

IT のお仕事について紹介します!

Yonashiro Yuu

目次

はじめに	1
茶番	1
このドキュメントを読むときにわかってほしいこと	2
おしごとの紹介	3
主にシステム開発に関する業務	3
システムを作る業務	4
システムを動かすための業務	7
セキュリティ	8
ユーザーサポート	11
データ (主にビッグデータ) を扱う	12
R&D(Research and Develop)	13
その他のシステム利用に関する業務	13
教育	13
IT っぽくない業務	14
おわりに	15

はじめに

茶番

小規模 toB IT ベンチャーの IT エンジニアの業務

(なお、創業者・経営者は IT に詳しくないものとする)

- インフラ構築 (クラウドだったら設定・オンプレだったらデータセンター選定&契約・回線選定&契約・サーバー&ネットワーク機器調達&設置&設定)
- 運用・障害対応・障害時の客対応・障害時の社内なだめ役・障害復旧後の謝罪&客対応&社内対応
- ユーザーサポート・クレーム対応
- 技術広報・PR blog 書き
- 社内インフラ (社内のネットワーク (Wi-Fi etc...)・プリンター・電話・コーヒーマーカー・IT 詳しくない社員の PC・時計の電池交換・空調 etc、メールサーバーとか) 整備

- 得意先インフラ (上記内容 + α) 整備
- 業務システムのなやつを作る・運用する
- 社員教育 (社員には社長なども含む)
- 社内ヘルプデスク
- R&D
- マーケティング (社長がいうからやるやつ) のためのログ分析、データ分析・分析のためのデータ洗浄
- データサイエンティストみたいなこと (ビッグデータあつめ・ビッグデータの学習)
- 営業のための資料作成
- バックオフィスチームの支援
- 自社 Web サイトの更新・メンテナンス

これらすべての業務を一人で行います!!!

90% くらいは実体験なのでアレなのですが、そもそも「IT のお仕事」とかいう主語デカすぎで、就活するときも「この会社どんな業務してんの?」っていつて「開発だよ!」とかいつても、それでもなお主語デカすぎると感じる感じなので、もうちょっと細かく砕いてどんな業務があるのかを紹介していきたいと思います。

いつても、正確さを求めるとめたくそ長文になるので、ざっくり説明するくらいになります。(思い入れのある業務は長文になるかも)

あと、網羅できてないと思うのでごめんなさい。

あと、クリエイティブな業界 (ゲーム開発とか Web 制作とか) は業界の制作物によって分担がかなり変わるのであんまりあてにならないかも。(ゲーム業界については、最近ソラの桜井さんが YouTube で十分わかりやすい動画出してたのでそっちみて)

このドキュメントを読むときにわかってほしいこと

割と「業務」にフォーカスした内容を書くようにしています。が、勘違いも多く含まれていると思って読んでください。

あと、主観と妄想と駄目な視点といろんなものがカオスな感じになってる雑文だと思って読んでいただければと思います。

もし仮にみなさんの進路選択の資料にすることがあるのでしたら、理解してほしいことがあります。

一人の人が一つの業務だけを担当することもあります、複数の業務を兼任することもあります。(茶番の例なんてものはその典型)

開発をしたくてこの会社に入ったのに、開発以外のやりたくない運用もやらないといけなくなった…とか、仕様を決める仕事がやりたかったのに、なんか開発行けとかいわれてダルい…とか、そういうことを思うことがあるかもしれませんが、どんな業務についたとしても別の業務に知識が活かしやすいのも IT 業界ばさなので、あまりにもやってみて適正がないことを振られた場合以外はちょっとは頑張ってみてもいいと思います。

あと、茶番のような過激なブラック企業には注意してね。死にそうになってまでやる業務はありません。

おしごとの紹介

主にシステム開発に関する業務

システムを設計したりする業務

このセクションはシステムを開発するために「どんなシステムを作るべきか」とかを考えたりする、直接開発はしないけど、開発するにあたって必要な環境を作る部分の説明をします。

システムの要件・仕様を決める

システムを作る!といっても「どういうシステムを作るか」といういわゆる**要件**や**仕様**が決まっていないとシステムは作れません。

そのために要件や仕様を決める業務があります。

要件や仕様を決めると一口にいても、
要件だと

- 開発するシステムを使用して何をするのかを決める
- 開発するシステムが業務の一部である場合(ほとんどの場合そう)、関連する他の業務との整合性をとったり、業務全体の流れを変えたりすることを検討・決定する
- システムやそのシステムの周りの業務などが抱えているリスクはどのようなものか、軽減できるのか
- ぶっちゃけ予算あるのか
- ぶっちゃけ開発できる人をどこから引っ張ってくるのか
- ぶっちゃけ導入に向けて上司・ユーザーをどういうふうに説得するのか

仕様だと

- どのような技術(動作環境・プログラミング言語・関連するサービス etc)を使うのか
- (UIを伴う場合)どのようなユーザーインターフェイス(画面構成とかメニューとかいろいろ)にするのか
- どのようなロジックでプログラムを書かせるのか(入力から出力までの流れや、プログラムの中でやってほしいことを決める)
- (DBを使用する場合)どのようなデータを保持するようにするのか、データの整合性はどのようにするのか
- (外部に公開する場合)どの範囲を公開するのか。しないのか。

などなど、考慮しないといけない点は多数におよびます。

ただ、なぜか重要な要素なのに、実際のシステムを使う側の人(とくに業務システムの場合)が不在のところで要件がきまったり、技術について詳しい人がいないのに仕様が決まったりすることがありがちです。

技術に詳しい人が

1. お客様の業務をヒアリングして 2. 「システム化すべき場所」を特定し(不要なところはシステム化しない)
2. 要件を定義して
3. 仕様に落とし込む

という流れが最強なのですが、それぞれ専門知識が必要(とくに業務システムはその業務について深い理解が必要) ため簡単にできるものではありません。

アドバイス・コンサルティングをする

いわゆるコンサルってやつです。

前述の

1. お客様の業務をヒアリングして 2. 「システム化すべき場所」を特定し (不要なところはシステム化しない)
2. 要件を定義して
3. 仕様に落とし込む

の部分について、外部からの視点でアドバイスをします。

「とくに業務システムはその業務について深い理解が必要」と書いたのですが、業界ごとに習慣があったり、似たような問題を抱えていたりすることが多いので「〇〇業界に強いコンサル会社です!」みたいな売り出し方をする人が多いです。(つよつよコンサル会社だと業界ごとにチームを持っていたりする)

また、1~4の項目のひとつ(もしくは前後ひとつも含む)に特化してアドバイスをする会社もあります。(業務のヒアリングが得意とか、仕様を落とし込むとき際「クラウドサービス」が得意な会社にアドバイスをもらうとか)

なので、自社製品を売り込むためにコンサルとして潜り込んで自社製品の得意な部分を活かすように分析するようなこともあります。(いいところだとちゃんとやってくれますが、あくどいところだと無理やり自社製品を入れるようにアドバイスをまげてくるので注意。営業的な要素を含む場合もあるのでそれはしょうがないけど。)

ただ、ハズレをひくと大変なので業者選定は慎重に。コンサルを仕事にするなら学習は大変だけどやりがいがありそう。

チームを作る・管理する

だいたいの場合、システムを作る場合はチームで行います。チームが円滑にまわるようにしたり、開発や運用が順調にまわるようにするための調整役やマネジメント役が必要になったりします。

システム発注側とシステム受注側の両方をとりまとめたり、要件を取りまとめるためのチームでファシリテーター(チームのやりたいことがうまくできるようによしなにしてくれる人)として動いてくれる人や、システム開発チームで開発の進捗管理やメンバーのタスク管理をする人がいたりします。

システムを作る業務

開発

システムを作ります。

大多数の「システム開発会社」でのシステム開発の場合はプログラムを書いてシステムを作る場合が多いです。

ただ最近はプログラムを書かないでシステムを構築するノーコード開発プラットフォームを使用してシステムを構築するケースも多いです。(所属している会社がノーコード開発プラットフォームを開発・提供してい

るので推しです)

とくに「システムを開発することを事業としていない会社・組織」が自社の業務をシステム化したいときに外注してしまうと時間もかかるし、開発会社に指示するのも大変だし、自分たちでえいやーってやってしまいたいみたいなときにノーコードプラットフォームが使われます。(弊社の場合は「システムを開発することを事業としている会社」ですが、自社で提供しているツールを使ったほうが開発速度爆上がりなので使ってます)

このパターンは、仕様検討の段階がそのまま自分たちの業務の整理につながって、そのままシステム化・効率化できる点でも最近よくあります。(あまり好きな言葉ではないですが、DX(Digital Transformation)の文脈でのシステムの内製化の際によく用いられる手法です)

といっても、プログラムを書いてシステムを作る場合でも、ノーコードなプラットフォームを使うにしても「やりたいことを実現するためにはどうしたらいいのか」をちゃんと形にする必要があるのには違いがないので、プログラミングやノーコードプラットフォームを使用するスキルは身につける必要はあります。

あと、ノーコードプラットフォームではできないことも多々あるので、システム開発ならなんでもできるようになる場合はプログラムのスキルは必要になります。(あと、ノーコードプラットフォーム間をつなぐためにプログラム作るとかもあります)

あとは、ゲームとかスマホアプリみたいなものを作る場合や、複雑なこと、ノーコードツールでは作れないようなもの、作りたいものを作るノーコードツールが提供されていないものを作りたい場合はプログラムを書かないといけません。

また、単に「プログラミングによるシステム開発」といっても、フロントエンド・バックエンド・ツール・ライブラリといったような細分化があるので、1つのカテゴリだけで完結することはあまりありません。(特化している場合もあるし、ひとつの分野を極めるということができる場合もあります)

プログラムを書く場合でも、ツールを使ってシステムを作る場合でも、それぞれの特徴や動かす環境についての知識は必要になります。

テスト

作ったシステムが正常に動作するかテストを行います。

テストにもいろいろあり

- 自分で作ったシステム (どういう挙動をするのか理解している) が意図した挙動をしているのかテストする
- 他人が作ったシステムをテストして、不具合が無いかを調べるテスト (作った人が意図しない挙動で不具合が起きないかとか、作った人ではやらないような操作をするテスト)
- わざと悪い使い方をしてシステムが壊れないかテスト

など、見る方向によってテストするやり方はさまざまです。

システム開発の現場ではテストを自動化 (プログラムにやらせる) させるための仕組みを作ることもよくあります。

あとはテストの分類として

- 機能が正常に動作しているかを確認する機能テスト
- 性能が十分に出ているかをテストする性能テスト

という分類をすることもあります。

QA

Question と Answer ではなく、「Quality Assurance」(品質保証)のための業務です。

テストとタブるところも多いですが、QA の場合は「製品・サービスの品質」を重きにおきます。

となると「品質とは何か。品質を評価する指標は何か。」という部分がキモになってきます。ここは会社やチーム・プロジェクトによって変わってくるところです。プロジェクトの偉い人が決めた「品質」が達成できているか、を常にチェックする業務といってもいいかと思います。

機能や性能だけではなく、「安定して稼働しているか」や、「使いやすさ」など、多岐にわたります。

QA は品質をチェックするだけではなく、品質を向上させるためのアドバイスをしたりする場合もあるため、よりシステム全体や、他社のシステムの「品質」について評価できる必要があります。

リリース

システムを作ったら、ユーザーに使ってもらうための環境にシステムを展開する必要があります。その作業をリリース作業といいます。

リリース作業を専業としている会社は僕の知る限りありませんが、システムを作っている会社であれば必ずリリースの業務というのが発生します。

インターネット上や社内ネットワークのサーバー上でシステムが動作する場合は、サーバーにプログラムをインストールする作業です。スマホアプリの場合は、各アプリストア (Google Play Store や AppStore) に指定の方法で作成したアプリをアップロードする作業です。各 PC や専用の機械で動作するシステムの場合は、それぞれの機械に配布・インストールを行います。

リリースの作業も自動化されていることが多いです。

とくにサーバーでシステムが動作する場合は、複数台のサーバーにシステムをインストールする必要があるため、それぞれのサーバーに漏れなく指定のシステムを入れる必要があるため、自動化してヒューマンエラーをへらすような工夫がされていることが多いです。

コンテナとよばれる仕組みを使ったりする方法もよく使われますがこのあたりは開発するプロジェクトによってさまざまなので自分が担当するプロジェクトで使うときに勉強してみたり、興味があったら調べてみてください。

PC にセットアップする系のものは、使う人の PC を一台一台まわってインストールすることもあれば、構成管理ツールとよばれる便利機能を使用して、複数台の PC にまとめてインストールするようなケースもあります。

リリース作業はミスったら即トラブルなのでミスをしにくい工夫がされていることも多いです。

システムの保守

システムをリリースして、使ってるうちに不具合がみつかったり、機能を追加する必要があることがあります。外部のサービスと連携していたりするときにサ終 (サービス終了) したりして連携を切らないといけなとか、サービスの仕様変更やバージョンアップに対応しないといけなことも多々あります。

こういったやらないといけな修正を行うのが保守になります。

システムを外注した場合、保守費用というのが開発費とは別にかかってくることが多いです。保守費用の範囲で改修をしてもらうようなことが多いです。

ただ、保守費用が実際の不具合の発生頻度や影響範囲に対して見合っていない場合に、保守をやめてしまうこ

ともよくあります。

だいたいそういうときに限って改修しないといけないことが発生して追加の開発費用がかかるパターンも多いです。(一般的に保守費用は開発費用よりは安く設定されていることが多いです。ただ、何も改修する必要がなくても保守費用は発生するため、どちらがいいかはなんとも…)

システムを動かすための業務

このセクションでは開発したシステムを動かす・動かし続けるための業務を説明します。

Layer 0

システムを動作させるためにはだいたいの場合、何かしらのコンピュータ上で動作させます。

そのコンピュータを動かすためには

- コンピューターを設置する場所の整備 (空調や火災発生時に消火する設備があったり、そもそも建物自体が頑丈でないといけないなど)
- コンピューターが設置してある場所のセキュリティ (主に物理的セキュリティ)
- 電源の安定供給 (停電発生時、いきなり電源断にならないようにするなど)
- その他「ファシリティ」とよばれる建物のメンテなど

が必要になっていきます。これらの維持を行う業務というものが Layer 0 と書いた業務です。(この表記は一般的なものではありません。僕が勝手につけた名前です)

建物の警備員さんや、回線・電源系統の管理を行う人、設備が壊れたら修繕する人なのがこの役割になります。

一見だれでもできそうな業務ですが、この人達をおこらせるとコンピュータが止まってしまったときに何もできなくなってしまうので無下にするべきではありません。マジリスpektです。個人的にはこういうあたりまえを維持する同じような作業を繰り返す行うことが苦手すぎるのでマジリスpektです。

とくにデータセンターとよばれるサーバーを大量に設置・稼働させる建物において、電源ランプがちゃんとついているか、機械をいれている箱 (ラック) の鍵がちゃんとかかっているか、知らない人が入っていないか、空調がちゃんと動いているかを監視する業務に従事することがあります。(特にサーバー運用会社の一番最初の業務とか)

これに耐えられる人は本当に尊敬できますが、会社によってはこれだけをさせるような会社もあるので適正がなかったり、スキルアップする要素がかなり限られてくる (= 給料上がらないことも多い) のでこういう会社に就職した場合は、キャリアパスをちゃんと考えるようにしてください。(こういう同じことをずーっとやっていたいという人にとっては天国かもしれません)

サーバー・ネットワーク機器の設置・セットアップ

システムを動かすためにはコンピュータが必要と前述しました。そのコンピュータ上でシステムが動くようにセットアップをする業務がこちらになります。

OS 入れたり、システムを動作させるのに必要なソフトを入れたり設定をしたり、ネットワークに接続したりする業務です。

また、昨今のコンピュータはほぼネットワークに接続して使用することが多いかと思います。その際には

ネットワーク機器の設定が必要になります。

ハブ・ルーター・スイッチ・ファイヤーウォール・セキュリティアプライアンスなどといったネットワーク機器の設定を行う仕事になります。

クラウド環境ではないいわゆる「オンプレミス環境」においては物理的なハードウェアをデータセンターやサーバー室のラックに設置をしたりするところや、UPS(停電とか起きたときに使用する短時間の予備電源みたいなやつ)の設置もあります。

こいつら異様に重たいので筋力が必要な業務だったりもします。

サーバー・ネットワーク機器の運用 (サーバーを使う側)

設置したサーバーやネットワーク機器が正常に動作しつづけるための業務です。

サーバーやネットワーク機器の状態を監視するツールがあるのでそれをモニタリングしたりすることが多いです。また、何かトラブルたら通知をするシステムが導入されていることも多いです。

通知の方法としては * 部屋においてある警告灯がくるくるしはじめる * なんか大きな音になる * スマホにメールがガンガンとんでくる * 電話がかかってくる みたいなのがあります。

これを業務中だったり、運用形態によっては 24 時間体制で組まれていることがあります。昼勤の仕事がかわって、ぐーすか寝てる午前 4 時にメールがガンガンきたりする。電話が鳴りまくるみたいなことも会社によってはあります。

そのあたりは…うん。がんばってください。(ちゃんとした会社の場合はメールが飛んできたとしても夜勤の人が対応してくれるはず)

サーバー・ネットワーク機器の保守 (HW 側)

設置したサーバーやネットワーク機器があるということは、多分どこからか買ってきているはず。サーバーやネットワーク機器が壊れたときに、保守契約を結んでいる場合は、販売元・もしくは保守の会社に連絡して機械を直してもらったり、正常な機械を送ってもらって交換することがあります。

こちらも 24 時間 365 日対応してくれる保守契約をしている場合は対応してくれます。平日の 9 時から 17 時までとか、契約によって対応してくれる時間はさまざまです。

…と書いたのですが、こういう保守業務があるという説明でした。

監視

システムの監視です。

だいたい前述してますが、ハード的な監視、施設的な監視、サーバーリソース (CPU 使用率やディスク使用率) の監視、サーバー上で動作しているシステムが正常に動作しているかの監視、セキュリティインシデントが発生していないかの監視など、いろんな監視項目があります。

監視業務担当の人は「異常が起きた場合に異常が起きた」と通知をするのが業務である場合が多いです。

通知してあとは運用の人ががんばってくれる…みたいなケースも多いです。(運用と監視を兼務している場合も多いです)

セキュリティ

このセクションではセキュリティに関する業務を紹介します。

セキュリティ設計

セキュリティの設計は、基本的に下記の内容を検証し、設計します。

- 何を守るべきか
- 守るべきものにはどんなリスク・インパクトがあるか
- 守るべきものが持っているリスクは潰すべきか
- リスクを潰さない場合は、残ったリスクがインシデントになったときにどうするか

そもそも、守るべきものがない場合はセキュリティなんてものは考える必要はありません。無人島に警備員は必要ない(海上保安庁的なものは必要かもしれないけど)し、金庫にいれておくものがなければ金庫は必要ありません。

なので、セキュリティ設計のはじまりは、何を守るべきかを洗い出すことになります。

システムなので、「顧客情報」や「データベース」かもしれないし、「稼働させているシステム」自体かもしれません。昨今はサイバー攻撃も多いので、システムを稼働させているハードウェアも対象になります。もっというと、ハードウェアを物理的に壊したりパクったりする輩がいるかもしれません。ハードウェアを設置している建物が天災やミサイル攻撃などで壊されてしまうかもしれません。実際に、大地震で建物ごと駄目になってシステムが駄目になったケースもあります。

「守るべきもの」の洗い出しを行ったら、次はどんなリスクがあるかを検証します。

- 前述した、顧客情報やデータベースは何かしらの理由で漏洩する可能性があります。
- システムを稼働させているマシンが乗っ取られたら踏み台にされて別のマシンを攻撃する可能性もあります。
- ハードウェアの盗難や破壊工作もあります。
- 建物の耐震性が駄目すぎて地震がきたら終わってしまったたり、ミサイルがとんでくる可能性も否定はできません。
- 最近では「経済安全保障」という考え方もあり、特定の国や地域に依存することがリスクになることもあります。

こういうふうに守るべきものがどういうリスクがあるのかを丁寧に洗い出していきます。

リスクの洗い出しが終わったら、そのリスクにどう対応していくかを検討します。

- 顧客情報の漏洩やデータベースの漏洩はだいたいの場合、システムの不具合が原因になりそうならシステムの不具合が起きにくいようなシステムの作り方をする
- 内部犯行による情報漏洩もよくあるので、データへのアクセスができる人を制限したり、アクセスの記録がきちんと残るように仕組みをつくる
- マシンの乗っ取り対策として、アンチウイルスソフトや、サイバー攻撃を検知できるツールを導入する
- 物理的なハードウェア攻撃や盗難を防ぐために、厳重に管理された空間にハードウェアを設置したり、警備員を配置したり、監視カメラできちんと記録をとる
- 建物の耐震性が駄目な場合は、耐震性を強化したり、お引越しを検討する。
- ミサイルみたいなのはもうあきらめる

などといった感じです。

ミサイルみたいな例はあきらめてしまって、こういう状態になったらシステムを捨ててしまうという判断もありますし、システムを冗長化しておいて、ものすごく距離の離れた複数の建物にバックアップをしておくというような対応も考えられます。

このようにリスクを潰せるだけ潰し、潰せないリスクがインシデントに変わったときにどうするかみたいな観点からセキュリティを設計していきます。

その中の一部に「ファイヤーウォールの設定」や「セキュリティアプライアンスの設定」のようなものが入ってきます。

セキュリティインシデント対応

いくらリスクを想定していたり、リスクを潰していても、だいたいの場合漏れはあるし、新たなリスクはどんどん発生していきます。

そしてそのリスクをつかれて守るべきものが破壊されてしまうことはあります。

そうなったときの対応がセキュリティインシデント対応になります。

だいたいの場合下記のようなことの調査をします。

- ・ 守るべきものが破壊されたことでどれくらいの影響・被害がでているのか
- ・ どのリスクをつかれて破壊されたのか。
- ・ 誰が犯人なのか (場合によっては刑事告訴することもありえるので)
- ・ 関係機関への報告
- ・ 破壊されたものは復旧できるのか。できない場合は代わりにどうするのか
- ・ 被害が出ている場合に補償をするのか
- ・ 次に同じようなインシデントが発生しないようにどうしたらいいのか

その他いろいろありますが、インシデントが発生してしまった場合の対応業務は結構たいへんです。

こちらは調査を専門に行っている会社もあります。

セキュリティ監査

インシデントが発生していないからといって、セキュリティが完璧というわけではありません。未知のリスクが発生していることも多々あります。

ですので、定期的にセキュリティの監査を行い、想定しているリスクに対応できているか、新たなリスクがある場合はそれにどう対応するのかを定期的に監査する必要があります。

脆弱性検査

こちらはどっちかというシステムよりの話ですが、作成したシステムに想定していない脆弱性が無いか、システムを動作させている環境に脆弱性がないか、ネットワーク設定に狙われるポイントがないかを検査する業務です。

いわゆるホワイトハッカー的な業務でセキュリティ業界的には花形業務ですが、システムの穴を探したり環境の脆弱性をつくような攻撃をしたり、最新の脆弱性情報を収集したり、ネットワークがどのように構成されているかを想像したりと、ものすごく幅の広い知識と経験が求められます。

セキュリティスキャナーを動かしてわかるようなところは誰でも検査できます。しかし、知らない人が作ったアプリやシステムの脆弱性を探すのは、開発者の視点が必要になってくるので、自然とプログラミングの知

識が必要になります。また、サーバーの設定のよくある設定漏れについても知っておく必要があるので、システムの開発・運用全般の知識が求められます。

いっぱい勉強しないとこの仕事はできませんよ。っと。(ただしこの仕事できたら多分給料めっちゃいいから頑張る価値はあると思います)

ユーザーサポート

このセクションはシステムを使用する人のサポートをする業務の説明です。

コールセンター・チャットサポート

ユーザーがわからないことがあった場合に、電話で問い合わせをうけたり、チャットでサポートしたりする業務です。

なんとなくイメージできると思いますが、意味のわからない質問がきたり、関係のない質問がきたりしてちゃんと対応する必要があります。

当然ですが、質問を受けるということは、問題を解決する必要があるので、問題を切り分ける能力・本当に聞きたいことを聞き出す能力・解決方法を調べる能力など、エスパーかな?みたいな能力が求められることが多いです。

「よくある質問と答え」みたいなのが準備されていてそれにマッチするような質問であればラッキーですが、そうでない場合は適切に調査し答える必要があります。

この手のコールセンターやチャットサポートを長らく運営しているところには過去にあった質問とその回答がデータとして蓄積されているので、それを見て回答する。よくわからなかったら分かる人に対応をぶん投げる(エスカレーションする)という運用をしているところが多いです。

エスカレーションを受けるくらいになってくると給料はよくなるんだと思います(想像)

フィールドエンジニア

フィールドエンジニアは、困っている人のところにかけつけて対応する業務です。

だいたいハードウェア的な故障とか、システムのエラーをみつけて直すのが仕事になります。

「困っている人のところにかけつける」というのが一番大きくて、社内で相談したりとかそういうのがやりづらい環境で対応することになるので、その場での適切な判断能力や自力での問題解決能力が求められます。(もちろん社内に電話して相談するってのは可能です。)

移動することと、機材を運搬する必要があるので、体力が求められる業務になります。あと、運転免許が必要になりがちかも。(小規模都市とかだと多分必要)

ヘルプデスク

コールセンター的なものや、フィールドエンジニア的なものを社内専用を持っている部署です。

社内専用とはいいつつも、その会社に雇用されているわけではなく、派遣である場合もよくあります。

ユーザーサポート全般の知識が必要になります。(雑)

データ (主にビッグデータ) を扱う

このトピックでは、データを扱う業務を紹介します。

データを活用するための基盤を作る

少ないデータであれば Excel で分析できるものもありますが、100 万件、1000 万件とかから数 PB クラスまで大量のデータを活用しようとした場合は専用の仕組みを使用する必要があります。

その専用の仕組みを動かすための基盤を構築する業務です。

クラウドサービスでででん!と展開したり、専用のハードウェアとソフトウェアを組み合わせたもの、汎用ハードウェアと専用のソフトを組み合わせで構築したりします。

普通のサービスの構成と違って、汎用ハードウェアを使用する場合はストレージの構成方法にコツがあったり、クラウドサービスだと、謎技術 (AWS Snowball とかで検索してね) を使用して大量のデータをクラウド上に移動したりと、特殊な構築をすることが多いです。

データを整備する

地味けどものすごく重要な業務です。

たとえば「名寄せ」とよばれる業務があります。

一方のデータでは「よなしろう」と登録されていて、一方のデータでは「與那城雄」と登録されていて、さらにもう一方のデータでは「與那城雄」(スペースが入っている) データの集まりがあったとき、人がパッと見たら同一人物かも?って思うけど、コンピューター側から見たら異なる文字列として判定されてしまいます。

これを別の ID を用意して同一人物だよ!ってというような集約をやったりする業務です。

つぎに「データ入力」の業務があります。紙の申込書をデータベースに入れるという業務です。データベースにいれてはじめてシステムから呼び出せるため、この作業は重要です。

過去にやらかしまくった例 (やらかしたらヤバそうな日本年金機構とかもやらかしている) もあるので、責任重大なはずなんだけど、(主に待遇面で) 軽視されがちな業務。(この辺は改善されてほしい…)

さいごに、「AI に学習させる」という業務の紹介です。

AI といっても、最初から答えを出せるわけではなく、大量の学習を行った結果、いろいろな答えを出すことが可能になります。「大量の学習」といっても、何が正解で何が間違っているかを判断できないうちは人間が教えてあげないといけません。その学習をさせる業務というのがあったりします。

それ専門に行っている会社もあります。

こういった丁寧なデータ入力や AI の学習を業務としてやっている人の支援があって、システムは大量のデータを適切に扱うことが可能になります。

データを活用する

データは溜まっているだけでは価値はありません (そもそもデータが溜まってないと何もできないので無価値ではないですが)。データを分析したり加工したりして「情報」として形にして、業務のために使用することではじめて (会社の求める) 「価値」を生み出します。

そのデータを活用するために「情報」に変換する業務があります。

いわゆる「データサイエンティスト」とよばれる職種はそのプロフェッショナルです。

昔から大量のデータから経営判断・マーケティング・その他業務の改善などのために情報を取り出すツールが提供されています。これらのツールを活用するのもデータを活用する職種になるかと思います。

R&D(Research and Develop)

日本語でわかりやすくいうと「研究開発職」です。

新たな技術が自社で使えないか、新たな技術を生み出していく職種です。

業務の内容はシンプルですが、やることは幅広いので、めたくそスキルが求められます。(会社の業務や技術セットによって研究する対象も変わってくるのであまり具体的な説明ができなくてすみません)

その他のシステム利用に関する業務

最近出てきたシステム利用に関する業務です。

RPA の学習

RPA(Robotic Process Automation) というツールがあります。これは何かというと、PC の画面操作をプログラムで行うものです。

ここにどんな内容が入っていたら、別の場所に特定の内容を入力して、指定のボタンをクリックする!

みたいなものをガンガン学習させることで、人間が行っていた PC 操作をプログラムにさせてしまおう。というものです。

その処理の学習 (プログラミングみたいな感じ) をさせる業務があつたりします。

RPA を導入することで人間より高速に (指示さえ間違っていれば) 間違いなく処理を休むことなく続けることができるので、導入をする会社も増えてきています。

教育

ここでは教育にかかわる業務です。教育大切。

■社内教育講師 自社で使用するツールの使い方を教育したり、セキュリティについての教育をしたり、、、コンプライアンスの教育をしたり…します。

思っているより社内教育しないといけないことも多いです。

専業でやるには教育の機会が少ないので、だいたい自分の普段の業務について教育したりと兼業でやることが多い気がします。

顧問

教育のセクションに入れていいのか微妙なのですが、普段は別の会社に所属していたり、フリーで活動しているが、たまに会社にやってきて相談に乗る業務です。

相談に乗るということはものすごい知識や技術をもってないと、顧問解除されてしまうのですごい人しか基本的にはなれません。(すごい人にしか頼まないけど)

研修講師

企業や団体の研修の講師です。特定の製品についての研修を担当したり、特定の分野についての研修を担当したり、所属している企業や団体(フリーでやっている場合は自分の強み)の強みについて講義を行うことが多いです。

また、会社によっては用意されたカリキュラムの講師を受け持つこともあります。

講義をするってことはそうかんたんなことではなく、少なくとも講義をする範囲の質問についてはサクッと答えられる必要はあるし、関連する話もしゃべれないといけないし、何より受講者にちゃんと伝えるスキルがないとやってられません。

難しい仕事ではありますが、ハマる人にはハマる仕事です。(逆にハマらない人にはまったくハマりません)

学校での講義を行う人・補佐する人

専門学校の講師がわかりやすいですが、それ以外にも IT に関する講義を学校で行うケースはあります。

小学校での IT 関連の授業には「ICT 支援員」とよばれる人が講義の補佐に入ったり、たまにメインをはったりします。

講師にフォーカスすると

- カリキュラムの準備
- カリキュラムに準拠した講義の準備
- 講義
- 評価 (および評価のための試験や課題の準備・採点など)
- クラスの環境の整備
- etc...

などまあまあやることいっぱいです。

たまに、「現場でやってられなくなったから講師になってるんだろ」とか言われることがある(どっちかっていうと学校の講師よりは研修講師のほうがいわれがち)のですが、現場には現場の辛さがあるし、講師には講師の辛さがあるのでそんなこというのはやめてください。

あと、コピペになりますが、講義をするってことはそうかんたんなことではなく、少なくとも講義をする範囲の質問についてはサクッと答えられる必要はあるし、関連する話もしゃべれないといけないし、何より受講者にちゃんと伝えるスキルがないとやってられません。

難しい仕事ではありますが、ハマる人にはハマる仕事です。(逆にハマらない人にはまったくハマりません)

IT っぽくない業務

このセクションは雑に書いてます。すみません。

営業・マーケティング

技術をお金にかえてくれる部署です。喧嘩ばかりしてはいけません。

広報

製品や会社の広報活動もありますが、最近では「技術広報」というのもあり、自社の技術力を発信していく広報もあります。

また、「DevRel(Developer Relation)」という自社の技術を使ってくれる人との接点を作るようなチームもあります。その中で「エバンジェリスト (伝道師)」とよばれる肩書を持った人 (最近だと自社以外の外部の人にこの肩書をつけるケースも多々あり) や、「Developer Advocate」「Cloud Advocate」とか対外的に自社の技術について発信するだけではなくコミュニケーションをとってユーザーとのつながりを大切にするような活動をしている業務をする人も多いです。(この辺は会社や所属コミュニティによって肩書がいろいろあるので…)

発信するということは少なくとも発信する内容については責任を持たないといけないし、会社・製品・コミュニティの顔となる人なので、まあ、いろいろ大変です。

バックオフィス

いわゆる総務・人事・経理とか、会社の建物の維持をしてくれる業務です。

この人達がいるおかげで給料がちゃんと振り込まれたり、会社の中で快適に仕事ができるのです。感謝しましょう。

おわりに

全然書き足りないのですが、時間がないのでこの辺で。誰かの参考になったらラッキーくらいに思ってます。

おまけ

この内容を書こうとおもったきっかけは、アドベントカレンダーに書くネタなにか無いかな…と思ったときに、過去のアレな仕事 (茶番参照) の紹介でもするか〜って悪ノリで思ったところからスタートしました。

それだけではつまらないので、もうちょっと読み応えのある文書が書けたらなあ…と思い、気づいたらこんなになっていました。

本当は雑に書いている部分 (IT ばくない仕事) のところの営業やマーケの部分も細かく書く予定だったんですが、時間がないのとそんなに知識がなかったので無理でした。すみません。