

システム開発保守QCD研究プロジェクト

2015年4月23日

一般社団法人 日本情報システム・ユーザー協会
システム開発・保守QCD研究プロジェクト

1. 2014年度の取り組みテーマと進め方

(1) プロジェクトの概要・方針

- ① システム開発・保守に関して、QCDを向上させるための仕組み、事例を議論
- ② JUASのソフトウェア・メトリクス調査(開発、保守)についての検討

(2) 本年度のテーマ

- ① システム再構築
- ② 短工期への対応策

(3) 部会活動の進め方

- ① 毎月第1火曜日 15時～18時(3時間)。4～5テーマの紹介。
- ② 部会メンバー各社から年に1回、「システム開発・保守のQCD向上」に関する自社の事例を紹介(質疑応答含め、30分～40分)
各社の最近の取り組み事例が紹介され、たいへん参考になる。
そして、部会メンバー各社のレベルアップに繋がる。
- ③ 合宿で内容を深掘り
2014年度は「システム再構築」、「短工期対策」

3. 部会活動経緯

回	開催日	発表・議論のテーマ
前年度	2014年 4月8日 (37名)	<ul style="list-style-type: none"> ①プロジェクト知見の組織展開—その具体策の紹介 ②SWEBOK2004との比較でのSWEBOK V3.0を概説する ③ERPパッケージ導入プロジェクト振り返り事例 ④ソフトウェアメトリクス調査報告速報
	2014年 5月13日 (35名)	<ul style="list-style-type: none"> ①俊敏なアプリケーション開発の実現に向けて ②大規模開発事例の紹介(QCD目標達成の要因) ③基幹系大規模システム再構築プロジェクトの振り返り ④システム障害事例に学ぶITサービス高信頼化教訓 ～再発防止のため、業界を超えた教訓共有の仕組みに向けて
1	2014年 6月3日 (44名)	<ul style="list-style-type: none"> ①基幹システム再構築の取り組み事例紹介 ②基幹システム再構築事例 ③「合宿テーマ:システム再構築」の進め方 ④「合宿テーマ:短工期対策」アンケートの説明 ⑤短工期への対応策について ⑥新年度部会参加者メンバーの自己紹介

回	開催日	発表・議論のテーマ 4
	2014年7月	7月11日(金)～12日(土)にかけて合宿予定であったが、台風のため中止。
2	2014年 8月5日 (43名)	<ul style="list-style-type: none"> ①【Final】IS企画・提案 教育に関する考察 要件・要求定義工程の品質向上 手戻りを少なくするための要求定義 ②リスクマネジメント再考 ③システム再構築アンケートのまとめと考察 ④短工期アンケートのまとめと考察 ⑤システム再構築への対策
3	2014年 9月2日 (44名)	<ul style="list-style-type: none"> ①OSS COBOLを活用したシステム再構築事例 ②ITダッシュボードについて ③システム再構築プロジェクトの事例紹介 ④開発業務の効率化への取り組みについて ⑤当社におけるシステム開発プロジェクトのQCD管理の取り組みと様々な課題
4	2014年 10月7日 (35名)	<ul style="list-style-type: none"> ①当社開発プロジェクトにおける品質向上への取組 ②設計工程におけるレビュー品質の向上について ③ビジネスへの貢献が求められる時代のシステム開発(プロセス)を考える ④WebPerformerによる短工期開発事例 ⑤合宿の進め方について システム再構築、短工期対応策

回	開催日	発表・議論のテーマ
5	2014年 11月7日 ～8日 (36名)	①再構築開発見積もり方式とその応用事例 ②楽々Frameworkのデモ 次のテーマで議論 「システム再構築」(3班)、「短工期対応策」(2班)について議論 (場 所は、湘南国際村センター)
6	2014年 12月2日 (39名)	①合宿の成果報告(5班の各班より報告) ②QCDの要だけど・・・ みんなPMにはなりたく・・・ない？ ③システム再構築プロジェクト外の事例 ④経営統合に伴うシステム統合事例のご紹介 ⑤QCD保守運営費用の投資時の考え方
7	2015年 1月13日 (41名)	①テスト工程における欠陥管理の取り組み ②システム再構築事例の紹介 ③ANA IT部門におけるプロジェクトレビューの取り組み ～チェックポイントレビュー(CPレビュー、P2レビュー) ④CACにおけるQCD向上の取り組み他 ⑤ソフトウェア品質の向上に向けた取組みについて ～ テスト標準化／派生開発 ～

回	開催日	発表・議論のテーマ
8	2015年 2月3日 (40名)	①コンセプトチェンジ ーシステム開発の作らずに作る ②品質向上に向けた取り組み ③システム再構築とその後の改修開発事例紹介 ④プロジェクト品質とリスク管理 ⑤ソフトウェアメトリックス調査2015に関する議論
9	2015年 3月3日 (38名)	①ソースコードメトリクス活用によるQCD向上施策の紹介 ②要件定義の充足度向上に向けた取り組み ③開発標準整備の取り組みについて ④サービスサイエンスによる顧客共創型ITビジネス ⑤大規模システム開発における最適な開発手法の適用 ⑥IT版失敗原因マンダラ図によるプロジェクト振り返り

4. 部会での各社の報告、議論より(1)

項 目	発 表 事 例
<p>超上流 要求定義、要件定義</p>	<ul style="list-style-type: none"> ・要件引き出し教育、演習(2.5日) 仮説検証ヒアリング教育 コーチング要素を取り入れたヒアリング教育 ・要件定義プロセス、作業成果物の見直し ・要件定義工程でのチェックシートの見直し ・要件定義に関する社内教育 <p>⇒要件定義工程の一段の強化が重要 : 今後の検討テーマの1つ</p>
<p>リスク管理</p>	<ul style="list-style-type: none"> ・リスクの識別 → リスクの分析 → リスクの優先順位付け → リスクの対策 → リスクの監視 のステップを踏む。 ・フェーズゲート(Phase Gate)の導入 ・ハードスキル(PMBOK, 共通フレーム等)のほか、ソフトスキル (問題解決法、交渉、モチベーション向上等)の重要性の認識
<p>見積もりモデル</p>	<ul style="list-style-type: none"> ・再構築開発の見積もり方式と応用事例(ジャステック) <p>改造型開発の見積もりは、</p> <ul style="list-style-type: none"> a. 母体となるソフトの調査分析、 b. 母体の品質を反映したコスト、 c. テストの巻き込みコスト <p>を考慮して行う必要がある。</p>
<p>プロセス改善</p>	<ul style="list-style-type: none"> ・CMMILレベル4、5への取り組み

項 目	発 表 事 例
プロジェクト管理 品質向上への 取り組み	<ul style="list-style-type: none"> ・各フェーズ終了時のレビュー (フェーズレビュー、チェックポイント・レビュー) ・設計工程のレビュー品質の向上 ・JUASメトリックスの採用(適正工期、工期比率、規模別予算超過率) ・第三者レビューを実施の会社の増加 ・Wモデルの採用 ・PMO、プロジェクトサポートグループの活動 標準の策定 プロジェクト評価会 知見の収集と展開(成功、失敗の教訓)――知見審議会 ・開発標準の整備→社内ポータルで一元管理 標準類、開発ツール、プロジェクト管理ツール、 プロジェクト管理情報も蓄積 ・派生開発に、清水吉男氏のXDDP(*)を採用 ・ソースコードの量や質を把握する取り組み(ソースコードメトリクス規程) 基本行数、差分行数(追加、削除、変更)
分割型開発	<ul style="list-style-type: none"> ・プロジェクトを一定の規模に分割し、開発(100～120月/システム) 要件定義、テスト、移行で有識者がボトルネックとなるため

(*)XDDP: eXtreme Derivative Development Process

項 目	発 表 事 例
システムの再構築	<p data-bbox="589 229 1787 335">今年度は、「基幹システム再構築」を主要テーマに取り上げたため、システム再構築の多くの事例紹介をいただいた。</p> <div data-bbox="1391 354 1877 429" style="border: 1px solid black; background-color: #e0ffff; padding: 5px; text-align: center;">約10社から事例紹介</div> <ul data-bbox="589 458 1362 1021" style="list-style-type: none">・基幹系システムの再構築・基幹系システム再構築・基幹系システム再構築・OSS COBOLを活用したシステム再構築・システム再構築・定期保安点検システム再構築・BI/ETLシステムの再構築・システム再構築とその後の改修開発 ・再構築開発の見積もり方式と応用事例
システム統合	<ul data-bbox="589 1049 1526 1149" style="list-style-type: none">・システム統合事例・システム統合におけるテスト工程の欠陥管理

項 目	発 表 事 例
生産性向上、コスト削減への取り組み	<ul style="list-style-type: none"> ・パッケージ活用による生産性向上、コスト削減 ・フレームワーク等のツールの採用による生産性向上、コスト削減
アジャイル開発 短工期開発 超高速開発	<ul style="list-style-type: none"> ・俊敏なアプリケーション開発 プロジェクト体制、開発支援ツール、テストの自動化、アジャイル、Wモデル、内製化、アプリケーション・ポートフォリオ、パッケージ利用等を組み合わせることで、開発の迅速化を図るのがよい。 ・WebPerformerを用いた短工期開発 <p>⇒アジャイル開発の事例は、まだまだ少ない。 JUASの今回のソフトウェアメトリクス調査とも合わせ、アジャイル開発については、今後もう一段掘り下げた検討が必要と思われる。</p>
パッケージ、クラウドの利用	<ul style="list-style-type: none"> ・「できるだけ作らずに作る開発」のコンセプトを変える ーパッケージ、クラウド等をまず検討

5. 合宿の概要

5-1. 概要

(1) 合宿のテーマ

「システム再構築」

「短工期対策」

(2) 合宿の進め方

①日程 11月7日(金)午後～11月8日(土)午前

(当初予定では、7月に合宿の予定であったが、台風で延期)

②場所 湘南国際村センター

③参加者 36名

④合宿テーマ展開事例

「再構築開発見積み方式とその応用事例」のご説明

⑤住友電工情報システムより、楽々Frameworkのデモ(短工期対応策の1案)

⑥システム再構築のリーダー、短工期対策のリーダーより

それぞれのチームの進め方につき、ご説明

⑦システム再構築チーム(3班)と短工期対策チーム(2班)に分かれて、

それぞれのテーマを議論。

最後に、各班のまとめを報告し、情報を共有

また、欠席されたメンバーもあるため、次回会合(12月)で合宿の成果を報告。

(1) 事前準備と進め方

1) QCD部会メンバーに、システム再構築のアンケートを依頼し、16社49件の回答を入手。

①再構築が成功したか、失敗だったかー 成功事例 31件、失敗事例 18件

②再構築の失敗要因

- | | |
|---------------------|-----|
| a. 要件定義(スコープ)に関する要因 | 10件 |
| b. 開発体制の不備 | 6件 |
| c. 現行機能の調査・確認が不足 | 5件 |

③成功要因

- | | |
|-----------------|----|
| a. 十分な現行調査 | 9件 |
| b. 開発体制 | 9件 |
| c. PMの技量、マネジメント | 7件 |
| d. 本番稼働前のテスト | 5件 |

2) 合宿の進め方

①再構築アンケート結果の説明

②自己紹介後、各社事例を説明

③各社事例、全体アンケートから成功要因、失敗要因を分析

④成功するため、失敗しないための対策を検討

(2)システム再構築の定義

詳細	リホスト	リライト		リビルド	
		ロジックを流用し 新しい言語で作り直す方法		ビジネスプロセスから見直し システムを構築する方法	
		単純 マイグレーション	システムリフォーム	スクラッチ	パッケージ +アドオン
	ソースにできるだけ 手を加えず 移行する手法	変換ツールによる 単純コンバージョン	リバースエンジニアリング 人が対応するので 対応言語が多く制約が 少ない	現行調査+ 要件定義	現行調査+ Fit&Gap
OS	置き換える	置き換える	置き換える	置き換える	置き換える
DB (*)	置き換える	置き換える	置き換える	置き換える	置き換える
言語	変更無し	ソース to ソース 変換ツールで書き換え	ソース to ソース 変換ツール + 人手で書き換え	新仕様 to ソース	パッケージ+新仕様 to ソース
仕様	変更なし	変更なし	変更なし	変更あり(新仕様)	パッケージ + 新仕様

(*) 同じタイプのDBへの置換・拡張と異なるタイプへのDB置換の方法がある。

Aチーム	<p>事前アンケートで失敗要因上位の「現行システム調査不足」を取り上げる。</p> <p>①旧システムの調査・分析が、システム再構築には必須。</p> <ul style="list-style-type: none">・そのため、調査・分析の進捗度、十分度を定量的に把握することが重要。・旧システムの練度、旧システムの保守用設計書の最新度の把握が必要。 <p>②課題としては</p> <ul style="list-style-type: none">・調査・分析の期間、コストの承認が得られるか。・旧システムの品質(設計書整備、整合性、潜在バグ等)の可視化・有識者(ユーザ側、情シス側)の確保、育成
Bチーム	<p>・マインドマップで、Bチームメンバー会社の成功・失敗要因を分析 その上で「現行システム分析」、「開発体制(ユーザ、システム部門、ベンダー)」を議論。</p> <p>①再構築については、現行システム分析を1工程と定義。</p> <ul style="list-style-type: none">コスト、基幹の確保、タスクを定義(ソース解析、設計書見直し等) <p>②体制については</p> <ul style="list-style-type: none">・ユーザ側-ToBeが分かり、仕様の決定ができる人 途中で、ベンダーに丸投げせず、最後まで主体的に関与が必要。・ベンダー-現行業務、現行システムに精通していること・システム化計画に体制を明記。
Cチーム	<p>・再構築のリスクとして、①経過年数が長い、②規模が大きい、③要員確保、④その他に分けて、分析。</p> <p>①経過年数が長いリスク(a.ドキュメントが古い、b.移行のリスク、c.ハード/ソフトの保守切れ)</p> <p>⇒a. 経験のあるユーザ(ステークホルダー)の参画</p> <ul style="list-style-type: none">b. 新旧システムの精査プログラム、旧システムとの並行稼働期間の確保c. 再構築のサイクルを普段から決めておくこと

Cチーム
(続き)

②規模が大きい場合の対策

- a. ステークホルダーが多くなる⇒ステークホルダー分析、コロケーション(場所の確保)
- b. **関係システムとの連携の複雑化⇒再構築の範囲を最適化**

③要員確保の対策

- a. **保守メンバーを再構築に参画させる⇒再構築中の保守の凍結 等**
- b. 旧システムを分かった人材が確保できない
⇒再構築の期間を決めておく(例えば、10年毎)、業務部門とIT部門のローテーション

④その他

- a. 再構築は、新規開発よりも軽視されがち
再構築のための説明をしっかり行い、予算の確保と体制の確保を行う。
再構築の難しさは、旧システムよりも要求仕様のレベルが高くなる。
一方、現行通りでよいところは、現行通りと言われる。
- b. 現行通りを禁句にし、現行通りの要件を可視化し、設計に織り込む。



1. 旧システム(現行システム)の調査・分析に時間と費用を確保することが重要
2. 現行システムの分かっているユーザ側の人、開発側の人々の参画が重要。
開発側は、保守メンバーを再構築に参画させる。
3. 現行通りを禁句にし、現行通りの要件を可視化し、設計に織り込む。
4. 規模が大きい場合、再構築の範囲を明確にすること。
例えば、システムの規模を一定規模に分割し、分割スタートを検討するなど

5-3. 短工期対策

(1) 事前準備と進め方

1) QCD部会メンバーに短工期に関するアンケートを事前に依頼

2) 合宿の進め方

2チーム(Dチーム、Eチーム)に分かれて、以下を実施

- ①自己紹介後、各社事例を説明
- ②短工期開発のアンケート分析
- ③短工期実現方法の検討(利点、欠点の洗い出し)
- ④リスク、課題の検討
- ⑤各チームの考え方をまとめる。

Dチーム	<p>(1)短工期のためには、開発工数を削減する。その対策として</p> <ul style="list-style-type: none">①ユーザ側の積極参画を促し、要求内容を精査(今やる必要があるか等)②パッケージを利用する③作る量を減らす<ul style="list-style-type: none">・リポジトリの活用、ツール類の活用・例外処理のカバー率を少なくする－人間系で対応した方がよいことは人間系で対応 <p>(2)それ以外</p> <ul style="list-style-type: none">①フェーズを分ける、アジャイル開発②並行開発③開発、着手の「さしみ化」
Eチーム	<p>チームメンバー各社の事例を分析し、短工期実現のための対応策を検討</p> <ul style="list-style-type: none">①設計時のレビュー回数、期間を短縮する。そのために、<ul style="list-style-type: none">⇒a. システム規模、錬度でレビュー回数を決めておくb. 要件ヒアリングのためのチェックシートを準備する②設計工程への高錬度要員の大量投入。 しかし高錬度要員の確保が課題。③CCPM(Critical Chain Project Management)の採用 確かに、工期は短くなる。 一方、適用にはメンバーへの説明、合意、 ストレスをためない環境作り等に留意が必要。④ユーザ側の協力が不可欠(要件を出すこと、決めること)⑤設計書、コードをできる限り書かないこと、人手を介さないこと。 そのためのツール類の採用。

短工期対策1

A: 作らない

そのまま使用する。(追加開発はしない)

パッケージの利用

FitGap分析結果は、運用で対応することを第一に考える

(参考)パッケージ費用の1/3(注1)以上の費用をかけて追加開発をする必要がある場合はスクラッチ開発をする

クラウドサービスの利用

クラウドサービスで使えるものは使う

汎用型ソフトの利用

汎用ソフトを個別には作らない(ワークフロー、BPM、印刷システム、EDIなど)

類似システムの再利用

再利用できるように、システムを標準化しておく(画面数推移、レイアウト、ワークフロー、...)

仕様の標準化、単純化

人の判断力の活用(複雑な条件・変更の多い条件は取り込まない)

開発要件定義の段階で機能を削除

- ・機能の使用頻度・効果と開発費用のバランスをチェック
- ・利用者目線と運用管理者目線の差を利用し仕様を縮小する

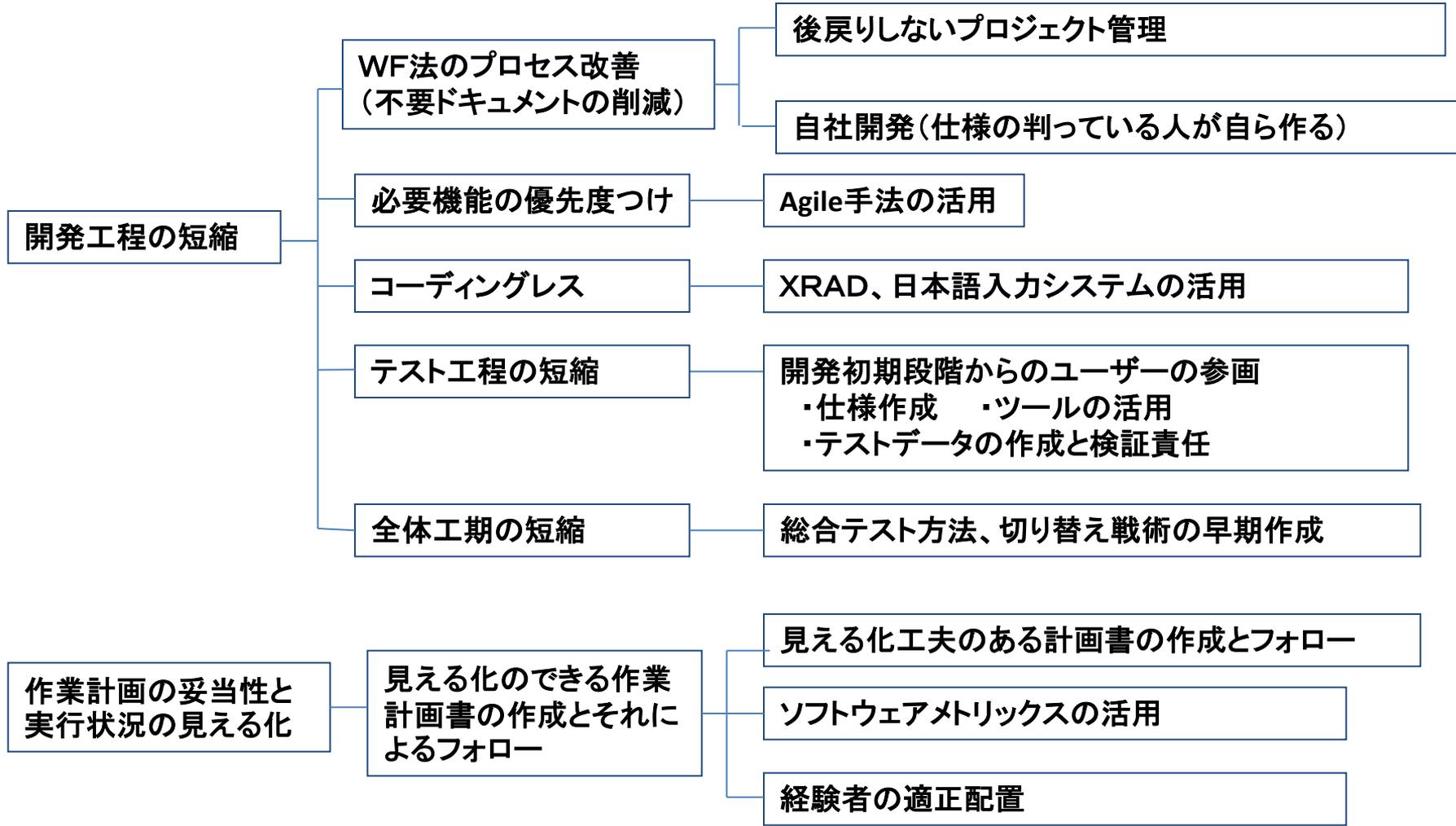
運用標準の明確化

運用者目線での運用計画書の作成

- ・誰が、いつまでに、入力するのか、明確化
- ・訂正作業のタイミングを含めて情報と現物の不一致をなくす(現場入力)
- ・入力ミスした場合でも直ぐにミスが判るように、作業の流れを一貫して管理する仕組みの取り込み
- ・例外作業が発生した場合の処置の取り方の指示・標準化
- ・プログラムミスの早期発見対策

短工期対策2

B: 早く作る



6. ソフトウェアメトリックス調査2015(暫定ご紹介)

6. 1 今年度は、アジャイル型と超高速開発型に絞り、調査。
ウォーターフォール型、U字開発型は、調査対象外とした。

6. 2 アジャイル型の設問

- ・プロジェクト特性、採用理由
- ・システム規模
- ・開発工期、スプリント回数、開発工数
- ・開発言語
- ・PMスキル
- ・管理手法(スクラム、XP、ユーザーストーリー、日次ミーティング他)(複数回答可)
- ・運用保守対応
- ・ドキュメント作成時期
- ・QCDのWFと比較しての評価
- ・ユーザ満足度
- ・技術者満足度
- ・開発チーム構成

6.3 超高速開発型の設問

- ・プロジェクト特性、採用理由
- ・開発ツールの名称
- ・開発期間(各工程ごとの期間)
- ・ツールの適用工程
- ・工程別投入人月
- ・開発工数、年間保守工数
- ・開発チーム構成

6.4 今回の調査の難しさ

(1) 実例が少なく、データ集めに苦勞

(2) 調査項目の設計、及び分析手法

何をデータとして集めて知見をまとめるか？

(3) アジャイル、超高速開発の品質データの取得は難しい。

特に、アジャイルは繰り返してテストを実施のため、
欠陥数(納入後の欠陥数)をどうとらえるか、など

詳しい紹介は、JUASの報告書をご参照ください。