SOFTWARE MAINTENANCE AND EVOLUTION --- DECADE OF REFACTORIZING ---

CS563
WEEK 1 - THU
Danny Dig
Objectives YOU created

Be a Better Engineer

- Knowledgeable experience with state of the art in refactoring and evolution
- Applications in Industry
- Understand about IoT
- Build maintainable SW
- Intro to Research in SE
Course Objectives

Software Maintenance and Evolution research & practice
- state of the art under the theme of Mobile and IoT

Develop critical thinking abilities

Practice giving scientific presentations and teaching others

Engage in active learning activities in class, such as discussions

Practice a research or novel-industrial project through all its stages

Have fun learning
Course Administration

Research-based course

Complete a research or industrial-novel project of your choice (teams of 2-3 students)

Read papers, write Critiques

Class Discussion

Put Your Dream to the Test
What is Refactoring?

“A change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behaviour” – M. Fowler [1999]

Top-level menu in all modern IDEs
- In 2000, I created the first open-source refactoring tool
Refactoring research growth

2,880 refactoring papers (4,944 authors) since 1990
The Humble Beginnings

First refactoring paper:
- Bill Opdyke and Ralph Johnson [SOPPA’90]: Refactoring, an Aid in designing application frameworks and evolving OO systems

PhD dissertations:
- Bill Griswold ‘91 at U of Washington
- Bill Opdyke ‘92 at U of Illinois
- Don Roberts ‘99 at U of Illinois

Refactoring research hard to publish in early 90s
- conflated with the compiler community
Most recent Decade of Refactoring Research

2,880 refactoring papers since 1990

2,442 papers between 2005-2016
Corpus of Papers

Work done by Marouane Kessentini and his team at Michigan

Scopus and Web of Science
- "Refactoring” in title, abstract, and keywords
- yielded 3277 papers

Refactoring definition:
- transformation with behavior preservation

Manual validation of ALL papers:
- each paper analyzed title, abstract (and sometimes content)
- 4 grad students who took a graduate class on Softw QA,
- Kessentini (faculty) looked at the contentious papers

In the end we removed 397 papers
O1: To Grow, Welcome Outsiders, Champions from Other Communities

Number of citations and publications, 2005-2016

800 citations
37 pubs
Expand focus to meet new needs that you can serve
O3: To Grow, Expand the Target Artefacts

Expand target: new refactoring research is about change to the code, models, architecture, DB, UI
O4: To Grow, Expand Objectives (the WHY)

Expand Objectives: new refactoring research is to improve performance, security, migration (beyond internal quality)
O5: To Grow, Adopt new Techniques (the HOW)

New refactoring research via static analysis, search-based, data mining
O6: To Increase Practical Impact, Work with Industry

Industrial collaboration levels:
- surveys with practitioners
- tool validated on industrial codebase
- tool licensed to industry, adopted in products
The Big Growth from 2012-2013
Big Growth of the Field: Expanding Definition

“A change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behaviour” – M. Fowler ‘99

Expanded Focus, Objectives, Techniques

“Automation/insight/testing/prioritization of changes to the artifacts of software to improve non-functional requirements and without changing its proper, intended behaviour” – D. Dig ‘17

Communities that thrive are going to be more accepting of new ideas
Big Growth Enabled by Community Engineering

Industrial champion(s): M. Fowler, Kent Beck, Ward Cunningham

Complementary skills: tool builders, paper writers, curators

Mindset for industrial collaboration and adoption

Shared platform:
- Eclipse (Erich Gamma + Frank Tip), analysis frameworks

Community infrastructure: 7 Refactoring Workshops, Dagstuhl
- first workshop in 2007, 50+ participants, 32 posters
- invited all major IDE providers
- growing new leaders