# Code Story: Information Tracking for Program Comprehension

Felix Grund and Nick Bradley
Department of Computer Science
University of British Colombia
Vancouver, BC, Canada
ataraxie,nbrad11@cs.ubc.ca

## ABSTRACT

Program comprehension impacts code quality, velocity and efficiency. However, it is a difficult task when only using code; many of the design and implementation decisions are lost when transferring the information into code. Unfortunately, capturing this information can be challenging, especially when it requires significant changes in developer workflow. Further, if the information is captured it must be linked to the section of code that it impacts and made available to reviewers in a way that enhances comprenshion without being obtrusive.

In this paper we describe a simple method for caputring developer reasoning during coding tasks. Our implementation, called CodeStory, integrates transparently into a developer's workflow by hooking into the standard copy-paste operation. In this iteration, when developers copy text or code snippets from StackOverflow, CodeStory will scrape the page capturing additional context and including it with the pasted content in their code. This information will later be seen by other developers giving them a better understanding of the author's reasoning.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

## General Terms

Theory

## Keywords

ACM proceedings, LaTeX, text tagging

## 1. INTRODUCTION

A program is the result of decisions made based on information obtained from a variety of sources. Knowledge of these decisions can quickly evaporate if there is no process for capturing it. In practice, documenting knowlegde is often considered a resource-intensive process without tangible, short-term gains, so often it is skipped or performed inadaquately[1, 2].

This information is used to implement code but it can quickly evaporate if there is no process for capturing it.

However, this information can be useful in program comprehension tasks

approach **introduce the information as a code story**

The contributions of this paper are as follows:

- A simple approach for capturing text-based collaborative reasoning among developers. This can include communications using email, instant messaging, websites and references.
- A prototype tool that implements this approach. It captures contextual information surrounding snippets copied from StackOverflow and stores it in a database using a Google Chrome extension. An Atom[1] package lets developers paste the snippet with a hyperlink to the code story.

Section 2 presents related work, ...

## 2. RELATED WORK

related-work.tex

## 3. APPROACH

approach.tex

## 4. EVALUATION

evaluation.tex

## 5. IMPLEMENTATION

implementation.tex

## 6. DISCUSSION

discussion.tex

## 7. FUTURE WORK

future-work.tex

[1] https://atom.io

## 8. CONCLUSION

conclusion.tex

## 9. REFERENCES


[1] O. Zimmermann, U. Zdun, T. Gschwind, and F. leymann, "Combining pattern languages and reusable architectural decision models into a comprehensive and comprehensible design method," in *Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*, WICSA '08, (Washington, DC, USA), pp. 157–166, IEEE Computer Society, 2008.

[2] N. B. Harrison, P. Avgeriou, and U. Zdun, "Using patterns to capture architectural decisions," *IEEE Softw.*, vol. 24, pp. 38–45, July 2007.