Time to branch out: An analysis of online user forum posts to inform Computer-Aided Design (CAD) branching

Kathy Cheng[1]*, Phil Cuvin[1], Dr. Shurui Zhou[2], Dr. Alison Olechowski[1]

[1]Department of Mechanical & Industrial Engineering, University of Toronto
[2]Department of Electrical & Computer Engineering, University of Toronto

*kathy.cheng@mail.utoronto.ca

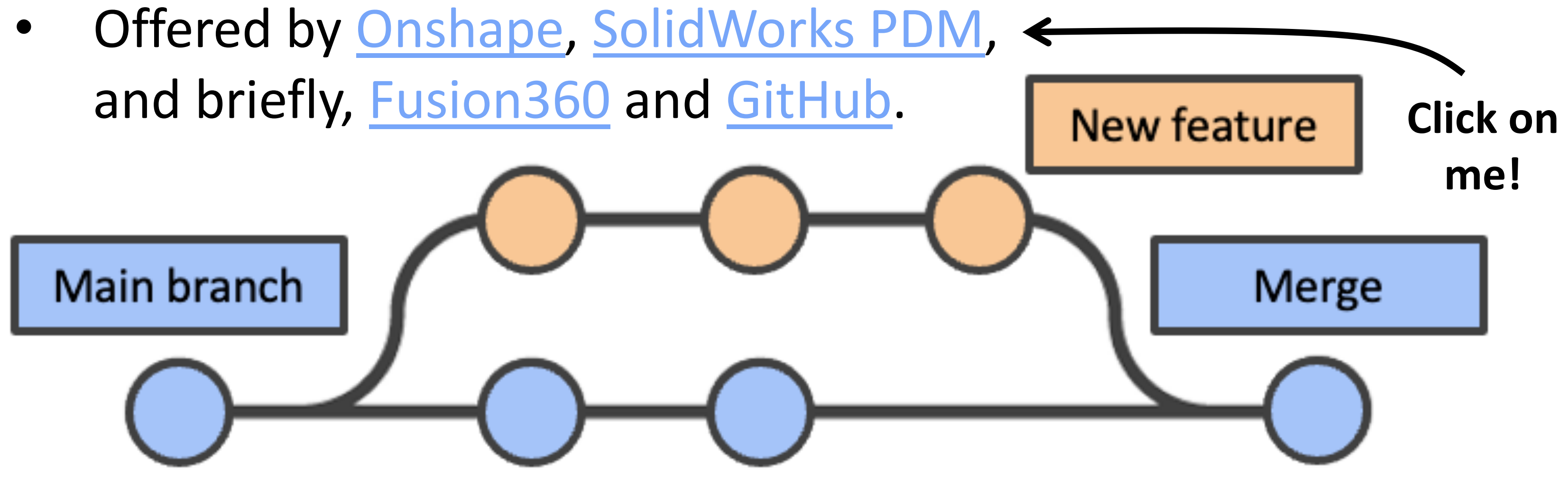
Background

Branching and Merging in Software Development

- Branching and merging is a feature of distributed version control systems (e.g., git) for software development.[1]
- Developers create branches to work with a copy of the code without modifying the existing version.
- Branches are used to develop parallel versions, isolate the risk of code changes, and enable collaboration.

Branching and Merging in CAD

- Is not a new concept in theory[2], but has not been studied from a practical standpoint, from the user's perspective.
- Offered by Onshape, SolidWorks PDM, and briefly, Fusion360 and GitHub.



Research Questions

RQ1: Why is branching used for CAD?

RQ2: What are the design shortcomings and gaps of existing branching tools?

Methods

1. Scrape text from online CAD forums (5 sites)
2. Qualitatively analyze forum post data (14,448 posts)
3. Develop branching use case taxonomy (719 posts)

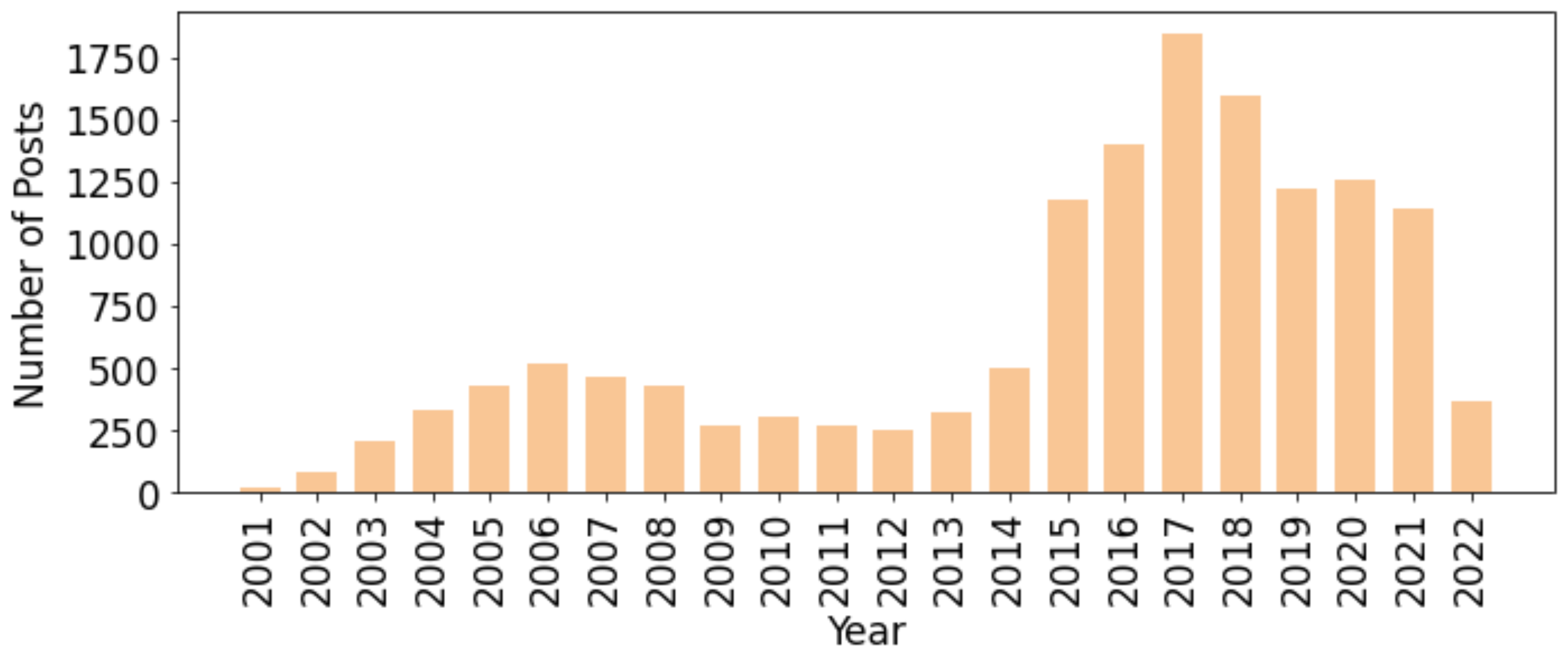


Figure 1. Number of forum posts collected from February 2021 to April 2022.

Taxonomy of Branching Use Cases (RQ1)

Product Line Management: serve the release/mainline version
Risk Isolation: isolate unapproved or unverified file changes
Designer Support: fulfill organizational or non-technical needs

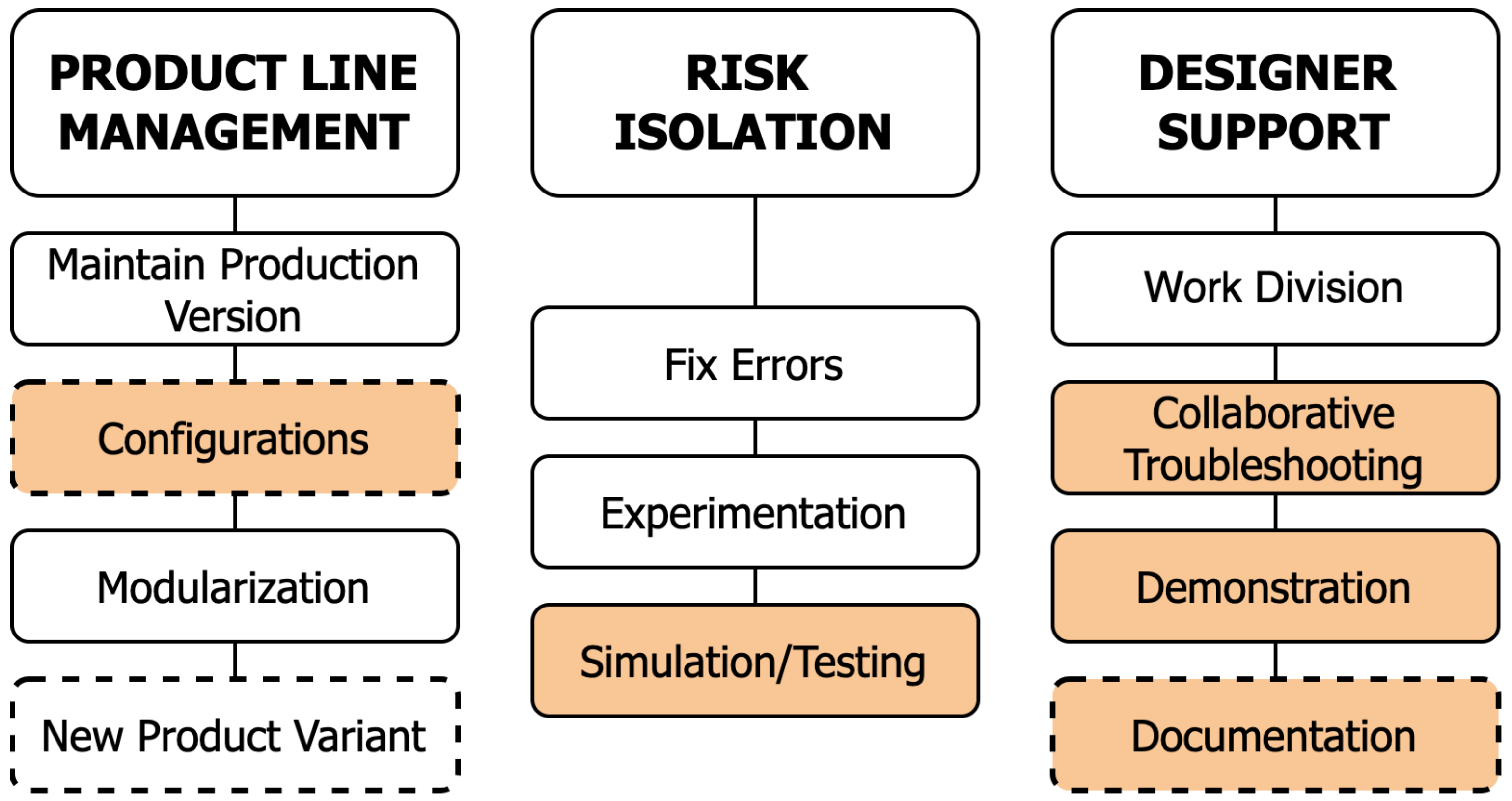


Figure 2. Taxonomy of CAD branching use cases. Orange use cases have not been previously mentioned in literature. A dashed border indicates the use case is an intended function of branching, but it is used as a workaround.

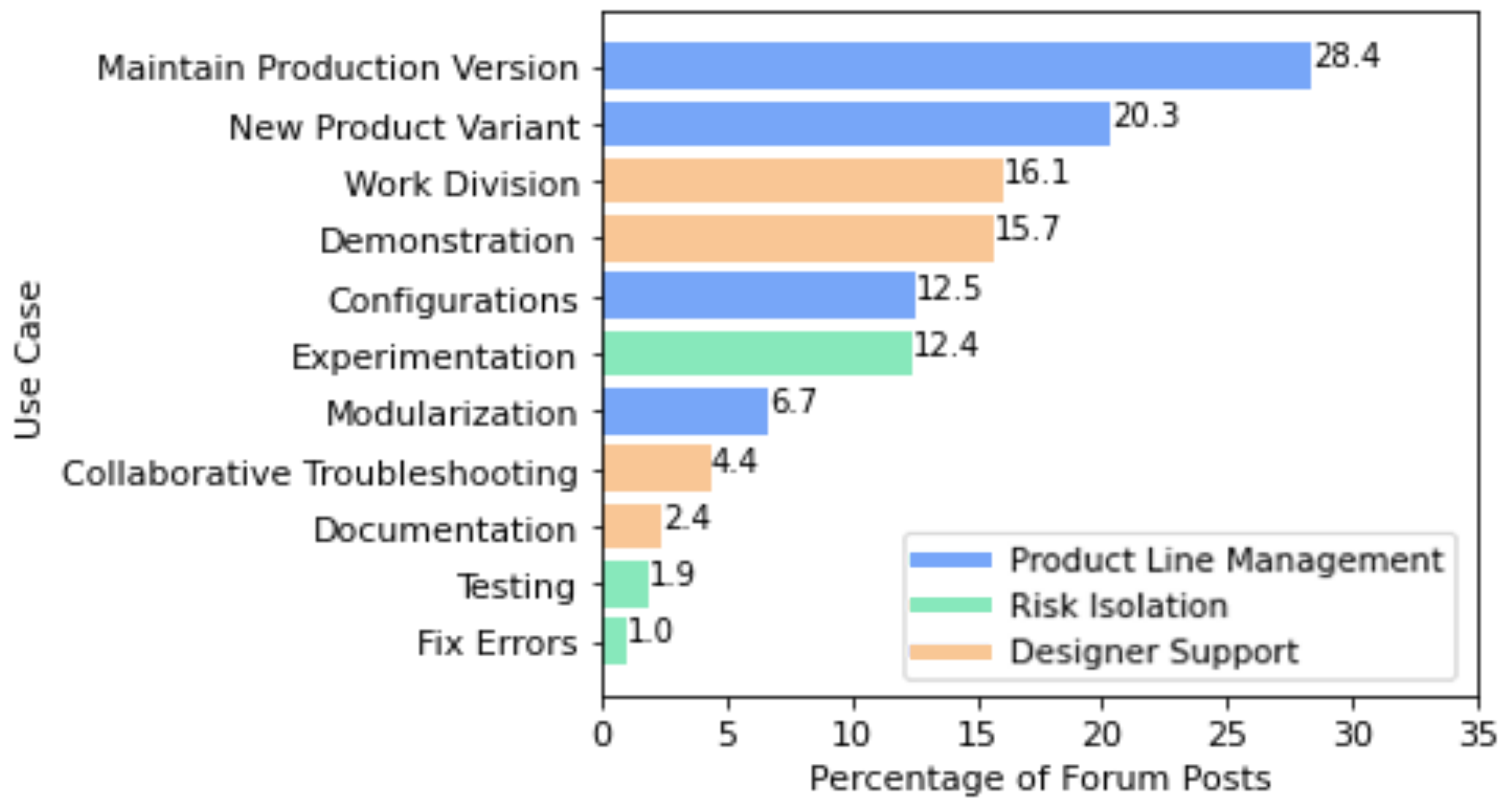


Figure 3. Frequency of mentioned use cases in the forum threads (n = 719).

Hardware vs. Software

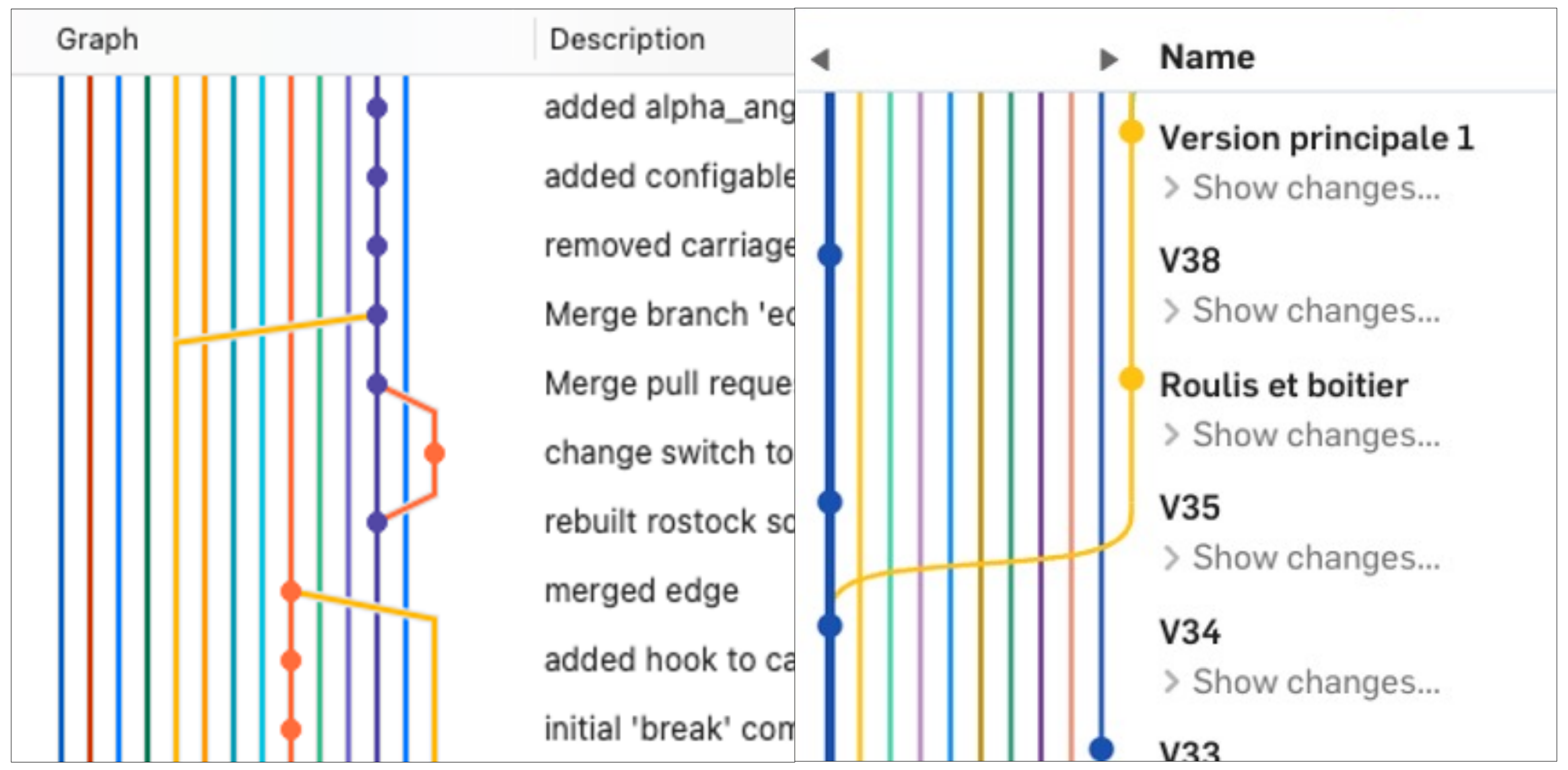
Hardware and software share common branching use cases, but there are some differences in branching practices:

1. CAD branches are often made with **no subsequent merge.**
2. Experimenting with **multiple alternative/candidate designs** is standard practice in CAD, but not in software.
3. Branching to **fix design errors is much less common** in CAD than in software (where bug fixing is widespread).

Shortcomings of Branching Tools (RQ2)

Product Line Management

- Ability to clean/prune the branch history (like git-rebase)
- Copy over notes or history to New Product Variant branch
- Poor visualization and navigation of long branch history:



added alpha_ang
added configable
removed carriage
Merge branch 'ec
Merge pull reque
change switch to
rebuilt rostock sc
merged edge
added hook to ca
initial 'break' com

Version principale 1
› Show changes...
V38
› Show changes...
Roulis et boitier
› Show changes...
V35
› Show changes...
V34
› Show changes...
V33

Figure 4. Comparison of software (left) and CAD (right) branch history, showing the challenge of poor visualization and navigation.

Risk Isolation

- Lack of branch granularity at the part or sub-assembly level
- Ability to selectively merge changes, or "cherry-picking"

Designer Support

- Configure specific access permissions (e.g., view only, copy only, edit, ownership, and branch- or version specific access)

Research Implications

Tool Builders

Address design shortcomings in existing branching tools

Practitioners

Increase awareness of branching to improve design

Researchers

Study further analogies between CAD and software

References

[1]S. Phillips, J. Sillito and R. Walker, "Branching and merging: an investigation into current version control practices", *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, 2011

[2]H-T. Chou and W. Kim, "A Unifying Framework for Version Control in a CAD Environment", *Proceedings of the 12th International Conference on Very Large Data Bases*, San Francisco, CA, USA, 1986