TOSHIBA
Leading Innovation

JaSST'12 Tokyo

上流設計工程における未然防止プロセスの提案

－未然防止リストの活用と欠陥の発想－

東芝ソシオシステムズ(株)
大谷　和夫　　塩谷　和夫
宮本　憲一　　久米　智己子
(株)東芝 ソフトウェア技術センター
夏目　珠規子

2012年1月25日

TOSHIBA
Leading Innovation

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　493名（2011年7月現在）
◆事業内容　東芝製品の設計・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
　製造部門　秋田事業所

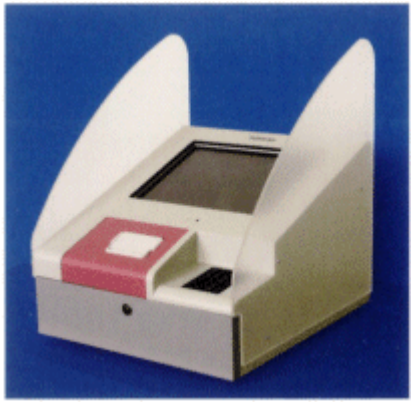
TOSHIBA

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



IC免許証情報確認端末

ETCシステム

定期券発行機

Agenda

- はじめに（背景と課題）
- 施策1　未然防止プロセスの確立
  - 観点リストから不具合モードの発想
  - 未然防止プロセスの検証と改善
- 施策2　未然防止リストの活用
  - 過去の不具合情報を知識化
- まとめ

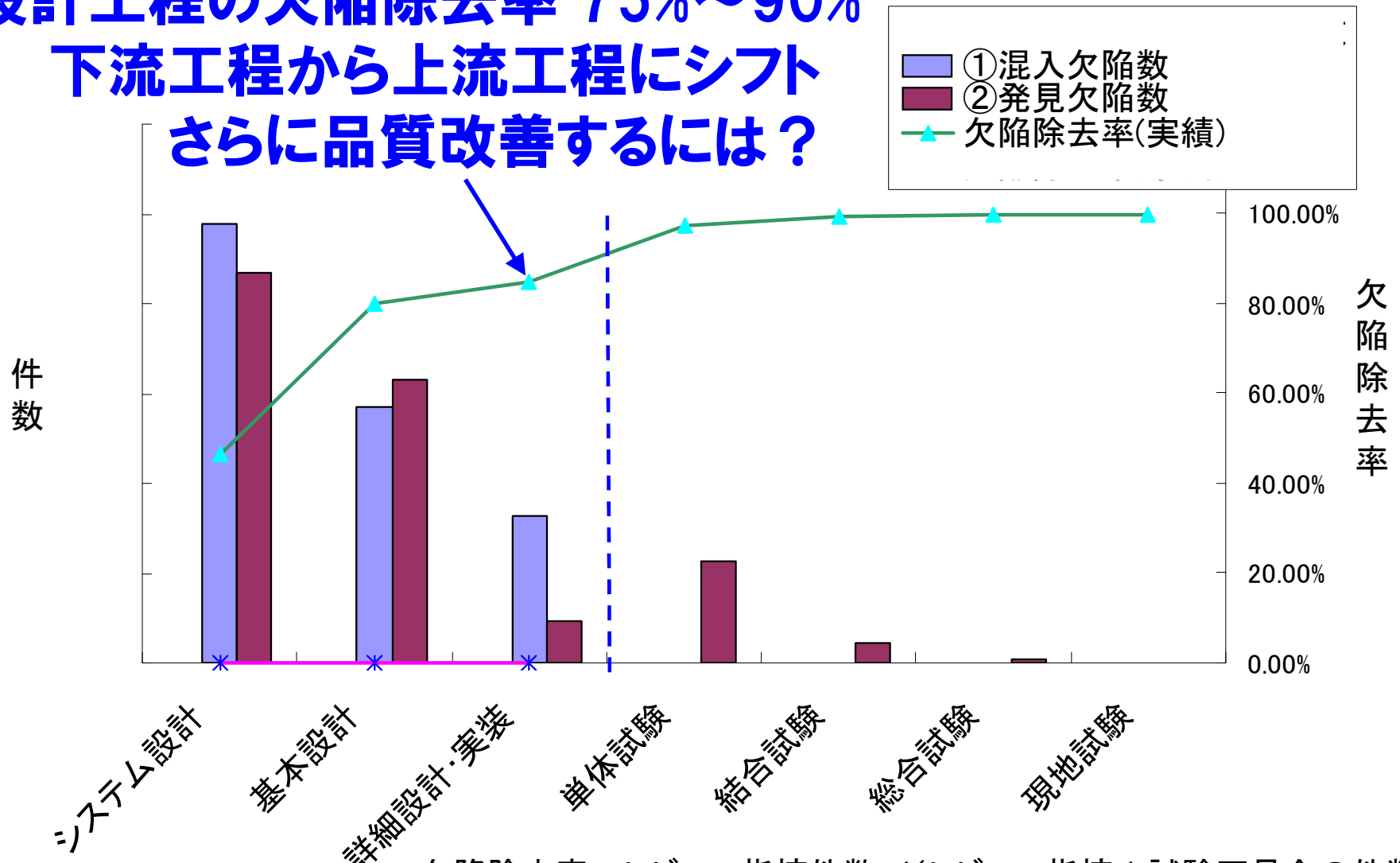
はじめに（背景）

ウォータフォール型開発プロセス

欠陥を摘出する取り組みが中心

はじめに（背景）

設計工程の欠陥除去率 75%～90%
下流工程から上流工程にシフト
さらに品質改善するには？



欠陥除去率＝レビュー指摘件数／（レビュー指摘＋試験不具合の件数）

はじめに（課題）

■ 課題１：設計工程で欠陥を混入させない仕組みの構築

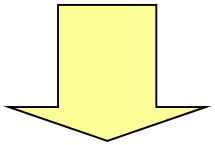
- 混入された欠陥の摘出と修正が後戻り作業
- 欠陥の事前予測が難しい

■ 課題２：過去に発生した不具合を類似した事象で再発させない

- 再発防止の適用範囲を広げた未然防止

はじめに（課題達成のための施策）

東芝グループでは、2010年よりソフトウェア
開発の未然防止にFMEAの適用を検討してきた



■ 課題達成のための施策

1．設計工程で欠陥を混入させない仕組みには
　　FMEAを利用した『未然防止プロセスの確立』

2．過去に発生した不具合の未然防止には
　　『未然防止リストの活用』

Agenda

- はじめに（背景と課題）
- **施策１　未然防止プロセスの確立**
  - **観点リストから不具合モードの発想**
  - **未然防止プロセスの検証と改善**
- 施策２　未然防止リストの活用
  - 過去の不具合情報を知識化
- まとめ

施策1　未然防止プロセスの確立

■ 欠陥の分類

– 欠陥の分類から観点リストとして整理

人による失敗

- 知識不足
- 判断ミス
- 注意不足
- ルール無視
- 検討不足

環境の故障や性能・容量不足

- ハードウェア故障
- OSとの相性
- メモリ不足

ソフトウェアの不具合

- 要求仕様化曖昧
- 論理の間違い
- タイミングずれ
- ・・・・・

設定・操作ミス

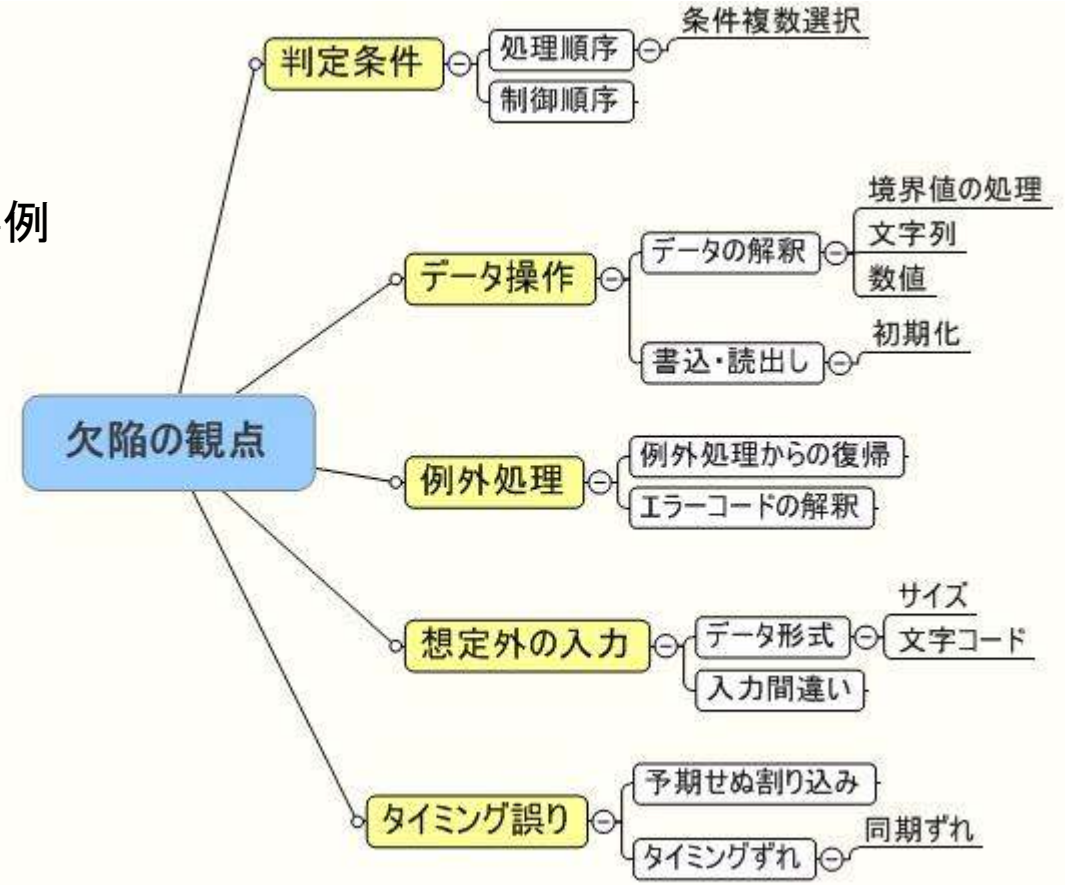
- システム環境の設定ミス
- パラメータ設定ミス
- 操作ミス
- 想定外の電源断から復帰

施策1　未然防止プロセスの確立

■ 欠陥の観点リスト

– ソフトウェア不具合を観点リストに整理

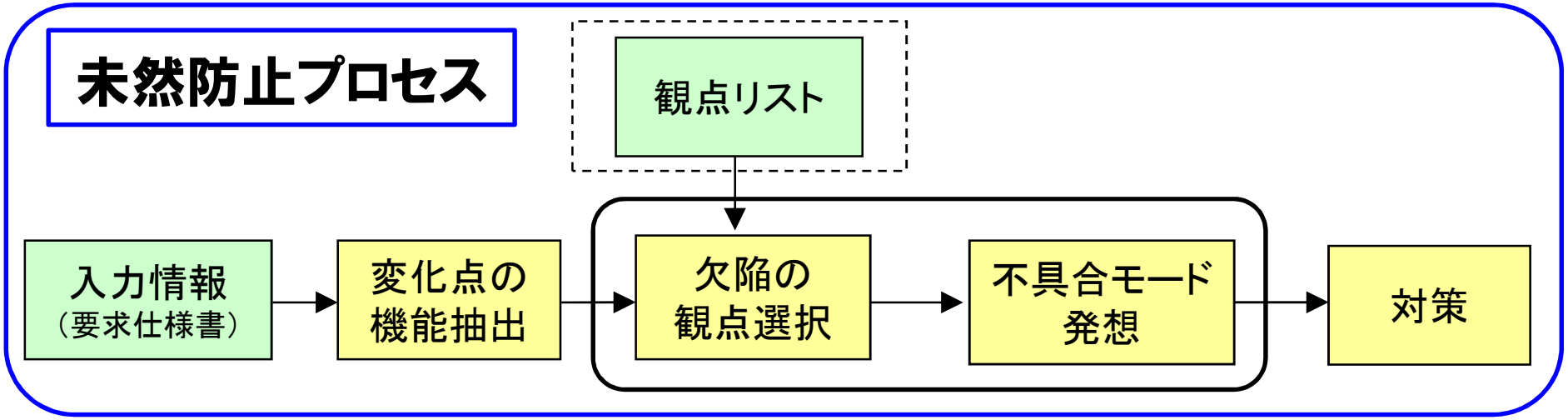
部分的な事例



施策1　未然防止プロセスの確立

■ 観点リストから不具合モードを発想

– 未然防止プロセスを立案



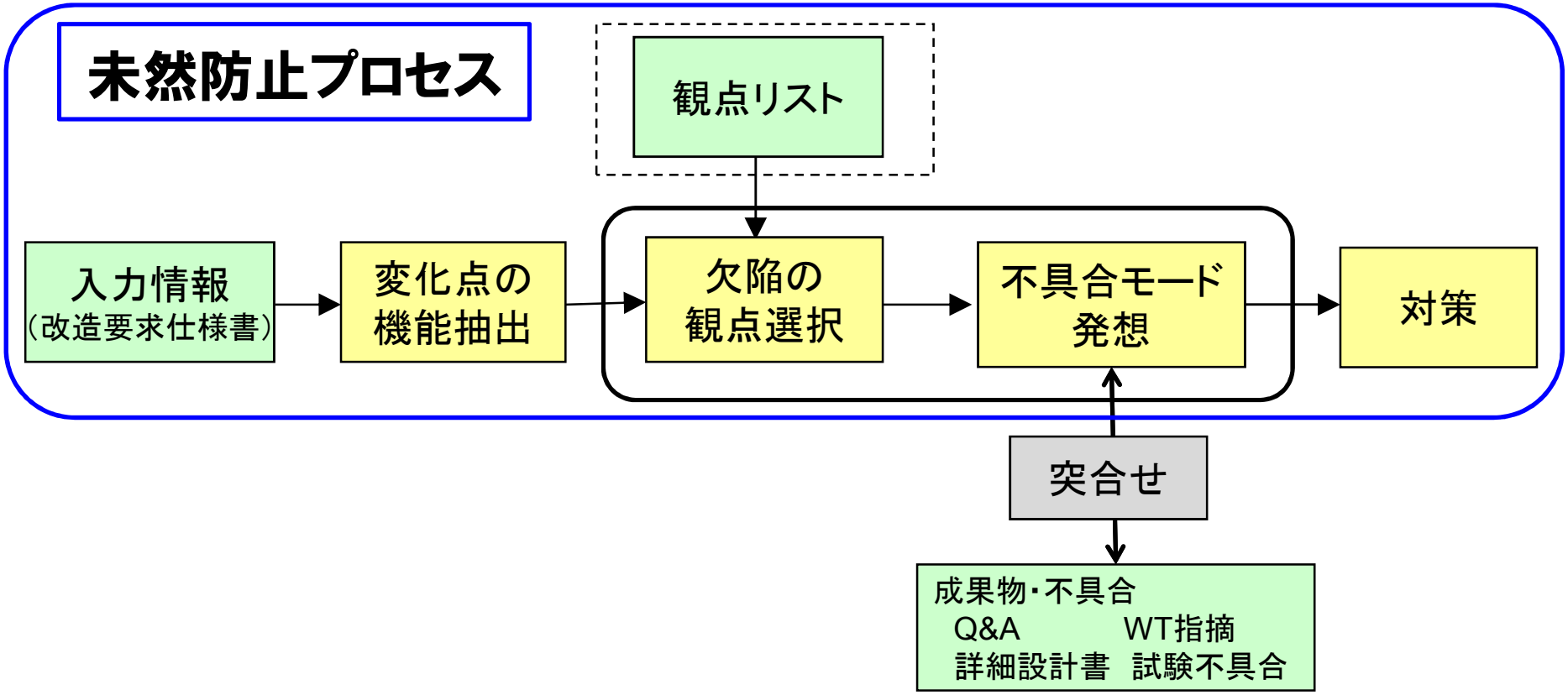
FMEAリスト

対象モジュール（構成要素、装置）／部品	要求（機能）	★①機能の裏返し	★②観点	不具合モード	考えられる原因	影響	検出方法	深刻度	発生頻度	検出難易度	RPN
Aタスク	情報の2重化処理を追加する	2重化に失敗する	例外処理からの復帰	データを正しく2重化できない	コピー元の正当性チェック漏れ	コピー先のデータを書き壊してしまう	障害観点レビュー	9	6	8	432

施策1　未然防止プロセスの確立

■ 未然防止プロセスの検証

- 過去の開発物件を使い、未然防止プロセスを試行
- 発想した不具合モードと成果物・不具合を突合せ



施策１　未然防止プロセスの確立

■ 検証事例

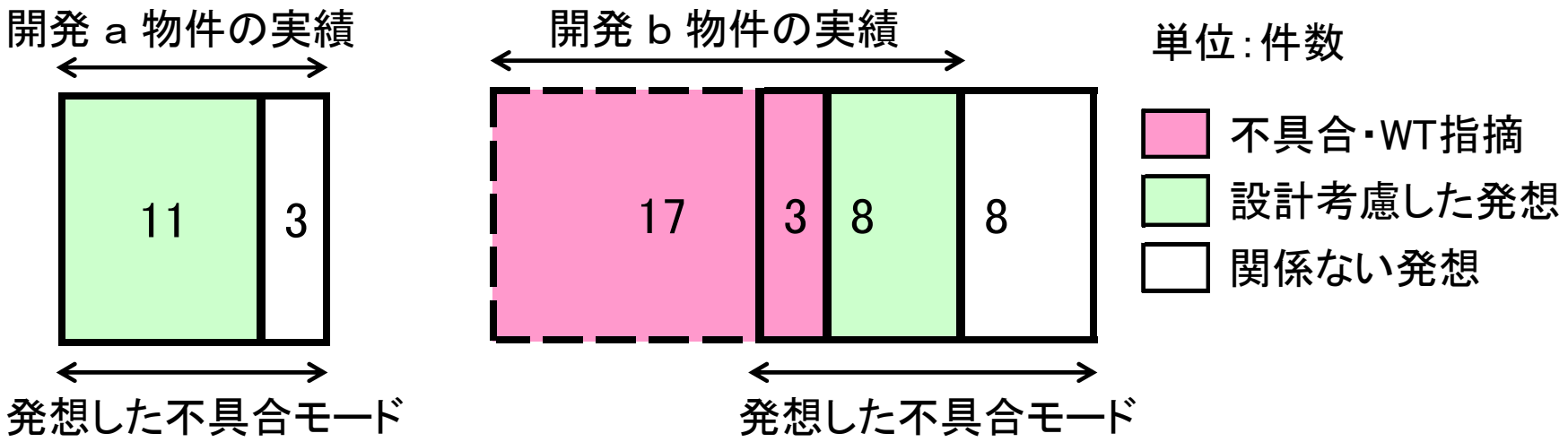
– 改造要求仕様（a物件）

・Aカードを最終利用日から起算して10年間使用していないと、失効とする機能の追加

＜機能＞	＜観点＞	＜不具合モードを発想＞
失効判定	想定外の入力	最終利用日が未来日となった場合の処理が抜ける
	処理順番	他の判定処理と優先度を誤る
	初期化忘れ	前に処理したデータを使用
最終利用日抽出	データ読出し	古いデータを読み出す

施策1　未然防止プロセスの確立

■ 検証結果



	率	a物件	b物件	効果
・不具合的中率	15%	—	3/20	×
・設計考慮発想率	50～79%	11/14	8/16	△
・関係ない発想率	21～50%	3/14	8/16	×

- 不具合モードが実際の**不具合に結びつきにくい**
- **関係ない不具合**モードまで発想

施策1　未然防止プロセスの確立

■ 検証者のコメント

– 効果

- 上流工程で、開発者の認識を揃えられる
- 過去に発生した不具合の再発防止だけでなく似たような不具合まで検討可能
- 観点リストを利用することにより、過去の不具合や経験をベースに検討漏れを防止

– 改善すべき点

- 時間がかかる
- ドメイン知識がないと、時間あたりの効果が小さい
- 技術者により不具合モードの発想がばらつく

■ 改善ポイント

発想のしやすさ　　発想力の向上

施策１　未然防止プロセスの確立

■ 未然防止プロセスの改善１

– 観点リストを「何が、どうして、どうなる方式」に変更し不具合モードを発想しやすくした

観点リスト

何が			どうして				どうなる
技術要素・環境			失敗メカニズムのパターン				機能不全のパターン
内部処理	例外処理	×	能力	スピード	不足	×	違う機能になってしまう
	異常発生時の処理			容量	不足		機能が欠落する
	データ操作				不安定		不要な機能が動く
	書き込み処理			性能	不足		性能不足
	読み出し処理				不安定		異常終了
	論理演算処理			サイズ	不足		DBを壊す
	通信・データ伝送処理			計算精度	不足		データを壊す
	割り込み処理		タイミング	誤内容			
	ポーリング処理			欠落			
	数値演算処理			速い			
	日付計算処理			遅い			
	タスク制御			ずれる			

東芝SFMEA-WG資料より

施策１　未然防止プロセスの確立

観点リストを「何が、どうして、どうなる方式」にあてはめ

＜何が＞ 技術要素・環境	＜どうして＞ 失敗メカニズム	＜どうなる＞ 機能不全パターン
①データ操作	想定外の入力	機能が欠落する
②分岐処理	処理順序が逆	不要な機能が動く
③読み出し処理	内部状態不定	違う機能になってしまう

不具合モードを発想すると

①最終利用日が未来日になった場合、処理が抜ける

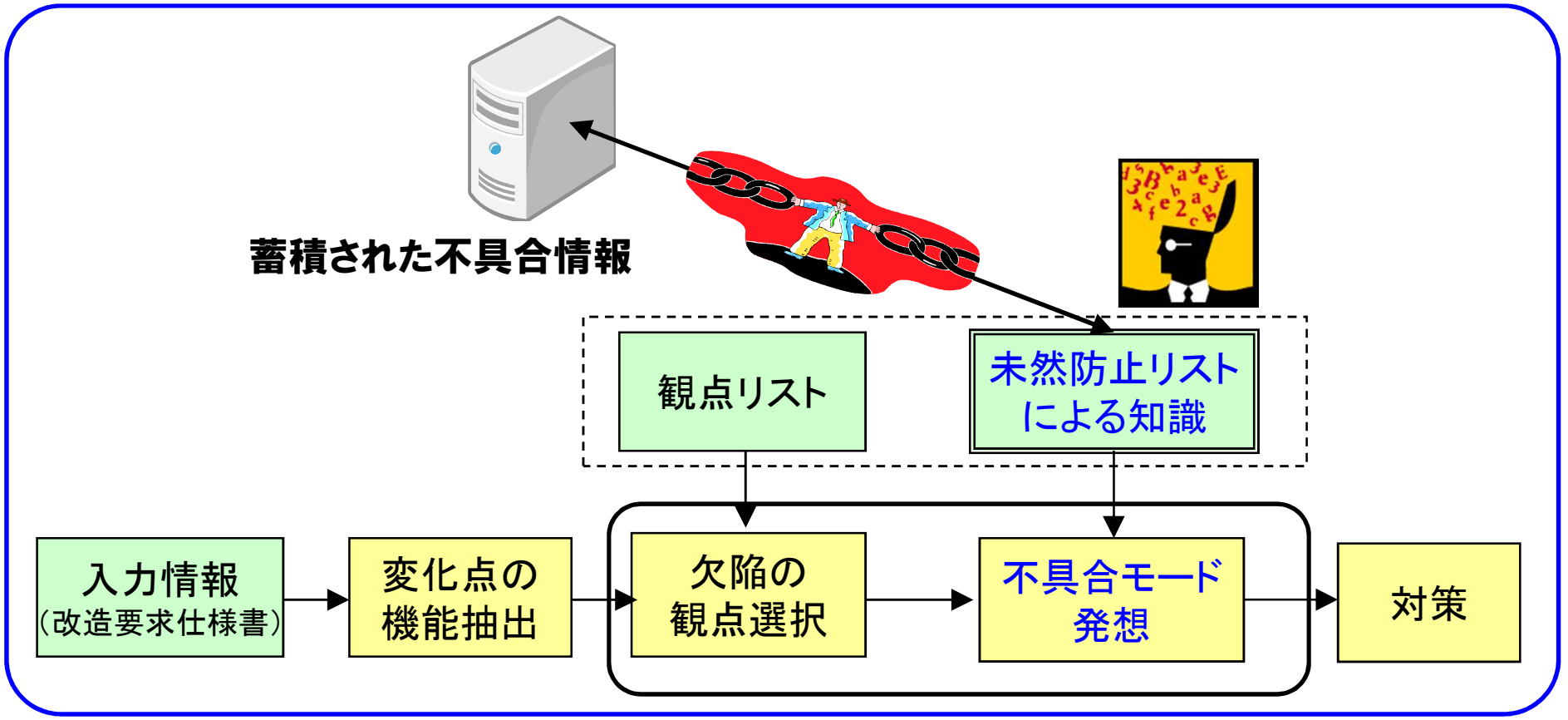
②他の判定処理と優先度を誤る

③初期化忘れにより前に処理したデータを使用してしまう

施策1　未然防止プロセスの確立

■ 未然防止プロセスの改善2

– 過去の不具合情報をベースにした未然防止リストを知識とし、不具合モードの発想力を向上



Agenda

- はじめに（背景と課題）
- 施策１　未然防止プロセスの確立
  - 観点リストから不具合モードの発想
  - 未然防止プロセスの検証と改善
- **施策２　未然防止リストの活用**
  - **過去の不具合情報を知識化**
- まとめ

施策2　未然防止リストの活用

■ 過去の不具合情報を知識化

– 目的

- 不具合情報から真意を伝える
  真因、発生メカニズム
- 痛みを伴う体験談を含める
- 次の開発では、不具合が形を変えて襲ってきても気づくような情報

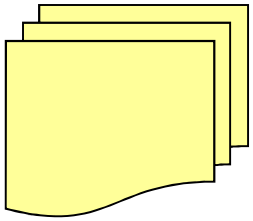
– 使い方

- 不具合情報を知識化した未然防止リストに変換
- 観点を付加し、未然防止プロセスに紐付け
- 観点から不具合モードを発想させる知識とし、技術者のスキルを補足

施策2　未然防止リストの活用

■ 未然防止データの蓄積

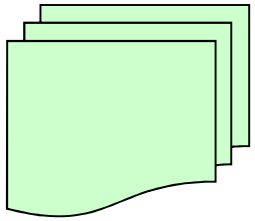
過去の不具合情報



・欠陥の観点
・ドメインキーワード

・不具合情報の知識化
・発生メカニズム図式化

未然防止リスト



■ 未然防止データの活用

検索キーワード

欠陥の観点

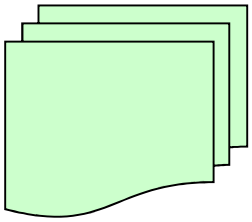
例：データ操作、想定外の入力

ドメインキーワード

例：払戻し計算機能

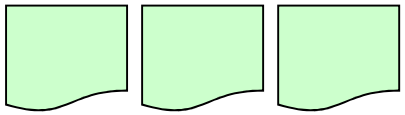
検索

未然防止リスト



抽出

欠陥を発想させる
知識データ



未然防止の知識を向上

施策2　未然防止リストの活用

■ 未然防止リスト

管理番号	発行日	タイトル
FC-0001	2011/12/15	大小比較の境界値ずれ

欠陥の観点1（何が）	欠陥の観点1（どうして）	不具合（現象・原因）の知識化1
内部処理 論理演算処理	境界値 ずれる	数値の大小比較において、未満の解釈を間違え境界値がずれた。

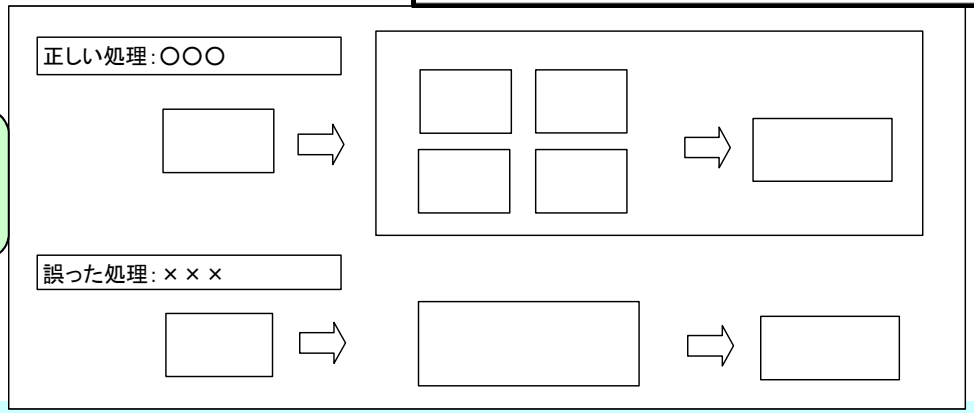
観点リストと紐付け

不具合を知識化

トラブル事例（正しい処理と不具合事象）	トラブル事例（原因）
入園料の小児料金が12歳未満と記載された仕様であるが、0～12歳を小児として取り扱ってしまった。 本来12歳未満とは11歳以下を示す。	設計書には未満の表現を記載しレビューを通過。 実装段階で、12歳未満を12歳を含めそれ以下と判断してコーディングしてしまった。

過去の不具合事例

発生メカニズムを図式化



Agenda

- はじめに（背景と課題）
- 施策1　未然防止プロセスの確立
  - 観点リストから不具合モードの発想
  - 未然防止プロセスの検証と改善
- 施策2　未然防止リストの活用
  - 過去の不具合情報を知識化
- **まとめ**

特に工夫した点

■ 未然防止プロセスの確立

– 不具合モードを発想しやすくするため、観点リストに改良を加え、「何が、どうして、どうなる方式」を採用

– 欠陥に対する知識レベルを高めるため、未然防止リストを利用し技術者スキルのバラツキを克服

■ 未然防止リストの活用

– 過去の不具合情報を知識化することにより、再発防止から未然防止が可能

活動成果と今後の展開

■ 活動成果

– ソフトウェア開発の設計検討段階で
欠陥の混入を防止することができる
未然防止プロセスの確立

– 過去の不具合情報を技術者全員が使える
技術資産の知識化（未然防止リスト）構築

■ 今後の展開

– 未然防止プロセスの適用事例を計画的に増やし
上流工程の品質作りこみを定着
– **未然防止の知識**を教育により**技術者に展開**

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
－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] 大和田尚孝：“システムはなぜダウンするのか
－知っておきたいシステム障害、信頼性の基礎知識”、
日経BP社、2009

ご清聴ありがとうございました

TOSHIBA

Leading Innovation