B-027

組込みソフトウェアに対するソフトウェア FMEA の試行実験とその考察
Trial and its prospects of software FMEA on embedded software

余宮 尚志† 小島 昌一‡
Hisashi Yomiya Shoichi Kojima

1. はじめに

製品の故障や不具合を分析する方法として、FMEA (Failure Mode and Effects Analysis)[1]がある。ソフトウェアに対してこれを適用するには、FMEA における部品(Item)や故障モード(Failure Mode)をソフトウェア向けに解釈しなおす必要がある。東芝では、部品をソフトウェアにおける機能と捉え、故障モードを発想しやすくするために観点リストを用いるソフトウェア FMEA を開発してきた[2], [3]。

ソフトウェア FMEA はソフトウェア開発の上流工程で、将来起こり得るソフトウェアの故障を未然に防止するための方法として導入が進みつつある。しかしその有効性については客観的な評価ができていない。ソフトウェアに対して FMEA を適用した実証実験はあるが[4], [5]、観点リストを用いたソフトウェア FMEA の方法ではなく、また有効性が十分に示されているとは言えない。

そこで、観点リストを用いたソフトウェア FMEA の有効性をはかるため、2 つの組込みソフトウェアに対して、ソフトウェア FMEA を用いる場合、およびソフトウェア FMEA を用いずに故障モードを抽出した場合で効果に差が出るかどうかを実験によって確かめた。本論文ではその実験結果について報告する。

2. 試行実験の概要

ここでは試行実験の概要を述べる。

2.1 2 つの実験対象

1 つ目のシステム X は、エレベータが扉を開いたまま走行することを防止する「戸開走行保護装置」である。故障モードの抽出対象は、このシステム内にある自己診断機能に関するソフトウェアである。

2 つ目のシステム Y は、組込み機器のデータ管理に適した「軽量データベース TinyBrace$_{TM}$」[6]で、レプリケーション機能に関するソフトウェアに対して故障モードを抽出した。

2.2 実験の制約事項

ソフトウェア FMEA には、その分析方法の手順と、一般の組込みソフトウェア向けに用意した観点リストを含むガイドラインが用意されている。実験では、システム X とシステム Y についての同じ仕様書に対して、表 1 にあるように、ガイドラインを用いて故障モードを抽出するグループと、ガイドラインを用いずに故障モードを抽出するグループを設定した。そして 3 時間を上限とし故障モードを抽出した。

グループ	ガイドライン	ドメイン知識	経験特性
X－1	無し	無し	若手
X－2	無し	無し	ベテラン
X－3	有り	無し	中堅
Y－4	有り	無し	若手
Y－5	無し	有り	ベテラン

表 1. 試行実験における 5 つのグループ

経験特性については、組込みソフトウェア開発における経験年数が若手は 3 年未満、中堅は 3 年以上 10 年未満、ベテランは 10 年以上の技術者とした。

また、ソフトウェア FMEA では、(1)部品の洗い出し、(2)故障モードの抽出、(3)影響解析、(4)対策の実施、を行うが、今回の試行実験の対象範囲は(2)までである。

ソフトウェア FMEA における観点リストは、実際には製品のドメインに応じて用意するものである。しかし今回の実験では、一般の組込みソフトウェア向けに用意した観点リスト[2], [3]を用いることとした。その理由は 2 つある。1 つは故障モードの発想には、ドメインに応じた観点リストを用いた方が効果が期待できるが、そうした観点リストを用いない場合でも効果があるかどうかを確かめるためである。もう 1 つは、ドメインに応じた観点リストは、過去に発生した故障の真因解析を行い、継続的にメンテナンスを行い更新し続けることが想定される。今回は試行実験のみのために、こうしたドメインに応じた観点リストを用意するのは難しかったためである。

2.3 試行実験の評価方法

抽出された故障モードに対する評価は、対象システムの開発におけるベテランが実施した。ここで言うベテランは、組込みソフトウェア開発の経験年数に加え、過去に実際その製品の開発に主体的に携わっている技術者である。評価の公平性を保つため、このベテランは実施者などの詳細は知らされない状態で評価を行った。

この評価では、抽出された故障モードを以下の 4 つに分類することとした。

分類 A: 間違い・誤解の可能性指摘
開発を INPUT から OUTPUT への変換(例えば、顧客要求をもとに仕様書を作成、仕様書をもとに設計書を作成など)と考えた場合の、間違い(INPUT から OUTPUT へ正しく変換されないなど)と誤解(INPUT の意味を勘違いするなど)の可能性を記述した内容

分類 B: 検討すべき故障発生条件の指摘
特定の条件(タイミング、動作環境、使い方など)や、それらの組み合わせによって不具合が起こり得るソフトウェア内部の構造や振る舞いを記述した内容

分類 C: 機能の裏返し
対象の機能が正しく動作しないなど、結果のみを記述した内容(不具合発生条件が曖昧なものも含む)

† 株式会社東芝 Toshiba Corporation
‡ 東芝ソフトウェア・コンサルティング株式会社
Toshiba Software Consulting Corporation

分類 D:　ハードウェアの故障モード
　ハードウェアの故障が原因で、ソフトウェアの機能が動作しないことを記述した内容

　このうち、開発の上流工程における故障の未然防止につながる、分類 B の割合を中心に評価した。分類 B は仕様書を検討した結果、潜在するソフトウェアの故障可能性を抽出することを意味するため、重視すべきであると判断した。これに対し、分類 A は仕様書を作成する時の検討ミスなどであり、分類 C は故障内容の具体化が足りず、故障の可能性を十分に特定できていない。分類 D はソフトウェアの故障ではないため、評価の対象外とした。

3. 試行実験の結果

　表 2 は各グループで抽出した故障モードを A から D に分類し、それぞれの個数を集計した結果である。FMEA は、部品や故障モードの粒度が一意に決まっているわけではないため、抽出された個数だけでなく、分類 B の比率も考慮する。

グループ	分類 A	分類 B	分類 C	分類 D
X－1	0	3	57	10
X－2	0	5	3	0
X－3	10	11	3	1
Y－4	6	27	5	6
Y－5	1	16	3	4

表 2. 各グループにおける分類ごとの故障モード抽出個数

　X－1 のグループは、多くの故障モードを抽出しているが、そのうち約 81%は「通信する」に対して「通信できない」といったように機能の裏返しを表現したものである。X－2 のグループは、個数は少ないが、その多くが分類 B、つまりソフトウェア FMEA で抽出すべき故障モードとなっている。X－3 のグループも同様に、分類 B を多く抽出できているが、分類 B 以外の故障モードも同程度抽出しているのが特徴である。
　Y－4 のグループは、多くの故障モードを抽出できており、分類 B が約 61%となっている。Y－5 のグループは、Y－4 より抽出した個数は少ないが、分類 B が約 67%と割合が高い。
　X－2 と Y－5 のベテランが実施したものの抽出個数が他グループと比較して多くないのは、発生確率がないか極めて低い故障を予め排除しているためであることが、実施後のインタビューによって明らかになっている。

4. 考察

　システム X では、ガイドラインを利用しない場合、若手技術者はシステムが機能不全になる条件を抽出できれば良いと考え、ソフトウェアに適した故障モードの抽出ができていないと考えられる。一方、ベテランはガイドラインを利用しなくても、ソフトウェアで事前に検討しておかなければならない故障を、これまでの開発経験から理解しており、具体的に踏み込んだ故障モードが抽出できていると考えられる。ガイドラインを利用した場合は、抽出すべき故障モードの種類が理解できること、また観点リストを用いて「何が」「どうして」「どうなる」という抽出手順に従えるため、掘り下げて故障モードを抽出できていることが分かる。
　システム Y では、ドメイン知識のない若手技術者がガイドラインを利用することにより、ベテランにも劣らない主要な故障モードが抽出できている。システム Y について B に分類された故障モードの内訳を見ると、Y－4 の 27 個のうち、約半数は発生確率が非常に小さなものであった。そして 7 個は Y－5 で抽出できていない故障モードだった。これは若手技術者にはドメイン知識がない分、先入観が少ないこと、そしてガイドラインによる観点リストを用いることで、発想が広がったためと考えられる。他方、Y－5 の抽出した故障モードにも Y－4 で発想できていない故障モードが含まれていた。これは過去のトラブル事例などの知識に差があるなど、開発経験による違いと思われる。

5. おわりに

　本論文では、実験を通して 2 つの組込みソフトウェアに対し、ソフトウェア FMEA を用いる場合と用いない場合で、抽出した故障モードを評価し、効果に差があるかどうかを確かめた。
　その結果、ソフトウェア FMEA を利用することによって、有効な故障モードがより多く抽出できることが分かった。中堅やベテラン技術者はソフトウェア FMEA を用いなくても有効な故障モードを抽出することはできるが、発想が広がらず十分な故障モードが抽出できない場合があることが分かった。
　ソフトウェアにおける故障の未然防止は、ソフトウェア FMEA を用いるだけでなく、ベテランによるレビューを行うなど、これまでの開発知識との共存が重要であると言える。
　今後は影響解析によって、抽出した故障モードのリスクの度合いに差が見られるかどうかを検証する必要がある。また、今回は観点リストとして一般の組込みソフトウェア向けに用意したものを用いたが、次はドメイン向けに開発したものを用いて実験を行う予定である。ドメインに応じた観点リストを用いることで、抽出する故障モードの数や質がさらに向上することが期待される。

参考文献

[1] S+IEC 60812 Ed.2.0: Analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA) (2006)
[2] 夏目珠規子, 小島昌一ほか: ソフトウェア開発における FMEA の適用可能性検討, 第 41 回信頼性・保全性シンポジウム発表報文集, p.359－364 (2011)
[3] 大谷和夫, 塩谷和夫ほか: 上流工程における未然防止プロセスの提案－未然防止リストの活用と欠陥の発想－, ソフトウェアテストシンポジウム JaSST ’12 (2012)
[4] 山科隆伸, 森崎修司ほか: 保守開発型ソフトウェアを対象としたソフトウェア FMEA 実証的評価, ソフトウェア品質シンポジウム 2008 発表文集, pp.157－164 (2008)
[5] 中西恒夫, 久住憲嗣ほか: ソフトウェア FMEA の一手法とプロダクトライン開発におけるその利用, 信学技法, SS2011-60, pp.19－24 (2012)
[6] 金松基孝, 山地圭: 組込み機器のデータ管理に適した軽量データベース TinyBrace$_{TM}$, 東芝レビュー, Vol.67, No.8, pp.11－14 (2012)