夏のソフトウェアプロセス改善セミナー 2008

「ソフトウェア開発委託の実態」

ー責任回避と丸投げの病理ー

2008年 6月23日

日本SPIコンソーシアム（JASPIC）
特別会員　岩見　好博

Opto Digital Technology

Agenda

- 「ソフトウェア開発委託の実態」
  - 責任回避と丸投げの構造
- 残された課題の解決に向けて
  - 各自がその責任を果たす
  - 内製化の動き

(c)2008 Yoshihiro Iwami, JASPIC

事例の概要

- 他プロジェクトが外注ソフトウェア不具合で混乱
- 外注ソフトウェア管理の支援を要請された

- 支援対象システム
  - Visual Basicの業務アプリケーション改修
    - » ベースソフトウェア開発元に委託
  - 規模
    - » 要件数　　20件 （詳細ベースで約100項目）
    - » 新規モジュール　4本、改修　20本
  - 委託内容
    - » 仕様書改修から単体テストまで
  - 期間　　1．5ヶ月
  - 工数　　12人月（委託先の見積り）

(c)2008 Yoshihiro Iwami, JASPIC

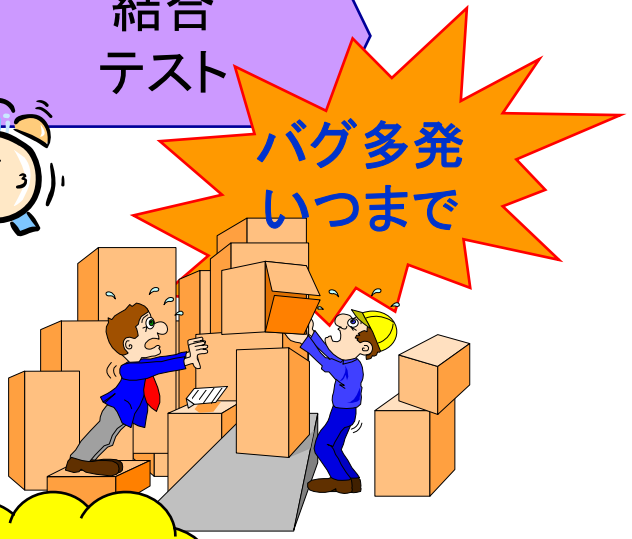
混乱したプロジェクトは、

要件定義 → 設計 → コーディング デバッグ → 単体テスト → 結合テスト

ソフトウェア委託先作業

確定遅れ開発期間に影響！

外注管理の強化を



「外注ソフトウェアにバグが多くテストが。。。」

「結合テストまでは順調だったのに？？？」

検収って受取っただけでは？？

対策会議

(c)2008 Yoshihiro Iwami, JASPIC

どう支援したか

- 以下を利用部門担当者に勧告
  - 「結合テスト、統合テストで苦労したいの？」
- レビューの強化
  - 委託先の要件理解度を確認
  - ソフトウェア設計書の精査
  - テスト仕様書の事前チェック
- 受入検収の強化
  - テスト結果報告のチェック
    - » 最終ではなく初回のテスト結果を見る
  - 重要機能モジュールの受入テスト実施
    - » これまでは、書面チェックで済ませていた

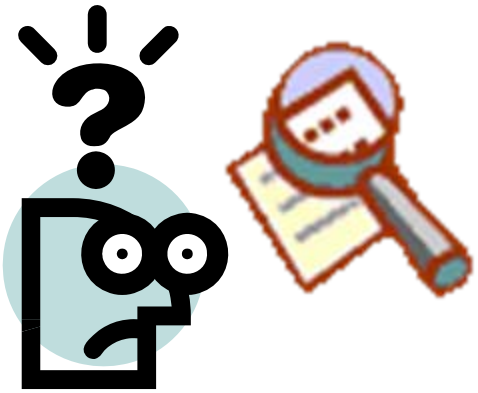
メンターが有効
・やり方を教える
・効果を体感させる

計測は力なり
・継続するともっと
強力に

(c)2008 Yoshihiro Iwami, JASPIC

レビューの強化

- 要件を文書化、委託先と合同でレビュー
  - 理解不足による誤り2件
    - » 弊社の説明不足が原因
- 仕様提示者とソフトウェア設計書をレビュー
  - 教育を兼ねて実施
  - 改修内容がきちんと設計書に反映されていなかった
- テスト仕様書の事前チェック
  - 弊社の指定フォーマットを使用



(c)2008 Yoshihiro Iwami, JASPIC

ソフトウェア設計書の精査

- 1次納入分をレビュー
  - 詳細改修要件100件当り約50件の不具合を発見
    - 「実担当者にはこれでわかるはず」との回答
- このままではバグ多発！
  - 設計書の再作成を指示

詳細仕様書 修理保管中の管理機能

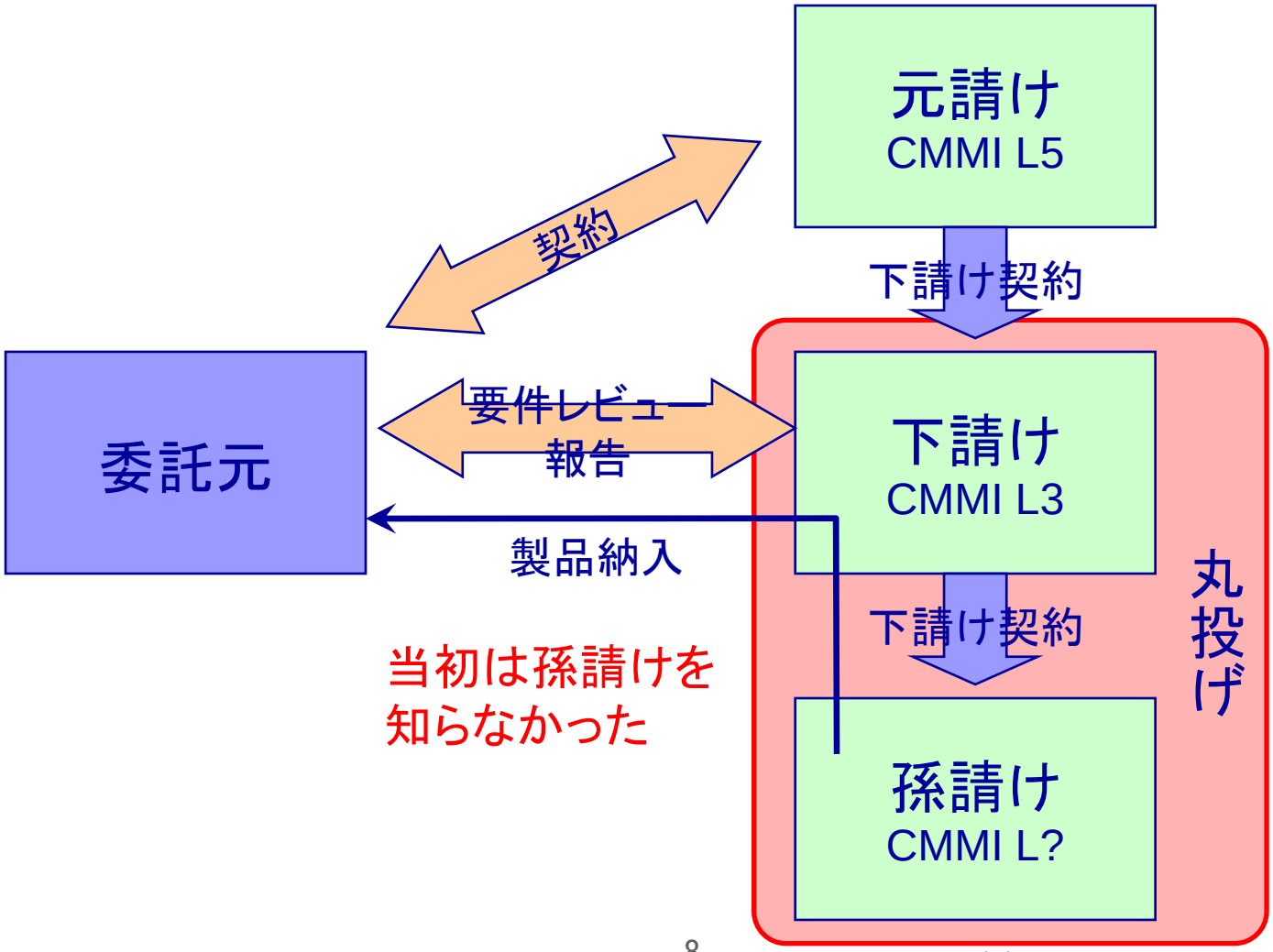
要望	1		CS-T画面で保管中の修理品を一覧表示できるようにする
		理由	保管中の修理品の管理をしたい
		説明	
	要望		入庫経過日数が表示される必要がある。
		理由	保管日数を管理できるようにしたい
		説明	
	(1)	■	現在の日付－入庫日を表示する。
	(2)	■	出庫日が入っているデータは表示しない。
	要望		各項目でソートできるようにしたい。
		理由	経過日数順にソートしたい
		説明	
	(1)	□	TrueDB Gridのヘッダーをクリックした時にソートできるようにしたい。
	要望		WEB公開用のコメントで絞り込めるようにしたい。
		理由	保管理由で絞り込んで管理したい
		説明	
	(1)	■	CS-Tの絞り込み項目にWEB公開用のコメントを選択できるようにす
	(2)	■	CS-Tの絞り込み条件でWEB公開用コメントを選択式で入力できるようにする。
	要望		保管ステータスで開示で絞り込めるようにしたい。
		理由	保管ステータスが保管中のデータだけを表示するため。
		説明	
	(1)	□	CS-Tの絞り込み項目に保管状況ステータスを選択できるようにす
	(2)	□	CS-Tの絞り込み条件で『保管中』のデータだけを検索できるようにす

詳細仕様書 完成予定日の管理

要望	1		完成予定日を二回以上変更した場合に、アラートを表示するようにする
		理由	お客さんとの約束した可能性がある作業者に注意を促して安易に変更しないようにさせる
		説明	
			<完成予定日変更画面での入力>
	(1)	■	完成予定日の変更ボタンを押した時に、その修理品の完成予定日が変更されているかチェックする
	(2)	■	完成予定日を変更した履歴があった場合はアラートを表示する
	(3)	■	アラートでは完成予定日を変更するか、変更しないかを選べるようにする
	(4)	■	完成予定日を変更する、とした場合は、社内用の変更理由を入力し、WEB用の変更理由を選択する
			<完成予定日の変更来歴の保存>
	(5)	■	完成予定日を変更した①修理番号、②変更者、③変更理由、④変更日時⑤変更前の完成予定日、⑥変更後の完成予定日を保存する（従来の仕様と同じ）
			<セット品のデータ更新処理>
	(6)	■	セット品がある場合は一覧表示する（検索に使用した番号は保持）
	(7)	■	一覧には、完成予定日が変更済みか、閲覧済みかを表示する
	(8)	■	完成予定日を変更する修理品を選択する
	(9)	■	完成予定日とWEBコメントと社内コメント（「修理番号〇〇の変更に伴い更新」）を変更できるようにする
	(10)	□	選択したデータは一括で更新する（エラーがあった場合は該当データを表示）
要望	2		WEB上で完成予定日を閲覧した時に、NERISで完成予定日を変更しようとするとアラートが表示されるようにする
		理由	WEB閲覧した場合は完成予定日を変更しないように注意を促すため
		説明	
			<WEBでの閲覧情報の取り込み>
	(1)	□	完成予定日を閲覧した①修理番号を保存する ②閲覧日時は、変更日時　項目に ③閲覧者（「WEB」などの作業者コードに固定）⇒ 変更者　項目に ④変更理由 には『WEBで閲覧』 を ⑤変更前の完成予定日と、⑥変更後の完成予定日には WEBで開示している完成予定日　を　保存する
要望	3		WEBで表示されている出荷予定日をNERISからでも見えるようにする
		理由	NERISの完成予定日を前提にしてユーザに説明すると、顧客のクレームにつながる可能性があるため
		説明	
	(1)	□	NERISの各画面で完成予定日以外に出荷予定日も表示する
	(2)	□	完成予定日が自動で変更される画面（受付、見積、進行処理画面）では出荷予定日も合わせて更新する（設定およびクリアーする時…）
要望	4		WEB上の出荷予定日を変更したい（NERIS側でWEB用のデータを強制送信する）
		理由	完成予定日を変更する場合はユーザに予め電話などで連絡することになっている しかし、ユーザに連絡が付かなかった場合は、WEBの出荷予定日を変更できるようにする必要がある
		説明	
			<データ転送処理>
		□	NERIS-01-01の仕様に従う

(c)2008 Yoshihiro Iwami, JASPIC

実は丸投げされていた！



(c)2008 Yoshihiro Iwami, JASPIC

受入検収の強化

- 初回テスト結果を解析
  - モジュールごとのバラツキ大
    - » 開発者によるのかを確認
  - 要注意モジュールを識別できた
    - » 初回テストでの不具合率20%以上
- 重要機能モジュールの受入テスト実施
  - 相当数の不具合を検出
    - » このまま結合テストに移行すれば大変だったかも
  - テスト環境について委託先から情報が来ない
    - » 妥当な環境下でテストしたのか？

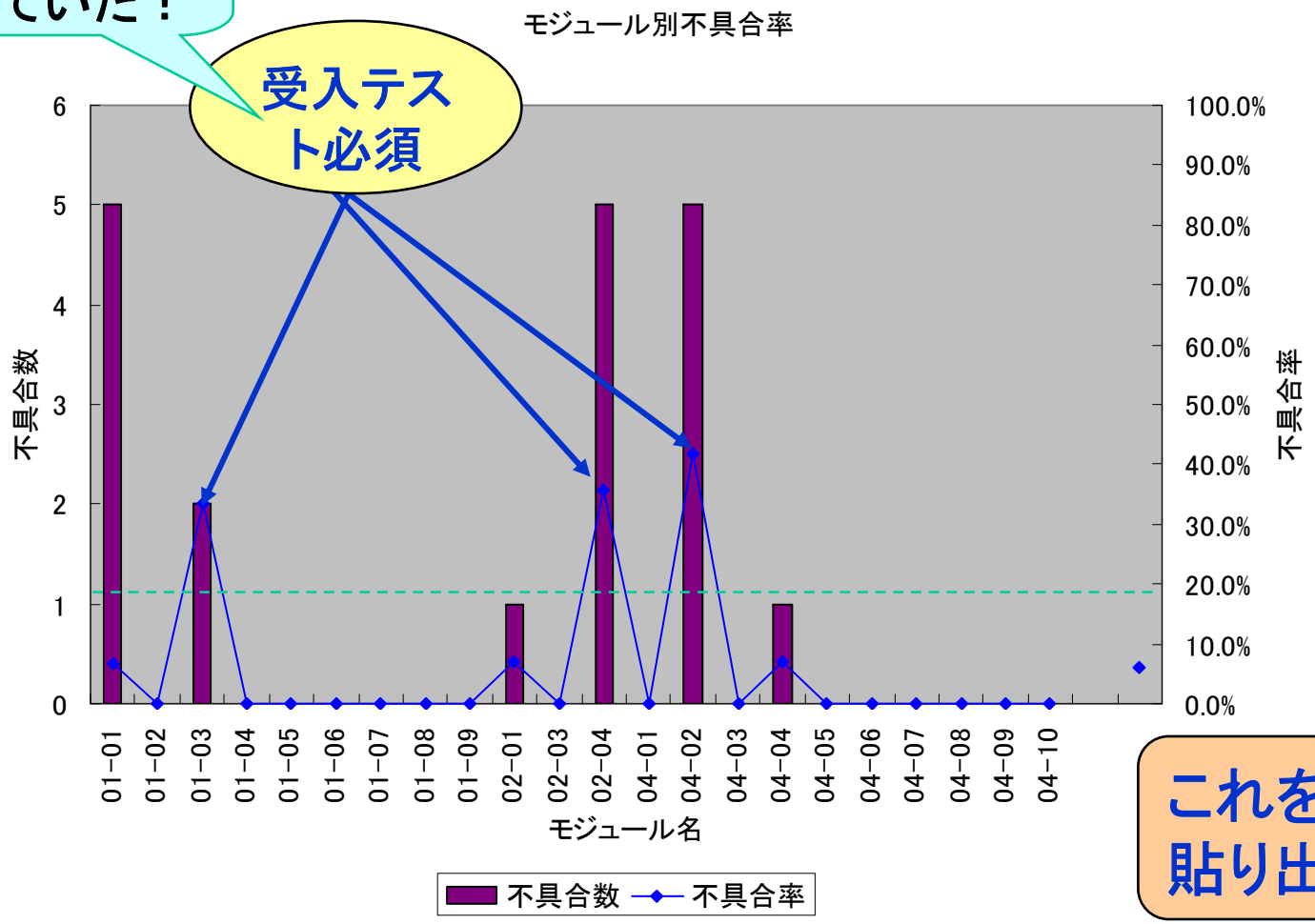
委託先にプロセスデータ提供を求めたが、収集していない、との回答！

(c)2008 Yoshihiro Iwami, JASPIC

モジュール別不具合グラフ

新人が担当していた！

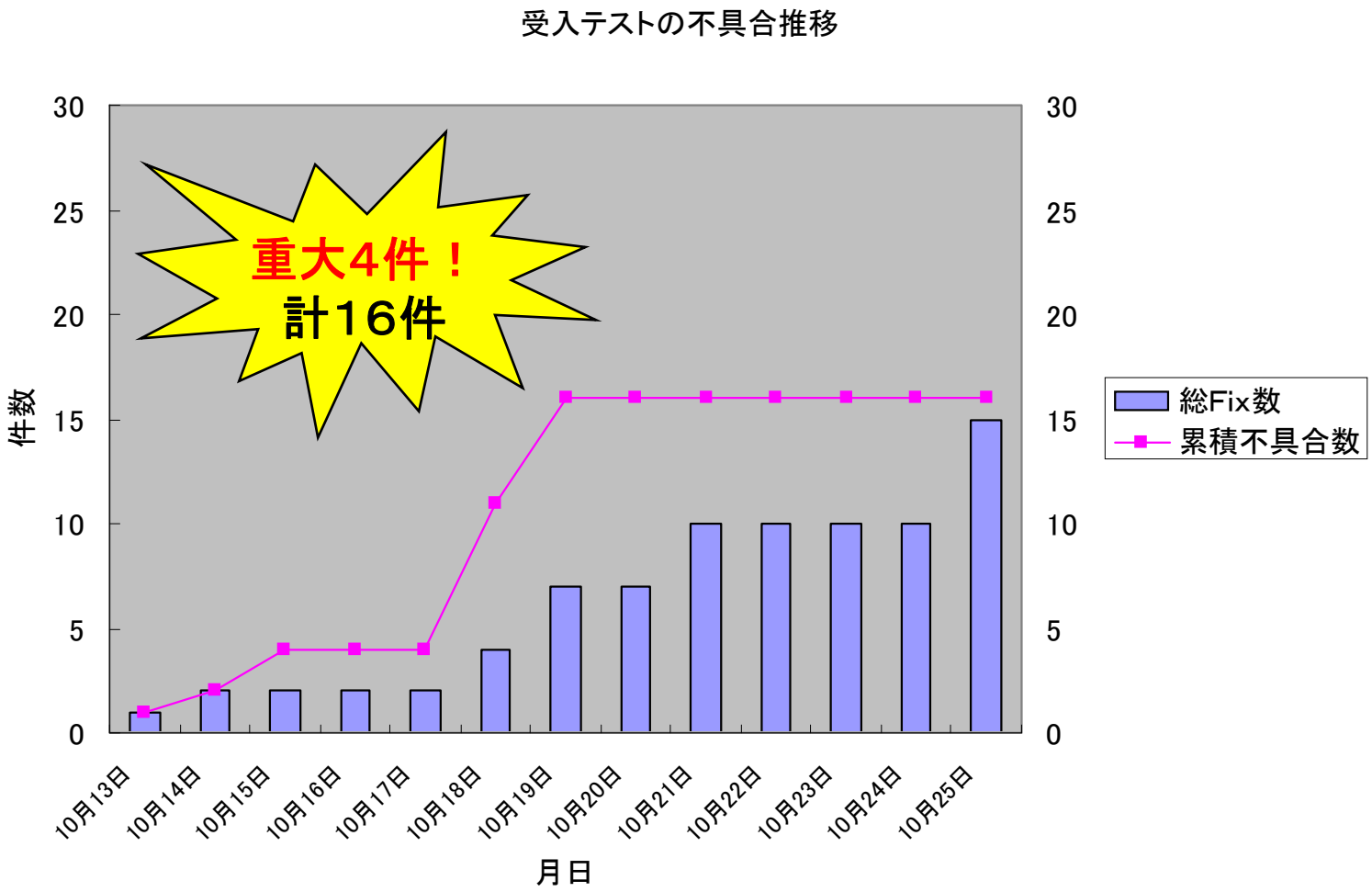
受入テスト必須



これを壁に貼り出した

(c)2008 Yoshihiro Iwami, JASPIC

受入テストで発見された不具合



(c)2008 Yoshihiro Iwami, JASPIC

改善効果

- 品質向上による日程確保
  - 以降のテストで大きなバグが出なかった
  - 無事、予定期日に稼動できた
- ソフトウェア受入検収プラクティスの強化
  - これまで(忙しいと)おざなりにしてきた受入検収の効果を現場が体感できた
- 委託先に緊張感が出てきた
  - 「あそこは検収が甘い」と委託先から甘くみられていた

(c)2008 Yoshihiro Iwami, JASPIC

残された課題

- 発注元(弊社): 「リスク(責任)回避」
  - 妥当な開発費見積もりがなかった
    - » 単価ベースでの値引き交渉に
    - » 委託先見積りを評価できず
  - A社ならば、大きいから安心
    - » 下請けされるのは承知。割高だが保険と思えば. . .
- 委託先: 「丸投げ」
  - 下請け構造
    - » A社(契約) ⇒ B社(A社子会社) ⇒ C社(開発担当)
  - C社へほぼ丸投げ
    - » 設計書レビューの痕跡なし(単純ミスがそのまま)
    - » C社名入りの文書をそのまま提出

(c)2008 Yoshihiro Iwami, JASPIC

残された課題の解決に向けて

- 外注の選定と維持
- 丸投げの背景
- 内製化の動き

(c)2008 Yoshihiro Iwami, JASPIC

ソフトウェア外注の選定、維持はむずかしい

- 外注を何で評価するか
  - 相見積りでコストの安いほう、は最悪
  - ソフトウェア品質が低いと単価が安くても結局は高くつく
- 実際に使ってみないと評価できない
  - まず小さな仕事を任せてみて、その結果で評価する
- ソフトウェア外注を長く使っていると劣化する
  - もう切られないと感じると、スキルの低い（安い）要員をアサインしてくる
  - 常に緊張感を与えておくとよい
- よい外注の見分け方（私見ですが）
  - 提示した要件を、外注自身の言葉で文書化しレビューを求めてくるか
  - 仕様書を確認して疑問点があると問い合わせてくるか
  - 担当窓口が開発者か

(c)2008 Yoshihiro Iwami, JASPIC

丸投げの背景

- 多重下請け構造
  - 政府調達でのワースト記録は7レベル
  - 重要社会システム開発での下請け利用を制限するほど深刻
- 多重下請け＝「丸投げ」ではないが。。。
  - 元請には、下請けの監理監督責任がある
  - ソフトウェア開発での丸投げは法律で禁止されていない
  - 大手ベンダーの「ゼネコン化」

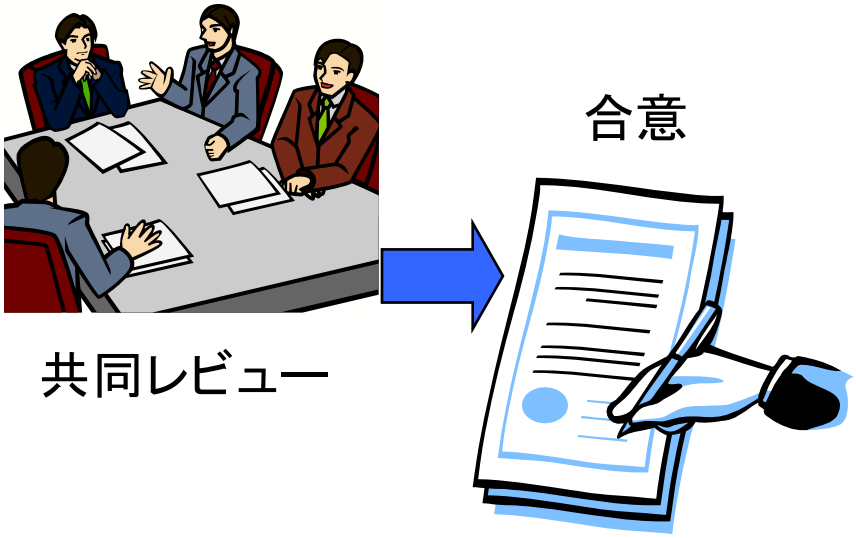
- 上流工程ほど「任したよ」に
  - ニーズだけ言えば、後は開発側で作ってくれるよ
  - 細かいことは、そちらで決めてよ
  - そして、ユーザテストで「こんなもの頼んでない！」

(c)2008 Yoshihiro Iwami, JASPIC

丸投げの解消

- 外注管理での事例
  - 子会社が外注を一元管理して、最適な外注を選定
  - 仕様提示と外注管理
  - 納入ソフトウェアの品質保証

- 「任したよ」からの脱却
  - 要件の確定、優先順位付けはユーザ(発注者)の責任
  - 開発側は「逆提案」して要件の合意をとる
  - 双方がこの合意を守る



- 発注側、受注側がともに各自の責任を果たす

(c)2008 Yoshihiro Iwami, JASPIC

内製化の動き

- 一転して内製化の動きが出てきた
  - Boeing　オフショア開発からソフトウェア内製に（約3,000名）
- 大手ベンダーが社内の開発組織を強化へ
  - 顧客が直接、下請けと契約（いわゆる中抜き）
  - このままでは仕事がなくなる
  - 社内に開発プロセスがないと、開発作業を管理できない
- パッケージメーカーの内製化
  - これまでは外注展開
    - » 品質が悪くリリース遅れに
    - » 社内に開発ノウハウがなくなる
      - 特にバグの原因分析とバグ修復のスキル
  - 開発ノウハウが蓄積できた
  - 品質が向上し、約束どおりリリースできた

単価が安くても品質が悪いと却って高くつく

(c)2008 Yoshihiro Iwami, JASPIC

米Boeing社の改善事例

- Boeing Parallel Model
- Personal Software Process (PSP)、Team Software Process (TSP) の導入
- ソフトウェア開発委託から内製化へ
- ソフトウェア品質向上でテスト期間を大幅短縮

外注部品が75%
その品質不良で
引渡しが15ヶ月
遅れると公表

機種	B777	B787
機体開発期間	4年	3年
ソフトウェア開発期間	7年	1年
ソフトウェア規模	7M steps	40M steps

(c)2008 Yoshihiro Iwami, JASPIC

ご清聴ありがとうございました

(c)2008 Yoshihiro Iwami, JASPIC