

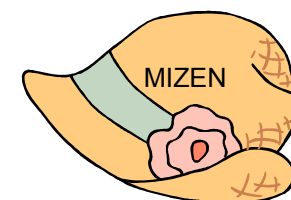
上流設計工程における 未然防止プロセスの実用化に向けて －不具合モード発想力を高める秘訣－

2013年1月30日
東芝ソシオシステムズ(株)

大谷和夫 久米智己子

塩谷和夫 渥美武雄

(株)東芝 ソフトウェアFMEA分科会



会社概要

- ◆会社名 東芝ソシオシステムズ株式会社
- ◆設立 1982年4月1日
- ◆人員 482名(2012年10月現在)
- ◆事業内容 東芝セキュリティ・自動化システムの設計・製造
自動化機器システム
セキュリティ関連システム
ICカードシステム
アプリケーション・ソフトウェアの開発
- ◆事業所 本社・設計部門 東芝小向工場内(川崎)
製造部門 秋田事業所

主力製品



海外向け選別取り揃え押印機



新幹線自動改札機



海外向け銀行券鑑査機



ETCシステム



定期券発行機

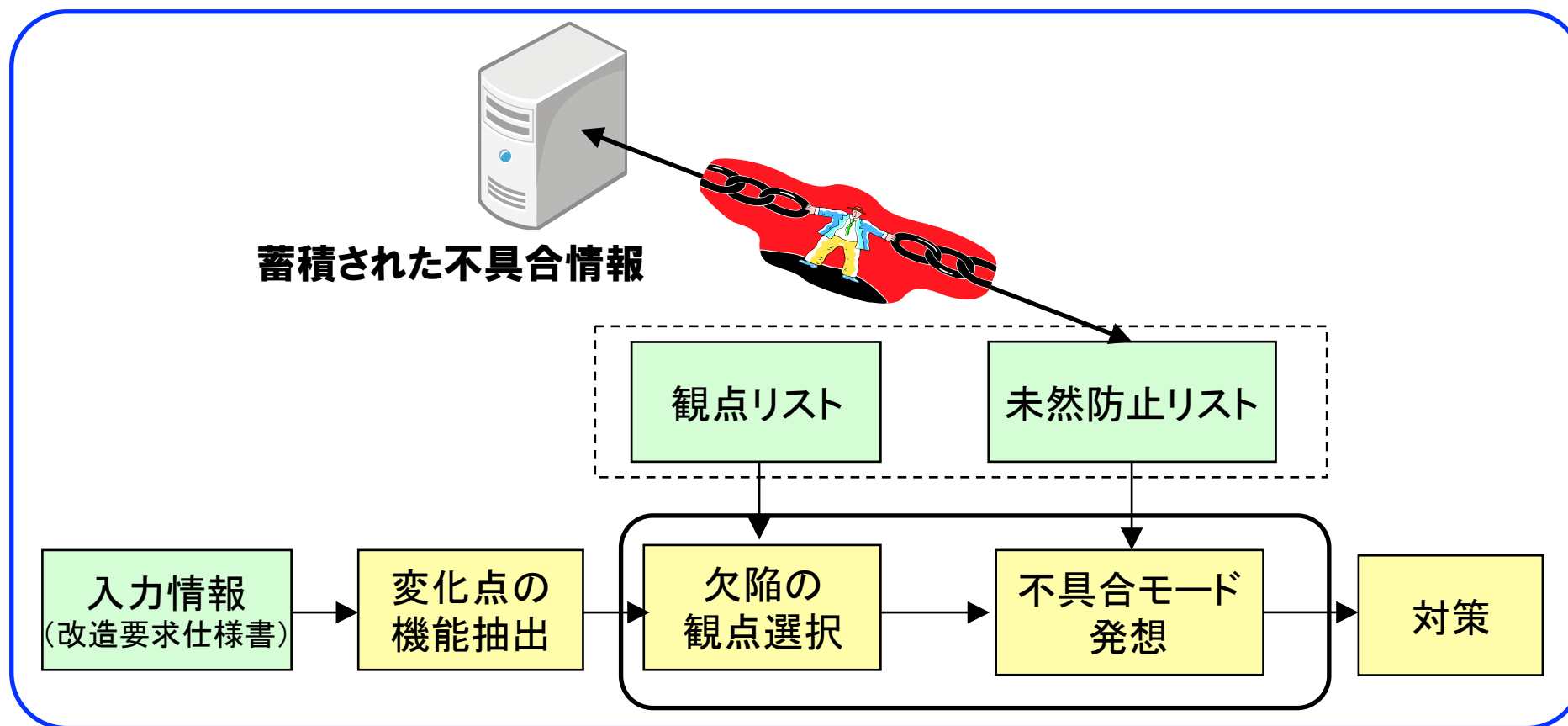
Agenda

1. 昨年JaSST' 12東京での提案(おさらい)
2. 未然防止実用化に向けての課題
 - 未然防止対象範囲の絞り込み
 - 不具合モード発想手順の定型化
 - 発想力の向上
3. パイロットプロジェクトへの展開
4. 特に工夫した点
5. まとめ

JaSST' 12提案:未然防止プロセスの確立

■ 不具合モードの発想

- 欠陥の観点リストと過去の不具合情報をベースにした未然防止リストを使って、不具合モードを発想



当時の問題点

■ 不具合モードの発想が発散

- 未然防止プロセスを検証した結果
実際の不具合モードと一致した割合が**10～38%**
と低く、議論が発散していた
- 原因
発想する対象の**範囲が広すぎる**
欠陥の**観点リスト**がうまく使えていない
参加者の発想するための**スキル**がバラつく

■ 観点リストの改良

- 観点リストを『**何が、どうして、どうなる**』タイプに
分解し、発想しやすくした
- この観点リストは適用できず提案で終わっていた

Agenda

1. 昨年JaSST' 12東京での提案(おさらい)
- 2. 未然防止実用化に向けての課題**
 - 未然防止対象範囲の絞り込み
 - 不具合モード発想手順の定型化
 - 発想力の向上
3. パイロットプロジェクトへの展開
4. 特に工夫した点
5. まとめ

未然防止実用化に向けての課題

■ 課題1：未然防止対象範囲の絞り込み

- － 入力となる機能や制約条件を整理

■ 課題2：不具合モード発想手順の定型化

- － 改良型観点リストを使い、発想手順パターン構築

■ 課題3：発想力の向上

- － 発想力を高める教育の実施

施策1 未然防止対象範囲の絞り込み

■ 処理モデルと想定される設計ロジック採用

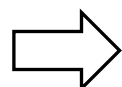
要求仕様

X年間未使用のICカードを入退場ゲートに
翳したとき、**取扱い停止**にする

処理モデル

入力(対象)

ICカードの
前回利用日
データ

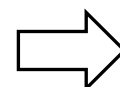


別入力:

端末の有効／無効設定

他の制約条件

未使用期間
チェック処理



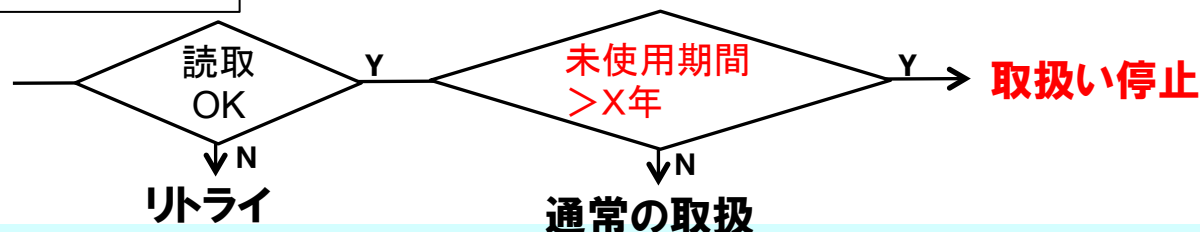
出力(振る舞い)

取扱い停止

条件:**X年間未使用**

想定される設計ロジック

前回利用日

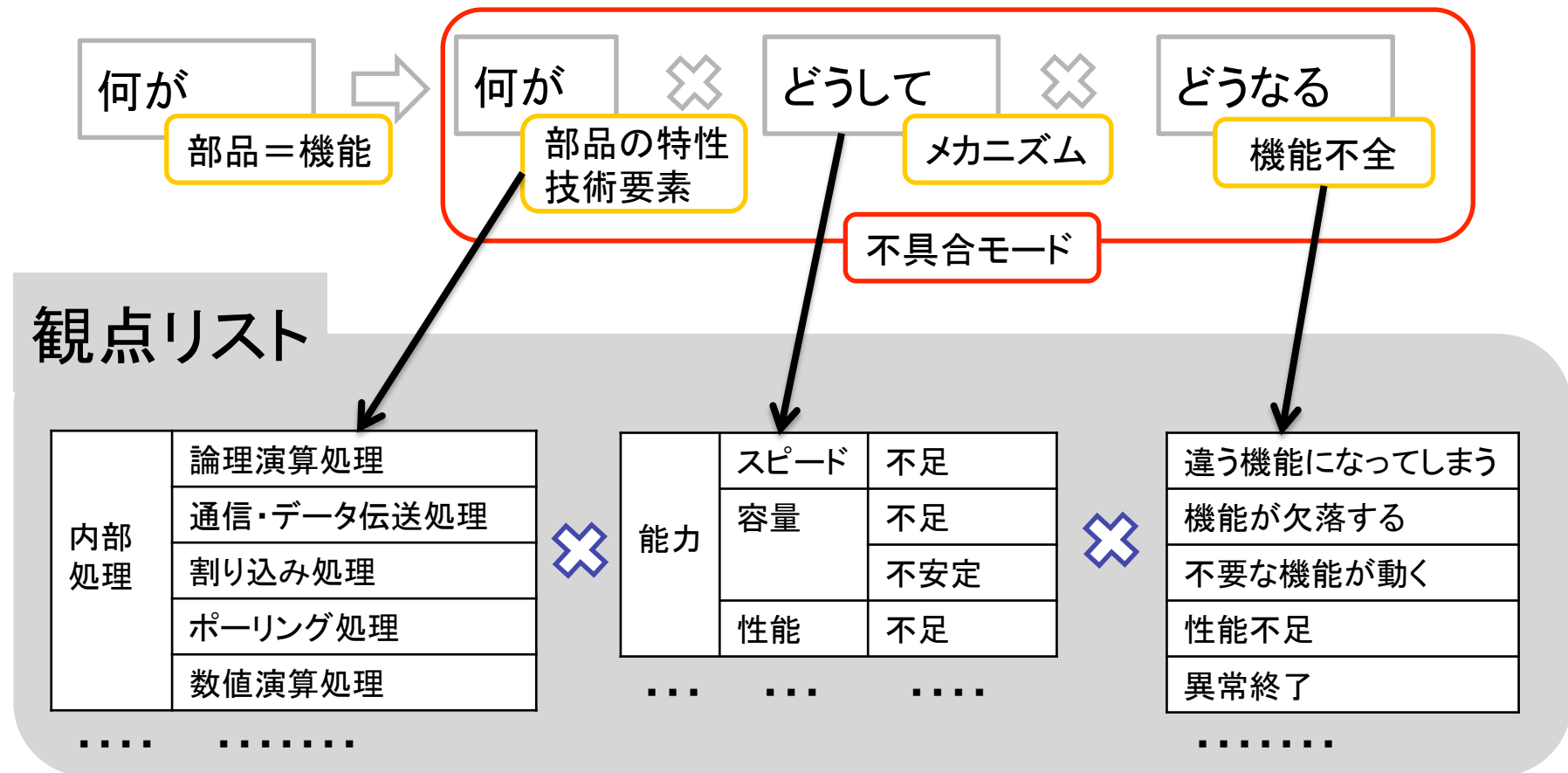


未然防止実用化に向けての課題

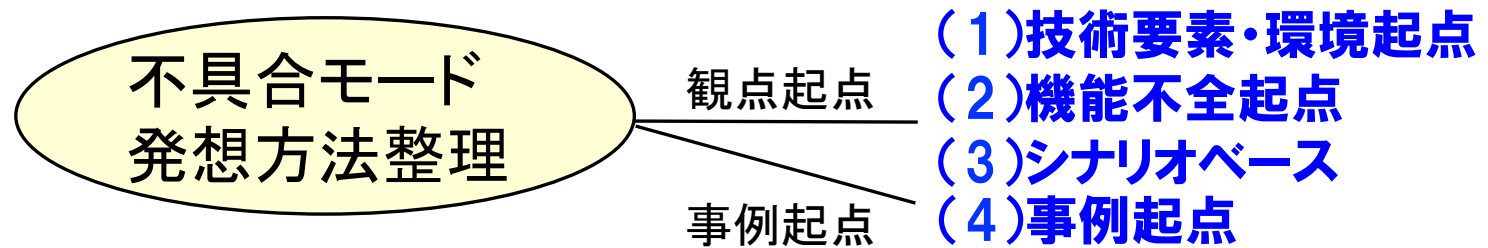
- 課題1：未然防止対象範囲の絞り込み
 - － 入力となる機能や制約条件を整理
- 課題2：不具合モード発想手順の定型化
 - － 改良型観点リストを使い、発想手順パターン構築
- 課題3：発想力の向上
 - － 発想力を高める教育の実施

施策2 不具合モード発想手順の定型化

■ 改良型観点リストとは

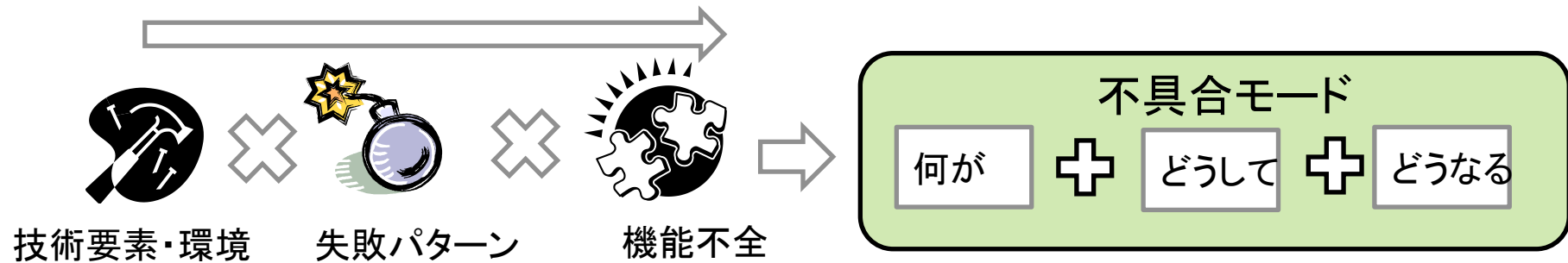


■ 不具合発想の起点を定型化



- 観点を利用して、手順に従って不具合モードを発想
 - 発想のパターンは大きく4つ
 - 手順にとらわれず、観点や事例を自由に組み合わせて発想することも必要
 - 一つの観点を固定し、別の観点の組み合わせを入れ替えてみる
 - 「何が」や「どうして」を複数組み合わせてみる
- 要求仕様 事例「X年間未使用カードを取扱い停止にする」を
(1)～(3)の手順で不具合発想

(1)不具合モード発想手順（技術要素・環境起点）

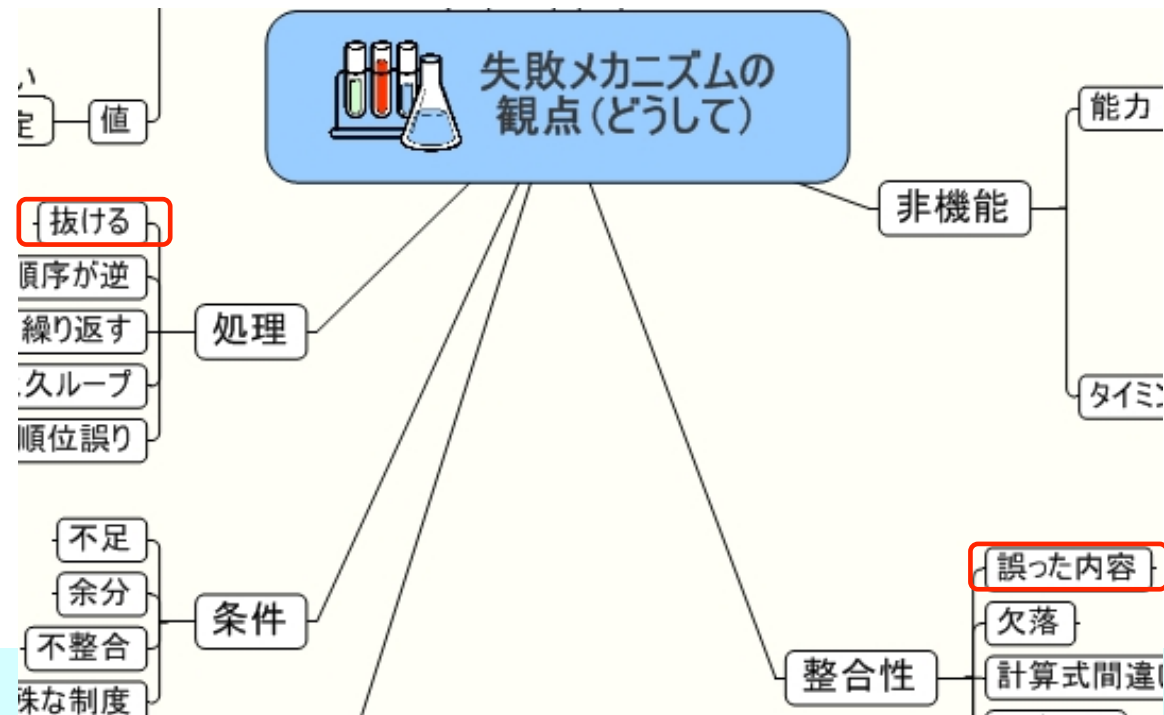
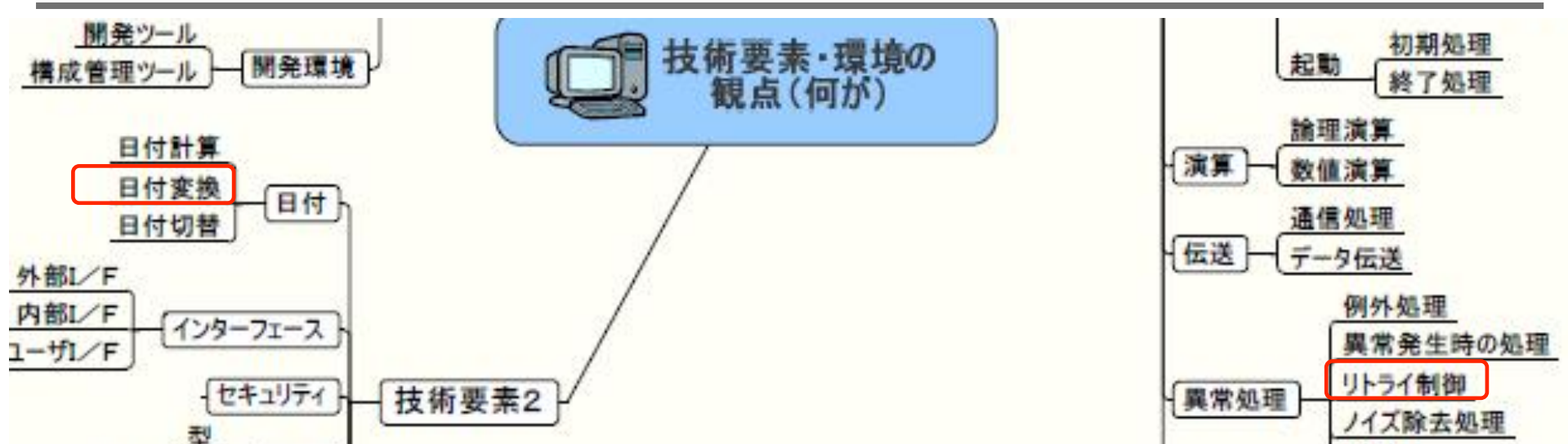


- ・ 技術要素・環境起点(何が)
⇒ 失敗メカニズム(どうして)

何が と どうして の観点を組合せて発想

⇒ 機能不全(どうなる)

観点リスト



(1)不具合モード発想手順（技術要素・環境起点）

・技術要素・環境起点(何が)

⇒ 失敗メカニズム(どうして) ⇒ 機能不全(どうなる)

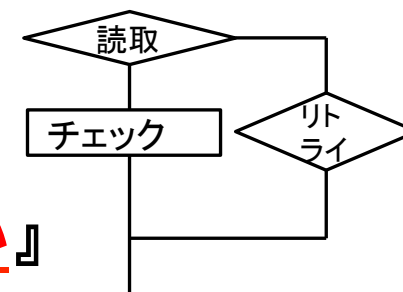
① 日付・変換処理(何が) ⇒ 整合性・誤った内容(どうして)

『前回利用日を 西暦年4桁を2桁として取扱い、
未使用期間を正しく計算できず、
取扱い停止とすべき日を間違える』

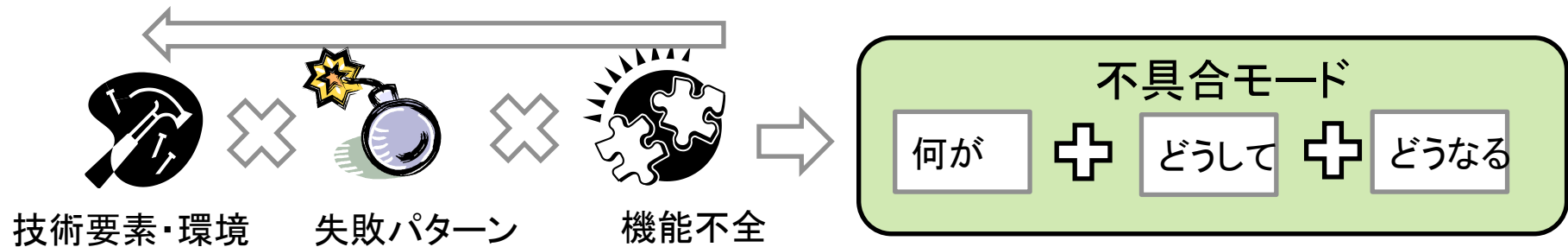
2012年07月04日
↓
20年12月07日

② リトライ制御(何が) ⇒ 処理・抜ける(どうして)

『カード読取りエラーのリトライ読取り時、
未使用期間チェック処理が抜け、
X年間未使用でも取り扱い停止にならない』



(2)不具合モード発想手順（機能不全起点）



・機能不全起点(どうなる)

- ① 機能欠落
- ② 不要な機能
- ③ 性能不足
- ④ 異常終了
- ⑤ 違う機能

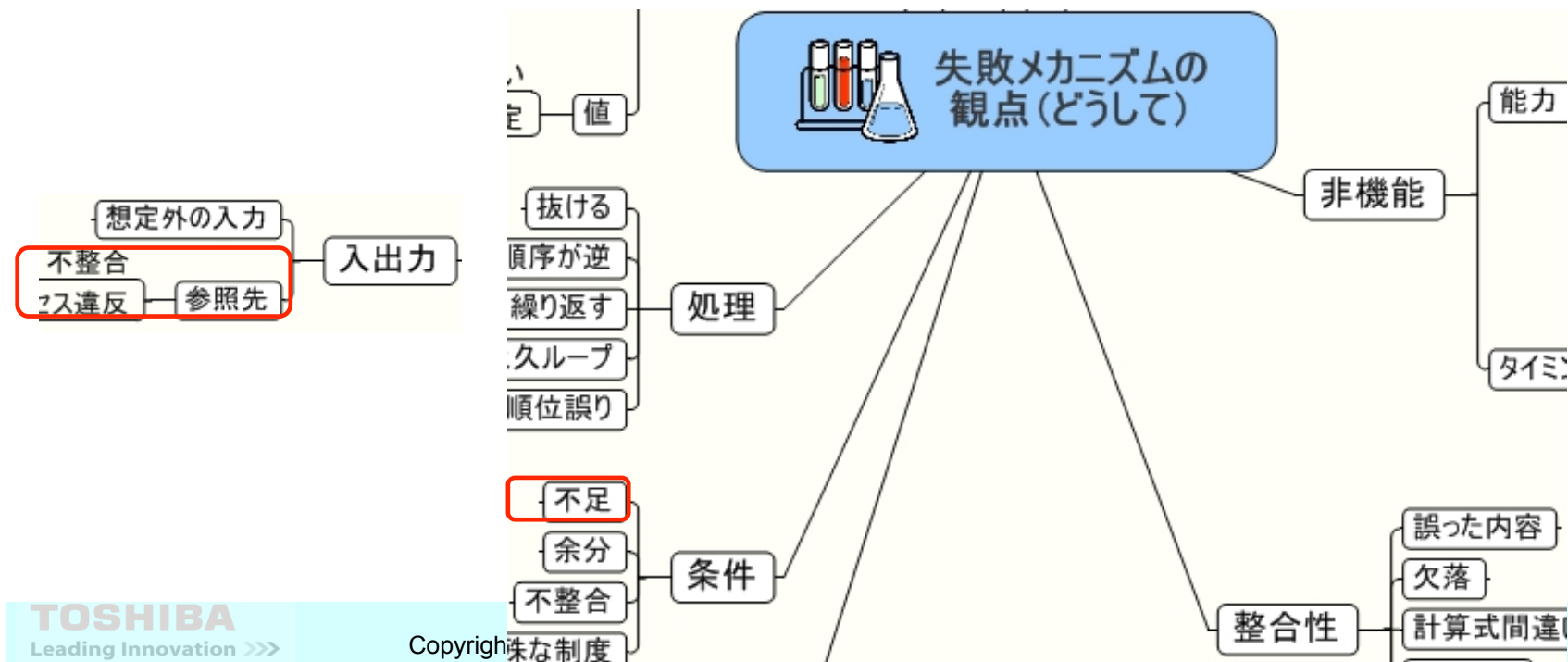
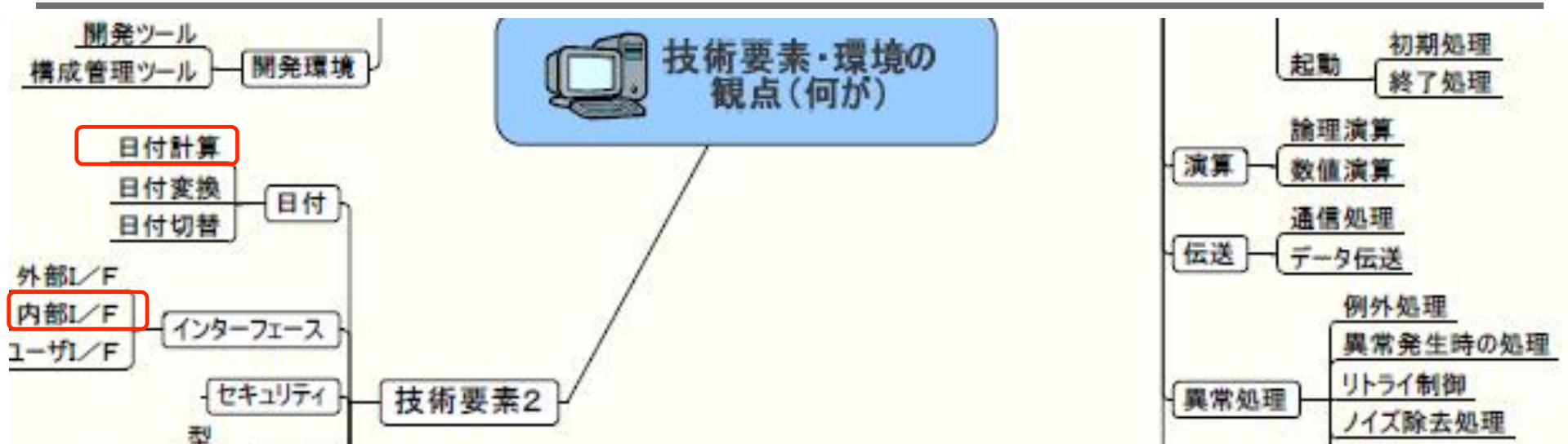
⇒ 何が ⇒ どうして を追加して発想

(2)不具合モード発想手順（機能不全起点）

・機能不全起点(どうなる)

- ① 機能欠落 取扱い停止にならない *
- ② 不要な機能 別条件から取扱い停止にすべきでない
- ③ 性能不足 取扱い停止になるまで時間がかかる
- ④ 異常終了 システムが停止してしまう *
- ⑤ 違う機能 X年より前(または後)に取扱い停止

観点リスト



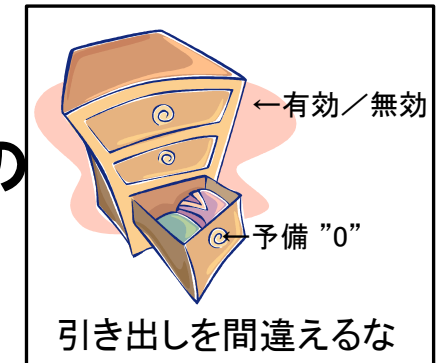
(2)不具合モード発想手順（機能不全起点）

・ 機能不全起点(どうなる) ⇒ 何が ⇒ どうして

① 取扱い停止にならない

内部I/F ⇒ 参照先・不整合

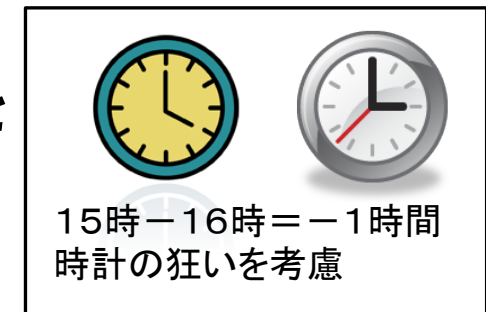
『未使用期間チェック有効／無効パラメータの
参照先アドレスを間違えて常に無効となり
取扱い停止にならない』



④ システムが停止してしまう

日付計算 ⇒ 条件・不足

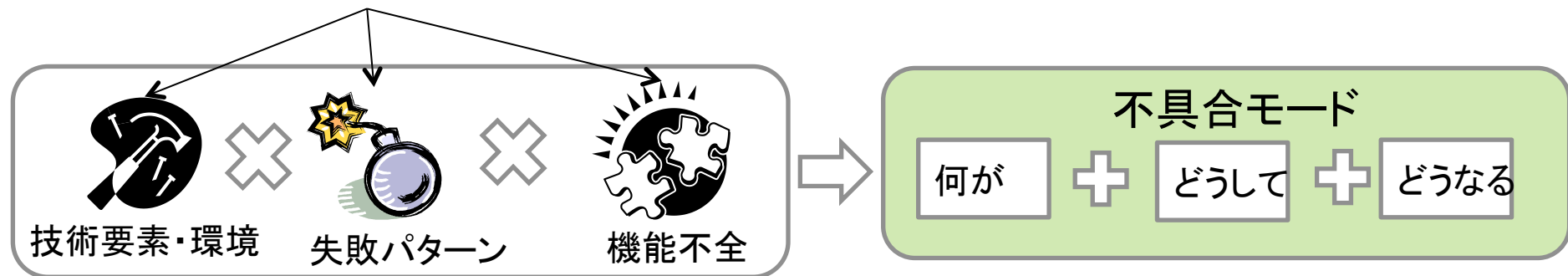
『未使用期間の計算結果が負となる条件を
考慮していなかったため
処理がアボートし停止してしまう』



(3)不具合モード発想手順（シナリオベース）

シナリオライティング(仕様段階)用観点	
利用環境	利用者
	利用手順・順番
	利用場所
	頻度
	接続
	自然環境
	非常事態

実際の利用形態を考える



- ・ シナリオベース起点
実際の利用形態や運用を考える

何が ⇒ どうして ⇒ どうなる 形式で記述

(3)不具合モード発想手順（シナリオベース）

- ・ シナリオベース起点（実際の利用形態を考える）

何が ⇒ どうして ⇒ どうなる

- － 条件組合せ

『残額0円など他の条件と組合わさった時、
残額不足エラーが優先され
取扱い停止にならない』



- － 特異日に利用

『前回を閏年の2/29に使用し、X年後に2/29がなく
閏年を考慮した期間計算ができず
取扱い停止にならない』



未然防止実用化に向けての課題

- 課題1：未然防止対象範囲の絞り込み
 - － 入力となる機能や制約条件を整理
- 課題2：不具合モード発想手順の定型化
 - － 改良型観点リストを使い、発想手順パターン構築
- 課題3：発想力の向上
 - － 発想力を高める教育の実施

施策3 発想力の向上

■ 発想力を高める教育

- － 技術者の経験度合いにより**発想力のバラツキ**があるため、発想力高める教育を実施
- － カリキュラム
再発防止と未然防止の違い、FMEA実施手順、
不具合モード抽出**事例の紹介**、演習
- － 講義1時間＋演習2時間

■ 演習内容

- － 課題の機能に対し、4種類の発想手順を使い、
実際に不具合モードを発想してもらう

施策3 発想力の向上

■ 受講対象者

- 2012年7月～11月に教育を4回開催し、延べ42名が受講
- 未然防止パイロツティングを実施するプロジェクトは必須教育とした

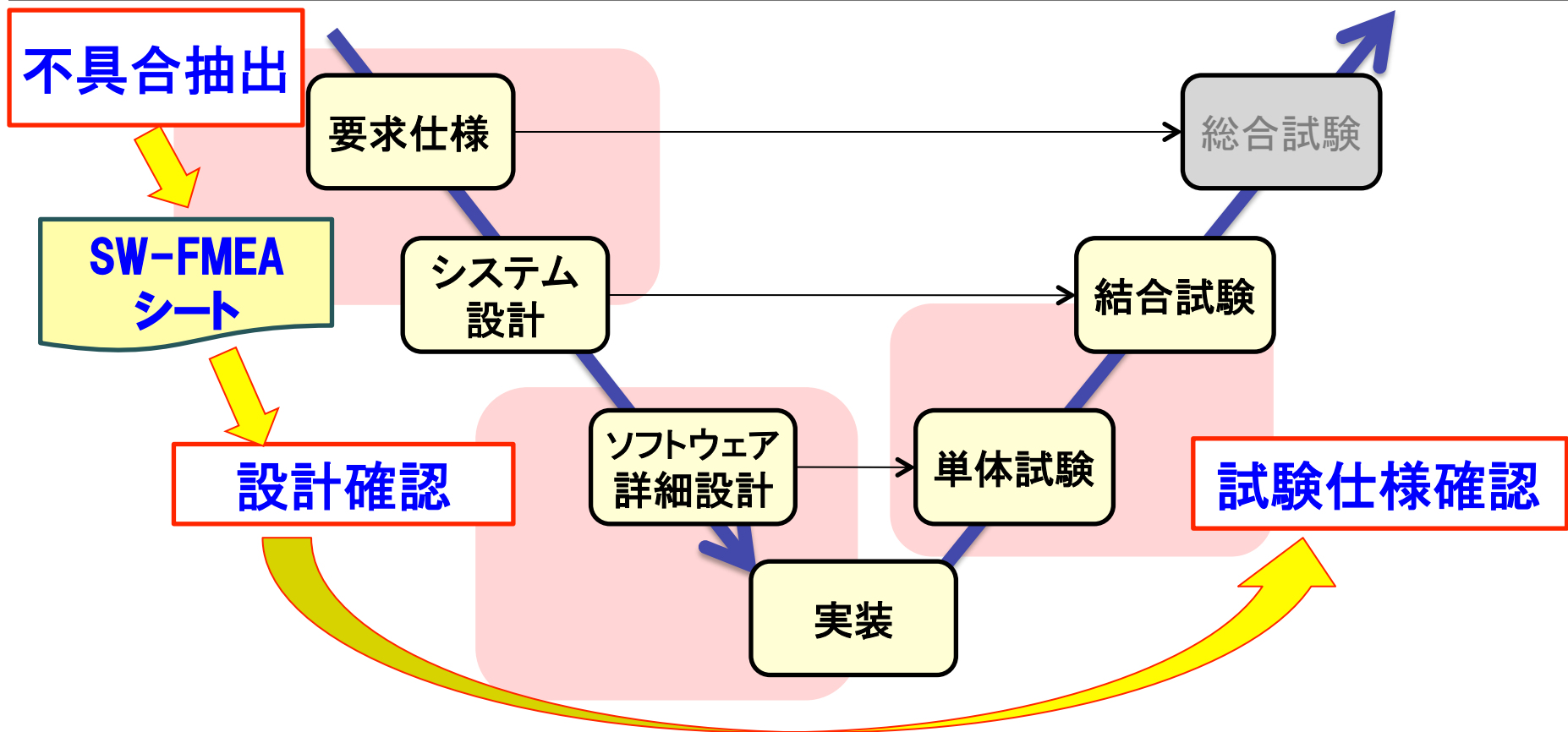
■ 受講者のアンケート結果

- 事例紹介があり、未然防止の適用方法、効果がわかりやすい
- 不具合の発想は難しいと感じた
- 発想手順が4つあるので、得意な手順を使って発想し、繰り返し行うことにより身に付くと思う

Agenda

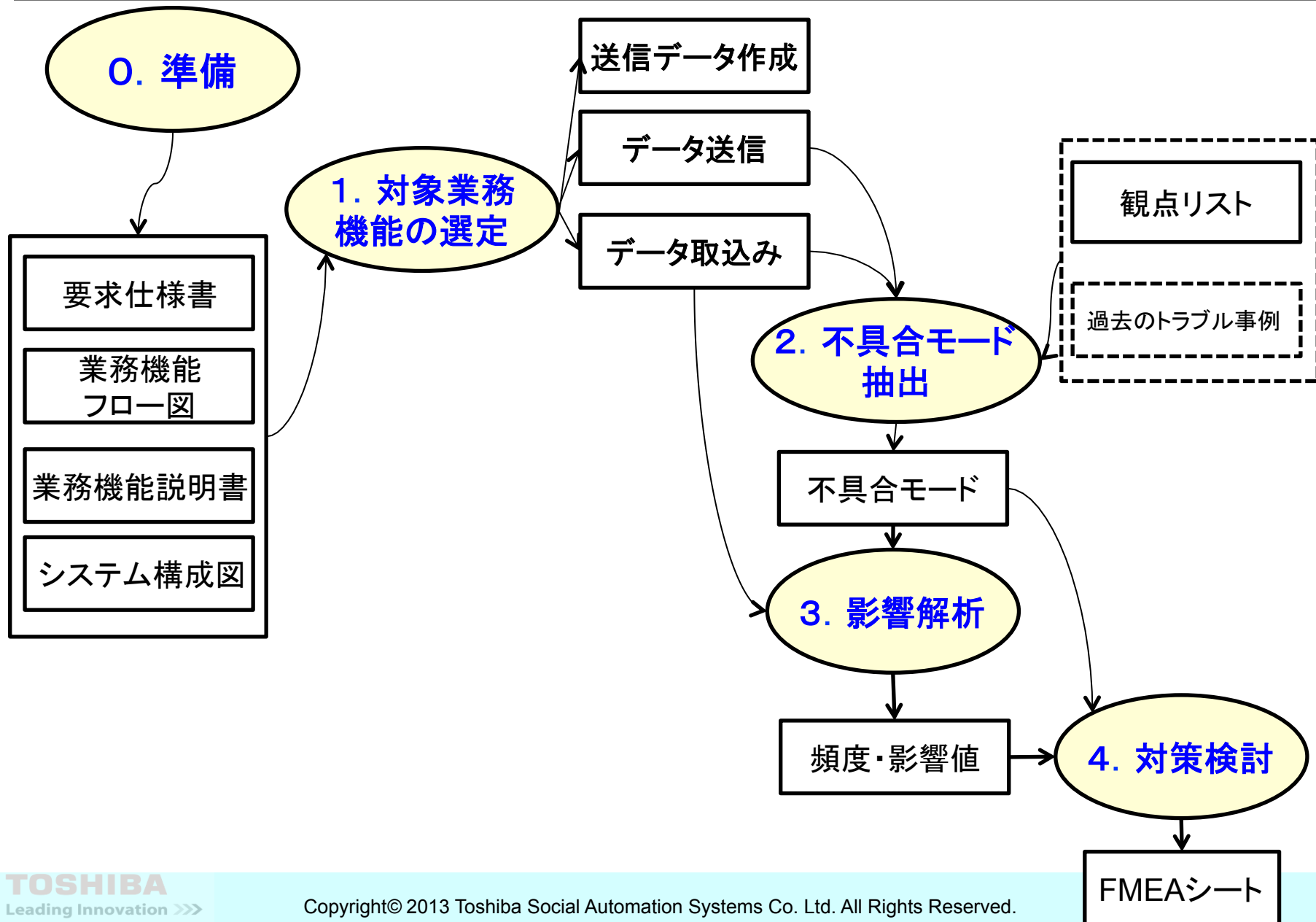
1. 昨年JaSST' 12東京での提案(おさらい)
2. 未然防止実用化に向けての課題
 - 未然防止対象範囲の絞り込み
 - 不具合モード発想手順の定型化
 - 発想力の向上
3. **パイロットプロジェクトへの展開**
4. 特に工夫した点
5. まとめ

“経理データシステム開発”への適用



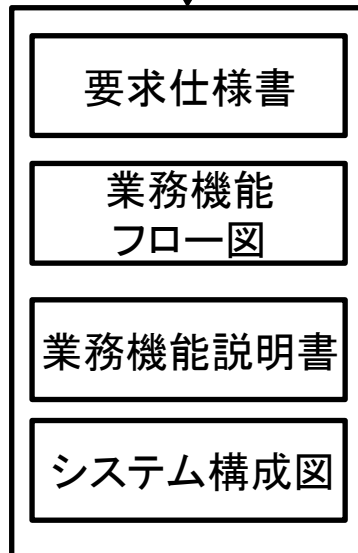
- ・不具合モードを抽出した後、発生頻度と影響度から対策検討
→「**SW-FMEAシート**」に整理
- ・設計着手前に対策を再確認
- ・試験仕様書作成前に対策を再確認

未然防止プロセス実施手順



準備

0. 準備



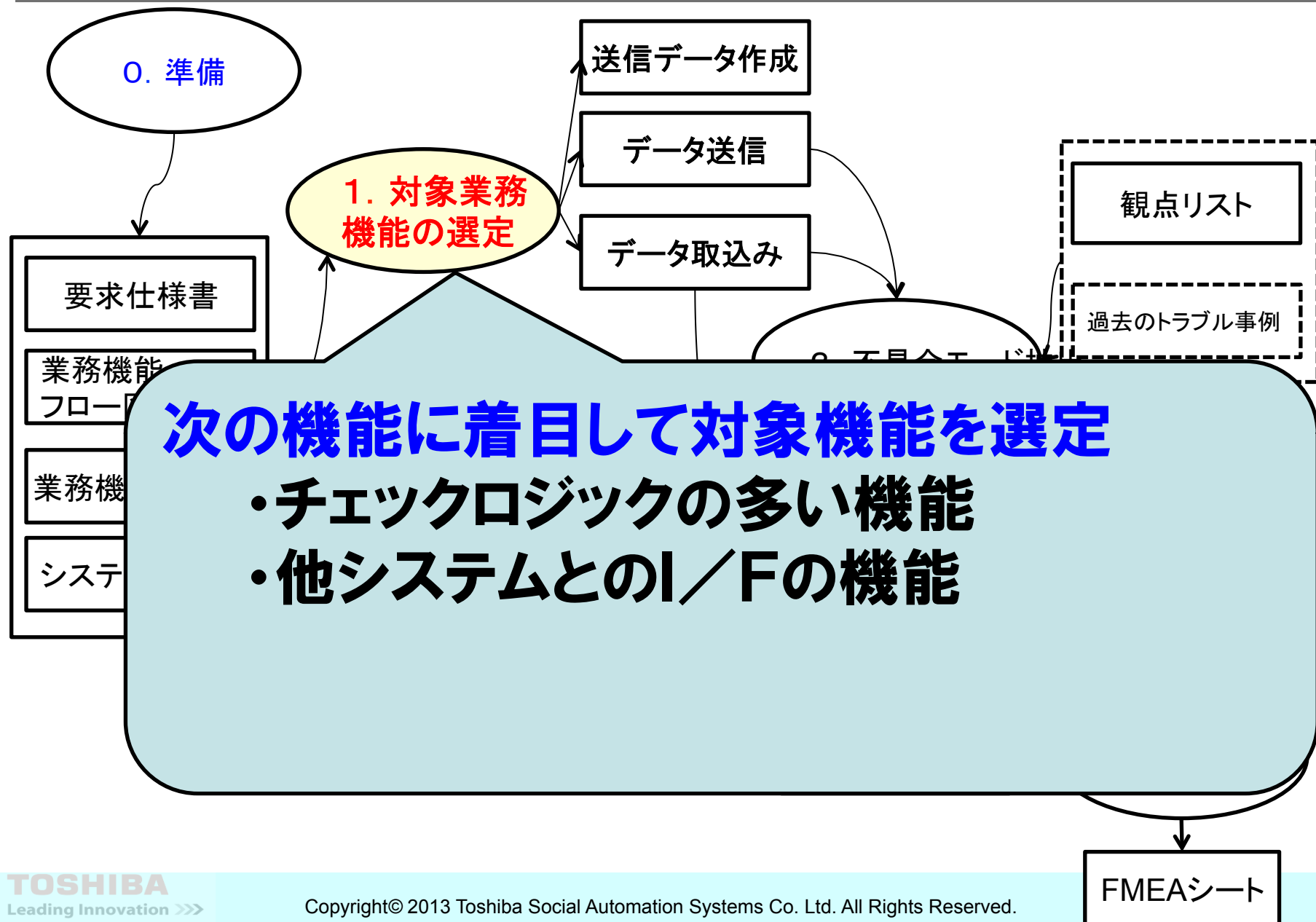
①メンバ選定

- PJの関係者(設計者 & レビューア)
- PJメンバ以外
- ファシリテータ
- アドバイザー

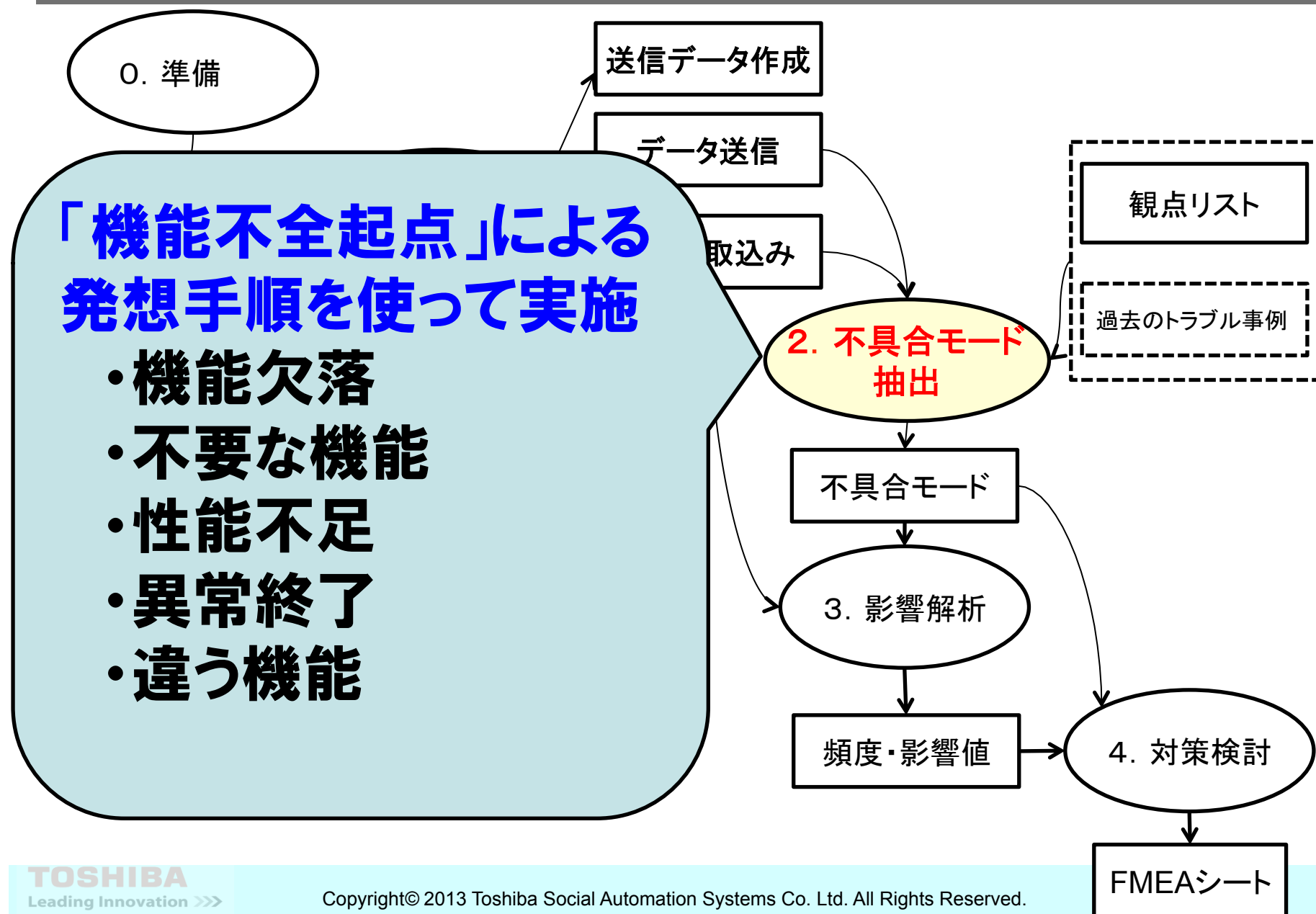
②資料準備

- システムの概要が分かるもの
- 要求仕様書
- 業務フローとシステムの振る舞いが分かるもの

対象業務・機能の選択



不具合モードの抽出



不具合モードの抽出事例

機能不全起点(どうなる) ⇒ 何が ⇒ どうして

【CSVファイルの読み込み】

(1)機能欠落

CSVファイルが読み込めない

- ファイル容量が大きい
- 異なるフォーマットのファイルを指定した
- 区切り文字(タブ／カンマ)の間違い
- ヘッダー有無判断の間違い(ヘッダー無なのに有で読んだ)

(2)不要な機能

取込まなくてよいファイルを取込む

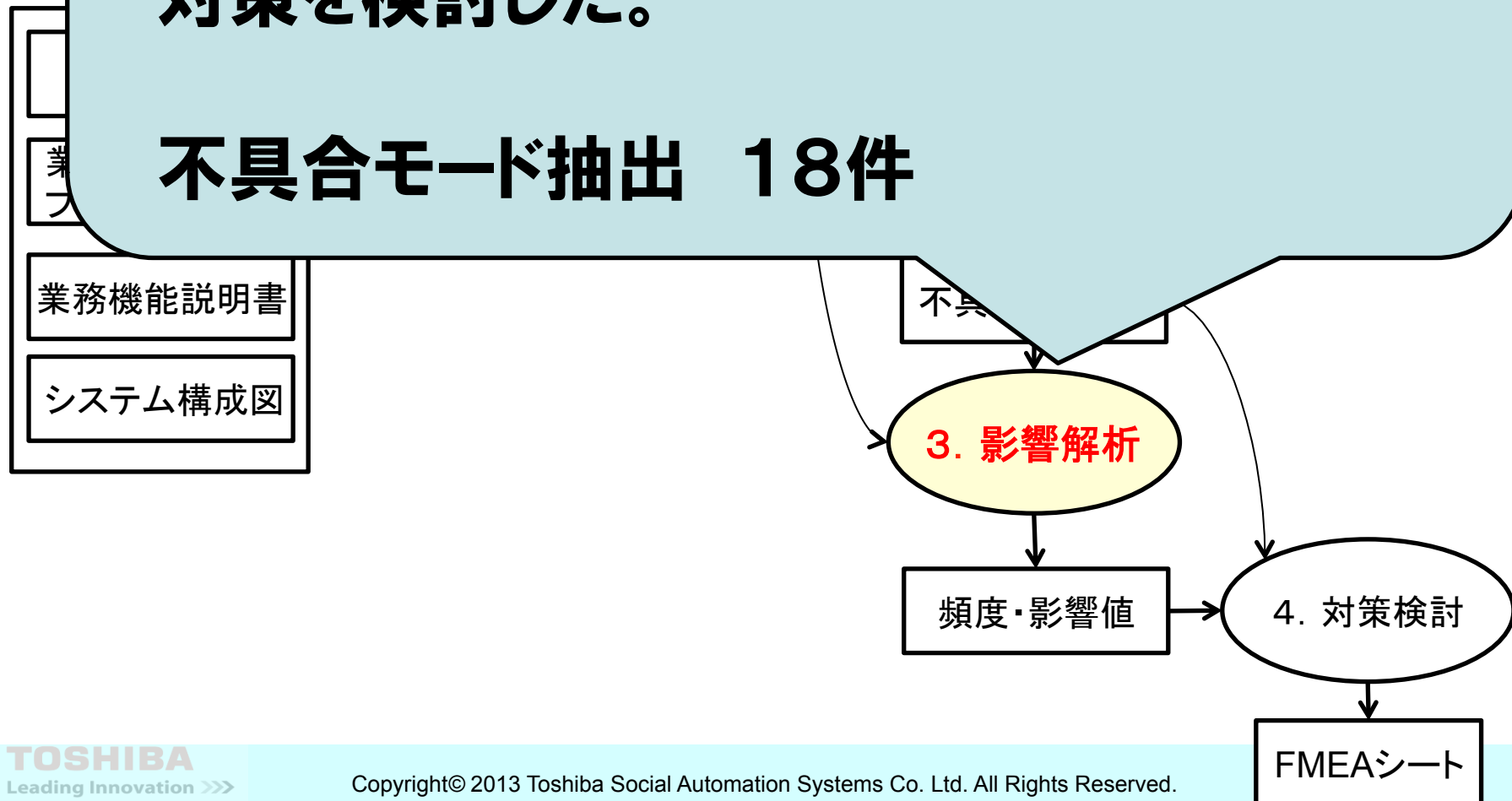
- 同じファイルの読み込みが何度でも出来る(データの重複)
- 異なるフォーマットのファイルを指定した

影響解析・対策検討

影響解析と対策検討

発生頻度と発生時の影響度を分析して、
対策を検討した。

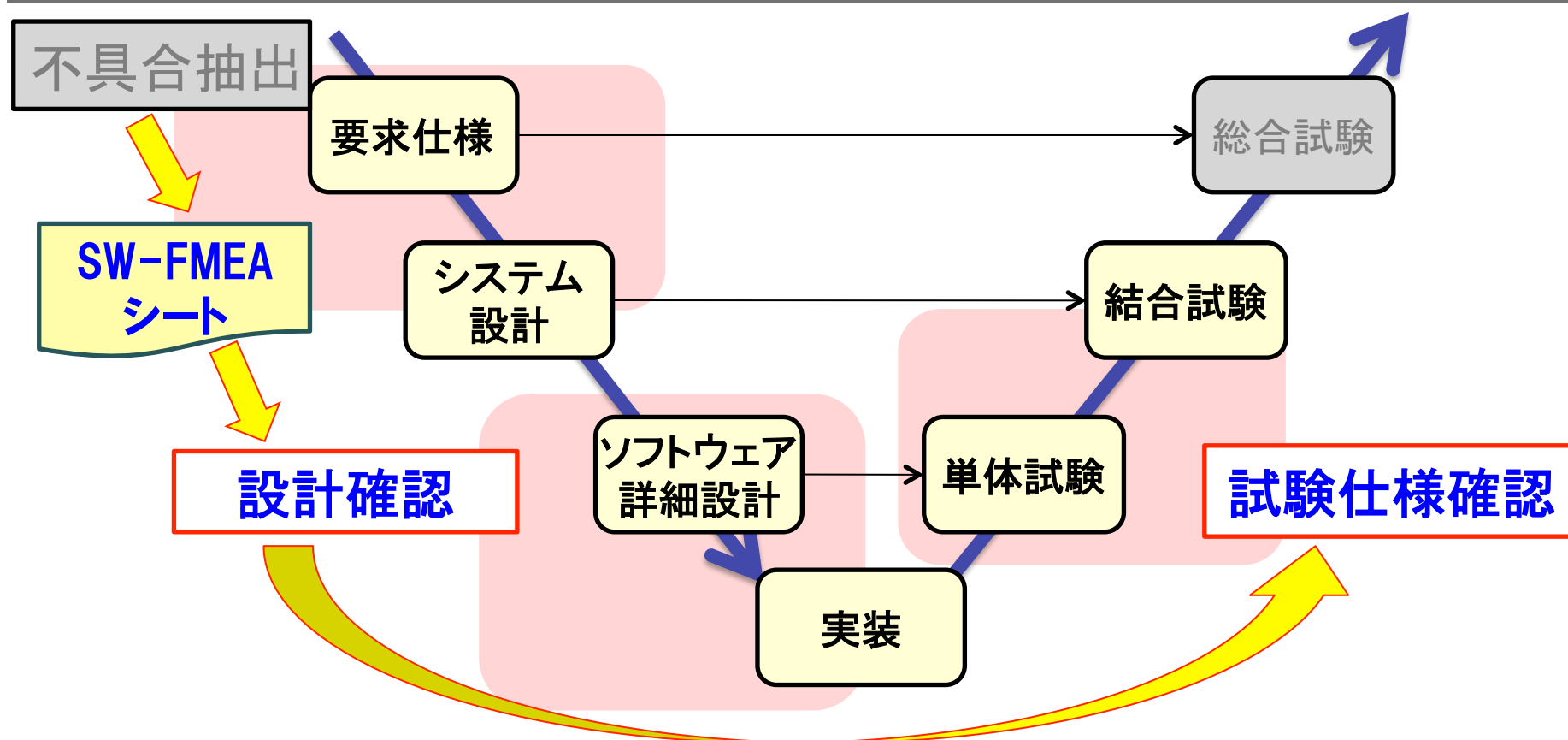
不具合モード抽出 18件



SW－FMEA事例

N O	機能／モジュール／処理	不具合モード			発生頻度	影響度	設計考慮した内容
		何が	どうして	どうなる			
1	送信データ作成	CSVファイルのデータ	データ数が多い	データが読み込まれない (時間がかかる)	極小 EXCELの制限行数は104万8000行。 実データはこんなに発生し得ない。	中 仮に発生したら(オペミス)・・・ データを最後まで読み込めない。 →IEが固まる	CSVを読み込む時に件数チェックして、閾値を超えたらエラーメッセージを表示して、DBからも消す。 →プログラム設計書に記載
2		区切文字	間違えた	データが読み込めない	小 読み込むCSVファイルはデータ連携システムで出力したファイルが多いので発生は少ない	小 CSVファイルを作り直す	区切文字はタブよりカンマの方が見やすいのでカンマにする。 →プログラム設計書に記載 項目数をチェックしてエラーメッセージを表示する →プログラム設計書に記載
3		ヘッダーの有無	有を無と間違えた	1行目のデータが取り込まれない	小 プログラムのバグ CSVファイル編集時のオペミス	小 CSVファイルを作り直す	CSVで出力する時に1行目にヘッダーマークをつけておき、なければヘッダー無し のエラーにする →プログラム設計書に記載
4		間違ったファイル	読み込んだ	データベースへ書き込み時にエラーとなる	小 ファイル読み込み時のオペミス	大 データベースに書き込まれてしまうデータもある	特定のCSVファイルしか選択できないようにしておく 項目数のチェックとヘッダーマークのチェックで判断する。 書き込みが全件完了しなかったらロールバックする仕組みを入れる。 →プログラム設計書に記載

SW-FMEAシートの活用(今後)



- ・不具合を抽出した後、発生頻度と影響度から対策検討
→「SW-FMEAシート」に整理
- ・設計着手前に対策を再確認
- ・試験仕様書作成前に対策を再確認

参加者の感想①

Q. 未然防止のやり方は理解できましたか？

- ・○ ほぼできた(全員)

Q. 開発メンバによる不具合の発想は有効ですか？

- ・○ 同じ観点リストを見ていても、メンバーによって
発想の観点が変わるので良い
- ・△ 有効だが**効果が出るまでに時間**がかかりそう
- ・△ 有効と思うが**工数とのバランス**が難しい

Q. 発想した不具合は的を射ることができましたか？

- ・○ 的を射ることができたが**取捨選択**が必要
- ・△ すでに**考慮されているもの**が多かった

参加者の感想②

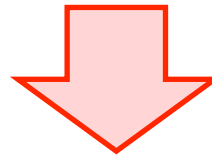
Q. 未然防止の改善案は？

- ・実践を繰り返すことで**不具合発想に慣れる**とよい
- ・システム設計レベル、基本設計レベル等、**工程毎に粒度を変えて発想**しても良い
- ・**観点リスト**を不具合モード発想の視点だけではなく**設計検討の場**で使うと良い
- ・受講した発想法を一通りやる時間のやり繰りをつける
と良い

問題点と解決策

■問題点

- ・効果的な不具合発想が難しそう
- ・時間が掛かり、費用対効果が低いのでは？
- ・当然考慮すると思われる不具合の発想が多い



■解決策

- ・設計検討会の中で観点リストを使って発想する
→ 時間が掛からない、お手軽にできる
- ・業務の有識者を交える、顧客になりきって発想する
→ 運用ベースでの不具合発想

Agenda

1. 昨年JaSST' 12東京での提案(おさらい)
2. 未然防止実用化に向けての課題
 - 未然防止対象範囲の絞り込み
 - 不具合モード発想手順の定型化
 - 発想力の向上
3. パイロットプロジェクトへの展開
- 4. 特に工夫した点**
- 5. まとめ**

特に工夫した点

■ 不具合モード発想力の向上

- 不具合モードの**発想手順を4種類**用意し、教育により技術者に修得してもらい、発想力を一気に向上

■ パイロットプロジェクトにて効果を確認

- 実際の開発製品に未然防止を適用し、設計工程に不具合となりそうな**気づきを与えられるか検証**

■ 重点課題としての取り組み

- ソフトウェア未然防止活動を**会社TOPの重点課題**とし、従業員全員が一丸となって取り組む

まとめ

■ まとめ

- ソフトウェア未然防止の実用化に向けて不具合発想手順を決め、パイロットティングを繰り返すことにより、本格展開の準備ができた



■ 今後の展開

- 未然防止プロセスの適用事例をさらに増やしPDCAをまわしながら、品質向上と生産性向上に貢献していく
- 過去事例ベースの不具合モード発想は、ハードウェアのFMEAに利用されているSSM(ストレス・ストレングス・モデル)の活用を具体化していく

参考文献

- [1] 夏目珠規子、小島昌一、村山知寛：
“ソフトウェア開発におけるFMEAの適用可能性検討”、
第41回信頼性・保全性シンポジウム発表報文集、2011

- [2] 濱口哲也：“失敗学と創造学”、早稲田大学オープンカレッジ”、
2011

- [3] 田村泰彦：“トラブル未然防止のための知識の構造化
ーSSMによる設計・計画の質を高める知識マネジメント”、
日本規格協会、2008

- [4] 大谷和夫、塩谷和夫、宮本憲一、久米智己子、夏目珠規子：
“上流設計工程における未然防止プロセスの提案”、ソフトウェア
テストシンポジウムJaSST2012 Tokyo

ご清聴ありがとうございました

TOSHIBA
Leading Innovation >>>