

---

---

# 当社の在庫の数量の公開と提供に ついての取り組み

シンワ測定株式会社

---

## ■ 執筆者 Profile ■



2012 年 シンワ測定(株) 入社  
社内システム担当

吉田 晃平

## ■ 論文要旨 ■

得意先様からの電話や FAX で在庫の数量についての問い合わせが営業所に集中し、業務を圧迫していた。問い合わせを減らすために、在庫の数量を公開する得意先様向けの会員サイトを作成することになった。

スマートフォンが普及し、Web で様々なサービスを受けることが当たり前の時代、情報システム部に求められる内容やスキルも拡大してきている。しかし、.NET を使用した自社向けのフォームアプリケーションの開発・保守をメインに行っている情報システム部には、大きな課題だ。解決のために ASP.NETMVC というフレームワークを使用し、開発の方法を大きく変えることなく、学習コストも抑えて課題を解決した。本稿では当社の事例を紹介するとともに Web アプリケーションへの移行の課題について考察する。

## ■ 論文目次 ■

<b>1.はじめに</b> .....	《 3》
<b>2.背景と問題点</b> .....	《 3》
2.1.背景	
2.2.システム構築の問題点	
<b>3.目的と方針</b> .....	《 4》
3.1.目的	
3.2.開発環境の選定	
<b>4.システムの構築</b> .....	《 5》
4.1.会員サイトの作成	
4.2.在庫数表示の工夫	
4.3.在庫検索数の工夫	
4.4.レスポンスデザインに対応	
<b>5.評価</b> .....	《 9》
5.1.開発の効果	
5.2.アクセスの推移と効果	
<b>6.課題と所感</b> .....	《 11》
6.1.開発の課題	
6.2.所感	

## ■ 図表一覧 ■

<b>図 1</b> 営業所ごとの月の在庫の問い合わせ数 .....	《 3》
<b>図 2</b> ユーザ登録の仕組み .....	《 5》
<b>図 3</b> 在庫の状態表示のしきい値について .....	《 7》
<b>図 4</b> 製品コードのみの在庫状況の検索 .....	《 7》
<b>図 5</b> レスポンスデザインに対応 .....	《 9》
<b>図 6</b> アクセスの推移 .....	《 10》
<b>図 7</b> 月の在庫の問い合わせ数 .....	《 10》
<b>表 1</b> フレームワークの比較 .....	《 5》
<b>表 2</b> セキュリティの検討 .....	《 6》

## 1. はじめに

当社は生活の基準となる「測定器」の製造・販売を行っている。特に「測定器」の中でも金属製スケールの分野では 圧倒的なシェアを誇っている。

常に新しい製品の開発・製品の品質化・生産の効率化に取り組み、お客様のニーズを捉え、そのニーズに応える製品作りを目指している。

## 2. 背景と問題点

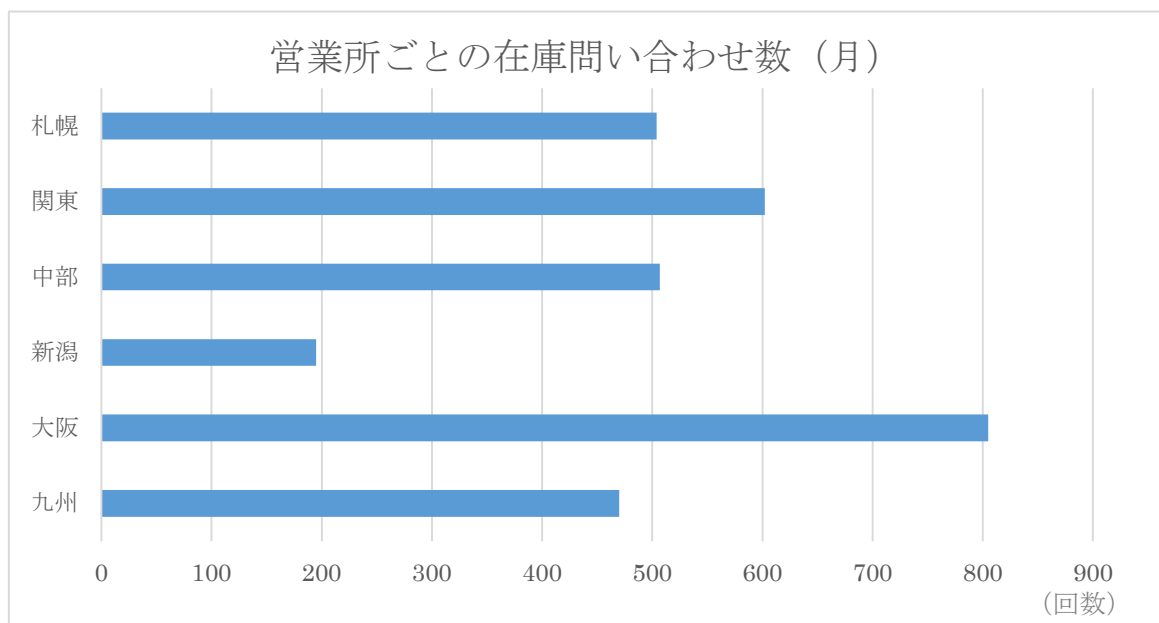
### 2.1 背景

当社の営業所へ在庫の状態の問い合わせ数が増加している。年々当社の取引先が増えるのに従い、在庫の確認が増え当社の営業業務の負担が増加している。営業業務は、国内の営業所 6 拠点にそれぞれ 1～4 名在籍しており、主にお客様からの受注や見積もりの対応、製品に関する問い合わせ対応など営業に関するサポートを行っている。いくつかの問い合わせの中から、在庫に関する問い合わせ数について図 1 にまとめた。

一部の大手商社からは CSV 形式のデータを毎日送ってくれば、自社システムで取り込むのが良いという意見をいただくことも多い。実際、そのような対応をしているメーカーもあるようだ。

しかし、すべての取引先のニーズに応えることは難しい。金物の商社は個人で細々やっているとところから大手まで非常に幅が広く、昔ながらのやり方で業務を行っているところも多いためだ。

図 1 営業所ごとの月の在庫の問い合わせ数



## 2.2 システム構築の問題点

スマートフォンが一般的になるに従い、どこからでも情報を取得したいというニーズが年々高まっている。大手メーカーでも在庫の確認から商品の購入まで出来るサービスが続々と登場してきている。

当社の営業からはシステムの構築にあたり、在庫の情報を外部に公開でき、パソコンやスマートフォンに関係なくどこからでも当社の在庫を確認出来るようにしてほしいと要望をもらった。

当社の社内システムの部署では、長年.NET と SQL を使用したフォームアプリケーションの開発と COBOL を使用しオフコンの保守管理と基幹システムの管理を中心に行ってきた。ホームページの更新を自社で行っていたこともあり、HTML とスタイルシート（画面の装飾を行う言語）、javascript（画面のアニメーションなどを制御する言語）については少しわかるくらいであった。

そのため、外部公開のシステムをどのように構築していけば良いかが大きな課題だった。

## 3. 目的と方針

### 3.1 目的

- ・会社の中のシステムや構成をなるべく活用すること。
- ・お取引先様の環境に関わらず、弊社の在庫を確認出来るようにすること。
- ・在庫確認を電話やFAXを行うよりもネットで確認するほうが手軽だと感じてもらうこと。
- ・amazonなどの通販サイトのような構成のデザインにすること。

### 3.2 開発環境の選定

まず外部への公開にあたり、現在主流となっている PHP や MySQL の環境で Wordpress（Webサイトを簡単に管理するシステム）を活用した Web アプリケーションの作成について検討を行ったが、当社の Windows を中心としたシステム構成と異なり、メインのプログラムは C# で Web アプリケーションだけが PHP という形では、メンテナンスだけでも複雑であるため、目的に沿わないことが分かった。加えて、当社では誰も作ったことの無い言語のため、導入しても学習することが多く難しいことが分かった。

そこで当社の Windows 環境でもスムーズな開発を行うために ASP.NET と ASP.NETMVC という2つのフレームワークに絞込みを行った。

ASP.NET という選択肢は、コントロールを配置するだけで簡単に作成出来るというのはとても魅力的でフォームアプリケーションのように開発することが出来そうだったが、自動で作成されるソースコードがわずらわしく、スマートフォン専用サイトを作成するのが難しく感じた。また、独自の文化が多くあり学習に時間がかかるように感じた。

ASP.NETMVC HTML とスタイルシートなどを自分で書く必要があるが、自由にサイトを構成することが出来るため細かな要望にも応えることが出来る。MVC（モデル・ビュー・コントロー

ラー) のルールさえ覚えれば少ないコードでサイトを構築出来ることが分かった。また Web 標準の技術である bootstrap を始めとした多くのサイトで採用されているフレームワークを使用することが出来るため、デザインに学習する時間を割かなくても簡単にモダンデザインでページを作成することが出来ることが分かった。

表 1 フレームワークの比較

	ASP.NET	ASP.NET MVC
コントロールの作成	○	△
自由なデザイン	×	○
必要なスキル	C#	C#・HTML/CSS・javascript・・・
当社の資産を活用	○	○
Web 標準のフレームワーク	×	bootstrap を始めすべてを使用することが出来る

#### 4 システムの構築

##### 4.1 会員サイトの作成

外部に公開し、誰でも閲覧出来るという状態にしてしまうと競合他社に弊社の在庫状態だけではなく在庫数の推移まで知られてしまうことになる。そのため、会員サイトを作成する必要がある。

また、IDとパスワードが何らかの原因で競合他社に漏れてしまい当社の在庫が知れてしまうのも防ぐというのも課題だ。

当社で会員サイトを作成するのは初めてだったため、どのような方式が適しているのか費用面とあわせて入念に検討を行った。

結果、Aの方式を採用することとした。IDとパスワード、あわせてHTTPヘッダから端末情報を取得することで、普段とは違う端末からログインしているユーザを探しやすくなり、ログイン先の画面に表示することでユーザへの抑止力につながると考えたからだ。また、費用面や保守の面でも優れている。

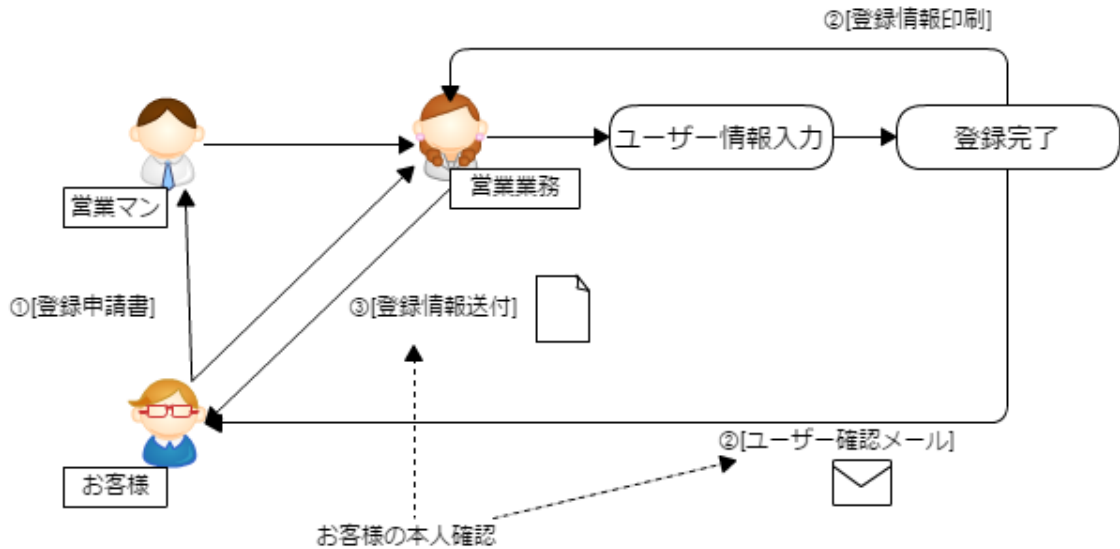
表2 セキュリティの検討

パターン		A	B	C
ログイン方法		パスワードとID、端末情報	パスワードとID、ワンタイムパスワード	パスワードとID、電子証明書
配布方法		1人に1つつ割り当て、 端末情報の登録	1人に1つつ割り当て、 パスワード生成機(トークン)の配布	1人に1つつ割り当て、 電子証明書の配布
操作		ログイン時に端末情報を送信するボタン押下	ログイン時にトークンから作成されるワンタイムパスワードを入力する	特になし
セキュリティ	パスワード更新	○ (使用者・管理者から可能)	○ (使用者・管理者から可能)	○ (使用者・管理者から可能)
	他人への無断譲渡	○ 抑制される (使用者名・端末情報を表示する)	○ (トークンを複製することは出来ないため)	○ (電子証明書をインストールしたパソコンでしか使用できないため)
	使用者の退職・異動時の対応	△ 旧使用者のアカウント停止 (定期的に確認する必要あり) 新使用者のアカウント配布	△ 旧使用者のアカウント停止 トークンの回収 新使用者のアカウントとトークンの配布	△ 旧使用者のアカウント停止 電子証明書の失効 新使用者のアカウントと電子証明書の配布
	不正利用のアラーム	○ (以前の端末情報との照合)	○	○
	端末の特定	○	○	○
	セキュリティ度	中	高	高
運用	ID管理の容易さ(シンフ側)	△	× トークンの電池が切れるところに新しいものを 送付する必要がある	×
	ID管理の容易さ(使用者側)	△	× トークンを紛失するとログインが出来ない 再発行作業が必要に	× 電子証明書をインストールしたパソコン以外での使用ができない 定期1年毎に電子証明書の更新が必要
	パスワード定期更新の手間	△	不要	不要
	可用性	中	低	低
モバイル端末への拡大		△	△	×
別途費用		○ 不要	△ 1,000円×個数(ID) サーバー構築費(50万～)	× 10,000円×個数(端末) サーバー構築費(50万～)
工数		× 1週間(端末情報の登録方法を模索)	× 4週間～(別途システム部門協力会社 との打ち合わせも必要)	× 不明(別途システム部門協力会社 との打ち合わせも必要)

このような会員サイトの作成には、NETIDENTITY というユーザ認証を行うためのフレームワークを使用する。当社で会員サイト作成の経験は無かったが、このフレームワークを使用することでパスワードのハッシュ化やロールの管理などセキュアな環境を作るための機能を簡単に利用することができた。

ユーザ登録の際にも簡単に登録出来るように工夫を行った。取引先には様々なユーザがありパソコンのスキルもそれぞれ大きく違うと考え、登録時に負担をかけないように Web から登録する方法をやめた。それぞれの営業マンが、紙の申込書を希望する取引先へ持っていき用紙に記入していただきこちらのほうで入力し、会員登録と同時に ID とパスワード、メールアドレス、氏名住所等が書かれた申し込み用紙を作成する。それを印刷して郵送するという方法にした。本人の確認には、メールアドレスへ認証のメールを登録時に送るように構築する。

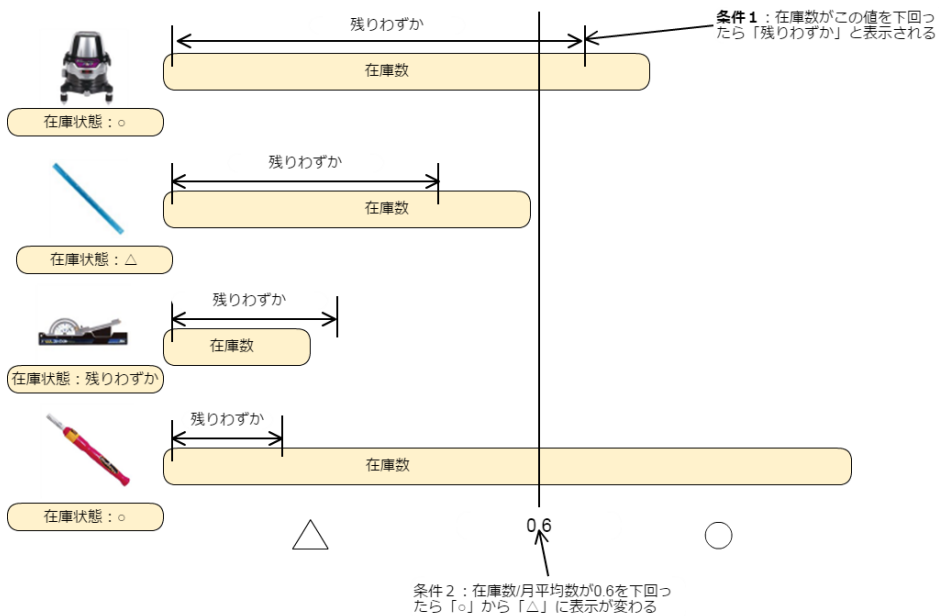
図2 ユーザ登録の仕組み



#### 4.2 在庫数表示の工夫

在庫数を直接表示してしまうと、買占め等が起こるため控えたほうが良いという意見があった。○△×での表示に変更し、直接在庫数がわからないよう変更する。製品によっては月の平均販売数が大きく違い、10個程度しか売れないものもあれば2000個以上売れるものもあるため、50個以下は△といったような一定の数字で区切る表示にしてしまうと常に△が表示されてしまう製品も出てくる。在庫数を月平均で割った値が0.6を下回る場合は、表示を○から△と表示するようにし、月の平均販売数や一度に売れる量を参考に在庫数が一定の値を下回ったものは「残りわずか」を優先して表示するようにした

図3 在庫の状態表示のしきい値について



#### 4.3 在庫検索時の工夫

在庫問い合わせの電話の内容を確認すると、各製品の番号のみを告げられそれに対して在庫数のみを答えていくといったやり方を行っている取引先がいくつかあることが分かった。対応させるために製品の番号を入れ、エンターキーを押していくと次々に在庫の状態を確認出来る画面を作成する。Ajax を使い、読み込みごとに画面が更新されないようにした。

図4 製品コードのみの在庫状況の検索

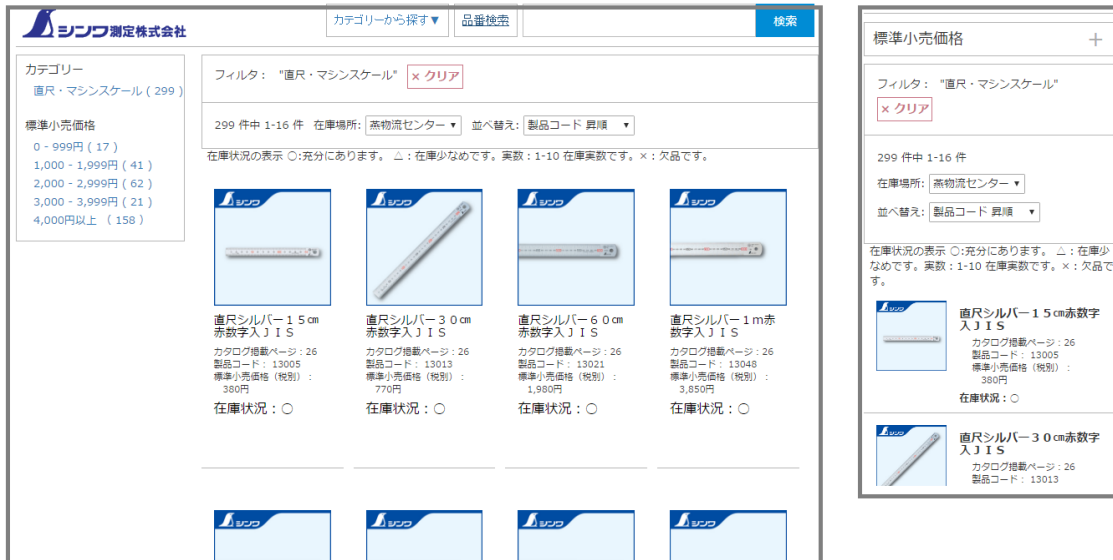
製品コード(品番)	在庫状況	製品名
13064	在庫状況：○	直尺シルバー 2m赤数字入 J I S
13072	在庫状況：×	直尺シルバー 3m
13056	在庫状況：○	直尺シルバー 1.5m赤数字入 J I S

#### 4.4 レスポンシブデザインの対応

どこからでも在庫を確認することが出来るという状態を作るために、レスポンシブデザインを採用した。レスポンシブデザインでは、端末の幅によって表示される項目が最適化されるため従来のようにパソコンサイトやスマートフォン専用サイトを作成する必要がなく工数を減らすことが出来る。



図5 レスポンシブデザインに対応



## 5 評価

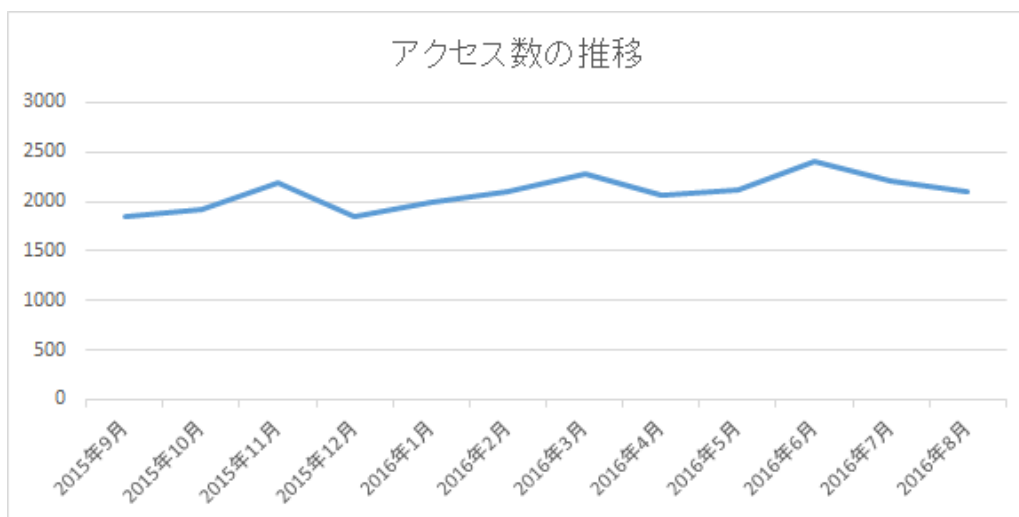
### 5.1 開発の効果

ASP.NETMVC を利用することで、当社では初めての Web アプリケーションを作成することができた。フレームワークでは、MVC モデル（モデル・ビュー・コントローラー）に従い、記述する言語が明確に分かれ、ASP.NET にあるような独特な記述が少ない点も開発を行いやすい点であった。また社内の専用 DLL も違和感なく使用することができた。

### 5.2 アクセスの推移と効果

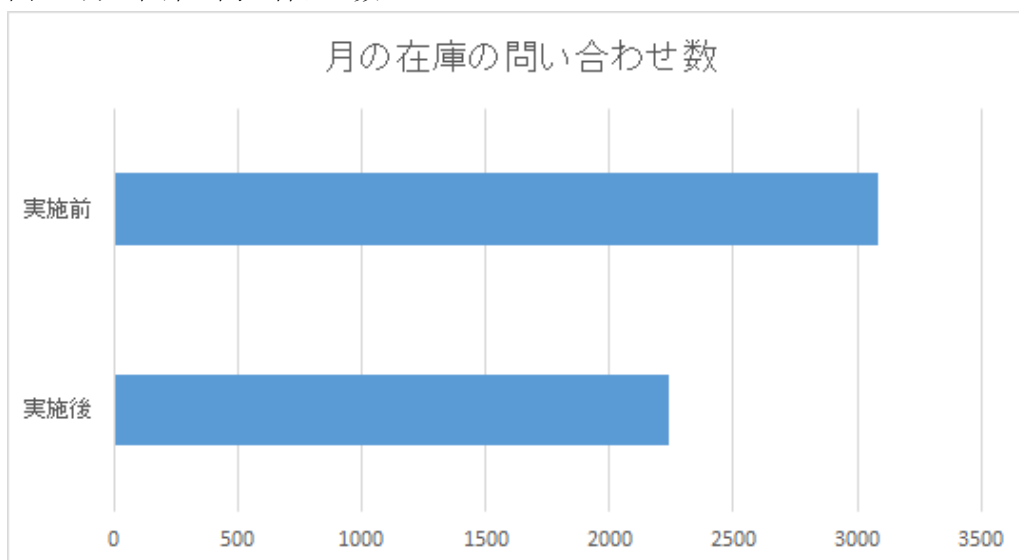
当システムでは、お取引先様から使いやすさを実感していただき電話の問い合わせを減らしていただくことを目標に考えている。月ごとのアクセスの推移は図のようになり、一定のユーザには何度も使っていただいていることが分かった。

図6 アクセスの推移



各営業所への問い合わせについては導入前が3083件あり導入後では2239件となり、三割減となった。個別のアクセス状況を見ていると多く利用していただいているところと、そうではないところに大きく分かれた。取引先の状況によって差があるのを感じる。

図7 月の在庫の問い合わせ数



## 6. 課題と所感

### 6.1 開発の課題

開発の多くをフレームワークに頼っているため、急な仕様変更などでプログラムが動かなくなっただけの対処が難しく感じる。Web の技術は変化のスピードがとても早いため継続的な学習が必要になってしまった。当社のような中小メーカーでは人員も限られるため、継続的に学習していけるかが課題となる。

在庫を電話しなくても手軽に確認出来るようになったのは良いという意見が多かったが、電話のままの方が手軽だということもありサービスを使っただけでない取引先もある。どのようにしたら使っただけなのかを、考える取り組みが一層必要だと思われる。また、直接購入することができたら良いという意見も多くあるが取引先によって納入価格等が違うなど課題があるためこちらも慎重に導入を考えていきたい。

### 6.2 所感

ASP.NETMVC を利用したシステム開発は、.NET を使用して社内のシステムを作成している中小メーカーには一番合うフレームワークだと感じた。また、弊社の新しい取り組みは、お取引先様に受け入れられたように感じた。これからも Web を利用した新しいサービスを生み続けるために努力をしていきたい。