

ソフトウェアのバグ等の問題を見つける ヒューマンファクターアプローチ

システム情報系 教授 伊藤誠

概要

人が開発するソフトウェアについて、バグなどの問題点が含まれる可能性の高いところを、ソースコードを見ることなく特定できます。従来のソフトウェア信頼性手法と大きく異なる新しい方法論です。

Carlos Andres Ramirez Catano, Makoto Itoh: "Actionable Insights from Developer Behavior: A Practical Approach to Software Defect Prediction," Automated Software Engineering, 33, 93 (2026).
<https://doi.org/10.1007/s10515-026-00642-9> IF=3.9 (2025)

1. なぜソースコードを見なくていいのか

- Git metadata に含まれる情報のみを利用します。
- 人間信頼性工学のError Forcing Context (EFC) の考え方をういて、Human Error (HE) を犯す環境要因を反映した説明変数を用います。
 - 前回更新からの時間経過による忘却 (Fig. 1)
 - 生体リズムによる注意力 (Fig. 2)

$$P(HE)=P(EFC)*P(HE|EFC)$$

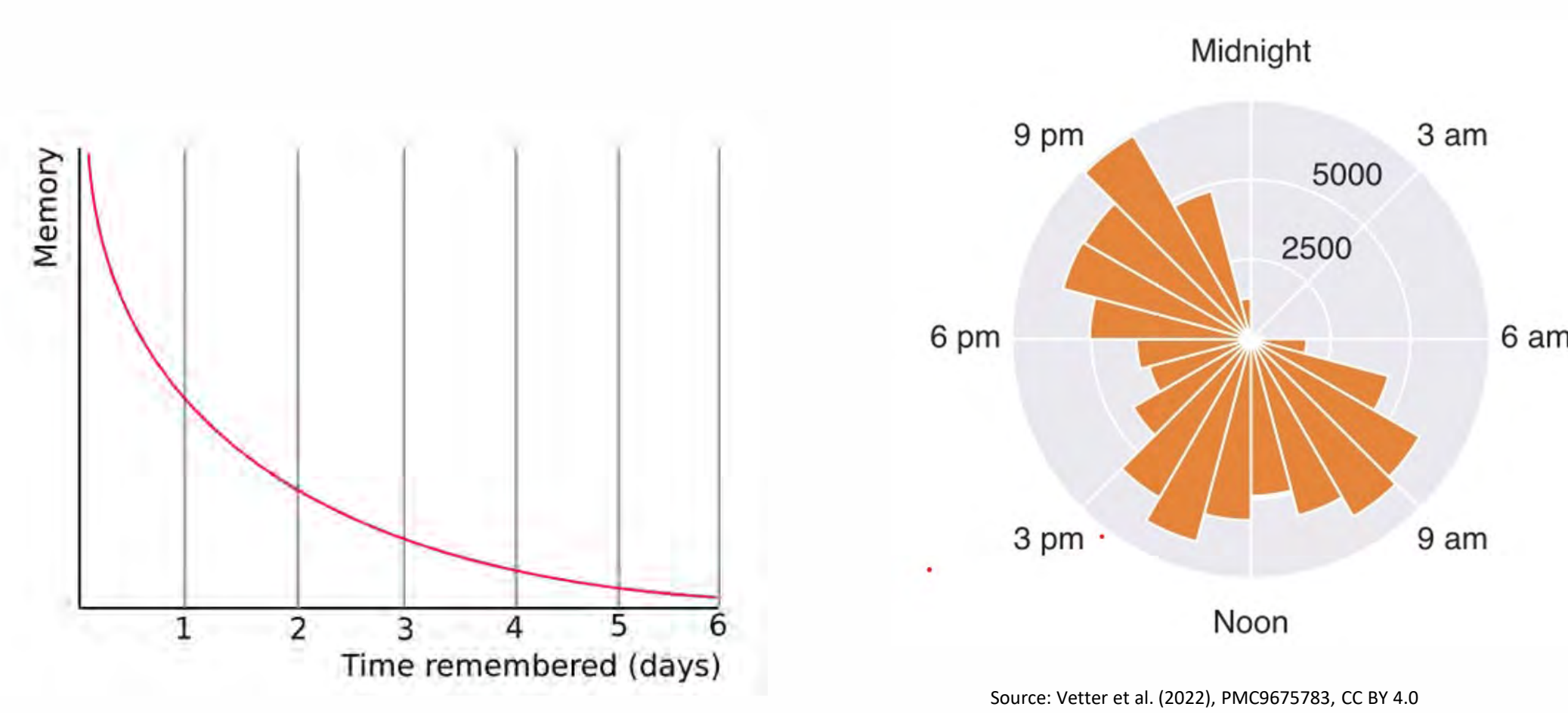


Fig. 1 忘却 Fig. 2 生体リズム

2. 本当にうまくいくのか

- 21のプロジェクトを対象として、様々な角度から検証をしました。
- 本提案の2指標は、プロジェクト間のばらつきが小さく、予測力も高い結果を得ました (Fig. 3)

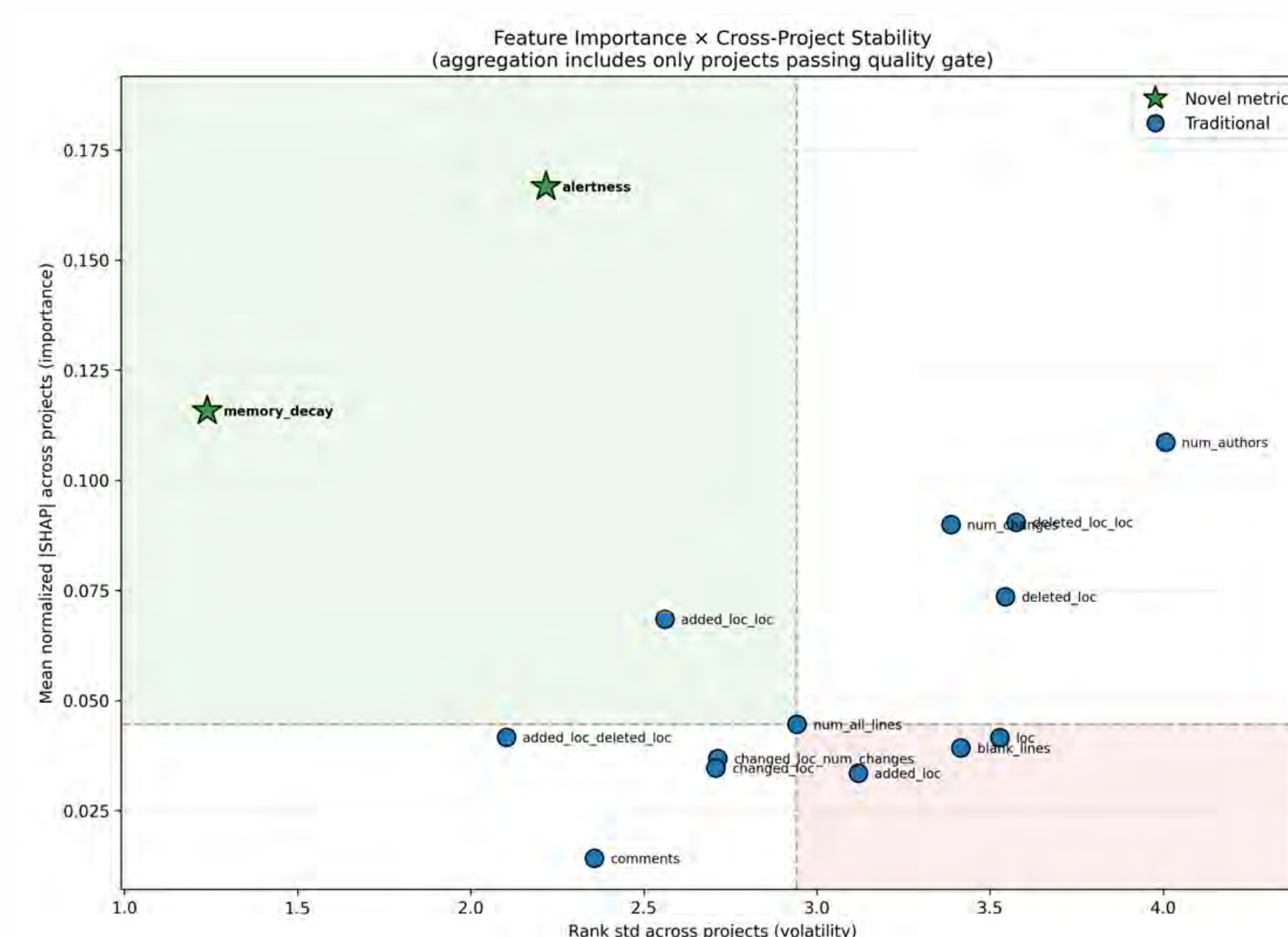


Fig. 3 説明変数の効果

3. 応用例

- NASAのプロジェクトなど、実用されているソフトウェアのバグ発見・修正の事例が多数あります。(Fig. 4はその一例)

4. 社会実装へ向けて

- 従来手法と異なり、なぜエラーが発生したかの説明がしやすく、対策を講じやすいといえます。
- AI利用のコーディングなどへも応用できる見通しを立てており現在検証作業進めています。

Javadoc Discrepancy in SortByID() Method – Copied from SortByMeanValue() #1986



Fig. 4 実ソフトウェアでの問題検出例

ポスターデータ
ダウンロード用

