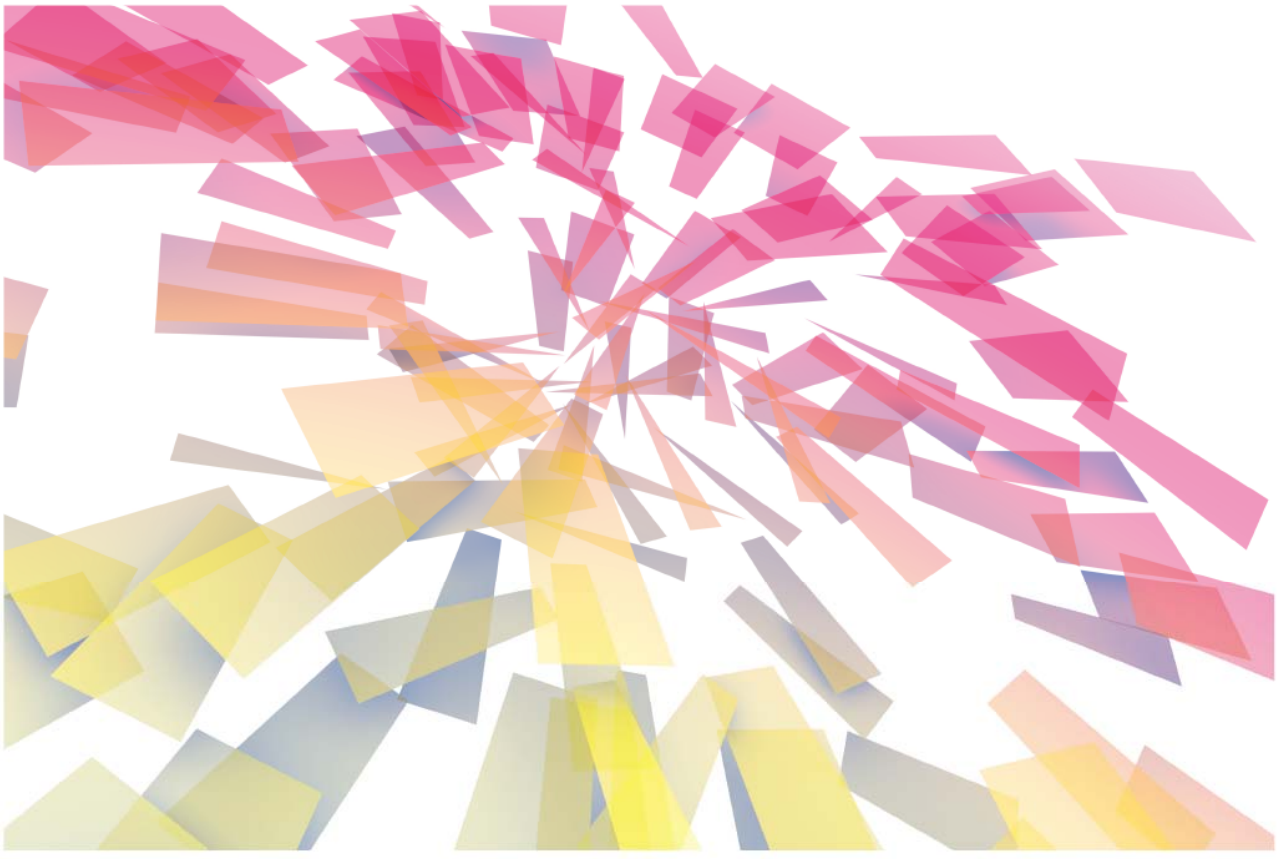
IPA Better Life with IT

現場で役立つ教訓活用のための
実践ガイドブック

（組込みシステム編）



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）

現場で役立つ教訓活用のための実践ガイドブック（組込みシステム編）

独立行政法人情報処理推進機構

目次

1 はじめに

1.1 概要

IPA/SEC では 2013 年度より障害事象から得られた経験・ノウハウを「情報処理システム高信頼化教訓集（組込みシステム編）」[1]（以下“教訓集”と称する）としてとりまとめてきた。本ガイドは、多くのものづくり企業で行われている実際の開発プロセスや社内教育などにおいて、この教訓集をどのように活用することができるか、その実践的な活用法を解説したものである。また企業内で実際に取り組まれている品質マネジメント、再発防止の活動事例も参考として掲載した。

「障害未然防止のための教訓化ガイドブック（組込みシステム編）」[2]とあわせて参照されたい。

1.2 対象読者

本ガイドの利用者としては組込みシステム開発に関わる以下の読者を想定する。

・ソフトウェア設計者
・システム設計者
・品質管理者
・技術者教育担当者

1.3 用語定義

本ガイドの中で重要と思われる用語について定義した。

再発防止：製品・システムで実際に発生した問題（故障・不具合）に関する根本的な原因を追究し、その原因に対して当該製品・システムで今後同じ問題が発生しないように対策を講じること。

（参考）問題の原因又は原因の影響を除去して、再発しないようにする処置（JIS Q 9024:2003）

未然防止：製品・システムで得られた再発防止の知見に基づくリスク要因を、他製品やシステムへ適用し類似の障害発生を予防する取組み

（参考）起こり得る不適合又はその他の望ましくない起こり得る状況の原因を除去するための処置（ISO9000「予防処置」）

教訓：製品ドメインに限定、特化することなく、当事者以外の人、他製品・システムにも役立たせることができる経験知識・ノウハウ

原因分析：障害発生時その情報を収集しシステム構造把握、問題構造把握を行い、障害を引き起こした直接原因を分析するまでの一連の活動

真因分析：原因分析によって明らかにされた直接原因に対し、その原因を発生させるに至った真因を抽出するための分析

障害の原因：障害を発生させた要因

障害の真因：障害の原因のうちの本質的な原因

1.4 本ガイドの構成

本ガイドは以下のような構成となっている。

２章　教訓集の構成と特徴
３章　組織学習のための基礎
４章　基本的な活用方法
５章　企業内での活動事例

第２章では、本ガイドが活用対象として参照している教訓集の構成と、その特徴を記している。
第３章では、教訓を活用する上で基本となる、教育・学習の基礎理論やバックグラウンドを解説している。
第４章では、本ガイドの中核部分で、教訓集を「社内教育・研修」「開発プロセス」「設計品質向上活動」などの利用場面を設定し、それぞれにおける活用のポイントと具体的な活用方法を示している。
第５章では、実際に企業内で行われている再発防止、未然防止に向けた教育、品質管理、開発プロセス管理の事例を紹介している。

2 教訓集の構成と特徴

本章では教訓集の構成について示す。まず、図 2-1.に想定される課題ニーズとそれに該当する参照先 PART を示し、各 PART の要点を解説した。なお、具体的な活用方法は活用シーンごとに 4 章に示した。

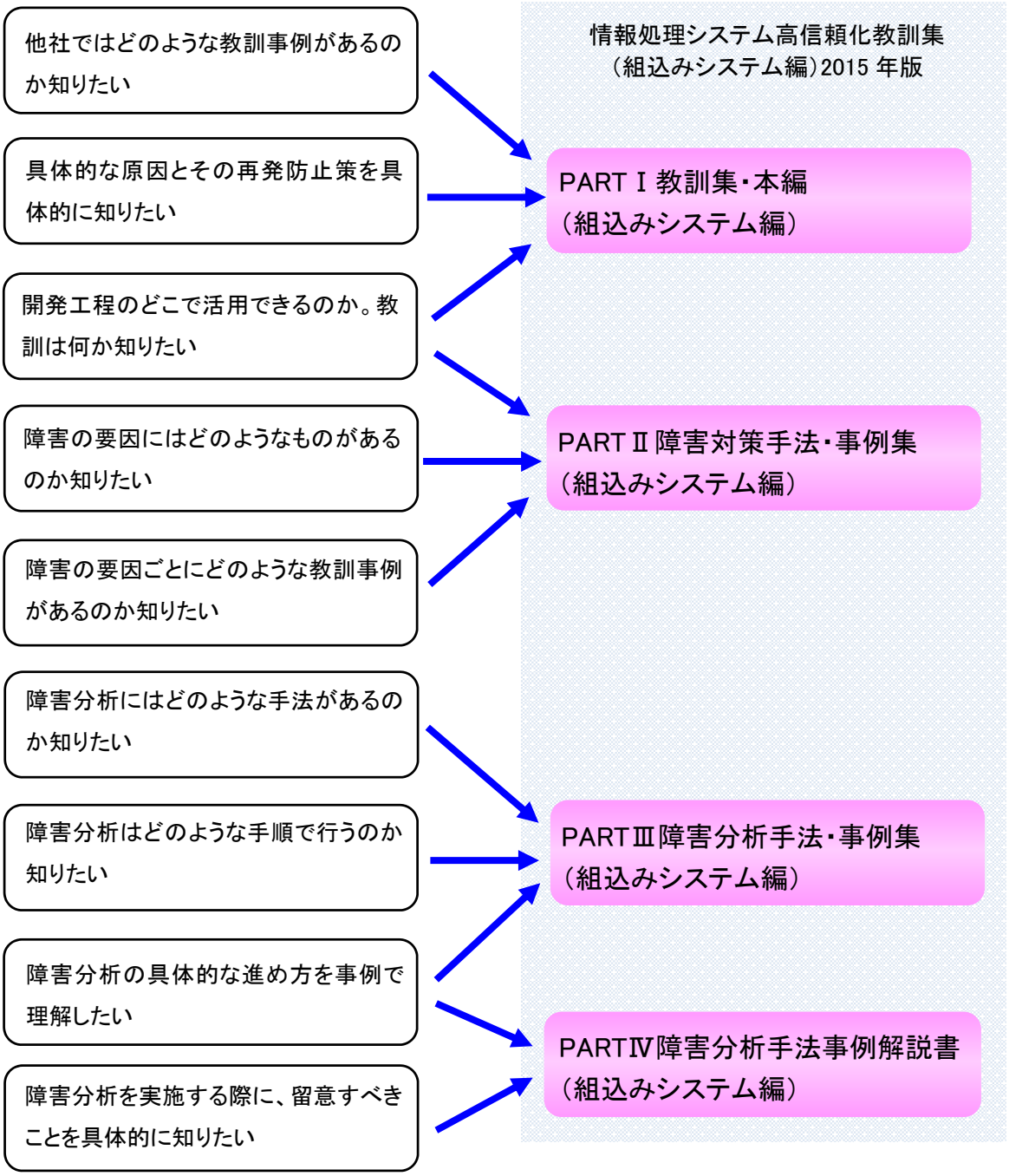


図 2-1. 課題ニーズと参照先

2.1 PART Ⅰ 教訓集・本編（組込みシステム編）

PART Ⅰには 2013 年度から収集した 35 の教訓事例を掲載しており、各教訓事例は図 2-1 のようなシートで整理している。ここではそのシート表記構成とポイントを以下に説明する。

①教訓タイトル：事例が示す教訓内容を端的に表現したもの。

②製品の特徴：障害発生した製品、システムの特徴をその重要度に照らしたシステム構成や運用などの観点から記述。

③観察できる現象：障害発生時に認識された具体的事象を記述。

④内部で発生した事象：上記③の発生事象を引き起こした直接原因を記述。

⑤原因となる要因：当該事象を引き起こすに至った背景要因について記述。これには不具合を作りこんだ要因と、それを流出させた要因がある。

⑥上記の未然防止に向けた対策：上記までに抽出された要因の対策を記述。

⑥-1　直接原因への対策

⑥-2　要因への恒久対策

⑥-3　ソフトウェア開発プロセスの該当工程と、その工程における教訓事項

教訓 9 ①	
教訓タイトル	システムを二重化する場合は、同期すべきデータ領域を適切に設定する
製品の特長 ②	高稼働率（無停止期間の長期化）が必要とされる遠隔監視システムにおいては、通常の故障や誤動作の発生頻度を小さくする以外に、故障が発生しても動作を続行する、故障から早く回復する機能が要求される等、高い信頼性が求められる。
観察できる現象 ③	二重化システムを採用して高稼働率を実現するためには、マスター側が故障した場合でも、制御の連続性を維持してスレーブ側に切り替わらなければならないが、スレーブ側に切り替わった直後に、異常を通知するアラームが発生した。
内部で発生した事象 ④	スレーブ側に切り替わった時点で、データ同期をとっていなかったデータの値が不正値となったため、パラメータ異常のアラームが発生した。
原因となる要因 ⑤	機能追加時に管理に必要なデータ領域を追加したが、同期をとるべきデータ領域を変更しなかった。また、追加したマスター側のデータ同期領域が使用されている状態の検査が漏れていたため、切り替わった場合にスレーブ側とデータ同期がとれていないことが分からなかった。
上記の未然防止に向けた対策 ⑥	直接原因への対策： ⑥-1 データ同期が必要なデータ領域を修正する。
	要因への恒久対策（対応工程を明記）： ⑥-2 データ同期の検査項目に、マスター側が追加データ領域まで使用している状態を加える。 データ範囲の境界の値の確認を検査項目に追加する。 これにより、二重化システムに関するデータの引き継ぎにかかわる動作不良の防止が容易になる。
⑥-3	■ソフトウェアアーキテクチャ設計（変更設計） ・二重化システムを変更設計する場合には、単体のみならず、二重化システム全体の影響解析をすること。 ・二重化システムを変更設計する場合は、同期させるデータの領域に注意すること。

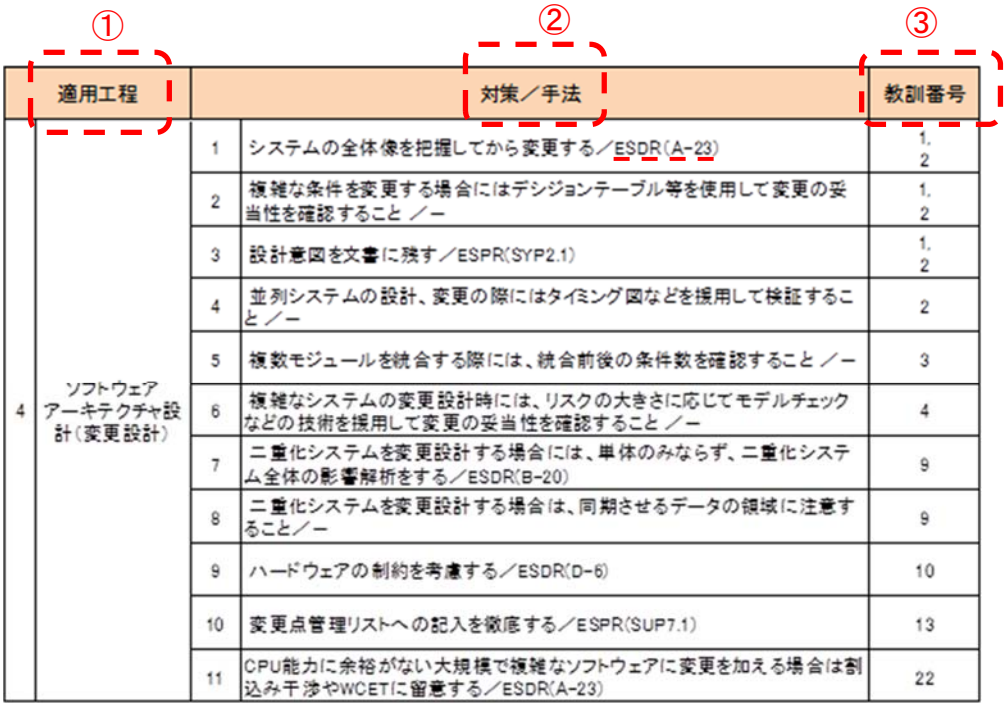
図 2-2. 教訓事例シートの表記構成

2.2 PART II 障害対策手法・事例集（組込みシステム編）

PART Ⅱは、PART I で収集した教訓事例を実務活用時に検索しやすくするため、各事例の分類体系を設け、各教訓の対策や活用法を整理している。

【工程別対策事例】

35 件の事例の中の真因への恒久対策を抽象化し、工程別の一覧表に対策事例として整理した。対策事例の多くは、IPA/SEC が発行している SEC BOOKS（ESPR[3]、ESDR[4]などの書籍）の開発手法が参考になるため、これに記載されているものを採用した。ただし、該当するものが無いものについては、“／－”を付している。



① 適用工程		② 対策／手法		③ 教訓番号
4	ソフトウェアアーキテクチャ設計（変更設計）	1	システムの全体像を把握してから変更する／ESDR(A-23)	1, 2
		2	複雑な条件を変更する場合にはデシジョンテーブル等を使用して変更の妥当性を確認すること ／－	1, 2
		3	設計意図を文書に残す／ESPR(SYP2.1)	1, 2
		4	並列システムの設計、変更の際にはタイミング図などを採用して検証すること ／－	2
		5	複数モジュールを統合する際には、統合前後の条件数を確認すること ／－	3
		6	複雑なシステムの変更設計時には、リスクの大きさに応じてモデルチェックなどの技術を採用して変更の妥当性を確認すること ／－	4
		7	二重化システムを変更設計する場合には、単体のみならず、二重化システム全体の影響解析をする／ESDR(B-20)	9
		8	二重化システムを変更設計する場合は、同期させるデータの領域に注意すること／－	9
		9	ハードウェアの制約を考慮する／ESDR(D-6)	10
		10	変更点管理リストへの記入を徹底する／ESPR(SUP7.1)	13
		11	CPU能力に余裕がない大規模で複雑なソフトウェアに変更を加える場合は割込み干渉やWCETに留意する／ESDR(A-23)	22

図 2-3. 工程別分類（抜粋）

①適用工程：「システム要求定義」から「運用」までの 10 工程に分類し、各工程に関係する教訓を記述

②対策／手法：PART I の各教訓事例の⑥-3 ソフトウェア開発プロセスの該当工程と、その工程における教訓事項の記述内容をこの欄に転記し、かつ ESDR、ESPR 等の既存 SEC BOOKS の関連がある場合はその書籍名と該当項目番号を記述している

③教訓番号：該当する教訓事例番号を記述

【観点マップ】

PART I の教訓事例をその障害を引き起こした要因で分類したもので、各教訓事例を自社・自部門製品に適用することを想定し、その利活用のトリガとなるよう、マインドマップを用いた整理をしている。各要素に付与されている番号は、該当する教訓の教訓番号である。

①直接原因観点マップ

35 事例の障害を引き起こした直接原因を整理したものである。

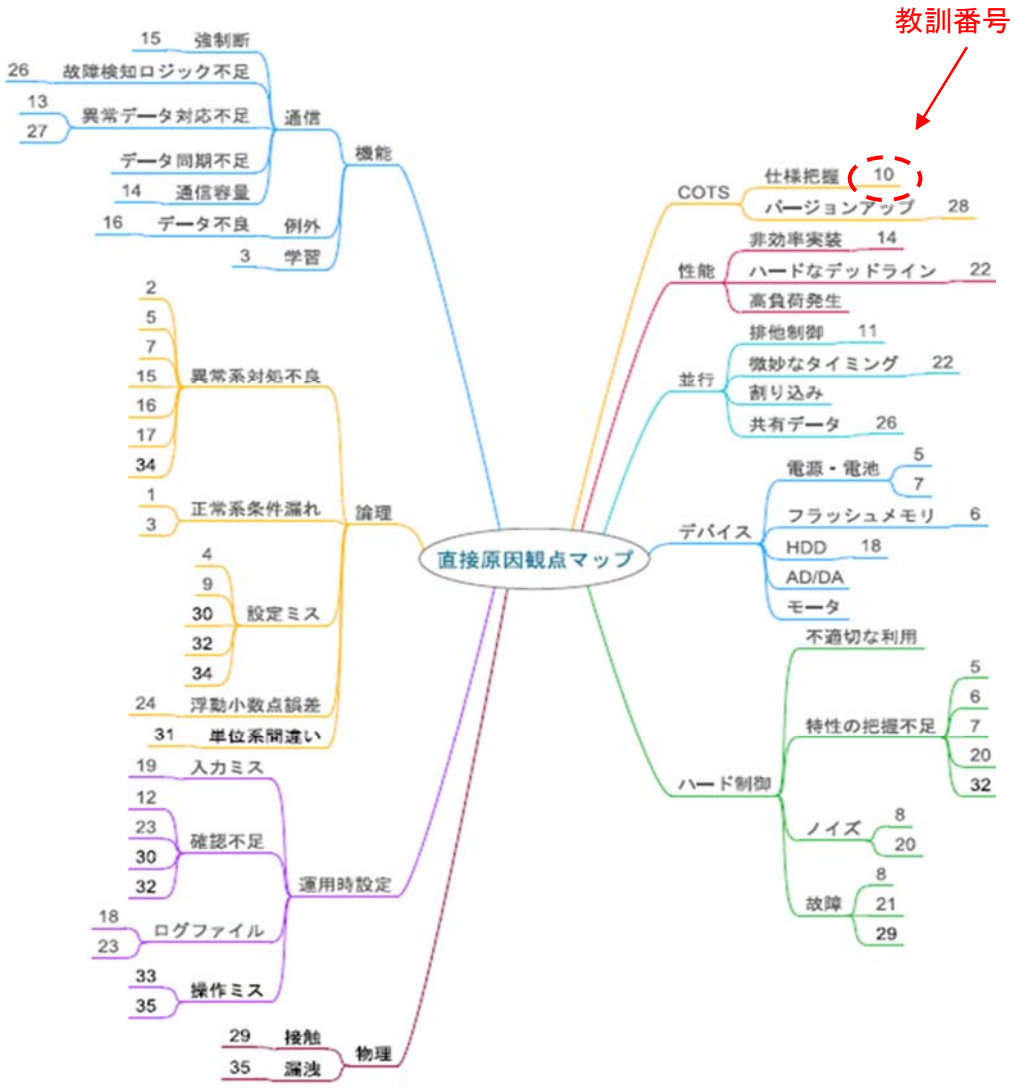


図 2-4. 直接原因観点マップ

②未然防止観点マップ

35 事例の障害を引き起こすに至った真因を階層的に分類し異なる製品ドメインに対しても未然防止に各教訓事例を活用できるように整理したものである。

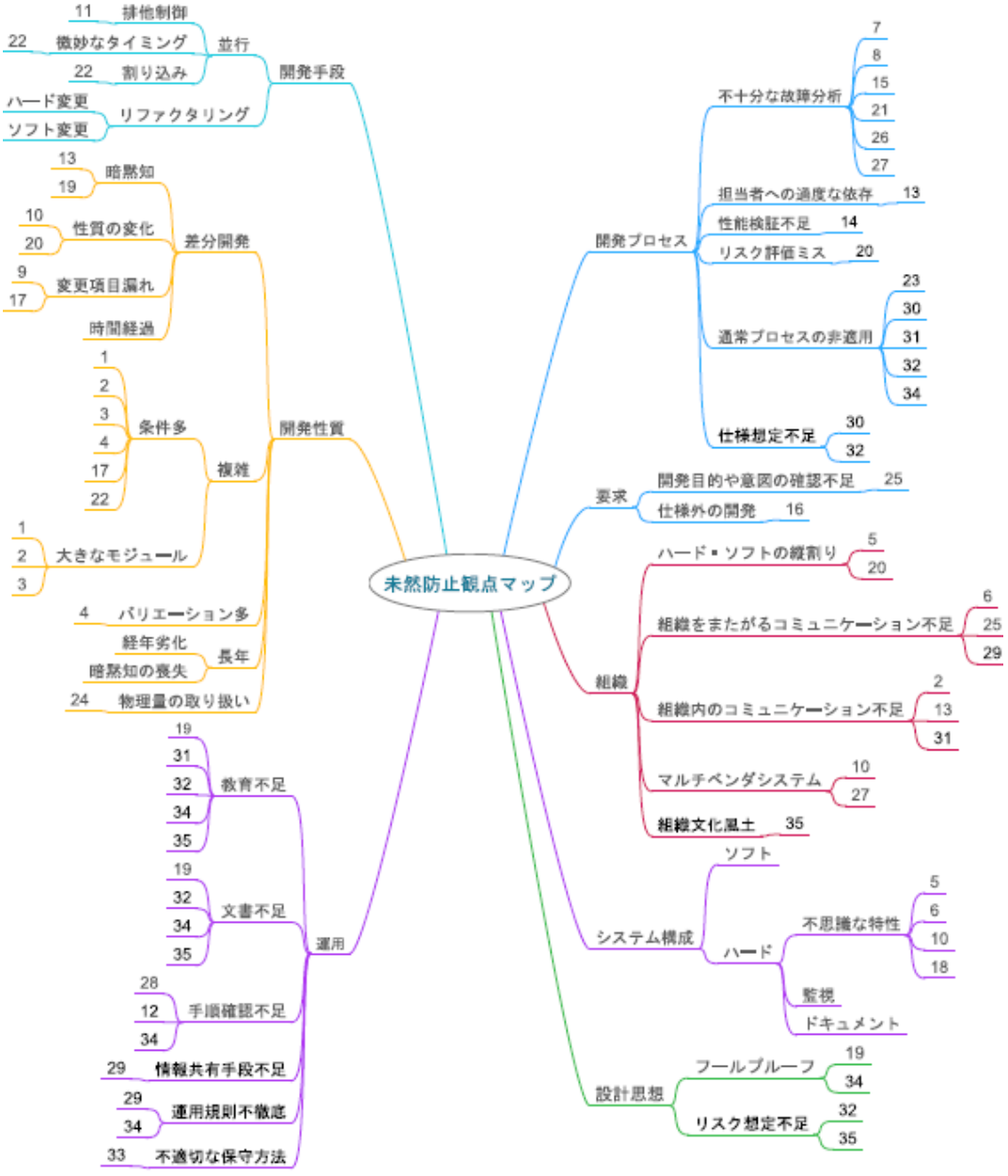


図 2-5. 未然防止観点マップ

2.3 PARTⅢ障害分析手法・事例集（組込みシステム編）

PART Ⅲでは障害発生時にその原因を分析し、真因を特定した上で再発防止の手立てを打つまでの対応手順の概要と適用される手法を、ステージごとに概説している。

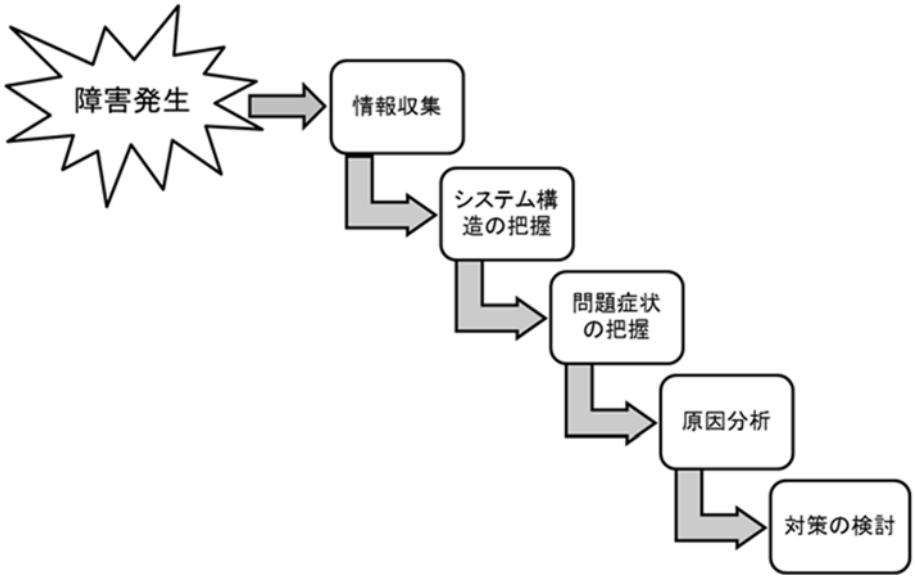


図 2-6. 障害対応手順

【原因分析の手法】

発生した障害からその原因を追究するための手法として下記手法を例示している。

①ブロック図：一般的に、システムを構成する機能、ハードウェアデバイス（個別のセンサ、アクチュエータ等も含む）、ソフトウェア等をそれぞれ独立したブロックとして記述する。

②事故経過表：時刻欄、発生した問題内容欄、内容の分類欄からなる表形式で作成し、システムの状況が変化する都度、行を追加して問題内容等を記載する。

③VTA（Variation Tree Analysis）：事象とシステム構成要素の関係及びシステム構成要素間の関係を図示する手法である。建設業界でよく利用されている。

④問題行動分析：事故経過表をもとに、事象を引き起こした直接的な問題行動（の候補）を列挙・分析する。

⑤PNA（プロセスネットワーク分析法）：ソフトウェア開発に関連するプロセス（業務）

の特徴と相互関係を明示して真因を探る手法。

⑥発生源・検出漏れ分析 : 作りこみ要因及び流出あるいはすり抜けさせた要因を分析する手法。

⑦例外分析 : 発生しうる例外事象を項目立てて整理し、それぞれの例外事象への対応を考察する手法。

⑧なぜなぜ分析：システムの障害や欠陥の原因を分析する方法として広く利用されている。

【分析手法を試行適用した事例】

上述した分析手法を下記のような実際に発生した障害事例（公開済）に対して適用している。

・湘南モノレール（2008 年 2 月 24 日。列車のブレーキ制御に起因する事故）
・天竜川水系阿知川の駒場ダム（2002 年 5 月 9 日。異常放流事故）
・アリアン 5 ロケット（1996 年 6 月 4 日。慣性制御異常による爆発事故）
・カンタス航空（2008 年 10 月 7 日。意図しない急降下が繰り返し発生）

2.4 PARTⅣ障害分析手法事例解説書（組込みシステム編）

PART Ⅳは、PART Ⅲで紹介した障害分析手法と障害分析作業を具体的事例に即して解説したもので、下記の点を考慮した記述内容となっている。

【有識者の知見による解説】

PART Ⅲの分析手法について、経験豊富な技術者が通常どのように障害分析しているのかを事例に則し、留意点などの項目を設け解説している。

> 原因分析
> ☞　留意点
> （ｉ）ハードウェア（HW）要因をまず疑い次いでソフトウェア（SW）要因を追求する
> ・HW が原因のことが多かったこともありこれを先に疑う
> ・HW は、“もの”に焦点を当ててヒアリングする
> ・HW でないとわかったら SW（コト、状況、ふるまいに注目）を疑い、関係者のヒアリングをする
> ・SW は、“もの”ではなく振る舞いとか、目的とか“こと”に焦点を当ててヒアリングするコミュニケーションスキルが前提
> ・その中で怪しい回答を拾い出しながらメモしていく（見当付けていく）
> ・ヒアリングした内容を詳細も確認して書き出す
> ・ベースになっているのは経験則。経験が薄い人には教訓集は必要と思う
> （ⅱ）SW 要因の追求は「技術」「プロセス」「マネジメント」の３つの観点で行う
> ・これで見ていくとどれかにひっかかる
> ・疑う順番
> プロセス　→　プロジェクトの制約　→　技術　→　マネジメント
> ：

【企業内の再発防止例】

障害の分析結果を再発防止につなげる取組みの事例を紹介している。

①Ａ社事例：障害は人により作り込まれるという前提にたち、障害を未然に防止するためには、人の作業を形式的に実行する障害を作り込まないフレームとフレームに実装する開発プロセス定義が重要。この方針に基づき、障害の真因分析による障害を未然に防止するフレームと開発プロセスを定義している。

②Ｂ社事例：社内外含めた品質問題の対応体制が構築され、不具合の解決手順やフォーマットなども定められたものがある。また、真因を分析した不具合例をケーススタディ化しており、結果を教材として新人等に教育している。

3 組織学習のための基礎

本章では教訓集を活用するための基礎となる、組織学習とプロセス改善についての概念を解説する。

3.1 インストラクショナルデザイン (ID)

インストラクショナルデザイン(Instructional Design; ID)とは、意図的な学習を支援する学習システムを設計、実施、評価するためのアプローチで今日では e ラーニングなどで広く適用されている。[5][6] 教訓などに基づく組織的な学習システムを社内教育に応用するための基礎として本ガイドでは 4 章及び 5 章の事例適用のバックグランドとして位置づける。

[教育設計プロセス ADDIE モデル]

組織的な学習システムを構築するためには、効果的な学習のためのプロセスを考える必要がある。ソフトウェアシステム開発のプロセスを一般化したプロセスモデルがあるのと同様に、教育システムの設計についてもプロセスモデルがある。ADDIE モデルはその代表的なものであり、分析(Analyze)、設計(Design)、開発(Develop)、 実施(Implement)、評価(Evaluation)という教育活動の各工程から成る。

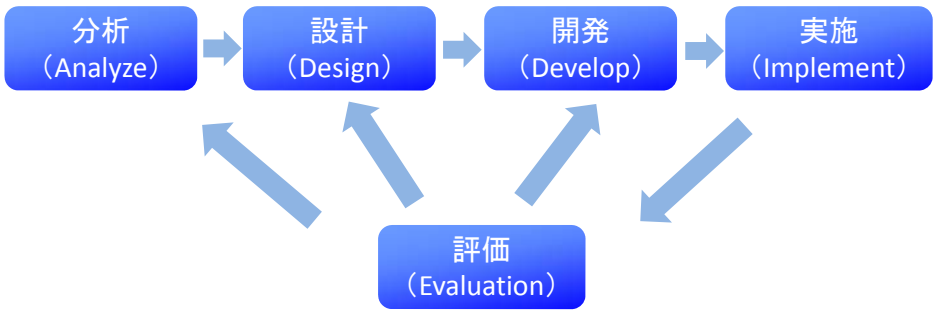


図 3-1. ADDIE モデル

- 分析(Analyze)
  教育に対するニーズを分析しゴールを決定する。さらに学習に関する諸条件（学習者の前提スキル、利用可能な時間、など）を決定する。
- 設計(Design)
  取り上げるトピックを決め、どの順番で教え、各トピックにどれだけ時間を利用するかを決定する。また、それぞれのトピックを具体化して達成すべき目標を決め、学習活動を定義する。さらに、学習状況を評価するための指標を決める。
- 開発(Develop)

学習活動と教材の種類について決め、それらの草案を作成する。草案の使用を依頼し、その結果をもとに教材と活動を改善する。

- 実施(Implement)
  開発した教育を実施する。
- 評価(Evaluation)
  開発した教育やそのプロセスを評価し、改善につなげる。教材、教育の開発プロセス、学習者の反応と達成度、教育の効果（研修の効果が職務に有効活用できているかなど）を評価する。

[教育目標とその分類]

教育活動の目標、すなわち定着を図るべき知識、技能などには様々な種類があり、それぞれごとに定着させるための方策が異なる。そのため、教育目標の分類について知っておく必要がある。ここではブルームの分類[3]のうち、未然防止知識に関連する分類について解説する。

- 認知的領域
  頭が働く領域であり、言語情報として記憶したり、記憶していた情報を利用して考えたりすることがそれにあたる。未然防止知識においては、ある教訓を述べる、教訓の分類を述べる、未然防止知識観点マップをもとに真因を特定するなどがあげられる。
- 情意的領域
  心に関する領域であり、ある状況を選ぼう、もしくは、避けようとする気持ちである。未然防止知識においては、品質を第一とする心構えをもつ、などが挙げられる。
  態度は命令して身につけさせることはできず、コミュニティでの活動を通じて身につくものであると考えられている。

3.2 プロセス改善

多くの企業ではシステム・ソフトウェア開発のプロセスを定め運用している。開発プロセスは ISO/IEC12207 (Software Life Cycle Processes)や ISO/IEC15288(System Life Cycle Processes)など国際規格でもモデルが示されており、これらを参考に組込みソフトウェア向け開発プロセスガイド（ESPR）が IPA/SEC からリリースされている。

障害発生時にはその再発防止の手段としてこうした開発プロセスに対策をフィードバックするといったプロセス改善の活動が行われている。プロセス改善とは国際規格（ISO/IEC15504 Information technology – Process assessment）に示される通り、改善すべきことを見つけ出しあるべき姿に向けて修正する、このサイクルを繰り返すということである。障害対応時に実施されるプロセス改善として代表的なものは、プロセス主要工程の節目で行われるレビュー方法やレビュー項目、プロセス定義の見直し、変更などがある。

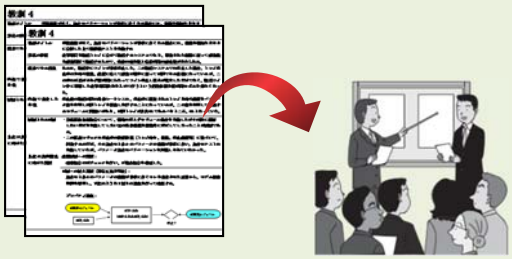
本ガイドにおいても 4.2 開発プロセスへの活用や 4.3.2 障害対応プロセスへの活用、5.3 開発プロセス管理事例などにこうしたプロセス改善に基づく取組みが示されている。

4 基本的な活用方法

本章では３章で示した教育モデルの理論、開発プロセスの基本的な考え方に基づき、教訓集を効果的に活用しうる適用シーンを下記のように想定し、そのシーンごとにどのような活用が可能かを示す。

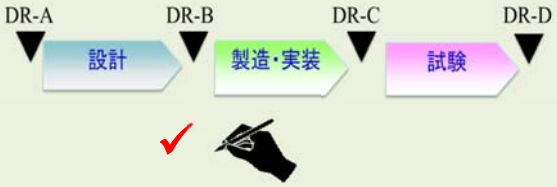
4.1 社内教育・研修に活用する

企業で取り組まれている社内教育・研修に教訓集の各事例や分析手法を適用する。3 章で示した教育設計の考え方に基づき、教育資料を作成、応用することで教育・研修の効果をさらに高めることができる。



4.2 開発プロセスに活用する

本教訓集中の事例から得られる設計留意事項や分析手法を、3.2 プロセスの改善の考え方に基づき、デザインレビューのチェック項目に反映する、現行プロセスの見直しを行う等の際に参考情報として活用する。



4.3 設計品質向上活動に活用する

障害発生に実施する真因究明の手順やプロセスに対し、本教訓集の障害対策手法や観点マップを適用する、あるいは各事例を自社製品に置き換えて設計危険予知訓練（ヒヤリハット）の題材として活用する。

図 4-1. 活用シーン

4.1 社内教育・研修への活用

4.1.1 障害分析に関わる教育資料作成における活用

➢想定される状況・課題

障害発生時に原因究明のための取組みとしてなぜなぜ分析を行っているが、自己流になっているため個人や部門による差が大きく、またなぜなぜ分析以外の手法も知らないため会社組織全体として効果のある再発防止を行うことができていない。

➢活用の狙い

組織標準的な再発防止手順、手法などの教育用資料を作成する際の参考として使用する。

➢活用方法

PARTⅢ、Ⅳで示されている障害分析手法を社内教育資料作成時に利用する。具体的には自社で行っている再発防止策を導出する手順に対して、PARTⅢで示されている各種手法の有効性を検討し、効果を見込めると考えられる手法があれば採用し教育用資料に反映する。

また、その中で例示する障害事例として PART Ⅰ 教訓集の中の事例をケーススタディサンプル等として採用する。

なぜなぜ分析については、会社ごとに独自の工夫が盛り込まれることが多く、この際にPARTⅣで示されている方法、および「留意事項」に記述されている内容、なぜなぜ分析で例示されている表のサンプルなどを参考にすることができる。

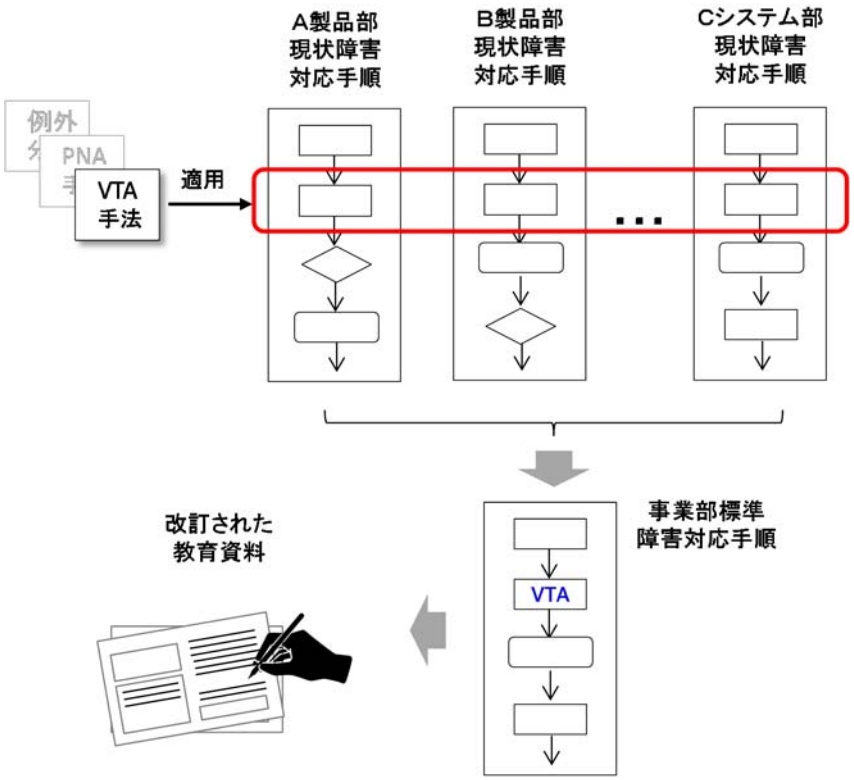


図 4-2. 社内教育資料作成への活用

➢期待効果

社外で実施されている分析手法を取り入れ、また同じ手法でもアプローチや考え方の違いを認識することにより、新たな視点で問題解決を行うことができる。

➢留意事項

教訓集中の事例に示されているエッセンスを理解し、分析手法の有効性を評価し自社内教育資料に反映する上では相応の経験を有することが前提となるかもしれない。

4.1.2 小集団活動等での活用

➢想定される状況・課題

障害時には真因を追求するためになぜなぜ分析などを行い再発防止に努め、日頃のルーチンワークにおいても小集団活動を通じた情報共有を定期的に実施しているが、障害内容に新規性が薄く活動が形骸化しがちとなっている。

➢活用の狙い

小集団活動の形骸化を防ぎ、新たな視点による気づきを得る。

➢活用方法

品質に関わる小集団活動の一環として職場のグループ（10 人程度）の朝ミーティング等で本教訓集の中の事例の輪読や、事例に基づいた KYT（危険予知トレーニング）を実施する。この際、従来経験したことが無い事例の場合はチェックシートに追加する等の施策を実施する。

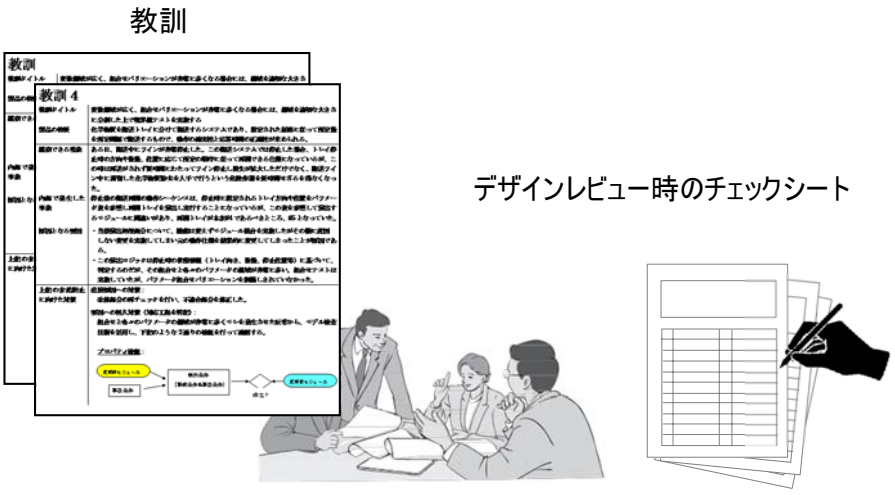


図 4-3. 小集団活動での活用

➢期待効果

同じような障害事例でも、タイプの異なるものや過去に経験のない内容を知ることで障害分析において新たな視点を与え、あるいはリスクを察知することができるようになる。

➢留意事項

要素技術や開発技術の隔たりが大きいような場合、事例集の教訓を自部門用に読み替えるなどの工夫が必要で、相応の経験者による事前準備とファシリテーションが前提となる。

4.2 開発プロセスへの活用

本節では開発プロセスに教訓事例集を活用する場合の応用例を示した。

4.2.1 真因分析と未然防止の進め方

➢想定される状況・課題

個人に依存した開発から組織的な開発に移行するためには、作業フレーム（様式およびソフトウェアフレームワーク）と作業フレームに実装するための開発プロセスの定義が必要である。

障害を未然に抑制するために障害の真因を特定して未然に抑制する作業フレームとフレームに実装する作業を開発プロセスとして定義してソフトウェア開発を行っている事例があり組織的な効果を得ている。しかし、障害の分析において真因に到達することが難しく正しいフレームと開発プロセスを定義することが困難な状況になっている。

➢活用の狙い

障害の真因への気づきを得ようとする際、多くの経験の中で自分の記憶にあるもののみ検出可能である。経験のない物事は理解が進まない。

日常の開発作業の中で、真因に到達するための素養を高めるとともに、未然防止に向けたプロセス改善に活用する。

➢活用方法

障害分析に基づくプロセス改善の事例を図 4-4 に示す。障害データを混入と流出・すり抜けた工程を分類する。各工程で障害要因を分類して真因を定義する。

水平展開として同種の障害が関連製品にはないことを確認すると同時に障害の真因を未然防止する作業フレームとフレームに実装する開発プロセスの定義により組織的な障害発生の未然防止を実現する。

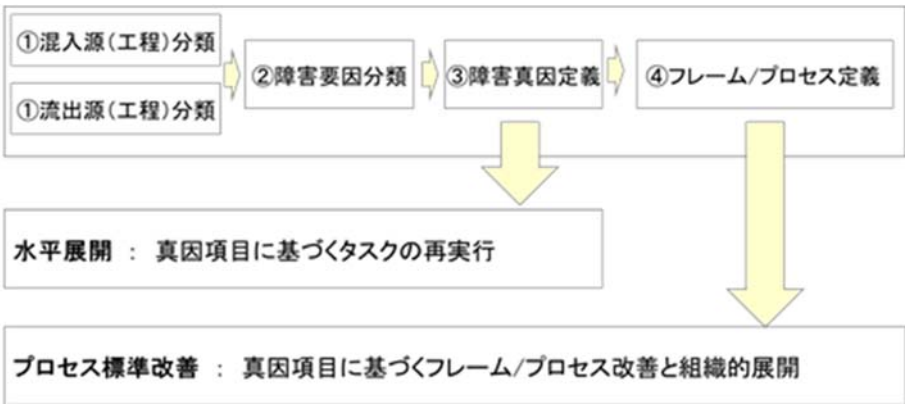


図 4-4. プロセス改善のプロセス例

図 4-5 に活用方法を示す。未然防止事例を基に観点マップと分析手法を活用して事例に依存しない真因と教訓を定義する。真因と教訓を自組織のケースに関連づける。真因と定義した教訓に基づき自部門の障害と関連付けてフレームとプロセスを定義して組織的改善へつなげる。

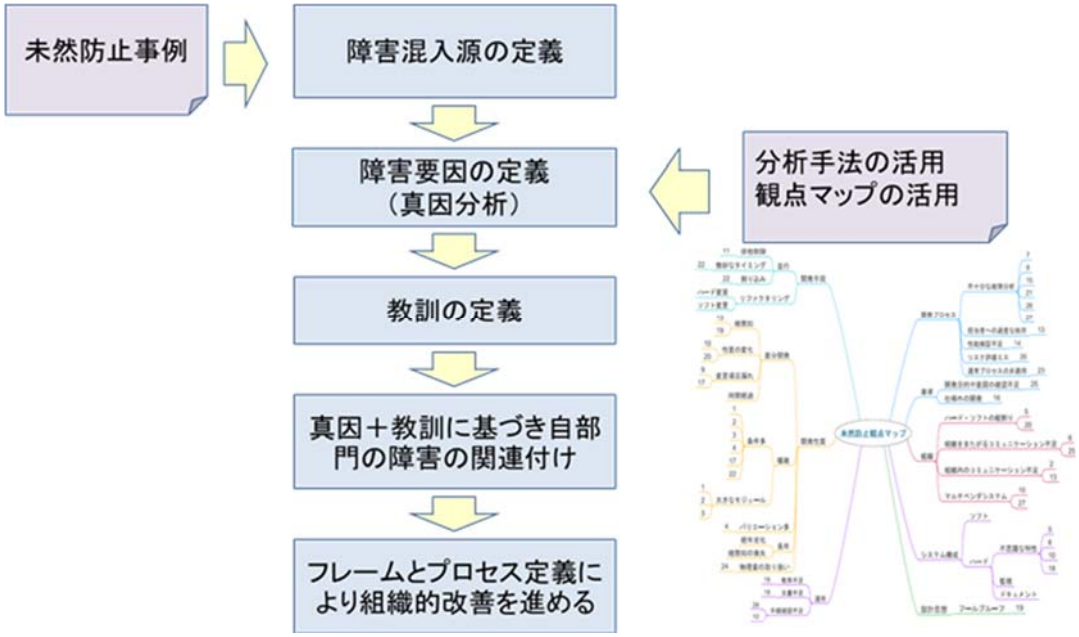


図 4-5. プロセス改善での活用例

真因分析の事例を図 4-6 示す。障害データに基づき障害が混入した工程及び流出・すり抜けた工程で障害を分類する。各工程で障害要因を分類して障害の真因を定義する。定義した真因を未然に防止するフレームと開発プロセスを定義する。

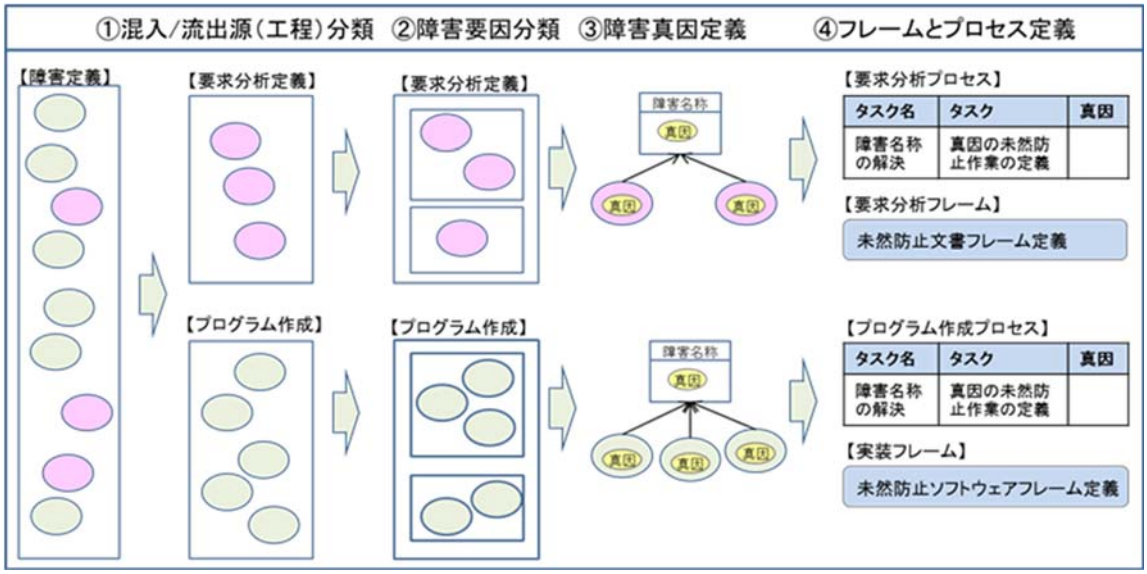


図 4-6. 真因分析での活用例

①混入源の分析の進め方

a)設計を厳密に定義しているか確認する

障害は、プログラムコードの想定外の動作により発生する。プログラムコードは、設計文書に基づき実装される。設計文書の曖昧性，誤り，漏れがあれば実装依存となり障害が発生する。

b)設計不備であれば分析を厳密に定義できているか確認する。

分析の曖昧性，誤り，漏れがあれば設計依存となり障害が発生する。

c)分析不備であれば要求を厳密に定義できているか確認する

要求の曖昧性，誤り，漏れがあれば無ければ分析依存となり障害が発生する。

②流出源（すり抜けた要因）の分析の進め方

a)要求に基づき分析文書がレビューできているか確認する。

b)分析文書に基づき設計文書がレビューできているか確認する。

c)分析・設計文書がテスト設計書に反映できているか確認する。

d)テスト設計書に基づきテストが実行されたか確認する。

③真因の分析の進め方

障害が発生した要因項目を階層的に定義する。各要因は、事実情報（設計書、ソースコード、計測データ、各種記録）に紐付けて真因を分析する。真因の検出を容易にするために真因カテゴリを使用する。真因カテゴリは、『規則』『方法』『要求』『資産』『時間』で構成される。以下に真因カテゴリと真因分析タスクの例を示す。

『規則』

・真因を未然防止するための規則が守れているのか確認する。

・未然防止が可能な規則か確認する。

『方法』

・真因を未然に防止する方法がフレームとプロセスで定義できているか確認する。

※未然防止の方法が無い場合は方法の開発を行いフレームとプロセスを定義する。

『要求』

・要求の変動、遅延。仕様品質が障害作り込みに起因していないか確認する。

『資産』

・既存資産やOSS含む流用ソフトウェアの制約や潜在不具合の影響はないか確認する。

『時間』

・過負荷な開発となりヒューマンエラーを誘発していないか確認する。

真因分析タスクを実行する際に観点マップを活用して新たな観点が見つかれば、真因分析タスクを追加していくことで対象組織に最適な真因分析を可能にする。

4.2.2 ソフトウェア改善プロセス活動への適用

➢適用例1

障害に基づくフレーム（様式およびソフトウェアフレームワーク）とフレームに実装するソフトウェア開発の作業手順をプロセスに定義することにより、障害に基づいた組織的なプロセス改善を実現させた事例を図 4-7 示す。

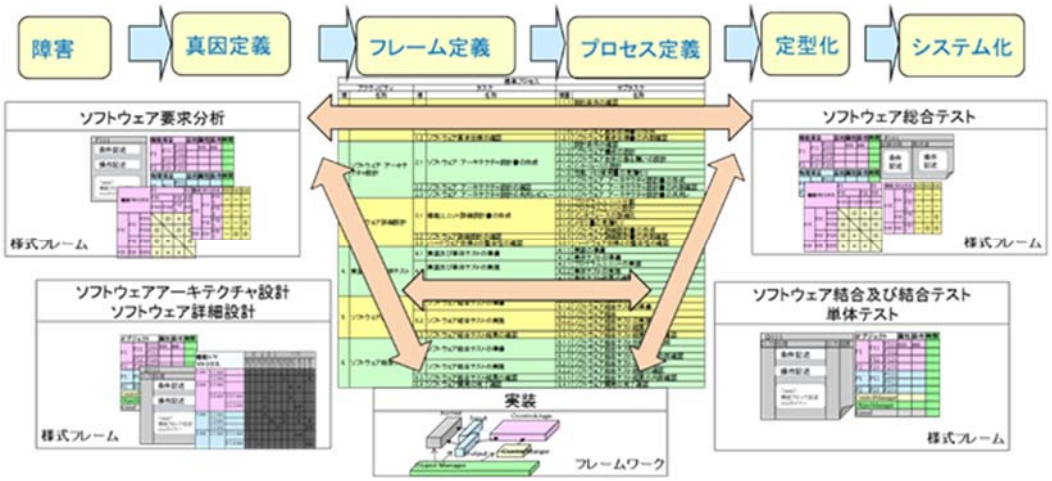


図 4-7. プロセス改善例

➢適用例2

障害に基づく開発プロセスの組織展開と育成の事例を図 4-8 示す。日常のソフトウェア開発で混入した障害を分析してフレームと開発プロセスを継続的に更新する。

フレームと開発プロセスの変化に対応した教育と開発プロセスの実行に必要なエンジニアリング技術教育を実施する。

更に成果物の確認と OJT を繰り返すことにより組織的な品質改善活動を実現させる。

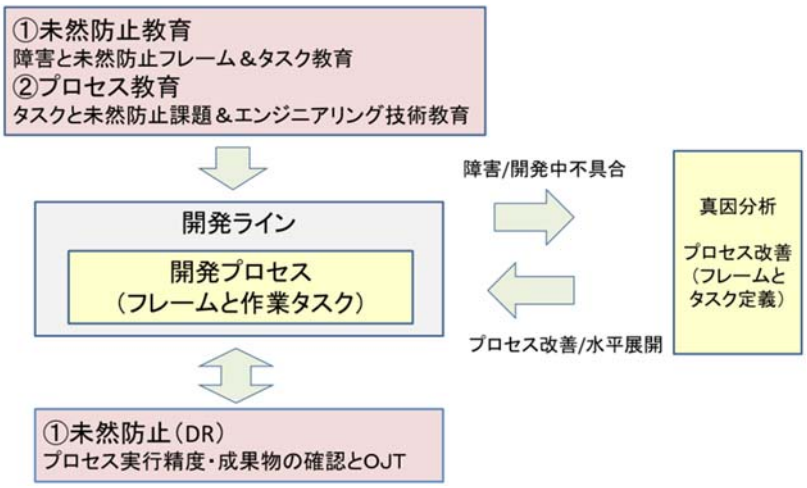


図 4-8. 開発プロセス組織展開・育成例

➢期待効果

障害の真因に基づくフレーム（文書様式およびソフトウェアフレームワーク）とフレームに実装する開発作業のプロセス定義により同種の障害は、未然に防止することが可能になる。経験の浅い技術者でも観点マップを用い真因分析カテゴリと真因分析プロセスの継続的な改善で真因定義を容易にして精度の高いプロセス定義が可能になる。

➢留意事項

人依存ではなく仕事の進め方（プロセス）の真因を定義する。
真因未達のよくある 3 つの記述パターンには以下のようなものがある。

『漏れてしまった』
『できなかった』
『不足した』

作業の進め方と成果物及び記録や計測データの現物に基づき、どのような作業を実施すれば『漏れなかったのか』『できたのか』『不足しなかったのか』を考える。

4.2.3 改善プログラムの効果的運用に向けた応用

（改善プログラム：改善のための活動実行計画）

➢想定される状況・課題

障害対応の経験に基づき、過去の対応情報を事例としてデータベースとして管理するなど再発防止に向けた組織的な取組みを行っているが、対策効果が限定的でさらに効率的で効果のある施策が必要となっている。

➢活用の狙い

効果のある対策を立案する際に、その有効性を判断するための指標として利用する。

➢活用方法

「未然防止観点マップ」を品質向上に向けた改善プログラム策定時に判断材料として活用する。自社、自部門の製品・システムにおいてどのような問題があるのか、分析結果の原因などがこうした改善プログラム体系に示された分類に従ってカテゴライズされることにより、改善のためにどこに集中投資すれば良いのかを可視化できる。その結果、組織の改善プログラムの策定を効果的に進めることができる

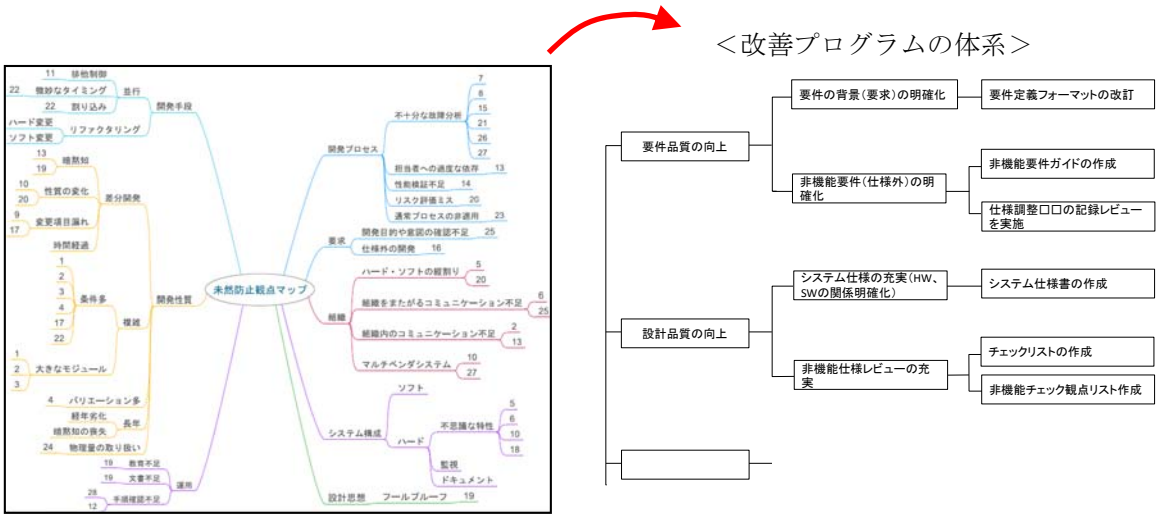


図 4-9. 改善プログラムへの適用

➢期待効果

これまでに発生した問題とその対応について、それらの原因が頻度や傾向などの統計情報として把握され、蓄積されることにより、未然防止や再発防止対策実行における有効性が明確になる。

➢留意事項

未然防止観点マップに従って発生した障害の原因分析が実行され、どの観点の障害が多いかといった集計が組織的に行われていることが前提となる。また観点としての網羅性、一般性が十分であるかに配慮を要する。

4.3 設計品質向上活動への活用

本節では障害発生時に行う再発防止対策あるいは未然防止に向けた設計品質向上の取組みに対し、本事例集がどのように活用できるかを示した。

4.3.1 障害要因の追求探索ガイドとして活用

➢想定される状況・課題

障害発生時にはなぜなぜ分析を行うが、原因を掘り下げてゆく際その観点次第で検討の方向性が変わり、結果に大きな影響が出る。また、分析の広さや深さは個人への依存性が高くなりがちであり、結果の安定性（一般性）を保つのが難しい。

➢活用の狙い

真因の追求の際に総花的な対策にならないよう、検討の方向性を与える。

➢活用方法

「未然防止観点マップ」を自部門で障害要因を追求してゆく際、検討の視点としてヌケ・モレがないか、偏りがないかなどをチェックするためのガイドとして活用する

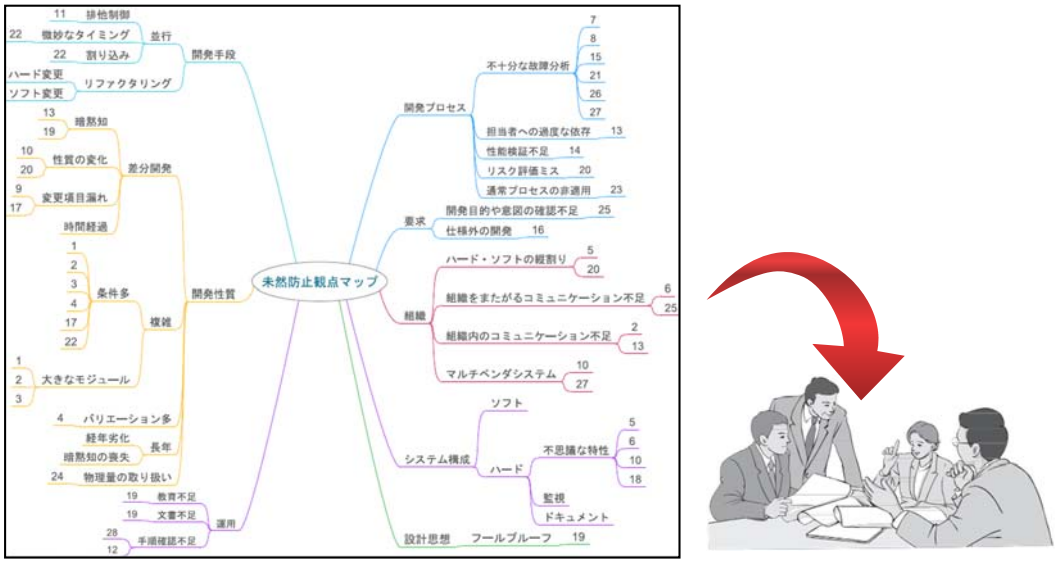


図 4-10. 設計品質向上活動への活用

➢期待効果

検討の方向性のバラツキを抑えることができ、同じような問題は同じような原因に帰着させやすくなる。その結果、未然防止や再発防止対策の投資対効果を高めることができる。

➢留意事項

観点としての網羅性、一般性が十分であるかに配慮を要する。自社、自部門の障害事例を観点マップに反映させてゆくといった継続的な取組みが大切である。

4.3.2 障害対応プロセスへの活用

➢想定される状況・課題

障害時には原因を調査し再発防止に努めているが、取扱い製品・システムのバリエーションも多く対応プロセスもまちまちになっている。このため、原因分析などに要する時間や顧客対応品質にも疎密が目立ち、個々の障害時対応プロセスについての見直しが必要となっている。

➢活用の狙い

障害対応プロセスを見直す際に他社、他部門取組み活動事例として参照・活用する。

➢活用方法

個別障害プロセスにおける改善を検討する場合に、PART I 教訓集中の類似障害例を参考にする。また PARTⅢ、Ⅳで示されている障害分析手法、他社再発防止活動事例などを自部門の障害対応プロセスを見直す際に利用する。例えば真因から再発防止策を導出する手順に対して、PARTⅣで示されている手法適用時の留意事項、追求時の観点ならびに他社事例の内容が自社においても有効であると考えられる場合には自社の仕組みに適した内容としてカスタマイズした上で反映する。

➢期待効果

組織として必要な対応プロセス・体制等を見直す際に他事例を参照することで、社内でのディスカッションを前向きに進めることができるとともに、あるべきプロセスを再考する上で新たな視点を得ることも可能となる。

➢留意事項

現状行われている製品・システム毎の対応ギャップが大きい場合、社内コンセンサスを得ながら再検討を推進してゆくための負荷は大きくなるため、プロジェクト体制など活動の進め方にも工夫を要する。

5 企業内での活動事例

本章では企業内で実際に発生した障害に基づき、再発防止や未然防止に向けた社内教育、品質マネジメント、開発プロセス管理の活動事例を示した。各事例の末尾には、教訓集で関連のある箇所を示した。

5.1 社内教育事例

社内教育例① （不具合発見実習教育）

・座学だけでなく、実習付きの教育を行っている。意図しない動作をするような不適合を含んだソフトウェアプログラムを用意し、受講生に発見させる内容となっている。
・参加者は面白がってやっている様子でありトラブルシュートは人気があり、役にたっていると感じている。（下記は実習ワークシート例）
・学習時間は１日コースから３日間コースまで内容に応じてさまざまである。
・実習でいくつバグがとれたかなどが指標になる。理解度テストである一定以上の得点がとれないと合格できない。若手や入社４～５年のエンジニアが受講している。

		不具合挿入箇所																							不具合発生部位						
		メッセージ													データ																
		① DERS01	② DACT01	③ DACT02	④ FUR9P01	⑤ FUR9P02	⑥ FUR9P03	⑦ NUMW01	⑧ NUMW02	⑨ NUMW03	⑩ LNKA01	⑪ LNKA02	⑫ LNKA03	⑬ FEMAK01	⑭ FEMAK02	① 出力値情報	② 軸回転速度	③ 排出量	④ 供給前圧値	⑤ 上限温度情報	⑥ 下限温度情報	⑦ N角偏向値	⑧ B角速度補正	⑨ B角速度基準	⑩ B角補正値	⑪ ベース圧力値	負荷	ゲージ表示 DP表示（流量）	ゲージ表示 DP表示（回転数）	ゲージ表示 DP表示（LED）	ALARM SV（L1）	ALARM SV（L2）	その他
メッセージ異常																															
	送信タイミング変更			●																											
		●																								●	●					
	TLINK ID		●																										●		●	
	TLINK ID			●																												
	不要メッセージ														●																●	
	不要メッセージ														●																●	

る	=1200ms							
・Link ID 設定ミス: F1: 2D1→2F1 F2: 2E1→2G1		✓					●	

図 5-1. 社内教育の例

関連 PART I　教訓 5 及び 10（ソフト技術者へのハード知識教育）、22（排他処理等基本概念の教育）

社内教育例②　（教育体系例）

・体得させるべき技術知識体系が定められており、その体系に基づいて不具合の学習を行っている。教育はコースごとに１～３日間かけて実施する。

・教育カリキュラム例：

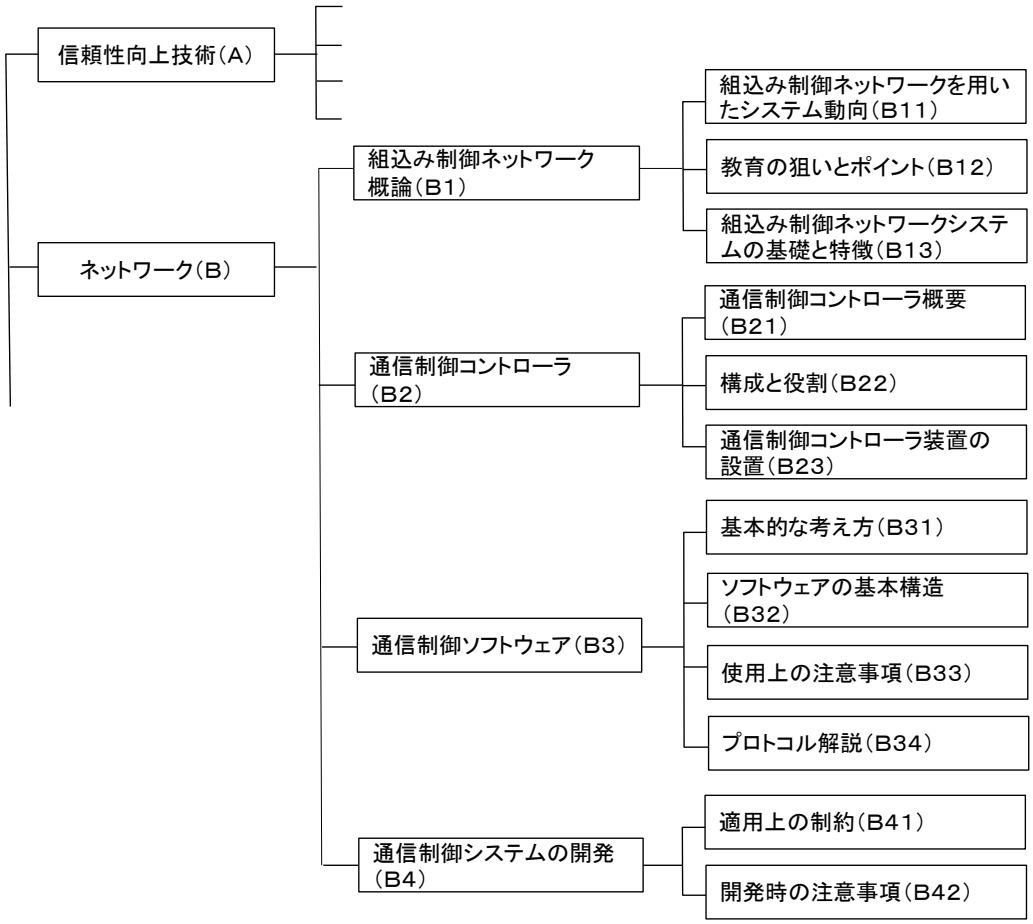


図 5-2. 教育体系の例

社内教育例③　（マインドマップの活用）

・品質を開発の上流で作りこむことの重要性を認識させるための工夫を入れた教育を行っている。例えばマインドマップによる設計観点マップなどを活用しながら、自社の品質に関する企業方針を繰り返しエンジニアに理解させる取組みを行っている。

社内教育例④　（事例紹介による共有）

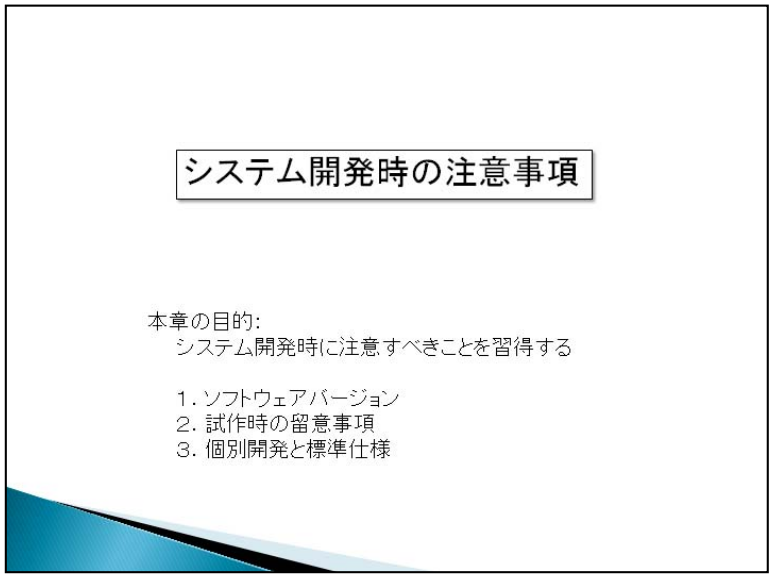
・品質教育含め、普段から品質向上への意識付けを実施している。
・他事業部でも発生し得る問題や重要な問題については事例紹介の形で発信・情報展開するための取組みを行っている。

社内教育例⑤　（ケーススタディ教育）

・チェックリストと不具合事例に関して、毎年エンジニア教育を実施している。
・各開発プロジェクトから１人以上は参加してもらうよう各プロジェクトマネージャに依頼している。
・参加者からはこうした教育の必要性は認識されており、また管理者側からもこうした教育を既定のカリキュラムに組み込んでほしい等前向きな反応を得ている。
・不具合の事例に基づくケーススタディ教育を本格的に展開するための取組みを推進中である。

社内教育例⑥　（未然防止教育資料例）

・過去に発生した障害対応の中から、同様の問題発生を未然に防ぐための教育を行っている。
・そのための教育資料を社内の教育担当部門が技術ジャンルごとに作成している。
・教育テキスト例：



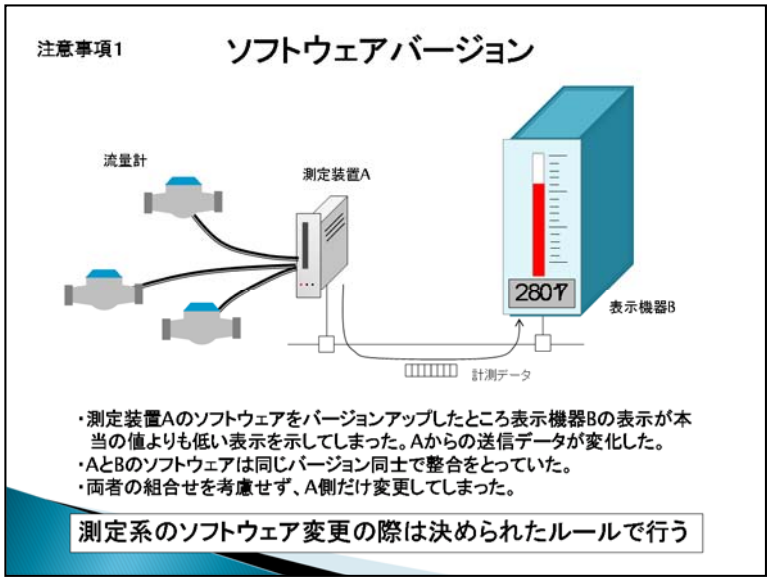
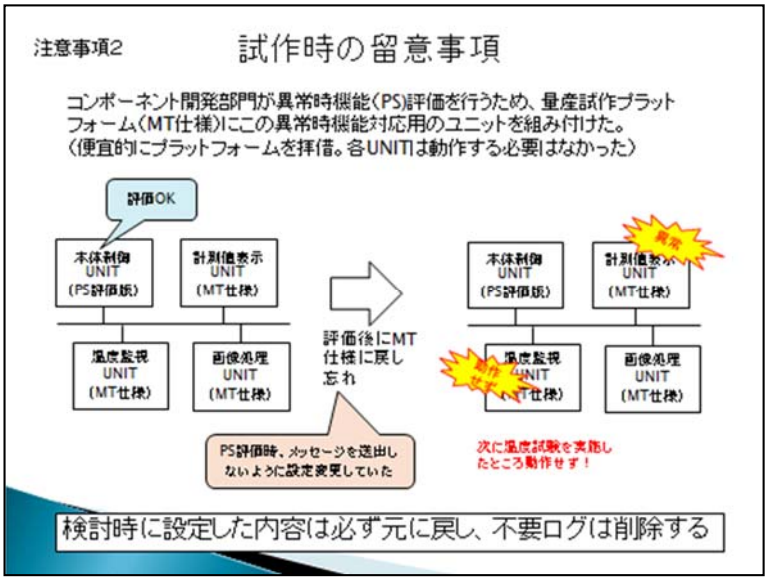


図 5-3a. 教育資料の例

（次頁へ続く）



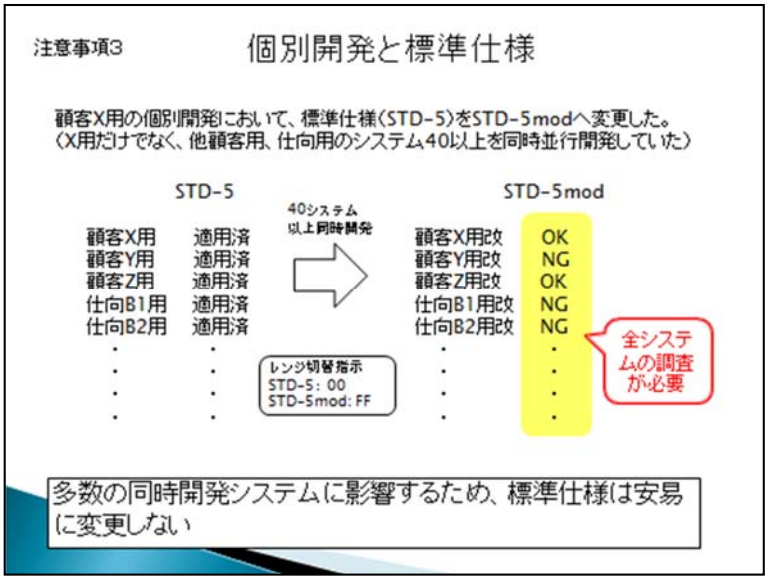
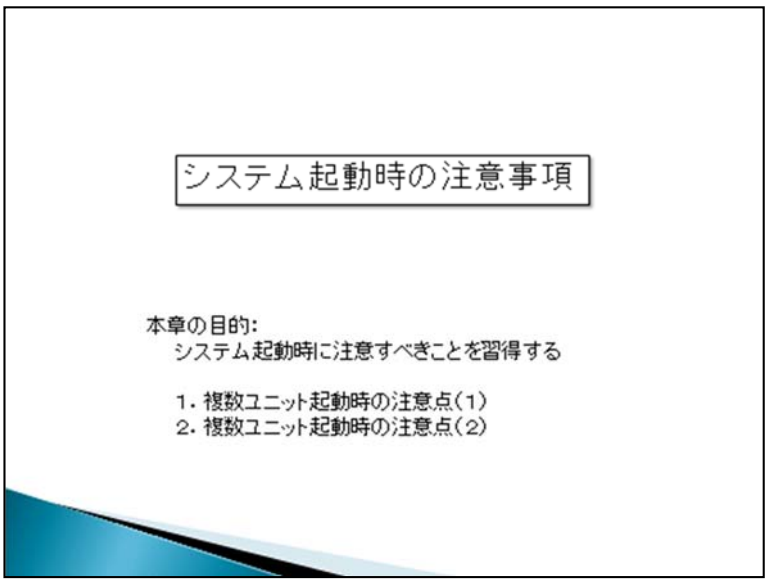


図 5-3b. 教育資料の例

(次頁へ続く)


システム起動時の注意事項
本章の目的:
システム起動時に注意すべきことを習得する
1．複数ユニット起動時の注意点(1)
2．複数ユニット起動時の注意点(2)

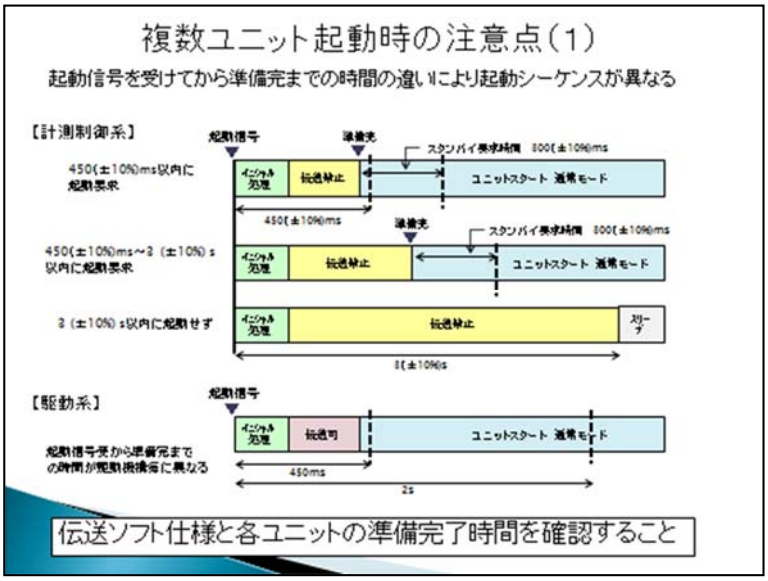

複数ユニット起動時の注意点(1)
起動信号を受けてから準備完までの時間の違いにより起動シーケンスが異なる
【計測制御系】
450(±10%)ms以内に起動要求
450(±10%)ms～3（±10%）s以内に起動要求
3（±10%）s以内に起動せず
【駆動系】
起動信号受から準備完までの時間が起動機構毎に異なる
伝送ソフト仕様と各ユニットの準備完了時間を確認すること

図 5-3c. 教育資料の例

社内教育例⑦　（参加者の失敗例を用いた実習教育）

・この例は、真因分析を“原因分析”と捉え、直接原因への対策後に実施する真因分析について、準備や進め方など、未然防止策や再発防止策の立案までのプロセスを組織的に教育するものである。
・始めに、分析の目的や要因と原因の定義など、分析における概念的な事柄、組織の標準的な分析手法であるなぜなぜ分析と、組織が推奨する分析手法である PNA（プロセスネットワーク分析）を座学で教育する。
・座学の後、学習者が持ち寄った失敗事例を題材に、講師あるいは学習者が分析者となり PNA を実践／演習する。
・学習時間は、持ち寄った失敗事例の数により、半日～１日の教育コースである。
・PNA の実践／演習では、分析者のヒアリングの仕方や意図を講師が解説しながら進めることで、学習者の理解がより深まることを狙っている。

教育コース名	内容	講習時間	受講対象者	前提知識	到達目標
原因分析	原因分析の準備，分析の進め方，再発防止策や未然防止策の立案までのプロセスを理解し，学習者が持ち寄った失敗事例を題材にPNA(プロセスネットワーク分析）法を学ぶ.	4	•プロジェクトで発生した品質問題を分析したい人 •原因分析力の向上を図りたい人	•学習者が持ち寄る失敗事例の概要	•原因分析の進め方や勘所，PNA法について説明できるようになる.

「原因分析」学習内容

章	学習項目	内容
1	原因分析の基本	分析活動，対象，目的，問題解決の定石，要因/原因，対策の立案，再発防止のポイント，など
2	なぜなぜ分析	なぜなぜ分析のポイント
3	PNA法	PNA法とは
		プロセス要素
		進め方と勘所
4	PNA法の実践	学習者の失敗事例を題材にしたPNA法の実演

図 5-4a. 実習教育の例

「原因分析」学習の進め方

座学　原因分析のこころ
弁解ではなく「事実を明らかに」
反省ではなく「次回の開発に活かす」
プロセスの書き出し方法　など

演習　講師による分析実演
分析者、当事者を二人の講師が務める
受講者も分析者として参加
受講者による分析実践
グループ毎に分析し、結果プレゼン、討議を行う

- 少人数(約10名)、受講者の階層のバランス
- 事前アンケートによるフィードフォワード
  原因分析への関り度合い、悩み等を把握
- 事後アンケートによる気付きの確認

図 5-4b. 実習教育の例

5.2 品質管理事例

今日多くの企業で品質管理（QC）を中心とした品質活動が行われており、その中で障害発生時の対応がルール化され、情報管理と共有の仕組みが構築されている。

品質管理例①　（報告書の記載事項）

・不具合発見時の対応手順は対応手順書として規定化されている。自社だけでなく協力会社も含めた対応フローとなっている。
・不具合現象を再現させ、発生条件を特定しその後、発生原因、発生確率を分析してから対策を検討する。
・不具合の作り込み原因と流出・すり抜け原因について、それぞれなぜなぜ分析を行い、得られた真因をもとに再発防止策を検討する。再発防止策を各開発工程移行時のレビューに使うチェックリストに反映している。
・チェックリストには、流出あるいはすり抜けた不具合だけでなく、社内のテスト工程で発見された不備で特に注意が必要であるとプロジェクトマネージャが判断したものも含まれる。
・発生時に作成する不具合報告書には、「発見された状況」、「発生する条件・確率」、「現象」、「原因」、「対策」、「対応日程」等を記載する。
・また再発防止策を検討する際には再発防止報告書も作成し、これには上記項目に加え、「作り込み原因」、「流出原因」「再発防止対策」、「対応日程」等を記載する。

品質管理例②　（全社的な品質保証体制）

・出荷した製品はその特性上長期にわたる市場対応が要求されるため、全社横断的な品質マネジメントを行う品質保証部門を中心に当該製品群の品質活動の運用が管理されている。
・個別の製品群ごとに具体的に定められた詳細な品質活動標準もありこれらは、上記品質保証部門の統制の元にその体系が定められ改訂などの見直しが定期的に進められている。

品質管理例③　（不適合情報管理）

・市場で不適合が発生した場合、その作りこんだ原因によっても対応が変わる。例えば設計段階に問題がある場合は、当該技術関係者でなぜなぜ分析を行いその検討結果は社内の品質情報データベースに格納保管し、一元的な情報管理を行っている。
・こうした情報は定期的な社内の TQM 活動などの場で報告されるなどの取り組みにより、関係部門で共有される。

品質管理例④　（チケットによる管理）

・全社レベルで不具合対応・品質改善のための枠組みがあり、各事業部門でさらに細分化して取り組んでいる。
・不具合を発生させた部門では、チケットを用いた各不具合の追跡を行いプロジェクトとして管理するようにしている。
・不具合対処の優先度は数段階のレベルに分けて管理している。このレベルは起草者の判断で設定し、チケットの転送先や閲覧者が必要に応じて変更を行う。記入者は、試験担当者の場合もあれば、ソフトプログラミング担当の場合や装置取りまとめ部門の人が行う場合もある。
・不具合情報はキーワードで検索できるようになっている。そのため不具合情報の記述粒度は結構詳細にわたるレベルとなっている。

品質管理例⑤　（要因の傾向分析）

・不具合の傾向分析を定常活動として進めている。
・機能面やどのタイミングで発生しているかなどの傾向、品質特性での層別分析などもしている。これらの分析結果はエンジニア教育にも反映させている。
・不具合事例はその発生のトリガとなった事象（例．ノイズ　等）、ソフトウェア内での原因（例．割込み干渉、処理タイミング考慮）などの観点で分類している。
・海外含めソフトウェア開発に携わる全ての部門で使用するチェックリストの共通化を図っている。またマイコンの機能毎（A/D 変換、タイマー、通信　等）にも分類整理している。

5.3 開発プロセス管理事例

開発プロセス管理例①　（陳腐化防止）

・不具合事象の対応策のうち設計工程で対応可能なものについてはチェックリストに反映し、デザインレビューのタイミングでチェックしている。
・不具合の真因を中長期的に分析し、再発防止に向け有益な教訓と考えられるものについては、ドキュメントとして記録保存し、社内で適宜教育を実施している。この際、内容の定期的見直しを行い時間経過に伴う陳腐化が起きないようにこころがけている。

開発プロセス管理例②　（チェックリスト＋α の工夫）

・チェックリストを用いた運用をしているが、このリストも膨れ上がりチェックリストのチェックというような工数も大きくなりややもすると見落としがちともなるので、直前のプロジェクトでの状況を把握することで網羅性を上げるといった工夫も行っている。
・バグ予測曲線によるコントロールも併用するなど基準に照らしてみて実際の乖離が大きい場合、テスト方法の見直しをする等その要因を追求するという管理をしている。
・またソフトウェアはモジュール化し再利用できるようにしている。

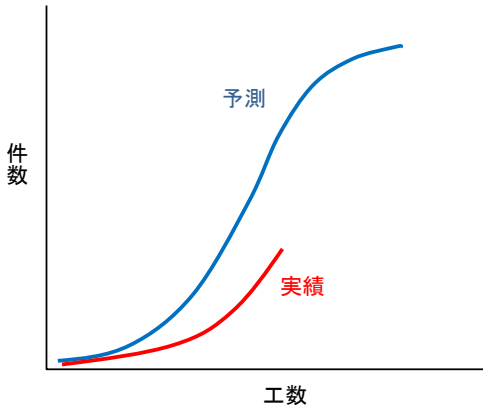


図 5-5. バグ予測曲線

開発プロセス管理例③　（チェックリストのメンテ）

・チェックリストのチェック項目は年々増大している。過去に登録された項目は、当時のことを知っている人も少なくなっていることもあり、登録された根拠が現状の技術、開発の状況からみて詳細吟味を要するものもある。
・記述の精細度、抽象度などのレベルが揃わないこともあるため、チェックリストの内容の見直しを行っている。

参考文献

[1]情報処理システム高信頼化教訓集（組込みシステム編）2015 年度版　独立行政法人技術本部　ソフトウェア高信頼化センター（平成 28 年 3 月）

[2] 障害未然防止のための教訓化ガイドブック（組込みシステム編）独立行政法人技術本部　ソフトウェア高信頼化センター（平成 28 年 3 月）

[3] ESPR :Embedded System development Process Reference
（SECBOOKS：ESPR Ver.2.0：【改訂版】組込みソフトウェア向け　開発プロセスガイド）
http://www.ipa.go.jp/sec/publish/tn07-005.html

[4] ESDR :Embedded System development Design Reference
（SECBOOKS：組込みソフトウェア向け設計ガイド［事例編］）
http://www.ipa.go.jp/sec/publish/tn12-003.html

[5]向後　千春, インストラクショナルデザイン ― 教えることの科学と技術 ―【2012 年版】, http://kogolab.chillout.jp/textbook/2012_ID_text.pdf, 2012.

[6]稲垣　忠, 鈴木　克明, 授業設計マニュアル Ver.2: 教師のためのインストラクショナルデザイン, 北大路書房, 2015.

[7] ESMR :Embedded System development Management Reference
（SECBOOKS：ESMR Ver.1.0：組込みソフトウェア向けプロジェクトマネジメントガイド［計画書編］）
http://www.ipa.go.jp/sec/publish/tn05-010.html

[8] ESTR :Embedded system development Testing Reference
（SECBOOKS：組込みソフトウェア開発における品質向上の勧め［テスト編～事例集～］）
http://www.ipa.go.jp/sec/publish/tn12-004.html

執筆者

【未然防止知識 WG】

主査	久住　憲嗣	国立大学法人九州大学
	内平　直志	国立大学法人北陸先端科学技術大学院大学
	石川　学	横河電機株式会社
	石原　鉄也	矢崎総業株式会社
	岩橋　正実	三菱電機メカトロニクスソフトウエア株式会社
	植武　信弘	株式会社日立産業制御ソリューションズ
	木村　裕之	日本電気株式会社
	鈴村　延保	アイシン・コムクルーズ株式会社
	高木　徳生	オムロンソーシャルソリューションズ株式会社
	土山　欽也	日本電気株式会社
	羽田　裕	日本電気通信システム株式会社
	細谷　伊知郎	トヨタ自動車株式会社

（50 音順）

	三原　幸博	独立行政法人情報処理推進機構
	十山　圭介	独立行政法人情報処理推進機構
	松田　充弘	独立行政法人情報処理推進機構
	石井　正悟	独立行政法人情報処理推進機構
	石田　茂	独立行政法人情報処理推進機構

監修

製品・制御システム高信頼化部会