
# Knowledge Based Reverse Engineering of Legacy Telecommunications Software

Mitel Corporation
University of Ottawa

Progress Presentation
CASCON November 1998

# Outline:

What have we done right

What could we have done better

Key progress over the last 2.5 years

Plans for the next 3 years

# What have we done right ...

Focused on the problem and adapted our research plans as new information and ideas arose

- Original plan was more knowledge intensive
- We discovered that other approaches would get us further initially
  - Focus on studying software engineers
  - Focus on search
  - Focus on usability

# ... What have we done right ...

Used a scientific approach to the problem

- Studying software engineers
  - Synchronized Shadowing
  - Analysis of results as Use Case Maps
  - Videotaping them use tools
  - Analyzing usage logs
  - Brainstorming

- (planned) Measuring usage and change in behavior to evaluate our work

# ... What have we done right

Industry/University collaboration has worked well

- Have met industrial needs, but at the same time pursuing basic research
- Our systems are actively and enthusiastically used at Mitel

NRC/University/Industry collaboration has been excellent

We would not have made so much progress without CSER interproject synergy

- Twice-yearly meetings have stimulated many ideas
- Collaboration with Ric Holt's group

# What could we have done better? (we plan to do better!)...

Need to better instill vision in graduate students and other university researchers

- Vision of working with users
  - They are both our source of
    --> initial data and ideas
    --> validation
- Vision of importance of usability (so incremental improvements are not buried in noise)
- Vision of getting ideas out to users early
  - To get maximum benefit
- Industrial vision for software analysis and visualization

# ... What could we have done better

Need more ‘project management’ of ongoing technology transfer

- Software is used, but ramp up has been slower than necessary (hence benefits less)
- University group has done too much ‘development’, ‘bug-fixing’ etc.

Need to expand collaboration within CSER

- Have tools used in other companies
- But how to do this without taking a lot of extra time!

# Key progress over the last 2.5 years...

## Source exploration infrastructure

- tkSee tool as user interface framework for testing new ideas
- Parsing, TA++ and CDB database framework for investigation of static analysis etc.
- Basing all work on an infrastructure will be a long-term time saver

## Knowledge base of SX-2000

- Rapid KB development techniques
- KB itself will be of use in near future

# ... Key progress over the last 2.5 years

Techniques for studying software engineers

- Synchronized shadowing etc.

Useful data about maintenance work patterns

- New representational formats (UCMs)

Planned participation in NRC workshop on empirical methods

Interesting results regarding parsing in the presence of conditional compilation

Clustering based on a variety of criteria including file names

# Benefits to software engineering education

Graduate software engineering course in January 1999 based on CSER research experience

- Empirical studies, usability, CASE tools
- Large registration expected

Education relevance surveys well received

- Not part of original plan, but developed from CSER's education mandate and with participation of CSER companies
- Information has helped shape SE programs

# What we would like to do in the next three years...

Goal: To continue to help improve the productivity of software engineers

Guiding principles:

- All work must be done with user interaction and involvement
- All work must be manifested in usable and testable extensions to the tkSee/ OPQ infrastructure
- All work must be evaluated for usability and incremental improvements must be measured

# ...What we would like to do in the next three years...

## Research area 1: Code analysis

- Key question: What forms of code analysis will help software engineers be more productive

- High-level hypotheses:
  - Analysis of interactions among states, processes etc. will be important
  - Dynamic analysis will be important

# ...What we would like to do in the next three years...

Research are 2: Knowledge based analysis

- Key question: How can knowledge be effectively integrated to help SEs be more productive:

- High-level hypotheses:
  - Search can be improved with the integration of small amounts of knowledge
  - One can use machine learning to study how SEs have solved software problems in the past and use the information to help them solve new problems

# ...What we would like to do in the next three years

## Research area 3: Visualization

- Key question: How can information be presented to software engineers to better help them understand software
- High level hypothesis: A variety of graphical techniques can help

## Research area 4: Empirical studies

- Key question: How can be better learn about and model the work of software engineers (or others) so that we can develop requirements
- High level hypothesis: Efficient techniques for gathering and coding observations, and developing work patterns can help.