかと私は妄想している。

そして、一部のアジャイル開発やDDDへの反感を持つと、ここでも「反反知性主義」と呼べるような過剰反応があるような気がする^{*15}。

DDDやアジャイル開発などの持つ反知性主義の出自を辿ると、ハッカー文化の他に、デザイン・パターンという考え方の源流であるクリストファー・アレグザンダーの思想に至る。

クリストファー・アレグザンダーは数学を学んだあとに建築家に転向し、都市計画などの研究をしていた。初期は、数学的な最適化の理論で都市計画が行えると考えていたようだが、『都市はツリーではない』[15]でそれが無理であると主張を変えた（「都市はツリーではない」とは、多くの都市計画が、問題を分岐が合流しないツリー構造で分類できることを仮定しているが、実際にはそんな単純ではなく、分岐が合流することがあることから言う。クリストファー・アレグザンダーはこのような構造をセミラティスと呼んでいた）。

そして、都市の複雑な問題をあらかじめツリー状に分類し切って計画を立てることが困難である以上、現場で住人とコミュニケーションを取りながら、生活と都市計画を同時に行っていくしかないと考えた。そのためには、住人と建築家が同じ言葉を喋る必要がある。そこで構想されたのがパターン・ランゲージと呼ばれるもので、それをソフトウェアの設計に応用したのがデザイン・パターンだった[17]。

ここで、クリストファー・アレグザンダーの健全な現場主義があらわれている。

しかし、実際にクリストファー・アレグザンダーの主要著書である『時を超えた建築の

^{*14} 「銀の弾丸ではない」とは「なんでも解決できる手法ではない」という意味で、元々は「狼男を殺す銀の弾丸は存在しない」つまり「そんな夢みたいなのは存在しない」みたいな意味だった。しかし「銀の弾丸」が「なんでも解決できる手法」みたいな意味に使われると、「そもそも銀の弾丸は狼男を殺すためのものであって、なんでも解決できる道具だったわけではないのでは？」とどうでもいいツッコミを入れてくなくなってしまう。つまり、銀の弾丸は銀の弾丸ではないのだ。そういう意味では「タダのランチなんてない」の方がまだ意味がわかる。

^{*15} 最近だと『ライティングソフトウェア』という本に、ドメインモデリングについて完全な勘違いをした上でドメインモデリングを批判する酷い議論が含まれていた

道』[16]を読んでみると感じられるのが、大学などのアカデミズムによる専門家教育に疑念を表し、原始社会の建築や職人仕事を称揚する、明らかな反知性主義が感じられる。

これがオブジェクト指向のコミュニティに受け入れられて、今のアジャイル手法やDDDへと繋がっていくのだ。だから底流に同じ反知性主義が流れていても不思議ではないと私は思う^{*16}。

ここまで紹介してきたハッカー文化、ソフトウェア開発文化における反知性主義はどれもどちらかといえば健全な現場主義であり、有害とは言えないものばかりであるが、それらがどれも無害なものかは私にはわからない。

あまり詳しくはないが、シリコンバレーなどのアメリカ西海岸のテック企業においては、加速主義やアイン・ランド^{*17}などの、過激な思想を標榜する者が多いとも聞く。これらは実際、反権威主義的で反知性主義な考え方と相性が良い。

もしテック文化の根底に反知性主義の影響があるのなら、私はそこから健全な現場主義を取り出し、無用な反権威主義を取りのぞきたいと思う。

そちらの方が単なる開発技術の今後のためにも、文化的摩擦の軽減のためにもいいのではないかと思うのだ。

しかしなぜそんなに話が錯綜としてしまっているのか。

反権威主義という成分を取り除けば話は簡単ではないのか。

単に「現場の形式化されない暗黙知」と「現場から離れた専門家の抽象的な形式知」のどちらも大切だということのはずだ。

ここまで単純化すると、「両方大事」という非常につまらない話になるのに、なぜこんなことで喧嘩するのだろうか。

私の非常に単純な見解を言うと、多分「自分のやっていることを少し違う見方をしている人がいるとなぜかムカつく」と言う人間の謎の本能が原因なのではないかと真面目に考

^{*16} その他にも、「ソフトウェア職人気質」[25] などという言葉でプログラミングの考え方をまとめようとするのにも反知性主義を感じるし、この業界が妙に「ジャーニー」と言う言葉を使うのもヒッピー文化が源流だと思われる。

ヒッピー文化の代表的バンドであるグレイトフル・デッドも自分たちの来歴を「ジャーニー」と読んでいる。最近ではグレイトフル・デッドのやり方をビジネスやマーケティングに取り込もうという本が数冊出ている [26] [27] ので、カウンターカルチャーとビジネスの奇妙な関係を観察したいなら読んでみるといい。

グレイトフル・デッドはまた、インターネット文化の初期の歴史にも関わっている。のちの雑誌文化に大きな影響を与えたヒッピーのための情報共有雑誌『全地球カタログ』の編集者であるスチュアート・ブランドはコミュニティ作りの道具としてのコンピュータに初期から注目していて、彼はその後最初期の電子掲示板 WELL を立ち上げる。グレイトフル・デッドのファンコミュニティも WELL 上に根付いていく。

そこから、国のコミュニケーション技術に対する規制に反対する団体「電子フロンティア財団」が誕生する。電子フロンティア財団が運営する無料の認証局「Let's Encrypt」は無料で Web ページを https 化するために広く使われている。

ちなみに電子フロンティア財団の創設者の一人は Lotus 1-2-3 の開発者であるミッチェル・ケイパーであるが、彼も非常にヒッピー的な人物で、プログラマーになる前はヨガによる瞑想のインストラクターをやっていた。彼の作ったソフトがなぜ Lotus (蓮) と名付けられているか、これでわかるだろう。

ことほど左様に、ハッカー文化とヒッピー文化の奇妙な関連を語り始めると止まらなくなるので、そろそろいい加減にしておこう。

^{*17} 実力主義や放任主義を肯定し、利他主義を徹底的に拒否した思想家。いわゆる新反動主義と親和性が高いと言われている。こころ辺の話は『ダークウェブ・アンダーグラウンド 社会秩序を逸脱するネット暗部の住人たち』[21] に詳しい。また、『闇 (ダーク) ネットの住人たち デジタル裏社会の内幕』[23] にもアイン・ランドを愛読する人物が登場する。

えている。

だからこそ、現場の人は現場の外からやってきた専門家を疑い、専門家は現場の人を軽く見がちなのだ。

この本能のせいで、ドメインエキスパートとソフトウェア開発者は喧嘩し、ソフトウェア開発者とコンピュータサイエンティストは喧嘩し、コンピュータサイエンティストと数学者は喧嘩し、数学者と数理論理学者は喧嘩する（これはあくまで例であり、いつも喧嘩しているわけではない）。

現場と専門家の関係は相対的なもので、ある場面で現場の人を軽く見る専門家な人が別の場面では専門家を疑う現場の人になることもあるのだ。何らかの摩擦が生じたら、自分が専門家側にいるのか現場側にいるのか少し反省してみるのもいいかもしれない。

先ほど「反権威主義を除けば単純になる」と言ったが、実際には反権威主義を取り除くのが大変なのだろう。それが人間の本能に根ざしているかもしれないのだから。

なので、私自身はこれが単純に解決できる問題であるとは思っていない。

当座の目標は整理である。

でも、どうせなら整理することで解決への手助けができればとは考えている。

これは解決すべき問題だと考えているからである。

1.9 連想と展望2：暗黙知と形式知

なぜ私がこんなことを気にしているのか、もう少し考えてみると、おそらくこれには「知識とはなにか」という個人的に気になっている問いと関係があるからだと思う。哲学で言う「知識論」の話だ。

論理実証主義などでは、知識というのは命題など、主に言語で書かれたものを意味していた。

しかし、哲学的な難点や脳科学の知見などを考えると、非言語的な知識も含めないと、どうも理屈が通りにくいと気づき始めた。非言語的な知識とは例えば、自転車の乗り方などの「～ができる」という形の肉体的な知識だ（こころ辺の話は戸田山和久の『知識の哲学』[11] などにある）。

これを化学出身の科学哲学者マイケル・ポランニーは「暗黙知」と呼び、それに対して言語化された知識を「形式知」と呼んだ [8]。この用語は説明なしですでに使っている。

これは最近の心理学の二重過程理論^{*18}とも相性が良い。

そしてアジャイル開発などの現場主義はこの暗黙知を重視する手法である。DDD もクリストファー・アレグザンダーのデザイン言語も、現場の暗黙知をどのように掬い取るか、という手法と考えられる。

これらはこれまで知識といえば形式知であった哲学などへのカウンターとしては非常に意味のあることであった。

科学などの形式知中心のアプローチでは、どうしても知識は酋長的で一般的なものにな

^{*18} 人間の思考には非言語的な早い思考と言語的なゆっくりの思考があるという理論。『ファスト&スロー』[22] と言う本が非常に有名だが、この本で紹介されている実験の多くは再現性について疑問が出されている。今のところ二重過程理論については今のところ問題がないとみなされていることが多い（将来的にどうなるかは知らない）。

る。そこでは、現場のさまざまな非言語的な知識は抜け落ちてしまう。

そして形式知は本来はそのような非言語的な知識と組み合わせて初めて活用可能なものになる。

例えばある場所に住んでいる生物の観察などでは、大学の生物研究者より、地元の自然観察者の方が詳しいことは多い。もちろん彼が生物学の理論に詳しいことはあまり期待できないだろう。そこには役割分担があり、どちらも欠けてはいけないのだ。

ところがこれまでの形式知偏重の反動か、ここでも振り子は逆側に振れすぎてしまう。

ポランニーは『暗黙知の次元』でなかなかすごい議論をしているので紹介したい [8]。暗黙知は目の不自由な人が杖で障害物を見つけるように働く。この時その人は杖を意識しない。杖を意識しないから、障害物を見つけることができる（二重過程理論が主張するように暗黙知は素早く働くので）。もしその人が杖を意識してしまったら、障害物を見つけることはできなくなってしまう（二重過程理論が主張するように形式知は素早く働かないので）。よって、暗黙知を形式知にすることはしてはいけない。

つまり哲学などはやってはいけない。

という議論だ。

ここには反知性主義への道が明らかに示されている。そしてそれはマイケル・ポランニーの生徒のエリザベス・シューエルが『オルウェイスの声：詩とナチュラリストリー』[10] で高らかに歌いあげた道だ。この本は、詩と科学は昔は同じものだったのに、今は別れてしまった。だからもう一度同じにしなければいけないと主張するのだ*19。

流石に正気の沙汰ではないと思う。

ポランニーがどこで間違っているかは私なりに理解していて、それは彼が知識を『個人的知識』と考えているところである（これは彼の著作 [9] の名前であり、エリザベス・シューエルも協力者である）。

知識を個人的なものと考えるのは、デカルトの『方法序説』などから続く古い伝統だ。これも、最近の哲学では疑問に付されている。

例えばフェルマー・ワイルズの定理の照明や超ひも理論の詳細などを一人の頭の中に収めている人など存在しないのである。我々が行っている行為は、論文を信用し引用することである。もちろんチェックは行われるが、必ずしも自分でしているわけではなく、チェック者を信用することで、論文を信用している。

するともし知識が個人的なものでなければいけないとしたら、多くの科学の知見は知識などではなくなってしまうのだ（[11] などの議論を参照）。

これを防ぐには知識は社会的なものと考えた方が良いことになる。

私は暗黙知が知識が言語的なものではなく肉体的なものと考えたことの、正しい続きだと思う。言葉の意味する抽象的なものではなく、脳などの肉体が知識を実現できるなら、複数の肉体の相互作用が知識を実現してもいいはずだ。

そして、言葉という制約を逃れることで、知識はさまざまな形を取りうるようになる。

『物のかたちをした知識 実験機器の哲学』[7] では、知識とは天球儀のような模型や様々

*19 似たような道筋で強い反知性主義にたどり着いてしまった例が『自然を名付ける』[13] である。この本では、ユキウス・キュルスの「環世界」などの議論 [14] を援用して、知識が肉体的で直感的なものであることを論じる。これは暗黙知と同様の議論だ。ここから、現代の分岐学的な生物の分類学では「魚類」などが存在しないことを論難し、「鯨を魚だと思ふ世界に帰ろう」と主張する。これも、専門的な形式知に対する現場の暗黙知の優位を強く主張する反知性主義の一種だと思われる。

な実験器具なども知識としていいのではないかと提案している。

それらは機能し、そしてそれを見た人に知識を伝えることすらできるのだから。

私はこれらのことを鑑みて、ソフトウェアなどはまさに「機能し、それを見た人に知識を伝えることができる」知識の一種だと考えている。

そう考えると、例えばアジャイルなどで、ドキュメントやコメントなどよりも機能するソースコード自体が重視される理由もわかる。つまり、ドキュメントやコメントは形式知であり、ソースコードとは暗黙知なのだ。

そしてドキュメントやコメントは形式知であるが故に、機能するのも変化するのも遅く、維持のコストが高い。だからドキュメントやコメントはソースコードが変更されたのに変更されず、陳腐化してしまいがちなのだ。

ドキュメントやコメントをあまり書かず、ソースコードに集中することで、維持のコストを安くし、変更に強くすることができる。これはまさに暗黙知の長所を使っていると言える。

ただ、暗黙知は小さな変更には強いが、一度途切れると修復がほぼ不可能であるという弱点がある。

チームが分解したり、しばらく開発が止まったりすることで、暗黙知は雲散霧消し、ソースコードからは何が何だかわからなくなってしまう。だから実際のアジャイル開発でも、ある程度のドキュメントやコメントを残すことで、形式知の力を利用する必要がある。

そう考えると、現代は知識の膨張とともにどんどん暗黙知の部分が大きくなっているのではないか、という懸念が生じる。形式知は高いコストがかかるが故に、現代の知識の膨張はほとんどは暗黙知の膨張によって賄われているのではないだろうか。かつてはそもそも人類の持つ知識が小さかったが故に、形式知こそが知識だと言い切ってそれほどの問題がなかったのかもしれないが、もう安閑とはしてはいられなくなったのではなかろうか、というのが私の考えだ^{*20}。

図書館の司書が形式知のためのインフラを支える仕事をしているのは分かりやすいであろう。博物館や美術館の学芸員は、さまざまな物品を保存することによって、暗黙知も支えている。分類学の知識は論文の中だけにあるのではなく、タイプ標本という「物のかたちをした知識」としても所蔵されている。それらが失われれば復元できない知識の欠損が起きかねない。

それ以外の人も、さまざまな形で人類の知識のインフラを保守している。例えば、現代のソフトウェアのほとんどはサポートありきで存在している。かつてのゲームはゲーム機のエミュレーターとゲームのソフトウェアがあれば遊べるが、最近のゲームは運営が終了すれば遊ぶことができなくなる。これらの運営も我々の知識の大事な一部分と言えると思う。

TwitterなどのSNSもデータの保存などはTwitterの運営とそれによって生じる収益に依存している。この収益構造も知識の大切な一部と言ってもいいのではなかろうか。Twitterの運営が終了した時に、それが保存され続ける可能性は非常に低い。

収益構造を知識のインフラに使うことで、インターネットは大きな発展を遂げたが、それは我々にコントロールできない不安定性も呼び込んだ。そして、そのようなインフラの

^{*20} いや大問題だ、当時だってそれを支えるのは民衆などの非知識人の持つものも含めた暗黙知だった、と言われればその通りだと思う。

上に作られた知識はどうしてもコストがかかる形式知ではなく暗黙知によって構成されるのではなからうか。

我々の社会の機能は、読むことが非常に難しい（そもそも読めない場合もある）プログラムや、それを運営する表に出ない人たちと、彼らを養う不安定な収益構造など、見えない不安定な活動に依存している。

我々が形式知だと思っていることは、未曾有の暗黙知の上に漂っている木の葉のようなものかもしれない。ちょっとしたことがあるだけで（例えば geocities の閉鎖で多くの web サイトが消えたように）、これらは消し飛んでしまうだろう。20 世紀後半から 21 世紀は後世、ほとんど資料の残っていない暗黒時代と見なされるかもしれない、というジョークがあまり笑えない理由だ。

本当にこれで大丈夫か、と私は疑問を感じずにはいられない。もちろん大した根拠のない、懸念ではあるが、大した根拠がなくても私の脳裏からなかなか離れてくれない。

何か、もうちょっと暗黙知をうまく形式知と繋げる方法はないものか。その抵抗になっ

てしまっている反知性主義をどうにか克服できないものか。そのようなことを頭の中でぐるぐるさせながら、ペあのしすてむが成功すれば、アカデミズムの手法とアジャイル手法をうまく繋げられ、反知性主義の抵抗なしに、暗黙知と形式知の上手な橋渡し方法が見つかったかもしれない、と夢みたいなことを考えてしまったりするのだった。

1.10 結び

というのも、実際今私が株式会社 RAKUDO でやっていることもそんな感じのことだったりするからだ。

ペあのしすてむでは数学の素養のある人にビジネスをさせようとして失敗したわけだが、今は会社員の経験はある人に、どうやってコンピュータサイエンスの知識を必要な分だけ学んでもらうのかを苦勞している。

みんな優秀なので、プログラミングや分析などは少しずつ着実に学んでくれるのだが、どうしても暗黙知ベースになってしまう。

ビジネスのサイクルの速さ的にはこれは正しいやり方だ。でも、そのままだと今までのやり方が古くなって新しい手法が主流になったときに、学び直しが発生する。暗黙知ベースだとその学び直しが大量になる可能性がある。

これが繰り返されると、学びを諦めてしまうのも仕方がないと思っているのだ。

なのでアカデミズムの形式知の積み重ねへのアクセスが持てるようになって欲しいと考えている。その基礎があれば学び直しを小さくすることができる、と個人的には考えている（根拠は微妙）。

学問がやれる人間にビジネス教えたほうが楽だという考え方は、ペあのしすてむの失敗を思うとあまり信じられない。とはいえ、ビジネスをやれる人間に学問を教えることが、いつでも可能ともとても思えない。

そもそもアカデミズムの形式知へアクセスするためには、アカデミズムの暗黙知を身につけなくては行けないが、これがなかなか大変なのだ。

こういうことを、なんとかしたいと思いながら、忙しくてたいして時間がとれない毎日である。

ペあのしすてむと直接関係のない妄想や繰りごとに紙幅を費やしてしまったので、ここで一度筆を置くことにしよう。

来年も忙しくなりそうだ。

1.11 2022-08-14 追記

実はこの号を出版した次の年 2022 年 7 月に、株式会社 RAKUDO も結局倒産することになった。

とほほ～。

潰れた理由はペあのしすてむと似たようなもので、どう考えても失敗しているプロジェクトに「もう少し頑張ればどうにかなる」と無駄なリソースをさきすぎていることである。ペアのシステムと違うところは、そこそこ仕事はあったことだが、利益に対して人が多すぎて（これもペあのしすてむと同じである）、しかも成功の見込みのないプロジェクトに時間と人員をさいては、キャッシュフローに大きな穴があいているということで、いくら仕事があってもうまくいきようがない。

なぜそんな会社が存続していたのかということ、ここでもペあのしすてむと同じ出資者がお金を出して存続させていたようだ。それならもっと早くに潰しておかなくてはいけなかったし、人を雇ってははいけなかった。それができなかったのは無責任だ、と私は考えている。

ペあのしすてむの時点では、僕は出資者は被害者と見ていたが、RAKUDO の失敗も鑑み、出資者の行動パターンも 2 回連続の失敗の原因であると見ている。簡単に言えば金を気楽に渡しすぎる、ということで、一見優しそうに見えるが、結局は人を狂わして、人間関係を破壊してしまっている。出資者との友人関係はその後も続いているが、彼が人に金を貸さないように、皆で監視をしている、という奇妙な状態が続いている。

と、2 回目の倒産も 1 回目と似ている部分はあるが、やはりそれなりの違いもある。RAKUDO の倒産がペあのしすてむの倒産と大きく違うのは、仕事が多かった分、RAKUDO のメンバーとは倒産後も友人関係を続けられている点だ。バラバラになってしまって、ほぼ音信不通のペあのしすてむのメンバーとはそこが違う。

RAKUDO の倒産後にその点が気になって、何人かには連絡を試みた。ペあのしすてむ倒産後には連絡も取りにくかった役員 C とも会ってみた。彼はペあのしすてむ倒産後も頑張っているようで、その点はよかったのだが、ペあのしすてむについては「あのまま続けていけばうまく行ったはずなのに、出資者が間違っことをしたからこうなってしまった」と考えているようで、その点はどうにも残念だった。

「ソフトウェア業界はこのままではいけない。なんとかしなくてはならない」という彼の熱意は私も共有できるのだが、ペあのしすてむの失敗の理由がそのような理解のままで、その熱意を成功に導けるのか、私には少し心配である。

おそらく彼の中では、形式手法について「こうすれば良い」という答えがあるのだろう。

私には、そのような答えがあるとはとても思えない。だからこそ迷い迷い、いつでもリスクを気をつけながら進まなくてはならないと思っている。そのような不確定な答えに従業員や出資者を巻き込むことに、もっともっと慎重にならなくてはいけない、と思っている。それが、「アジャイル」の重要な考え方はずだ。

まあ、彼が頑張って僕の考えが単なる杞憂であることを実証してくれるなら、それで構

わないが。

参考文献

- [1] トーマス, デイブ ハント, アンディ./角谷信太郎 木下史彦 監訳. アジャイルプラクティス 達人プログラマに学ぶ現場開発者の習慣. オーム社. 2007.
- [2] メイヤー, バートランド./土肥拓生 前澤悠太 末永俊一郎 訳. アジャイルイントロダクション: Agile 開発の光と影. 近代科学社. 2018.
- [3] エヴァンス, エリック./和智右桂 牧野祐子 訳./今関剛 監修. エリック・エヴァンスのドメイン駆動設計. 翔泳社. 2011.
- [4] ジュヴァル・ローウィー./長尾高弘 訳. ライティング・ソフトウェア. 翔泳社. 2021.
- [5] リース, エリック./井口耕二 訳. リーン・スタートアップ ムダのない起業プロセスでイノベーションを生み出す. 日経 BP. 2012.
- [6] ウェンガー, エティエンヌ リチャード, マクダーモット スナイダー, ウィリアム・M./櫻井祐子 野中郁次郎 野村恭彦 訳. コミュニティ・オブ・プラクティス—ナレッジ社会の新たな知識形態の実践 (Harvard Business School Press). 翔泳社. 2002.
- [7] ベアード, デービス./松浦俊輔 訳. 物のかたちをした知識 実験機器の哲学. 青土社. 2005.
- [8] ポランニー, マイケル./高橋勇夫 訳. 暗黙知の次元. 筑摩書房. 2003.
- [9] ポランニー, マイケル./長尾史郎 訳. 個人的知識—脱批判哲学をめざして. 地方・小出版流通センター. 1985.
- [10] シューウェル, エリザベス./高山宏 訳. オルフェウスの声; 詩とナチュラル・ヒストリー (高山宏セレクション〈異貌の人文学〉). 白水社. 2014.
- [11] 戸田山和久. 知識の哲学 (哲学教科書シリーズ). 産業図書. 2002.
- [12] 岡田猛 戸田山和久 田村均 三輪和久. 科学を考える—人工知能からカルチュラル・スタディーズまで 14 の視点. 北大路書房. 1999.
- [13] ヨーン, キャロル・キサク./三中信宏 野中香方子 訳. 自然を名づける—なぜ生物分類では直感と科学が衝突するのか. NTT 出版. 2013.
- [14] ユクスキュル クリサート./日高敏隆 羽田節子 訳. 生物から見た世界. 岩波文庫. 2005.
- [15] アレグザンダー, クリストファー./稲葉武司 押野身邦英 訳. 形の合成に関するノート/都市はツリーではない. 鹿島出版会. 2013.
- [16] アレグザンダー, クリストファー./平田翰那 訳. 時を超えた建築の道. 鹿島出版会. 1993.
- [17] アレグザンダー, クリストファー./平田翰那 訳. パタン・ランゲージ—環境設計の手引. 鹿島出版会. 1984.
- [18] アンダーセン, カート./山田美明 山田 文 訳. ファンタジーランド—狂気と幻想のアメリカ 500 年史. 東洋経済新報社. 2019.

- [19] マルコフ, ジョン./服部桂 訳. パソコン創世「第3の神話」—カウンターカルチャーが育んだ夢. NTT 出版. 2007.
- [20] ヒース, ジョセフ ポター, アンドルー./栗原百代 訳. 反逆の神話〔新版〕「反体制」はカネになる. 早川書房. 2021.
- [21] 木澤佐登志. ダークウェブ・アンダーグラウンド 社会秩序を逸脱するネット暗部の住人たち. イースト・プレス. 2019.
- [22] カーネマン, ダニエル./村井章子 訳. ファスト&スロー. 早川書房. 2012.
- [23] バートレット, ジェイミー./星水裕 訳. 闇（ダーク）ネットの住人たち デジタル裏社会の内幕. CCCメディアハウス. 2015
- [24] サイド, マシュー/. 有枝春. 失敗の科学 失敗から学習する組織、学習できない組織. ディスカヴァー・トゥエンティワン. 2016.
- [25] マーティン, ロバート./角征典 訳. Clean Coder プロフェッショナルプログラマへの道. ドワンゴ. 2018.
- [26] スコット, デイヴィッド・ミーアマン ハリガン, ブライアン./渡辺由佳里 訳. グレイトフル・デッドにマーケティングを学ぶ. 日経 BP. 2011
- [27] バーネス, ベリー./伊藤富雄 訳. グレイトフルデッドのビジネスレッスン#. 翔泳社. 2012
- [28] 永瀬唯. 疾走のメトロポリス—速度の都市、メディアの都市. INAX. 1993.
- [29] 石井遼介. 心理的安全性のつくりかた 「心理的柔軟性」が困難を乗り越えるチームに変える. 日本能率協会マネジメントセンター. 2020.
- [30] https://ja.wikipedia.org/wiki/メーデー!:_航空機事故の真実と真相
- [31] Claessen, Koen Hughes John. Quickcheck: a lightweight tool for random testing of haskell programs. In International Conference on Functional Programming, pages 268–279. ACM. 2000.

ぺあのしすてむ跡地訪問記

淡中 圏

株式ぺあのしすてむの住所は社長の自宅であった。

実際には株式会社ぺあのしすてむの本社は社長の実家の離れである。

そこを改造してパソコンなどが備えられて仕事ができるようになっていた。

個人事業時代からそうやって仕事をしていたのだろう。

ぺあのしすてむは完全リモートを目指した（潰れる寸前以外は）会社だったので、ここが本社でも大きな問題はなかったのだ。

ただそれでも時々ここに来てはいけなくて、交通の便が悪くて苦勞した。

周りは畑と工場しかない。

社長のお父さんは今でも畑仕事が好きらしく、さまざまな野菜や果物を育てて近所に配っているのだという（商売で農業をやっているわけではないらしい）。

ぺあのしすてむ時代に「西瓜が余って捨てなくてはいけなくて困っている」と言われたことがきっかけで今でも一年に一回は、余った野菜を受け取りに行って、簡単なバーベキューをしたりしている。

大きな西瓜半分をスプーンですくって食べて、お腹を冷やしたりした。

心の調子を崩した社長の様子を見ることも兼ねていたのだが、社長は今でも「心も体もボロボロだから普通の仕事はできない。でもこんな自分でもできる仕事はないだろうか」と前向きに物事を考えていて、むしろ自分が元気付けられている形だ。

本当に尊敬する。

一社員の分際ではあるが、今世話になっている RAKUDO で何か協力ができないか、考えている。

教育用の資料作成などなら、動ける時間で少しずつできないか、いやそういうことこそまとまって集中できる時間がないといけないから心と体の健康が重要か、などいろいろな悩んでいる。

さて、今年の夏も西瓜を食べに行ったら、こんなものを見せられた。



ぺあのしすてむの住所印である。

自分でも理由はよく分からないが興奮してしまって、上から下からためつすがめつしてしまった。

その様子を見た社長は一言

「いる？」

と聞いた。

一も二もなく

「います！」

と答えた。

一体これが何にいいのかは今もって分からない。

「ファンみたいだ」

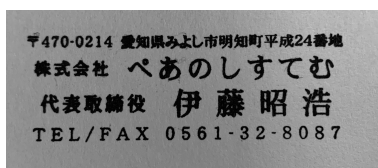
とその場にいる人は笑った。

確かに私は今でも「ぺあのしすてむ」のやろうとしたことのファンなのかもしれない。

これを今は自宅の仕事場の机の引き出しに入れている。

そして時々取り出しては、白い紙にボンと押している。

私は印鑑を押すことの苦手さだったらかなり自信のある人間^{*21}だが、これなら簡単だ。



さすがブラザー。元々は米国製ミシン「シンガー」に対抗しようと語感の近い「シスター」という社名にするつもりが、すでに商標登録されて使えなかったらしい。元々は「安井ミシン商会」だったのが創業者の6男4女の長男が事業を引き継いだときに「安井ミシン兄弟商会」になり、「シスター」が使えなかったのでこの社名から「ブラザー」にしたという。

というわけで、2021年12月31日、忍び寄るコロナのオミクロン株の恐怖に怯えながら開かれるコミケ C99において、この本を購入してくれた人は、白い部分などに自由にこのスタンプを押していいですよ。

消毒用のアルコールなども持っています。

^{*21} 横になる、逆さまになる、薄れる、違う場所に押してしまう、ならいい方で、一度慌てて印鑑の背の方に朱肉をつけてボンと押して、真っ赤な満月を作ってしまったことがある。多元の職員さんを笑わせてしまった。

変な会社が確かに存在したという、何かの名残を感じてくれたら幸いです。

執筆者一覧:

淡中 圏 本名：田中健策

本文にも書きましたが、前回のコミケでは本をキンコーズで印刷した帰りに会社が潰れると聞いて、その次の日にブースに座っていました。

今こうして本を後書きを書いていると、会社というのはなんとも脆く、そして数学界隈や同人などのコミュニティというのはそれよりずっと息の長いものだ実感しています。

だからこそ会社というものは割と簡単に作れ、コミュニティというものはよほど運が良くないとそう簡単には新規に作ることはできないのであろう。

なので、しばらくやめる気はないので、みなさまお付き合いください。会社は潰れても同人は続くよ、どこまでも、というわけなのです（どういうわけなんだ）。

よく分からない自作 web ページ <https://tannakaken.xyz/>

編集後記:

前回、「さあ、The Dark Side of Forcing の明日はどっちだ？ まあ、Dark Side に決まっています」で編集後記を締めましたが、結局 The Dark Side of Forcing の明日はどっちだったんでしょうかね？

明日になったらまた「明日はどっちだ？」と言っているのでしょう。

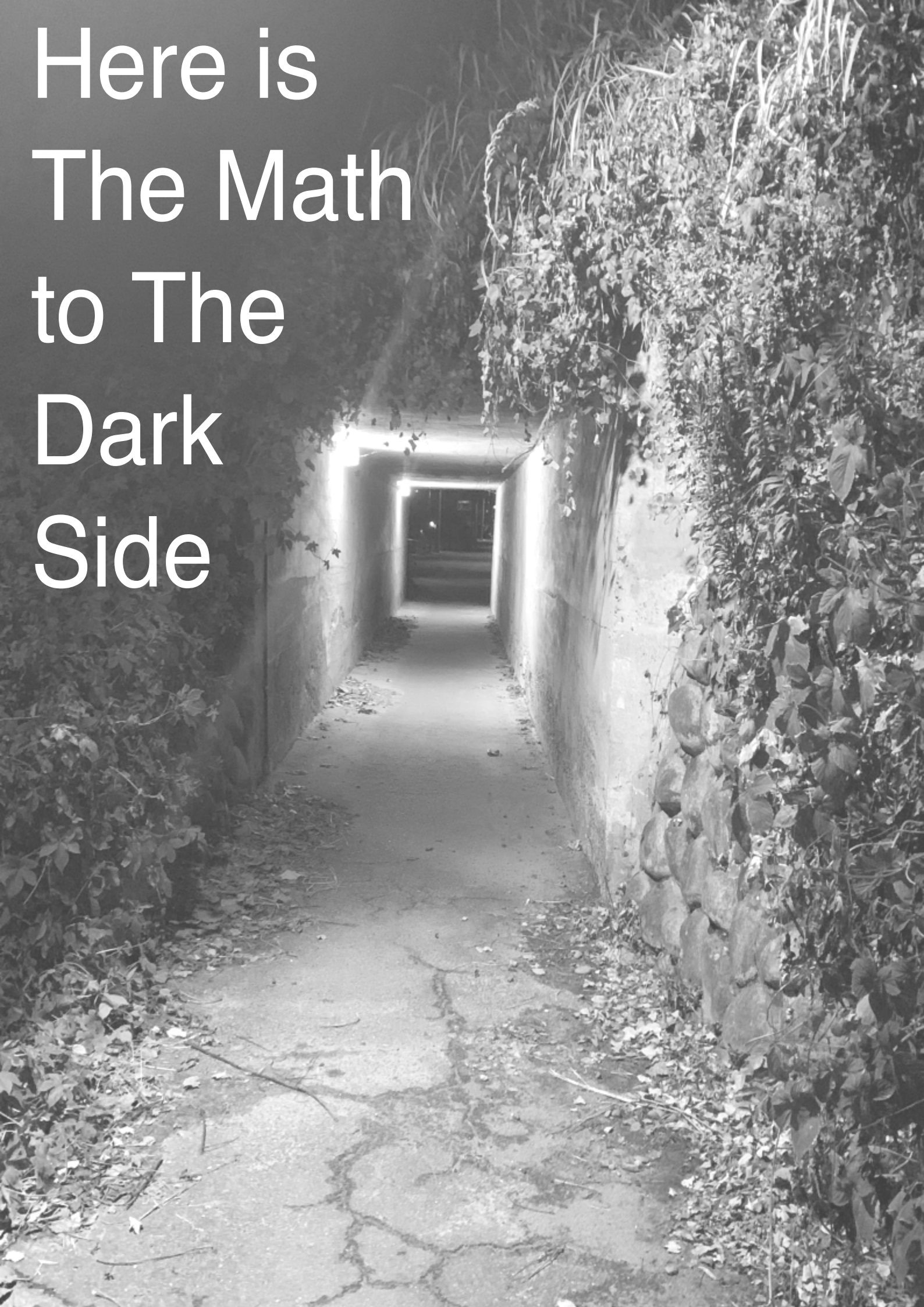
あと、誰か編集後記だけでも書いてくれないかな？ 編集しなくてもいいからさ。

【淡中 圏】

発行者 : The dark side of Forcing

連絡先 : <http://forcing.nagoya>

発行日 : 2021 年 12 月 31 日



Here is
The Math
to The
Dark
Side