Detection of Requirement Errors and Faults via a Human Error Taxonomy: A Feasibility Study

Wenhua Hu

Jeffrey C. Carver

Vaibhav K. Anu

Gursimran S. Walia

Gary Bradshaw

10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement（ESEM ’16）

背景

- 様々な工程の中で特に要件定義はヒューマンセントリックな工程
  - 開発者と顧客が話し合って要件を抽出，要件定義書を作成

- 要件定義書の作成には問題が発生しやすい
  - 自然言語の曖昧さ
  - 全ての顧客と同時にコミュニケーションしないといけない難しさ
- 要件定義の段階で要件定義書の問題を修正しなければコストが非常にかかる
  - 修正の4,50%は開発工程の前半で起こった問題を対象[1]

[1] B. Boehm and V. R. Basili. Software defect reduction top 10 list. Foundations of empirical software engineering: the legacy of Victor R. Basili, 426, 2005.

関連研究

- ヒューマンエラーの概念を用いることでソフトウェア要件の品質が向上する[2]
- 要件定義で発生するヒューマンエラーの分類する研究がされている[3]
  - ヒューマンエラーを構造化することで欠陥の検知精度を向上することができる
  - 欠陥を含む予防策にもなる

- [3]は認知科学のヒューマンエラーの研究に影響され行われたが，既存のヒューマンエラーの分類とは関連性が低い

[2]M. E. R. F. Lopes and C. H. Q. Forster. Application of human error theories for the process improvement of requirements engineering. Information Sciences, 250:142–161, 2013.
[3G. S. Walia and J. C. Carver. A systematic literature review to identify and classify software requirement errors. Information and Software Technology, 51(7):1087–1109, 2009.

目的・アプローチ

- 目的
  - 認知科学のヒューマンエラーに基づいた要件定義における
    ヒューマンエラーの分類を定義する
    - Human Error Taxonomy（HET）

- アプローチ
  - ソフトウェア工学ドメインでのヒューマンエラーに関する知
    見を調査
  - 心理学のヒューマンエラーの知見と一致するものを抽出
    - 15件のヒューマンエラーが抽出
  - 認知心理学のヒューメンエラーの分類にマッピング

Human Error Taxonomy（HET）

- 15種類のエラーを大きく3つに分類

	概要	例
Slip	タスク自体を間違う，もしくは，間違った順番でタスク行うときに発生するエラー	シリアルに牛乳を入れるつもりがオレンジジュースを入れてしまう
Lapses	タスクを行う途中で目的を忘れてしまう，ルーティーンに従うことによって起こるエラー	部屋に入る前はやることを決めていたのに，部屋に入ると忘れる
Mistakes	そもそも設定しているタスクが間違っているためタスクを正しく実行しても結果が不適切になるエラー	カーナビに目的地を設定して運転していたが，そもそも設定した目的地が間違っていた

Slip

- タスク自体を間違う，もしくは，間違った順番でタスク行うときに発生するエラー

- Clerical Errors
  - 抽出した要件を仕様書に書く時に不注意で間違えることが原因

- Lack of consistency in the requirement specification
  - 要件を定義する人がはっきりと言うことができない，一貫して言うことができないことが原因
    - 解釈がバラバラ

Lapses

- タスクを行う途中で目的を忘れてしまう，ルーティーンに従うことによって起こるエラー

- Loss of information from stakeholders
  - ステークホルダーから得た要件を記録し忘れてしまう

- Accidentally overlooking requirements
  - ステークホルダーがこの要件は当たり前だろうと思い言い忘れる

Mistakes

- そもそも設定しているタスクが間違っているためタスクを正しく実行しても結果が不適切になるエラー

- Information Management errors
  - 標準操作の知識不足が原因
- Wrong Assumptions
  - 開発者が利用者の意見について誤った解釈をしている
- Low Understanding of one another’s Roles (LU)
  - 役割によって必要なドメイン知識が異なる

HET

Table 1: Human Error Taxonomy

Reason's Taxonomy	Human Error Taxonomy	Error Description
Slips	Clerical Errors (CE)	Due to carelessness while performing mechanical transcriptions from one format or from one medium to another.
	Lack of consistency in the requirement specification (LC)	occur when requirement authors are not able to articulate and organize the requirements in a consistent manner, even when they have a clear idea of user needs.
Lapses	Loss of information from stakeholders (LI)	Forgetting, discarding or failing to store information or documents provided by stakeholders.
	Accidentally overlooking requirements (AO)	Many requirements, both functional and non-functional, are accidentally overlooked.
Mistakes	Application Errors (AE)	Arise from a misunderstanding of the application or problem domain.
	Environment Errors (EE)	Due to a misunderstanding or misuse of the hardware or software environment of a given project.
	Information Management Errors (IM)	Due to a lack of knowledge about standard procedures.
	Wrong Assumptions (WA)	Happens here is creation of mistaken assumptions about features and opinions of stakeholders.
	Low Understanding of one another's Roles (LU)	Domain knowledge and perspectives vary between roles, which necessitates considerable communication among members of the software engineering team.
	Mistake belief that it is impossible to sppcify non-functional requirements in a verifiable form (MB)	Major causes of this problem are the prevalent myths that it is too costly, too difficult, and even impossible to produce good requirements, especially nonfunctional requirements, during the software engineering process.
	Not having a clear distinction between client and users (CD)	Fail to gather and analyze requirements from the userâĂŹs point of view.
	Lack of awareness of sources of requirements (LA)	Requirements gathering person is not aware of all stakeholders which he/she should contact in order to gather the complete set of user needs.
	Problem-Solution Errors (PS)	Due to not knowing, misunderstanding, or misuse of problem solution processes.
	Inadequate Requirements Process (IR)	The requirements engineering plan is incomplete or inadequate for the current project.
	Syntactic Errors (SE)	Syntax describes the grammatical rules for specifying correct arrangement of elements to produce meaning.

リサーチクエスチョン

- HETの実現可能性を確かめるために以下の3つのRQを設定

- RQ1：HETは要件定義で発生する間違いや欠陥を分類するのに役立つのか？

- RQ2：HETを用いることで発見できる欠陥は増えるのか？

- RQ3：開発者はHETを役立ったと考えるか？

実験方法

- 被験者：28名の学部生→6つのチームに分割
- それぞれのチームでシステムの要件定義書を作成

Table 2: Teams and System Description

Team #	System Name and Description	# Page	# Req
1	**Grade Keeper**: manage student course information	21	14
2	**Tenant Tracker**: provide communication between property managers and tenants	22	12
3	**Nutrition**: keep track of the food the users have eaten	15	7
4	**Opus**: share pictures of interesting items in the local community	14	7
5	**Robotics Score**: manage the scores for a robotics competition	15	19
6	**Promo Pass**: manage business promotion information for customers	22	4
7	**Bowling**: create and manage a bowling game session	17	13
8	**Is Open**: manage detailed information about local businesses	8	9

- 作成した要件定義書に含まれるエラーや欠陥をHETを使って特定できるか評価する
  - 特定したエラーの個数と0個の場合をt検定し評価する

実験手順

1. 各参加者に個別で違うチームが作成した要件定義書をインスペクションし，欠陥を特定してもらう
   - 予め欠陥の種類は伝えている
   - 特定した欠陥をリストにまとめる
   - 合計28個の欠陥リストができる
2. 1の結果を受け取り，チームごとの欠陥リストを作成する
   - 1の結果をマージする
3. チームごとにHETを用いてエラーを抽出・分類する
4. 3の結果を元に，要件定義書をHETを用いて再度インスペクションしエラーに関連する欠陥を特定する
5. 最後に4の結果を元にチームごとの欠陥リストを作成する

RQ1：HETは要件定義で発生する間違いや欠陥を分類するのに役立つのか？

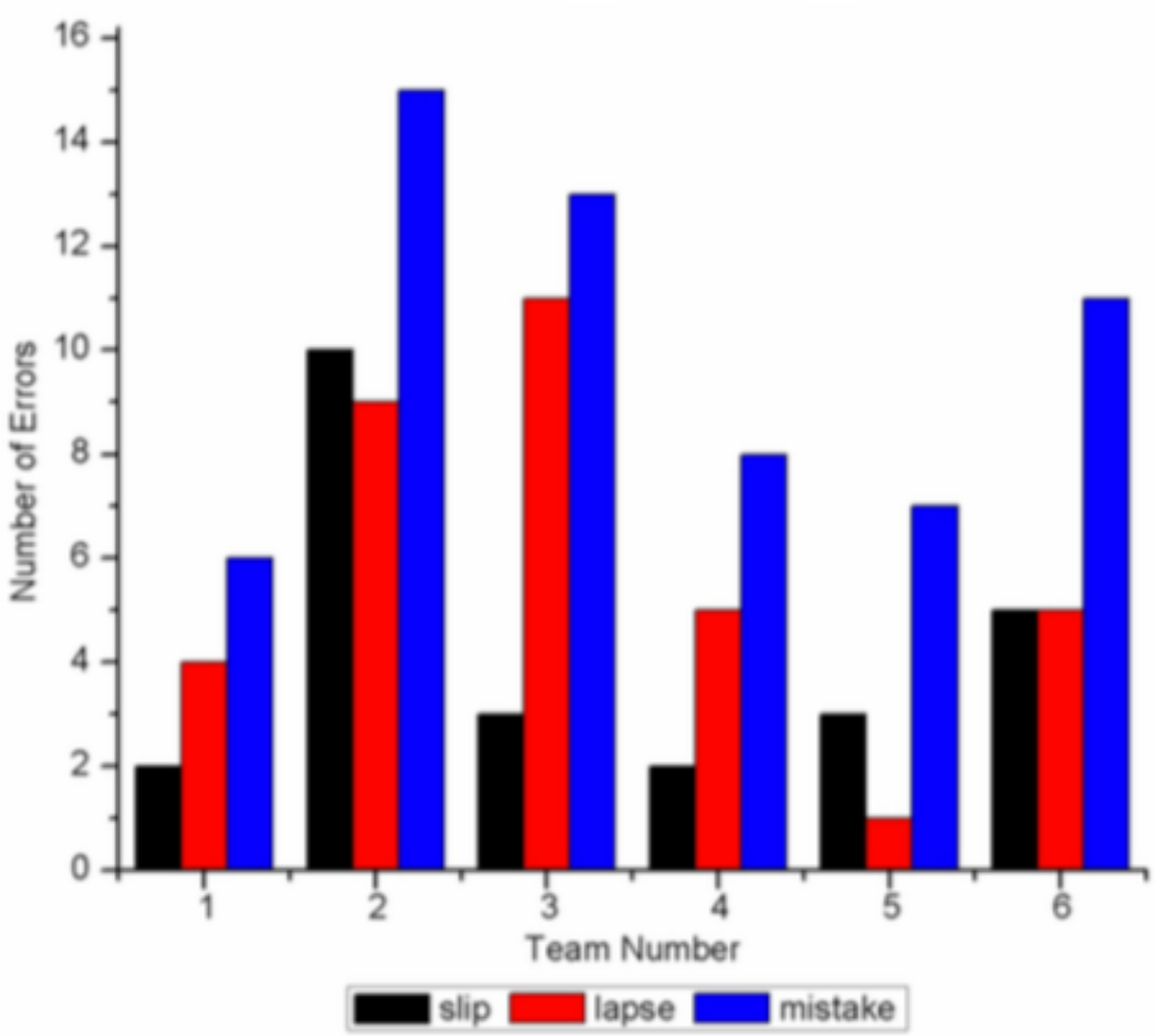


Table 3: Team Errors by Error Class

Error types	T1	T2	T3	T4	T5	T6	% of total errors
CE	-	2	3	1	3	-	8%
LC	2	8	-	1	-	5	13%
LI	-	4	3	-	-	3	8%
AO	4	5	8	5	1	2	21%
AE	-	-	1	-	1	-	2 %
EE	2	-	-	-	1	-	3%
IM	2	6	-	7	3	-	15%
WA	2	3	-	-	-	6	9%
LU	-	-	-	-	-	-	-
MB	-	-	10	1	1	-	10%
CD	-	-	-	-	-	-	-
LA	-	-	-	-	-	-	-
PS	-	-	-	-	-	-	-
IR	-	-	1	-	1	5	6%
SE	-	6	1	-	-	-	6%

shaded cells indicate result was significantly greater than 0

3つの大分類で統計的有意差を確認

また，表の黒で示しているエラーに関しても統計的有意差を確認

RQ1：HETは要件定義で発生する間違いや欠陥を分類するのに役立つのか？

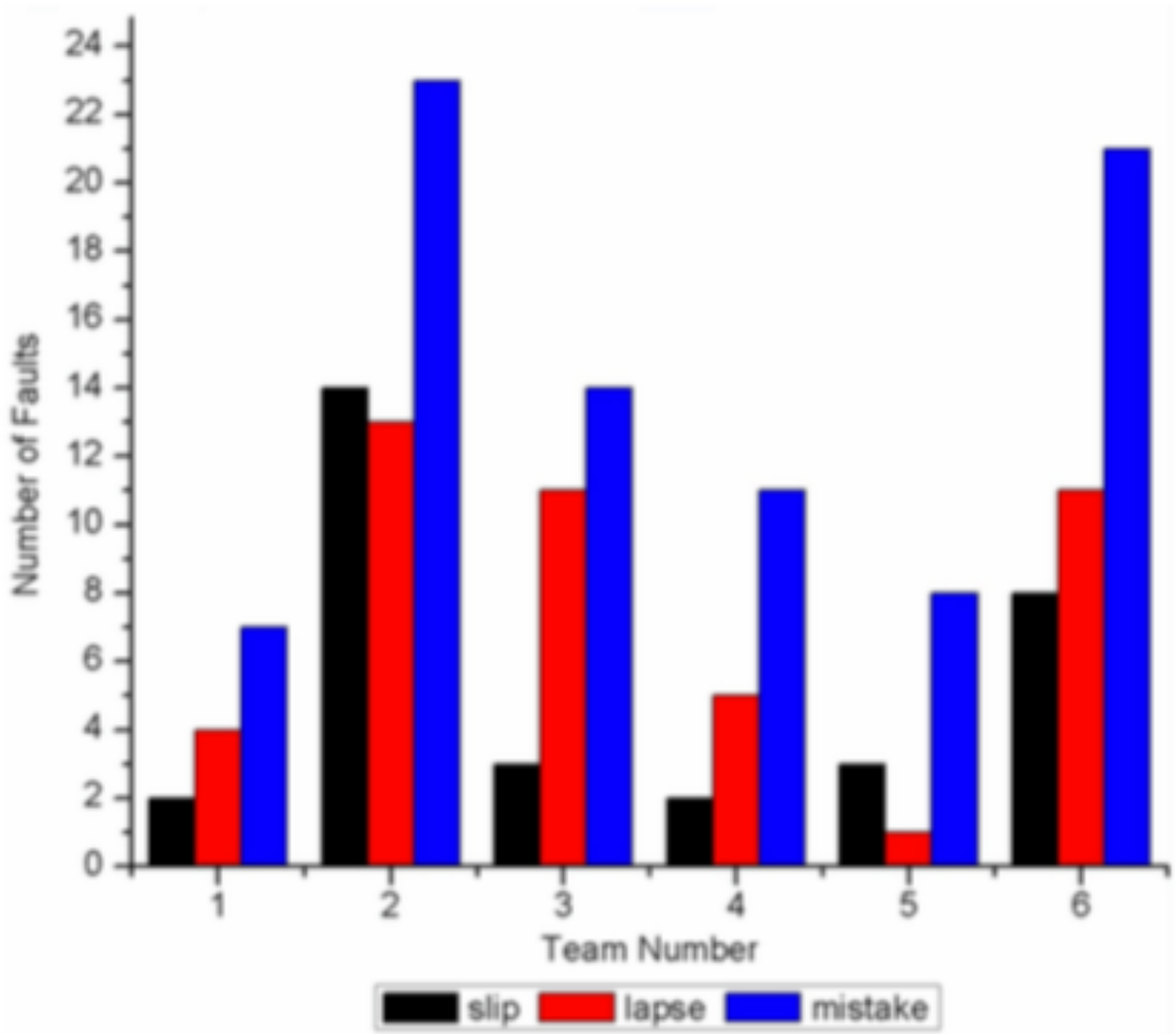


Table 4: Team Faults by Error Class

Error types	T1	T2	T3	T4	T5	T6	% of total faults
CE	-	2	3	1	3	-	6%
LC	2	12	-	1	-	8	14%
LI	-	8	3	-	-	8	12%
AO	4	5	8	5	1	3	16%
AE	-	-	1	-	1	-	1%
EE	2	-	-	-	1	-	2%
IM	3	7	-	10	3	-	14%
WA	2	9	-	-	-	9	12%
LU	-	-	-	-	-	-	-
MB	-	-	10	1	1	-	7%
CD	-	-	-	-	-	-	-
LA	-	-	-	-	-	-	-
PS	-	-	-	-	-	-	-
IR	-	-	1	-	2	12	9%
SE	-	7	2	-	-	-	6%

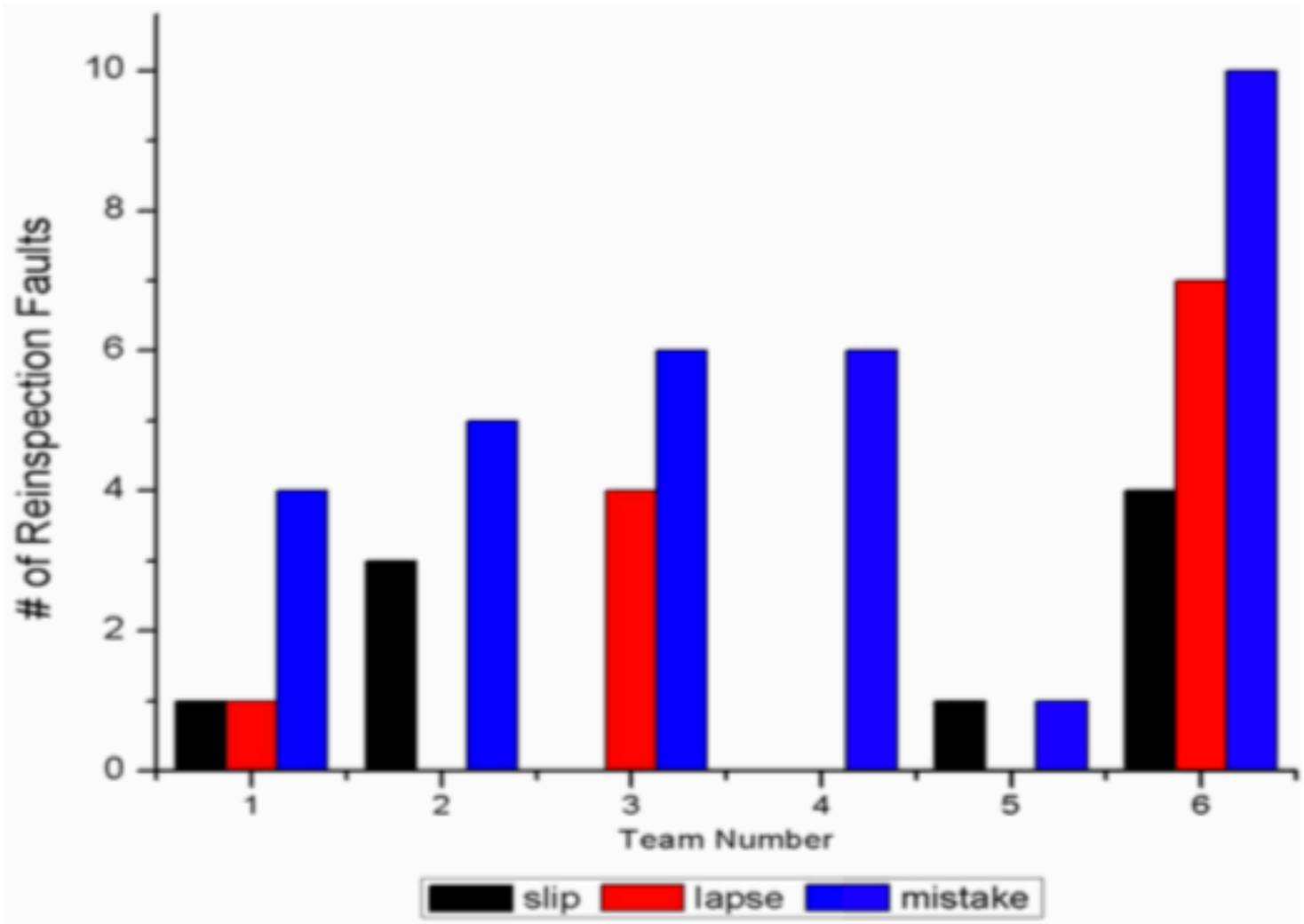
shaded cells indicate result was significantly greater than 0

3つの大分類で統計的有意差を確認

また，表の黒で示しているエラーに関しても統計的有意差を確認

RQ2：HETを用いることで発見できる欠陥は増えるのか？

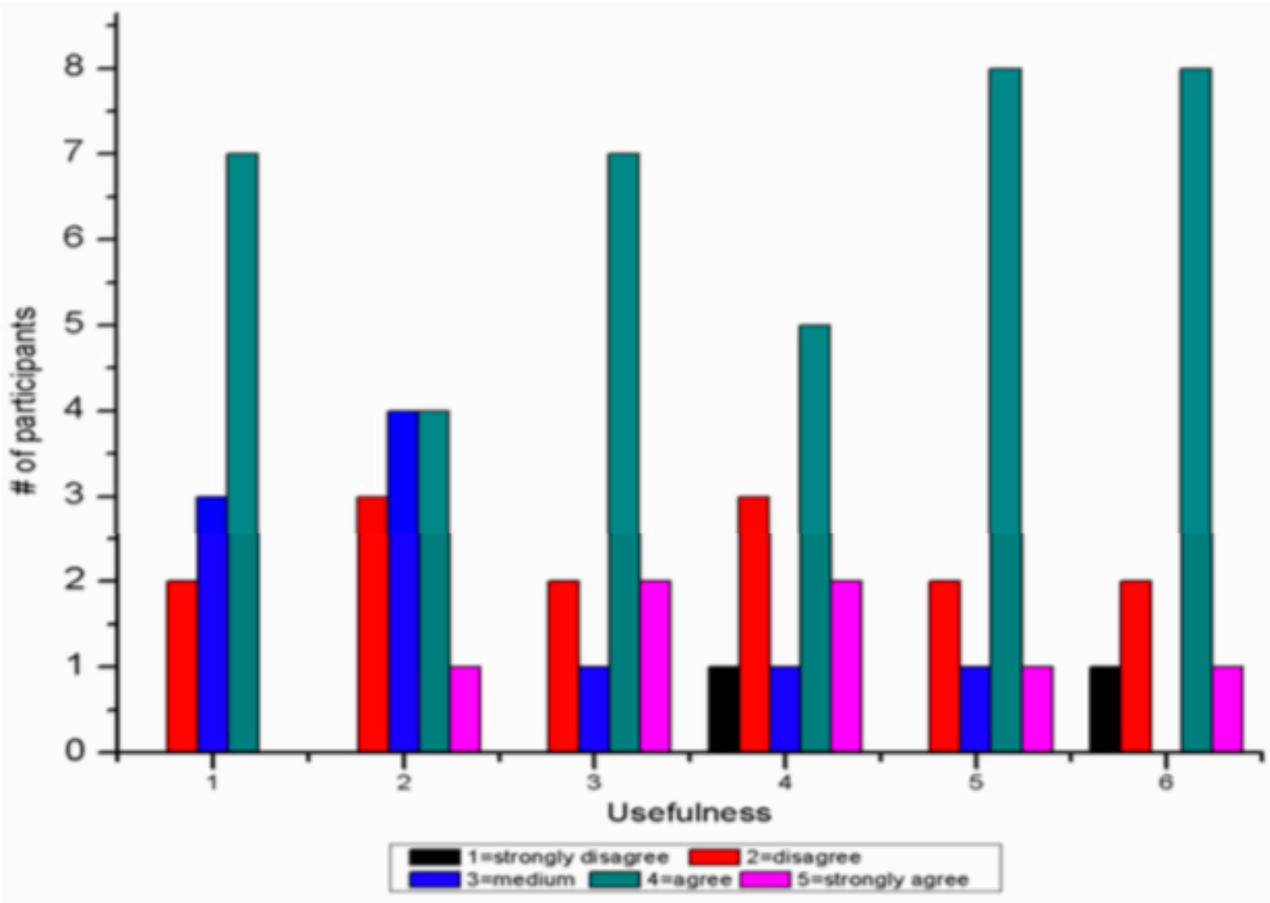
- 実験のステップ4で発見した欠陥を調査

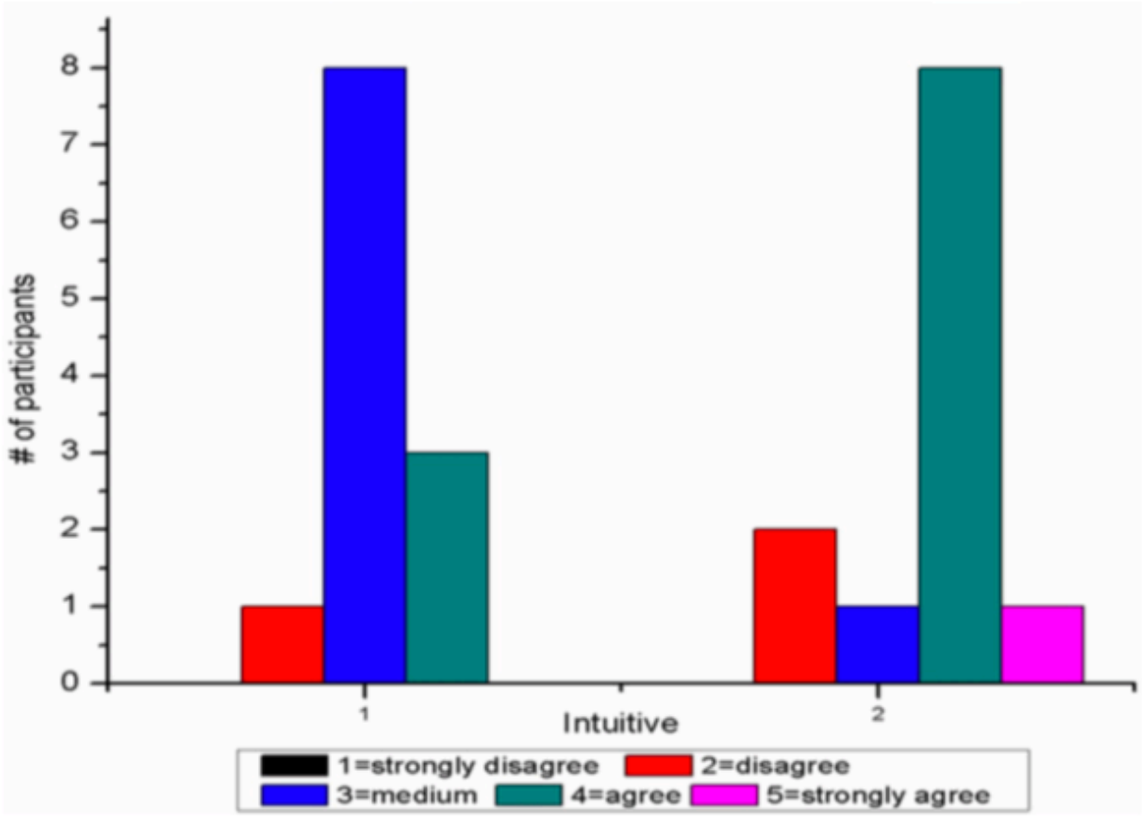


SipsとMistakeで統計的有意差を確認

RQ3：開発者はHETを役立ったと考えるか？

- ほとんどの学生が直感的で役に立つと回答





まとめ

- まとめ
  - 要件定義で発生するヒューマンエラーHETを定義した
  - HETを用いることで，要件定義中に発生したエラーや欠陥を抽出することができる

- 所感
  - 要件定義の欠陥に関する分析は珍しくて新鮮だった
  - 要件定義で問題が生じやすい，という点は引用できそう