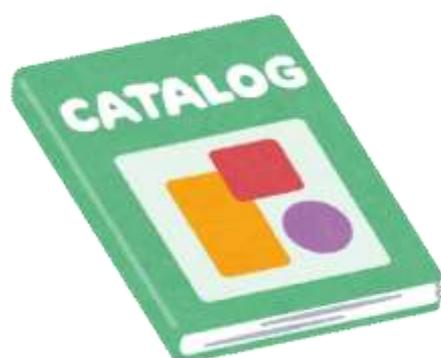


# ITプロジェクト

実行すれば宝の山 実行しなければ負債の山

## カイゼン活動カタログ



PMファクトリー 2016年

## 目次

はじめに	p1
改善 No.1 開発プロセスの改善	p2
改善 No.2 開発準備レビュー・チェックシートの運用	p4
改善 No.3 要件定義書(要求仕様書)の精度向上	p5
改善 No.4 要求仕様の変更管理	p7
改善 No.5 仕様変更影響度表の導入	p8
改善 No.6 要求仕様レビュー・チェックシートの運用	p9
改善 No.7 見積りリスクの回避	p10
改善 No.8 見積りヒット率の向上	p11
改善 No.9 見積り作業生産性の向上	p12
改善 No.10 プロジェクト・リスクの低減	p13
改善 No.11 プロジェクト目標値の設定	p15
改善 No.12 コミュニケーションの活性化	p16
改善 No.13 手抜き問題の撲滅	p17
改善 No.14 無駄の排除	p19
改善 No.15 機能モジュール進捗管理表の導入	p20
改善 No.16 修正残件管理表の導入	p21
改善 No.17 ソフト資産の流用	p22
改善 No.18 初期設計レビュー・チェックシートの運用	p23
改善 No.19 詳細設計レビュー・チェックシートの運用	p24
改善 No.20 コードレビュー・チェックシートの運用	p25
改善 No.21 テスト効率の向上	p27
改善 No.22 単体・結合テストレビュー・チェックシートの運用	p28
改善 No.23 総合テスト準備レビュー・チェックシートの運用	p29
改善 No.24 総合テストレビュー・チェックシートの運用	p30
改善 No.25 構成管理作業チェックシートの運用	p31
改善 No.26 リリースレビュー・チェックシートの運用	p32
改善 No.27 ラップアップの実行	p33

## 改革コンセプト

改革コンセプト#1 時間の確保	p35
改革コンセプト#2 自律型開発	p37
改革コンセプト#3 変化即応型開発	p38
改革コンセプト#4 フォーカス活動	p39

## はじめに

日々の仕事に忙殺されているソフトウェア開発に携わるみなさんにおいては、いま以上の仕事はとてでもないがもう背負いきれないというのが実感でしょう。しかし一方では、問題山積の現状を少しでも改善し、世間の人並みに休日くらいは休みたいとお考えのことでしょう。

もしそのように少しでも改善活動に取り組みたいとお考えの方々のために本資料をまとめました。

カイゼンという言葉は今や世界共通語として知られていますが、本家本元の日本において果たしてどれ位の人々が改善活動に取り組んでいるのかは、昨今のジリ貧状態の日本においては余り期待できません。日本における構造改革が進んでいないと言われていますが、それはカイゼン活動が進んでいないことを意味しています。

本資料は「カイゼン活動カタログ」という名称をつけた通り、ソフトウェア開発の上流工程から下流工程に至るまでの全工程において筆者が経験したカイゼン活動を元に、カイゼンを志している方々に具体的にどのようなカイゼンのテーマがあり、それらはどのような問題をどのようにしてカイゼンし所定の成果を獲得したのかをカタログ的に示したものです。

それぞれのカタログの記述構成は、テーマ名、改善すべき問題点、改善策および改善の期待成果となっています。改善の期待成果の欄には、その改善を実現するにあたって重要な役割を果たしたドキュメントのサンプルをエビデンスとして記載しました。

なおカタログの次に記載した「改革コンセプト」はカイゼン活動を進めるにあたっての基本的な方針を著したものです。

それぞれのカイゼン項目は一過性の改善活動では単発的な成果しか得られず、永続的な組織力の維持向上のためには継続的な活動が必要であることに強く留意していただきたいと思えます。一旦これらのカイゼン活動を中止してしまった開発組織は品質の悪化、生産性の低下、赤字および人心の荒廃を招き、必ず衰退していくことでしょう。

記載の順は基本的に上流工程から下流工程に至る流れとなっていますが、読者のみなさんの興味のあるところから読んでいただいても構いません。いずれにしろ、このカタログの中から何らかのヒントを得てみなさんのカイゼン活動が活性化するならば喜ばしいことです。

**改善 No.1**      **テーマ： 開発プロセスの改善**

- 改善すべき問題点**
- ・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ISO・CMMI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管理【プロセス管理表】

No.	業務	種別	目的	内容	実施	備考	ISO	CMMI	品質	コスト	納期	顧客満足
1	開発プロセス管理表の作成	標準	開発プロセス管理表の作成	開発プロセス管理表の作成	開発プロセス管理表の作成	開発プロセス管理表の作成						
2	開発プロセス管理表の実行	標準	開発プロセス管理表の実行	開発プロセス管理表の実行	開発プロセス管理表の実行	開発プロセス管理表の実行						
3	開発プロセス管理表のレビュー	標準	開発プロセス管理表のレビュー	開発プロセス管理表のレビュー	開発プロセス管理表のレビュー	開発プロセス管理表のレビュー						
4	開発プロセス管理表の更新	標準	開発プロセス管理表の更新	開発プロセス管理表の更新	開発プロセス管理表の更新	開発プロセス管理表の更新						
5	開発プロセス管理表の運用	標準	開発プロセス管理表の運用	開発プロセス管理表の運用	開発プロセス管理表の運用	開発プロセス管理表の運用						
6	開発プロセス管理表の維持	標準	開発プロセス管理表の維持	開発プロセス管理表の維持	開発プロセス管理表の維持	開発プロセス管理表の維持						
7	開発プロセス管理表の廃止	標準	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止						
8	開発プロセス管理表の改善	標準	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善						
9	開発プロセス管理表の検証	標準	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証						
10	開発プロセス管理表の承認	標準	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認						
11	開発プロセス管理表の廃止	標準	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止						
12	開発プロセス管理表の改善	標準	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善						
13	開発プロセス管理表の検証	標準	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証						
14	開発プロセス管理表の承認	標準	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認						
15	開発プロセス管理表の廃止	標準	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止						
16	開発プロセス管理表の改善	標準	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善						
17	開発プロセス管理表の検証	標準	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証						
18	開発プロセス管理表の承認	標準	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認						
19	開発プロセス管理表の廃止	標準	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止						
20	開発プロセス管理表の改善	標準	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善						
21	開発プロセス管理表の検証	標準	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証						
22	開発プロセス管理表の承認	標準	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認						
23	開発プロセス管理表の廃止	標準	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止						
24	開発プロセス管理表の改善	標準	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善						
25	開発プロセス管理表の検証	標準	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証						
26	開発プロセス管理表の承認	標準	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認						
27	開発プロセス管理表の廃止	標準	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止	開発プロセス管理表の廃止						
28	開発プロセス管理表の改善	標準	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善	開発プロセス管理表の改善						
29	開発プロセス管理表の検証	標準	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証	開発プロセス管理表の検証						
30	開発プロセス管理表の承認	標準	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認	開発プロセス管理表の承認						



**改善 No.2**      **テーマ : 開発準備レビュー・チェックシートの運用**

**改善すべき問題点**  
 開発開始を阻害する問題として次のようなものがある。  
 ・開発内容の概要および背景が把握されていないこと。  
 ・公式の開発指示書が発行されていないこと。  
 ・妥当な開発体制が構築されていないこと。  
 ・リスク管理表、課題管理表、プロジェクト計画書、管理表、スケジュール表などのドキュメントの準備ができていないこと。

**改善策**  
 開発開始に必要な不可欠な人・モノ・資金・情報等についての準備が完了しているか否かをフォローするための「開発準備レビュー・チェックシート」を作成し開発準備工程においてレビューを行う。

**改善の期待効果**  
 ◎開発開始の前に準備しておかなければならないものについての欠落を防止することにより、後に続く開発工程のリスクを予防しQCDおよび生産性の向上を図る。「開発準備レビュー・チェックシート」はそれぞれの開発組織の文化や実力に応じて最適化を図る必要がある。

No.	大項目	確認内容	例・補足	確認結果	備考
1	PM仕様	PMはPMの概要をレビューする。  本文書が事前の中立指示書は発行されているか？ (PMが確認する項目) PMが作成するPMに必要な資料の作成状況は？ ①、プロジェクト計画書 ②、プロセス管理表 ③、リスク管理表 ④、課題管理表 ⑤、大日程表および日程表 ⑥、ステータスレポート ⑦、外部委託計画書（協力会社がある場合） ⑧、外部委託ソフトウェア開発計画書（協力会社がある場合） ※その他必要に応じてPMがプロセス管理表を作成。	・存在の概要やプロジェクト発生の経緯 ・物種？ 投機？ ・開発予定は？ ・公式レビューの開催日時や日程	□済 □特有 □不要	
		今現在何をもって課題は何か？ 未解決項目はいつまでに解決するのか目標を立てる。 ①優先すべき項目や時間的にかかるとの項目は注意 ②計画の進捗/管理に記録すべき項目の確認を行う。 進捗していないものは課題管理表へ記録する。		□済 □特有 □不要	
2	開発体制	役割/担当、評価/担当、職務管理者や更新ファイルの作成は明確になっているか？ ※プロジェクト計画書に留意記載し。		□済 □特有 □不要	
3	スケジュール	工務スケジュールの確認を行う。リスクは何か？ 件数/発注/受領/リリース/手探りの確認を行う。 開発メンバーの現在の仕様の作業は何か？ メンバーの進捗作業は何か？ 懸念点/作業による遅れは発生しているか？ スケジュール調整が必要か？ 上記項目確認確認の行われた日を決める。 開発/管理する材料の確認/活用/在庫の記録 <注意すべき事項> サーバー、端末管理、外部機器、カーネル、システム、開発PC 製造/評価/リリースされる結果物を確認する。 仕様が作成完了日を決める。 (開発メンバーに配布する。内容が明確になって、レベルの仕様書。 最終的な成果物であること。) (重要事項) 仕様書が用意されている場合は仕様変更(移行) (→大項目は確認済) 用意されている場合は、仕様変更/変更の履歴日を決める。		□済 □特有 □不要	

Ref. 『要件定義リスク クイックリファレンス 第2章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/requirement\\_risk.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/requirement_risk.pdf)

改善 No.3	テーマ :要件定義書(要求仕様書)の精度向上
---------	------------------------

**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
- ・要求仕様の正確な記述

**改善策**

- ・開発要件を過不足なく網羅した要求仕様書ひな形の作成
- ・要求仕様書ひな形の運用
- ・要求仕様書の採点評価

**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

【要件定義書記述項目表】

要件仕様書記述内容	
【目的】	1. 系統・要素整理-目次 2. システムの企画-目的 2-1 システム化の理由 2-2 システム化前後の業務 2-3 システム化のメリット
【概要】	3. システム構成図
【概要】	4. ワード構成 記述順序を明記
【概要】	5. ソフト構成 記述順序を明記
【概要】	6. 運用一覧 記述順序を明記
【運用】	7. 運用スケジュール 運用条件-運用手順を併せて記述する 上記記載内容にインストール手順-初回準備-メンテナンス項目を盛り込む
【運用】	8. 稼働ビジネスフロー(運用フロー) 運用時-稼働時の運用フローを記載(記述順序含む) 社内業務-業務連携の連携者-連携種別、外部連携-連携種別を記載
【概要】	9. 業務一覧 担当者-依頼処理の名称は、ここで記載 前システムからの業務継承を記載
【概要】	10. ビジネスルール定義 システム利用-運用を規定する 全業務-依頼処理の名称と種別を併せて記述し、ここで運用条件を記載する。
【概要】	11. 詳細ビジネスフロー 運用-稼働業務記載 担当業務担当者(タスク/課室)を全て明記し記載 依頼処理名称について記載 エラー処理について詳細を記載
【概要】	12. 画面一覧
【概要】	13. 画面設計 入出力要件に準拠した記述がガイドラインを参照 依頼処理について注釈(変更時のシステム種別)を併記 入出力フィールド詳細コントロールを明記
【概要】	14. 画面項目説明
【概要】	15. 権限一覧
【概要】	16. 権限設計 ユーザー別権限を記載(権限を子項目で明記)
【概要】	17. 機能項目説明
【概要】	18. 機能定義書 記述順序を明記
【概要】	19. 画面遷移図
【概要】	20. 要求データ一覧 データ要件を明示 各データの要素-属性/内容を明記
【概要】	21. 要求データ仕様書 運用-発生データ内容、運用タイミング、発生範囲、取得タイミングを記載 仕様書項目については注釈(規定/非規定-伝送リークエス-データフォーマット)を併記 外部-会社間連携種別、伝送方式(手続)-伝送リークエス(データ連携種別)/データフォーマットを記載 注釈の補記は、メタデータ項目を記載
【概要】	22. オンライン一覧
【概要】	23. 要求仕様 依頼データの運用時間記載
【概要】	24. その他 業務連携システム-システム連携を記載

【要件定義書採点表】

要件仕様書 採点方法							
ユーザー名:XXXX			開発担当部署:XXXX				
仕様変更:V016.010(XXXX対応)			採点者:XXXX				
0: 記述なし 1: 仕様としては十分 2: 多少不精確はあるが内容はわかる 3: 要件として十分							
*記述が90点未満の場合は、基準時には採点不可とする。 *記述内容得点が90点未満の場合は、基準時には採点不可とする。							
項目ID	大項目ID	中項目ID	要件仕様内容	記述必須項目	記述の有無	記述内容得点	備考
1	1	1	業務・営業履歴・目次	1	1	3	
2	2	2-1	システム化の理由	1	1	3	
3	2	2-2	システム化前後の運用	1	1	3	
4	2	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	4	4	バックアップ				
7	5	5	オフサイト	1	1	2	
8	6	6	運用一覧				
9	7	7	運用スケジュール	1	1	3	
10	8	8	既設システムフロー(運用フロー)				
11	9	9	運用一覧	1	1	3	
12	10	10	システムフロー(定数)				
13	11	11	既設システムフロー				
14	12	12	運用一覧				
15	13	13	既設設計				
16	14	14	既設運用説明				
17	15	15	運用一覧				
18	16	16	既設設計				
19	17	17	既設運用説明				
20	18	18	既設仕様書	1	x	x	
21	19	19	既設構成図				
22	20	20	既設データ一覧				
23	21	21	既設データ仕様書	1	1	2	
24	22	22	オンライン一覧	1	1	2	
25	23	23	既設作成	1	1	2	
26	24	24	その他				
合計				12	11	29	

記述の有無	記述内容得点	記述内容満点
満点ポイント ▼ 12	36	12×3 =36点
満点ポイント ▼ 11	29	
得点(100点満点換算)	81	

採点率 = (11 / 12) × 100 = 92点	記述内容得点 = (29 / 36) × 100 = 81点
-----------------------------------	--------------------------------------

Ref. 『要件定義リスク クイックリファレンス 第3章・第4章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/requirement\\_risk.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/requirement_risk.pdf)  
 『ソフトウェア開発のメソッド クイックリファレンス 第3章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/method.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/method.pdf)  
 『プロジェクト レスキューマニュアル 第6章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/projectrescue2016.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/projectrescue2016.pdf)

## 改善すべき問題点

- ・要件定義ベースラインの確立および変更管理がなされないまま開発に着手し、度重なる仕様変更および想定仕様に基づく開発による手戻り、品質劣化、コスト高、生産性低下、必要時間の喪失を招いている。
- ・発注社および受注社ともに要件定義のベースラインの早期凍結意識が不足している。
- ・発注社および受注社ともに要求仕様は全て書面化し、契約ベースにて管理する意識が不足している。

## 改善策

## ◎仕様変更管理表の作成および運用による要求仕様ベースラインの確立

- ・要件定義工程のプロセス化
- ・仕様凍結期日の明確化
- ・放縦・無定見な仕様変更の防止
- ・仕様変更の質、量および費用の管理

## 改善の期待成果

## ◎要求仕様の早期凍結、開発QCDの向上

仕様変更管理表													
No.	仕様変更 番号	発生日	業種名	仕様変更の内容	区分	グルー プ	凍止数値	影響度	難易度	対応予 算	対応日	リリース/クォン	開発工程
2					仕様変更	APL			B				
3					仕様変更	APL			A				
4					仕様変更	APL			A				
5					仕様変更	APL			B				
6					仕様変更	APL			A				
7					仕様変更	APL			A				

Ref. 『要件定義リスク クイックリファレンス 第3章・第4章』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/requirement\\_risk.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/requirement_risk.pdf)

『ソフトウェア開発のメソッド クイックリファレンス 第3章』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/method.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/method.pdf)

『プロジェクト レスキューマニュアル 第6章』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/projectrescue2016.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/projectrescue2016.pdf)



改善 No.6      テーマ : 要求仕様レビュー・チェックシートの運用

**改善すべき問題点**

- ・見積り範囲からの逸脱仕様
- ・不明・疑問仕様
- ・不明確な制限事項・条件
- ・不明確な異常処理・エラー処理
- ・不明確な非機能要件
- ・未確定仕様
- ・不明確な仕様の優先順位
- ・不明確な仕様の背景・意味
- ・不明確な性能要件

等。

**改善策**

要件定義工程において確定すべき要求仕様からの抜け、誤り、疑問、不明点、制限、条件などの未確定要素を排除するために「要求仕様チェックシート」を作成し要件定義工程においてレビューを行う。

**改善の期待効果**

◎要求仕様の品質を向上させ、その後続く設計・製造・評価工程のQCDおよび生産性を向上させる。

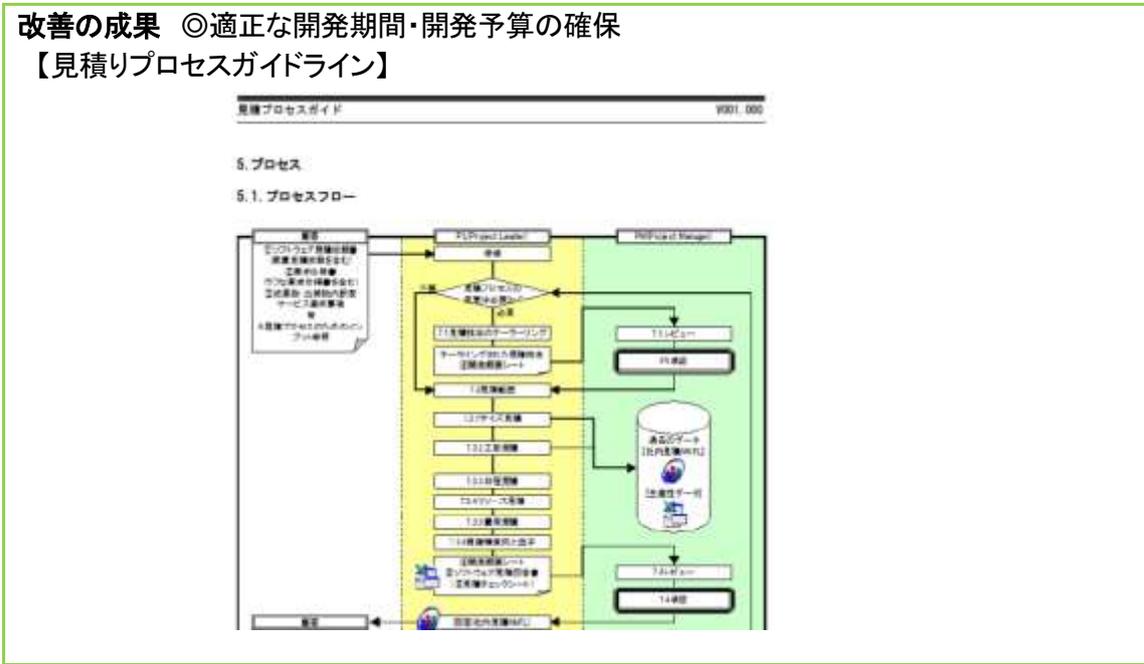
要求仕様レビュー・チェックシート		実施日	実施者
No.	大項目	確認内容	判定
1	仕様レビュー 仕様説明	見積り時の要求仕様から漏れ項目はないか？	□済 □特有 □不要
		仕様で不明確な項目はないか？	□済 □特有 □不要
		不明確な項目(注釈)として明記された項目は、納期が確定しているか？	□済 □特有 □不要
		不明確な項目の内、内容や発注への補足が必要な項目について、ロジック/開発メンバーへの作業に影響を及ぼさない補足納期が確定しているか？	□済 □特有 □不要
		不明確な項目は優先順位が付けられ、優先度が低い項目から解決する段階になっているか？	□済 □特有 □不要
		システム構成や関係は明確になっているか？	□済 □特有 □不要
		制限事項が明確になっているか？	□済 □特有 □不要
		また、制約になっている理由が明確になっているか？	□済 □特有 □不要
		仕様変更の経緯(いきさつや理由)が明確になっているか？	□済 □特有 □不要
		仕様書へ記載する操作/画面/印刷/実行内容/ファイルフォーマットの仕様は明確になっているか？	□済 □特有 □不要
		エラーが発生した時の仕様は明確になっているか？	□済 □特有 □不要
		運用上、その場で異動できないエラーが発生した場合の回避方法は明確になっているか？	□済 □特有 □不要
		特に「ログ」仕様では異常発生時の運用に支えられるので注意が必要。	□済 □特有 □不要

Ref. 『要件定義リスク クイックリファレンス 第3章・第4章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/requirement\\_risk.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/requirement_risk.pdf)  
『ソフトウェア開発のメソッド クイックリファレンス 第3章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/method.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/method.pdf)

改善 No.7	テーマ :見積りリスクの回避
---------	----------------

- 改善すべき問題点**
- ・非現実的な超短納期・低開発費の要求。
  - ・開発着手後の納期短縮要求や開発費減額の要求。
  - ・正しくない要求内容またはお粗末な要求仕様による見積り回答。
  - ・顧客案件が数週間前に発生しているにも関わらず見積り依頼が今日の明日。
  - ・システム全体を見ない見積り依頼、回答。
  - ・見積りリスクを開発費の大小でしか判断できていない。
  - ・開発途中での仕様変更管理ができていない。

- 改善策【見積りリスクの回避】**
- ◎見積り依頼・回答基準の策定および運用
  - ◎見積り方法・見積り回答シートの改善
    - ・開発範囲の明示 ;仕様の骨子決定済みのシステム範囲・機能範囲についてのみ見積る。
    - ・未決定仕様に関しては見積り回答に含まれないことを明示すること。
    - ・開発着手時期および開発期間を明示すること。
    - ・表現、必要項目の再検討を行うこと。
    - ・自動計算、ツール化できる部分の検討(FP 計算等)による見積り者毎のバラツキの低減および記載漏れの防止
    - ・見積り時点の要求仕様精度レベル(採点)による開発リスク係数の考慮を行うこと。
    - ・見積り有効期限を明示すること。



Ref. 『ソフトウェア開発のメソッド クイックリファレンス 第3章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/method.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/method.pdf)  
 『プロプロマネ銀の弾丸・チェックリスト 3.』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/silverbullets\\_20161015.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/silverbullets_20161015.pdf)

改善 No.8

テーマ : 見積りヒット率の向上

改善すべき問題点

受注につながらない見積りによって失われた開発者の時間および工数金額は、少なくない時間および費用のネガティブインパクトを開発組織に与えている。

改善策

見積り依頼を受けた開発部門別および見積り依頼を出した部門別の2種類の「見積り時間&ヒット率集計一覧表」を作成し、無駄になった時間および金額を把握すると同時に受注ヒット率が低い見積り依頼部門および担当者を特定し、該当する部署および担当者に対して無駄な見積りは結局コスト増として相手に跳ね返る旨の警告を発し無駄な見積り依頼要求を抑制する。

改善の期待効果

◎平均ヒット率よりかなり低い見積り依頼を頻発する部署ないしは担当者に対して確率の低い無駄な見積り要求を抑制させることにより、開発者の貴重な時間の浪費を防ぐと同時に開発部門のコスト負担を軽減する。

見積り時間&ヒット率集計一覧表(見積り受託部門別)									
開発部門別	件数	ヒット数	ヒット率	見積り時間数(H)	1件当たりの見積り時間(H)	見積りにかかった総時間×時間単価(5.7)	ヒットした見積り総時間×時間単価(5.7)	ヒットしなかった見積り時間×時間単価(5.7)	ヒットしなかった見積り時間×時間単価(5.7)
A	87	23	26.4	355.0	4.1	2023.5	534.9	1488.6	261.1
B	187	43	23.0	409.0	2.2	2331.3	536.1	1795.2	315.0
C	41	7	17.1	447.0	10.9	2547.9	435.0	2112.9	370.7
D	54	51	94.4	171.6	3.2	978.3	924.0	54.4	9.5
E	412	129	31.3	982.5	2.4	5600.3	1753.5	3846.8	674.9
F	164	90	54.9	1027.0	6.3	5853.9	3212.5	2641.4	463.4
<b>合計:</b>	<b>945</b>	<b>343</b>	<b>36.3</b>	<b>3,392.1</b>	<b>3.6</b>	<b>19,335</b>	<b>7,018</b>	<b>12,317</b>	<b>2,160.9</b>
	件	件	%	H	H	千円	千円	千円	H

見積り時間&ヒット率集計一覧表(見積り依頼部門別)									
見積り依頼部門別	件数	ヒット数	ヒット率	見積り時間数(H)	1件当たりの見積り時間(H)	見積りにかかった総時間×時間単価(5.7)	ヒットした見積り総時間×時間単価(5.7)	ヒットしなかった見積り時間×時間単価(5.7)	ヒットしなかった見積り時間×時間単価(5.7)
A	109	22	20.2	772.0	7.1	4400.4	888.2	3512.2	616.2
B	78	23	29.5	192.0	2.5	1094.4	322.7	771.7	135.4
C	44	6	13.6	244.0	5.5	1390.8	189.7	1201.1	210.7
D	55	31	56.4	188.0	3.4	1071.6	604.0	467.6	82.0
E	397	137	34.5	925.0	2.3	5272.5	1819.5	3453.0	605.8
F	111	64	57.7	932.0	8.4	5312.4	3063.0	2249.4	394.6
<b>合計:</b>	<b>794</b>	<b>283</b>	<b>35.6</b>	<b>3253.0</b>	<b>4.1</b>	<b>18542</b>	<b>6609</b>	<b>11933.3</b>	<b>2093.6</b>
	件	件	%	H	H	千円	千円	千円	H

改善 No.9	テーマ : 見積り作業の生産性向上
---------	-------------------

**改善すべき問題点**

不適切な見積り依頼方法や内容のために調査・検討に余分な時間がかかり見積り作業の生産性を低下させると同時に見積り精度の低下を招き、ひいてはプロジェクトのQCDの低下を招いている。不適切な依頼方法や内容の例としては次のようなものがある。

- ・要件定義書なしの見積り依頼(口頭・電話のみによる依頼など)。
- ・要件定義書内容の不備(必要事項の記述なし、記述ミスなど)。
- ・短時間の見積り回答要求。
- ・バラバラに出される仕様内容、二転三転する仕様内容など。
- ・最初から仕切り値で開発費や納期を要求してくる。

**改善策**

「見積り依頼内容調査票」を作成し、不適切な見積り依頼を頻発する要求部署および要求者を特定し、改善要求を行う。見積り不可能な事例に対しては、その理由を明示し、適切な見積り依頼書の再発行を促す。

**改善の期待効果**

◎見積り依頼方法および内容の適正化により不要な調査・検討・コミュニケーション時間を削減し、見積り作業の生産性を上げると同時に見積り精度の向上を図り、プロジェクトのQCDを向上させる。

見積り依頼内容調査票(見積り依頼の悪い例)				更新日:2016/06/01				
No.	見積り依頼内容	依頼元	依頼元担当	見積り依頼の仕方(状況)	依頼日時	受付部署	受付担当	備考
1	〇〇社△△システム	〇〇SE	〇〇〇	見積り内容がアメリカからパナリに変更になり、見積中に前提条件が変わったりして見積りに時間がかかった。	xx/xx/xx	〇〇開発	〇〇〇	その後、受注済み
2				カスタムユーザーに対する標準システムの追加の動作確認で運用が全く記されていない。				2005/4にも同様な見積り依頼があった。
3				標準見積りであるがERPのペーJ-1読み取り可や、3ペーJ-1の選定等、仕様/0調査から始める必要がある				
4				見積り依頼に書いた日に回答を求められる。				電話対応にて、1週間後に納期延長
5				要求仕様書ではなく、客先標準書を提示され要求打ち合わせするも、遠慮、確認(メール)の実が出てこず、確定するに時間を要した。				
6				客先仕様書の内容が白紙で打ち合わせ済みの認識のみであった。				
7				要求仕様書の概要にある項目の記述が詳細に存在しない。見積りWPL上では大目にて見積り積算とある。見積り項目及び見積り工数が増える				見積りで調整継続
8				要件定義不明確				
9				要件がまとめでできなく、開発スケジュールの調整をタカに依頼があり、空欄状態で回答を行う仕様書の時もあった				
10				メールで超概要説明のみ				詳細確認中
11				適用が前提の見積り依頼だが、客先適用と合わない部分があり変更が発生した。				
12				過去に見積りした複数の見積り依頼を1つにまとめた見積りだが、一部要件が必要であるにも関わらず抜けている。				

改善 No.10    テーマ : プロジェクト・リスクの低減

- 改善すべき問題点**
- ・見積り精度が悪く赤字を招いている。
  - ・要求仕様の精度が悪く開発の生産性低下や品質悪化を招いている。
  - ・類似不具合がなくなる
  - ・不具合による仕損費が減らない

- 改善策**
- ・過去に経験した主要なリスク問題の収集とカテゴリ分類。
  - ・類似不具合情報の収集とカテゴリ分類。
  - ・開発チームにおけるリスク管理表の作成および運用。
  - ・マネジメントによるリスク監査の組織的・継続的な実行。

改善の期待効果

◎リスク排除による目標QCDの達成

リスク管理表		Project Driver										Project Risk				Project Status			
項目	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
1	1-1	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
2	1-2	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
3	1-3	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
4	1-4	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
5	1-5	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
6	1-6	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
7	1-7	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
8	1-8	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
9	1-9	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
10	1-10	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
11	1-11	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
12	1-12	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
13	1-13	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
14	1-14	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
15	1-15	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
16	1-16	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
17	1-17	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
18	1-18	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
19	1-19	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生
20	1-20	リスク	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生	発生

受注リスクチェックシート				
	リスク項目	リスク度合い	ウェイト	リスク点数
1	仕様連携度合いは			
2	開発依頼書の発行有無			
3	パッケージソフトの信用率			
4	物件の利益は十分あるか			
5	ソフトの利益はあるか			
6	納期は遅いか			
7	システムの複雑さは			
8	他社の基盤接続の複雑度			
9	パッケージ利用の場合、指定された規模の範囲か			
10	データの保証について責任範囲は明確か			
11	客先への条件提示は明確か			
12	換ラデータは整次か			
13	客先店舗の総、膨大か			
14	客先の情報システム部門の有無			
15	客先のレベルは			
16	客先のキーマンは明確か			
17	リプレイスか新規か			
18	現行業務の引継ぎの集について			
19	客先はテックに対して好意的か			
20	客先保証の有無			
21	パイロット店舗の導入の有無			
22	客先指定の新技術の有無			
23	システムダウン時の客先の営業や業務に与える影響度合いは			
24	客先からのペナルティ条件提示の有無			
		リスク点数合計		0
記入上の注意				
リスク度合いの欄は(大、中、小)(良い、普通、悪い)(有り、無し)等、を リスクの低い順に○ △ × で記入する。				

見積リスクチェックシート				
	リスク項目	リスク度合い	ウェイト	リスク点数
1	発注元からの要求仕様の精度は			
2	開発における見積り精度は			
3	未経験分野の業務か			
4	提案システムの追加的財産権の扱い明確か			
5	必要な場合機密保持契約の有無は			
6	マルチベンダーの場合、責任範囲は明確か			
7	特別な保守体制、保守契約があるか			
8	客先からの要件定義資料は有るか			
9	物件全体の適正利益は確保されているか			
10	ソフト開発の適正利益は確保されているか			
11	開発部門の仕様打合せ支援費は含まれているか			
12	開発部門のテスト立会い/店舗立ち会い支援費は含まれているか			
13	品証のテスト費用は含まれているか			
14	客先提出のマニュアル作成の費用は含まれているか			
15	ソフトメンテナンス料の見積りは含まれているか			
16	リモートメンテナンスの設備費用は含まれているか			
17	リモートメンテナンス契約はなされているか			
18	立ち上げ費の見積りは含まれているか			
19	初期導入の立会い費用の見積りは含まれているか			
20	マスター作成費用の見積りは含まれているか			
21	設定変更料の見積りは含まれているか			
22	客先検証の立会い費用は含まれているか			
23	客先教育費は含まれているか			
24	見積回答書の見積り条件記述に漏れはないか			
		リスク点数合計		0
記入上の注意				
リスク度合いの欄は(大、中、小)(良い、普通、悪い)(有り、無し)等、を リスクの低い順に○ △ × で記入する。				

Ref. 『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](http://www.geocities.jp/pmfactory_ambitious/pmnote/requirement_risk.pdf)

『プロマネ 銀の弾丸・チェックリスト』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/silverbullets\\_20161015.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/silverbullets_20161015.pdf)

『プロジェクト レスキューマニュアル』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/projectrescue2016.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/projectrescue2016.pdf)

**改善すべき問題点**

目先の仕事に追われプロジェクトの目標も立てずに開発を行っているチームが多く、いつまでたっても優れたQCDの成果を残せず、人材の成長も達成できない。

目標値のない開発とは、地図のない旅あるいはゴールの見えない競技のようなものであり、出たとこ勝負の人的資源消耗型のプロジェクトに等しい。

**改善策**

開発着手前にプロジェクトのQCDに関する達成目標値を設定することにより、その目標を達成するためのあらゆる活動によってプロジェクトを遂行すること。QCDの目標値を設定するためには過去のプロジェクトのQCDデータが収集されている必要がある。現時点での開発チームのQCDに関する実力値が分からなければ目標値を設定することは不可能である。

**【目標値設定の条件と設定方法】**

- ・現時点における自チームのQCD実力値が把握されていること。
- ・現実的で達成可能であること。      ・測定可能な数値で表されること。
- ・現状の問題点を洗い出し、その数値化を行うこと。
- ・問題点の真因を明確にし、具体的な改善策を立案すること。
- ・改善策の期待効果に見合った目標値を設定すること。
- ・目標値は、その変更も含めて、常時、チーム内の全員で共有されていること。

**【目標値の例】****◎品質指標**

- ・テスト工程発生不具合のレベル・件数・発生率(／kstep)
- ・市場発生不具合のレベル・件数・発生率(／kstep)
- ・仕様ミス率(発生件数／kstep)      ・設計ミス率(発生件数／kstep)
- ・製造ミス率(発生件数／kstep)      ・評価ミス率(評価ミス発生件数／kstep)
- ・プロセス品質;レビュー漏れ率(レビュー漏れ不具合件数／kstep)

**◎コスト指標**

- ・目標利益率または利益額

**◎生産性指標**

- ・見積りヒット率／要求仕様書精度達成率／ソフトウェア流用率(モジュール化率、リピート率)／プログラム製造ステップ単価

**改善の期待効果**

◎目標値の設定はデータ・ドリブンなプロジェクトの第一歩と言え、ソフトウェア開発を科学的・工学的に遂行する唯一の手段である。数値に基づく開発は、プロジェクトの代表的な問題であるあいまいさや情緒的な判断を一掃する。

**【目標値設定による効果】**

- ・現状の開発力が明確になる。
- ・到達すべきゴールが見え、モチベーションの維持につながる。
- ・改善すべき問題を明確にする。
- ・あいまいさや情緒性を排し、合理的ないしは妥当性のあるチームを育成する。

**改善すべき問題点**

開発チームにおけるQCDおよび生産性に致命的なインパクトを与えているコミュニケーション問題には以下のようなものがあります。

- ・人の話を最後まで聞くことができない。
- ・部下の意見を十分に聞けない。
- ・質問がしづらくて失敗を招いている。
- ・自分の主張の根拠について適切に説明できない。
- ・臆病な性格がコミュニケーションを阻害している。
- ・忙しい時、一方的な自分の意見の押し付けになってしまう。
- ・伝わったと思った内容が伝わっていなかった。
- ・要求内容を勝手に深読みして仕様を膨らませてしまった
- ・まともな業務文書が書けない。
- ・何が書いてあるのか分からない仕様書・設計書。
- ・表面的な言葉の理解では本質が見抜けない。

**改善策**

## ◎「日次情報共有会議」の実行

毎日の短時間ミーティングをチーム内で実行すること。メンバー全員から「①昨日何を実行したか ②明日何を実行する予定か ③今抱えている問題は何か」の3点について報告を受け、リーダーはそれぞれに的確なアドバイスを行う。

数十人以上での課や部における一方的な通達に偏りがちな朝礼・昼礼は何の効果もないと思った方がいい。最大10人以下のメンバーにおいて「日次情報共有会議」を実行することによって、全員の進捗・問題点および全体的な情報共有は間違いなく実現され、多くのメンバーが持っているコミュニケーションの苦手意識も改善されていく。

**改善の期待効果**

◎下記課題の改善によりコミュニケーション不良に起因する間違いや失敗は激減する。

- ・チーム内の情報共有。
- ・毎日の行動結果・行動予定の確認。
- ・毎日の課題／リスクの掘り起こしと対応状況の確認。
- ・開発目的・範囲・情報の徹底。
- ・開発仕様の情報共有。
- ・開発仕様の優先度順位情報の共有。

Ref. 『コミュニケーション自習講座』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/communication1.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/communication1.pdf)

**改善すべき問題点**

時間不足やあせりによって必要な工程の中断や工程自体のスキップが行われた場合、間違いなく品質問題に直結してしまいます。本来ならばプロジェクトのQCDはみな必須条件のはずですが、時間不足に陥ったプロジェクトでは形だけでも納期優先となり必要な作業の希薄化ないしはスキップという愚行に手を染める結果を招きます。その結果、市場で大トラブルを発生させることとなります。

**改善策**

◎基本的な対策は見積りおよび要件定義工程において妥当な開発期間を獲得することや改善活動によって生産性を向上することにあります。ここでは具体的な手抜き行為を列挙した「手抜き防止のチェックリスト」の導入による対策を示します。

**【設計・製造工程における手抜き防止のチェックリスト】** 

- 仕様の事前調査・検討をスキップしていないか。
- ドキュメントに基づく設計・製造・評価が行われているか。
- 基本設計工程をスキップしていないか(基本設計書未作成)。
- 詳細設計工程をスキップしていないか(詳細設計書未作成)。
- 仕様実現に必要なメモリー容量、ハードウェア諸元を確認したか。
- パフォーマンス性能、レスポンス性能の考慮が抜けていないか。
- 異常系処理の考慮が抜けていないか。
- 各工程における自己チェックをスキップしていないか。
- 各工程の内部／外部レビューをスキップしていないか。
- ソースコードやドキュメントの無節操なコピー&ペーストが行われていないか。
- プロジェクト完了時の振り返りやラップアップをスキップしていないか。

**【テスト工程における手抜き防止のチェックリスト】** 

- インターフェースにおける最終的な出力確認をスキップしていないか。
- テスト項目未消化のまま次工程に進んでいないか。
- 単体テストをスキップしていないか。
- 結合テストをスキップしていないか。
- 総合テスト項目未消化のまま客先にリリースしていないか。
- 発見済み未修正バグを含んだままリリースしていないか。

### 【全般的な手抜き防止のチェックリスト】

- 仕事の意味を考えて作業を行っているか。
- 時間がないことを手抜きの言い訳にしていないか。
- 予算不足を理由に手抜き仕事が行われてはいないか。
- 合理的なプロセス管理表に基づいて開発を実行しているか。
- 重要な手順抜けを防ぐためのチェックリストを運用しているか。
- 手順慣れで必要なチェックを飛ばしてはいないか。
- コーディングやドキュメント作成において安易なコピー&ペーストを行ってはいないか。
- ソフトウェアやドキュメント等の成果物の出荷前の現物確認や動作確認を行っているか。
- 深刻な疲労に陥ってはいないか。
- 手抜きで発生した実例を本人に示しているか。
- 日次情報共有会議等で問題の共有を毎日実行しているか。
- 前記問題点の改善活動をリーダー主導で行っているか。

### 改善の期待効果

◎「手抜き防止のチェックリスト」は「レビュー・チェックシート」と合わせて使用することで意図的な手抜きと意図しないミス両方の防止に効果を発揮し、QCDの改善に寄与する。

Ref. 『プロプロマネ銀の弾丸・チェックリスト 15. 』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/silverbullets\\_20161015.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/silverbullets_20161015.pdf)

**改善すべき問題点**

◎開発の全ての工程の中で見られる代表的な無駄としては次のようなものが挙げられる。

- ・繰り返される同じ失敗
- ・受注につながらない見積り
- ・意味不明な報告書
- ・結論の出ない会議
- ・長時間の井戸端会議
- ・タイミングが遅いレビュー
- ・保険かけのCCメール
- ・長時間にわたる世間話的な電話
- ・遅刻により相手を待たせる時間
- ・優先順位を考えない場当たりの仕事のやり方
- ・責任の押し付け合い、ケンカ、派閥争い、勢力争いによって失われる時間とエネルギー
- ・多重請負構造・多層組織構造によるコミュニケーション不良

**改善策**

上記に列挙されたさまざまな無駄は特段の注意を向けなければ無駄だとは認識されにくい人間行動の不適切さに起因するものが多い。これらの問題に対する個人的ないしは個別的な対応はほとんど効果を期待できない。これらの問題は組織文化の問題であるため組織的な躰、すなわち組織的な運動が必要であり、例えばそれは伝統的なアプローチである5S運動が有効である。

**【ソフトウェア開発における5S】**

- ・整理： 開発に必要なものは残し、不要なものは廃棄すること。
- ・整頓： 開発に必要なものは常時使用可能な状態にしておくこと。
- ・清掃： 開発に必要なものの欠陥は常時修正しておくこと。
- ・清潔： 開発に必要なものは最新の状態にメンテナンスしておくこと。
- ・しつけ： 上記の4Sを実行する習慣を身につけること。

これらの運動は、まさにさまざまな改善活動の中において組織員全員の活発なコミュニケーションの実行のもとに、組織的にかつ継続的に実行する必要がある。

**改善の期待効果**

◎優れた開発活動を支えるものは、言うまでもなく、個人および組織における健全なヒューマンスキルに他ならず、これらの人間能力は、その自律性を基本的な土台とし、妥当性と合理性の双方の力によって、柔軟な思考・行動を生み出し、個人ないしは組織間における相互義務および相互扶助の行動を促進するでしょう。最後に、これらの魅力的かつ人間的な思考・行動によって獲得された有形・無形の心的・物的な資産を、将来の自他に譲る行為によって、その個人ないしは組織に永続的な繁栄をもたらすことができると考えられます。







**改善 No.18**      **テーマ : 初期設計レビュー・チェックシートの運用**

- 改善すべき問題点**
- ・システムの全体像が表されていない。
  - ・設計のコンセプトが不明。
  - ・要求機能を構造化した設計になっていない。
  - ・仕様の変更および仕様の間違いが反映されていない。
  - ・機能を実現するための情報の欠落や誤りが多い。
  - ・柔軟性に乏しい構造設計になっている。
  - ・詳細設計書作成のベースドキュメントとして使用可能な設計品質ではない。
  - ・結合テスト仕様書のベースドキュメントとして使用可能な設計品質ではない。
  - ・要求仕様が完全に網羅されていない。
  - ・理解が困難ないしは不可能な内容。
  - ・実運用との整合性が分からない。

**改善策**  
要件定義書における誤りや抜けを発見すると同時に初期設計書としての基本要件を満たすための諸条件を網羅した「初期設計チェックシート」を作成し、初期設計工程においてレビューを行う。

**改善の期待効果**  
◎要件定義書の欠陥を早期に発見すると同時にシステムの基本設計書である初期設計書の品質を確保することで、その後続く詳細設計・製造・評価工程のQCDおよび生産性を向上させる。

項目	内容	確認方法	確認結果	備考
1	システム全体の構成(システム、サービス、機能)が明確に示されているか?	図面・仕様書	○	
2	設計のコンセプトが明確に示されているか?	図面・仕様書	○	
3	要求機能を構造化した設計になっているか?	図面・仕様書	○	
4	仕様の変更および仕様の間違いが反映されているか?	図面・仕様書	○	
5	機能を実現するための情報の欠落や誤りが多いか?	図面・仕様書	○	
6	柔軟性に乏しい構造設計になっているか?	図面・仕様書	○	
7	詳細設計書作成のベースドキュメントとして使用可能な設計品質ではないか?	図面・仕様書	○	
8	結合テスト仕様書のベースドキュメントとして使用可能な設計品質ではないか?	図面・仕様書	○	
9	要求仕様が完全に網羅されていないか?	図面・仕様書	○	
10	理解が困難ないしは不可能な内容があるか?	図面・仕様書	○	
11	実運用との整合性が分からないか?	図面・仕様書	○	

Ref. 『プロジェクト レスキューマニュアル 第5章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/projectrescue2016.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/projectrescue2016.pdf)  
『設計製造評価リスク クイックリファレンス 2.』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/sekkei\\_riskreference.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/sekkei_riskreference.pdf)







## Check List プログラマーがやってはいけない手抜きの12ヶ条

### ▶Check Timing: プログラミング着手前

- ソースコードの”コピペ”を無節操に行なっていないか。
- ベースのプログラム構造を理解しないままソースコード変更に着手していないか。
- データ処理の仕掛けを理解しないままソースコード変更に着手していないか。
- 変更対象の機能の呼び出され方、関連機能との連携の仕方を理解せずソースコード変更に着手していないか。
- 作られたデータはどの機能がどのタイミングでアクセスしているのか理解しないままソースコード変更に着手していないか。
- 他人の書いたソースコード理解するためにそれを読むことだけしかしていないのでは。
- 一つの変更要件に対して複数の担当者が関係している場合、早い段階で互いに検討内容等の相互確認・レビューをしているか。また、みなバラバラにソース修正にとりかかり、寄せ集めたソースで一気にテストをしているのではないか。
- お互いにどこを変更したのか誰も知らないのではないか。
- 一つの機能に関する多数の関数において同一巨大パラメータ構造体(256バイト等)を共通に使用してはいないか。
- 開発中に発見された、以前から含まれていた潜在バグについていきなりソースコードの修正を行なっているのではないか。
- 「プロジェクト計画書」を書いていないのではないか。
- 「データ」をとっていないのではないか。

(出典:清水吉男著「派生開発を成功させるプロセス改善の技術と極意」および「要求を仕様化する技術 表現する技術」)

Ref. 『プロジェクト レスキューマニュアル 第4章』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/projectrescue2016.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/projectrescue2016.pdf)

『ソフトウェア開発のメソッド クイックリファレンス 第9章』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/method.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/method.pdf)

『プロプロマネ銀の弾丸・チェックリスト 15. 』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/silverbullets\\_20161015.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/silverbullets_20161015.pdf)

『設計製造評価リスク クイックリファレンス 3. 』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/sekkei\\_riskreference.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/sekkei_riskreference.pdf)







**改善 No.24**      **テーマ : 総合テストレビュー・チェックシートの運用**

**改善すべき問題点**  
 総合テスト仕様書(テスト・チェックリスト)の欠陥が多い。  
 ・総合テスト仕様書(チェックリスト)そのものが揃っていない。  
 ・抜け・誤記が多く精度が悪い。      ・異常系のチェックが抜けている。  
 ・実運用テストのチェックが抜けている。      ・性能評価のチェックが抜けている。  
 ・非機能要件のチェックが抜けている。      ・外部 I/O系のチェックが抜けている。

**改善策**  
 総合テストの内容が要求仕様通りに動作することを確認する項目を網羅した「総合テストチェックシート」を作成し、テストの前にレビューを行うことで、作成された総合テスト仕様書における誤りや抜けを発見する。

**改善の期待効果**  
 ◎総合テスト仕様書の欠陥を早期に発見することで総合テストの品質および生産性を向上させ、最終的には、顧客の要求通りの品質および機能を確認する。

項目	内容	確認方法	確認結果	備考
1	記述内容	作成すべきチェック項目は揃っているか?	○	確認結果
		作成したチェックリストには下記が記載されているか? ①チェックリストのタイトル、文書ファイル名 ②検査項目/システム名を記載する欄 ③検査内容/検査方法/手順 ④検査結果記入欄(X/○/N/A)記入欄や計算機記入欄 ⑤検査日記入欄 ⑥検査者記入欄	○	確認結果
		各種チェックリスト(仕様書に紐づくチェックリスト)の元ネタである仕様書の完成度は十分か? 既存のチェックリストを流用する場合、内容は十分か?	○	確認結果
		作成したチェックリストに記述されている文章は明確か?	○	確認結果
		作成したチェックリストは、他の類似した仕様にも適用可能な内容か?	○	確認結果
		シロカウチェックリストを初期段階、(1)ターンオーバーまで実行(2)運用検査可能な場合がある。早期に検査し、上記(1)は確認にポイントとなる項目が含まれているか? 異常系チェック項目の内容は十分か?	○	確認結果
		完成度の高いチェック項目があるか?	○	確認結果
		システム全体の範囲よりチェック項目が抜けられているか? 資料の更新に伴ってチェック項目が更新されているか? 更新事項は通知されているか? ユーザ確認による動作確認のチェック項目は確認されているか?	○	確認結果
		ログチェックを行うという検査項目はあるか?	○	確認結果
		テストスケジュールや検査責任が明確になっているか? テスト担当者間の連携がスムーズにできるか?	○	確認結果

Ref. 『設計製造評価リスク クイックリファレンス 4.』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/sekkei\\_riskreference.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/sekkei_riskreference.pdf)  
 『ソフトウェア開発のメソッド クイックリファレンス 第13章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/method.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/method.pdf)  
 『プロブロマネ銀の弾丸・チェックリスト #55、#71』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/silverbullets\\_20161015.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/silverbullets_20161015.pdf)  
 『プロジェクト レスキューマニュアル 第3章』  
[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/projectrescue2016.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/projectrescue2016.pdf)

改善 No.25	テーマ : 構成管理作業チェックシートの運用
----------	------------------------

**改善すべき問題点**  
 ソフトウェアの適正な管理がなされなかったことにより以下のような問題が発生している。

- ・出荷ソフトウェアにおけるモジュール抜け。
- ・出荷ソフトウェアにおける誤ったモジュールの混入。
- ・出荷ソフトウェアにおけるバージョンの取り違い。
- ・改修時または不具合修正時のデグレージョン。
- ・度重なる仕様変更による混乱。

**改善策**  
 効果的な構成管理ツールを使用すると同時に人間系に由来するエラーを防ぐために「構成管理作業チェック・シート」を使用する。

**改善の期待効果**  
 ◎構成管理ツールとともに構成管理作業チェック・シートを組織的に運用すれば人的ミスによるソフトウェアの構成不具合を防止できスムーズな市場投入が可能となる。

■構成管理作業チェック・シート

No	手順	作業	チェック項目	<input checked="" type="checkbox"/>
0	全般		構成管理サーバと開発サーバは、別マシンになっているか？  構成管理サーバ（フォルダ）にセキュリティをかけ、S C M A 及び P L 以外は更新できないようになっているか？	
1	事前作業	フォルダ作成	チェックアウト用のフォルダを作成しているか？ チェックイン用のフォルダを作成しているか？	
		チェックアウト依頼書作成	依頼書を作成しているか？ 記号間違い(対象の抜け等)をチェックしているか？(依頼された物を素直にチェックアウトしているのであれば無し)	
2	構成管理	チェックアウト作業	チェックアウト依頼書の確認(承認漏れや記入漏れがないか等)	
			修正対象バージョンの確認(指定された対象バージョンに間違いがないか確認しているか？)	
			チェックアウトされたモジュールの一覧が管理されているか？	
			チェックアウト作業後にチェックアウト依頼書の作業完了項目に記入しているか？	
	チェックイン作業	チェックアウト依頼書とのチェック(チェックアウト依頼書と照合して確認しているか？)		
		修正不要分の確認(チェックアウトされていてチェックインされないものを確認しているか？)		
	追跡表作成作業	追跡表作成作業	実際に修正が施されているかソースコンペア等の確認を行なっているか？	
			チェックインされるモジュールの一覧が管理されているか？	
			チェックイン作業後にチェックアウト依頼書の作業完了項目に記入しているか？	
			修正内容の追跡は可能か？	
			S C I 追跡管理表が作成されているか？	
			S C I 追跡管理表がチェックイン作業後に更新されているか？	

改善 No.26      テーマ :リリースレビュー・チェックシートの運用

**改善すべき問題点**  
 ソフトウェアおよび要求仕様書・設計書・評価仕様書等の最終正式版が揃っていない。  
 ・リリース成果物が揃っていない場合がある。  
 ・リリース成果物が最終正式版になっていない場合がある。  
 ・開発途中で発生した変更内容が更新されていない場合がある。  
 ・リリースされるソフトウェアが書き込まれたマスターメディアを使用した最終確認テストが行われていない場合がある。

**改善策**  
 顧客にリリースされる全ての成果物項目を網羅した「リリース・チェックシート」を作成し、リリース前に確認することで間違いのない製品であることを確認する。

**改善の期待効果**  
 ◎リリースされる成果物が正しい最終成果物であることを確認することで、顧客に間違いのない製品をリリースすることができる。

リリースレビュー・チェックシート					
製品名/バージョン/レビュー対象項目		実施日	実施者		
№	大項目	確認内容	例・補足	確認結果	備考
1	仕様書	仕様書フィードバックがされているか？ 開発途中で「仕様書へ反映」となった項目の記載が完了しているか？		□済 □残有 □不要	
2	設計書	設計書フィードバックがされているか？ 開発途中で「設計書へ反映」となった項目の記載が完了しているか？		□済 □残有 □不要	
3	プログラム 単位・結合PCL	（A）ソース物がすべて揃っているか？  単位・結合チェックリストをすべてチェックしているか？ 結合テストで発生した不具合の数が確認済完了しているか？ 不具合修正の履歴がわかるか？ チェックリストに単位・結合テスト履歴が記載されているか？ チェックリストに修正者の氏名が記載されているか？ チェックリストに単位・結合テストの不具合件数が記載されているか？	不具合修正履歴などが存在しているか？ 不具合数が確認済完了しているか？ ソースのリリース漏れなどないか？	□済 □残有 □不要 □済 □残有 □不要 □済 □残有 □不要 □済 □残有 □不要 □済 □残有 □不要	
4	影響度表	追加機能の行が追加されているか？ 新たに影響が発生した項目は反映されているか？ 対応すべき過去不具合/未済不具合は対応されているか？ 対応した不具合の項目は記載されているか？ 備考欄に前回の対応が記載されているか？	本機で○○業務が増加している場合、○○業務の行が増えているか？ 単位・結合テスト中に発生した項目 結合テストで発生した不具合 対応すべき過去不具合は何か？が明確になっているか。項目は精査すること。対応しない不具合は、対応しない理由がわかるか？	□済 □残有 □不要 □済 □残有 □不要 □済 □残有 □不要	
5	その他	評価Gへリリースする場合、評価対象外一覧と評価Gへのリリース物フェーズが合っているか？ 品質評価データの集約の作成担当者を確認する。 バージョン管理チェックリスト/変更モジュール一覧の作成担当者を確認する。 不具合修正履歴が必要か？ 必要の場合、用意されているか？また、内容は評価Gの項目でも分かる内容となっているか？ 入れ替え手順書やセットアップ手順書が必要か？必要の場合、用意されているか？また、内容は評価Gの項目でも分かる内容となっているか？		□済 □残有 □不要 □済 □残有 □不要 □済 □残有 □不要 □済 □残有 □不要 □済 □残有 □不要	

改善 No.27	テーマ : ラップアップの実行
----------	-----------------

**改善すべき問題点**  
 プロジェクト終了時に行う振り返り会議(ラップアップ・ミーティング)を実行しない場合に起こる問題は次の通りです。

- ・失敗に学ばないために同様の問題を再発させる。
- ・成功要因を他のチームへ横展開できない。
- ・目標と実績の対比を行わないために、失敗／成功のレベルを数値で把握できない。
- ・失敗／成功の要因を全メンバーで共有できない、学習できない。
- ・次なる目標値の設定ができなくなる。

**改善策**  
 必ず「ラップアップ・ミーティング」を実行し、下記項目の目標値と実績値の差異の分析を行い、その原因を特定し、対策案の作成を行うこと。

◎品質問題の振り返り

- ・発生不具合(総合テスト)のレベル、件数および発生率(／Kstep)
- ・発生不具合(市場リリース後)のレベル、件数および発生率(／Kstep)
- ・仕様品質;仕様ミス率(仕様ミス発生件数／Kstep)
- ・設計品質;設計ミス率(設計ミス発生件数／Kstep)
- ・製造品質;製造ミス率(製造ミス発生件数／Kstep)
- ・評価品質;評価ミス率(評価ミス発生件数／Kstep)
- ・プロセス品質;レビュー漏れ率(レビュー漏れ不具合件数／Kstep)

振り返りシート 品質問題 不具合発生率					
	定義	目標値	実績値	主な原因	対策
仕様品質	仕様ミス率 (仕様ミス発生件数／Kstep)				
設計品質	設計ミス率 (設計ミス発生件数／Kstep)				
製造品質	製造ミス率 (製造ミス発生件数／Kstep)				
評価品質	評価ミス率 (評価ミス発生件数／Kstep)				
プロセス品質	レビュー漏れ率(レビュー漏れ不具合件数／Kstep)				
総合テスト品質	Aレベル不具合発生率(件数／Kstep)				
	Bレベル不具合発生率(件数／Kstep)				
	Cレベル不具合発生率(件数／Kstep)				
市場稼働品質	Aレベル不具合発生率(件数／Kstep)				
	Bレベル不具合発生率(件数／Kstep)				
	Cレベル不具合発生率(件数／Kstep)				

### ◎コスト・利益問題の振り返り

適正な利益確保の振り返りのポイント

1. 見積りと実績の差異の分析を行うこと。
2. 適正利益率が確保されなかった場合、見積り自体の問題と開発の問題に分けて問題は何か、問題のレベルはどうだったのか、何故その問題が発生したのかの原因および真因の分析と対策案の作成を行うこと。

振り返りシート コスト問題					
イベント	見積り額	実績金額	差異	原因	対策
開発費合計					
<内訳>					
直接労務費					
間接労務費					
直接材料費					
外注費					

### ◎生産性問題の振り返り

見積りヒット率／要求仕様書精度達成率／ソフトウェア流用率(モジュール化率、リポート率)／プログラム製造ステップ単価などの目標値と実績値の差異の分析を行い、その原因を特定し、対策案の作成を行うこと。

振り返りシート 生産性問題					
イベント	目標値	実績値	差異	原因	対策
見積りヒット率					
要求仕様書精度達成率					
ソフトウェア流用率(モジュール化率)					
ソフトウェア流用率(リポート率)					
プログラム製造ステップ単価					

### 改善の期待効果

◎「ラップアップ」は「目標値の設定」とセットになって機能し、開発チームをデータ・ドリブンな組織に変身させ、次なるプロジェクトの目標設定を可能とし、チームメンバーの確実な成長を約束する。

- ・失敗に学び同様な失敗を防止する。
- ・成功要因を他のチームへ横展開可能とする。
- ・目標と実績の対比を行うことで、失敗／成功のレベルを数値で把握できる。
- ・失敗／成功の要因を全メンバーで共有でき、大きな学習効果を与える。
- ・次なる目標値の設定が可能となる。

## 改革コンセプト#1

キーワード : 時間の確保

**改革のコンセプト** : ものごとを遂行するにおいて時間の確保は究極の必要条件である。

ヒト・モノ・カネ・情報などのリソースはなんとかなる可能性があるが、失われた時間を取り戻すことはできない。プロジェクトの究極の成功要件は、必要時間の確保にあることを深く認識し、必要時間の確保および無駄な時間の削減に必要なあらゆる対策を講じること。

### 必要なアクション

- (1) 先行調査によりリスクを排除し、後工程で発生する不始末による膨大な時間ロスを防ぐこと。
- (2) 要件定義工程において顧客とのコミュニケーションを緊密にすることで要求仕様の誤解やモレをなくし後工程で発生する膨大な時間ロスを防ぐこと。
- (3) リスクを考慮した妥当かつ正確な見積りを行うことで開発に必要な絶対時間の確保をすること。
- (4) 開発プロジェクト内のコミュニケーションを緊密にすることで指示の誤解やモレをなくし後工程で発生する膨大な時間ロスを防ぐこと。
- (5) 重要な機能から先に開発を行い、完成の都度顧客と現物にて動作確認を行うことで、やり直し・修正などの膨大な時間ロスを防ぐこと。
- (6) 上記をより可能にするために顧客および開発チームは場所および情報の共有を行い開発を共同して遂行すること。

### アクションの具体例

#### ◎先行調査

要求仕様の疑問点・不明点の払拭、未経験の仕様・機能の調査・理解、プロトタイプ作成による未知の新技术領域のリスクの排除など。

#### ◎顧客との密接直接コミュニケーション

要件定義に関して顧客との集中合宿検討会を含む密接かつ頻繁な仕様検討会の実行。

#### ◎妥当かつ正確な見積り

要求仕様および開発全体に関するリスクを把握し、開発に必要な時間と予算の確保を行うこと。

#### ◎開発内部のコミュニケーション

毎日ベースの定例日次情報共有会議の実行、週次のまとめ振り返りおよび次週予定検討会の実行、月次のまとめ振り返りおよび次月予定検討会の実行など。

◎顧客に対して7日～10日ごとの重要仕様順のリリースを繰り返し顧客および開発チームによる共同検証を行い、不具合修正・仕様変更等は顧客との直接対話による即断・即決による対応を行うこと。

### 期待効果

- ◎プロトタイプ作成による未知の新技术領域特有の不具合のあぶり出しによる、本番開発における致命的な不具合の予防。
- ◎顧客との密接直接コミュニケーションにより要件定義における相互の理解を促進するだけでなく相互の信頼関係を醸成する。
- ◎要求仕様の早期明確化をベースにした妥当かつ正確な見積りは、プロジェクトの絶対的な成功要因である必要時間の確保と必要予算の確保を確かなものにする。
- ◎開発チーム内の日次・週次・月次の頻繁なコミュニケーションはメンバー全員による問題の共有および情報の整理整頓を促進し、開発者における仕様の誤解・設計ミス・段取りミス等を大幅に削減すると同時に大勢の開発者における開発完遂の強いモチベーションを最後まで維持させる。
- ◎短期間開発ごとに逐次リリースされた動作する現物のソフトウェアにての顧客との直接的検証・対話は開発における後戻りや不要な修正等の無駄な作業を大幅に削減する結果となり、同時に顧客における安心感・満足感の醸成に大きく寄与する。

Ref. 『ソフトウェア開発のメソッド クイックリファレンス 第5章』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/method.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/method.pdf)

『プロマネ 銀の弾丸・チェックリスト 16. 』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/silverbullets\\_20161015.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/silverbullets_20161015.pdf)

『プロジェクト レスキューマニュアル 第7章4. 』

[http://www.geocities.jp/pmfactory\\_ambitious/pmnote/projectrescue2016.pdf](http://www.geocities.jp/pmfactory_ambitious/pmnote/projectrescue2016.pdf)

### 改革のコンセプト

マンネリ化したウォーターフォール型開発組織を自律性のある組織へ転換させる。

### 必要なアクション

1. 毎日やろう
2. 毎日聞こう
3. 毎日見よう
4. 毎日まとめよう
5. マネジャーは心理的・地理的にいつも開発担当者のそばに居よう。
6. 顧客の声を聞こう
7. 早くしよう
8. 見えるようにしよう

これらの行動は開発者における自律性・協調性を促し、顧客との密な連携を促し、日々刻々の変化に即応するための基本的な行動のポイントとなる。

### アクションの具体例

1. 毎日やろう
  - ・課題/バラシおよびその対策を毎日実行しよう。
  - ・開発担当者を毎日巡回しフォローしよう。
  - ・対策の指示を毎日行なおう。
2. 毎日聞こう
  - ・お客様の話を良く聞こう。
  - ・開発担当者の不平・不満・グチを聞こう。
  - ・関連部署の状況を聞こう。
3. 毎日見よう
  - ・毎日、リスク管理表・課題管理表を見よう。
  - ・毎日、開発担当者たちの進捗状況を見よう。
  - ・毎日、開発担当者たちの健康状況を見よう。自分の健康状況も。
4. 毎日まとめよう
  - ・変化する課題・問題を毎日、整理更新しまとめよう。
  - ・明日以降に必要なアクションをまとめよう。
5. マネジャーは心理的・地理的にいつも開発担当者のそばに居よう。
  - ・マネジャーはいつでも担当者の相談にのろう。
  - ・マネジャーは事務所から開発ルームに引っ越そう。
6. 顧客の声を聞こう
  - ・お客様を早い時点で開発パートナーとして巻き込もう。
7. 早くしよう
  - ・自部署の担当ではないと判断した仕事は即刻担当部署に渡そう。
8. 見えるようにしよう
  - ・リーダーはサブリーダーの、サブリーダーは担当者・外注の仕事の範囲を明確にしよう。
  - ・必要な情報は口頭だけに頼らずまず手書きでも白板のコピーでも良いから書面として見えるようにしておこう。
  - ・スケジュール進捗表・管理表は毎日更新しておこう。

### 改革コンセプト#3

キーワード : 変化即応型開発

#### 改革のコンセプト

1. 顧客要求に対して俊敏な対応をすること。
2. 開発チーム全員による意識・価値観の共有による目標の達成。

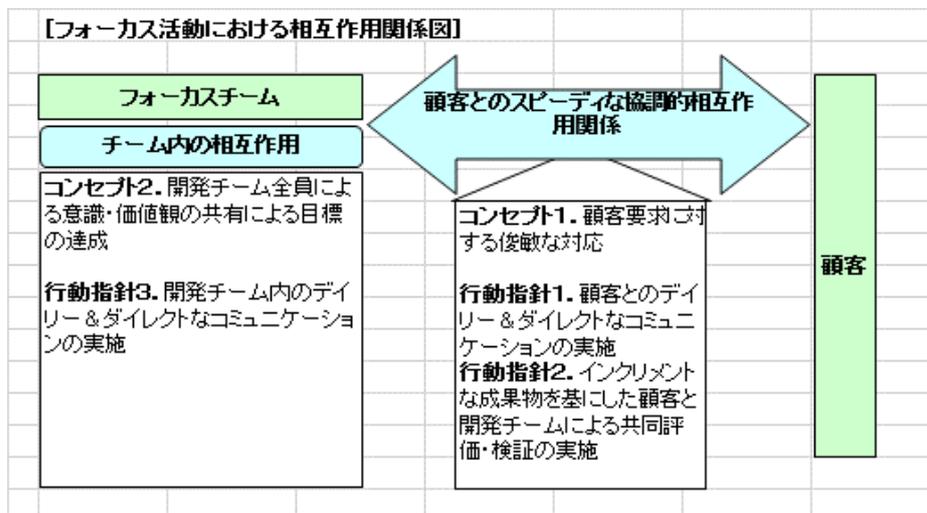
#### 必要なアクション

- (1)顧客とのデイリー&ダイレクトなコミュニケーションの実施。
- (2)インクリメントな成果物を基にした顧客と開発チームによる共同評価・検証の実施。
- (3)開発チーム内のデイリー&ダイレクトなコミュニケーションの実施。

#### アクションの具体例

◎顧客と開発チーム間あるいは開発チーム内において発生する課題・問題を、成果物である現物に基づいて直接的なコミュニケーションによって毎日ベースに俊敏に解決しようとする。プロジェクトの主要な問題は人と人之间に発生するものであり、それらの問題をすばやく解決することからしかプロジェクトの成功は導き出せない。

フォーカス活動(カイゼン活動)における顧客とフォーカスチーム(カイゼン主導チーム)間およびフォーカスチーム内の相互作用関係は下図のように表せる。



フォーカスチームと顧客間における相互作用関係を実現するための基本コンセプトは「顧客要求に対する俊敏な対応」であり、これを支える行動指針は「顧客とのデイリー&ダイレクトなコミュニケーションの実施」および「インクリメントな成果物を基にした顧客と開発チームによる共同評価・検証の実施」である。

またフォーカスチーム内におけるメンバー間の相互作用関係を実現するための基本コンセプトは「開発チーム全員による意識・価値観の共有による目標の達成」であり、これを支える行動指針は「開発チーム内のデイリー&ダイレクトなコミュニケーションの実施」である。

**改革のコンセプト**

フォーカス活動(カイゼン活動)は、顧客に対してスピーディな協調的相互作用関係の中で、顧客要求の変化に対応すべく密接なコミュニケーションを実行し、正しく機能するソフトウェアを提供し、さらにこれらを実現すべくフォーカスチーム内においては自律的なメンバー間においてフェイス・トゥ・フェイスの会話を実行し、一定の開発ペースを維持すべく顧客要求の価値の高い順にインクリメンタルな開発を継続し、有効な技術の採用に注意を払い、定期的な振り返りによるフィードバックにより無理・無駄・失敗の予防を行なおうとするものである。

**必要なアクション****◎コンセプト1. 顧客要求に対する俊敏な対応**

行動指針1. 顧客とのデイリー & ダイレクトなコミュニケーションの実施

行動指針2. インクリメンタルな成果物を基にした顧客と開発チームによる共同評価・検証の実施

**◎コンセプト2. 開発チーム全員による意識・価値観の共有による目標の達成**

行動指針3. 開発チーム内のデイリー & ダイレクトなコミュニケーションの実施

**期待効果**

◎フォーカス活動(カイゼン活動)はウォーターフォール開発における 3 つの問題点の解消に大きな効果があるものと考えられる。

- ① 工程間のコミュニケーション・ギャップによる問題
- ② 開発能力による問題
- ③ 認識能力による問題

フォーカス活動は顧客とのあるいは開発チーム内における密接なコミュニケーション、顧客要求順位の高いものから行われるインクリメンタルな開発等により下記のような効果を生む。

- ・分割リリースごとに動作確認・顧客検証を終了して進むため、開発完了に向けて見通しが良く、予測可能である。
- ・低リスクである。失敗に対するリカバリー期間確保が容易である。顧客および開発者における安心感が確保される。
- ・要件の変更・追加に対する柔軟性・融通性がある。
- ・品質確保の効果がある。
- ・顧客満足度の獲得がしやすい。
- ・開発者の自主性・自律性・モチベーションの向上が期待できる。
- ・チームワークの強化が行われる。
- ・コミュニケーション能力の育成・定着化が行われる。

上記はいわゆるウォーターフォール開発にアジャイル的な要素を組み込んだ開発方式であり、筆者がリードした大規模な開発においてもその効果は実証済である。