現場で使える!自動化入門

北崎 恵凡、山田 雄一、鎌田 誠、瀧川 大樹、青木 敬樹、大野 泰歩、松岡 光隆、新山 実奈子、百合宮 桜、小崎 肇、澁谷 匠、慈英　著

2023-07-31 版　　発行

はじめに

お読みいただきありがとうございます。

某通信会社でインフラ設備の運用保守業務を担当し、日夜、自動化・効率化に取り組み、日常の課題を解決しています。

SRE(Site Reliability Engineering) を目指して活動していますが、運用保守業務はいわゆる「コストセンター」と呼ばれ、サービスやシステムの信頼性を高める活動や付加価値を創造する活動にもあまりコストを掛けられません。

既刊「現場で使える!Google Apps Script レシピ集」、「図解と実践で現場で使える Grafana」の執筆メンバーに加えて、本書は RPACommunity の有志が自動化のノウハウをまとめたものです。

普段、現場で自動化に取り組んでいる経験が誰かのお役に立てれば幸いです。

2023 年 7 月

北崎 恵凡

表記関係について

本書に記載されている会社名、製品名などは、一般に各社の登録商標または商標、商品名です。

会社名、製品名については、本文中では©、®、™マークなどは表示していません。

目次

第 1 章

RPACommunity とは

RPACommunity 代表 Mitz (松岡 光隆)

IT を活用した業務の自動化や IT 推進全般に興味を持つ全国の有志が集うコミュニティで、2018 年 3 月に立ち上げました。

名称には「RPA」(ロボティック・プロセス・オートメーション) が付いていますが、RPA 専用のツールやサービスだけに限らず、様々なツールやプログラムも含め IT 技術全般を対象にした学びの場を提供しています。

発足から 5 年以上経ちメンバー数も増え続け 8,000 名を超えているのですが、今現在でもコミュニティ内では様々な手法での業務自動化トークが飛び交っています。

そんな RPACommunity 内の「自動化ネタ」を、まだ RPACommunity を知らない方々にも知っていただきたいという想いを持つ有志が集って今回の書籍作成に至りました。

RPACommunity[*1]では 2023 年 5 月時点で 300 回以上のイベントを実施しており、その資料はこちらに掲載されております。

[*1] https://rpacommunity.connpass.com/presentation/

図 1.1: connpass の QR コード

また、2020 年以降のイベントは全て YouTube[*2]に公開しております。

[*2] https://www.youtube.com/@RPACommunity

図 1.2: YouTube の QR コード

ご興味ある方はぜひご覧ください。

本書はこれら RPACommunity での情報や登壇をベースに、RPACommunity 有志メンバーが書き上げております。

第 2 章

RPA をいれたり自動化を検討する前に考えたい事

山田 雄一

2.1 はじめに

RPA を導入したり、自動化を検討したり、DX を始めたい、またそれ以前の話として現状の情報システムを改善したい…… と思ったこと、ありませんか？ また、自分は乗り気でなくても、上から社命として言われてしまったり[*1]したこと、ありませんか？

筆者としては、ここ数年 (2015 年、16 年以降) は、RPA やノーコード、ローコードツールが一般的になった事により、単純なシステムリプレース以外に、取れる選択肢がずいぶんと増えたな……という印象があります。また既存システムにちょい足ししてみたり、既存システムのデータエントリー部分をノーコード、ローコードツールを使うようにすることで、こういった業務上の課題を解決できるかも……？ という機運が高まってきており、実際に活用例も多くなってきているように思います。

これらの RPA 系のツールは、「RPA 系のツールを使えば業務改善ができる」っぽく見える事が多いのですが、実際は打出の小槌のようにあらゆる業務改善や課題解決に効いてくれ、私達の業務を楽な方へ導いてくれるものなのでしょうか？

この章では、こういった時に何を判断基準にしたり、どういった懸念点があるのかをまとめてみました。RPA 導入以前の検討段階として、「RPA を導入したり、自動化を検討する」までに必要な検討の助けになり、ある程度[*2]の道筋を示せれば良いなと思っています。

[*1] こっちの方がもっと深刻である

[*2] もちろん筆者の個人的な考えと独断と偏見によるものです

2.2　世の「情報システム」にも色々ある

世の中には色んな会社があり、色んな業務プロセスがあり、そして色んな情報システムがあります。

情報システムについては、色々なものがありますが、ターゲットを絞らないとこの手の話はついつい発散してしまうものなので、参考となりそうなケースを次の様に分けてみましょう。

- スクラッチ開発の自社基幹システム
- パッケージソフトを使った自社基幹システム
- 自前システム（SQLServer+AccessVBA とかでしっかり目の構成のもの）
- 自前システム（と呼べるかどうかわからないけど Excel+VBA ぐらいのもの）
- システムなし

これらについて、それぞれ次のような対応策を今回は考えてみましょう。

- 今のシステムを改修して隙間を埋める
- RPA で隙間を埋める
- RPA じゃないツールで隙間を埋める
- 今のシステムをそのまま諦めて使う
- 人を増やす

上で挙げたそれぞれの **システム×対応策** について、それぞれの対応策の場合の検討ポイントをそれぞれ考えていきたいと思います。

2.3　SIer によるスクラッチ開発の自社基幹システム

SIer による受諾開発（スクラッチ開発）で納品され、運用されている自社基幹システムの場合、受諾開発という契約の特性を考える必要があるケースがあります。

2.3.1　今のシステムを改修して隙間を埋める

この対応策は、開発してもらった SIer に再度追加改修をお願いする事になることが多いですが、改修時のコストがかさむ事もあります。

また、SIer の開発人員も暇ではないので、継続的に追加改修をお願いしていない場合、改めて追加改修をお願いしたとして、実際に当時作っていた人が継続して対応してくれる

か、品質が担保[*3]されるか、みたいな所が悩ましくなることもあります。だいたい裏側は分かんないので……

また筆者の実体験として、継続的に改修を行っているシステムであれば、継続改修されているのでコストもそこまでかからない事もありますが、しばらく触っておらず、時間が経ってから改修を行うようなケースでは、過去のことを思い出す/既存システムの解析にもコストがかかるので、結構地味に大変だったりします。(大変なときは総じて見積もりに跳ね返ります。)

2.3.2 RPA で隙間を埋める

現行システムの改修と比べ、RPA で隙間を埋めた時の最大の利点は、「システムの内部を修正していないのに、業務オペレーションは簡単にできる (かもしれない)」点です。

RPA は良くも悪くも「画面操作をするツール」で、システムの内部を修正しませんが、これはケースによっては大きなメリットになります。

○うれしいケース 1

たとえば、社内で ISO 認証などの外部監査がある認証を取得しており、この監査情報の取得や集約がシステムに組み込まれている場合、社内システムの改修をすると、改修箇所(時によっては監査箇所全般)の処理内容について、再度監査が必要になってしまうケースがあります。

ところが、RPA での業務改善を行うとあら不思議！　社内システムから見ると、「画面操作をしているだけ」ですね！

これならどんな改善をしても安心！

○うれしいケース 2

たとえば、最初のスクラッチ開発は SIer に頼んだが、その後の改修については他社に頼んだり、自分たちで行うケースもあるかもしれません。(プロジェクト炎上したりするとありがち)

よく受諾開発の契約書にはあるあるなのですが、システムを開発納入した会社以外が改修した場合は、納入会社はシステムの瑕疵担保責任をその後一切追わない　とする契約となっているケースがあったりします。(責任を負えなくなるのでそれはまあ当たり前といえば当たり前)

こういった場合に、システムを直接改修するのではなく、RPA を使って改善を行った場合、やはり社内システムから見ると、「画面操作をしているだけ」になるのでシステム利用時の瑕疵担保責任が継続され、そういった点でも安心だったりします。

[*3] ここで言っている品質とは単に「バグがない」というだけではなくて、ソースコードの保守性であったり、応答速度であったり、データ量であったりといった非機能要件も含みます。

2.3.3　RPA じゃないツールで隙間を埋める

RPA じゃないツールで隙間を埋める　というのもケースによっては非常に効きます。

基幹システムを作ったはいいが、そんなに今後長いことこれを使わないと思うので、システム自体に**改修は入れたくない、でも現状の業務は厳しいので改善をしたい**だとか、RPA ではやりにくい**大量データの捌きを行った上で、処理を実現**する必要がある……といった場合に、たとえば ETL ツールがハマったりする事があります。

ただ、RPA じゃないツールの場合において、データの I/O は API 経由だったり、DB への直接アクセスが手段になりがちなんですが、パフォーマンスが劣化したり、本来の基幹システムからのデータ I/O ではない I/O（特に書き込み処理）が発生するので、既存システムとのそういった所の統制を取ったりするのが思った以上に大変なのと、当然ながら前述したような監査があるシステムの場合、悩ましい事になります。

2.3.4　今のシステムをそのまま諦めて使う

改修コストとそれにともなって得られるメリットが見合わない場合はこうなりがちです。やむを得ないのです……

2.3.5　人を増やしてオペレーションで解決

人を増やして対応するのはある意味最強の一手です。ハネムーン係数なども改善するかもしれません。やったね！ 業務が強くなるね！

※ただしボトルネックに人が増えない場合はなにも変わらず教育コストだけがかさむケースも……

2.4　パッケージソフトを使った自社基幹システム

パッケージソフトの場合にはパッケージソフトならではの検討ポイントがあるのでは？と筆者は考えています。

2.4.1　今のシステムを改修して隙間を埋める

そもそも、大規模な基幹パッケージ（例えば SAP とか）を入れていて、既に運用が回っているフェーズならば、システムを改修したいと考える会社であれば、ほぼほぼアドオンによるカスタマイズが入った状態になっているはずで、その上で現在のシステムと業務にギャップが発生しているので、そこに更にアドオンによるカスタマイズを付け加える形（付け加えたいと考えている状態）になっているのでないかなと考えられます。

ただパッケージソフトのカスタマイズというのは、一般論でいうと２割の処理にカスタマイズを入れるのであればフルスクラッチで作るのとコストが変わらないと言われているぐらいのもので、非常に開発コストがかかるものです。

また、一般的な大規模パッケージのベストプラクティスは「極力カスタマイズなしで使い、業務をパッケージに合わせる」となっている事が多く、そういった意味でも改修は悪手となりがちですが、現場要望もあって、情シスとしても押し負けてしまうことが多く、なかなかそういった理想的な形にはならない実情がありますね……

■コラム: コラム：パッケージソフトにおいて、カスタマイズがないとなにが嬉しいのか？

カスタマイズなしで使うと何が嬉しいかというと、パッケージのバージョンアップの際にカスタマイズはすべて見直しが要る事が多く、カスタマイズがなければパッケージのバージョンアップがサクサクで進むので、維持管理コストがずいぶん軽くなるのです……（カスタマイズがあればあるほど、再検証や再開発のリスクが上がる）

なお、大規模パッケージを使う時にも、会計系の箇所については原則として一切カスタマイズを入れないケースが多く、これは前述の「監査がある場合」に近い話が待っている事が多いです。

具体的には、こういったパッケージを入れる時の大きなメリットの一つとして、会計監査の省力化があげられます。

フルスクラッチで会計システムを構築した場合、大きな企業さん[*4]では担当の会計事務所にシステムの監査をして貰う必要があり、これが経理担当の悩みのタネだったりするのですが、有名な会計パッケージであれば、それをそのまま素直に使ってさえいれば、ある程度の会計システムとしての正しさをパッケージが保証してくれるため、会計的にチェックすべき点が相当シンプルになり、会計事務所が嬉しいので、経理担当も非常に嬉しいんだそうで……

2.4.2 RPA で隙間を埋める

こういったケースでもやっぱり RPA はオイシイです。だって、画面しか触らないですからね。

また、前述の「２割の処理にカスタマイズを入れるのであればフルスクラッチで作るのとコストが変わらない」みたいな話を、パッケージソフト側ではなく、その外で持てるよ

[*4] 上場してたりだとか…

うになるので、そういった意味でも良いです。

加えてコスト面でも有利なことがあります。フルスクラッチ開発と比べても、大規模パッケージのカスタマイズができる人というのは、そのパッケージに精通したコンサルレベルの SE であることが多く、そういう人にお願いするとどうしても単価そのものが高くて、結果これが開発コストとして高くなりがちなのですが、RPA であればそういった高単価の人をアサインせずとも、業務改善が実現できるかもしれません。これは嬉しいポイントですね。

2.4.3 RPA じゃないツールで隙間を埋める

RPA じゃないツールも前項の「RPA で隙間を埋める」と同じ観点でオイシイです。ただ、内部データを直接 API で触る事になるので、投入したデータがパッケージ上でうまく処理されることを検証しておく必要があります。（筆者は実際にそういうテストをやったこともあります……笑）

2.4.4 今のシステムをそのまま諦めて使う

人生諦めも肝心です。投入コストより回収リターンのほうが少ない場合は「何もしない」が最適解だったりしますよね…（特にパッケージは改修コストがかかりがちなので、ノーカスタマイズで使って人が合わせる方が良かったりすることもままあります。）

2.4.5 人を増やしてオペレーションで解決

人こそパワー！（※前の「スクラッチ」の話と全く論点は変わらないです）

2.5 自前システム（SQLServer+AccessVBA とかでしっかり目の構成）

さて、コストをがっつりかけて SIer に基幹システムを作ってもらったりはしておらず、パッケージソフトを入れていない会社というのも意外に多いものです。（私も前職は社内 SE だったのですが、このパターンでした。）

2.5.1 今のシステムを改修して隙間を埋める

自前システムだと改修しやすい面と、改修が難しくなる面があります。

改修し易いポイントは、自前で作っているのでコスト面や対応時間がコンパクトですみます。改修が難しいポイントは、そもそもそのために人数を多く割いていなかったりする

のでマンニングが難しい事があるのと、少人数で開発をする事になるので、そんなチームにエキスパートがごろごろいるわけでもなく、結果として技術的負債が蓄積しやすい点でしょうか。（そのため、改修コストがどんどん上がっていく傾向にあります。）

2.5.2 RPA で隙間を埋める

RPA については自前システムだと悪手になるケースが多いです。（自分たちでシステムの表も裏も全部ハンドリングできている筈なので、RPA を使うより直接改修したほうが普通は望ましい筈）さて、RPA を使ったほうが良いケースはあるのでしょうか？

たとえば、自前システムを構築している基盤に無い機能を RPA で簡単に付け足せる場合などは、RPA を導入するメリットがありそうです。(正直、前職ではこのシナリオで少し導入を検討したことがあります。結局入れなかったのですが。)

2.5.3 RPA じゃないツールで隙間を埋める

外部データ連携などをする場合や、新しい機能を実現したい場合など (最近だと AI とか？)、RPA じゃない他のツールが刺さるケースもあるかもしれません。

2.5.4 今のシステムをそのまま諦めて使う

諦めも肝心です。でもどこかで重い腰をあげないといけないタイミングというのも……

2.5.5 人を増やしてオペレーションで解決

もうここまで来ると、既存のシステムを守るという観点よりは、新しいシステムや業務プロセス改善を実現するために人を増やす必要があるかもしれません。

2.6 自前システム（と呼べるかどうかわからないけど Excel+VBA ぐらいのもの）や、システムっぽいものが無いかも…？

小さい会社で簡単なシステムを作って運用しているケースも多いでしょう。ExcelVBA などがよく活躍するケースです。あとは、「情報システム」と言われて思い浮かぶものが無いケースもあるでしょう。

でも、IT が基幹システムのようにしっかりと、連動して動くようになっていなくても、IT を使った業務プロセスが回っているケースなども広義のシステムと言えるかもしれません。

2.6.1　今のシステムを改修して隙間を埋める

簡単なシステムや小さなマクロに分割されているほど、改修というよりは 1 つのシステムにまとめていくほうが良いかもしれません。

2.6.2　RPA で隙間を埋める、RPA じゃないツールで隙間を埋める

ここはもうツール導入の可否みたいな話で論点がまとまってしまいます。

そもそもの話として、RPA やその他のツールで隙間を埋めるにせよある程度の粒度のシステムになってからな気がします。その際に、**ハンドオペレーションが入っている箇所をシステム化し、一気に自動化**できるかもしれません。

なお、それは一般的には「DX」と言います。

改善の伸びしろがいっぱいある状態は、ある意味では良い事なのかもしれません。

2.6.3　今のシステムをそのまま諦めて使う

まあそんな判断になることもあるかもしれません。投入できるコストは有限ですし。

2.6.4　人を増やしてオペレーションで解決

これぐらいの規模感だと、「増やせるものなら増やしたい」というのが本音になりそうな。

2.7　色々と書いてきましたが…

色々なケースを挙げ、RPA が効きやすそうなケース、効きにくそうなケースを様々に検討してみましたが、いかがだったでしょうか？

ちなみに情報システムの再構築における様々な論点については、「SEC BOOKS：システム再構築を成功に導くユーザガイド　第 2 版〜ユーザとベンダで共有する再構築のリスクと対策〜」という書籍があったのですが、絶版になってしまいました。ただ、PDF は無料公開されていて、誰でも読めるようになっており、ここに様々な参考となる情報が掲載されています。

`https://www.ipa.go.jp/archive/publish/secbooks20180223.html`

RPA での改善を検討する際にも、本来的なアプローチとしては

- システムリプレースや改修するほどではないね
- RPA で改善をしようね

という検討結果を経て、その後から RPA での改善が始まるのでは？　と思いますので、こういったソースにも当たってみて、一読した上で論点整理をしておくと、少し導入がスムーズになるかもしれません。[*5]

この記事が最適な IT システム構築、ツール選定戦略の一助となれば幸いです。

[*5] 上長に説明するときとか、資料があると説明力が段違いだったりします

第 3 章

業務断捨離のすすめ

北崎 恵凡

3.1　今日から私は庶務担当

これから話す内容は架空の会社の技術部門の一組織で、本業であるエンジニアリングの傍ら、庶務担当を命ぜられ、業務断捨離を進めてきた体験談 (フィクション) です。どのように考えて行動し、苦楽を経験し、業務改善してきたかの一節を共有できれば幸いです。

3.2　はじまり

少子高齢化、働き手世代の人数減少と団塊世代の引退 (2025 年の崖問題[*1])。じわじわと差し迫ってくる不安がある日、現実のものとなる。担当者が 3 人→ 1 人に減り (2 人は円満引退・シニア再雇用で別の道へ)、技術系管理職の宿命とは言え、すべての庶務業務が自分に回ってきた。組織変更や人事異動も重なり、業務整理をせざるを得ない状況に陥った。何から始める (た) か最初に頭の中に浮かんだのは、業務のデューデリジェンス (本来は投資を行うにあたって、投資対象となる企業や投資先の価値やリスクなどを調査すること) だった。表現として「業務整理」「業務棚卸し」でもよかったが、判断軸を設定し、価値がないものに工数 (コスト) をかけてもムダだと思ったからだ。

営利企業の目的は 2 つしかない。

- 売上 (収益) 増加 (P)

[*1] 経済産業省が「DX レポート」にて提示した日本の近い将来に対する警鐘で、日本企業が DX の取り組みを十分に行わなかった場合、2025 年以降に年間で最大 12 兆円の経済損失が発生し、国際競争力を失うという課題。合わせて老朽化した IT 設備が残ると、技術的負債によりサービス競争力も低下することが懸念される。

- コスト削減 (L)

本業はインフラ設備の運用保守業務を担当し、SRE(Site Reliability Engineering)[*2]を目指して活動していますが、運用保守業務はいわゆる「コストセンター」と呼ばれ、サービスやシステムの信頼性を高める活動や付加価値を創造する活動にもあまりコストを掛けられず、日々昼夜、自動化・効率化に勤しんでいます。庶務業務もエンジニアリングの一種と捉えて、同じように取り組めないか、と考えはじめました。

3.3 何をする (した) か

- 庶務業務一覧の作成
- 当番表の作成＋ロギング・数値化
- 報告フォーマット (1 枚) の作成
- Before / After シートの作成
- セルフサービス化
- コミュニケーションコストを下げる (情報の公開、チャットボットの活用)
- 機械化・自動化
- 事務局の立ち上げ

3.4 なぜ庶務業務は属人化しやすいか

本業でないことは誰も興味が無いし、担当者に丸投げ・任せきりになり、属人化しやすいです。担当者が突然いなくなると困る潜在的な問題を抱えています。業務知識を公開 (属人化→形式知化) し、業務理解に努める仕組みが重要です。

3.4.1 庶務業務一覧の作成

庶務業務一覧を作成しました。

[*2] Google 社が提唱したサービス・システムの信頼性、安定性、拡張性、効率性を高めるためのエンジニアリングの一種で、システムのエラーを防止するために、自動化、モニタリング、テスト、リカバリなどのプロセスを導入することにより、人為的なミスを減らし、高いシステム信頼性を実現することを目指した方法論。システムの性能改善を目的としたメトリクスやデータ分析に基づく意思決定も重要な要素となっている

No.	業務名称	業務内容	責任者	担当者	業務確認書	更新日	工数(h/月)
1	業務A	- 項目A	責任者A	担当者A 担当者a	資料A	2023年4月30日	12
2	業務B	- 項目B - bはbbをXXまでに行う	責任者B	担当者b	-	2022年1月15日	2
3	業務C						
4							
5							
6							

図 3.1: 庶務業務一覧

- 採番
  - 業務に詳しい担当者間であれば「XX の件」で通じるかもしれませんが、業務内容を知らなければ説明や指示にコミュニケーションコストが発生するので、庶務業務の「No.3」の件、というようにミニマムコストで認識の齟齬が発生しないコミュニケーションが大事です。
- 業務名称
  - 誰にでも (業務を知らない人でも) 内容をある程度把握できる分かりやすい名称にします。
- 業務内容
  - 軽微な内容であればここに記載します。量が多く内容が煩雑な場合は業務確認書を作成します。
- 責任者
  - 責任者不在は担当者が困ります。相談相手も報告先もなく、業務が属人化・形骸化しやすい原因を作ります。
- 担当者
- 業務確認書
- 工数 (h/月)
  - 定量的に数値で判断できるのは重要です。工数＝コストです。業務内容に対して適切なコストかどうか (サンクコストも含めて) 判断します。

カテゴリ分け、優先順位づけ (重要度と緊急性の 2 軸マトリクスで行うことが多いですが、困り度も考慮に含めて) 行います。

3.4.2 当番表

リマインダー、チェックリスト、ロギングの機能を持ち、自動的に数値化まで行います。SRE ではトイル (Toil)[*3]と呼ばれる種類の業務で、1 人あたりのトイルに割かれる時間を 50% 未満にすることが推奨されています。

2023年5月8日	当番	担当A		
作業種別	業務	頻度	内容	実施記録
アテンド				未
確認・収集				済

図 3.2: 当番表

3.4.3 情報の公開

密室の議論は内容が不透明で属人化を生みやすいので、基本的に情報は公開するようにします。(ただし、セキュリティ上の情報機密区分に応じて内容と公開範囲は適切に設定します) 情報が公開されれば情報の方向性が PUSH 型から PULL 型になり、コミュニケーションコストが下がります。

[*3] 時間 (Time)、オペレーション (Operation)、インシデント (Incident)、負荷 (Load) の頭字語。同じことを繰り返し手動で実行する、スケールしない作業を指します。

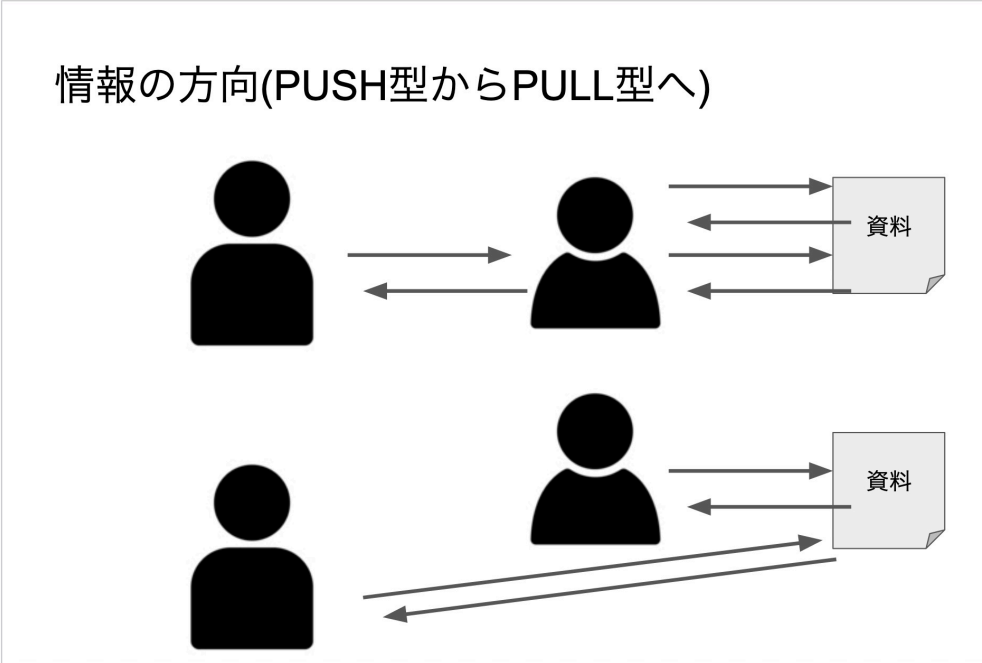

図 3.3: 情報の方向

3.4.4 業務確認書

異なる業務を同じフォーマットで見られるようにすることが重要です。

図 3.4: 業務確認書

3.4.5　報告フォーマット

庶務業務は煩雑で細々とした内容が多く、情報量が多くなりがちです。整理前と整理後を 1 枚で見えるようにすることが重要です。

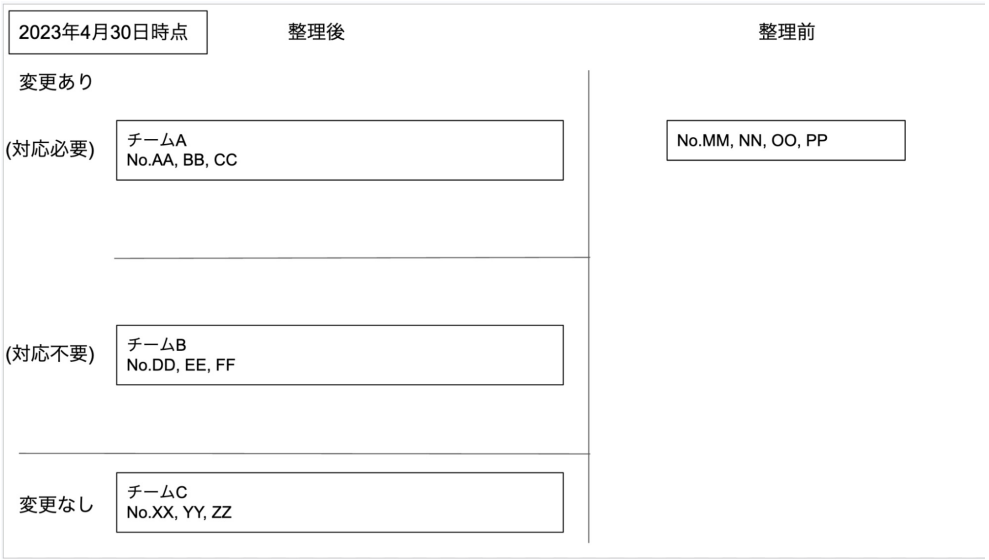

図 3.5: 報告フォーマット

3.4.6　Before / After シート

これまでの習慣や慣れた行動を変える、変えさせるのは大変です。なかなか理解は得られなくても、シンクコストを下げるために、分かりやすく伝える工夫をします。基本は「人を仲介・介在させない＝セルフサービス化」、「コミュニケーションコストは最小限に」です。

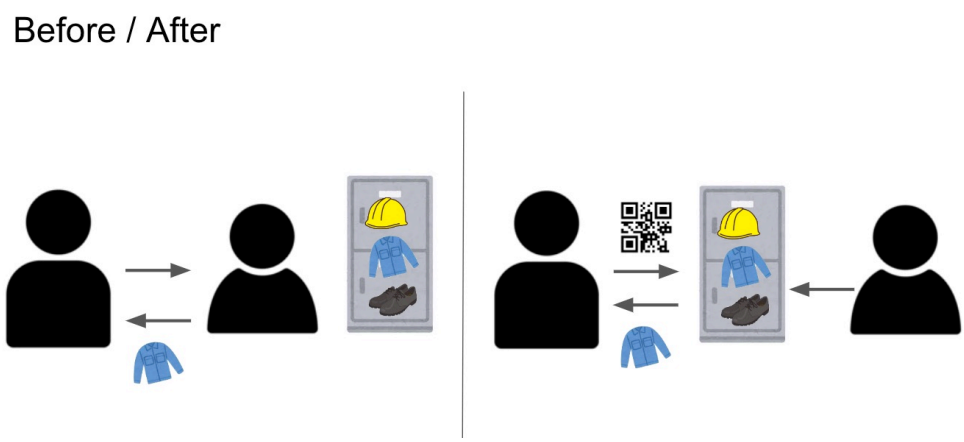

図 3.6: Before / After

3.4.7　事務局の立ち上げ

業務を前へ進めるためには、利害関係が対立した場合に備えて、取り敢えず、落とし所、最後の砦、困った時の駆け込み寺を用意することが重要です。

3.5　WHY から始めよ

ゴールデンサークル理論[*4]を利用して業務を査定していきました。「何の業務をやっているのか」「なぜ、その業務をやっているのか」「その業務のやり方はベストなのか」業務を担当している方への敬意と感謝を忘れずに、問い続けることが重要です。

[*4] 経営者やマーケターによって使用される、より深い目的や価値観を明確にするためのフレームワーク。

なぜ、その業務が必要なのか

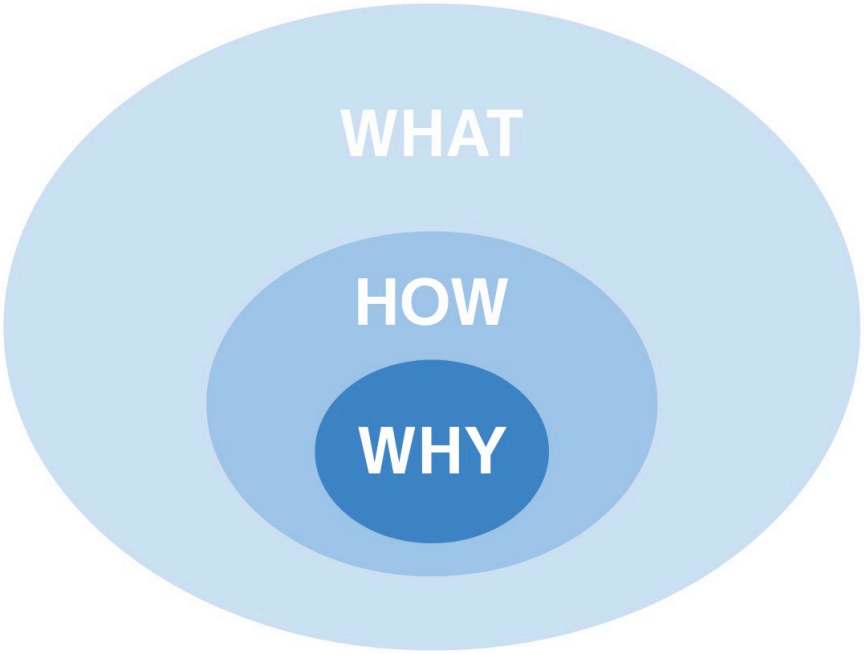

図 3.7: ゴールデンサークル

3.6 断捨離 (できたケース)

3.6.1 根本原因が既に解決していた

時間経過と共にルーティン化・マンネリ化・形骸化していた。

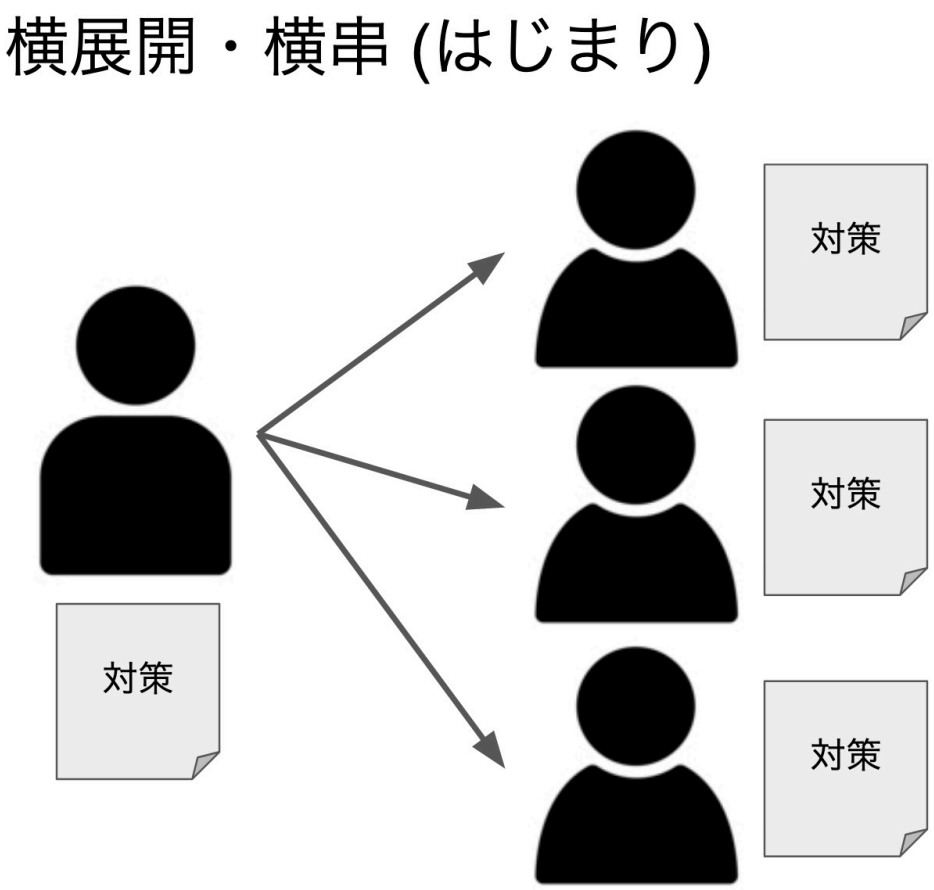

図 3.8: はじまり

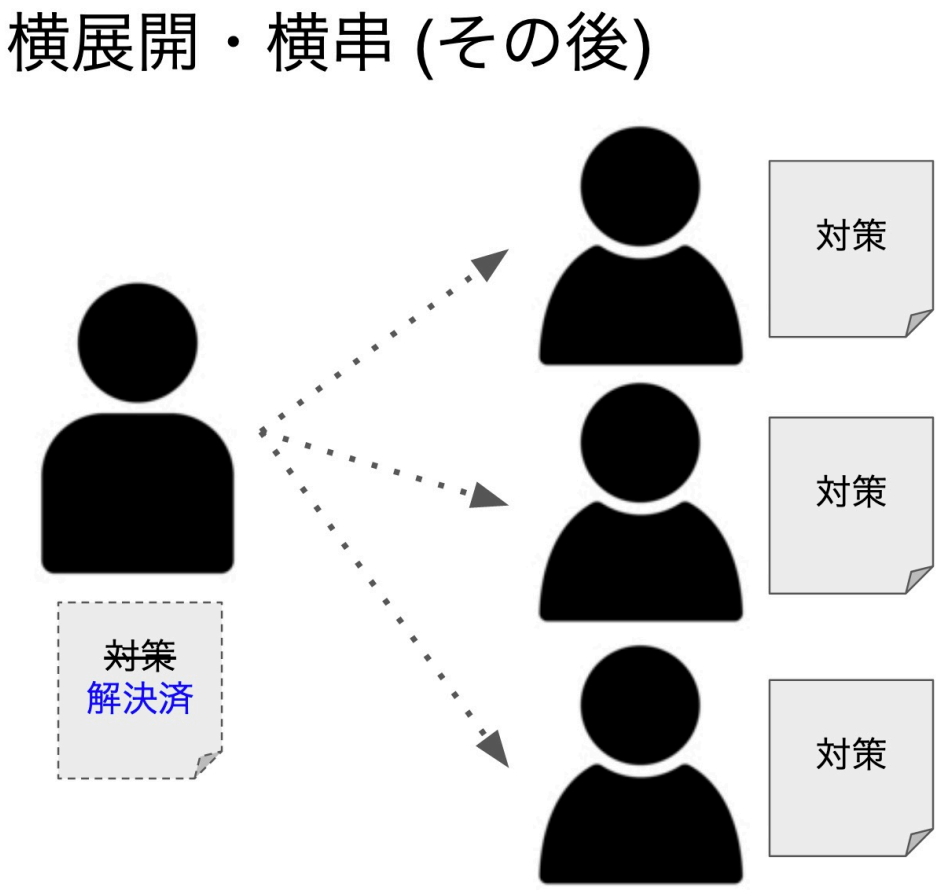

図 3.9: その後

3.6.2　他では簡易に実施、または、実施していなかった

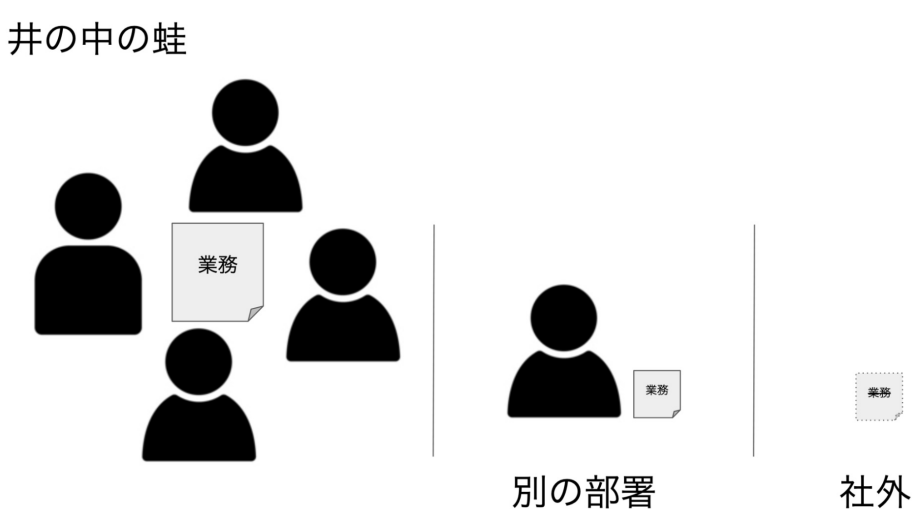

図 3.10: 井の中の蛙

3.6.3　機械化・自動化

巡回業務を無くすことができました。

・在庫確認 (スマートマットの活用)　・・本書後述

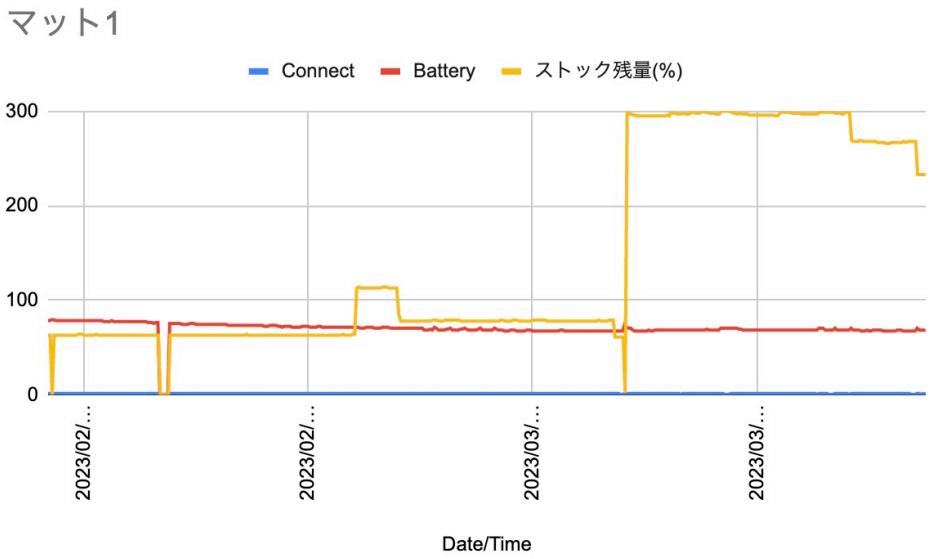

図 3.11: スマートマットライト 1

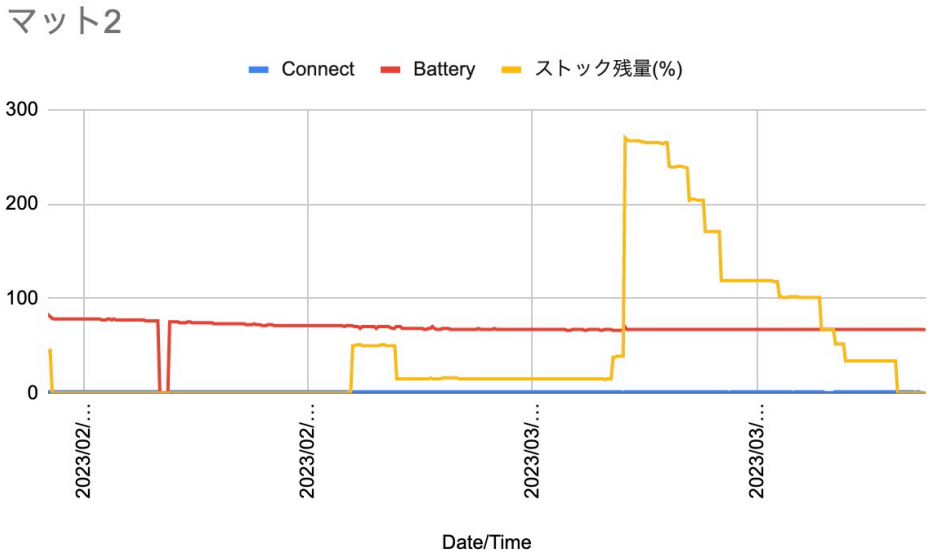

図 3.12: スマートマットライト 2

・滞在人数の可視化 (スマート AI カメラの活用)

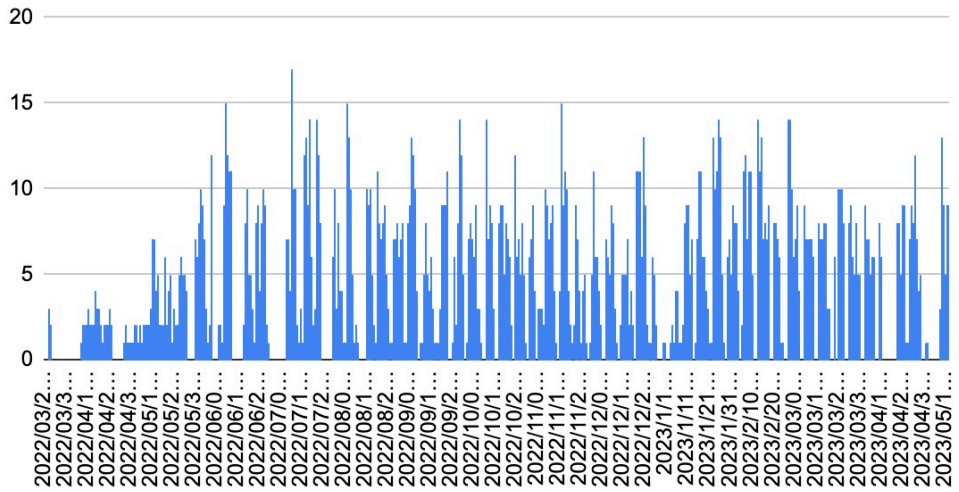

図 3.13: スマート AI カメラ 1

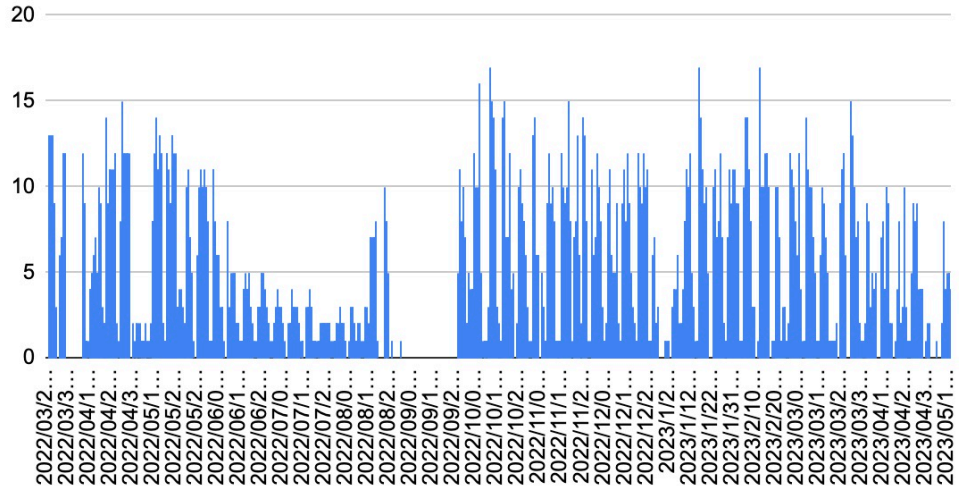

図 3.14: スマート AI カメラ 2

・職場環境の快適化 (IoT デバイスの活用)

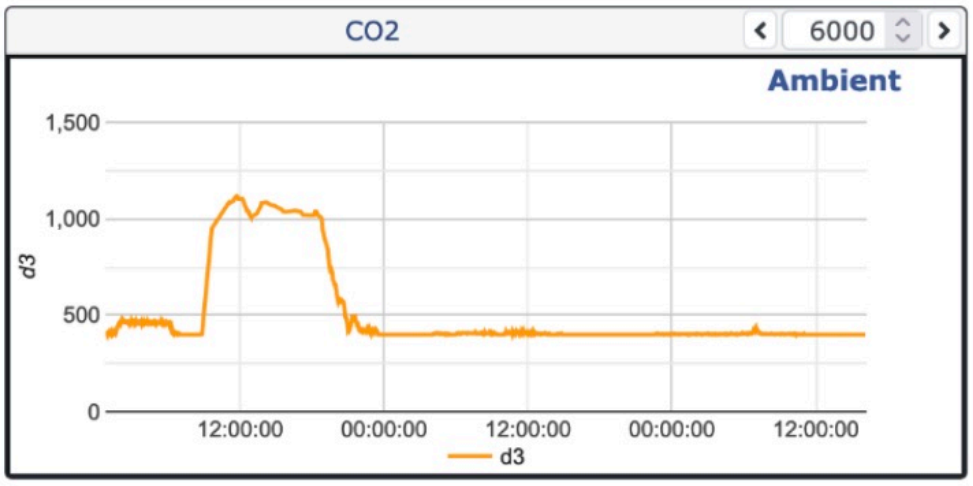

図 3.15: CO2 濃度

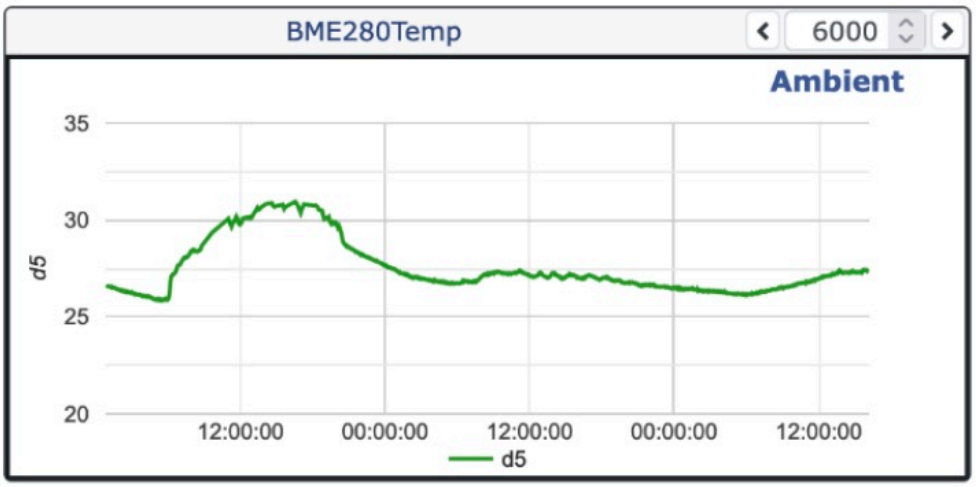

図 3.16: 気温

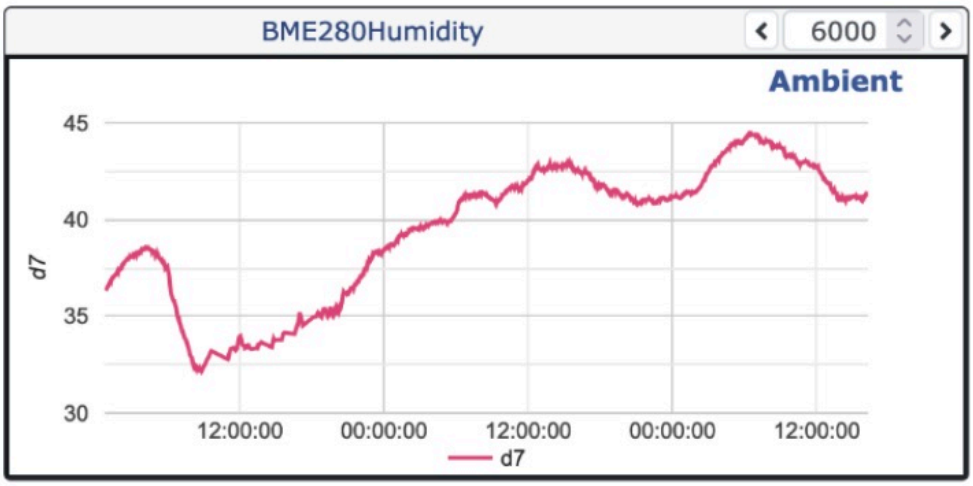

図 3.17: 湿度

3.7 断捨離 (できなかったケース)

- 現場業務
  - 物理的に人を動かす必要がある場合には効率化に限界がある。
    - (郵便物、宅配便、請求書、訪問者対応、など)
  - 頻度を減らす、まとめて対応する、などの対応方法になる。
- 法的規制
  - XX 法、施行規則、関連細則、安全衛生管理、消防法など、ルールを守る必要がある
    - 業務は勝手に止められない。頻度・回数など。
  - 用語の解釈にも注意が必要。
    -「巡視」・・人が見て回ること。ロボット化できない。
  - 政府によるアナログ規制撤廃[*5]には期待しているが、対象・候補があまりなかった。
- 事務局の廃止 (解散)

[*5] デジタル原則を踏まえたアナログ規制の見直し (https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/c43e8643-e807-41f3-b929-94fb7054377e/1420dca1/20221221_meeting_administrative_research_outline_08.pdf)

3.8 悩み (悩んだこと)

- GAS の引き継ぎ (技術スキルが必要)
  - 背伸びをせず、自分たちの身の丈にあったスキルレベルで実装が求められます。
    - 高いスキルを求めると、後任者がメンテナンスできなくなり、技術的負債を生みやすくなります。
- 心構え
  - そもそも、その業務を無くせないか
  - 面倒なこと、苦痛なこと (つらみ) を無くす、緩和できないか
  - 自分の仕事が無くなる不安がある
    - (別の仕事の割り当てを先に示す必要がある)

3.9 気づき

3.9.1 良かったこと

- 一緒に活動する仲間ができた (いろいろな人と知り合えた)
- チームワークが良くなった (団結力が向上した)
- 相談相手ができた (メンタル面の支えができた)
- 業務の中身を知ることができた
- 問題・課題を具体的に把握できた
- 集合知を生かすことができた (集合知の力を発揮できた)
- 上司/部下それぞれの視点・視野・観点を理解するようになった
  - (客観的・中立的に物事を見られるようになった)
- 事務/技術スキルが身についた
  - スプレッドシート、フォーム、GAS、など
- 技術部門なのでアナログ業務 (巡視) よりデジタル (コードベース) の方が
  - 理解しやすい、引き継ぎしやすい
- 技術で解決する欲求が高まった (活動のモチベーションを維持できた)
  - 滞在人数可視化、職場環境の IoT 化、計測 (スマートマットもその一つ)

3.9.2 反省点

- ジョブディスクリプション (職務定義書) までは踏み込めなかった
- メンバーシップ型→ジョブ型雇用の夜明けが見られなかった

- (差し迫った状況ではなく) 職場メンバーと (ふりかえりとして) 座談会をやりたかった

3.9.3 副次的効果

- ES サーベイ[*6]の向上 (責任者・担当者の明確化、ひとり IT 用務員の相談相手ができる)
- 心理的安全性の確保 (庶務業務の一覧化、見える化 (誰でも見られる場所へ公開)、数値による定量化・客観化、負担の平準化 (得手・不得手、適材適所の場合を除く))
- 担当者のローテーション (サイクル、任期、計画性が生まれた (我慢しよう、先が見えない不安から開放))

3.10 さいごに

二年間に渡る庶務業務整理がひと段落し、成功・失敗いろいろとありましたが、本書の執筆を機会にふりかえることができて良かったです。私が経験したベストプラクティスが、少しでもみなさまの「見える景色を変える」[*7]お役に立てれば幸いです。

[*6] ES(Employee Satisfaction) 従業員満足度調査。従業員エクスペリエンス、従業員エンゲージメントと表現されることもある

[*7] 敬愛する沢渡あまね氏の著書から表現を借用

第 4 章

スマートマットをハックしてさらに便利にする

北崎 恵凡

4.1　スマートマットとは

あらかじめ登録した商品の重さを定期的に計測し、指定した条件になると自動で発注するスマートデバイスです。製品としてスマートマットライト[*1]とスマートマットクラウド[*2]の 2 種類があります。スマートマットライトは Amazon の日用品 (水やペーパータオルなど、約 3,000 の対象商品) の自動注文に特化しており、管理画面が用意されています。マットの価格[*3]は安価 (セール時〜通常: 1,980〜2,980 円) で、マット 10 枚まで利用可能です。スマートマットクラウドは法人向けサービスで、棚卸自動化、入出庫管理、自動注文など、フルスペック・フルサービスの在庫管理・発注プラットフォームです。違いの詳細は製品のヘルプセンター[*4]に掲載されています。

[*1] https://service.lite.smartmat.io/
[*2] https://www.smartmat.io/
[*3] https://www.amazon.co.jp/dp/B08G8B2928
[*4] https://smartshopping.my.site.com/help/s/article/000001605

図 4.1: 戸棚のスマートマットライト

図 4.2: ペーパータオルの在庫

4.2 スマートマットライト

スマートマットライトはマット本体を購入すれば月額利用料は発生しません (買い切り)。マットは単三電池 4 本で動作し、Wi-Fi へ接続します。定期的に (デフォルトでは 1 日 4 回) 重さを測定し、AWS へデータが送信されます。(通信をキャプチャーして確認) データは専用の管理画面[*5]から、対象のマットを選択→詳細情報→消費レポートへ移動して確認することができます。

[*5] https://lite.smartmat.io/

図 4.3: スマートマットライトの管理画面


詳細情報
商品情報
Ar ハンドタオル 200枚入 5個パック
100%時の商品重量
2.07kg
注文設定
注文方法
買いどき通知
注文タイミング
20%以下
重複注文防止機能
オフ
重複注文防止機能とは？
残量情報
先週は 67.1%
昨日は 33.3% 商品を消費しました
商品残量
216%
消費ペース/日
8.1%
最終計測日時
2023/03/29 12:49
計測頻度
1日8回
消費レポートを詳しく見る

図 4.4: 詳細情報


消費レポート
先週は67.1%、昨日は33.3%商品を使いました
注文回数
2回
消費ペース/日
8.1%
使い切るまでの
平均日数
12.3日
現在の商品残量
計測誤差がある場合
216%
残量グラフ
1週間
1ヶ月
1年
%
300
270
240
210
180
150
120
90
60
30
0
注文タイミング（20%）
03/01
03/05
03/09
03/13
03/17
03/21
03/25

図 4.5: 消費レポート

4.3　スマートマットライト 1 と 2 の違い

結論から書くと、機能上の違いはほとんどありません。スマートマットライトの裏側のシール (図 4 の矢印) を剥がすとネジが 1 本現れ、プラスドライバーで外すことができます。側面はガラス板とプラスチックの境界 (図 5 の矢印) にマイナスドライバーを差し込んで開けることができます。スマートマットライト 1 は ESP-WROOM-02(ESP8266 チップ) 基板の Wi-Fi アンテナを使用していますが、スマートマットライト 2 は Wi-Fi アンテナ (図 6 の矢印) が外出しとなり、ネットワークの接続性が向上しています。

図 4.6: スマートマットライトの裏側

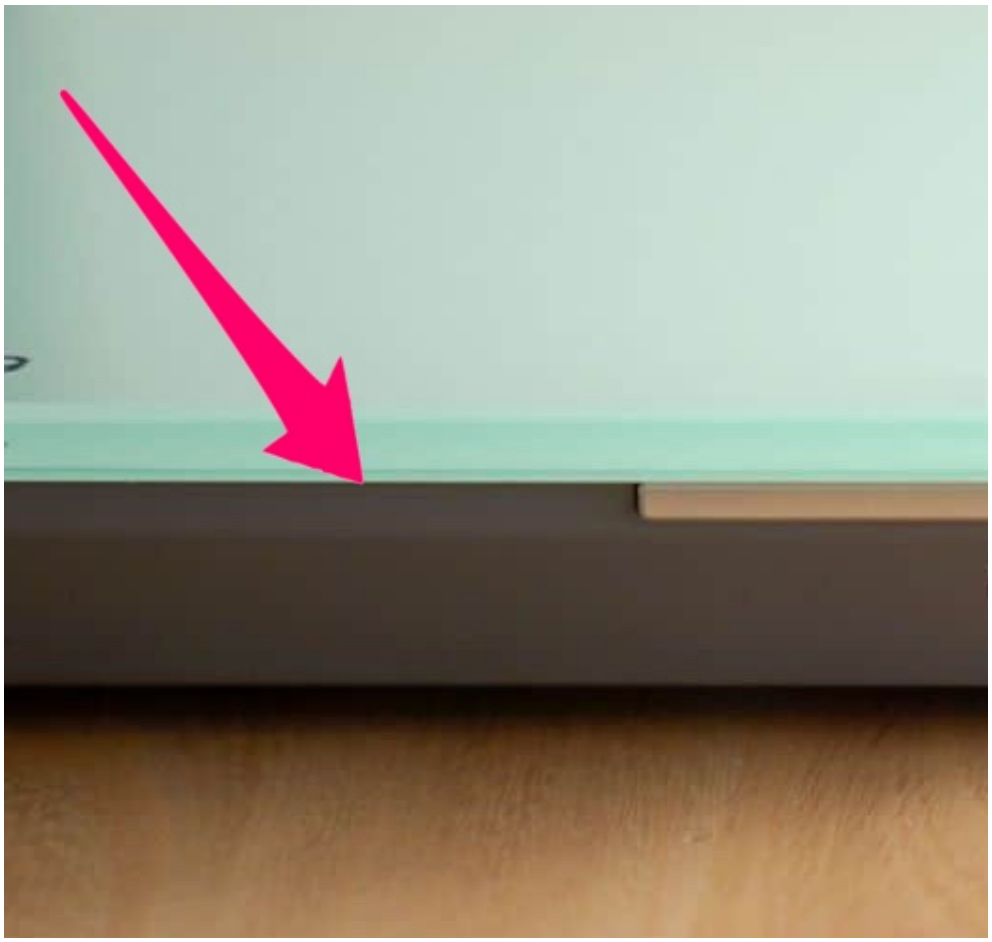

図 4.7: スマートマットライトの側面

図 4.8: 分解した中身 (バージョン 1)

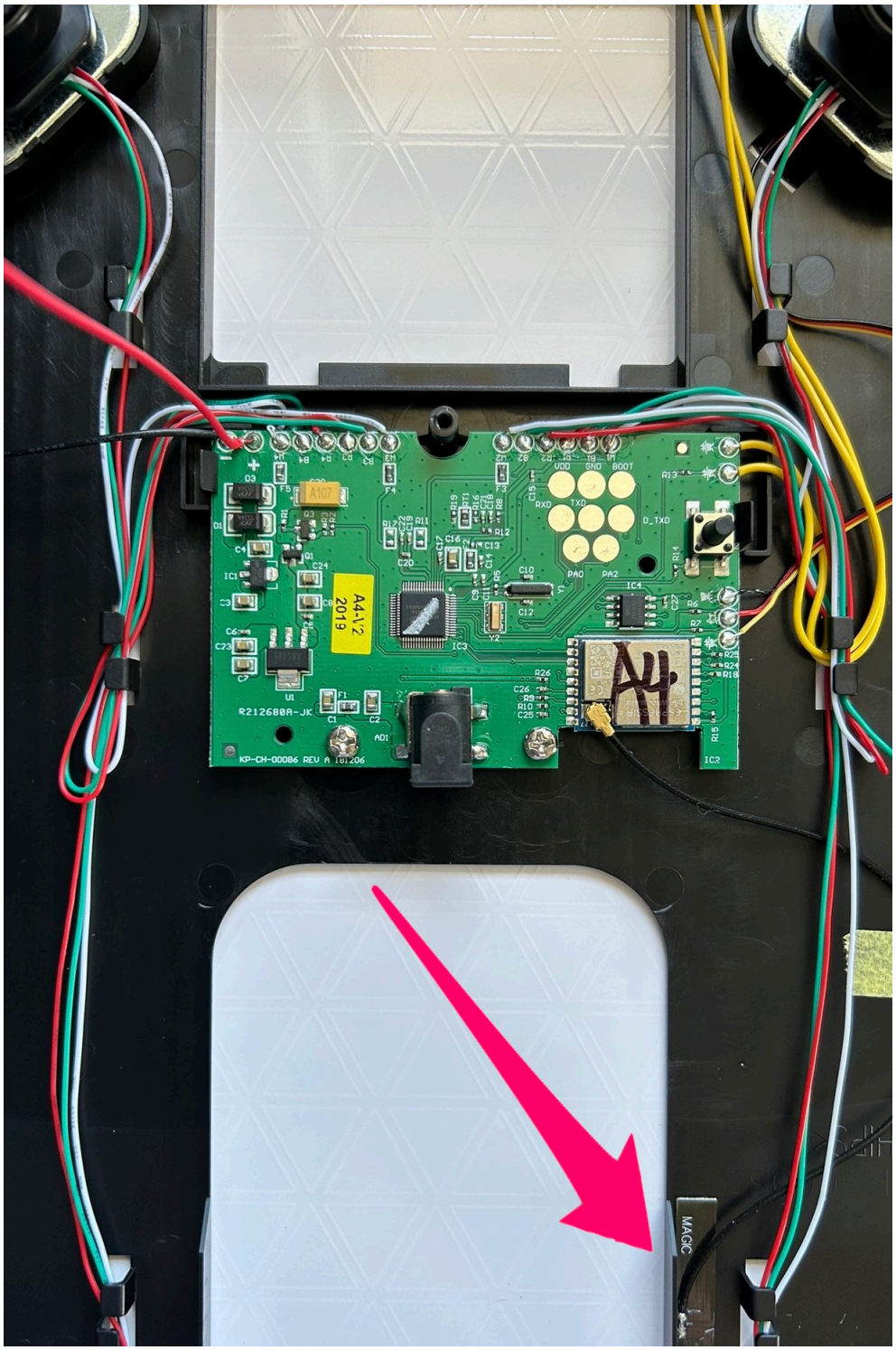

図 4.9: 分解した中身 (バージョン 2)

筆者紹介

北崎 恵凡 (きたざき あやちか)

趣味でモノづくり活動やコミュニティ「野良ハック」や技術書の執筆や月刊 I/O(工学社)、シェルスクリプトマガジン (USP 出版) などへの寄稿を行う。共著書に「Jetson Nano 超入門」(ソーテック社)、「現場で使える！ Google Apps Script レシピ集」(インプレス R&D)、「図解と実践で現場で使える Grafana」(インプレス R&D)、「はじめての Node-RED MCU Edition」(工学社) がある。コミュニティではオンライン・オフライン・ハイブリッドイベントの企画・運営・配信・アーカイブ編集/管理支援を担当。

山田 雄一 (やまだ ゆういち)

SIer での受諾開発 SE、社内 SE(内製型) を経て、現在はサポートエンジニア。技術同人誌を中心とした技術書の執筆や、たま〜に月刊 I/O(工学社) のライターだったりもする。(主に近畿圏近郊のイベントレポートを担当) 共著書に「エンジニアのための見積もり実践入門」、「エンジニアのためのオンライン生活ガイドブック」、「エンジニアのための目標設定の技術」、「エンジニアのための「カジュアル面談のトリセツ」、「積み基板を作らないための電子工作入門」などがある。(全て ditflame 名義 ㈱インプレス刊)

鎌田 誠 (かまだ まこと)

地元工場の情シスを 20 年経験した後、現在はその経験を活かして、名古屋を拠点にネットワーク系の技術営業に従事。お客様の課題解決等を行う。最近はコミュニティ活動に積極的に取り組んでおり、RPA Community ではライトニングトーク支部の主催。Babylon.js 勉強会では「Babylon.js レシピ集 vol.1」(インプレス R&D) 及び「Babylon.js レシピ集 vol.2」の執筆も一部担当。IT 知識は広く浅くをモットーに日々精進。

瀧川 大樹（たきがわ だいき）

学生時代は社会情報学を専攻、推薦アルゴリズムについて研究。強調フィルタリングとコンテンツベースフィルタリングを組み合わせた独自のアルゴリズムを考案、LAMP で実装。また、研究の傍らデータセンターのアルバイトでサーバーの保守運用業務を経験。2014 年より通信会社にてサーバー/アプリケーション保守運用業務に従事。保守運用業務に注力する傍ら、情報処理安全確保支援士の資格を取得し、システムのセキュリティー管理業務にも従事。

青木 敬樹（あおき ひろき）

学生時代は機械工学を専攻。三次元画像の特徴量抽出に関して研究。2021 年に某通信会社に入社。入社後はサーバー・インフラの保守運用業務を担当。社内の「プロアクティブな運用」の推進チームの一員として活動し、運用可能な自動化を実現するために日々勉強中。

大野 泰歩 (おおの やすほ)

学生時代は機械学習の中でも強化学習という分野について研究。RoboCup 2D Soccer に触れておりました。2020 年に通信会社に入社し、サーバ・インフラの運用保守業務に従事する傍ら、運用保守業務を自動化するためのシステム開発にも関わっておりました。

松岡 光隆 (まつおか みつたか)

XXX

新山 実奈子 (にいやま みなこ)

XXX

百合宮 桜 (ゆりみや さくら)

XXX

小崎 肇 (こさき はじめ)

XXX

澁谷 匠 (しぶや たくみ)

株式会社 ASAHI Accounting Robot 研究所 テクニカルエバンジェリスト一般社団法人中小企業個人情報セキュリティー推進協会認定 DX アドバイザースペシャリストプログラミング知識ゼロの状態から RPA/AI-OCR の技術・知識及び概念、全社展開の導入スキーム等を独学で勉強。モットーは "Make everyone's work easier. (全ての人の仕事を楽にする)"

慈英 (じえい)

XXX

現場で使える!自動化入門

2023 年 7 月 31 日　初版第 1 刷　発行
著　者　北崎 恵凡、山田 雄一、鎌田 誠、瀧川 大樹、青木 敬樹、大野 泰歩、松岡 光隆、新山 実奈子、百合宮 桜、小崎 肇、澁谷 匠、慈英