AI for Software Engineering Research

Pitfalls and Challenges

Willian Oizumi

CBSoft'23
CONGRESSO BRASILEIRO DE SOFTWARE
CAMPO GRANDE - MS

iSE'23
THE 3RD BRAZILIAN WORKSHOP ON
INTELLIGENT SOFTWARE ENGINEERING

About me

Willian Oizumi

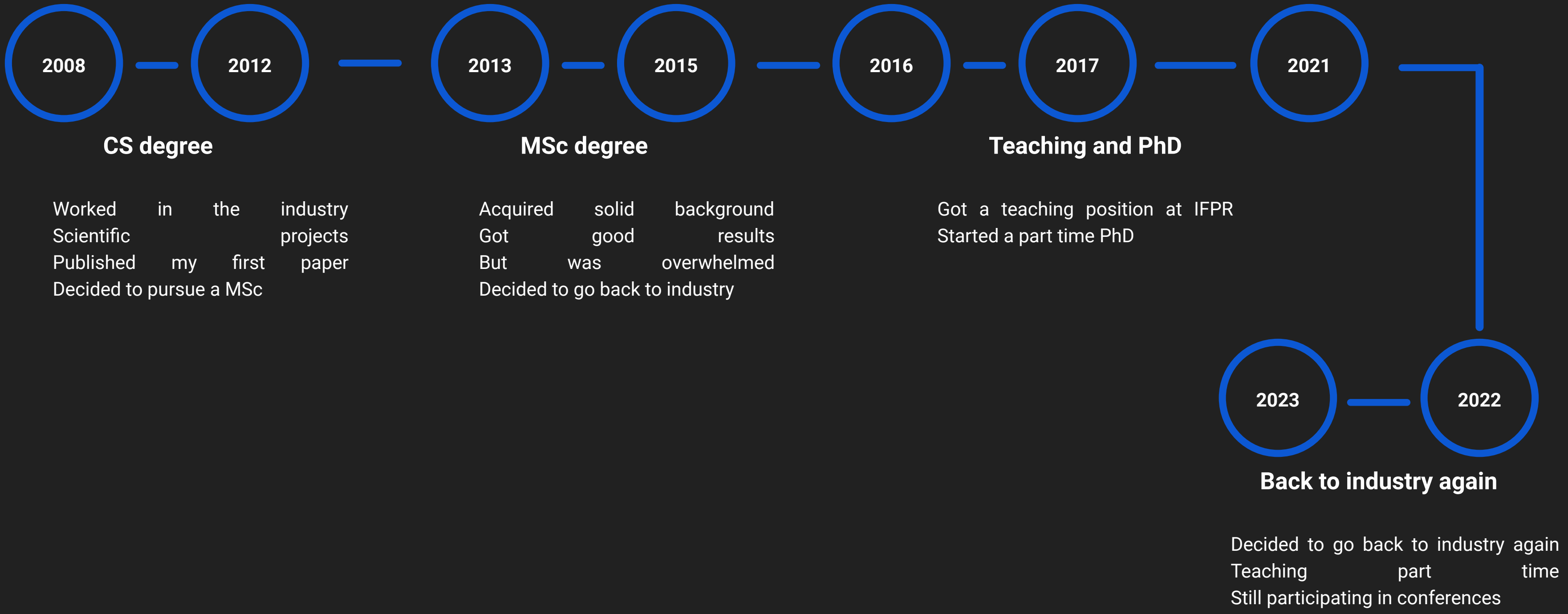
Senior Software Engineer @GoTo

Teaching Professor @Senac-PR

PhD in Informatics - PUC-Rio (2022)

http://wnoizumi.github.io/

My career so far



CS degree

Worked in the industry
Scientific projects
Published my first paper
Decided to pursue a MSc

MSc degree

Acquired solid background
Got good results
But was overwhelmed
Decided to go back to industry

Teaching and PhD

Got a teaching position at IFPR
Started a part time PhD

Back to industry again

Decided to go back to industry again
Teaching part time
Still participating in conferences

My Research Experience

- Internal Quality Metrics
- Code Smells
- Software Architecture and Design
- Refactoring
- Search-based Software Engineering
- Software Product Lines
- Code Review

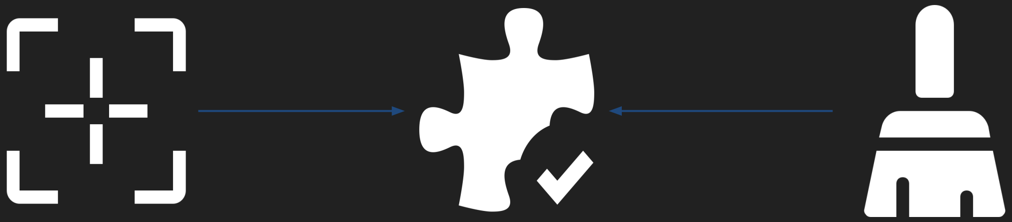
- Exploratory Study
- Controlled Experiment
- Repository Mining
- Survey
- Grounded Theory

Our Goals Today

- Provide an overview of my PhD research
- Explore the pitfalls and challenges I faced during the journey
- Discuss recommendations for conducting a strong (PhD/MSc) research

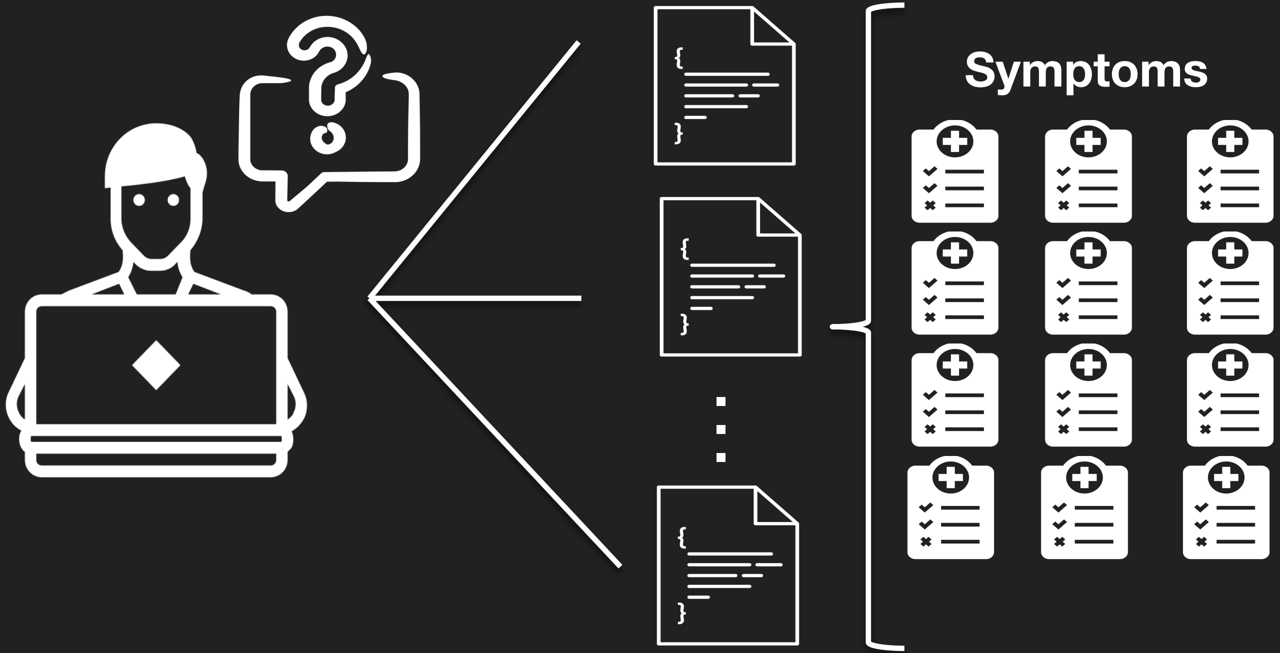
My PhD Research

Main Goal was to provide **effective support** for developers in the **identification** and **refactoring** of **design problems**



Design Problem

Where and How to Refactor?



Symptoms

After multiples studies and years…

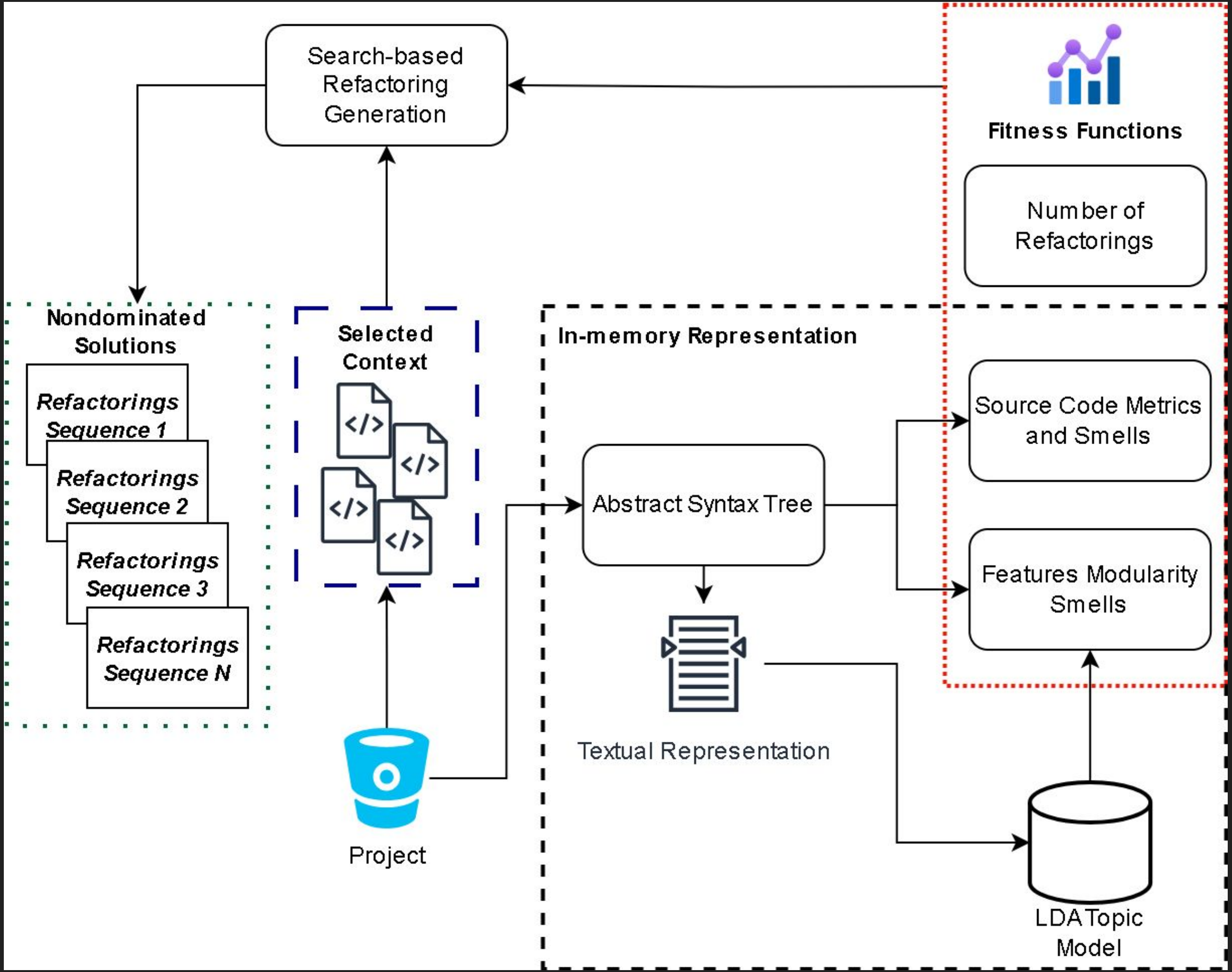
Requirements for Refactoring Recommendation Techniques

1. Consideration of Heterogeneous Information

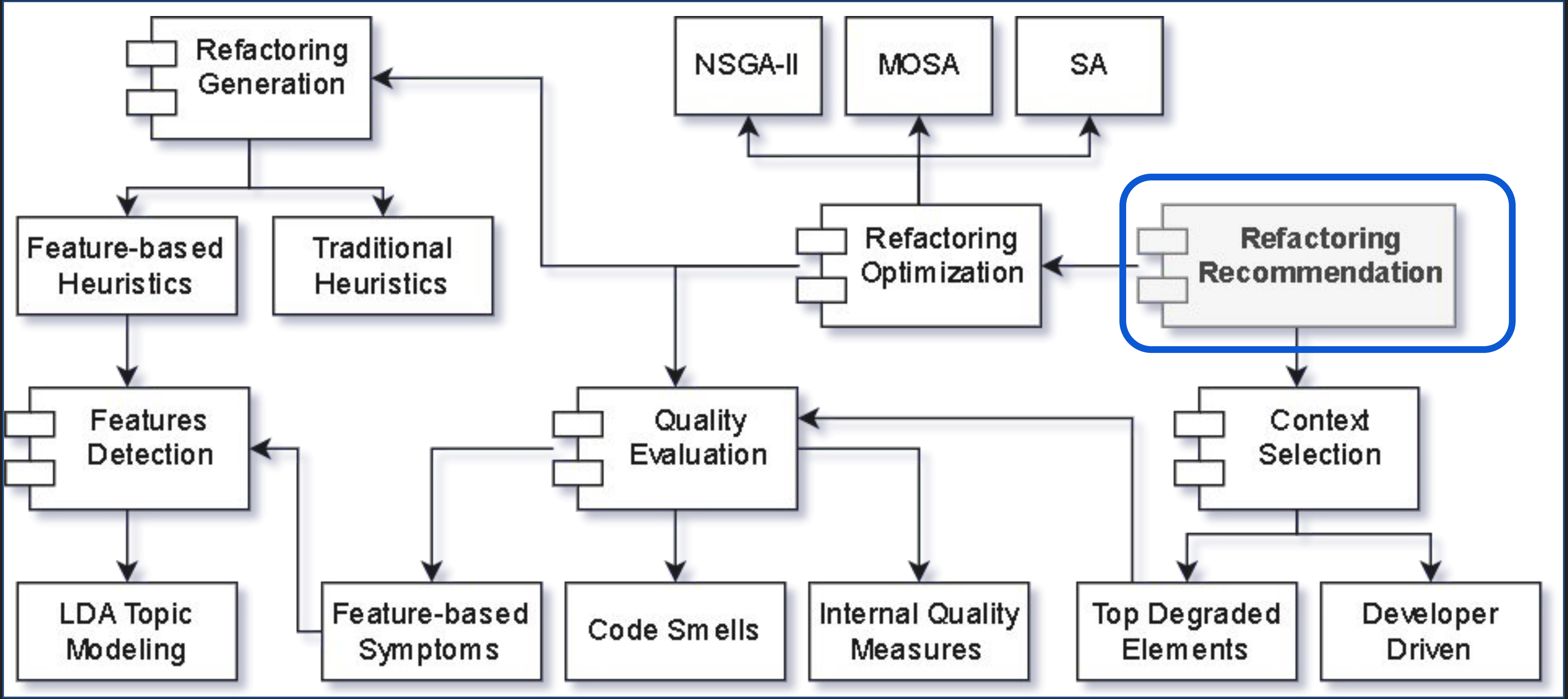
2. Context-Sensitive Detection

3. Feature Modularity Awareness

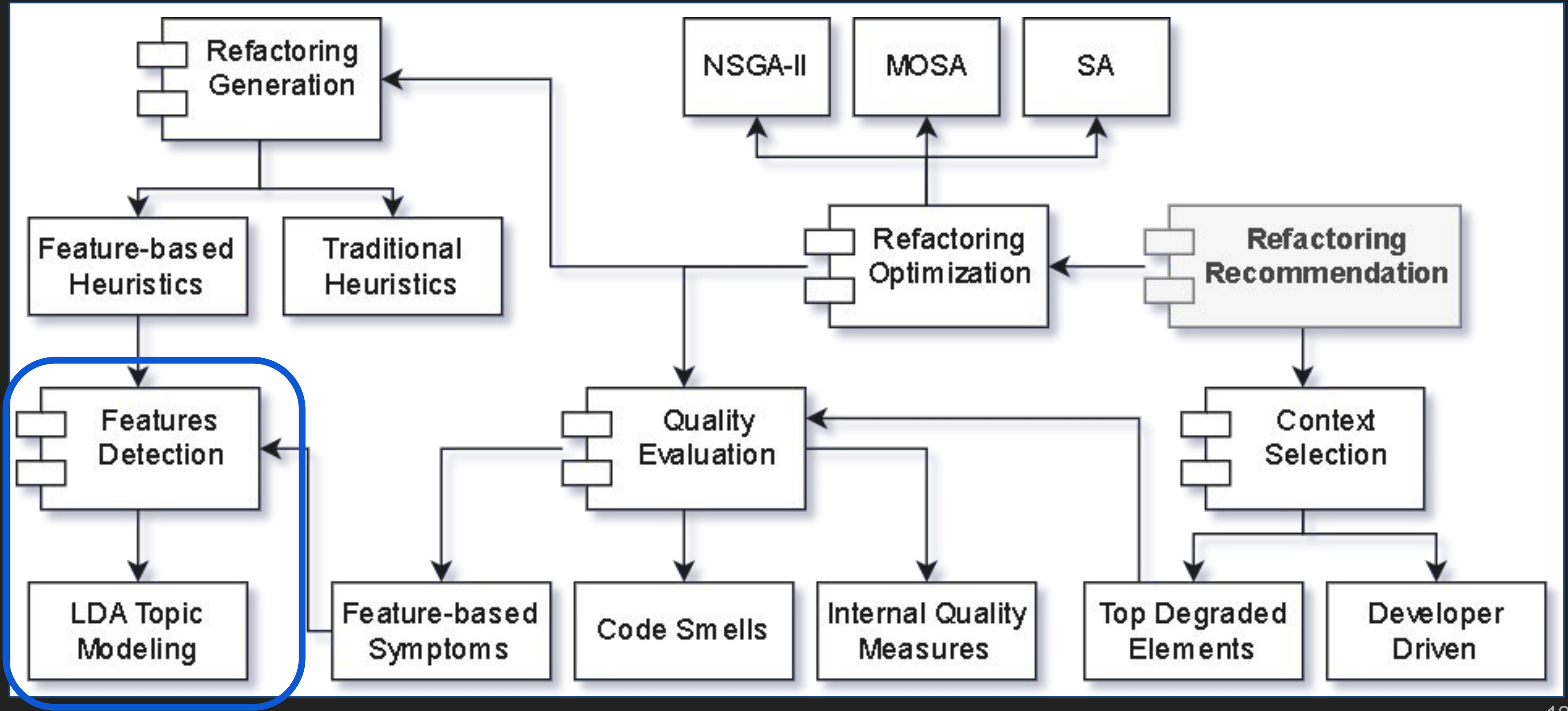
4. Effective Recommendations


Search-based Refactoring Generation
Fitness Functions
Number of Refactorings
Nondominated Solutions
Refactorings Sequence 1
Refactorings Sequence 2
Refactorings Sequence 3
Refactorings Sequence N
Selected Context
In-memory Representation
Source Code Metrics and Smells
Abstract Syntax Tree
Features Modularity Smells
Textual Representation
Project
LDA Topic Model

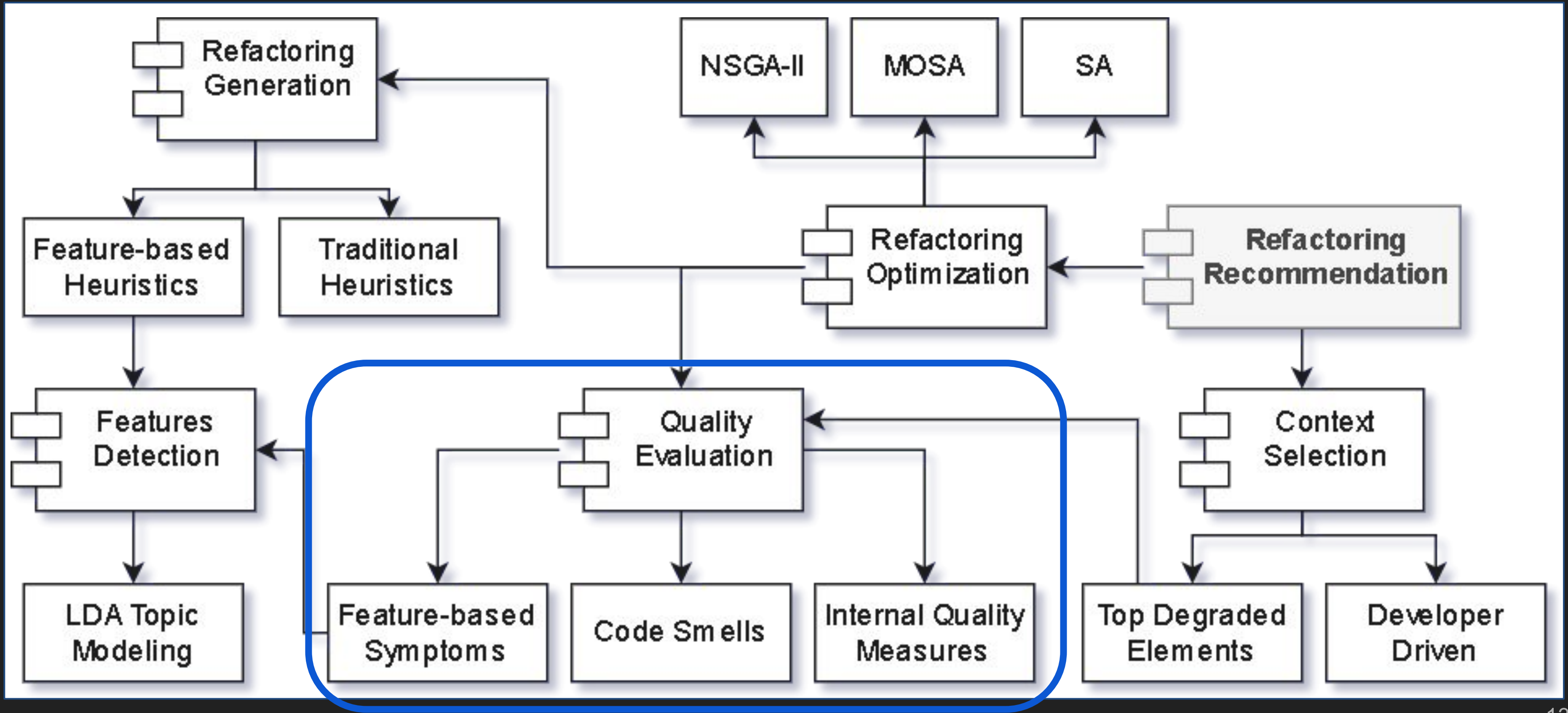
Overview of Main Components



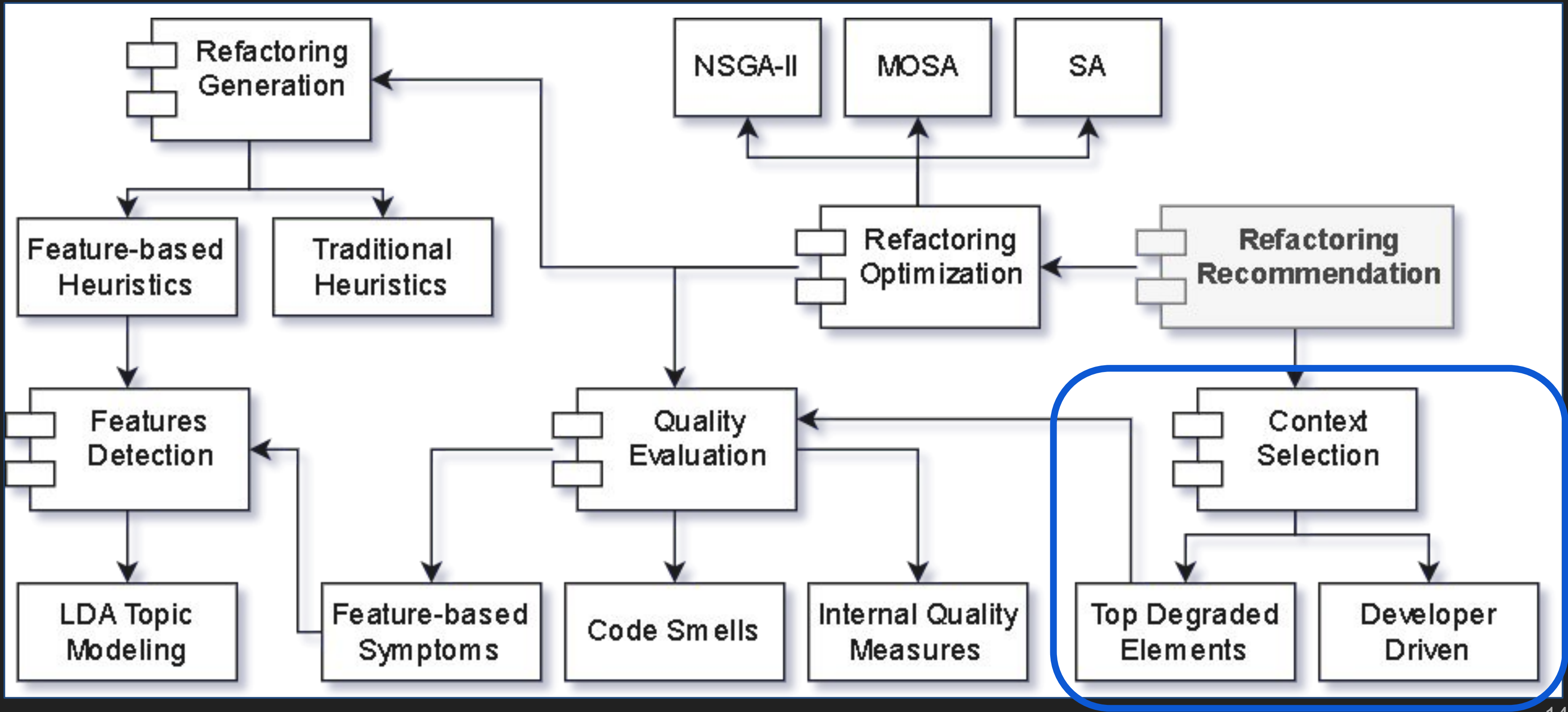
Overview of Main Components



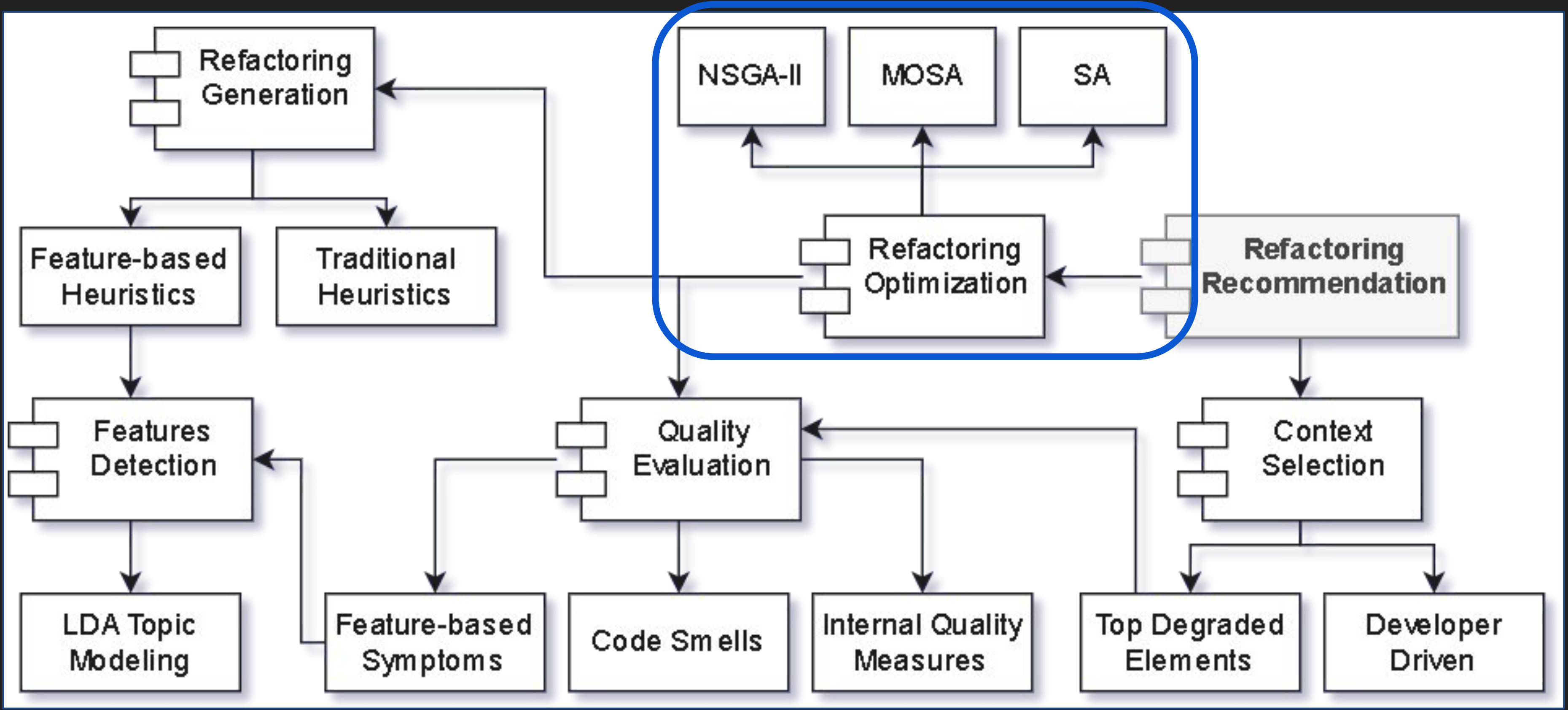
Overview of Main Components



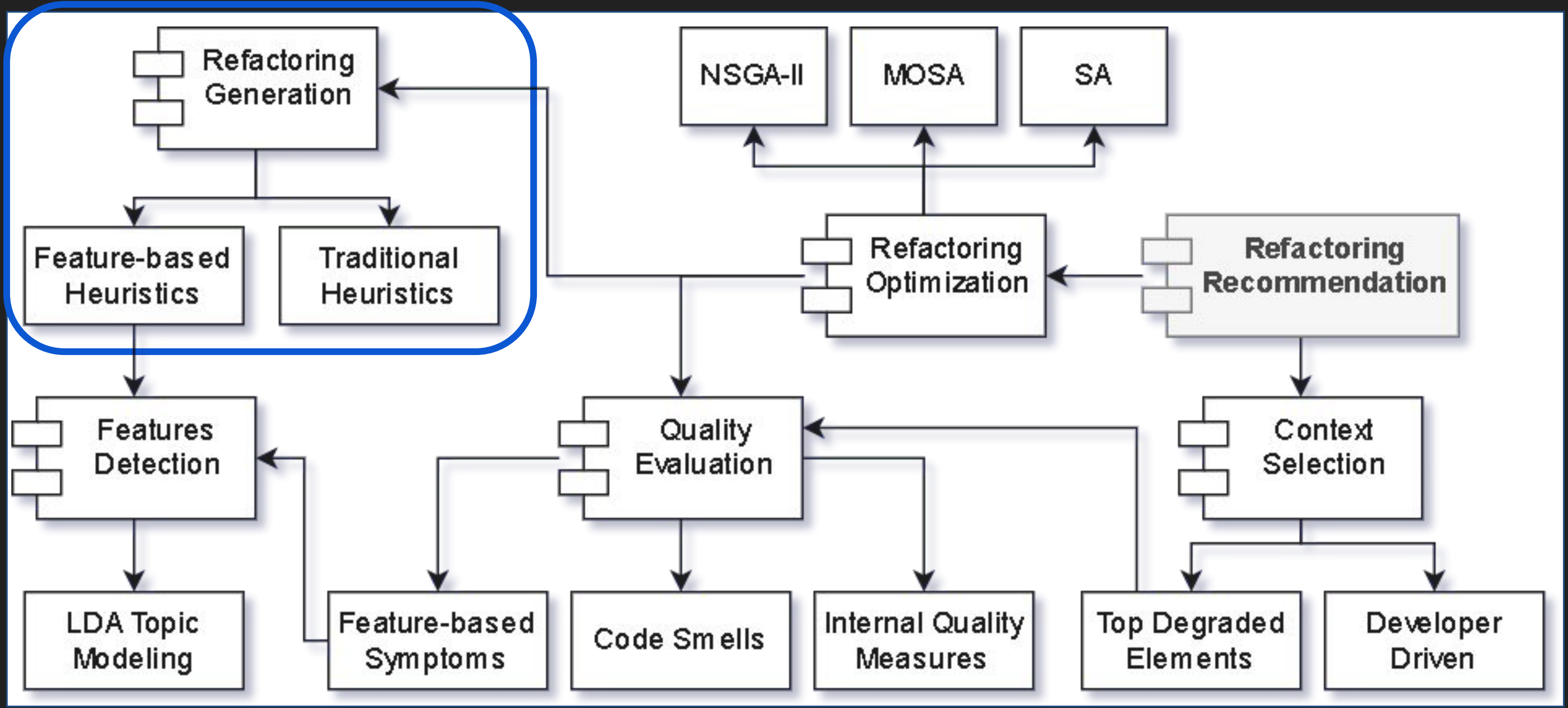
Overview of Main Components



Overview of Main Components



Overview of Main Components

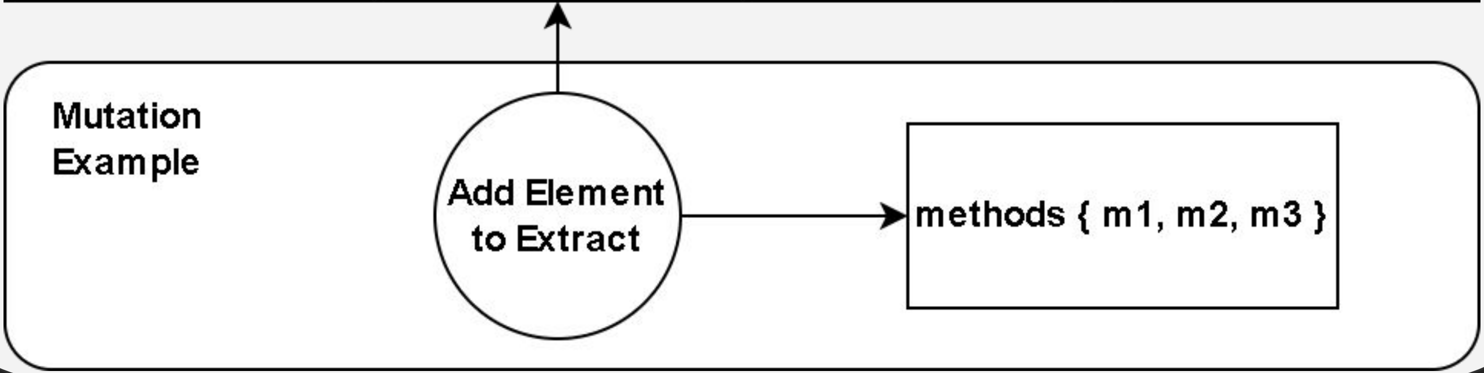


Solution Representation

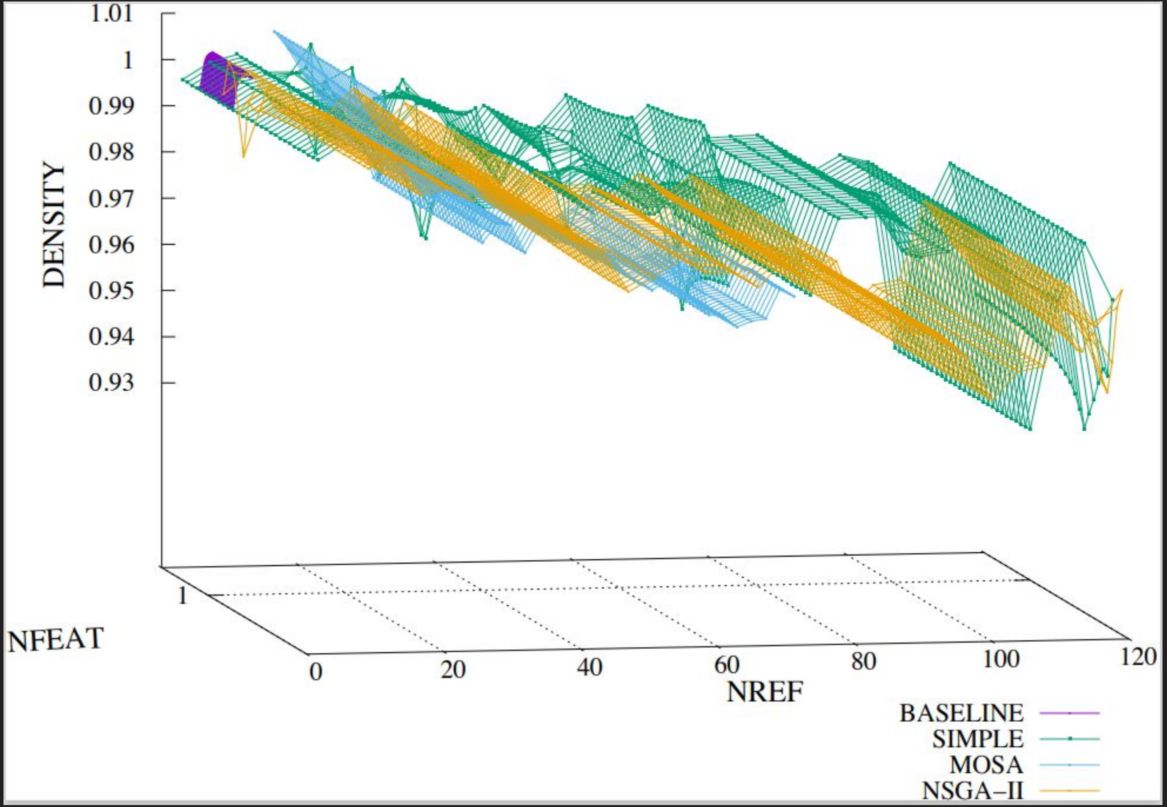
ClassA	ClassB	ClassC	ClassD

MM {m1->ClassD}
MM {m2->ClassF}
MF {f1->ClassD}
MF {f2->ClassF}
	EC [
	fields {f1, f2, f3}
	methods {m1, m2}
	]
		MM {m1->ClassE}
		EC [
		fields {f1, f2}
		methods {m2, m3}]
			No Refactoring

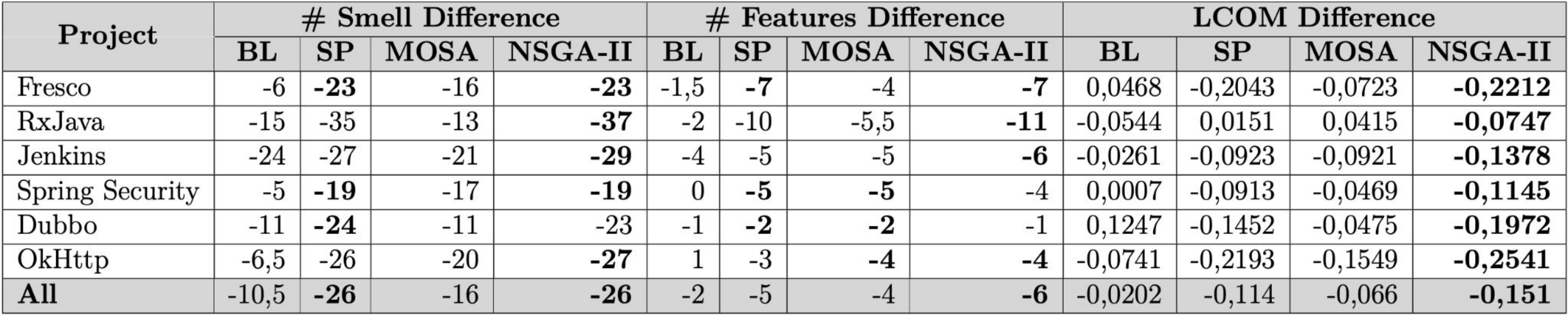
0	1	2	3



Example of Solution Space for the okhttp Project

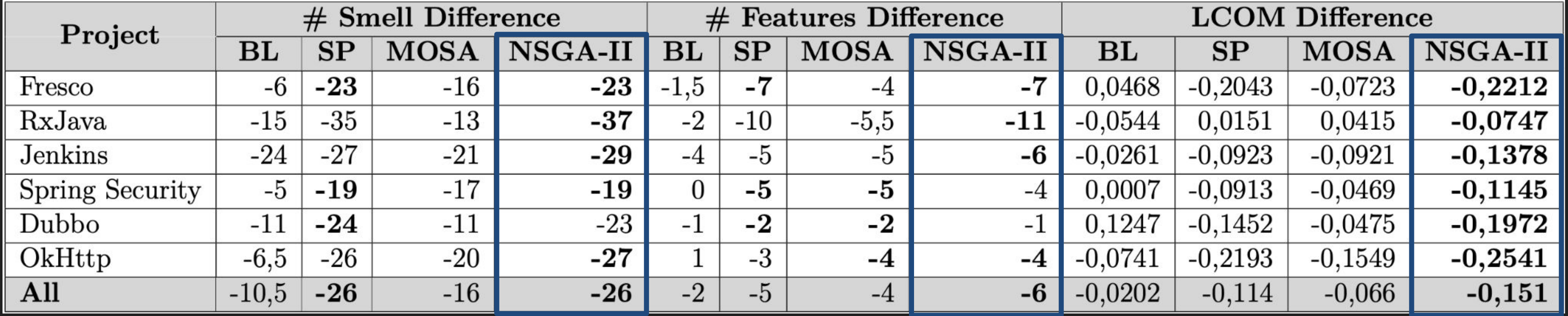


Quality Impact of Solutions



Project	# Smell Difference				# Features Difference				LCOM Difference			
	BL	SP	MOSA	NSGA-II	BL	SP	MOSA	NSGA-II	BL	SP	MOSA	NSGA-II
Fresco	-6	**-23**	-16	**-23**	-1,5	**-7**	-4	**-7**	0,0468	-0,2043	-0,0723	**-0,2212**
RxJava	-15	-35	-13	**-37**	-2	-10	-5,5	**-11**	-0,0544	0,0151	0,0415	**-0,0747**
Jenkins	-24	-27	-21	**-29**	-4	-5	-5	**-6**	-0,0261	-0,0923	-0,0921	**-0,1378**
Spring Security	-5	**-19**	-17	**-19**	0	**-5**	**-5**	-4	0,0007	-0,0913	-0,0469	**-0,1145**
Dubbo	-11	**-24**	-11	-23	-1	**-2**	**-2**	-1	0,1247	-0,1452	-0,0475	**-0,1972**
OkHttp	-6,5	-26	-20	**-27**	1	-3	**-4**	**-4**	-0,0741	-0,2193	-0,1549	**-0,2541**
All	-10,5	**-26**	-16	**-26**	-2	-5	-4	**-6**	-0,0202	-0,114	-0,066	**-0,151**

Quality Impact of Solutions



Project	# Smell Difference				# Features Difference				LCOM Difference			
	BL	SP	MOSA	NSGA-II	BL	SP	MOSA	NSGA-II	BL	SP	MOSA	NSGA-II
Fresco	-6	**-23**	-16	**-23**	-1,5	**-7**	-4	**-7**	0,0468	-0,2043	-0,0723	**-0,2212**
RxJava	-15	-35	-13	**-37**	-2	-10	-5,5	**-11**	-0,0544	0,0151	0,0415	**-0,0747**
Jenkins	-24	-27	-21	**-29**	-4	-5	-5	**-6**	-0,0261	-0,0923	-0,0921	**-0,1378**
Spring Security	-5	**-19**	-17	**-19**	0	**-5**	**-5**	-4	0,0007	-0,0913	-0,0469	**-0,1145**
Dubbo	-11	**-24**	-11	-23	-1	**-2**	**-2**	-1	0,1247	-0,1452	-0,0475	**-0,1972**
OkHttp	-6,5	-26	-20	**-27**	1	-3	**-4**	**-4**	-0,0741	-0,2193	-0,1549	**-0,2541**
All	-10,5	**-26**	-16	**-26**	-2	-5	-4	**-6**	-0,0202	-0,114	-0,066	**-0,151**

NSGA-II outperformed other strategies in most projects

Thesis Contributions

01 **Requirements and Guidelines**

- Empirically identified requirements
- Definition and evaluation of key requirements

02 **Complete Approach for Dealing with Design Problems**

- Robust evidence on the identification and removal of design problems
- Single solution addressing all key requirements

03 **Search-based Refactoring**

- Alternative and effective solution representation
- Heuristics-based initial population
- Domain-specific mutation operators

04 **Open Science**

- Open source tools (Organic and OrganicRef)
- Replication package for all studies

What went wrong?

(what could be better?)

#1
Too much indecision at the beginning

Code Smells

Technical Debt

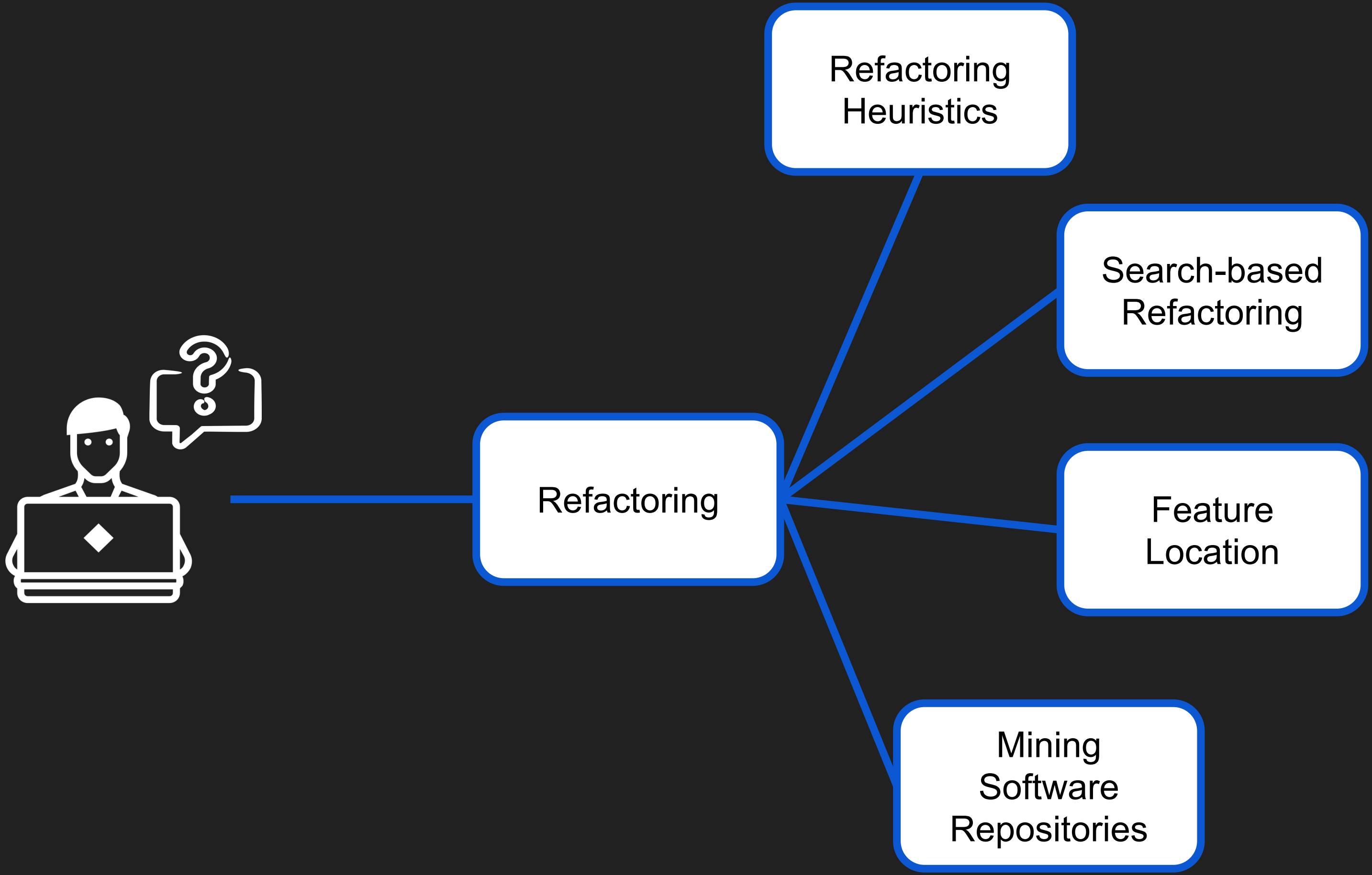
IHC

Source Code Metrics

Refactoring

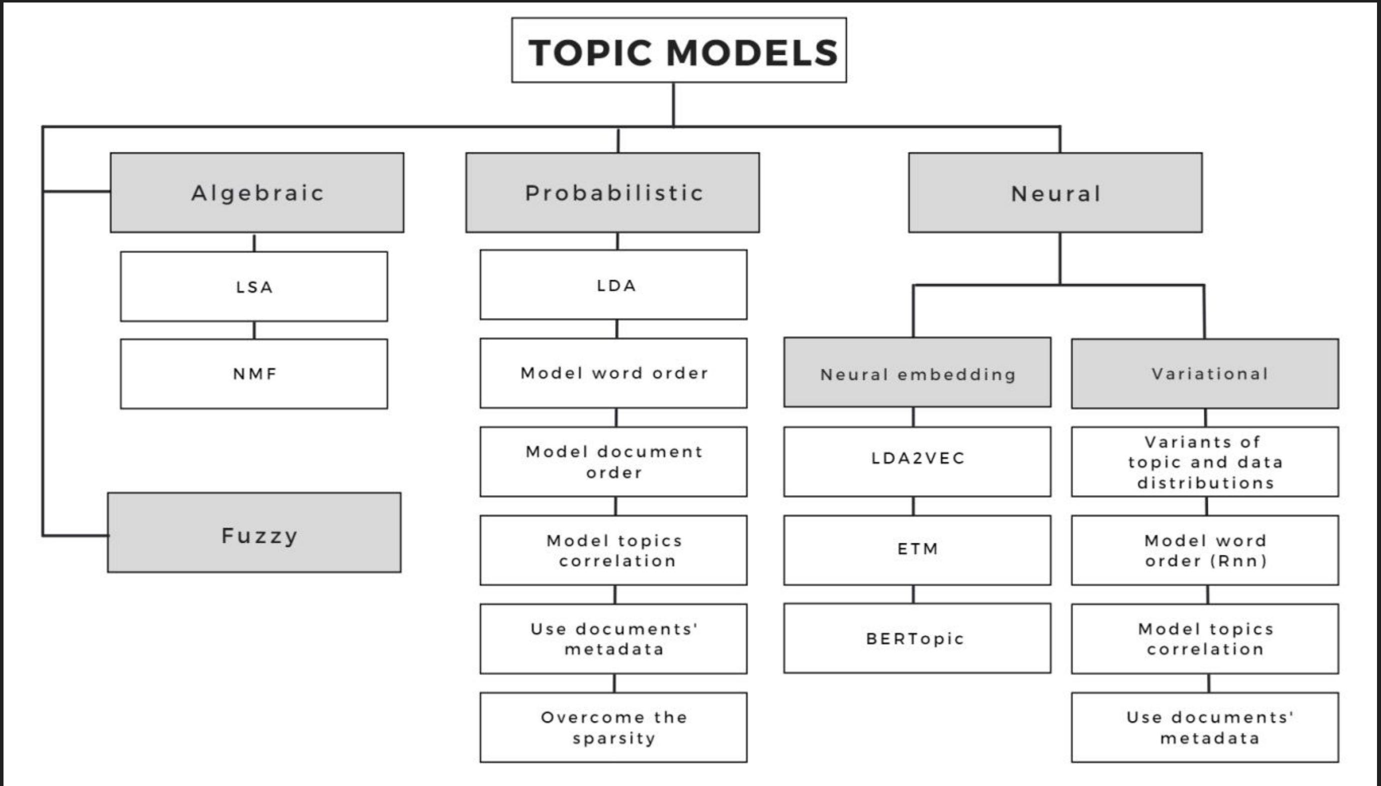
Software Architecture

Internal Quality Attributes


Refactoring Heuristics
Search-based Refactoring
Refactoring
Feature Location
Mining Software Repositories

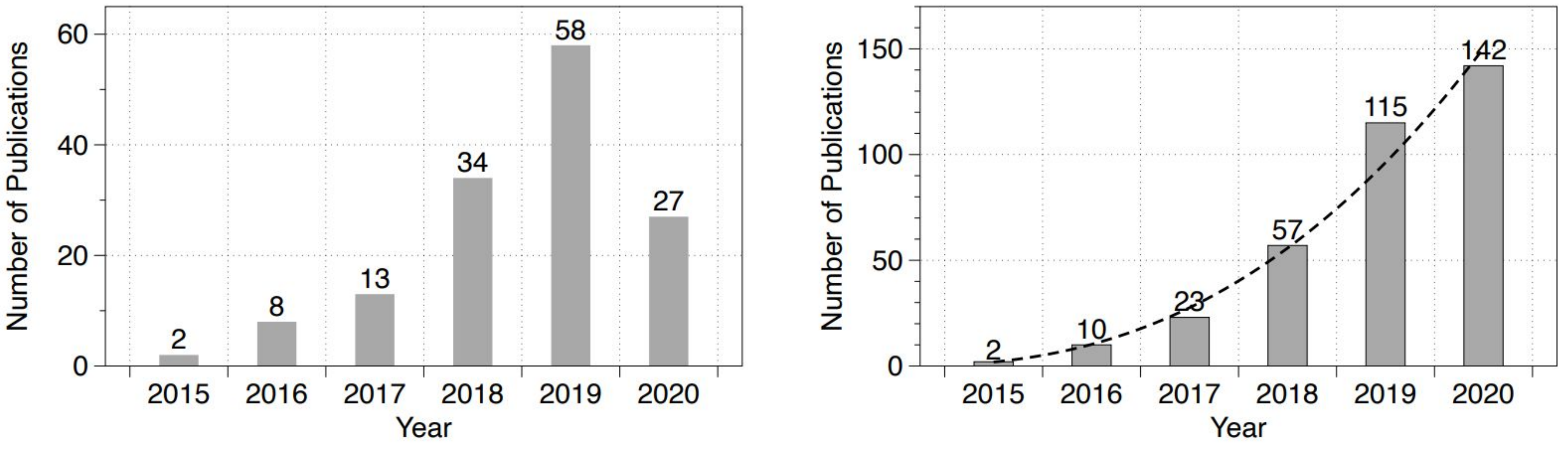
#2
Lack of attention to the fast-evolving literature

Example: State-of-the-art for Topic Modeling



Source: Abdelrazek, Aly, et al. "Topic modeling algorithms and applications: A survey." Information Systems (2022): 102131.

Another Example: Deep Learning for Software Engineering



(a) Number of publications per year.

(b) Cumulative number of publications per year.

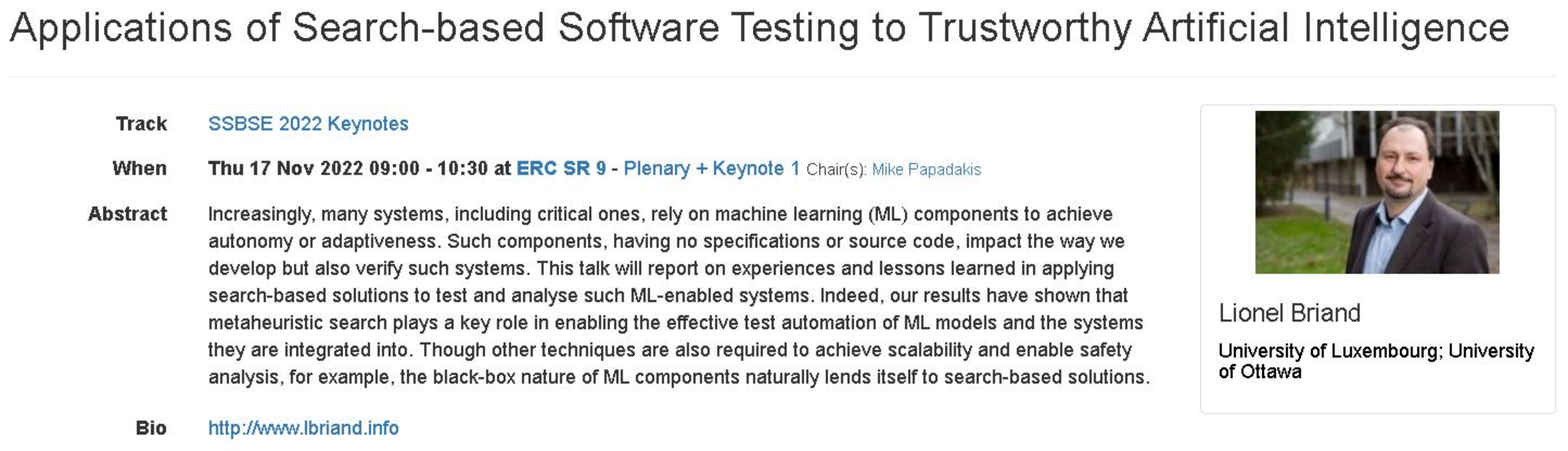
Yang, Yanming, et al. "A survey on deep learning for software engineering." ACM Computing Surveys (CSUR) 54.10s (2022): 1-73.

Number of Papers per Topic - ICSE

Quick search on dblp, only looking at the sessions and papers titles:

Year	* Learning	NN/DL	LLM / Pre-trained	SBSE
2016	3	0	0	**1**
2017	**4**	**1**	0	**3**
2018	**1**	**3**	0	**9**
2019	**4**	**11**	0	**1**
2020	**18**	**10**	0	0
2021	**16**	**12**	**1**	0
2022	**39**	**22**	**8**	0
2023	**33**	**17**	**11**	0

Is SBSE dead?



Applications of Search-based Software Testing to Trustworthy Artificial Intelligence

Track SSBSE 2022 Keynotes

When **Thu 17 Nov 2022 09:00 - 10:30 at ERC SR 9** - Plenary + Keynote 1 Chair(s): Mike Papadakis

Abstract Increasingly, many systems, including critical ones, rely on machine learning (ML) components to achieve autonomy or adaptiveness. Such components, having no specifications or source code, impact the way we develop but also verify such systems. This talk will report on experiences and lessons learned in applying search-based solutions to test and analyse such ML-enabled systems. Indeed, our results have shown that metaheuristic search plays a key role in enabling the effective test automation of ML models and the systems they are integrated into. Though other techniques are also required to achieve scalability and enable safety analysis, for example, the black-box nature of ML components naturally lends itself to search-based solutions.

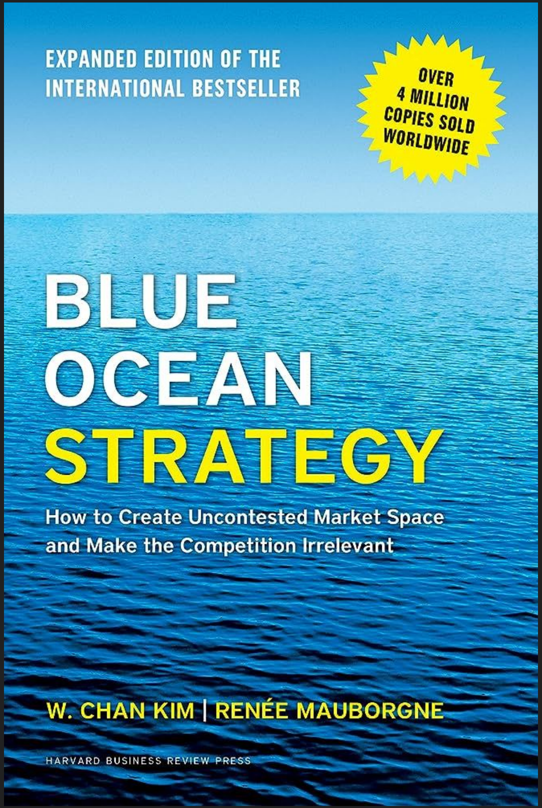
Bio http://www.lbriand.info

Lionel Briand

University of Luxembourg; University of Ottawa

https://conf.researchr.org/home/ssbse-2022

Red/Blue Ocean Strategy



EXPANDED EDITION OF THE
INTERNATIONAL BESTSELLER

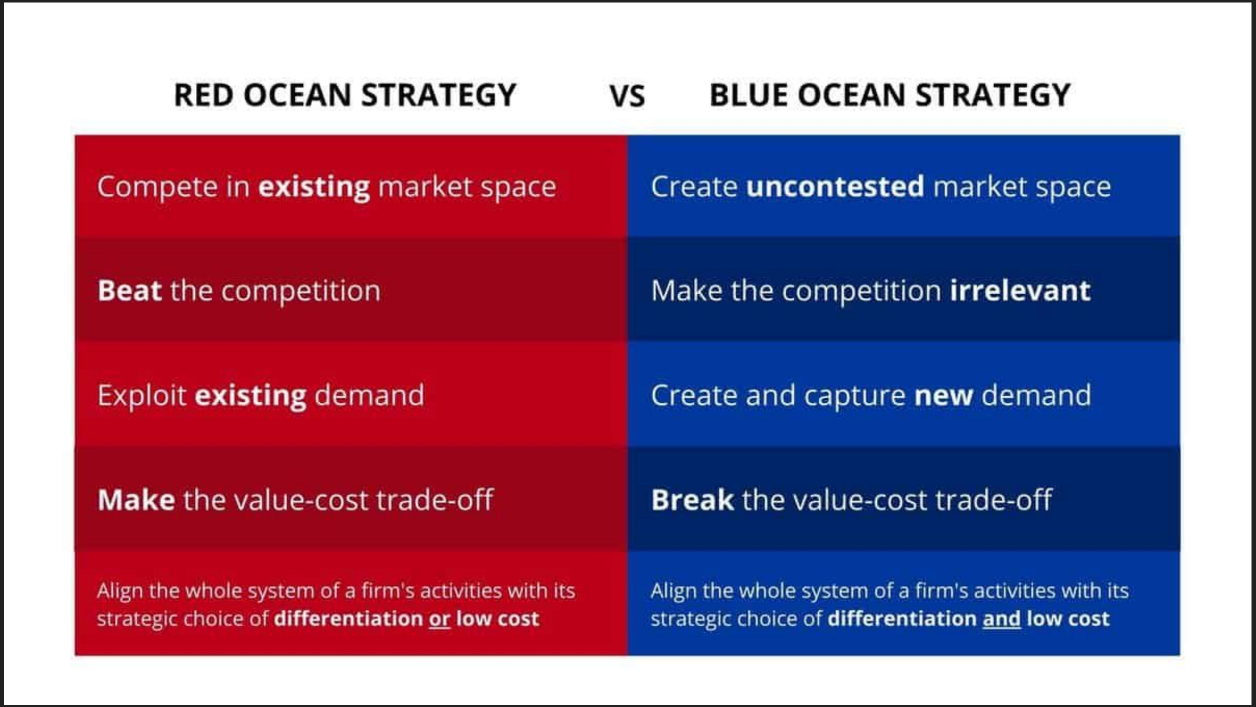
OVER 4 MILLION COPIES SOLD WORLDWIDE

BLUE
OCEAN
STRATEGY

How to Create Uncontested Market Space
and Make the Competition Irrelevant

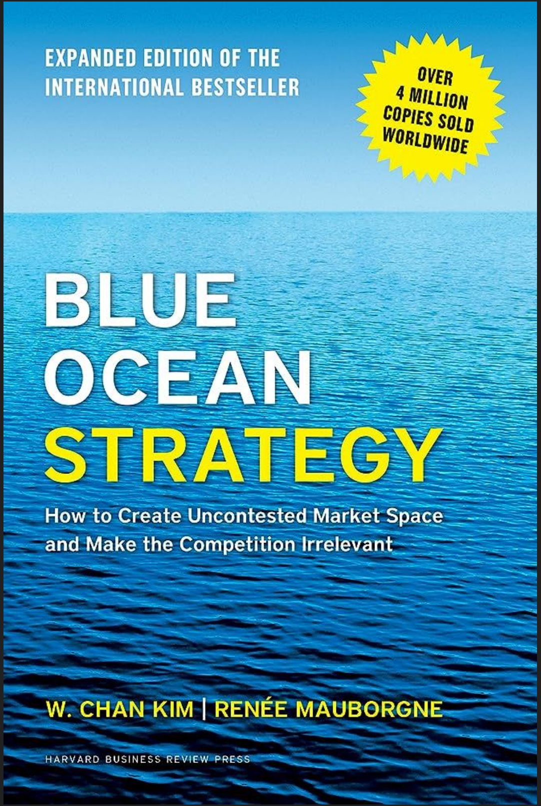
W. CHAN KIM | RENÉE MAUBORGNE

HARVARD BUSINESS REVIEW PRESS

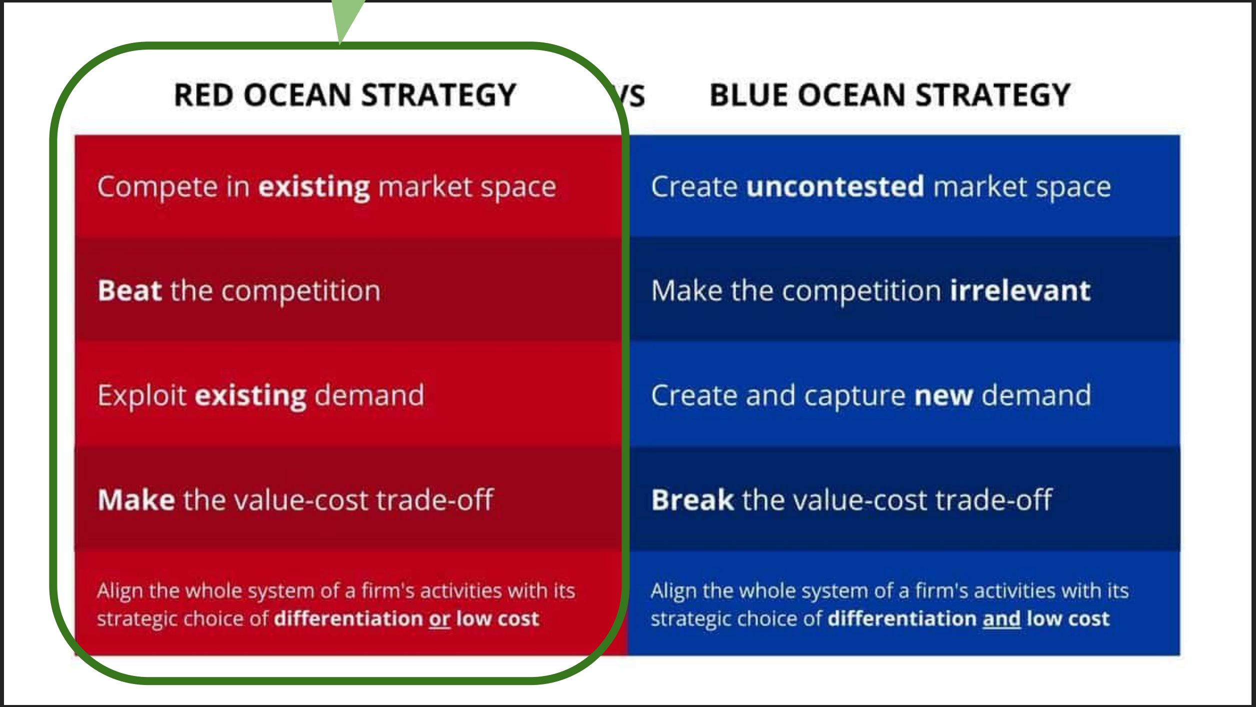


RED OCEAN STRATEGY	VS	BLUE OCEAN STRATEGY
Compete in **existing** market space		Create **uncontested** market space
Beat the competition		Make the competition **irrelevant**
Exploit **existing** demand		Create and capture **new** demand
Make the value-cost trade-off		**Break** the value-cost trade-off
Align the whole system of a firm's activities with its strategic choice of **differentiation or low cost**		Align the whole system of a firm's activities with its strategic choice of **differentiation and low cost**

Red/Blue Ocean Strategy


EXPANDED EDITION OF THE
INTERNATIONAL BESTSELLER
OVER 4 MILLION COPIES SOLD WORLDWIDE
BLUE OCEAN STRATEGY
How to Create Uncontested Market Space and Make the Competition Irrelevant
W. CHAN KIM | RENÉE MAUBORGNE
HARVARD BUSINESS REVIEW PRESS

Most research is in the **red ocean**



RED OCEAN STRATEGY	VS	BLUE OCEAN STRATEGY
Compete in **existing** market space		Create **uncontested** market space
Beat the competition		Make the competition **irrelevant**
Exploit **existing** demand		Create and capture **new** demand
Make the value-cost trade-off		**Break** the value-cost trade-off
Align the whole system of a firm's activities with its strategic choice of **differentiation or low cost**		Align the whole system of a firm's activities with its strategic choice of **differentiation and low cost**

#3
Too much effort spent in contributions unrelated to my thesis

#4
Too much focus on technical aspects rather than on the scientific ones

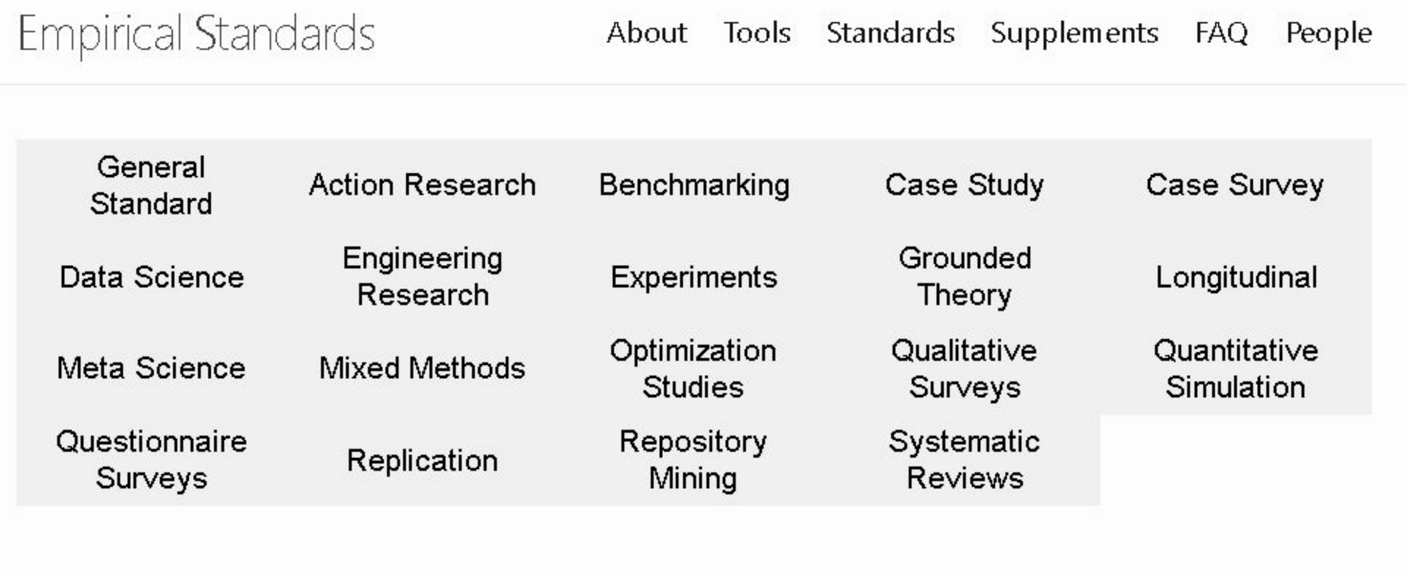
Lots of effort on…

- Writing beautiful code
- Creating a tool that is easy to use
- Testing different frameworks and libraries to use in the tool
- Collaborating in other students’ studies

Rather than…

- Being up to date with the literature
- Finding out which datasets could be used in my research
- Validating and adapting the proposed technique
- Designing a sound and robust evaluation study
- Doing preliminarily studies and publishing first author papers

Empirical Standards for Software Engineering Research



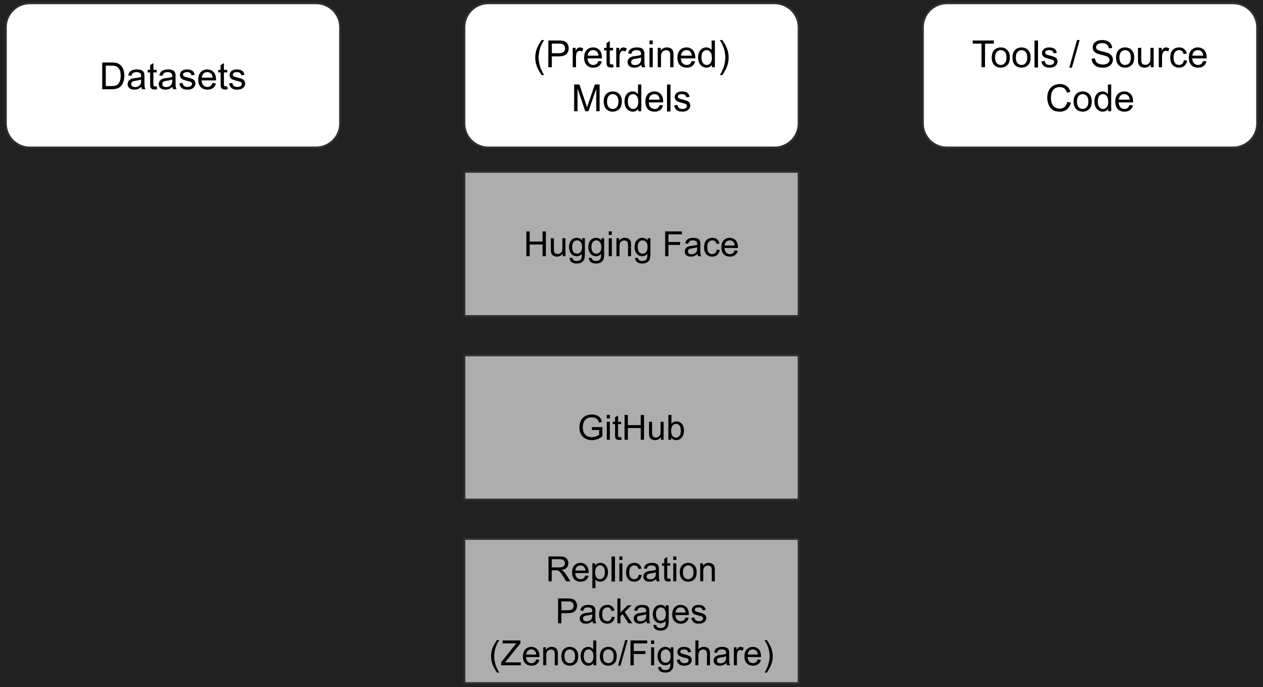
General Standard	Action Research	Benchmarking	Case Study	Case Survey
Data Science	Engineering Research	Experiments	Grounded Theory	Longitudinal
Meta Science	Mixed Methods	Optimization Studies	Qualitative Surveys	Quantitative Simulation
Questionnaire Surveys	Replication	Repository Mining	Systematic Reviews	

Available at: https://sigsoft.org/EmpiricalStandards/docs/

Get to Know What is Available

- Datasets
  - Replication Packages (Zenodo/Figshare)
  - MSR Mining Challenge
  - Create and publish your own dataset
- (Pretrained) Models
- Tools / Source Code

Get to Know What is Available



Datasets

(Pretrained) Models

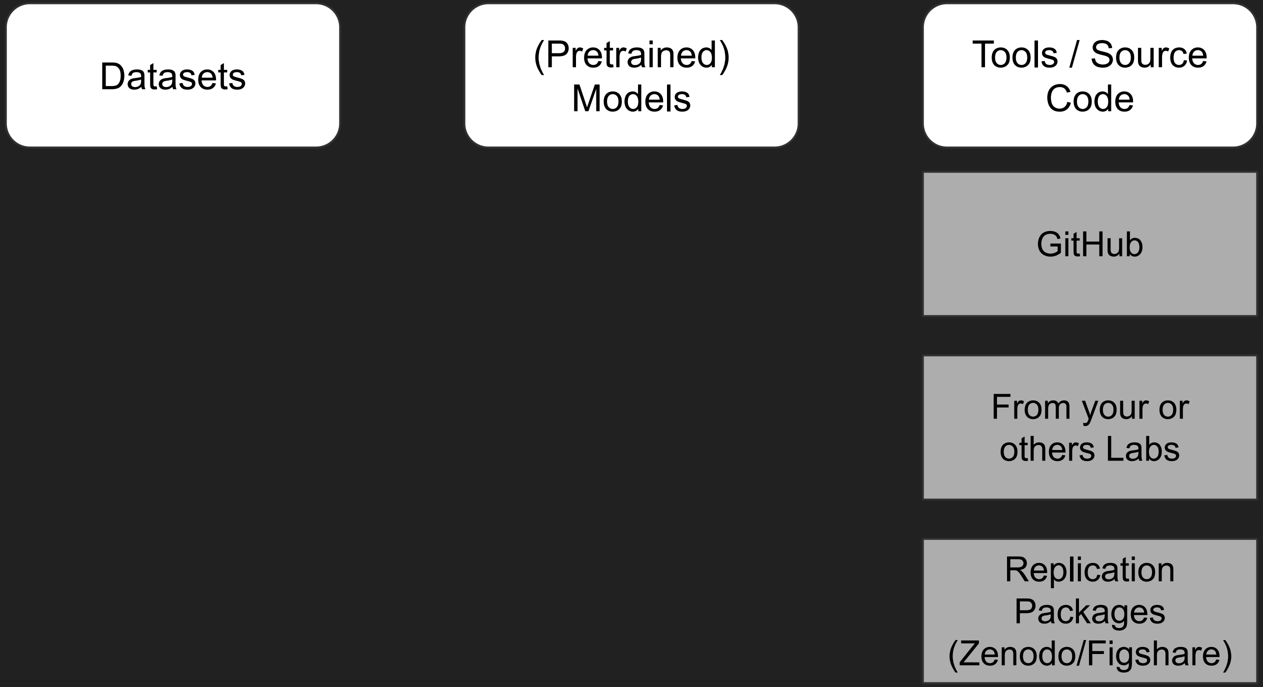
Tools / Source Code

Hugging Face

GitHub

Replication Packages (Zenodo/Figshare)

Get to Know What is Available



Datasets

(Pretrained) Models

Tools / Source Code

GitHub

From your or others Labs

Replication Packages (Zenodo/Figshare)

#5
The chosen scope was larger than it should have been

Revisiting the Requirements

1. Consideration of Heterogeneous Information
2. Context-Sensitive Detection
3. Feature Modularity Awareness
4. Effective Recommendations

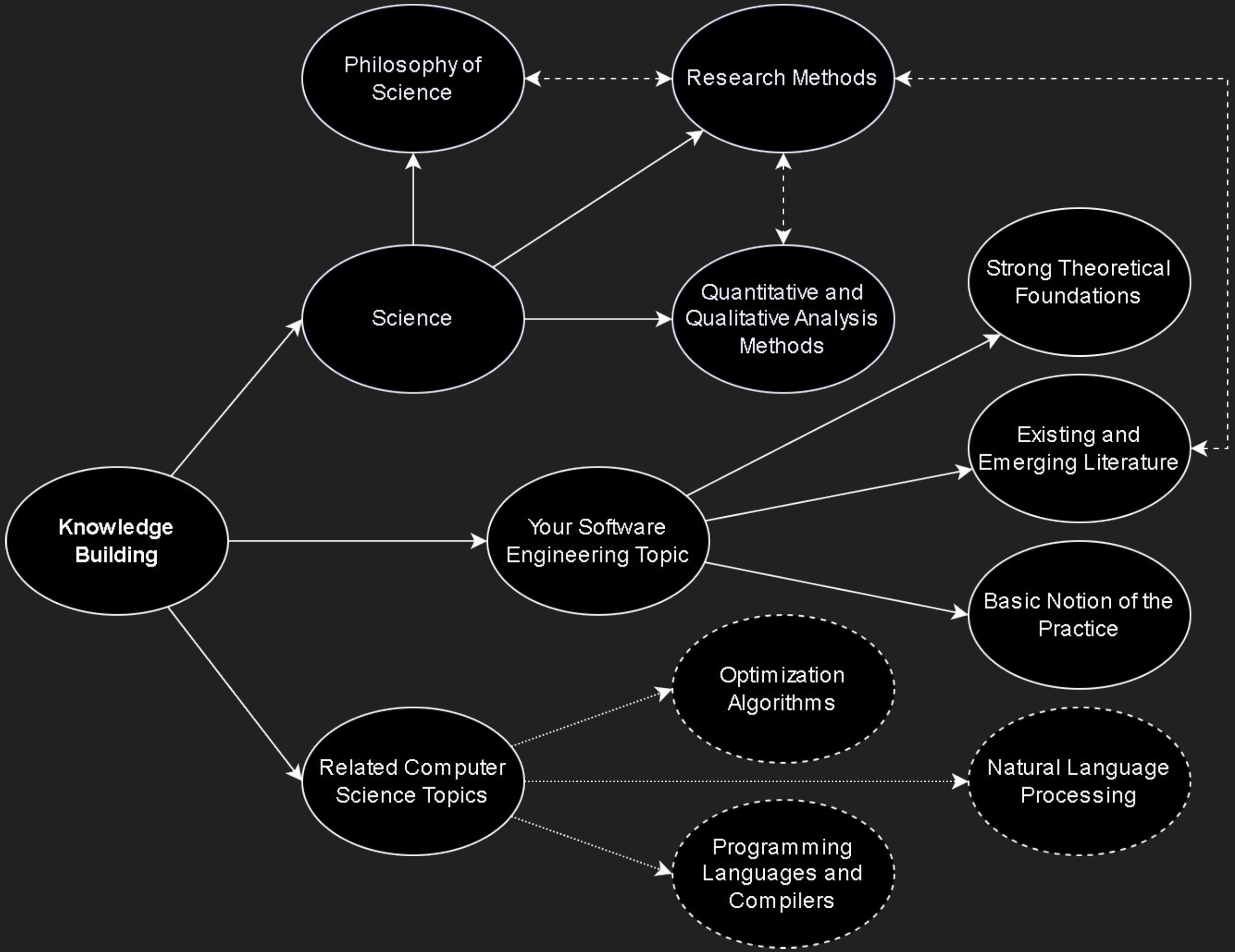
Revisiting the Requirements

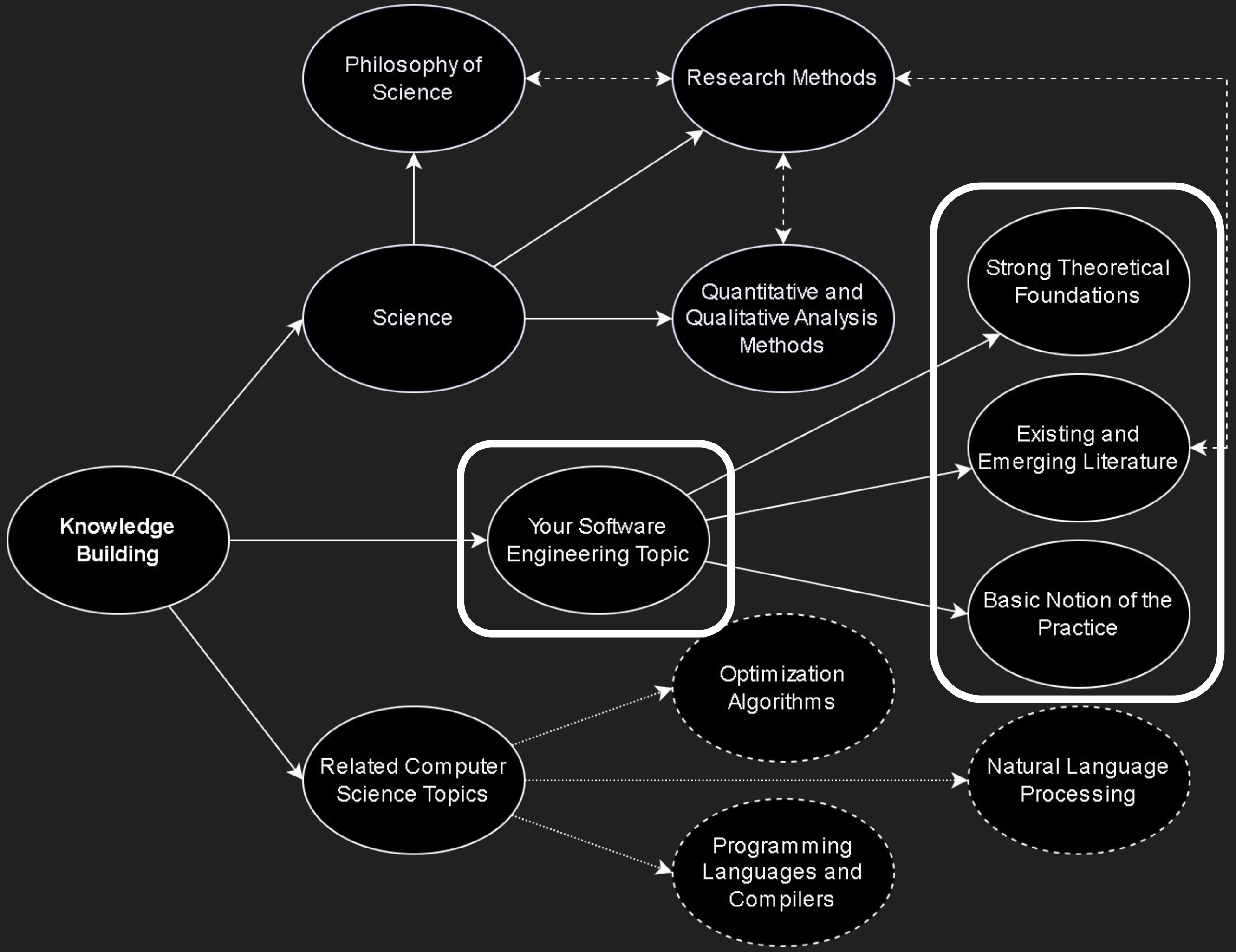
1. Consideration of Heterogeneous Information
2. Context-Sensitive Detection
3. Feature Modularity Awareness
4. Effective Recommendations

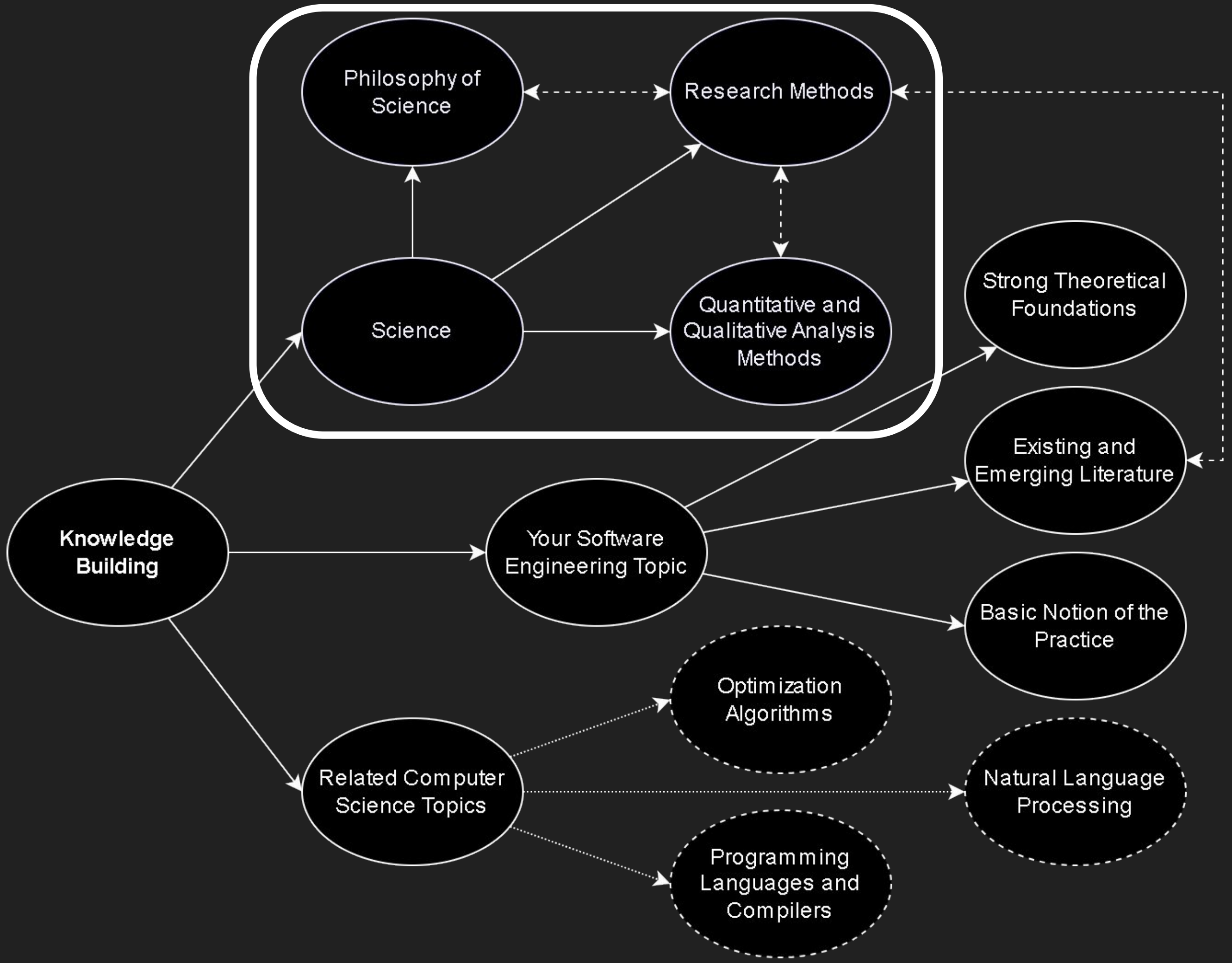
Each requirement alone could be deeply explored in a thesis

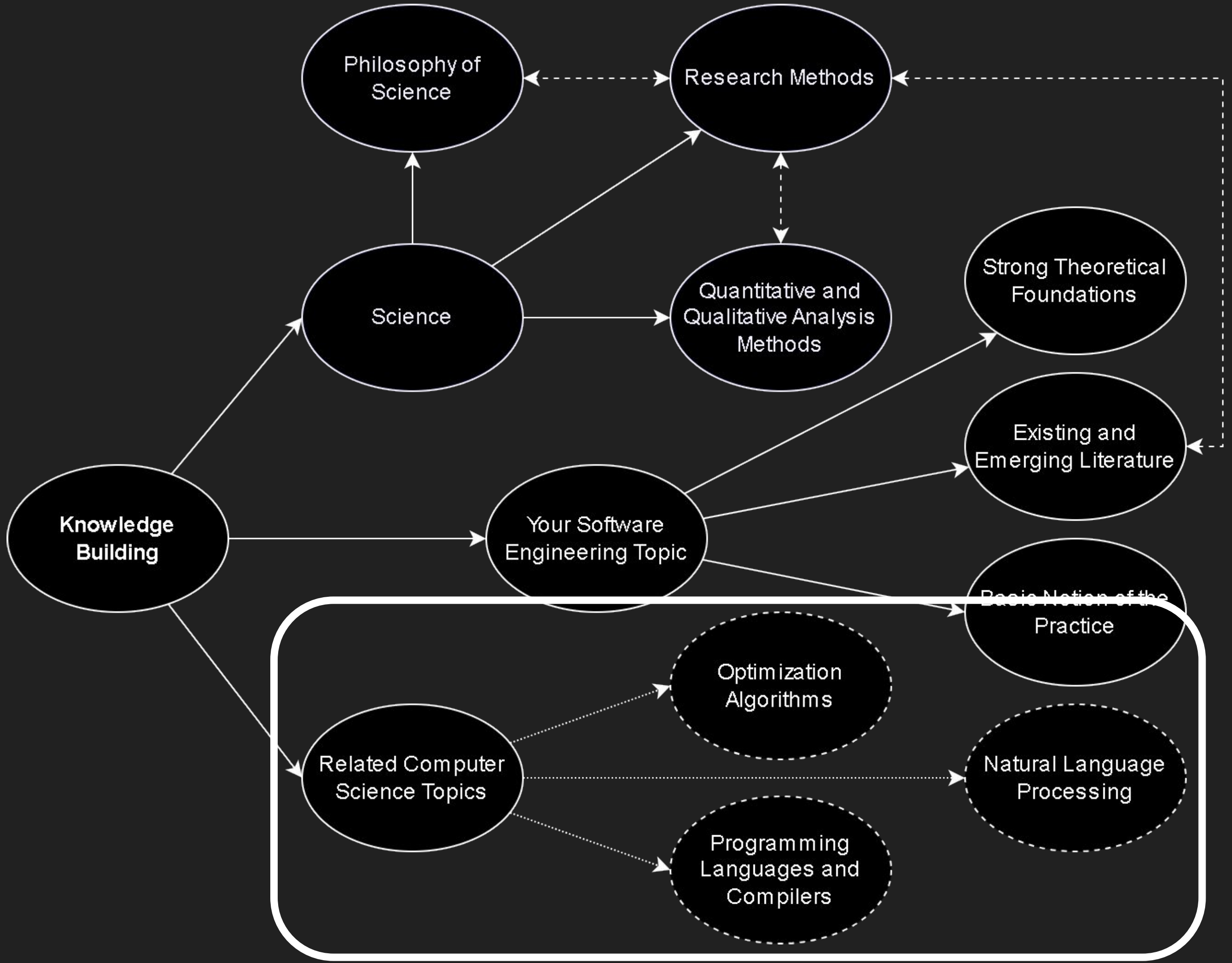
Mind Maps to Guide your Journey

Mind Map 1: Knowledge Building


Philosophy of Science
Research Methods
Science
Quantitative and Qualitative Analysis Methods
Strong Theoretical Foundations
Existing and Emerging Literature
Knowledge Building
Your Software Engineering Topic
Basic Notion of the Practice
Optimization Algorithms
Related Computer Science Topics
Natural Language Processing
Programming Languages and Compilers


Philosophy of Science
Research Methods
Science
Quantitative and Qualitative Analysis Methods
Strong Theoretical Foundations
Existing and Emerging Literature
Basic Notion of the Practice
Knowledge Building
Your Software Engineering Topic
Optimization Algorithms
Related Computer Science Topics
Natural Language Processing
Programming Languages and Compilers


Philosophy of Science
Research Methods
Science
Quantitative and Qualitative Analysis Methods
Strong Theoretical Foundations
Existing and Emerging Literature
Knowledge Building
Your Software Engineering Topic
Basic Notion of the Practice
Optimization Algorithms
Related Computer Science Topics
Natural Language Processing
Programming Languages and Compilers



Philosophy of Science

Research Methods

Science

Quantitative and Qualitative Analysis Methods

Strong Theoretical Foundations

Existing and Emerging Literature

Knowledge Building

Your Software Engineering Topic

Basic Notion of the Practice

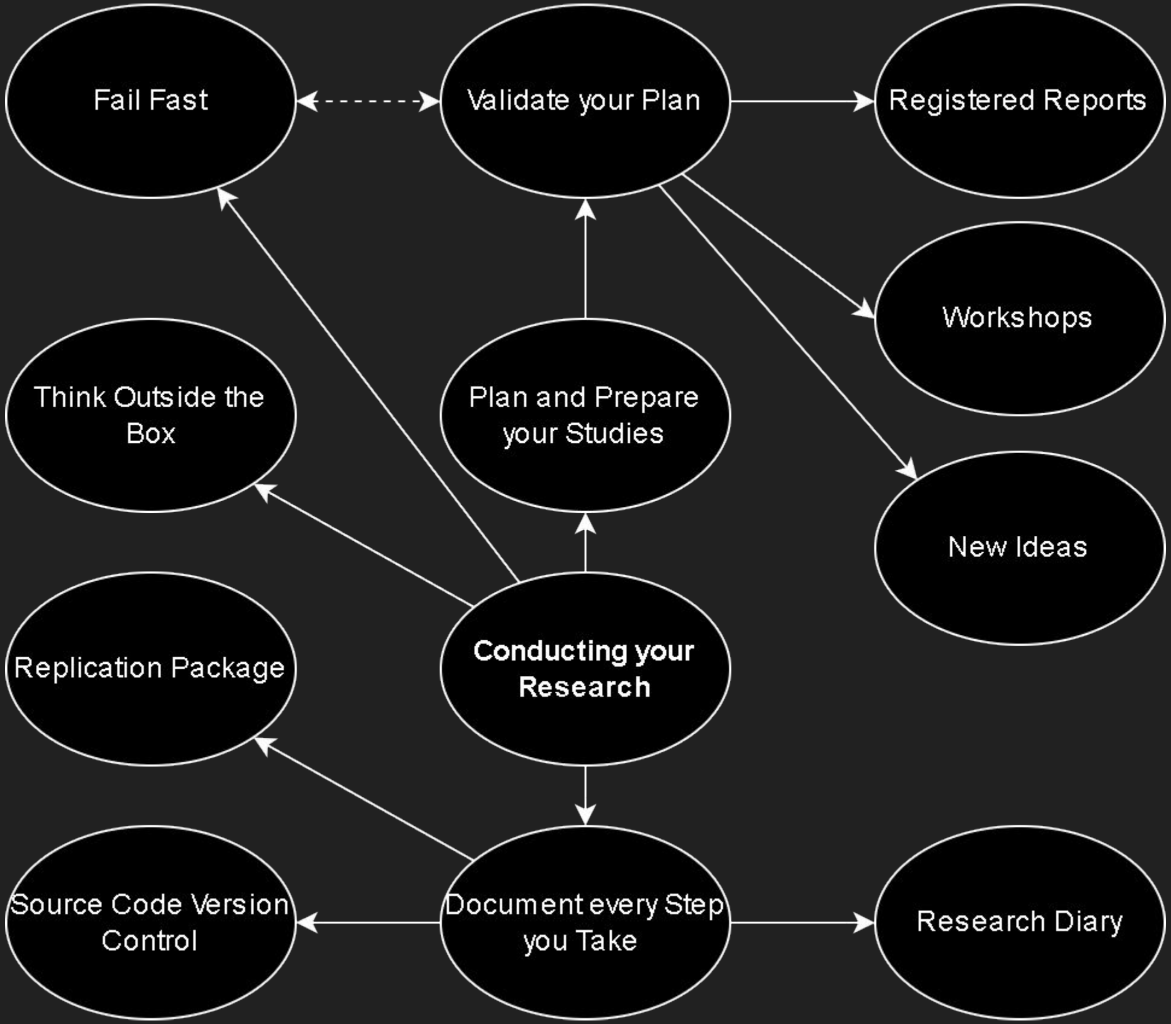
Optimization Algorithms

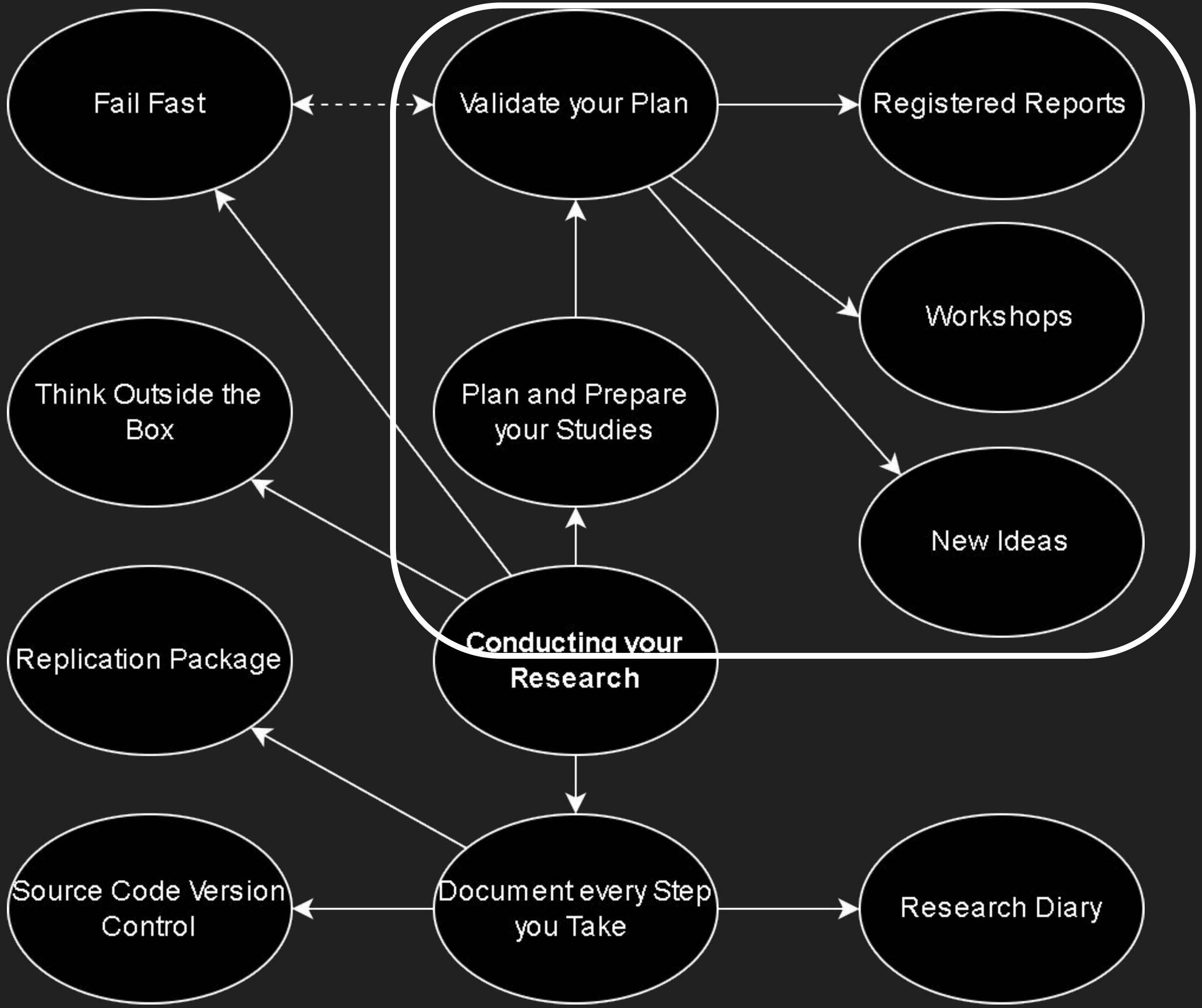
Related Computer Science Topics

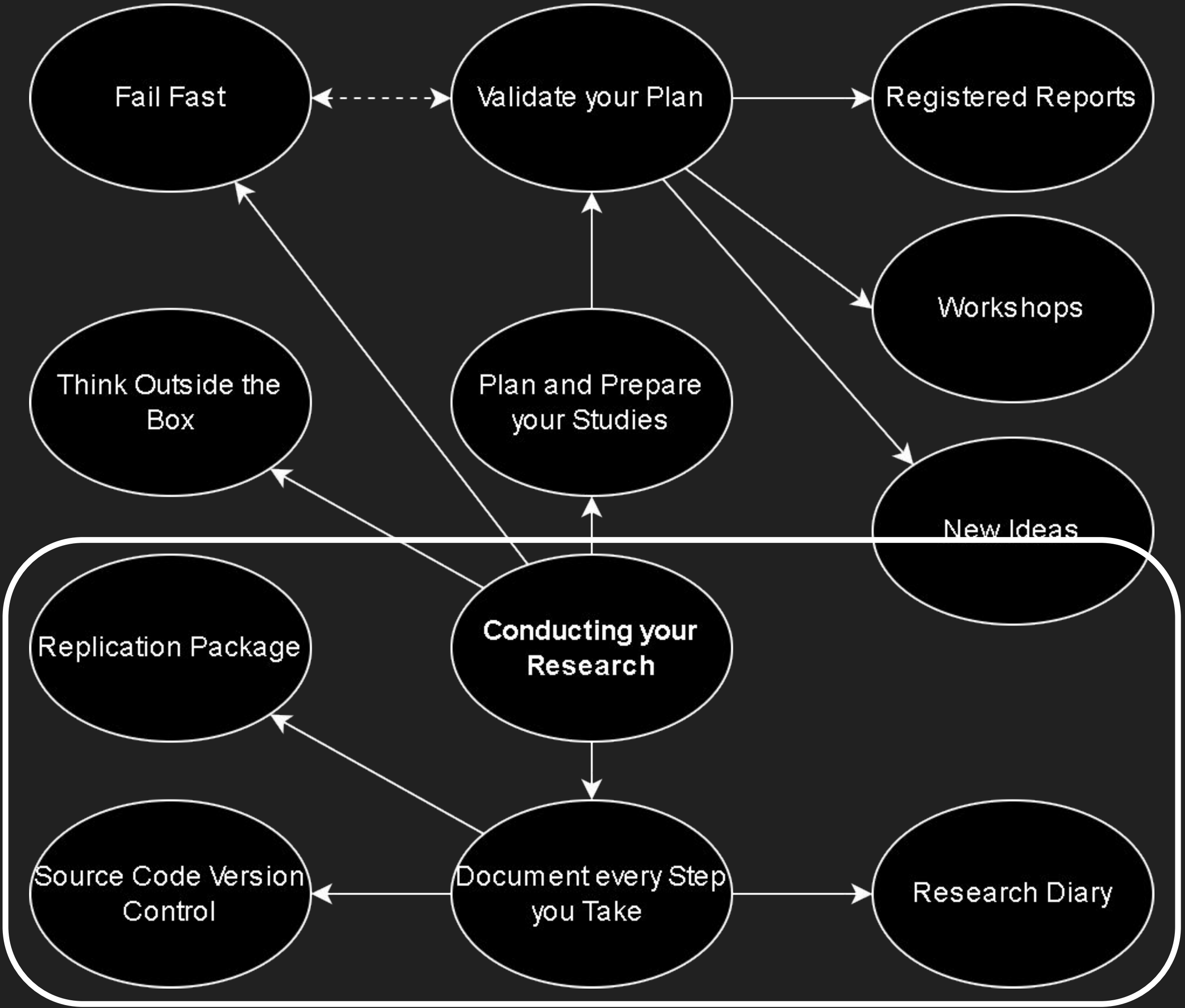
Natural Language Processing

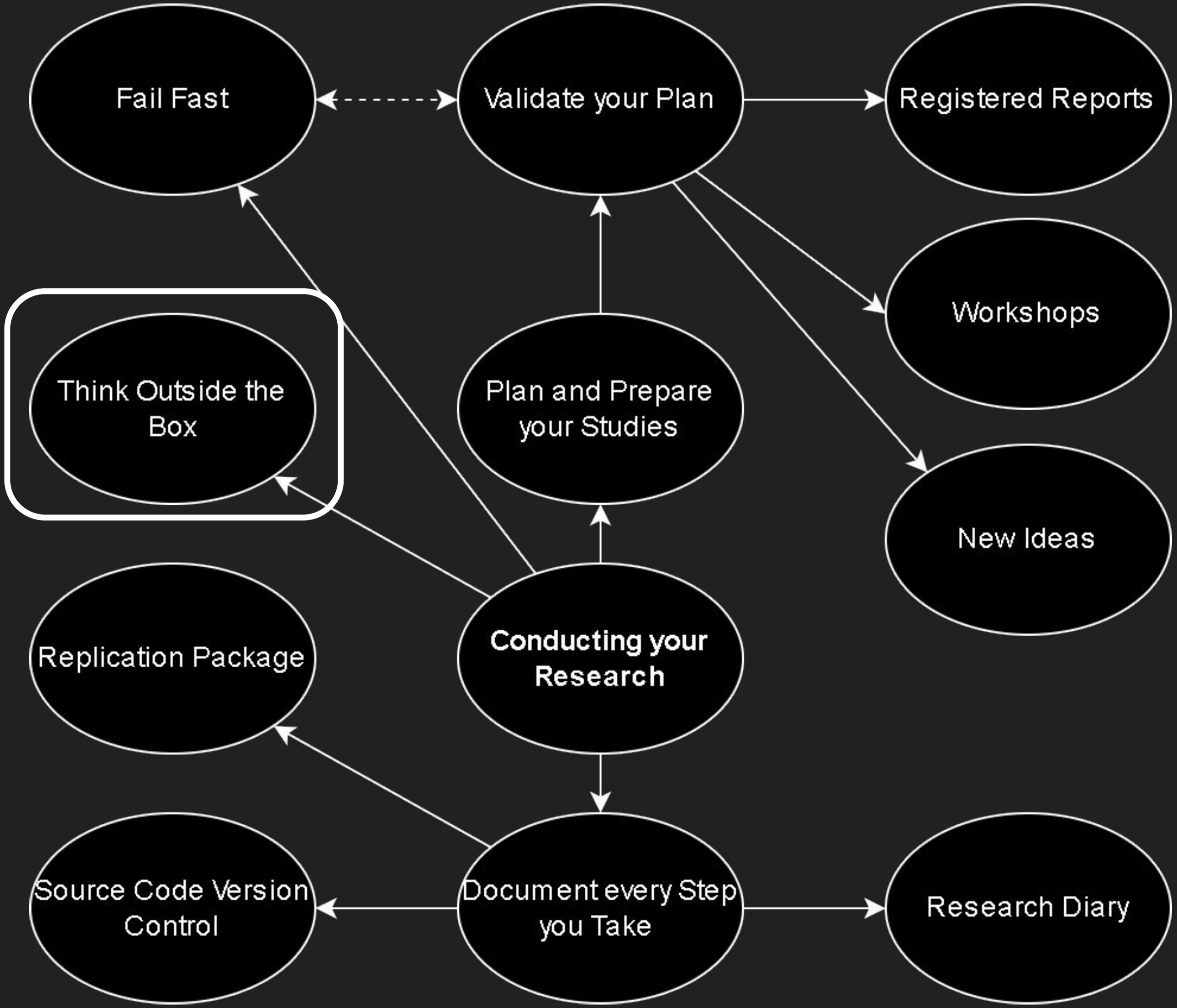
Programming Languages and Compilers

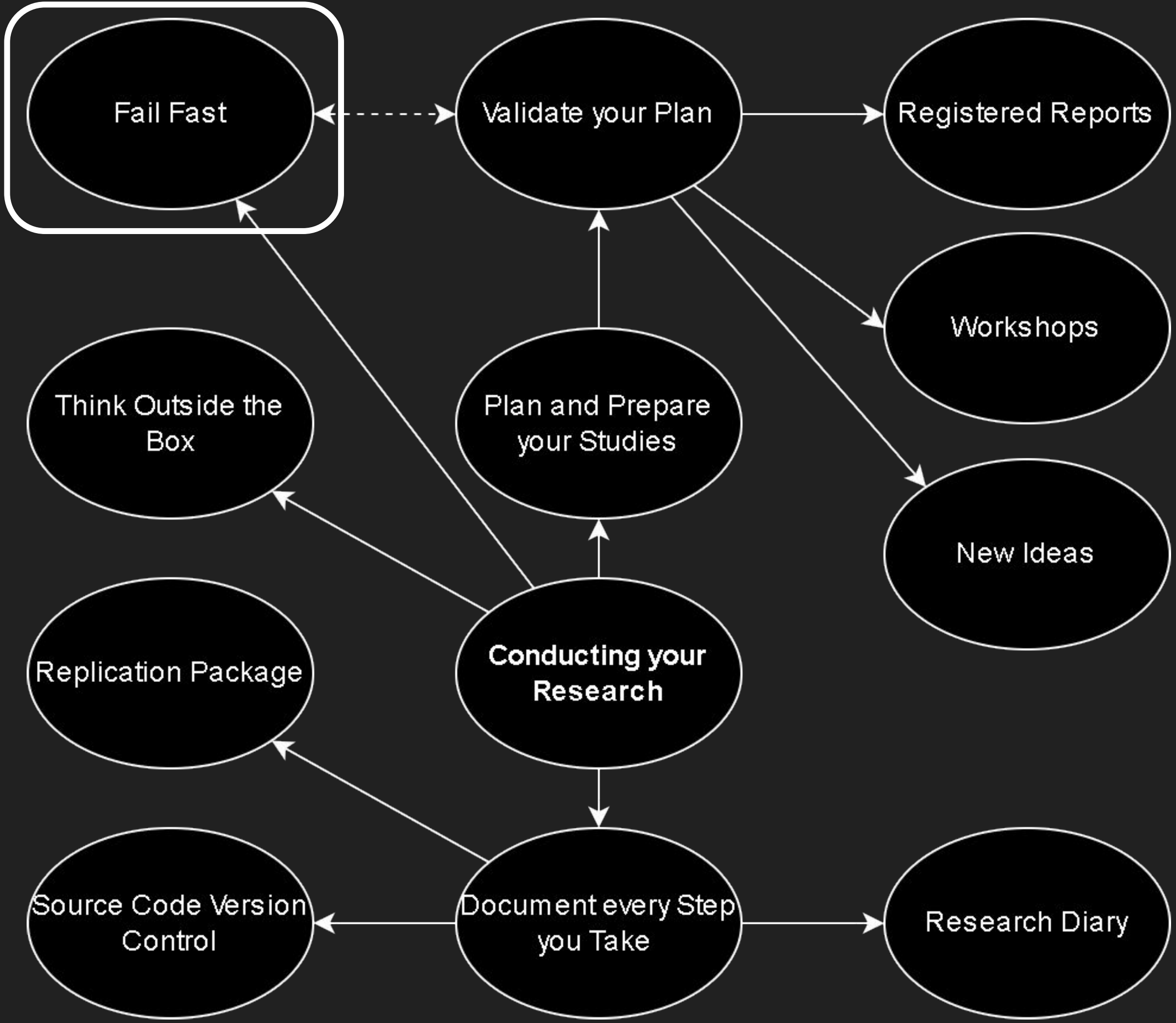
Mind Map 2: Conducting your research


Fail Fast
Validate your Plan
Registered Reports
Workshops
Think Outside the Box
Plan and Prepare your Studies
New Ideas
Replication Package
Conducting your Research
Source Code Version Control
Document every Step you Take
Research Diary


Fail Fast
Validate your Plan
Registered Reports
Workshops
Think Outside the Box
Plan and Prepare your Studies
New Ideas
Replication Package
Conducting your Research
Source Code Version Control
Document every Step you Take
Research Diary


Fail Fast
Validate your Plan
Registered Reports
Workshops
Think Outside the Box
Plan and Prepare your Studies
New Ideas
Replication Package
Conducting your Research
Source Code Version Control
Document every Step you Take
Research Diary


Fail Fast
Validate your Plan
Registered Reports
Workshops
Think Outside the Box
Plan and Prepare your Studies
New Ideas
Replication Package
Conducting your Research
Source Code Version Control
Document every Step you Take
Research Diary


Fail Fast
Validate your Plan
Registered Reports
Workshops
Think Outside the Box
Plan and Prepare your Studies
New Ideas
Replication Package
Conducting your Research
Source Code Version Control
Document every Step you Take
Research Diary

In a Nutshell

- Select your research topic as soon as possible
- Avoid too large scopes
- Put effort on building the required knowledge
- Try to fill the gaps with collaborations
- Keep tracking the literature and adapt if needed

AI for Software Engineering

Pitfalls and Challenges

Willian Oizumi

CBSoft'23
CONGRESSO BRASILEIRO DE SOFTWARE
CAMPO GRANDE - MS

iSE'23
THE 3RD BRAZILIAN WORKSHOP ON
INTELLIGENT SOFTWARE ENGINEERING