

COMP 762

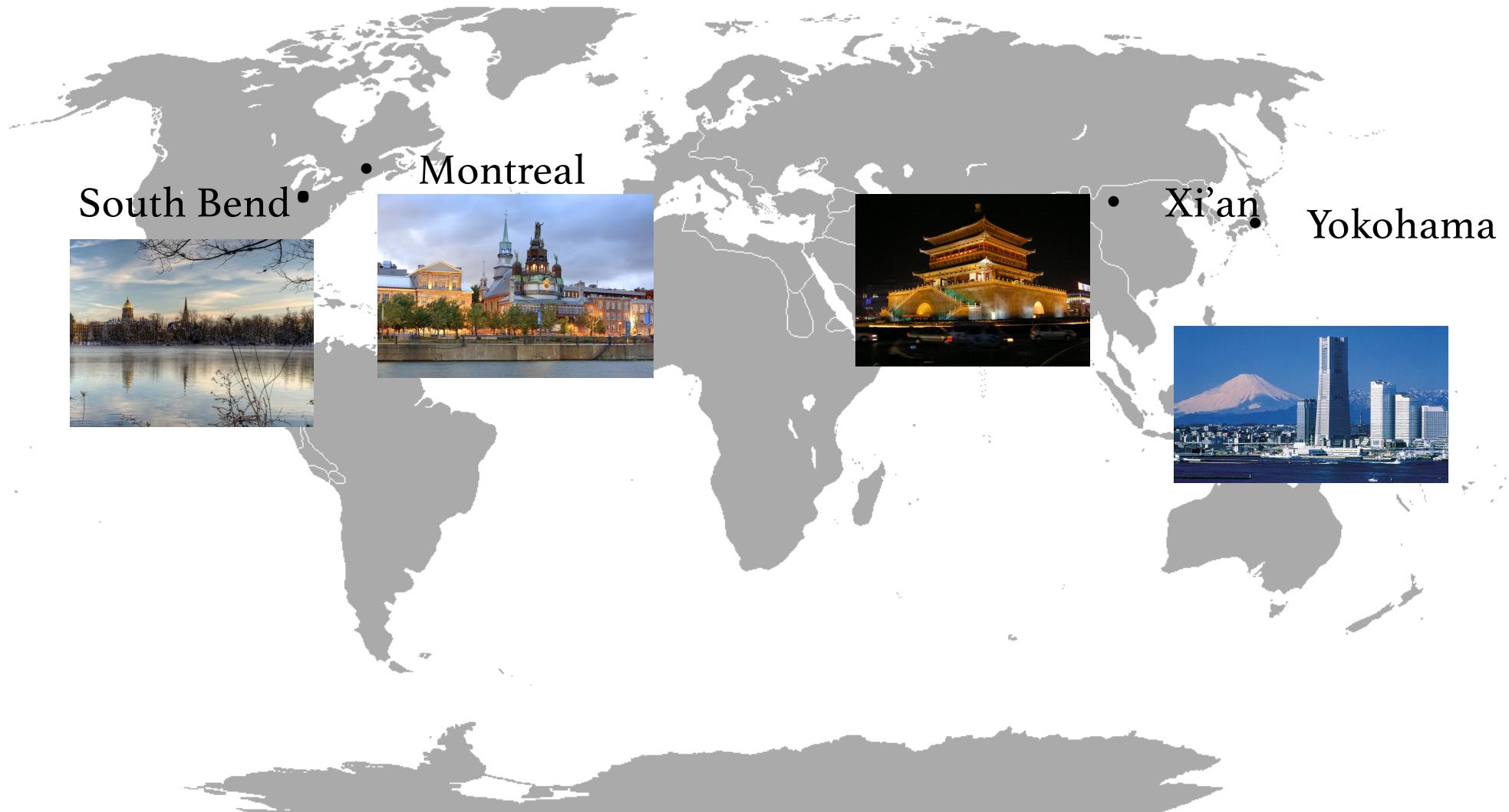
ML & NLP Methods for Software Engineering

Week1-b

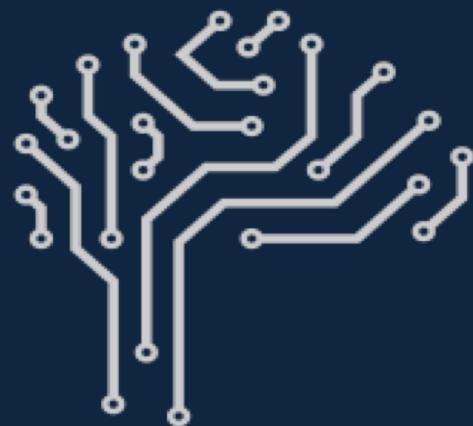
Jin L.C. Guo

Assistant Professor of Software Engineering
School of Computer Science
McGill University
jguo@cs.mcgill.ca

Self Introduction

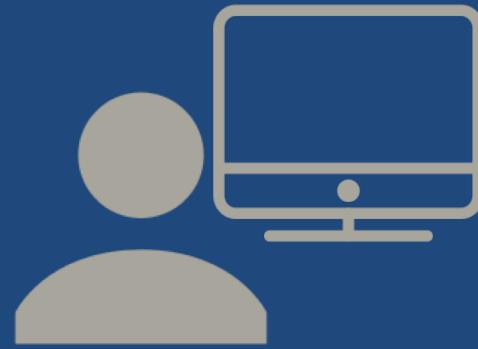


Automated Knowledge Construction



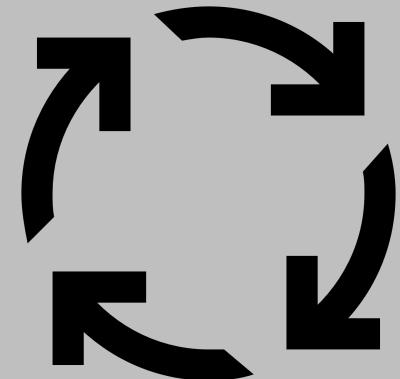
*Scalable, Trusted
Knowledge
Construction*

Software Connection Inference



*Configurable,
Cost-effective
Traceability*

Connectivity Supported SE Tools



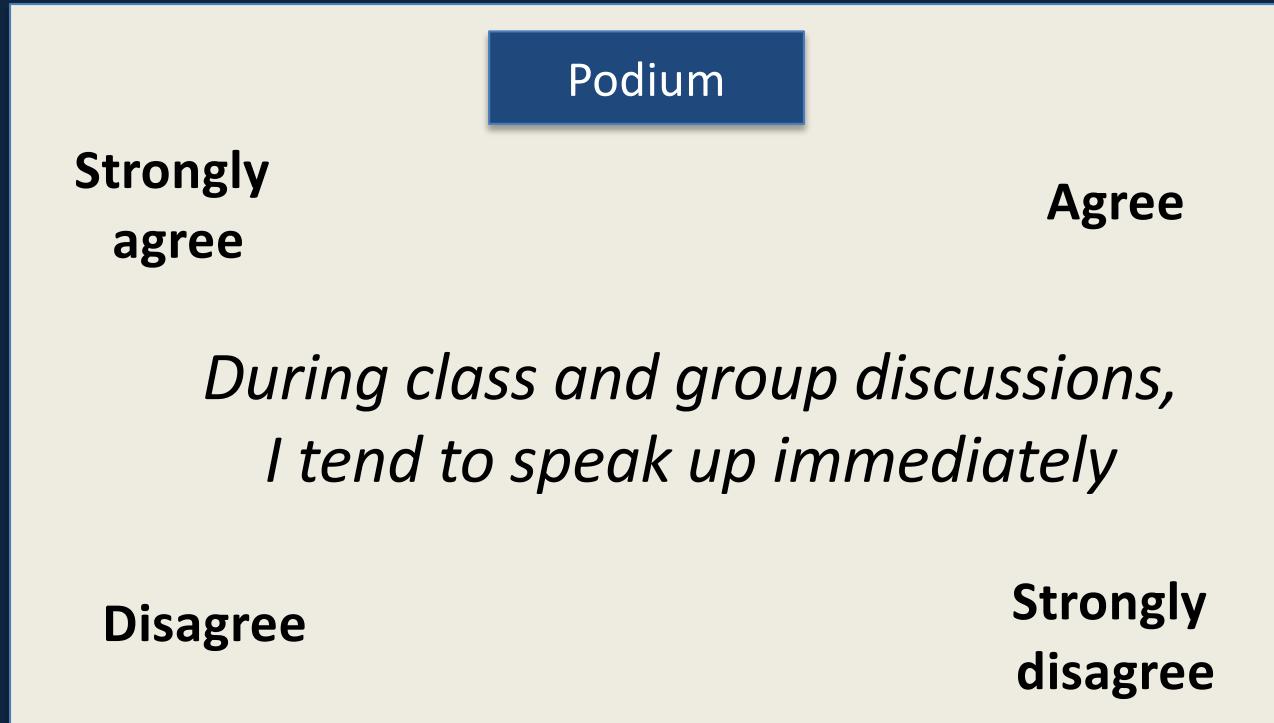
*Purposed , Valued
Connectivity*

Introduce yourself

- To the class
 - Who you are
 - Why do you enroll this class
 - One boring fact about yourself

Activity 1

- Find your corner



Activity 1

- Use 5 mins to discuss with your corner mates the reasons of your choice.
- Discuss and share the benefit and drawbacks of this tendency of your corner

Course Logistics

- Time

Monday & Wednesday 10:05am – 11:25am

- Location

McConnell Engineering Building 103

- Syllabus

http://jguo-web.com/COMP762_Fall2019/

- Online Discussion

piazza.com/mcgill.ca/fall2019/comp762/home

- Office hour

By appointment

Course Overview

- Graduate level *Seminar* Style class
 - Paper presentations and discussions
 - Course projects
 - Small size
 - Background Survey

Class Structure

- Before class
 - Read assigned paper
 - Write paper report (more later)
- During class
 - Activities about paper content
 - Present related topics
- After class
 - Think up and implement ideas for your project

Paper Presentation (from Week 3)

- One to two student per class
- About the assigned paper
- And the related papers either
 - Solving the same or similar SE problems
 - Using the same or similar techniques in a different SE context
 - Addressing a similar concern on SE research
- Post the paper titles on Piazza two days before class by 11:59pm.
- Give a 20 minutes presentation and answer questions.

Project

- Individually or group (max 4 people)
- Ideation phase:
 - Post and discuss your project ideas on piazza
 - Form teams if there are common interests
- Report project proposal phase:
 - Proposal report (less than 2 pages)
 - Presentation [week 7-8]
- Carry out and complete project:
 - Final Project report (4-8 pages)
 - Final presentation [week 13-14]

Evaluation

- Paper reading report [15%]
- Participation [25%] ✓ Survey and discussion [15%]
 ✓ Feedback [10%]
- Presentation [20%] ✓ Paper Presentation [5%]
 ✓ Project proposal presentation [5%]
 ✓ Final project presentation [10%]
- Project Report [40%] ✓ Proposal [10%]
 ✓ Peer review [10%]
 ✓ Final report [20%]
- Bonus [5%]

Schedule

- Syllabus http://jguo-web.com/COMP762_Fall2019/

Activity 2

- Think about three tasks/activities related to software development, maintenance and evolution that you are interested in.
- Identify the actor, action, and product of these activities.
- What are the tools the actor uses to produce the products?

Software Engineering is ...

- Basically, *develop* and *evaluate* Software

Develop: synthetic activities that creating and modifying the software, including the code, design documents, documentation, etc.

Evaluate: analytic activities associated with predicting, determining, and estimating properties of the software systems, including both functionality and non-functional properties such as performance or reliability.

Software Engineering Research is ...

- Seek better ways to develop and evaluate Software
- Communicate the findings

Paper Abstract Anatomy

- Current state of the art, identifying a particular problem **Problem**
- What this paper contributes to improving the situation **Contribution**
- The specific result of the paper and the main idea behind it **Findings**
- How the result is demonstrated or defended **Validation**

COMP 762

ML & NLP Methods

for Software Engineering

Week2-a

Jin L.C. Guo

Assistant Professor of Software Engineering
School of Computer Science
McGill University
jguo@cs.mcgill.ca

From our first survey

- What do you want to gain from this course?

ML NLP + SE

Coding Discussion Presentation Paper reading, writing

Paper submissions time management skills

Meet and work with people with related interests

Your contribution

Industry experience on practices

Novel ML or NLP techniques (Vision, Deep RL, Graph NN, etc)

Experience on designing a research strategy

Considerations for Socio-technical Systems (e.g. biases)

Communicate your opinions and ideas in varied ways

The diagram consists of two large speech bubble-like shapes. The left shape is light blue and contains the words 'Effort' and 'Activity'. The right shape is dark blue and contains the words 'Result' and 'Achievement'. Below these shapes, on the dark blue background, are the words 'Report?' and 'Objective?'. A horizontal line connects the bottom of the light blue shape to the bottom of the dark blue shape. A small rectangular callout at the bottom center contains the text 'Aiming for the whole set'.

Effort

Result

Activity

Achievement

Report?

Objective?

Aiming for the whole set

Paper Abstract Anatomy

- Current state of the art, identifying a particular problem **Problem**
- What this paper contributes to improving the situation **Contribution**
- The specific result of the paper and the main idea behind it **Findings**
- How the result is demonstrated or defended **Validation**

Paper Abstract Anatomy

Abstract: As software systems become more complex the overall system structure - or software architecture - becomes a central design problem. An important step towards an engineering discipline of software is a formal basis for describing and analyzing these designs. We present a theory for one aspect of architectural description, the interactions between components. The key idea is to define architectural connectors as explicit semantic entities. These are specified as a collection of protocols that characterize each of the participant roles in an interaction and how these roles interact. We illustrate how this scheme can be used to define a variety of common architectural connectors. We provide a formal semantics and show how this lends to a sound deductive system in which architectural compatibility can be checked in a way analogous to type checking in programming languages.

“Formalizing Architectural Connection.” Proceedings of 16th International Conference on Software Engineering, 1994

Paper Abstract Anatomy

Abstract: What is it that makes software fail? In an empirical study of the post-release defect history of five Microsoft software systems, we found that failure-prone software entities are statistically correlated with code complexity measures. However, there is no single set of complexity metrics that could act as a universally best defect predictor. Using principal component analysis on the code metrics, we built regression models that accurately predict the likelihood of post-release defects for new entities. The approach can easily be generalized to arbitrary projects; in particular, predictors obtained from one project can also be significant for new, similar projects.

“Mining Metrics to Predict Component Failures”, Proceedings of 16th International Conference on Software Engineering, 2006

Categorize SE papers

- Problem dimension:

- Development method

identification of features useful for improving the design, implementation, maintenance and automation during the different phases of the software development.

- Analysis method

approaches for software evaluation, such as testing or verification and validation, as well as quality analysis including performance metrics

- Specific instance
 - Generalization or characterization
 - Feasibility study or exploration

Categorize SE papers

- Contribution dimension:

- Theoretical

Grounded theory, procedure and algorithm applied to different software development process phases, techniques for bug prediction, dynamic and static analysis, reliability analysis structure or taxonomy for a problem area, etc.

- Technological

Tools, prototypes, infrastructures, frameworks, etc.

- Empirical

Case study, survey, lesson learned reports, etc.

- Perspectival

systematic literature review, interdisciplinary study or exploratory study, historical perspective, etc.

Categorize SE papers

- Validation dimension:

- Analysis

*providing proofs or complexity analysis or run-time analysis
provided by applying the contribution to a controlled situation or collecting statistically significant results and data.*

- Evaluation

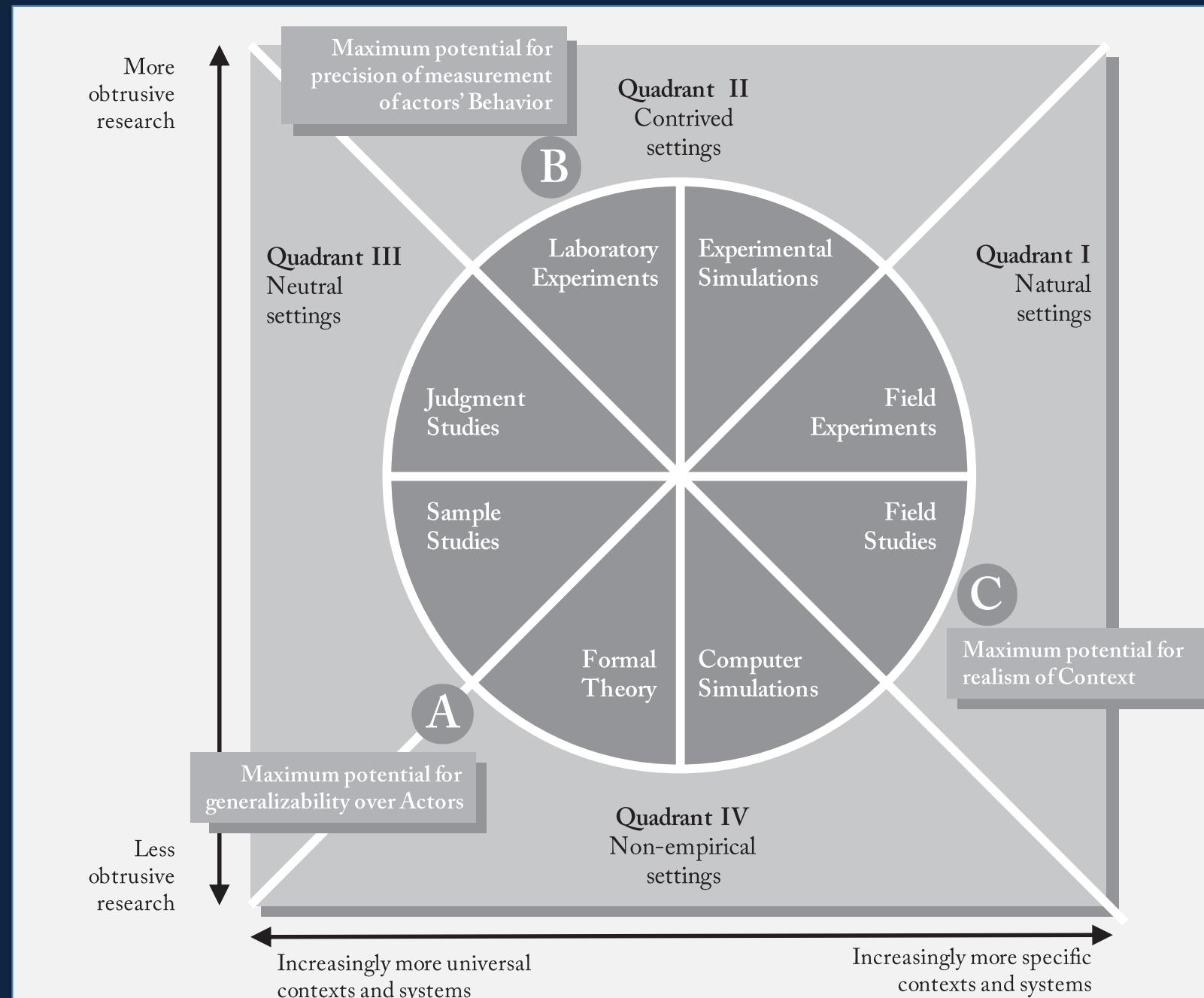
*adequately describes phenomena of interest (descriptive model)
accounts for the phenomena of interest (qualitative model)
generates results that fit actual data (empirical model)*

- Experience

evidences supporting the contribution by using third party result collection

- Example

illustrations derived from practical situation, toy examples.



Stol, Klaas-Jan, and Brian Fitzgerald. "The ABC of software engineering research." *ACM Transactions on Software Engineering and Methodology (TOSEM)* 27, no. 3 (2018): 11.

Paper Abstract

- Current state of the art, identifying a particular problem **Problem**
- What this paper contributes to improving the situation **Contribution**
- The specific result of the paper and the main idea behind it **Findings**
- How the result is demonstrated or defended **Validation**

Expand to full Paper

- **Abstract**
- **Introduction**
 - Expands the abstract, also discusses limitations of existing work, and how the proposed solution has been evaluated
- **Related Work**
 - Describes what has already taken place in this area, and how is this paper different
- **Background**
 - introduce concepts help understanding proposed approach
- **Proposed Approach**
 - Detailed explanation of the proposed solution
- **Evaluation**
 - Performance comparison with existing work
- **Thread to Validity**
- **Conclusions**
 - Summarize result, contribution, limitation and future work.
- **References**

Paper Reading Report

- Using your own word:
 - Motivation of this work
 - What are the assumptions this paper makes?
 - What is the proposed solution?
 - How is solution evaluated?
 - What elements might threaten the validity of this work?
 - Limitations or *extensions* for this work
 - Major takeaway message

Paper Reading Report

- Five reports throughout the semester (1+4)
 - “Who should fix the bug” (on Piazza)
 - Four more papers based on bidding using EasyChair
 - Submit your report on those four papers using EasyChair

Where to find good SE papers?

- ICSE - ACM/IEEE International Conference on Software Engineering. Since 1975.
- ESEC/FSE - ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering.
- ASE - IEEE/ACM Automated Software Engineering
 - foundations, techniques, and tools for automating the analysis, design, implementation, testing, and maintenance of large software systems

Where to find good SE papers?

- Specific Field

MSR: International Conference on Mining Software Repositories

ESEM: International Symposium on Empirical Software Engineering and Measurement

RE: International Requirements Engineering Conference

ICSME: International Conference on Software Maintenance and Evolution

...

Activity 3

- Go back to the software engineering activities you come up with.
- What are the problems/potential improvements for the current process?
- What is needed for the improvement?

Homework

- Background survey
- Goal through the required paper list, read abstract
 - Signup paper presentation (aiming for understand the paper)
 - Bid on papers to write report (aiming for criticize the paper) – Email about bidding later today
- Paper report for the first paper: “Who should fix this bug”