

思考のリファクタリング

1-1. 全てのバグは思考の中にある

組織で、エンジニアリングを考えると、個人で書く場合には発生しない問題へ直面することが多々ある。

その問題に対して、他人と関わりを踏まえてどう振る舞うかを求められる。

その中でコミュニケーションの理不尽など、多くの感情に包まれ対立が起こる。

それらバグをよくしていくには、考え方を少し変えていく必要がある。

「思考のリファクタリング」が必要。

言い方を変えれば、「不確実性に向き合う」 考え方を考えていく。

1-2 不確実性とエンジニアリング

エンジニアリングとは「実現」していくための科学分野と言える。

「実現」には始まりと終わりが必ずある。

はじめ

「何を食べたいのか？」 → 曖昧

おわり

「ハンバーグを食べたい」 → 実現

不確実性コーン

実現とは不確実な状態から確実な状態に持っていくこと。

つまり、「どうしたら効率よく不確実性を減らしていけるのか」がエンジニアリングでは重要

組織も不確実なものを確実にできる組織が強い

「具体的で細かい指示」が必要 → マイクロマネジメント

「抽象的で自由度のある指示」

これらを比較した時、指示する人がいなくなると回らなくなるのでマイクロマネジメントされた組織は、組織能力に上限が存在する。

抽象的で自由度のある指示で動ける組織は、「不確実性」を「確実」なものに変えていける力を持ち、自己組織化されていると言える。

マイクロマネジメント

「何が起こるかわからない」 「ある程度わかる」 の差

天気で 晴れと曇りの確率が

- 50:50
- 80:20

の二つでは不確実性の度合いに大きな違いが生まれている。

こうした不確実性をより多く減らしていくことが「実現」への道となる。

不確実性 = 「わからないこと」

わからないことは本質的には2つしかない。

- 未来 = 「環境不確実性」
- 他人 = 「通信不確実性」 「コミュニケーション不確実性」

人間は不安から逃げる生き物

人間にとって「わからない」は「不安」につながる。

そして、「わからない」に向き合うことを避けてしまう習性がある。

例： 人間ドックに行くのが怖い。なぜなら将来の病気が不安だから。

だから、わかっている「安心」を優先したり、攻撃したりしてしまう。

でも、「不安」は減らない。

減らすためには、「情報を生み出すこと」が「不確実性を下げること」つながる。

エンジニアリングの本質は「不確実性の削減」

1-3. 情報を生み出す3つの考え方。

- 経験主義と仮説思考
- システム思考
- 論理的思考の盲点

経験主義と仮説思考

情報を入手するために、行動を起こして、その結果を観察し、そこから問題解決を行う考え方を「経験主義」と言い、

限定された情報であってもその情報から全体を想定し、それを確かめることで少ない情報から問題解決に向かう思考様式を「仮説思考」という。

システム思考

正解は1つではない。

そのため、全体像を見極めて正解を設定する必要がある。

時間と資源の制約の中で、より正解に近づく一手を打ち続けることが重要。

論理的思考の盲点

人間は感情的な生き物。

いつでも、論理的に考えられるわけではない。

それを乗り越えて、問題を正しく認知する必要がある。

人は論理的でなくなる可能性があると思った上で揉んだ解決に臨む必要がある。

1-4. 論理的思考の盲点

論理的思考とは演繹思考のこと。

この論理的思考には、以下の前提がある。

- ・ ルールと思考を正しく認知できること。
- ・ 正しく演繹できること

画像

これらの前提が崩れると、正しくない答えが導かれる。

事実を正しく認知できる

認知は大小の違いはあれど、変化してしまうもの。

だから、ありのままの事実を完全に正しく認知することはできない。

だから、自分の認知がいつ、どのように歪むのか把握する必要がある。

技術領域でも、「事実」「意見」は区別する必要がある。

感情に囚われず判断する

感情による短絡を排除する。

「AさんはBさんを見下しているからこのようなことを言うのだ」のような曲解を避ける。

非論理的に考えない = 論理的に考える

「論理的に考える」に注目するよりも「非論理的に考えてしまう」瞬間を知ることが重要
論理的思考能力とは、「感情的になる瞬間を知り、その影響を少なくできる」能力

人は正しく事実を認識できない

雨が降った時、人はいろんな認知が入ってくる。

苛立ったり、憂鬱な思いになったりするイメージが湧いてきたりもする。

ただ、事実は「雨が降った」だけであり、その他のイメージは全て認知。

自分の認知がどう歪むのかを把握し、客観視することが重要。

画像

ベーコンの4つのイドラ

フランシス・ベーコンは経験主義の祖であり、人間の認識には様々な錯覚や誤謬が含まれることを説いた。

「ノヴム・オルガヌム」という書で「イドラ」という人間の錯覚・認識の違いが存在することを伝えてた。

- 種族のイドラ
- 洞窟のイドラ
- 市場のイドラ
- 劇場のイドラ

種族のイドラ

人間が持っている性質から生じる偏見

「遠くにあるものは小さく見える」「暗い場所ではものがはっきりと見えない」など

洞窟のイドラ

個々の取り巻く環境から、外の世界を知らずに一般的なことだと決めつけて理解することから生じる偏見

「自分がそうだから、他の人もそうだと思う」

「自分の家では目玉焼きにマヨネーズだからみんなそうだ」

市場のイドラ

言葉の不適切な使用から生じる誤解や偏見

「噂話やデマを信じてしまう」

劇場のイドラ

伝統や権威を無批判に受け入れて、誤った考えでも信じてしまうこと

「偉い人の言っていることは正しいだろうと思う。」

など

認知の歪み

不安や抑鬱状態を固定化させる思考が、どのようなもので、どのような認知の歪みとして現れるか分類したもの。

ネガティブな感情や不安は「認知の歪み」を発生していて、それを正していくことで、ネガティブな感情を取り除くことができる考えた。

- ゼロイチ思考
- 一般化のしすぎ
- すべき思考
- 選択的注目
- レッテル貼り
- 結論の飛躍
- 感情の理由づけ

ゼロイチ思考

白か黒か。敵か味方か。など、間のグラデーションを認識できずに捉えてしまうという認知の歪み

一般化のしすぎ

「主語が大きい」

事例の1つ2つを全体の問題としてしまうこと

すべき思考

すべきだからとそれらを強制する思考パターン。

他人に対して強制したりしてしまう。

これらは自分にも向かうので、鎖となって自身を追い詰めることにもなる。

選択的注目

「心のフィルター」

人は見たいものしか見ないとも言い換えられる。

一度そうなのだと思い込むと、フィルターを通して見てしまうことが多い。

例えば、ネガティブな印象をもった人は何をしてもネガティブな風に見えてしまったり。

レッテル貼り

一般化のしすぎが加速するとレッテル貼りになる。

属性のみに注目して、それが原因としてしまうもの。

「男だから」「女だから」「エンジニアだから」「営業だから」など

結論の飛躍

「先回りのしすぎ」「心のよみすぎ」

「既読スルー」はきっと怒っているからだ。など

感情の理由づけ

感情を根拠に自分の考えが正しいと判断するような認知の歪み。

「不安」だからこれは失敗する。

「あの人のこと嫌い」だから「あの人は価値がない」など。

ここから

認知的不協和

自分のなかの矛盾（不協和）を解決するために、認知自体を歪めること。

例

タバコは体に悪い。

だが、タバコは体に悪くない。他の原因で死ぬ人が多いなど正当化できる情報を取りに行く。

本人もわかっていることを上から「健康に悪い」と伝えることはその行動を起こせない理由を作ることになってしまう。

扁桃体をコントロールする

動物の脳は危機を察知したり、生命を脅かされると脳の「扁桃体」が活発化し、

「逃げる」「吠える」「体を戦闘に備える」などの生き残りの行動が素早く行えるようになっている。

それらは原初的なトリガーに過ぎない。

その後、理性や経験などが総動員されて、やがて「怒り」に変わる。

恐怖は「怒り」になる。

怒っている人は自分の知的な能力を使ってそれらの危機を乗り越えようとする。

「怒り」が日常で発生するのは「自分の大切にしているもの」に被害が及びそうだと感じているということ。

「怒り」を感じた時、同時に「大事なこと」を知るときでもある。

「自分自身の延長線上にあるもの」「自分自身を構成するもの」だと感じる何かでもある。

自分のアイデンティティの範囲を知る。

「自分自身を構成していること」をアイデンティティという。

アイデンティティと思っている範囲を攻撃されると怒りを感じる。

例：仕事の進め方、チームメンバ、宗教、会社、国家

アイデンティティが広い人ほど怒りっぽく感じ

狭い人は優しい人でもあり、何にも関心のないひともかもしれない。

画像

「怒り」を「悲しみ」として伝える

「怒り」をそのままの感情で伝えてしまうと「恐怖」を生み「怒り」の連鎖が広がってしまう。

意図的に、他人のアイデンティティを攻撃するというのはよっぽど悪感情がない限りはほとんどない。

全く知らずにぞんざいに扱ってしまったりがほとんど。

相手が何を大事にしているかを知るのは難しい。

それは友人・恋人・長年連れ添った夫婦でさえ同じ。

ついうっかり「その人の大事にしていることを傷つけるようなことをしてしまった。」という
ことに
気づかないことの方が多い。

もし、自分が誰かに怒りを感じたなら、これらの前提を元に

「それは自分にとって大事なことで、その発言は大事なものをぞんざいに扱われたようで悲
しい」と相手に伝えた方が良い。

なぜなら、その発言が攻撃していると気付かなかっただけだから。

問題解決より問題認知の方が難しい

以上のことから、感情による歪みや認知の歪みは誰にでも発生しうる。

だからこそ、人や自分が非論理的になるのかを知ることが重要。

実際の問題では感情的な対立や、論理的な正当性があるかのようなオブラートが何重にも包まれている。

正しく認知するためにはこの歪みを取り除く必要がある。

「自分は間違っているかもしれないが、それに早く気付く方が良い」と思考のパターンを変える必要がある。

kokokara

経験主義と仮説思考

わからないことは調べるしかない

Q. 3枚のトランプのうち、ハートのエースはどれか？

A. めくってみる

考えてもわからないことはある。

考えたらわかると思ってしまうことがあるが、そうじゃない時もある。

わからないことを行動で突き止める

正解に辿り着くためには確かめること。行動することが正解への一歩となる。

それが経験主義にも通じる。

× 「わからなかった」 → 「頭が悪かったから」

○ 「わからなかった」 → 「何をすればわかるかを思考する」

不確実性と夏休みの宿題

「自由研究」「苦手な教科の宿題」「美術の課題」

どのくらい時間がかかるか読めないものが多く残ってしまうと憂鬱となってしまう

これは「不確実性」が高いものが残ってしまったから。

逆にこれらを先に片付けてしまえば、完了時期の予測精度は上がる。

これは「経験主義的に不確実性を効率よく下げるという発想に基づいたプロセス」

ただ、不安に立ち向かう行為のため難易度は高い。

プロフェッショナルの仕事

プロほど、短い期間に一定のクオリティに仕上げていくが、

アマチュアは残りの短い時間で急速に出来上がる。

これは不確実性の低いものから取り掛かるかの違いとも言える。

コントロールできるもの/できないもの

「行動できることは何か」「行動の結果起きたことを観察できるか」が経験主義で重視される。

コントロールできるできないを把握することが重要

「上司が自分を評価してくれない」

「新入社員が「ゆとり」で全然仕事ができない」

コントロールできない

- 上司の自分に対する内心での評価
- 上司の評価基準
- 新入社員の能力
- 新入社員の内心

コントロールできる

- 上司の評価基準を詳しく聞くという行動
- 上司の評価基準に合わせた自身の行動の変化
- 上司を変えるための異動などの行動
- 自分の仕事を上司に詳しく説明するという行動
- 自分自身の思いを上司に知ってもらうための行動
- 新入社員への指示の出し方
- 新入社員の行動へのフィードバック

観測できるもの / できないもの

先ほどのコントロールできるかできないかは「行動する人の意志」で変えられるものを挙げた。

つまり「直接」コントロールできるもの

「観測できないもの」は「コントロールできない」

「内心」は観測もできないのでコントロールできない。

間接的にコントロールできるものは「新入社員の行動」

行動を変えさせるために自身の行動を変化させ観測することが重要。

kokokara

仮説思考

少ない情報で大胆に考える

画像

- ・ 演繹法
- ・ 帰納法
- ・ 仮説法

仮設法で大事なこと

「痕跡がわずかであっても」「確かめる行動につながる」
がポイント。以下はNG

「十分な証拠が揃ってないから、仮説が作れない」

「今までの前提から導けなかったからこの仮説は間違っている」

これらのせいで「確かめる行動につながらない」ことは多い。

PDCAサイクル

仮説といえば、有名なものは「PDCAサイクル」

Plan（計画）

Do（実行）

Check（検証）

Act（改善行動）

これを仮説とその検証にフォーカスして考えると
画像となる。

仮説検証のコツ

現実には「仮説」の定まらないままに計画が立てられ、何を検証するのかが明らかではない状態というのがよく見られる。

仮説検証サイクルで重要なのは「何が仮説なのか」を明らかにすること。

そしてそれは「検証できるのか」というアイデアを持つこと。

この二つが揃わないとPDCAが回らない。

データ駆動な意思決定の誤解

「データ駆動な意思決定」と聞くと

「データから結論や決定が導かれて」「次にとるべき正しい行動」が見えるような気がする。

ただ、実際にはデータは常に不完全なもの。

そこから意思決定は導くことはできない。

「データ駆動な意思決定」とは

- 「仮説」を推論するために、持っているデータの可視化をする
- 「仮説」が正しかったのか、統計的に検証する

ということにおいて、有効的な考え方であって、
データから演繹的・帰納的に答えが導けるものではない。

そのため、数少ないデータから大胆に顧客のインサイトや仮説を推論し正しいかを検証していくことが大事。

リアルオプション戦略

未来はどうかかわからないので、仮説が正しいのか間違っているのかを検証するには「やってみる」しかない。

この不確実性を逆手に取ってメリットに変える戦略を

「リアルオプション戦略」という。

遅延した意思決定

「遅延した意思決定」をするための戦略

早期に大きな投資判断などの意思決定を行うよりも

小さく失敗をして、成功しそうなときに大きく投資するという手法。

つまり、不確実性が少ない時と大きい時で投資の割合を変える。

開発で置き換えると

チャレンジングなことをするときには小さく検証を試みる。

成功したら、大きくでて行う。

株などで行われる戦略ではあるが、開発手法にも使える検証の考え方。

実際に、「リーンスタートアップ」という書籍でも、
新規事業での小さな仮説検証の重要性を説かれている。

そして検証から得られたものを元に新しい仮説に方向転換していくことを「ピポット」と呼んでいる。

[まとめ]問題解決よりも問題の明晰化の方が難しい

必要十分な情報が揃っている状態での問題解決はそこまで難しいものではない。

難しいのは情報が不完全な状態から始まる問題。

そのため「問題をはっきりさせる」経験主義と仮説思考は重要となる。

経験主義・・・自分のコントロールできるものを通じて、観測できるものを改善する

仮説思考・・・限られた情報の中から、大胆にモデルを推論し、そのモデルの確からしさを発見するための行動。

この二つから「問題は何なのか」を明晰することができれば不毛な思考にならないはず。

