

PDCAサイクル

1. PDCAサイクルの意味と重要性

(1) PDCAとは

PDCAとは「Plan(計画)」、「Do(実行)」、「Check(評価)」、「Action(改善)」の頭文字をとったもので、業務の効率化を目指す方法の1つです。

日本では1990年代後半からよく使われるようになった方法で、計画から改善までを1つのサイクルとして行います。



(2) 目標を確実に達成するために重要となる

計画を単に実行しただけでは、問題点が改善されることはありません。

PDCAで重要なのは、計画を実行した後、成功しても失敗しても必ず評価を行い、改善につなげていくことです。

そして、PDCAが威力を発揮するのは、むしろ失敗した時だとも言えます。



掲げた目標の数値を達成できなかった場合、新たな施策を考え直す必要があります。

目標を達成するための施策は、

**手当たり次第やってみたり、直感的に動いてみることで
打ち出せるものではありません。**



**どのような施策を講じた時に「成功」と呼べる結果が出るのか、
PDCAサイクルを回すうちに最適解へと近づいて行くことができます。**

**目標達成のためのPDCAサイクルを回し続けることによって、次第に
失敗(目標の不達成)は回避できるようになるというメリットもあります。**

(3) 定期的にサイクルを回すことが重要である

PDCAは1度で終わりではありません。
「PDCAサイクル」といわれるように、何度も実行し、改善された計画を繰り返し評価することで、どんどん精度が高められていきます。定期的にPDCAサイクルを回すことで、精度の高い計画を練ることができるのです。



【PDCA サイクル】

2. PDCAプランを設計する前に明確にすべき2つのこと

(1) 目標の明確化

このPDCAを行う前に明確にしておかなければならないことがあります。

それは「目標」です。

目標が明確にされていないと、評価するときどの角度から評価してよいかがあやふやになってしまう可能性があります。

せっかくPDCAを導入しても、結果的にあまり意味がないものとなってしまいます。

① 目標は売上か？純利益か？

たとえば、ある商品の売上金額が上がり、純利益が下がった時、目標が「売上金額を上げる」というのであれば、この計画は成功です。

しかし、企業としてその計画で「純利益を上げる」ということが目標であった場合は、この計画は失敗となります。

そして



純利益を上げるために経費を削ってしまうと、売上金額は下がってしまうかもしれません。

② 大幅に変わってくるアクションプランとサイクル内容

目標の定め方によってその計画アクションプラン自体が大幅に変わってきます。

評価基準や改善方法も全く変わってしまいますので、PDCAに取り組むときは「何のために行っているのか」という目標をしっかりと設定しておく必要があります。

(2) 期間の明確化

期間を設定することにより実現性を高めることもできます。
「1年後までに現在の売り上げを20%上回る」といった目標設定により、必要な行動を逆算して考えることができます。
改善までの時間短縮が可能となり、競合に対して先手を取れる可能性もあります。



ゴールを明確にし緊張感を持たせることで、PDCAを効果的に回せるようになります。

3. PDCAサイクルを効果的に回すポイント

効果を生み出す重要なポイントは、ゴール・目標を達成する為のPDCAになっているか、という点です。

日々のルーチン業務をPDCAに落とし込んでも、成果は得られにくいものです。



(1) Plan:計画

目標を達成するために、できるだけシンプルで実現性が高いものにします。



達成までの期限が決まっているのであれば、タスクや行動などを整理し、スケジュールに落とし込んでいきます。

重要なのは



- ① 達成するための行動
- ② リソース、スケジュールなど



(2) Do:実行

まずは、なるべく計画通りに実行する。
重要なのは、あとで評価・分析(Check)できるように活動内容を必ず残すことです。



計画通りにいかなかったことや、発生した課題も記録に残す。
(記入者によるバイアス(先入観)がかからないように工夫する)



的確な指標が汲み取れるよう、体制を工夫しましょう



(3) Check:評価

計画通りに進んでいるのか、目標がどの程度まで達成できているのかを評価します。
計画の段階で設定した指標を元に、客観的な数値で判断します。



良かった点と悪かった点を客観的に定量数値などで分析し、
どうしてそうなったかという要因を振り返りましょう。



その数値になった因果関係を明確にし、どのようにすれば
改善(Action)につながるのかを議論することが重要です。



(4) Action:改善

目標達成を行うための施策の精度を高めましょう。
評価を見ながら、良かった点は継続的にいき、悪かった部分はどうのように改善すべきかを考えます。



それらの選択肢のなかから、最も実現性の高い選択を行い、計画(Plan)につなげていきます。



見込みがないものに対しては、計画そのものの修正や中止などの検討・判断も必要です。



4. PDCAを効果的に回せない時に陥りがちなパターン

PDCAサイクルは途切れることなく、迅速かつ効率的に回すことがとても重要です。

しかし、実際の企業ではうまく回っているところもあれば、なかなかうまく回らない所も当然存在します。

- ① 実現不可能な計画のため、実行できないパターン
- ② 実行したが、評価で止まってしまうパターン
- ③ 評価を正しく行えず、適切な改善ができないパターン
- ④ 改善を計画につなげられず、循環しないパターン

PDCAサイクルを高速回転できる組織は強くなります。



まずはサイクルを回せる体制を確立しましょう。

(1) ありがちな失敗例と改善策

- ① PDCAサイクルが途切れる → どれかが抜けているので
漏れのないサイクルを実行

きちんと行っているつもりでも、PDCAサイクルが途切れてしまっていることもよくあります。



「評価しただけで改善はしていない」パターンや「評価が適切でなく、改善に失敗した」といった場合もありますので、抜け漏れがなく、効果を発揮するPDCAサイクルを心がけましょう。

抜けや漏れのないようにするために

- 評価を行う人材が適任者であることが重要なポイント

- * 評価(Check)は客観的に行うこと
- * 客観的で科学的、そして一連の業務に精通している人物が評価する



業務を管理する立場の人材が客観的に評価することが望ましい。

**② 目標設定が高すぎる → 計画倒れになってしまう場合は
ハードルを下げる**

最初は、適切なものに設定し、実現可能なものから始めるとよい。
実行する中で、必要のないものの切削や適正値への修正は、思った以上に重要になってきます。

③ やりっぱなし、1度しか回さない → 継続的に繰り返す

一度のサイクルでは気づけなくても、何回か繰り返し行うことで
気づく問題点もあります。
PDCAは評価、改善を繰り返し行うことで高い効果を得るものなので、やりっぱなしや、1度かぎりではなく、継続的に行いましょう。

5. PDCAを回す為の「具体的ToDo」への落とし方

(1) ただの報告会ではなく生きる会議体へ

捗報告は、ただの報告会になってしまうとなんの生産性向上もなく、逆効果です。

次回の計画に結びつけるためのCAP(評価→改善→計画)会議にすることが重要であり、仕組み作りとその為の会議体を策定しましょう。

(2) 課題分解や重要度/緊急度の概念

ただなんとなくタスクについてのPDCAを回すだけでなく課題の分解や「重要度」や「緊急度」も加味しましょう

6. PDCAの成功が仕事の成功にもつながる

(1) 無駄な時間は最小限とし、価値のある時間にフォーカスを

PDCAを効果的に活用することによって、さまざまな物事を改善することができます。

しかし、PDCAを行うこと自体を目的にしてはいけません。

PDCAはあくまで物事を効率的に行い、企業の業績を上げるなど、設定した目標を達成するために行うためのものです。



ツールをうまく活用して、現代に合った体制を模索しましょう。

(2) 多くの業務にPDCAを適用し解決を試みよう

PDCAは、多くの業務に対して適用され、この仕組みが最適化されればされるほど業務の効率とクオリティが上がっていきます。人や環境によって課題やその要因は多岐に及びますが、PDCAサイクルを回していくことで解決に導けます。



課題解決に悩んでいる企業はぜひ取り入れてみてください。

【 まとめ 】

無駄な時間は最小限にし、価値のある時間にフォーカスを

PDCAを効果的に活用することによって、さまざまな物事を改善することができます。

しかし、PDCAを行うこと自体を目的にしてはいけません。

PDCAの目的は

- ① 物事を効率的に行うため
- ② 企業の業績を上げるため

など、設定した目標を達成するために行うためのものです。

各々がいろんなツールをうまく活用して、現代に合った体制を模索しましょう。

多くの業務にPDCAを適用し解決を試みよう

PDCAは、多くの業務に対して適用され、この仕組みが最適化されればされるほど業務の効率とクオリティが上がっていきます。

人や環境によって課題やその要因は多岐に及びますが、PDCAサイクルを回していくことで解決に導けます。

END