

『100分de分かる品質の作り込み』

目次

はじめに p 1

1. バグの正体 p 2

2. 『品質の作り込み』とは何か p 3

(1) 不具合が出る前にその芽を潰す p 3

3. プロジェクトの成功と失敗 p 4

(1) プロジェクト成功の指標 QCDについて p 4

(2) ものごとの後先をわきまえよ～優先順位認識問題 p 5

(3) 失敗プロジェクトの原因分析結果 p 6

(4) 失敗にいたるまでの地獄の一里塚 p 7

(5) 多重請負構造がもたらす丸投げと組織力の衰退 p 8

(6) プロジェクトの成否は、見積りと要件定義の品質によって決定づけられる p 9

4. 品質の作り込みに必要な活動 p 10

(1) 妥当な見積りによる妥当な開発期間および開発費用の獲得を p 11

(2) 要求仕様の早期凍結を p 11

(3) リーダーが開発仕様の全体像を押さえておくこと p 12

(4) 開発メンバーに対して適切な仕事の配分を p 12

(5) 開発メンバーに仕事内容を十分に理解させておくこと p 13

(6) リスクの解消および改善活動の実行を p 14

5. 不条理な顧客要求 p 15

(1) 最初から間に合っていない納期 p 15

(2) 不条理な要求への対抗策 p 16

6. 品質について p 17

(1) 製品品質と業務品質 p 17

(2) 製品品質 (モノの品質) について p 18

(3) どこまでも譲れるものではない p 19

(4) バグが出なければ、それで良いと言うわけにはいかない p 20

(5) 儲かりますか、その製品で p 21

(6) 業務品質 (ヒトの品質) について p 22

7. リスクについて p 23

(1) リスク回避の失敗～あなた自身がリスクなのかも p 23

(2) リスク化してしまった人たち～リスクの自覚 p 24

8. カイゼン活動のすすめ p 2 5

- (1) 業務品質（ヒトの品質）を上げるためには p 2 5
- (2) 仕事に対する認識 p 2 6
- (3) 何をすれば、それが実現できるのか～私たちが実行したカイゼン活動 p 2 7
- (4) 開発プロセスの改善 p 2 7
- (5) 要件定義書の精度向上 p 2 9
- (6) カイゼン活動の実施項目一覧 p 3 2
- (7) カイゼン活動の成果 p 3 2
- (8) 簡単には実行できないカイゼン活動 p 3 3

さいごに p 3 4

参考 p 3 5

『100分de分かる品質の作り込み』

はじめに

現実のプロジェクトを成功に導くためには、成功の方程式を探すよりも、失敗に学ぶ方法が最も効果的だと感じています。PMBOKやCMMIは成功の方程式ですが、その通りにやろうとすると挫折する人やプロジェクトが多いのも事実です。これらの成功の方程式の知識を持ちつつも、現実の取り組み方は、「二度と同じ失敗はしない」という振り返り行動を行う、つまり「成功させる」ではなく「失敗しない」というアプローチの方法が結果として成功に一番近い道だと感じています。

これからお話しする内容は、下記の目次に示した通りですが、話の流れとしては、そもそも『品質の作り込み』とは何か、ということを中心に、『プロジェクトの成功・失敗』の分析から品質が実際にはどのように位置づけられているのかを把握し、さらには『品質の作り込みに必要な基本的な活動』にはどのようなものがあるのかを発見し、続いて、QCDの失敗に大きな影響力を与える『顧客の不条理な要求』について触れ、続いて、そもそも『品質とは何かに』についてもう一度よく考え、さらに品質の発生源である『リスク』は何かを明らかにし、結局、『品質の作り込み』にはリスクの解消行動である『カイゼン活動』が非常に有効であることを明らかにしたいと思います。

目次大項目

1. バグの正体
2. 『品質の作り込み』とは何か
3. プロジェクトの成功と失敗
4. 品質の作り込みに必要な活動
5. 不条理な顧客要求
6. 品質について
7. リスクについて
8. カイゼン活動のすすめ

1. バグの正体



◎バグはバグのような顔をしてはいない

コードは書かれたように動くだけ、バグはそれを書いた人間の方にある

◎結合則の怪

何故か、虫のないものと虫のないものを結合しても、やはり虫のあるものになる。

◎バグはプログラムにだけあるのではない

あいまいな要求仕様書はドキュメントのバグ、ドキュメントのバグはそれを作成した人の業務バグ

ソフトウェアの品質といわれて、最初に思いつく言葉は“バグ”です。不具合と言ったり、インシデントと言いつくってもバグはバグなのです。最初にバグの奇妙な特徴について話をしたいと思います。

◎バグはバグのような顔をしてはいない

バグはバグのような顔をしてはいません。コードは書かれたように動くだけ、バグはそれを書いた人間の方にあります。コードは無色無臭で、バグは最初からバグのような顔をしてはいません。元は正常なコードであっても移植されることによってバグに変身することもあるでしょう。コードもドキュメントも単純にコピーをすれば正常なものが異常に、異常なものはやはり異常に移植されることでしょう。コードにしても設計書にしてもコピーはバグの拡大再生産の行為だと思った方がいいでしょう。

木下恂氏はその著書『ソフトウェアの法則』（p223）の中でバグについて次のように語っています。

【結合則の怪】

虫のあるものと虫のあるものを結合すると、虫のあるものになる。

虫のないものと虫のあるものを結合すると、虫のあるものになる。

虫のないものと虫のないものを結合すると、やはり虫のあるものになる。

◎バグはプログラムにだけあるのではないということ

一般的にはコード上の論理ミス**をバグと呼んでいますが、その真因はそのミスを引き起こしたプログラマーの業務バグであり、遡ればミスのある詳細設計書を作成した設計者の業務バグであり、更に遡ればあいまいな要求仕様書を作成した要求者の業務バグであり、妥当な開発費と期間の確保に失敗した見積り責任者の業務バグだとも考えられます。**

不適切な業務、不十分な業務がバグの本質だと言えます。

2. 『品質の作り込み』とは何か

リスクの解消を!



品質の作り込みとは

- ◎ 不具合が出る前にその芽を潰すということ
- ◎ 品質失敗リスクの事前解消活動のこと

(1) 不具合が出る前にその芽を潰す

『品質の作り込み』とは、不具合が出る前にその芽を潰すということでしょう。

すなわち、不具合の未然防止を図るために、テストや検査で不良を取り除くことに力を入れるよりも、そもそもの不良の発生源を絶つという考え方、および要件定義工程・設計工程・実装工程などで品質確保に有効な方法を計画的に採用する。ということに他なりません。

品質の作り込み活動とは、

= Q I P (Quality In Process) = 開発プロセスの中で品質確保をすること。

すなわち

= 高品質な要件定義・設計業務・実装業務を実行すること。

開発業務自体のそもそもの品質が高くなければ、その成果物としてのソフトウェアの製品品質が高くなるわけありません。

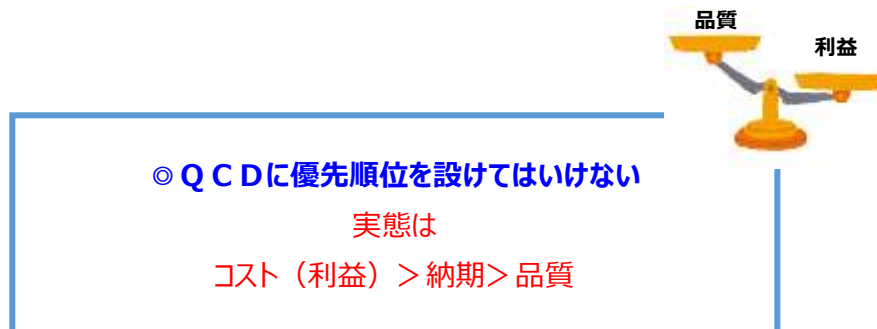
⇒そのためには、それらの業務の失敗リスクの事前解消を行うこと。

品質の作り込みとは要件定義・設計・実装業務を失敗させる可能性のある原因やリスクを事前に排除しておく、ということになります。

品質の作り込みとは、品質失敗リスクの事前解消活動に他なりません。

3. プロジェクトの成功と失敗

(1) プロジェクト成功の指標 Q C Dについて



◎プロジェクトの成功と失敗について、日経 B P 社による 2009 年の「プロジェクト実態調査 8 0 0 社」によればプロジェクトの成功率は **3 1 . 1 %** だったという分析結果が出ています。

(Q 品質は成功 : 51.9%、C コストは成功 : 63.2%、D 納期は成功 : 54.6%)

◎この調査結果から見えることは次の通りです。

開発会社は表向き品質や納期優先だとは言っていますが、多くの失敗プロジェクトにおける Q C D の結果を見る限りにおいて、**その実態はコスト、すなわち利益第一優先**で、見かけ上納期を守ったように装い**品質を犠牲にしている**姿が見えてきます。

すなわち 現実の Q C D の優先順位は **コスト (利益) > 納期 > 品質** の順になっています。

◎これで良いのでしょうか？ / これで良いわけがありません、顧客は怒ります。

製品の品質はボロボロで、これでお客様が怒らないわけがありません。

◎顧客側にとっては、お互いの合意に基づいて開発費を支払ったのだから納期を守り品質問題のない製品の提供を受けることは当然のことです。

◎ Q C D には優先順位などなく、いずれも達成しなければならないものであるという認識が必要です。

(2) ものごとの後先をわきまえよ～優先順位認識問題

Walk, Don't run!

◎優先順位判断の三つの視点

1. 一つの業務内の処理の順番の妥当性の視点
2. 複数の業務の緊急度・重要度の優先順位判断の視点
3. 手持ち時間と必要時間の比較判断の視点

◎限られた時間の中で業務品質を上げるためには**優先順位の判断**が非常に重要になってきます。
ものごとの優先順位の決定にあたって必要なことは、

1. 一つの業務内における**処理の順番の妥当性**の視点、
2. 複数の業務の**緊急度・重要度の優先順位判断**の視点、および
3. **手持ち時間と必要時間の比較判断**の視点、

この三つの視点における判断ミスおよび必要時間の確保の失敗は、Q C Dを大きく毀損する結果を招くことになります。

☆**顧客や上司からの割り込み依頼が常に第一優先とは限りません**

優先順位認識の誤りとして以下のようなものがあります。

- ① 割り込み作業等の優先度判断
- ② 必須業務に優先順位はないという認識の欠如
- ③ 納期第一優先の誤り

☆**間違っても、優しい順とか自分の好きな順にというような情緒的な選択をすべきではありません。**

(3) 失敗プロジェクトの原因分析結果 (プロジェクト実態調査 800社)



◎ Q C D の失敗は要件定義の失敗にある！

【品質問題を起こしたプロジェクト】

1. テストが不十分・移行作業問題：41.7%
2. **要件定義が不十分**：36.7%
3. システムの設計が不正確：31.7%
4. システム開発の質が悪い：31.7%
5. エンドユーザーへの教育不十分：31.7%

【コスト問題を起こしたプロジェクト】

1. **追加の開発**作業が発生：58.9%
2. **追加の設計**作業が発生：47.5%
3. **追加の企画**作業が発生：32.6%
4. テストが長引いた：16.3%
5. ハードの追加購入や高性能化：7.8%

【納期問題を起こしたプロジェクト】

1. **要件定義が長くなった**：43.6%
2. 設計作業が長くなった：33.0%
3. 開発作業が長くなった：33.0%
4. テストが長くなった：26.8%
5. 企画作業が長くなった：16.2%

要件定義の業務品質の悪さがプロジェクト全体に悪影響を及ぼしていることは一目瞭然。

◎ **失敗プロジェクトの原因分析結果**は上記に示した通りです。

これらの分析結果は、いずれにしても**要件定義の業務品質の悪さがプロジェクト全体に悪影響を及ぼしていることを示しています。**

◎ これらから見えるプロジェクトの主な問題点は、

1. **要件定義が不十分であり長引いている。**
2. **システム設計が不正確であり質が悪い。**
3. **追加開発が発生する。**
4. **テストが不十分。**

の4点に集約することができます。

(4) 失敗にいたるまでの地獄の一里塚



◎ドミノ倒れ

要件定義の失敗が設計の失敗を招き、設計の失敗が製造の失敗を招き、すべての失敗ゴミが評価工程になだれ込んでいる。

◎プロジェクトが失敗にいたるまでの経過を、工程順に見てみましょう。

◎要件定義工程において

☆顧客およびベンダーともに下請け依存的な姿勢により要件定義能力が低下し、そのために要件定義工程は所定の期間内に終了できず、さらに要件定義内容も貧弱であり漏れも多い。

◎設計・製造工程において

☆要件定義工程における問題が設計・製造工程においては質の悪い設計・製造や予定外の追加開発を発生させ開発工程を遅延させてしまっている。

◎評価テスト工程において

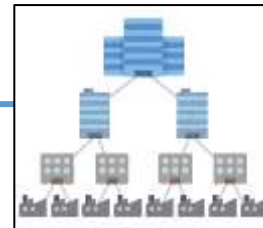
☆これら上流・中流工程の品質の悪さおよび工期遅延は下流工程におけるテスト時間の大幅な不足を招き、結果としてテスト不足の粗悪な品質の製品を納期遅れでしかも赤字でリリースしている。

◎遅れて決まった要件定義内容は、すでに決まっている開発費や開発期間より多くの期間と費用を必要とする場合が多く、コストと時間の前提が崩れたプロジェクトの結果は、時間切れ・工数不足による中途半端な仕事が招く品質低下しかありません。

プロジェクトは開始される前から失敗を運命づけられているようなものです。

(5) 多重請負構造がもたらす丸投げと組織力の衰退

◎ 丸投げはヒト・モノ・カネ・組織のすべてを破壊していく。



顧客・ベンダー・下請け会社間の下請け依存的な関係による要件定義能力の低下は、開発のさまざまな局面においてプロジェクトの真の阻害要因としてその名前を変えて登場してきます。

例えばベンダー側から協力会社への仕事の丸投げが一括請負というもっとうまい名前で実行されています。更にまたオフショア開発という名前で他国にも丸投げされています。発注元は請負側に実務を丸ごと依存し、請負側は発注元に丸ごと売り上げを依存している構図になっているのです。

そのような状況は間違いなく関係組織全体に仕事をやり抜く情熱や責任感の低下を招くことになります。ものづくりの指標を利益第一主義でしか見られないマネジメントの下では開発組織はまちがいに衰退していく運命にあるでしょう。

◎ **丸投げとは**、上位会社が自分の**責務を放棄**し下位の会社に仕事の責任も含めて**押し付ける**行為のことを指しています。無責任な上司が部下に対して仕事を押しつけることも同様に丸投げなのです。

◎ **丸投げが及ぼす害毒**としては以下のようなものがあります。

- ◎ スキルダウン：すなわち人材および組織の能力の劣化
- ◎ モチベーションダウン：すなわちやる気の喪失
- ◎ 品質悪化：すなわちトラブルの多発
- ◎ コスト増加：すなわち無駄な出費の増大
- ◎ 損益悪化：すなわち利益減少・売上げダウン
- ◎ スピードダウン・生産性悪化：すなわちむだな作業・やり直しの多発、効率ダウン
- ◎ 人材の劣化：すなわち専門職の無能化
- ◎ 関係者間の協調関係の破壊：すなわちコミュニケーションの分断・共有分業の破壊
- ◎ 協力者の離反：すなわち代替りの協力者はどこにも居なくなる
- ◎ 組織の崩壊：すなわち職能集団から無能集団への転落

それでも、あなたは丸投げをやりますか？

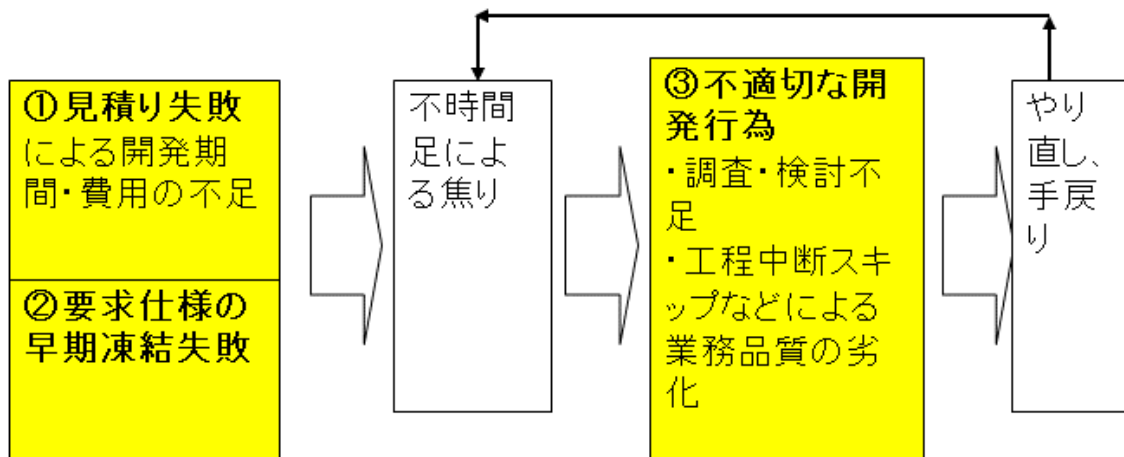
◎ 詳細は、PMファクトリーサイトに記載の『IT組織 失敗の方程式～分断と丸投げの行き着く所』をご覧ください。

http://www.geocities.jp/pmfactory_ambitious/subpage1/shippai_houteishiki.html

(6) プロジェクトの成否は、見積りと要件定義の品質によって決定づけられる



◎失敗プロジェクトは最初から失敗している。



◎失敗プロジェクトの主な原因は、プロジェクトの最初の工程である**見積りの失敗**および**要求仕様の早期凍結の失敗**にあります。見積りの失敗によって必要な開発期間や費用の獲得に失敗し、あいまいな要求仕様や二転三転する要求仕様によって、さらに多くの時間を失ってしまうと、開発チームは間違いなく焦りモードに陥ります。時間不足と焦りは、開発内容の調査不足や検討不足を招き、さらには必要な業務や工程のスキップという手抜きを招いてしまうこととなります。その結果、多数の仕様誤認・機能欠落やバグの修正のために、設計・製造・評価テストなどの手直しややり直しなどが頻発することになります。

◎失敗プロジェクトは、最初から失敗しているのです。

◎図に示した失敗の構図でも分かるように、**品質の作り込み**は、最初の二つの工程である**見積り工程**および**要件定義工程**に集中して実行される必要があります。

4. 品質の作り込みに必要な活動



1. 妥当な見積りによる妥当な開発期間および開発費用の獲得を
2. 要求仕様の早期凍結を
3. リーダーが開発仕様の全体像を押さえておくこと
4. 開発メンバーに対して適切な仕事の配分を
5. 開発メンバーに仕事内容を十分に理解させておくこと
6. リスクの解消および改善活動の実行を

◎品質の作り込みに必要な活動として、6点をあげました。

いずれも簡単な活動ではありませんが、これらの活動の地道な実行は間違いなく業務品質および製品品質を向上させます。

◎それぞれについて見てみることにしましょう。

(1) 適切な見積りによる適切な開発期間および開発費用の獲得を



◎「適切な見積りによる適切な開発期間および開発費用の獲得」は何としてでも達成しなければならないでしょう。失敗すると、プロジェクトは必ず失敗します。

開発に必要な適切な期間および開発費の獲得が、プロジェクトの重大な成功要因となります。見積り時において、要求仕様の骨子および全体像を限りなく正確に把握しておく必要があり、適切な開発費と期間を確保する必要があります。

(2) 要求仕様の早期凍結を



◎見積りと同じくらいに重要なことは、「要求仕様の早期凍結」です。何をつくるのが決まっていなければ開発チームは一步も前に進むことはできません。

何を実現すべきか、ということをごだけ早く明確化できるかがプロジェクトの成功の鍵となります。何を作るべきかが分かっていなければ、我々開発者は一步も前に出ることできません。とりえず決まったところから始めようとするようないい加減な姿勢ではプロジェクトの成功は望めません。顧客ないしはベンダー側から要求仕様が出てくるのを待つような受身的な姿勢は不適切です。仕様検討は、仕様変更管理表をもとに、顧客と同意した仕様凍結日に向けて、顧客価値の重要度の順に仕様決定を行う必要があります。仕様の明確化がその後の開発行為である基本設計・詳細設計・プログラム製造・評価などの全ての業務品質を決定づけることとなります。

◎この二つのことが達成されなければ、残りの4項目の実行もきつと無理でしょう。

◎このような常識的なことがもできていないのなら何故できないのか良く考えてみる必要があります。

(3) リーダーが開発仕様の全体像を押さえておくこと



◎リーダーは要求仕様の全体像を理解しておくこと。

◎ 3番目に挙げたことは、「リーダーが開発仕様の全体像を押さえておく」ということです。
☆顧客との仕様検討に消極的なコミュニケーションが苦手なリーダーにおいては、往々にして仕様の骨子を把握しないまま、部下に仕事を丸投げすることがあります。
そのようなチームにおいては仕様の理解不足によって、見積りの失敗や開発のやり直しなどの重大な失敗を招いています。

リーダーが要求仕様の全体像を理解していることによって、妥当な見積りが可能となり、適切な開発も可能になります。要求仕様の全体像を把握するために、リーダーは仕様検討に参加し、不明点や疑問点を全て解消するよう、要求者と密接なコミュニケーションを実行する必要があります。

(4) 開発メンバーに対して適切な仕事の配分を



◎リーダーは適切な仕事の配分を

◎ 4番目として、「開発メンバーに対する適切な仕事の配分」があります。
☆それぞれのメンバーの実力や性格もわきまえず、おまけに仕様の骨子も把握できないようなリーダーには、仕事の適切な配分ができるわけありません。
無責任な上司やリーダーたちは、今日も仕事を丸投げしており、部下たちを疲労困憊の地獄に落としているのです。このような「ブラック・リーダー」にならないように気をつけたいものです。

公平な仕事の配分を行うためには、部下の能力や性格および成果物を日ごろから良く把握しておく必要があります。同じ量や質の仕事を均等に割り振ることが公平なのではありません。仕事の割り振りは、各人のレベルや適性に応じて配分する必要があり、各人と良く話し合い納得を得た上で仕事をやっていただくことが重要です。また特定の優秀な人に多くの仕事が集中してしまうことはリスク上も避けなければいけないことです。

(5) 開発メンバーに仕事内容を十分に理解させておくこと



◎メンバーの仕事内容の確実な理解のためにはドキュメントによる説明・情報共有が必要。

◎5番目として、「開発メンバーに仕事内容を十分に理解させておくこと」が必要です。

☆仕事の内容が明確にメンバーに伝わっていない場合、まともな仕事が行われるわけありません。多くの職場において、仕事の指示や仕様の内容やその他の伝達事項などについて、今でも口頭によるコミュニケーションがその主流を占めています。

口頭によるコミュニケーションは、多くの誤解や伝達忘れを生みQCDを大きく傷つけ続けています。

◎「エビングハウスの忘却曲線」についてご紹介したいと思います。

☆ドイツの心理学者、ヘルマン・エビングハウスの実験によると、人の記憶忘却度は20分後には42%、1時間後には56%、1日後には74%、1週間後には77%、一ヶ月後には79%であったということです。

それでもあなたは開発における複雑・複数の情報を口頭で仲間に伝えますか。すべからず、ドキュメントベースの仕事を実行すべきです。

部下に仕様を説明する場合は、必ず書かれた資料・ドキュメントに基づく説明を行い、理解不足や誤解を生まないようにする必要があります。仕様の説明においては、最初にその仕様の意味・背景を説明する必要があります。説明の後で、部下にその内容を復唱させ、不明点・疑問点を相互に確認することで、部下における理解度を確認することができるでしょう。

(6) リスクの解消および改善活動の実行を



◎改善活動なしでは品質の作り込みは達成できない。

◎最後に挙げたことは「**リスクの解消および改善活動の実行**」と言うことです。

☆繰り返される失敗について、今度は気をつけようという程度の反省では、また同じ失敗を必ず繰り返すことになります。

リスクの解消、チーム全体の能力向上、生産性の向上を図るために改善活動を組織的に進める必要があります。

プロジェクトにおいて想定される全てのリスクを前もってリスク管理表として登録しておき、プロジェクトの全員でこれらのリスクの排除活動を行う必要があります。リスク項目の代表的なものは過去のプロジェクトにおけるQ C Dの各項目で起こした数々の失敗、特に繰り返される失敗について挙げ、リスクの解消を実行する必要があります。リスクの解消のみならず、チーム全体の能力向上および生産性の向上を図るためにも改善活動を組織的に強力に推し進める必要があります。

5. 不条理な顧客要求

(1) 最初から間に合っていない納期



◎顧客やベンダー側の不条理な要求に負けるということはQもCもDも捨てて無条件降伏をすることと同じ。

◎プロジェクトのQ C Dの失敗に大きな影響を及ぼす**顧客の不条理な要求**について考えてみたいと思います。

◎みなさんは誰が考えてもとうてい無理だと思われる要求をされたことはありませんか。その要求者は、顧客だけではなく自分の上司やその他の関係者も含まれます。

◎顧客や上司からの要求に対しては、なかなか「できません」とは言えないでしょう。実現不可能な短納期・低予算の要求、守られない約束や要求仕様の提示期限などさまざまな納得できない問題に対して、「仕方がない」とか「言っても無駄だ」とあきらめてはいませんか。

インスタントラーメンを煮るのに普通は3分かかります。これは常識。この常識が通用しない不思議な世界があります。どう考えてみても3ヵ月かかる仕事を1ヵ月で納入しろと迫る顧客や営業もいます。権力者の圧力やどうしても受注したいという欲によって、この不条理な要求を受けてしまう開発側責任者もいます。このようなプロジェクトは、必ず失敗しているのを何度も経験しているにもかかわらず、また今日も繰り返そうとしています。実に愚かなことです。

不条理な顧客要求問題の代表的なものは次の通りです。

- ・ 実現不可能な短納期・低予算
- ・ いつまでも提示されない要求仕様
- ・ 仕様説明・提示責任の丸投げ

あなたならどうしますか？

(2) 不条理な要求への対抗策



◎不条理な要求には無理な実行条件の提示を。たとえば、一週間以内の要求仕様の凍結の要求とかプレミアムコストの要求とか、要求元からの強力な人的支援の要求とか。

◎では、不条理な要求に対抗するにはどうしたら良いのでしょうか。

◎たびかさなる不条理な顧客要求やベンダー側要求に悩まされ続けている開発会社は少なからずあることでしょう。しかしながら、いつまでも“仕方ない”とあきらめてばかりいる訳にもいきません。

◎「無理が通れば道理が引込む」と言う通り、無理を重ねた挙句の品質劣化という罰は、結局ブーメランのように不条理な要求を強要した顧客ないしはベンダーに戻ってくることになります。

◎交渉のキーワードとしては、「無理な要求には無理な実行条件の提示」とか「過去の貸しを返していただく」とか「無理を通せば結局共倒れ」とかがありますが、みなさん自身ではどのようなキーワードが思い浮かぶのでしょうか。

◎無理な要求への対抗策は条件闘争が最も効果的です。無理な要求には無理な実行条件の提示を行うことです。「目には目を、歯には歯を」は、物ごとに対する対応の原則のひとつです。

敵を知り己を知れば百戦危うからず～タフネゴシエーション

自分の弱みにばかり気を取られていると敵が見えなくなります。敵の弱みにも注目するところからタフな交渉が可能となります。“All or Nothing”的なデジタル思考をやめ、敵をよく知ることから交渉という戦いを始めたいと思います。

【もの柔らかなお断り】

言葉、口調、しぐさを使って、心から感謝を示し、残念そうに、しかしはっきりと、言い訳せずにノーと言う。将来、別の関係を作るきっかけを示す。(G. M. ワインバーグ、『コンサルタントの道具箱』)

* 参考資料：PMファクトリーのサイトに掲載している『プロマネバトル100戦』を参考にしていたきたいと思います。

http://www.geocities.jp/pmfactory_ambitious/subpage1/battle.html

6. 品質について

◎さて、そもそも「品質とは何か」についてもう一度考えてみたいと思います。

(1) 製品品質と業務品質

◎ **品質には、製品品質と業務品質の二通りがある。**

製品品質とはモノ自体の品質

業務品質とはヒトの品質



- ◎ 品質には、製品品質と業務品質の二通りがあります。
- ◎ 製品品質とは、モノ自体の品質のことです。
- ◎ 業務品質とは、物を作り出した人間の業務行為の品質のことです。
- ◎ 最初に業務品質があり、その結果が製品品質となって現れて来ます。
- ◎ 当然のことながら、業務品質以上の製品品質が生まれることはありません。

製品品質（モノの品質）は業務品質（ヒトの品質）によって規定されてしまいます。ヒトがモノ（ソフトウェア）を作っている限りにおいては、業務品質レベル以上の製品品質の結果は生まれません。

◎ **業務品質以上の製品品質は生まれない。**



(2) 製品品質 (モノの品質) について



◎ **品質の数値化 (指標化) は品質改善の必須要件。**

- ◎ 製品品質を高めるためには、業務品質を高める必要があるということです。
- ◎ 品質の本質を見極めるには、結果ばかりに目を奪われることなく、その原因である業務品質のカイゼンに注目する必要があります。

製品品質とは、モノ (ソフトウェア) の切り口から見た品質であり、開発能力 (業務品質) 以上の製品品質が生まれることはありません。製品品質の主な指標例としては以下のものがあります。

【製品品質の指標例】

- ・単体テスト品質：不具合件数 / 1 Kステップ
 - ・結合テスト品質：不具合件数 / 1 Kステップ
 - ・総合テスト品質：不具合件数 / 1 Kステップ
 - ・市場品質：市場障害件数 / 障害率
 - ・市場障害レベル：重度障害・中度障害・軽度障害
- 重度障害の例：全システムダウン、会計数値違算、全取引 / 発注 / 在庫管理機能不正
中度障害の例：一部システムダウン、一部端末ロック、一部取引 / 発注 / 在庫管理機能不正
軽度障害の例：会計数値に影響しないレベルの印字・表示の不正等

- ◎ 製品品質の一般的な指標例は上記の通りです。
- ◎ 品質改善の出発点は、製品品質の数値化にあります。
- ◎ 数値化することによって、現在の状態の正確な把握が可能になり、さらに改善の目標値の設定も可能になります。
- ◎ スローガンや掛け声だけでは目標の達成は全く無理なことです。

(3) どこまでも譲れるものではない



◎「これくらいは良いだろう」とか「そこまでやる必要はないだろう」とかいうような、
自分勝手な判断は手抜きや丸投げを生み出し、必ず品質に傷をつけることになる。

「スレッシュホールド (threshold) , 閾値 (しきいち) 」という言葉があります。ON/OFF の境となる境界線のことです。この線を超えると ON と認識され、下回ると OFF と認識されます。ON 状態を正常領域、OFF 状態を異常領域としてみると、正常領域にも異常領域にも領域の幅があります。

仕事の業務品質を保つための正常領域における複数の条件を一つずつ外していくと業務品質は段々と低下して異常領域に近づきます。そしてある最後の条件を外した時に異常領域に陥ることになります。この最後の条件だけは外すわけにはいきません。業務品質も製品品質もどこまでも譲れるものではありません。

◎これくらいは良いだろうと思っている内に、品質は段々と悪化していくものです。どこまでも譲れるものではありません。

◎そのような自分勝手な考え方は、社会常識とはかけ離れた悪しき文化に汚染された組織や会社の中で生まれるものです。

◎自分の会社では当たり前と思っていることが、本当に当たり前なのか疑ってみる必要があります。

(4) バグが出なければ、それで良いと言うわけにはいかない



◎ 品質の条件はNOバグだけではない

◎ バグが出なければ、それで良いと言うわけにはいきません。

☆ソフトウェアにおいても、

- ◎ ノンストップ・ノーダウンで連続 24 時間 365 日稼働に耐えるのか。
- ◎ 仕様変更が容易な構造体になっているのか。
- ◎ 容易な操作性を確保するための人間工学的な考慮がなされているのか。
- ◎ 設置方法が簡単であるか。
などが品質とともに作り込まれていて、はじめて顧客満足が得られます。
- ◎ 以下に記述された項目はハードウェアにも要求される品質条件です。

製品品質について RASUI

商品の命は信頼性だけではない

優れた商品の 5 つの条件は RASUI にある。耳慣れない言葉ですが次の 5 つの特性を備えたものが良い商品の条件となります。

- ・Reliability : 信頼性 (不具合・障害率の低さ)
- ・Availability : 可用性 (継続的稼働能力)
- ・Serviceability : 保守性 (保守・メンテのしやすさ)
- ・Usability : 使用性 (使いやすさ)
- ・Installability : 設置の容易性

総合テストは、これらの 5 つの特性検証の場であって、単なるバグ潰しの最終工程ではないのです。

(5) 儲かりますか、その製品で



◎ **懸命に働く、ではなくて、賢明に働く。**

◎ みなさんの仕事は儲かっていますか。

◎ ソフトウェア開発は一応知的な労働とされていますが心身を酷使するような土方的労働に堕ちてしまってはいませんか。成果を出すということは毎日深夜まで働くことではなく、どれだけ知恵を出して創造的な仕事ができただということでしょう。

◎ ソフトウェア開発という仕事は精密機械を作る仕事と同じようなもので知恵と工夫を出せなければ品質守ることも利益をだすこともできないでしょう。利益を出せない仕事はプロの仕事だとは言えません。

「当たり前のことだが、儲かるかどうかは、製品に盛り込まれた技術が高度か否かで決まるのではない。自社以外に同等品を提供できる会社が何社あるかで決まっている。たとえ盛り込まれる技術が高度であっても、同じくらい高度な知識を製品に盛り込める会社が世界中に何社も存在するなら、残念ながら利益は手に入らない。」(沼上幹、一橋大商学部長、2012.03.16 朝日新聞)

(6) 業務品質 (ヒトの品質) について



◎業務品質の筆頭は、プロマネやリーダー自身の業務品質にある。

◎つづいて、業務品質、すなわちヒトの品質について考えてみましょう。

業務品質とは、ヒトの切り口から見た品質であり、プロジェクトチームの開発能力のことを意味するものと言えます。端的に言えばヒトの能力であり、具体的業務能力としてはプロジェクトマネジメント能力、プロセス遂行能力、技術能力、生産性レベルなどを挙げるすることができます。

◎開発チームの業務能力の筆頭に挙げたいのが、**プロジェクトマネジメント能力**です。

◎失敗プロジェクトの原因の8割・9割はプロジェクトリーダーのマネジメント能力不足にあるでしょう。

◎標準的な技術レベルのメンバーがそろっており、妥当なプロジェクトマネジメントにより要求仕様が早期に凍結され、開発実行に必要な期間と費用が確保され、やるべき順に・やるべきことが・やるべき時に実行されれば、そうそうプロジェクトが失敗することはあり得ません。

業務品質の主な指標例としては以下のものがあります。

【業務品質の指標例】

- ・ 仕様品質：仕様ミス件数／率、仕様変更影響度表作成率
- ・ 設計品質：設計ミス件数／率
- ・ 製造品質：製造ミス件数／率
- ・ 評価品質：評価ミス件数／率、バグ発見数／率
- ・ 構成管理品質：構成管理ミス件数／率
- ・ プロセス管理品質：プロセス管理表実施率
- ・ リスク管理品質：リスク対策実施件数／率、リスク管理表実施率
- ・ 課題管理品質：課題対策実施件数／率、課題管理表実施率
- ・ 各レビュー品質：レビュー指摘件数／率、レビュー実施率

いずれの指標もソフトウェア業界全体を網羅できるような統一的基準値で表すことは困難であり、同一組織における過去・現在の実績値の変動値の比較によって判断せざるを得ません。

これらの指標はチーム全体および個人別の把握が必要です。さらにプロジェクトの成否を決定づけるプロジェクトマネージャやプロジェクトリーダーにおける業務品質は、プロジェクトごとにQCDの目標達成率などによって測定していく必要があります。

7. リスクについて

◎ 次にリスクについて考えてみたいと思います。

(1) リスク回避の失敗～あなた自身がリスクなのかも



◎ ソフトウェア開発の三大リスク

1. 見積りの失敗 2. 決まらない要求仕様 3. 開発行動の失敗

◎ 品質の確保とは、すなわち失敗リスクの回避と言うことです。

ソフトウェア開発プロジェクトにおける主なリスク：

◎ **見積りの失敗** ◎ **決まらない要求仕様** ◎ **開発行動の失敗**

◎ この三つが開発行動における失敗の大部分を誘発している。

◎ このリスクはプロジェクトの初期工程に集中している。

◎ ターゲットは、妥当な見積りおよび要件定義の実現にある。

◎ 主なリスク：

- ・ 仕様決定の遅れ
- ・ 短納期
- ・ 未経験分野の開発
- ・ 仕様凍結前の事前着手

◎ 自分および自分のチームにおける過去の失敗に学ぶカイゼン活動を継続すれば、おのずと失敗は減っていくものです。

◎ 科学合理性に基づいた反省をしない者、反省ができない者は、いつまでたっても成功を手にすることはできません。

* 参考資料：『リスクヘッジ クイックリファレンス』（PMファクトリーのサイトに掲載中）

http://www.geocities.jp/pmfactory_ambitious/pmnote/riskhedgeqr.pdf

(2) リスク化してしまった人たち～リスクの自覚



◎リスクはヒトにあり。モノやカネそのものにあるわけではない。

◎リスクはヒトにあり、モノやカネそのものにある訳ではありません。

人は必ずミスをします。自分も含めてミスをしない人間など見たこともありません。

◎失敗や事故の原因について、失敗学の権威である畑村洋太郎は、

「すべての失敗はヒューマンエラー（人的要因が主因となる失敗）である」と言っています。

◎要求仕様を決められないAさんがリスクであり。無理な短納期や開発費のカットをいつも要求してくるBさんがリスクなのです。さらに言えば、そのような事態に対処できない、あなた自身が最大のリスクなのかも知れません。

全ての仕事で最初にやることは、その仕事は何であるのかを知ることからしか始まりません。プロジェクトにおけるリスク回避は、まずそのプロジェクトの特質を十分に“自覚”することにあり、準備段階において情報収集等の処置を講じ、作戦を練り、それに基づいた体制の準備をすることが必要です。仕様要件も決まらないまま開発着手を行うことは、それだけで失敗・敗戦の十分な条件となります。分かっているでもそれを繰り返す人たちは、自分自身が“リスク化”してしまった人なのでしょう。

目的を共有・理解しない人のリスクは高くなります。顧客の求める価値が分からない人のリスクは高くなります。

◎品質の改善にあたっては、「リスクは、人にあり、自分にあり」という新たな認識が必要になって来ます。

8. カイゼン活動のすすめ



◎ 品質の作り込みを実現するのは組織的なカイゼン活動

◎さて今回のテーマである「品質の作り込み」を実現するために必要な「カイゼン活動」について、みなさんと一緒に考えて行きたいと思います。

◎キーワードは、「個人戦ではなく組織戦を」です。

これはリーダーシップの発揮とメンバー同士のノウハウの継承であり、弱い者も強いものも互いに手を携えてプロジェクトをやり抜くことだと確信されます。

(1) 業務品質（ヒトの品質）を上げるためには



◎ 業務品質（ヒトの品質）を上げるためには、凡人たちの自主的な力の結集を

業務品質を上げるためには、プロジェクトマネジメント能力の向上、プロセス遂行能力の向上、技術能力の向上、生産性の向上などが必要になりますが、一体どのようにすればヒト依存的なこれらの能力の向上が実現できるのでしょうか。

われわれの開発という仕事は基本的には個人戦ではなく組織戦です。リーダーシップとチームプレーなくしては決してプロジェクトを成功させることはできません。

ポイントは、秀でた個人の能力に頼ることなく、どうすれば平凡な普通の開発者たちの力を強制によらず自主的に結集させることができるかにかかっています。

(2) 仕事に対する認識



◎ **チーム全員の仕事に対する認識を揃えておくこと。**

良い仕事 = 日常業務 + カイゼン活動

- ◎ まず改善活動に取り掛かる前に、チーム全員の仕事に対する認識を揃えておく必要があります。
- ◎ 高品質・高利益・高生産性を実現するためには、従来の仕事のやり方を習慣的に漫然とこなしているだけでは全く無理であり、仕事の中心にカイゼン活動をすすめる必要があります。

日常業務をこなしているだけでは品質の高い業務や製品を生み出すことはできません。日常業務と合わせて自他の過去の失敗に学ぶカイゼン活動の実行が高品質な業務および製品を生み出すことはトヨタのカイゼンで実証済みであり世界中に知られていることです。

良い仕事 = 日常業務 + カイゼン活動

- ◎ 仕事とは、日常業務とカイゼン活動の二つが同時に行われることです。
- ◎ カイゼンは、仕事のあらゆることに「何故？」という疑問を発することから始まります。「何故？」と思うことが少なくなったら開発者としての能力が劣化してきたと思った方がいいでしょう。

◎ **カイゼン活動の基本コンセプト**は以下の通りです。

- ・個人戦ではなく組織戦（チームプレー）を。
- ・一人だけが成長するのではなく、みな共に成長すること。
- ・能力に長けたものは後進の者にその知恵（ノウハウ）を譲ること。
- ・後進の者もその成長に従って、さらに後に続く後進の者にその知恵（ノウハウ）を譲ること。
- ・先に進んでいる者は、その持てるあらゆる価値あるもの、資産・資金・知恵（ノウハウ）の全てを後進のものに順に譲り渡し、永続的な繁栄の循環を実現させる義務がある。

◎ **カイゼン活動計画のポイント**

- ・問題点の定義を明確に記述すること。
- ・具体的な改善策を数値の裏付けをもって明示すること。
- ・期待成果の目標値を明示すること。

改善の目標値は少なくとも現状よりも10%～30%以上の向上を目指すこと。

(3) 何をすれば、それが実現できるのか～私たちが実行したカイゼン活動



◎ **良い結果は良いプロセスからしか生み出されない。**

◎最後に、カイゼン活動の実例を示します。

皆さんに語れることは、私たちが実際に実行し体験した事しかありません。うまく行ったものもあれば、うまく行かなかったものもありますが、全体としてみれば大きな成果を獲得できたと自信をもってお話できます。

私たちが最も苦しんでいた問題を解消するために、最初は数人でカイゼンチームを結成し、活動を始めました。

最初に取り組んだことは、『開発プロセスの改善』

最初に取り組んだことはプロセス管理表の作成とその実行でした。

◎何故プロセス管理に目をつけたのかと言うと、仕事と言うものは「段取り7分の後3分」と言われているように、われわれの過去の失敗は準備の悪さや実行の順番の不適切さに起因する問題が多かったからです。

多くの品質問題の本質は、仕事の段取りの悪さにあります。段取りとは、すなわちプロセス管理のことです。

(4) 開発プロセスの改善

開発プロセス改善計画書の抜粋は下記の通りです。

◎プロセス管理における改善すべき問題点

- ・成果物を自分の目で確かめていないこと。
- ・各工程の実行手順忘れ。
- ・各工程の手抜きの防止。
- ・人依存・経験依存のバラバラな開発手順。
- ・視覚化されていない開発手順。

◎改善策

- ・標準プロセス管理表の作成（開発業務手順のビジュアル化）
- ・標準プロセス管理表に基づく開発手順の実行（開発業務品質の確保）
- ・標準プロセス管理表実行のレビュー
- ・プロセス管理表にI S O・C M M I 項目を絡めたチェック機能の強化

◎改善の期待成果

標準化されたプロセス管理表による全工程の手順および成果物の現物確認

プロセス管理表の一部は次の通りです。縦軸には、時系列順に主要な開発イベントを記述し、横軸にはそれぞれのイベントで実行されるべき作業内容・担当者・作成物・注意書き、チェック欄、実施予定日、実施日、レビュー結果などが記述されます。

【プロセス管理表】

【プロセス管理表】(サンプル)		*PL (or PM) , SPL (or SPM)									
	NO	手順	進捗	作業	担当者	成果物	開始 チェック	PL チェック	実施 予定日	実施日	レビュー 結果
事前準備	1	顧客・ステークホルダーの情報収集		顧客の要求内容・予算・納期等の事前把握	開発部	・調査報告書					
	2	新OS・新言語・新システム対応の準備		新ソフトウェア・新ハードウェア等に対する技術知識のマスターおよびプロトタイプによる事前調査	開発部	・調査報告書					
計画プロセス	1	見積もり依頼 見積もり回答		仕様、スケジュール、要員検討、見積依頼	PL, SPL	・見積回答書 ・リスク管理表					
	2	要求仕様の明確化		要求事項の明確化	PL	・要求事項の明確化チェックリスト作成					
	3	仕様検討、承認		SEとの仕様打ち合わせ	SE, PL, 開発メンバー	・通事録 / 要求仕様書の承認					
	4	不具合防止策の選定、決定		開発に対する有効な不具合防止策を	PL, SPL, 開発メン	・不具合防止策チェックリスト					
	5	スケジュール		大、小日程の作成	PL, SPL	・大日程表、小日程表					
	6	プロジェクト品質目標の作成		プロジェクト内品質目標の作成	PL, SPL	・プロジェクト品質目標					
	7	プロジェクト計画書		プロジェクト計画書の作成 リスク管理表更新	PL, SPL	・プロジェクト計画書					
				プロジェクト計画レビュー 開発開始会議	SQAグループメンバー、 PM, PL	・レビュー記録					
	8	外注見積委託作成		外注見積委託DBに登録	PL, SPL	・見積書					
	9	開発管理		開発管理資料の作成	PL, SPL	・プロセス管理表 ・スケジュール表 ・課題管理表 ・進捗進捗					
10	デザインレビュー		要求事項の明確化チェックリスト	SE → 開発部	・要求仕様の採点確定						

(5) 要件定義書の精度向上



◎ 要件定義書の精度は見積りおよび開発行為そのものに重大なインパクトを与える

以下は、「要件定義書の精度向上」に関する改善計画書の抜粋です。

◎ 改善すべき問題点

仕様変更の基本的要件の記述モレによる開発生産性および品質の低下。

- ・要求仕様の目的の記述モレ
- ・要求仕様の背景の記述モレ
- ・要求仕様の概要の記述モレ
- ・要求仕様の運用説明モレ
- ・要求仕様の不正確な記述

◎ 改善の期待成果

要件定義書の精度向上による見積り精度向上、設計書等の精度向上

◎ 要件定義書ガイドラインの目次を添付に示しました。

このガイドラインの本文は、要求仕様書に何をどのように書くべきかを示したものとなっています。
ご自分の組織に合ったガイドラインの作成が必要です。

【要件定義書ガイドライン】

【要求仕様書 記述項目】			開発要件 必要項目	見積要件 必要項目	今回提出 の項目	
表紙						
変更履歴						
1. システム企画・目的				*		
2. システム概要	2.1 システム構成図	2.1.1 全体構成図		*		
		2.1.2 店舗機器構成図		*		
		2.1.3 ネットワーク図				
	2.2 ハードウェア構成					
	2.3 ソフトウェア構成					
	2.4 コード体系					
3. 要求機能概要 (要求機能を漏らさず箇条書きにする)	3.1			*		
	3.2			*		
	3.3			*		
	3.4			*		
	3.5			*		
4. 運用フロー						
5. 要求機能説明	5.1 要求機能詳細 (開発項目毎に要求仕様の詳細説明をする。)	5.1.1				
		5.1.2				
		5.1.3				
		5.1.4				
		5.1.5				
	5.2 入出力情報	5.2.1 入力画面				
		5.2.2 入力モニター				
		5.2.3 出力画面				
		5.2.4 出力リスト				
		5.2.5 レシート				
	5.3 その他	5.3.1 集信データ				
		5.3.2 配信データ				
		5.3.3 オンラインフォーマット				
5.3.4 メモリーバランス						
6. 非機能説明	6.1 データ要件	6.1.1 データ件数		*		
		6.1.2 データ保有期間		*		
	6.2 要求性能			*		
7. 条件・制限事項	7.1 運用条件			*		
	7.2 制限事項			*		
	7.3 信頼性					
	7.4 拡張性					
	7.5 セキュリティ					
8. 概略スケジュール	8.1 稼働スケジュール					
	8.2 展開スケジュール					
9. その他	9.1 従来システム構成					
	9.2 従来システムからのデータ移行					

要件定義書の精度は見積りおよび設計・製造・テストに重大なインパクトを与えるため要件定義書精度の評価を行う必要があります。

開発部内で要求仕様書（要件定義書）の精度に対する評価基準を設け、それに従った採点を行います。採点結果については合格の基準点を設けておき、それに満たないものは理由を明記の上、要求仕様書作成者にフィードバックし、修正・追記を行っていただく必要があります。

要求仕様書の採点法については、ある程度の妥当性および合理性が確保されていなければ、採点表自体の信頼性が失われてしまいますので注意が必要です。

◎要件定義書の出来栄を評価した採点表のサンプルは次の通りです。

【要件定義書採点表】

要求仕様書 採点方法								
ユーザー名：XXXXX プロジェクト名：○○○○○				開発担当部署：XXXXX 採点者：XXXXX				
0：記録なし 1：仕様としては不十分 2：多少不明点はあるが内容はわか 3：要件として十分								
0：なし 1：あり								
*記量が0.0点未満の場合、基本的には見積不可とする。 *記述内容得点が0.0点未満の場合、基本的には見積不可とする。								
項目	分類	大項目	中項目	要求仕様内容	記録必須項目	記録の有無	記録内容得点	備考
1		1	1	表紙・変更履歴・目次	1	1	3	
2	目的	2	2-1	システム化の理由	1	1	3	
3			2-2	システム化前後の運用	1	1	3	
4			2-3	システム化のメリット	1	1	3	
5			3	3	システム構成図	1	1	3
6	概要	4	4	ハード構成				
7		5	5	ソフト構成	1	1	2	
8		6	6	運用一覧				
9	運用	7	7	運用スケジュール	1	1	3	
10		8	8	概要ビジネスフロー（運用フロー）				
11	機能	9	9	業務一覧	1	1	3	
12		10	10	ビジネスルール定義				
13		11	11	詳細ビジネスフロー				
14		12	12	画面一覧				
15		13	13	画面設計		0		
16		14	14	画面項目説明				
17		15	15	帳票一覧				
18		16	16	帳票設計				
19		17	17	帳票項目説明				
20		18	18	機能定義書	1	X	X	
21	19	19	画面遷移図					
22	20	20	要求データ一覧					
23	21	21	要求データ仕様書	1	1	2		
24	22	22	オンライン一覧	1	1	2		
25	23	23	要求性能	1	1	2		
26	24	24	その他					
合計					12	11	29	

項目	記録の有無	記録内容得点
両点ポイント	12	36
得点ポイント	11	29
得点 / (100点満点係数)	92	81

記録内容満点 = 12 × 3 = 36点
 記録内容得点 = (29 / 36) × 100 = 81点
 記載率

(6) カイゼン活動の実施項目一覧

カイゼンチームが実行したその他の改善項目の一部を以下に示します。

1. 要求仕様の変更管理
2. 仕様変更影響度表の導入
3. 見積りリスクの回避
4. プロジェクト・リスクの低減
5. コミュニケーションの活性化
6. 無駄の排除
7. レビュー・チェックシートの運用
8. ラップアップの実行

* 参考資料：『カイゼン活動カタログ』（P Mファクトリーのサイトにて掲載中）

http://www.geocities.jp/pmfactory_ambitious/pmnote/kaizen.pdf

(7) カイゼン活動の成果

数年間に渡って実行された開発全組織をあげてのカイゼン活動の成果は以下に示した通りです。

1. Q C D 数値の大幅な改善
2. チームメンバーの技術能力の向上
3. チームリーダーのプロマネ能力の向上
4. コミュニケーションの活性化
5. 全員の自律・自立能力およびモチベーションの向上
6. 改善活動が開発業務の中核業務として実行されたこと。

これらの成果は実証済みのことですが、継続して取り組むためにはそれ相応の努力と情熱が必要です。

(8) 簡単には実行できないカイゼン活動

最後になりますが、カイゼン活動が簡単には実行・継続できない理由についてお話しておきたいと思います。

今を変えたくない

何かに行き詰ったとしても人は今までやってきたことをなかなか変えることができません。例えその行動が非効率だったとしても、それでよしとされる環境にあればいつまでもやり方を変える気にはなれません。しかし、その問題が限界点を越えたときにそれが問題として立ち現れますが、もう後の祭りです。悪い状態に陥らないようにするためには、地道な改善活動を継続する以外に方法はありません。

【現状維持の法則】

「人は、選択肢を前にして迷った場合、新しいものに手を出すことを選択するよりも、今と同じ日常をそのまま送っていくことを選択する。」

(ダニエル・カーネマン、行動経済学者)

この壁を越えられたひとたちだけが、『品質の作り込み』をはじめとするカイゼン活動の果実を確実に手にすることができるでしょう。



最も重要なことは、

1. 開発準備ないしは初期工程のリスクを最小化すること。
2. 品質の作り込みもQCDの目標を達成するのも、あなたたちしかないということ。
3. 個人戦ではなく組織戦を行うこと。

品質の作り込みとは、開発の準備段階ないしは初期工程において大きなリスクをすべて解消する活動のことを意味しています。開発の準備段階や初期工程に位置する主な仕事は、プロセス管理表を初めとするプロジェクト計画の策定、見積り交渉、要求仕様の早期凍結などがあります。これらの初期重要職務のリスクを最小化し品質を高めることが「品質の作り込み」そのものであると言えます。

品質の作り込みを実行するものは、ヒトしかいません。そのヒトは特定の選ばれた担当者だけやリーダー・管理者だけを指すのではなく、開発チームの関係者全員のことを指しています。

「リスクはヒトにあり」ということを常に念頭に置き、個人戦ではなく組織戦に切り、改善活動をプロジェクト活動の中軸に置くことで、品質の向上のみならず利益の獲得、納期の遵守、人材の育成を図ることができます。



カイゼン活動の実行は、

- ◎ 品質の向上
 - ◎ 利益の獲得
 - ◎ 納期の遵守および生産性の向上
 - ◎ 人材の育成
- を実現させる。**

参考

P Mファクトリー ホームページのご紹介

◎ U R L : http://www.geocities.jp/pmfactory_ambitious/

◎ 主な掲載記事

- ・『プロマネバトル 1 0 0 戦』、『今日のチェックリスト』、『プロマネ折々のことば』
- ・『S E 稼業は亡己利他』 ⇒ 技術評論社のサイトで掲載中
<http://gihyo.jp/dev/serial/01/engineer-proverbs>
- ・『I T 組織 失敗の方程式』
http://www.geocities.jp/pmfactory_ambitious/subpage1/shippai_houteishiki.html
- ・『S E プロマネ愚行録』
http://www.geocities.jp/pmfactory_ambitious/pmnote/gukouroku.pdf
- ・『あなたはなぜヒトの話しを聞けないのか』
http://www.geocities.jp/pmfactory_ambitious/pmnote/communication.pdf
- ・『カイゼン活動カタログ』
http://www.geocities.jp/pmfactory_ambitious/pmnote/kaizen.pdf
- ・『リスクヘッジ クイックリファレンス』
http://www.geocities.jp/pmfactory_ambitious/pmnote/riskhedgeqr.pdf
- ・『要件定義リスク クイックリファレンス』
http://www.geocities.jp/pmfactory_ambitious/pmnote/requirement_risk.pdf
- ・『銀の弾丸チェックリスト』
http://www.geocities.jp/pmfactory_ambitious/pmnote/silverbullets_20161015.pdf

◎ P Mファクトリーのホームページには皆さんの参考になりそうな記事を多数掲載しておりますので、関心のある資料などありましたら、どうぞお読みいただきたいと思います。