システムグループ

重要インフラ等システム障害対策（製品・制御システム）

SEC 調査役
三原　幸博

SEC 研究員
松田　充弘

SEC 調査役
石田　茂

SEC 調査役
十山　圭介

SEC 調査役
石井　正悟

交通機関や電気・水道システムなどの機器の制御を行う製品・制御システム（組込みシステム）の障害事例の収集・分析と対策の検討を行い、その結果を普遍化した「教訓」として取りまとめた「情報処理システム高信頼化教訓集（製品・制御システム編）」2014 年度版※1 を公開した。併せて企業における品質文化を醸成するための「製品・制御システム高信頼化のための行動指針」※2、障害発生時の真因分析のための「障害分析手法事例解説書」※3、モデルベースアプローチ、システムズエンジニアリング手法に基づく障害診断のための「大規模・複雑化した組込みシステムのための障害診断手法」※4 を公開した。

1 製品・制御システム分野における高信頼化

近年、機器や製品（以下、製品・制御システム）の機能の大半がコンピュータを利用してソフトウェアで実現されるようになってきている。それらには社会インフラとして重要な役割を担うものも多く、高い信頼性が求められる場合が少なくない。しかし、製品・制御システムは、実現する機能規模が肥大化すると共に異なるシステム同士が複合化する傾向にあり、システム全体として信頼性を確保するための技術面での工夫や運用管理での工夫が求められている。

一方、企業間競争の激化により、差分開発といった短納期の製品開発が主流となり、システム高信頼化のための技術やノウハウが企業内でうまく伝承されていないといった問題も顕在化している。

そこで IPA/SEC は、製品・制御システムのシステム信頼性に関する現状を鑑み、産業界におけるシステム高信頼に関する知見を集積し、将来に向けたシステム信頼性向上に関する技術的な布石を打ち、その結果としてシステム信頼性に関する社会的な認識レベルを上げていくことを目的に、「製品・制御システム高信頼化部会」とその傘下の下記３つの WG を設置し、産学の有識者を交えた議論を進めた（図１）。

(1) 未然防止知識 WG：製品・制御システムの障害を未然に防止するためのノウハウや知見を収集分析し、産業界で共通利用できるよう教訓化する。

(2) 障害事例検証 WG：製品・制御システムにおけるシステム障害に関する事例研究を通して、システム障害発生時の対処法や障害要因分析の手法を整理する。

(3) 障害原因診断 WG：製品・制御システムに障害事象が発生したときに、ソフトウェア面の原因を、モデルベースアプローチ、システムズエンジニアリングに基づき迅速かつ的確に、更に透明性・客観性を確保しつつ指摘できるようにする。事後 V&V（Verification and Validation）として体系化し、人材育成につなげる。

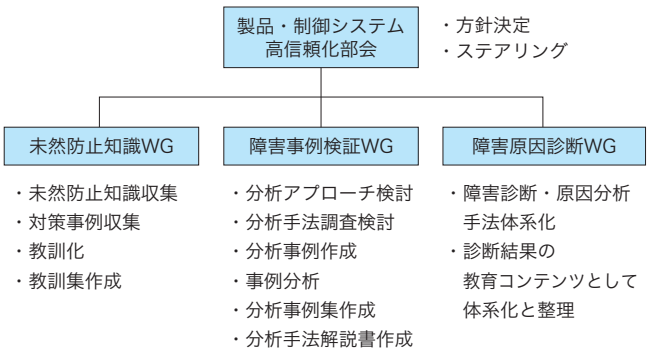

図１　製品・制御システムに関する部会・WG 構成

【脚注】
※１　http://www.ipa.go.jp/sec/reports/20150327_2.html
※２　http://www.ipa.go.jp/sec/reports/20150330.html
※３　http://www.ipa.go.jp/sec/reports/20150327_2.html
※４　http://www.ipa.go.jp/sec/reports/20150331_2.html

2 障害対策事例の収集と教訓集・対策事例集作成

2.1 背景と狙い

2013 年度に引き続き、産業界で実際に活用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知識をもとに、製品・制御システムの障害から得られた知見やノウハウを抽象化、一般化することによって、組込みシステム開発企業において幅広く利活用できるための『智恵』として教訓を作成し、前年度に公開した教訓集を改訂した。併せて教訓を実践するための対策の事例集を改訂した。

2.2 教訓の分類

(1) 観点による分類

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、展開するなどといった、障害を未然防止化するための取り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、各事例の障害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。そこで事例を抽象化することにより未然防止の教訓として自社・自部門製品で適用する際のトリガーとなるよう、28 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（図２、図３）。また開発現場での活用方法についても参考例として記載した。

図２　直接原因観点マップ

図３　未然防止観点マップ

(2) 教訓と開発工程による分類

開発工程においてどのような対策を施せば、教訓が解決しようとしている問題を未然防止できるのかという観点から解決策を整理した。プロセスモデルは「【改訂版】組込みソフトウェア向け開発プロセスガイド（ESPR Ver.2.0）」のモデルを採用した。また、直接開発にかかわるシステム・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）も対象とし、ESPR では定義されていない教育に関する工程も工程別未然防止知識の一部として採用した。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も、切り出して記載した。

教訓と対策が必要な工程との対応例を表 1 に示す。

3 障害分析手法事例解説書

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。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、表にまとめると共に、作業の流れをフローで示した。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。開発現場で役立つ知識の提供を目指して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、事例に沿った形で解説した。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、より深い分析を行うため、障害事例として本書内で公開した情報では、不明情報を創作して補って具体的に障害分析を行った。分析に際して HOW だけでなく WHY を記載することにより理解を容易にしている。

また、分析手法の解説とは別に、分析結果を再発防止につなげる取り組みの事例も紹介している。

4 製品・制御システム高信頼化のための行動指針

重要インフラを支える製品・制御システムにおいて求められる信頼性を発揮するためにシステムのライフサイクル（企画・設計・開発・保守・運用）全体を通して経営層及び開発責任者、品質責任者が遵守すべき事項を行動指針として取りまとめた。

主な特徴は以下の通り。

・ 重要インフラなどの製品・制御システムの開発企業の

表1　教訓一覧と対策が必要な工程との対応例

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など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、組み合わ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、フリーズなど）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，手順など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	遠隔地など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	データベースなど COTS 製品のバージョン、動作仕様の相違などの情報が関係者にタイムリーに参照できるようにする							○	○	○	○

有識者・専門家と組込み系ソフトウェア工学の学識者による「製品・制御システム高信頼化部会」にて検討・整理し、実際に産業界で活動している方の声を反映。

- IT サービスと異なる製品・制御システムに特有の状況に鑑みた技術的及び事業的特性に配慮。
- 経営層及び開発責任者、品質責任者が取り組むべきエンジニアリング・コンピテンシーを記述。

本指針が製品・制御システムの関係者に積極的に活用され、信頼性にかかわるコンピテンシーの維持向上が図られ、社会経済活動全体の信頼性向上、ひいては安心・安全な社会生活の実現に資することを期待している。

5 障害原因診断手法

5.1 背景と狙い

複数のシステムが連携することで、システム全体が大規模・複雑になり、システムを管理・操作する人とシステムの関係も複雑になる傾向がある。そのため、大規模・複雑なシステムで、人とシステムまたは複数システム間に起因する複合的な要因による障害が発生すると、原因を見つけ出すことは困難であり、その影響も広範囲かつ深刻になる。

また、システムの制御を担う要素が、ハードウェアから、制御ソフトなどソフトウェアを主体としたものに替わって

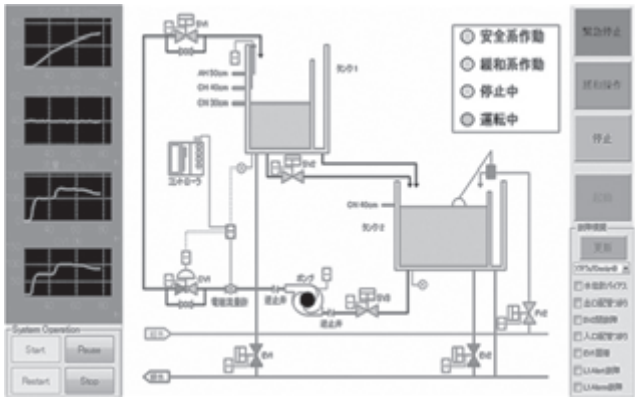

図４　化学プラントシミュレータの表示画面

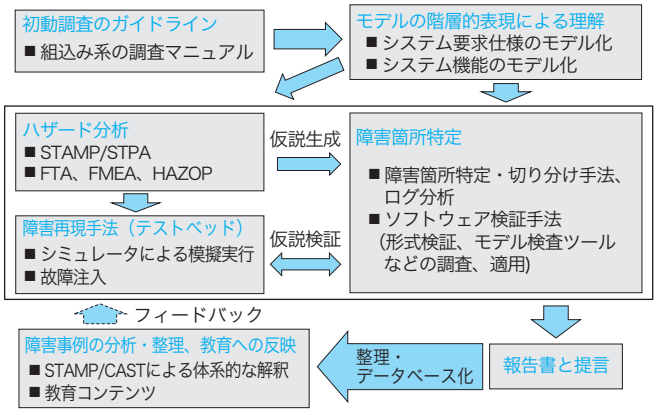

図５　事後 V&V 体系

きており、障害発生時のシステムの診断や原因の分析と対策も、従来の物理的要因中心の視点から、ソフトウェアやシステム中心のシステムズエンジニアリングに基づいた視点に変える必要がある。

そこで、従来の方法よりも容易に障害診断を可能にするため①障害発生後の初動調査と情報収集の手法、②人為的な要因も含め、システムとソフトウェアのどこに原因があるかを見極める障害診断手法、③障害原因となるソフトウェア制御ロジックの不具合を見極める形式検証手法やテスト手法を組み合わせて事後 V&V（Verification and Validation）として体系化し（図 5）、「大規模・複雑化した組込みシステムのための障害診断手法～モデルベースアプローチによる事後 V&V の提案～」として公開した。

5.2 事後 V&V の特徴

モデルベース手法をベースに、対象システムを故障原因仮説に沿って動作をシミュレートすることにより障害を再現させ、診断を行う。

これにより、以下の点が期待できる。

- システム全体の振る舞いを確認しながら分析できるため、原因個所の特定がしやすく、分析に要する作業時間の短縮が期待できる。また、類似の原因による障害の再発防止にもつながる。
- 実際のシステムを動作させることなくシミュレータ上で仮想的に動作検証ができるため、実際のシステムを一切毀損することなく障害を再現でき、通常は試験が困難な障害まで確認できる。
- システム実装前の動作検証にも使用できるため、障害の未然防止にもつながる。
- 人（操作員）の動作とシステム機能間の不整合などを含む複合要因に対する障害診断が可能。
- 実際のシステムをシミュレーションできるため、操作員向けのトレーニングや、非常事態を想定した訓練などへの活用が可能。

5.3 事後 V&V の有効性の確認

ソフトウェアの欠陥が関連している過去の障害事例を調査し、事後に検証を行うためのテストベッドの基本設計と、その実現例とすべくモデルベース開発ツールを用いて、化学プラントシミュレータを開発して検証を行った（図 4）。

その結果、人間の操作と制御ロジックの間に矛盾があるようなシステム障害の模擬や、ソフトウェアロジックの形式検証が可能であることを確認し、障害診断手法の有用性を確認した。

5.4 今後の取り組み

米国 MIT※5 の Nancy Leveson 教授が提唱し、欧米を中心に活用されている STAMP※6 のような新しい安全解析手法を取り入れるなど、実用性の向上を進める。更に、

- 障害が発生した際に、第三者として診断活動を行う役割の確立
- 障害の再発防止
- 透明性・客観性の確保
- 診断作業の迅速化を図るため、障害原因の診断結果を抽象化
- 教育コンテンツとして体系的に整理・蓄積

を進め新たなフレームワークとして普及展開を図っていく。

6 今後の課題

「教訓集」を始め「製品・制御システム高信頼化のための行動指針」、「障害分析手法事例解説書」、「大規模・複雑化した組込みシステムのための障害診断手法」について、現場での適用を促進するため、セミナーなどの開催と、活用を意識した質・量両面からのブラッシュアップを進めていく。また、現場からの評価の収集にも努め、今後の活動にフィードバックしていく。

今後は、行動指針に示された考え方にのっとり、各企業が自ら高信頼なものづくりを継続的に取り組んでいくための教材作成や教育の普及に向けた取り組みを推進していく。

【脚注】

※5　MIT（Massachusetts Institute of Technology）：マサチューセッツ工科大学

※6　STAMP（Systems-Theoretic Accident Model and Process）：システム理論に基づく事故モデル