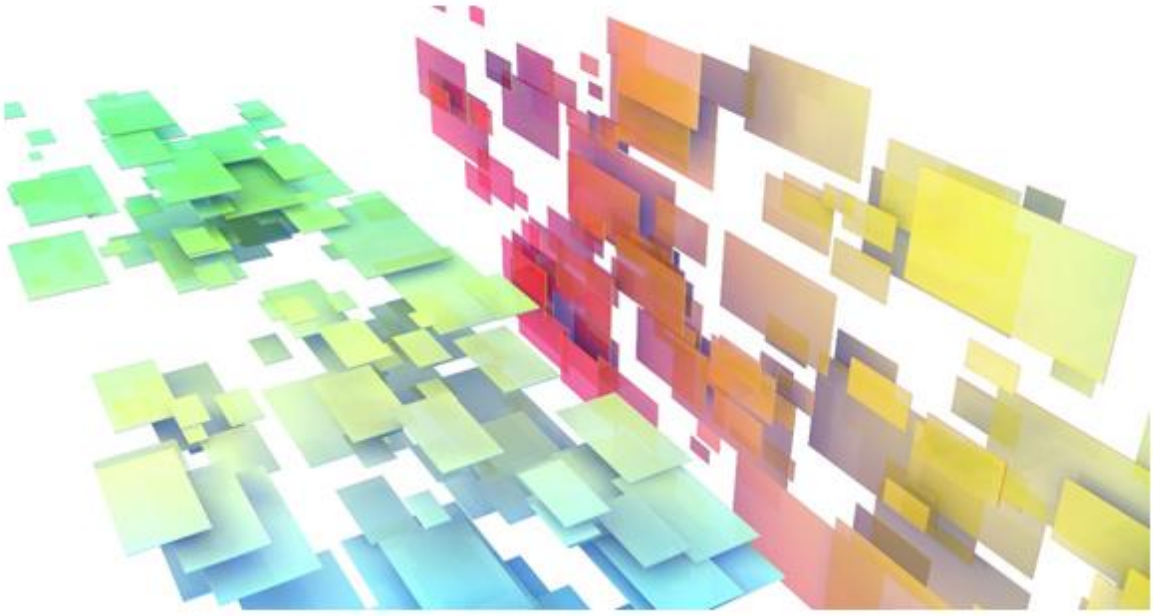
IPA Better Life with IT

情報処理システム高信頼化教訓集

2015年度版

(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）

情報処理システム高信頼化教訓集（組込みシステム編）

独立行政法人情報処理推進機構

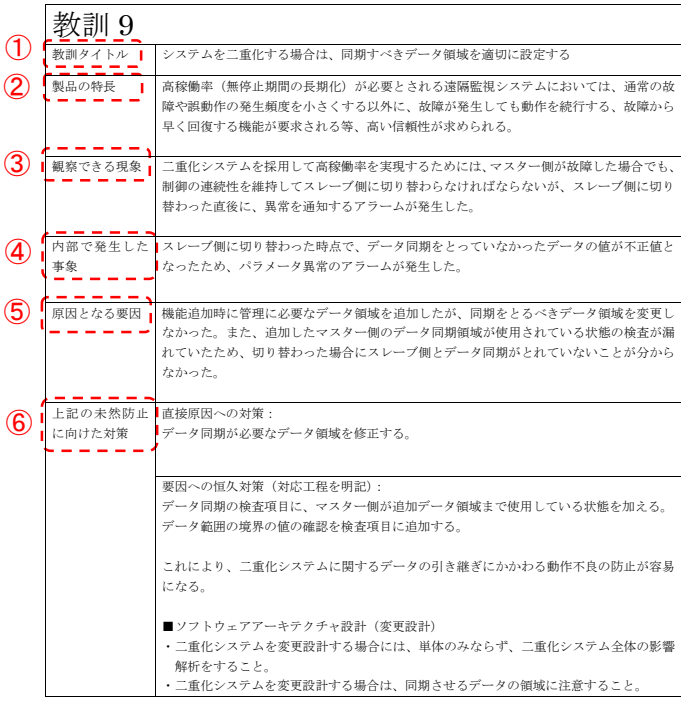
本書の構成と使い方

本書は、次の 4 つの PART と付録から成る：

PART Ⅰ　教訓集（本編）
PART Ⅱ　障害対策手法・事例集
PART Ⅲ　障害分析手法・事例集
PART Ⅳ　障害分析手法事例解説書
付録 A　障害情報の取扱いルール
付録 B　信頼性を表す評価指標

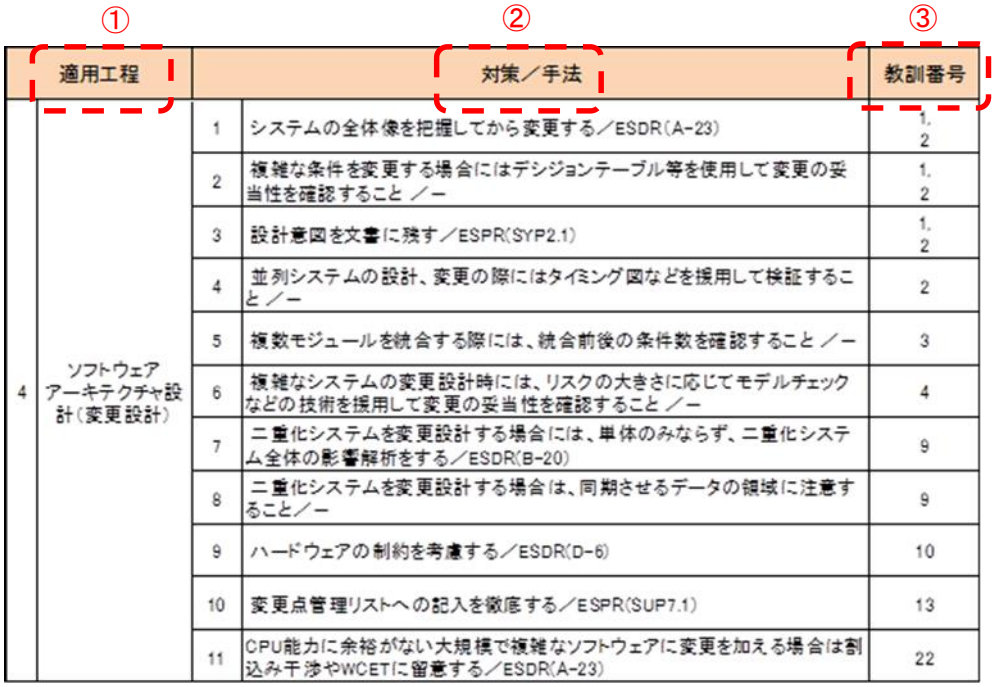
PART Ⅰ　教訓集（本編）は、情報処理システム高信頼化教訓集（組込みシステム編）の本体である。PART Ⅰ には 2013 年度から収集された 35 の教訓事例が掲載されており、各教訓事例は次図のようなシートで整理されている。ここではそのシート表記構成とポイントを以下に説明した。

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：障害が発生した製品、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②～⑤で抽出された要因の対策を記述。



	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
②	製品の特長	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。
③	観察できる現象	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。
④	内部で発生した事象	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。
⑤	原因となる要因	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。
⑥	上記の未然防止に向けた対策	直接原因への対策： 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）： 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。 ■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。

PART Ⅱ　障害対策手法・事例集は、教訓集中の各教訓を実践するために必要な手法を整理したものである。教訓に含まれる具体的な対策を実施するに際して、対策の必要な背景等をより深く理解するために役立つ。また、教訓に含まれる対策以外の周辺対策・関連対策を検討する際の参考としても活用することができる。



	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 ／ー	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援用して検証すること ／ー	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 ／ー	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援用して変更の妥当性を確認すること ／ー	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／ー	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

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に記述された対策・手法および該当する教訓番号を記述

②対策・手法：PART Ⅰ の各教訓の「⑥上記の未然防止に向けた対策」の欄には各対策を行うべき工程（対応工程）が明記されており、対応工程が上記①の適用工程と一致する対策内容をこの欄に転記する。また、ESDR、ESPR 等の既存 SECBOOK（PART Ⅰ の参考文献[1]～[4]）に関連する内容がある場合はその該当項目番号を記述

③教訓番号：②に転記した対策が含まれる教訓番号を記述

PART Ⅲ　障害分析手法・事例集は、今回の活動の中で実施した、障害の分析手法・事例の調査結果をまとめたものである。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した手法を、ステージごとに概説している。原因分析に利用した手法と手法の適用を検討した事例は以下の通りである。

【原因分析の手法】

ブロック図、事故経過表、VTA (Variation Tree Analysis)、問題行動分析、PNA（プロセスネットワーク分析法）、発生源・検出漏れ分析、例外分析、なぜなぜ分析

【分析手法を試行適用した事例】

・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）

PART Ⅳ　障害分析手法事例解説書は、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下記のような有識者からいただいた意見も取り入れて解説したものである。障害発生時の原因分析、手法の適用手順の参考として活用できる。

【有識者からいただいた意見の例】

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
：

付録 A　障害情報の取扱いルールでは、障害情報を報告・記録する共等通様式と、それらの収集・公開に際しての機密保持等のルールをまとめたものである。本書の読者が自ら障害情報の収集と分析、それに基づく教訓の共有を行う際に、参考として活用して頂きたい。
付録 B　信頼性を表す評価指標は、産業分野毎に評価指標が定められている例を示すものである。

全体目次

序言

世界中のすべてのモノがネットワークで繋がる IoT (Internet of Things)の時代が到来しつつある。IoT は今まで捕捉できなかったモノの情報の収集と活用により、産業や生活のムリ・ムダ・ムラを大きく削減し、新しい価値を生み出すことが期待されている。このように IoT への社会の期待は高いが、すべてのモノがネットワークで繋がってしまった世界において、モノを制御するソフトウェアの障害のインパクトは想像以上に大きく、その信頼性をどのように担保するかは極めて重要かつ深刻な課題となるであろう。IoT の活用が必須となる自動車の自動運転システムなどはその典型例である。生産性の観点でソフトウェア危機が叫ばれたのは１９６０年代であったが、人類は様々な技術革新により生産性のソフトウェア危機を乗り越えることができた。しかし、今後は信頼性に関するソフトウェア危機を乗り越えていく必要がある。

そのような迫りくる危機に対応するために、情報処理推進機構（IPA）の SEC（ソフトウェア高信頼化センター）では、重要インフラ等の製品・制御システム[1]の障害事例情報の収集・分析と対策の整理・体系化等を、問題意識を持つ企業等の協力を得ながら推進してきた。本書は、部会およびワーキンググループで検討してきた 3 年間の整理・体系化の成果をまとめたものであり、事例も昨年度から 7 事例増えている。

障害事例および対策の整理・体系化ができても、その成果が障害の再発・未然防止に活用されなければ意味がないが、実はこれが簡単ではない。これまでも、各企業では過去の障害事例をデータベース化することは行われてきたが、せっかく蓄積したデータが活用されていないという話はよく耳にする。整理・体系化した事例を、組織として学習し活用する仕組みの構築が必要である。そこで、今年度は、「現場で役立つ教訓活用のためのガイドブック」および「障害未然防止のための教訓化ガイドブック」を作成するとともに、教訓集を用いた技術者教育を開発し２回の試行を行った。今後も、ガイドブックの充実と教育の洗練化を行っていく予定である。

IoT の時代の想定外の障害を未然に防止するためには、単なる過去事例の学習ではなく、過去の事例から「気づき力」をどのように養成するかが重要となる。これは、知識科学における知識継承や組織学習の重要な研究テーマでもある。知識科学の最新の知見も取り入れながら、このチャレンジングな課題に取り組んでいく。

2016 年 3 月

組込みシステム高信頼化部会
主査　　内平　直志
北陸先端科学技術大学院大学
知識科学研究科　教授

[1] 本書の 2013 年度版および 2014 年度版では対象とするシステムを「製品･制御システム」と呼称していたが、2015 年度版より「組込みシステム」の呼称を使用する。

ＰＡＲＴ　Ｉ

教訓集・本編

（組込みシステム編）

PART　I　目次

1.はじめに

1.1　背景と目的

組込みシステムは、私たちの生活や社会の隅々にまで広まり、重要なインフラとして必要欠くべからざるものになっている。一方、システムの複合化によりシステム全体の信頼性を守ることが困難になりつつある。

従来、個々の製品の信頼性を損ねる要因に関する分析と対策は、個別に行われ公開されることはなかった。そのため、別の製品あるいは業界において、特に事前の検証やテストでは発見の難しい原因によって引き起こされる類似の障害が発生することがあった。

また、製品を全く新規に開発することが少なくなり、製品の信頼性に関わるノウハウを活用する機会も無くなりつつあることから、企業の経験を世代間で共有・伝承する手立てが必要になりつつある。

製品の信頼性を維持するためには、各企業等の経験と情報を企業・業界・世代横断で共有することが大切である。そのためには、各企業等が個別に保持する経験とノウハウを第三者が理解し実践できる形に要約する必要がある。そこで独立行政法人情報処理推進機構（以下、IPA/SEC）では、重要インフラ[2]等の組込みシステムの開発・構築を担う企業等の協力を得ながら、障害事例情報の収集・分析と対策の検討、既に実施済の対策を含めた整理・体系化等を行うこととした。この取組みは、その結果得られる「教訓」を業界・分野を越えて幅広く共有し、類似障害の未然防止・影響範囲の縮小につなげる仕組みの構築を目指すものである。（図 1.1 参照）

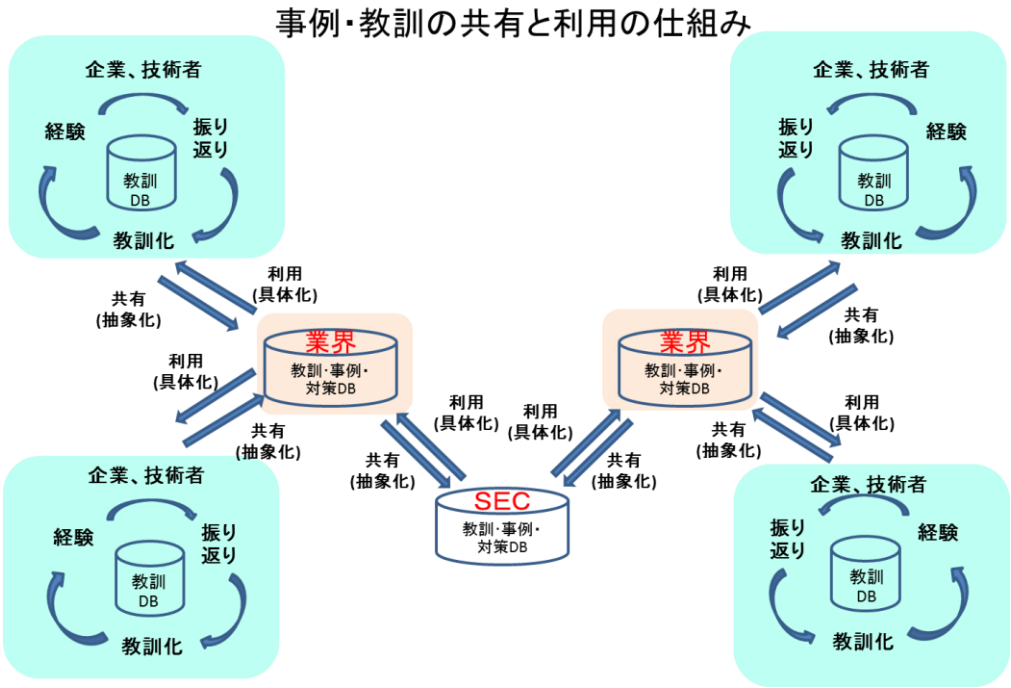


図 1.1　知識の共有と活用

[2] 内閣サイバーセキュリティセンターでは「情報通信、金融、航空、鉄道、電力、ガス、政府・行政サービス、医療、水道、物流、化学、クレジット、石油」の 13 分野を重要インフラに定義している。（平成 26 年 5 月 19 日「重要インフラの情報セキュリティ対策に係る第 3 次行動計画」）

1.2 IPA/SEC の取組みの意義

製品の信頼性を維持するためには、各企業等の経験と情報を分野・世代横断で共有することが大切である。そのためには、各企業等が個別に保持する経験とノウハウを第三者が理解し実践できる形に要約する必要がある。そこで IPA/SEC では、重要インフラ等の組込みシステムの開発・構築を担う企業等の協力を得ながら、障害事例情報の収集・分析と対策の検討、既に実施済の対策を含めた整理・体系化等を行うこととした。この取組みは、その結果得られる「教訓」を業界・分野を越えて幅広く共有し、類似障害の未然防止・影響範囲の縮小につなげる仕組みの構築を目指すものである。

一般には企業から生の障害情報を提供してもらうことはほとんどできないため、IPA/SEC では機密保持等に関する所定のルール（国家公務員法、IPA 委員会規定等）に則り、公的機関の立場として企業内で一次分析された情報（対策を含む）を提供いただいた。その情報を一般化・抽象化して教訓を「情報処理システム高信頼化教訓集（組込みシステム編）」として取りまとめた。

1.3 本教訓集の特徴

組込みシステムでは、利用者や利用環境を事前に特定することが困難な場合が多く、事前の検証やテストでは発見の難しい要因によって障害が引き起こされることがある。

しかし、個々の製品の信頼性を損ねるこのような要因に関する分析と対策は、企業ごとに個別に行われ公開されることはなかった。そのため、別の製品あるいは業界において、このような事前の検証やテストでは発見の難しい要因によって引き起こされる類似の障害が発生することがあった。本教訓集では、このような障害の発生を未然に防止するための教訓を収集した。

また、製品を全く新規に開発することが少なくなり、製品の信頼性に関わるノウハウを活用する機会も無くなりつつあることから、企業の経験と知識を世代間で共有・伝承する手立てが必要になりつつある。本教訓集はこのような知識の伝承も視野に入れて作成した。

2.情報処理システム高信頼化教訓集（製品・制御（組込み）システム編）

2.1　未然防止知識収集の方針

2.1.1　概要

本教訓集は、産業界で実践されているシステムの品質上の問題を未然に防ぐための未然防止知識をもとに、それらを抽象化、一般化することによって、幅広い組込みシステム開発企業において利活用できるための教訓として整備したものである。

ここで、品質上の問題を未然に防ぐことができる知識のことを未然防止知識と呼ぶ。

未然防止知識には再発防止と未然防止のための知識との 2 種類がある。再発防止のための知識は、同分野、同企業、同組織が起こした事故、障害に関する知識をもとに再度類似の問題の発生を防ぐための知識である。また、未然防止のための知識とは、他分野、他企業、他組織が起こした事故事例を抽象化、一般化することによって、自分野、自組織において類似の問題を防ぐための知識である。

また、未然防止知識には、内在的な障害を取り除くための品質向上のための知識と、内在的な障害が発現したとしても問題として外化するのを防ぐフォールトトレランスのための知識があるが、両者とも収集の対象とした。

なお、障害、故障、欠陥などの用語は、様々な意味で使用される。本教訓集では障害という用語は JIS X 0014 に定義される障害(fault)の意味で用い、故障(failure)や欠陥(defect)は IEEE 1044 の定義や IEEE 982.1 の記載内容に基づく用語を用いる。障害は計算機プログラム内の不正確なステップ、プロセスまたはデータの定義とする。故障は要求された機能を遂行する製品の能力が尽きる状態、または事前に仕様化された制限内での機能を遂行する能力が無い状態とする。

2.1.2　未然防止知識の想定利用者

未然防止知識の利用者としては組込みシステム開発に関わる以下の 3 者を想定する。

- ソフトウェア設計者
- ベンダー側システム設計者
- ユーザー側システム設計者

なお、システムを利用する特別な訓練を受けていないエンドユーザーは、知識の利用が困難であると考え、想定利用者からは除外してある。

2.1.3　収集、整理手順

未然防止知識教訓化については、それぞれの企業活動の中で実施することが望ましい。ここでは、そのための未然防止知識の収集、整理等の手順と方法の一例を紹介する。

[収集手順の背景]

未然防止知識の収集、整理を実施するにあたり、収集した各企業内外の事例や知見をそのまま未然防止知識として収録するのではなく、肝となる部分を抽出し、別の事例に書き換えたものを未然防止知識とする。これは、これらの事例や知見をそのまま公開情報とすると、当事者にとっては事例や知見の開示が困難であることが予想されるためである。さらに、組込みシステムの製品分野は多岐にわたるため、同一の企業内であっても他の分野の事例や知見では実感しにくいため、なるべく利用者の分野に近い事例に書き換えて未然防止知識としたいと考えるためである。

[収集の枠組み]

未然防止知識の収集、整理プロセスは繰り返し実施する過程において改善すべきである。これは未然防止知識の収集、整理を通して、各企業で活用できる未然防止知識の収集、利用プロセスを確立するためである。収集プロセスを確立するためには、未然防止知識の分類、フィルタリング、肝となる知識の抽出、抽象化、事例の書き換えの方法を整備する必要がある。これらの方法を一挙に確立することは難しく、収集、整理プロセスを繰り返して実行し、修正を図る必要がある。

[具体的な収集手順]

上記の背景を踏まえて未然防止知識は以下の手順で収集する。

1) 未然防止知識シートを使用したヒアリング

後述する未然防止知識シートを使用し、関係者にヒアリング、または、直接記入させる。この段階では生の情報であり他の事例への書き換えは行わない。

2) 抽象化、他分野、他事例への書き換え

未然防止知識としての重要なエッセンスを残して抽象化し、他分野、他事例への知識の書き換えを行う。

3) 工程別の未然防止知識の抽出

工程ごとの対策と、対策しなかったときにどのような問題が発生しうるのかを一覧にするために、工程別の未然防止知識を抽出する。

4) レビュー

未然防止知識を有識者によりレビューし、記述の過不足、理解が難しいところ等を補足する。

5) 修正

レビューの結果に基づいて修正する。

[未然防止知識シートの設計]

未然防止知識シートは、知識の提供者に記入してもらうために使用するシートである。一次情報として必要な情報を過不足無く記入してもらうためにシートとして構成する。なお、未然防止知識シートは後述する記載項目を参考に各ドメインの特性も考慮して設計することが望ましい。

図 2.1 に未然防止知識シートの設計に関わる概要図を示す。図 2.1 にあるような手順で順番に整理していくことによって、未然防止知識シートが完成する。

まず、最初に問題が発生したときの状態を整理する。問題が顕在化したときに直接観察できる、システムが暴走する、システムが不意に停止する等の現象をまず整理する。次にその現象を直接発生させた内部で発生した事象を整理する。

次にその問題を作り込んだ直接的な原因を整理する。この原因には開発技術に関わる技術面、人・組織面等がある。

その次に原因に対してどのように恒久的な対策を行うのかということを検討する。

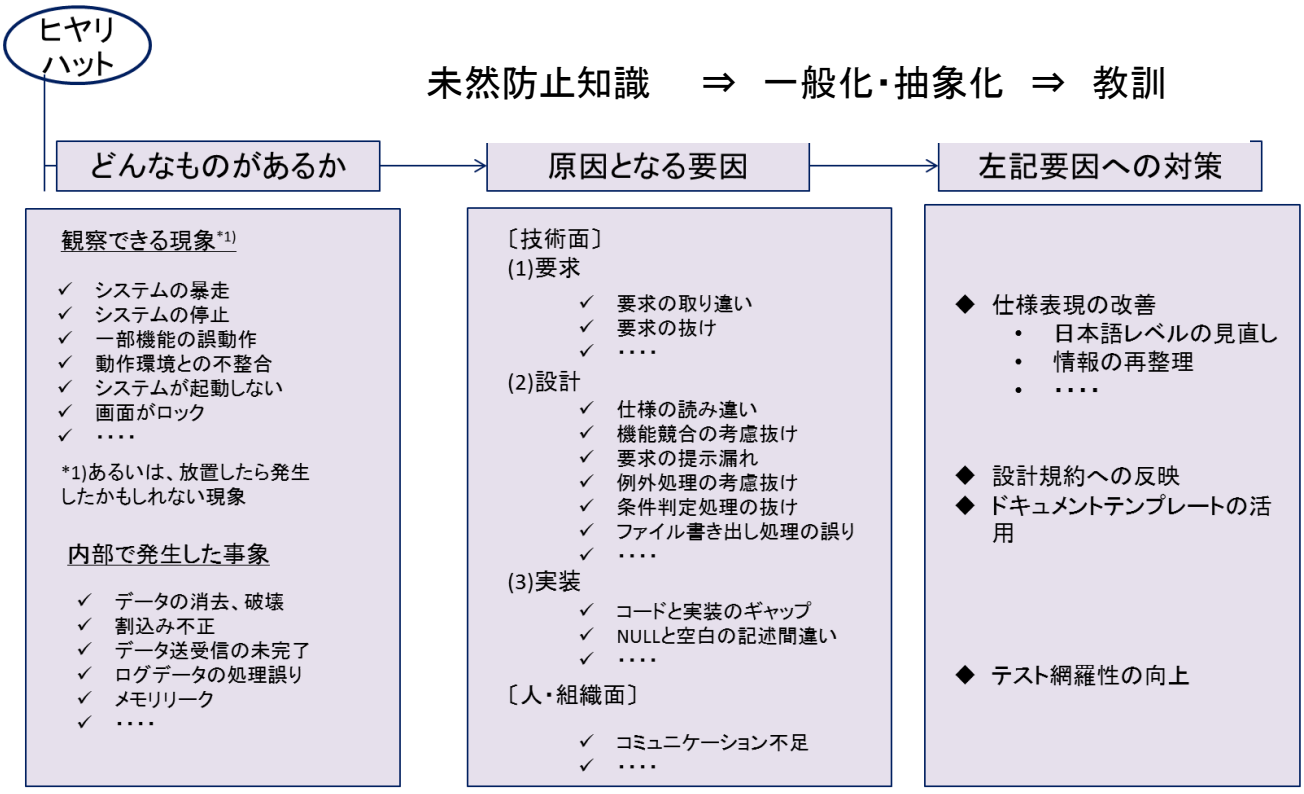


図 2.1 未然防止知識シートの設計方針

[未然防止知識シート記載項目]

前記した未然防止知識シートの設計に基づき、以下の項目を考慮して未然防止知識シートを作成する。

- 教訓タイトル
- 背景のキーワード
- 製品の分野
- 製品の特徴
- 観察できる現象
- 内部で発生した事象
- 原因となる要因
- 上記の未然防止に向けた対策

設計のところで前述した項目（観察できる現象、内部で発生した事象、原因となる要因、上記の未然防止に向けた対策）以外に、未然防止知識の背景を記述する項目、工程別の未然防止知識を追加する。

未然防止知識の背景は、取り扱う製品分野における信頼性要求の度合い、ハードウェアの特徴等により、未然防止知識における対策等が大きく異なると考え、追加してもよい。

また、工程別の未然防止知識については、どの工程においてどのような対策を施せば、未然防止知識シートに記載している問題が発生しないのかを整理するために追加する。

[工程別未然防止知識整理の方針]

工程ごとの対策と、対策しなかったときにどのような問題が発生しうるのかを一覧にし、工程ごとに未然防止知識を俯瞰できるようにするために、工程別に整理する。工程ごとのそれぞれの未然防止知識から、もとの未然防止知識をたどれるように整理する（図 2.2）。

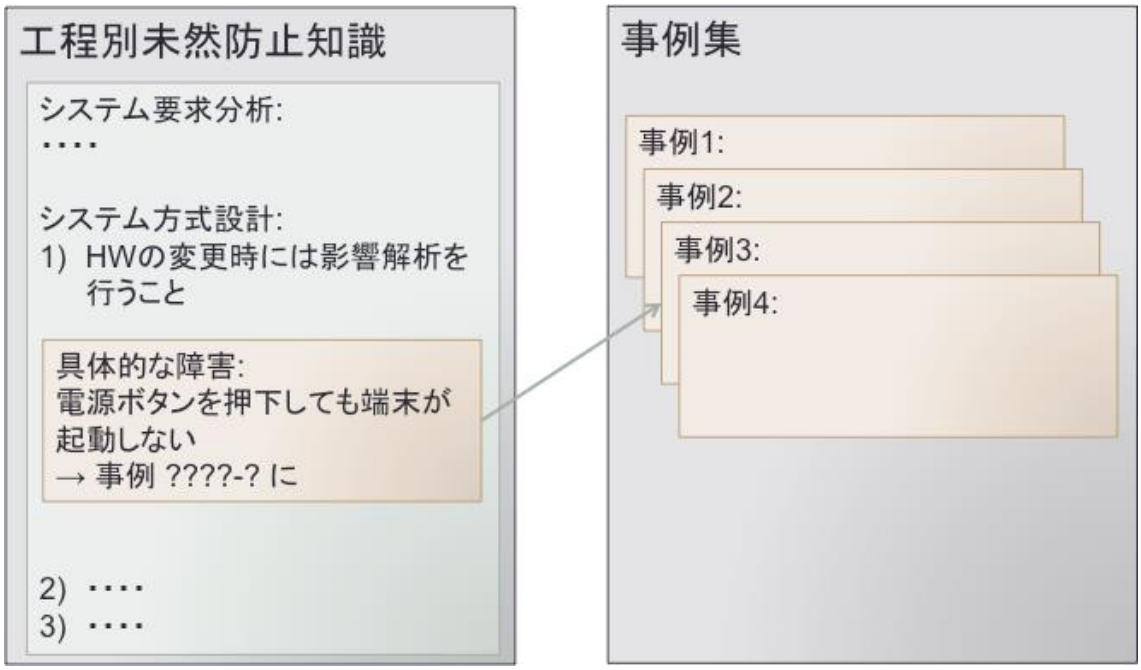


図 2.2 工程別未然防止知識と事例との対応関係

工程別未然防止知識の整理に当たり、この例ではプロセスモデルは「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」[1]のモデルを基本的には採用している。また、直接開発に関わるシステム・エンジニアリング・プロセス（SYP）やソフトウェア・エンジニアリング・プロセス（SWP）だけでなく、サポートプロセス（SUP）もできる限り対象とすることが望ましい。なお、ESPR では教育に関する工程が定義されていないが、これも工程別未然防止知識の一部としては採用を検討すべき項目であるといえる。また、組込みシステム開発において多く見られる差分開発特有の未然防止知識については、各工程に差分開発である旨、特記することが望ましい。

2.2　教訓

本章では収集した未然防止知識事例を抽象化した教訓を紹介する。

教訓 1	
教訓タイトル	複雑な条件式のロジック変更を行う場合は、デシジョンテーブル等による検証が有効である
製品の特徴	空気圧で非常用ドアの開閉を行うシステムで、開閉のための圧力を一定に保ち、速度や各種インターロック等多数のセンサー入力に基づいて開閉動作する。非常時動作の確実性が求められる。
観察できる現象	試験運用中、いくつかある非常用ドアが開かないという事態が露見し、当該システムの信頼性だけでなく製造メーカの信用をも損なうこととなった。
内部で発生した事象	調査したところ、本ドアの開閉を行うための空気圧を制御するロジックにおいて矛盾する条件が設定されていることがわかった。
原因となる要因	本システムの非常ドア開閉のためのコンプレッサー駆動電源は電源効率を考慮し他系統と融通し合うようになっているが、ソレノイド余熱用の電流もこの系統から得ている。変更前は以下のように余熱用電流の確保に関する条件が設定されていたのだがある時電源構成が変更になりこの条件を削除することになった。　ところが、担当者は当該部分だけを単純に削除すればよいとの思い込みで単純に 1 行削除したため、変更後パターンが全体要求仕様を満たしていないことを認識できなかった。 変更前： （前提条件） &&　（空気圧が設定値以上） &&　（（●電流要求無） ||（速度センサー異常検知） ||（■温度センサー異常検知）） 変更後の不具合状態： （前提条件） &&　（空気圧が設定値以上） &&　（~~（●電流要求無）~~ ||（速度センサー異常検知） ||（■温度センサー異常検知））

<table>
<tr><td></td><td>対策後：
（前提条件）
&& （(速度センサー異常検知）
||（■温度センサー異常検知）
||（空気圧が設定値以上））</td></tr>
<tr><td>上記の未然防止に向けた対策</td><td>直接原因への対策：
・当該条件式を確認の上、ロジックの修正を実施</td></tr>
<tr><td></td><td>要因への恒久対策（対応工程を明記）：
複雑な設定条件を変更する場合には、ロジックに矛盾や不適合がないかを、以下のようにデシジョンテーブルを作成して確認する。元の条件パターンから削除されたもの、追加されたものを色分けする等により比較確認することで気づきを得ることができ、未然に誤りを防止することが容易になる。
変更前：</td></tr>
</table>

			条件式	成立条件				
				#1	#2	#3	#4	#5
&&			（前提条件）	○	○	○		
&&			空気圧 >= 設定値	○	○	○		
&&	||		●電源要求 == FALSE	○				
	||		速度異常 == TRUE		○			
	||		■温度異常 == TRUE			○		

変更後の不具合状態：

			条件式	成立条件				
				#1	#2	#3	#4	#5
&&			（前提条件）	○	○			
&&			空気圧 >= 設定値	○	○			
&&	||		速度異常 == TRUE	○				
	||		■温度異常 == TRUE		○			

	■ソフトウェアアーキテクチャ設計（変更設計） ・システムの全体像を把握してから変更すること ・　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 ・　設計意図を文書に残すこと ・ ■ソフトウェアアーキテクチャ設計 ・適切な規模にモジュール分割し，複雑さを減らすこと ・ハードウェアの制約を超えないように設計，監視すること

教訓 2

教訓タイトル	条件が整理されていない状態で、トータルの条件数が 100 を超えるような機能、または 10 個以上の条件を有する機能を修正する場合、関連する条件を全て洗い出して整理し不整合がないことを確認する
製品の特徴	溶剤処理を伴うプロセス制御のコントロールシステムで、温度等の環境条件が厳しい中で使用され高い信頼性が要求される。
観察できる現象	運転中にドレン系の配管に亀裂が発生し対応処理中に排気処理系ポンプの一部が所定の排気動作終了後も停止せず加熱焼損した。火災事故には至らなかったが、当該系統の一部機能が喪失してしまい延焼すれば大事故に至る可能性があった。
内部で発生した事象	この排気処理系は A, B 2 台のポンプがあり、ドレン開始指示信号 ON に基づいてポンプ起動される。起動時には排気モニタ制御が一定時間動作する仕様になっていて、A ポンプ起動後このモニタ機能完了後に A ポンプを停止し B ポンプを起動して排気を行う。今回、配管の亀裂の際に温度センサーも損傷が発生し、この影響で異常時の制御シーケンスが一部効かなくなりこの状態で A ポンプ起動後排気モニタ制御が完了する前にドレン開始指示信号が OFF してしまい、排気モニタが完了しない状態になり、A ポンプが停止せず焼損してしまった。
原因となる要因	・従来は、ドレン開始信号は排気完了まで ON 状態が継続する前提でロジックが設定されていたが排気処理動作仕様が変わりその対応を実施した際、温度センサー故障でも対応できるロジックであったものを意図せずポンプ動作シーケンスを変更した際に排気完了を待たず OFF となってしまうようなロジックにしてしまったことが原因であった。 ・本排気処理は前段の反応工程との連携で制御が行われているが、担当者はそうした制御全般に関する知識・経験が薄く、当該反応機能の変更だけを考えて修正作業を行った。
上記の未然防止に向けた対策	直接原因への対策： ・当該条件式を確認の上、ロジックの修正を実施
	要因への恒久対策（対応工程を明記）： ① 変更前後で変数の数等を比較することには一定の効果はあるが、この場合のように前後で大きな差がない場合は、数だけでは違いに気が付きにくい。 複雑で量も増大した制御動作ロジックを特定モジュールに集中させ過ぎており、メンテ

ナンスの観点から適切に分割する等の方策をとることが望ましい。思考の範囲を小さくすることにより誤りが少なく、かつ短期間での設計が容易になる。

変数	変更規模	変更前　複雑度		変更後　複雑度	
		入力変数	条件数	入力変数	条件数
A ポンプ実行状態	なし	8	7	8	7
A ポンプ起動	小	6	5	7	6
○○弁動作	小	7	8	9	11
○○弁開状態	なし	5	4	5	4
ドレン開始指示	－			26	43
ドレン実行状態	－			30	58

変更前後で大きな差がない

② A、B ポンプとも焼損に至るような連続運転を避けるため一定時間以上は動作させないようなロジックを実装し、ベテランエンジニアが設計者に工程全般に関する技術知識を教育する。従来から組織内で所有している信頼性に関わる知識の伝承が可能になる。

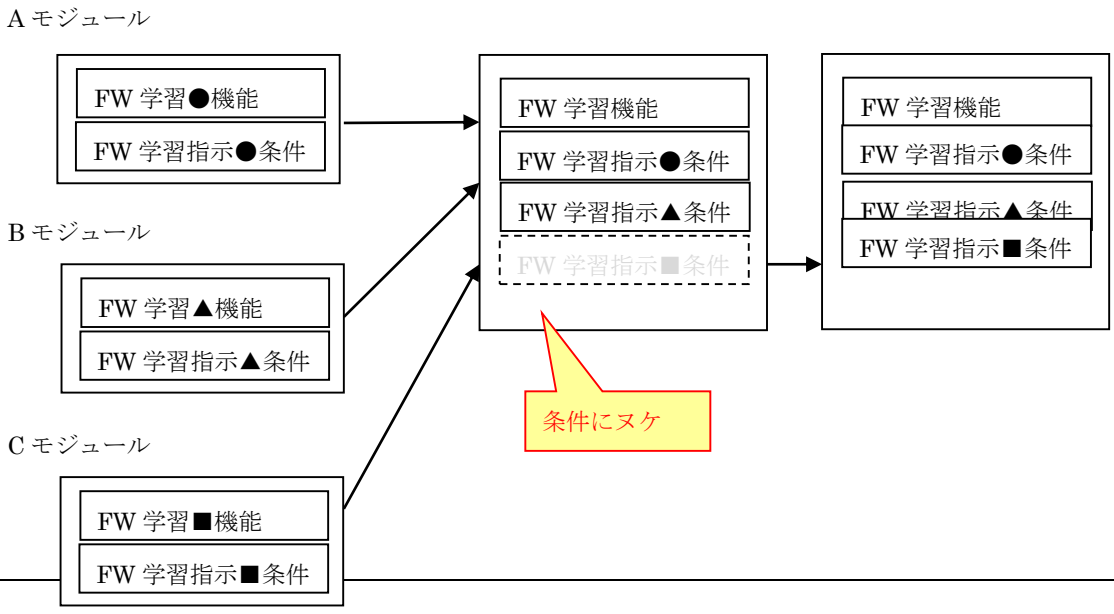
複雑さの定義と閾値の目安：
　条件が整理されていない状態で、トータルの条件数が 100 を超えるような機能、または 10 個以上の条件

■ソフトウェアアーキテクチャ設計（変更設計）
・システムの全体像を把握してから変更すること
・　設計意図を文書に残すこと
・　並列システムの設計，変更の際にはタイミング図等を援用して検証すること

■ソフトウェアアーキテクチャ設計
・適切な規模にモジュール分割し，複雑さを減らすこと．
・ハードウェアの制約を超えないように設計，監視すること

教訓 3

教訓タイトル	複数機能モジュールを統合する場合、統合前の条件数の総和と統合後の条件数を比較し差がある場合は、条件の抜けがないか確認する
製品の特徴	工場内の製品・仕掛品の自動搬送制御（AGV: Automatic Guided Vehicle）システムで、工場の生産性向上を達成する上で搬送経路の最適制御を行えるとともに、高い稼働率が要求される。
観察できる現象	稼働中に多数ある位置センサーの一部に故障が発生し、そのメンテナンス準備中に自動搬送車同志の接触事故が発生し、長時間の搬送停止による出荷停滞という事態となった。
内部で発生した事象	移動に際しては他走行車や人やモノの存在や距離等を測定するセンサー入力情報等を最適制御ルーチンの中で自己学習しながら動作するが、故障したセンサー入力情報が存在しない条件においては前方位置に関する学習を指示する条件の中に一部抜けが存在していた。
原因となる要因	・当該機能は複数のソフトウェアモジュールで構成されてきたが、機能統合する事情が発生しモジュール構成の見直しを実施した際に、元々の動作仕様条件の一部に転記モレが発生した。 ・通常開発環境でのデバッグと構造ビジュアル化ツールによる確認作業を実施していたが、レアケースで発生しており担当者が、仕様条件に抜けがあることを認識できなかった。



<table>
<tr><td>上記の未然防止に向けた対策</td><td>直接原因への対策：
変更前の条件と変更後の条件を突合せ確認し、ヌケ部分を追加修正した。</td></tr>
<tr><td></td><td>要因への恒久対策（対応工程を明記）：
・一般的に学習機能の確認はパラメータ類の組み合わせが膨大になる傾向があり、実機による検証でないと発見が難しいものが多い。事前に仮想環境を構築してテストは実施しているが、網羅度合には限界がある。

・ソフトウェア変更作業の際の条件の設定モレを防ぐ対策として、構造化ビジュアルツールによる目視確認の際、

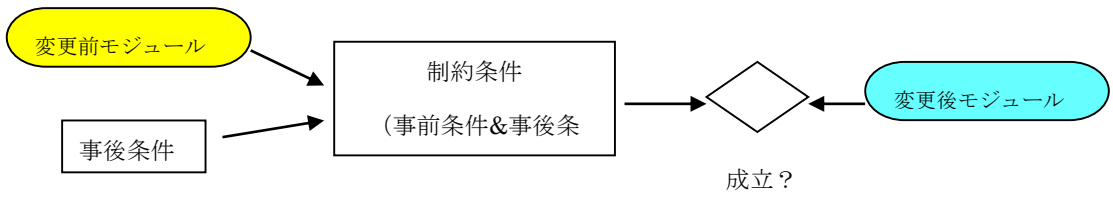
統合前のモジュール中で出現する指定条件数の総和：X
統合後モジュール中で指定される条件数：Y

と定義し、 X ≒ Y となっているかを本ツール上で確認し、結果を記録する。

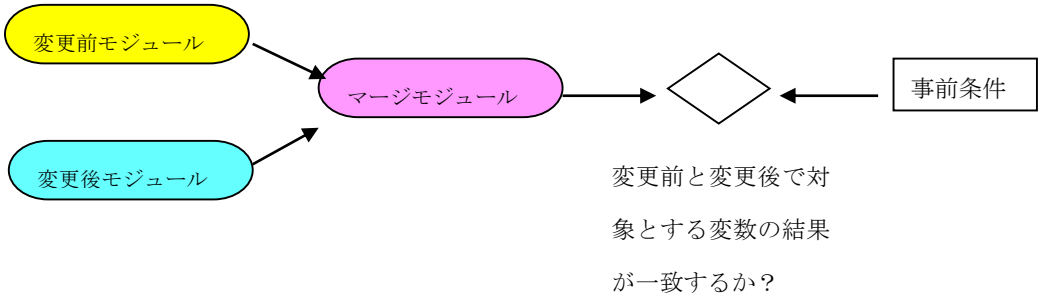
<table><tr><th></th><th>統合前と統合後不具合</th><th>統合前と修正後</th></tr><tr><td>ビジュアルツール上での大きさ比較</td><td>大きく異なる</td><td>ほぼ同じ</td></tr><tr><td>ビジュアルツール上で条件数比較</td><td>統合前：総数＞19 以上
統合後不具合：総数＝4</td><td>統合前：総数＞19 以上
修正後：総数＞19 以上</td></tr></table>
この方法により、簡便に誤りの有無の可能性のチェックが容易になる。

■ソフトウェアアーキテクチャ設計（変更設計）
・ 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
・
■システムテスト
学習機能を持つモジュールのテストは、網羅度を上げられないことを認識すること</td></tr>
</table>

教訓 4

教訓タイトル	変数値域が広く、組合せバリエーションが非常に多くなる場合には、値域を適切な大きさに分割した上で境界値テストを実施する
製品の特徴	化学物質を搬送トレイに分けて搬送するシステムであり、指定された経路に従って所定量を所定間隔で搬送するもので、動作の確実性と応答時間の正確性が求められる。
観察できる現象	ある日、搬送中にラインが非常停止した。この搬送システムでは停止した場合、トレイ停止時の方向や数量、位置に応じて所定の順序に従って再開できる仕様になっているが、この時は再送がされず長時間にわたってライン停止し損失が拡大しただけでなく、搬送ライン中に滞留した化学物質除去を人手で行うという危険作業を長時間せざるを得なくなった。
内部で発生した事象	停止後の搬送再開の動作シーケンスは、停止時に想定されるトレイ方向や位置をパラメータ表を参照し再開トレイを算出し実行することになっているが、この表を参照して算出するモジュールに間違いがあり、再開トレイが本来#4 であるべきところ、#5 となっていた。
原因となる要因	・当該算出処理部分について、機能は変えずモジュール統合を実施したがその際に意図しない変更を実施してしまい元の動作仕様を結果的に変更してしまったことが原因である。 ・この算出ロジックは停止時の状態情報（トレイ向き、数量、停止位置等）に基づいて、判定するのだが、その組合せと各々のパラメータの値域が非常に多い。組合せテストは実施していたが、パラメータ組合せバリエーションを網羅しきれていなかった。
上記の未然防止に向けた対策	直接原因への対策： 改修部分の再チェックを行い、不適合部分を修正した。
	要因への恒久対策（対応工程を明記）： 組合せと各々のパラメータの値域が非常に多くモレを発生させた反省から、モデル検査技術を活用し、下記のような 2 通りの検証を行って確認する。 <u>プロパティ検証</u>： 

完全一致検証：



また、変数値域が広く、組合せバリエーションが非常に多くなると一般的にモデル検査ではパラメータ組み合わせの爆発が生じるため抽象化を行う。本事例も値域を適切な大きさに分割した上で実施した。これにより、妥当な時間内で検証することが容易になる。

複雑さの定義と閾値の目安：
　条件が整理されていない状態で、トータルの条件数が 100 を超えるような機能、または 10 個以上の条件

■ソフトウェアアーキテクチャ設計（変更設計）

・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等の技術を援用して変更の妥当性を確認すること

教訓 5

教訓タイトル	内蔵電池を使用する場合には、深放電時の起動シーケンスを考慮すること
製品の特徴	ディスプレイと無線通信機能を有し、内蔵電池により AC 充電器の接続が無くても使用が可能な可搬型業務用端末
観察できる現象	電源ボタンを押下しても端末が起動しない。 AC 充電器を接続しているのに充電が出来ない。
内部で発生した事象	・電池電圧が極度に低下した状態（深放電状態）では、充電等によりある一定電圧まで電池電圧が回復するまでの区間（深放電電圧から端末起動電圧閾値までの区間）では端末起動を行わない様、ソフトで起動抑止している。 ・深放電状態にある時に、AC アダプタを接続すると充電により電池電圧が上昇するが、この状態で電源 ON 操作を行うと、充電電力を含んだ電池電圧で端末起動可否判定が行われ、端末起動電圧閾値を上回り、起動処理を開始してしまった。 ・一方、端末起動処理の一部で充電を一度停止する仕様となっており、充電を停止すると電池電圧は元の状態（深放電状態）に戻るため、電源 IC の持つ端末起動電圧閾値を下回ってしまう。 ・よって、ハードウェアは端末起動電圧閾値を下回った状態では、電池電圧不十分として端末起動と全系統の電源供給を停止してしまい、充電停止のままシステムが停止してしまった。
原因となる要因	・電池電圧が極度に低下した状態（深放電状態）を想定した評価が行われていなかった。 ・電池電圧が極度に低下した状態（深放電状態）での AC 充電器接続状態の配慮が不足していた。
上記の未然防止に向けた対策	直接原因への対策： ・AC アダプタ接続時の端末起動電圧閾値の変更
	要因への恒久対策（対応工程を明記）： ・電池で動作する端末においては、長時間放置等により電池電圧が極度に低下した状態（深放電状態）になり、その状態に対する対策が必要であることを認識すること。 ・電池電圧が極度に低下した状態（深放電状態）にならない工夫や、その状態になった場合の対策を検討し、必ず実際の装置で検証を行うこと。 ・端末の起動電流や電圧降下及びそのバラツキを考慮し設計に織り込むこと。 ・客先のユースケースを調査／検討し、同様の条件／環境で評価を行うこと。 この対策により、 ・電源ボタンを押下しても端末が起動しない

	・AC 充電器を接続しているのに充電が出来ない といった製品の充電に関する動作不良を設計時に防止することが容易になる。 ■システムアーキテクチャ設計 ・ハードウェア開発部門とソフトウェア開発部門の連携が重要 ・ハードウェアの特性を文書化してソフトウェア設計の入力として与える ・ハードウェア特性の文書をメンテナンスし続けること ■ソフトウェアアーキテクチャ設計 ・ハードウェアの制約を考慮すること ■システムテスト ・物理的な条件を網羅すること ・実機検証を行うこと ■レビュー ・ステークホルダを集めて分野を越えた人たちでレビューすること ■教育 ・ハードウェア技術者とソフトウェア技術者の文化交流の場を設けること ・文化交流を行うこと

教訓 6

教訓タイトル	フラッシュメモリを使用する場合には、書き込み寿命回数を考慮すること
製品の特徴	ディスプレイと無線通信機能とストレージとしてフラッシュメモリを内蔵し、内蔵電池により AC 充電器の接続が無くても使用が可能な可搬型業務用端末
観察できる現象	・電源ボタンを押下しても端末が起動しない。 ・端末の起動中にフリーズする。 ・端末使用中に電源断やリセット、フリーズが発生する。
内部で発生した事象	・端末のストレージ（OS、データ、アプリ等の保存用）としてフラッシュメモリを採用。 ・フラッシュメモリの特定領域の値が壊れており、起動出来ない等の不具合が発生。
原因となる要因	・ストレージとして採用したフラッシュメモリの特定領域の値が壊れていた。 ・フラッシュメモリの特定領域に、書き込み寿命回数を超えた書き込みが行われたため。
上記の未然防止に向けた対策	直接原因への対策： ・書き込み回数増の要因を修正し、書き込み回数を低減する。
	要因への恒久対策（対応工程を明記）： ・端末のストレージとしてフラッシュメモリを採用する場合、寿命部品と認識すること。 ・フラッシュメモリは寿命回数を超えない様に注意する。 ・寿命回数を超える可能性がある場合には、回数を監視する機能や、寿命回数を超えない仕組みを準備する。 ・採用したフラッシュメモリの寿命回数や平準化方法等については、実際の装置での使用方法で問題が無いか、事前にメモリメーカーに確認する。 ・フラッシュメモリを採用するときは、実際のお客様の使用環境や取り扱うデータを把握するとともに、使用中のどの様なタイミングで、どの程度の頻度で読み書きされるのかを把握する。 ・自社で開発したソフトウェア領域だけではなく、使用する OS や、購入した SW や FW の動作も含めて検討する。 ・フラッシュメモリにバリエーションがある場合は、その差分を把握し評価を行う。 ・フラッシュメモリを変更する場合（NOR フラッシュ→NAND フラッシュ、SLC→MLC→TLC 等）は、寿命回数が異なるので、特に注意する。 ・フラッシュメモリを使用する際は、ハードウェア開発部門とソフトウェア開発部門との連携が重要であり、協力して不具合の未然防止に取り組む。

	これにより、 ・電源ボタンを押下しても端末が起動しない ・端末の起動中にフリーズする ・端末使用中に電源断やリセット、フリーズが発生する といった製品の部品寿命に起因する動作不良を設計時に防止することが容易になる。 ■システム要求定義 ・システムの利用されかたをあらかじめ想定すること ■システムアーキテクチャ設計 ・HW の変更時には影響解析を行うこと ・特殊な HW を使用する時にはあらかじめ特性を把握しておくこと ・システム設計上でダイアグ（診断）機能を実装し、結果を通知する機能を具備すること ■プロジェクトマネジメント ・ハードウェア開発部門とソフトウェア開発部門のコミュニケーションを密にすること ■運用 ・お客様がアプリケーションソフトウェアを追加する際の制約条件を伝えること

教訓 7

教訓タイトル	消費電力の多い機能を追加する場合には、一時的な電圧降下による影響（リセット、フリーズ等）や電源の種類、電池の場合は残量を考慮すること
製品の特徴	ディスプレイと 3G 無線通信機能を有し、内蔵電池により AC 充電器の接続が無くても使用が可能な可搬型業務用端末
観察できる現象	・端末起動時や使用中に、「3G モデムが停止しました」とのエラー通知が表示され、以降 3G 通信が使用出来なくなる。 ・電源ボタンを押下しても端末が起動しない。
内部で発生した事象	・3G モデムの動作開始直後に FLASH への書き込みが行われており、書き込み中に電源断が発生し、File System が破壊され、3G モデムが使用出来なくなった。 ・3G モデムの動作開始時の突入電流による電源電圧降下により、リセットが発生。 ・3G モデムリセット対策として、3G モデムの電源にコンデンサを追加したが、今度は電池容量が少ない場合の端末起動時に、3G モデム電源オン時の突入電流により、電池電圧が降下してシステムリセットが発生し、端末が電源オフとなった。
原因となる要因	・3G 無し版と 3G 有り版の 2 つの商品バリエーションあり。 ・3G 無し版の開発が先行しており、3G 有り版の評価が不十分であった。 ・搭載部品により、3G モデムの電源電圧量にバラツキがあり、評価が不十分だった。 ・3G モデムの電源にコンデンサを追加した際に、周辺回路への影響検討が不十分だった。 ・電池容量が少ない状態（電池電圧が低い状態）での評価が不十分だった。
上記の未然防止に向けた対策	直接原因への対策： ・3G モデムリセット対策として、3G モデムの電源にコンデンサを追加する。 ・電池容量が少ない場合のシステムリセット対策として、電源 IC の根元へのコンデンサ追加と、電池容量が少ない状態では端末が起動しない様にソフトウェア変更する。
	要因への恒久対策（対応工程を明記）： ■システムアーキテクチャ設計 ・商品バリエーションがある場合は、その差分を把握し、評価を行う。 ・端末や各機能の起動電流や電圧降下及びその回路部品のバラツキを考慮し、端末起動シーケンスも含め、設計に織り込む。 ・電源ラインに電圧降下対策を追加する場合は、対策部品の配置場所や他の電源ラインへの影響を確認する。 ・電池で動作する端末においては、電池容量が少ない状態に対する対策が必要であることを認識する。これにより、 ・端末起動時や使用中に通信が使用出来なくなる

	・電源ボタンを押下しても端末が起動しない といった製品の電源容量に起因する動作不良を設計時に防止することが容易になる。 ■システムアーキテクチャ設計 ・商品バリエーションがある場合は、その差分を把握し、評価を行うこと。 ・端末や各機能の起動電流や電圧降下及びその回路部品のバラツキを考慮し、端末起動シーケンスも含め、設計に織り込むこと。 ・電源ラインに電圧降下対策を追加する場合は、対策部品の配置場所や他の電源ラインへの影響を確認すること。 ・電池で動作する端末においては、電池容量が少ない状態に対する対策が必要であることを認識すること。 ・消費電力の多い機能を追加する場合には、一時的な電圧降下による影響（リセット、フリーズ等）や電源の種類、電池の場合は残量を考慮すること。 ■ソフトウェアアーキテクチャ設計 ・バリエーションの差分を把握して評価すること ・新たな機能を追加するときには影響解析をすること

教訓 8

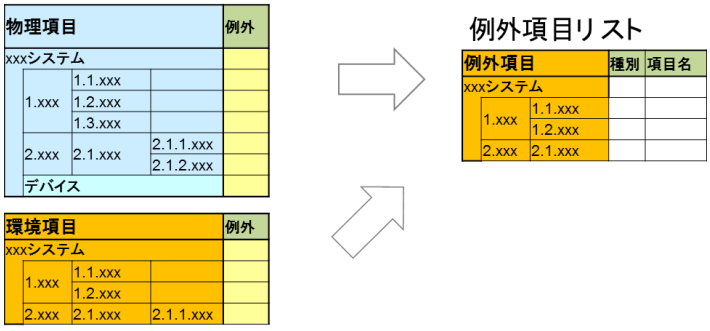
教訓タイトル	想定可能な例外を形式的に漏れなく分析する
製品の特徴	物理現象を制御する組込システムにおいて多くの機能項目の並行動作と様々な周辺デバイス及ぶサブシステムを操作参照しながらシステム目的を達成する制御システム。
観察できる現象	航空機システムが自動操縦中に急降下を引き起こした。原因は，センサーから情報を取得して姿勢情報を取得する姿勢情報管理装置がある。姿勢情報管理装置の姿勢情報を自動操舵装置が読み取り飛行を制御している。この姿勢情報管理装置が想定外の大きな姿勢情報を出力し続ける事で自動操舵装置が誤った操作を行い急降下に繋がった。姿勢情報管理装置は電源リセットで復旧した。 姿勢情報管理装置の現象としては，以下の 1～3 が想定される。 現象 1：運転中にシステムが急に停止してシステム障害にとなる。再始動操作で正常運転となる。 現象 2：外部入出力を正しく参照/操作できず機能が正常に動作せずシステム障害となる。外部入出力の種別及び機能の種別により現象は特定できない。 現象 3：運転中に特定の機能の実行が終了せず他の機能が動作せずシステム障害となる。電源リセットしなければ復旧できない。
内部で発生した事象	誤った姿勢情報を出力し続ける姿勢情報管理装置の内部発生の可能性の事象は、以下の事象 1～事象 6 が想定される。 事象 1：MPU 内部データの破壊 MPU へのノイズ侵入源は、MPU デバイスのピンから入り込む。この時に IO の方向レジスタが最初に破壊、MPU の内部まで到達するエネルギー量のノイズであれば、MPU 内部レジスタが破壊され MPU 誤動作（入力が変化しない，出力が変化しない、割り込みが起動しない、etc）又は暴走する。→現象 1、現象 2 事象 2：想定外の割り込みプログラムの実行 ノイズにより外部割り込みが発生して入力処理を実施する事で不定な入力値を使用してシステム障害が発生する。→現象 2 事象 3：割り込み連続起動による CPU 処理占有 割り込みがバースト的に発生して割り込みプログラムが CPU を占有することにより MPU に接続される周辺デバイスの時間制約を満足することが出来ず入出力を正しく操作することが出来なくなる。→現象 2 事象 4：入力デバイス故障

	ハードウェアの破壊により入力値が不定となりシステム障害が発生する。→現象 2 事象 5：出力デバイス故障 ハードウェアの破壊により出力操作を実施してもハードウェアの操作が出来ずシステム障害が発生する。→現象 2 事象 6：入出力デバイス及びサブシステムの故障又は断線 入出力デバイス又はサブシステムの応答を待つ場合、故障又は断線時に応答が返らず特定の機能実行から抜けることが出来ず、他の機能がどうさできなくなりシステム障害となる。
原因となる要因	例外を想定した仕様が定義できていない。 組込みシステムが参照、操作する入出力デバイス、サブシステムの例外を考慮できていないのが発生要因である。
上記の未然防止に向けた対策	直接原因への対策： 事象 1：MPU 内部データの破壊 MPU 内部レジスタのリフレッシュによりノイズにより破壊したレジスタを復旧 割り込み及びタスクを監視して一定時間動作しなければリセット操作を実施する事で復旧させる。 入力デバイスから取得したデータのノイズ処理を実施する。 入力データ範囲以外であれば入力範囲に丸める。 事象 2：想定外の割り込みプログラムの実行 割り込みの起動時に正常割り込みであるかを判断してノイズによる割り込みであれば何もせずに割り込みプログラムを終了。 事象 3：割り込み連続起動による CPU 処理占有 同上 事象 4：入力デバイス故障 入力データ範囲以外であれば入力範囲に丸める。 入力デバイスを一定周期で監視して入力デバイスの応答が無ければ入力デバイスをリセットする。入力デバイスが復旧出来ない場合は、MPU のリセット操作を実施する。 事象 5：出力デバイス故障 出力デバイスを一定周期で監視して入力デバイスの応答が無ければ出力デバイスをリセッ

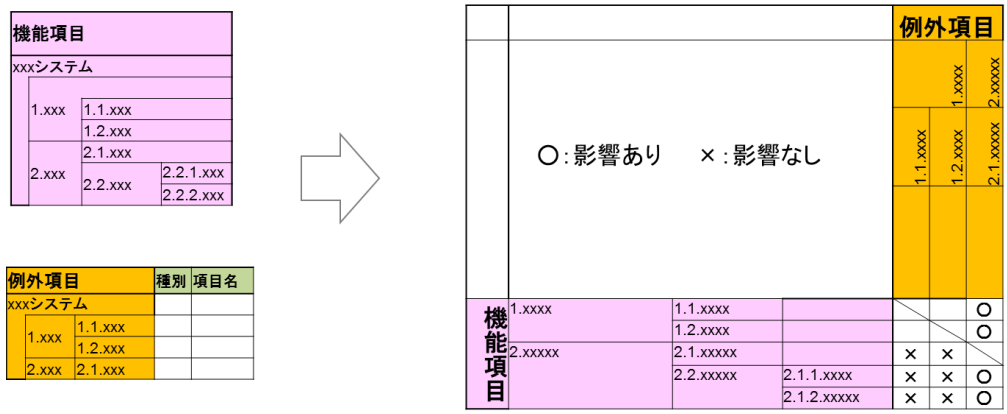
トする。出力デバイスが復旧出来ない場合は、MPU のリセット操作を実施する。

事象 6：入出力デバイス及びサブシステムの故障又は断線
入出力デバイス又はサブシステムの応答を待ちのタイムアウトを設ける。一定時間応答が無ければ対象機能の実行を中断して異常処理を実施する。

要因への恒久対策（対応工程を明記）：
要求分析定義にて以下の作業を実行する事で例外の障害を未然に防止する。
・例外項目を物理的観点、環境的観点により定義する
・定義した例外項目から例外リストを定義する



・機能項目リストと例外項目リストのマトリクスを生成して例外の機能仕様を定義する



これにより、外部入出力を正しく参照/操作できず機能が正常に動作しなくなることから発生するシステム障害の可能性を減少させることが容易になる。

■システム要求分析
・例外項目を物理的観点、環境的観点により定義すること。
・定義した例外項目から例外リストを定義すること。

	■システムアーキテクチャ設計 ・機能項目リストと例外項目リストのマトリクスを生成して例外の機能仕様を定義すること。

教訓 9

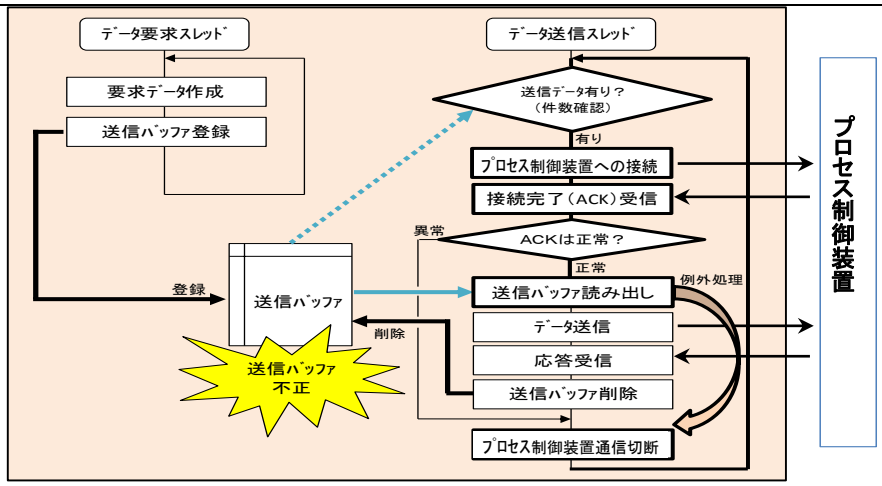
教訓タイトル	システムを二重化する場合は、同期すべきデータ領域を適切に設定する
製品の特徴	高稼働率（無停止期間の長期化）が必要とされる遠隔監視システムにおいては、通常の故障や誤動作の発生頻度を小さくする以外に、故障が発生しても動作を続行する、故障から早く回復する機能が要求される等、高い信頼性が求められる。
観察できる現象	二重化システムを採用して高稼働率を実現するためには、マスター側が故障した場合でも、制御の連続性を維持してスレーブ側に切り替わらなければならないが、スレーブ側に切り替わった直後に、異常を通知するアラームが発生した。
内部で発生した事象	スレーブ側に切り替わった時点で、データ同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	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	直接原因への対策： 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）： 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。 ■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。

教訓 10

教訓タイトル	制御対象のハードウェアが同一でも、運用条件が変わるときは、ハードウェア仕様を再確認する
製品の特徴	・無線 LAN 経由で通信する他用途の製品をベースに開発 ・ハードウェア，ソフトウェアともベース製品を元にして開発 ・対象製品に帰属可能なハンディ端末の最大数がベース製品より増加
観察できる現象	ハンディ端末が最大数帰属すると、製品がリブートする場合がある。
内部で発生した事象	ハンディ端末を帰属させる処理において、帰属させる端末に対応する key エントリの無線 LAN チップへの登録が失敗する。
原因となる要因	〔技術面〕 (1) 設計：要求仕様に対する無線 LAN チップ・データシートの影響箇所の確認漏れ 帰属するハンディ端末の Key エントリ管理に、無線 LAN チップ内部メモリにある管理テーブルを使用。暗号化方式ごとに Key エントリのエントリ方法が異なり、帰属する暗号化方式の順番によっては最大数帰属時に Key エントリ登録不可となる。ソフトウェア設計者は、無線 LAN チップの仕様を理解しておらず、影響を認識しなかった。 (2) 設計：レビューアの選定漏れ レビューア（開発チームメンバ）が無線 LAN チップの仕様を理解していなかった。 (3) テスト：条件の組合せ漏れ 結合テストにおいて最大数帰属の通信テストを実施していたが、その場合に帰属させる暗号化方式の順序に対する網羅性が不足していた。
上記の未然防止に向けた対策	直接原因への対策： 無線 LAN チップの管理テーブルとして、チップ内部メモリではなく、チップ外部メモリを利用するように、ソフトウェアの無線 LAN チップ制御部を修正した。
	要因への恒久対策（対応工程を明記）： 1. 要求仕様の影響範囲の特定（要求定義工程） 「ハードウェア仕様に対するベース製品との要求仕様差分の影響の確認」をレビューチェックシートへ追加する。 2. テスト網羅性の向上（結合テスト工程） 最大数帰属時に，帰属させる暗号化方式の組合せテストを追加する。 3. 適正なレビューアの選定（要求定義工程、結合テスト工程） ベース製品のソフトウェア設計者、システムテスト設計者を設計レビュー、結合テスト仕様レビューへ参加させる。 これにより、メモリ容量不足に起因するシステム障害の発生の可能性を減少させること

	が容易になる。 ■システムアーキテクチャ設計 ・ハードウェア開発部門とソフトウェア開発部門で継続的に連携すること ・ハードウェアの特性を文書化してソフトウェア設計の入力として与えること ・ハードウェア特性の文書をメンテナンスし続け，劣化を防止すること ■ソフトウェアアーキテクチャ設計（変更設計） ・ハードウェアの制約を考慮すること ■レビュー ・ステークホルダを集めて分野を越えた人たちでレビューすること ■教育 ・ハードウェアの教育をソフトウェア技術者にすること ・ハードウェア技術者とソフトウェア技術者の文化交流の場を設けること

教訓 11

教訓タイトル	プロセス間、スレッド間でデータを共有（引き渡し）する場合は、排他・同期処理が正しく行われているか、あるいはデッドロックが発生していないかどうか注意する
製品の特徴	Windows サーバで処理を実施し、現場側の**プロセス制御装置**から通信によりコンベアの分岐制御や実績収集する生産管理システム
観察できる現象	プロセス制御装置とサーバ間の通信が停止して、工場のライン停止に至った。
内部で発生した事象	 ①送信バッファに対し、データ送信スレッドの削除タイミングとデータ要求スレッドの登録タイミングとが重なり、送信バッファが不正状態となった。 ②データ送信スレッドは、不正状態の送信バッファの読み出しを行なったため、例外処理によりプロセス制御装置との接続/切断を繰り返す状態となった。
原因となる要因	プロセス内でスレッド間のデータリンケージ用の送信バッファに ArrayList を使用していた。ArrayList はスレッドセーフでないため排他を掛ける必要があったが、コーディングした人は ArrayList がスレッドセーフでないことを知らなかった。
上記の未然防止に向けた対策	直接原因への対策： 1）ArryList をスレッド間で使用できるように、排他処理を追加した 2）スレッドセーフ関連の類似見直しを実施した
	要因への恒久対策（対応工程を明記）： マルチスレッド処理でスレッドセーフであることの観点を、プログラムチェックシート及びコーディング規準に追加した。 これにより、コードレビュー時にスレッド間のデータの排他漏れに関わる不具合の指摘が容易になる。

	■ソフトウェアアーキテクチャ設計 ・並列処理を考慮して設計すること ・共有データはあらかじめ洗い出してチェックすること ■コーディング ・　マルチスレッド処理でスレッドセーフであることの観点を、プログラムチェックシート及びコーディング規準に追加すること ■教育 ・　マルチスレッド処理の教育をすること

教訓 12

教訓タイトル	歩留りのある製品の良品／不良品を検査する装置では、全てが良品あるいは、不良品との検査結果は異常と判断すべきである
製品の特徴	半導体がスペック通りの機能・性能を満たしているかを検査する装置。 半導体の検査は複数のテストで構成され、その全てのテストが良の場合は検査結果が良品となる。一方で、あるテストで不良になった場合は不良品と判断され、検査時間の効率化のため、通常その後のテストは行わない。
観察できる現象	半導体の検査では、一定の割合で不良品が発生するが、検査した全てが良品となった。 しかし、その後の検査の工程で通常より多くの不良品が検出された。 全て不良品の場合と同じく、全て良品になった場合でも、検査自体に異常があると考えなければならない。
内部で発生した事象	調査担当者が不良品の調査のため、不良にならないようにマスク設定した。調査終了後は、マスク解除しなければならないが、調査担当者がマスク設定を解除しない状態で、量産を開始したため、検査は全て良品となった。全て不良品の場合、あるいは全て良品の場合には、検査自体が異常の可能性があることを通知する対応を忘れた。
原因となる要因	全て良品となった場合に、異常を通知する処理を忘れた。 全て不良品の場合は直感的に検査が異常であるとわかるが、全て良品の場合にも検査が異常であると考えが及ばなかった。
上記の未然防止に向けた対策	直接原因への対策： 全て不良品の場合と同様に、全て良品の場合も異常を通知するよう修正した。
	要因への恒久対策（対応工程を明記）： ・良／不良の条件に関わる仕様を明確にする。 ・検査装置の検証項目に加え、できるだけ検証を自動化する。 これにより、保守時の捜査員のミスに起因する問題の発生を減少させることが容易になる。

	■　システム要求定義 ・歩留りのある製品の良品／不良品を検査する装置では、全てが良品あるいは、不良品との検査結果は異常と判断すべきである。 ■システムアーキテクチャ設計 ・システム設計の中でもメンテナンスモードに対する設計に注意すること。 ■運用 ・メンテナンスモードを有するシステムでは、その取扱いについて、手順書で明確にして、ダブルチェックも考慮すること。

教訓 13

教訓タイトル	既存ソフトウェアの性能改善を実施する際には、アイドリングタイムの発生、処理の同期ずれの発生等と影響を確認する
製品の特徴	電子機器製品を検査するための製造工程の検査装置であり、PC と複数の組込み機器（信号発生機器、電圧・電流計測機器、電流計測機器、通信機器、カメラ等）で構成される。mS オーダーの計測精度・制御フローの確認が要求される。 開発形態としては、現行品への機能追加となる。
観察できる現象	電子機器製品が有効な通信データを送信しているにも関わらず、検査装置が「データ無しエラー（no data)」を出力してしまい、一時、ラインがストップした。
内部で発生した事象	検査装置の PC は、一定期間内に受信された通信データを通信機器のバッファからサイクリックに読み出し、ログファイルに保存していく。ログファイルは、検査実行後に詳細に解析される。 電子機器製品によっては、正常動作中であっても、一定期間内に通信がまったく存在しない状況が存在するが、その際、空のデータが、タイムスタンプ 0s の有効なデータとしてログファイルに保存されたため、正常な 2 分探索ができず、有効なデータを検出できなかった。
原因となる要因	現行の検査装置への機能追加時に、処理速度の改善を併せて実施した際、下記要因によりデグレードし、且つテストでも検出できなかった。 〔技術面〕 ■通信データが存在しない場合のテスト項目の不足 〔人・組織面〕 ■現行品の設計意図を記録に残していなかった ■変更点の管理の甘さ ■（変更規模が小さいと判断したことによる）担当者への依存度大
上記の未然防止に向けた対策	直接原因への対策： もともとは存在していた「通信機器のバッファから通信データを読み出す際、通信データの有無を確認する処理」を追加し、通信データが無しの場合は、ログファイルへの保存をしないようにソフトウェアを変更。
	要因への恒久対策（対応工程を明記）： ・変更点管理リストへの記入の徹底（ソフトウェア変更設計） ・設計書への設計意図の記載（ソフトウェア設計） ・設計・実装レビューでの「変更点の確認」「変更による影響範囲の確認」の確実な実施（レビュー）

	・テスト網羅性の向上（テスト） ・テスト項目の抽出観点追加 ⇒データが存在しない場合の振る舞い確認 ⇒意図的に通信データの存在しない期間を途中に入れる） ・レビューアの能力向上（レビューアのスキルマップ作成、レビューアの選定） これにより、通信系システムにおいて通信データの異常に対する考慮漏れの防止が容易になる。 ■ソフトウェアアーキテクチャ設計（変更設計） ・変更点管理リストへの記入を徹底すること ■ソフトウェアアーキテクチャ設計 ・設計意図を文書に残すこと ■レビュー ・設計・実装レビューでの「変更点の確認」「変更による影響範囲の確認」を確実に実施すること ■システムテスト ・テスト観点リスト等作成を通してテストの網羅性向上を図ること. ■教育 ・レビューアのスキルマップを作成する等してレビューアの能力向上を図ること ■プロジェクトマネジメント ・レビューアのスキルマップを作成する等して適切なレビューアの選定を行うこと. ・ひとりの担当者に過度に依存したタスクの割り当てを行わないこと

教訓 14

教訓タイトル	大量のデータを通信経由で扱う場合、一連の処理の流れの中にボトルネックを作りこまないように注意する。 また、時間帯による負荷変動についても考慮する
製品の特徴	複数の営業担当者が携帯する業務用端末と、リモートでのデータ授受サービスを提供するデータベースサーバと、データベースサーバから営業担当者の情報を受け取り、営業の支援をするクライアント端末（本社・支店）からなるシステムで、常時稼動ならびに遅滞のないデータの授受サービスが要求される。
観察できる現象	利用者の多い時間帯にサーバのレスポンスが著しく低下し、営業情報ならびに営業支援情報の授受に非常に時間がかかることになり、サービスを断続的に停止するに至った。
内部で発生した事象	・サーバ内のデータ解析処理が高負荷になり、業務用端末から送られてくるデータが一時記憶部（通信バッファ）に大量に滞留した。 ・サーバへアクセスするプログラムに、最適化されていない記述があり、これが処理時間のボトルネックとなっていた。(データの文字列を直接比較せず、一旦、数値に変換してから比較していたため、必要以上に処理時間がかかっていた。)
原因となる要因	・利用者が増え、想定した以上のデータが特定の時間帯にサーバに送られてきた。 ・ボトルネックを想定したサーバへのアクセス条件に関する仕様記述がなかった。 ・サーバの有識者が実装レビューへ参加できていなかった。 ・システム全体での統合負荷テストが実施不足だった。 ⇒実環境に近い試験環境をタイムリーに用意できなかった。
上記の未然防止に向けた対策	直接原因への対策： ・文字列を直接比較するようプログラムを修正。(最適化) ・実環境に近い試験環境にて、サーバへのアクセス処理時間を計測し、処理速度が改善されていることを確認。
	要因への恒久対策（対応工程を明記）： ・ボトルネックを想定したサーバへのアクセス条件に関する仕様を記述する。 ⇒　仕様に則ったプログラミングの実施 ・サーバの有識者が、必ずレビューに参加するしくみ（ルール化）構築 ・実環境を模擬した試験環境の早期手配 これにより、性能設計よる性能の確保・改善の確実性の向上を容易にする。

	■システム要求定義 ・　クライアント端末数等の非機能要求を整理すること ■システムアーキテクチャ設計 ・　仕様変更にともなう影響解析をすること ■ソフトウェアアーキテクチャ設計 ・大量のデータを通信経由で扱う場合、一連の処理の流れの中にボトルネックを作りこまないように注意する。また、時間帯による負荷変動についても考慮する。 ・データの出と入りを等しくなるように設計すること。 ■レビュー ・性能に関わる箇所については確実にレビューすること

教訓 15

教訓タイトル	納入後、お客様が運用する業務システムでは、業務シーケンス中のあらゆる異常操作（リセット、電源断、放置も含め)、への対応を考える
製品の特徴	店舗用窓口対応業務装置で、日次の業務処理に必要なデータを受信したり、当日行った処理の集計データを所定の時間にセンターサーバに送信したりするバッチ処理が自動的に実行されることになっている。
観察できる現象	ある日、窓口対応員が当該装置の電源を OFF にして帰宅し、翌朝窓口対応業務を行おうとしたところ、業務システムが正常に立ち上がらず店舗業務運営ができない状態が発生した。
内部で発生した事象	業務処理集計データの送信状態が未完了状態であったため、装置システムが正常に起動しない。
原因となる要因	昨日の業務処理集計データをセンターサーバに送信中にバッチ処理を強制されたためデータ送信処理が未完了状態になったことが直接原因であることがわかった。　本来、送受信中に強制終了されたとしても、次回再開した際に送信未終了分を再送する等のリカバリ処理が実施されるべきであったが、そうした事態が想定されておらず、リカバリ機能が未実装だった。
上記の未然防止に向けた対策	直接原因への対策： センターサーバとの送受信処理中に強制終了された場合の再送、再開時処理を実装する。
	要因への恒久対策（対応工程を明記）：要件定義 要求仕様を検討する際の確認事項に強制終了された時の復旧処理を盛り込む。 これにより、業務実行中に操作員が強制システム終了を行う等の操作ミスへの対応力の向上が容易になる。 ■システム要求定義 ・要求仕様を検討する際の確認事項に強制終了された時の復旧処理を盛り込む ■システムテスト ・瞬停やネットワーク異常対応処理を盛り込んだテスト設計をすること

教訓 16

教訓タイトル	障害解析時の保守メンテ用ログ処理であっても、仕様書を作成し、影響評価を実施すること
製品の特徴	多くのプロセス製造工程を経て最終製品として作られる製品があり、途中には加工処理だけでなく人体に有害な化学薬品処理も含まれている。このシステムでは各工程品質状態を管理するために工程ごとの作業情報管理をログトレースデータとして保存し、ロットごとにそのロギングデータを集計してサーバに送信し、次工程以降で使用している。
観察できる現象	ある時ある製造工程でログデータを集計中に当該処理が異常終了し、製造工程が停止してしまった。
内部で発生した事象	次工程に引き渡すべき製品に対応したログデータが一部消失した。
原因となる要因	当該工程中では、薬品処理状態によって品質にムラが生じるため処理後に工程最終段で状態確認を行いログデータとして保存しているが、この際、正常時と不適合時のデータが混在しないようそれぞれのログデータを別ファイルに出力するため別モジュールでロギングしなくてはならないところ、なぜか共通のログ処理で行っていることが真因であると判明した。不適合時のログ出力機能は、障害解析時のためのもので、仕様書に記載されていなかった。
上記の未然防止に向けた対策	直接原因への対策： ログを採取する処理は、正常時データと異常時データは別々のモジュールで行い、別ファイルに書き出すように修正する。 ■ソフトウェアアーキテクチャ設計 ・デバッグやログ等の直接要求に関わらない機能であっても、設計とレビューをすること

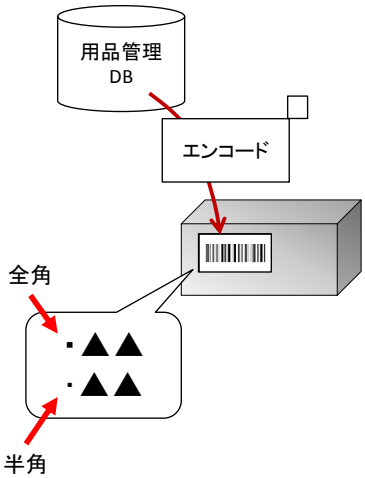
教訓 17

教訓タイトル	判断処理は、必要条件だけでなく、制限すべき条件も漏れなく抽出する
製品の特徴	ある重要施設の入退出ゲート管理システムでは、施設建屋への通行に際しては職員の電子通行証に対して、その施設ごとの関連情報に通行証保有者の ID 情報を関連付けて運用している。この重要施設は当該敷地内に多数あり、各施設は物理的に連結しているが施設ごとにゲートが敷設されていて、通過可能な人の識別と通過可能期間、また施設によっては当日中の入退出回数制限等多くの制限情報に基づいて区域への出入りを監視制御している。
観察できる現象	ある日、職員が A 施設に入場しようとしたところ、異常事態を知らせる警報が発生し入場できなくなり、一時当該施設に関わる職員全員が出入りできなくなり当日の業務に重大な支障が発生してしまった。
内部で発生した事象	この職員は別の X、Y 施設にも何回か出入りし回数制限上限に達した後に A 施設へ立ち入ろうとしたところで、入場判定処理で異常となっていた。
原因となる要因	他施設から立ち寄って入場する場合、入退出回数制限のチェック機能も装備されていたが、なぜか Y 施設の入場回数制限確認処理が抜けており、判定条件ルーチン中で Y 施設が判別できず異常終了していたことが原因だった。
上記の未然防止に向けた対策	直接原因への対策： X、Y 他施設での入退出制限回数確認処理や他条件確認の処理に抜けがないかを確認し実装する。 ■システム要求定義 ・要求条件の抜け漏れを防止するためには、不変条件を論理式で記述する等、形式手法の適用を検討する。 ・要求条件の抜け漏れを防止するためには、観点表に知識を蓄積する。

教訓 18

教訓タイトル	ログファイルの断片化に注意すること
製品の特徴	Windows サーバで処理を実施し、現場側のプロセス制御装置から通信によりコンベアの分岐制御や実績収集する生産管理システム
観察できる現象	障害が発生したため障害解析のためにログファイルをコピー操作したところ、制御処理が遅延して、工場のライン停止に至った。
内部で発生した事象	ログファイルがディスク上で断片化していたため、ログファイルコピー時にディスク IO 負荷が高騰し、オンラインで動作しているプロセスのログ書き込みが遅延したため処理遅れが発生した。
原因となる要因	① 複数プロセスが日々1～300MB 程度の可変長のログファイル 30 ファイルを作成して 30 日経過したログファイルを自動削除するような仕組みとなっていた。 ② 可変長のためログファイルのサイズが増えると断片化しながら自動拡張した。 ③ 30 日経過で断片化したファイルを自動的に削除したが断片化は解消されずログを作成したため断片化が加速した。
上記の未然防止に向けた対策	直接原因への対策： 1）工場停止日にデフラグを実施して断片化を解消させた。
	要因への恒久対策（対応工程を明記）： 1）ログファイルを日々別パーティションへ移動して日々の断片化を回避した。 2）無駄なログを削除してログファイルの削減を図った。 これにより、記録用データにおいて可変長データを扱うことによる不具合発生を減少させることが容易になる。 ■ソフトウェアアーキテクチャ設計 ・可変長ファイルの読み書きには注意する。 ・ログ設計は、ログファイルの断片化に注意して、予め固定長ファイルのログを設計する等の考慮する。

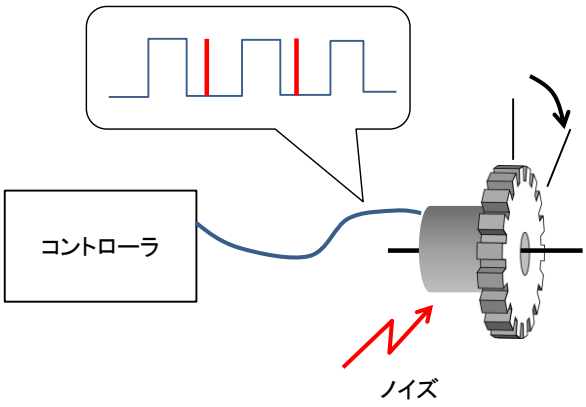
教訓 19

教訓タイトル	人による変更作業ではミスが起きることを前提に、ツール活用等で不具合の作り込みや流出の防止に心がける
製品の特徴	保守用用品の生産管理システムの一部で、重要施設向けの用品発送のためのデータ管理を行うと共にバーコード等の荷札情報を発行、刻印するシステム。
観察できる現象	ある重要施設の定期点検で急遽保守部品の交換が必要になり用品を発送したところ、重要施設の受入検査でいくつかの用品でバーコード情報読み取りエラーのため施設入荷できなくなった。この結果当該施設のメンテナンス作業そのものに支障を来し、点検日程全体に大きな影響を及ぼす事態になってしまった。当該用品類は厳格な識別管理を要するもので、バーコードをレーザ刻印した金属加工部材も含まれている。
内部で発生した事象	重要施設側で読み取れない状態を現地調査し確認したところ、レーザ刻印された識別情報の一部に文字化けが発生していた。（〇〇〇・〇〇〇・▲▲・□□□　という文字列の▲▲部分）
原因となる要因	・原因を調査したところ、この▲▲の直前にある、「・」（なかてん）が本来全角であるべきところが、半角になっていたためと判明した。この識別データは用品出荷にあたってデータベース上の用品管理情報を担当者が直接手作業で編集したのだが、その際 ・▲▲ と全て全角入力後にこの単位で変換すると「・」（なかてん）部分が半角になってしまうが、担当者は半角になっているかを目視で確認しきれなかった。 

	・この識別データは用品と施設ごとに様々な組み合わせがありその量も多い。また対象施設によっては当該識別データに半角文字を含む場合もあり、従来経験者の経験に依存した運用となっていたが、今回は急遽大量の出荷要請があり確認しきれなかったという背景も判明した。
上記の未然防止に向けた対策	直接原因への対策： ・手作業で編集する際は、編集箇所対してチェックツールで半角の「･」（なかてん）がないかどうかを確認する。 ・同様の問題が発生しそうな文字列、箇所がないかを点検し類似のミスが発生すると思われる場合にはツール対応を検討する。
	要因への恒久対策（対応工程を明記）： システム設計工程 ・属人的技量に依存した作業とせず、誰がやっても確実に間違いを検出・修正できるような方策をとることが望ましい。 例）チェックツールや Web 入力等で採用されているような全角・半角入力モードの自動切り替え機能 ■システム要求定義 ・識別データのような情報を手入力しなくてはならない場合は入力規則を定めること。 ■ソフトウェアアーキテクチャ設計（変更設計） ・属人的作業だけに依存せず自動入力やチェックツール等で変換ミスを未然に防ぐ ■システムテスト ・手入力を要するシステムでは範囲外テストを実施する

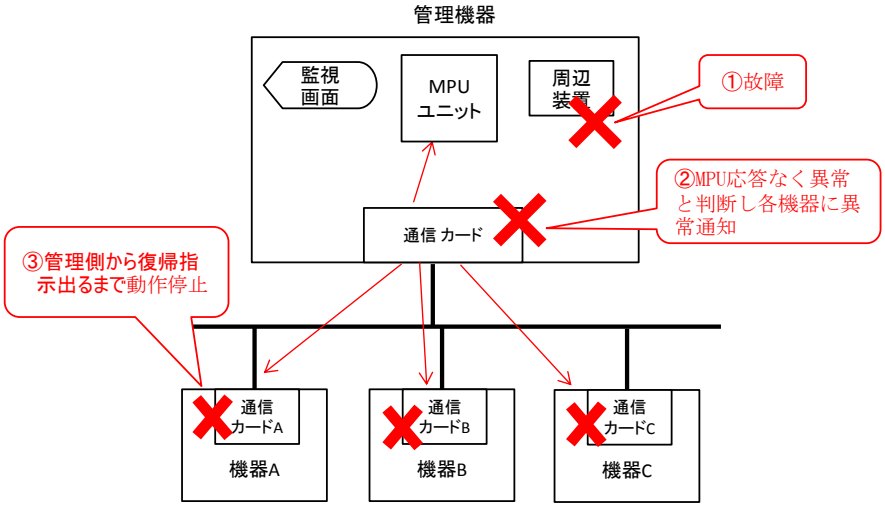
教訓 20

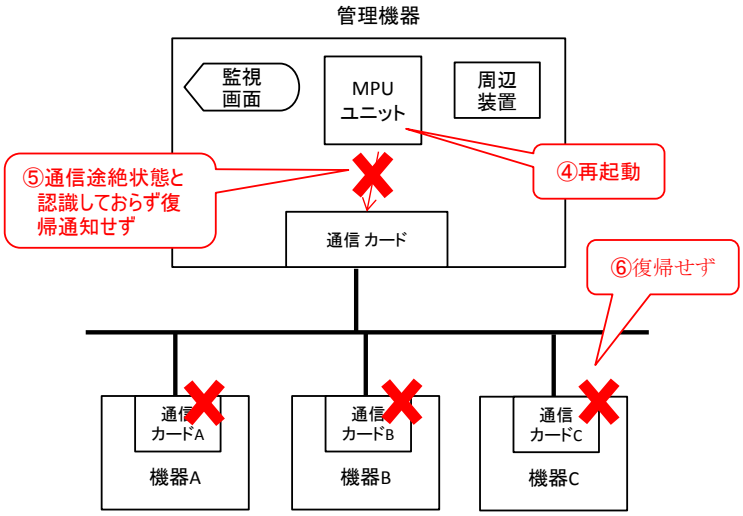
教訓タイトル	信頼性向上施策を採る場合は、故障発生確率と影響の定量評価を行い、対策は確実に実装する
製品の特徴	ある産業用化学製品を製造する工場内で生成される様々な中間精製物を、純度等で小分けし次工程に搬送制御するためのシステム。搬送中に危険状態となることを防ぐための仕組みが組み込まれており、多くのアクチュエータやセンサーが設置され、ms 単位の応答性が求められる。
観察できる現象	固形化中間生成物の小分けを行う工程において、作業中に搬送コンベアの異常を通知する警告灯が突然点灯し警報が発報しコンベアを管理する搬送制御システムが停止した。 現場保守員が確認し点検したが、コンベア含めた機器装置に異常は見受けられず搬送制御システムを再起動したが、しばらくすると再び警報が発報されるという事象が散発的に発生するようになってしまい、操業度に影響を及ぼすことになってしまった。
内部で発生した事象	・搬送コンベアを動かす際には、生成物の数量カウントや所定位置にあるかの確認、モータに連結されている駆動機構の位置や回転数等さまざまな諸元をセンサーで検出しながら制御している。調査したところ、これらの中のある駆動機構の回転数センサーのカウントアップ値が、通常なら 1→2→3→4→　というようにインクリメントされるはずのところが、当該センサーがノイズの影響を受けて 1→2→5→7→　のような遷移となり、異常と判断したために非常停止していたことがわかった。
原因となる要因	ノイズによる一時的なセンサー異常から復帰しても誤警告が継続したのは、以下のようなソフトウェア含めたいくつかの要因が重なっていたためであることが判明した。 ・回転数を測るためのセンサー（磁気センサー）は、回転速度を算出して表示するとともに速度を制御するための装置で、120 度ごとにパルス出力するものである。ノイズあるいは回転子の瞬間的逆転により磁気センサーのパルス出力間隔が急変すると機器異常と判断し警報を発報する仕組みになっている。 ・この中間生成物の小分け作業は危険性が高いということで、全体的に信頼性を向上させる検討が段階的に実施されており、センサー故障に対してもケーブル断線だけでなく、データ間隔等検出機能のレベルを上げる処置を追加した。またこれに先立ちモータの省電力化機能を実装した結果、ノイズの発生、瞬間的逆転の発生頻度があがり、機器異常の判断不具合が顕在化し警報を連続して発報していた。

	・異常事象の可能性だけを考えそうした事象の発生確率の算出とその程度に鑑みた対策実施の必要性判断を経ないまま改造作業がなされていた。 ・検出レベルを上げる際にはシステム停止、再起動アルゴリズムの改造において例えばカウントアップ値が異常に増加減したらノイズによるエラーと判断する等のロジックも組み込まれてはいたが、不十分かつ誤りがあることもわかった。 ・ノイズが乗りやすくなったかもしれないことは機構担当者は想定できたものの、事象発生頻度も高くないものと考えまたシステムとしての所掌が不明確なこともあり他メンバーには話さなかった。 
上記の未然防止に向けた対策	直接原因への対策： ノイズによる一時的なセンサー使用不可状態から復帰する際のシステム停止、再起動アルゴリズムソフトウェアを修正した。
	要因への恒久対策（対応工程を明記）： システム設計工程 ・信頼性向上施策を採る場合は、故障発生確率算出と対策効果評価を行い、システム全体としての対策実施の必要性を判断するようなプロセスとする。 ・ノイズ等対応を要する事象に対してその対処を実装する場合は、そのハンドリングや復帰の手順等にヌケモレがないように識者によるレビューを行う。 ・トータルでの責任範囲を明確にするとともに、メンバー間の情報共有・コミュニケーションを促進する。

	■システムアーキテクチャ設計 ・故障発生確率算出等の定量評価を行い対策効果を評価すること。 ■ソフトウェアアーキテクチャ設計 ・異常データを検知し適切に対処できるロジックを実装する ■レビュー ・ドメイン知識を有するメンバーよるレビューを必須とする。 ■プロジェクトマネジメント ・対策時は 2 次リスク（副作用）の発生を検討した上で実施判断する ・ハードウェア開発部門とソフトウェア開発部門のコミュニケーションを密にすること

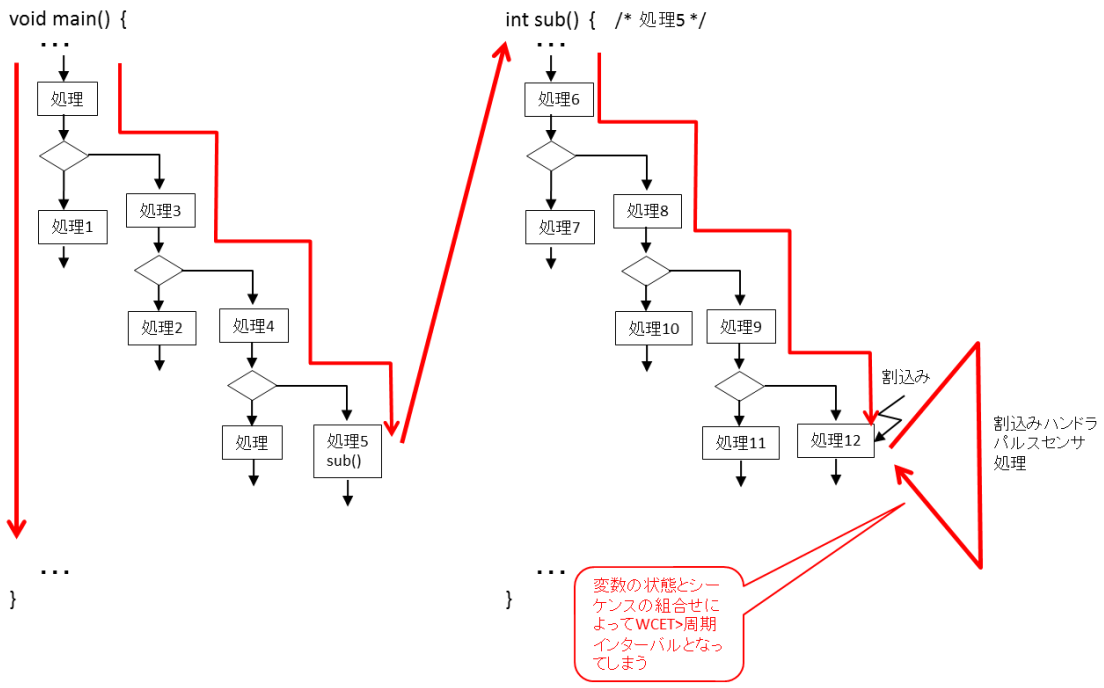
教訓 21

教訓タイトル	高い信頼性対策が求められるシステムでは重大な影響を及ぼす事象の想定と復旧手順を十分に検討する
製品の特徴	ネットワーク上に複数の機器が繋がっている。その各機器とそれらを管理する機器（管理機器）があり、各機器と情報のやり取りをしながら、システム目的を達成する制御システム。
観察できる現象	管理機器でシステムエラーが発生したが、管理機器の監視画面では正常稼働しているようにみえる。一方で、他の機器からは管理機器はエラー状態と認識されていて、実際には管理機器に情報を送信していないため管理機器の監視画面は更新されていなかった。
内部で発生した事象	・ネットワーク上の機器は全て専用の通信カードを介して通信している。通信カードは自身が接続されているホスト機器から定期的にイベントを受信し一定時間受信しないとエラーであると判断する。管理機器側の通信カードと各機器側の通信カード同志も互いの動作状況を通知する仕様となっていて、復帰後は相手に復帰指示を出す。 ・ある時、管理機器に接続されている周辺装置に故障が発生し、この影響で通信カードへのイベント全てが数分間滞り、一定時間応答受信できなかった管理機器の通信カードはシステムエラーになった。管理機器 MPU ユニットにエラーのイベントを通知し、ネットワークに繋がっている全ての機器にも通知した。各機器は管理機器からの復帰の通知があるまで、管理機器への通信が停止したままの状態になった。  ・その後、管理機器ユニットは自動復帰し、通信カードも自動的に再起動したが、通信カードが MPU ユニットに通知したシステムエラーのイベントが消失していたため、管理機器は通信カードにエラーが発生していたという事実を認識できず、各機器へ復帰の通知も出さず、各機器からの通信も停止したままになった。管理機器の通信カードのログに

	システムエラーの記録は残っていたがそれを参照する仕様にはなっていなかった。 
原因となる要因	周辺機器故障等で長時間イベントが処理されない状況になった際、システムに重大な影響を及ぼすシステムエラー等の情報を記録保持する仕組みになっておらず、再起動時に異常状態であったことを認識できなくなるケースがあることを想定できていなかった。
上記の未然防止に向けた対策	直接原因への対策： ・故障した周辺機器を交換した。 ・通信カードの起動時にログを調べ、システムエラー等重大な影響がある事象が発生している場合には該当する機器（本件では管理機器）に通知する。
	要因への恒久対策（対応工程を明記）： ・高い信頼性対策が求められるシステムでは重大な影響を及ぼす事象の想定と復旧手順を十分に検討する ・障害から復旧させるリスタート手順はコールドスタートとは異なる手順があるということを考慮する ■システムアーキテクチャ設計 ・故障からの復旧設計では部位や順序によって復旧できないケースがないかを検討する ・障害からの復帰シーケンスを十分に検討する ■ソフトウェアアーキテクチャ設計 ・通信復旧時には障害発生前の状態を確認し復帰させる ■運用 ・障害復旧手順は文書化して保守関係者に周知する

教訓 22

教訓タイトル	処理時間がクリティカルなシステムでは、ツールを活用し変数やその取りうる状態数とそれぞれの状況における動作処理に最大バラツキを意識し余裕を把握し設計する。
製品の特徴	・ある成型製品は溶融金属を成型加工しながら、薬剤処理等多くの工程を経て製造される。科学的組成が重要なため溶融や成型時の温度管理や加工時間の制約が厳しくかつ諸条件の組み合わせも複雑で、温度、位置、速度検知のために多数のパルスセンサーが設置され製造中間品の処理が管理されている。 ・この工程制御システムは上記多数のセンサー情報を受け加工アームのアクチュエータ位置やモータ回転制御等をリアルタイムに処理する必要があり、タイマ割り込みによるメイン一定周期処理を行っている。 ・工程制御のメインルーチンは 5ms 周期で動作し、このインターバル中に割り込み処理を受け付けたり、PID 制御に必要な I 要素、D 要素の計算の算出も行っている。
観察できる現象	増産体制の指示を受け製造品種の変更、増加が決定され、当該工場の製造ラインも段階的に変更が行われつつあり、ラインの増設や機器更新に伴い工程制御のソフトウェアも段階的に変更・追加された結果大規模化していた。その最中、加工成型時間の管理が厳しい工程において不良中間品が散発的に製造される事象が発生するようになり、ライン変更計画に支障を来し、少なくない機会損失を発生させてしまった。
内部で発生した事象	調査したところ、直近で実施されたソフト変更を行った際に、複数の処理を行うことで当該工程の処理をコントロールしているメインルーチンで行っている経過時間算出等の制御量計算にズレが生じ、この結果当該成型時の熱処理時間に影響を与えたためとわかった。
原因となる要因	さらに調査したところ以下のような状況であることがわかった。 ・ソフト変更の結果本システムで行う処理の組み合わせが増加しその状態の場合分けをするロジックも長くなり、熱処理における PID 時間を計算する際にメインルーチンのタイムインターバル 5ms を超えてしまう非常に稀なケースが存在することになり、計算値に論理的な矛盾が生じることになってしまった。

<table>
<tr>
<td></td>
<td>


・元々本システムは、外界センサー入力の変数が多く、その取りうる値域（状態の数）、及びそれら変数条件の組み合わせごとの動作シーケンスのバリエーションが非常に多く、仕様書にも全てのケースを記述しきれていなかった。また割り込み処理の負荷最大の加算も不十分な把握情況だった。
・担当者の経験不足もあり、本工程制御システムにかかる処理負荷の最大状態を設計的に予見しきれていなかった。またそのため検査仕様で複雑で負荷のかかるシーケンス組み合わせ試験が不足していた。（負荷テスト、限界テストが不足していた）
・さらに割り込み処理とメインルーチンの間でも同じ変数を処理（相互に書き込み）していたことで、まれな頻度で、同様な事案が起きる可能性があることが、確認作業の中でわかった。
</td>
</tr>
<tr>
<td rowspan="2">上記の未然防止に向けた対策</td>
<td>直接原因への対策：
問題となった PID 制御時間を行う際に、タイムインターバルを最悪でも超えることがないように処理を修正し、関連箇所についても同様の確認作業を実施。</td>
</tr>
<tr>
<td>要因への恒久対策（対応工程を明記）：
システム設計工程
・特にパルスセンサー処理では、メイン処理と割り込み処理の間の変数の割り込み干渉（Mutual Exclusion：相互排他問題）が発生することが常であり、基本的な配慮が必要。
・メインループの処理時間がクリティカルなシステムでは、変数やその取りうる状態数とそれぞれの状況における機能動作処理時間にバラツキが出るため、WCET（最悪実行時間：Worst- case execution time）を意識し把握した設計とする。</td>
</tr>
</table>

	・このために静的解析ツールを利用し確認するような手順を設計プロセスに組み込み、人による差を可能な限り抑止することが重要である。 ・排他処理、等時性、参照透過性等組込み系システムで重要となる基本概念をケーススタディ等を通じて理解習得を促進させる。 ■ソフトウェアアーキテクチャ設計 ・静的解析ツールを利用し確認するような手順を設計プロセスに組み込む ■ソフトウェアアーキテクチャ設計（変更設計） ・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り込み干渉や WCET（最悪実行時間）に留意する。 ■レビュー ・ドメイン知識を有するメンバーよるレビューを必須とする。 ■システムテスト ・複雑で負荷のかかるシーケンス組み合わせ試験による限界テストを行う ■教育 ・割り込みにおける排他処理等の重要かつ基本概念を理解させる

教訓 23

教訓タイトル	開発を伴わない保守案件でも、システム構成変更が発生する場合は，手順等作業内容の妥当性を確認できるようなプロセスを経る
製品の特徴	• 該当システムのハードウェア構成は、1 つの共有ストレージを 2 つのサーバで管理する対称型マルチプロセッシング方式を取る。2 つのサーバで現用系/代替系の二重化構成をとり、フェイルオーバーを実現する。 • 現用系/代替系のソフトウェアはいずれも、共有ストレージのデータベースを管理する商用 RDBMS と、商用 RDBMS を介してデータベースにアクセスして該当サービスを実現するアプリケーションから構成される。 • 共有ストレージは、データベース領域とファイルシステム領域で構成される。 • 該当システムは予防保全のため半年に 1 回、システム再起動を実施する（定期保守作業)。システム再起動は運用コマンドを使用し、以下の手順で実施する。 1. 現用系サーバのアプリケーション停止 2. 現用系サーバの商用 RDBMS 停止 → 起動 3. 現用系サーバのアプリケーション起動 • ある定期保守作業が終了した 1 か月後に保守案件実現のため、データベースの一部を運用コマンドにてファイルシステム領域に追加した。
観察できる現象	次の定期保守作業にてシステムを再起動したところ、現用系/代替系とも立ち上がらず、顧客サポートサービスのカスタマーコントロールや新規加入登録ができなくなった。
内部で発生した事象	現用系の商用 RDBMS が再起動したときにデータベースへのアクセス不可となり、通常起動できず。結果、アプリケーションも起動失敗し代替系に切り替えて起動を試みるも、同様に商用 RDBMS 再起動に失敗した。
原因となる要因	• 定期保守作業にて現用系のアプリケーションを停止し、ファイルシステム領域がアンマウントとなり、現用系の商用 RDBMS 再起動時にデータベースへアクセスできず、メンテナンスモードとなったため。 • 既存機能へ影響が及ぶリスク回避のため、保守案件で利用するデータベースを、データベース領域ではなく、アプリケーションが起動する際にマウントするファイルシステム領域に作成した。 • サービスを実現する各種機能でアクセスする領域を確認し、上記対応で既存機能へは影響がないと思い込み、システム再起動テストは未実施。 • 問題となった保守案件は運用コマンドでの対応のため、通常の開発プロセスとは異なる作業プロセスを適用し、要件定義書、手順書、テスト項目のみの作成に留まっていた。 • 既存ドキュメントに、システム再起動時に関する記述がない。

上記の未然防止に向けた対策	直接原因への対策： • 保守案件でファイルシステム領域に追加した一部のデータベースを、データベース領域に再登録。 • データベース領域、ファイルシステム領域の確保と参照の順序性について記載したドキュメントを作成。 • 必須のテスト項目として、システム再起動テストを追加。
	要因への恒久対策（対応工程を明記）： （プロジェクトマネジメント） ・開発を伴わない保守案件でも、システム構成変更が発生する場合は、手順等作業内容の妥当性を確認できるよう通常の開発プロセスを適用する。 ■レビュー ・ドメイン知識を有するメンバーよるレビューを必須とする。 ■システムテスト ・データベースバージョンの相違による影響をテストで確認する ■プロジェクトマネジメント ・保守対応でも作業内容の妥当性を確認できるようなプロセスとする ■運用 ・保守対応員へ事前に情報提供を確実に行うこと

教訓 24

教訓タイトル	物理量（時間、重量等）を扱う場合は単位、桁数を確認する。
製品の特徴	電子機器製品の長時間稼動試験の状況をカメラで連続撮影しつつ、各種イベントを発生させたときに電子機器製品の表示器の画像が確実に切り替わることを自動判定する機能を有する試験装置である。PC と複数の組込み機器（信号発生機器、カメラ等）で構成される。
観察できる現象	試験装置としては、電子機器製品の表示器の画像の切り替りに応じた正常な判定をするが、判定のエビデンスとして出力する判定対象の画像ファイル名を誤って出力する場合が稀にある。画像ファイルはデータサイズが大きいため、判定対象以外のファイルをテスト実行後に自動的に削除する機能があり、エビデンスが保存されないという問題が発生した。
内部で発生した事象	試験装置は、カメラが連続撮影した画像（テスト開始からの経過時間を示すタイムスタンプ付[単位：ms]）を受信し、“Pic_<試験番号>_<カメラのタイムスタンプ値>.bmp”というファイル名で保存する。一方、試験装置内部では、秒単位（浮動小数点数）で時間管理を行っているため、カメラのタイムスタンプ値を秒単位（浮動小数点数）に変換して各種判定を実行している。テスト及び判定実行後、試験装置は、保存された画像ファイル群の中から、時間が一致する画像ファイルを探索して判定対象画像ファイル名として出力する。このとき、浮動小数点数の誤差の影響で一致するファイルが存在せず、エラーを出力していた。
原因となる要因	判定対象の画像ファイルを探索する処理で、試験装置内部の時間情報に浮動小数点数誤差が生じたため、一致する画像ファイルを検出できなかった。 例、a.画像ファイル名：Pic_3_8139.bmp b.試験装置内部で記録していた画像収録時間：8.1389999･･･S（double） ⇒bの値を1000倍した値（double型）と、aの時間情報である8139（long型）を単純に比較してしまったため、一致しなかった。
上記の未然防止に向けた対策	直接原因への対策： ・試験装置内部で記録していた画像収録時間（double）を1000倍した後、小数点以下1桁目を四捨五入し、long型に変換してから、long型同士で比較する。 ・浮動小数点誤差の考慮漏れがないかチェックシートのチェック項目として追加する。
	要因への恒久対策（対応工程を明記）： 【設計工程】 ・物理量（時間、重量等）を扱う場合は単位、桁数を確認する。

	■システムアーキテクチャ設計 ・物理量の計算では単位及び単位系の違いに留意する ・小数点を含む処理では計算機依存の要素（型と有効桁数等）に配慮する ■実装（コーディング） ・データ生成時と参照時では型を揃える ■システムテスト ・小数点を含む場合には小数点誤差に配慮したテストを行う

教訓 25

教訓タイトル	顧客が要求していることの目的と背景に遡って、その意図を確認することが、要求仕様のあいまいさ排除に役立つ
製品の特徴	顧客からの注文を受付けてから、梱包・発送する製品があり、この受注登録と発送管理を行うシステムがある。顧客が求める所定の発送日に指定の拠点センターや宛先に製品が発送されるようこのシステムでは、(1) まず配送指定日の順にスケジューリングし、(2) 発送日が重なった場合には顧客からの注文を受け付けた順番に発送する。この順番通りになるように作業スケジューリングを行い、出荷指示を出す。 例えば下記のような状況の場合、A→B→C→D→E　の順番になる。
観察できる現象	・ところが近年、作業工程では必ずしも予定した通りに作業が進まないことが多くなり、出荷業務が混乱するようになってきた。ある時、現場責任者から「予定はあてにならない。当日内であれば出来た順にどんどん出荷できるように至急改造してほしい」との要求を受けた。 ・この要求に基づき、情報システム部門で本システムを担当している N 氏は他改善業務も担当していて多忙であったため、外部の協力会社社員 X 氏に本改造を依頼した。X 氏は本業務の経験は多くなかったが、改造は難しい内容と思えず、また最近の出荷状況の実績データも渡してあったのでこれらでテストすれば可能と考えた。 ・こうしてスケジューリングルールを改造したモジュールを適用したところ、目論見と異なる状況になり大きな混乱を招き、トラック配送業者からもクレームを受けるという事態になってしまった。

受付日	受付時間	発送指定日	出荷予定	顧客名
8 月 2 日	9:30	8 月 10 日	0810-0900	A
8 月 2 日	13:00	8 月 10 日	0810-1000	B
8 月 2 日	16:00	8 月 10 日	0810-1030	C
8 月 3 日	11:30	8 月 10 日	0810-1200	D
8 月 3 日	14:00	8 月 10 日	0810-1430	E

<table>
<tr><td>内部で発生した事象</td><td>下記に問題が起きた状況を示す。

・出荷予定はあてにならないということで準備完了フィールドを設け、実際の作業完了タイミングでこのフィールドに完了時刻を記録し、出荷予定フィールドは参照しないようにした。

・改造後システムは準備完了フィールド中の完了時刻の出来た順に出荷指示をすることになり、この結果 5 月 22 日は B→A→C→E→D の順のスケジュールとなっていた。しかし、現場の意味していたのはそうではなく、A→B→C→D→E であった。

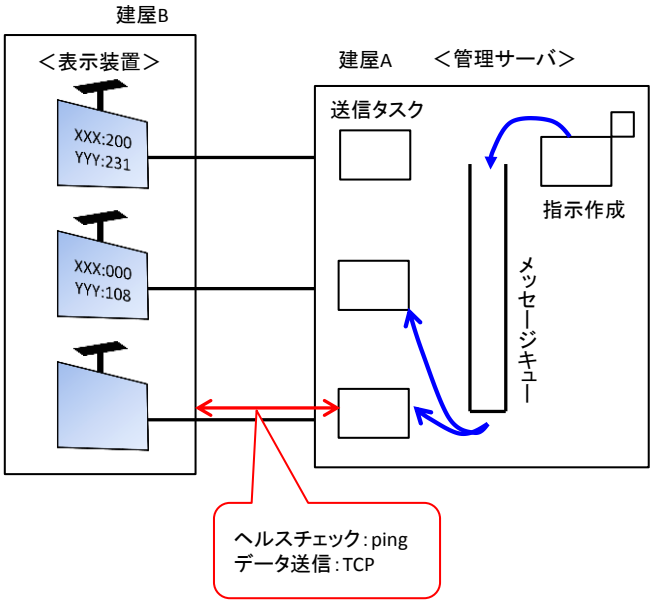
<table>
<tr><th>受付日</th><th>受付時間</th><th>発送指定日</th><th>出荷予定</th><th>準備完了</th><th>顧客名</th></tr>
<tr><td>5 月 10 日</td><td>12:30</td><td>5 月 22 日</td><td>0522-1000</td><td>0522-1115</td><td>A</td></tr>
<tr><td>5 月 11 日</td><td>9:00</td><td>5 月 22 日</td><td>0522-1100</td><td>0522-1100</td><td>B</td></tr>
<tr><td>5 月 11 日</td><td>14:00</td><td>5 月 22 日</td><td>0522-1300</td><td>0522-1400</td><td>C</td></tr>
<tr><td>5 月 12 日</td><td>11:30</td><td>5 月 22 日</td><td>0522-1500</td><td>0522-1715</td><td>D</td></tr>
<tr><td>5 月 12 日</td><td>15:00</td><td>5 月 22 日</td><td>0522-1700</td><td>0522-1700</td><td>E</td></tr>
</table>
</td></tr>
<tr><td>原因となる要因</td><td>・X 氏は“出来た順”という言葉の意味を「準備完了フィールド中の最も古いつまり先に更新された順」と解釈してスケジューリングルールを変更していた。
・現場責任者は 1 時間以内の差であれば配送業者へのトラック手配の都合もあるので、当初のままでよいと考えていた。当然これは、(2) 発送日が重なった場合には顧客からの注文を受け付けた順番に発送する　という要件を逸脱していたが、これまでの経験で問題ないと責任者はわかっていた。
・経験のある N 氏であればそうした現場の事情を理解できていたが、経験のない X 氏には理解できておらず、そうした詳細について N 氏も説明が不十分であった。
・N 氏より受領した実績データも、今回のように完了時刻が大きく遅延している状態を示しているデータがなかったため、テストからももれていた。</td></tr>
<tr><td>上記の未然防止に向けた対策</td><td>直接原因への対策：

・実際の準備完了時刻に配送業者指示の都合も加味した出荷指示となるようソフトウェアプログラムを修正した。</td></tr>
</table>

	要因への恒久対策（対応工程を明記）： 要求定義工程 ・言葉の意味、解釈では齟齬を生じやすいため、変更要求を受領する場合は、「仕様確認票」（下記項目含）を用い顧客意図に遡って確認する。 ［目的と背景］：仕様変更・追加の背景事情や理由等を記述 ［要求内容］：変更要求仕様を記述 ［変更仕様］：上記要求に対するプログラム変更内容を具体的に記述 ［確認署名］：内容確認後の承認印またはサイン ・過去経緯や現場で慣習的に運用されている内容は、文書化して残し、実務担当者に理解できるよう教育、説明する。 ■システム要求定義 ・要求仕様はその意図や背景に遡って確認しあいまいさを排除する ・背景事情や慣習等も含め重要事項は文書化する ■レビュー ・ドメイン知識を有するメンバーよるレビューを必須とする。

教訓 26

教訓タイトル	遠隔地等物理的に離れた装置をネットワーク接続して稼働させるシステムでは、故障等の状態検知やメンテナンスも容易ではないため、システム的視点での状態把握を行う。
製品の特徴	ある商品の配送・仕分けを行う物流管理システムで、配送品を仕向地や配送品の種類、配送法等の種別によって自動的に振り分け、コンベアを動作させながら、仕分け配送員に指示表示を行う。就業中の業務中断が起きないよう管理サーバは 2 重化されている。表示装置はシングル構成となっている。
観察できる現象	ある時システムの構成変更に伴いあるエリア向けのラインを別建屋に移設した。そのしばらく後に、移設したラインの表示装置の一つが表示されなくなり、正しい仕分け作業ができなくなる事態が発生した。事態を確認しようとしているうちに当該エリアの表示装置の多くで表示内容が切り替わらないという状態となり配送センターの稼働率に影響を与えてしまった。
内部で発生した事象	調査したところ、最初に配送指示ができなくなった表示装置がハングアップしていることがわかり、また管理サーバ内の作業指示表示処理が全体的に遅滞していることがわかった。
原因となる要因	さらに確認してみると、以下のような状況であることが判明した。 ・このシステムは表示装置の故障に備え、管理サーバから定期的にヘルスチェックを ping で行い、メッセージデータの送信は TCP 上のアプリケーションで行われていた。 ・最初に異常となった表示装置はハングアップしていたが、LAN ドライバは生きている状況であったため、ヘルスチェックには応答が返されていた。このためサーバ側はこの装置は故障と判断せず作業指示メッセージを送信しつづけた。 ・ヘルスチェックでは応答があっても、表示装置自体はハングアップしていたため、送信メッセージを受け取らずメッセージキューに送信メッセージが滞留しつづけたためとわかった。 ・別建屋となったため状況の迅速な確認と対処も遅れてしまったことも混乱を助長した。

<table>
<tr><td></td><td> 
</td></tr>
<tr><td rowspan="2">上記の未然防止に向けた対策</td><td>直接原因への対策：
・送信メッセージキュー長とその変化をモニタして管理するように変更した
・ping によるヘルスチェックだけでなく、送信処理（TCP）アプリケーションでの相手装置の状況確認を行うような修正を検討する。</td></tr>
<tr><td>要因への恒久対策（対応工程を明記）：

・遠隔地等物理的に離れた装置をネットワーク接続して稼働させるシステムでは、故障等の状態検知やメンテナンスも容易ではないため、システム的視点での状態把握を行う。

■システム要求定義
・遠隔地機器に対する保守・メンテナンス仕様を検討する

■システムアーキテクチャ設計
・遠隔機器を監視する手段の信頼性を十分検討する

■システムテスト
・遠隔機器の障害を模擬した異常テストを行う</td></tr>
</table>

教訓 27

教訓タイトル	マルチベンダーシステムでは仕様に外れた想定外事象が発生することを前提とした自己防衛策を採る。
製品の特徴	ある公共的な業務を行うシステムがあり、これらは各拠点等に設置されている多数の端末装置と、センターサーバで構成されている。業務データの送受信量も多く、業務によっては即時性が要求される。またこの業務システムはマルチベンダーであり、他社製品も本ネットワークに多数接続されている。
観察できる現象	業務処理の追加と変更を行うことになりプログラムを変更しこれに伴い端末装置側のプログラムも拠点ごとに順次更新作業をしていたところ、更新した端末装置が一斉に異常停止してしまう事態が発生した。これらの多くはリブートで復旧したが、一時的にとはいえ公共業務の運営に支障を来す事態となった。
内部で発生した事象	ログ等を調査したところ、あるタイミングで不正フレームが外部から配信された形跡がありこれが原因であると判明した。
原因となる要因	・端末装置のメッセージ受信プログラムでは、送信フレーム中の制御フィールドを以下のように解釈して関係アプリケーションを行う処理になっている。またチェックサム値も確認する。 ＜制御フィールド＞　＜解釈＞ 1　アプリケーション A　} 従来仕様範囲 2　アプリケーション B 3　アプリケーション C　} 新規追加 4　アプリケーション D 5　追加予定アプリケーション X ・このデータ送受信においては、1 か 2 のみが送信されるというのが従来のシステム全体の共通仕様と定義されていたため、制御フィールドが 1 でも 2 でもない場合には棄却するルーチンになっていた。 ・一方、業務処理変更に伴い新たなアプリケーションとその対応フィールド値が追加され、その時の状況に応じて処理を振り分けながら実行するという仕様が追加されたため、端末側の処理ルーチンを変更した。さらにアプリケーション X の追加が予定されていたため将来に備えて拡張性を盛り込んでいた

	・一方アプリケーション X の仕様はなかなか確定せず、そのフィールド値（5）を除外するようなコーディングが未実装となり、テストも不十分な状態であった。 ・当日の外部から送信された不正フレームの制御フィールドを確認したところ、未確定のはずのフィールド値（5）となっており処理が異常停止した。
上記の未然防止に向けた対策	直接原因への対策： ・当初仕様である制御フィールドが所定値以外のフレームは不正として棄却するようにプログラムを修正した。
	要因への恒久対策（対応工程を明記）： 要求定義工程 ソフトウェア設計工程 テスト工程 ・不正フレームを配信したのは自社以外の装置システムと推察されたが、追求が困難であった。マルチベンダーシステムではこのような仕様外事象も障害要因のひとつと考え、自己防衛策（仕様外の場合に備えた例外処理）を確実に実装しもれなくテストする。 ■システム要求定義 ・マルチベンダー環境におけるリスクを想定すること ■ソフトウェアアーキテクチャ設計 ・仕様外事象時のエラー処理を十分に検討する ■システムテスト ・境界値テスト項目にヌケモレがないようにする

教訓 28

教訓タイトル	データベース等 COTS*製品のバージョン、動作仕様の相違等の情報が関係者にタイムリーに参照できるようにする
製品の特徴	某団体向けの基幹業務システムの一つで、24H のノンストップ運転が必要なシステム。全国に何ケ所も同様の業務システムが稼働しており、それぞれのシステムは主系・従系の 2 重化システムとして構成されている。
観察できる現象	業務運用仕様の変更に伴い、ある拠点の 2 重化システムの主系ソフトウェアプログラムの修正を行ったところ、主系システムが異常になり従系システムに切り替わった。システム全体の稼働停止には至らなかったが両系停止すれば、当該業務に支障を来してしまい大きな経済損を出してしまうところであった。
内部で発生した事象	調査したところ、データベースシステムのログに書きだされるメッセージをモニタしている監視アプリケーションによってログ領域溢れと判定され系切り替えが発生していることがわかった。
原因となる要因	・このデータベースはバージョンによってログに出力するメッセージ内容や記録対象に微妙な違いがあり、かつどのようなメッセージをどの程度まで記録させるか等を構成ファイルのパラメータで設定することもできる。 ・従来よりデータベースのメジャーバージョンが変わった場合には、その環境での動作試験を行っていたが、マイナーバージョン変更の場合には必ずしも全ての詳細項目のテストを実施していなかった。 ・今回業務仕様の変更に伴い、データベースのマイナーバージョンアップを実施したのだが、この詳細内容を認識できていなかったため、パラメータ修正せずに稼働させたため、メッセージの採取量が従来に比し増加したことが原因であると判明した。

上記の未然防止に向けた対策	直接原因への対策： ・次回のメンテナンスタイミングでパラメータ変更するまでの暫定処置としてログファイルを定期的に削除する処理を追加した。
	要因への恒久対策（対応工程を明記）： ・現地変更作業では、当日作業手順や、不測事態への対処についての事前確認を確実に実施する。 ・データベースのバージョンによって異なる挙動やパラメータ設定仕様の相違を専門組織による事前検証や、有識者レビューで確認することも検討する。 ・COTS*のバージョンによる動作や仕様の違いについて、社内ポータル等で既知情報を提供し設計関係者が常に閲覧・確認できるような環境を整えることが有益である。 *COTS：Commercial Off-The-Shelf　商用既製製品 ■システムテスト ・COTS のバージョンによる影響を事前に検証する ■教育 ・COTS 技術情報は容易にアクセスできるようにする ■プロジェクトマネジメント ・ブラックボックスであることを前提とした開発プロセスを考慮する ■運用 ・不測事態への対処を計画する

教訓 29

教訓タイトル	複数の事業体にまたがる重要システムでは関係者の立場・ニーズの視点から、想定しうる障害発生リスクを同定し効果的な危機管理体制を構築する
製品の特徴	北米の広域にわたる地域をカバーする総延長 200,000 マイル（950,000 メガワット）に及ぶ電力送電網(Power Grid)システム。
観察できる現象	2003 年 8 月 14 日、北米地域での大規模停電が発生した。5000 万人が停電の影響を被り、被害コスト総額は 40～100 億ドルに及んだ。
内部で発生した事象	当日、北オハイオ州にある FirstEnegy 社の 5 つの発電システムが過負荷になり自動停止が発生した。この時、送電網監視システムは警報の発報に失敗し 1500MW の負荷が不平衡にあることが通知されず、送電線上に電力サージが流れ、加熱した送電線が周囲にある伸びすぎた木立の枝に垂れ下がって接触し送電が停止した。この結果、当該送電線に関係する送電網が連鎖的に停電する事態となった。
原因となる要因	調査したところ背景要因含め以下のような状況であることがわかった。 ・UNIX で作られていた FirstEnegy 社の送電網監視システムで警報発報に際してエラーが生じ、処理されないイベントが開始され 30 分以内に主系サーバがクラッシュし従系を含む関係システムに異常が伝搬した。 ・コンピュータ画面の応答が遅くなり、FirstEnegy 社の IT 要員はシステムがクラッシュしたことを認識したが、運転員には知らせなかった。運転員は顧客からの電話連絡を受けて異常を知るに至った。 ・FirstEnegy 社の運転員と地域統括センターの要員は当該システム全体に関する理解が不十分であったため、適切な対処を行うことができなかった。 ・送電線周辺の木立は送電線が垂れ下がって短絡事故が起きることがないよう剪定することが定められているが、剪定されておらず垂れ下がった電線がひっかかってしまった。 ・FirstEnegy 社の運転員と地域統括センターの要員はシミュレーション等のトレーニングを受けてはいたが、ある地域網に存在していた脆弱性を十分に把握できておらず、非常時の対応についての訓練もできていなかった。
上記の未然防止に向けた対策	・改良された警報と正確かつ迅速に状況を診断できる機能、送電網の問題を迅速に確認することができるユーザインターフェースなどを搭載したコンピュータシステムにリプレースするなど、システム全般の見直しを実施。コントロールセンターには広範囲にわたるシステム監視のための“マップボード”が設置され、さらにコントロールセンターの代替となることが可能なバックアップセンターも建設された。

	・当該システムやネットワークの全てを十分に理解できるように運用手順書と教育プログラムを改訂した。 ・非常事態への対応計画が策定され、木立の剪定作業実施も強化された。 本件から得た NASA の教訓： ・全体設計要件は、当該ミッションを果たす関係者のニーズと合致していなくてはならず、またシステム全体の運用パフォーマンスに関する情報を正確かつリアルタイムに提供するものでなくてはならない。 ・そのミッションに関して、可能な限り全ての不測の事態を考慮することが、チームの状況判断能力を向上させ、効果的な危機管理計画の策定・運用の助けとなる。 ・運用関係者に当該システム全体に関して確実に理解させることが、被害の連鎖拡大を低減させることを可能にする。 ・業務ミッションの成功において、チームコミュニケーションの大切さは最大限に強調するべきである。 （出典：NASA SYSTEM FAILURE CASE STUDIES, https://nsc.nasa.gov/SFCS/SystemFailureCaseStudy/Details/16） ■システム要求定義 ・障害発生時の対処を考慮した全体システム要件を明確にする ■システムアーキテクチャ設計 ・運用状況情報のリアルタイム表示など障害発生時に必要なシステム機能に留意する ■教育 ・運用関係者に当該システム全体に関する知識を確実に理解させる ■プロジェクトマネジメント ・システムの安定的な運用に関わる部門やチームのコミュニケーション向上を図る ■運用 ・運用手順書などは確実に文書化し関係者に周知する

教訓 30

教訓タイトル	過去のハードウェア、ソフトウェア資産を使用する場合は、その内容や当時の方法について考慮する
製品の特徴	スペースシャトル通信システム： スペースシャトルと地上局との無線通信は、打ち上げ時には S バンド（1,700～2,300MHz）を使用し、軌道上では Ku バンド（15,250～17,250MHz）に自動的に切り替わる仕組みになっている。一度 Ku バンドに切り替わると S バンドは Ku バンドの異常に対処するためのバックアップとして使用できるようになっている。 またシャトル本体の中央部にあるペイロード（貨物）とシャトルとの通信は PSP（Payload Signal Processor）を介して行われる。これは無線もしくはケーブルによるリンクが確立される仕様になっている。無線は貨物を移動させる際にまたケーブルは貨物室に貨物が存在する際に使用される。
観察できる現象	2008 年 11 月 14 日に打ち上げられたスペースシャトルエンデバーにおいて、シャトルが軌道上に達した時、管制局は 2 つの通信機構が打ち上げモードから軌道上モードに自動的に切り替わっていないことに気がついた。無線通信は Ku バンドではなく S バンドのままであり、貨物室との通信はケーブルに切り替わらず無線のままになっていた。幸いにも S バンドと Ku バンドの切り替えと、貨物のケーブルへの切り替えは手動操作により実施することができ、無事帰還することができた。
内部で発生した事象	・通信システムソフトウェアに潜んでいたバグが原因であったが、これはこの打ち上げに先立つ 1989 年のプログラム変更以降の数回にわたる変更作業により作り込まれたものであった。 ・シャトルのソフトウェアはデータと出力コマンドのセットが、compool（command data pool)と呼ばれるコードブロックに収められている。この際出力コマンドは compool の偶数アドレスに配置される決まりになっている。コードは慣習的にフルワード配置が使用され、このフルワード配置はデータアドレスを指定するために 4 バイトを使用する。 ・通信の自動切替を実行するために、管理システムは compool からコマンドを抽出して当該通信機能を制御する GCIL（Grand Command Interface Logic）と呼ばれるモジュールにコマンドを送出する。
原因となる要因	・1989 年の変更作業において、新たなコードが追加された。これは GCIL へのフルワード配置がなされていたが、出力が必ず偶数アドレスになるようなテクニックは使用されなかっ

	た。その代わりプログラマは、将来の変更に際して留意が必要であることをコード中に警告メッセージとして残していた。（コーディング標準では、プログラマは上記テクニックを用いることが要求されていたが不明瞭な記述だった） ・2000 年の変更では、compool の変更履歴はコードの最後部に記載するようルールが変更され、再配置された。しかし 1989 年に記述されたコード中の警告メッセージは、誰でも気がつくはずと考え元の位置の compool の先頭に置かれたままであった。この結果、この警告メッセージはコードそのものをレビューしない限り知ることができない状態になってしまった。 ・2007 年の変更では、compool にハーフワード（2 バイト）の新しいデータが追加された。この追加により 3 つの GCIL 出力コマンドが偶数アドレスから奇数アドレスに移動させられることになった。エンデバーにこの変更済プログラムが搭載された結果、GCIL は S バンドを Ku バンドに切り替えたり、PSP のモードを切り替えたりするコマンドを受け取らなくなってしまった。レビューやテストの段階でもこのコードに記述されていた警告メッセージに気付くことができなかった。 　また、貨物室の通信を無線からケーブルに切り替えるテストは 2007 年変更の時には実施されず、S バンド、Ku バンドの自動切替がうまくいかないことは把握されていたが、当該テストの目的とは関係ないとして記録されず、さらにその修正作業は別部門が行うものと仮定して放置された。
上記の未然防止に向けた対策	**明確にされた手順と訓練：** ・偶数アドレスに配置するテクニックは優れた手法であると認識はされていたが、適用が必須であることがコーディング標準に明示されておらず、レビューのチェックリストにもこれに該当する項目がなかった。そこでコーディング標準を改訂し、こうした全ての“優れた手法”を明記し教育内容にも反映させた。 **End-to-End の検証：** ・ソフトウェアの変更の際に行うテストにはいくつかの段階があるが、いずれの段階においても PSP に出力されるコマンドが正しいかどうかをチェックするテスト仕様になっていなかった。PSP のモード切り替えのような小規模機能のテストでも、コマンドが妥当かどうかモレなく確認するべきである。 **実際の飛行を想定した試験と訓練：** ・貨物室の通信を無線からケーブルに切り替えるテストや、S バンド／Ku バンドの自動切替のテストと実際の状態とでは差異が生じていた。テスト手順では実際の飛行を模擬した状

態で実施するべきであり、エラー検出可能な手法やシミュレーションを用いるべきである。

異常状態の記録：

・ミッションクリティカルなシステムのテストでは、全ての失敗結果を記録しフォローするべきである。もしテスト目的と合致していなくても、異常な結果については常に調査を行う。

過去の資産：

・方針、手順や手法は時間とともに変化する。過去のハードウェア、ソフトウェア資産を使用する場合は、それ以前の要求内容や方法について考慮することが必要である。

（出典：NASA SYSTEM FAILURE CASE STUDIES,
https://nsc.nasa.gov/SFCS/SystemFailureCaseStudy/Details/5）

■ソフトウェアアーキテクチャ設計（変更設計）

・ソフトウェア変更時の記録の残し方には一貫性を持たせる

・過去のハードウェア、ソフトウェア資産を使用する場合は、その内容や当時の方法について考慮する

■実装（コーディング）

・適用が推奨されるコーディング手法は標準として明記し順守させる

■システムテスト

・全ての失敗結果を記録しフォローする

・実際の動作を模擬した状態でテストを実施すること。またエラー検出可能な手法やシミュレーションを用いること

教訓 31

教訓タイトル	ミッションクリティカルシステムではリスク管理や V&V を確実に実施する
製品の特徴	火星気象観測衛星（MCO: Mars Climate Orbiter）: MCO は NASA の第二次火星探索計画の一環として開発された火星の気象調査のための衛星で、3 週間後に打ち上げられた Mars Polar Lander の通信中継局としても機能させるべく 1998 年に打ち上げられた。
観察できる現象	・打ち上げ後の 4 ケ月間に航法ソフトウェアに問題があることがわかり修正作業を行った。この修正が完了した時、オペレータはこの航行ソフトウェアフィイルに異常なデータがあることに気がついたが、非公式な話合いをしただけでその問題解決を放置した。 ・打ち上げの 9 ケ月後、火星周回軌道にのせるためナビゲーションチームは軌道修正用の姿勢制御ロケット噴射を計画した。予定では火星上空 226km の楕円周回軌道に入る予定だった。しかし噴射の 1 時間前に目標高度を再計算したところ 110km（危険高度のわずか 30km 上空）になることがわかった。 ナビゲーションチームは緊急噴射を実施して軌道修正することを検討したが、最終的には当初の計画通り行うことにした。緊急噴射した場合、軌道投入が当初計画とズレてしまい、Mars Polar Lander との通信に支障を来す可能性があり、緊急噴射実施のための検討も不十分で、変更準備も間に合わなかった上、MCO の予想高度も危険下限よりは高いと想定されたためである。そして軌道投入の噴射から 4.5 分後、MCO からの通信が途絶してしまった。
内部で発生した事象	・実際には MCO は火星大気に高度 110km ではなく約 57km をめがけて突入していた。 ・チームが調査したところ、火星に到達するまでの間に行われた軌道修正量は常に本来の数値よりも 4.45 倍大きい値になっていた。これは航行ソフトウェアにおいて、本来メートルで計算すべきところをヤード換算していたためであった。 ・この処理以外は全てメートルで計算していた。このような食い違いの結果、計算された値よりも MCO の機体は実際には火星に近い位置を目標にしてしまった。
原因となる要因	調査したところ以下の様な要因があることがわかった。 ・航行ソフトウェアの開発業者は本来メートル法で計算しなくてはならないところを、ヤード計算で出荷していた。 ・このソフトを開発したプログラマは Mars Surveyor のプログラムと互換性がある SIS

	（Software Interface Specification）を適用せず、試験員も適用確認をしていなかった。 ・V&V の実施においては厳格な実施が要求されているにもかかわらず、当該チェック作業が確実に実施されたことを示すエビデンスは残されていなかった。 ・ナビゲーションチームとオペレーションチームやプロジェクトマネージャ間のコミュニケーションが悪く、正式な報告手続きではなく非公式な話し合いで意思決定していたため、事故に至るまでの間に不具合に気付くチャンスがあったにもかかわらず、気付くことができなかった。 ・ナビゲーションチームは打ち上げ間近になって編成されたため MCO の航法やオペレーションについて熟知していなかった。また、オペレーションチームは Mars Global Surveyor、MCO、Mars Polar Lander の３プロジェクトを兼務し、非常に多忙でサポート体制も不十分であったので、仮に 226km の周回軌道に移動させる変更が必須だと理解したとしても準備不可能であった。
上記の未然防止に向けた対策	**リスク管理：** ・V&V ではミッション要求にダイレクトに関連づけた検証がなされなくてはならない。 ・重大リスクを認識させるためにもプロジェクト開始の早い段階から担当者をアサインすべきである。 **クリティカルタスクと定義責任：** ・プロジェクト計画においては、全ての重要な役割と情報が識別され、開発から実行段階に確実に引き継がれなくてはならない。 ・常に第三者による V&V（IV&V）が実施されることが求められる。 ・チームの一員として、他チームの役割とミッションへ及ぼす影響を理解すること。 **コミュニケーション：** ・どのような問題であっても適切なエンジニアリング規範に従い最優先課題としてエスカレーションされるよう、チームメンバー間の開かれたコミュニケーションを促進させることが大切。 **認知から行動へ：** ・問題があることを認識するだけでは不十分であり、各ミッションチームは以前のミッションからの教訓をどのように取り入れるのかを決定しなくてはならない。 ・自分のミッションとそれにおける役割の実行に繰り返し立ち戻り評価しなくてはならない。

	（出典：NASA SYSTEM FAILURE CASE STUDIES, https://nsc.nasa.gov/SFCS/SystemFailureCaseStudy/Details/144） ■レビュー ・ミッションクリティカルシステムでは第三者検証も含めた V&V を確実に実施する ・V&V ではミッション要求にダイレクトに関連づけた検証を行うこと ■プロジェクトマネジメント ・ささいな問題も見逃されることがないよう、チームメンバー間の開かれたコミュニケーションの促進を心がける ・重要な役割と情報が識別され、開発から実行段階に確実に引き継がれるような計画とプロジェクト運営とすること ■教育 ・対象システムの技術や運用に関わる知識のあるメンバをアサインする、あるいは既存のメンバに教育を行う

教訓 32

教訓タイトル	不測事態においても適切に動作するかの検証を十分に行い、条件変更時には潜在的なリスク許容度合いの変化を見逃さない
製品の特徴	人工衛星の航行システム： 人手を介することなく、自律的に衛星同士のランデブーを行うことが可能であることを実証するために 2005 年に打ち上げられた人工衛星（DART: Demonstration of Autonomous Rendezvous Technology）。この DART は MUBLCOM（Multiple Paths, Beyond-Line-of-Communications）と呼ばれる衛星とのランデブーを自律的に行うため、3 つの GPS 情報を用いる映像誘導センサー（AVGS：Advanced Video Guidance Sensor）を使用して、衛星の位置と速度を計算する。近距離では航行制御は全て AVGS で行う。
観察できる現象	DART は打ち上げ後、航行システムソフトウェアで発生した異常により、MUBLCOM とのランデブーのためのエンジンを過噴射してしまい、燃料を余計に使用してしまった。近距離飛行になると地上局ではミッション遂行に向けた燃料消費量が多すぎることを認識したが、全自動システムであるため何もすることができなかった。GPS データ処理のエラーによりエンジンの過噴射が続き、DART は MUBLCOM に衝突した。DART の燃料は不足し、結局ミッションを果たすことはできなくなった。
内部で発生した事象	DART の速度や位置を算出するために、航行システムは GPS と AVGS の実測値とソフトウェアで計算した推定値をある値（ゲイン設定値）で重み付けして比較する。推定値と実測値の誤差が許容値以上の場合、ソフトウェアはリセットして再計算する。調査したところ以下のようなエラーが発生していた。 ・エラー１： GPS 入力値とソフトウェア推定値との速度比較の誤差範囲は±1m/s となっていて、これを超えた場合、結果をリセットしなくてはならない。しかし、GPS の誤差範囲は±2m/s となっていたため、3 分ごとにリセットを繰り返す無限ループに陥っていた。 ・エラー２： ゲイン設定値はテスト工程の最後に発見されたユニット変換エラーに対応するために打ち上げ直前に変更された。これにより推定値に対して不適切に大きな重み付けがなされることになった。当初のゲイン設定値であれば上記の無限ループにならずに済んだはずだったが、変更したゲイン値によって無限ループしてしまうことになった。

	GPS を使用しない AVGS モードに切り替える地点があったが、過噴射により DART はそのポイントを過ぎていた。もし切り替えができていたら無限ループに陥ることはなかったはずであった。
原因となる要因	調査の結果以下の様な要因があることがわかった。 **不十分なソフトウェア要求仕様理解と検証：** ・無限ループがバグであることはわかっていたが修正されなかった。これは GPS 出力がソフトウェアによる推定値算出に使用されることが適切に文書化されず、周知もされなかったため DART チームの誰にも理解されず、テスト項目にもなっていなかったためであった。 ・大きなゲイン（重み付け）値となってしまったが、打ち上げ期日が迫っていたこともあり適切なテストや検証もなされなかった。 **不適切な設計の選択：** ・DART のコマンドシーケンスは過去のソフトウェア資産を利用して開発されたが、この資産は予期しない入力への対応ができず、自律制御には適していないものであった。 ・DART は地上局から制御できなかった。自律制御の設計方針にもかかわらず、エラーに対する余裕度や万が一への備えがなかった。 **経験、訓練及び管理監督不足：** ・政府及び開発企業からなる DART チームは経験と訓練が不足していたため、過去の NASA プロジェクトの教訓を生かせず不適切な設計、テストを行ってしまった。 ・DART はハイリスク、低予算なプロジェクトであり、ほとんど開発企業に依存し政府の管理監督がなされなかった。
上記の未然防止に向けた対策	・重要なフライトソフトウェアは早期の段階で以下のような観点で仕様検証することが便益をもたらす。 当該ソフトウェアは必要な機能を実行するか。 〃　　は不必要な機能を実行することはないか。 不測の事態においても適切に動作するか。 ・適切なテストや手順、安全策が取られ、文書化されているか、コミュニケーションが図られているかを保証する上で、第三者検証・アセスメントや相互レビューは有用なツールとなる。

	・条件変更によるリスク許容度合いの潜在的遷移に対応するために、プログラムやプロジェクト管理を定められた手続き通りに実施しリスクレベルを確認すべきである。 ・プログラムやプロジェクトチームは、当該事項に詳しい経験者を入れるとともに、NASAの過去の経験を十分に活かすべく、以前の教訓を適用すべきである。指定されたツールや参照情報、有用と思われる教訓は、他のプログラム・プロジェクトチームにとっても便益をもたらすことができるように認知されるべきである。 （出典：NASA SYSTEM FAILURE CASE STUDIES, https://nsc.nasa.gov/SFCS/SystemFailureCaseStudy/Details/12） ■システムアーキテクチャ設計 ・不測の事態の想定を行い、そうした事態に対処可能な設計とすること ■ソフトウェアアーキテクチャ設計（変更設計） ・条件変更によるリスク許容度合いが潜在的遷移していないかリスクレベルを確認する ■レビュー ・第三者検証・アセスメントや相互レビューを確実に実施し、過去の教訓を設計やテストに反映させる ■システムテスト ・適切なテストや手順を用いかつ文書化を行うこと ■プロジェクトマネジメント ・条件変更によるリスク許容度合いの変化を見逃す事のないように、定められた手続きを確実に実施し変更後のリスクレベルを確認する

教訓 33

教訓タイトル	不十分な設計となっている回避策は根本的に見直す
製品の特徴	空港の航空交通管制官と航空機との通信を行うための音声切替制御システム（VSCS: Voice Switching and Control System）。航空管制官はこの VSCS タッチスクリーンを使用して、運航乗務員と話をしたり他のコントローラや無線周波数に接続したりするための電話回線を選択する。
観察できる現象	2004 年 9 月、ロサンゼルス国際空港と当該地域の他空港がカリフォルニア州の FAA の無線システムの障害のために業務を停止した。VSCS が故障した場合に備え、プライマリシステム障害時に引き継ぐことになっていたバックアップ· システムもクラッシュした。幸いなことに、飛行中の民間航空機上にある衝突回避システム（CAS）が役立ち、この停止期間中の大惨事発生を回避することができた。この通信機能停止によって（150 フライトキャンセル含む）約 600 便、30,000 人の乗客が影響を受けた。
内部で発生した事象	この VSCS 異常は 3 時間続いた無線の故障によるものである。VSCS システムはデータの過負荷を防止するために約 50 日ごとにリセットする必要があり、現地技術者が 30 日ごとにリセットする作業を行っていたがこの作業に失敗したため、VSCS がシャットダウンしさらにこのシステムへのバックアップ切り替えも失敗した。
原因となる要因	このシステムのソフトウェアは、1ms ごとにデクリメントする、32 ビットのカウントダウンタイマーを使用していた。これはカウンタがゼロに達するまで $2^{32}-1$ms の間のタイマ値をとることを意味し、ゼロになるとソフトウェアはシャットダウンする。 そのため、カウンタが 2^{32} ms（約 50 日）に達する前に、手動による無線システムのリセットが必要であった。カウンタがゼロに達したときに人手でリセットされていない場合、警告なしに VSCS をシャットダウンするようにプログラムされていたのだが、これはそもそも設計上の誤りと考えられる。
上記の未然防止に向けた対策	・FAA は後に、人間の介入なしにカウンタを定期的にリセットするソフトウェアパッチを適用した。 NASA の教訓： ・不十分な設計となっているソフトウェアによる回避策を設計者は受け入れてはならない。

	・その回避方法がどのタイミングで重要システムに影響を与えるかを識別する能力に対して、システムとしての視点が求められる。 （出典：NASA SYSTEM FAILURE CASE STUDIES, https://nsc.nasa.gov/SFCS/SystemFailureCaseStudyFile/Download/535） ■システムアーキテクチャ設計 ・システムの視点に立ち重要システムに与えるリスクを識別すること ■ソフトウェアアーキテクチャ設計 ・不十分な設計となっているソフトウェアによる回避策は根本的に見直すべきである

教訓 34

教訓タイトル	重要なソフトウェアを変更する際は、変更管理を確実に実施する
製品の特徴	太陽観測衛星 SOHO(Solar Heliospheric Observatory)は、1995 年に太陽と太陽風を研究するために打ち上げられ、1997 年にはその活動を 2003 年まで延長することになった。
観察できる現象	SOHO には 3 つのジャイロがあり、これらは機械的な摩耗、角度変更、大きな温度変化などによるドリフト偏差を定期的に補正する必要がある。こうした機械的、温度的なストレスのため当初の予定期間を過ぎた後に継続動作することは困難であると予想された。そこで運用延長に対応できるよう、1998 年 6 月 24 日、姿勢検出用ジャイロスコープを効率的に動作できるよう変更されたソフトウェアが姿勢制御ユニット（ACU）コンピュータにアップロードされた。すると直ちに、SOHO を“安全モード”にするアラームが発生した。このアラームは地上局からの操作でしか解除できないものだった。 地上局は、ソフトウェア変更における不具合がアラームを出したことをつきとめたが、ジャイロの 1 つが回転を開始しないという別の重大な問題を見過ごした。回転停止用ジャイロを再調整しようとすると、SOHO の制御を失わせるような回転を与えるロケットエンジンの点火をもたらした。 1998 年 6 月 25 日 12:43 までに、SOHO の姿勢はズレてしまい全ての電源、通信、テレメタリ信号が消失した。懸命な作業を続けた結果、3 ケ月後になんとか回復させることができた。
内部で発生した事象	変更したソフトウェアではジャイロ C は通常運行時に回転速度を検出し、ジャイロ B は過回転を検出、ジャイロ A は保全用に回転停止する。過回転が検出されると“安全モード”に移行し、ジャイロ A がジャイロ C の代わりに回転速度検出を行うべく再び回転を開始する。 変更したソフトウェアには次の２つの不具合が存在していた。

ジャイロ	機能	正常時の使用	安全モード時の使用
A	回転速度検出	しない	する
B	過回転検出	する	する
C	回転速度検出	する	しない

	1．安全モードにおけるジャイロ A の再回転を行うコーディングが不注意により抜けていた。 2．ジャイロ B の過回転検出の設定値は、仕様の 20 倍も高い感度の値が設定されていた。 1998 年 6 月 24 日の変更により、すぐにジャイロ B は過回転と判断し安全モードに移行してしまった。地上局オペレータはすぐにエラーを修正したが、姿勢調整のための再較正を開始する前に、ジャイロ A の回転状態を確認することができなかった。回転停止ジャイロの計測値はロケットエンジン点火に変化を与えなかったため、他の安全モードが動作するまで SOHO の実回転数は上昇してしまった。 安全モード時にジャイロ C が動作しなかったが、オペレータはジャイロ B の異常検出が故障して停止しているせいであると間違った判断をしてしまい、ジャイロ A の正常時の回転速度の値に基づき、エンジン点火の再較正を再開してしまった。
原因となる要因	**変更管理の欠如**： ・プログラム変更は正規の文書化、試験に基づいておらず、当局の承認手続きも経て行われていなかった。また第三者による検証も未実施であった。 ・ジャイロの回転状態は、その回転速度が訂正されるべきであるか明確に地上局から把握できず、誤判断した。 **次手順に移行する上での問題**： ・安全モードに移行する手順では、リカバリしようとする前に、最新の 3 つのテレメータデータを評価しジャイロの回転速度を確認することと明示されているが、いずれもその手順は守られていなかった。 ・レビュー委員の承認を得ることなく、ジャイロ B の回転速度低下が実施されていた。 **無理なスケジュール**： ・6 月 24 日～29 日に計画されていた科学的活動スケジュールでは不測事態を検討する時間をとることができず、かつ追加のスタッフなしで行われた。 **不適切なスタッフアサインと教育**： ・航行運用チームは SOHO の設計と運用に関する正規の教育を受けていなかった。 ・SOHO の全体的な理解をしているメンバは 2 人しかおらず、ジャイロのコマンドシーケ

	ンスで使用されている言語の専門メンバはいなかった。
上記の未然防止に向けた対策	・航行上重要なソフトウェアを変更またはアップデートする際は、正規の手続きによる V&V を行い、実装前にはしかるべき承認を必ず得ること。 ・装置の「オン･オフ」の状態は、冗長性や信頼性の誤判断を防止する上で、間違った判断をすることがないほどに明確になっていなくてはならない。 ・実際には非アクティブであるにもかかわらず、センサーが有効であると誤って解釈することがないような設計としなくてはならない。 ・目的を達成できるかどうかは多くの場合、衛星自体の健全性と安全に依存する。延長フェーズで限られた予算という現実の中で、実行可能性検討が確実に行われなくてはならない。 ・手順を守ることは、ものごとの進行が適切であるか否かを判断するために不可欠であり、確立された手順を回避してはならない。 ・プログラムやプロジェクトは関係スタッフが当該衛星に関連する具体的な知識を持っていることを保証できるよう、きちんとした計画とコントロールを行う必要がある。 （出典：NASA SYSTEM FAILURE CASE STUDIES, https://nsc.nasa.gov/SFCS/SystemFailureCaseStudy/Details/10） ■システムアーキテクチャ設計 ・無効であるにも拘らず有効であると誤判断される可能性が排除されるような設計を行う ■プロジェクトマネジメント ・重要なソフトウェアを変更する際は、正規の手続きに基づく V&V を行い、実装にあたっては承認を得る ■教育 ・航行運用メンバに対して対象システムの設計と運用に関する知識を与える

教訓 35

教訓タイトル	リスク分析によるハザード識別を行い、非常時には関係者が即応できる体制を構築する
製品の特徴	オーストラリア・ヴィクトリア州　天然ガスプラント。
観察できる現象	某社の天然ガスプラントで、アブソーバー装置（天然ガスに含まれる一部の成分を吸着し除去する装置）の熱交換器が破損して内容物が漏洩、これによって形成された蒸気雲が着火・爆燃を起こし、付近の作業員 2 名が死亡、8 名が負傷した。 また、この事故により、この工場は全面的に操業を停止したため、ヴィクトリア州の都市ガス供給が約 2 週間にわたって停止、市民生活に大きな影響を与えた。
内部で発生した事象	プラント内のシリンダーに亀裂が生じ、そこから液状の炭化水素が地面へと漏れ出した。最初の段階では滴る程度であったが、事故当日の朝には流れるまでに至っていた。 そして、通常は高温であるはずのパイプが、氷が付着するほどの低温に至っていた。また、通常なら動作しているはずのポンプは停止しており、通常は安定しているはずである貯蔵タンクの液面が急激に下がった。この状態でオペレータが高温のリーンオイル注入作業を実施したところ、反応容器が破損し内容物が漏洩し気化した蒸気に引火爆発した。
原因となる要因	当初この会社は、当該要員へは十分な教育研修も実施しており本件はオペレータのミスによるものであると主張したが、詳細調査の結果、以下の様な状況にあることがわかった。 ・この会社が実施していた研修はガスプラント操作の知識を網羅するもので、研修では理解を確認するための記述テストが行われる。この回答内容の程度に応じて評価がなされ、不十分とみなされると再研修を行い、理解できたかどうかを再度尋ね、理解できたという回答をもって合格とされる。しかし、実際の現場では「それでもよくわからない」とは言いにくい雰囲気があり、さらに説明を求めるには勇気が必要であった。このため、エンジニアは不完全で表面的な学習状態にあった。 ・この会社は、同工場内に 3 つある製造プラントのうち 2 つに対しては HAZOP によるリスクアセスメントを行っていたが、最も古いこのプラントに対してのみ実施していなかった。これは、他 2 つのプラントが国の監督省庁の所管であり HAZOP 実施を求められていたのに対し、当該プラントは州所管でこうした対応が求められておらず、リソース不足もあり未実施だったためであった。もし HAZOP を実施していれば、低温による脆弱性を認識することができたはずである。 ・この会社の安全管理システムではマニュアルなどもあったが、不必要な相互参照や理解しにくい言葉などで書かれていた。この安全管理システムに対する監査も不十分な内容

	となっていて重要なハザードの識別を行うことができていなかった。また、悪いニュースが組織上層部に伝わらない風土が存在していた。 ・社内にはインシデント報告の仕組みが存在し経営幹部は毎日目を通していたが、主に個人への傷害に関わる事象報告に使用され、プロセス上の問題(process upset)はインシデントとして扱われておらず報告されていなかった。 ・警報システムは存在していたが、実際には警報は一日中出ている状態であり、操作員は警報が出続けている状態に慣れてしまい、何が重要で何が軽微なものかの区別もできなくなっていた。
上記の未然防止に向けた対策	・操作員のミスを重大事故の説明にすべきではない。 ・系統的にハザードを識別することは事故防止のために不可欠である。 ・操作員への研修は、プラント運転の手順およびその実践を含めた理解につながるよう適切に行われるべきである。 ・悪いニュースを識別し、それが経営上層部に伝わることを確認するために十分な監査が実施されなくてはならない。 ・重要なトラブル警報が識別され正しく認識されるように警報システムを慎重に設計する必要がある。 ・幹部管理職は、危険なプロセスの管理のための責任を負わなくてはならない。 ・非常事態に備える体制がすべての危険施設に適用されるべきである。 （出典 : Andrew Hopkins、「Lessons from Esso's Gas Plant Explosion at Longford」Australian National university） ■システムアーキテクチャ設計 ・重要システムでは網羅的にリスク分析を行い系統的にハザードを識別する ・重要な警報が正しく認識されるように警報システムは注意深く設計する

	■教育 ・関係者への教育は表面的でない実践上の理解につながるよう適切に行う ■運用 ・責任の所在を明確にし非常事態に備える体制がすべての危険施設に適用されること ・悪い知らせが経営層にタイムリーに伝達されるような監査体制とする

*参考文献

[1] 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
[2] 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
[3] 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
[4] 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

ＰＡＲＴ　Ⅱ

障害対策手法・事例集（組込みシステム編）

PART Ⅱ 目次

1.分類の体系

本章は、収集した未然防止知識教訓事例の開発現場での活用を容易にするため、各事例の分類体系を設け、その中にそれぞれの教訓の対策や活用法を整理したものである。この際、開発の各プロセスで参照されることを念頭に工程別の対応づけとそれぞれにおける対策事例と手法を示した。また、各教訓事例の持つ特性や原因等の要素に着目し、その要素から活用を考える状況を想定し観点マップとして整理した。用途に応じ適宜参照されたい。

2.工程別対策事例と手法

実際の障害事例から導かれる未然防止知識を教訓化する作業では、問題を引き起こした直接原因を分析し、問題を混入させたり流出させたりする真の要因（真因）を特定する。特定された真因はソフトウェア開発の作業や運用時の作業に潜んでいると考えられ、類似問題が起こらないように打つ真因への対策は、工程作業を定義する組織や企業の開発プロセスに織り込まれる。ここでは、PART I の未然防止知識を導いた 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ている。対策事例の多くは、IPA/SEC が発行している SECBOOKS の開発手法が参考になるため、SECBOOKS の略称と手法が解説された箇所を併せて紹介している。（注 1)。また、対策が導かれた過程を参照しやすいように元の未然防止知識に付した教訓番号を付している。

注 1）対策手法は、SECBOOKS[1][2][3][4]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。

表 1.1　工程別一覧

教訓番号	教訓タイトル	システム要求定義	システムアーキテクチャ設計	ソフトウェアアーキテクチャ設計	ソフトウェアアーキテクチャ設計（変更設計）	実装（コーディング）	レビュー	システムテスト	教育	プロジェクトマネジメント	運用
1	複雑な条件式のロジック変更を行う場合は、デシジョンテーブル等による検証が有効である			○	○						
2	条件が整理されていない状態で、トータルの条件数が100を超えるような機能、または10個以上の条件を有する機能を修正する場合、関連する条件を全て洗い出して整理し不整合がないことを確認する			○	○						
3	複数機能モジュールを統合する場合、統合前の条件数の総和と統合後の条件数を比較し差がある場合は、条件の抜けがないか確認する。				○			○			
4	変数値域が広く、組合せバリエーションが非常に多くなる場合には、値域を適切な大きさに分割した上で境界値テストを実施する				○						
5	内蔵電池を使用する場合には、深放電時の起動シーケンスを考慮すること		○	○			○	○	○		
6	フラッシュメモリを使用する場合には、書き込み寿命回数を考慮すること	○	○							○	○
7	消費電力の多い機能を追加する場合には、一時的な電圧降下による影響（リセット、フリーズ等）や電源の種類、電池の場合は残量を考慮すること		○								
8	想定可能な例外を形式的に漏れなく分析する	○	○								
9	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			○							
10	制御対象のハードウェアが同一でも、運用条件が変わるときは、ハードウェア仕様を再確認する		○		○		○		○		
11	プロセス間、スレッド間でデータを共有（引き渡し）する場合は、排他・同期処理が正しく行われているか、あるいはデッドロックが発生していないかどうか注意する			○		○			○		
12	歩留りのある製品の良品／不良品を検査する装置では、全てが良品あるいは、不良品との検査結果は異常と判断すべきである	○									○
13	既存ソフトウェアの性能改善を実施する際には、アイドリングタイムの発生、処理の同期ずれの発生等と影響を確認する			○	○			○	○	○	
14	・大量のデータを通信経由で扱う場合、一連の処理の流れの中にボトルネックを作りこまないように注意する ・時間帯による負荷変動について考慮する	○	○	○			○				
15	納入したあと、お客様が運用するような業務システムでは、業務シーケンス中のあらゆる異常操作（リセット、電源断、放置も含め）、への対応を考える				○			○			
16	障害解析時の保守メンテ用ログ処理であっても、仕様書を作成し、影響評価を実施すること			○							
17	判断処理は、必要条件だけでなく、制限すべき条件も漏れなく抽出する				○						
18	ログファイルの断片化に注意する			○							
19	人による変更作業ではミスが起きることを前提に、ツール活用などで不具合の作り込みや流出の防止に心がける	○			○			○			
20	信頼性向上施策を採る場合は、故障発生確率と影響の定量評価を行い、対策は確実に実装する		○	○			○			○	
21	高い信頼性対策が求められるシステムでは重大な影響を及ぼす事象の想定と復旧手順を十分に検討する		○								○
22	処理時間がクリティカルなシステムではツールを活用し、変数やその取りうる状態数とそれぞれの状況における動作処理に最大バラツキを意識し余裕を把握し設計する。			○	○		○	○	○		
23	開発を伴わない保守案件でも，システム構成変更が発生する場合は，手順等作業内容の妥当性を確認できるようなプロセスを経る						○	○		○	○
24	物理量（時間、重量など）を扱う場合は単位、桁数を確認する。		○			○		○			
25	顧客が要求していることの目的と背景に遡って、その意図を確認することが、要求仕様のあいまいさ排除に役立つ	○					○				
26	遠隔地等物理的に離れた装置をネットワーク接続して稼働させるシステムでは、故障などの状態検知やメンテナンスも容易ではないため、システム的視点での状態把握を行う。	○	○					○			
27	マルチベンダーシステムでは仕様に外れた想定外事象が発生することを前提とした自己防衛策を採る。	○		○				○			
28	データベース等COTS製品のバージョン、動作仕様の相違等の情報が関係者にタイムリーに参照できるようにする							○	○	○	○

29	複数の事業体にまたがる重要システムでは関係者の立場・ニーズの視点から、想定しうる障害発生リスクを同定し効果的な危機管理体制を構築する	○	○						○	○	○
30	過去のハードウェア、ソフトウェア資産を使用する場合は、その内容や当時の方法について考慮する				○	○		○			
31	ミッションクリティカルシステムではリスク管理やV&Vを確実に実施する						○		○	○	
32	不測事態においても適切に動作するかの検証を十分に行い、条件変更時には潜在的なリスク許容度合いの変化を見逃さない		○		○		○	○		○	
33	不十分な設計となっている回避策は根本的に見直す		○	○							
34	重要なソフトウェアを変更する際は、変更管理を確実に実施する		○						○	○	
35	リスク分析によるハザード識別を行い、非常時には関係者が即応できる体制を構築する		○						○		○

2.1　システム要求定義における対策事例

適用工程		対策／手法		教訓番号
1	システム要求定義	1	システムの利用されかたをあらかじめ想定する／ESPR(SYP1)	6
		2	クライアント端末数などの非機能要求を整理する／ESDR(A-28)	14
		3	例外項目を物理的観点、環境的観点により定義する／ESDR(B-12)	8
		4	定義した例外項目から例外リストを定義する／ESDR(B-12)	8
		5	歩留りのある製品の良品・不良品を検査する装置では、全てが良品あるいは、不良品との検査結果は異常と判断すべきである／ESDR(B-1)	12
		6	要求仕様を検討する際の確認事項に強制終了された時の復旧処理を盛り込む／ESDR(B-1)	15
		7	要求条件の抜け漏れを防止するためには、不変条件を論理式で記述するなど、形式手法の適用を検討する ／ー	17
		8	要求条件の抜け漏れを防止するためには、観点表に知識を蓄積する ／ー	17
		9	識別データのような情報を手入力しなくてはならない場合は入力規則を定める／ー	19
		10	遠隔地機器に対する保守・メンテナンス仕様を検討する／ESPR(SYP1)	26
		11	マルチベンダー環境におけるリスクを想定すること／ESDR(B-21)	27
		12	要求仕様はその意図や背景に遡って確認しあいまいさを排除する／ESPR(SYP1.1)	25
		13	背景事情や慣習なども含め重要事項は文書化する／ESPR(SYP1.1)	25
		14	障害発生時の対処を考慮した全体システム要件を明確にする／ESPR(SYP1.1.4)	29

2.2　システムアーキテクチャ設計における対策事例

適用工程		対策／手法		教訓番号
2	システム アーキテクチャ設計	1	ハードウェア開発部門とソフトウェア開発部門で継続的に連携する／ESDR（D-6）	5, 10
		2	ハードウェアの特性を文書化してソフトウェア設計の入力として与える／ESPR(SYP2.1, SWP1)	5, 10
		3	ハードウェア特性の文書をメンテナンスし続け、劣化を防止する／ESPR(SUP7.1)	5, 10
		4	HWの変更時には影響解析を行う／ESPR(SUP7.2)	6
		5	特殊なHWを使用する時にはあらかじめ特性を把握しておく／ESDR(D-5)	6
		6	システム設計上でダイアグ（診断）機能を実装し、結果を通知する機能を具備する／ESDR(A-20)	6
		7	商品バリエーションがある場合は、その差分を把握し、評価を行う／ESDR(A-13, 14, 15)	7
		8	端末や各機能の起動電流や電圧降下及びその回路部品のバラツキを考慮し、端末起動シーケンスも含め、設計に織り込むこと／－	7
		9	電源ラインに電圧降下対策を追加する場合は、対策部品の配置場所や他の電源ラインへの影響を確認する／ESDR(A-24, B-15)	7
		10	電池で動作する端末においては、電池容量が少ない状態に対する対策が必要であることを認識すること／－	7
		11	消費電力の多い機能を追加する場合には、一時的な電圧降下による影響（リセット、フリーズ等）や電源の種類、電池の場合は残量を考慮すること／－	7
		12	機能項目リストと例外項目リストのマトリクスを生成して例外の機能仕様を定義する／ESDR(B-12)	8
		13	システム設計の中でもメンテナンスモードに対する設計に注意すること／－	12
		14	仕様変更にともなう影響解析をする／ESPR(SUP7.2)	14
		15	故障発生確率算出等の定量評価を行い対策効果を評価すること／ESPR(SUP3.1)	20
		16	故障からの復旧設計では部位や順序によって復旧できないケースがないかを検討する／ESPR(SYP2.1)	21
		17	障害からの復帰シーケンスを十分に検討する／ESPR(SYP2.1)	21
		18	物理量の計算では単位及び単位系の違いに留意する／ESDR(B-5)	24
		19	小数点を含む処理では計算機依存の要素（型と有効桁数など）に配慮する／ESCR(R2.1, 2.3, 2.4)	24
		20	遠隔機器を監視する手段の信頼性を十分検討する／－	26
		21	運用状況情報のリアルタイム表示など障害発生時に必要なシステム機能に留意する ／－	29
		22	無効であるにも拘らず有効であると誤判断される可能性が排除されるような設計を行う ／－	34
		23	重要システムでは網羅的にリスク分析を行い系統的にハザードを識別する／ESPR (SUP3, SAP1)	35
		24	重要な警報が正しく認識されるように警報システムは注意深く設計する ／ESPR (SYP2.1.3)	35
		25	不測の事態の想定を行い、そうした事態に対処可能な設計とすること ／ESPR (SYP2.1.3)	32

2.3　ソフトウェアアーキテクチャ設計における対策事例

適用工程		対策／手法		教訓番号
3	ソフトウェア アーキテクチャ設計	1	適切な規模にモジュール分割し、複雑さを減らす／ESDR(A-3, 9)	1, 2
		2	ハードウェアの制約を超えないように設計、監視する／ESDR(D-4)	1, 2
		3	デバッグやログなどの直接要求に関わらない機能であっても、設計とレビューをする／ESPR（SUP2, SUP8）	16
		4	ハードウェアの制約を考慮する／ESDR(D-6)	5
		5	並列処理を考慮して設計する／ESDR(C-7)	11
		6	共有データはあらかじめ洗い出してチェックする／ESDR(C-7, 16), ESPR(SYP2.1)	11
		7	可変長ファイルの読み書きには注意／ESDR(B-13)	18
		8	ログ設計は、ログファイルの断片化に注意して、予め固定長ファイルのログを設計する等の考慮する／ESDR(B-16)	18
		9	設計意図を文書に残す／ESPR(SYP2.1)	13
		10	大量のデータを通信経由で扱う場合、一連の処理の流れの中にボトルネックを作りこまないように注意する。また、時間帯による負荷変動についても考慮する／ESDR(B-20)	14
		11	データの出と入りを等しくなるように設計する／ESDR（A-17）	14
		12	属人的作業だけに依存せず自動入力やチェックツール等で変換ミスを未然に防ぐ／－	19
		13	異常データを検知し適切に対処できるロジックを実装する／ESPR(SYP2.1.3)	20
		14	通信復旧時には障害発生前の状態を確認し復帰させる／ESDR(B-1)	21
		15	静的解析ツールを利用し確認するような手順を設計プロセスに組み込む／－	22
		16	仕様外事象時のエラー処理を十分に検討する／ESDR(B-1)	27
		17	不十分な設計となっているソフトウェアによる回避策は根本的に見直すべきである ／ESPR (SWP2.2)	33

2.4　ソフトウェアアーキテクチャ設計（変更設計）における対策事例

適用工程		対策／手法		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 ／ー	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援用して検証すること ／ー	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 ／ー	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援用して変更の妥当性を確認すること ／ー	4,32
		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／ー	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
		12	ソフトウェア変更時の記録の残し方には一貫性を持たせる ／ESPR (SUP7）	30
		13	過去のハードウェア、ソフトウェア資産を使用する場合は、その内容や当時の方法について考慮する ／ESDR(C-12）	30

2.5 実装（コーディング）における対策事例

適用工程		対策／手法		教訓番号
5	実装 （コーディング）	1	マルチスレッド処理でスレッドセーフであることの観点を、プログラムチェックシート及びコーディング規準に追加する／ESPR(SWP3.2)	11
		2	データ生成時と参照時では型を揃える／ESCR(R2.3)	24
		3	適用が推奨されるコーディング手法は標準として明記し順守させる ／ESCR(3.2, 3.3)	30

2.6 レビューにおける対策事例

適用工程		対策／手法		教訓番号
6	レビュー	1	ステークホルダを集めて分野を超えた人たちでレビューする／ESPR(SUP8)	5, 10
		2	設計・実装レビューでの「変更点の確認」「変更による影響範囲の確認」を確実に実施する／ESPR(SUP7, SUP8)	13
		3	性能に関わる箇所については確実にレビューする／ESPR(SUP8), ESDR(B-20)	14
		4	ドメイン知識を有するメンバーよるレビューを必須とする／ESPR(SUP8)	20, 22, 23, 25
		5	ミッションクリティカルシステムでは第三者検証も含めたV&Vを確実に実施する ／ESPR(SUP8)	31,32
		6	V&Vではミッション要求にダイレクトに関連づけた検証を行うこと ／ESPR(SUP8)	31

2.7　システムテストにおける対策事例

適用工程		対策／手法		教訓番号
7	システムテスト	1	学習機能を持つモジュールのテストは、網羅度を上げられないことを認識する／ESDR(A-6)	3
		2	瞬停やネットワーク異常対応処理を盛り込んだテスト設計をする／ESDR(A-10)	15
		3	物理的な条件を網羅する／ESTR(1.1.3), ESPR(SYP4.1)	5
		4	実機検証を行う／ESTR(1.1.3), ESPR(SYP4)	5
		5	テスト観点リストなど作成を通してテストの網羅性向上を図る／ESTR(3.1.5)	13
		6	手入力を要するシステムでは範囲外テストを実施する／ESTR(3.1.2), ESDR(B-3)	19
		7	複雑で負荷のかかるシーケンス組み合わせ試験による限界テストを行う／ESDR(B-3),ESPR(SWP5)	22
		8	データベースバージョンの相違による影響をテストで確認する／ESDR(A-8)	23
		9	小数点を含む場合には小数点誤差に配慮したテストを行う／ESTR(3.1.2)	24
		10	遠隔機器の障害を模擬した異常テストを行う／ESTR(1.1.3)	26
		11	境界値テスト項目にヌケモレがないようにする／ESTR(3.1.2)	27
		12	COTSのバージョンによる影響を事前に検証する／ESDR(A-8,B-21)	28
		13	全ての失敗結果を記録しフォローする ／ESPR(SYP4.2, SYP4.3, SUP6)	30
		14	実際の動作を模擬した状態でテストを実施すること。またエラー検出可能な手法やシミュレーションを用いること ／ESPR(SYP4.1)	30
		15	適切なテストや手順を用いかつ文書化を行うこと ／ESPR(SYP4.1)	32

2.8　教育における対策事例

適用工程		対策／手法		教訓番号
8	教育	1	ハードウェアの教育をソフトウェア技術者にすること／－	10
		2	ハードウェア技術者とソフトウェア技術者の文化交流の場を設けること／－	5, 10
		3	マルチスレッド処理の教育をすること／－	11
		4	レビュアのスキルマップを作成するなどしてレビュアの能力向上を図ること／－	13
		5	割込みにおける排他処理などの重要かつ基本概念を理解させる／ESDR(B-7, D-1)	22
		6	COTS技術情報は容易にアクセスできるようにする／ESDR(A-8,B-21)	28
		7	運用関係者に当該システム全体に関する知識を確実に理解させる ／－	29,31,34,35

2.9　プロジェクトマネジメントにおける対策事例

適用工程		対策／手法		教訓番号
9	プロジェクト マネジメント	1	ハードウェア開発部門とソフトウェア開発部門のコミュニケーションを密にすること／－	6, 20
		2	レビュアのスキルマップを作成するなどして適切なレビュアの選定を行うこと／－	13
		3	ひとりの担当者に過度に依存したタスクの割り当てを行わない／ESMR(4章)	13
		4	対策時は2次リスク（副作用）の発生を検討した上で実施判断する／ESPR(SUP3)	20
		5	保守対応でも作業内容の妥当性を確認できるようなプロセスとする／－	23
		6	ブラックボックスであることを前提とした開発プロセスを考慮する／ESDR(B-21)	28
		7	システムの安定的な運用に関わる部門やチームのコミュニケーション向上を図る ／ESMR(3章)	29,31
		8	重要な役割と情報が識別され、開発から実行段階に確実に引き継がれるような計画とプロジェクト運営とすること ／ESPR(SUP1),ESMR(3章)	31
		9	条件変更によるリスク許容度合いの変化を見逃す事のないように、定められた手続きを確実に実施し変更後のリスクレベルを確認する ／ESPR（SUP3）	32
		10	重要なソフトウェアを変更する際は、定められた手続きを確実に実施し変更後のリスクレベルを確認する ／ESPR（SUP7）	34

2.10　運用における対策事例

適用工程		対策／手法		教訓番号
10	運用	1	お客様がアプリケーションソフトウェアを追加する際の制約条件を伝えること／－	6
		2	メンテナンスモードを有するシステムでは、その取扱いについて、手順書で明確にして、ダブルチェックも考慮すること ／－	12
		3	障害復旧手順は文書化して保守関係者に周知する／－	21,29
		4	保守対応員へ事前に情報提供を確実に行うこと／ESPR(SYP1)	23
		5	不測事態への対処を計画する／ESMR(7章),ESPR(SUP3)	28
		6	責任の所在を明確にし非常事態に備える体制がすべての危険施設に適用されること ／－	35
		7	悪い知らせが経営層にタイムリーに伝達されるような監査体制とする ／－	35

3.観点マップ

発生した障害から得られる知見を他製品や産業領域に適用、展開する等、障害を未然防止化するための取組みはその重要性が認識されてはいても、開発形態やプロセス、技術の違いにより容易でないという現実もある。一方、各事例の事象を引き起こした原因には共通する要素も見受けられるため、それらを未然防止の教訓として自社・自部門製品に適用する際の抽象化に活用することを想定し、その利活用のトリガーとなるよう、35 教訓事例の直接原因と真因を観点マップとして抽出・整理した。また開発現場での活用方法についても参考例として記述した。

3.1　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。マップ中の番号は事例番号を表している。

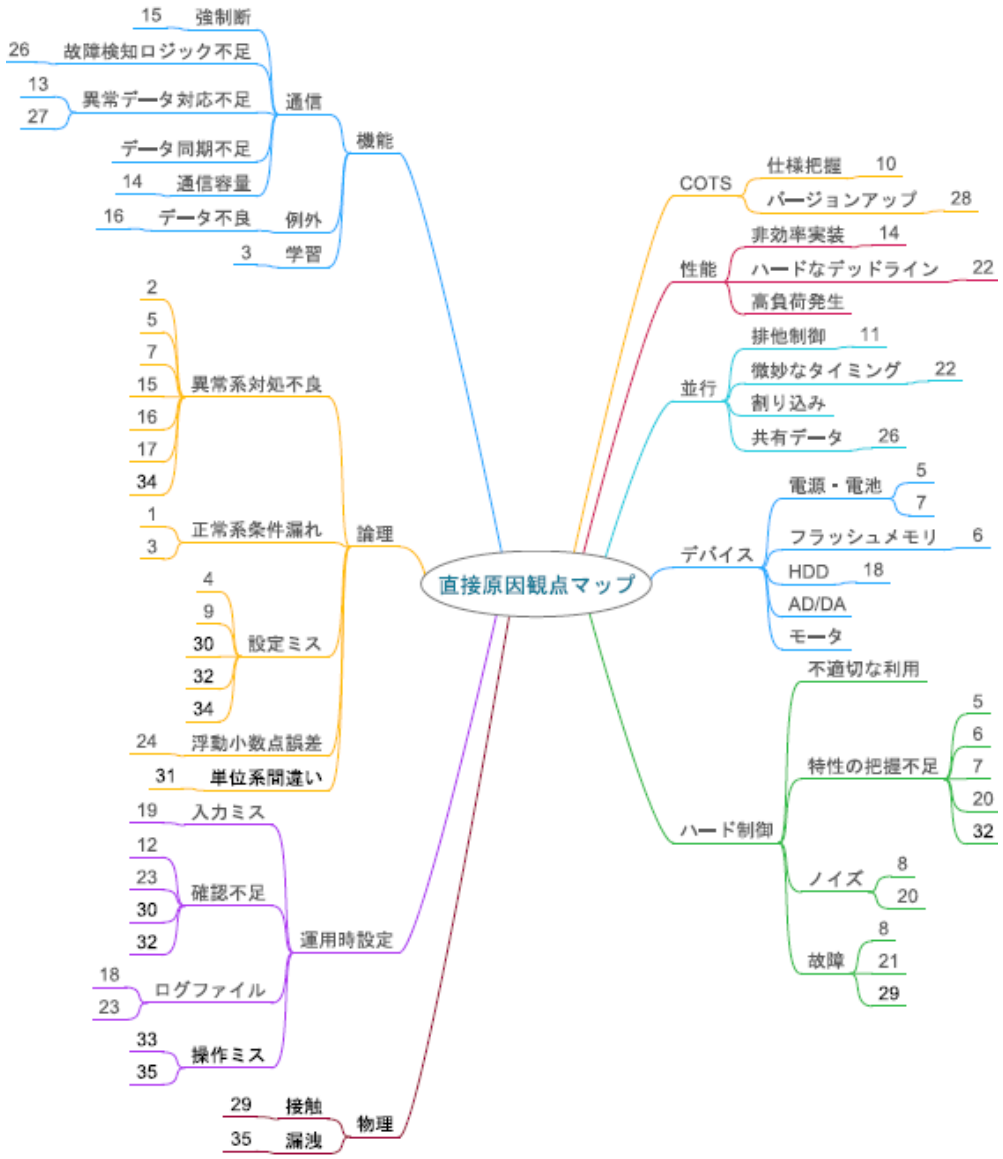


図 3.1　直接原因観点マップ

3.2　未然防止観点マップ

同じく 35 事例の障害を引き起こすに至った真因から未然防止の観点を抽出し整理したものである。マップ中の番号は事例番号を表している。

条件数や構造の複雑さやバリエーションの多さといった開発性質、要求仕様の条件の見落とし等が多く、近年の組込みシステムが置かれた状況を推測できる。

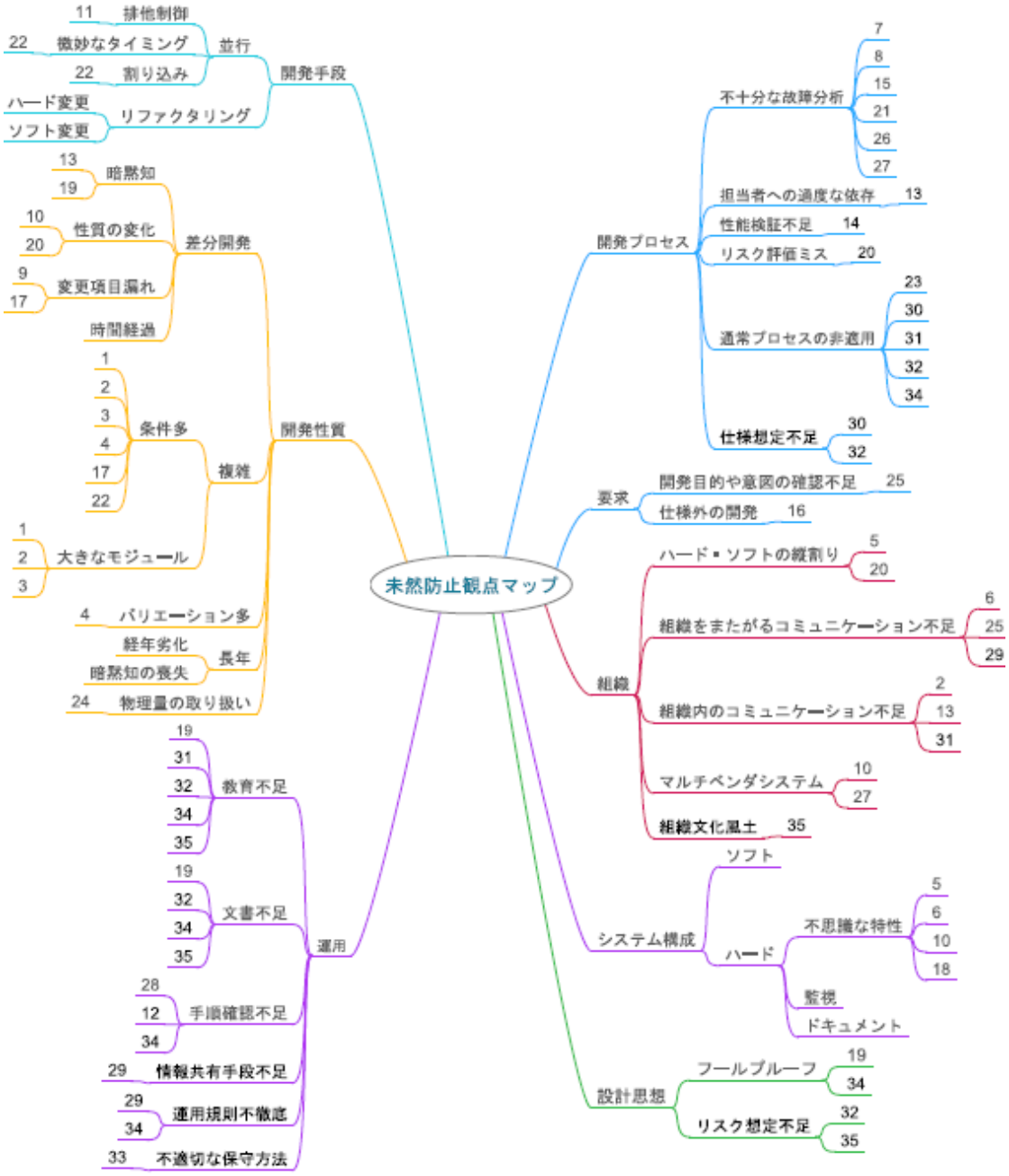


図 3.2　未然防止観点マップ

3.3　活用方法

本節では観点マップの具体的な活用方法のイメージを例示した。なお、これらは開発実務の状況に即した組合せや応用を行うことが望ましい。

3.3.1　教育・研修への活用

【教訓の選択】

直接原因観点マップまたは未然防止観点マップの中から活用したい観点要素を選択し、該当する教訓番号から教訓情報を選択する。

【抽象化】

選択した教訓事例をよく読み、その本質的内容を理解し必要に応じた抽象化を行う。
この際、製品は違っても障害を作り込んだり、流出させたりしてしまった要因に対して、自社製品、自部門の状況に置換可能な抽象化を行うために、観点マップのレイヤを適切に遡る等の工夫を行う。

【資料作成】

類似の社内事例等も参考に、研修等で使用するケーススタディ資料等の作成を行い、教育を実施する。

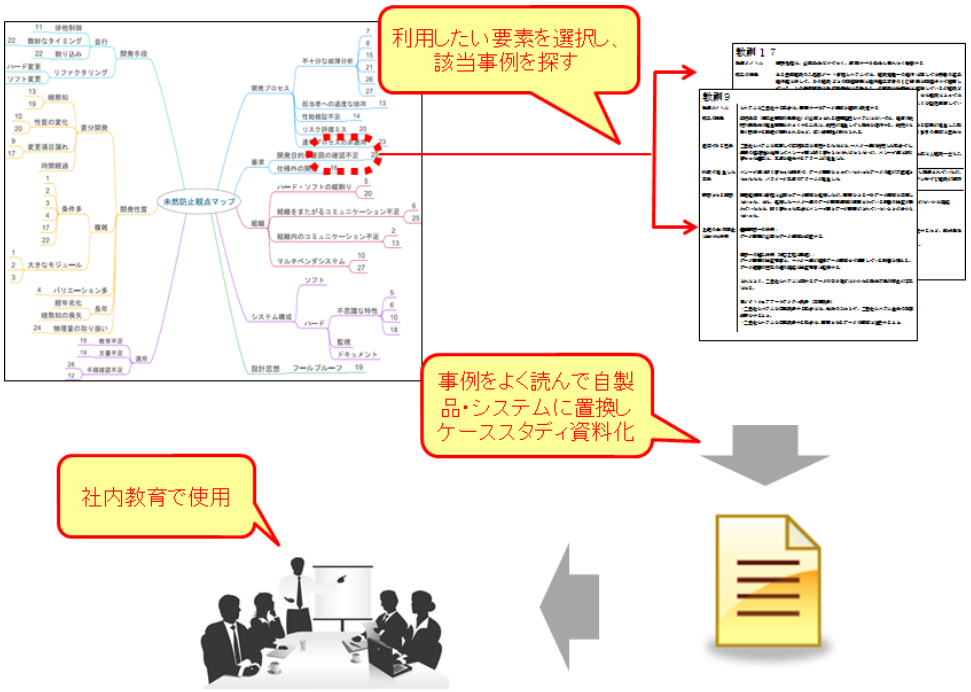


図 3.3　教育・研修への活用

3.3.2　開発プロセスへの活用

【観点マップのカスタマイズ】

自製品や技術の実態に即した要素項目を追加したり、表現法を修正したりする等の定期的なメンテナンスを行い実務応用に適した状態を保つように心がける。

【チェックシートへの反映】

開発工程でのデザインレビューで使用するチェックシートや、レビューアが留意すべき項目等に活用するため、観点マップ中の要素とそれに該当する教訓事例情報を自社製品、技術に置換して反映する。

【開発プロセスに適用】

開発プロセスのデザインレビュー等のマイルストーンで観点マップと教訓情報も利用する。また、このマップ中の要素を抽象化し設計タスク定義を行い、開発プロセスに組み込むといった応用も可能である。

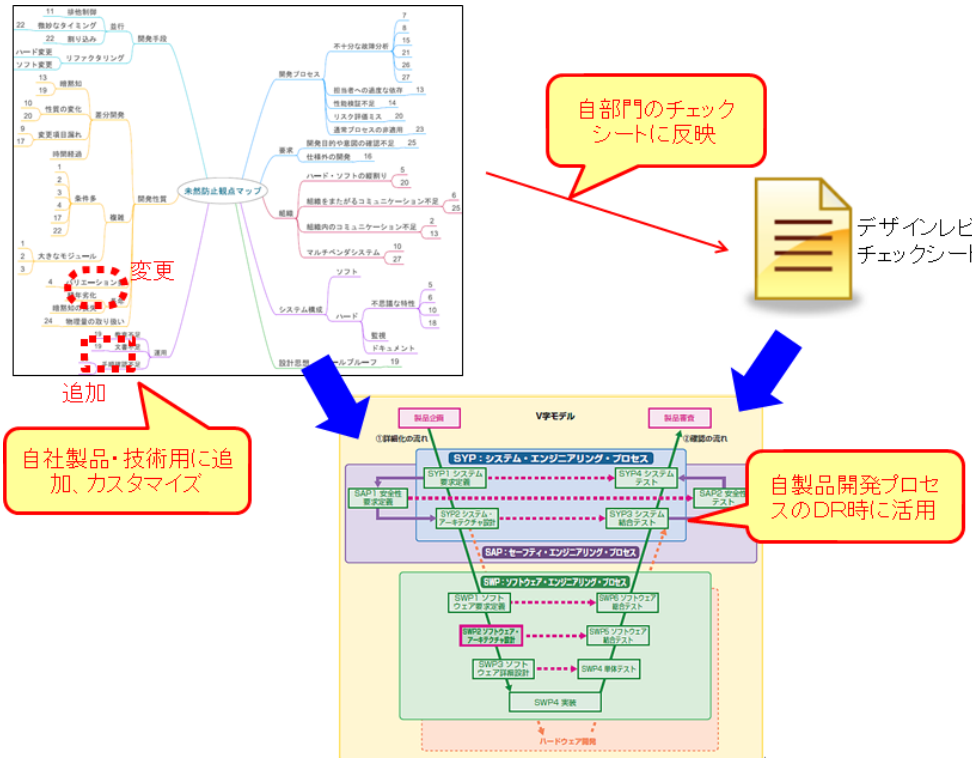


図 3.4　開発プロセスへの活用

ＰＡＲＴ　Ⅲ

障害分析手法・事例集（組込みシステム編）

PART Ⅲ 目次

1. はじめに

1.1 本事例集で使用する用語について

障害、故障、欠陥などの用語は、様々な意味で使用される。本書では、障害という用語は JIS X 0014 に定義される障害(fault)の意味で用い、故障(failure)や欠陥(defect)は IEEE 1044 の定義や IEEE 982.1 の記載内容に沿って用いる。

【障害】要求された機能を遂行する機能単位の能力の、縮退又は喪失を 引き起こす、異常な状態。

【故障】要求された機能を遂行する製品の能力が尽きる状態、または事前に仕様化された制限内での機能を遂行する能力が無い状態とする。

【欠陥】設計者の認識の有無にかかわらず、すべての成果物において要求定義の誤り、仕様設計の誤り、プログラミングの誤り、システム構築の誤り等により「期待される結果」と乖離があるために、何かしらの対策・対応が必要と考えられる事象またはその原因。

【要因】原因の候補。

1.2 分析手法概覧

本事例集で扱う分析手法と分析手順の関係を表 1 に示す。本事例集は 2 つのシーンで参照されることを想定している。

1 つ目は、システムの稼働中に障害が発生した場合である。システムの運用者や障害の発見者が重要な情報源となるため連絡を取れるようにする必要がある。

2 つ目は、システムの開発中に障害が発生した場合である。稼働中の障害と異なり、開発中の障害への対応が緊急性を要することはまれである。しかしながら、障害の分析に必要となる情報を収集する活動が開発プロセスに組み込まれていると後々の分析作業を円滑に進めることができる。そのため、障害を分析する時にどのような情報が必要となるか整理しておくとよい。

表 1 の下部は障害分析の手順を示している。各段階で、誰が（作業の主体者）何を行うか（分析作業)、また、各タスクではどのような手法が利用されているのかをまとめている。障害を分析する際には、分析がどの段階まで進んでいるのかこの表を参照して把握することができる。

表 1 分析手順と分析手法の概要

参照シーン

①運用中の障害
・現場の状況を保存・記録する（可能な限り）
・関係者への連絡手段を確保する
・関連する文書の用意

②開発中の障害
・障害が発生した状況を保存・記録する
・事実関係の資料を収集（実行ログなど）
・関連する文書の用意

入力

分析作業

タスク	入力	作業	主体者	出力	手法
情報収集	①，②の情報，人	情報収集と整理	プロジェクトリーダ 開発担当者 営業担当者	収集した情報を整理した文書	障害管理票の調査 関係者へのヒアリング 再現実験
システム構造の把握	収集した情報を整理した文書 設計書・仕様書	システム構造の整理	プロジェクトリーダ 開発担当者 有識者（HW設計者）	システム構造図	ブロック図 UML SysML ネットワーク図
問題症状の把握	収集した情報を整理した文書 システム構造図	問題症状の把握	プロジェクトリーダ 開発担当者 有識者（HW設計者）	問題症状を整理した図表	表 VTA
原因分析	収集した情報を整理した文書 システム構造図 問題症状を整理した図表 担当者の経験・過去事例 FTA/FMEA(設計時作成)	原因箇所の特定	プロジェクトリーダ 開発担当者 有識者（HW設計者）	原因箇所の候補一覧表	問題行動分析 例外分析 FTA/FMEA
	原因箇所の候補一覧表 設計書・仕様書 観点表（設計時に作成）	直接原因の特定 影響度・範囲の分析 暫定的対処案の立案	プロジェクトリーダ 開発担当者 製品の品質保証部	直接的原因の一覧表 ・人為的ミス ・設計誤り	レビュー コードインスペクション 再試験 シミュレーション モデル検査
	直接的原因 開発記録（実績） 開発プロセス（定義） ヒアリング（開発担当者）	根本的な原因の推定	プロジェクトリーダ 開発担当者 SQA(品質管理部門)	根本的原因 ・プロセスの欠陥 ・未実施プロセス 視点で整理 ex. 制御可能な項目化	PNA 発生源・検出漏れ分析 なぜなぜ分析 KJ法（課題）
対策の立案	根本的原因 原因に伴う制約事項	対策の検討	プロジェクトリーダ 開発担当者 管理職	短期対策 長期対策	影響レベル評価尺度・評価手法

本事例集では

2 章： 分析の手順と流れ

3 章： 分析に利用できる手法

を紹介する。なお、3 章でとりあげる分析手法としては、表 1 の手法欄に掲載した手法の中から、今回調査した企業で普段利用されているものを中心に選んで掲載してある。また、3 章の分析手法の適用例は、4 章に掲載する実フィールドで発生した障害事例を題材にまとめてある。

2. 障害発生から分析結果までの流れ

2.1 障害発生から対策の検討まで

図 1 は障害が発生した際に行う活動を示したものである。本節では各活動の詳細について述べる。

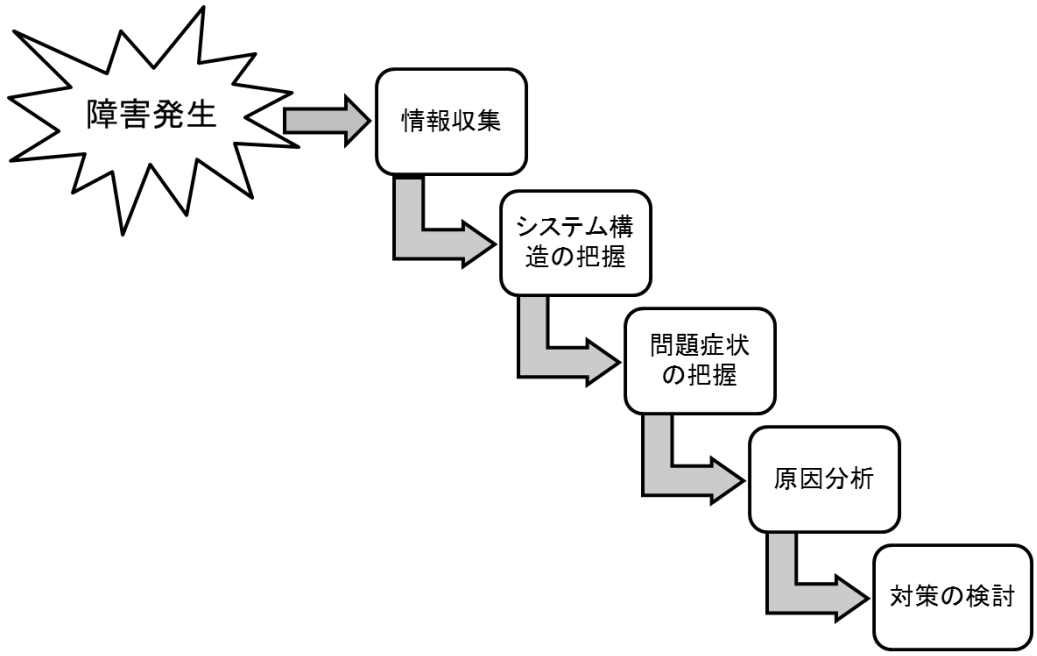


図 1　障害分析の流れ

2.2　分析の各タスク詳細

(1) 情報収集

障害の分析に際してまず行うのが情報収集である。ただし、稼働中のシステムで発生した障害については事態の収拾を優先する。障害の分析に必要な情報の収集は事態の収拾後速やかに行う。収集した情報は整理して文書にまとめる。

障害の分析に必要となる情報の例として以下のものがあげられる。

- 障害の発生状況（関係者に対するヒアリングを通じて）
- システムの稼働ログ情報
- 障害の再現手順
- システムの構成
- 外部環境

システムの種類によって収集が必要な情報は異なる。また、収集できる情報が限られる場合もある（大規模な事故等)。このような場合でも必要と思われる情報をできるだけ集めることが以降の分析を円滑に進めるために重要である。

(2) 全体を把握する（システム構造）

収集した情報は大きく分けて 2 種類に分類できる。すなわち、障害発生の経緯に関する情報と、システムの動作・構造に関する情報である。これらの情報を整理して対応付けることで障害に至った問題症状の把握が可能になる。

まずはシステムの動作・構造に関する情報を整理して、システムの全体像を見渡せるような簡潔なシステム構造図を用意する。システムが複数のサブシステムから構成されている場合は、サブシステム内の構造は別の図に分ける等簡潔さが損なわれないように気を付ける。全体を見渡すことができる図を用意

することは、サブシステム間の相互作用の関係が明確になる等、対象システムの構造把握に役立つ。

システムの設計時に作成した UML 等の図がそのまま利用可能であることが多いが、システム全体の概要をとらえるために、簡易なブロック図等を用意してもよい。

(3) 問題症状の把握（事象経過）

次に、時系列と相互作用を意識しながら事象を整理して問題症状を把握する。手順としては、まず、ヒアリング等で得た事象に関する情報を時系列に沿って整理する。このとき、観点を定めて事象を整理すると分析しやすくなる。例えば、システム、システムの運用に従事している人、システムの利用者等の観点が考えられる。

システムが複数のサブシステムから構成されている場合は、サブシステムごとに情報を整理する。また、システム構造の把握の際に作成したシステムの構造図を利用して、収集した情報に矛盾する点や曖昧な点がないことを確認する。

最後に、障害に関わった人やシステムの構成要素の間の相互作用が分りやすくなるように事象を整理して一覧性の高い形式で図表にまとめる。

(4) 原因分析

原因分析は 3 つのタスクで構成されている。すなわち、①原因箇所の推定、②直接原因の特定、③根本原因の特定、である。各タスクについて以下で述べる。

① 原因箇所の推定

まず、問題症状を引き起こしたと考えられるシステム上の原因箇所を推定する。複数の観点から原因を推定することで漏れや抜けを防ぎやすくなる。例えば、以下のような観点が考えられる。

- ソフトウェアの観点
- ハードウェアの観点
- 人の観点
- 環境・想定条件の観点

事象について整理した図表やシステム構造図に加えて、原因分析の担当者の経験等も貴重な情報源である。社内で過去の障害事例のデータベースが整備されている場合は参照する。また、システムの設計時に作成した FTA や FMEA 等を利用することで、考慮すべき故障モードの漏れや抜けを防ぎやすくなる。

情報を得ることが難しい場合は一定の仮定を置いて原因箇所を推定する。このような仮定については、事実と区別できるように記録する。

② 直接原因の特定

原因箇所の推定結果に基づいて直接原因を特定する。この段階では、再試験によって障害の再現条件を調査する等の作業が必要となる。ソフトウェア部分に問題があると推定される場合には、コードレビューやインスペクション等を実施する。

原因箇所の特定作業が不十分であるために直接原因が見つからない場合もある。その場合は前のタスクに戻って再分析を行う。

直接原因を特定した後、その影響度や影響範囲を分析する。分析結果に応じて緊急の対処方法を考える必要がある。

③ 根本原因の特定

次に、直接原因を引き起こしたと考えられる根本原因を分析する。多くの場合は設計誤りや人為的ミスが直接原因となるが、そのような誤りやミスが行われた要因や見逃された要因を複数の視点に立って分析する。例えば、開発プロセスの観点から直接原因を眺めることで、障害を引き起こす欠陥を作りこんだ

工程やその工程の不備を特定する。この場合、開発担当者へのヒアリング等が貴重な情報源となる。

(5) 対策の検討

特定された根本原因を取り除く対策を検討する。ただし、全ての原因が取り除けるわけではない。そのような場合は影響を軽減する対策を検討する。開発プロセスや体制に関する現状や制約を踏まえたうえで 2 種類の視点で再発防止策を考える。

- 短期的対策
- 長期的対策

開発中に発生した障害であれば、開発中のシステムを点検して同種の障害が発生していないことを確認する。また、その後の開発で同種の障害が再発しないような対策を立案する。

開発終了後には、他のプロジェクトにも適用できる長期的な再発防止策を考える。長期的な対策としては

- ガイドラインの整備（技術面について）
- 規定の改定（プロセスの定義等）
- 教育体制の整備
- 知見に関するデータベースの整備

等が挙げられる。再発防止策を考える際に重要なのは、技術的課題であるのか管理的な課題であるのか等、複数の観点から検討することである。また、根本的な原因が影響を及ぼす範囲や度合いを評価する手法を整備することで、妥当な再発防止策を選びやすくなる。

3. 分析手法と分析事例

3.1 ブロック図

(a) 手法の利用シーン

システム構造を把握する際に利用する。

(b) 手法の概要

ブロック図は物事の構造等を図示する方法のひとつである。詳細を捨象することで全体的な構造を把握することが容易になる。複雑なシステムの構造を把握するためには、システムの設計図をそのまま用いるよりも適している場合もある。

(c) 記法ならびに分析法

ブロック図については、様々な書き方のルールが存在するが、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（個別のセンサー、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し、各ブロック間の関係をブロック間のリンクとして線でつないで表記し、システム全体の構成を表現する。

(d) 分析の例 1

アリアン 5（4.3 節）の事例について作成したブロック図を図 2 に示す。この図ではシステムの構成要素間の関係だけでなく、障害に関連した事象が発生した箇所が分るように説明されている。

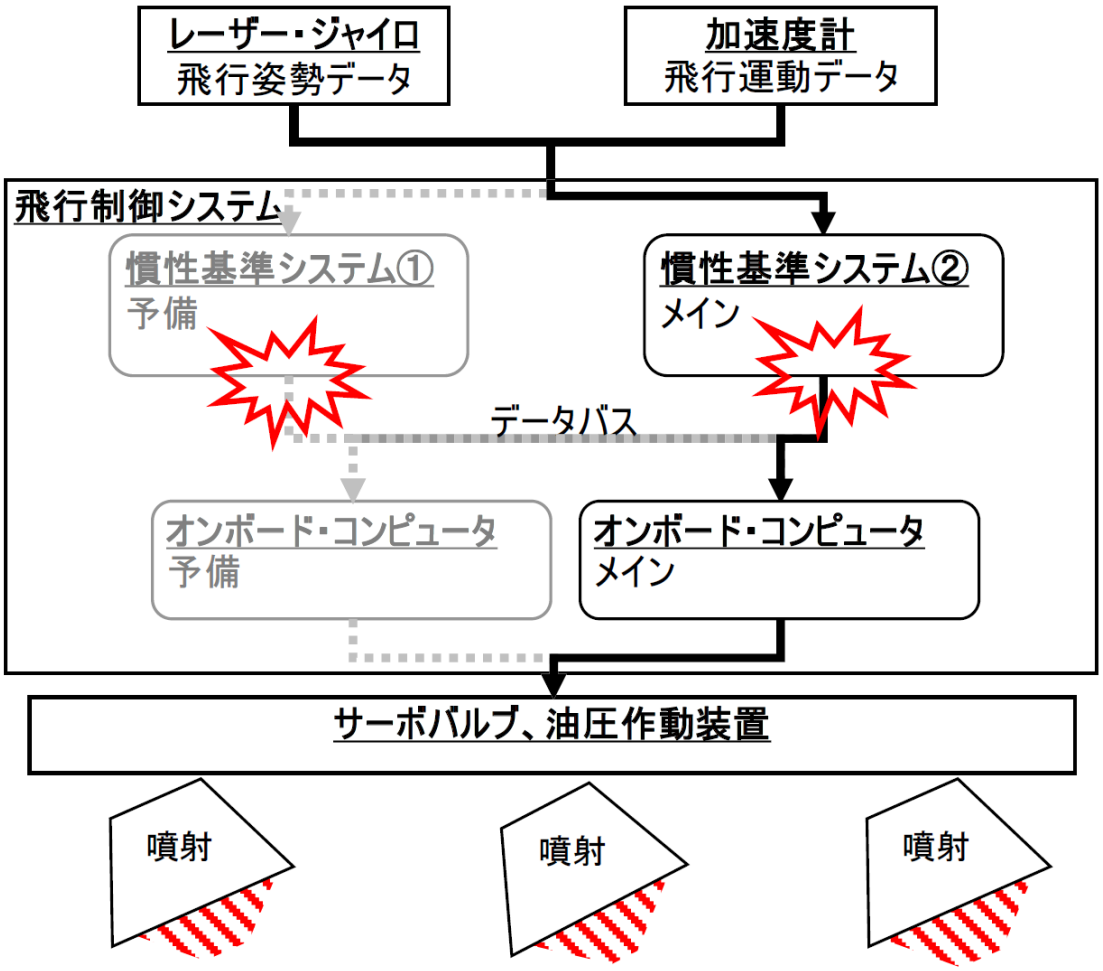


図 2 ブロック図の例（アリアン 5）

(e)　分析の例 2

駒場ダム（4.2 節）の事例について作成したブロック図を図 3 に示す。この図でもシステムの構成要素間の静的な関係だけでなく、障害に行った経緯（事象）についても盛り込まれている。

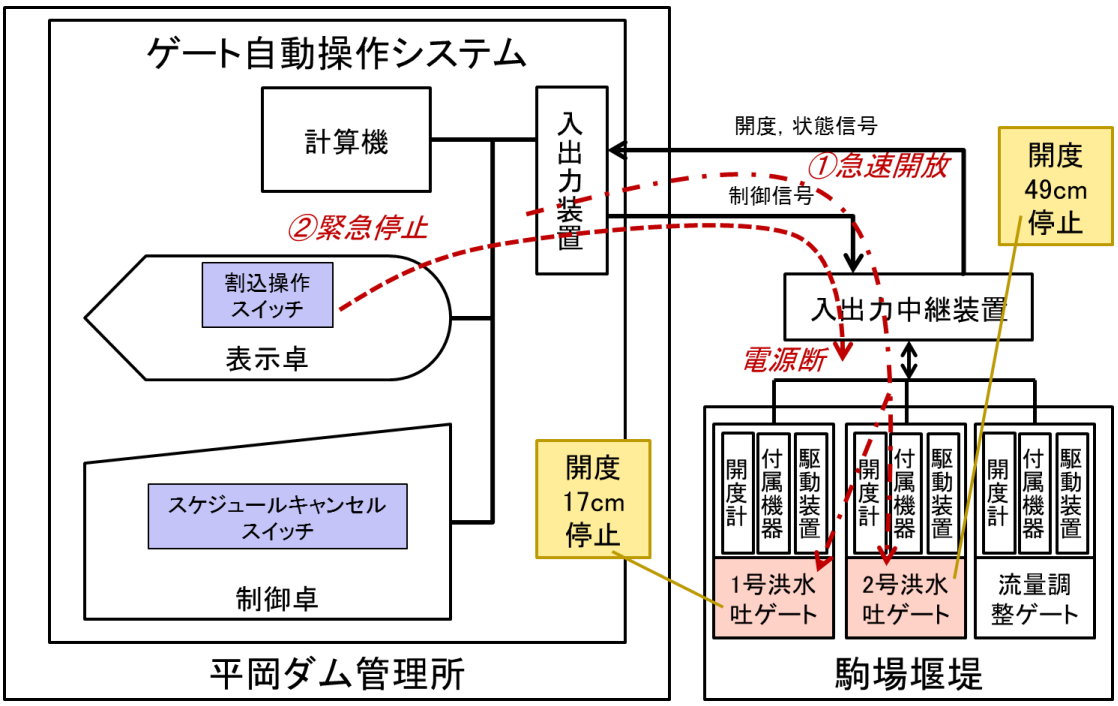


図 3 ブロック図の例（駒場ダム）

3.2 事故経過表

(a) 手法の利用シーン

問題症状の把握に利用する。

(b) 手法の概要

障害の経緯を把握するとき、全ての事象が一覧できるよう表に配置する方法がよく用いられる。表の作成では、時系列に沿って事象を配置することで漏れや抜けを防ぎやすくできる。

この方法は記法の習得が不要で、表計算ソフト等を用いて簡単に作成できる利点がある。一方で、テキスト主体で記述するため、システム構造図との用語の対応等に注意する必要がある。障害に関連する要素が多い場合には、システムの構成要素や関係者ごとに事象を書き出すことでシステム構造図との対応を確認しやすくなる。

(c) 記法ならびに分析法

事故経過表は時刻欄、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していく。分類欄については、問題発生の端緒となった不良事象、その結果としてシステムに発生した機能不全、外部から観測可能となる症状等を代表的な分類として表記する。

(d) 分析の例 1

駒場ダム (4.2 節) の事例から作成した表を表 2 に示す。この例では、システムの構成要素は区別せず、時系列に沿って配置している。さらに、事象の分類を示す項目が追加されている。ここでは、「失敗まんだら」[1]の「失敗結果の分類」に挙げられている分類を利用している。このような分類は障害の原因を推定する段階で役立つ。

表２　事故経過表による事象の整理（駒場ダム）

	問題の症状/経過	
	分類	内容
14:05		目標ダム水位9.9mに設定
14:10	不良現象	水位変更画面上で設定水位を確認後、「変更取消」操作を行い計画起動 → 水位設定値が初期値になってしまっていた
18:05	不良現象	吐水ゲート作動
18:06		2号ゲートが開度49cmで停止
18:06		1号ゲート停止をこころみるが停止せず
18:07		「スケジュールキャンセル」スイッチにより停止すべきところ、「緊急停止」により作動停止
		「緊急停止」したことにより、現地の電源が止められ遠隔操作不可となる

(e)　分析の例２

湘南モノレール（4.1 節）の事例から作成した二次元表を表 3 に示す。分析の例 1 と同様に事象の分類を示す項目が追加されている。また、「失敗まんだら」[1]の「失敗結果の分類」に挙げられている分類を利用している。

表 3 事故経過表による事象の整理（湘南モノレール）

問題の症状	
分類	内容
不良現象	VVVFインバータ内のゲート電源装置の高周波ノイズが、同装置の電源マイナス極側である低圧車体接地線に重畳した。
不良現象	未使用のモニタ伝送回路がノイズの影響により、異常動作した
機能不全	加減速シーケンス処理が実行されなくなり、力行継続状態となった
機能不全	ウォッチドッグタイマによる保護動作が働かなかった
破損	列車の車両及び分岐器等の施設に物損
破損	すべてのブレーキディスクに亀裂が発生した。
起こり得る被害	ブレーキディスクとして必要な強度を有していない可能性がある。

3.3 VTA（Variation Tree Analysis）

(a) 手法の利用シーン

問題症状の把握に利用する。

(b) 手法の概要

VTA（Variation Tree Analysis）は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おり、品質学会の論文等でも良く見受けられる手法である。

(c) 記法ならびに分析法

VTA では表と同様に縦軸を時系列に取って事象を並べる。さらに、横軸に各事象と関連する構成要素を配置し、各構成要素に関連ある関係を記述する。この際、配置された事象が構成要素にとって正常動作であったかといった情報や、事象間の因果関係についての情報を定められた記法で記述する。図 4 に VTA の凡例を示す。

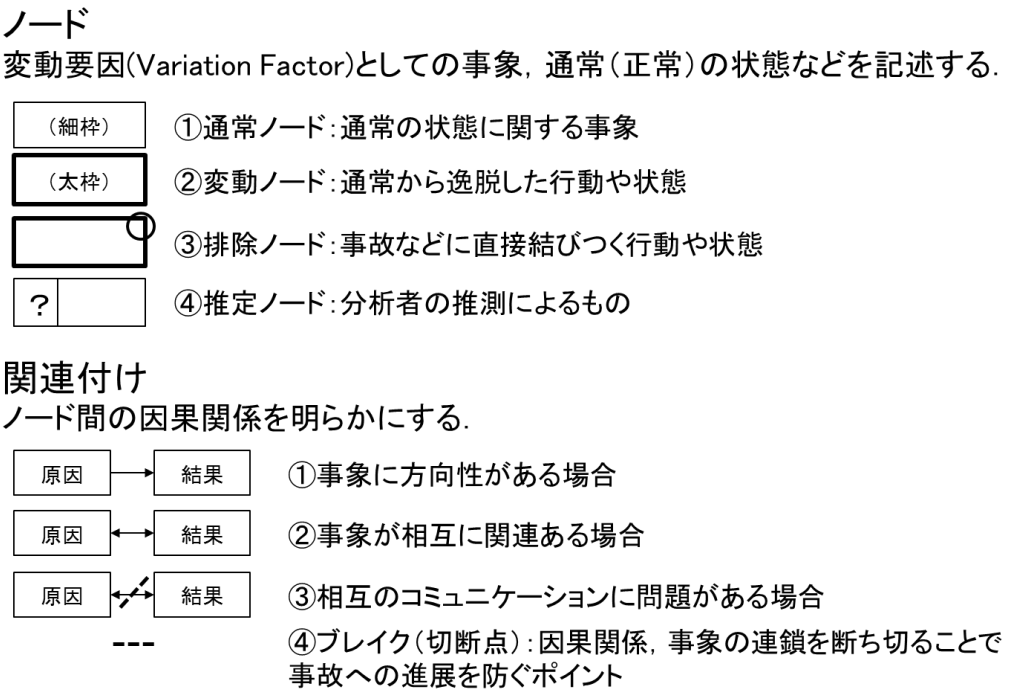


図 4 VTA の凡例

(d) 分析の例

駒場ダム（4.2 節）の事例を VTA で整理した例を図 5 に示す。事象が時系列に沿って関連する構成要素ごとに並べられている。また、事象間の関連性の種類を区別できる記法が整えられており、適切に記述できれば障害の直接的原因を推測する際にも役立つ。今回の事例では、目標水位の確認のための操作が障害の起点であることが図から読み取れる。

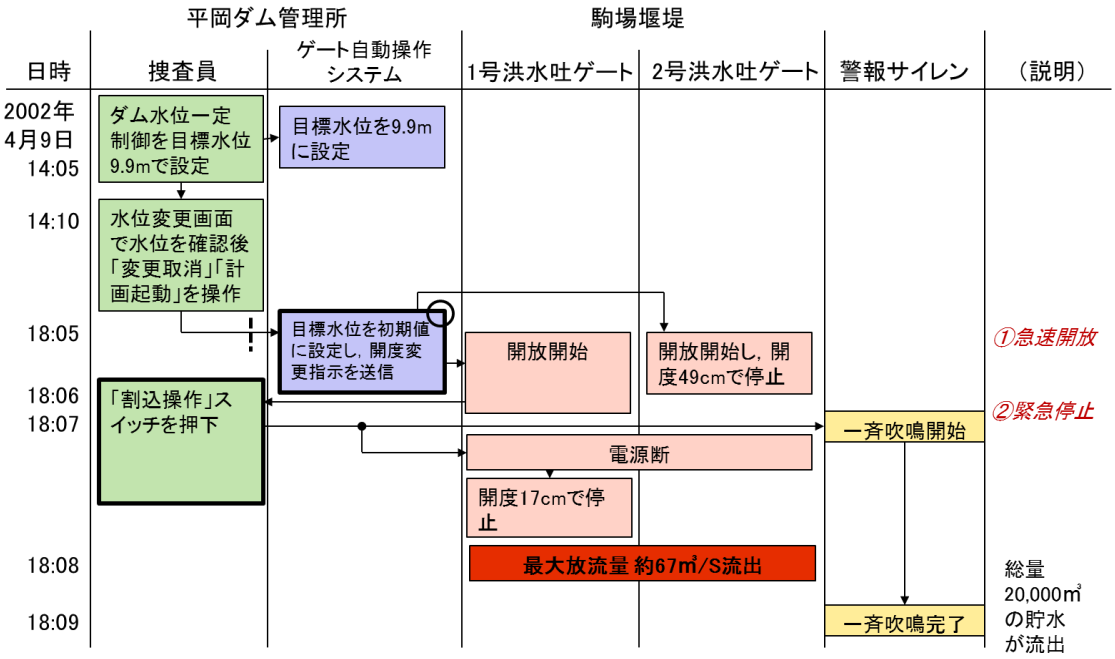


図５　VTA による事象の整理例（駒場ダム）

3.4 問題行動分析

(a) 手法の利用シーン

原因箇所の特定に利用する。

(b) 手法の概要

問題行動分析では、整理された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。分析の結果は根本原因を推定する際に利用するなぜなぜ分析の起点となる。

＊本手法は事例集検討のための部会委員から提供されたオリジナルのものである。

(c) 記法ならびに分析法

事故経過表をもとに、各事象を引き起こした可能性として考えられるシステム操作者や開発者の行動を検討して問題行動/内容欄に記載する。また記載した問題行動/内容について、運用時の操作と開発時の作業に分類する。

(d) 分析の例 1

問題行動分析を駒場ダム（4.2 節）の事例に適用した例を表 4 に示す。左側の問題症状は表 2 で示したのと同じである。問題症状と同様に問題行動にも分類を示す項目が追加されている。この項目は 「失敗まんだら」[1]の「失敗行動の分類」を参考にしているが、必ずしもそれにとらわれない分類項目を設けている。

症状がシステムの意図しない動作である場合、開発上の問題行動（行動誤り）が記述される。

表 4　問題行動分析の例（駒場ダム）

	問題の症状/経過		問題行動	
	分類	内容	分類	内容
14:05		目標ダム水位9.9mに設定		
14:10	不良現象	水位変更画面上で設定水位を確認後、「変更取消」操作を行い計画起動 → 水位設定値が初期値になってしまっていた	潜在危険	確認のために設定画面を開いた？
			ソフト制作	設定状況を確認できない画面になっている
			運転・使用	最終的な設定値の確認をしていない
18:05	不良現象	吐水ゲート作動	ソフト制作	設定状況を確認できない画面になっている
18:06		2号ゲートが開度49cmで停止	ソフト制作	ゲート停止する仕様とソフト仕様の不一致
18:06		1号ゲート停止をこころみるが停止せず	非定常操作	停止のために「割込操作」スイッチを押した
18:07		「スケジュールキャンセル」スイッチにより停止すべきところ、「緊急停止」により作動停止	非定常操作	電源断になることをしらず「緊急停止」を押してしまった
		「緊急停止」したことにより、現地の電源が止められ遠隔操作不可となる	非定常操作	管理所なのに遠隔操作不可となる

(e)　分析の例 2

問題行動分析を湘南モノレールの事例に適用した例を表 5 に示す。分析の例 1 と同様に、「失敗まんだら」[1]の「失敗行動の分類」を参考にして分類項目を設けている。また、開発上の問題行動も一緒に記述されている。

表 5 問題行動分析の例（湘南モノレール）

問題の症状		問題行動	
分類	内容	分類	内容
不良現象	VVVFインバータ内のゲート電源装置の高周波ノイズが、同装置の電源マイナス極側である低圧車体接地線に重畳した。	ハード制作	マイナス極と車体間の接地線の断面積が小さく、抵抗値が大きい
		ハード制作	低圧車体接地線の配線の引き回しが長い
不良現象	未使用のモニタ伝送回路がノイズの影響により、異常動作した	ハード制作	VVVFインバータ内の分圧抵抗盤で終端処理がされていなかった。
機能不全	加減速シーケンス処理が実行されなくなり、力行継続状態となった	ソフト制作	搭載されていない運転台モニタの伝送回路からの割り込みを有効な割り込みとして処理するプログラムになっていたので、不正割り込みが発生した。
		ソフト制作	不正割り込みにより、すべての割り込みが禁止されたため、加減速シーケンス処理起動カウンタの値が更新されなく
機能不全	ウォッチドッグタイマによる保護動作が働かなかった	ソフト制作	加減速シーケンス処理が実行されない場合でもWDTがかからないような処理フローとなっている。
破損	列車の車両及び分岐器等の施設に物損	非定常行為	マスコンを操作しても加速し続けるという異常状態を知りながら、運行を継続した。
		ハード製作	非常ブレーキを使用したが、駆動力の方が勝っていたため、止まらなかった。
破損	すべてのブレーキディスクに亀裂が発生した。	非定常行為 無為	1次車のものと比較すると同じブレーキをかけた場合に発生する応力が大きく、ブレーキディスク自体の引張り強度が小さいので、熱疲労によるき裂が発生しやすい材料であったものと推定される。
起こり得る被害	ブレーキディスクとして必要な強度を有していない可能性がある。		

3.5　PNA（プロセスネットワーク分析法）

(a)　手法の利用シーン

原因分析に利用する。特に根本原因を推定する際に適している。

(b)　手法の概要

PNA[3]は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根本原因を探る手法である。PNA では開発プロセス毎に想定される欠陥混入の要因（(確認ポイント)）があらかじめ整理されており（(図 6)）、それらの要因と事象との関連性を分析することで問題がある工程及びその原因を特定する。開発プロセスが詳細に決まっており遵守されている状況で特に有効な手法である。

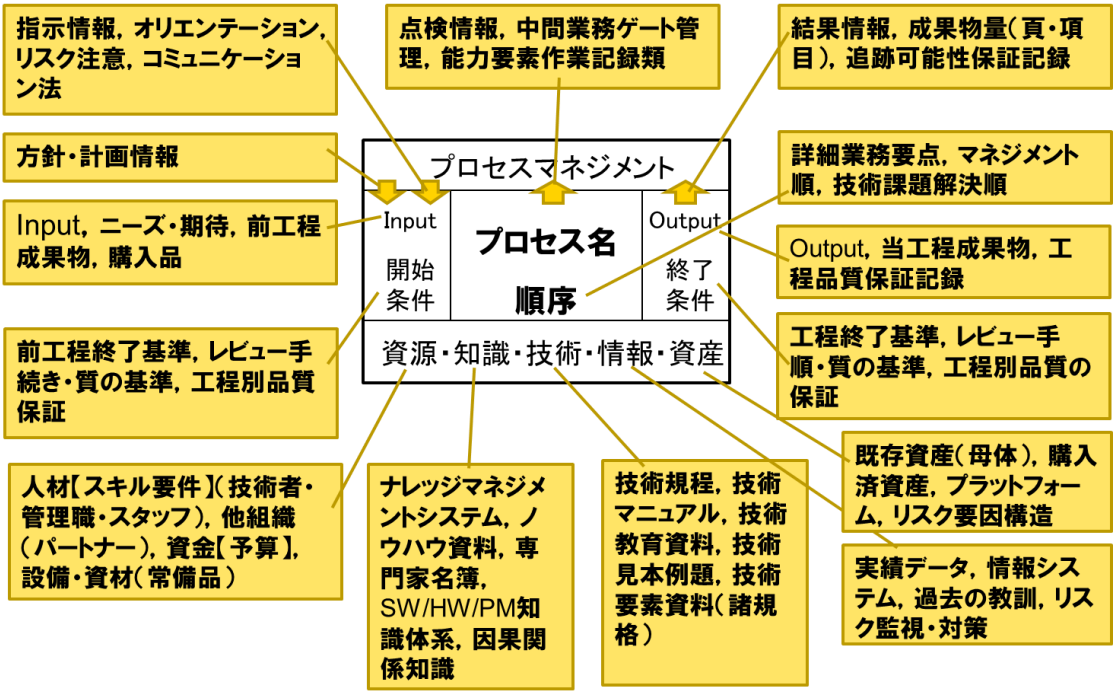


図 6 プロセス要素における確認ポイント

(c)　記法ならびに分析法

PNA を行う際には、まず、図 6 に示したプロセステンプレートに従い、当該システム開発におけるプロセスを記述する。その上で、各プロセスの中で障害の原因として疑われる作業（アクティビティ）や情報を洗い出し、プロセス記述の上に注記していく。

(d)　分析の例

湘南モノレール（4.1 節）についての分析例を図 7 に示す。本事例の開発プロセスモデルは不明であるが、ここでは V 字モデルが採用されていると仮定して要因候補を洗い出している。

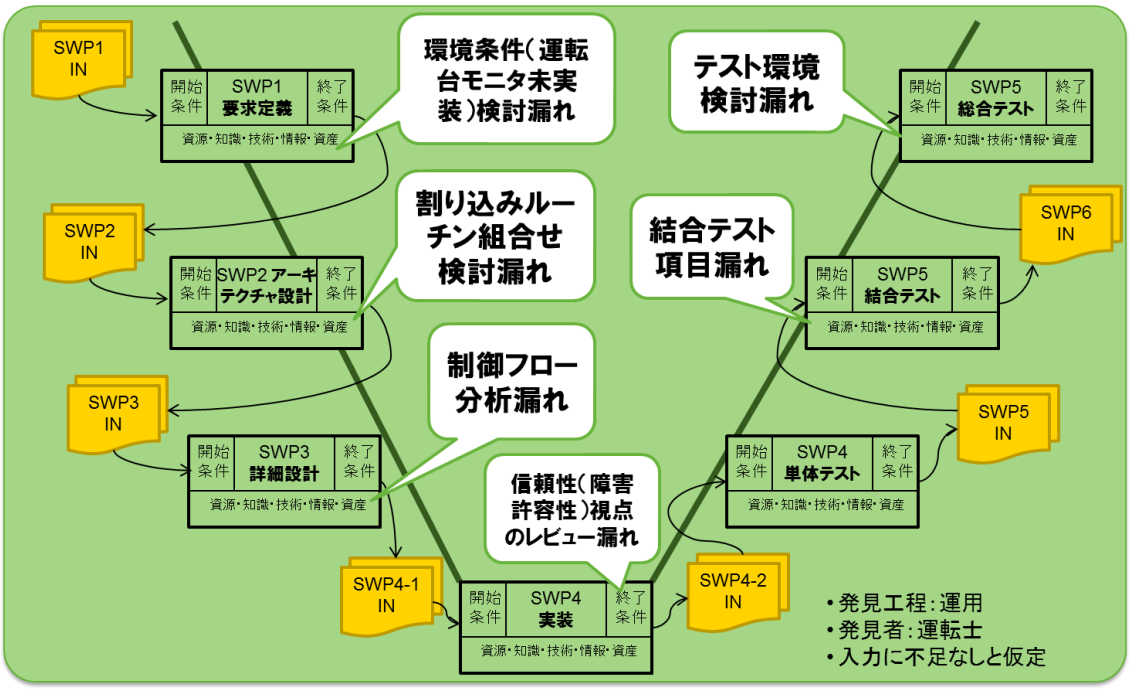


図 7　PNA による作りこみ要因の洗い出し（湘南モノレール）

3.6　発生源・検出漏れ分析

(a)　手法の利用シーン

原因分析に利用する。特に根本原因を推定する際に有効。

(b)　手法の概要

この分析方法は、作りこみ要因及び流出要因を分析する手法である。この手法では V 字モデルを想定しており、PNA のようにタスク（プロセス）を基点として分析を開始する。大まかな手順は以下の通り。

1. 前工程の分析
   V 字モデルで検出した工程と上流工程の作りこみ要因を分析する
2. インプット精度の分析
   タスクの入力に欠陥混入要因がないか検証する
3. タスク安定性の分析
   作業タスクが個人に依存せずに安定して実行できるレベルに詳細化されているか検証する
4. 検出漏れ分析
   発生源の工程のレビューによる流出漏れの分析及び V 字モデルの試験設計レビューによる検出漏れの分析
5. 計画の検証
   仕様インプット、ソフトウェアリリースポイント、制約、リスク、プロセス、実行計画、見積もり等が計画され精度が確保されているか検証する
6. 外乱要因の特定
   仕様変更の多発や出荷直前の無理な変更等の外乱要因により十分な時間確保ができない状況になっていないか分析する

＊本手法は事例集検討のための部会委員から提供されたオリジナルのものである。

(c)　記法ならびに分析法

発生源・検出漏れ分析では、まず表 1 列目に事象を記載し、その事象に関係すると思われるシステムの構成ユニット名とその故障モードを 2 列目、その故障モードに対する対処とその結果を 3 列目に記載する。それぞれの対処/結果につながったと考えられる要因を洗い出して 4 列目以降に記述する。その上で、当該事故に最も関係すると思われる要因を選び、その要因の再発を防止するための是正措置を検討し右端の列に記載する。システムの障害では複数の事象が同時発生する場合もあるため、それらの事象ごとに新たに行を追加していく。

(d)　分析の例

カンタス航空（4.4 節）の事例における作りこみ要因の分析例を表 6 に示す。この例は事象に着目して直接的な原因を推測し、そこから想定される要因を列挙している。

表６　作りこみ要因の分析例（カンタス航空）

現象分析（現象から要因を推定）							
事象			要因			是正	
急降下発生	ADIRUデータスパイク故障モード	故障モードは到着まで解除できなかった	想定外のロジックで動作	バグ	－		
		故障モード起動後異常動作は電源OFF→ONで正常復帰		ROM破壊（PG破壊）	－		
				ハードウェア異常	－		
				物理的環境要因（温度、振動）	－		
				EMI	－		
				機内の外乱	－		
				機外の外乱	－		
				SEE(電子線）※機上装置の継続的なリスク	○	メモリ破壊	クリティカルデータの演算範囲チェック メモリWhite時の再度読み出し照合 データのリフレッシュ
					○	CPU内部回路破壊	
				1.2秒周期でのスパイク未想定	○	例外仕様未定義	例外の継続の仕様を定義する
		手動操作に反応出来ない2秒	想定外の手動切り替え				

3.7　例外分析

(a)　手法の利用シーン

原因分析に利用する。原因箇所の特定に有効。

(b)　手法の概要

例外分析は、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である。物理的な構成要素に関する例外と環境の構成要素に関する例外の 2 種類が考えられる。FTA 等と同様に設計段階の検討で整理されている情報であるが、根本原因を特定する際に要因の考慮漏れの防止に役立つ。

(c)　記法ならびに分析法

予めシステム設計時等に、システムを構成する物理要素（ハード要素）を洗い出し、それぞれの要素について考えられる異常動作を列挙した表を作成しておく。その上で当該システムに障害が発生した際に、関連すると思われる物理要素とその異常動作を選び出し、その部分に対する対応策を検討確認する。

＊本手法は事例集検討のための部会委員から提供されたオリジナルのものである

(d)　分析の例

湘南モノレール（4.1 節）の事例について、物理的な構成要素の関する例外を分析した結果の一部が表 7 である。構成要素によってはより詳細な部品単位で分析を行う。

表 7　例外分析の例（湘南モノレール）

物理例外分析				
物理項目			例外項目	対応
ブレーキシステム			ー	ー
ATS			ー	マスコンを非常ブレーキ位置にするとATSがOFFとなる。 並行動作してアラームによる人間系に伝える改善が良いと思える。
VVVFインバータ	回路		切粉等によるショート	
	通信		データ化け	受信データチェックして、データ化けは読み捨てる。 データ化けを考慮した定時通信等を設計する。
			ノイズ受信	割り込み発生がノイズによる割り込みであるかを判断する設計にする。
	未使用回路	ポート	ノイズ	未使用ポートは出力モードとなるように設計する。
		割り込み	ノイズ	未使用割り込み動作処理を設計する。

3.8　なぜなぜ分析

(a)　手法の利用シーン

根本原因の推定に利用する。

(b)　記法ならびに分析法

なぜなぜ分析は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。基本的な記法は、システムの障害事象を最左側に配置する。その事象を引き起こしたと考えられる原因をその隣に記載し、更にその原因がなぜ起きたかをその右側に追記する。この手順を複数回繰り返すことで、障害の根本原因を探していく。それ以上、展開が困難な原因に達したところで分析を打ち切り、その原因に対する対応策を検討して記述する。

繰り返しの原因分析の視点や粒度等については、後述するように様々な考え方がある。

3.8.1　方法 1

(a)　手法の概要

方法 1 は、事象から遡って根本的な原因を探る各段階で、何を問うのか明確に記述する。すべての要因は各段階で同じ問いに答える必要がある。各段階の問いを具体的に記述することで要因が思いつきやすくなる。また、各段階での問いを揃えることで、論理的な飛躍を伴う要因が記述されることが少なくなる。

(b)　分析の例

湘南モノレール（4.1 節）の事例に対して方法 1 を実施した例を図 8 に示す。この例ではシステムの意図しない動作という事象を起点とした分析となっている。各レベルの問いが、行動誤り→判断誤り→判断根拠→根拠への判断誤り→作り込み要因といった順で統一されている。

縦方向に統一された問いへの返答が配置されているため、おおよそ同じレベルの要因が並ぶようになっていることが見て取れる。

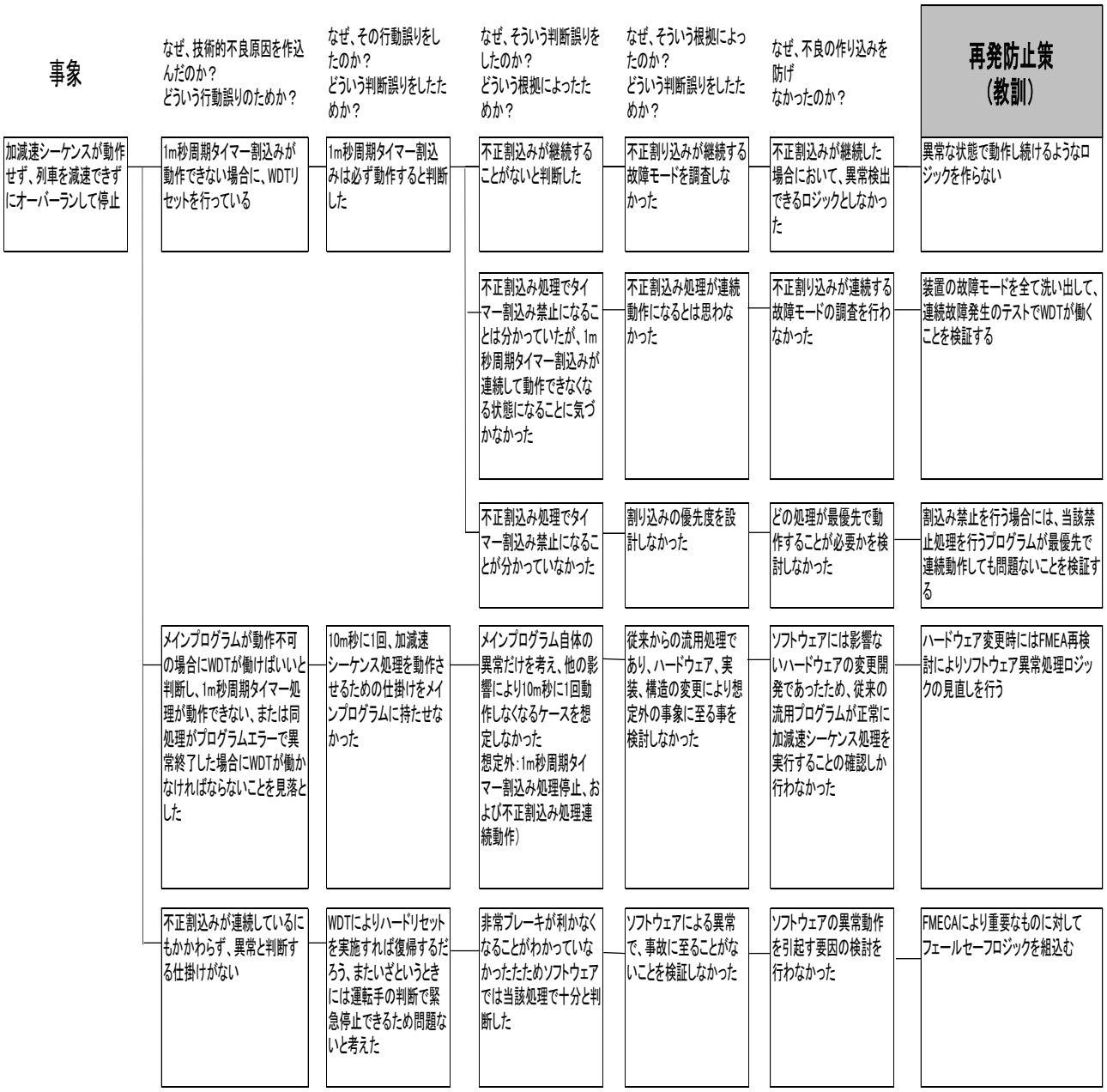


図 8 なぜなぜ分析１の例（湘南モノレール）

3.8.2　方法 2

(a)　手法の概要

方法 2 では、あらかじめ洗い出した問題症状を起点としている。また、根本的な原因を探るためのヒントとして、標準等からの逸脱を考えるようにしている。作りこみの工程もしくは流出の工程を特定することで再発防止策を考えやすいようにしてある。

(b)　分析の例

この方法でアリアン 2（4.3 節）の事例を分析した結果を表 8 に示す。「五なぜの法則」を想定して要因分析が五次まで用意されているが必須とはしていない。また、再発防止策については何を行うかを記述し、具体的な防止策の策定については IPA/SEC の ESxR を挙げて別途行うように区別している。

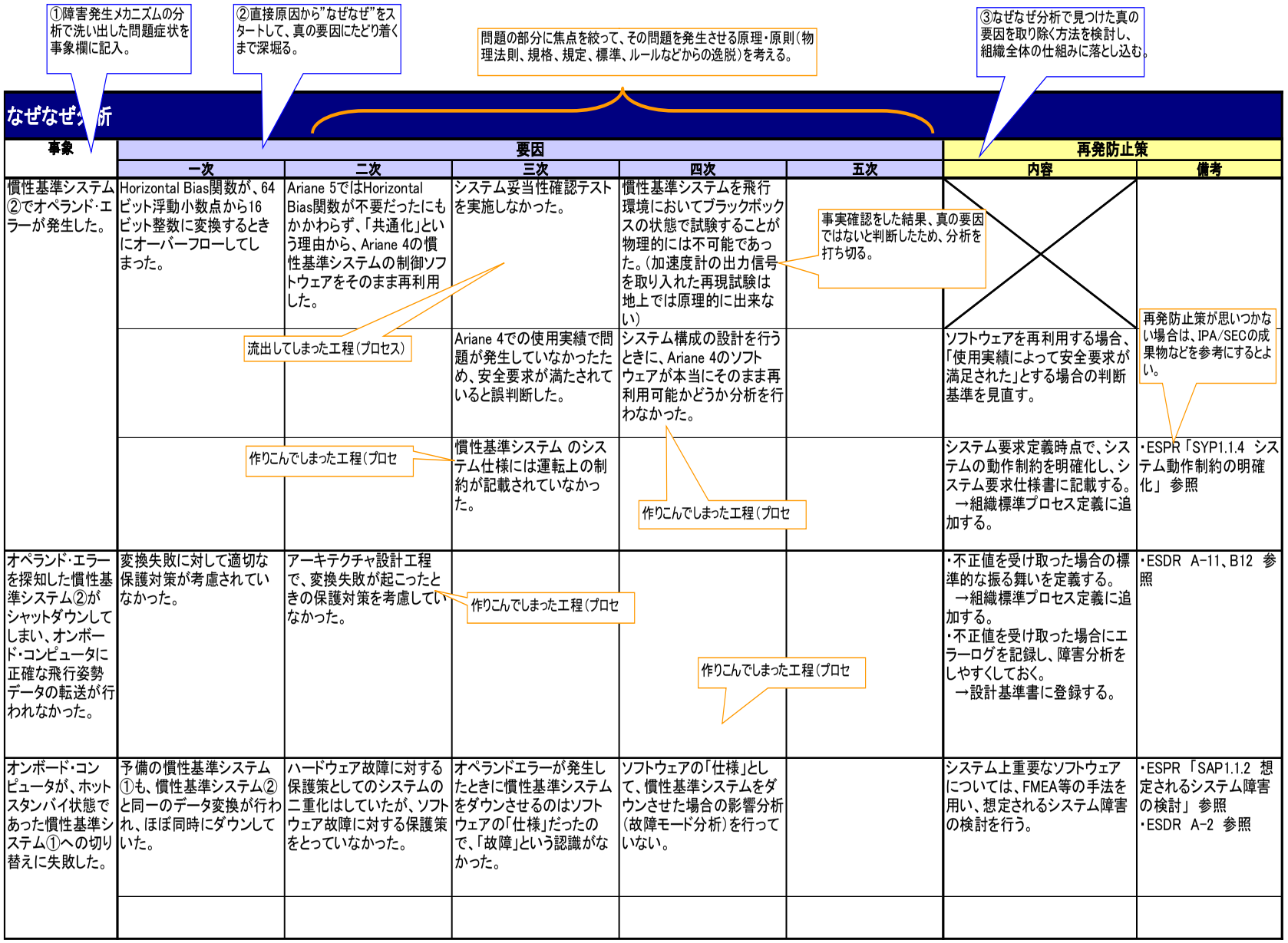


表８ なぜなぜ分析２の例（アリアン５）

3.8.3　方法 3

(a)　手法の概要

方法 3 は、分析・設計・実装・試験のそれぞれについて、作りこみ要因と流出要因の根本原因を定義して対応方法を定義する。方法 3 では、問題が行動や症状を例外と考えて、物理的な例外なのか、環境的な例外なのかの区別を意識しながら分析を行っている。

(b)　分析の例

方法 3 を湘南モノレール（4.1 節）の事例に適用した結果を表 9 に示す。表 9 は設計の作りこみ要因のみ分析している。個別の問題行動が起点となって分析が始まり、その問題行動を引き起こしたと考えられる要因が例外として記述されている。

表 9　なぜなぜ分析 3 の例（モノレール）

根本原因分析					対応
運転指示に応じず加速する	VVVFインバータ誤動作	加減速シーケンスが実行できていない	未使用回路からの不正割り込みで全割り込みが停止する		未使用割り込みの処理を組み込む
			タイマ割り込みが止まる		10msec毎の加減速シーケンスを確実な動作が目的。 タイマ割り込みを監視して、10msec毎の起動を確実に行う。タイマ割り込みが停止すればWDTを止めるアーキテクチャ設計。
			WDTアーキテクチャ不備		
		H/W不良 【物理例外→】			-
	人為的ミス 【環境例外→】	異常事態を検知しながら運転を継続	運行計画通り正しく列車を運転するタスクを優先した	過去の経験値で、非常ブレーキで対応できるかも知れないという誤った認識を持った。	人為的ミスを抑制するシステムからのアラームが必要。 ATSをOFFにしない仕様改善が必要

4. サンプル事例の概要

4.1 湘南モノレール

(a) 障害の概要

この事例は、2008 年 2 月 24 日に湘南モノレール株式会社江ノ島線の湘南深沢駅から西鎌倉駅にかけて発生した鉄道事故である。湘南深沢駅を出発した列車がブレーキ力不足の状態で西鎌倉駅へ進入したために停止できず、同駅通過後に設置された分岐器に衝突して停止した。(図 9)

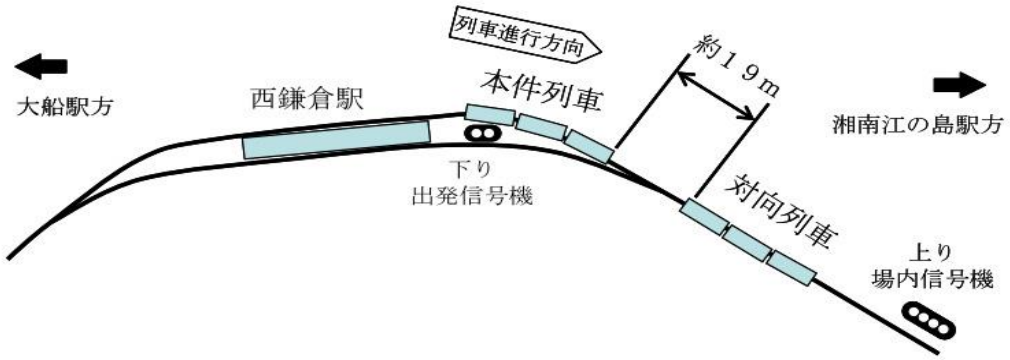


図 9　事故現場略図（文献[4]より引用）

(b) 障害の経緯・及び原因について

鉄道事故調査委員会[4]によると、湘南深沢駅を出発直後に列車は急加速をはじめ、運転士がアクセルをオフしたにも関わらず、加速が継続する状態となっていた。また、非常ブレーキを含むブレーキ操作にも関わらず、分岐器に衝突するまで完全に停止しなかった。

同報告書に挙げられた障害の直接的原因のうち、主なものは、回路へのノイズの重畳及び VVVF インバータの誤動作であった。

(c) 再発防止策について

同報告書では以下の再発防止策が講じられたと述べられている。

・運転取り扱いに関わる指導の改善
・VVVF インバータの制御プログラムの改善

4.2　駒場ダム

(a)　障害の概要

この事例は、2002 年 5 月 9 日に天竜川水系阿知川にある中部電力の駒場ダム（長野県）において発生した異常放流である。駒場ダムを管理する平岡ダム管理所において同ダムの水位変更操作を行ったところ、意図せずに水門が開放された。(図 10)

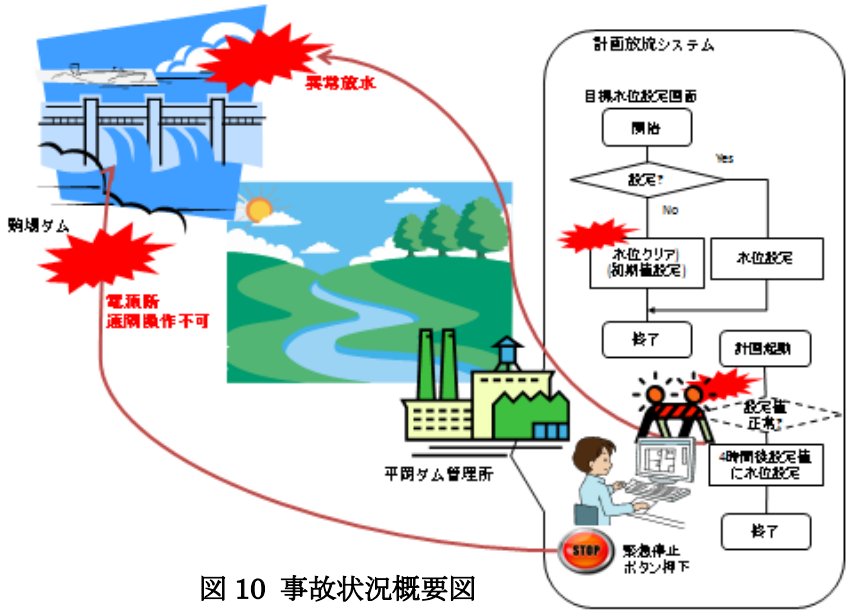


図 10 事故状況概要図

(b)　障害の経緯及び原因について

文献[5] [6] [7]によると、ダムの水位を確認するためには、水位変更操作と同じ操作を変更直前まで行い、変更取り消し操作を行う必要があった。同手順を行った後に同ダムの目標水位が意図せず初期値に変更された結果、水門が開放され、異常放流につながった。さらに、異常に気づいた操作員が誤ってダムの電源を遮断したため、結果的に総量約 2 万立方メートルの貯水が放流された。

同文献に挙げられた異常放流の直接原因は以下の通り。

・ダム水位一定制御プログラムの欠陥
・操作員の操作誤り

(c)　再発防止策について

同報告書では、以下の再発防止策が挙げられている。

・ダム水位一定制御プログラムの欠陥対策
・異常作動ゲート停止操作方法の改善

4.3 アリアン５

(a) 障害の概要

この事例は、1996 年 6 月 4 日に打ち上げられたロケット・アリアン 5 の事故[8] [9] [10]である。同機体は打ち上げ直後に進路から大きく逸れて、39 秒の後に高度 3700m において爆発、四散した。

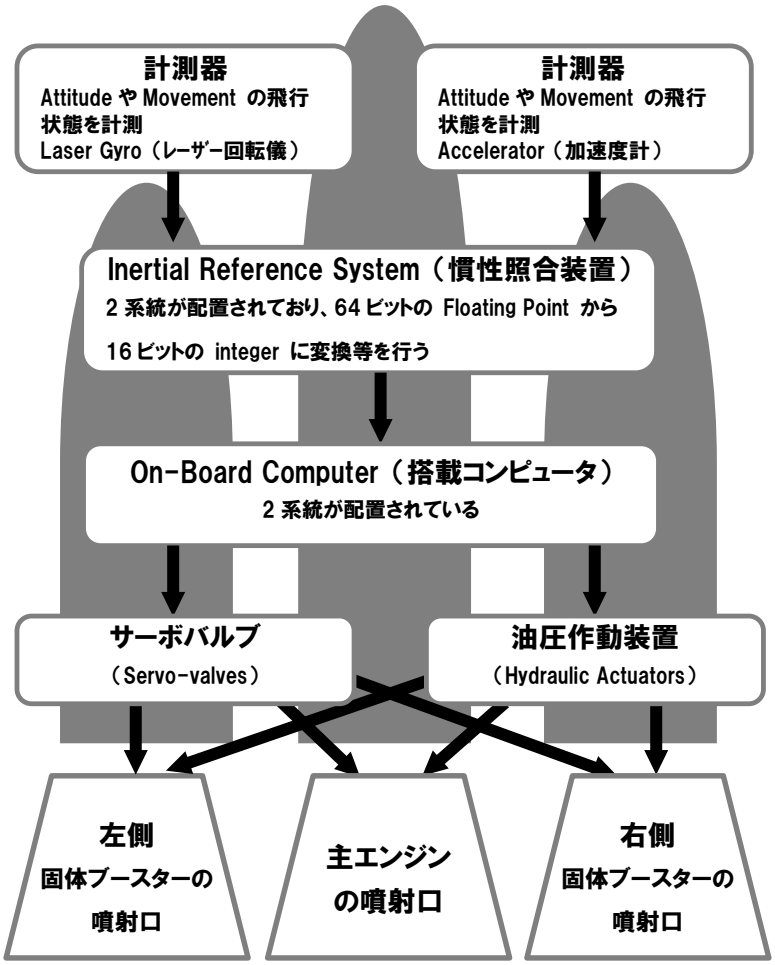


図 11 アリアン 5 の構造概要図（文献[10]より引用）

(b) 障害の経緯・原因について

文献によると、打ち上げ直後に飛行制御システム内の 64 ビットの浮動小数点を 16 ビットの符号付き整数へ変換するルーチンでオペランド・エラーが発生した。同エラーを受けて進路の制御に関わる慣性照合装置（図 11 参照）がシャットダウンし、ロケットの噴射口が不適切な方向に向けられた。その結果、空力的負荷が加わり分解・爆発した。

文献に挙げられた障害の直接的な原因のうち、主なものは以下の通り。

・数値変換ルーチンの不適切な処理

・不正な値に対する保護策の不徹底

本事例集で参考にした文献では、要因として以下の点を挙げている。

・実績があるソフトウェアの盲目的な流用

・不適切なレビュー活動

(c) 再発防止策について

再発防止策については、文献[8]が広い範囲で扱っている。例えば、適切なレビュー活動の実施やソフトウェアの多重化の検討等が挙げられる。

4.4　カンタス航空

(a)　障害の概要

この事例は、2008 年 10 月 7 日、カンタス航空の航空機が飛行中に意図しない急降下を繰り返し、乗員乗客が負傷した航空事故である（図 12）。乗員乗客は急降下によって天井に投げ出され負傷したが、同航空機は最終的には近隣の空港へ着陸できたため死者はいなかった。

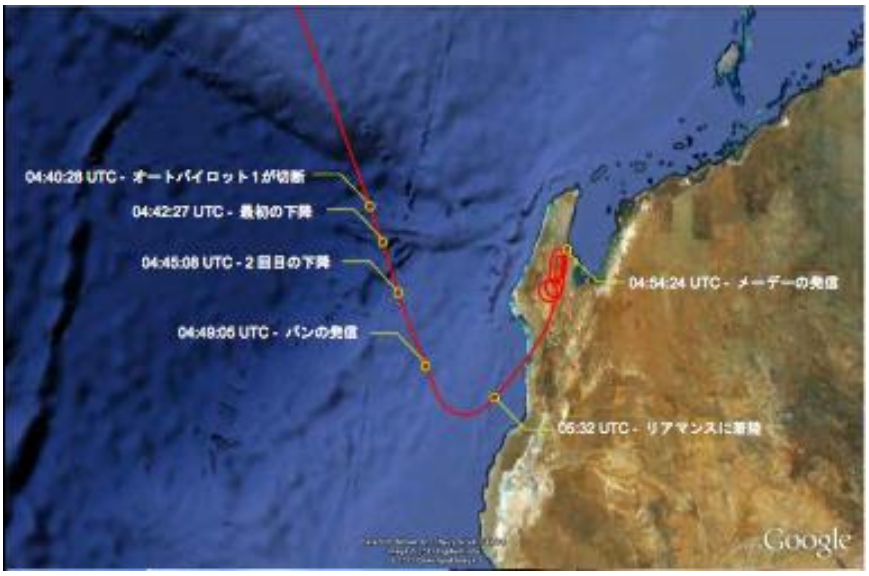


図 12 航空機の経路と主要なイベント（文献[11]より引用）

(b)　障害の経緯・原因について

文献[11]によると、シンガポールを出発した同航空機は、上空 37,000 フィートを飛行中に航空機システムに慣性基準システムより不正な値が入力され、オートパイロットが切断された。その結果、同航空機が急激に機首を下げたため、乗員乗客が天井に投げ出されて負傷した。手動操縦によって最終的に近隣の空港へ着陸した。

文献に挙げられた障害の直接的な原因のうち、主なものは以下の通り。

・断続的なデータスパイク（不正な値）による制御の失敗
・宇宙からの電子線による ADIRU（Air Data and Inertial Reference Unit）のメモリ破壊

(c)　再発防止策について

同報告書では、以下の対策が示されている。

・当該器機のデータ伝送異常の検出能力向上
・シートベルト着用の徹底

*参考文献

[1] 畑村創造工学研究所　畑村洋太郎：失敗知識データベースの構造と表現
http://www.sozogaku.com/fkd/inf/mandara.html
[2] 日本ヒューマンファクター研究所：品質とヒューマンファクター 安心と安全の考え方，財団法人日本科学技術連盟
[3] 金子　龍三：原因分析「プロセスネットワーク分析法（PNA)」の勘所
http://juse-sqip.jp/archives/vol8/qualityone_01.html
[4] 運輸安全委員会：鉄道事故調査報告書：湘南モノレール株式会社　江ノ島線西鎌倉駅構内　鉄道物損事故平成 21 年 6 月 26 日。
[5] 失敗百選 ～長野の駒場ダムの異常放流（2002）～
http://www.sydrose.com/case100/325/
[6]「駒場堰堤ゲート異常作動原因調査　報告書」（平成 14 年 6 月　国土交通省中部地方整備局）
[7]「ダム等ゲート類の異常作動等の再発防止について　－　点検・確認結果　－　」（平成 14 年 7 月 26 日　原子力安全・保安院）
[8] N. G. Leveson: The Role of Software in Space Craft Accidents.
[9] 失敗百選 ～アリアン 5 型ロケット爆発事故～
http://www.sydrose.com/case100/284/
[10] アリアン5の爆発事故とソフトウエア安全性に関する国際規格(横浜国大　清水久二)安全工学　安全工学 41（1）, 39-42, 2002-02-15 安全工学協会
[11] オーストラリア運輸安全局：運輸安全報告書　航空事象調査 AO-2008-070.

ＰＡＲＴ　Ⅳ

障害分析手法事例解説書（組込みシステム編）

PART Ⅳ 目次

1. はじめに

1.1 本解説書の概要

組込みシステムには高い信頼性が求められる。開発中や出荷後に発生した障害をきちんと分析して根本的な原因を特定し、同種の障害が再発しないように開発の進め方を改善していくことが信頼性を高めるために重要となる。本解説書は開発現場での障害分析に役立つ知識の提供を目指して作成された。本解説書の構成は以下の通りである。

2 章：障害分析手法と分析作業

3 章：障害分析事例解説

4 章：再発防止活動の事例

2 章では、障害が発生した後に行われる作業の種類や内容について解説している。また、分析に必要となる情報の種類についても述べている。

3 章では、経験豊富な技術者が普段どのように障害を分析しているのか、事例に沿ったかたちで解説している。具体性を持たせることで分析作業を行う際の思考の過程や分析手法の使われ方を読み取れるように努めた。なお、本解説書における事例は全て架空のものである。実際に発生した障害を参考にしているが、開発体制や根本原因等は執筆に携わった技術者による経験や推測による創作である。

4 章には、障害の分析事例に加えて、分析結果を再発防止につなげる取組みの事例も紹介している。分析の結果得られた知見を他の開発に活かすための参考となれば幸いである。

1.2 本解説書で使用する用語について

障害、故障、欠陥等の用語は、さまざまな意味で使用される。本解説書では、障害という用語は JIS X 0014 に定義される障害（fault）の意味で用い、故障（failure）や欠陥（defect）は IEEE 1044 の定義や IEEE 982.1 の記載内容に沿って用いる。

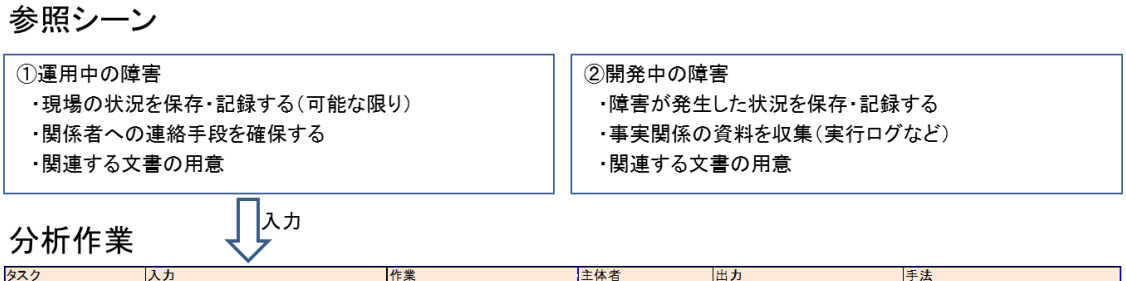
【障害】要求された機能を遂行する機能単位の能力の、縮退又は喪失を 引き起こす、異常な状態。
【故障】要求された機能を遂行する製品の能力が尽きる状態、または事前に仕様化された制限内での機能を遂行する能力が無い状態とする。
【欠陥】設計者の認識の有無にかかわらず、すべての成果物において要求定義の誤り、仕様設計の誤り、プログラミングの誤り、システム構築の誤り等により「期待される結果」と乖離があるために、何かしらの対策・対応が必要と考えられる事象またはその原因。
【要因】原因の候補。

2. 障害分析手法と分析作業

本書の第 3 章で解説する障害分析手法と障害分析作業を紹介する。

2.1 障害分析手法と分析作業の概覧

第 3 章で解説する分析事例では、障害が発生してから対策を検討するまでの分析作業の中でいくつかの手法が使われている。図 2.1 は、組込みシステムを開発する日本の先進企業が使っている手法や一般にシステム障害の分析手法として知られている手法をそれが使われる場面に対応させて整理したものである。



タスク	入力	作業	主体者	出力	手法
情報収集	①, ②の情報, 人	情報収集と整理	プロジェクトリーダ 開発担当者 営業担当者	収集した情報を整理した文書	障害管理票の調査 関係者へのヒアリング 再現実験
システム構造の把握	収集した情報を整理した文書 設計書・仕様書	システム構造の整理	プロジェクトリーダ 開発担当者 有識者（HW設計者）	システム構造図	ブロック図 UML SysML ネットワーク図
問題症状の把握	収集した情報を整理した文書 システム構造図	問題症状の把握	プロジェクトリーダ 開発担当者 有識者（HW設計者）	問題症状を整理した図表	表 VTA
原因分析	収集した情報を整理した文書 システム構造図 問題症状を整理した図表 担当者の経験・過去事例 FTA/FMEA(設計時作成)	原因箇所の特定	プロジェクトリーダ 開発担当者 有識者（HW設計者）	原因箇所の候補一覧表	問題行動分析 例外分析 FTA/FMEA
	原因箇所の候補一覧表 設計書・仕様書 観点表（設計時に作成）	直接原因の特定 影響度・範囲の分析 暫定的対処案の立案	プロジェクトリーダ 開発担当者 製品の品質保証部	直接的原因の一覧表 ・人為的ミス ・設計誤り	レビュー コードインスペクション 再試験 シミュレーション モデル検査
	直接的原因 開発記録（実績） 開発プロセス（定義） ヒアリング（開発担当者）	根本的な原因の推定	プロジェクトリーダ 開発担当者 SQA(品質管理部門)	根本的原因 ・プロセスの欠陥 ・未実施プロセス 視点で整理 ex. 制御可能な項目化	PNA 発生源・検出漏れ分析 なぜなぜ分析 KJ法（課題）
対策の立案	根本的原因 原因に伴う制約事項	対策の検討	プロジェクトリーダ 開発担当者 管理職	短期対策 長期対策	影響レベル評価尺度・評価手法

図 2.1　分析作業の各タスクと分析手法の概要

図 2.1 の上部には障害分析の起点となる二つの参照場面、すなわち、運用中の障害発生と開発中の障害発生を配している。運用中に障害が発生した場合は、組込みシステムの運用者や障害の発見者が重要な情報源となるため連絡が取れるようにする必要がある。一方、開発中に障害が発生した場合は、稼働中の障害と異なり、障害への対応が緊急性を要することはまれである。しかしながら、障害の分析に必要となる情報を収集する活動が開発プロセスに組み込まれていると後々の分析作業を円滑に進めることができる。そのため、障害を分析する時にどのような情報が必要となるか、折に触れて整理しておくとよい。

図 2.1 の下部は障害分析作業で行われるタスクを示している。各タスクにおいて、誰が（作業の主体者）何を行うか（分析作業）、また、どのような分析手法が利用されているのかを例示している。障害を分析する際には図 2.1 を参照することで何をすべきか把握することができる。

2.2 障害発生から再発防止策の立案まで

図 2.2 は障害が発生した際の分析作業で行われるタスクを示したものである。障害が発生すると障害に関する情報取集にまず取り組む。そしてシステム全体の構造を把握したうえで問題症状を整理する。整理された情報をもとに問題の原因を探り、再発防止のための作を立案する、という流れとなっている。ただし、この流れは一方向ではない。例えば、原因分析において重要な情報が不足している場合には追加の情報を収集する。

図 2.2 のタスクは ESPR のサポートプロセスの一つである問題解決管理（(SUP6)）に含まれるタスクと共通する部分も多い。一方で、障害分析では出荷後に発生した障害も対象となり、長期的な信頼性向上のための取り組みとして再発防止策を立案する必要がある。

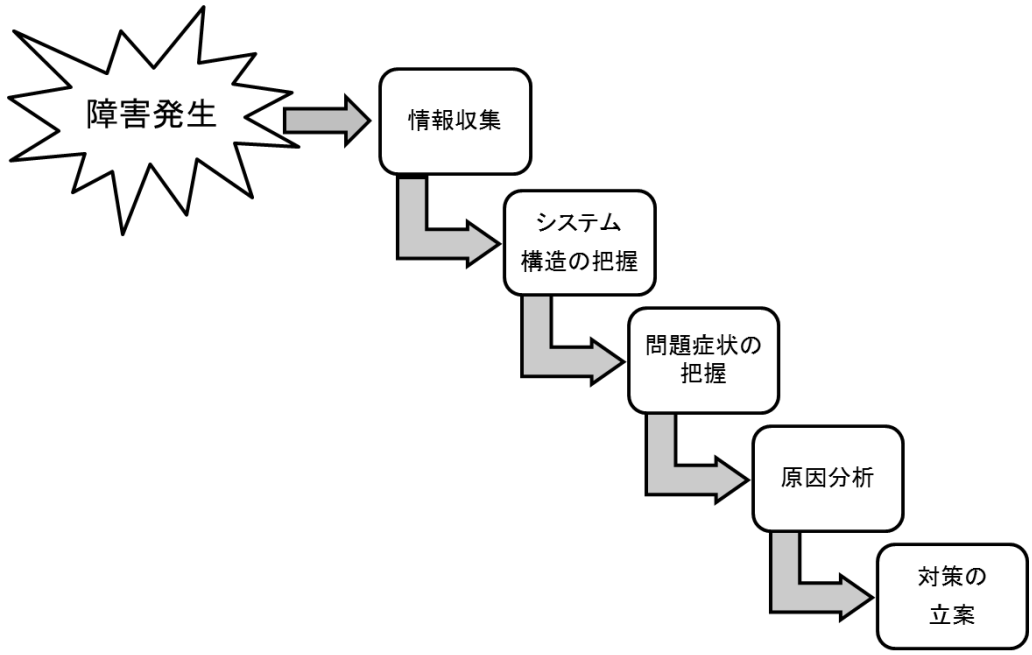


図 2.2　障害分析の流れ

2.3 各タスクの詳細

(1) 情報収集

障害の分析に際してまず行うのが情報収集である。ただし、稼働中のシステムで発生した障害については事態の収拾を優先する。障害の分析に必要な情報の収集は事態の収拾後速やかに行う。収集した情報は観点別に整理して文書にまとめる。

障害の分析に必要となる情報の例として以下のものがあげられる。

- 障害の発生状況（関係者に対するヒアリングを通じて）
- 組込みシステムの稼働ログ情報

- 障害の再現手順
- 計算機システムの構成
- 外部環境

システムの種類によって収集が必要な情報は異なる。また、収集できる情報が限られる場合もある（大規模な事故等)。このような場合でも必要と思われる情報をできるだけ集めることが以降の分析を円滑に進めるために重要である。

(2) システム構造の把握

収集した情報は大きく分けて 2 種類に分類できる。すなわち、障害発生の経緯に関する情報と、システムの動作・構造に関する情報である。これらの情報を整理して対応付けることで障害に至った問題症状の把握が可能になる。

まずはシステムの動作・構造に関する情報を整理して、システムの全体像を見渡せるような簡潔なシステム構造図を用意する。システムが複数のサブシステムから構成されている場合は、サブシステム内の構造は別の図に分ける等簡潔さが損なわれないように気を付ける。全体を見渡すことができる図を用意することは、サブシステム間の相互作用の関係が明確になる等、対象システムの構造把握に役立つ。

システムの設計時に作成した UML 等の図がそのまま利用可能であることが多いが、システム全体の概要をとらえるために、簡易なブロック図等を用意してもよい。

(3) 問題症状の把握

次に、時系列と相互作用を意識しながら事象を整理して問題症状を把握する。手順としては、まず、ヒアリング等で得た事象に関する情報を時系列に沿って整理する。このとき、観点を定めて事象を整理すると分析しやすくなる。例えば、システム、システムの運用に従事している人、組込みシステムの利用者等の観点が考えられる。

システムが複数のサブシステムから構成されている場合は、サブシステムごとに情報を整理する。また、システム構造の把握の際に作成したシステムの構造図を利用して、収集した情報に矛盾する点や曖昧な点がないことを確認する。

最後に、障害に関わった人やシステムの構成要素の間の相互作用が分りやすくなるように事象を整理して一覧性の高い形式で図表にまとめる。

(4) 原因分析

原因分析は 3 つのステップで構成されている。すなわち、①原因箇所の推定、②直接原因の特定、③根本原因の特定、である。各ステップについて以下で述べる。

① 原因箇所の推定

まず、問題症状を引き起こしたと考えられるシステム上の原因箇所を推定する。複数の観点から原因を推定することで漏れや抜けを防ぎやすくなる。例えば、以下のような観点が考えられる。

- ソフトウェアの観点
- ハードウェアの観点
- 人の観点

- 環境・想定条件の観点

事象について整理した図表やシステム構造図に加えて、原因分析の担当者の経験等も貴重な情報源である。社内で過去の障害事例のデータベースが整備されている場合は参照する。「情報処理システム高信頼化教訓集（組込みシステム編）」（参考文献[1]）の PART1、PART2 等も参考にするとよい。この際、システムの設計時に作成した FTA や FMEA 等を利用することで、考慮すべき故障モードの漏れや抜けを防ぎやすくなる。

情報を得ることが難しい場合は一定の仮定を置いて原因箇所を推定する。このような仮定については、事実と区別できるように記録する。

② 直接原因の特定

原因箇所の推定結果に基づいて直接原因を特定する。この段階では、再試験によって障害の再現条件を調査する等の作業が必要となる。ソフトウェア部分に問題があると推定される場合には、コードレビューやインスペクション等を実施する。

原因箇所の推定作業が不十分であるために直接原因が見つからない場合もある。その場合は前のステップに戻って再分析を行う。

直接原因を特定した後、その影響度や影響範囲を分析する。分析結果に応じて緊急の対処方法を考える必要がある。

③ 根本原因の推定

次に、直接原因を引き起こした過程を分析し根本原因を推定する。多くの場合は設計誤りや人為的ミスが直接原因となるが、そのような誤りやミスを混入させた要因や流出させた要因を複数の視点に立って分析する。例えば、開発プロセスの観点から直接原因を眺めることで、障害を引き起こす欠陥を作りこんだ工程やその工程の不備を特定する。この場合、開発担当者へのヒアリング等が貴重な情報源となる。

(5) 対策の立案

特定された根本原因を取り除く対策を検討する。ただし、全ての原因が取り除けるわけではない。そのような場合は影響を軽減する対策を検討する。開発プロセスや体制に関する現状や制約を踏まえたうえで 2 種類の視点で再発防止策を考える。

- 短期的対策
- 長期的対策

開発中に発生した障害であれば、開発中のシステムを点検して同種の障害が発生していないことを確認する。また、その後の開発で同種の障害が再発しないような対策を立案する。

開発終了後には、他のプロジェクトにも適用できる長期的な再発防止策を考える。長期的な対策としては

- ガイドラインの整備（技術面について）
- 規定の改定（プロセスの定義等）
- 教育体制の整備
- 知見に関するデータベースの整備

等が挙げられる。再発防止策を考える際に重要なのは、技術的課題であるのか管理的な課題であるのか等、複数の観点から検討することである。また、根本的な原因が影響を及ぼす範囲や度合いを評価する手

法を整備することで、妥当な再発防止策を選びやすくなる。

3. 障害分析事例解説

3 章では、経験豊富な技術者が普段どのように障害を分析しているのか、事例に沿って解説する。具体的な事例を用いることにより、分析作業を行う際の思考の過程や分析手法の使われ方を読み取ることができる。なお、事例は実際に発生した障害を参考にしているが固有名称は架空のものに変えており、開発体制や根本原因等不足する情報は、執筆者の経験や推測によって創作している。

3.1 未来都市モノレール障害の分析

「未来都市モノレール」障害事例について、「なぜなぜ分析」手法を使って、2.2 項の障害発生から再発防止策の立案までの流れ（手順）に沿っての分析の考え方、結果を解説する。

障害事例は架空のものであるが、実際に発生した障害を参考に開発体制や根本原因等を創作している。（参考文献[2]）

次の順番で分析の手順を説明する。

3.1.1　情報収集とシステム全体の把握（システム構造の把握）

3.1.2　問題症状の把握（事象経過）

3.1.3　原因分析

3.1.4　対策の検討とまとめ

3.1.1 情報収集とシステム全体の把握（システム構造の把握）

先ず、以下の 3 つの視点で事実情報を収集する

・何（モノ）で発生したか

・登場人物（ヒト）は誰か

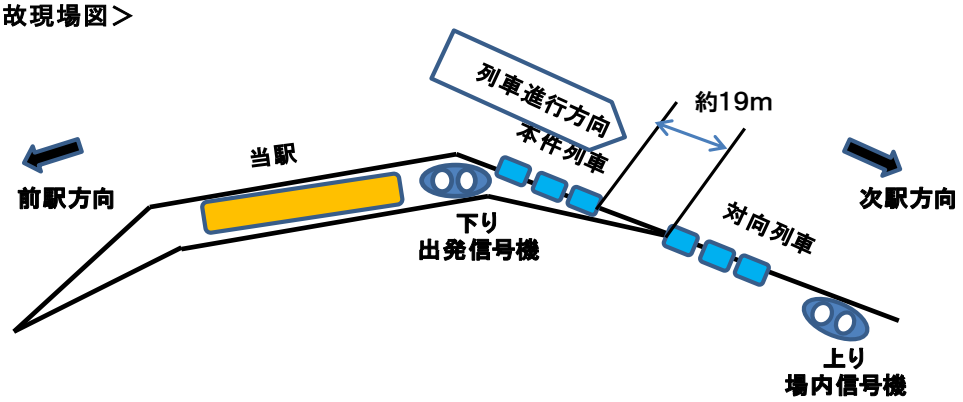
・何が起きたのか（コト）

(1) 事故の概要

原因分析では、原因は必ず結果の前にくることを前提に行うので、事故の概要を時系列に記載する（図3.1.1）。

鉄道事故の概要

＜事故現場図＞



未来都市モノレール株式会社の始発駅発終着駅行き3両編成の下り普通列車は、平成26年3月31日、前駅を定刻（9時50分）に出発した。
列車は、前駅に到着するまでは特に異常はなかったが、
同駅を出発する際に急加速し、その後、
運転士がワンハンドルマスコン(*1)を力行(*2)位置としていないにもかかわらず加速する状態になった。
列車は当駅進入時にブレーキ力不足の状態となり、運転士は非常ブレーキ(*3)及び保安ブレーキ(*4)を使用したが所定位置に停止せず、停止信号を現示していた同駅の下り出発信号機を冒進した。
同下り列車は、その先の分岐器に衝突し、接近していた上り普通列車の進路を支障して停止した。
一方、次駅を定刻より約30秒遅れて（9時52分30秒ごろ）出発し、当駅で同下り列車とすれ違う予定であった同上り列車の運転士は、同駅の約60m手前で同下り列車が当駅の下り出発信号機を冒進してくるのを認めたため、非常ブレーキを使用し、同下り列車の約19m手前で停止した。
車両の異常に気付いたが、すぐに列車を止めようとは思わなかった。最初はマスコンの入れ間違いか何かでノッチオフ(*5)すれば戻るのではないかと思ったが、変わらなかった。おかしいとは思ったが、そのまま行ってしまった。最終的には非常ブレーキを使用すれば止まるかもしれないという思いがあったのかもしれない。
当駅下り場内信号機付近で、本件運転士が「ブレーキが効かない」と言ったので本件運転士の方を見ると、確かにマスコンは非常ブレーキ位置にあったが、体感ではブレーキがほとんど効いていない状態であった。急いで右側にある非常ブレーキ引きスイッチを扱ったが、減速する感覚はほとんどなかった。本件運転士は左側の非常ブレーキ引きスイッチに続いて保安ブレーキスイッチを扱ったが、それでも減速する感覚はほとんどなかった。
同下り列車の車両及び分岐器(*6)等の施設に物損が生じたが、双方の列車の乗客及び乗務員（同下り列車には乗客22名及び乗務員2名、同上り列車には乗客16名及び乗務員2名）には死傷者はなかった。

＊1：「ワンハンドルマスコン」とは、列車の加減速を制御する主幹制御器とブレーキハンドルを一つのハンドルで操作可能にしたもの

＊2：「力行」とは、列車に駆動力をかけて走行させること

＊3：「非常ブレーキ」とは　非常事態及び列車分離等に場合にも作用するようにフェイルセーフ機構になっているブレーキで空気ブレーキのみ作用させる

＊4：「保安ブレーキ」とは、常用・非常ブレーキ系に異常が生じて使用できない場合に使用するために設けられたブレーキをいい、指令回路や空気源等が常用・非常ブレーキ系とは独立して設けられている。

＊5：「ノッチ」とは列車の速度を制御するマスコンの刻み（段）のこと。「ノッチオフ」とは、ワンハンドルマスコンを駆動力がかからない状態にすること

＊6：「分岐器」とは、モノレールの軌道を合流・分岐させ列車の進路変更を可能にするもの

図 3.1.1　事故の概要

(2) 製品の概要

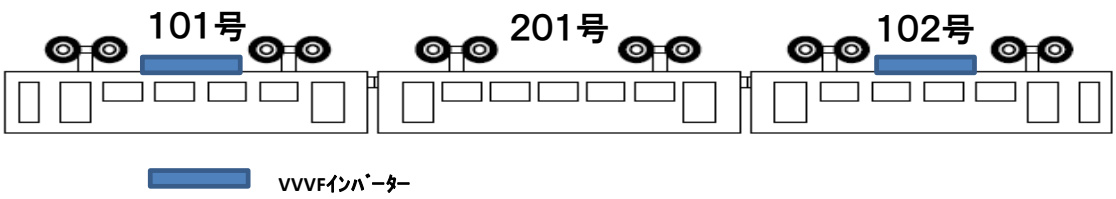
製品の機能ブロックを構成要素とするシステム全体図・システム要素の概要を把握するため機能ブロック図を作成する。

・既存資料があれば参考にする
・自分の担当部分に原因があるのではないかと感じている人が作成開始
・自分の所掌範囲を中心にその周囲を関係者にヒアリングしながらまとめ、自分の理解が正しいかを確認する

☞ **留意点**

・自分の範囲の外も含めて書くことで全体から眺める
・模式図はもともと存在しないのが当たり前と考える

主回路



3両編成全車が電動車で、走行用に12台の主電動機（モーター）が搭載されている。
本件編成のVVVFインバータ(*1)は、1両目である101号と3両目である102号に搭載され、それぞれ自車の主電動機4台と中間車である201号の主電動機2台の計6台を制御している。

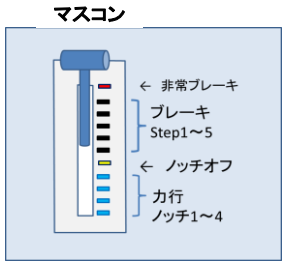
＊1：「VVVF インバータ」とは、電圧及び周波数ともに変えることが可能なインバータ（直流を交流に変換する装置）をいう。

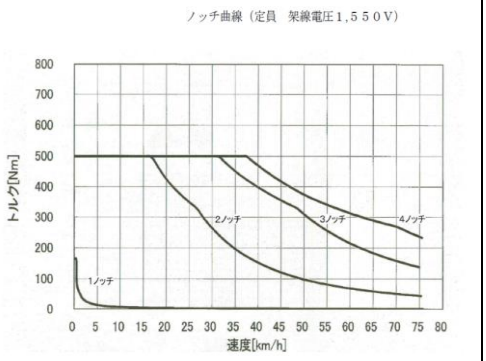
図 3.1.2　主回路

制御回路

力行ノッチと加圧される指令線の関係

		マスコン設定							
		後進（上り方向）				前進（下り方向）			
		1	2	3	4	1	2	3	4
指令線番号	2	○	○	○	○	―	―	―	―
	3	―	―	―	―	○	○	○	○
	4	―	○	○	○	―	○	○	○
	5	―	―	○	○	―	―	○	○
	8	―	―	―	○	―	―	―	○





本件編成の制御回路はマスコンによるマニュアル操作で、力行は4ノッチまである。マスコンからの力行指令を受けたVVVFインバータは、加速制御のシーケンスをソフトウェアで処理してトルク指令を出し、力行ノッチに応じたトルクを発生させ、列車を加速させる。

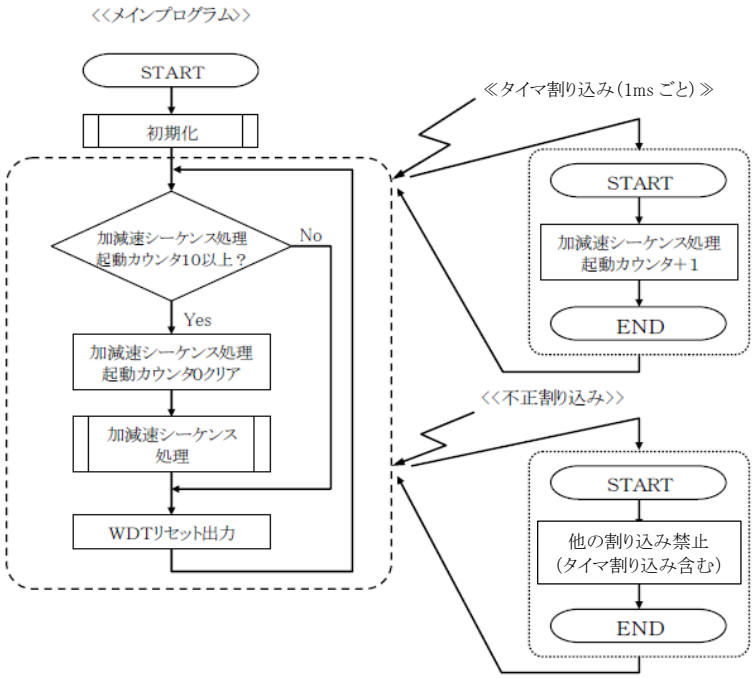
加速制御のシーケンスには戻しノッチ機能がないため、いったん投入された力行指令は高位（数字の多い位置）の指令があるか、又は、ノッチオフされるまで保持される。

「戻しノッチ機能」とは、高位のノッチから低位のノッチに戻すと、直接戻した低位のノッチ指令に移行する機能をいう。

4ノッチから1ノッチ、又は2ノッチに戻すと、走行速度が15km/h 以上の場合は操作時の速度を基準として一定の速度を保つ定速モードに入る仕組みとなっ
ている。

図 3.1.3　制御回路

加減速制御プログラムとウォッチドッグタイマ(WDT)による保護動作



本件編成の本事故発生時におけるVVVFインバータ制御のメインプログラムは、加減速シーケンス処理が実行されないとき(カウンタの値が9以下のとき)はWDT(ウォッチドッグタイマ)リセット出力を繰り返すようになっていた。

1ms 間隔のタイマ割り込みで更新されるカウンタの値が10になる(10ms 経過)と加減速シーケンス処理が実行され、その間はWDTリセット処理が実行されない。

加減速シーケンス処理に一定時間以上要するような異常が発生したときはWDTの保護動作が働き、主回路の電流をいったん遮断してからVVVFインバータ制御のメインプログラムを再起動させる仕組みとなっていた。

VVVFインバータ制御プログラムにおいて不正割り込みが発生した場合、タイマ割り込みを含む他の割り込みを禁止してメインプログラムに戻る。この場合、加減速シーケンス処理起動カウンタが更新されなくなるため、加減速シーケンス処理が実行されずにWDTリセット出力を繰り返すこととなる。

何らかの理由により不正割り込みが発生し、加減速シーケンス処理が実行されなくなると、運転士の操作がモーターのトルク指令値に反映されず、かつ、WDTによる保護動作が働かないために、列車の加減速を決めるモーターの駆動力は、不正割り込みが発生する直前の状態を継続する仕組みとなっていた。

VVVFインバータ制御プログラムにおいて、運転台モニタ用の伝送回路からの伝送開始データ受信に続く受信割り込みは有効な割り込みとして処理されるようになっている。

VVVFインバータには、その動作情報を常時監視し、故障内容及び故障が主回路の過電流やWDT保護動作など重故障である場合に、故障発生の0.7秒前から0.3秒後までのVVVFインバータの動作情報を記録する機能がある

図 3.1.4　加減速制御プログラムとウォッチドッグタイマ（WDT）による保護動作

ブレーキシステム
常用ブレーキは、回生ブレーキと空気ブレーキを併用している。主として回生ブレーキ(*1)が作用するが、回生電流が立ち上がっていないブレーキ開始時と停止直前には空気ブレーキが作用する。
回生ブレーキ作用中は、空気ブレーキにはBC圧(*2)約30kPa の初込め圧が作用する。
「初込め圧」とは、電気(回生)ブレーキから空気ブレーキへの切換を円滑にするため、電気(回生)ブレーキ作用中も制輪子を車輪に軽く押し付ける程度の圧力をいう。
回生ブレーキ作用中は、運転台右側にある表示灯群の中にある黄色の「電制」表示灯が点灯する。
各運転台には、回生ブレーキの開放用のスイッチとして‘「電制ブレーキ」スイッチ’(以下「電制スイッチ」という。)が設けられている。このスイッチを「切」とすると、常用ブレーキにおいても回生ブレーキは作用せず、常に空気ブレーキのみが作用する。
非常及び保安ブレーキでは、空気ブレーキのみが作用する。
ブレーキ指令はすべて電気指令となっている。常用及び保安ブレーキは指令線の加圧で動作し、非常ブレーキは常時加圧された指令線が無加圧となることで動作する。
常用及び非常ブレーキを動作させるための圧力空気は「供給空気だめ」から供給される。
保安ブレーキを動作させるための圧力空気は、「保安だめ」と呼ばれる空気だめから常用及び非常ブレーキとは別の配管を経由して供給され、常用及び非常ブレーキの空気系統に異常があった場合にバックアップする仕組みとなっている。

＊1:「回生ブレーキ(電気ブレーキ)」とは、駆動のためのモーターを発電機として作用させ、発電するのに必要なトルクを制御することでブレーキとして利用する

＊2:「BC 圧(力)」とは、車両の圧力空気を使って車輪の回転を止めるため基礎ブレーキ装置に出力する必要な圧力のこと

図 3.1.5　ブレーキシステム

3.1.2 問題症状の把握（事象経過）

まず問題の結果を整理することで、問題の症状を把握する。具体的には、「失敗まんだら（失敗結果の分類)」（付録１　図 1-2 参照）の内側の円内に書かれている項目を参考に問題（結果）を抽出する。

［物への結果］

①　機能不全…諸元未達・ハード不良・ソフト不良・システム不良、等。

②　不良現象…機械現象・熱流体現象・化学現象・電気故障、等。

③　破損…劣化・減肉・変形・破壊と損傷・大規模破損、等。

［外部への影響を伴う結果］

④　二次災害…損壊・環境破壊、等。

［人への結果］

⑤　身体的被害…人損・発病・負傷・死亡、等。

⑥　精神的被害…PTSD（(心的外傷後ストレス障害）等

［組織・社会への結果］

⑦　組織の損失…経済的損失・社会的損失、等。

⑧　社会の被害…社会機能不全・人の意識変化、等。

［これから必ず起こる結果］

⑨　未来への被害…未出来（みしゅったい）の結果・予想可能な結果・予想不可能な結果、等。

［起こるかもしれない結果］

⑩　起こり得る被害…潜在危険・ヒヤリハット、等

☞ **留意点**

・問題症状は最終状態（結果）から遡って最終原因（直接）を発見するまで結果に対する要因を聞いていく

・ポイントを掴む、技術的知識を持っていることが前提

・途中の症状に対する原因と対策も重要・・・根本原因につながる（1 の事実の整理計に戻ってみる）

問題症状
運転士がワンハンドルマスコンを力行位置としていないにもかかわらず加速する状態になった。
車両の異常に気付いたが、すぐに列車を止めようとは思わなかった。最初はマスコンの入れ間違いか何かでノッチオフすれば戻るのではないかと思った。
運転士は非常ブレーキ及び保安ブレーキを使用したが所定位置に停止せず
同下り列車は、その先の分岐器に衝突し、接近していた上り普通列車の進路を支障して停止した。

図 3.1.6　問題症状

3.1.3 原因分析

(1) 障害メカニズム分析

(ア) 行動分析

「失敗まんだら（失敗行動の分類）」（付録 1　図 1-3 参照）を参考に問題行動を抽出する。

［物への行動］

①　計画・設計…計画の不良・他からの設計をそのまま使ってしまう流用設計、等。

②　製作…ハード製作中の行動・ソフト製作中の行動、等。

※ソフト製作については開発プロセスを意識し、「作り込み要因」と「流出要因」について深堀りして分析を行う。

③　使用…機械の運転や使用・保守や修理・輸送や貯蔵・廃棄、等。

［人の行動］

④　定常操作…手順の不遵守・誤操作、等。

⑤　非定常操作…操作の変更・緊急操作、等。

⑥　定常動作…不注意動作・危険動作・誤動作、等。

⑦　非定常動作…状況変化時の動作・体調不良時の動作、等。

⑧　誤対応行為…連絡不備・自己の保身のための間違った行為、等。

⑨　不良行為…倫理や道徳の違反・規則の違反、等。

⑩　非定常行為…変更・非常時行為・無為、等。

☞ **留意点**

（ｉ）ハードウェア（HW）要因は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　　例：テストしたか

→ プロジェクトの制約

→ 技術

→ マネジメント

・プロジェクトの目的、置かれた状況（コストをかけられる案件ではなかった等）を聞いていくと判明する

（ⅲ）怪しそうなモジュールを抽出しテストして直接原因を探索する

・多くの場合、単体テストは実施されており問題がなくても結合して動かした際の I/F、状態遷移において問題が検出される

・どうしても問題が見つからないことがある。その場合、マイコンのバグやコンパイラのバグ等も疑ってかかる必要がある

・ミドルウェア等購入製品に問題があると思い込んで考えないこともある

・モノ（マイコンのバグ等）の知識やインタビュースキルも重要

（ⅳ）自分の推定が正しいかどうかを検証する

・時系列的に、結果→原因というように追求する

・ゴールにたどりついてもその途中で気になったものは残さず追求する

☞ **留意点**

怪しそうだというのは経験則で判断する・・・ベテラン

・・・こうあるべき

・・・こういうときはこうするでしょきっと

教訓集で見つける・・・・・経験のない人

原因
2台あるVVVFインバータのうちの1台が、誤動作により力行継続状態となった
運転士が本件列車の異常に気付きながら運転を継続した
低圧回路のマイナス極側に重畳したノイズの影響を受けやすい状態となっていた
未使用のモニタ伝送回路に対して適切なノイズ対策がなされていなかった
ウォッチドッグタイマによる保護動作が働かなかった

図　3.1.7　原因

(2)　根本原因

下記の 4 つの観点で原因を深堀していく。個人の責任を追及するためではなく、あくまで原因が何に起因するかの観点で進める。

以下対策を打ちやすくするために「失敗まんだら（失敗原因の分類）」（付録１　図１－４参照）を利用する。

［個人に起因する原因］

①　無知…知識の不足・伝承の無視、等。

②　不注意…理解の不足・注意や用心の不足・疲労や体調不良、等。

③　手順の不遵守…連絡不足・手順の無視、等。

④　誤判断…狭い視野・誤った理解・間違った認知・状況に対する誤判断、等。

⑤　調査・検討の不足…仮想演習の不足・事前検討の不足・環境調査の不足、等。

［個人・組織のいずれにも起因しない原因］

⑥　環境変化への対応不良…使用環境の変化・経済環境の変化、等。

［組織に起因する原因］

⑦　企画不良…権利構築の不良・組織構成の不良・戦略や企画の不良、等。

⑧　価値観不良…異文化・組織文化の不良・安全意識の不良、等。

⑨　組織運営不良…運営の硬直化・管理の不良・構成員の不良等。

「誰にも起因しない原因」

⑩　未知…未知の事象が発生すること・異常事象が発生すること、等。

☞　**留意点**

・問題症状から選択して事象に書く

・やり方、そうなってしまった理由

・流出と作り込みのプロセスで分けてみる

表　3.1.1　なぜなぜ分析

なぜなぜ分析									
事象	要因					再発防止策			
	一次	二次	三次	四次	五次	対策内容	弊害	実行部門	実行時期
2台あるVVVFインバータのうちの1台が、誤動作により力行継続状態となった。	搭載されていない運転台モニタの伝送回路からの割り込みを有効な割り込みとして処理するプログラムになっていたので、不正割り込みが発生した。	運転台モニタが未接続のまま使用されると知らずに、未使用の端子を有効なままにした。	運転台モニタが未接続の状態で使用されるとは知らなかった。	VVVFインバータに対する要求仕様が不明確だった。		開発に着手する前に、要求定義を実施し、VVVFインバータに対する要求仕様を明確にする。		車両メーカー	
	不正割り込みにより、すべての割り込みが禁止され、加減速シーケンス処理が実行されなくなった	割り込み禁止としたときに、他の割り込みへ与える影響を検討できていなかった。				ソフトウェア詳細設計を行う前に、基本設計について十分検討する。		制御プログラム作成者	
						基本設計の妥当性を確認するために、基本設計書を作成後、必ずレビューを実施する。		制御プログラム作成者	
		不正割り込みが発生したときの振る舞いについて、十分に検証が行われていない。	正常系は設計時に十分に検討したが、異常系の検討が漏れていた。			「設計では、異常系も考慮する。」ことを設計基準書に追記する。		制御プログラム作成者	
			異常系に配慮したテストを行っていなかった。			「テストを実施する際には、正常系だけでなく、異常系も考慮したテストを実施する。」ことを設計基準書に追記する。		制御プログラム作成者	
		1ms 間隔のタイマ割り込みを禁止すれば、WDTの保護動作が働くと思っていた。	タイマ割り込みを禁止したときに、期待通りにWDTの保護動作が働くことをテストで確認していなかった。			フェイルセーフを実装した場合は、実装したフェイルセーフが期待通りに動作することを必ずテストで確認する。		制御プログラム作成者	
	プログラムがデッドロックしても、ウォッチドッグタイマによる保護動作が働かなかった	加減速シーケンス処理が実行されない場合でもWDTがかからないような処理フローとなっている。	加減速シーケンス処理起動カウンタ値の判定結果にかかわらず、必ず通るパス上にWDTリセット処理があった。	WDTの利用についてのノウハウが不足していた。		「WDTタイマの利用に関するノウハウ」を集め、設計基準に追記する		制御プログラム作成者	
						設計の妥当性を確認するために、必ず有識者にレビューをしてもらう。		制御プログラム作成者	
列車が当駅をオーバーランし、車両及び分岐器等の施設を壊して停止した。	運転手がマスコンの操作と関係なく力行し続けるという異常状態を知りながら、運行を継続した。	ノッチオフや非常ブレーキを操作すれば、問題なく停止できると誤った認識を持っていた。	非常ブレーキを使用したが、駆動力の方が勝っていたため、止まらなかった。			異常発生時の手順について、業務手順書を見直す。		鉄道事業者	
						ブレーキにのみ依存せずに、機械的に主回路を遮断するなど、確実に駆動力をゼロにする手段を用意する。		車両メーカー	
		運転手安全運行よりも定時運行を優先する意識があった。				車両運行に関わる社員の安全に対する意識を維持向上するために、定期的な安全教育を実施する。		鉄道事業者	

(3)　障害発生シナリオ

「原因」－「行動」－「結果」のキーワードをまとめて、障害を構造化する。

表　3.1.2　障害メカニズム分析

障害メカニズム分析								
問題の症状		問題行動		根本原因		対策案		
分類	内容	分類	内容	分類	内容	対策方法	対策者	備考
不良現象	VVVFインバータ内のゲート電源装置の高周波ノイズが、同装置の電源マイナス極側である低圧車体接地線に重畳した。	ハード制作	マイナス極と車体間の接地線の断面積が小さく、抵抗値が大きい					
		ハード制作	低圧車体接地線の配線の引き回しが長い					
不良現象	未使用のモニタ伝送回路がノイズの影響により、異常動作した	ハード制作	VVVFインバータ内の分圧抵抗盤で終端処理がされていなかった。					
機能不全	加減速シーケンス処理が実行されなくなり、力行継続状態となった	ソフト制作	搭載されていない運転台モニタの伝送回路からの割り込みを有効な割り込みとして処理するプログラムになっていたので、不正割り込みが発生した。	環境調査不足	運転台モニタが未接続のまま使用されると知らずに、未使用の端子を有効なままにした。	開発に着手する前に、システム要求定義プロセスを実施し、VVVFインバータに対する要求を明確にする。	VVVFインバータメーカー	組み込みソフトウェアの開発プロセスについては、ESPR Ver2.0を参照
		ソフト制作	不正割り込みにより、すべての割り込みが禁止されたため、加減速シーケンス処理起動カウンタの値が更新されなくなった。	調査・検討の不足	不正割り込みが発生したときの振る舞いについて、十分に検証が行われていない。	ソフトウェアを設計する際は、正常系を設計すると同時に異常系も併せて設計する。	制御プログラム作成者	ESDR p.22「A-2」を参照
				調査・検討の不足	すべての割り込みを禁止としたときに、他の割り込みへ与える影響を検討できていなかった。	ソフトウェアの詳細な設計を行う前に、アーキテクチャ（振る舞いと構造）について十分検討し、「アーキテクチャ設計書」として文書化する。アーキテクチャ設計の妥当性を確認するために、有識者を交えてレビューを実施する。	制御プログラム作成者	「ソフトウェア・アーキテクチャ設計」プロセスについては、ESPR Ver2.0 p.76 を参照。
				誤判断	タイマ割り込みを禁止すれば、WDTの保護動作が働くと誤認識していた。	「設計はできる限り単純化し、テストしやすさを考慮して設計する。」といったような、設計基準を設ける。	制御プログラム作成者	ESDR p.28「A-6」を参照。
				手順の不遵守	タイマ割り込みを禁止したときに、期待通りにWDTの保護動作が働くことをテストで確認していなかった。	フェイルセーフを実装した場合は、実装したフェイルセーフが期待通りに動作することを必ずテストで確認する。	制御プログラム作成者	
				手順の不遵守	ソフトウェアに対する安全要求が不明確だった。	ソフトウェア開発に着手する前に、安全要求定義プロセスを実施し、ソフトウェアの機能に対する要求を明確にする。	制御プログラム作成者	「安全要求定義」プロセスについては、ESPR Ver2.0 p.132 を参照。
				価値観不良	加減速制御プログラムが「システムの安全にかかわる重要な要素である」という認識が欠けていた。	機能安全の「セーフティインテグリティレベル」の考え方を取り入れ、システムの安全性に関わる部品かどうかを判断する仕組みを構築する。	VVVFインバータメーカー 車両メーカー	機能安全については、IEC61508などの国際規格を参照。
機能不全	ウォッチドッグタイマによる保護動作が働かなかった	ソフト制作	加減速シーケンス処理が実行されない場合でもWDTがかからないような処理フローとなっている。	不注意	必ず通るパス上にWDTリセット処理があった。	WDTタイマの利用に関するノウハウを設計基準やノウハウ集といった形で文書化し、形式知化する。	制御プログラム作成者	
						機能ユニットの詳細な振る舞いを設計書として文書化し、有識者を交えてレビューを行う。	制御プログラム作成者	「ソフトウェア詳細設計」プロセスについては、ESPR Ver2.0 p.89 を参照。
破損	列車の車両及び分岐器等の施設に物損	非定常行為	マスコンを操作しても加速し続けるという異常状態を知りながら、運行を継続した。	誤判断	ノッチオフや非常ブレーキを操作すれば、問題なく停止できると誤った認識を持っていた。	異常発生時の手順について、業務手順書を見直す。	モノレール会社	
				価値観不良	運転手に定時運行を安全運行よりも優先する意識があった。	車両運行に関わる社員の安全に対する意識を維持向上するために、定期的な安全教育を実施する。	モノレール会社	
				価値観不良	会社全体として、安全を最優先と考える意識が低かった。	安全第一とする経営方針を明確に打ち出す。	モノレール会社の経営層	
		ハード製作	非常ブレーキを使用したが、駆動力の方が勝っていたため、止まらなかった。	誤判断	車両システムの設計時点で、複数のブレーキ系統があるので、十分停止できると誤判断した。	ブレーキにのみ依存せずに、主電源をカットするなど、手動で駆動力をゼロにする手段を用意する。	車両メーカー	
				調査・検討の不足	システムの構成要素の一部の機能が欠落した場合、システム全体としてどのような障害が発生するかを検証していなかった。	HAZOPやFMEAなどの手法を用いて、システムに内在するハザードを分析し、システムの安全性を確保する手段を検討する。	車両メーカー	
破損	すべてのブレーキディスクに亀裂が発生した。	非定常行為 無為	1次車のものと比較すると同じブレーキをかけた場合に発生する応力が大きく、ブレーキディスク自体の引張り強度が小さいので、熱疲労によるき裂が発生しやすい材料であったものと推定される。	誤判断	2次車で使用した水平連続鋳造材の方が組織が緻密で均一な硬さ分布を有した優れた材料と判断したため			
				調査・検討の不足	材料の特性の変化がブレーキディスクの機能・性能に及ぼす影響を十分に検討しなかった			
起こり得る被害	ブレーキディスクとして必要な強度を有していない可能性がある。							

3.1.4 対策の検討とまとめ

(1) 対策

再発防止策
非常ブレーキを作動させても減速感がない場合には「レバースハンドル『切』」とすることを指導し、電車運転士作業基準を改訂し、同内容を「非常の場合の処置」に加えた。
VVVFインバータの加減速制御プログラムに、非常ブレーキが投入された場合に主回路をしゃ断する処理を追加した。
VVVFインバータの加減速制御プログラムのWDTリセット処理のタイミングを、加減速シーケンス処理の直後に変更し、加減速シーケンスが停止した場合はWDTにより主回路の停止とVVVFインバータの再起動が確実に行われるようにした。

図　3.1.8　再発防止策

☞　**留意点**

・エンジニアに教育しているということではなく、設計プロセスへ反映するようにしている

・設計への反映としては「○○する時は△△すること」というように表現しているが
　○○の部分をどのように表現するかは悩みどころ

・設計は ESDR（参考文献[3]）、プロセスは ESPR（参考文献[4]）が参考になる

(2) 教訓

問題症状、障害メカニズム分析は時系列に並べることを意識するとモレ・ヌケなく考えやすい。以下に例を示す。

教訓
制御プログラム作成者
割り込みを使用するシステムにおいて割り込み禁止処理を行う場合、有効としているすべての割り込みへの影響を検討する。この検討は設計工程で実施し、検討結果は文書化した上でレビューを行い、エビデンスとして残す。
設計では、異常系も考慮する。
テストを実施する際には、正常系だけでなく、異常系も考慮したテストを実施する。
フェイルセーフを実装した場合は、実装したフェイルセーフが期待通りに動作することを必ずテストで確認する。
運転手
異常な力行を認識した時点で安全を最優先し、すぐに停車すべきだった。
車両メーカー
運転手が異常を検知したときに、確実に駆動力を0にできる仕組み（例えば、機械的に主回路を遮断するなど）を車両に備えておくべきだった。
鉄道事業者
「異常な力行動作」の発生を想定した非常時対応策を策定し、運転手に周知徹底しておくべきだった。

図　3.1.9　教訓

3.2 未来都市モノレール障害のなぜなぜ分析事例

「未来都市モノレール」障害事例について「なぜなぜ分析」手法を使い、2.2 項の図 2「障害分析の流れ」の中の原因分析タスクで行われる直接原因から根本原因分析の具体的流れと考え方を解説する。障害内容については、3.1 を参照。

次の順番で分析の手順を説明する。

3.2.1　なぜなぜ分析結果

3.2.2　なぜなぜ分析の解説

3.2.1 なぜなぜ分析結果

図 3.2.1 になぜなぜ分析の結果を示す。

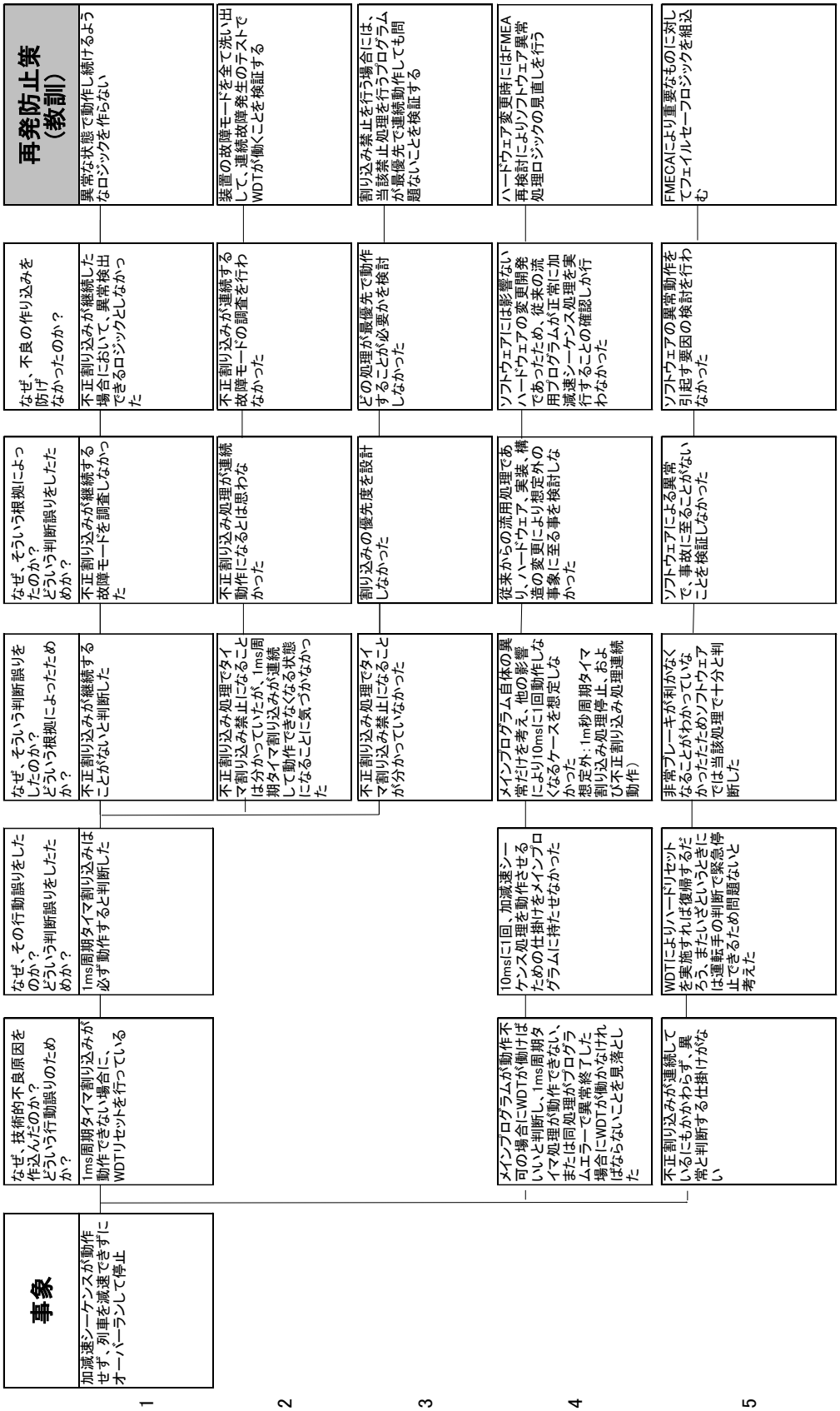

事象
再発防止策
(教訓)
A
B
C
D
E
1
2
3
4
5
加減速シーケンスが動作せず、列車を減速できずにオーバーランして停止
なぜ、技術的不良原因を作込んだのか？
どういう行動誤りのためか？
なぜ、その行動誤りをしたのか？
どういう判断誤りをしたためか？
なぜ、そういう判断誤りをしたのか？
どういう根拠によったため？
なぜ、そういう根拠によったのか？
どういう判断誤りをしたためか？
なぜ、不良の作り込みを防げなかったのか？
異常な状態で動作し続けるようなロジックを作らない
1ms周期タイマ割り込みが動作できない場合に、WDTリセットを行っている
1ms周期タイマ割り込みは必ず動作すると判断した
不正割り込みが継続することがないと判断した
不正割り込み処理が連続故障モードを調査しなかった
不正割り込みが継続する故障モードの調査を行わなかった
不正割り込みが継続しているにもかかわらず、異常と判断する仕掛けがない
WDTによりハードリセットを実施すれば復帰するだろう、またいざというときには運転手の判断で緊急停止できるため問題ないと考えた
非常ブレーキが利かなくなることがわかっていなかったためソフトウェアでは当該処理で十分と判断した
ソフトウェアによる異常で、事故に至ることがないことを検証しなかった
ソフトウェアの異常動作を引起す要因の検討を行わなかった
メインプログラムが動作不可の場合にWDTが働けばいいと判断し、1ms周期タイマ処理が動作できない、または同処理がプログラムエラーで異常終了した場合にWDTが働かなければならないことを見落とした
10msに1回、加減速シーケンス処理を動作させるための仕掛けをメインプログラムに持たせなかった
メインプログラム自体の異常だけを考え、他の影響により10msに1回動作しなくなるケースを想定しなかった
想定外：1m秒周期タイマ割り込み処理停止、および不正割り込み処理連続動作）
不正割り込み処理でタイマ割り込み禁止になることが分かっていなかった
不正割り込み処理でタイマ割り込み禁止になることは分かっていたが、1ms周期タイマ割り込みが連続して動作できなくなる状態になることに気づかなかった
割り込みの優先度を設計しなかった
従来からの流用処理であり、ハードウェア、実装、構造の変更により想定外の事象に至る事を検討しなかった
どの処理が最優先で動作することが必要かを検討しなかった
ソフトウェアには影響ないハードウェアの変更開発であったため、従来の流用プログラムが正常に加減速シーケンス処理を実行することの確認しか行わなかった
不正割り込み時に、WDTが働くことを検証する
装置の故障モードを全て洗い出して、連続故障発生のテストでWDTが働くことを検証する
割り込み禁止を行う場合には、当該禁止処理を行うプログラムが最優先で連続動作しても問題ないことを検証する
ハードウェア変更時にはFMEA再検討によりソフトウェア異常処理ロジックの見直しを行う
FMECAにより重要なものに対してフェイルセーフロジックを組込む

図　3.2.1　なぜなぜ分析

3.2.2 なぜなぜ分析の解説

以下になぜなぜ分析表の各ボックスの記述内容となぜそう記述するかの根拠（考え方）を分析の流れ（なぜなぜ分析の深さ方向）に沿って解説する。さらに到達した根本原因について、再発防止策（教訓）とその根拠を示した。

1-A

記述内容：

「1ms 周期タイマ割り込みが動作できない場合に、ウォッチドッグタイマ（WDT）リセットを行っている」

根拠：

異常が発生しているにもかかわらず WDT リセット出力処理を実行すること自体考えが間違っている。（図 3.1.4 参照）

1-B

記述内容：

「1ms 周期タイマ割り込みは必ず動作すると判断した」

根拠：

タイマ割り込みで起動され、カウンタに+1 するだけの処理と言うこともあり、この処理が動作しなくなる考えにいたらなかったのではないかと判断した。

1-C

記述内容：

「不正割り込みが継続することがないと判断した」

根拠：

なぜ 1ms 周期タイマ割り込みが必ず動作すると考えてしまったのかを記入するところだが、ほとんど検討していなかったのではないかと推測した。

1-再発防止策

記述内容：

「異常な状態で動作し続けるようなロジックを作らない」

根拠：

異常ルートでインループしてしまう処理になっていることを見逃している。本来はどうすればこのようなロジックを作らないようにできるかが再発防止であるが、とりあえずは教訓とした。しかし、この動作自体をおかしいとおもわなければ、ソフトロジックについてあるべき教育が必要。

2-C

記述内容：

「不正割込み処理でタイマ割り込み禁止になることは分かっていたが、1ms 周期タイマ割り込みが連続して動作できなくなる状態になることに気づかなかった」

根拠：

ソフトのロジックは分かっていたが、ハード故障に起因しての検討が不足した（ものと推測）ことの

追求。FMEA 検討不足

2-D

記述内容：

「不正割込み処理が連続動作になるとは思わなかった」

根拠：

ハードが故障すれば、プログラム全体が動作しなくなり WDT が働くため問題ないと判断して深く検討しなかった。

2-E

記述内容：

「不正割り込みが連続する故障モードの調査を行わなかった」

根拠：

①タイマ割り込みが入らなくなった場合

②タイマ割り込みプログラムがカウンタ+1 する前にエラー終了等も同じ事象となるが装置の故障モードによる動作の洗い出しが出来ていない

2-再発防止策

記述内容：

「装置の故障モードを全て洗い出して、連続故障発生のテストで WDT が働くことを検証する」

根拠：

初期開発時に FMEA による異常試験の網羅性検証と試験の実施をすることが必要。

＜不正割込み発生故障モードの想定＞

（1）連続／間欠／単発

（2）10ms（以上／以下）連続後に復旧を繰り返す

3-C

記述内容：

「不正割込み処理でタイマ割り込み禁止になることが分かっていなかった」

根拠：

ソフトのロジックが分かっていなかった、このような設計が必要であることを知らない。

優先度を考えないで設計すると、思わぬ動作になることがある。構造レビュー、試験評価が必要。

①複数プログラムが非同期動作する場合

②シングルプロセッサ／マルチプロセッサによっても作りが変わってくる

4-A

記述内容：

「メインプログラムが動作不可の場合に WDT が働けばいいと判断し、1ms 周期タイマ処理が動作できない、または同処理がプログラムエラーで異常終了した場合に WDT が働かなければならないことを見落とした」

根拠：

この行は、4-D/E についての再発防止について追求すべきと考えた。これはありがちなミスで、従来のハード構成では起き得なかったことが、部品や構造変更により思わぬ故障モードが起きやすくなってしまうことの事例も少なくない。

4-B

記述内容：

「10ms に 1 回、加減速シーケンス処理を動作させるための仕掛けをメインプログラムに持たせなかった」

根拠：

他ソフト処理で周期を作らせている。

カウンタの更新はハードロジックで実施すべき。

5-A

記述内容：

「不正割込みが連続しているにもかかわらず、異常と判断する仕掛けがない」

根拠：

この VVVF 内プログラムの重点は、10ms に 1 回加減速シーケンスを動作させなければならないことにある。普段動作してはならない不正割込みが 1 日 1 回とかであればありえる動作（想定内）としても、連続動作することは異常事態という認識で処理設計すべき。

5-B

記述内容：

「WDT によりハードリセットを実施すれば復帰するだろう、またいざというときには運転手の判断で緊急停止できるため問題ないと考えた」

根拠：

考えが浅く検討していないために WDT が毎回リセットされる動作の存在や最悪は緊急停止が絶対動作するから大丈夫と安易に考えた

5-C

記述内容：

「非常ブレーキが利かなくなることがわかっていなかったためソフトウェアでは当該処理で十分と判断した」

根拠：

ソフトは、WDT での復旧さえあれば十分と思った。

5-D

記述内容：

「ソフトウェアによる異常で、事故に至ることがないことを検証しなかった」

根拠：

ソフトウェアによる異常

①プログラムエラー

②インループ

③処理遅延

5-再発防止策

記述内容：

「FMECA により重要なものに対してフェイルセーフロジックを組込む」

根拠：

マイクロプログラムやソフトウェアもシステムの構成要素の 1 つとして考え、システムの重要品であれば、異常時のフェイルセーフロジックを組込むことが必要。

3.3 堰堤洪水吐ゲート異常作動の分析事例 1

「堰堤洪水吐ゲート異常作動」障害事例について、VTA 手法（Variation Tree Analysis : 変動木分析）（付録 2 参照）を使って、2.2 項の障害発生から再発防止策の立案までの流れ（手順）に沿って分析の考え方、結果を解説する。

障害事例は架空のものであるが、実際に発生した障害を参考に、開発体制や根本原因等を創作している。（参考文献[5]、[6]、[7]、[8]）

次の順番で分析の手順を説明する。

3.3.1　事実整理

3.3.2　分析

3.3.3　まとめ

3.3.1 事実整理

(1) 事故概要の把握

事例報告書から、事故の概要を把握する。

- 発生日：200x 年 4 月 1 日
- 発生場所：千石ダム（堰堤）
- 障害内容：
  文京川水系千石川にある千石ダム（堰堤）で、ダム水位一定制御（ゲート自動操作システム）の制御プログラムの欠陥により、総量 20,000m³ の貯水が異常放流された。人命被害はなし。

(2) 製品概要の把握

事故報告書から、製品（ゲート自動操作システム）の概要を把握する。報告書のみでは製品の概要が把握しにくい場合は、必要に応じて関連情報を収集する。本分析では、ダム管理用制御処理設備のシステム構成等の把握のため、参考文献　[9]を参照した。

また、障害を概観できるように、製品を含むシステム全体の概要図を作成する。

【製品の概要】

① 千石堰堤は、最大出力 5,600kw の発電を目的に文京川水系千石川に建設した取水堰であり、昭和 xx 年 10 月から運転を開始している。放流設備は洪水吐ゲート 2 門、流量調整ゲート 1 門を有している。

② 千石堰堤は、計画放流のため、約 36km 下流にある駒込ダム管理所のゲート自動操作システムにより遠隔操作でダム水位一定制御が行われている。

③ 駒込ダム管理所の操作員が、ゲート自動操作システムを使用して、ダム水位一定制御を行う。

④ ダム水位一定制御とは、目標水位の入力設定とその時刻を設定することによって千石堰堤

の洪水吐ゲートの開度制御を行うことである。

⑤ ダム水位一定制御では、1 回の設定により 4 時間後まで、設定された操作間隔時間ごとに実績の流入量・放流量・水位データを更新し設定水位と比較され、ゲートの開度が計算される。このとき、開度変更が必要であれば千石堰堤へ開度変更指示が送信されゲート操作が行われる。

⑥ ダム水位一定制御では、設定から 4 時間が経過すると、変更操作しなければ、直前の設定を使用して、水位一定制御を継続する。

⑦ ダム水位一定制御では、開いている洪水吐ゲートの停止には「スケジュールキャンセル」スイッチを押す。「割込操作」スイッチを押下してもゲートは停止しない。

⑧ 駒込ダム管理所から千石堰堤を「緊急停止」させると、千石堰堤の電源が止まる。

⑨ 洪水吐ゲートは、1 回の作動時間を超過すると停止する。

⑩ 警報サイレンは、一斉吹鳴開始すると、2 分後に一斉吹鳴を停止する。

⑪ ゲート自動操作システムの遠隔操作制御プログラムは、事故の 5 年前に納入されている。

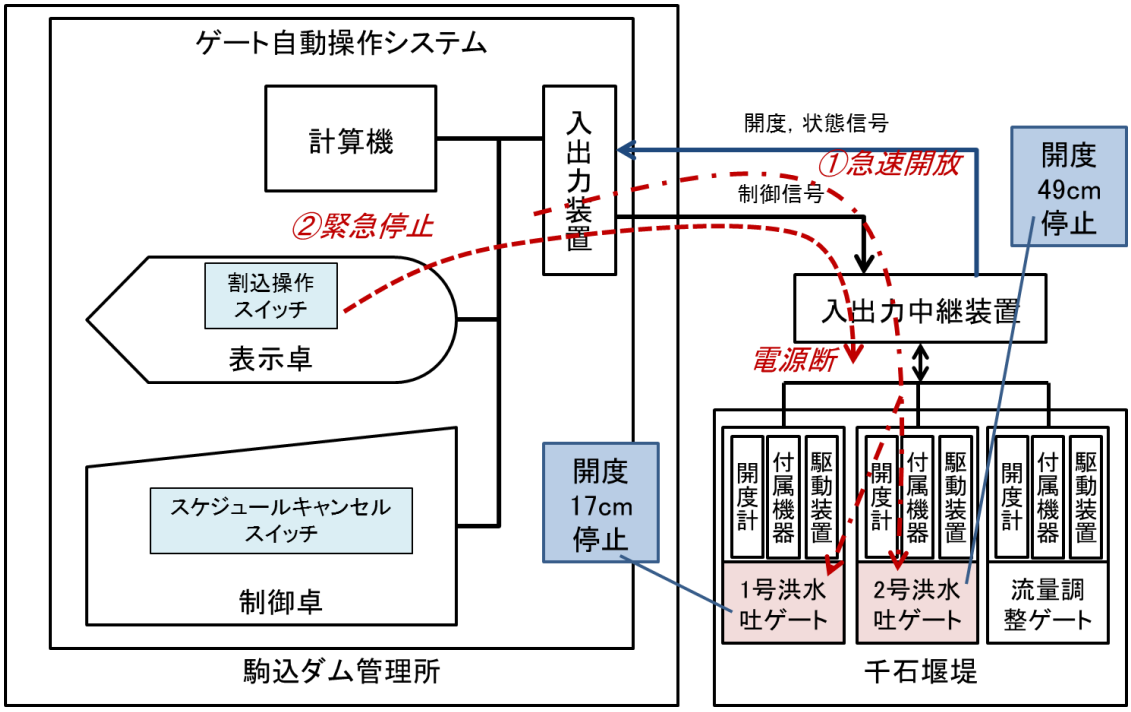


図 3.3.1　システム全体概要図

(3) 発生事象の整理

事故報告書から「何が起きたか」「誰が何をしたか」を漏れなく拾い出す。本分析では下記①ごとに②、③をグループ化した。事象を整理する際に一般的に「誰」「何」に注目する。

① 「誰」「何」

関係者	駒込ダム管理所	操作員
設備	千石堰堤	洪水吐 1 号ゲート、2 号ゲート、流量調整ゲート、警報サイレン
	駒込ダム管理所	ゲート自動操作システム（計算機：制御プログラム、表示卓：割込操作スイッチ、制御卓：スケジュールキャンセルスイッチ）

② 「いつ」

③ 「何をした」「どうなった」

200x/04/01 14:05	駒込ダム管理所・操作員	水位一定制御を水位 9.9m に設定
200x/04/01 14:05	駒込ダム管理所・ゲート自動操作システム	目標水位を 9.9m に設定
200x/04/01 14:10	駒込ダム管理所・操作員	設定水位を確認後、「変更取消」操作、「計画起動」操作実施
200x/04/01 18:05	駒込ダム管理所・ゲート自動操作システム	目標水位を初期値に設定し、開度変更指示を千石堰堤に送信
200x/04/01 18:05	千石堰堤 ・洪水吐 1 号、2 号ゲート	開放開始（急速開放）
200x/04/01 18:06	千石堰堤・洪水吐 2 号ゲート	開度 49cm で停止（制限タイマ）
200x/04/01 18:06	駒込ダム管理所・操作員	「割込操作」スイッチ押下
200x/04/01 18:07	千石堰堤～三穂発電所間・警報サイレン	一斉吹鳴開始
200x/04/01 18:07	駒込ダム管理所・操作員	「緊急停止」（千石堰堤の電源断）
200x/04/01 18:07	千石堰堤・洪水吐 1 号ゲート	開度 17cm で停止（電源断）
200x/04/01 18:08	千石堰堤	最大放流量約 67 ㎥/s が流出
200x/04/01 18:09	千石堰堤～本郷発電所間・警報サイレン	一斉吹鳴完了

3.3.2 分析

(1) 分析手法の選択

上記 3.3.1 で整理した情報を元に分析する。分析手法としては、次のような理由により、VTA（Variation Tree Analysis：変動木分析）（付録 2 参照）を選択した。

- 該当事象は、複数の要因が相互に関連しており、さらに時間的経過と要因間の関連が重要と思われるため。
- 分析手法として比較的理解が用意で、分析に対する高度な専門知識を必要としないため。

(2) VTA を用いた分析

① 上記 3.3.1（3） ①（「誰」「何」。以下、関係要素）を M-SHEL モデル*の要素と比較し、それぞれ何に対応しているかを確認する。確認した結果、M-SHEL の他の要素がないか、等について、報告書を再度確認する。もし、他の関係要素が見つかった場合は、該当の関係要素についても、3.3.1（3） ②（「いつ」）と③（「何をした」「どうなった」。以下、事象）を報告書から拾い出す。

＊M-SHEL モデル：

機械やシステムを安全に、有効に機能させるために必要とされる人間の能力や限界、特性等のヒューマンファクターを表現するためのモデル。元は国際航空運送協会（IATA）で提案され、現在では幾つかの類似モデルがある。

M：マネジメント、S：ソフトウェア、H：ハードウェア、E：環境、L：人間（相手、関係者、第三者、等）、中央の L：人間（当人、当事者、本人、自分、等）

図 3.3.2　M-SHEL　モデル　（出典：日本ヒューマンファクター研究所、参考文献 [10]）

【M-SHEL モデルの要素との対応】

- 中央の L（当事者）：操作員が該当する。
- H（機器、機材、設備、等）：
  洪水吐 1 号ゲート、2 号ゲート、流量調整ゲート、警報サイレン、ゲート自動操作システムが該当する。ただし、流量調整ゲートについては、事故に与えた影響について報告書から読み取ることができないため、除外する。
- M（コミットメント、体制、分担、リスク管理、等）：
  操作員に対する教育カリキュラムが該当する。ただし、その内容、及び、経緯について調査報告書から読み取ることができないため、除外する。
- S（規程、規則、細則、要領、等）：
  運転操作マニュアルが該当する。ただし、その内容、及び、経緯について報告書から読み取ることができないため、除外する。
- E（気温、湿度、換気、騒音、照明、空間、遠近、利便、安全文化、風土、慣習、等）：
  千石堰堤と駒込管理ダム管理所との距離（約 36km）が該当する。ただし、経緯について報告書から読み取ることができないため、除外する。

- L（相手、関係者、第三者）：
  該当なし。

② 上記①で決定した関係要素を横軸に配置する。

③ 縦軸を時間軸とし、関係要素ごとに事象をノードとして時間経過順に並べる。ノードの文体は俯瞰しやすいように、簡潔に、客観的に記述する。1つのノードには1つの事象のみを記述するようにする。ノード相互に因果関係があれば、矢印で結ぶ。

④ 排除ノード（事故に直接結びつく行動や状態）と、変動ノード（通常から逸脱した行動や状態）を特定する。特定するには、本来なら運転操作マニュアル等を確認し、通常の行動や状態を把握する必要があるが、本分析では報告書から想定した。

⑤ 最後に全体を見直し、事故が妥当に表現できているかを確認する。このとき対象者（本例では操作員）の視点で確認する。

【VTA】

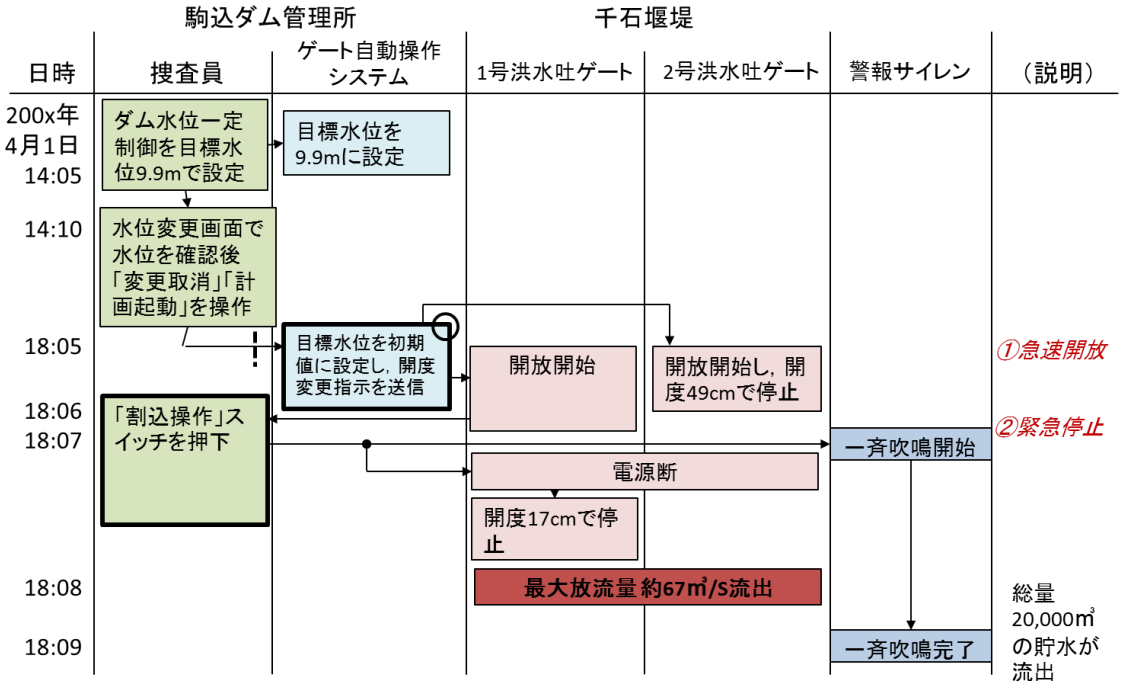


図 3.3.3　VTA による分析図

⑥ VTA の変動ノード（通常から逸脱した行動や状態）と排除ノード（事故等に直接結びつく行動や状態）から、原因を特定する。主体（誰）別に色分けすると見やすくなる。

【原因】

A) ゲート自動操作システム（制御プログラム）の欠陥
洪水吐ゲートを開度制御するための目標水位のプログラム設定値誤り（水位変更キャンセル直前に設定した値になるべきところ、初期値〔極端に低い水位〕になった）のた

め、仕様の規定時間（4 時間）後に洪水吐ゲートが開いた（急速開放）。

B) 同システムの操作員の操作誤り

上記 B）の異常に気づいた操作員が、ゲート停止に「スケジュールキャンセル」スイッチを押すべきところ、「割込操作」スイッチを押したため、ゲートが停止しなかった。

(3) なぜなぜ分析を用いた分析

VTA により得られた原因に対して、さらに分析することで、深く背後要因を追及し、真因を特定する。原因をさらに深く掘り下げるため、分析手法としては、なぜなぜ分析を選択した。

分析の方針として、不具合を作り込んだ設計の再発防止、広く運用も含めたシステムとしての事故の再発防止、等が考えられる。本分析では、原因としてプログラムの欠陥（原因 A）が特定されたことから、前者の視点でなぜなぜ分析を実施することが望ましいが、前者について分析するには情報が不足しているため、操作員のヒューマンエラー防止を対象としたフールプルーフの視点で分析した。真因に達すると色を変える。

【なぜなぜ分析】

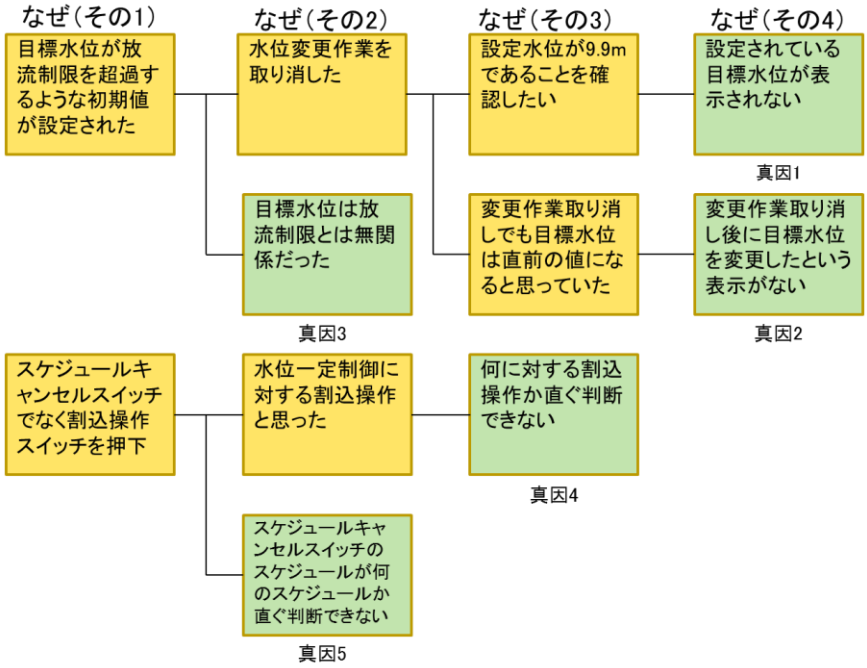


図 3.3.4　なぜなぜ分析

3.3.3 まとめ

(1) 真因に対する対策の立案

なぜなぜ分析により特定した真因を解決するための対策を立案する。立案した対策は、8つの観点（普及性、整合性、具体性、実行性、的中性、確実性、経済性、永続性）で評価することが望ましい。

【対策】

① 目標水位の設定なし、ならびに水位変更キャンセル操作時に設定される目標水位の確認をプログラムに追加（真因 1、2）
上記のケースで、4 時間後に設定される目標水位を表示し、操作員に確認し承認後、該当水位を 4 時間経過したときの目標水位とする。

② ゲート開度変更指示を送信する場合に、目標水位による放流量確認機能をプログラムに追加（真因 3）
目標水位による放流量が堰堤下流の放流制限を超える場合は、ゲート開度変更指示を堰堤に送信しない。

③ スイッチ名称及び機能の適正化（真因 4、5）
最終的な目的に直接結びつくスイッチ名称とする。また、緊急時に使用するスイッチは単一の機能のみとする。
例：「スケジュールキャンセル」→「ゲート停止」

④ 操作教育の強化（真因 4、5）
シミュレータにより過去の事故事例や異常時操作を訓練する。

(2) 教訓の作成

分析をとおして得られた原因や対策を一般化・抽象化し、未然防止に繋がるよう教訓を作成する。本分析では、操作員とシステムが頻繁にインタラクションを取りながら、所定の目的を達成するようなシステムに対する、システム設計を対象とした。

【教訓】

① 操作員は、システムが設定する設定値をすべて正しく把握している訳ではない。

② 緊急時に利用する設備は、「使い方を説明しなくとも、ユーザーが適切に使ってくれる」アフォーダンス（Affordance）を重視して設計する。

3.4 堰堤洪水吐ゲート異常作動の分析事例 2

「千石ダム」障害事例について「問題行動分析」手法を使った分析の考え方、結果に焦点を当てて解説する。従って、障害発生から分析結果までの流れ（手順）の 3.4.2、は、終わった後の 3.4.3 から 3.4.5 に沿って解説する。

次の順番で分析の手順を説明する。

3.4.1　「問題行動分析」手法
3.4.2　障害概要の把握
3.4.3　問題症状の把握（事象経過）
3.4.4　原因分析
3.4.5　対策の検討

3.4.1 「問題行動分析」手法

(1) 手法の概要

問題行動分析では、整理された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。分析の結果は根本原因を推定する際に利用するなぜなぜ分析の起点となる。

(2) 記法ならびに分析法

事故経過表をもとに、各事象を引き起こした可能性として考えられるシステム操作者や開発者の行動を検討して問題行動/内容欄に記載する。また記載した問題行動/内容について、
運用時の操作と開発時の作業に分類する。

3.4.2 障害概要の把握

この事例は、200x 年 4 月 1 日に文京川水系千石川にある千石ダムにおいて発生した異常放流である。千石ダムを管理する駒込ダム管理所において同ダムの水位変更操作を行ったところ、意図せずに水門が開放された。（図 3.4.1）

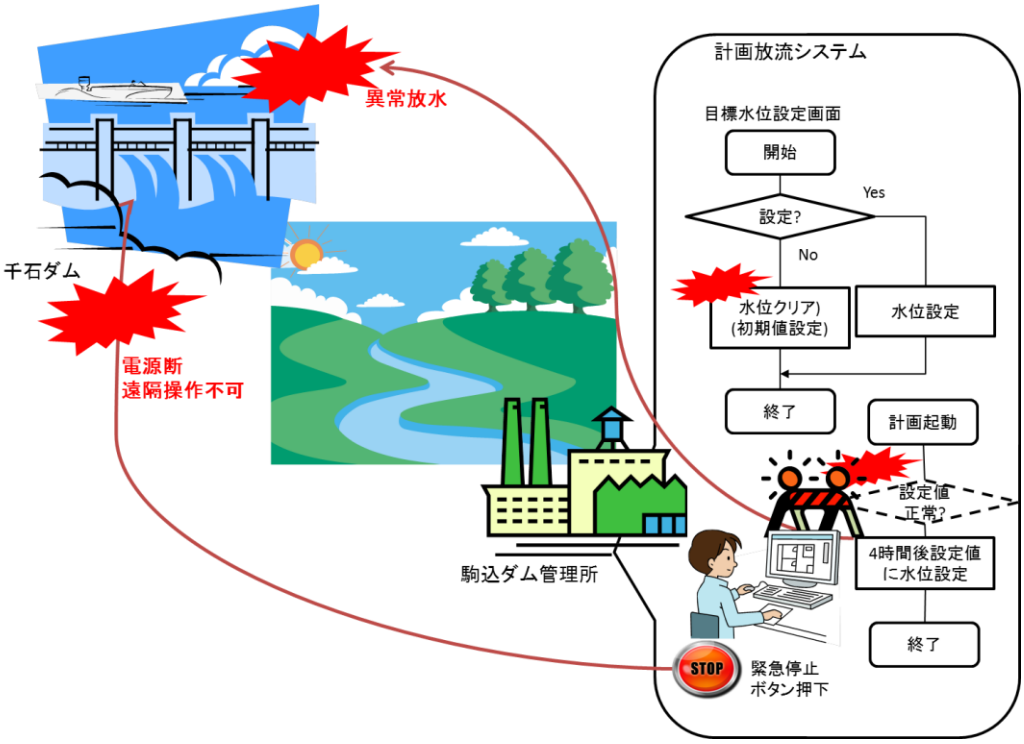


図 3.4.1 事故状況概要図

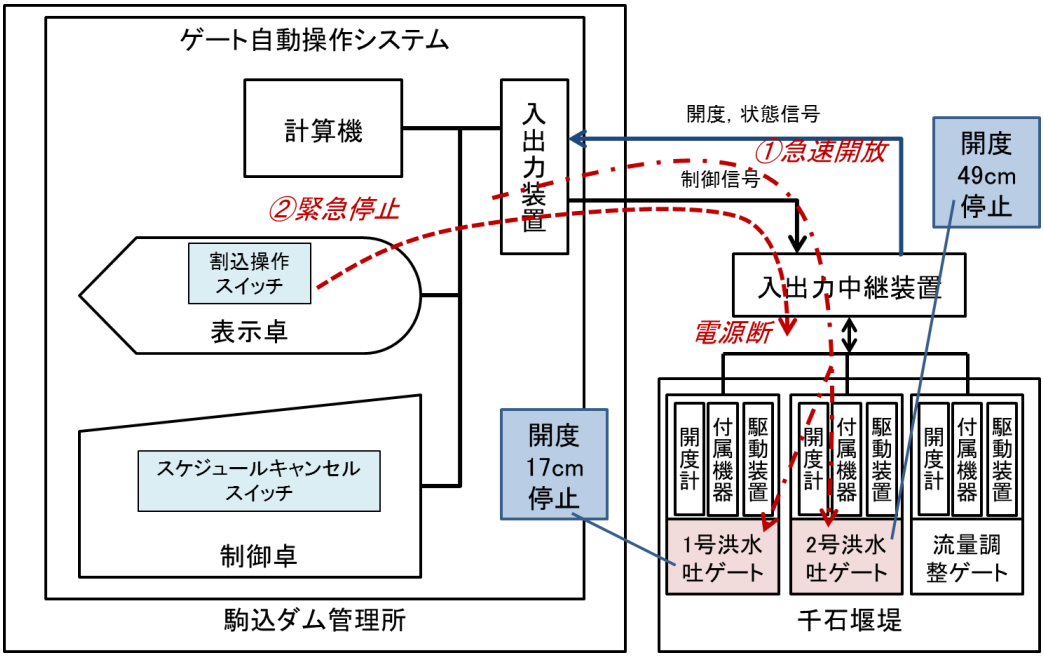


図 3.4.2　システム全体概要図

ダムの水位を確認するためには、水位変更操作と同じ操作を変更直前まで行い、変更取り消し操作を行う必要があった。同手順を行った後に同ダムの目標水位が意図せず初期値に変更された結果、水門が開放され、異常放流につながった。さらに、異常に気づいた操作員が誤ってダムの電源を遮断したため、結果的に総量約 2 万立方メートルの貯水が放流された。（図 3.4.1）

3.4.3 問題症状の把握（事象経過）

(1) 事故経過表の作成

事故経過表の時刻欄、発生した問題内容欄、内容の分類欄　に、システム状況の変化を一つの項目として問題内容等を記載する。分類欄については、問題発生の端緒となった不良事象、その結果としてシステムに発生した機能不全、外部から観測可能となる症状等を代表的な分類として表記する。機器の状態に影響を及ぼしたソフトウェア設計、実装における直接的な要因（不具合原因）をリストアップする。この例では、システムの構成要素は区別せず、時系列に沿って配置した。さらに、事象の分類を示す項目を追加した。分類には、「失敗まんだら（失敗結果の分類）」（付録１　図１－２参照）に挙げられているキーフレーズを利用した。このように分類することが障害の原因を推定する段階で役立つ。

表　3.4.1　事故経過表

	問題の症状/経過	
時刻	分類	内容
14:05		目標ダム水位9.9mlに設定
14:10	不良現象	水位変更画面上で設定水位を確認後、「変更取消」操作を行い計画起動 →**水位設定値が初期値になってしまっていた**
18:05	不良現象	吐水ゲート作動
18:06		2号ゲートが開度49cmで停止
18:06		1号ゲート停止をこころみるが停止せず
18:07		「スケジュールキャンセル」スイッチにより停止すべきところ、「緊急停止」により作動停止
		「緊急停止」したことにより、現地の電源が止められ遠隔操作不可となる

直接根本原因につながりそうなものは赤で示す。

(2) 問題行動の列挙

前記事故経過表（表　3.4.1）に対応して、各事象を引き起こした可能性として考えられるシステム操作者や開発者の行動を検討して問題行動/内容欄に記載する。また記載した問題行動/内容について、運用時の操作と開発時の作業に分類する。ここでは、問題症状と同様に問題行動にも分類を示す項目を追加する。この項目は「失敗まんだら（失敗行動の分類）」（付録１　図１−３参照）を参考にしているが、必ずしもそれにとらわれない分類項目を設けた。症状がシステムの意図しない動作である場合、開発上の問題行動（行動誤り）として記述する。

表　3.4.2　問題行動

	問題の症状/経過		問題行動	
時刻	分類	内容	分類	内容
14:05		目標ダム水位9.9mに設定		
14:10	不良現象	水位変更画面上で設定水位を確認後、「変更取消」操作を行い計画起動 →**水位設定値が初期値になってしまっていた**	潜在危険	確認のために設定画面を開いた？
			ソフト制作	設定状況を確認できない画面になっている
			運転・使用	最終的な設定値の確認をしていない
18:05	不良現象	吐水ゲート作動	ソフト制作	設定状況を確認できない画面になっている
18:06		2号ゲートが開度49cmで停止	ソフト制作	ゲート停止する仕様とソフト仕様の不一致
18:06		1号ゲート停止をこころみるが停止せず	非定常操作	停止のために「割込操作」スイッチを押した
18:07		「スケジュールキャンセル」スイッチにより停止すべきところ、「緊急停止」により作動停止	非定常操作	電源断になることをしらず「緊急停止」を押してしまった
		「緊急停止」したことにより、現地の電源が止められ遠隔操作不可となる	非定常操作	管理所なのに遠隔操作不可となる

3.4.4 原因分析

前記問題行動表（表　3.4.2）の各項目に対してなぜなぜ分析を行い、結果として得られた根本原因を以下の表（表　3.4.3）に記載する。設計、実装における根本原因を分析し、特にインフラ設備として安全設計、フェイルセーフという観点でどういう問題がありそうかをリストアップする。

表　3.4.3　問題行動・根本原因対応表

問題行動		根本原因		疑問点
分類	内容	分類	内容	
潜在危険	確認のために設定画面を開いた？		**オペミスの可能性あり**	**確認画面ない？**
ソフト制作	設定状況を確認できない画面になっている	調査検討の不足	設定画面でないと確認できない	
		調査検討の不足	具体的なオペレーションを理解しないまま画面設計している	
		手順の不順守？	取り消し操作の場合は以前の設定値を使うようになっていたが初期値にしていた	仕様確認していない？設定しないまま初期値が使われるようになっているのか不明
運転・使用	最終的な設定値の確認をしていない	手順の不順守	確認手順の不備	
		価値観不良	要求事項がきちんとできているか確認できていない	
ソフト制作	設定状況を確認できない画面になっている	調査検討の不足	**実際に動作するまで異常な設定値がなされている状況が確認できない**	
ソフト制作	ゲート停止する仕様とソフト仕様の不一致	調査検討の不足	**作動時間を制限するタイマーで停止するような設定値でもチェックする機能がない**	
		価値観不良	作動時間を制限するような要求仕様がきちんと開発者に伝わっていない	
非定常操作	停止のために「割込操作」スイッチを押した	手順の不順守	ゲート停止の手順が理解できていない	
			スイッチの名前がオペレーションとマッチしていない	
非定常操作	電源断になることをしらず「緊急停止」を押してしまった	手順の不順守	ゲート停止の手順が理解できていない	
			スイッチの名前がオペレーションとマッチしていない	
非定常操作	管理所なのに遠隔操作不可となる	調査検討の不足	**最終手段でゲート電源断で止めるのはOKとしても遠隔操作不可は問題**	

明らかな直接原因は、

・ダム水位一定制御プログラムの欠陥

・操作員の操作誤り

であるが、ここでは全ての問題行動に対して根本原因の分析を行った。さらに可能性のある原因も追記

し、疑問点として改めて調査し解決策まで書く。

根本原因をなぜなぜ分析を使って分析する。一部を次表（表　3.4.4）に示す。

表　3.4.4　なぜなぜ分析

直接原因										
プログラムのミスにより水位設定がクリアされた	なぜ? →	変更画面で設定水位確認後取り消し	なぜ? →	水位設定を確認した	なぜ? →	**確認画面がない:設計不備**	なぜ? →	運用方法を考えない操作画面設計だった	なぜ? →	**運用を理解しないで設計した要件定義での不備**
	なぜ? →	異常な値が設定できてしまう	なぜ? →	**異常値を設定できないようにする or 警告をするような設計になっていない:設計不備**	なぜ? →	フェイルセーフな設計方針の欠如	なぜ? →	**運用を理解しないで設計した要件定義での不備**		
			なぜ? →	異常系のテストが行われていない	なぜ? →	**網羅確認していない:プロセス不備**				
	なぜ? →	取り消し操作後の水位が初期値となっていた	なぜ? →	**取り消し後の値をもとに戻すか初期値とするか仕様として明確になっていない?:設計不備**	なぜ? →	運用フローが明確になっていない	なぜ? →	**運用を理解しないで設計した要件定義での不備**		
			なぜ? →	元の値に戻す処理が抜けていた?	なぜ? →	**レビューやテストをしていない:プロセス不備**				
1号ゲート停止をこころみるが停止せず	なぜ? →	停止のために「割込操作」スイッチを押した	なぜ? →	ボタン名がオペレーションとマッチしていない	なぜ? →	ボタン類は既存のを流用?				
			なぜ? →	異常時の操作方法が理解できていない	なぜ? →	**マニュアル不備、教育訓練のプロセス不備**				

＊:黒太字は、一般的な分析で真因となるところを示している。

3.4.5 対策の検討

前記問題行動・根本原因対応表（表 3.4.3）の各根本原因に対応して対策を立案し記載する（表 3.4.5）。対策の検討範囲として、手順書・訓練教育等組織文化的な点についても考察に含める。

表　3.4.5　根本原因・対策表

根本原因		対策案			疑問点
分類	内容	対策方法	対象者	備考	
	オペミスの可能性あり				**確認画面ない?**
調査検討の不足	設定画面でないと確認できない	**オペミスを防ぐ意味でも設定を確認するだけの画面を用意すべき**			
調査検討の不足	具体的なオペレーションを理解しないまま画面設計している	一連の設定、確認のオペレーションユースケースを網羅してから画面設計すべき			
手順の不順守 **?**	取り消し操作の場合は以前の設定値を使うようになっていたが初期値にしていた	仕様を確認して実装 or 他できちんと設定しているか確認			仕様確認していない**?** 設定しないまま初期値が使われるようになっているのか不明
手順の不順守	確認手順の不備	計画起動後に設定値を確認するよう手順を見直す			
価値観不良	要求事項がきちんとできているか確認できていない	チェック観点を明記してレビューを行う			
調査検討の不足	**実際に動作するまで異常な設定値がなされている状況が確認できない**	**状況をモニタリングできる画面があるべき**			
調査検討の不足	**作動時間を制限するタイマーで停止するような設定値でもチェックする機能がない**	変更可能な範囲の閾値を設定し、現在値と設定値の差が閾値内かどうかチェックをする 確認画面を出す			
価値観不良	作動時間を制限するような要求仕様がきちんと開発者に伝わっていない	要求仕様を明確にする			
手順の不順守	ゲート停止の手順が理解できていない	手順書の整備、教育、予行訓練など定着のための活動			
	スイッチの名前がオペレーションとマッチしていない	ボタン名称の変更			
手順の不順守	ゲート停止の手順が理解できていない	手順書の整備、教育、予行訓練など定着のための活動			
	スイッチの名前がオペレーションとマッチしていない	ボタン名称の変更			
調査検討の不足	**最終手段でゲート電源断で止めるのはOKとしても遠隔操作不可は問題**	**遠隔操作は可能とすべき**			

他に操作マニュアル不備、操作員教育不備、警報一斉吹鳴自動停止不可等の原因も考えられる。

最後に表 3.4.1 から 3.4.5 を 1 つの表にまとめて一覧できるようにする（表 3.4.6）。

表　3.4.6　分析結果表

問題の症状/経過			問題行動		根本原因		対策案			疑問点
時刻	分類	内容	分類	内容	分類	内容	対策方法	対象者	備考	
14:05		目標ダム水位9.9mに設定								
14:10	不良現象	水位変更画面上で設定水位を確認後、「変更取消」操作を行い計画起動 **→水位設定値が初期値になってしまっていた**	潜在危険	確認のために設定画面を開いた？		**オペミスの可能性あり**				**確認画面ない？**
			ソフト制作	設定状況を確認できない画面になっている	調査検討の不足	設定画面でないと確認できない	**オペミスを防ぐ意味でも設定を確認するだけの画面を用意すべき**			
					調査検討の不足	具体的なオペレーションを理解しないまま画面設計している	一連の設定、確認のオペレーションユースケースを網羅してから画面設計すべき			
					手順の不順守？	取り消し操作の場合は以前の設定値を使うようになっていたが初期値にしていた	仕様を確認して実装 or 他できちんと設定しているか確認			仕様確認していない？設定しないまま初期値が使われるようになっているのか不明
			運転・使用	最終的な設定値の確認をしていない	手順の不順守	確認手順の不備	計画起動後に設定値を確認するよう手順を見直す			
					価値観不良	要求事項がきちんとできているか確認できていない	チェック観点を明記してレビューを行う			
18:05	不良現象	吐水ゲート作動	ソフト制作	設定状況を確認できない画面になっている	調査検討の不足	**実際に動作するまで異常な設定値がなされている状況が確認できない**	**状況をモニタリングできる画面があるべき**			
18:06		2号ゲートが開度49cmで停止	ソフト制作	ゲート停止する仕様とソフト仕様の不一致	調査検討の不足	**作動時間を制限するタイマーで停止するような設定値でもチェックする機能がない**	変更可能な範囲の閾値を設定し、現在値と設定値の差が閾値内かどうかチェックをする 確認画面を出す			
					価値観不良	作動時間を制限するような要求仕様がきちんと開発者に伝わっていない	要求仕様を明確にする			
18:06		1号ゲート停止をこころみるが停止せず	非定常操作	停止のために「割込操作」スイッチを押した	手順の不順守	ゲート停止の手順が理解できていない	手順書の整備、教育、予行訓練など定着のための活動			
						スイッチの名前がオペレーションとマッチしていない	ボタン名称の変更			
18:07		「スケジュールキャンセル」スイッチにより停止すべきところ、「緊急停止」により作動停止	非定常操作	電源断になることをしらず「緊急停止」を押してしまった	手順の不順守	ゲート停止の手順が理解できていない	手順書の整備、教育、予行訓練など定着のための活動			
						スイッチの名前がオペレーションとマッチしていない	ボタン名称の変更			
		「緊急停止」したことにより、現地の電源が止められ遠隔操作不可となる	非定常操作	管理所なのに遠隔操作不可となる	調査検討の不足	**最終手段でゲート電源断で止めるのはOKとしても遠隔操作不可は問題**	**遠隔操作は可能とすべき**			

ここでは対策の担当者を記載していないが、組織として決定の上記入する。

4. 再発防止活動の事例

4 章では、障害の分析結果を再発防止につなげる活動に取り組んだ事例を 2 件紹介する。

4.1 A 社の再発防止活動事例

4.1.1 概要

障害は、人の作業により作り込まれる。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である。本章では、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の定義の事例を紹介する。

4.1.2 開発プロセスによる障害抑制

開発プロセスは、ソフトウェア開発の作業をタスクとして定義して、作業領域（アクティビティ）ごとにタスクを階層的に定義したものである。アクティビティは、一般的に工程とも呼ばれる.本節では、工程で統一する。要求は、曖昧である場合が多く、ソフトウェア要求定義で厳密に要求を定義して設計・実装・テストに落とし込んでいく。各工程の概要を以下に示す。

工程名称	概要
ソフトウェア要求定義	当該製品を実現するためにソフトウェアとして実現が必要となる要求を明確にする。
ソフトウェアアーキテクチャ設計	開発する組込みソフトウェアのアーキテクチャ（＝動作［振る舞い］と構造）を決定する。
ソフトウェア詳細設計	ソフトウェアアーキテクチャ設計で定義された機能ユニットをプログラムユニットに分割し、詳細な振る舞いや論理構造等を設計する。
実装と単体テスト	ソフトウェアを構成する個々のユニットの実装と単体レベルでの動作確認を行う。
ソフトウェア結合テスト	ソフトウェアを構成する個々のプログラムユニットを順次組み立て、それぞれ組み合わせた際の機能が動作するかどうかを確認する。
ソフトウェア総合テスト	ソフトウェアを構成する個々の要素（機能ユニット）をすべて結合した状態で、ソフトウェアとしての総合的なテストを実施する。

障害は、人の作業により作り込まれる。開発工程（ソフトウェア要求定義、ソフトウェアアーキテクチャ設計、ソフトウェア詳細設計、実装、単体テスト、ソフトウェア結合テスト、ソフトウェア総合テスト）に定義したタスクを実行することにより人の作業を安定化して障害を作り込みにくい状態にすることが大切である。

4.1.3 障害抑制方法

障害の真因を分析して障害を未然に防止する作業タスクを定義する。作業タスクを確実に実行させるために各工程の障害に対応した様式フレーム及びソフトウェアフレームワークの定義により確実に作業を実行せざる終えない状況を作り出すことが大切である。作業を安定して実行するための作業タスクを開発プロセスに定義で実現することにより組織的な障害の抑制を可能にする。

図 4.1.1 に示すようにフレームとして様式フレームとソフトウェアフレームワークがあり、ソフトウェア要求定義で定義した内容が各工程の各様式フレームとの双方向に紐付けが大切である。これによりソフトウェア要求定義で厳密に定義した要求を設計・実装・テストのフレームに確実落とし込むことが大切である。更にフレーム間の変換作業を開発プロセスに定義することにより組織的な開発を可能にする。

全くプロセスを持たない組織に置いても障害発生のイベント又は、過去の数年間の障害の分析によるフレームと作業タスクの定義で段階的に開発プロセスを定義することによる改善効果が期待される。

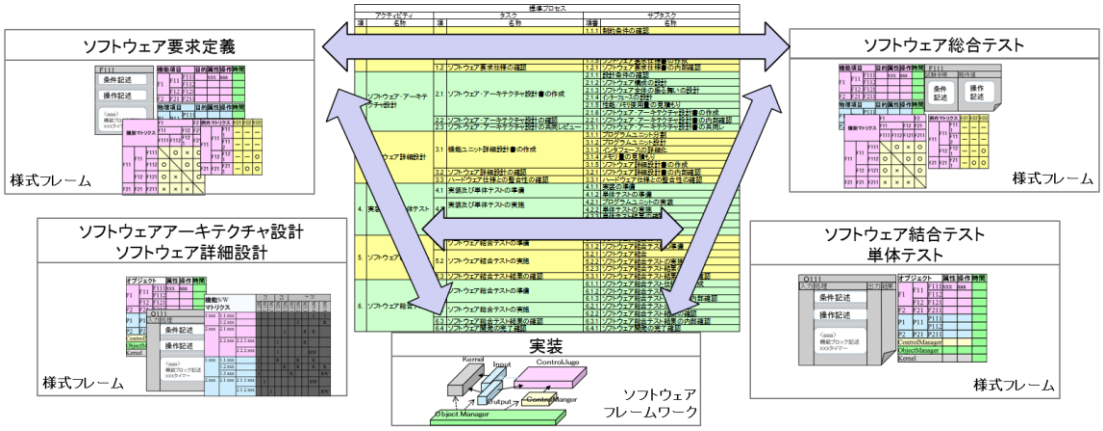


図 4.1.1　フレームとプロセスによる障害抑制・障害分析手順

4.1.4 障害分析

障害分析で重要なことは、障害を分類整理することである。まずは、障害を定義する。定義した障害ごとに障害の発生源である障害発生工程、即ち障害を作り込んだ工程を定義する。次に工程ごとの障害に対して同種の要因で分類して要因カテゴリを作成する。各要因カテゴリの要因項目間での類似性を分析して真因を定義する。定義した真因を未然に防止する作業タスクと様式フレームとソフトウェアフレームワークの定義を実施する。

(1) 障害定義

障害を検出した時点で障害定義として名称、現象、検出工程（障害を発見した工程）、発生工程（障害を作り込んだ工程）、原因（障害の原因）、要因（障害を作り込んだ要因）を定義する。

(2) 発生工程分類（図 4.1.2-①）

障害項目ごとに次節に述べる発生源の分析に基づき発生工程ごとに障害項目を分類する。

(3) 要因分類（図 4.1.2-②）

障害項目ごとに次節の述べる障害真因の分析に基づき発生工程ごとに障害項目を分類して要因カテゴリを定義する。

(4) 真因定義（図 4.1.2-③）

要因カテゴリ内の各要因の共通要因を定義する。定義した要因が発生する真因をなぜなぜ分析により定義する。真因定義は、次節の真因の分析に基づき実施する。

(5) プロセス定義（図 4.1.2-④）

定義した真因に対して、真因を作り込まない作業をタスクとして定義する。タスク名称は、「真因名称」「の解決」の文言でタスク名を定義する。タスク名に対する作業をタスクとして定義する。定義したタスクの右列に定義した真因を定義することでタスク実行の意味を理解できるようにする。

(6) フレーム定義（図 4.1.2-④）

定義したタスクを実行するための様式フレームと必要に応じてソフトウェアフレームワークを定義する。

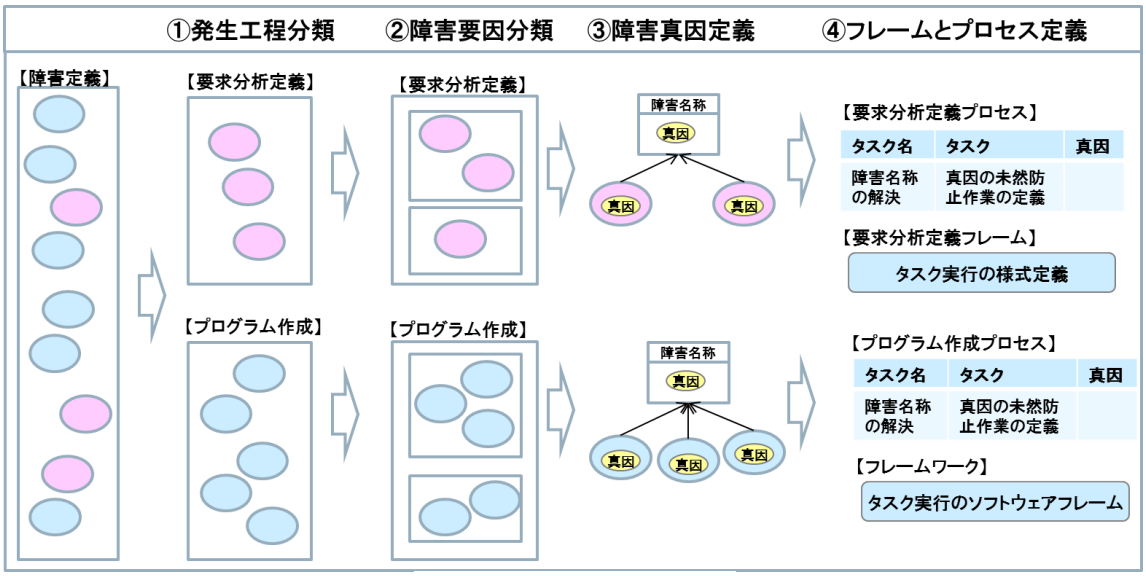


図 4.1.2　障害分析手順

4.1.5 真因分析

真因分析は、発生源の分析と真因の分析に分かれる。それぞれについて説明する。

(1) 発生源の分析

a) 入力品質

タスクへの入力文書が不安定であれば、正しい成果物を定義することができない。タスクの入力に不具合混入要因がないか入力の現物を基に定義漏れ、曖昧性、矛盾がないか分析する。

b) 前工程品質

前工程で定義できていない情報を後工程で定義することはできない。V 字モデルに基づき障害発生源を分析する。障害検出工程の上流工程の成果物の現物を基に定義漏れ、曖昧性、矛盾がないかといった品質を分析する。

特にソフトウェア要求定義の前工程である仕様開発の工程で作成された要求の品質の分析が大切である。要求にない情報はソフトウェア要求定義で定義ができない。そのため設計・実装・テストに落とし込めずに障害を発生させていないかの分析が大切である。

(2) 真因の分析

c) 成果物品質

障害発生工程の成果物の品質を現物のドキュメント、ソフトウェアと様式フレーム及びソフトウェアフレームワークを分析する。

d) プロセス品質

障害発生工程の成果物を生成した作業タスクを文書化して、定義された開発プロセスの作業タスク通り確実に作業が実行できたかタスク実行品質を分析する。

e) レビュー/テスト品質

障害発生工程の成果物のレビューの漏れ・精度及びテスト実行の漏れ・精度を分析する。

f) 計画品質

仕様インプット、ソフトウェアリリースポイント、制約、リスク、プロセス、実行計画、見積り等が計画され精度が確保されているか分析する。

g) 外乱要因

仕様遅延、仕様変更、出荷直前の無理な依頼等の外乱要因により十分な時間確保ができない状況になっていないか分析する。

4.2 B 社の再発防止活動事例

品質不具合の処理フローはルール化されており、発注元会社に出荷して不具合が発生した場合のアクティビティフローが決まっている。　B 社は、ソフトウェア開発を子会社 C 社に開発委託しており、問題発生時には B 社のシステム開発部門と、このソフトウェア開発子会社 C 社が対処する。

ソフトウェア開発子会社がやるべきことは不具合解決手順として決まっている。

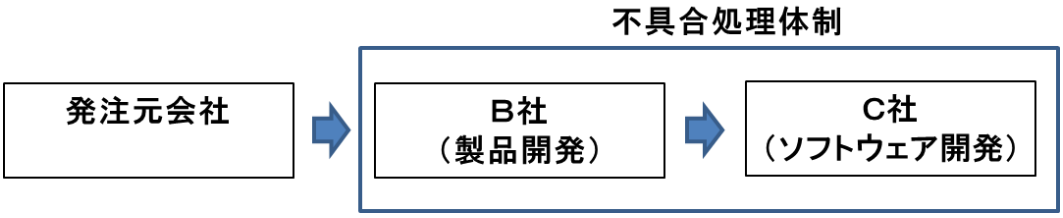


図 4.2.1　不具合処理体制

4.2.1 不具合解決手順

プロジェクトマネージャを活動の起点として以下の手順で不具合解決から対策展開まで行う。製品によっては、発注元会社への報告も行う。

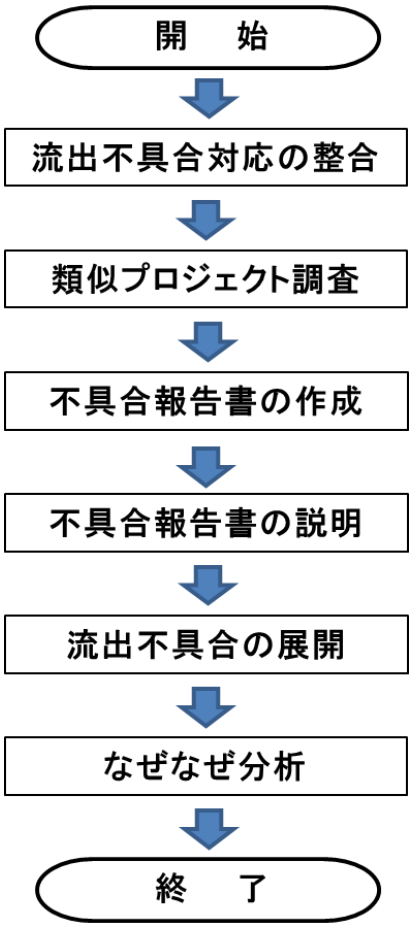


図 4.2.2　アクティビティフロー

(1) 流出不具合対応の整合

システム、ハードウェア、ソフトウェアを付きあわせて本当に不具合か仕様なのかを切り分ける。原因究明は B 社が主体になり、FTA によりハードウェアとソフトウェアの切り分けを行う。あまり長時間はかからない。

(2) 類似プロジェクト調査

コンポーネント化しているので先に関係者を集めて既に解決済か否かの確認を含め調査する。

(3) 不具合報告書の作成

社内報告書としてだけでなく発注元会社への報告にも使用する。

(4) 不具合報告書の説明

発注元会社へ説明

(5) 流出不具合の展開

社内関係部門へ対策含め報告する。(報告書の通知)

(6) なぜなぜ分析

不具合の根本原因の分析と再発防止具体策の作成、展開のためになぜなぜ分析を行う。

ソフトウェアリーダを中心に実施し、プロジェクトマネージャが責任を持つ。実施体制(ソフトウェアリーダ、プロジェクトマネージャ、システム開発部隊、(発注元会社))はルール化されており、「なぜなぜ分析手順」ガイドブックがある。

ガイドブックは、30 項目のガイドから構成されており、分析は、基本として 5 段階まで行う。なぜなぜ分析のシートは流出側と作り込み側の 2 つに分かれている。また問題の特定ではプロセス特定まで行っている。また、なぜなぜ分析では、真因に行きつかない場合は、真因を想定して行うこともある。

※(発注元会社の)なぜなぜ分析資料中の注意点(抜粋)

- ✓ 一人称で書く
- ✓ 問題は事象やメカニズムでなく引き起こした直接の要因とする
- ✓ 結論を最初に設定して分析しないこと
- ✓ 推論から導かれた要因をそのまま当てはめず必ず事実の裏付けをとること
- ✓ 言い訳を書かない

上記対応に則した所定の不具合再発防止報告書フォーマットがあり、障害の「発生確率」の欄も用意している。

4.2.2 報告

不具合再発防止報告書には、以下の項目が含まれている。

(1) 作り込み要因

(2) 作り込み工程

(3) 発生頻度

通常操作・通常条件で発生、特定操作・通常条件で発生等

(4) 混入サイクル

先行、試作、生産

(5) 重要度

安全性に影響、客先金銭的損害、安全性イメージ低下等

(6) トリガー

使用上ありえる操作、チャタリングノイズ、電圧変動（マイコンリセット有無)、ハードウェアの変更、公差ばらつき

(7) ソフトウェア内原因リスト

処理タイミング考慮もれ、処理順序考慮もれ、データ初期化ミス、割り込み干渉、データ算出タイミングミス、データ授受タイミングミス、データの同時性考慮もれ、データの異常値考慮もれ、排他制御ミス、マイコン仕様の理解不足、分析内容の反映漏れ、等

（※データ初期化ミスが多い）

(8) 根本対策

なぜなぜ分析の結果から根本対策を策定し実施する。

4.2.3 再発防止

根本原因を分析した不具合例をサンプルにしてケーススタディしており、結果を教材として新人等に教育している。半期ごとに実績効果を把握しており、レビューアやプロジェクトマネージャの資質もルールで明確にしている。

社内教育は、不具合対策がメインではなく、機能／実現方法のノウハウ紹介をメインに行っており、不具合に起因するケーススタディだけでなく、自社の設計ノウハウを伝授するための取り組みが元々あり、それに不具合のケーススタディを追加している。

再発防止策は「チェックリスト」に蓄積し共有している。不具合事例から対策が導かれると追加するため項目は増加していくが、一方でチェックリストを全て事前に確認することが難しくなりつつある。再発防止の徹底には、さらなる取組の検討も考えている。さらに、製品全体として再発防止の徹底を図るために、ソフトウェア開発を委託している C 社以外の協力会社に対して実施している教育の中に付加的に再発防止策を入れている。再発防止策はチーム内で合意を得るまで検討するため時間を要することがある。

付録 1：失敗知識データベースと失敗まんだら

独立行政法人科学技術振興機構（JST）では失敗知識データベース構築のために、「失敗まんだら」と呼ばれる表現方法で失敗の結果、失敗を起こす行動、失敗を起こす原因を分類している。失敗まんだらとは、失敗を生かそうとしている人が頭の中に持っている失敗の知識を、仏教で悟りの世界や仏の教えを示した「まんだら図」を参考に、階層的に構造化したものである。（参考文献［11］）

失敗知識データベースの構造と表現

（「失敗まんだら」解説）

平成 17 年 3 月

独立行政法人科学技術振興機構（JST）

失敗知識データベース整備事業

統括　畑村 洋太郎

http://www.sozogaku.com/fkd/inf/mandara.html

失敗の脈絡と構造化

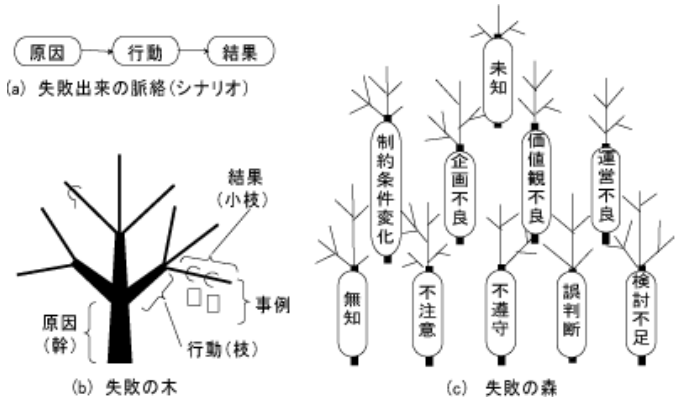


図 1－1　失敗知識データベースの構造と表現

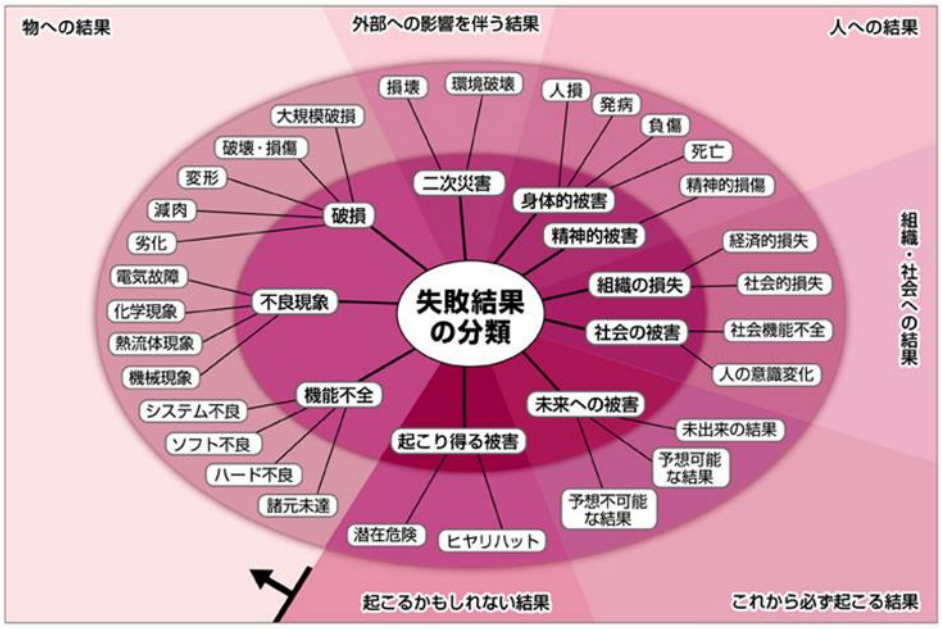
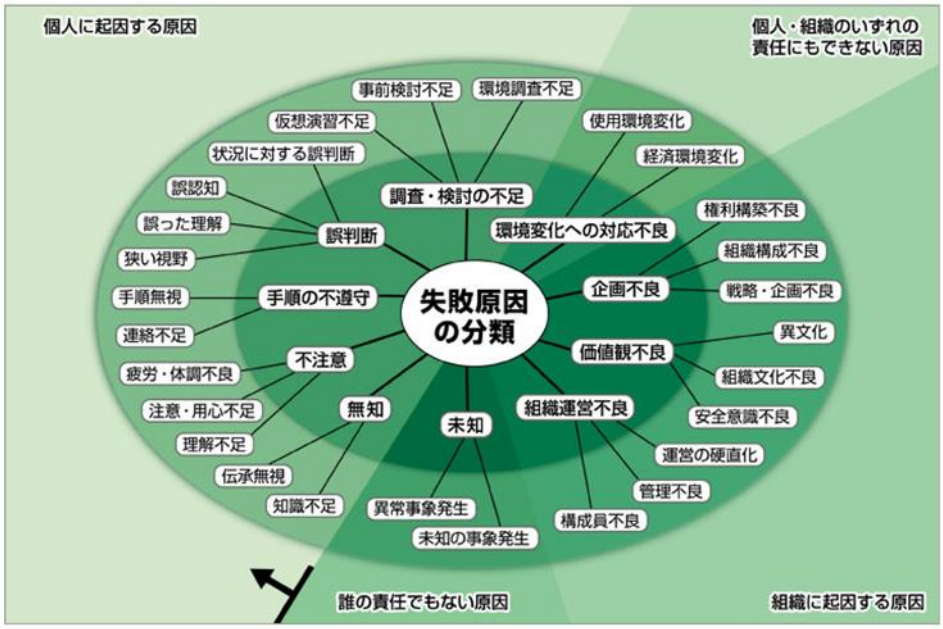


図 1－2　失敗まんだら（失敗結果の分類）

図 1－3　失敗まんだら（失敗行動の分類）



JST畑村委員会作成 2002

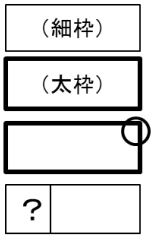
図 1－4　失敗まんだら（失敗原因の分類）

付録 2：VTA（Variation Tree Analysis：変動木分析）概説

VTA（Variation Tree Analysis）は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おり、品質学会の論文等でも良く見受けられる手法である。

（1）ノード

変動要因(Variation Factor)としての事象，通常（正常）の状態などを記述する.



①通常ノード：通常の状態に関する事象

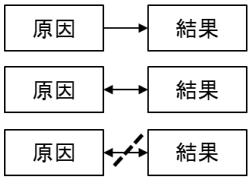
②変動ノード：通常から逸脱した行動や状態

③排除ノード：事故などに直接結びつく行動や状態

④推定ノード：分析者の推測によるもの

（2）関連付け

ノード間の因果関係を明らかにする.



①事象に方向性がある場合

②事象が相互に関連ある場合

③相互のコミュニケーションに問題がある場合

---

④ブレイク（切断点）：因果関係，事象の連鎖を断ち切ることで事故への進展を防ぐポイント

本書 PARTⅢの 3.3 節等にも VTA が解説されているので、そちらも参照のこと。

*参考文献

[1] 情報処理システム高信頼化教訓集（製品・制御システム編）　独立行政法人　情報処理推進機構　技術本部　ソフトウェア高信頼化センター（平成26年5月13日）
http://www.ipa.go.jp/sec/reports/20140513_2.html

[2] 運輸安全委員会：鉄道事故調査報告書：湘南モノレール株式会社　江ノ島線西鎌倉駅構内　鉄道物損事故　平成21年6月26日
http://www.mlit.go.jp/jtsb/railway/rep-acci/RA2009-6-1.pdf

[3] 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

[4] 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

[5]「駒場堰堤ゲート異常作動原因調査　報告書」（平成14年6月　国土交通省中部地方整備局）

[6]「ダム等ゲート類の異常作動等の再発防止について－　点検・確認結果　－　」（平成14年7月26日　原子力安全・保安院）
http://warp.ndl.go.jp/info:ndljp/pid/286890/www.meti.go.jp/kohosys/press/0002951/0/020726damu.pdf

[7] 原子力安全・保安院電力安全課：中部電力（株）駒場堰堤洪水吐ゲートの異常作動について　平成14年04月11日（木）
http://warp.ndl.go.jp/info:ndljp/pid/286890/www.meti.go.jp/kohosys/press/0002611/

[8] 失敗百選 ～長野の駒場ダムの異常放流（2002）～
http://www.sydrose.com/case100/325/

[9] 島田啓史、青木信、新標準準拠 ダム管理用制御処理設備の開発、日本無線技報、No.53, pp.37-40（2007）
http://www.jrc.co.jp/jp/company/html/review53/pdf/JRCreview53_10.pdf

[10] 日本ヒューマンファクター研究所：品質とヒューマンファクター安心と安全の考え方、財団法人日本科学技術連盟

[11] 畑村創造工学研究所　畑村洋太郎：失敗知識データベースの構造と表現
http://www.sozogaku.com/fkd/inf/mandara.html

付録 A：障害情報の取扱いルール

障害情報を記録する共通様式の設計、機密保持・情報提供の方法のルール等をまとめた。

1. はじめに

情報システムの障害事例情報を収集・分析し、その教訓を社会で共有する仕組みの構築が望まれる。そのためには、国民生活や経済活動に一定以上の影響を及ぼした障害について、事業者が積極的に情報提供を行えるよう、障害情報を記録する共通様式を設計するとともに、機密保持・情報提供の方法に関するルールを作成することが必要である。ここでは、今回の IPA/SEC における活動を通して検討され、一部試行された、障害情報を記録する共通様式の設計、機密保持・情報提供の方法のルールについて説明する。

2. 障害を記録する共通様式

障害情報の記録様式としては、内閣官房情報セキュリティセンター（以下、NISC）の「政府機関統一基準適用個別マニュアル群」[3]の中で参考資料として提供されている。また、金融庁のように、システム障害発生時に法令等に基づく届出を行う際の書式例[4]を定めている政府機関もある。民間の事業者においても、障害やインシデントの記録様式を定めて障害管理を行っている[5]。

ここでは、NISC の様式で定める項目群をほぼ踏襲し、一部に IPA/SEC での試行結果を反映した様式を作成した。

（1）障害情報を記録する報告書の項目

NISC の様式と異なるところは、主に次の部分である：

・障害の対処は、その原因が徐々に詳細に判明していくのに伴い、影響等を考慮しつつ幾次にもわたって行われることもある。従って、対処内容を時系列に記載することとした。

・今回の目的は、教訓共有のための障害の事実記録である。従って、発見者等の個人に関する項目は、組織を越えて共有する必要はなく、報告不要とした（下記における下線部分）。

[3] http://www.nisc.go.jp/active/general/kijun_man.html

[4] 「申請書等様式集」 http://www.fsa.go.jp/common/law/guide/seisan_b.pdf

[5] IPA/SEC「情報システム障害の再発防止のための組織的マネジメントの調査 WG 報告書」（平成 24 年 4 月 5 日） 2.2 節 http://www.ipa.go.jp/sec/softwareengineering/reports/20120405.html

今回作成した障害情報を記録する報告書の項目を以下に記す。なお、今後の運用の中で、必要に応じて追加や精緻化等を行っていきたい。

（障害情報）

・障害等管理番号　　：障害をユニークに管理するために番号を付与
・システム名　　：障害の発生したシステム名を記載
・発見日時　　：障害の発生した日付、時間を記載
・対象　　：障害の発生した箇所を記載
・状況　　：障害状況を記載，現象，影響度（・重大度）を含む
・発見者の情報　　：障害を発見した人の情報を記載
・受理者の情報　　：障害報告を受理した人の情報を記載
・通知先の情報　　：障害の通知先を記載
・照会先の情報　　：障害に関する照会窓口を記載

（障害対処情報）

・実施した対処内容　　：実施した対処内容を時系列（①、②、...）に記載
・対処日時　　：対処した日付、時間を内容と対応（①、②、...）させて記載
・再発防止策内容　　：今後実施予定の再発防止策を記載（できれば時系列に）
・防止策実施予定日　　：再発防止策の実施予定日を記載
・原因　　：障害の発生原因を記載
（直接的な原因や根本的な原因等、複数レベルあり）
・対処の実施者情報　　：対処を実施した人の情報を記載
・対処の承認者情報　　：対処を承認した人の情報を記載

(2) 障害情報を記録する報告書の記載例

(1) の様式に基づく障害情報報告の記載例を以下に示す。

なお、今回の目的は、教訓共有のための障害の事実記録であるため、網掛け部分に示したような発見者等の個人に関する項目は、組織を越えて共有する必要はなく、報告不要である。

(障害情報)

障害等管理番号	分類(分野)コード+連番
システム名	銀行システム/残高照会
発生日時	2013 年 1 月 23 日 9 時 00 分
対象	○○銀行
状況	○○銀行の残高照会処理で、以下の条件に該当する場合、異なる口座名義人の残高が表示された。 ・同姓同名 ・生年月日情報が口座情報ファイルに未登録 (約 1,000 人が対象)

(障害対処情報)

実施した対処内容	①(当面の対応)状況欄記載の該当者が残高照会を行った場合、「銀行窓口で照会ください」のメッセージを表示する ②(当該障害への対応)経路 X から口座情報ファイルにデータを反映する場合は、■■ファイルに保持している生年月日情報を付加する
対処日時	①2013 年 1 月 23 日 14 時 00 分 ②2013 年 1 月 30 日 9 時 00 分
再発防止策内容	③(類似障害への対応)組織内において、エラー発生時に表示させるメッセージ内容を管理する。また、IT システムエラー時の事務マニュアルの整備状況を管理する。
防止策実施予定日	③2013 年 4 月

発見者	氏名	山田　太郎
	所属	東京支店
	連絡先	03-1111-2222
	発見日時	2013 年 1 月 23 日 11 時 30 分
受理者	氏名	鈴木　一郎
	所属	営業企画部
	連絡先	03-1111-1234
	受理日時	2013 年 1 月 23 日 13 時 20 分
通知先	氏名	伊藤　花子
	所属	品質管理部
	連絡先	03-1111-9999
照会先	氏名	佐藤　はじめ
	所属	品質管理部
	連絡先	03-2222-9999

原因	①エラーが発生した場合の運用を決めていなかったため。要件定義漏れ。 ②人を特定するためのキー情報(名前、生年月日、性別等)のうち、特定の経路 X から反映されたデータで生年月日が反映漏れ。 ③組織内で統一したエラー発生時の対応手順を決めていなかったため。

対処実施者	氏名	①②山田　花子
	所属	①②口振収納システム課
照会先	氏名	①②田中　一郎
	役割	①②アプリケーションオーナ(責任者)
	所属	①②口振収納部

3. 機密保持・情報提供の方法のルール等

3. 1　考え方

障害情報には、その提供元の機密事項や信用にかかわる事項が含まれることが多い。そのため、障害情報を収集する際は、情報提供者や共有活動への参加者が不利益を被らないよう機密保持等のルールを取り決めて、事前に情報提供者の同意を得る必要がある。また、情報の提供を受ける側も、その管理のルールを遵守する必要がある。ここでは、機密保持・情報提供の方法に関するルールについて、以下の手順でまとめる。

(a)　情報の流れのモデル化

機密保持等のルールの適用局面は、情報の収集方法（情報の提供者）によって異なることが想定される。障害情報を収集した後、それらを教訓として取りまとめ、公開するまでの情報の流れをモデル化する。また、収集した障害情報の管理方法等、機密保持等のルールの適用局面を整理する。

(b)　情報の流れと決めるべきルール

情報の開示可能な範囲、施すべき情報の加工のレベル、またその確認方法等、上記モデルの各ケース、各局面において、どのようなルールを取り決めればよいか、具体的なルールを示す。

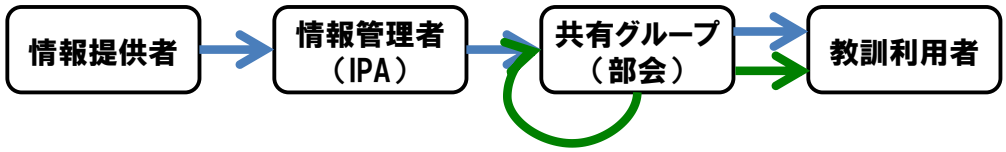
(c)　適用例

ルールの実効性を確認するため、IPA/SEC における「重要インフラ IT サービス高信頼化部会」や個別ヒアリング等の活動において、実際に使用した適用例を示す。

3．2　情報の流れのモデル化

障害情報の収集から、それを教訓として公開するまでの情報の流れを、情報の収集方法ごとに、下図に示すようにモデル化した。さらに、このモデルに基づいて、機密保持等のルールの適用局面を整理した。

情報の流れのモデル



No.	収集方法	ルールの適用局面（義務を負う者）			
		情報提供者から預かった障害情報の管理（情報管理者）	共有グループへの情報開示時（情報管理者）	共有グループでの議論時（共有グループメンバ）	教訓の公開時（情報管理者）
1	情報管理者が、個別ヒアリング等により提供された障害情報を取り扱う場合	情報提供者との間で、機密保持のルールを適用	情報提供者との間で、加工情報の共有に係るルールを適用	情報管理者との間で、機密保持のルールを適用	情報提供者との間で、公開のための情報加工のルールを適用
2	共有グループのメンバが、障害情報をグループ内に直接開示した場合	適用なし	適用なし	共有メンバ間で、機密保持のルールを適用	情報提供者との間で、公開のための情報加工のルールを適用

(注)ここでは、情報管理者をIPA、共有グループをIPA内に設置された部会として説明する。
また、共有グループの運用を情報管理者が行うことを前提としている。
情報管理者は共有グループのメンバでもある。
これらは、本ルールの適用先に応じ、適切な組織・会議体に置き換えることになる。

ここで、情報管理者とは、情報提供者から提供された障害情報を、適切に管理する責任がある者を指す。

共有グループとは、情報提供者から提供された、一般には公開できないような機微な内容が含まれた障害情報を、ここに示す機密保持等のルールの下、再発防止のための教訓として作り上げる議論を行う者の集まりを指す。

上記の表の No.1 では、情報管理者が、個別ヒアリング等により提供された障害情報を取り扱う場合に必要となる、ルールの適用局面を示した。

No.2 では、共有グループのメンバが、障害情報をグループ内に直接開示した場合に必要となる、ルールの適用局面を示した。

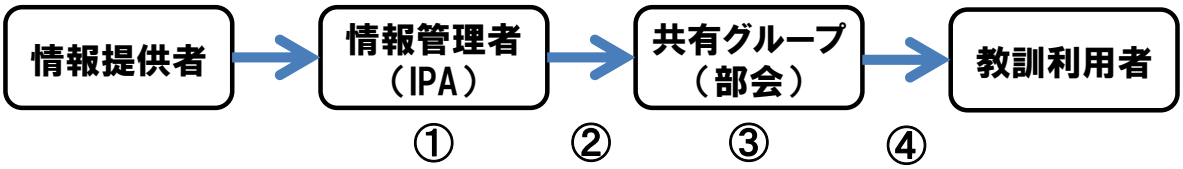
以降の節で、各ケース、各局面において、どのようなルールを取り決めればよいか、情報の収集方法によって異なる機密保持等のルールの適用局面について、情報の流れに沿って整理した。

3．3　情報の流れと決めるべきルール

3．3．1　情報管理者が、個別ヒアリング等により提供された障害情報を取り扱う場合

情報管理者が、個別ヒアリング等により提供された障害情報を取り扱う場合の、情報の流れとルールの適用局面を下図に示す。丸数字で示した箇所が、ルールを適用すべき局面である。

情報管理者が、個別ヒアリング等により提供された障害情報を取り扱う場合



（1）情報の取扱いに関するルール

①　情報管理者における情報提供者から預かった障害情報の管理

情報提供者から預かった障害情報は適切に管理されなければならず、情報管理者に対して、機密保持のルールの適用が必要となる。以下に、検討すべき機密保持レベルの例を示す。情報管理者は、機密保持レベルに応じた情報開示範囲を遵守しなければならない。

・特定の情報取扱者に限る

情報管理者において別途定めた、機密情報取扱者に限り、情報にアクセスできる。

・管理組織内

情報管理者が所属している組織内（IPA/SEC 内）まで、情報にアクセスできる。

なお、IPA/SEC の活動においては、上記レベルの中で「管理組織内」を標準として設定した。（ただし、状況に応じて情報提供者と協議した）

②　共有グループへの情報開示時

共有グループへ情報を開示する場合、情報管理者に対して、情報加工のルールの適用が必要となる。以下に、検討すべき情報加工レベルの例を示す。情報管理者は、情報加工レベルに応じて情報をマスキングした上で、共有グループに開示しなければならない。

・障害情報の抽象化

個別システム名称・企業名等が分からないように匿名化すると共に、システム／サービス内容が分からないように抽象化する。ただし、システム構成や処理・運用手順等は、教訓や対策手法に関する議論に資するレベルの情報とする。

・匿名化

個別システム名称・企業名等が分からないように匿名化した上で、共有グループへ開示する。

・加工なし

あらかじめ情報提供者の同意が得られている場合に限る。

ここではさらに、上記ルールにて加工した情報を、共有グループに開示する際の確認のルールを取り決める必要がある。

・情報提供者への確認

共有グループに開示する前に、加工済み障害情報の開示範囲について、共有グループにおける情報取扱いルールを説明した上で、情報提供者へ同意を得る必要がある。

なお、IPA/SEC の活動においては、上記レベルの中で「匿名化」を標準として設定した。（ただし、状況に応じて情報提供者と協議した）

③　共有グループでの議論時

共有グループに情報を開示し、議論を行う場合、議論に参加する共有グループメンバに対して、機密保持のルールの適用が必要となる。以下に、検討すべき機密保持レベルの例を示す。共有グループメンバは、機密保持レベルに応じた情報開示範囲を遵守しなければならない。

・メンバ限り

共有グループのメンバに限り、情報にアクセスできる。

・メンバ及び所属部門内

共有グループのメンバと、そのメンバが所属する企業の部門内まで、情報にアクセスできる。

・メンバ及び所属企業内

共有グループのメンバと、そのメンバが所属する企業内まで、情報にアクセスできる。

なお、IPA/SEC の活動においては、上記レベルの中で「メンバ及び所属企業内」を標準として設定した。（ただし、状況に応じて情報提供者と協議した）

④　障害事例・教訓の公開時

教訓を一般利用者へ公開する場合、共有グループ運営者に対して、情報加工のルールの適用が必要となる。以下に、検討すべき情報加工レベルの例を示す。共有グループ運営者は、情報加工レベルに応じて情報をマスキングした上で、一般利用者へ公開しなければならない。

・障害事例・教訓の抽象化

個別システム名称・企業名等が分からないように匿名化すると共に、システム／サービス内容が分からないように抽象化する。ただし、システム構成や処理・運用手順等は、問題を理解し、対策として実践できるレベルの情報とする。

・匿名化

個別システム名称・企業名等が分からないように匿名化した上で、一般利用者へ公開する。

・加工なし

あらかじめ情報提供者の同意が得られている場合に限る。

ここではさらに、上記ルールにて加工した情報を、一般利用者に公開する際の確認のルールを取り決める必要がある。

・情報提供者への確認

一般利用者に公開する前に、障害事例・障害情報から導かれた教訓事例の公開について、情報提供者へ同意を得る必要がある。ただし、あらかじめ情報提供者の同意が得られている場合はこの限りではない。

なお、IPA/SEC の活動においては、上記レベルの中で「障害事例・教訓の抽象化」を標準として設定した。（ただし、状況に応じて情報提供者と協議した）

(2) ルールの確認方法

この情報の流れの中で、上記各局面で設定した機密保持等のルールについては、情報提供者とどのように確認するかを決めておく必要がある。以下に、検討すべき確認方法の例を示す。

なお、この場合、情報管理者は、共有グループの運営者及びメンバとの間で、機密保持等のルールについてあらかじめ確認しておくことが前提となる。情報提供者に対する確認内容には、それらをどのようにしたかの確認結果を含めることになる。

・書面等による確認

情報管理者から、情報取扱いに関する上記①～④の局面で適用する機密保持等のルールを書面等により説明し、情報提供者の承諾を得る。（誓約書の提示等を含む）

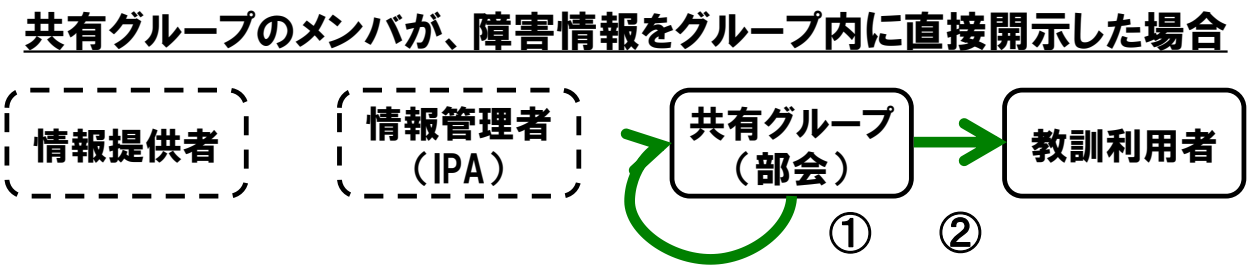
・機密保持契約書（NDA）の締結

特に機微性の高い情報を扱う場合や情報提供者からの求めがある場合等には、情報管理者と情報提供者との間で、機密保持契書（NDA）を締結し、上記①～④の局面で適用する機密保持等のルールを確認する。

なお、IPA/SEC の活動においては、上記方法の中で「書面等による確認」を標準として設定した。（ただし、状況に応じて情報提供者と協議した）

3．3．2　共有グループのメンバが、障害情報をグループ内に直接開示した場合

共有グループのメンバが、障害情報をグループ内に直接開示した場合の、情報の流れとルールの適用局面を下図に示す。丸数字で示した箇所が、ルールを適用すべき局面である。



（1）情報の取扱いに関するルール

①　共有グループでの議論時

共有グループメンバが、障害情報をグループ内に直接開示し、議論を行う場合、議論に参加する共有グループメンバに対して、機密保持のルールの適用が必要となる。以下に、検討すべき機密保持レベルの例を示す。共有グループメンバは、機密保持レベルに応じた情報開示範囲を遵守しなければならない。

・メンバ限り

共有グループのメンバに限り、情報にアクセスできる。

・メンバ及び所属部門内

共有グループのメンバと、そのメンバが所属する企業の部門内まで、情報にアクセスできる。

・メンバ及び所属企業内

共有グループのメンバと、そのメンバが所属する企業内まで、情報にアクセスできる。

なお、IPA/SEC の活動においては、上記レベルの中で「メンバ及び所属企業内」を標準として設定した。（ただし、状況に応じて情報提供者と協議した）

②　障害事例・教訓の公開時

教訓を一般利用者へ公開する場合、共有グループ運営者に対して、情報加工のルールの適用が必要となる。以下に、検討すべき情報加工レベルの例を示す。共有グループ運営者は、情報加工レベルに応じて情報をマスキングした上で、一般利用者へ公開しなければならない。

・障害事例・教訓の抽象化

個別システム名称・企業名等が分からないように匿名化すると共に、システム／サービス内容が分からないように抽象化する。ただし、システム構成や処理・運用手順等は、対策として実践できるレベルの情報とする。

・匿名化

個別システム名称・企業名等が分からないように匿名化した上で、一般利用者へ公開する。

・加工なし

あらかじめ情報提供者の同意が得られている場合に限る。

ここではさらに、上記ルールにて加工した情報を、一般利用者に公開する際の確認のルールを取り決める必要がある。

・情報提供者への確認

一般利用者に公開する前に、障害事例・障害情報から導かれた教訓事例の公開について、情報提供者へ同意を得る必要がある。ただし、あらかじめ情報提供者の同意が得られている場合はこの限りではない。

なお、IPA/SEC の活動においては、上記レベルの中で「障害事例・教訓の抽象化」を標準として設定した。（ただし、状況に応じて情報提供者と協議した）

(2) ルールの確認方法

この情報の流れの中で、上記各局面で設定した機密保持等のルールについては、共有グループ内でどのように確認するかを決めておく必要がある。以下に、検討すべき確認方法の例を示す。

・書面等による確認

共有グループ運営者からの共有グループ内への配布文書等の書面説明にて、上記①～②の局面で適用する機密保持等のルールを確認する。（誓約書の提示等を含む）

・機密保持契約書（NDA）の締結

特に機微性の高い情報を扱う場合や情報提供者からの求めがある場合等には、共有グループで情報を共有する者は、情報管理者との間で同一文面の機密保持契書（NDA）を締結し、上記①～②の局面で適用する機密保持等のルールを確認する。

なお、IPA/SEC の活動においては、上記方法の中で「書面等による確認」を標準として設定した。具体的には、誓約書の提示により確認を行った。（ただし、状況に応じて情報提供者と協議した）

3．3．3　情報管理者が、共有グループのメンバから障害情報を提供された場合

本文書にて、情報の流れのモデル化はしていないが、他に想定される障害情報の収集方法として、「情報管理者が、共有グループのメンバから障害情報を提供された場合」が考えられる。この場合、情報管理者に対して機密保持等のルールの適用が必要となるため、決めるべきルールは、上記 3．3．1 情報管理者が、個別ヒアリング等により提供された障害情報を取り扱う場合　に準ずることとなる。

3．4　適用例

IPA/SEC 内に設置した共有グループ「重要インフラ IT サービス高信頼化部会」にて、実際に適用した資料の例を以下に示す。なお、共有グループの運用者は、IPA/SEC である。

～情報提供の取扱い～【開示可能範囲：委員及び委員所属企業限り】

機密保持・情報提供者保護のため、情報取扱いルールを以下のとおり制定する。

■提供頂いた機微情報は、SEC 内でも必要最低限の範囲で利用する。
【開示可能範囲：特定の担当者限り】

■提供頂いた機微情報及び共同で教訓化した情報について、「教訓の認定・公開」等のため、第三者に情報を提出する際には以下を遵守する。

・システム／サービス名や個人名等を匿名化する。また、これらが特定できるような情報も匿名化／加工する。
・教訓情報は、情報提供者の承認なく第三者に提供しない。
（教訓情報の提供者の意向に沿って必要な整形を施し、あらかじめ情報提供者の了解を得ることが、第三者への提供条件）

※資料提供者の要望により、開示範囲をさらに限定する場合もある。

（注）ここで、「委員及び委員所属企業限り」は、上記 3．3　情報の流れと決めるべきルール　で記載した「メンバ及び所属企業内」と同等である。

～部会資料の取扱い～【開示可能範囲：委員及び委員所属企業限り】

◆部会配布資料の開示可能範囲
配布資料については、守秘を徹底するため、以下の通りとする。

■原則、以下の範囲とする。
【開示可能範囲：委員及び委員所属企業限り】
（委員には、委員の代理＋登録オブザーバも含む）

■情報提供者の要望に応じて、開示範囲をさらに限定する。
（例）【開示可能範囲：委員限り】
【開示可能範囲：委員及び委員所属部門限り】
（委員には、委員の代理＋登録オブザーバも含む）

■上記文言を資料の全ページに表示する。

～議事録の開示範囲～

■原則、以下の範囲とする。
①事務局作成及び、委員に確認依頼中の段階
【開示可能範囲：委員限り】
（委員には、委員の代理＋登録オブザーバも含む）
②委員確認後、議事録内容が確定した段階
【開示可能範囲：委員及び委員所属企業限り】
（委員には、委員の代理＋登録オブザーバも含む）

■議事録への記載を希望しない内容については、当該部分を会議中又は議事録確認中に、指摘を依頼する。

付録 B：信頼性を表す評価指標

産業分野	評価指標	説明	備考
鉄道	平均遅延時間（総遅延時分）	対象となるトラブル 1 件で生じた列車の遅れ時分の総計	東海道新幹線関連記事（平成 26 年 1 月 5 日）
	輸送障害件数	運休 30 分以上@旅客、1 時間以上@旅客以外の遅延	
航空機	遅延率	遅延とは出発予定時刻より 15 分を超えて出発した便	国土交通省情報統計
	欠航率	—	国土交通省情報統計
自動車	Consumer Reports Score	Predicted Reliability、Owner Satisfaction、Accident Avoidance	米国コンシューマリポート ACSI
	日本自動車初期品質調査（IQS）	新車購入後 2～9 ケ月経過したユーザーから聴取した不具合経験に基づく評価（ブレーキの反応、部品が壊れる、異音、振動等）	JD パワー調査
医療装置	稼働率、修理件数	—	
プラント	稼働率、メンテナンス時間、MTBF、MTTR	—	
昇降機	稼働率、修理件数	—	
家電機器	故障率、クレーム件数	—	コールセンタ問合せ
サーバシステム	MTBF、MTTR	—	JD パワー調査
移動体通信	通話接続率、パケット接続率	—	各社調査（ipsos、ソフトバンク等）

注）

MTBF（Mean Time Between Failure）：平均事故間隔。
MTTR（Mean Time To Recovery）：平均修復時間。
JD パワー（J. D. Power and Associates）は、アメリカ合衆国・カリフォルニア州を拠点とする、世界的な市場調査及びコンサルティング会社。

ACSI（American Customer Satisfaction Index）

ipsos：イプソスは、世界規模を誇るグロ―バル市場調査会社。

編著者

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

協力者

【製品・制御システム高信頼化部会】

主査	内平 直志	国立大学法人北陸先端科学技術大学院大学
副主査	三原 幸博	独立行政法人情報処理推進機構
	安達 和孝	日産自動車株式会社
	天嵜 聡介	公立大学法人岡山県立大学
	岩崎 新一	日本電気株式会社
	河合 浩明	アイシン精機株式会社
	小泉 忍	株式会社日立製作所
	五味 弘	一般社団法人電子情報技術産業協会（JEITA）/沖電気工業株式会社
	鈴木 哲雄	富士電機株式会社
	鈴村 延保	アイシン・コムクルーズ株式会社
	高木 徳生	オムロンソーシアルソリューションズ株式会社
	中岡 邦夫	三菱電機株式会社
	野本 安栄	株式会社日立産業制御ソリューションズ
	長谷川 賢一	株式会社富士通コンピュータテクノロジーズ
	久住 憲嗣	国立大学法人九州大学
	細谷 伊知郎	トヨタ自動車株式会社
	三浦 邦彦	矢崎総業株式会社
	薬袋 正和	横河電機株式会社
	門田 浩	一般社団法人組込みシステム技術協会（JASA）

【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
	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
	海野 和由	矢崎総業株式会社

	鈴村　延保	アイシン・コムクルーズ株式会社
	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
	光田　貴志	オムロン株式会社
	武藤　貴志	日本電気株式会社

【障害事例検証 WG】

主査	天嵜　聡介	公立大学法人岡山県立大学
	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
	小島　正	株式会社日立産業制御ソリューションズ
	津田　昌之	パイオニアシステムテクノロジー株式会社
	羽田　裕	日本電気通信システム株式会社
	馬場　匡史	株式会社富士通コンピュータテクノロジーズ

（所属は 2015 年 3 月時点のもの）