

## Web アプリケーションを対象とした故障モード影響解析の試行

高井 雄 治<sup>†1</sup>    田村 晃 一<sup>†1</sup>  
森崎 修 司<sup>†1</sup>    松本 健 一<sup>†1</sup>

故障モード影響解析はハードウェアの品質向上を目的とした技法であり、設計段階で起こりうる故障のパターンを列挙し、パターン毎に、致命度を評価し、致命度の高い順に対策を検討する。本稿では、ソフトウェアを対象としてハードウェアと同様の手順で故障モード影響解析を実施し、有益な故障モードが抽出できるか試行した結果を報告する。具体的には、設計ドキュメントから有識者によって、14 件の故障モードを列挙し、抽出された故障モードを欠陥発見の指針としてコードレビューを実施した。特に指針を与えずコードレビューした場合と比較して、システム上で影響の大きい欠陥を発見できることを示す結果が得られた。

### Experimental evaluation of failure mode effect analysis on Web application

YUJI TAKAI,<sup>†1</sup> KOICHI TAMURA,<sup>†1</sup>  
SHUJI MORISAKI<sup>†1</sup> and KEN-ICHI MATSUMOTO<sup>†1</sup>

Failure mode and effect analysis (FMEA) is a technique aiming quality improvement in hardware design. In FMEA, failure modes are enumerated and their effects are evaluated. Each failure mode are ranked by fatality according to its effect. In this paper, we show result of experiment of software. The purpose of the experiment is to evaluate effectiveness of obtained failure modes. First, experts enumerates failure mode from a design document of a web application. Then, with the failure mode, subjects did code inspection. In the experiment, 14 failure modes are obtained and with the failure mode more fatal failures are detected.

### 1. はじめに

FMEA(Failure Mode and Effect Analysis: 故障モード影響解析)は、ハードウェアの品質向上を目的とした技法であり、設計段階で起こりうる故障のパターンを列挙し、パターン毎に、致命度を評価し、致命度の高い順に対策を検討する。現在までハードウェアを対象とした多くの FMEA の研究が行われてきて、その有効性が明らかになっている。例えば、イギリスの自動車企業に対して、FMEA を使用し、その使用者に対してアンケートを行い評価を行った研究がある<sup>1)</sup>。そこで FMEA の開発プロセスの向上、チームでの活動の促進、さらに準備や更新の時間短縮が報告されている。

一方で、ソフトウェアの品質を確保するための活動の一つとして、レビューが注目されている。レビューとは、ソフトウェア開発プロジェクトの各工程で作成された成果物などの欠陥を目視によって発見することである。レビューは優れた欠陥検出方法であることに加えて、開発の早期から実施可能であることから、欠陥の早期発見に伴う開発効率の向上にも効果があることが報告されている<sup>2)</sup>。しかし、大規模なソフトウェアではドキュメントの量が膨大であり、また人間の目視によるレビューは作業負荷が高いため、労力や時間の効率が求められている<sup>3)</sup>。そのため、多くのレビュー手法の研究が行われてきた。

本稿では、FMEA を小規模 Web アプリケーションを対象に行い、過去の類似の故障(故障モード)を列挙することを試みた。さらに列挙された故障モードに基づき後述する方法で致命度を推測し、故障モードを致命度の順に並べ替えた FMEA ワークシートを作成する。そして、作成した FMEA ワークシートを使用したレビューを実施し、どのような欠陥を発見しているのか、その結果を報告する。FMEA の結果を FMEA の作成者以外がレビューに使用することは、新しい試みである。

以下、2 章では FMEA の手順を述べる。3 章では本稿で行った実験の概要を述べる。また Web アプリケーションを対象に行った本稿に適用した FMEA ワークシートの作成手順を述べる。4 章で実験結果を報告する、5 章でまとめおよび今後の課題を述べる。

### 2. 故障モード影響解析の手順

FMEA(故障モード影響解析)はハードウェアに広く用いられてきた手法で、発生しうる故障モードを予測し、それぞれの致命度を評価したものである。致命度は対象となるシステムがその故障により受ける信頼性、保全性、安全性などの影響により、重大な故障となる要素であるかを解析したものである。致命度を評価することにより重大な故障となる部分が

<sup>†1</sup> 奈良先端科学技術大学院大学 情報科学研究科

Graduate School of Information Science, Nara Institute of Science and Technology

ら対策を講じることができるのが FMEA の特徴である。

電球を例として FMEA の具体的手順を次に示す<sup>4)</sup>

(1) システム構成要素の抽出

解析対象システムを理解し、故障の原因となるシステムを構成する機器や部品を抽出する。解析を行う対象を細かい単位で抽出することにより、故障がどの部分で影響しているのかを判断することが容易になる。図 1 左は、電球を構成する部品の抽出を示す。

(2) 故障モード抽出

システムを十分に理解しているメンバにより (1) で抽出した構成要素ごとに故障モードを列挙する。図 1 右は、抽出した部品ごとの故障モードの事例を示す。

(3) 影響解析

(2) で抽出された故障モードが発生した場合の影響を記述し、故障モードの発生頻度と利用者への影響度の評価を行う。また影響度と発生頻度の積を致命度として求める。求めた致命度の大きい故障モードに対して優先して対策を検討する。表 1 は影響解析の事例である。影響度、頻度、致命度をそれぞれ程度によって 4 つに分類している。ランクの評価方法は解析対象によって異なる。

(4) 故障モード影響解析ワークシートの作成

以上の結果に対して故障モード影響解析ワークシートを作成する。ワークシートは表形式で構成され、各行が構成要素に対応している。各列には、“機器”、“故障モード”、“故障の影響”、“致命度”が含まれる。表 2 に FMEA ワークシートの事例を示す。

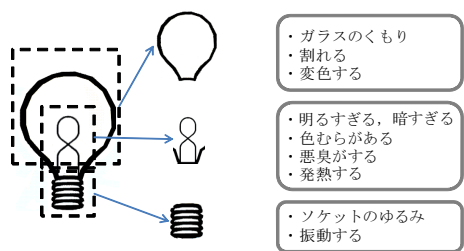


図 1 システム構成要素ごとの故障モードの抽出

表 1 影響解析

影響度		頻度		致命度		
程度	ランク	程度	ランク	程度	A × B	ランク
大いに影響がある	4	頻繁に起きる	4	非常に致命的である	12 以上	4
影響がある	3	よく起きる	3	致命的である	6 ~ 12	3
少し影響がある	2	たまに起きる	2	少し致命的である	2 ~ 6	2
ほとんど影響がない	1	めったに起きない	1	ほとんど致命的でない	1 ~ 2	1

表 2 FMEA ワークシート

機器	故障モード	故障の影響	影響の致命度
ガラス	割れる	安全性に問題あり, ガラスの破損	3
	変色する	使用時に支障を生じる, 商品価値の低下	1
ソケット	発熱がする	火傷, 火災の原因	2

### 3. 実験概要

#### 3.1 概要

Web アプリケーションを対象に故障モードを列挙することが可能であるか、また故障モードを列挙することができれば、その故障モードに該当する欠陥が発見できるか実験的に試みた。どのような欠陥を発見することができたのかを評価するため、AHR (Ad-Hoc Reading) 手法を比較対象とする。

被験者は、奈良先端科学技術大学院大学の情報科学研究科の博士前期課程及び博士後期課程の学生 8 人と、C 言語での開発経験のある実務経験者 2 人である。FMEA, AHR の 2 つの手法に学生 4 人、実務経験者 1 人ずつ分割した。その際に、レビューアの C 言語及びその他のプログラミング言語の経験年数を用いて、スキルの差が生じないようにグループの分割を行った。

#### 3.2 レビュー対象

対象は会議等のイベントの参加登録を実現するシステムのソースコードであり、Apache Web サーバの CGI として動作する C 言語で実装されている。規模は約 1,500LOC である。システムは、類似するシステムでの開発経験がある開発者により実験用に開発されたものである。また別の開発者により大まかなレビューを実施している。図 2 はイベント受付システムの情報のやりとりを示したものである。以下にその内容を記述する。

(1) 「イベント管理者」が「システム管理者」に対して、イベント作成の依頼をし、イベ

ント作成画面からイベントを作成する。

- (2) イベントを作成すると、作成されたイベントの登録用 Web ページの URL が生成される。
- (3) 「イベント管理者」は「イベント参加者」に参加の呼びかけを行い、「イベント参加者」はイベント参加者画面から、必要な情報を「イベント管理者」に送信する。
- (4) 「イベント参加者」はイベントに登録されると、イベント参加登録完了の通知が送信される。「イベント管理者」は、イベント管理者画面から参加登録者のリストを閲覧することができる。

レビューアには外部仕様書、内部設計書、ソースコードが渡される。また FMEA でのグループには FMEA ワークシートが渡される。外部仕様書、内部設計書は自然言語で書かれていて、誤りはないものとし、レビューの対象ではない。

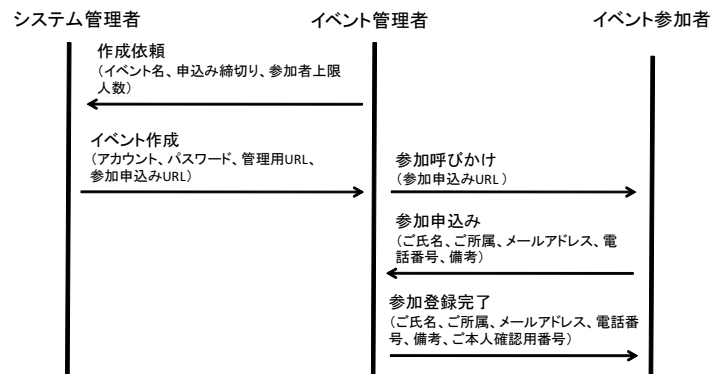


図 2 イベント受付システムの全体像

### 3.3 実験手順

以下に、Web アプリケーションを対象とした故障モード抽出の実験と、作成された FMEA ワークシートを使用したレビュー実験の二つの実験手順を述べる。ただし、2 つの実験は異なる被験者によって行う。

故障モード抽出の実験を以下の手順で行う。

- (1) システムを把握した後に 1 時間のブレインストーミングを行い、故障モードを列挙する

- (2) 挙げられた故障モードに対して、後述する手順で FMEA ワークシートを作成する。レビュー実験を以下の手順で行う。

- (1) レビュー対象としているシステムを説明する。
- (2) FMEA ワークシートの見方を説明する。AHR を適用したレビューアにはレビュー手法に関する説明はない。
- (3) システムの概要理解を各自で行い、その後レビュー作業を行う。欠陥を発見した際にはその内容と発見時刻を記録している。

所要時間は、システムの説明が 10 分、レビュー手法の説明が 10 分、システムの概要理解に 20 分、レビュー作業が 60 分である。

### 3.4 FMEA ワークシート作成の手順

本稿では、2 章で述べた FMEA をレビュー対象のソフトウェアに適用する。

以下に本稿で使用する FMEA の具体的な手順を図 3 に示す。

- (1) システム構成要素の抽出 (2 章での (1) と対応)

本稿で使用するイベント登録システムの構成要素を抽出する。システムは、「参加登録機能」「イベント管理機能」「システム管理機能」で構成している。その構成要素それぞれで異なる画面ごとにファイルを分割する。機能によってファイルが分割されているため、レビューの際に故障がどのファイルに対応しているのかが理解しやすい。

- (2) 故障モード抽出 (2 章での (2) と対応)

分類した機能に対して、類似システムの開発経験がある 2 名で 1 時間のブレインストーミングを行い、故障モードを列挙する。故障モードを十分に抽出するために、ブレインストーミングを行う前に外部仕様書と内部設計書を渡して、十分にシステムを把握した後に行う。

- (3) 影響解析 (2 章での (3) と対応)

(2) によって挙げられた故障モードに対して、その原因、影響を解析する。故障の影響は、その故障によってどのような影響があるのか (1 次影響) と、その故障が起こるとシステム上でどのような影響があるのか (2 次影響) の二つに分類している。

FMEA は故障要因の抽出を行うことを目的としているので、システムを構成する機器のどの「故障モード」の発生を防止するののかについての評価が必要である。致命度の計算は、以下の式によって求めている。

$$\text{致命度} = \text{影響度} \times \text{故障モード発生頻度}$$

影響度は、解析した故障の原因、影響に基づいて、挙げられた故障モードで順位付けを

行う。順位付けはイベント登録システム上でユーザへの影響の大きさにより判断する。また故障モード発生頻度は「ほとんど発生しない」「たまに発生する」「よく発生する」の3つの頻度によって判断する。

(4) 故障モード影響解析ワークシートの作成 (2章での(4)と対応)

(3) で計算した致命度が大きい故障モードの順に上から記述を行う。作成した FMEA ワークシートの各列には、「画面名」と「機能」, 「故障モード」と「致命度」が含まれる。

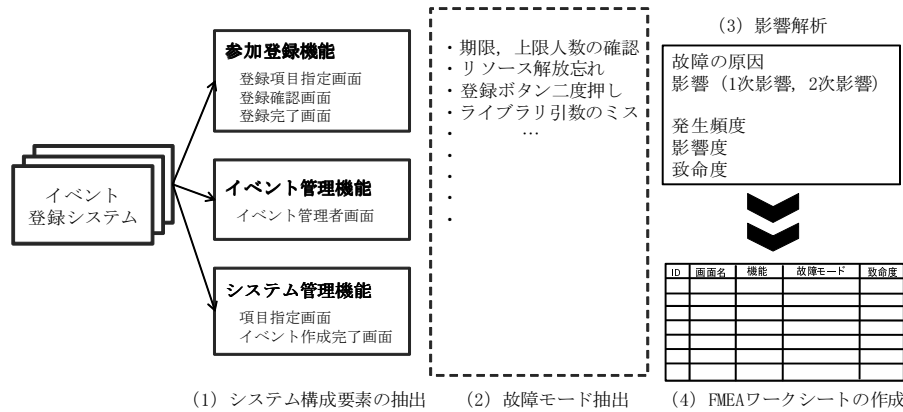


図3 FMEA ワークシート作成手順

4. 実験結果

4.1 得られた故障モード

Web アプリケーションを対象に FMEA を適用した結果, 3.4 の手順により 14 件の故障モードを列挙することができた。表 3 は得られた故障モードに基づき作成した FMEA ワークシートである。致命度の数値幅を 1~10 になるように調整した。ユーザへの影響が大きい故障モードも列挙することができたことが分かった。

致命度が最も高い故障モードは, 受付 html 生成時, 期限または上限人数の超過確認をしていないことであった。期限, 上限人数の超過確認をしていない場合, 正しく登録できず, イベント登録ができない可能性が高い。また同じ致命度の故障モードであったライブラリ引

数のミスは, システム全体が正しく動作しなくなり, システムが停止する可能性がある。

一方で, 致命度が最も低い故障モードは, イベントが多すぎるため表の行数が多くなりブラウザの性能劣化が起こることであった。リソース限界の考慮不足が原因であるが, 発生頻度は低く, パフォーマンスが低下する欠陥である。

表 3 FMEA ワークシート

ID	画面名	機能	故障モード	致命度
1	登録項目指定画面, 登録確認画面, 登録完了画面	イベント参加登録機能	受付 html 生成時, 期限または上限人数の超過確認をしていない	9
2	イベント作成完了画面 イベント管理者画面 登録項目指定画面	イベント作成機能 イベント閲覧機能 イベント参加登録機能	ライブラリ引数のミス(例, 引数の数, 引数の型の不一致, 同じ引数の型で順序が違う(a(x,y) a(y,x) など), ライブラリ関数の戻り値解析ミス(例, 0 正常とすべきなのに-1 正常としている)	9
3	イベント作成完了画面 登録項目指定画面 登録確認画面 登録完了画面	イベント作成機能 イベント参加登録機能	入力画面において, 仕様書で定義されている入力項目と実際の入力項目が異なる(例, 英数字を入力する欄に日本語を入力しても受け付けられてしまう, 全角を入力する欄に半角を入力しても受け付けられてしまう, 入力可能文字数が仕様書で定義されている文字数より多いまたは少ない, 数字入力欄の桁指定ミス)	5
4	イベント作成完了画面 登録項目指定画面 登録確認画面	イベント作成機能 イベント参加登録機能	入力に含まれる HTML タグ(例, 入力欄に対する <SCRIPT language="JavaScript"> ~ </SCRIPT> といった Javascript の入力), SQL 文, シェルスクリプト等の無効化(エスケープ)忘れ	4
5	登録項目指定画面 登録確認画面 登録完了画面	イベント参加登録機能	登録項目指定画面を表示後または登録確認画面を表示後, 期限または上限人数の超過確認をしていない(例, A(開始) B(確認) C(完了)(A と B の間または B と C の間に制限を超えていても登録できてしまう))	4
6	項目指定画面 登録項目指定画面 登録確認画面	イベント作成機能 イベント参加登録機能	画面上の登録ボタン二度押しにより, 二重にイベント作成またはイベント参加登録してしまう, もしくは, 作成または登録できない	3
7	イベント作成完了画面 イベント管理者画面 登録項目指定画面 登録確認画面 登録完了画面	イベント作成機能 イベント閲覧機能 イベント参加登録機能	配列の不正アクセス(オーバフロー)及び解放忘れ(メモリリーク)	3

ID	画面名	機能	故障モード	致命度
8	イベント作成完了画面 登録項目指定画面 登録確認画面 登録完了画面	イベント作成機能 イベント参加登録機能	送信された文字コードを正しく解釈できず、登録者ごとに日本語文字コードが違う	3
9	イベント作成完了画面 イベント管理者画面 登録項目指定画面 登録確認画面 登録完了画面	イベント作成機能 イベント閲覧機能 イベント参加登録機能	リソース解放忘れ(ファイル閉じ忘れ)	3
10	イベント作成完了画面 イベント管理者画面	イベント作成機能 イベント閲覧機能	アカウント、パスワードの設定が誤っており、正しいアカウント、パスワードでもページを表示できない、または、異なるアカウントのページが不正に参照できる	2
11	イベント作成完了画面 イベント管理者画面 登録項目指定画面 登録確認画面 登録完了画面	イベント作成機能 イベント閲覧機能 イベント参加登録機能	ファイルなどに記録されている過去の DB またはデータの移行に失敗し、正しい情報が取得できない(文字コード、データ区切り)	2
12	イベント管理者画面 登録項目指定画面 登録確認画面 登録完了画面	イベント閲覧機能 イベント参加登録機能	DB にアクセスする前に必要な DB の存在確認が抜けている (DB として使用しているファイルの存在確認がない)	2
13	項目指定画面 登録項目指定画面 登録確認画面	イベント作成機能 イベント参加登録機能	画面のユーザインタフェースが紛らわしく、ユーザが誤って操作する可能性がある(例: “Yes”, “No” ボタンの位置が他のページと異なる)	1
14	イベント管理者画面	イベント閲覧機能	ブラウザで表示する内容が巨大すぎて(1万行を超える html のテーブルなど)ブラウザの性能劣化が起こる(イベント多すぎる、表が大きすぎてブラウザが大量のメモリを使用したり、反応がなくなる)	1

## 4.2 欠陥分類

実験で発見された欠陥は、IEEE 1044 Standard Classification for Software Anomalies<sup>5)</sup>の Recognition classification scheme に基づいて分類した。表 4 にそれぞれの分類項目と、以下の理由で分類した重要度を示した。

(1) 重要度 A (高): 作業に対して、その内容が失われる。また動かなくなる恐れが高いシステムが停止する、入力したファイルが消去される可能性がある欠陥である。ユーザ

が入力した情報が失われ、間違った結果やデータを出力するものも、ユーザに大きな影響を与えるため重要度を高く設定している。

(2) 重要度 B (中): 動作が正しく行われない。表示が正常でない

エラーチェックがないため、そのまま入力が行われることにより表示が正常に行われないものや、ファイルが消去されないが、ファイルに入力した値が正しく記入されないために、ユーザが誤った動作を行う可能性があるものなどを重要度を中に設定している。

(3) 重要度 C (低): ユーザに与えている情報に矛盾がある。また思いもよらないエラーが生じる。

プログラムに関して仕様を満たしていないものや、設計書とプログラムの名称が違うものなどが挙げられる。実際にユーザに影響が少ないが、開発者の共通認識を妨げたりユーザが誤って判断する可能性があるものを重要度を低く設定している。なお、実験に使用したソフトウェアに含まれる欠陥のどれにもあてはまらない分類は、重要度を設定していない。

表 4 IEEE1044 の欠陥分類

重要度	分類コード	分類
A	RR510	オペレーションシステムのクラッシュ
A	RR520	プログラムの支障
A	RR530	プログラムクラッシュ
-	RR540	入力の問題
B	RR541	- 正しい入力が受けられない
B	RR542	- 間違った入力を受けている
C	RR543	- 説明が間違っている ミスがある
C	RR544	- パラメータが不完全である ミスがある
-	RR550	出力での問題
B	RR551	- 間違ったフォーマット
A	RR552	- 間違った結果 / データ
B	RR553	- 不完全さ / ミス
C	RR554	- スペル / 文法
-	RR555	- 飾り
C	RR560	求められている性能が満たされていない
-	RR570	わかっている製品欠陥の合計
-	RR580	システムエラーメッセージ
-	RR590	その他

### 4.3 発見した欠陥の種類の違い

実験により発見した欠陥の平均個数を表 5 に示す。AHR を適用した学生 4 人の発見した欠陥の平均個数が 1 つであるのに対して、FMEA では 2 つであり、2 倍の欠陥を発見できた。それに対して実務経験者では、各手法とも同じ個数の欠陥を発見した。

次に、各手法において学生と実務経験者の発見した欠陥の重要度を表 6 に示す。4.2 で示したように、発見した欠陥の重要度を分類した。

AHR を適用した学生と比較して、FMEA では多くの重要度「A」の欠陥を発見できた。実務経験者ではどちらの手法でも重要度「A」の欠陥を発見できたが、FMEA では発見した全ての欠陥が、重要度「B」以上であることが分かった。

表 5 発見した欠陥の平均個数

	AHR	FMEA
学生 (4 人)	1	2
実務経験者 (1 人)	3	3
合計	1.4	2.2

表 6 発見した欠陥の重要度

	重要度	AHR	FMEA
学生 (4 人)	A	1	3
	B	2	1
	C	1	3
	合計	4	7
実務経験者 (1 人)	A	1	1
	B	0	2
	C	2	0
	合計	3	3

次に、発見した欠陥の具体的な内容を示す。

表 7 は、学生が発見した欠陥の具体的な内容を示している。FMEA では、イベント登録の際に記録しておくファイルが消去されてしまう恐れのある欠陥を発見した。発見した重要度 B の欠陥は FMEA と AHR の大きな差は見られず、時間に関するエラー処理が行われないことで誤って行動する可能性があるものや、引数の間違いにより実際の表示が異なるため、ユーザが誤って判断する可能性があるものがあつた。また FMEA によって、ユーザへの影響が少ない重要度 C の欠陥も発見した。

次に、実務経験者が発見した欠陥の具体的な内容を表 8 に示す。FMEA では、セッションハイジャックの危険性がある欠陥を発見した。また募集に関する最大数が分からないものや、文字数のチェックが正しく行われない欠陥を発見した。AHR では実際の表示には影響しないものや、エラーメッセージ不足により、伝えるべき情報が不足している欠陥を発見した。

表 7 発見した欠陥 (学生)

	重要度	分類コード	分類	内容
AHR	A	552	出力での問題-間違った結果 / データ	現在より前の時間でイベント登録ができてしまうため、ユーザが誤って行動する可能性がある
	B	551	出力での問題-間違ったフォーマット	文字化けにより、内容が正確に理解できなくなる
		541	入力での問題-正しい入力が受けられない	設定された募集人数が一人少なくなる
	C	554	出力での問題-スペル / 文法	実際の表示に関しては問題はない
FMEA	A	530	プログラムクラッシュ	人数、締切日時、イベント名などが入力されているファイルが消える可能性がある
		530	プログラムクラッシュ	人数、締切日時、イベント名などが入力されているファイルが消える可能性がある
		552	出力での問題-間違った結果 / データ	現在より前の時間でイベント登録ができてしまうため、ユーザが誤って行動する可能性がある
	B	553	出力での問題-不完全さ、ミス	引数が誤っていることにより実際の表示が異なるため、ユーザが誤って判断する可能性がある
	C	560	求められている性能が満たされていない	設定された文字数でエラー処理が行われないため、多くの情報を受け入れてしまう
		543	入力での問題-説明が間違っている	ユーザへの影響はないが、その他設計者・開発者に対して共通認識できなくなる
		560	求められている性能が満たされていない	設定された文字数でエラー処理が行われないため、多くの情報を受け入れてしまう

### 4.4 考 察

本稿で得られた故障モードに基づき、Web アプリケーションでの欠陥の発見を試みたが、次のような知見が得られた。4.3 で明らかにしているように、学生は FMEA の方が 2 倍の欠陥を発見している。また実務経験者は同じ個数の欠陥を発見している。FMEA ワークシートは故障モードの内容と、その故障モードが含まれている可能性がある画面名 (ファイル名) が書かれている。そのため学生では、FMEA ワークシートが欠陥を発見する上でツールとしても上手く機能している。しかし、開発経験がある実務経験者は、経験から欠陥が混

表 8 発見した欠陥（実務経験者）

	重要度	コード	分類	内容
AHR	A	552	出力での問題-間違った結果/データ	現在より前の時間でイベント登録ができてしまうため、ユーザが誤って行動する可能性がある
		554	出力での問題-スペル/文法	実際の表示に関しては問題はない
	C	560	求められる性能が満たされていない	日付や募集人数が超えていても、その旨が表示されないため、ユーザは理解しにくい
FMEA	A	530	プログラムクラッシュ	セッションハイジャックなどの危険性があり、登録情報が盗まれたり、消されたりする可能性がある
		553	出力での問題-不完全さ、ミス	仕様書から許容できる募集人数の最大値がわからない
	B	542	入力の問題-間違った入力を受けている	全角を考慮した文字数のチェックになっていないため、ユーザが誤って判断する可能性がある

入しやすい箇所が予測できるため、発見した欠陥の個数に差が表れなかったと考えられる。また学生は重要度の高い欠陥を AHR では発見しにくいですが、FMEA では発見することができた。実務経験者は、効率よく重要度の高い欠陥を発見できる可能性が示唆できた。ただし、今回の実験では都合上、1 人の実務経験者でしか行っていないため、人数を増やした上での検討が必要である。さらに、実際に FMEA によって発見している欠陥は、プログラムクラッシュにより重要なイベント登録情報を入力したファイルが消失する可能性があることが示された。FMEA ワークシートを使用したレビューを適用することで、システムに影響の少ない重要度の低い欠陥よりも、先に重要度の高い欠陥に対しての対策を施すことできた。効率の良いレビューを実施することが期待できる。

## 5. ま と め

本稿では、Web アプリケーションを対象に故障モードの抽出を試みた。その結果、ユーザへの影響が大きい欠陥を含んだ故障モード 14 件を列挙することができた。列挙された故障モードを学生、実務経験者に与え、Web アプリケーションのレビューを実施した。特にガイドラインを与えない AHR と比較して、FMEA を使用したレビューではシステム上で影響が大きい欠陥を効率よく発見していることが示された。実験では実務経験者の被験者が少ないため、今後被験者を増やすことによりその有効性を明らかにする予定である。

今後の課題は、チェックリスト手法に対しても同様の実験を行い、発見した欠陥にどのよ

うな違いがあるのかを示すことにより、FMEA の特性、有効性を明らかにしていく。

## 参 考 文 献

- 1) B.G. Dale and P.Shaw. " Failure mode and effect analysis in the motor industry: A state-of-the-art study ", Quality and Reliability Engineering International, 6(3), pp.179-188,1990.
- 2) M.E.Fagan, " Design and Code Inspection to Reduce Errors in Program Development ", IBM Systems Journal, Vol.15, No.3, pp.182-211,1976
- 3) 細川 宣啓, " 第三者インスペクションによる品質検査と欠陥予防 ", 情報処理学会, VOL.50, NO.5, pp405-411
- 4) IEC INTERNATIONAL STANDARD NORME INTERNATIONALE, IEC60812, 2006
- 5) IEEE Standard Classification for Software Anomalies,IEEE1044,1993