INFORMATION VISUALIZATION IN SOFTWARE TESTING AND MAINTENANCE A LITERATURE SURVEY

Peer Project Review 2

Marjane Namavar

University of British Columbia

Information Visualization

Fall 2019

What does Visualization in Software Testing and Maintenance mean?

- ✓ **Test techniques:** Executing a program or application with the intent of finding software bugs
- ✓ **Maintenance:** Addresses bug fixes and minor enhancements
- **Software Visualization:** Mapping from software artifacts—including programs—to graphical representations. It's needed because Software itself and software artifacts such as bugs and fixes are invisible.
- **Software Testing and Maintenance Visualization:** Use the above definition for corresponding artifact.

How Visualization helps Software Testing and Maintenance?

- Artifacts are textual, use textual visualization
- Specific ways of graphical visualization work **better**
- **Facilitates** testing and maintenance tasks
- Different techniques
- Example: Fault localization

Goals

- Survey the **existing literature** focusing on the use of visualization for software testing and maintenance
- Analyze the data from empirical experiments under **What/Why/How framework**
- Abstract gathered information to **categorize/compare** existing approaches.

Contributions

- Literature **review**
- Organizing past works under a **certain framework**
- Analysis and synthesis of the **findings** of past researches
- New **categorization/comparison**
- Suggesting some **possible future** directions

Main Steps

- Gather (23-25) relevant papers
- Review some relevant survey papers to gain an idea about doing survey project in this area
- Review all papers one time to achieve a big picture
- Analysis of all selected papers under what/why/how framework
- Prepare final paper and presentation

What has been done?

- Collect and read **relevant papers**

  - All of them from **VISSOFT** conference under testing, maintenance, debugging and evolution categories

- Review some relevant **survey** papers

  - **Limited** survey papers on reverse engineering and repository mining

What has been done? (cont.)

- Analysis of all selected papers under **what/why/how framework**

  - Variety of techniques: feature extraction, clustering, matrix views, multiple view coordination

  - Variety of data types: Networks, Trees, Text, Sequences and Events

Categorization (to be completed)

Context: Bug, test case, source code, etc.

Scalability: If the system supports millions of LOC and/or thousands of classes and/or source files

Tasks supported: Detecting code smells, trace analysis, debug support

Techniques used: “How” part in What/Why/How framework

What to do next?

- Improve categorization
- Prepare final presentation
- Prepare final report