

Code Today, Deadline Tomorrow: Procrastination Among Software Developers

Zeinabsadat Saghi*, Thomas Zimmermann[§], Souti Chattopadhyay*

*Department of Computer Science, University of Southern California, Los Angeles, CA, USA

[§]Department of Informatics, University of California, Irvine, CA, USA

2025 IEEE/ACM 47th International Conference on Software
Engineering (ICSE)

背景 | 『先延ばし』とその研究

先延ばしという，身近な現象に関する研究が行われている

例：大学生の先延ばし

- 学生の80~95%が，時々先延ばしを経験
- 20~50%が慢性的な先延ばし傾向
- ストレス増加，成績低下などの影響

一方，ソフトウェア開発/開発者の文脈では，

- 先延ばしの影響が体系的に研究されていない
- 開発現場での実態が不明
- 対策も経験論に頼っている状況

➤ 開発者特有の先延ばしを解明する必要性

動機 | 開発者特有の影響（仮説）

一般的な先延ばしに比べ、開発者特有の要因や影響があるのでは？

- 抽象的かつ複雑なタスク特性による要因
- チーム活動・協働という特性による要因
- 技術的側面での影響
- チーム・プロジェクトへの影響 など

➤ 開発者特有の理解と支援が必要？

目的

開発者の先延ばし行動の包括的理解と実践的支援

- 先延ばしによる影響の解明
- 先延ばしの要因の特定
- 先延ばし管理手法の調査

➤ 最終的には、開発者/チームの生産性向上に貢献

アプローチ

- 15名の開発者への半構造化インタビューを実施
 - 7業界・多様な経験レベルの開発者15名
 - 90分間の詳細インタビュー
 - 質的分析による体系的なテーマ抽出

研究プロセス①

①事前調査

- 既存文献から、先延ばし要因リストを作成
 - 心理学・CS分野の先延ばし研究を計36論文調査
 - 既存の先延ばし要因や、開発者特有の要因の新規仮説を設定
 - 既存例：タスク興味度，不明確性，作業負荷 など
 - 新規例：タスクの曖昧性，重要性，学習要求度 など
- ✓ インタビューで聞くべき要因を事前に整理

②半構造化インタビューの実施

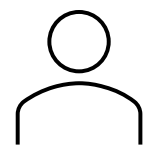
- 90分間実施，録音・転写・匿名化
 - 先延ばしの定義→具体例→要因確認→対策 の流れ
- ✓ 開発者の実体験と先延ばしの実態を深く理解

研究プロセス②

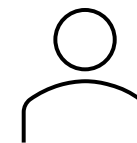
③収集した大量の発言を整理，共通パターンを導出

- 15人×90分の発言から，先延ばしに関する部分を抽出
- 似たような体験・意見を分類，共通パターンを発見

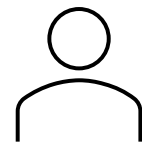
例：個人の発言→一般的知見



急いで品質が下がる



バグが増える



テストを省略



研究者

→先延ばしは“個人パフォーマンス低下”を引き起こす？

実験結果として体系的に整理

- RQごとに適した分類軸で整理

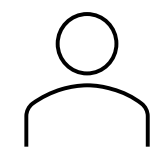
RQ1 | 先延ばしが多たらす影響

先延ばしは、開発者に負の影響を与える

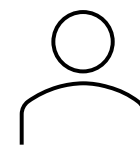
- 5カテゴリ, 14の負の影響を特定 (SPACEフレームワークを活用)

1. 満足度・幸福感への影響

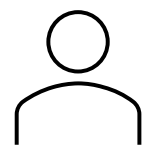
- 感情的苦痛
- 不安とストレス
- ワークライフバランスの低下


P9, 15

“罪悪感”


P3

“膨大なタスクによるストレス”

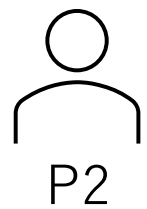

P2

“先延ばしを頻繁にやりすぎると、だらしなくなる”

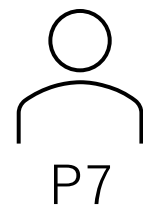
RQ1 | 先延ばしが多たらす影響

2. パフォーマンスへの影響（最も深刻）

- 個人パフォーマンスの低下
- 技術的負債の増加，期限遅れ
- チームパフォーマンスの低下



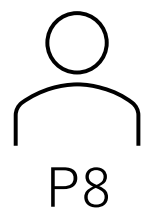
“品質・生産性に影響”



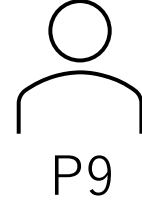
“チーム・ビジネスに影響”

3. 活動への影響

- コード生産量の減少



“少ないコード・機能の完了”

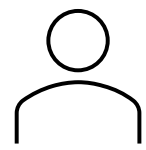


“コメント量減少”

RQ1 | 先延ばしがもたらす影響

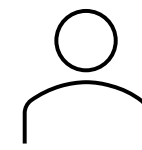
4. コミュニケーション・協調への影響

- チーム文化の悪化
- 信頼の弱体化



P13

“その人のイメージ・評判に影響”

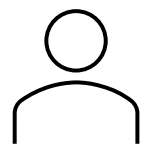


P12

“信頼を失う”

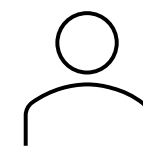
5. 効率・フローへの影響

- 補填的努力
- ワークフロー中断
- 連鎖的遅延



P8

“補填のために夜遅くまで働く”



P10

“次のフローに影響”

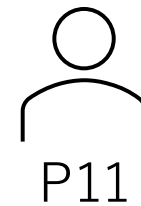
RQ1 | 先延ばしが多たらす影響

一方、先延ばしには正の効果もある

- 8の正の影響を特定

1. 満足度・幸福感の向上

- 休憩
- 短期的ストレス軽減
- 個人タスクの時間



“あえて一歩離れて、
新鮮な心構えで挑める”



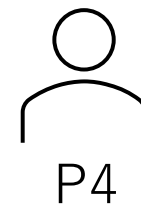
“日中、5~10分の休憩
が非常に有益”

2. パフォーマンスの向上

- 創造性の向上

3. 活動の最適化

- 不必要な作業の回避

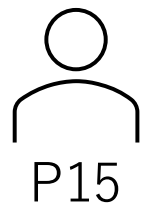


“自分のタスクが他人に
割り当てられるケース”

RQ1 | 先延ばしがもたらす影響

4. コミュニケーション・協調の向上

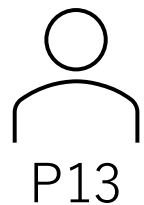
- 職場での社交



“同僚と会話による先延ばしは、社会的つながりの改善”

5. 効率・フローの向上

- 期限近くでの効率



“確実に、期限直前が作業効率のピーク”

RQ1 | 先延ばしがもたらす影響

4. コミュニケーション・協調の向上

• 職場での社交

開発者の先延ばしは、正負の両方の影響をもたらす

- パフォーマンス, コード品質, 創造性など, 開発者特有の影響を特定
- 先延ばしは, 単に悪い習慣とは限らない

➤ **先延ばしの排除ではなく, 適切な管理が重要**

P13

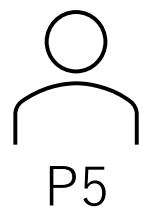
RQ2 | 先延ばしを引き起こす要因

先延ばしを引き起こす要因を3カテゴリに分類

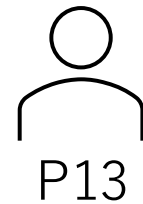
- 22の要因を特定

1. タスク関連要因（最多言及）

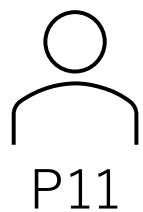
- タスク興味度
- 主観的・客観的重要性
- タスクの難易度・曖昧性



“文書作成は後回し”



“重要でないタスクは後回し”

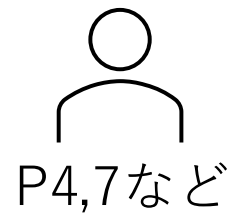


“タスクが難しすぎるor簡単すぎると先延ばし”

RQ2 | 先延ばしを引き起こす要因

2. 個人要因

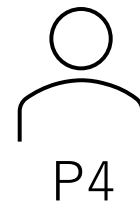
- 精神状態
- 不確実性
- 失敗への恐怖



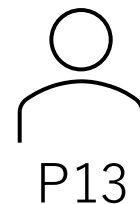
“気分が乗らない時は何もできない”

3. 外部要因

- 気晴らし
- コミュニケーション問題
- 他者依存



“チームとの調整不備で”



“他人の助けが必要な時”

RQ2 | 先延ばしを引き起こす要因

2. 個人要因

- 精神状態
- 不確実性



“気分が乗らない時は何もできない”

開発者特有の先延ばし要因を特定

- 特に、タスクに対する認識（興味関心，重要度，難易度）が大きく影響
- 個人・外部要因も複合的に影響

コミュニケーション問題



“他人の助けが必要な時”

- 他者依存

P13

RQ3 | 先延ばしをどのように管理する？

開発者が行っている先延ばし管理法を，4カテゴリに分類

- 14の実践的緩和技術を特定

認識・意識（Awareness）

- 進捗パターン追跡
- 積極的対策
- 他者への遅延通知

タスクフォーカス

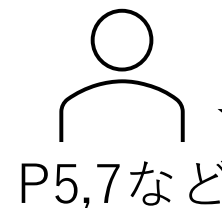
- 日々のタスク計画を立てる（最多）
- 好き・嫌いタスクのバランス
- タイムボクシング/偽の期限
- 気晴らし・誘惑の除去



“過去の先延ばし傾向を参考”



“遅延を早期に伝える”



“タスクの優先順位付け”

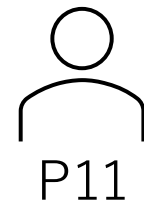


“タイムボクシングの適用”

RQ3 | 先延ばしをどのように管理する？

タスク戦略

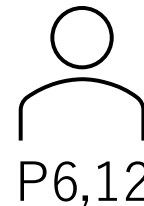
- タスクの分解・細分化
- タスク依存関係の回避



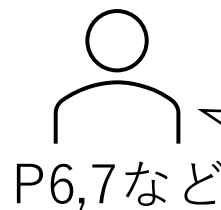
“タスクを細かく分解しより良く管理”

チームサポート

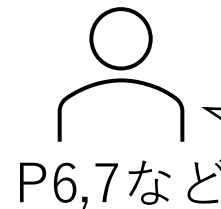
- 適切な期限設定
- 進捗ミーティング
- 作業中断の管理
- 助けを求める



“コミュニケーションをとって設定”



“週一のアジャイルミーティング”



“他人からの助けがタスク課題を解決”

RQ3 | 先延ばしをどのように管理する？

タスク戦略

- タスクの分解・細分化



“タスクを細かく分解しより良く管理”

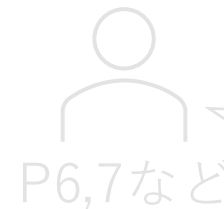
タスク計画やチームでの支援が重要視

- あらかじめ明確で適切な計画を立てることで、後回しを低減
- 個人・外部要因に対して、他者の介入が有効

- 作業中断の管理
- 助けを求める



“週一のアジャイルミーティング”



“他人からの助けがタスク課題を解決”

議論① | 先延ばし支援のためのツール

JIRAやGitHubなどには、カレンダーやガントチャートあり

- RQ3で、タスク計画・開発者への介入の重要性を特定
- 既存ツールにはすでにそれを補助するツールがあったりする

一方、先延ばしの認識との統合は不十分

- RQ3で登場した手法全てをカバーできるツールがあるわけではない

➤ 先延ばしの原因・管理手法に基づいたツール開発をすべき

- 開発者のスキルや関心，パフォーマンスに基づいてスケジュール適応を行えるAI駆動システムなど

議論② | ソフトウェア開発の独自性

他分野では特定されなかった影響・要因

- チームパフォーマンス低下
- 技術的負債の増加
- タスクの主観的重要性 など

他分野との矛盾

- 他分野：先延ばしとパフォーマンスに有意な相関なし
- 開発：明確なパフォーマンス低下が報告

➤ チーム依存性の高さ， 技術的複雑性など独自性によるもの？

まとめ

開発者における先延ばしの影響・要因を調査

先延ばしには、正負両面の影響があることを特定

開発者独自の先延ばし性質があることを特定

軽減手法などの知見を活かし、先延ばし対策やそのツール開発に貢献

所感

テーマは面白かった

- 先延ばしは、身近な現象どころか、日常
- ソフトウェア開発環境独自の性質があるのも面白い
- 他分野の人間と開発者の性格の違いとかも出てそう

計画を立てることを後回しにする場合はどうしたらいい？

- 対策手法は楽しみにしていたが、まあ定番っちゃ定番
- 真の怠惰は後回し対策を後回しにするのさ