ソフトウェアテストシンポジウム2010東京
(JaSST' 10 Tokyo)

HITACHI
Inspire the Next

HitachiSystems

レビュアーの3K(勘、経験、感性)を活発化させる発想術
「レビュー眼」のご紹介

2010年1月28日

株式会社 日立システムアンドサービス
品質保証部
角口 勝隆

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

Opening

＜自己紹介＞
- (株) 日立システムアンドサービスで、主にパッケージソフト およびソリューションサービスの品質保証業務に従事中。
- JaSST' 09 Kansaiで「レビュー眼」と称して、レビュアーの持つ暗黙知(勘・経験・度胸)の解明について考察。
  http://www.jasst.jp/archives/jasst09w/pdf/A2-4.pdf
- JaSST' 10 Tokyoでは、JaSST' 09 Kansaiの内容を、より具体化し かつ 応用可能なノウハウを含めて紹介する。

Kadoguchi Yoshitaka
角口 勝隆
y-kadoguchi@hitachi-system.co.jp

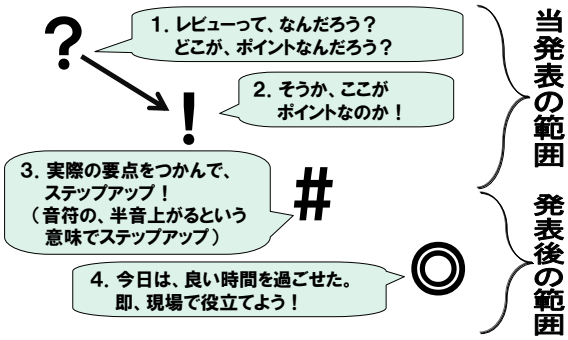
All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

◆本発表の、対象とする方

- いつも納期直前にバグが多発し、**もっと早くバグに気づけないものか**と、悩んでいる方
- 仕様誤りによる、**作り直しや再テストを軽減させたい**と考えている方
- テスト設計で「**検討漏れや不足がある**」と悩んでいる方
- 「**仕様書の行間を読め**」と言われたが、見えないものをどうやって見れば良いのか、まじめに悩んでいる方
- 「**レビューって、なんだろう？**」という方
- レビューで**指摘されるのが、うんざり**という方
- レビュアーとして、**ステップアップしたい方**
- 上述するような部下を抱えている方

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

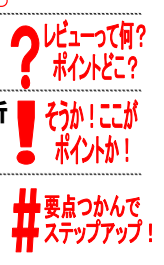
◆本日の流れ



All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

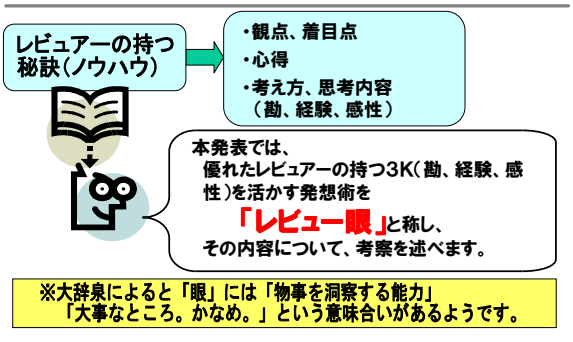
Contents

はじめに
1. レビューの一般論
2. 現状のレビューの問題点
3. レビュアーの思考プロセスの分析
4. レビュー向け発想術の考察
5. レビュー向け発想術のまとめ
6. 今後の展望



All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

◆「レビュー眼」とは



※大辞泉によると「眼」には「物事を洞察する能力」「大事なところ。かなめ。」という意味合いがあるようです。

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

◆「レビュー眼」の位置づけ

レビュー眼：発想術
- 観察法
- 発想法

レビューに必要な要素

パーソナル・スキル

- 計画、方針
- プロセス
- 規格,規則
- 技法・ツール

ウォーターフォール、スパイラル、アジャイル など

インスペクション、ウォークスルー、バスアラウンドレビュー、ピアレビュー など

パーソナルスキル（レビュアーの質）が向上すると、各種レビュー技法やプロセスとの相乗効果を期待できる。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

◆「レビュー眼」を極めれば、

①プロジェクト全体の品質リスクを、事前に予測できる。

②テスト設計技術が向上する。

レビュー眼で不具合(リスク)を先読み

（レビューとテストは、補完関係）

先読みした不具合が存在しないことを、テストや監査で実証。

不具合の作り込みを防止

・レビュー時の着眼点を実機検証
・不具合が、存在しないことを確認

早期に「品質」を確保することで、コスト／納期が予定通りに推進され、プロジェクトを成功に導くことが可能。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

◆QCDについて

Q：Quality（品質）　例）うまい
C：Cost（費用）　やすい
D：Delivery（納期）　はやい

Case	品質	費用	納期	事例	評価
1	△	○	○	費用の限界と、納期が近づいたので、ある程度の品質（最低限の動作が可能）状態で、完了させた。	必ずしも、お客様にとっての品質が良いとはいえない。納品後のクレーム対応の損失あり。
2	△ ↓ ○	○ ↓ △	○	納期直前で品質に問題があることが判明。短期間で品質を確保するために、人員をかなり投入した。	プロジェクトとして、必ずしも成功とはいえない。
3	× ↓ △	×	×	品質が悪く、デグレードが多発。後戻り作業が多く、修正／見直しに人員と期間を費やした。	最悪の事態
4	◎	○	○	品質が早期に確保され、後戻り作業や追加作業が無く、予定通りの費用と納期で完了した。	理想形

「品質第一」とすれば、費用・納期も成功へ導くことが可能。
逆説的に、品質を疎かにすると、費用・納期が危うくなる。
「品質」だけを、後から付け足すようなことはできない。

「レビュー」が有効だと、言われている

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

Contents

はじめに
1. レビューの一般論
2. 現状のレビューの問題点
3. レビュアーの思考プロセスの分析
4. レビュー向け発想術の考察
5. レビュー向け発想術のまとめ
6. 今後の展望

レビューって何？ポイントどこ？

そうか！ここがポイントか！

要点つかんでステップアップ！

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

1-1 レビューの定義について

レビュー（Review）：批評、再検討、再確認

Re(再)-View(見)：つまり、見直し・再確認という意味合いでしょうか。

レビューの目的

①埋め込まれた不具合の摘出

②埋め込まれる不具合の予防

上流工程から、発生する不具合を除去することで、「品質を作りこむ」ことを目的とする。

レビューの考え方は様々ですが、ここでは、プログラムを動かす前に、再確認を行うものと定義し、いわゆる「デザインレビュー」を対象とします

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

1-2 レビューの重要性

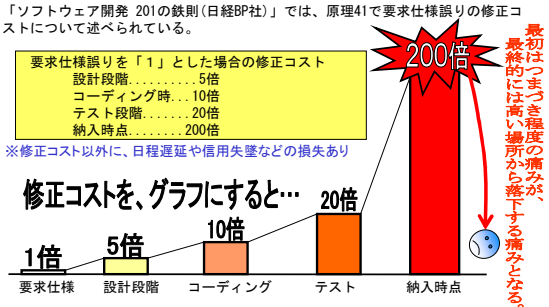
レビューによる問題摘出の、コストへの貢献度について

「ソフトウェア開発 201の鉄則（日経BP社）」では、原理41で要求仕様誤りの修正コストについて述べられている。

要求仕様誤りを「1」とした場合の修正コスト
- 設計段階……5倍
- コーディング時…10倍
- テスト段階……20倍
- 納入時点……200倍

※修正コスト以外に、日程遅延や信用失墜などの損失あり

修正コストを、グラフにすると…



All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

1-3:レビュー技法の一例について

主な目的	技法	概要
対象の合否判定とプロセス改善	インスペクション	専門チームによるレビューを行う（または、レビューを受ける）
不明点や、間違い/勘違いの発見	ピアレビュー	「同僚によるチェック」の意味合い。 ※幅広い意味があるが、ここでは狭義に捉える。
成果物のエラーの発見	ウォークスルー	役割を決めて、実際の処理流れに沿って、抜けや検討漏れがないか確認する。
多数の意見を収集	パスアラウンド	「回覧・回し読み」などで、たくさんの人の意見を集める。（多重同時進行）

参考文献:「ソフトウェア品質保証入門(日科技連出版社)」、「ピアレビュー(日経BP社)」

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

Contents

はじめに
1. レビューの一般論
2. 現状のレビューの問題点
3. レビュアーの思考プロセスの分析
4. レビュー向け発想術の考察
5. レビュー向け発想術のまとめ
6. 今後の展望

レビューって何？ポイントどこ？

そうか！ここがポイントか！

#要点つかんでステップアップ！

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

2-1 レビュー効果が上がらない現場の例

◆形式的なレビューになっている

機能仕様レビュー結果報告：
- 100ページあたり、30件の不具合を指摘しました。（指摘内容は、誤字脱字，様式不正などばかり）
- 40項目のチェックリストに従い、全て確認し、問題はありませんでした。

⇒ 本当に大丈夫なのか??

◆不具合を見逃した反省は、あまり活かされていない

社外不良反省会：
- そのような使われ方は想定していませんでした。
- テストケースが思い浮かびませんでした。
- 仕様書に記載がなく、テストが漏れました。

⇒ 何故、レビューやテスト設計時に思い浮かばない?

※再発防止策は、作り込み側へ求めることが多い。
※チェックリストを強化するだけでは、確認項目が膨大になり、結局見なくなる。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

2-2 レビューで問題を見逃す主な原因

①着眼漏れ（見ようとしない、気付かない）

②思考の偏り（思い込み、思い違い）
（かたよ）

※思い込み：指示内容の制約等に、気付かない
※思い違い：知らない または 間違った確信。

- ソフトウェアに対する「要求」や「仕様」は、目に見えない。
- 仕様書があるといっても、完全な情報を得る／伝えることは難しい。

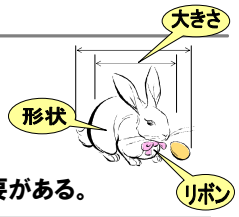
見えないものを、どうやって読む??

いわゆる、「行間を読め」ということ？

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

例えばですが、、、

隣の部屋にいるなど、右の図が見えない人に、ウサギの詳細を伝達する場合。
見た目だけではなく、様々な観点で観察をし、情報を得る必要がある。



ソフトウェアも同じで、内容を深く理解するには、対象物を見る目(観点)が大切と考える。
（観点を持てば、「仕様書の行間」も見えてくる）

様々なレビュー技法の活用や、プロセス改善を行ったとしても、レビュアーが「観点」を持たなければ、効果は上がらないと考える。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

Contents

はじめに
1. レビューの一般論
2. 現状のレビューの問題点
3. レビュアーの思考プロセスの分析
4. レビュー向け発想術の考察
5. レビュー向け発想術のまとめ
6. 今後の展望

レビューって何？ポイントどこ？

そうか！ここがポイントか！

#要点つかんでステップアップ！

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

3-1 レビュー時の思考プロセス

レビュアーは、どのような観点で対象物を見るのか？

下記、架空Webサイト「モクシィ」の新機能(仕様概要)を題材に、レビュー時の思考プロセスを分析する。

新機能の仕様概要について：

携帯電話から、画像データをWebサーバへアップする機能を備える。ただし、関連者のみ利用可能。

(詳細は、仕様概要を説明後に実施されるという前提で実施)

※思考プロセスは人それぞれ多様であり、あくまで一例として挙げます。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

まずは、気になる点に着目する。

- 契約との齟齬
- 提案書との整合性

新機能の仕様概要について：

携帯電話から、画像データをWebサーバへアップする機能を備える。ただし、関連者のみ利用可能。

<レビュー時の思考プロセス>
1)関心を持つ(感性)
2)ヒラメキ、勘が働く(勘)
3)仮説を立てる(経験)
4)指摘する(度胸)

気になるキーワードを捕らえる。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

関連者って、誰？

新機能の仕様概要について：

携帯電話から、画像データをWebサーバへアップする機能を備える。ただし、関連者のみ利用可能。

<レビュー時の思考プロセス>
1)関心を持つ(感性)
2)ヒラメキ、勘が働く(勘)
3)仮説を立てる(経験)
4)指摘する(度胸)

キーワードから、何か、ヒラメキが生じます。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

関連者って、誰？

- "関連者"の定義は？
- 認証の仕組みは大丈夫？
- 関連者の登録は、どのように行う？
- 関連者以外がアクセスすると、どうなる？

<レビュー時の思考プロセス>
1)関心を持つ(感性)
2)ヒラメキ、勘が働く(勘)
3)仮説を立てる(経験)
4)指摘する(度胸)

この内容で正しいのか、仮説を立てた上で、検討する。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

関連者って、誰？

- "関連者"の定義は？
- 認証の仕組みは大丈夫？
- 関連者の登録は、どのように行う？
- 関連者以外がアクセスすると、どうなる？

<レビュー時の思考プロセス>
1)関心を持つ(感性)
2)ヒラメキ、勘が働く(勘)
3)仮説を立てる(経験)
4)指摘する(度胸)

仮説に基づいて質問・指摘を行う。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

3-2 優れたレビュアーの指摘内容を分析

1) 優れたレビュアーは、
怪しいポイント(着眼点)を押さえている！
2) 優れたレビュアーは、
的確に不具合を摘出できる！
3) 優れたレビュアーは、
仕様の矛盾・抜けを見破れる！
4) 優れたレビュアーは、
不具合を予防するための、
適切なアドバイスを行うことができる！

優れたレビュアーの秘訣は、後述する。

QA技術分野であり、紙面 および発表時間制約の都合上、別の機会に取り扱う。

課題①

これら、優れたレビュアーのノウハウを一般化できないか？

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

3-3 エラーを作り込む箇所の分析

レビュー時の思考プロセス：

1）関心を持つ（感性）
2）ヒラメキ、勘が働く（勘）
3）仮説を立てる（経験）
4）指摘する（度胸）

ここが、エラーの根本原因！！
着眼漏れ
思考の偏り

課題②
着眼漏れや、思い込み／思い違いを低減し、蓄積された勘や経験・感性を活かすには？

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

3-4 レビューにおける改善課題のまとめ

課題① 優れたレビュアーのノウハウを一般化
- 怪しいポイントを押さえるコツとは？
- 的確に不具合を摘出できるコツは？
  そして、
  仕様の矛盾，抜けを見破るコツは？

課題② 着眼漏れや、思い込み／思い違いを低減し、蓄積された勘や経験・感性を活かすには？
- 対象物を見る目（観点）を体系化
- 思考の偏りをなくす
  （陥りやすい死角をなくす）
- 蓄積された勘や経験を、
  スムーズに引き出す

これらの課題を、解決する発想術を、次章で考察する

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

Contents

はじめに
1. レビューの一般論
2. 現状のレビューの問題点
3. レビュアーの思考プロセスの分析
4. レビュー向け発想術の考察
5. レビュー向け発想術のまとめ
6. 今後の展望

レビューって何？ポイントどこ？
そうか！ここがポイントか！
#要点つかんでステップアップ！

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

レビュー向け発想術「レビュー眼」

ここでは、レビュー向け発想術を、レビュー眼と称します。

レビュー眼
- 観察法
  - ・怪しいポイント
  - ・秘訣（心得）
  - ・思考フレーズ
- 発想法
  - ①定型連想型
  - ②エラー推測型
  - ③自由発想型

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

4-1-1 怪しいポイントを押さえるコツとは？

怪しいポイントの共通点

優れたレビュアーは、バグのありそうな箇所を狙っている。
そのときの意識は「バグがあるかも知れない」。

ここが大事！と考える。

トム・デマルコ と ティモシー・リスターの著書
「熊とワルツを ～リスクを愉しむプロジェクト管理～」では、
「…かも知れない」というフレーズが多数出てくる。

このポイントを、身近な例に置き換えて考えてみよう。

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

4-1-2 怪しいポイントの身近な例

車の運転で、見通しの悪い角を曲がる場合。

初心者ドライバー	ベテランのドライバー
大丈夫だろう！ GoGo！	ひょっとしたら、人がいるかも知れない。いったん、停止しよう。
リスクを予知しない、危険運転	リスクを予知した、安全運転

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

4-1-3 優れたレビュアーの秘訣(心得)

1	不良を見つけようとするから、見つかる。(第三者検証は効果あり)
2	実際にその仕様に基づいて次工程で作れるのか、想像する。
3	「How(どのように)」の議論よりも「Why(なぜ)」「What(なに)」が大事。
4	「こうあるべき」という姿を思い浮かべて対比する。(価値感を持つ)
5	後から仕様を付け加えたような、「ただし」で始まる箇所が怪しい。
6	「A」でなく、「B」でないなら「C」とは限らない。(論理的に考察)
7	「○○○のはず」という発言は怪しい!(曖昧さを排除) 先入観を持たない
8	「絶対」という言葉は「絶対」にない。(「絶対安全!」って、ホント?)
9	設計者が面倒臭がるようなところに、バグがある。(ある意味、執念!)

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

4-1-4 思考フレーズ

怪しいポイント発見のコツ(心構え)

よく観察して「的確に情報をキャッチ」

レビュー時は「かもしれない」推論

レビュー時の思考フレーズ

[着目点]について[考察]		[推論]
もし、○○が □□を	→ ××すれば △△だったら	→ ×××になるかもしれない (または、どうなるの?)

レビュー時の思考フレーズを使うことで、
多くの事柄をひとつずつ分解して、考察を行うことが可能。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

レビュー向け発想術「レビュー眼」

ここでは、レビュー向け発想術を、レビュー眼と称します。

レビュー眼
- 観察法 ・怪しいポイント ・秘訣(心得) ・思考フレーズ
- 発想法
  - ①定型連想型
  - ②エラー推測型
  - ③自由発想型

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

4-2-1 発想メカニズムの考察

ベテランのレビュアーは、
無意識に【ヒラメキ、勘】で不具合を連想している。
いわゆる「アナロジー」が働いている状態。

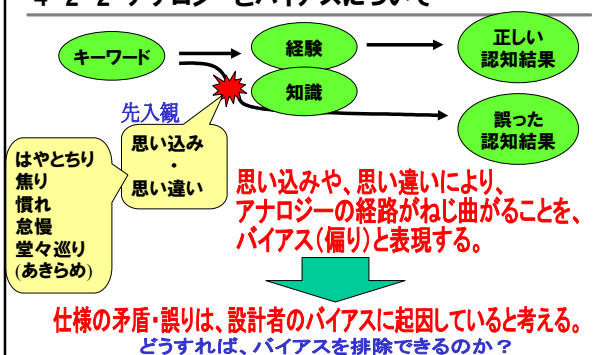
アナロジー:「類推する」という意味。

だれしも無意識に、キーワードから類推される
知識・経験を連想している。

ヒラメキ・勘は、「アナロジー」により産み出されると考える。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

4-2-2 アナロジーとバイアスについて



思い込みや、思い違いにより、
アナロジーの経路がねじ曲がることを、
バイアス(偏り)と表現する。

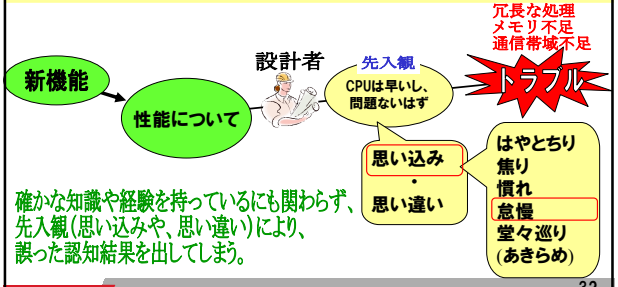
仕様の矛盾・誤りは、設計者のバイアスに起因していると考える。

どうすれば、バイアスを排除できるのか?

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

バイアスの一例

作成した当事者は、「大丈夫だろう」という、
先入観を持った物事の見方をしてしまいがち。

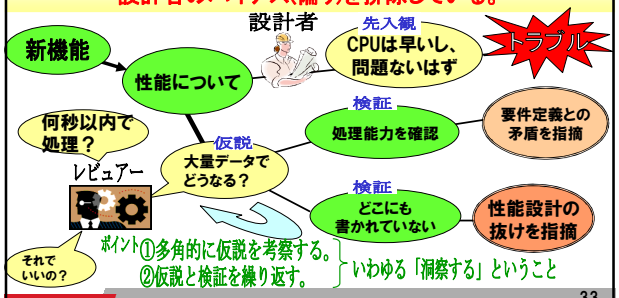


確かな知識や経験を持っているにも関わらず、
先入観(思い込みや、思い違い)により、
誤った認知結果を出してしまう。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

レビュー時のバイアス排除例

先入観を除いて、「なぜなの？」「大丈夫なの？」と、
単純な問い(仮説)と確認(検証)を繰り返し、
設計者のバイアス(偏り)を排除している。



All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

4-2-3 的確な仮説や推論を産み出すには？

レビューのポイントは、先入観を取り除いて、
- 多角的に仮説を考察する。そして、
- 仮説と検証を繰り返す。

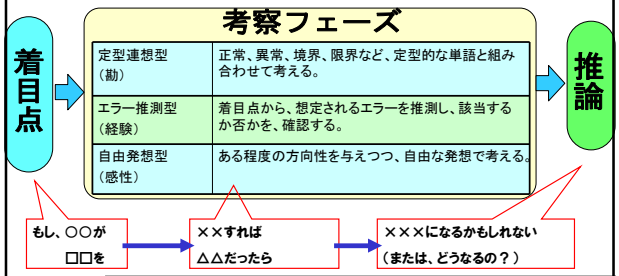
では、どうすれば、的確な仮説や推論を出せるのか？

仮説を産み出す(発想する)方法、
つまり、「発想法」について考察。

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

4-2-4 レビュー向け発想法の形式について

レビューの目的は、不具合の発見と予知。
不具合を摘出する指向で考察を行う場合、下記3種類の
発想形式があると考える。



All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

レビュー向け発想術「レビュー眼」

ここでは、レビュー向け発想術を、レビュー眼と称します。

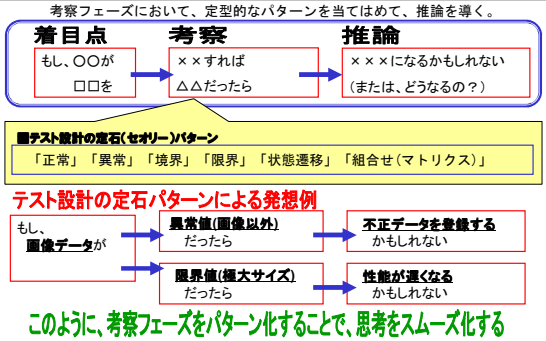
- レビュー眼
  - 観察法
  - 発想法
    - ①定型連想型
    - ②エラー推測型
    - ③自由発想型

・怪しいポイント
・秘訣（心得）
・思考フレーズ

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

①定型連想型発想法について

考察フェーズにおいて、定型的なパターンを当てはめて、推論を導く。



このように、考察フェーズをパターン化することで、思考をスムーズ化する

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

その他の、定型連想型発想法について

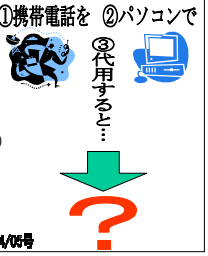
発想法の例：SCAMPER

複数のキーワードを関連付けるなどして、発想する。

これらの頭文字をとって、「SCAMPER」。

- Substitute（代える、代用する）
- Combine（組み合わせる）
- Adapt（適応させる）
- Modify（修正する）
- Put to other uses（ほかの使いみち）
- Eliminate（省略する、除去する）
- Rearrange（再調整する）

「モクシィ」において、
①携帯電話を ②パソコンで
③代用すると…



参考文献：日経ITプロフェッショナル(日経BP社)、2004/05号

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

その他、使えそうな定型連想パターン(１)

■テンプレート

規格、規則類
類似システムなど、過去の成果物
観点リスト など

テンプレートに当てはめたり、
何かと比較をすることで、
新たな気付きを得たり、
検討漏れを防止することが可能。

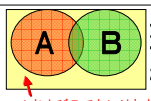
■トランザクション

追加(作成)　競合
更新　排他
参照(検索)　中断
削除
復元

データ操作の定型的パターン。
住所録やスケジュール帳など、
データを扱うケースに当てはめることで
検討漏れの防止につながる。

■データ組合せ

AはBと同じ
AにBを含む
BにAを含む
AとBの一部が重なっている
AとBは一致しない



AおよびBでないけれど、Cとは限らない

左記はデータ組合せのパターンであるが、
マトリクス表で組合せを網羅することで、
検討漏れを防止することが可能。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

その他、使えそうな定型連想パターン(２)

■ソフトウェア品質特性ISO/IEC9126(JIS X 0129)

特性	品質副特性
機能性	合目的性, 正確性, 相互運用性, セキュリティ, 標準適合性
信頼性	成熟性, 障害許容性, 回復性, 標準適合性
使用性	理解性, 習得性, 運用性, 魅力性, 標準適合性
効率性	時間効率性, 資源効率性, 標準適合性
保守性	解析性, 変更性, 安定性, 試験性, 標準適合性
移植性	環境適応性, 設置性, 共存性, 置換性, 標準適合性

参考URL: http://ja.wikipedia.org/wiki/ISO_9126

■非機能要件のパターン

非機能要件(10個の主特性)
機能性，信頼性，使用性，効率性，保守性，移植性，障害抑制性，効果性，運用性，技術要件
詳細は、JUAS(日本情報システム・ユーザー協会)発行の「非機能要求仕様定義ガイドライン」で述べられている。

■システムテストのカテゴリ

機能部分テスト，ボリューム，ストレス，有用度(ユーザビリティ)，セキュリティ，効率(パフォーマンス)，ストレージ，構成，互換性/構成/変換，設置，信頼性，回復，サービス性，文書，手続き
参考：Glenford J.Myersは、著書「ソフトウェア・テストの技法　第２版」で、システムテスト実施内容について詳細を述べている。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

レビュー向け発想術「レビュー眼」

ここでは、レビュー向け発想術を、レビュー眼と称します。

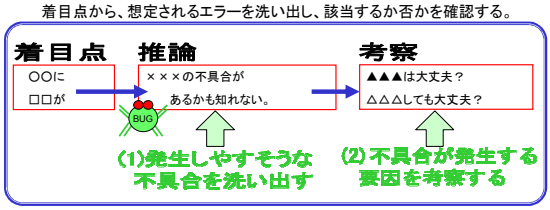
レビュー眼
- 観察法
- 発想法
  - ①定型連想型
  - **②エラー推測型**
  - ③自由発想型

・怪しいポイント
・秘訣(心得)
・思考フレーズ

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

②エラー推測型発想法について

着目点から、想定されるエラーを洗い出し、該当するか否かを確認する。



エラー推測による発想例

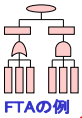
新システムに
- 脆弱性がある かも知れない
  - ・OSのセキュリティ脆弱性
  - ・プログラムの不具合
  - ・通信の傍受
- 移行に失敗する かも知れない
  - ・移行作業の手順ミス
  - ・データ移行プログラムの不良
  - ・インタフェース不良

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

不具合を考察する時の問題点

FMEAまたはFTAで、発生する不具合(現象)から要因を検討すると、少し手間。
(図表や、リスク分析までは、必須ではない)
※FMEA(Failure Mode and Effect Analysis)
FTA(Fault Tree Analysis)

発生し得る不具合(現象)を予測するには、ある程度の知識・経験が必要。
不具合事例やナレッジデータベースは存在するけれど、探すのが手間。
一文字違うと、検索に引っかからない。



FTAの例

発想が妨げられる。

分析/抽象化

そこで、

過去の不具合を分析/抽象化し、レビュー観点として再利用する。

不具合の発生手順を分析し、レビューへ応用させる一連の手法を、
ここでは「FM2R」と称する。(FM2R:Failure Mode To Reviewの略。)

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

「モクシィ」で発生した不具合事例

現象：画像データが流出してしまった。
発生条件/手順：クロスサイトスクリプティングによる被害
原因：実装漏れ(特殊文字のサニタイジング(無害化)漏れ)
要因：考慮不足
動機的要因：(1)仕様書に記載されておらず、検討漏れ
(2)フレームワークで実装済みと、思い込み
作りこみ工程：詳細設計
見逃し要因：テスト観点欠落(フレームワークで実装済みと誤解)

では、

どの情報をレビュー観点として再利用すれば良いのか？
(つまり、どのような観点が漏れたのか？)

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

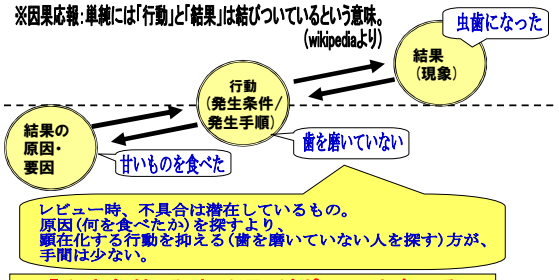
原因から結果までの経過を考察

身近な四字熟語に、「因果応報」がある。

※因果応報:単純には「行動」と「結果」は結びついているという意味。(wikipediaより)

顕在化した不具合

潜在する不具合



レビュー時、不具合は潜在しているもの。
原因(何を食べたか)を探すより、
顕在化する行動を抑える(歯を磨いていない人を探す)方が、
手間は少ない。

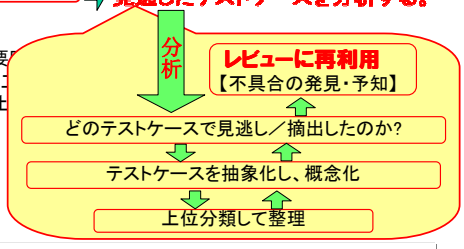
「発生条件/発生手順」がポイントと考える

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

不具合を、レビュー観点として利用可能なように分析するには

不具合事例
- 現象
- 発生条件/手順
- 原因
- 要因
- 動機的要
- 作りこみ
- 再発防止

発生条件/手順から、レビュー時に見逃したテストケースを分析する。

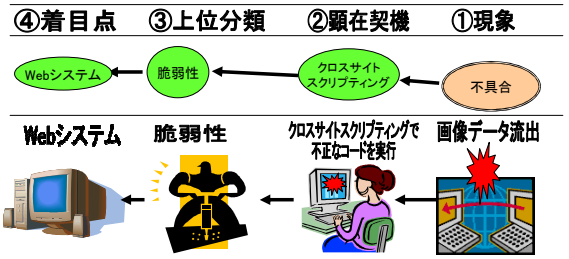


※「発生条件/手順」の明確化により、デバッグ時および修正確認作業の精度向上も期待することが可能。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

エラー推測のための、不具合分析例

不具合を、上位分類と結びつけて、蓄積。

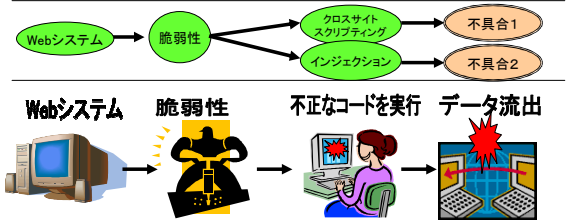


All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

上位分類からの、エラー推測例

蓄積した上位分類から、不具合が複数連想される。

(アナロジーを活用するための道具として、利用可能)

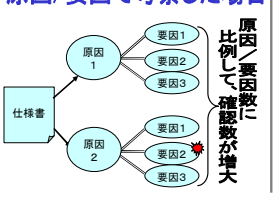


※筆者としては、発生した現象(不具合)と発生契機から、原因箇所の特定へ結びつける、トラブルシューティングにも応用が可能と考える。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

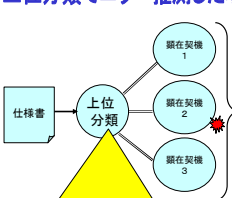
不具合の上位分類化による効果について

原因/要因で考察した場合



原因/要因数に比例して、確認数が増大

上位分類でエラー推測した場合



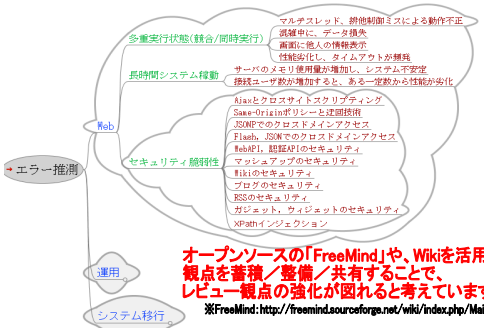
少ない観点で、多くの不具合にたどり着く

上位分類は、用途(レビュー目的)に
合わせて、類推辞典(シソーラス)として
整理するのが良いと考える。

■参考
※「基本から学ぶソフトウェアテスト(日経BP社)」の巻末付録に、
400種類以上の「よくあるソフトウェア不具合」が記載されており、参考となる。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

不具合事例をマインドマップで整理した例



**オープンソースの「FreeMind」や、Wikiを活用し、
観点を蓄積/整備/共有することで、
レビュー観点の強化が図れると考えています。**

※FreeMind: http://freemind.sourceforge.net/wiki/index.php/Main_Page

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

レビュー向け発想術「レビュー眼」

ここでは、レビュー向け発想術を、レビュー眼と称します。

- レビュー眼
  - 観察法
  - 発想法
    - ①定型連想型
    - ②エラー推測型
    - ③自由発想型

・怪しいポイント
・秘訣（心得）
・思考フレーズ

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

③自由発想型発想法について

発想法の例：飛び石

着目点(キーワード)をもとに、発想を飛躍させる。

誇張:極端な条件や状況を設定する

願望:現実や実現可能性などを一切考えずに理想形を思い浮かべる

歪曲:組織や人、モノなどの関係をねじ曲げてみる

逆転:関係を逆転させる

着目点	考察	推論
もし、○○が □□を	××すれば △△だったら	どうなるの？

参考文献：日経ITプロフェッショナル(日経BP社), 2004/05号

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

「モクシィ」における、「飛び石」での発想例

誇張:極端な条件や状況を設定する
→ システム利用者が10倍になったら…
→ 10年後も正常稼動させるには…

願望:現実や実現可能性などを一切考えずに理想形を思い浮かべる
→ 誰でも、ワンボタンで操作可能な、使いやすさ

歪曲:組織や人、モノなどの関係をねじ曲げてみる
→ 利用者が、他社のスパイだったら…

逆転:関係を逆転させる
→ アップロードができるなら、ダウンロードは？

※筆者としては、「願望」×「逆転」で、「反目標」を検討するなど、制約条件についても応用を利かせれば良いと考えている。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

その他、使えそうな、自由発想例を取り上げる。

■情報の構造化

「分類」：類似点を分類整理する
「関連」：相反、相互作用、影響範囲
「配列」：時間や量で並べる

キーワードを、分類・関連付け・並べかえることで、何かが見えてくる。

■ペルソナ

仮想ユーザを定義し、それぞれの仮想ユーザの立場から、利用方法（用い方）や、利用される場面を検討する。（例：日立花子 25才、うっかりミスが多い、etc…）

■発想を変える4つの視点

視点（立場）	いまの位置・立場そのままでなく、相手の立場、他人の視点、子供の視点、外国人の視点、過去からの視点、未来からの視点、上下前後左右、表裏等々
見かけ（外観）	見えている形・大きさ・構造のままに見ない、大きくしたり小さくしたり、分けたり合わせたり、伸ばしたり縮めたり、早くしたり遅くしたり、前後上下を変えたり等々
意味（価値）	分かっている常識・知識のままに見ない、別の意味、裏の意味、逆の価値、具体化したり抽象化したり、まとめたりわけたり、喩えたり等々
条件（状況）	「いま」「ここ」だけでのピンポイントでなく、5年後、10年後、100年後、1000年後あるいは5年前、10年前、100年前、1000年前等々

参考文献：日経ITプロフェッショナル(日経BP社), 2004/05号

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

自由発想を成功させるポイント

レビュアーが、思いついたことを述べるだけでは、話が発散するだけで、的を得た協議を行うことはできない。

以下の事項をレビュー開始時に明確化し、レビュアーが共有することで、ベクトルを合わせることが大事と考える。

1. レビュー方針、進め方の明確化
   （例：検討すべきポイントや、アジェンダの明確化）
2. 成功条件の明確化
   （例：ピーク時でも、5秒以内のレスポンス 等）
3. 反目標の明確化
   （例：15分以上のシステムダウンは回避 等）

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

発想法における心得（コツ）など

定型連想型	正常、異常、境界、限界など、定型的な単語と組み合わせて考える。
エラー推測型	着目点から想定されるエラーを推測し、該当するか否かを、確認する。
自由発想型	ある程度の方向性を与えつつ、自由な発想で考える。

【ヒラメキはキラメキ】
一瞬のヒラメキ（インスピレーション）を得るには、
"一度は見直す"というクセが大事。

マックス桐島談
（ハリウッド映画プロデューサー）

自動車教習所の教官談

【ミラーに死角あり】
車線変更する際、ミラーには死角があることを認識した上で、
［バックミラー］→［サイドミラー］→［目視確認］の
3つを確認しなければならない。
（レビューも同様に、多角的な確認が必要）

今日はもう帰って、明日考えよう。
（キーワードを詰め込んでおくと、
翌朝の通勤中に思いつくこと多数）

筆者談

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

Contents

はじめに

1. レビューの一般論
2. 現状のレビューの問題点
3. レビュアーの思考プロセスの分析
4. レビュー向け発想術の考察
5. レビュー向け発想術のまとめ
6. 今後の展望

？レビューって何？ポイントどこ？

！そうか！ここがポイントか！

＃要点つかんでステップアップ！

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

5-1 レビュー眼のまとめ(１)

課題①:優れたレビュアーのノウハウを一般化

怪しいポイントを押さえるコツ

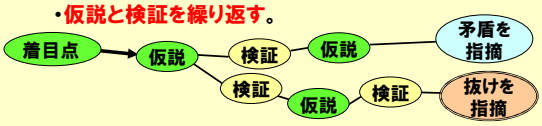
- よく観察して「的確に情報をキャッチ」
- レビュー時は「かもしれない」推論

的確に不具合を摘出できるコツ、仕様の矛盾，抜けを見破るコツ

先入観を取り除いて、

- 多角的に仮説を考察する。そして、
- 仮説と検証を繰り返す。

いわゆる「洞察する」ということ



All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

5-2 レビュー眼のまとめ(２)

課題②:着眼漏れや、思い込み／思い違いを低減し、蓄積された勘や経験・感性を活かす

レビュー向け発想術(3種類の発想形式)で考察することにより、「着眼漏れ」「思考の偏り(思い込み／思い違い)」に起因するエラーを低減させつつ、蓄積された情報を有効に活用することが可能。

レビュー向け発想術に求めた要件	定型連想型(勘)	エラー推測型(経験)	自由発想型(感性)
対象物を見る目(観点)の体系化	○	△ 抽象化と分類が必要	○
思考の偏りをなくす(陥りやすい死角をなくす)	○ 抜け・誤り防止	○ リスク予測	○ 新たな気付き
蓄積された勘や経験を、スムーズに引き出す	△ 現在の知識以上のものは得られない	○ キーワードをもとに、知識を増やせる	※ 既成概念に捕らわれない発想が可能
的確な仮説と推論を導き出す	○	○	△ 「反目標」を検討することで、的確性が増す

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

Contents

はじめに

1. レビューの一般論
2. 現状のレビューの問題点
3. レビュアーの思考プロセスの分析
4. レビュー向け発想術の考察
5. レビュー向け発想術のまとめ
6. 今後の展望

？レビューって何？ポイントどこ？

！そうか！ここがポイントか！

＃要点つかんでステップアップ！

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

6-1 レビュー観点の、現場での活用例

レビュー観点なし	レビュー観点あり
レビュー担当者は、A機能は○○くん、B機能は△△くん。よろしくね。	○○くんは、この観点で、△△くんは、こちらの観点で、機能を横断的に見てね。
・知識,経験が少ないと、指摘できない問題あり。 ・仕様書に書いてあることしか、思い浮かばない。 ・機能ごとに、レビュー結果のバラツキあり。 ・機能間にまたがるような問題は気付きにくい。 ・観点漏れによる見逃しリスクあり	・知識,経験が少なくても、観点に基づく質問や仮説で、問題を摘出しやすくなる。 ・仕様書に書いていないことも、見破りやすい(洞察しやすい) ・機能ごとの、レビュー内容が一定する。 ・機能間にまたがるような問題も発見しやすい ・観点が明確。(見逃し漏れが少ない)

なお、以下のいずれかが不足しても、レビュー効果を十分に発揮できないと考える。
(1) レビュアーに与える情報 →入力される情報の精度を向上させること
(2) 経験(蓄積された知識) →適切なレビュアーを割り当てること
(3) 多角的な観点 →複数人(2~4名)でレビューを実施すること

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

6-2 今後の展望について(１)

チェックリストによるレビューでは、「一元的品質要素」の確保までが実現可能な範囲であった。レビュー眼での発想法「自由発想型」は、「魅力的品質」についての議論を活発化させる要素となる。これにより、真のお客様価値の実現を図ることが可能と考える。

品質要素	品質の定義	チェックリスト	発想法
当たり前品質要素	それが充足されれば当たり前と受け取られるが、不十分であれば不満を引き起こす品質要素。(実現されない場合、プロジェクトの失敗を意味する)	○：定性的な要素はチェックリストで対応可能。	◎：「エラー推測型」発想法により、過去の障害情報を有効活用し、より確実な不具合摘出／障害予防が可能
一元的品質要素	それが充足されれば満足、不十分であれば不満を引き起こす品質要素。(要求、要件の実現)	○：「要件定義」に従った検証で、対応可能	◎：「定型連想型」発想法により、要件の曖昧さや、機能を実現をする上での品質リスクを早期に検討可能
魅力的品質要素	それが充足されれば満足を与えるが、不十分であってもしかたがないと受け取られる品質要素。(安全性、使いやすさ、利便性等)	×：真の「要求」を捉えた検討が必要だが、チェックリスト化は難しい	◎：「自由発想型」発想法に基づき、「真のお客様価値」について検討可能

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

6－2　今後の展望について（2）

「**上流工程で品質を作りこめ**」と言われるが、これは単に「**不具合をレビューで除去すること**」だけではない。

「品質」って、一言で言うと何？

品質を「お客様の満足度」と定義した場合、
「品質（お客様の満足度）を創造する」タイミングは、「デザイン・レビュー」が最適であると考える。
今一度「デザイン・レビュー」を活発化させることが、真のお客様価値の実現を図る要点と考える。

「デザイン・レビュー」の活発化により、より高度なお客様価値の実現を期待することができると考える。

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

6-3　最後に

Think! it's yourself.

我々の発見力には、どんな最新ツールもかなわない！

・各種テスト技法や、高額なツールも、“観点”が無ければ実施されないし、不良も発見されない。何ごともレビュー眼のような**“観点”を持つことが大切**と考える。

レビューはクリエイティブ(創造的)に！

・どんなにコンピュータが発達しても、何かを創造できるのは、人間だけの取り柄。
・レビューは問題発見だけではなく、お客様にとっての、**真の品質（満足）を創造できる場**でもある。
「品質（満足）を作りこむ」タイミングは、デザインレビューが最適。
・今一度、**ものづくりの大切さ・素晴らしさを振り返ろう**。

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

Closing

－謝辞－

・当プレゼンテーション資料を作成するあたり、JaSST Kansai実行委員、TEF関西の皆様、ならびに 弊社の諸先輩方に“多大な”ご協力を頂きましたことを、この場をお借りして、御礼申し上げます。
・また、予稿集作成 ならびに シンポジウム開催準備，会場運営にあたられた方々にも、この場をお借りして、御礼申し上げます。

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

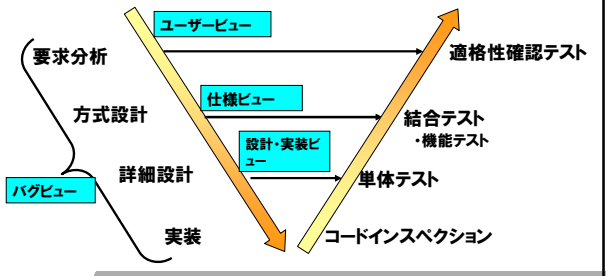
HITACHI
Inspire the Next

HitachiSystems

付録1－1　場面に応じた、レビューの視点について

Ostrandの4つのビュー

ユーザービュー、仕様ビュー、設計・実装ビュー、バグビュー



All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

付録1－2　場面に応じた、レビューの視点について

発想法・目的 / ビュー	定型連想 抜け・誤り防止	エラー推測 リスク予測	自由発想 新たな気づき
ユーザービュー	○	△	◎
仕様ビュー	◎	○	○
設計・実装ビュー	◎	○	×
バグビュー	○	◎	△

All Rights Reserved, Copyright © 2010, Hitachi Systems & Services, Ltd.

付録2 レビューの実施種類

代表的なレビュー実施方法

レビュー方法	概要	利点	難点
アドホック・リーディング	特にやり方を決めず、レビュアーの勘や経験で内容を考察する	準備は不要	レビュアーのスキルに依存する
チェックリスト・ベース・リーディング	チェックリストの内容に従い、内容を確認する	精度の高いレビューを実施可能	対象物に適合したチェックリストが必要
シナリオ・ベース・リーディング	予め定めたシチュエーションや、不具合／問題発生の可能性で検討する	勘や経験に依存する部分を一般化しやすい	対象物に適合したシナリオを準備する必要があり、事前準備が大変

いずれにせよ、共通的なポイント

対象物を見る「レビューアの視点(眼)」が、重要。

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

付録3 レビューツールの一例

レビューツール	概要
ブレーンストーミング	テーマに基づく、複数人数での自由発想により、思いつきや気になる事項を挙げていく。
5W1H ＋ 1D	Why,Whatなどの質問で詳細を具体化する。 さらに、Do(動作)を検討することで、より具体化される。
テンプレート／ワークシート	品質特性や、過去事例など、テンプレートになるものを活用する。
チェックリスト	想定する出来事や過去の経験を一覧化しておき、対比させる
マインドマップ	連想するキーワードを増やす、メモ技術
FMEA/FTA	不具合の可能性から仕様を検討する

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.

付録4 レビュー技術を磨くのに有効なコミュニティのご紹介

- **TEF**
  （Testing Engineer's Forum: ソフトウェアテスト技術者交流会）
  http://www.swtest.jp/wiki/index.php?swtest.jp/wiki/forum

- **SQiPコミュニティ（日科技連：Software Quality Profession）**
  http://www.juse.or.jp/software/sqip_community.html

- **ソフトウェアレビュー／インスペクションを議論するコミュニティ**
  （奈良先端科学技術大学院大学 情報科学研究科 ）
  http://se.naist.jp/html/review/ml.html

All Rights Reserved. Copyright © 2010, Hitachi Systems & Services, Ltd.