

2014年11月20日発行

VERISERVE NAVIGATION

ベリナビ

2014
Autumn



OSSリスク管理ソリューション

Series

D-Case 入門

品質の真実

ベリナビ

Autumn 2014

「ベリナビ」について

当冊子は株式会社ベリサーブが情報発信および情報活性化を目的として主に社内向けに手作りで発行していたものを、お客様向けに再編集したものです。内容的には一部社内向けの表現になっておりますが、当社の取り組みを広く知って頂きたいという思いから、お客様向けとして改めて発刊する運びとなりました。

様々な分野で検証に携わるベリサーブが、「今とこれから」を皆様にお伝えます。

ベリナビ編集部

Contents

3 品質の真実

4 連載 | Series D-Case入門

10 インタビュー | Interview

現場スペシャリストに聞く、
検証の将来について

14 ソリューションを通じて現場貢献！ ユーザビリティPRJ展開 事例

D-Case PRJ 展開事例

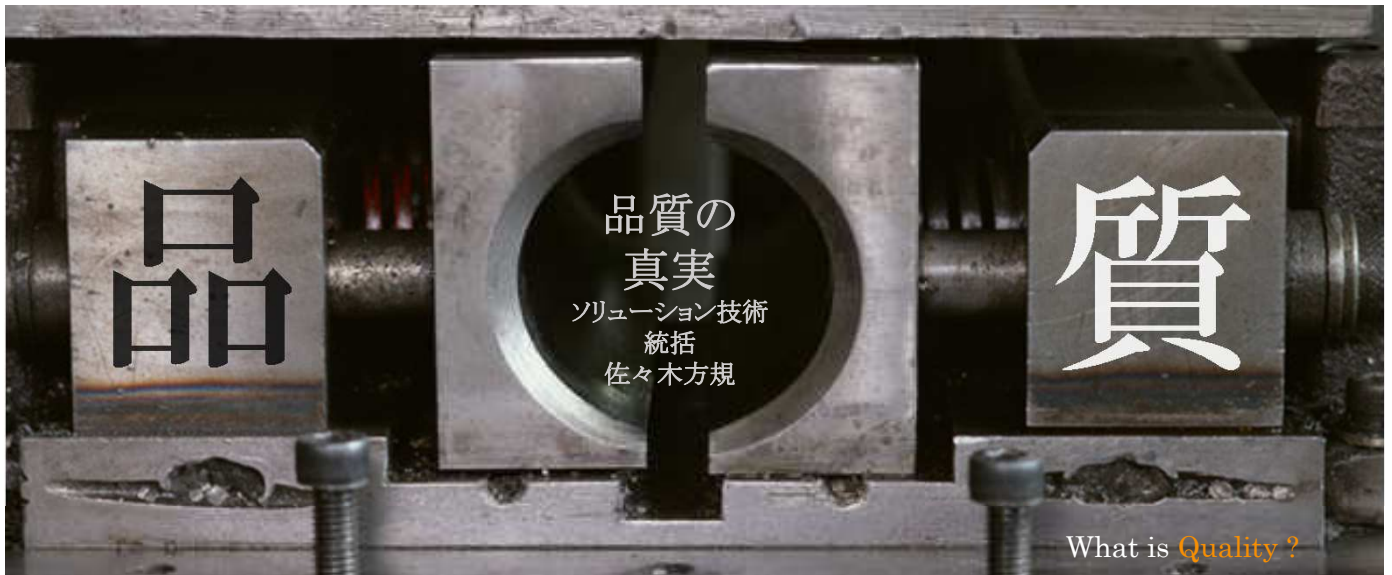
18 派生開発の世界と検証

20 検証理論とその手法の活用について テスト体系

21 『IT検証フォーラム2014』 参加レポート

24 クラウド・コンピューティング

26 モヤモヤするテスト用語



品質には多面性の特徴があり、柔軟性が重要です。これを解説するために、品質の説明でよく利用される狩野モデルを用います。狩野モデルでは、下記のような5つの要素が定義されています。

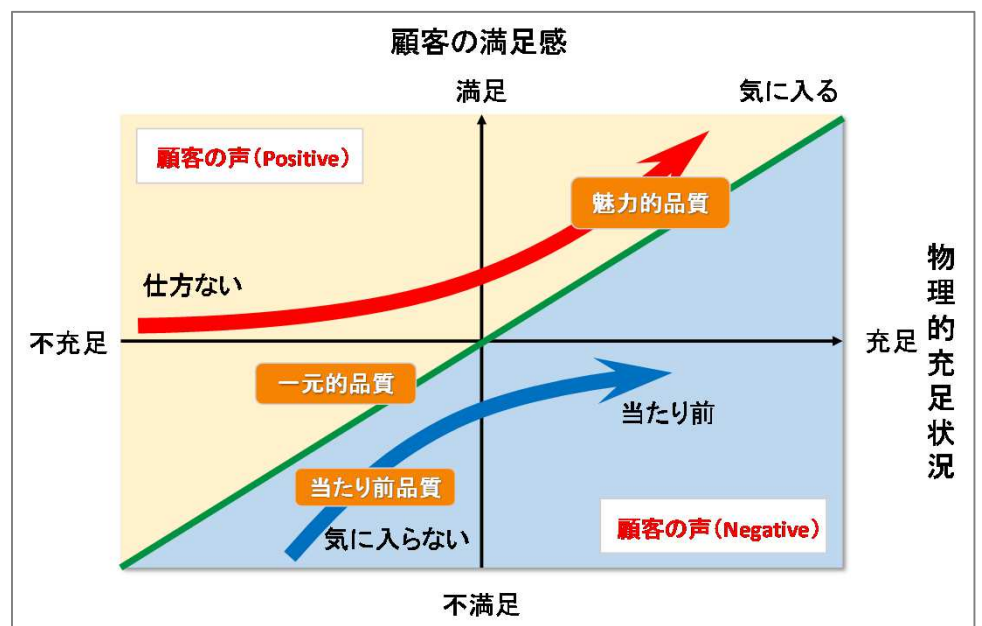
- ・魅力的品質要素:それが充足されれば満足を与えるが、不充足であっても仕方がないと受けとられる品質要素。
- ・一元的品質要素:それが充足されれば満足、不充足であれば不満を引き起こす品質要素。
- ・当たり前品質要素:それが充足されれば当たり前と受け止められるが、不充足であれば不満を引き起こす品質要素。
- ・無関心品質要素:充足でも不充足でも、満足も与えず不満も引き起こさない品質要素。
- ・逆品質要素:充足されているのに不満を引き起こしたり、不充足であるのに満足を与えたりする品質要素。

“当たり前品質”や“魅力的品質”という用語は聞き覚えがある人も多いと思います。狩野モデルでは2次元の4象限で表現されていますが、“当たり前品質”や“魅力的品質”を維持するためには、**時間軸**を加える必要があります。従来のハードウェアでは、アップグレードの概念が少なく、製品の機能を満足し続けるためには、新製品を消費者が購入し続ける必要がありました。

ソフトウェアではその概念を破壊し、ランチェン（ランニングチェンジ(Running Change)）を行いながら、消費者の要求を満足し続けることを可能としました。

モバイルアプリなどに代表されるサービス開発では、変化する要求に対応するための常に変更開発が進められています。

ソフトウェアの品質を維持するとは、常に変更に対応していると言っても良いと思います。要求に対応し続けるためには、保守性の**変更容易性**(変更しやすい度合い)、**安定性**(変更しても影響が少ない度合い)など、柔軟性が重要になります。テスト視点においても、デグレードや変更への影響視点などの考慮が重要になります。



皆さんは、ソフトウェア品質特性をご存知ですか。知っている人は多いと思います。では、機能性、信頼性など x x 性の"性"はどのように考えられていますか？

よくテスト観点でソフトウェア品質特性を利用していると聞きます。ここでいう"性"には、定性的な定義であって、定量的な定義ではありません。つまりテストの目的であって、目標(値)ではないので、基準としての利用できません。

では、「信頼性を満足する」、「使用性を満足する」とはどんなことでしょうか。

xx性と表記しているものを筆者は、**特性の度合い**としています。例えば、機能性を「ソフトウェアが指定された条件下で、明示的・暗示的に必要とされた機能を提供する度合い」となります。つまり度合いとは1or0でなく、「まあまあこれでいいよね。」という曖昧な基準も含まれています。こんな曖昧な定義をテスト観点に利用しています。

これがテスト観点として成立する背景は、テスト設計者が定量的な目標値を経験から算出したり、仕様やヒアリングから取得することに基づいているからで、この曖昧な目標値に

対する説明責任が果たせると、テストの妥当性につながると考えています。さて柔軟性も定性的な"性"がついています。柔軟である度合いとは、どんなものでしょうか。まず、ソフトウェアシステムのステークホルダーを考えてみましょう。

ソフトウェアシステムのライフサイクルでは、5者が関係しているといえます。5者とは、**発注者・開発者・運用者・保守者・利用者**です。

発注者は、利用者に対して許容範囲を大きいシステムで多くのユーザを取り込むことを求めます。

運用者は、ダウンタイムが少なくなるような多様な運用方法を求めます。

保守者は、利用者のニーズに対応しやすい改変しやすいシステムを求めます。

このように柔軟性は、許容であり、多様(可用)であり、可変であり、ステークホルダーでも各々の異なる特性を持ちます。柔軟性は多面性をもつため、柔軟である度合いが重要です。今回は、柔軟性の視点を説明しました。次回は、その方法論(品質構造)です。

D-Case 入門

連載 | Series

D-Caseの基本の「キ」

近年、欧米で高い安全性が要求されるシステムの開発運用において、提出が義務付けられるまでに普及している安全性ケース(*)として使われている「D-Case」について、ご紹介します。

D-Caseとは？

「D-Case」の「D」は、「Dependable」の頭文字で、「Dependability Case」を指していますが、要するに、高品質、高信頼かつ、高安全であることを証拠を元に確認する議論とその手法と考えれば良いと思います。
ちなみに、D-Caseの公式サイトであるwww.dcase.jpには、以下のように記載されています。

D-Caseの定義：

D-Case とは、システムのディペンダビリティをシステムに関わる人たち（ステークホルダ）が共有し互いに分かり合い、そのディペンダビリティを社会の人々にわかってもらい、説明責任を果たすための手法とツールです。

* 安全性ケース(Safety Case) とは、テスト結果や検証結果をエビデンスとしてそれらを根拠にシステムの安全性を議論し、システム認証者や利用者などに保証する、あるいは確信させる(assure) ためのドキュメントです。

ディペンダビリティの定義:

ディペンダビリティ (英: dependability) とは、一般的な信頼性にとどまらず、たとえ一部が壊れても残りの部分でうまく働くといった自立的自己修復的な動作をさす概念である。大規模集積回路、コンピュータシステムなどで使われる言葉である。

従来の「歩留まりを高めて欠陥を作らない」「検査によって欠陥品を外に出さない」ことがこれまでの信頼性とすれば、ディペンダブルな技術は「欠陥があってもへこたれない」技術ということになる。

マルチコア・マルチチップで高性能並列組込みシステムを構築して、冗長性を持ったソフトウェア分散共有メモリ機構により実現する例がある。

D-Caseの考え方:

D-Caseは、対象システムが「ディペンダブルである」、「へこたれない安心感がある」というシステム製作者の主張に対して、それを裏付ける証拠を論理的な説明によって示せるようにし、主張を否定する訴えや言いがかりに対して、きちんと責任を果たしていることを明確にしようとする考え方です。



D-Case図とは:

D-Case図は、GSN(Goal Structuring Notation)図とも呼ばれています。主張のゴールを頂点とした議論の展開を図で示すもので、これによって議論の抜け漏れを確認し、合意していくための1つの見せ方の手法です。

D-Case図で用いるチャートの種類と意味を、表-1に示します。

名称	図式要素	説明
ゴール		システムが達成すべき性質を示す。下位ゴールや戦略に分解される。
戦略		ゴールの達成を導くために必要となる論証を示す。下位ゴールや下位戦略に分解される。
コンテキスト		ゴールや戦略が必要となる理由としての外部情報を示す。
未展開要素		まだ具体化できていないゴールや戦略を示す。
証跡		ゴールや戦略が達成できることを示す証跡。
コンテキスト関係		コンテキストとゴール、戦略との関係。
ゴール関係		ゴールと戦略、ゴールと証跡の関係。

山本修一郎教授の「要求工学」サイト

(<http://www.bcm.co.jp/site/youkyu/youkyu93.html>)より

表-1

D-Case図は、ゴール分析という要求工学の研究成果から生まれたものです。詳しくは、要求工学のサイトに連載されています。(www.bcm.co.jp/site/youkyu/index.html)

システムにおける一般的ゴールは、「要求通りの機能・非機能要件を満足できている」ことです。そして、このゴールに対して証跡を使って具体的に説明していくための論理展開ツリーの図がD-Case図となります。この論理展開の途中で、証跡となるテスト結果やテスト計画がないとか、そもそも、暗黙の要求については、曖昧なのでシステムの設計に、考慮できていないというようなことが、さまざまに起こります。つまり、この図を作成することによって、そうした抜け漏れに気が付くことができるということです。

D-Case図のイメージサンプル:

D-Case図のサンプルを図-1に示します。

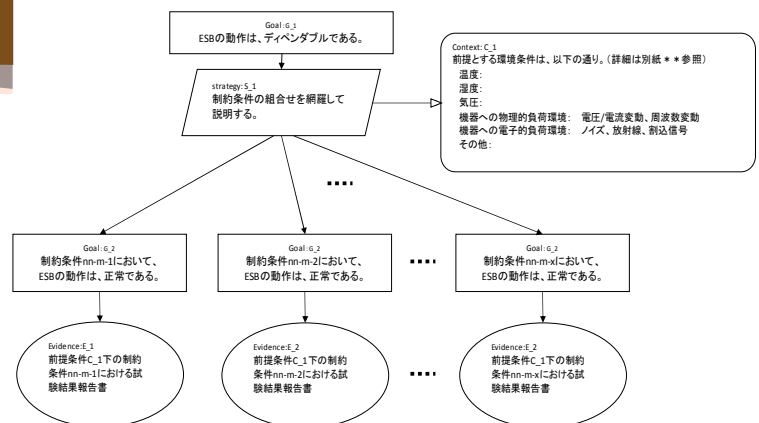


図-1 D-Case図のサンプル

D-Case手法が注目される背景と期待されている効果

今回は、D-Case手法が注目されている背景と期待されている効果をご説明します。

D-Case手法が注目されている背景:

D-Case手法は、自社製品やサービスの正当性の主張を、具体的に証拠を示して説明を展開する手法です。この手法が注目されている理由は、いくつか考えられます。

- ①事故や災害が発生してから、原因分析をやっていたのでは信用失墜で、企業は事業継続できない。
- ②グローバルビジネス展開において、1つの安全性欠陥といえども、致命的損害を発生しうる。
- ③企業は、CSR(社会貢献)を実践していることを 自社製品を通して示す説明責任があると認識されている。
- ④これまで「日本品質」と呼ばれ、世界一の品質と謳われている製造業は空洞化し、品質改善が進んでいない。
- ⑤「おもてなし」のサービス精神は、ソフトウェアに具現化する知恵やアイデアの企画力が乏しい。

上記の①～③は、自動車メーカーのリコール問題のように、世界中で訴訟問題を引き起こすことを少しでも抑止する必要から、極めて重視されています。そして、組込みマイコンでは、ディペンダブルVLSIの研究と実用化がかなり進んできました。(参考URL = <http://dvlsi.jst.go.jp/>)

しかし、自動車の将来は、単に自動車が知的にロボット化するだけではありません。ITS(高度交通社会)システムは、自動車と道路システムとの連携を前提として、安全性と快適性を実現しようとしています。このシステムは極めて多くの機能を持った複雑なシステムとなります。

しかも、その中には、天候や自然災害なども想定し、電力系統や通信系統のダウンや切断といった事態にも耐えられる高いディペンダビリティが求められます。誤作動により、交通事故を誘発し、人が死亡する事態になってはいけません。

このような高い安全性、信頼性の要求を満足する品質をどのようにすれば実現できるのかということを考える際に、D-Case手法が役立ちます。そして、従来の安全分析手法との違いは、繰り返し実証実験を行ってデータを積み重ねるという方法では、永遠に実用化できないというスパイラルから脱出して、早く実用化できることです。

もちろん、D-Case手法は完璧な手法ではありません。単なる議論展開の方法にすぎませんが、重要なことは、議論を尽くしたということ、証拠をもって説明できること、議論を尽くした内容は、そのステークホルダーと合意できていることです。従来の安全性分析やリスク分析でも、議論しているはずですが、合意をとって説明できているかどうかの違いがあります。いつでも説明できるという状況になっていれば、万一、事故や災害が発生しても、すぐに責任ある対応と説明が可能ですが、そうっていない場合、説明は「調査中」とか「対応中」という表現で終わってしまい、被災者からは、何が原因で、誰が責任を取ってくれるのかが不明となるだけでなく、あの企業は対応が遅いとか、対応が悪いという判断になっていきます。

次に上記④、⑤について考えてみましょう。これまで日本の製品競争力を確保していた「品質」は、自動化やIT化によって、差別化ができなくなってきました。また、製造自体も海外で行うため、品質検査の合格基準を厳しくしているだけの差になってきています。また、システムやソフトウェアの企画段階での「品質つくりこみ」の観点からは、まだ確立されていないところが多いばかりか、独創的な製品を積極的に採用することも、ベンチャー以外では、ほとんどなくなってきました。



交通事故、死亡事故抑止

これは、経営者がリスクを取らないという問題も含みますが、何を事前に検証しておけば良いのかもわからない技術部門の中間管理層や、技術の現場から離れた瞬間に技術の進歩から取り残された、管理者層にリスク分析をさせているという問題が背景にあります。こうした問題を放置しておく、いずれ競争力を失い、その企業は、クリエイティブな商品開発が活発にできる企業に追い抜かれて消滅します。このような最悪の事態になってしまう前に、企画段階でのその製品の品質をしっかりと議論し、商品戦略と共に品質戦略(*)を立てて、ビジネスに勝ち残っていく挑戦を繰り返し、企業の企画層が、経営層を説得し、事業部門や製造部門に活力を与える必要があります。

D-Case手法に期待されている効果:

このような企画工程での品質議論において、D-Case手法は、非常に有効です。すべてが議論できるわけではありませんが、課題が明確になることで、何の対策をすれば先に進めることができるか、何が明確になれば、品質を確保することができるかといったことがわかってきます。

開発工程に入る前の企画工程でD-Case手法を用いるということは、ユーザーの運用現場で考えているという事と実態は同じですが、企画工程では、開発計画として、予算、納期、体制を画策する作業が含まれます。ここで、費用と納期を決めています、それは暗黙に品質も決めていることを意味します。

要件定義工程で決めるのは**テスト計画**ですが、テスト環境構築という費用と時間のかかる見積は企画工程で対応していますので、ここで曖昧な品質方針と品質戦略のまま開発工程を進めてレビューで不備を指摘されたとしても、もう開発を止めてやり直すということは、企業としては何としても避けたいところです。品質方針は、ISO9000の理念に準拠していれば、具体的には、従来と変わりありませんが、**品質戦略は、開発する対象、新規か保守開発か、規模や重要度などで変わります。**いきなり品質戦略(*)を考えろと言われても無理かもしれませんが、運用する状況を想定し、どんな障害が最も危険な事態を引き起こすことになるのか、それをきちんと回避できているか確認する方法は何か、どうやって確認すれば良いかということを考える際、メンバーと議論の過程をD-Case手法で記録に残すことで、議論をトレースすることが可能になり、漏れや不足に気付いて改善していくことができます。



(*)品質戦略の参考:<http://qualab.jp/materials/SQIP-QAdirector.130703.pdf>

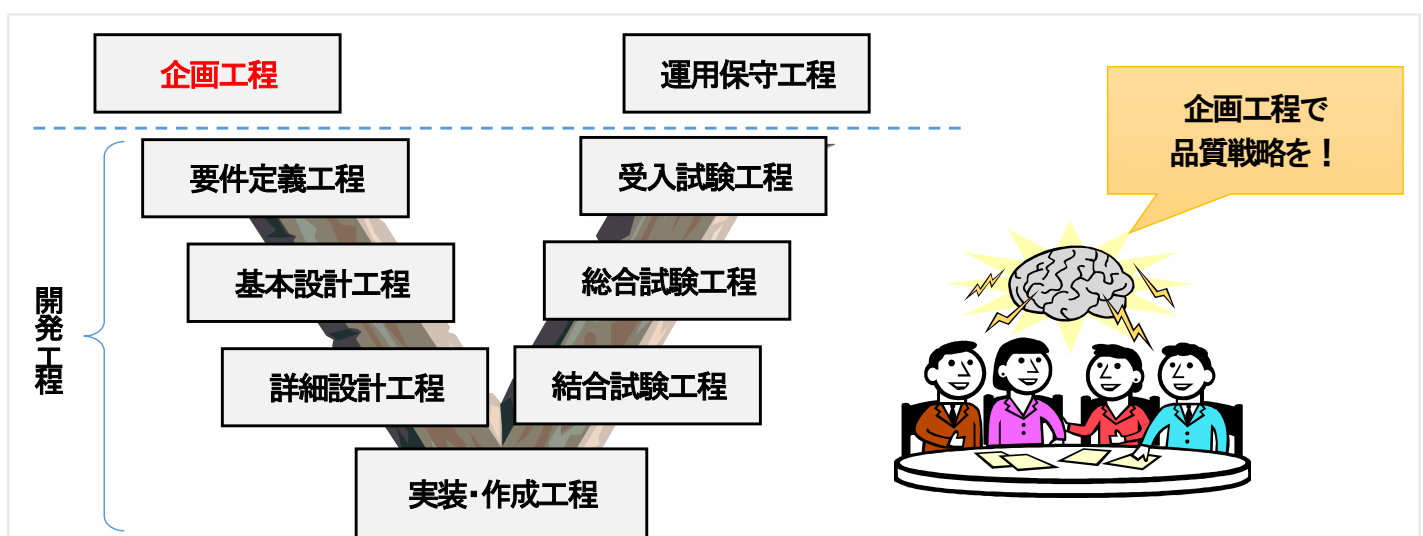


図-1 V字型開発モデルを適用する前の企画工程で要求を固める際に、品質戦略(*)を考えるイメージ

D-Case手法をPRJで活用しよう

D-Case手法を、PRJで活用する場合についてご紹介します。

D-Case手法の活用－１

D-Case手法は、PRJで実施している業務内容について、お客様と合意してゴールを設定し、その内容を具体的に実施していく時の論理的な展開をイメージで表現する際に利用できます。

この場合は、要求記述表や要求逸脱分析表は使いませんが、テスト設計の構造を、D-Case図を使って説明する例です。D-Case図の中で証拠となる部分に、具体的なテスト項目やテスト項目数を記載します。

PRJ月次報告会などで、D-Case図を使ってテスト設計の構造を説明し、テスト項目が適切なテスト技法やテスト観点から抽出されていることを、イメージとして示します。

そうすることで、テスト設計に対するお客様の理解が深まり、競合他社と差別化した報告が可能になります。また、テスト設計の論理展開パターンを蓄積できるので、末端の証拠となる具体的なテスト項目がテスト対象ごとに違っていても、PRJを越えて共有していくことができます。

D-Case手法の活用－２

従来のテスト設計方法と違う方法で、障害や事故といったハザードに特化したテスト設計を考え、安全性、堅牢性を確認する必要がある場合に、お客様とその設計内容をまとめていく際に利用できます。

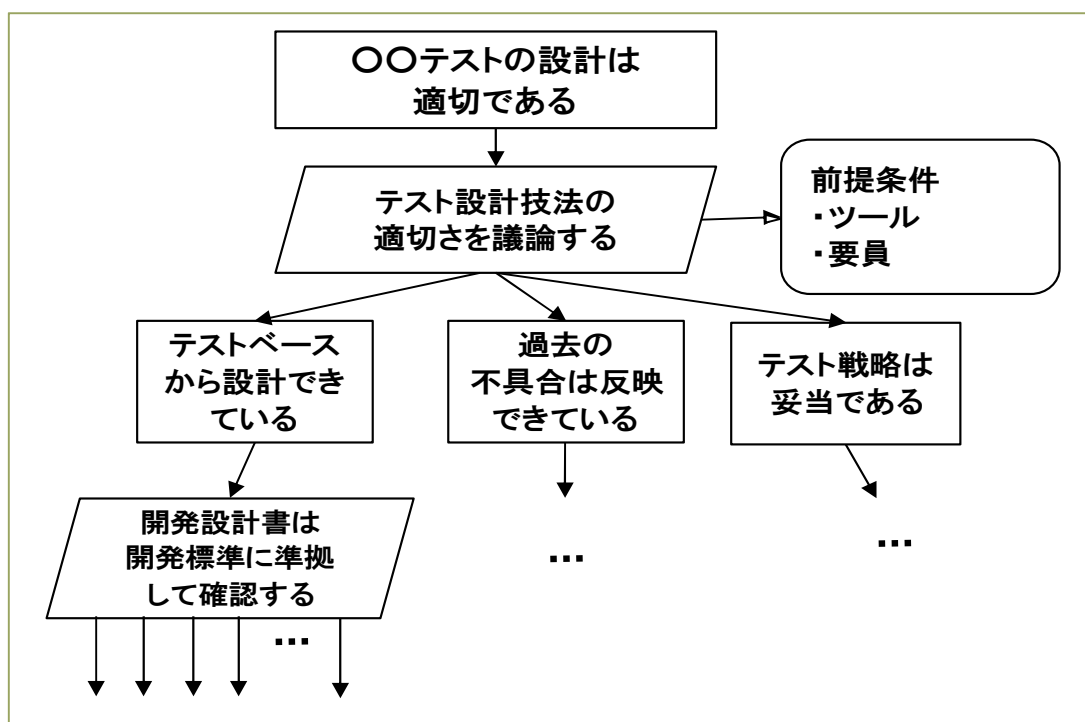
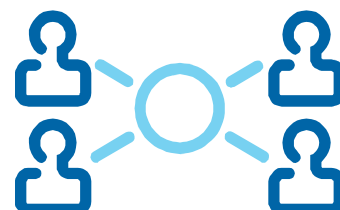


図-1 D-Case手法で、テスト設計についてお客様に説明して合意する際のイメージ

この場合は、本格的に、開発仕様書やシステム設計書、要件定義書などから、要求記述表、要求逸脱分析表、D-Case図を作成します。

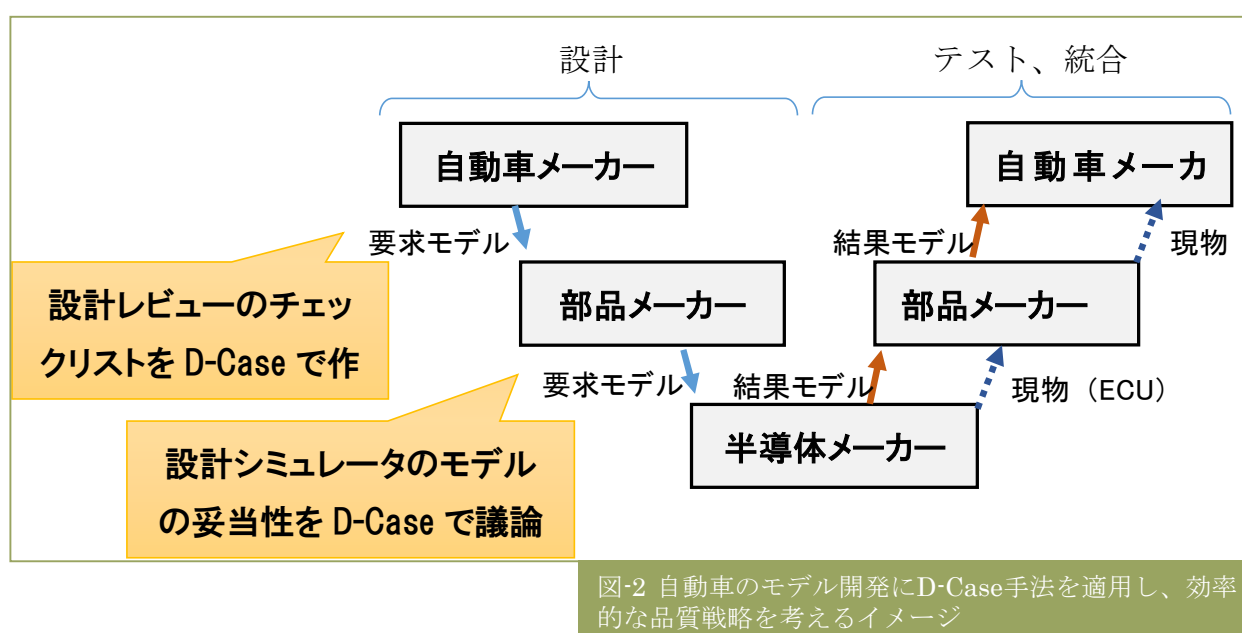
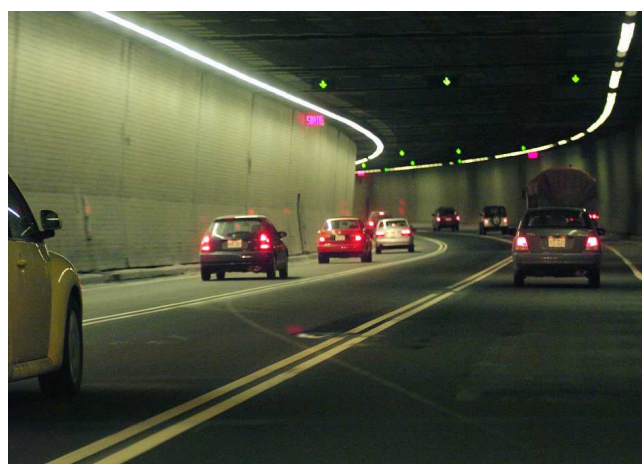
この使い方は、本来のテスト工程におけるD-Case手法ですが、実際に、こうした障害テストをきっちり実施するシステムとしては、人の命に関わるような医療系や、たった1回の障害でも、甚大で広範囲な影響を及ぼすような金融システム系、交通システム系、宇宙システム系、原子力システム系などが考えられます。これらのシステムを構成する装置や機器に対するコントローラ用ソフト(組込み系ソフト)に限らず、冗長構成のサーバーやストレージ、データセンターなどの品質戦略の立案から、具体的なW字型検証モデルへの展開における品質確保の議論に対して、D-Case手法を活用していくことができます。

開発プロジェクトを実施することが決まり、実際にプロジェクトを開始する際に、プロジェクト計画の立案と共に品質戦略を立てて、開発工程の進行に沿った品質の作りこみと、そのチェック方法(KPI:Key Performance Indicator [評価指標])を具体的に決定します。これにより、開発工程ごとの品質管理が具体的に実施できますので、プロジェクトの品質が可視化できます。

これまで、試作しながら要件定義していたような工程においても、要件の抜け漏れを早く検出できるほか、保留事項も具体的にし、何が明確になれば保留事項は対策済みとなるかが、わかるようになっていきます。

自動車メーカーでは、MATLABやSimLinkなどのツールを使って、設計シミュレーションを行い、実機による安全性テストの項目を最小限にし、開発期間の短縮、開発コスト削減を推進していますが、そうしたシミュレータによる安全性評価項目の実施だけでも自動車安全規格のISO/IEC26262の審査をパスすることが可能となっています。

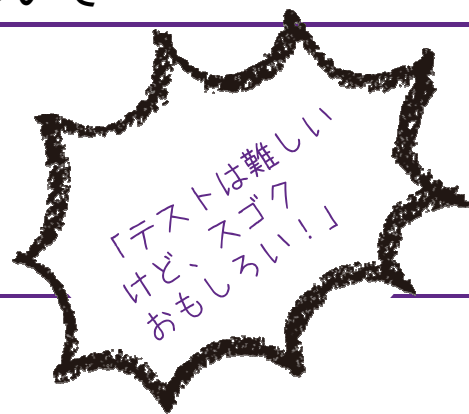
設計シミュレーションは、ハードウェアだけでなく、ソフトウェアについても可能になりつつありますが、そのシミュレーション上での網羅的な機能テスト/非機能テストは、外注先の部品メーカーに対しても同じですので、新車の設計工期が大幅に短縮され、開発コストも大きく削減できるようになります。この設計工程において、具体的な設計レビューのチェックリストを検討する際にも、D-Case手法を応用して、チェック項目の設定をすることができます。



現場スペシャリストに聞く、検証の将来について

金融系分野で活動中の
小宮洋行さんに直撃！

ソリューション部：斎藤功



SIer様の中で金融系分野に取り組んでいる、小宮洋行さんに語ってもらいました。

——“金融系分野”ということですが、具体的にはどのような内容ですか？

「金融系のパッケージ開発の中で、ソースコードの検証と、総合テストの企画と設計、テスト実施を行っています。」

——“このプロジェクトは上流から参画できた”と聞いていますが、どんな流れで入れたのでしょうか？

「最初は“テスト標準の導入”から始まったプロジェクトです。お客様の全社テスト標準の構築と導入を担当させていただき、その知見を活かして同じお客様で金融系のパッケージ開発のテスト標準作成を担当させていただきました。その後、上流での”仕様書検証”を提案して採用していただき、検証の対象をソースコードや単体・結合テスト仕様書に広げ、テスト範囲も曖昧さだけでなく上位要求との整合性チェックまでを担当することができました。最近は、総合テスト計画の作成と、非機能要求のテストも行っています。」



——上流工程から入って理想的な流れかと思いますが、何がポイントとなったのでしょうか？

「テストの専門家による第三者検証の必要性を理解いただいたことが大きいと思います。また、お客様の管理職の方々の理解者が増えていったことも大きな要因だと思います。」「また、これまでお客様では(動的な)テストで品質を担保することを行っていましたが、最近では開発やテストの標準を作り、レビューを強化して上流で品質を作り込む流れに変わってきています。」

——開発の方たちと接して仕事を行っているようですが、開発者と検証者との違いは何でしょうか？

「開発側では仕様に書かれていることを基に機能を作り込みますが、検証側では仕様書に書かれていないことも考えながらテスト設計し実施を行っています。この“書かれていないこと”の扱い方が違うということでしょう。更にSIer様の中は担当が細分化されていて、担当範囲が境界線として引かれています。往々にしてこの境界線の中にグレーゾーンがあり、そこに不具合の作り込みの危険性があるということです。」

——“書かれていないことを”考えてテスト設計・テスト実施するのは難しいと思いますが、それが出来るには何が必要なのでしょうか？

「“書かれていないこと”をテストに盛り込むにはその根拠が必要となります。例えば”国際標準規格の知識”や、最終製品の品質に対する考え方を持っている必要があります。その意味では”検証は開発より難しい”と言えるのではないかと最近つくづく感じています。」

例えばテスト計画の草案を作るにしても、色々な人と話をして情報を入手し、それぞれの立場や役割を意識しながら調整を行い、根拠を提示して納得してもらう必要があります。」

——色々なことを知っていたり、合意形成を作り上げる力も必要ということですね。

「そうですね。知識の領域を広く持つ必要性を痛感しています。ある意味細かなことを知らなくても話せないといけなく、提示できないといけなくということです。何故そのテストを行わなければならないかの根拠や、それを行う戦略や方針を提示する必要があります。また、単体テストから総合テストまでの全ての工程をまたいで検証全体で品質を考えることも必要となります。」



——この“合意形成”、“見える化”はD-Caseに繋がるかも知れないですね？

「その可能性はあるので、少し考えてみます。別途相談させてください。」

——大変な仕事ですね。何かそれ以外にも思っていることがありますか？

「最近感じていることは、検証というものが変わって来た、検証(テスト)のことだけを考える時代は終わったのではと。検証業務はイノベーションの領域ではないかと、『ホンダ イノベーションの真髄』を読んで思ったこと

ですが、ソフトウェア開発はオペレーションが重要な作業で、検証は小さなイノベーションを繰り返す行うことが重要な作業ではないかと感じています。

また、開発担当者と検証担当者は明らかに思考ロジックが違うということも痛感しています。(開発担当者に品質の話をして、皆さん、反応が鈍いです...)」

——最後に“これだけは伝えておきたいこと”があればお願いします。

「テストは難しいけどスゴクおもしろい！色々なことを知らないといけなく、最近ではテスト以外に要求から実装まで幅広い分野の書籍を読むようになりました。知識をフル活用して対応した仕事がお客様に評価されるのは、エンジニアとして非常にありがたいことです。皆さんにもこれを感じて欲しいです。」



——まさに“検証の未来”に繋がるヒントがたくさんあった感じです。是非、イノベーションにより『絶対価値(違い)』を皆で生み出していければと思います。インタビューにご協力いただき、ありがとうございました。



OSSリスク管理ソリューション に取り組む、銭本 崇さんに直撃！

今回は、OSS（Open Source Software）リスク管理ソリューションの新たなビジネス展開に取り組んでいる銭本さんに語ってもらいました。

——まず“OSSリスク管理ソリューション”は簡単に言うとどのようなものでしょうか？

「BlackDuck社（以下、BD社とします）のOSS検査ソフトウェアを利用し、OSSライセンスに遵守していない不適切な利用（コンプライアンスリスク）を洗い出すサービスです。言い換えると著作権違反のチェックサービスです。最近では上場企業を中心にキッチリOSSライセンスのチェックを実施する会社が増えてきており、例えばB2Bの開発の場合には、お客様（発注者）に求められるケースも増えています。」



インタビュー： 斎藤 功

——これまでのサービスから“新たなビジネス展開”ということですが、具体的にはどのような内容でしょうか？

「BD社のソフトウェア自体が、コンプライアンスチェックから、総合的なOSS管理ツールに変わっていることに合わせ、当社のサービスを拡充したサービスです。」

——何故このような動きになっているのでしょうか？

「そもそもOSSを利用するということは、開発のコストダウンやソフトウェアモジュールの再利用などの狙いがありますが、明確なOSS管理プロセスがない会社が多く、OSSの効率的な利活用の必要性が取上げられるようになった背景があります。」

——どのような利活用が求められるのでしょうか？

「OSSの利用だけではなく、OSSを組み込んだソフトウェアモジュールやコンポーネント、或いは社内のソフトウェアモジュールの再利用が求められています。これは、社内で似て非なる複数のソフトウェアの開発が発生したり、そのメンテナンスの負荷が増大したりと、開発への影響が大きくなってきたためです。その為、再利用をシステム化できるソフトウェアプラットフォームも求められています。」

——新サービスですがどのようなものなのでしょうか？

「内容的には先の背景を踏まえて、サービスレベルを段階的に上げていくようなサービスとしています。大きく分けて3つのサービスレベルです。」

- ①コードの再利用へ対応するサービス
- ②脆弱性へ対応するサービス
- ③コンプライアンスリスクへ対応するサービス

考え方としては、BD社のソフトウェアと人の作業の組合せでサービスを提供する形です。BD社のソフトウェアを導入すればフルオートメーションで処理できるわけではないため、人による作業をアウトソーシングする形です。」

——それぞれの概要を説明していただけますか。

「“コードの再利用への対応”は、製品がどのようなソフトウェアコンポーネントで構成されているのかを管理するソフトウェアの運用サービスになっています。」

「“脆弱性への対応”は、OSSコードが開示されているが故に、脆弱性も見つけやすく攻撃手法も作られやすい背景があります。現状では各社リリースした後に人力で対処する形ですが、如何に早く気付き対処できるかがポイントとなります。BD社のソフトウェアでは、デイリーでOSSの脆弱性のチェックを行い、製品担当者への連絡を自動化できます。」「また、当社のネットワーク検証サービスビジネスユニット（脆弱性検証チーム）と連携して脆弱性への対応を行うことも進めています。」



「“コンプライアンスリスクへの対応”は、これまで提供してきたサービスですが、お客様のソースコードに含まれるOSSとライセンスの調査と、ライセンス違反の可能性指摘を行います。」



銭本 崇さん

以前はOSS検査ツールのアウトプットを技術者が2次解析し、レポートしていましたが、現在はツールのアウトプットに対して技術者が判断していた項目を自動化した独自アプリケーションをアドオンするとともに、独自レポートの自動出力等、当社開発のアドオンアプリケーションで当社ならではの価値を出しています。」

——実際にお客様の反応はどうか？

「OSSを使った開発の効率化を狙っているお客様からは、“OSS利用時の課題にマッチしているね”という反応をいただいています。また、OSSライセンスのチェックサービスを導入されたお客様には、過去に開発したソフトウェアに対してOSSライセンスのチェックを掛けるケースも出てきます。」

——当社のPRJからもお客様へ訴求できそうですね。

「お客様は製品やサービスの品質を高めるために色々な違う視点でのアプローチを考えているはずですので、それに対応できる一つの武器になると思います。お客様の良いパートナーになる為には、色々な相談相手になる必要がありますので。」

——当社の新しいビジネスの観点からは何が言えますか？

「BD社のソフトウェアのライセンス販売(年間契約)とアウトソーシングの持ち帰り業務としているため、ナレッジを本社で集約して運用することが出来るサービスです。また、独自アプリケーションで技術者の作業を最小化するようにサービスを仕上げていますので、人に依存しない／アウトプットに明確な根拠がある検証サービスの形になっているといえると思います。」

——検証の将来についてはどうでしょうか？

「ソフトウェア部品の組合せによる開発が進めば、検証自体も部分的にはある程度組合せになるのではと考えています。I/F自体をチェックすればこのコンポーネントは問題ないみたいな。結果、テストも効率化できます的なサービスになっていく可能性があるのではと思っています。今後、コード解析からの検証範囲の拡大に繋がるのではと考えています。」

——最後に新しいサービスやビジネスを作っている訳ですが、そこに必要なモノは何でしょう？

「お客様を良い意味で驚かせたいという志向だと考えています。以前社内の先輩に聞いた言葉が印象的でした。“停滞は衰退だと”、同じことをやっているとはそれは衰退で、常に先を走っている必要があるということです。日々の業務の中でバランスが難しいため、なるべく自分自身を追い込むようにコントロールしています。例えば、定期的にセミナーに登壇することを掲げ、講演内容も以前と同じ話しにならないよう、お客様に驚きを与えられるよう、セミナーをマイルストーンとして新しいサービスをつくることを行っています。そのためにも世の中の動向を掴みながら、常にアンテナを張り停滞しないように前に進むことをモットーとしています。」

——正に皆が考えるべき話しだと思います。お客様の品質を上げるパートナーとして日々活動していきたいですね。

本日はインタビューへのご協力ありがとうございました。



ユーザビリティ PRJ展開事例

ソリューション部：原義雄

前回はPRJ・事業部向けに行ったD-Case研修をご紹介させていただきましたが、今回は東日本第三事業部のあるPRJ(Webアプリ系のPRJ)で行ったユーザビリティ研修を御紹介させていただきます。

【研修概要】

○実施日

1. 事前打ち合わせ	2014年5月29日(木)15:00～16:00	1時間
2. 研修1日目	2014年6月20日(金)14:00～16:00	2時間
3. 課題1(宿題)	2014年6月23日(月)～6月24日(火)	約2時間
4. 研修2日目	2014年6月25日(水)15:00～17:00	2時間
5. 課題2(宿題)	2014年6月26日(木)～7月4日(金)	約7時間

○参加者：PRJメンバー 2名（3年目社員1名と1年目社員1名）

○研修目的:ユーザビリティ評価の一つであるインスペクション評価の理解と体感

○研修内容:

- ・1日目 ① ユーザビリティの基礎知識の説明
② ユーザビリティチェック(まずはやってみる)
PRJ先の受け入れテストのケースを使用して、各自が思うユーザビリティ観点で対象画面を操作し、使いにくい箇所などを指摘し、結果を発表する。
※診断対象は「業務アプリ」(社内管理システムのPKG)です。
- ・2日目 ① 課題1の診断結果を整理分析(ワークショップ形式)
KJ法を使用し整理・分類を行った後、影響範囲を検討
② 総評(事例やレポート方法などの説明)



(図2)研修での課題抽出の様子



(図3)ユーザビリティ診断
報告書サンプル(イメージ)

2日間の流れ

1日目		2日目
座学	宿題	ディスカッション
やってみよう		課題の確認
ユーザビリティって何？		問題をピックアップ
ユーザビリティを調べる		問題を分類しよう
インスペクション評価		改善方法を考えよう
実施の注意		最後の振り返り
もう一度、やってみよう		
振り返り		
次回までに		

(図1)ユーザビリティ研修の流れ



(図4) 研修のテキストの一部

【参加者の声】

～この研修で学んだこと～

- 「ユーザビリティのテスト」と言っても漠然としていたが、
どういった観点で対象をテストすればよいかを体系立って学習・体験ができた。
- 自分が持っていない他者の視点に気づくことができた。
- 誰もが何かしらの「ユーザー」であり、比較的活かしやすいので新卒
でも即戦力となり得る。
- 課題をこなすことでメンバー間に一体感が生まれた。
- 探索的テストへの適用を行ってみる。
⇒ 探索的テストに、ユーザビリティの観点を追加して実施する。
⇒ 報告時にユーザビリティの観点をを入れてお客様へアピール。

発見した問題点	改善案
1. 画面のレイアウトが分かりにくい。特にメニューの配置が不明瞭である。	メニューの配置をわかりやすく整理し、色やアイコンで視覚的に区別できるようにする。
2. 入力欄のラベルが小さく、読み取りにくい。	ラベルのサイズを大きくし、色を黒に統一して可読性を高める。
3. 確認画面の表示が、操作の完了を明確に示していない。	確認画面に「操作が完了しました」というメッセージを追加し、緑色のチェックマークで視覚的に示す。
4. 検索機能の使い方が複雑で、必要な条件が多すぎる。	検索条件を簡素化し、必須条件と任意条件を分けて表示する。
5. 印刷機能の出力結果が、必要な情報が省略されている。	印刷オプションを追加し、必要な情報をすべて出力できるようにする。
6. 画面のレスポンスが遅く、待ち時間がある。	データベースの最適化を行い、処理速度を向上させる。
7. 画面のデザインが古く、印象が悪い。	最新のデザイントレンドを取り入れ、UI/UXを刷新する。
8. 画面の操作性が悪く、誤操作しやすい。	操作性を向上させるため、ボタンやメニューの配置を見直し、フィードバック音を付ける。
9. 画面の表示が、特定のブラウザやデバイスで正常に動作しない。	クロスブラウザ・クロスデバイス対応を確認し、修正を行う。
10. 画面の表示が、特定の言語で正常に動作しない。	多言語対応を確認し、修正を行う。

(図5) 研修で抽出された問題一覧 (イメージ)

今回の研修はユーザビリティ評価手法の一つであるインスペクション評価を行いました。ユーザビリティ評価を専門的なイメージで捉えているかも知れませんが、インスペクション評価は、**バイアスを排除し、想定されるユーザー像をイメージできれば、専門家でなくとも評価できる手法**です。(一人だと意識の偏りや思い込みで問題に気付かないこともあるので、一般的に2～3人で問題抽出と整理・分析を行います。)

D-Case PRJ展開事例

7月上旬に行った、中部事業部向けのD-Case研修を御紹介させていただきます。

【研修概要】

- 実施日 : 2014年7月5日(土曜日) 10:00～15:00
- 実施場所 : ベリサーブ 中部事業部 SC伏見ビル3F(セミナールーム1)
- 参加者 : 中部事業部 計10名 【内訳】管理職1名、PRJ長7名、PRJメンバー1名、営業1名
- 実施内容 : D-Caseの概要説明の後、各グループでテーマを決め、ワークショップ形式で演習を実施。グループ内討議で、自ら考えることで理解度が上がるためワークショップ形式を採用する。



【セミナーの内容】(図1)
セミナーの内容と時間割

セミナー内容(内容と時間割り)

1.【前説】	D-Caseの背景(なぜD-Case)	30分
D-Caseの活用ビジネスについて説明いたします。		
2.【説明】	D-Caseの概要について説明	45分
D-Caseセミナー資料を用い、ポイントについて説明いたします。		
3.【WS 1回目】	テーマを元にグループ討議	45分
選択したテーマ(例:分室を拡大・継続させる)を題材にグループ内で討議いたします。グループメンバーは状況により決めます。グループ毎に講師が指摘・アドバイスをいたします。		
～ 休憩(60分) ～		
4.【振り返り】	グループ討議の発表と結果チェック	30分
作成結果の発表と振り返りを行い、D-Caseの考え方を事例で修得する。		
5.【WS 2回目】	テーマを元にグループ討議	30分
選択したテーマを題材にグループでD-Caseを使って構造的に可視化する。講師による個別指摘がグループ毎に入ります。		
6.【振り返り】	グループ討議の発表と結果チェック	30分
作成結果を発表と振り返りを行い、D-Caseの考え方を事例で修得する。		
7.【Q & A】	質疑応答	30分

- 実施目的 : PRJ長/PRJメンバーがPRJ拡大につなげられるヒントを得る。



【研修のゴール】(図2)
研修のゴール

研修のゴール

D-Caseの概念が理解でき、分室拡大・継続にどう使えるか理解できる。

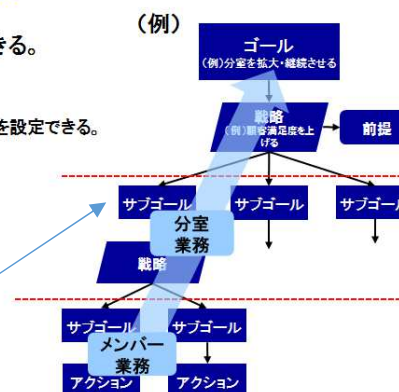
D-Caseを使って構造的に可視化できる。

→ 要求に対し、分析、分解、整理ができる。

→ 分室業務へ利用できる。

ゴールを設定し、前提・戦略・戦術・手段・を設定できる。

GSN (Goal Structuring Notation) サンプル



※GSN (Goal Structuring Notation) : システムが達成すべき目的や性質について、その達成を導く方法・思考を可視化する際に用いる記法

この研修を通して、PRJの皆さんが**お客様との合意形成などにD-Caseを使えるようになる**ことを目標に設定しました。そのため、単に説明ではなく、理解度が上がるように研修の中にグループ別のワークショップ形式を取り入れ、チーム毎にテーマ(TOP GOAL)を決め、実際にGSNを作成していただきました。

ワークショップでは、チーム毎に様々な「TOP GOAL」が設定されました。「D-Caseが使える」「結果が高品質である」「お客様の課題を解決する提案ができる」「PRJが拡大する」・・・などなど。「PRJが拡大する」は事業計画の一つでもあるため、「戦略」と「前提」を決め、それを達成するためのいくつかの「Sub GOAL」を決め、更にその「Sub GOAL」を達成するための「戦略」、「手段」と作成していくと、PRJやメンバーが実施

する業務に該当することになります。(図2のGSNサンプル)

現場に即した目的や、事業計画を例に取ることで、メンバーやPRJの業務が担っている役割が明確になり、存在価値も明らかになることでモチベーションアップにも繋がります。逆に、今やっている業務がGSNの「手段」に出て来なかったり、前提に相違があったりすると、そもそも目的が違う業務かもしれませんので、上長に確認と合意を取る必要があるということも分かります。これは、お客様との業務(PRJ業務)に置き換えれば、お客様への説明材料(確認事項)になり、お客様内の事では、思考の流れを捉えることができるため、上流工程での要件のズレ、部門間での認識のズレなどを把握することができ、**合意形成の材料**としてご利用いただくことができます。

ワークショップでのグループ討議の時間は45分程度でした。当然その時間内で全ての議論が出尽すことはないと思いますし、そうそう網羅できるものでもありません。しかし、そのような網羅性は重要なことではありません。その時点での前提と議論内容(証跡)で合意をとっておけばよいことで、「不明点」や「未定事項」は、「不明」、「未定」であることを記載しておけばいいのです。修正が入れば、都度合意を取っていくことが重要となります。お客様との合意形成を計るための説明材料としては最適な方法です。

これまでD-Caseの紹介になってしまいましたが、最後に研修で出た主な意見をご紹介します。

全体的にアンケートや質疑の内容から、8割の方が満足傾向にありました。

【良い意見】

- ・実習があって良かった。
- ・正しく使用できれば顧客にアピールできるものとなる気がする。
- ・論理展開ロジックの表し方なので、業務への適用ができる。いくつか実践が必要？
- ・お客様への説明に使えると思います。まずはPRJ業部で試してみます。

【悪い意見】

- ・D-Caseの理解としては「ふわっ」とした感じである。
- ・合意形成する以外の要求逸脱分析などの内容も気になる。
- ・記載ルールが曖昧である。
- ・戦略を決めるのが難しい。



それを裏付ける結果として、早速参加された2名の方より実業務に利用された結果なども頂いております。その例を以下にご紹介させていただきます。

(1) テスト品質改善活動の計画レビューにD-Caseを利用する。

【利用概要】 お客様との定例会議で思考プロセスを可視化するため、D-Caseを用いて業務を説明する。

【結果】 お客様より「大変わかり易い」と言った高評を得る。

【ソリューション部からの支援】

- ・作成された説明資料を頂き、D-Caseの展開面からの指摘と修正案の提示
- ・類似サンプル(GSN)の提供



(2) 中間報告・終了報告にD-Caseを利用する。

【利用概要】 機能性評価が完了し、これから信頼性試験が始まるため、以下の3点について、お客様との合意形成にD-Caseを利用(予定)。

- ・現時点での品質を論理的に説明する
- ・信頼性試験観点を明確にする
- ・信頼性試験観点について顧客と合意をとる

【ソリューション部からの支援】

- ・D-Caseを用いた終了報告書(報告書)のサンプルを提供する。
- ・作成された報告資料を入手し、作成支援とD-Caseの展開支援(指摘と修正)を行う。



派生開発の世界と検証

派生開発とXDDP

ソリューション技術統括 佐々木 方規

派生開発とXDDP*1 (eXtreme Derivative Development Process) については、AFFORDD (派生開発推進協議会: <http://www.xddp.jp/index.shtml>) のHPを参照すると詳しく紹介されています。ここでは検証の視点で、派生開発とXDDPを解説したいと思います。最近では、新製品の開発といっても0から全てを設計するスクラッチ開発はほとんどありません。新製品の開発であっても、既成のソフトウェアを部分的に手を加えた変更開発です。それらに加え短納期での開発が要求され、このことが派生開発に注目があつまる背景となっています。ここに発生するのが変更部分にフォーカスした部分理解による開発ですが、なぜ部分理解で開発することが可能なのでしょう。

構造的に作成されたソフトコードは、呼び出し型の関数(型)になっています。部分的な理解を進めるには、変更対象となるソフトコードが含まれる関数とその引数を理解することから始まります。そして、対象となるのソフトコードを変更することによる影響範囲の考慮を広げていきます。影響範囲を検討することは、検証でも実施されています。例えば、不具合修正後の回帰テストでは、影響がありそうな部分を中心にデグレードを起こしていないことを

確認します。回帰テストを計画する場合に、影響範囲はどのように考えられているでしょうか。”全テストのやり直し”を実施しているプロジェクトは少ないと思います。参考ですが私の場合は、次のような影響分析を実施しています。

1. 変更モジュール (ソースコード) の対象部分
変更したプログラムのアルゴリズムや、コードカバレッジなど
2. 変更モジュールと直接I/Fをとる関数群
変更したプログラムを呼び出す関数との関係など
3. 呼び出し/復帰I/F周辺
変更したプログラムの呼び出し変数の型式など
4. 状態の変化および状態が関連する機能群
変更したプログラムに影響する状態の変化など
5. グローバル変数で関連する機能群 (要注意)
変更したプログラムが利用するグローバル変数など

上記は、序列になっていて、後半部分の分析では、広範囲に及びます。気づいた方も多いと思いますが、私の回帰テストの分析方法は、俗に「グレーボックステスト」に近い分析を行います。

開発環境の変化にどう対応するか？

派生開発での成果物の分析方法を解説する前に、開発環境の変化と検証についてお話します。開発環境などを説明するのには理由があります。

検証にとって、「**開発環境の変化にどのように対応すれば良いのか**」が、大きな課題です。

XDDPをはじめとする派生開発やリーンスタートアップ開発、イテレート開発などのアジャイル開発がとり沙汰されていますが、検証がテストに対応できていないとよく聞かれます。

実は、大きな落とし穴がそこに存在するのです。

- ・スクラッチからの開発ができないから派生開発を取り入れた。
- ・要求が決まらないからアジャイル開発を採用した。

何とも正論に聞こえますが、実はメソロジーやプロセスの適用を、開発への要望が正しく分析されていない状態で、トレンドの技術を採用する安易さが状況を複雑化させています。

テストの現場でも同じで、組合せ爆発を回避するためには直交表、漏れが無いようにテスト観点を適用するなど、総括された課題のみフォーカスしています。

これらについて何が問題であるかという点、開発であればアジャイル開発を適用したり、派生開発を実施することではなく、要求の変化や定義が決めきれない環境下でどのように進めれば適切な開発ができるか、に対応するプラクティスの集合体が開発手法やプロセスです。検証も同様です。

より適切なプラクティスを選択することが重要で、プロセスやメソロジーに定義されている全てのプラクティスを選択する必要はありません。

パッケージ製品に例えると、豊富な機能が提供されているから全てを利用しなければいけない、ということです。無理に利用しようとすると、必ず矛盾がおこります。

派生開発などで必要な影響分析をせずに、ソースコードのみ簡単に変更してしまう。影響分析する時間が無い、分析する方法がわからないなどの課題が棚上げ状態で、開発されてしまう。これらは、ソースコードを変更するときにリファクタリングを行わないとコードが劣化していく。劣化したコードは、分析が難しい。悪いスパイラル状態で、収集がつかなくなっていく。

このような状態で、後はテストに任せられても、品質の劣化を抑えることは所詮無理な話です。テストは銀の弾丸にはなりません。**テストを実施する場合に重要なことは、開発プロセスの分析です。**しかし、そこに気が付いている人は少ないと思います。

テストの要件分析には、プロジェクトの特性を分析するアクティビティが存在しますが、プロダクト(テスト対象)だけフォーカスして分析してしまうプロジェクトも多いと思います。

ます。

ソフトウェアの欠陥は、人がミスして混入していきます。人が介在するところに、ミスが生じます。つまり、開発プロセスで人が介在する部分に欠陥が入り込むのです。かくれんぼで、鬼が目隠ししないで、隠れるところをみている。遊び(ゲーム)としてはずるいですが、品質を確保する視点では正しい方法です。テスト担当者は、開発のあり方に踏み込まないことが正しいわけではありません。影響度分析もされていない派生開発で開発されたものを、テスト担当者が臭いものには蓋をしてテストをする。皆さんは、どのように考えますか。

*1 XDDPは、(株)システムクリエイツの清水吉男氏が提案された開発アプローチ



テスト体系

~What's “Validation” ?~

技術部 : 江澤宏和

私が、システム検証理論研究会で発表しているテーマは、「テスト体系」です。「テスト体系」は、**対象となる製品やシステムにおける全てのテストを、一枚の地図に示すことを目指しています。**なぜ、「テスト体系」が必要なのかというと、例えば、開発部門と品質保証部門の間で認識に齟齬があり、本来実施すべきテストが実施されていない、ということがあります。極端にわかりやすい例でいうと、機能単位で組織が分かれていたとします。A機能チームは、A機能のテスト、B機能チームは、B機能のテストを担当します。じゃあ、A機能とB機能の間の割り込みや遷移のテストは、誰が担当するのか？というと、実は明確に決まっていない、といったことがあります。こういった隙間での漏れを防ぐために、ステークホルダが集まり、テスト体系をもって担当するテスト範囲を明確にし、全体の認識を共通にすることが、目的の一つです。5月度の研究会で、テスト体系の話をしたのですが、これまでと違い、「Validation」という概念をテスト体系に示す、というものでした。

私が思っていた「Validation」は、「正しい製品をつくっているか」ということだと思っていたので、要求仕様書どおりに作られていることを確認することだと思っていました。しかし、「**正しい製品をつくっている**」というのは、「**ドキュメントに書かれていないこと**」を指摘することだそうです。そして、7月度の研究会では、中部事業部のSさんが、「書かれていない要件を導き出す方法論」というテーマで、折りたたみ自転車部のSさんが、「障害対応テスト体系」というテーマで発表しました。

Yさんのテーマは、そのものズバリですし、Sさんのテーマも、IPAの非機能要求グレードをベースにした検討で、これも「Validation」の一つであると思います。

「Validation」が何なのか、実は私もまだわかっていません。ただ、様々な手法や技法を用いて類推することで、「**書かれていないこと**」を指摘する方法は、ありそうな気がしています。

例えば、不具合モード分析の「例外」は、簡単に説明すると「書かれていることの中で、特殊なもの」ということになります。この方法を応用し、仕様変更や機能追加があったときに、「なぜ、それが追加もしくは変更されたのか」を分析し、体系化することで、「仕様に書かれないこと」のモードができ、それを持って類推することができるのかも知れません。

Validation・・・深すぎてよくわかりませんが、今後、当社にとって絶対必要になる技術であると思っています。



一般社団法人 I T 検証産業協会 (IVIA) 主催

『 I T 検証フォーラム 2 0 1 4 』

参加レポート

ソリューション部 : 西村 憲一郎



JP TOWER Hall & Conference

去る2014年7月4日(金)13:00～ 東京駅丸の内側の目の前に建つJPタワー(旧東京中央郵便局)で一般社団法人I T検証産業協会(IVIA)主催の「IT検証フォーラム2014」が行われました。

『明日の日本を創る第三者検証』～あなたの知らない検証の世界～と題して、これから求められる新たな「IT検証の世界」へと、皆様を繋げていきたいという思いで催されました。本フォーラムは今回で2回目となり前回より開催規模を拡大しました。今回の集客目標は600名と置きました。当日はあいにくの雨でわずかに届きませんでしたが、574名のお客様がお越し頂きました。当日のスケジュールはオープニング、来賓ご挨拶、基調講演、特別講演、分科会(2トラック×3セッション)で午後の半日でとり行われました。(※表1)また、出展ブースについても過去最高の17ブースが出展し、イベントの盛上りに一役買いました。

冒頭で主催者を代表してIVIA会長の浅井清孝氏が挨拶に登壇しました。「当協会は2005年発足して今年は第一次中長期(5ヵ年)計画の最終年度にあたります。標準化による検証技術の確立、検証技術者の育成と地位向上、検証業界団体活動の啓蒙の3つの柱で検証業界の基盤を作りました。現在、第二次中長期(5ヵ年)計画を策定中です。その最終年度は2020年、東京でオリンピックが開催される年になります。それに向けて交通インフラや生活空間における安心で快適な日本をアピールする絶好の機会であると思います。IT化、システム化という話が増えていき、そこに検証という役割が非常に重要になってくると思います。

※表1 : IT検証フォーラム2014 スケジュール

■オープニング: IVIA会長 浅井清孝			出 展 展 示
■来賓ご挨拶: 経済産業省 商務情報政策局 情報処理振興課 地域情報化人材育成推進室長 小池雅行 氏			
■基調講演: 脳科学者 茂木健一郎 氏			
■特別講演: 「なぜITシステムのトラブルは減らないのか」 独立行政法人 情報処理推進機構(IPA) 技術本部 ソフトウェア高信頼化センター(SEC) 所長 松本隆明 氏			
「ICT 第3世代プラットフォーム時代を 迎えるの検証技術」 元 日立製作所 篠崎 雅継 氏	「あなたの会社は企業秘密が守れるか？」 株式会社アールジェイ 代表取締役 石田 淳一 氏	「安心快適なホームネットワークの 発展への貢献」 IVIA 技術部会 大瀧 裕史	
「『全自動車』のトレンドを、 ‘検証’ から検証する」 日鉄住金総研 研究主幹 倉沢 鉄也 氏	「情報セキュリティ分野における IT製品等の認証制度の状況」 東京工科大学 コンピュータサイエンス学部 教授 手塚 悟 氏	「スマートハウスの品質を守れ！」 IVIA 技術部会 水野 健	

これに向けて投資が拡大していくのは容易に想像がつきます。今後も協会だからこそできる検証の事業化を進めていきたい。

そこで皆様に3つのお願いがあります。来年の3月に向けて中長期ビジョンを作っていく中、会員企業の方には事業計画策定PRJに参加してほしいです。会員意外の方は本日のフォーラム参加を契機に是非協会へ入会して頂きたいと思います。また、お客様の方々は今後ますます素晴らしい製品を開発することに注力して頂いて、検証の仕事はわれわれに任せて頂きたい。」という力強いお話がありました。

次に来賓挨拶として、経済産業省 商務情報政策局 情報処理振興課 地域情報化人材育成推進室長 小池雅行氏が登壇しました。

「日頃、我が国の情報処理の発展に多大なる貢献をして頂き感謝しております。安倍政権の新成長戦略においては、今後は世界最高水準のIT社会実現に向けた改革の本格的実行段階であります。新たなイノベーションの基盤となる無料公衆無線LAN環境の整備、サイバーセキュリティ対策の強化、ビッグデータの活用環境整備など力強い成長へつなげる為に手を打っています。今後さまざまな製品がネットワークに繋がる中、IT社会の産業分野を検証という手法によりシステムの下支えして頂き感謝しております。トラブルがない社会を実現するには、皆様の力が必要であると考えます。産業を日本経済のエンジンという表現をしますが、皆様はこのエンジンをチェックするまさにレース場のピットのような役割を担っていると表現できます。大変かつ地道な仕事であるとは思いますが、大変重要な仕事でありますので、産業の下支えをしていただきたく思います。経済産業省といたしましても、今後も皆様と協力し、

エンジンといえる産業界とピットといえる皆様とともに進行していく所存でございます。」

事前に予定していた方が急に異動になりピンチヒッターでの登壇でしたが、その役を果たしました。

次に基調講演として皆さんテレビなどでご存知の脳科学者 茂木健一郎氏が登壇しました。

「成長にITは欠かせないですが、これから皆様の想像をはるかに越えたITが出てくるでしょう。アメリカでは自動運転の車がすでに公道を走っています。実際走っている時は誰も気が付かないですが、もう実際にやっています。Googleが特許申請してトヨタのプリウスを使ってやっています。さて、日本で同じことができるでしょうか？更にハンドルが無い車も出ています。ここで問題なのは、事故が起こったら誰の責任か？（ドライバー、メーカー、通信、IT）ということが問題になります。そこにチャンスがあります。セグウェイが日本の公道を走っているのを見たことがありません。これでは、日本は自動運転の競争に負けるでしょう。日本は自動運転の国際標準化を目指しています。自動パーキングはよい評価のようですが、しかしこの団体にGoogleは入っていないのです。大丈夫ですか？勝てますか？また、自動運転ができるとなにか起こるか？ここで想像力を働かせなければいけません。タクシーがいらない、地方の飲食店が儲かる。運転代行がなくなる・・・。



車内での案内がいろいろ充実してきます。また、自動車の地図への広告も増えるでしょう。そこで、地図といえばGoogle MAPです。またここで心配になります。日本は本当に勝てるのかと・・・。

政府はもっと真剣に経済成長戦略考えて欲しいです。OSを執られたら終わりです。日本は慎重な国民性です。鉄道の指差し確認のように。でも新幹線はすばらしいですが、ITにおいては非常に心配です。自動運転の実験が公道でできるようにしてほしいです。実際に走らないと道路状況などでの自動運転の検証は非常に難しいですね。



話は変わりますが、大学のオンライン教室“edx”をご存知でしょうか？ハーバード大学やMITでは普通に行われています。実際とまったく同じ授業が受けられ、しかも無料ですし、定員もありません。世界中、オンラインで授業が受けられます。また、ただビデオを見るだけではありません。コミュニケーションもとれます。議論、質問ができる、単位を正式にもらう場合に料金を払う、という仕組みです。



それではテストや論文の採点をどうするかです。定員が無いのでものすごい数の生徒が世界中にいます。そこで、自動採点技術を開発しています。どう検証するか？ビジネスチャンスですね。

また、これらのシステムを外販しています。1コース:1000万円です。日本の大学にもオファーが来ています。独自でシステムを構築すると1000万円ではとうてい無理です。”edx“はすでに全世界に知られていますのでこの仕組みに乗るしかなくなってきます。これが戦略なのです。日本はここが非常に弱いのです。

日本はただコンテンツを提供するだけになっています。アマゾンも同じような感じですね。日本の出版社はアマゾンの業者に成り下がっています。パソコンやスマートフォンと同じ構図が見えてきます。



株式会社ベリサーブ ブース

また、Googleの“Project Loon”をご存知でしょうか？気球を高度に飛ばして地球全体に適度に分布させ3G程度の通信の基地局にします。これで全世界どこにいてもアクセスできるような環境を提供しようとしています。

外洋のインターネット接続は当然のこと山、砂漠、どこでも繋がります。そうするとまたいろいろな変化が現れます。Googleは全世界の通信を握ることができるのです。ITの未来はすごいですが、日本は主導権が握れないのではと思います。自動運転はビックビジネスですが、日本には戦略がありません。ここで脳の話をするとな「望ましい困難」というのがあります。脳にいい栄養になるそうです。ちょっと話が反れますが、失読症というのがあります。先天的なもので文字を読んで理解することが苦手なのです。ハリウッド俳優のトムクルーズやバージングループCEOのリチャードブランソンが有名です。トムクルーズは台本が読めないで、撮影中は他の人に読んでもらって覚えさせます。リチャードブランソンは財務諸表が読めません。

では何が優れているかというと、人を見極める力、交渉術、コミュニケーション能力が非常に優れています。ですから仕事を常にアウトソーシングするという考え方に立っています。優等生は何でも自分でできるので実はあまり経営者に向いていないといえます。経営者は人にうまく仕事をやらせるのが仕事ですから。



さて、日本はこの困難にどのような戦略で生き延びるのか、はたまた世界をコーディネートできるような人材が登場するのか？

最後にITを検証するというのは日本人に非常に向いている仕事だと思います。これだけ検証するネタがあるのですから、日本のIT検証が世界を制するかもしれませんね。皆様のご活躍をお祈りしております。」

かなり話しがあちこちへ飛んでいましたが、非常に面白く、ためになる情報を提供して頂き大変勉強になりました。そしてわれわれはもっと頭を使わなくてはいけなかった次第です。

その後もIPA/SECの所長である松本隆明氏をはじめいろいろな分野の方々をご講演されましたが、ここでは省略致します。



今回誌面の都合で全ての講演を掲載することはできませんでしたが、少しは臨場感が伝わったかと思います。ITの検証というものは誰がやるにしても、絶対に必要なことだということを改めて認識した次第です。

全ての講演が終了したあともロビーでは出展ブースへの人だかりがいつまでも途絶えることが無く、盛んに説明と議論が行われていました。その喧騒を横目で見ながら会場を後にしました。

クラウド・コンピューティング

技術部： 二井誠

Makoto Nii

■ クラウドの障害から

2014年の国内における大規模なクラウド障害は、今のところ発生していませんが、海外では、未だに頻発しています。日本のクラウド品質は、海外と比べると非常に良いと言えます。それは、電力事情や、国民の教育レベルの高さなども影響していると思います。つい、2～3年前までは、日本のクラウドも毎年、それなりの大きな障害事故を引き起こしていましたが、再発防止策が効いているためか、非常に少なくなりました。過去の障害を分析して見ると、運用に関連する問題、中でも、途中でバージョンアップを実施した際に障害を発生させてしまうケースがよくありました。国内で大規模な事例として、F社のデータ消失事件が有名ですが、ハッカーや

海外からのウィルス攻撃や機密情報・個人情報漏洩事件などは、年々、エスカレートしていると言われています。

■ クラウドを検証するとは

クラウドには、基本的にSaaS、PaaS、IaaSという3つのサービス形態があると前号で紹介しましたが、それぞれのサービスに対して、第三者検証サービスを提供していけるように取り組む必要があります。ちょっと話はそれますが、ASPIC(アスピック)という団体があるのをご存知でしょうか。この団体は、クラウド業者が提供するサービスに対して、ある一定の審査を行い、合格証を発行することで、利用者が安心して利用できるようにし、クラウドを普及させることを目的として設立されたNPO法人です。

SCSKの「USIZE」も審査を受けて合格しています。合格の有効期間は、2年間です。

クラウドは、通販サイトやSNS、ツイッターなどのコミュニケーションサイトだけでなく、医療介護福祉分野、交通システム、自治体システムといった公共系や、企業の基幹業務系への対応も進んできていますが、それと併にASPICの認定審査も、国内では標準となりつつあります。図-1にASPICの認定制度の経緯について、概略を示します。

この認定審査を受けて合格したからと言って、そのサービスが予告なくダウンしたり、情報漏洩事件が絶対に起こらないということを保証するものではありませんが、怪しいサイトだったり、

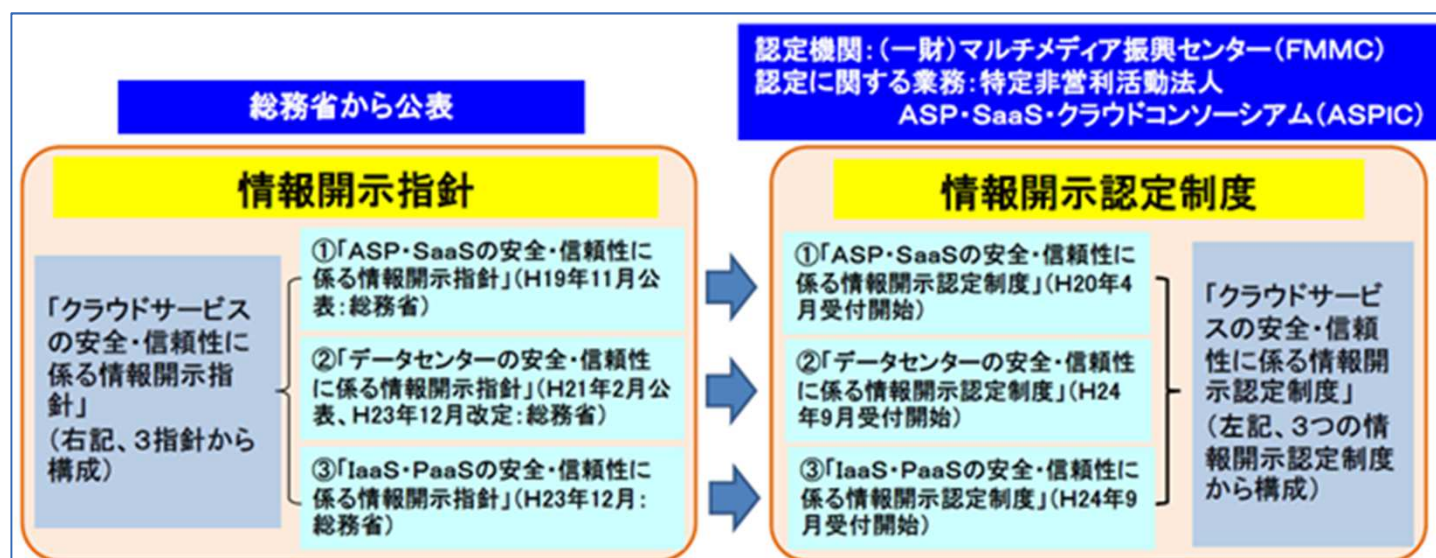


図-1 ASPICの情報開示認定制度の概略経緯

一発屋のサイトのようなオーナーが一儲けしたら突然サイトをクローズしてしまうというようなヤクザな企業かどうかを審査するようです。審査の概略項目を図-2に参考として示します。この審査も1つの検証の形態と考えられます。

しかし、医療福祉や自治体、交通などのような社会的影響、生命に直接影響のあるクラウドについては、具体的に障害が発生しにくいような十分なテストをやっているか、開発プロセスごとのレビューをやっているか、などのソフトウェア検証だけでなく、データセンターとしての耐震、耐火、防犯、防災、自立運転、運用監視ルール、顧客とのサービスレベル保証など、システム監査人が金融機関のシステムに対して実施する検査以上のチェックを必要とします。

扱う情報の安全性確保に対する要求レベルに応じた検証が必要です。この検証が弱くて、クラウドのデータセンターを乗っ取られるようなことが起こると、最悪の場合、国家の存続が危ぶまれるということさえ起こります。ちなみに、自治体が保有する住民票の個人情報、国外のクラウドに保存することは、法律違反になります。わかりやすく言うと、アマゾンやグーグル、マイクロソフトなどの海外のセンターと連携しているクラウドに住民票データをバックアップと言えども保存することは許可されていません。万一、そういう情報が、海外のテログループや諜報部門等に奪われて、事件が起きたとしても不思議はありません。

話を元に戻しまして、クラウド検証では、

Webサービスのソフトウェアを検証する従来型の検証分野以外に、運用保守について、ITIL基準に準拠した運用ができていないかを検証する分野とデータセンターの設備と建物について、各々の基準に準拠しているかを検証する分野があります。ベリサーブが得意としているソフトウェア検証分野は、SaaSクラウドにおいて、いろいろ実績が出てきております。

機能系だけでなく、負荷検証、セキュリティ検証なども実施しています。仕様書検証も行っています。ベリサーブが検証したものに責任を持ち、ASPICのような認定証(図-3)を発行できるといいなと個人的には思っています。

今回は、アマゾンのクラウドを使った1つの検証事例について、具体的にご紹介していきます。

■ASP・SaaS(中項目数:93、小項目数:133)		■IaaS・PaaS(中項目数:110、小項目数:131)		■データセンター(中項目数:103、小項目数:137)	
分類	情報開示項目	分類	情報開示項目	分類	情報開示項目
事業者	事業所・事業	事業者名、設立年、主要事業概要など	事業所・事業	事業者名、設立年、主要事業概要など	事業所・事業
	人 材	代表者、役員、従業員数	人 材	代表者、役員、従業員数	人 材
	財務状況	財務データ、財務信頼性	財務状況	財務データ、財務信頼性	財務状況
	資本関係・取引関係	株主構成、大口取引先、主要取引先金融機関	資本関係・取引関係	株主構成、大口取引先、主要取引先金融機関	資本関係・取引関係
サービス	コンプライアンス	組織体制、文書類	コンプライアンス	組織体制、文書類	コンプライアンス
	サービス基本特性	サービス内容、サービスの変更・終了、料金体系、品質、利用量	サービス基本特性	サービス概要、サービス構成、サービス品質、サービスの変更・終了、サービス料金・解約、サービス利用量、データ管理	ハウジング(建物・設備)
	アプリケーション、プラットフォーム、サーバ・ストレージ等	主要ソフトウェア、連携・拡張性、セキュリティ	システム運用(システム基盤運用、セキュリティ等)	システム基盤運用、セキュリティ(基盤・ストレージ等、ネットワーク)	ハウジング(ネットワーク)
	ネットワーク	回線、セキュリティ	ハウジング(サーバ設置場所)	建物、電源設備、消火設備、避難対策設備、空調設備、セキュリティ	ハウジング(サービスの内容)
	ハウジング(サーバ設置場所)	施設建築物、非常用電源、消火設備・報知システム、避難対策、空調設備、セキュリティ	サービスサポート	サービス窓口、サービス保証・継続、サービス通知・報告	ハウジング(サービスのサポート)
	サービスサポート	窓口、サービス保証・継続、通知・報告			IaaS・PaaS(サービスの内容)
					※当該DCと一体として提供しているIaaS・PaaSサービスで、申請の対象とする場合は記述。
					サービス構成、データ管理

図-2 ASPICの審査項目概要



図-3 ASPICの認定証イメージ

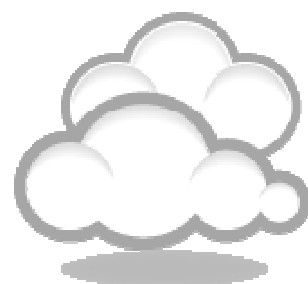


図-1, 2, 3 - 特定非営利活動法人ASP・SaaS・クラウド コンソーシアム(ASPIC)資料
<http://www.aspicjapan.org/event/forum/2013/file/aspic02.pdf> より抜粋

モヤモヤする テスト用語



てすとデザイン倶楽部

※Zep、Aloha、MMの会話

(人物紹介)

Zep→色々と不満を持っていて、どうにかしたいと思っている。

Aloha→経験豊富で、意見や状況を整理できる。

MM→分からないことが多いけど、鋭いことに気づく。

Zep:さて、今回は何について話しましょうか？

Aloha:そうですねー「テスト観点」はまだ先の予定ですね？

Zep:そうですね。まあ今回でもいいのですがね。

Aloha:「ユーザビリティ」とか「バリデーション」はどうですか？

Zep:確かに最近よく聞きますし、モヤモヤした用語ではありませんね。

MM:あの一・・・

Zep:何でしょう？

MM:私、「トレーサビリティ」について話してみたいです。

Aloha:おお、トレーサビリティですか？

MM:はい。いわゆるテスト用語というわけではないと思うのですが、頻繁に使いますし。

Zep:確かに、よく使うね。でも、何かモヤモヤポイントがある？あまり意味がブレるようなことは無いのかなと思って。

MM:んー、用語の捉え方ということではそうですね。でも、何かモヤモヤするんですよ。。

Zep:そうかあ、何だろう・・・

Aloha:こんな感じですか？用語そのものではなく、トレーサビリティを確保するための目的やHow to的ところがモヤモヤするような。結局、何のためにトレースできるようにするの？どうなっていればいいのか？みたいなモヤモヤでしょうか？

MM:ああ、そうです！そうです！そういうモヤモヤです。

Zep:なるほど！では、今日はその辺を議論しましょうかね！

Aloha:了解ですー

MM:ありがとうございます！

Zep:意味としては平気なんだよね？

MM:構成や内容、変更などを後から追跡できることですよね？

Zep:まあ、そうですね。

Aloha:それで、追跡できるということは、一体何のために、

どうしておけばいいのかがモヤモヤするってことですよ。

MM:はい、そうですね。お客さんとの会話でも「トレーサビリティ」という用語はよく使われるのですが、結局そのあたりがいつも分からなくなります。

Zep:では、そのあたりを議論しましょうか。何のためにトレーサビリティが必要なのでしょう？

Aloha:抜け漏れを把握でき安心感や納得感を得るってものと、関連や影響範囲の把握できて流用や改善をしやすくするためってことでしょうか。

MM:抜け漏れって例えばどういうことでしょうか？

Aloha:要求されたものがしっかり盛り込まれているとか、複数人で作業している場合に各担当にしっかり割り振れているとかってことだと思います。

Zep:では、関連や影響範囲の把握の例も挙げてもらえますか？

Aloha:こっちは、既にドキュメントなどがあってそれを流用して別のドキュメントを作成するときとか、何か問題があって振り返るときに、どういう意図でそうなっているのか理解するときかなと思います。

Zep:なるほど。では、何かを作っている時に確認するためのものと、作り終わった後に確認するためのものと、2点を言っていることになるんですね。

MM:私、トレーサビリティは作り終わった後の話なのかと思ってましたが、作っている時にも必要になるものなんですね。

Aloha:はい、そうだと思います。

Zep:では、どうすれば、この目的が達成できるのでしょうか。

Aloha:確かにこちらは難しいですよ。

MM:普段はどうしてますか？

Aloha:各要素にIDを振ったり、線で結ぶといったことはありますね。

Zep:もっと大きな括りでは、変更履歴を書くとか、参照ドキュメント名を書くとかもある意味そうですね。

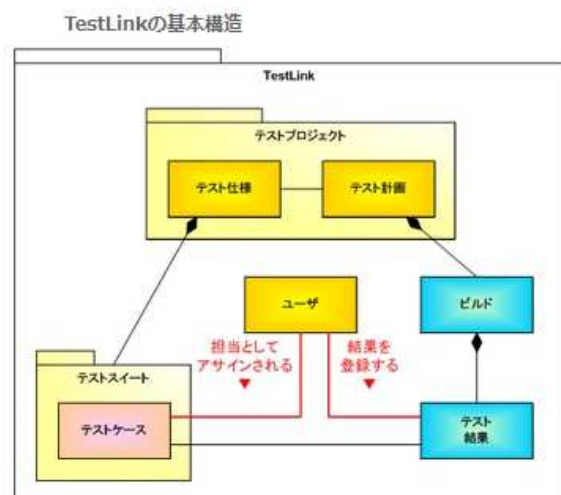
Aloha:まあそうですね。ただ、どちらにしても完璧に辿れるようにするのは難しいですよ。

MM:確かにトレーサビリティに気をつけて作ったつもりでも、後々遡りできないっていったことはよくある気がします。

Aloha:そうですね。それで、もっと気をつけてなんでもかんでも関連づけようとして、カオス状態になってしまうこともありますね。

Zep:量が多いとか複雑な場合、例えば検出不具合が多いとか、複数の関連テストケースがあるとか言った場合ですけど、ツールによってどうにかできる部分はありますよね。

MM:ああ、Testlinkとかですか？



Zep:そう。でもツールでどうにかできるようにするためにも、トレーサビリティの目的を考え、必要となる場面をどれだけ想定できるかが大事なんでしょうね。

Aloha:そうですね。どのようにトレーサビリティを確保するか、どう運用するかを決めないと破綻してしまいますよね。

MM:そういう課題があるとして、どういうポイントが分かると、抜け漏れ、関連や影響範囲の把握のできるのでしょうか？

Aloha:少し抽象的になりますが、「主張⇔根拠の関係」みたいなものと、「データの変換/加工の意図」みたいなところがポイントなのかなと思います。

Zep:なんとなく分かりますが、具体例を挙げてもらえますか？

Aloha:「主張⇔根拠」で言うと、例えばテストケースで実施していない箇所があったとして、主張は「そのテストケースはやらなくていい」となり、根拠は「何故なら、この機能は搭載されていないからだ」みたいな感じなんですけどね。

MM:なるほど、確かに必要な内容が気がしますし、そういう場面は良くありますよね。

Zep:他にも例えば、不具合を登録しなくていい場合の主張と根拠、このテスト条件を実施する必要がある場合の主張と根拠、この環境で実施しなければいけない場合の主張と根拠など、色々な場面でこの関係性は登場してきそうですね。

Aloha:はい。そういうところが明示できれば、抜け盛れの把握はしやすいのかなと思います。

MM:では、「データの変換/加工の意図」は、こういったものですか？

Aloha:例えば、機能仕様を基にテストアイテムを抽出するとか、機能を細分化するとか、不具合から不具合モードを抽出するとか言った場合に、基にしたものから何かを変換、加工しているわけで、その意図を理解する必要があるのかと思います。

Zep:確かに、そのまま使うことは少ないですからね。テスト側の視点で、機能を変換したり、不具合を蓄積してますからね。そのあたりが見えないと流用や活用は難しいですよ。

MM:テストの視点か。確かにそれが見えないと他の人、例えば開発者が見ても理解できないかもしれませんね。

Zep:ああ、そうですね。その辺を考慮できるようになると、トレーサビリティの質も一段上がりますねー

Aloha:そうですねー

Zep:とりあえず今回はこのあたりでやめておきます？

Aloha:MMさん、少しはモヤモヤ消えた？

MM:はい！ただ、どう実現できるかのところはまだアイデアが浮かばないです。。

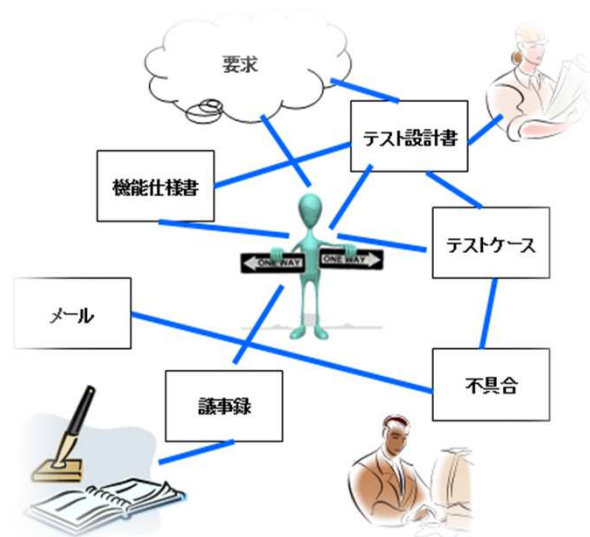
Zep:そうだね。今回の議論はあまり踏み込んで出来なかったからね。

Aloha:「トレーサビリティ」の実現方法みたいなところはまた別途検討していきましょう。

MM:はい、そうしましょう。

Zep:そうですね。では今回もありがとうございました～

Aloha&MM:ありがとうございました！



From the Editor

ベリナビ秋号を最後までお読み下さり、ありがとうございました。

今回初めて正式にお客様向け冊子を発刊するという事で、それぞれの技術者による記事の執筆から編集作業まで、想像以上の苦勞がありました。しかしその分、全14記事全てに魂がこもっております。読んでいただいた方に、少しでも「ああ、ベリサーブはこんな取り組みをしているんだな」と感じていただければ幸いです。では、次回の『ベリナビ冬号』もどうぞお楽しみに。





VERISERVE NAVIGATION『ベリナビ』 2014年 秋号

編集・発行:株式会社ベリサーブ

〒160-0023 東京都新宿区西新宿6-24-1西新宿三井ビル14F

ソリューション部:03-5909-5700

本誌についてのお問い合わせ先:ソリューション部

発行責任者:斎藤功 編集責任者:西村憲一郎 編集委員:豊本奈美江

verinavi@veriserve.co.jp

※本ベリナビの記事中に掲載する社名または製品名は、各社の商標または登録商標です。