論　文

不具合履歴に基づくソフトウェアIV&V活動の定量的見える化手法

松本　真佑[†a)]　上野　秀剛[†]　門田　暁人[†]　松本　健一[†]
片平　真史[††]　神武　直彦[††*]　宮本　祐子[††]　氏原　頌悟[††]
吉川　茂雄[†††]

Quantitatively Characterizing Software IV&V Activities Based on Defect History

Shinsuke MATSUMOTO[†a)], Hidetake UWANO[†], Akito MONDEN[†],
Ken-ichi MATSUMOTO[†], Masafumi KATAHIRA[††], Naohiko KOHTAKE[††*],
Yuko MIYAMOTO[††], Shohgo UJIHARA[††], and Shigeo YOSHIKAWA[†††]

あらまし　本論文では高信頼性ソフトウェア開発現場で用いられる Independent Verification & Validation (IV&V) 活動プロセスの評価を目的として，IV&V 活動の「見える化」手法の提案を行う．ここでいう「見える化」とは，活動の成果を分かりやすく整理することで共通認識の形成，定量的な議論，改善点の洗い出し等を可能とすることである．提案手法では IV&V 活動により検出された不具合を三つの機能的な種別，及び Verification と Validation の種別という独立した二つの属性を用いて 6 種類に分類し，理解容易な形で可視化する．提案手法を用いることで，従来の不具合分類方法で指摘されている分類の困難さの排除，及び IV&V 活動の評価に必須となる Verification と Validation の両面からの活動の評価が可能となる．宇宙航空研究開発機構 (JAXA) において実施された IV&V 活動の不具合検出履歴データを用いたケーススタディにより，提案手法の有効性を確認するとともに，対象とした IV&V プロセスにおける改善点を発見することができた．

キーワード　IV&V，見える化，不具合検出履歴

1. まえがき

人工衛星や探査機，及びそれらを地上で運用・監視するシステムに用いられる組込みソフトウェアには，極めて高い信頼性と安全性が求められる．このような高信頼性ソフトウェアの開発現場では，Software Independent Verification and Validation（ソフトウェア独立検証及び有効性確認：SW IV&V，以降 IV&V と略す）が実施されている．IV&V とは，ソフトウェアの開発組織や開発委託組織から独立した組織がドキュメントやソースコードなどの成果物について品質保証及び妥当性確認を行うことにより，ソフトウェアの信頼性を向上させる技術や枠組みのことである [1], [2]．独立した組織が成果物の品質保証及び妥当性確認を実施することで，客観的かつ幅広い視点での評価が期待される．アメリカ航空宇宙局（NASA）では IV&V を用いたソフトウェアの評価活動（IV&V 活動）を行う組織が設置されており，多くの宇宙機ソフトウェアの開発プロジェクトで IV&V が実施されている．航空システムや軍事システムなどに搭載される組込みソフトウェアにおいても IV&V が実施されており [2], [3]，高信頼性の確保のために利用されている．

IV&V は独立した組織にて実施され，そのプロセスが外部から見えないため，何らかの方法でその活動を定量的に「見える化」し，現状把握や評価を行うことが必要である．ここでいう「見える化」とは，活動の成果を分かりやすく整理することで共通認識の形成，定量的な議論，改善点の洗い出し等を可能とすることである．このような見える化は IV&V に限らずソフ

† 奈良先端科学技術大学院大学情報科学研究科，生駒市
Graduate School of Information Science, Nara Institute of Science and Technology, 8916–5 Takayama, Ikoma-shi, 630–0192 Japan
†† 宇宙航空研究開発機構，つくば市
Japan Aerospace Exploration Agency, 2–1–1 Sengen, Tsukuba-shi, 305–8505 Japan
††† 有人宇宙システム株式会社，土浦市
Japan Manned Space Systems Corporation, 1–1–26 Kawaguchi, Tsuchiura-shi, 300–0033 Japan
* 現在，慶應義塾大学
a) E-mail: shinsuke-m@is.naist.jp

トウェア開発全般において重要であり，一般的な品質保証活動である文書レビューやテストにおいても，不具合の数や種類に基づいて活動結果を見える化する手法が提案されている [4], [5]．これらの研究は，検出された不具合を分類して整理することで，検出数の足りない（若しくは多すぎる）不具合種類やモジュールを特定し，改善すべきプロセスを明らかにすることを目的としている．よく用いられている不具合の分類方法として IEEE Std 1044-1993 [6] や Chillarege らの提案する Orthogonal Defect Classification (ODC) [7], 実行可能なコードに含まれる不具合の分類を用いるために ODC を詳細化した ODC-CC [8] 等がある．

しかし，従来の不具合分類法を，IV&V で検出された不具合の分類に適用することは適当でない．IV&V は，成果物に含まれる不具合を検出するという点ではレビューやテストと同じであるが，より高い信頼性を得るために，複数の不具合検出観点（Lessons Learned の反映，上位文書整合性，ハザード関連機能の識別など）及び複数の不具合検出手法（文書レビュー，モデル検査，プログラム解析など）が採用されており，これらを考慮したよりきめ細かい評価が求められる．加えて，IV&V では，Verification によってドキュメントやソースコードなどの成果物間の整合性を確保し，Validation によってシステム要求との整合性を確保するという 2 面性をもつため，これらの両面からの結果の整理が必要である．また，従来の不具合分類法は，各分類項目の独立性が弱く，検出した不具合を適切に分類することが難しいという批判もあり [9]，分類項目の独立性を確保することが必要となる．

本論文では，IV&V 活動の見える化に求められる要件を整理し，その結果に基づいて，IV&V 活動により検出された不具合の分類方法を提案する．提案手法では不具合を三つの機能的な種別，及び Verification と Validation の種別という独立した二つの属性を用いて 6 種類に分類する．従来手法において多くの種類に分割されていた不具合の種類を三つの機能的な種類にまとめることで，本手法ではそれぞれの不具合種類を独立させ，分類を容易にしている．また，本論文では不具合検出手法や検出観点ごとに，どのような属性の不具合を検出しているかを容易に把握するための可視化手法についても提案する．不具合の分類結果を可視化することで，不具合の検出傾向の把握の容易化，他プロジェクトや他工程との検出傾向の比較の容易化が期待できる．

提案手法のケーススタディとして，本論文では宇宙航空研究開発機構 (JAXA) においてある宇宙機ソフトウェア開発プロジェクトのソフトウェア要求分析フェーズで実施された IV&V 活動の不具合検出履歴データを用い，提案手法の有効性を確認する．

以降，**2.** で IV&V についての説明を行い，**3.** で提案する分類方法と可視化方法の詳細について述べる．**4.** では JAXA で実施された IV&V 活動を題材としたケーススタディについて説明し，**5.** でケーススタディの結果と考察について述べる．最後に **6.** で本論文のまとめと今後の課題について述べる．

2. Independent Verification and Validation

2.1 IV&V の概要

IV&V とは品質保証における基本的な考え方の一つである Verification（検証）と Validation（妥当性確認）の二つの観点から，開発組織や開発委託組織から独立した組織がプロダクトを評価する技術や枠組みのことである [1], [2]．IV&V の Independent（独立性）については IEEE Std 1012-2004 により，技術的独立，管理的独立，財政的独立の三つの点が定められている [10], [11]．開発組織から技術的に独立した組織が IV&V 活動を実施することで，開発技術に縛られない客観的な視点での評価が可能となり，管理的・財政的な独立性を保つことで，組織や管理体制に縛られない視点からの評価が可能となる．

ソフトウェアの開発サイクルに対する Verification と Validation の関係を図 1 に示す．Verification とは開発の各工程において，より上位の工程の成果物（上位文書）に基づいて具現化・詳細化が正しく行われているかどうかを精査し誤りを取り除く活動のことであり，主として信頼性の確保に寄与する．ここでの信頼性は，ISO-9126-1 よりソフトウェア故障が発生しないで持続的に動作する性質や度合と定義する．Verification によって，開発の各工程において上位文書と下位文書の間で正しく整合性を確保，すなわち抽象的なシステムの振舞いが完全に具現化されていれば，システムとしては故障なく正しい動作をするといえる．一方，Validation とはシステムに求められた最上位のシステム要求について，各工程で作成された成果物が機能や品質について十分に満たしているかを確認する活動のことであり，主として安全性の寄与に貢献する．ここでの安全性は，ISO-9126-4 より動作中に

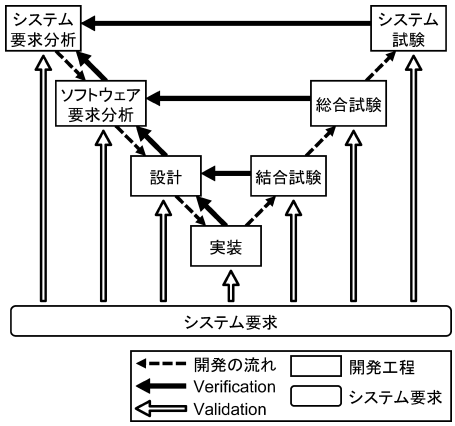


図 1　ソフトウェア開発プロセスに対する Verification & Validation 活動
Fig. 1　Verification and Validation activities for software development process.

ハザードを誘発しない性質や度合と定義する．JAXA における宇宙機のシステム開発では，システムに対する最上位の要求を「システムの振舞いや動作が人的喪失や宇宙機の喪失につながらないこと」と定義しており，Validation 活動を正しく実施することにより高い安全性を確保できるといえる．ソフトウェアの高い信頼性と安全性を達成するためには，開発サイクルの上流から下流までの各工程に対して，Verification と Validation の両方の観点から評価を行うことが重要である．

また IV&V 活動では様々な視点からソフトウェアの品質を確認するために，複数の品質評価手法が用いられる．IV&V において一般的に用いられる評価手法の一例を以下に示す．

- チェックリストを用いたレビュー
- FTA，FMEA に基づくレビュー
- モデルチェッキング
- シミュレーション
- コードレビュー
- テスト

このうち，例えばレビューでは開発工程全般において成果物を評価することが可能であるが，多くの工数を必要とするため網羅的な検証を行うことが困難である．一方，モデル検査の場合，状態遷移や条件式を厳密かつ網羅的に検証できるが適用可能な成果物や工程が限定的である．このように各評価手法は検出対象とする不具合の種類や実施可能な工程が異なっており，

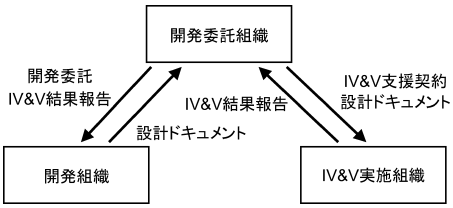


図 2　IV&V 活動の実施体制
Fig. 2　Structure of IV&V activity.

網羅的でかつ効率的な IV&V 活動の実施のためには，対象プロジェクトや実施工程に適した手法を選択することが必須である．

典型的な IV&V 活動の実施体制を図 2 に示す．IV&V 実施組織は開発委託組織（JAXA）を通じて開発組織（メーカ）にて作成された対象ドキュメントを受け取り不具合(注1)を検出する．検出された不具合は IV&V 結果報告として開発組織に渡され，開発組織がそれぞれの報告内容に対してシステム開発におけるリスクの大きさを分析し，修正が必要なものについては開発組織により適宜修正される．

2.2　IV&V 活動の評価における課題

IV&V 活動における効率性・有効性の評価においては三つの課題が存在する．

2.2.1　不具合の機能的な分類

ソフトウェアの検証プロセスの改善やテスト工程の終了，出荷の可否の判断に役立てるために，検出された不具合を分類し検出数の偏りや網羅性を分析する研究が従来から行われている [8]．不具合分類の代表的なものには ODC [7] や階層的欠陥分類法 [12]，Beizer による不具合分類 [13] がある．しかし，従来提案されている不具合の機能的な分類方法は分類の粒度が小さく，また各項目が必ずしも独立していない [9]．一例として ODC で提案されている分類項目を以下に示す．

- Function
- Interface
- Checking
- Assignment
- Timing/Serialization
- Build/Package/Merge
- Documentation
- Algorithm

(注1)：IV&V ではソフトウェアの不具合に関する指摘のみならず，不具合の可能性となる問題点や課題についても指摘が行われるが，本論文ではこれらをまとめて不具合と呼ぶ．

例えば「ソフトウェア要求仕様のデータ設計が，システム設計書やインタフェース定義書と整合していない」という不具合を ODC の分類方法に従い，一つの項目に分類する場合，Interface と Documentation のどちらにも分類することが可能である．したがって，分類を行う作業者の解釈によって分類結果が変化し，妥当な分類結果が得られない可能性がある．

また，日本の大手ソフトウェアベンダで用いられている不具合分類において，検出された不具合のうち約30%が「その他」に分類されているという事例も報告されている [14]．このような「その他」に分類されやすい分類手法は分類項目が多く，検出した不具合がどの項目に分類されるかを判断しにくいと考えられる．したがって IV&V 活動によって検出された不具合の種類を適切に分析するためには，開発者が容易に分類することが可能でかつ分類項目間の独立性が確保された分類項目の作成が必要である．

2.2.2 Verification と Validation の分類

IV&V 活動によって検出される不具合は，Verification と Validation のどちらの観点によって検出されたかを区別する必要がある．IV&V 活動では Verification によって成果物間の整合性を評価し，Validation によって要求との整合性を評価することで対象ソフトウェアの信頼性と安全性を確保する．したがって，IV&V 活動によって不具合が正しく検出されているか評価するためには，Verification と Validation の両面において十分に不具合が検出されているかを確認する必要がある．

2.2.3 不具合検出手法が適切に選択されているかの識別

IV&V 活動は様々な不具合の検出観点や検出手法が IV&V 実施組織によって定義されており，IV&V 活動の実施の際に対象とする成果物や工程に適した組合せが選択され実施される．検出手法や検出観点の組合せの妥当性や，検出手法ごとの効果を評価するためには検出手法や検出観点ごとの検出傾向についての分析が必須である．検出手法や検出観点に加え，不具合種類の分類と Verification と Validation の分類を行う場合，三つの軸が必要であり単一な表では表すことが難しい．IV&V 活動全体としての検出傾向に限らず，個々の検出手法や観点の検出傾向について分析可能な方法が必要である．

3. 提案手法

3.1 概　　要

提案分類方法では，IV&V によって検出された不具合を三つの機能的な種別，及び Verification と Validation の種別という直交する二つの属性を用いて 6 種類に分類する．不具合の機能的な種別を互いに独立性の高い三つに集約することで，不具合の分類を容易にするとともに，検出された不具合の網羅性の評価を可能としている．また，Verification に関する不具合と Validation に関する不具合を区別することで，信頼性と安全性の両面からの評価を可能としている（**3.2**）．更に本論文では上記の不具合の分類方法に加え，分類された不具合の可視化方法の提案を行う（**3.3**）．

3.2 不具合分類

3.2.1 不具合の機能的な種別

本手法ではシステム全体を入力，処理，出力の三つから構成されるというモデル（IPO：Input-Process-Output モデル）に基づき，分類項目を機能，インタフェース，シナリオという独立性の高い三つに集約することを考える．提案分類方法では「処理」に問題がある場合を機能の不具合,「入力」と「出力」のいずれかに問題がある場合をインタフェースの不具合とみなし，処理や入出力のレベルにまで具体化されていない抽象的なシステムの振舞いや，シナリオに関する問題をシナリオの不具合とみなす．提案分類法と IPO モデルの対応を図 3 に示す．

IPO モデルはシステム全体を三つの要素で大別するモデルであり，このモデルに基づいた分類項目に加え，シナリオなどの抽象的な記述を包括する分類項目を設けることで高い網羅性を確保できると考えられる．ま

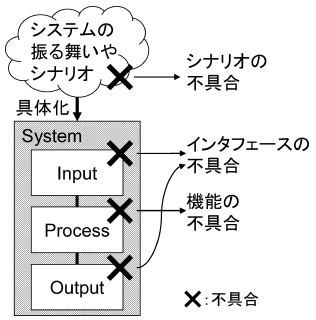


図 3　機能的な種別項目と IPO モデルの対応
Fig. 3　IPO model and our proposed functional types of defect.

た，IPO モデルは抽象度が高いにもかかわらず，個々の要素の区切りが直感的でかつ明確であるとうい点から，少なくとも従来の分類項目よりも高い独立性を確保できると考えられる．これら分類項目の網羅性と分類項目間の独立性を確保することにより，分類者による解釈の違いを減らすことができる可能性がある．三つの分類項目について以下で具体的に説明する．

- **機能**

システムの「処理」に関する不具合を指す．システムやモジュールが正しい入力に対して，正しく結果を出力できない場合や，入力データと出力データは定義されているがその処理方法が定義されていない場合，「機能」の不具合に分類される．

- **インタフェース**

システムの「入出力」に関する不具合である．ソフトウェア間やハードウェア/ソフトウェア間のインタフェースについての矛盾や不整合に関する不具合は「インタフェース」の不具合に分類される．

- **シナリオ**

システムの「シナリオ」に関する不具合である．シナリオに関する矛盾や，外部環境の変化やハードウェア障害などによって引き起こされるシステム環境の変化についての想定不足に対する不具合は「シナリオ」の不具合に分類される．

3. 2. 2 Verification/Validation の種別

提案手法では不具合が Verification/Validation のどちらに関するものかを区別し分類を行う．以下で Verification と Validation について説明する．

- **Verification（検証）**

成果物間の整合性に関する不具合を指す．例えば成果物間での記述の不一致や不整合，上位の成果物に記述されている項目の下位の成果物への展開漏れは Verification に分類される．

- **Validation（妥当性確認）**

システムに対する要求についての不具合を指す．例えば，設計された機能がシステムに対する要求を満たしていない場合や，必要な機能に関する記述が成果物中にない場合など，システム要求の「抜け」に関する不具合は Validation に分類される．

3. 3 分類結果の可視化

本論文では **3. 2** で述べた不具合の分類方法の提案に加え，検出手法や検出観点の組合せの評価を目的とした分類結果の可視化方法を提案する．図 4 は提案する可視化方法の基本となる分類枠である．提案する分類方法は，不具合の機能的な種別と Verification/Validation の種別が互いに独立しており，それぞれの種別がもつ分類項目についても高い独立性が確保されているため，図 4 のように表現することが可能である．この図に対して個々の不具合を該当する枠内にプロットすることで，不具合全体の分布を表現できる．更に，図 5 に示すように各不具合を検出手法（若しくは検出観点）ごとに異なる記号を用いてプロットすることで，検出手法の組合せの効果と個々の検出手法の効果を一元的に表示することができる．また分類結果を可視化することで，容易に検出傾向が理解できるようになるほか，他の評価工程や他のプロジェクトとの比較の容易化が期待できる．

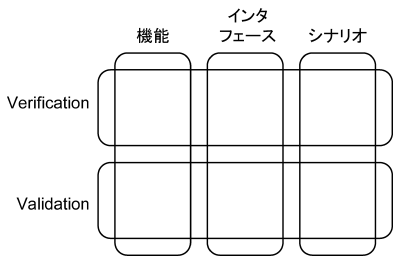


図 4 提案分類方法の可視化
Fig. 4 A visualization of proposed defect classification.

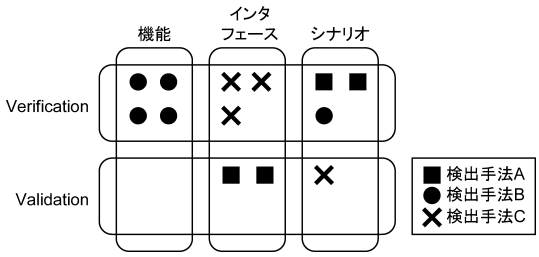


図 5 可視化例
Fig. 5 An example of visualization.

4. ケーススタディ

4. 1 分析対象データ

本論文で分析対象としたデータは，ある宇宙機ソフトウェア開発プロジェクトのソフトウェア要求分析フェーズにおいて JAXA で実施された IV&V 活動の際に収集された不具合の検出履歴である．

対象データには IV&V 活動によって検出された 49 件の不具合についての概要や，検出時に用いられた不具合検出観点，及び検出手法，また不具合の指摘に対する開発組織の回答，及びその回答に対する判定など

表 1　メトリックス一覧
Table 1　A list of metrics used in the analysis.

メトリックス名	概要	例
ID	各検出不具合を一意に識別するための番号.	1, 2, 3
不具合検出観点	不具合を発見する際に用いられていた観点.	Lessons Learned の反映.
不具合検出手法	不具合を発見する際に用いられていた手法.	ボイジャーガリレオチェックリストによる文書レビュー.
対象成果物	不具合が含まれていた成果物.	ソフトウェア設計仕様書.
不具合概要	発見された不具合の概要.	一時的な電源瞬断に対する対策が不明.
不具合原因	発見された不具合の原因となっている事象.	電源瞬断に対する異常対策が運用定義で考慮されていない.
リスク概要	不具合によって存在していると思われるリスクの概要.	正常なデータが得られない．また，機器に対して誤った制御が行われるため，機器が破壊される.
リスクの有無	ソフトウェア開発組織に対して検出不具合についての問合せを行うことで決定したリスクの有無.	リスクなし.

が記載されている．対象データに含まれている項目のうち，分析に利用した項目の一覧を表 1 に示す．本論文では 49 件のデータから，記述漏れや不明りょうな点があった 6 件を除いた 43 件を対象として提案手法を用いて分析を行った．

4.2　分 析 手 順

分析の手順を以下に示す．

（1） 不具合の機能的な分類：各不具合を不具合概要，不具合原因及びリスクの記述を参考に，機能，インタフェース，シナリオのいずれかに分類する．

（2） Verification/Validation の分類：各不具合を不具合概要と不具合原因の記述を参考に Verification と Validation のいずれかに分類する．

（3） 分類結果の可視化：手順 (1)，(2) により分類された各不具合を図 4 にプロットする．本論文では，(a) 不具合検出観点，(b) 不具合検出手法，(c) リスクの有無の三つの観点から分析を行うために，分析観点（**4.3**）ごとに複数の記号を用意し，三つの観点に対応したプロットを行う．

4.3　分析の観点

本論文では三つの項目（不具合検出観点，不具合検出手法，リスクの有無）を分析の観点として IV&V 活動の評価を行う．

4.3.1　分析観点 1：不具合検出観点

JAXA における IV&V 活動では定められた不具合検出観点に基づいて対象ドキュメントの評価作業が行われる．定められた観点に従って成果物の評価を行うことで，効率的な IV&V 活動が可能となる．ただし，観点ごとの検出不具合は必ずしも独立しているとは限らず，効率的な観点を適切に選ぶ必要がある．不具合検出観点が適切に選択されているかを調べるために，各不具合検出観点についての検出傾向について分析を行う．また，不具合検出観点はそれぞれある特定の種類の不具合を検出することを目的としており，検出が期待される不具合の種類と異なった種類の不具合を検出することは検出効率の低下を招く可能性がある．そのため，実際に検出された不具合の種類と期待される効果の不一致についても分析を行う．具体的な不具合検出観点は以下のとおりである．

- Lessons Learned の反映：システム仕様に規定されている機能や性能，及びインタフェース仕様に，過去の宇宙機プロジェクトで発生した事故や検出された不具合から得られた Lessons Learned が網羅的に反映されているかを評価する．主に宇宙機開発で用いられるボイジャーガリレオチェックリスト [15] を用いた文書レビューや，タイミングチャートの状態遷移モデルを用いたシミュレーションなどによって評価が行われる．
- 上位文書整合性確認：評価対象成果物の上位の成果物に記載される事柄が評価対象にも記載されているか，その記述内容に矛盾がなく一致しているかについて評価する．手法としては，要件・構成管理ツールである DOORS [16] を用いた文書レビューなどが行われる．
- ハザード関連機能の識別：宇宙機の喪失といった重大な障害にかかわるソフトウェア機能，及び障害制御にかかわるソフトウェア機能が網羅的に識別されているか，またその機能の妥当性について評価を行う．主に，FTA（Fault Tree Analysis）[17] や FMEA（Failure Mode and Effects Analysis）[18] を用いた文書レビューが行われる．
- インタフェースの妥当性確認：ソフトウェア間

の入出力タイミングや送受信時におけるプロトコルの妥当性，共通データの名称といったインタフェースに関する評価を行う．手法としては主に文書レビューが用いられるが，未定義のデータ受信といった想定外の入力に対する処理についてはボイジャーガリレオチェックリストが用いられる．

- その他：本ケーススタディでは，検出不具合数が少なかった完全性評価，一貫性評価，タイミング評価の三つの不具合検出観点を「その他」としてまとめた．完全性評価では，状態遷移の条件のすべての組合せに対して遷移先の抜けについて評価を行う．一貫性評価では一つの遷移条件に対して複数の遷移先が定義されていないかを評価する．タイミング評価では，データの送受信や割込みの処理などのタイミングに関する評価を行う．いずれの観点も手法としては文書レビューやモデル検査が用いられる．

4.3.2 分析観点 2：不具合検出手法

各不具合検出観点を達成するための不具合検出手法としては，大きく分けて文書レビューとモデル検査の 2 種類の手法が用いられている．文書レビューでは評価対象成果物の上位の成果物や，チェックリストなどの資料を用いて行われる．モデル検査としてはトレーサビリティ解析ツールや状態遷移モデルの解析ツールなどが用いられる．JAXA における IV&V では，文書レビューは成果物の評価に広く用いられているがモデル検査は比較的導入実績が浅く，その実施効果については不明な点が多い．各不具合検出手法がどのような不具合を検出しているかを分析するために，各不具合検出手法についての検出傾向について分析を行う．

4.3.3 分析観点 3：リスクの有無

IV&V 活動では開発組織とは独立した組織が評価を行うため，不具合検出履歴の中には開発者に対する確認事項のような実際に修正を必要としない指摘も含まれる．本論文では，このような修正を必要としない指摘をリスクなし不具合，修正を要する指摘をリスクあり不具合と呼ぶ．効率的な IV&V の実施のためには，より多くのリスクあり不具合を検出する必要がある．IV&V 活動の効率性を評価するために，リスクあり不具合の検出傾向を分析する．

5. 結果と考察

5.1 分析観点 1：不具合検出観点

不具合検出観点ごとの検出数を表 2 に，検出観点ごとの検出不具合の可視化結果を図 6 に示す．JAXA で

表 2 各不具合検出観点における不具合の検出数
Table 2 The number of detected defects in each perspective.

不具合検出観点	#	%
Lessons Learned の反映	14	32.6%
上位文書整合性確認	8	18.6%
ハザード関連機能の識別	9	20.9%
インタフェースの妥当性確認	6	14.0%
その他	6	14.0%
合計	43	100.0%

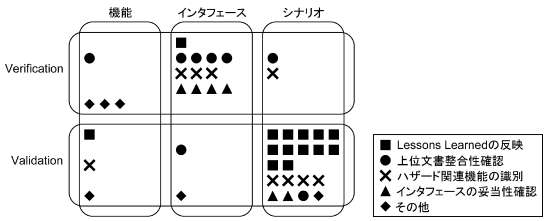


図 6 不具合検出観点ごとに集計された不具合の分布
Fig. 6 Classification of defects by detection perspective.

定義されている IV&V ガイドラインより抽出した，観点ごとの検出が期待される不具合の種類を図 7 に示す．この IV&V ガイドラインは JAXA で実施する IV&V 活動についての実施要綱が記載されており，IV&V 活動の実施体制や実施方法，評価観点や評価手法の概要などが記述されている．図 6 と図 7 を比較をすることで，期待される検出不具合と実際に検出された不具合の傾向のずれを分析することが可能である．

観点「Lessons Learned の反映」で検出された不具合は Validation に関する不具合が 14 個中 13 個と多く（以下，13/14 のように表す），シナリオに関する不具合も多く検出されている（12/14）．図 7 に示すとおり Lessons Learned の反映により検出が期待される不具合は，要求誤りや運用シナリオの矛盾といった Validation とシナリオに関するものであり，実際に検出された不具合とのずれは小さく，期待どおりの不具合が適切に検出されていることが分かる．このことから検出観点に沿った適切な IV&V 活動が実施されていると考えられる．

Lessons Learned の反映と同様に，観点「上位文書整合性確認」と「インタフェースの妥当性確認」についても期待どおりの不具合が検出されており，適切に評価が行われているといえる．

一方，観点「ハザード関連機能の識別」により検出された不具合の約半数は，検出が期待されない Verification に関する不具合であった（4/9）．これら

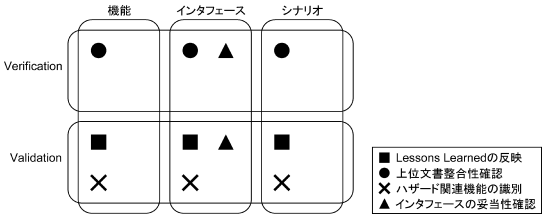


図 7　各不具合検出観点により検出が期待される不具合の種類
Fig. 7　Assumed defect classification in each detection perspective.

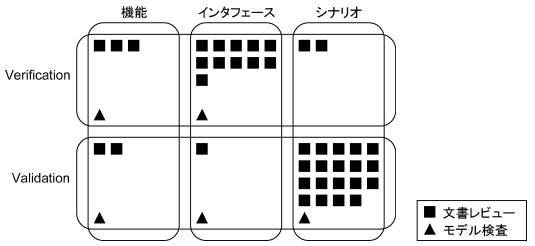


図 8　不具合検出手法ごとに集計された不具合の分布
Fig. 8　Classification of defects by detection technique.

表 3　各不具合検出手法における不具合の検出数
Table 3　The number of detected defects in each method.

不具合検出手法	#	%
文書レビュー	38	88.4%
モデル検査	5	11.6%
合計	43	100.0%

Verification の不具合の内容について確認したところ，データ名称の不整合といった検出目的からずれた不具合が含まれていた（3/9）．評価を実施した順番にも依存するが，これらの不具合はインタフェースの整合性を確認する観点で検出されるべきである．例えばハザード関連機能の識別を実施する前にインタフェースに関する不具合検出観点が実施されていた場合，これはインタフェースの妥当性確認で見逃された不具合であるといえる．このような不具合検出観点からずれた検出を分析し，IV&V 実施プロセスに反映することで効率的な IV&V の実施が可能になると考えられる．

検出の数という面では，Lessons Learned の反映とハザード関連機能の識別で多くの不具合が検出されており，二つの観点で全検出の約半数を占めている（23/43）．このことから，分析対象のプロジェクトにおいては多くの不具合を検出するという面では Lessons Learned の反映とハザード関連機能の識別の二つの観点が有効であると考えられる．

不具合の種類についての検出網羅性という点では，機能に関する不具合が他の種類の不具合と比べて少なかった（機能：7，インタフェース：14，シナリオ：22）．考えられる原因としては，インタフェースやシナリオの不具合の検出と比較して，該当機能に関してより深いドメイン知識が必要となる一方で，IV&V 実施組織は開発組織ではないことから，評価者のドメイン知識が不足していた可能性がある．機能に関する不具合を効果的に検出するためには，ドメイン知識の豊富な技術者を IV&V 活動に参加させることが重要と思われる．

5.2　分析観点 2：不具合検出手法

検出された不具合を不具合検出手法ごとに集計した分類結果を図 8 に，個々の検出手法の検出数を表 3 に示す．評価手法としては主として文書レビューが用いられており，ほとんどの不具合は文書レビューにより検出されている（38/43）．

モデル検査によって検出された不具合 5 個のうち，3 個が Validation に関するものであった．モデル検査では，評価対象とする機能を有限状態機械としてモデル化し，時相論理式を用いて数理的に検証する [19] ため，本来は Verification に関する不具合（遷移条件の矛盾や抜け，上位成果物との矛盾など）を網羅的に発見することが期待されている．一方で，要求や運用シナリオなどの厳密に定義することができないものをモデル化することは難しく，Validation に関する不具合の発見は必ずしも期待されていない．本ケーススタディにおいて，モデル検査により Validation の不具合を多く検出していた原因としては，これらの不具合が作成したモデルにより検出されたものではなく，モデル作成時に要求仕様書及び設計書を精読した際に，レビュアーによってその矛盾や誤りを発見したものであったと考えられる．このことから，モデル検査では作成したモデルによって Verification に関する不具合を発見できるのみならず，モデルの作成時においても Validation に関する不具合が検出されるという副次的な効果が期待できるといえる．

5.3　分析観点 3：リスクの有無

検出された不具合をリスクの有無ごとに集計した分類結果を図 9 に示す．リスクの有無ごとの検出数を表 4 に示す．

リスクを含む不具合が多く存在していたのは Veri-

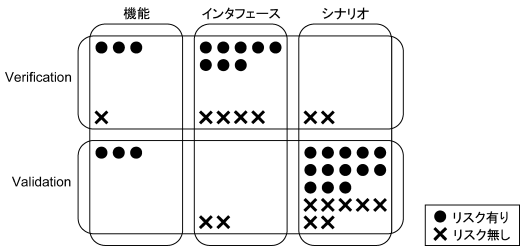


図 9　リスクの有無ごとに集計された不具合の分布
Fig. 9　Risk presence/absence of each defect.

表 4　リスクの有無ごとの不具合の検出数
Table 4　The number of detected defects of risk presence/absence.

リスクの有無	#	%
リスク有り	27	62.8%
リスク無し	16	37.2%
合計	43	100.0%

fication かつインタフェースと Validation かつシナリオに関する不具合であった．リスクを含む不具合は実際に修正を要する不具合のことであり，これらの種類の不具合を適切に検出することが本対象システムにおける IV&V 活動の効率化につながると考えられる．

一方で Verification かつシナリオ（上段右）と Validation かつインタフェース（下段中央）の不具合は検出数が少なく，リスクを含む不具合が存在しない．Verification かつシナリオについては，運用手順などを想定して作成されるシナリオには上位の成果物が存在しないため，不具合を検出することが難しく，また検出されたとしてもリスクにつながるような不具合ではなかったと考えられる．一方，Validation かつインタフェースについては，要求などの上位の仕様書においては，インタフェースなどの具体的な仕様が記述されていないため，不具合数が少なかったと考えられる．つまり，これらの不具合はそもそも混入しにくくリスクを含む可能性も低いため，他の不具合を重点的に検出することで IV&V 活動の効率化につながると考えられる．

不具合の機能的な種別ごとに見ると，リスクあり不具合の割合はインタフェースが 57%（8/14），シナリオが 59%（13/22）とそれぞれ約半分を占めているが，機能は 86%（6/7）であり，他の不具合の種類に比べてリスクあり不具合が多い．このことは，本 IV&V 活動の対象システムでは，機能に関する不具合が修正を要する重要な不具合である場合が多かったことを示している．分析対象 IV&V 活動においては機能の不具合を検出目的とする観点が含まれていないため，このような観点を加えることで IV&V 活動の効率化とソフトウェアの高信頼化につながる可能性がある．

5.4　提案手法に対する考察

本節では **2.2** で述べた IV&V 活動の評価における課題について考察する．

不具合の機能的な種別について，項目数の集約と独立性の確保によりぶれの少ない分類が可能であるかを確認するために，2 名の大学院生を対象に提案分類方法と従来 JAXA で用いられていた分類法（分類項目：インタフェース，運用シナリオ，論理，タイミング，負荷，データ）を用いて，筆者らの分類結果との比較実験を行った．実験に用いたデータは本論文で利用した 43 件の不具合データの中から機密情報を含む不具合 8 件を除いた 35 件の不具合データである．結果としては，提案分類方法のぶれは平均 20%（7/35），従来手法は平均 51%（18/35）であり，提案手法を用いることで分類者間のぶれを抑えることが可能であることが明らかとなった．しかしながら，提案手法でもある程度のぶれが存在した．これは，被験者が学生であり対象システムに関するドメイン知識がないことによるものだと考えられるが，対象システムに関する共通認識をもつ現場の作業者であれば，よりぶれを抑えつつ分類できる可能性がある．付録の表 A.1 に各分類者のぶれの詳細を示す．従来方法のぶれの傾向として，データとインタフェースの間で分類結果の不一致が多く，負荷やタイミングと運用シナリオの間でも多くの不一致が存在した．一例として「負荷に起因する障害の対策シナリオがない」という不具合に対しては，負荷と運用シナリオのどちらかにぶれる傾向があった．これらの分類項目間は十分な独立性が確保できていない可能性がある．

不具合の機能的な種別とは独立に，Verification と Validation を分けて分類することで，本分析対象 IV&V 活動において，信頼性（Verification）と安全性（Validation）に関する不具合の両方を検出できていたことが確認できた．また，モデル検査では本来想定していない Verification の不具合の検出という副次的な効果が得られることが分かった．Verification と Validation を区別せずに IV&V 活動の評価を行った場合，検出不具合がどちらか一方に偏ってしまうケースを判断できず，適切に IV&V 活動が実施されているかを評価できない．すなわち，客観的に Verification

と Validation の両面からのプロダクトの評価を実施できるという IV&V の利点を十分に引き出せない可能性がある．

また，これらの分類結果を可視化することにより，IV&V 活動で検出された不具合の定量的な「見える化」が可能であることが確認できた．この見える化によって，不具合検出状況の定量的な把握や，他の IV&V 活動や他の評価工程との検出傾向の比較が容易に行えると考えられる．例えば，現行の IV&V 活動における検出状況と過去に実施された IV&V 活動での検出傾向を比較し，評価が不十分な点を洗い出すといった IV&V 活動の改善につながることが期待できる．

ただし，本ケーススタディでは不具合の検出数のみに基づいて分析を行っており，不具合の質や重要度という点については分析の対象外とした点に注意されたい．JAXA における IV&V 活動では重要度の低い不具合を多数見つけるよりも，極めて重要度の高い不具合 1 個を発見する方が有用であるとされている．これら不具合の重要度についても提示可能なように提案手法の拡張を行うことは今後の課題である．

6. むすび

本論文では IV&V 活動の評価手法の提案と，JAXA において実施された IV&V 活動を用いたケーススタディを行った．ケーススタディでは提案手法を用いて不具合を分類し，分類結果を不具合検出観点，不具合検出手法，リスクの有無の三つの観点から分析を行った．その結果，提案手法について以下のような知見が得られた．

- 分類項目の少なさと独立性の確保により，ぶれの少ない機能的な種別が可能である．
- 機能的な種別とは別に Verification と Validation を区別することで，信頼性と安全性の両面からの評価が可能である．
- IV&V 活動により検出された不具合の，定量的な見える化が可能である．

またケーススタディの題材となった JAXA における IV&V 活動に関して，次のような評価結果を得ることができた．

- 検出観点「ハザード関連機能の識別」において，検出が期待されないインタフェースに関する不具合が検出されていた．このような検出目的からずれた不具合を分析し，IV&V 実施プロセスに反映することで IV&V 活動の改善につながると考えられる．
- 不具合検出手法としてモデル検査を用いることで，モデルそのものによる Verification の不具合の検出のみならず，モデルの作成時においても Validation の不具合が検出されるという副次的な効果があった．
- 機能に関する不具合は重要な（修正を要する）不具合である割合が高かった．この不具合を検出可能な不具合検出観点と不具合検出手法の組合せを重点的に実施することが望まれる．

なお，本論文ではケーススタディの対象として，ソフトウェア要求分析フェーズの成果物を対象に実施されたある一つの IV&V 活動のみを扱った点に注意されたい．IV&V 活動による検出不具合の網羅性や効率性についての議論を行うためには過去の検出傾向との比較が必須であり，他の開発工程や他のプロジェクトとの比較を行うことは今後の課題である．

謝辞 本研究の一部は，文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われた．また特別研究員奨励費（課題番号：21・8995）の研究助成を受けて行われた．

文　献

[1] J.D. Arthur, M.K. Groener, K.J. Hayhurst, and C.M. Holloway, “Adding value to the software development process: A study in independent verification and validation,” Technical Report, TR-98-15, Virginia Tech, 1998.

[2] R.O. Lewis, Independent verification and validation: A life cycle engineering process for quality software, Wiley, 1992.

[3] A.B. Bassam, B. Jonathan, and R.M. Philippe, “Independent validation and verification of the tcas ii collision avoidance subsystem,” Aerospace and Electronic Systems Magazine, vol.15, no.8, pp.3–21, 2000.

[4] M.E. Fagan, “Design and code inspection to reduce errors in program development,” IBM Syst. J., vol.15, no.3, pp.182–211, 1976.

[5] O. Laitenberger, “Studying the effects of code inspection and structural testing on software quality,” ISSRE ’98: Proc. Ninth International Symposium on Software Reliability Engineering, pp.237–246, IEEE Computer Society, Washington, DC, USA, 1998.

[6] IEEE/ANSI, “IEEE standard classification for software anomalies,” 1993.

[7] R. Chillarege, I.S. Bhandari, J.K. Chaar, M.J. Halliday, D.S. Moebus, B.K. Ray, and M.Y. Wong, “Orthogonal defect classification — A concept for in-process measurements,” IEEE Trans. Softw. Eng., vol.18, no.11, pp.943–956, 1992.

[8] D. Kelly and T. Shepard, “A case study in the use of defect classification in inspections,” CASCON ’01: Proc. 2001 Conference of the Centre for Advanced

Studies on Collaborative Research, p.7, IBM Press, 2001.
[9] L.O. Damm and L. Lundberg, “Identification of test process improvements by combining fault trigger classification and faults-slip-through measurement,” ISESE ’05: Proc. 4th International Symposium on Empirical Software Engineering, pp.152–161, 2005.
[10] IEEE/ANSI, “IEEE standard for software verification and validation plans,” 2004.
[11] 加藤　淳, 星野伸行, 片平真史, 石濱直樹, 宮本祐子, 神武直彦, “クリティカルソフトウェアに対する IV&V：Independent Verification and Validation,” 情処学研報. 第 155 回ソフトウェア工学研究会報告, vol.2007-SE-155, no.33, pp.81–87, 2007.
[12] R.B. Grady, Practical software metrics for project management and process improvement, Prentice-Hall, Upper Saddle River, NJ, USA, 1992.
[13] B. Beizer, Software testing techniques, 2nd ed., Van Nostrand Reinhold New York, NY, USA, 1990.
[14] 松村知子, 門田暁人, 森崎修司, 松本健一, “マルチベンダ情報システム開発における障害修正工数の要因分析,” 情処学論, vol.48, no.5, pp.1926–1935, May 2007.
[15] R.R. Lutz, “Targeting safety-related errors during software requirements analysis,” Proc. 1st ACM SIGSOFT Symposium on the Foundations of Software Engineering, pp.99–106, 1993.
[16] Telelogic, “Doors tools,” http://www.telelogic.com/doors
[17] U.S. Nuclear Regulatory Commission NRC, Fault tree handbook, NUREG-0492, 1981.
[18] D.J. Reifer, “Software failure modes and effects analysis,” IEEE Trans. Reliab., vol.28, no.2, pp.247–249, 1979.
[19] M.C. Edmund, O. Grumberg, and D.A. Peled, Model checking, The MIT Press, 1999.

付　　録

表 A·1　筆者らの分類結果との不一致（ぶれ）の個数
Table A·1　The number of differences with our classification result.

	被験者 A	被験者 B	平均
提案分類法	5 (14%)	9 (26%)	7 (20%)
従来手法	17 (49%)	19 (54%)	18 (51%)

（平成 21 年 2 月 24 日受付，7 月 6 日再受付）

松本　真佑　（学生員）

平 18 京産大・理卒．平 20 奈良先端科学技術大学院大学情報科学研究科博士前期課程了．現在，同大学博士後期課程在籍中．平 21 日本学術振興会特別研究員（DC2）採用．エンピリカルソフトウェア工学，特にソフトウェアメトリクスの研究に従事．情報処理学会，IEEE，ACM 各会員．

上野　秀剛　（学生員）

平 16 岩手県立大・ソフトウェア情報卒．平 21 奈良先端科学技術大学院大学博士課程了．同年奈良工業高等専門学校情報工学科助教．博士（工学）．ソフトウェア開発におけるヒューマンファクタの研究に従事．IEEE，ACM 各会員．

門田　暁人　（正員）

平 6 名大・工・電気卒．平 10 奈良先端科学技術大学院大学情報科学研究科博士後期課程了．同年同大学助手．平 16 同大学助教授．平 19 同大学准教授．平 15〜16 Auckland 大学客員研究員．博士（工学）．ソフトウェアメトリクス，ソフトウェアプロテクション，ヒューマンファクタの研究に従事．情報処理学会，日本ソフトウェア科学会，IEEE，ACM 各会員．

松本　健一　（正員）

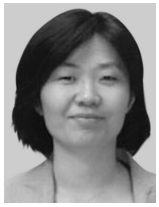
昭 60 阪大・基礎工・情報工学卒．平元同大大学院博士課程中退．同年同大学基礎工学部情報工学科助手．平 5 奈良先端科学技術大学院大学助教授．平 13 同大学教授．工博．エンピリカルソフトウェア工学，特に，プロジェクトデータ収集/利用支援の研究に従事．情報処理学会，日本ソフトウェア科学会，ACM 各会員，IEEE Senior Member．

片平　真史

1987 宇宙開発事業団（NASDA）入社．国際宇宙ステーション，H-IIA ロケットなどの開発業務に従事．マサチューセッツ工科大学航空宇宙工学部 Complex Systems Research Lab (Prof. Nancy G. Leveson) 研究員・講師を経て，宇宙航空研究開発機構主幹開発員として高信頼ソフトウェア開発保証及びソフトウェア安全性に関する業務に従事．現在，ソフトウェア独立検証及び有効性確認（IV&V），ソフトウェアプロセス改善，高信頼性 RTOS 開発に関する業務のチームリーダ．フロリダ工科大学修士課程了．IEEE，情報処理学会各会員．

神武　直彦

1998 慶應義塾大学大学院理工学研究科了．同年宇宙開発事業団入社．H-IIA ロケット搭載機器の研究開発に従事．慶應義塾大学政策・メディア研究科助手，欧州宇宙機関訪問研究員を経て，宇宙航空研究開発機構主任開発員として，宇宙機搭載ソフトウェアに対する独立検証及び有効性確認（IV&V）に従事．現在，慶應義塾大学先導研究センター准教授．システムズエンジニアリング，宇宙技術を利用したインテリジェントシステムなどの研究に従事．PMP．博士（政策・メディア）．INCOSE，IEEE，情報処理学会各会員．



宮本　祐子

宇宙航空研究開発機構開発員として，モデルベース開発，ソフトウェアプロセス改善，ソフトウェア独立検証及び有効性確認（IV&V），プロセス改善に従事．筑波大学大学院物理学研究科前期課程了．

氏原　頌悟

2004 名城大・理工・情報科学卒．同年三栄ハイテックス社入社．携帯音源に関連したソフトウェアの開発やテスト業務に従事．2008 JSTQB 認定テスト技術者資格 Foundation Level 取得．2008 より宇宙航空研究開発機構開発員として，ソフトウェア独立検証及び有効性確認（IV&V）に従事．

吉川　茂雄

平 11 北大・工・機械卒．平 16 同大大学院博士課程了．同年有人宇宙システム（株）入社．博士（工学）．ソフトウェア独立検証及び有効性確認（IV&V）に関する業務に従事．情報処理学会会員．