

Summary Review: With Great Humor Comes Great Developer Engagement

Paul Wooseok Lee
University of Waterloo
Waterloo, Canada
w69lee@uwaterloo.ca

1 What is the Problem Being Solved?

- New perspective: exploration of humour as a tool for engagement and collaboration in software engineering.
- How is humour used in software engineering?
- What impact does humour have in software engineering?
- What do people think about humour in software engineering?

2 What is the New Idea they are proposing?

The paper explores the role of humour in software engineering, proposing it as a tool to boost developer engagement and collaboration. By analyzing three case studies: Faker, Lolcommits, and Volkswagen, the authors highlight how humour can be integrated into different stages of software development.

(1) Case Studies

- Faker - A library designed to generate fake but realistic data for software testing. Beyond its practical functionality, Faker incorporates humorous elements (e.g., fake names or pop culture references) to make testing more engaging and enjoyable for developers.
- Lolcommits - A tool that captures a photo of developers each time they make a commit. This adds an element of fun and tradition to the routine task of committing code, fostering collaboration and team bonding.
- Volkswagen - A satirical software inspired by the real-life Volkswagen emissions scandal. It humorously tricks continuous integration pipelines into marking all tests as passed. This case led to the development of the legitimate tool, IS-CI, which helps detect CI environments.

(2) Developer Survey - The paper conducted the first developer survey focused on humour in software development, collecting responses from 125 developers. Key findings include:

- Over 93% of respondents had positive reactions to humour in software.
- Humour is most appreciated in comments, documentation, and test inputs but less in class and variable names.
- Developers cited humour as a tool for fostering community, aiding understanding, and making challenging tasks enjoyable.
- Some developers emphasize the need for responsible use of humour to avoid negative impacts on code quality.

3 Class Discussion

During the class discussion, various perspectives were shared on the topic of humour in software engineering, focusing on its appropriateness and impact across different fields.

(1) Appropriateness

- Some students raised concerns about the survey's limited scope, as it primarily included responses from English-speaking developers on platforms like Reddit and Discord, which may not represent the perspectives of non-English-speaking communities, such as French or Chinese developers. The class noted that humour is highly subjective, with individual and cultural differences affecting what people find funny. Even though inclusive humour tries not to offend anyone, people were worried that thinking about it too much or forcing it in a strict way could kill creativity and make some people feel left out.
- Students highlighted that humour should emerge naturally within an organization, such as through shared experiences or inside jokes. For example, "remember when" stories, while considered a simple form of humour, can be an effective way of bonding. However, creating a universally humorous environment is challenging, as not everyone may share the same sense of humour.
- The professor contributed to the discussion by sharing a personal story. As a "Graduate Advocate," he placed a humorous sign outside his office reading "Graduate Advocado," complete with a drawing of an avocado. While this may amuse some students (no feedback, unfortunately), the professor expressed concern that others with serious concerns might not find it appropriate. This example underscored the importance of considering the context and audience when incorporating humour into professional environments.

(2) Humour Across Different Fields

- The class also discussed whether humour in the workplace is unique to software engineering or has broader implications. It was noted that the use of humour is not exclusive to software development and likely exists in other fields. Students suggested that the paper could have been strengthened by referencing studies from other disciplines, such as business or creative industries, where humour may already be recognized as a tool for productivity and engagement.

4 Positives and Negatives

4.1 Positives

- New insight to the field of software engineering on how humour creates positive effects through multiple case studies and surveys.
- The paper did a great job promoting humour as a strong weapon for software engineers but it also went over some potential consequences as well.
- The paper also highlights impacts beyond software such as positive effects on organizational culture.
- The paper is Very relatable as a software engineer.

4.2 Negatives

- Although the paper went over some potential consequences that developers mentioned through the survey, more concerns could have been observed.
 - What about security and privacy issues?
 - Cultural sensitivity?
- It lacks in-depth analysis on how effective humour is to software engineering but the paper provides a solid foundation for future research in humour-based software engineering.

More discussion of the above points is in section 5.

5 Future Work

- Potential case study that covers failures/backfires/harms in the usage of humour in software engineering settings:
 - Where's the line? Some form of humour can offend someone.
 - Is humour for everyone? Some may not agree with using humour openly in a professional setting.
 - Security and privacy risk? Users of Lolcommits may leak company secrets when photos are taken. They could also take pictures of people who may not want to be in the picture.
 - Inclusive? What about people with different cultural backgrounds? They may feel left out.
- Performance analysis between teams with humour vs no humour. It will be interesting to compare and analyze two control groups where one team works on a project with humour while the other works on a project with minimal humour.
- Comparison analysis between different fields of software engineering. Different fields may require different levels of humour or none due to many factors.
 - Finance? Money is involved so can we risk using humour?
 - Healthcare? Medical histories and lives are involved which may require more of a serious atmosphere.
 - Gaming? It is common to see more humour in the gaming community than in others. For example, game developers tend to leave many Easter eggs.

6 Rating (out of 5)

I would rate this paper as 5 out of 5 because of its introduction to innovative and new perspectives on the usage of humour in software engineering with interesting case studies.

7 Discussion Points

- Should there be a guideline for incorporating humour into software engineering? How do we ensure humour is respectful and inclusive, especially in global communities? How about a potential security or privacy breach?
- Do you think humour degrades code readability over time? A new generation of coders may look at codes that the older generation has written and they may not understand the humour (Boomer!).
- Points from Future Works:
 - Humour in different fields? (Finance, Healthcare, Gaming, etc.)
 - Thoughts on the gap of performance between an SE team with humour and no humour.