Analyzing the influence of uncertainty on architectural decision making

Oleksandra Klymenko, 27.11.2017, Garching b. Muenchen

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Agenda

1. Motivation
2. Related Work
3. Research Questions
4. Research Process
5. Timeline
6. References

Motivation

- Software development stages are highly communication-intensive
- Developers, architects, testers, end users and other stakeholders constantly communicate
- System or its parts have to be regularly upgraded, fixed, or replaced
- Task management systems structure the process by implicitly capturing design decisions

Being documented in natural language text,
tasks descriptions often lack clarity and completeness,
causing more time and effort to be spent on their resolution

Can **uncertainty**
be one of the possible reasons?

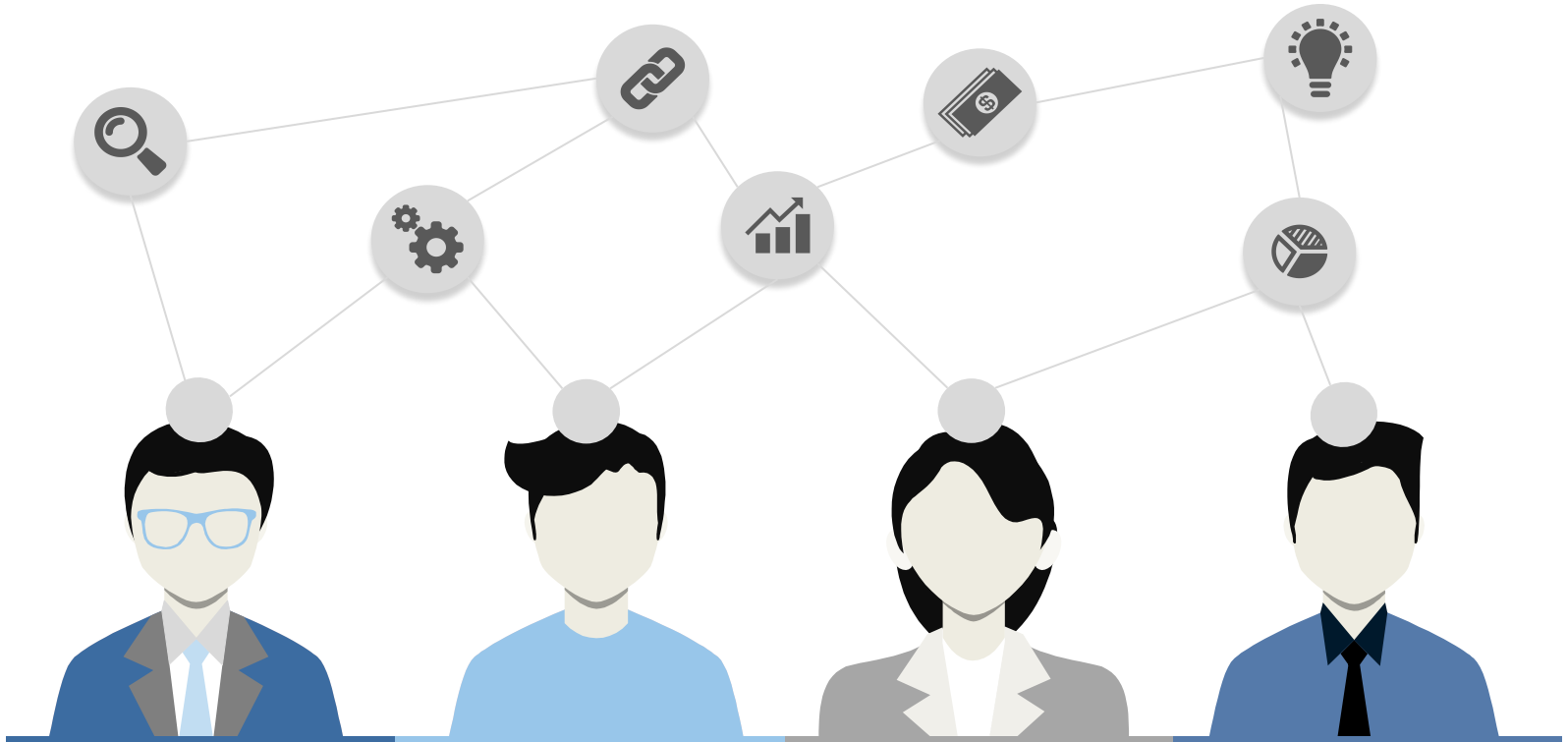
GOALS

1. Derive a method for measuring uncertainty in decision-making process
2. Provide interpretation of obtained measures for specific projects

Agenda

1. Motivation
2. Related Work
3. Research Questions
4. Research Process
5. Timeline
6. References

Related Work



Related Work

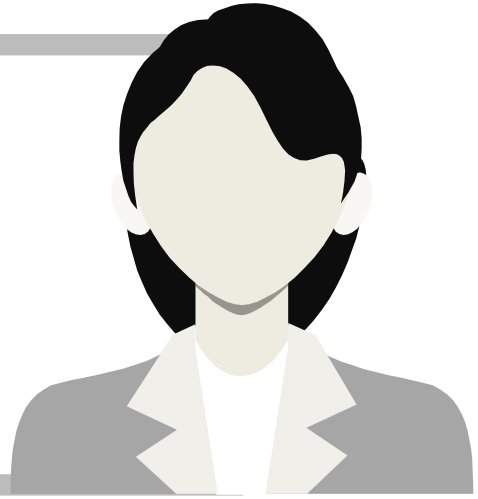
Linguistic uncertainty in natural language texts

- Linguistic uncertainty is an important linguistic phenomenon that is relevant in many fields (Vincze)
- In its most general sense, it can be interpreted as lack of information (Vincze)
- Different machine learning methods were proposed to detect uncertainty (Medlock et al.)
- A classification of linguistic uncertainty, providing words, that indicate uncertainty in a sentence was proposed by Jean et al. and Vincze
- E.g. Szarvas et al examined the recognition of uncertainty cues in context and introduced an uncertainty cue detection model for different domains and genres

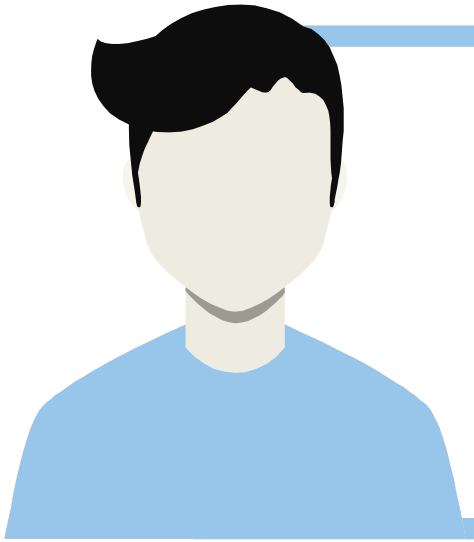
Uncertainty in Software Engineering

According to Letier et al.:

- Uncertainty is inevitable in software engineering
- Uncertainty complicates architecture decisions and may expose a software project to significant risk
- Software architects lack support in evaluating uncertainty, its impact on risk, and the value of reducing uncertainty before making critical decisions



Related Work



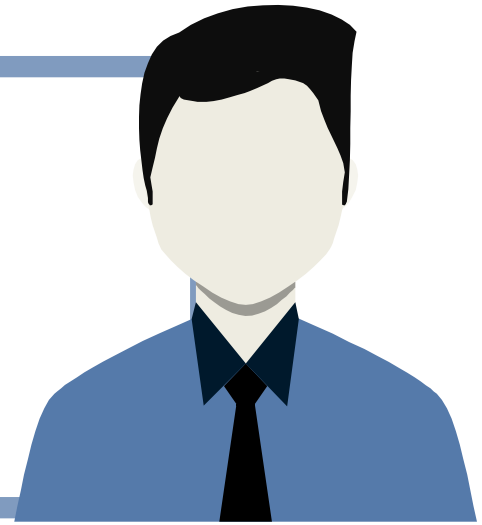
Software Architectural Design Decisions

- Software Architectural Design Decisions are implicit and rarely documented (Capilla et al)
- It is important to maintain clarity and completeness of knowledge expressed by SA stakeholders (K. A. de Graaf et al.)
- Several AKM tools were proposed, e.g. PAKME by Babar and Gorton and Decision Architect by Manteuel et al.
- There is a need for substantial improvement of AKM tools (Capilla et al.)
- Stakeholders tend to rather use agile tools (Miesbauer et al.)

Architectural Assumptions in decision-making process

Recent study (2017) by C. Yang et al. claims:

- concept is not commonly used in industry
- important in architecting and software development
- architects frequently make AA in their work
- lack of approaches, tools, and guidelines for AA management
- there is a connection between AA and certain software artifacts



Agenda

1. Motivation
2. Related Work
3. Research Questions
4. Research Process
5. Timeline
6. References

Research Questions

Research Question 1

How to quantify uncertainty density throughout the decision life-cycle?

Research Question 2

How can the change of uncertainty density over the decision life-cycle be interpreted?

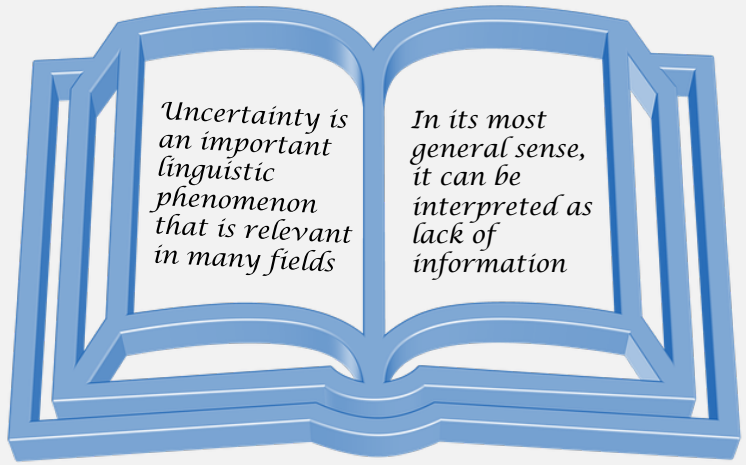
Agenda

1. Motivation
2. Related Work
3. Research Questions
4. Research Process
5. Timeline
6. References

Research Process

Step 1:

Review the literature on the topics related to the research

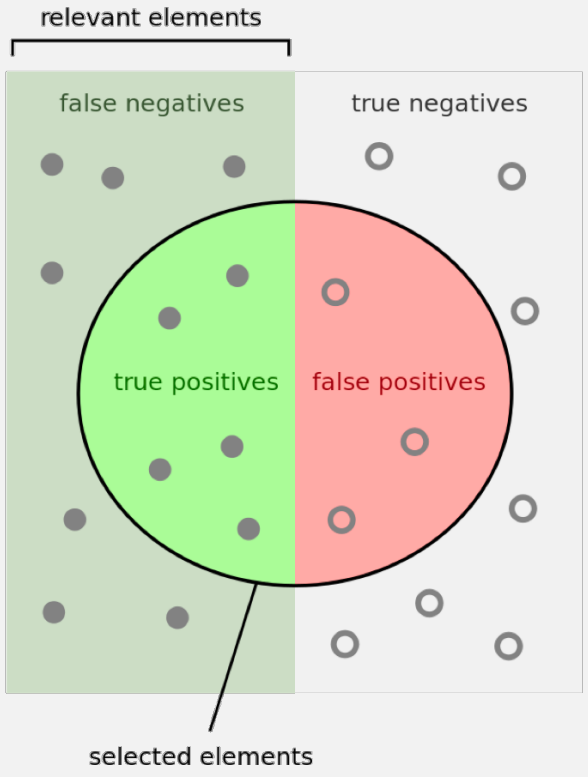


Step 2:

Explore the data and define the scope of research

Research Process

Step 3.1:
Calculate the efficiency of
the uncertainty detection algorithm



Decision title:add poll function for BoundedPriorityQueue /
Decision link: https://issues.apache.org/jira/browse/SPARK-21401
Description:

The most of BoundedPriorityQueue usages in ML/MLLIB are:

Get the value of BoundedPriorityQueue, then sort it.

For example, in Word2Vec: pq.toSeq.sortBy(-_._2)

in ALS, pq.toArray.sorted()

The test results show using pq.poll() is much faster than sort the value.

For example, in PR https://github.com/apache/spark/pull/18624

We get the sorted value of pq by the following code:

{quote}

var size = pq.size

while(size > 0) {

size -= 1

val factor = pq.poll

}{quote} If using the generally UNCERTAINTY used methods: pq.toArray.sorted() to get the sorted value of pq. There is about 10% performance reduction. It is good to add the poll function for BoundedPriorityQueue, since many usages of PQ need the sorted value.

Comment 4 (Author: peng.meng@intel.com, Timestamp: 2017-07-13T15:24:21.614+0000)

Sure, I will add isEmpty and maybe UNCERTAINTY some other functions, and tests cases. Thanks.

None
Comment 5 (Author: srowen, Timestamp: 2017-07-15T08:27:15.437+0000)

[~peng.meng@intel.com] on re-reading this, where is something sorted only to get the top element? word2vec seems UNCERTAINTY to sort the list when it needs to produces a fully sorted list. The PR you link to isn't replacing a sort, and it's not in general UNCERTAINTY faster to sort something this way. What is the benefit here?

Research Process

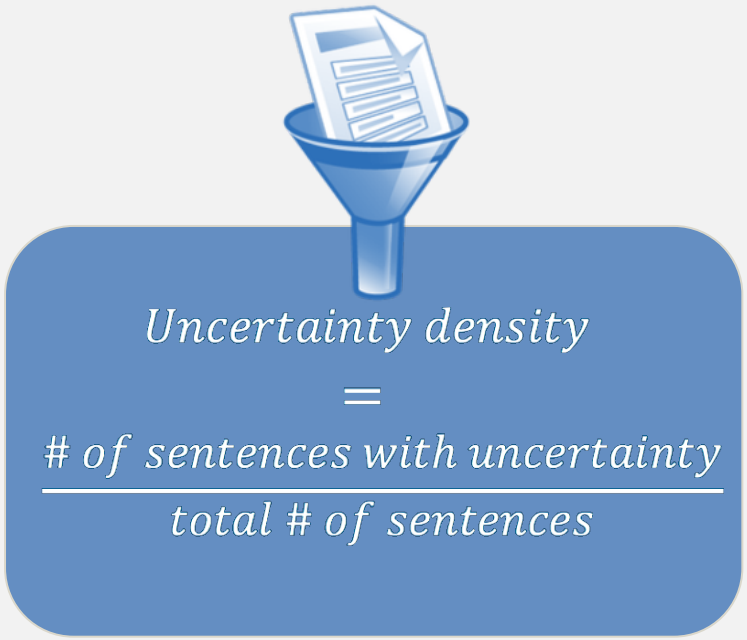
- Python code snippet for detecting uncertainty:

```
uncertainties = [
    "think", "thought", "thinking", "almost",
    "apparent", "apparently", "appear", "appeared", "appears", "approximately", "around",
    "assume", "assumed", "certain amount", "certain extent", "certain level", "claim",
    "claimed", "doubt", "doubtful", "essentially", "estimate",
    "estimated", "feel", "felt", "frequently", "from our perspective", "generally", "guess",
    "in general", "in most cases", "in most instances", "in our view", "indicate", "indicated",
    "largely", "likely", "mainly", "may", "maybe", "might", "mostly", "often", "on the whole",
    "ought", "perhaps", "plausible", "plausibly", "possible", "possibly", "postulate",
    "postulated", "presumable", "probable", "probably", "relatively", "roughly", "seems",
    "should", "sometimes", "somewhat", "suggest", "suggested", "suppose", "suspect", "tend to",
    "tends to", "typical", "typically", "uncertain", "uncertainly", "unclear", "unclearly",
    "unlikely", "usually", "broadly", "tended to", "presumably", "suggests",
    "from this perspective", "from my perspective", "in my view", "in this view", "in our opinion",
    "in my opinion", "to my knowledge", "fairly", "quite", "rather", "argue", "argues", "argued",
    "claims", "feels", "indicates", "supposed", "supposes", "suspects", "postulates", "IMHO"
]
ents = ['UNCERTAINTY']
decision = [d for d in decisions if d['key']=='SPARK-22180'][0]
```

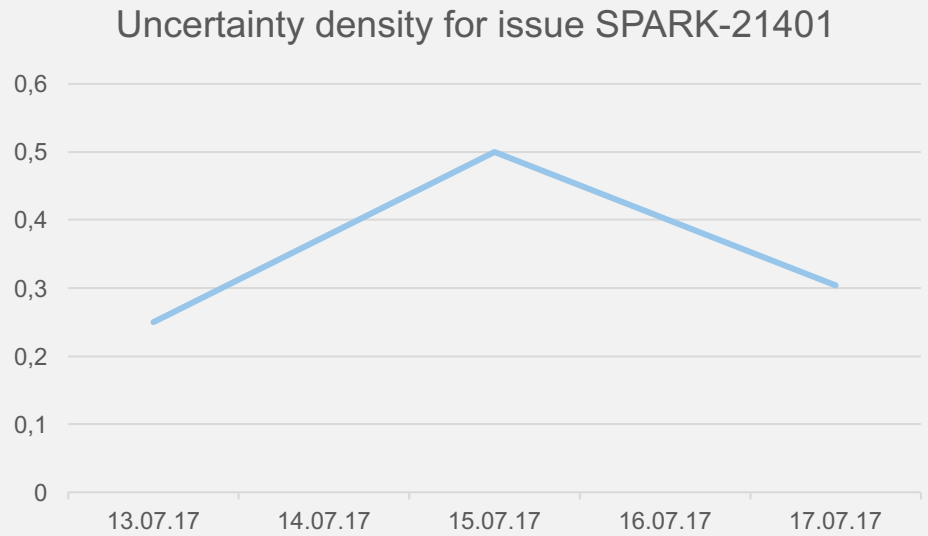
- The set of uncertainty words was compiled by Hyland in his book from year 2005.

Research Process

Step 3.2:
Calculate
the uncertainty density per issue



- Line graph representing the change of uncertainty density throughout the issue life-cycle

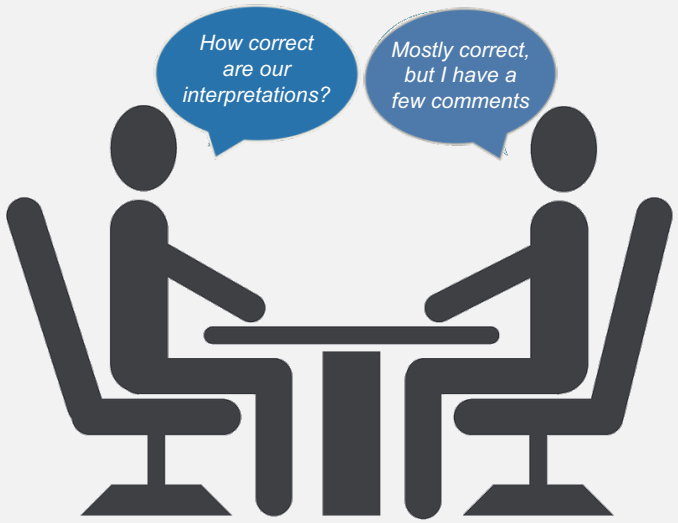


Hypotheses:

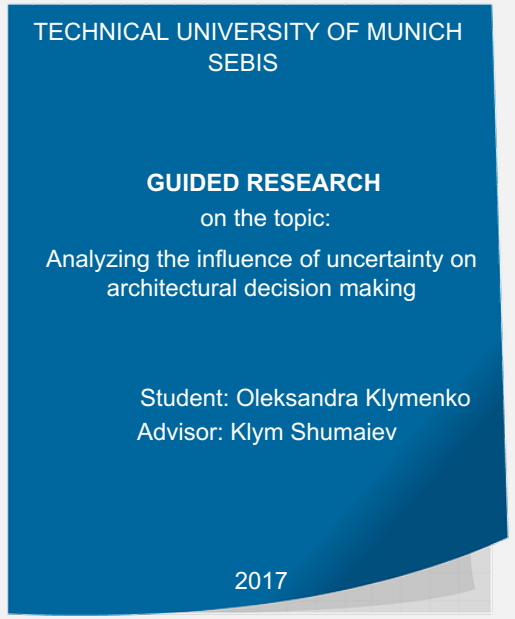
- *Uncertainty density will decrease towards the end of the issue life-cycle*
- *There will be peak(s) in the middle when new solution approaches will be discussed*

Research Process

Step 4:
Interpret and validate
the results



Step 5:
Write a scientific report
on the research



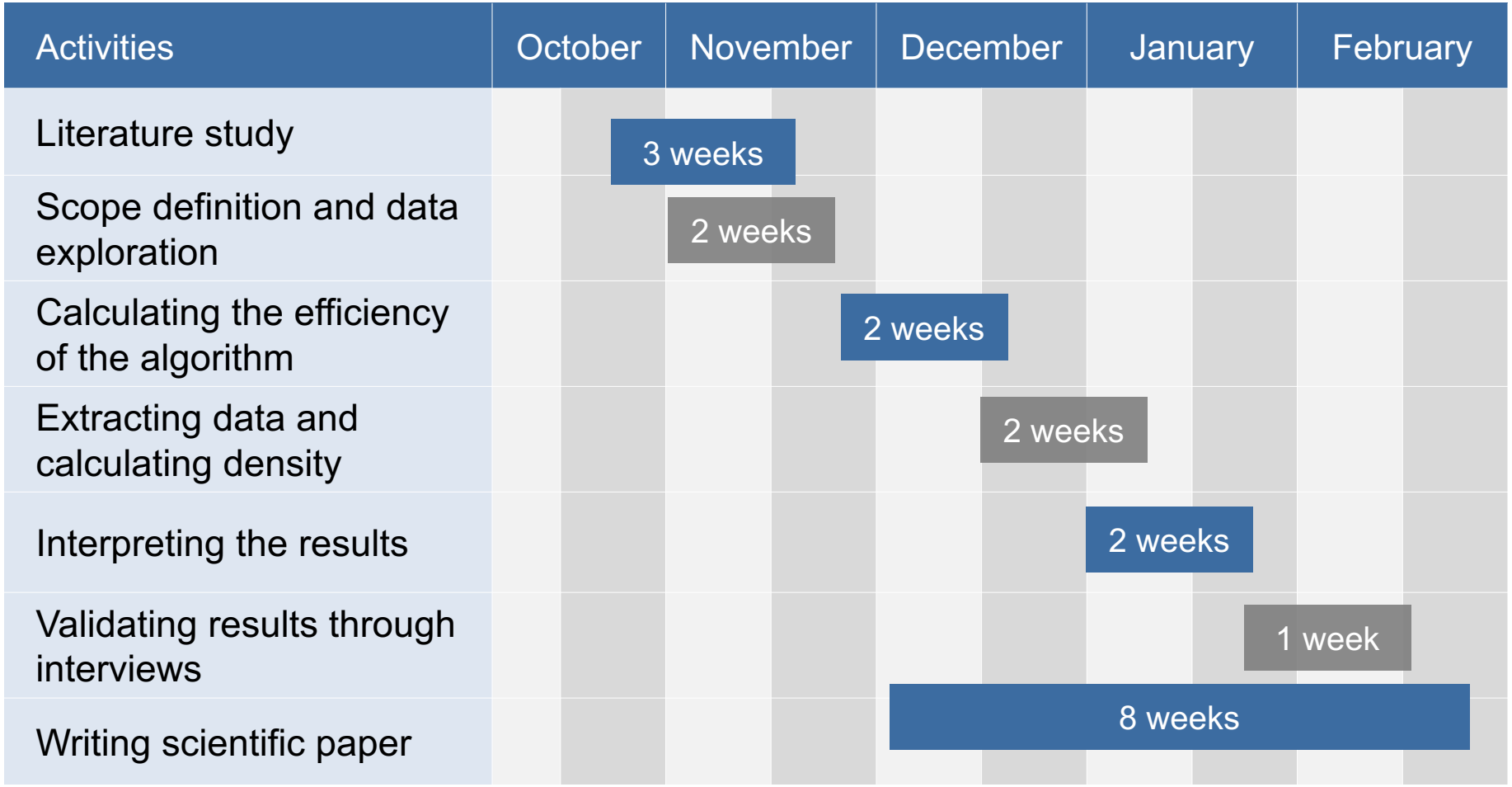
Agenda

1. Motivation
2. Related Work
3. Research Questions
4. Research Process
5. Timeline
6. References

Timeline

Start date: 20.10.2017

End date: 20.02.2018



Agenda

1. Motivation
2. Related Work
3. Research Questions
4. Research Process
5. Timeline
6. References

References

1. Shradhanand, A. Kaur, S. Jain, “Use of fuzzy logic in software development”.
2. M. Bhat , K. Shumaiev , A. Biesdorf , U. Hohenstein , F. Matthes, “Automatic Extraction of Design Decisions from Issue Management Systems: A Machine Learning Based Approach”.
3. K. Shumaiev, M. Bhat, “Automatic uncertainty detection in software architecture documentation”.
4. V. Vincze, “Uncertainty Detection in Natural Language Texts.”
5. G. Szarvas, V. Vincze, R. Farkas et al., ”Cross-Genre and Cross-Domain Detection of Semantic Uncertainty.
6. E. Letier, D. Stefan, E.T.Barr, “Uncertainty, Risk, and Information Value in Software Requirements and Architecture”
7. D. Garlan, “Software Engineering in an Uncertain World.”
8. R. Capilla et al., “10 years of software architecture knowledge management: Practice and future.”
9. C. Miesbauer, R. Weinreich, “Classication of design decisions – an expert survey in practice.”
10. J.S. van der Ven, J. Bosch, “Making the right decision: Supporting architects with design decision data.”
11. C. Yang, P.Liang, P. Avgeriou et al., “Architectural Assumptions and Their Management in Industry – An Exploratory Study.”
12. C. Yang, P. Liang, P. Avgeriou, “Assumptions and their management in software development: A systematic mapping study.”

M.Sc.
Oleksandra Klymenko

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

alexandra.klymenko@tum.de
wwwmatthes.in.tum.de

Backup Slides

System Architecture Decisions

- Issues from two project are analyzed.