Presented By:

Syed Ishtiaque Ahmad

Exploranative Code Quality Documents

Haris Mumtaz, Shahid Latif, Fabian Beck, and Daniel Weiskopf

InfoVis'19

Good code quality is needed for efficiently developing maintainable and extendable software

Goal:

Self-explanatory system

(Explanation + Exploration)

Specially for less experienced software developers | less technical stakeholders



Overview Visualization

The sum of all method complexity values for a class The maximum of all the method-level complexity values of a class The number of other classes that depend on a class (incoming dependencie Used to detect 4 types of code smell Large Class Lazy Class To show bug-proneness Spaghetti Code

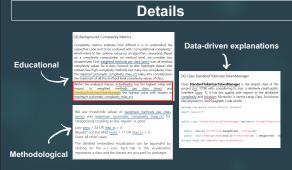
Data and Analysis

Use these software metrics with threshold to measure quality attributes

ode Quality Documents 3. Details 1. Summary Text

Summary Text analysis of software metrics along four quality attributes (1) shows generally mixed quality. mplexity: The code complexity is okay ★ as 51 classes 26.2% are rated as having low quality, still few ultiple smells. For instance Embedded visualization shows actual value of the software metrics in the respective category the classes 45.7% have been associated wit lity, but many as good (135) (1), I+I

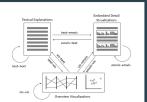
2855), QueryParser (loc: 2830), and StandardTokenizerTokenManager (loc: 3709).



DEMO

https://vis-tools.paluno.uni-due.de/cqd/

Interaction model



ransient selection on hovering over a class name anywhere highlights:

- > text-vis : polyline in parallel coordinates, dot in scatterplot
- > text-emvis: bar in embedded visualization > text-text: other occurrence of class name

Persistent selection on clicked: encoded by plack color (good for comparing classes)

Persistent range selection on the axes of parallel

Design Process and Evaluation

Formative Evaluation Iteration # 1

- 4 participants (3 PhD, 1 postdoc)
- · Mix of visualization and software experts
- Study included 3 phases (45 minutes)
- Identify different aspect of code quality in a document
- Participant reviewed features of the system and provided feedback
- Interview the participants asking general questions

Design Process and Evaluation

Formative Evaluation Iteration # 2

- 3 previous participants (2 PhD + 1 postdoc) + 1 new participant (PhD)
- · New participant is currently conducting visualization research and has a software engineering background
- Study included 2 phases (30 minutes)
- Participants reviewed features of the system and provided feedback
- Interview the participants focusing on specific improvements

Results

Iteration #1

- Added methodological and educational All the participants agreed that system explanation improved overall
- Added interaction between all representations (only text-vis interaction was present in prototype)
- Iteration #2

- More information about the bug history

Recommendations for Interactive Documents

You just learned on the sides! Consider brushing text, really!

Captions! And make them dynamic

What-Why-How Analysis Java project source code (Xerces 1.2, Lucene 2.0, Forrest); Multivariate data; What: Data 11 metrics (4 quality attributes, 4 code smell, number of bugs) Why: Tasks Self explanatory system to teach and report about software code quality How: Encode Parallel coordinates; Scatterplots; bar charts; Consistent colouring; Glyphs; How: Facet Multiple view panel coordinated with link highlighting and colouring Hover and click interaction to link texts, visuals and embedded visuals in a bidirectional way; Brushing interaction with mouse press and hold for paralle Filter class by brushing parallel coordinates axes Java project source code consisting about 200 ~ 300 classes

Strengths and Weaknesses

Does not provide solution to occlusion problems with scatterplot Provides more context to the data and explain the findings in detail Follows an incremental design Unable to view and compare all Provides recommendations for interactive documents with multivariate data Using same person for second iteration of design evaluation introduces biasness

References

- Talk: https://vimeo.com/370669433
- 2. Tool: https://vis-tools.paluno.uni-due.de/cqd/
- Paper: https://www.computer.org/csdl/journal/tg/5555/01/08807349/1cG6mtDwLNr

Thank you! Questions?

17