

【NCPA 認定 ノーコード基礎講座】 動画文字起こし版テキスト

まず、大まかな流れです。1つ目は、ノーコードとはなにか。そもそもノーコードとは何を指すのか、どういう発展を遂げてきたのか、という歴史と今の状況についてお話をします。

その後、ではなぜ今ノーコードが使われているか、使われ始めたのか、というところをお話します。

3つ目、ノーコードの種類とツール。実際にどういうサービスがあってどういうふうに使われているかについてお話をします。

最後、実際に使うにあたって、ノーコードのメリット・デメリットがありますので、そこも細かくお話をしたいと思います。

まず、ノーコードとはなにか。ノーコードという名前・単語で言うと、いろんな会社やいろんな所でノーコードという名前が使われていると思います。ただそれらは、雑誌やメディア、テレビだったり、新聞だったり、各ツールを開発している企業によって、その定義がかなり違います。かなり違いますので我々がこのコースを作る時に「どういうものをノーコードと呼ぶか」その定義を決めたので、まず先にお伝えします。

まず1つ目は、プログラミングが不要である。プログラミングをしなくても使えるツールであるということですね。

2つ目、インターネット上で使用できる。これは主にブラウザ上で使用ができるということを目指します。

裏側の管理であったり、各ユーザーがインターネットにつながっていれば利用ができる。インストールをしなくてもいいという定義になります。(なお、ノーコードツールの中には Web アプリケーションのみならず、ネイティブアプリケーションを開発できるものもあります。)

もう一つが、機能を拡張できる。他のクラウドサービスによくある経費精算や勤怠管理などは、ある程度機能が決まっていて、それ以上の拡張ができないものが多い。ただ、ノーコードは機能拡張できる事が一つ大きなメリットになります。

自分たちでコードを書く場合や、どこかの会社や個人が作ったものを取り込んで機能を拡張させるというツールもあります。

この3つをノーコードの定義として話を進めていきたいと思っています。

ノーコードの発展の歴史を簡単に振り返りたいと思います。世界で初めてノーコードツールを目指したソフトウェアは何でしょうか？少しお考え頂ければと思います。

この場合、ノーコードというにはインターネットに繋がっていなかったり、インストールが必要だったので完全なノーコードとは言えないんですが初めて目指したツールで世界で、一番最初のもは何かというと、VISICALC という今で言う Excel のような表計算のソフトになります。

これは 1979 年に発売された世界初のパーソナルコンピューターである Apple 社のパーソナルコンピューターに表計算ソフトとして使われてきました。

この時のパソコンというと、まだクリックして画像を押してとか、アイコンクリックとかがないので、全部コマンドラインでしたね。手で文字を打って命令文を実行するっていうことがほとんどでコンピューター自体も、まだ個人のおもちゃというものになっていて、手で文字を打って命令文を実行するっていうことがほとんどでコンピューター自体も、まだ個人のおもちゃというものになっていて、これを機にビジネスで使っていこうという風になってきています。

実際、表計算なので計算式に数字を入れると、勝手に計算が出てきて合計が出たり、税金の計算などができるソフトウェアになります。

もちろん、インターネットには繋がっていないし、機能拡張もできません、インストールも必要ですが、この後のソフトウェアと大きく違う点は、例えば家計簿であったり、同じ計算をするものでも何か特定の用途に繋がっているものではなく、汎用的に計算ができる今の Excel もそうです。いろんな業務で使えると思うのですが、そういう形で表計算ソフトとして使われることを想定してるソフトウェアというところが、これが初めてになります。

1970 年代から、どのようにすれば一般の人が、コンピューターの恩恵やソフトウェアの恩恵を受けられるかというのを考えて、ノーコードに近い概念のようなものを作って、発展をさせている。ちなみにこの 6 年後、マイクロソフトが Windows を出して、Excel を売り出したということになります。

ノーコードで発展の歴史というのが主に大きく 4 つあります。

ノーコード第 0 世代: サイトが簡単に作れるソフトウェアです。

その後、第 1 世代: 記事を作ることができるサイトの管理サービス。

細かく後ほどお話しますが、ノーコード第 2 世代と呼ばれるのがサイトの管理以外にも様々な機能を持つサービスです。

最後に、ノーコード第 3 世代: 汎用的用途で利用できるサービス。

この 4 世代で分けて話をできればと思います。

ノーコード第 0 世代の一例。サイトが簡単に作れるソフトウェアとしていろいろなホームページを作るツールが出たのですが、日本だとホームページビルダーがよく使われていたと思います。

左側の画面のように、大きく中央に実際の画面のデザインですね。どういうふうに表示されるかというのがあって、それぞれどういう画像をどこに配置するかは、このツールでできて、そして、出来上がったファイルをサーバーにアップロードすると、ウェブサイトとして表示ができる、というようなものになります。

HTML のコーディングを覚えないとウェブサイトが作れなかった時代に、ノーコードでウェブサイトを作れたことが大きな特徴になります。これもインストール型なので厳密に言うとノーコードではないんですが、ここもノーコードの世界を目指したツールの一つなのかなと思います。

いま受講頂いている方でも、ホームページビルダーを使用されたことがある方もいると思うのですが、実際はウェブサイトだけではなくて、裏側にデータベースを持って、クリックした内容によって表示が変わる等さまざまな機能があります。

その後よく皆さんも名前は聞いたことがあると思うのですが、WordPress というのを第 1 世代としています。これはホームページビルダーと違って記事を作ることができる、というところが大きいです。

実際はホームページビルダーと同時期にリリースされているのですが、ホームページビルダーは、パソコンにインストールして作っていたもの、WordPress はサーバーにインストールするので、インターネット経由で編集ツールを表示して、その中で文字を書いたり、記事を書いて保存をするとサーバーに保存されて、それが表示されます。

あと大きいのが、それまでは 1 枚 1 枚ページを作っていたものが、この WordPress だと同レイアウトでたくさんのページができる。記事の管理も、もちろんノーコードでできるので 文字の変更なども、HTML を知らなくてもできる。もちろん記事も新しく作成できる。

今現在も WordPress はかなり使われてます。世界中のウェブサイトの半分以上が WordPress で出来ているとも言われています。ただし今もノーコードとしてカウントされていますが、第 1 世代として、ここでは説明をしています。

第2世代として、サイト以外のさまざまな機能を持つサービスとして、Shopify という EC サイトを作成するサービスをご紹介します。第1世代・第2世代はウェブサイトを作るということがメインでしたが、EC でいえばカートの機能。商品を買う際に一つのところにまとめておく。購入前の商品の一覧を保存することができる場所。もう一つは、決済ですね。実際にクレジットカードであったり、銀行振込みで、お金のやり取りができる、ということが大きな違いだと思っています。

実際この Shopify が出て、これまで EC という決められた機能をどう使うかというところだったのが、それらをノーコードで管理できます。機能拡張ができたりとか、あとはウェブサイトも好きに編集できる、ということも、これまでと違うところです。

どういうところが使っているか？ Shopify 自体あまり表に出てきません。もちろん、ノーコードのツールなので出てこないんですが、大きな事例だと、NIKE がこれまで Amazon とかのプラットフォームで販売していたのを自社 EC サイトとして作る時に使ったと言われています。国内でもキャンプ用品のスノーピーク、金沢のゴーゴーカレーや、オリオンビールなども自社の EC サイトとして Shopify を使っています。ウェブサイト以外にいろいろな機能を持つサービスが第2世代です。

続いて第3世代。サイトもいろいろな機能も、より簡単により汎用的に作ることができるようになったところがノーコード第3世代の違いです。左側が、Bubble というアメリカニューヨーク発の会社ですが、ウェブデザイン、データベース構成、その中でデータをどう確保するかも、このボタンを押したら、どう加工されて次の画面に行くかというところを、ほぼプログラミングと同じような形で作ることができる。もちろん、ノーコードで作ることができるというものになっています。

ここまで来るのにノーコード第0世代からずっと続いて、ようやくこういうサービスが出てきている。右側が Kintone ですね。これは日本初のサービスになるんですが、画面にあるとおりに、左側のフォームの部品ですね。例えばリンクだとか、文字列とかラジオボタンというのをドラック&ドロップしていくと、入力画面が完成する。裏側ではデータベースもできて編集画面もあって、編集画面と一覧画面が作成されるようなツールになります。

より汎用的になって、よりプログラミングに近い形のものができる・作れる、というのが第3世代の例になります。この後、ノーコードの進化の過程は第4世代に移ろうとしています。ノーコードという言葉がなくなってしまうんじゃないかというぐらい、コードを書かなくても出来るサービスがこの後たくさん出てきます。一つ特徴的なのが、2023年にすごい今ブームになっている ChatGPT ですね。これはチャットのインターフェースでこちらからチャットをすると、裏側で AI が動いて AI が回答を作って回答してくれる。わかりやすく言うとそういうもので、これはもちろん AI をノーコードで使っている。この場合だと自然言語で指示命令をして、それに対して適切な情報を返してくるというサービスになります。

ここまで来るともう自然言語「日本語」で何かを指示をすると、それが返ってくる。そもそもドラッグ&ドロップやクリックなどではなく、もちろんプログラミングもしなくても、例えば、社内にある情報を全部 AI に保存し、必要な情報を質問をして返答させる事なども出来ます。

ここから画像・動画などのデータでも同じように出来るようになる。ここからノーコードの発展が 2 次関数的に増えていくと思います。

ここまでで「ノーコードとはなにか」について終わりたいと思います。次の動画で「なぜノーコードが使われ始めたのか」というところをお話したいと思います。

前回はノーコードとは何かについてお話をしました。ここではなぜノーコードが使われ始めたのかをお話したいと思います。こういうノーコードの歴史だったり定義だったり、後はノーコードが使われる背景ですね、今からお話するような背景・理由などは直接ノーコード導入には関係ない部分ではあるんですが、一般教養として知っていると、今後どういうふうにツールが変遷して行って、どういう新しいテクノロジーが出ていくのか出てくるのか、それをどのように会社で使うのかという未来予測ができるので、ぜひ頭の片隅に入れておいて頂ければと思います。

では、なぜノーコードが使われ始めたのかというところ、大きく 4 つの側面からお話したいと思います。一つは、クラウドという技術の進化、技術面になります。二つ目は、API 機能の一般化。これは主にアプリケーションの機能の側面になります。三つ目、社会の複雑化によるニッチ需要の顕在化。これは主に社会面、社会の側面になります。最後は、よく言われている IT 人材の不足ですね。人材面。この 4 つで、それぞれ深く詳細にお話をしていきます。

まず一つ目、クラウドという技術の進化についてお話をします。クラウドという技術自体、今はインターネットが繋がればブラウザがインストールされていると、さまざまな情報をたくさん取れますが、2010 年頃まではインターネットのサービスを作ろうと思うとサーバーが必要でした。サーバーも 50~100 万円するような機材を個人なら自宅に、会社だとデータセンターに置いたり、まず初期投資が必要でした。そのサーバーの中に OS をインストールをしたり、いろいろなものを設定する必要がありました。ですのでプログラミングする前にたくさんのお金を必要があった。お金もかかりました。Amazon の AWS と呼ばれるクラウドのサーバーが登場し、それが解消される。クラウドサーバーとはインターネット経由でそのサーバーを使え、プログラムをアップロードすると直ぐに動く状態になるもの。元々 Amazon のサービスですね。Amazon ドットコムと呼ばれるオンラインで買い物ができるサービス自体は世界中で利用されています。特定の国になりますが、多くの国で展開をされているサービスです。色んな人が色んな時に買物するので、凄いいアクセスがサーバーに来る。なので Amazon 自体もたくさんのサーバーを購入していました。ただ、あまりアクセスがない時は

使われてない状態になっていた。それを一般の人たちに使ってもらえるような仕組みにしよう。これが、クラウドサーバーの仕組みの一番最初の原点になります。

どういうふうに進化していったか？ノーコードに繋がっていくところなんですが、元々は自分たちでサーバーを持っていないといけなかった。そこからインターネットに繋がっていた、というところになります。一番問題なのは、もちろん家にサーバーを置くことでスペースを取ると言うこともあります。室内の温度・耐震・ネットワーク設備・電源設備というのを全部自分で管理したり購入しないといけなかった。それをクラウド側に置けば良いのではないかと。というところが先ほどの Amazon の AWS の仕組みです。クラウドというのは別に雲の上ではなくて、どこか見えないところに保存してあるというような意味合いですね。ですのでインターネット上に、物理的には Amazon のデータセンターと呼ばれる倉庫に入っているのですが、そこにインターネットを繋いで使えるようになります。ただ一方でクラウドのサーバーが出始めた時は、顧客情報や個人情報によく分からない場所のサーバーには置けない。最初はゲームとかですね。オンラインのゲームとか、もの凄いアクセスが来るのでオンラインゲームや SNS、FaceBook や Twitter も最初の時期はクラウドのサーバーを使っていました。後はコンシューマ向けですね。消費者向けサービスが多かったです。ですので大切なデータは未だ自分達の手元に置いてある状況でした。

今はどうなのかと言うと、こういうデリケートなデータもクラウド上に格納されています。一つは、さまざまな機能が強化されセキュリティが向上していること。もう一つは技術が進歩して、サーバーダウンが限りなく少ないこと。今はどんなシステムもクラウドに移行するのが、かなり一般的になっていると思います。クラウドが出たことで、開発者はプログラミングにより集中できより簡単にサービスを作ることができます。もちろんクラウドというものがなければ、ノーコードツールを作ることも皆さんが使うことも出来なかったと言えます。これがクラウドのデータセンターと呼ばれるものになります。サーバーがたくさん入るラックがあって右側が千葉県にある新しいグーグルのデータセンターですね。こういう大きな倉庫の中で保存がされている。

次は API 機能の一般化についてお話をします。そもそも API とは何だということもお話をしていきます。左側にアプリですね。皆さんが使うアプリがあります。右側にデータベースがある。そこに対して直接アクセスすると直感的に危ないというのは、何となく分かると思うんですね。

簡単なサービスであれば、直接データベースを見に行くというのは、もちろんできます。やってもいいかなと思いますが、では例えば、皆さんの口座情報や行政の住民票情報みたいなものを簡単にプログラムを書いてアクセスできるというのは問題ですよ。ですので直接アクセスすることはできないけど、一部の情報だけ授受ができるという仕組みが作れないかと。

そこで間に関所というか門を設けてですね。こういう命令であれば、こういう情報が返ってきますよ。ただ、こういう情報が欲しいと言っても、その情報は返ってきませんよ。というのが、この API と呼ばれる仕組みになります。細かい技術情報は少し調べて頂ければと思うのですが、こういう仕組みで安全にかつデータが相互にやりとりができる。

いま例えば、銀行のシステムを例にとると、銀行の中の皆さんの口座情報の勘定データですね。お金のやりとりの所を勝手にデータベースで書き替えられるのは、凄いリスクですし、社会が混乱すると思うんですね。ただ一方で、アプリで残高照会とかできるとすごく便利だと思います。後は振込みは公式のサービスだと思うんですが、残高照会やアカウント情報が取れるというところがあると色んなサービスで使われて使い勝手が良くなると言えると思います。

ですので API は データにアクセスし、より便利にするための機能です。いま銀行側でも、もちろん口座取引や振込みみたいなところはデータベースが隠されています。セキュリティも高く隠されていますけれども、残高照会であったり、一部データは API で公開し皆が情報を取れるようになってきています。

それによって、いろんなアプリやサービスが出てくることによってより銀行口座が使いやすくなる。預金などを確認し、それに合わせて色んなサービスを紹介してくれると銀行側にもメリットがあるという風になります。もう一つですね。他にも例がありますが、例えば天気予報とか皆さんアプリを入れて毎日見ている方もいらっしゃると思うんですが、これも気象庁のデータベースにいきなりアクセスしに行くのは、誤入力・誤消去などあった場合、すごく問題だし社会は混乱します。

一方で、例えば気象庁が発行する資料だけであれば、リアルタイムに天気予報もわからないし、凄く不便ですね。ですので気象庁や天気予報ができる免許を持っている会社などが API という機能で関所や入り口を設けて、こんなデータ下さいというのを出します、明日の天気は出します。ただ、どこの GPS でどのようにみたいな情報は隠されてる仕組みです。

会社の中でも勤怠管理とか人事システムですね。勤怠管理の中でタイムカード改竄できたら問題ですけど、勤怠の情報を受け取って、それを人事のシステムに渡すみたいなこともできますし、人事のシステムの中に入っている例えば、個人への会社の評価情報は絶対に取り出すことはできないけど人事情報に入っている名前・個人の ID・社員番号みたいなところは、常に最新の情報が取れて、勤怠管理システムにつながられます。

こういう形でサービス同士やアプリ同士のやりとりがよりしやすくなるということが API 機能になります。それがどんどん一般化することで、ノーコードというサービスがより色んな形でサービスを作ることができるようになりました。

次は社会面です。社会の複雑化によるニッチ需要の顕在化の話をしたと思います。2000年代、インターネットの使われ方はかなり画一的でした。みなさん Yahoo!で情報を得ていたし、昔でいうiモードのアプリで遊んでいましたし、Twitter もない時代、2ちゃんねるに書込むみたいな事がほとんどの方がそうだったと思います。後は、マスメディアをよく見て情報を得るところだったと思います。

仕事に関しても Windows で Excel や PowerPoint を使ったり、メールやファックス送信みたいなところが一般的だったと思います。翻って現在は多種多様な人生観・価値観があり、趣味嗜好がありインターネットの発展でどんどん個別化されてきています。特定の趣味で集まる、特定の情報だけ受け取るとか「特定の嗜好」グルメの中でもこの食品大好きで集まるみたいなところ。それはもう Twitter・Facebook もそうですし、他の SNS やコミュニティーも同じだと思います。

これら心身ともに多様な形になる中で、個別のニーズがどんどん細分化され、かつニッチだけど深い需要が出てくる。もちろんビジネス的には画一的なサービスですね。広く間口をとれないと、もちろん売上げが上がらないので。これまでは、全員同じシステムを使いましょう的なところがありましたが、今はノーコードという技術がある中で、手軽にサービスが使える。必要とされる人が使って、ニッチだけど必要なところを埋められる。

ちょっと具体的に話をします。EMPTYDRESSY という無人のレンタルドレス屋さんがあります。ドレスを着ない人とか、レンタルで服を借りない人にとってはピンと来ないと思うんですが、そこには強い需要があります。今だとインスタグラムだったり、写真を撮る機会が多い中で、たくさんの服を買う余裕はないけれども同じ服は着たくない。レンタル衣装で他の人と同じにならない服を着たい。でも一方でレンタル店で顔を合わせるのはちょっと気まずい。

今までは無人でレンタルドレスショップをするならかなり投資が必要システムもちゃんと作り込んで開業する必要があった。もちろんその後うまくいかないこともある。この場合は全部ノーコードでこの辺のオペレーションですね。実際にレンタル予約をしたい、レンタル予約しこの時間帯借ります、無人店舗なので鍵の受け取りもデジタルでやりとりします。暗証番号で開錠でき返却管理もできるというような仕組みをエンジニアではなく、ビジネスを考えた人が一人で作ったという事例です。

もちろん実際これを作った上で、ニーズや店舗運営に合わせてノーコードだといっぱい変更ができる。これがノーコードのメリット。普通のシステムだとちょっと文言を変えたり処理が変わるだけでは 10 万、20 万みたいな感じになると思うので。そこを自分でできるというところで、ニッチなところですね。無人でドレス借りたいという人たち向け、刺さる事業を作っていく。もちろん無人なのでレンタルドレスが高いイメージも変えられます。数万円するような

ドレスも 3,480 円で 1 週間借りられる。レンタルドレスにありがちなサポートスタッフの配置。それがないので より安くなるっていうのも出せています。

もちろんスタッフが必要な人はお金を払って高い所に行けばよくてここの利用者にとっては自分たちでやるから安くして欲しい。なので初期投資とかランニングコストを少ないまま価格も抑えてスモールにスタートができるというようなところがあると思います。これだと全国に展開をしてというのはなかなか難しい大手の衣料品店をやるというのはなかなか難しいですが、ただ、こういう需要は確実に存在しビジネスになるという意味だとノーコードで作って早く提供し変更するという点が良いメリットです。例えばお子さんを育てる環境も今すごく変わっているかと思えます。なかなか共働きで外に遊ばせに行けないとか、子供達だけで遊ばせられないとか色々状況が変わる中で、近所にどうい公園があって、そこが安全なのかどうかパッとわかるようなマップがあるといいですね。というので、この左側ですね。横浜市瀬谷区の公園マップになります。直感的にこれビジネスにならない、お金取れないと思うのですが、でも需要は一定数ある。社会的にもこういうのあったらいいよねというものだと思います。

ただ、一方で区役所のウェブサイトを見ている、細かい Excel をそのまま PDF にしたような使いづらい情報しかとれないけれど、こういう形でマップにして使えるようになると、この地域に住むお子さんだったり親御さんが使いやすいという風になります。このツールは、Glide というノーコードツールを使っています。裏側が Google のスプレッドシートになっていて、必要なデータを入れると自動的にマップ上に展開されるサービス。右も同じく Glide で作られた品川区のテイクアウトのマップです。

コロナ禍で飲食店が打撃を受けたのは皆さんご存知と思いますが、そこでいろんな飲食店がテイクアウトするような事業転換をしました。ただ一方で大手グルメサービスだと掲載に多額の費用が掛かるので、これは区民の有志が作った無料のサービスになります。ほとんどの飲食店はもともとテイクアウトやっていたいなかったので、このサービスで利用者はテイクアウトの有無がわかってそこに行って、じゃあ、もともとお店に行ってお飯を食べていたけれどもテイクアウトをやっているのならテイクアウトにしよう、ということが出来る。これも社会の複雑化によりニッチな需要が顕在化した例だと思います。

最後、人材の不足についてもお話をしたいと思えます。ちょっと大きなマクロ視点で情報を整理すると、少子高齢化によって労働人口が減少してきている。そもそも労働者が少なくなっている。IT 職はこれまですごく不人気でした。職場環境はブラックであったり、報酬の割に責任が重い、暗いとかのイメージがついて不人気でした。もう一つ、団塊の世代と呼ばれる人達がリタイアをしていく。この人達は今までのシステムを支えてきた人たちなのですが、若い世代が入ってこないとなウハウが引き継がれない。今後、今までのシステムが安定して動くのだろうか不安視されています。

経済産業省の DX レポートの「2025 年の崖」と呼ばれる予測です。団塊世代がリタイアしてもシステムは安定稼働するのかどうかというのを「崖」という言葉で表現しています。こうなってくると IT 人材の採用が困難で且つ給与が高くなっていく。もちろん人材が少なくなると採用競争になり給料が高騰する、そうするとどうなるかというと、高い給料を払って人材を獲得するのが一つ。もう一つは、新しく人材を採用するのではなくて社内人材の育成。そこにノーコードが貢献できるだろうと思われます。

もう一つ、今回もう少し会社の中に入って話をすると、IT 技術者が正社員で働くほど、IT の仕事は社内になかったよねと、必要な時はシステム会社の人を呼んできて丸投げして、その時だけ作ってもらったらしいよねと。でも、ここ 10 年 20 年で、IT が不可欠なビジネス環境になった。でも、これまで正社員じゃない人達にお願いをしていたので社内に全然ノウハウがなく、あってもすごい属人化している。でもどんどん IT がなくてはならない状態になってきている。そうすると、IT 業務が多いにもかかわらず専門の人材がいらない。

IT 人材育成に時間とお金がかかる。外部の委託業者に全部任せて結局、使うノウハウはあっても作るノウハウがなくなる。そうすると専門人材がいなくて、というようなループに陥ってしまう。ただノーコードという観点で言うと、社内人材育成に可能性が出ます。右側の人材育成する時間とお金という部分でループを断ち切れます。

実際どれくらい時間とお金がかからなくなるか、漠然と完全に私の主観でお話すると、私自身プログラミングの講師もやっていました。本当にコードを書くのが初めてという初学者がシステムを開発するのに大体 6~8 ヶ月。一人前で独立するのに 1 年半程かかる。一方でノーコードの初学者がシステム開発できるようになるまで大体 3~4 ヶ月でできます。

実際 2 ヶ月ぐらいで副業で仕事をしているという人達もいっぱいいます。やはりここまで短くなると社内で人材育成できるよねとなって今ノーコードがブームになってきている。皆さんが使える、使おうと思う人達や会社が増えてきている。という事になります。

ここまで、なぜノーコードが使われ始めたのかという 4 つの理由についてお話をしました。

続いてノーコードの種類とツールについてお話します。

ここからいろいろなツールの名前が出てくるのですが、まず、ざっくり覚えて頂いて後でご自身で調べて頂いたりだとか、この後でも何度も出てくるツールがあるので、そこで具体的な機能だったりをおさえて頂ければと思います。

まずはちょっと大きくノーコードの種類で分けます。カテゴリーで分けていきます。大きく機能を分けると 3 つに分類されます。一つは Web デザイン・Web 制作ですね。Web サイトを作るためのものや、フォーム機能や CMS だったり記事を管理するような機能も含めてのものになります。

主に広報とか PR・マーケティングですね。対外発信・お客様とのやりとりに使用するノーコードになります。2 つ目は、データの管理ですね。これも所謂データベースだと思って頂ければと思います。データを保存・管理する場所、適切な形でそのデータを表示するもの。

企業業務は、ほぼデータが重要だと思しますので、これをノーコードで使うという意味でも非常に重要なカテゴリー。もう一つがタスクの自動化ですね。これは入力データや表示出力されるデータの加工をしたり、あとは自動で通知をしたり、保存データを PDF にしたり、PDF でメールを送ったり、分析とかいうところですね。処理を人を介在せずに自動です。そういうノーコードもあります。

で、これらを兼ね備えて全部持っているノーコードもあります。オールインワン汎用型、オールインワン業務特化型の2つ。ですので全部で5つに分類してお話をしたいと思います。

まず少し入る前の導入ですが、今ノーコードツールって大体どれくらいあると思いますか？頭に浮かぶ中でも10~20あります。WordPressも含めるとそれくらいあるかなと思います。

今どれくらいあるかという、これは毎月新しいものが出てるので正確な数字は分かりませんが、国内だけでも150。海外を含めたら500以上のノーコードと呼ばれるツールが出てきている。

段々とノーコードツールも特化していたり個別具体のサービスになっていくところがあって、この表示されているものでいくとIoT、AI、VR、チャットボット、フォームとかですね。より個別の機能に特化して、ただその中でノーコードとしていろんな事ができるというのが増えてきているという状況です。

具体的なツール少しお話をしたいと思います。まず1つ目、Webデザインの制作です。STUDIOというWebデザイン制作のノーコードを説明します。デザイナーさんがプログラムのコーディングなしでできるとなると大きなメリットになるかと思います。後、この右側ですね。CMS/記事を管理する機能もあるので、実際テキストだけ書いてアップロードすると記事の画面ができるとか、あとは今、AIを取り込んで日本語/自然言語で指示を出すと、AIがデザインの修正をするみたいなのも新しく出てきている。このツールって別に海外でも使えるので、国外からの利用も多いと言われています。

もう一つ「ペライチ」というサービスを紹介したいと思います。先程紹介したのはデザイナーさん向けで細かいチューニングができるもの。ペライチは、Web制作をしたことがない、そもそもIT職でもない個人商店の方々や年齢高めの方々でも使えるWeb制作ツール。

この右側画面を見て頂ければと思うのですが、見出し・ボタン・画像などのプラスを押すと部品を入れることができ、そのボタンや写真をドラッグドロップして配置をしていくと Web サイトが出来上がるというものになります。

もちろん細かいレイアウト変更などは STUDIO 程にはできないのですが、初心者でも簡単に作れるという意味だと大きな魅力だと思います。大体 Web 制作会社に依頼するとお金が掛かるが、それを自分たちで出来る。IT 職以外の方が 0 を 1 にできる場所は大きな魅力ですね。

今は決済や予約の機能が付いていて、より個人事業主・小規模事業者にやさしいツールになってきている。

続いて、データ管理。クラウドのデータベースで国産ノーコードでは Kintone がまず最初に出てきます。Kintone はノーコードの概念が出る前からノーコードをコンセプトにやっています。

入力の画面ですね。左側の例えば日報という画面にラベルとラジオボタンとかがあると思うのですが、それを右側画面にドラッグ&ドロップで配置をしていくと画面ができて、かつデータベースもできるというようなサービスです。

Excel 業務をクラウド化し Kintone に載せることなどがよくあります。Excel 自体はインストールされている自分のパソコンで使うため、複数人で一緒に操作をすることはできないと思うのですが、Kintone だとそれができる。アクセス権限や組織図に合わせてユーザー設定する事もできるので大規模組織でも活用ができる。

もう一つは API の機能が付いているので、他社で開発したプラグインを使うことで、自分たちでコーディングしなくても機能追加できる。Kintone の凄いところは、フォームでも、カレンダーでも、Gmail でもなんでもいいですが、そこに何か情報が入ってきた時に自動処理が始まり、どこかにデータを送る、例えば LINE でもメールでもいいと思います。他のシステム、例えば顧客管理に情報が行くということもできる。作り方なのですが、丸くなっている所がコネクターと呼ばれる部分で、このコネクターを選択して、スプレッドシートの例でいうと、コネクターを選択して行を追加・消去などを選んで、実際の中身のデータを入れると、そのコネクターが出来上がって、それを隣につなぐ。例えば Google カレンダーと LINE を繋いで、LINE でどんなメッセージを送信するか。みたいなのを決められる。ですのでかなり直感的に使うことができる。そこが良さかなという風に思っています。

同じタスク自動化で国産「Yoom」も紹介します。国産なので free とか Kintone など国内の勤怠管理・労務管理と接続ができるところが大きなメリット。データベースやフォームもあるので、データベースに一時的に情報を保存するとか、フォームを作って登録情報を Google

スプレッドシートに保存とか、問い合わせ管理に保存、Kintone に保存みたいなことができる。

例えばいま Excel で管理している情報、顧客情報や案件情報、商談情報などあると思うのですが、それを請求書にしたりとか、請求書作成ツールにデータを送り自動で作る、経理クラウドサービス free などに送って請求タイミングで金額と顧客名を計上する、途中で経理の担当者がチェックして「この請求書の期日違いますよ」など確認するという事もできます。

日本語で使えて且つ日本語のサポートができるというのが大きなメリットかと思います。

次はオールインワン汎用型について話をしたいと思います。オールインワン汎用型で一番有名なのは、この「Bubble」です。ここまで汎用的なノーコードはないと言われるぐらい使われている。元々、アメリカニューヨーク発の Bubble。コードを書かなくてもある程度のプログラミングの知識があれば、誰にでもすぐアプリやシステムが作れるというものです。

画面を見ていただけると、画面のデザインにテキスト・ボタン・リンクイメージ等があると思うのですが、それを右側にドラッグ&ドロップをして、そこで設定を変えていく。すると表示のデザインができる。そこで、Bubble の場合そのボタンを配置してボタンを押すとどういう動きになるのかというのも設定ができます。入力された情報でボタンを押すと、そのデータがデータベースに保存される、というようなこともできますし、データベースを保存する前に、ちょっとデータを加工する姓と名を分けるとか、何かそういうこともできるようになります。

プログラミングで基本的にできるものは、ほぼ Bubble でできると捉えて良いかなと思いません。ただ、一方で大規模のアクセス。オープンな誰でも使える状態だとアクセスの負荷が上がる可能性があるので、そういうサービスは、やはり 0 から作った方がいいですが、最初は Bubble で作り、そこからコードに書き換える事例もあります。

あとは社内システムに使う場合は、英語の管理画面が難しいのでどちらかというと業務システムというよりも社内の新しいシステムや、そもそも新規事業や新規サービスを立ち上げるときのアプリケーションだったりとして使うところが多いかなと思います。何かよくあるようなマッチングのアプリですね。出会い系のマッチングではなくて人材のマッチングとか、売り手・買い手がいて取引をするみたいなビジネスモデル特化型で業界ごとによくあるようなモデルだと思うのですが、あとは何かを診断・シミュレーションしたりとかのシステムを新しく作る時に使われる。新規事業を立ち上げる時のプロトタイプです。

まずどういうシステムがあればビジネスが実現できるか最初のステップとして Bubble を使うことも結構あります。もう一つですね。先ほどお話をした「Glide」です。品川区のテイクアウトや公園マップのツールになります。これも左側画面のように、部品をドラッグ&ドロップして作る。ユニークなのが裏側は Excel データなどでいいんです。それに合わせてデザインを作っていくというのができるのでより直感的にできる。

あとはタスクの自動化機能のように、ボタンを押すとデータ加工や別のところに保存などができたり、自動返信したりできる Glide も海外ツールでアメリカ・サンフランシスコのツール。Bubble と比べるとデザインの制約等はあるけれどより簡単に作れデータベースも Excel 等で比較的分かりやすく編集がしやすいその辺りの理由から使われているケースも結構あります。

もう一つオールインワンの業務特化型を紹介します。「Shopify」カナダ・オタワ発の EC サイトの開発サービスです。少し前の動画でも紹介しましたが、これまで EC サイトを持つとすると機能が限定されたものが多かった。もちろん昔からあるカートや決済の機能がついているサービスは結構あったけどカスタマイズがしづらいというようになっていて、一方でインターネットの普及により Amazon や楽天などの大きなプラットフォームで販売した方が収益化があがるようになってきて、集客はもう全部任せちゃう。というような時代がありました。

今やファッション・コスメ等は SNS で紹介されているものをそこから買うパターン。必ずしも Amazon・楽天に沢山の手数料を払ってまでやらない。すると自前で EC サイトを作り、自前のブランドとしてカスタマイズし機能も拡張して使う、というのができます。Shopify の場合、画面デザインもできるし、データベースもあるし、あとはタスクの自動化というのもできます。メールが飛んだり、お客さんに情報が入ったり、カスタマイズされたお客さんごとに違う情報を提供をしたりとかできます。そういう意味でオールインワンなんですけど、ただ Bubble とか Glide のように何でもできるかという点、このツールはあくまでも EC 機能を持ったサービスになります。

次にノーコードのメリット・デメリットについて話します。

まずノーコードのメリットからお話します。一つは開発スピードの速さですね。しかも圧倒的に速いです。単にコードを書かないからではなくて面倒な設計資料を作らない、デザインも 0 から考える必要ないですし、一旦作ってから見せるという意味だと、そこまでのスピードも速い、改善も速いということでトータルとしてスピードが速い。あとコミュニケーションミスも少ないというのはあります。

ひとつ事例をご紹介しますと、加古川市のコロナ給付金の申請システムになります。コロナ給付金というのが、コロナが始まって 1~2 カ月ぐらいですかね給付が政府で決まって、各自治体で申請や給付の準備をしていたところにいきなり新たなルールが決まって、いきなりやるという感じになりました。当時を思い出すと、自治体によって給付の早い遅いに差がありどれだけ早く給付できるかの競争も自治体間であったと思います。その中でこの加古川市の事例を話したいと思うのですが、画面に表示されている右側が紙です。

この紙を郵送します。給付対象となっている世帯に給付をします。自治体側に持っている住所に郵送する。そうすると、その場所にちゃんと世帯があるという確認ができます。郵送で送られた用紙に照会番号というのが書かれていてこれと一緒に申請用紙も入っていて

紙で申請ができるんですが、その後、申請システムです。左側 Kintone のシステムですけども Kintone 上でこの照会番号を入れて OK にすると申請が終わる。

何がいいかというと、紙の申請よりも Web で申請してくれた方が最終的に給付が早くなる。住民側も市側もメリットもあります。加古川市側のメリットの話をするると既にこの市は Kintone を使っていたそうです。そのノウハウがあったので、給付金の受付どうしようとなった時に Kintone 使おうとなって、このシステム自体は数日で作られたそうです。

その後、もう物ができてるので担当者から上司そして最終的に市長へプレゼンをするのも、実際の実機でプレゼンしたそうです。その場で決裁が降りて、このシステムで運用しようとなったようです。あまりにシステムができるのが早すぎて給付しますという広報など紙側の準備の方が遅かったと言われてます。

もちろんコードも書けるので最終的にコードを書くという手法もあるのですが、まずはノーコードの範囲でも柔軟性が高いと言えます。

もう一つですね。学習コストの低さが挙げられます。前の動画でも話しましたが学習コスト。プログラミングだと 1 年かかって、ノーコードだと半年もかからない。ここの話をできるだけ定量的にお伝えしたいなと思ってこのスライドを作ったのですが、あるプログラミングスクールは、返金可能期間が 2 週間設けられていてその時点で約半分が挫折します。最後までいくのは全体の 20% 以下らしいです。

ですので 80% 以上の人達は、やる気を持って自分で申し込んでいるけど途中で挫折してやめている、というのがプログラミングです。その挫折する気持ちもわかるんです。いろいろな知識を持っていないといけませんし、単にプログラミングだけでなく、サーバー、ネットワーク、セキュリティ等多くの知識が必要なので、途中で挫折するポイントが沢山ある。

一方でノーコードであれば、その挫折が 0 ではないけれども少なくなると思います。「Udemy」というオンライン講座を購入できるサービスがあるんですが、左側が Bubble の開発コースで約 11.5 時間のコース内容になります。対して右側 Python と呼ばれるプログラミング言語の一つですがシステム開発できるまでに約 54 時間かかります。時間が 5 倍違うってことは、5 倍挫折するポイントがある。

学習コストが低ければ低いほど実際に人材として育成される可能性が高まるという言えると思います。もう一つですね。これも関連するんですが、社内の人材を活用できるというのも大きなメリットになります。これは星野リゾートさんですね。全国にあるホテルであったり旅館を運営する会社ですが、競争に勝つための全社員 IT 人材化計画っていうのを DX と呼ばれる前段階からかなり実践されているようです。

実際に社内人材育成をすることは聞いた感じでは、なかなか難しいと思うのですが、自分達はいち消費者としては多かれ少なかれ IT を使って生活しているしどうすれば動くかというのは、何となく直感的に分かる筈なんですね。プログラミングになると、そこが直感的に分かるまで時間がかかっていたのが、ノーコードだと結構手前の段階で分かるというのがあります。

この星野リゾートさんの事例でも全員を IT 人材化するんですけど、全員プログラミングまで出来なくてもいいと。

紙、電話、口頭指示だと人間が介在するのでヒューマンエラーが起きます。ヒューマンエラーを起こさせないために 2 回も 3 回も他の人がチェックして、結局そこに人が介在している。マニュアルを作って、手順を作って、どんどん業務が増えていくという事がかなり多いと思います。

このスライドのように、日本中の社会人が自分の仕事でミスをしないうちにこれだけ本が数多く出ていて、これだけ読んでみると、ミスをしないうち、という本がこんなにたくさん出ている。それだけ現場が疲弊している。実際「人間はミスする」という前提に立っていない事が、実は大きな問題です。

システム化すると、そのヒューマンエラーが減るのは勿論ですが、日に何度も同じチェックをしていた人達の仕事が、もっと別の仕事になる。よりビジネスに近い部分であったり、もっと多種多様な仕事ができると思っています。すごいルーティンの地味な作業なんだけどすごく精神的な負荷もかかる、というところから解放される。そういう働き方の改善という意味だとヒューマンエラーを低減という大きなメリットのひとつだと思います。

メリットの最後ですね。初期導入費用が安い。これもノーコードの良さですね。使いたいと思ったらすぐ使えるものが多いです。無料プラン・無料体験・デモ環境というのも多数あります。プログラミングするシステムだと出来上がる直前にならないと実際にもものを見れないということになります。その時点でイメージが違うとなると修正にもお金・時間がかかるしもう引き返せないで、そのまま使うというような悲劇が起きます。

ノーコードの場合、すぐ使って使いやすさのイメージをつかんで同僚・上司に見せてイメージ共有もしやすいですし、もし無料プランで行けるのであれば、そのまま無料で使えばいいし有料プランの機能が欲しければ、有料プランにすれば良い。そもそも初期導入の費用が安い分、検討がすぐできる。初期導入費用が安いノーコードのメリットが享受できるかと思っています。

次にノーコードのデメリットについてお話します。

今までノーコードの良いところばかり学習したんですが万能ではない。例えば、仕様、使い方、機能等がすごく複雑な基幹システム。製造業でいうと生産管理とか、金融ならお金勘定のところなど。自社サービスの根幹となるシステムは、そもそも他と差別化している筈。世の中にあるツールだけでは出来ない部分がある程度出てくる。

複雑になっているシステムをノーコードで置き換えるのはなかなかの悲劇になると思います。私の経験したプロジェクトで守秘義務から全部公開はできないのですが、複数のプロジェクトを合わせた感じでお話ができればと思います。

それはスマートフォンから親がベビーシッターを予約し、その会社が稼働可能なシッターを探し依頼し、当日保育をして完了というサービス。この会社の中で一番根幹となるサービス。事業の肝となっているシステムです。

その地域だったり、どこからどこに移動するかであったり、移動する方法とか移動する時間とかで金額が変わってくる。あと子供の状況を詳細にヒアリングし、熱があるとか、二人いるとか時間など細かい情報。電話やトラブルにもすぐ対応しないといけない。

もともと顧客管理のシステムをノーコードのシステムを使って開発する予定でしたが、実際に仕様を聞いていくなかなか難しい。ノーコードの機能だけだとできないし、カスタマイズだけでもできない。結局コードを書かないといけない。それも結構書かないといけない。

商談して見積もりをしている時だと、そこまで複雑ではない。これは我々もそうだったし、お客さんもそういう認識だったので、実際に現場にヒアリングして、いま使っているシステムや保存しているデータを見て、ということが起きました。

結局お客さんとも再確認して、コードで開発した方が絶対いいですという話でコーディングに切り替えた。ノーコードだけでは絶対できないことがある。そこはコーディングと違う部分。なので何でも安請け合いですと後で自分の首を絞める。それはそのままお客さんの事業に対しても響いていくので注意が必要だと思います。

もう一つは、サービス終了のリスクですね。プログラミング言語であればどこでも動きますし、ひとつのサービスが終わってもサーバーがある限り継続して使えます。ただ、一方でノーコードは、そのサービスにすごく依存をするのでそのサービスが終わると別サービスで作るという訳にもいかない。これまで私は経験したことがないですし、なかなかそういう話も聞かないのですけれども0ではない。

一番大きな例は Google App Maker というもの。いま Google の中に App Sheet と言われる機能があるのですが、それに切り替わるタイミングで「App Maker 終わります」という案内が来ました。もうこうなってしまうとどうしようもないので、別のノーコードで作り替えるしかないという風になります。

こうしたサービスの整理だけではなく、例えば倒産するとか、サービス提供者が終了したくない場合でも終了する可能性がある。そこは事前に察知するのも、なかなか難しいので見極めが必要です。

もう一つはセキュリティリスクですね。ノーコードって簡単に使えるので、一方でセキュリティーの知識は社内で共通認識を持っておかないといけない。プログラミング開発だとするとプロが開発をするので、セキュリティーの部分をちゃんと担保して作るのですが、ノーコードはプログラミング知識がなくても使えるという意味だとセキュリティー知識がなくても使えるということなんですよ。ですので悪意を持っていなくても設定一つで変更ができてしまって個人の情報が漏れてしまうということがあります。

これはあるゲーム会社で名前は伏せてますが採用の情報ですね。学生の情報を恐らく Google スプレッドシートに載せていた。そこまでは良かったんですが、そのデータベースの公開設定に問題がありました。Google シートの設定を全公開にした。リンクが分かれば情報漏洩。なのでインターネットを検索したらすぐ出てくる訳ではないので個人情報が漏洩してるリスクとしては少ないとは思いますが、とはいえ 0 ではないというので、お詫びを含めての対策や再発防止策等も含めすごくコストがかかってしまったという事例です。

特にセキュリティー面。IT に明るい人が管理者になって権限とセキュリティー知識を持ちノーコードを使う事が非常に重要です。