

欠陥エンジニアリングの重要性と欠陥メタ情報の定義／管理 －欠陥マスターデータベースの普及と産業寄与に向けた提言－



Nobuhiro Hosokawa (carvin@jp.ibm.com)
Quality Engineering, Services Quality,
IBM Japan Ltd.

第1章－医学との対比 #1病理学



はじめに



ようこそ! 欠陥エンジニアリングの世界へ!

欠陥エンジニアリングは、一般的なあまり馴染みのない言葉ですが欠陥の記録／分類／活用を通じて品質管理活動全般へ貢献する、いわば「ケンパノチカラ」基礎体力と言っても過言ではない、重要な概念です。

しかし日々の技術者としての活動では、欠陥管理DBに検出した欠陥を記録しますが、実は「欠陥を記録してどうする?」という目的意識が欠如している場合が少なくありません。

また「欠陥管理DBを参照して利用するのは誰?」「欠陥はどうやって分類整理したらいいの?」という概念等も、国内ではあまり知られていないトピックです。

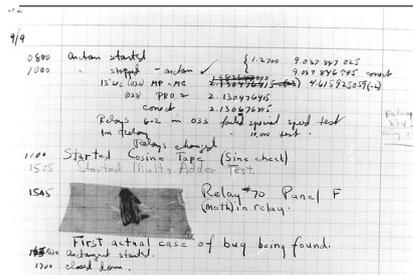
本プレゼンテーションでは、欠陥管理の一般的な問題点となぜ欠陥管理が負荷ばかり大きくうまく活用されないか、またどのように整理すればよいかを皆さんと一緒に考え、「ケンパノチカラ」としてあるべき姿を紹介します。

Nobuhiro Hosokawa
IT Architect
Quality Engineering
AIS - Global Business
Service
IBM Japan Ltd.
carvin@jp.ibm.com
Twitter://_Dreamstate

本日のプレゼンテーションでは、一部に刺激の強い画像が含まれています。

気分が悪くなった方は、挙手するなどスタッフまでお申し付けください。

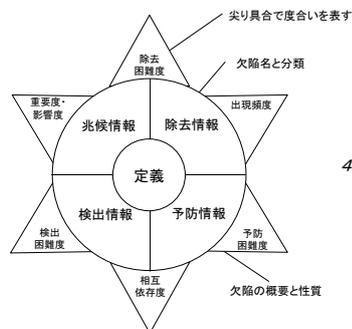
世界で最初のバグ 1947年9月9日(1947-09-09)



概要
The First "Computer Bug" Moth found trapped between points at Relay #70, Panel F, of the Mark II Aiken Relay Calculator while it was being tested at Harvard University, 9 September 1947. The operators affixed the moth to the computer log, with the entry, "First actual case of bug being found". They put out the word that they had "debugged" the machine, thus introducing the term "debugging a computer program". In 1999, the log, with the moth still taped by the entry, was in the Naval Surface Warfare Center Computer Museum at Dahlgren, Virginia, which erroneously dated it 9 September 1946. The Smithsonian Institute's National Museum of American History and other sources have the correct date of 9 September 1947 (Object ID: 1994.0191.01). The Harvard Mark II computer was not complete until the summer of 1947.

Removed caption text: Photo # NH 96566-40 First Computer "Bug", 1945
資料 1947年9月9日 NH 96566-40
提供 U.S. Naval Historical Center Online Library Photograph NH 96566-40
著作権 © 1994 The Naval Surface Warfare Center, Dahlgren, VA, 1992

欠陥の形 - Shape of Defects -



6つのトゲ
+
4象限で欠陥の全体像を表す
||
欠陥を可視化

第2章－ 欠陥マスターの必要性



全ての検出欠陥は“何でもいから”DBに登録せよの法則

バグ報告者側（テスター、開発者…）

1. 検出順序を意識せず登録する
2. プロジェクトで検出した欠陥は全て、BTS (Bug Tracking System) に記録する
3. 同種の欠陥があるかもしれないがとりえず 入力・入れる
4. 誰がどう使われて、どう意思決定されるか わからないけどとりえず登録する
5. 粒度はばらばらで入力する
6. 明確な「何を欠陥として記録するか」というルールと欠陥定義がない

報告される側（PM、品質管理担当…）

1. 未整理で「何を言っているかわからない」が たりあえず多量に登録されてくる
2. 欠陥の関連が不明。全ての欠陥は単独と 考える
3. 出てきた順に欠陥除去しようとする
4. 簡単なものから対処するよう指示する
5. 週次報告書以外にバグ情報を使わない
6. 再発している欠陥に気付かない、気付けない
7. プロジェクトを終わったら使わない

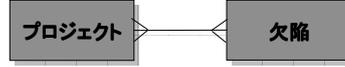
➢ せつかく時間と工数をかけて記録した欠陥情報を、実は利用していない・活用できていない事例が散見される

➢ 欠陥管理DB・BTSの情報が活用されない理由はなにか？

プロジェクトと欠陥の関係

□ プロジェクトと欠陥の関係を整理すると次のような関係。(ER図風に解説)

1. プロジェクトには複数の欠陥が発生する
2. 一つの欠陥は複数のプロジェクトで検出される



□ 上記M対Mの関係は、1対1関係の「関連エンティティ」をおくと整理できる

1. プロジェクトと欠陥の間に「プロジェクト欠陥」エンティティを設定。
2. 各1対Mの関係でリレーションを整理



医学との対比、再び

医学には...

- ✓ 医学では、稀な症例や現場の事例を学会等を通じて共有する仕組みがある
- ✓ 現場の実例から抽象化して「病気」だけを定義・管理・維持する学会・権威機関がある。そういう医学分野がある
- ✓ 現場医師は最新動向や検査・対処方法を継続的に学ぶ習慣が付いている

IT業界には...

- ✓ 稀なバグや被害の大きな欠陥を業界全体で共有する仕組みがない
- ✓ そもそもバグを研究する「バグ屋」という分野の人材も、技術分野も存在しない
- ✓ 現場は「除去」さえできればよいための欠陥情報の獲得を行わない

PubMed

PubMed comprises more than 20 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites.

医学分野で世界最大の文献データベース。1966年からNLM(米国国立医学図書館)でデータ収集が始まり、現在毎月約3万の文献が新たに追加される。現在では、米国を中心に約70か国から、900万件を超える文献が収録される。



第3章 - 欠陥マスターDBの構築

欠陥の情報分類: #1「病理学」風の整理

医学の視点では、定義と用途を重視した永続的・長期的な情報整理を行います

● 定義・基礎情報

1. 名前・別名
2. 欠陥の定義・分類
3. 一般的な推定原因発生・確率
4. 併発症・合併症
5. 特記事項

● 検出情報

1. どのように検出するか？
2. どのメトリクスを計れば欠陥兆候を捕らえることができるか？
3. 定性情報？定量情報？

欠陥の情報

● 兆候情報

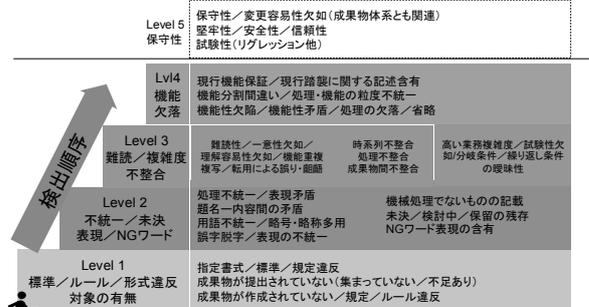
1. どうやったらその欠陥に気付くか？
2. どのメトリクスを計れば欠陥兆候を捕らえることができるか？
3. 定性情報？定量情報？

● 除去方法・予防方法

1. 関連する欠陥・連鎖欠陥
2. 除去時の注意点
3. 副作用・除去リスク
4. 再発予防方法

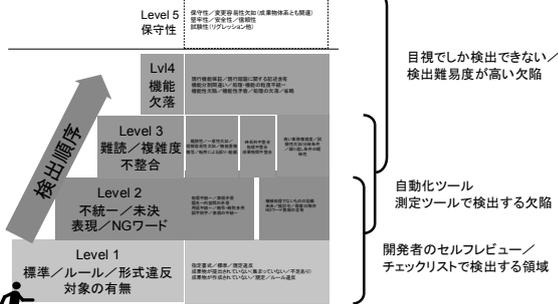
欠陥の情報分類: #2「レビュー・テスト」専門家の整理

レビュー・テスト担当者の視点では、検出順序を考慮した階層的な欠陥分類が必要で



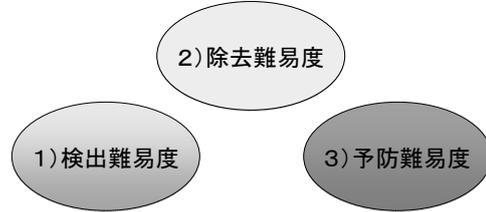
欠陥の情報分類: #2 「レビュー・テスト」専門家の整理

それぞれのレベルに応じて最適な検出方法を用いるべきです。



欠陥の情報分類: #3 「品質管理」専門家の整理

品質管理の専門家の立場からは、以下の3つの観点での欠陥分類が必要です。



- 検出困難度--開発者またはプロジェクト内部メンバーによる自己レビューで検出しにくい欠陥
- 除去困難度--欠陥の存在は確認できたが、本番環境から除去しにくく、除去工数のかかる欠陥
- 予防困難度--他の困難度に関係なく顕発し、根本原因対策が難しいため、再発予防の難しい欠陥

誰が使う? メリットは?

■ プロジェクトマネージャー(開発中)

- ②兆候情報+③検出情報=品質管理上の必要アクション指示
- ③検出情報+④除去・予防方法=品質管理上の予防アクション指示

■ プロジェクトマネージャー-経営層(プロジェクト計画時)

- ②兆候情報+④除去・予防方法=当該プロジェクトのリスク判定+プロジェクト計画
- ①定義情報+④除去・予防方法=品質管理計画

■ プログラム・設計者

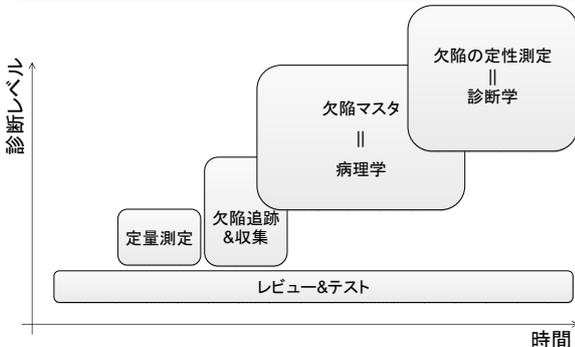
- ①定義情報+③検出情報=自己レビュー・自浄

■ テスト/レビュー担当者

- ①~④の全部!! どうやって?



欠陥エンジニアリングの未来



まとめ: 欠陥エンジニアリングの重要性

- まず“銀の弾丸”で倒すべきモンスターを知る
- 欠陥は単独では存在しない
- 欠陥は定量側面/定性側面がある
- 欠陥は企業にとっての財産である
- 欠陥を収集し、分類して蓄積する
- 欠陥の学習が人と組織を成長させる
- 欠陥を知る事から全てがはじまる

全ては安心のために