TDD89

Introduction

Ola Leifler, 2017-10-31

LINKÖPINGS UNIVERSITET

Part I

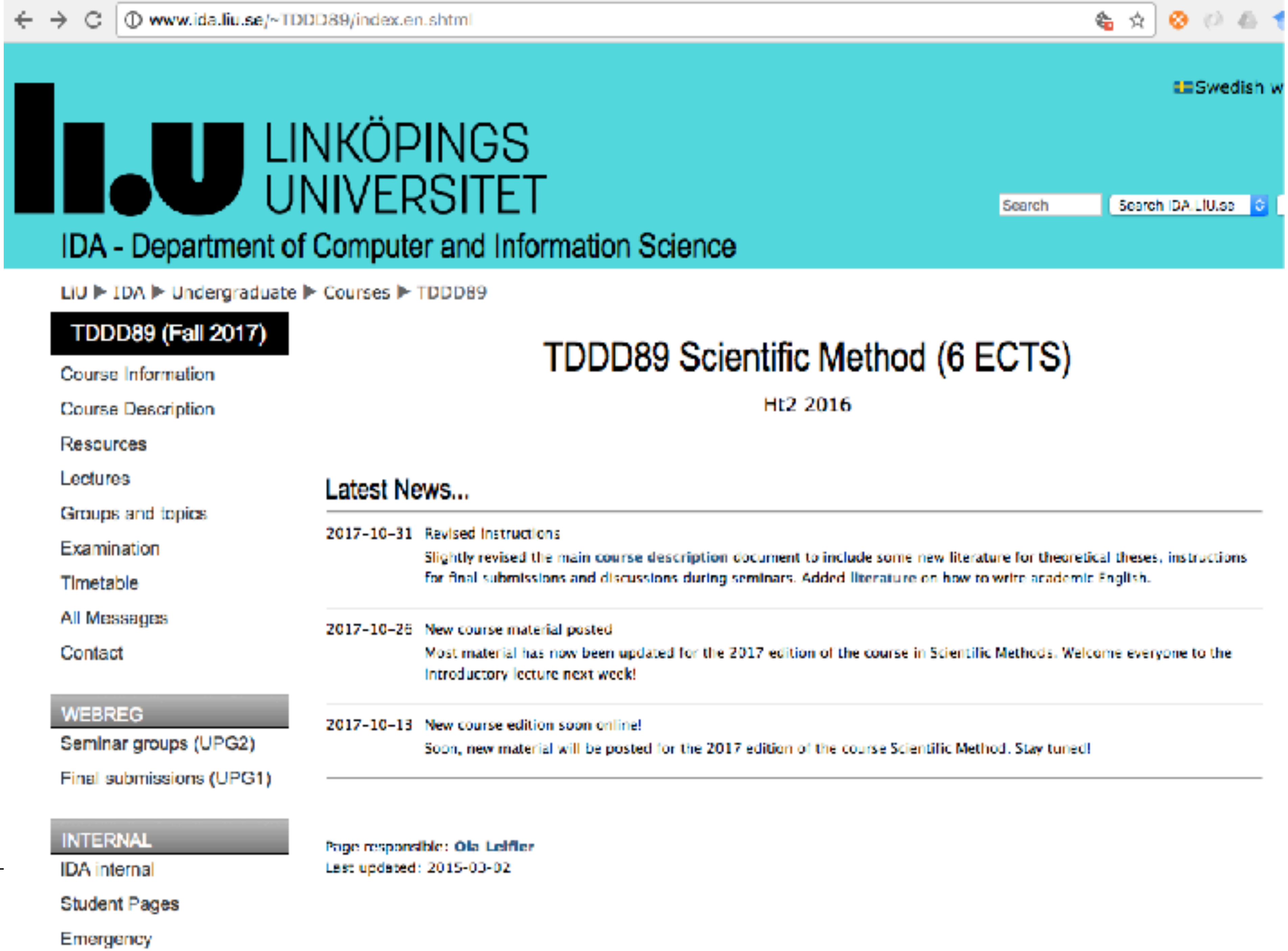
- Course format
- Activities
- Examination

Part II

- Starting your thesis project:
  - What is a *good* thesis project?
  - How do you start?

Part I

Course web



TDDD89 Scientific Method (6 ECTS)

Ht2 2016

Latest News...

2017-10-31 Revised Instructions

Slightly revised the main course description document to include some new literature for theoretical theses, instructions for final submissions and discussions during seminars. Added literature on how to write academic English.

2017-10-26 New course material posted

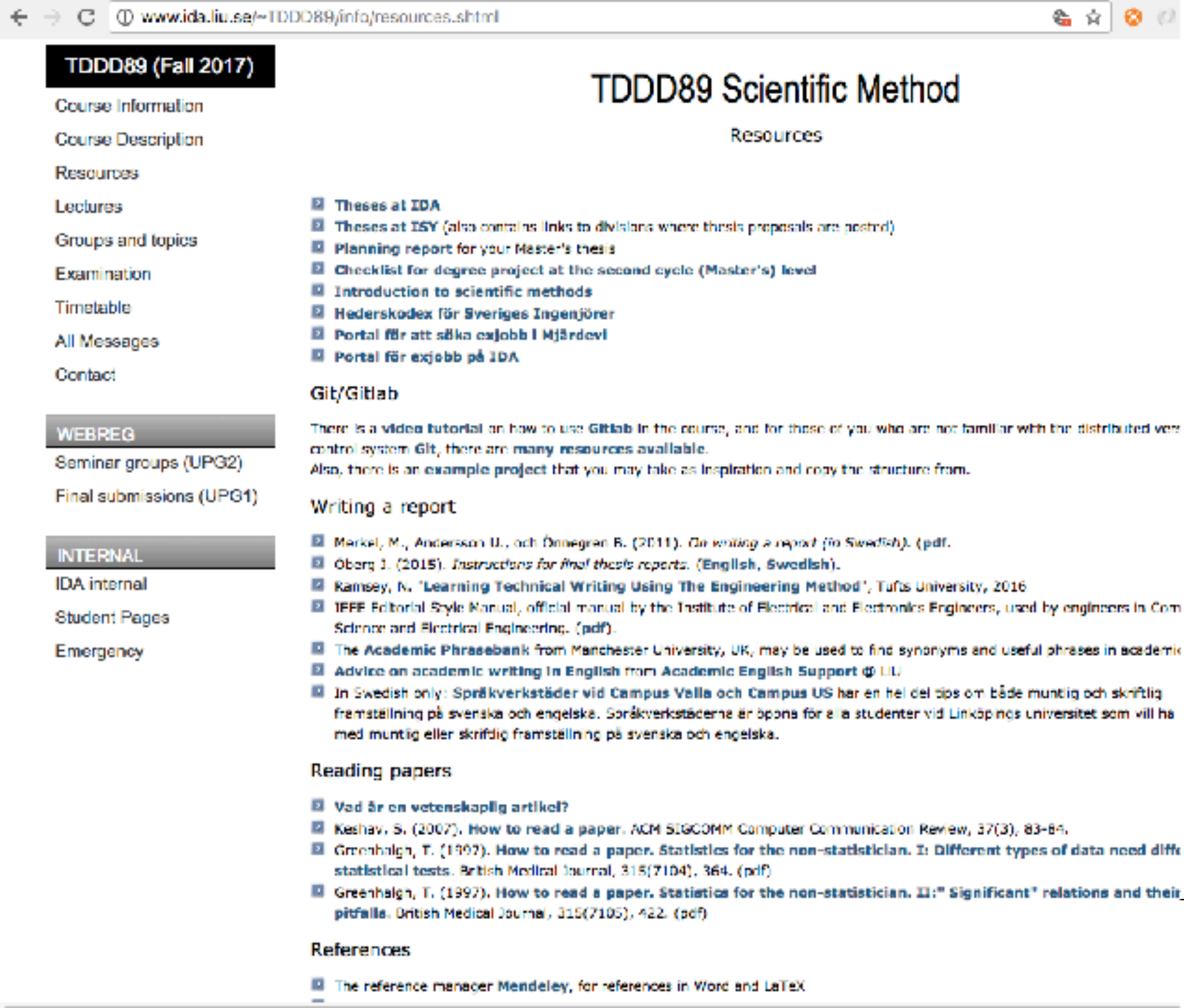
Most material has now been updated for the 2017 edition of the course in Scientific Methods. Welcome everyone to the introductory lecture next week!

2017-10-13 New course edition soon online!

Soon, new material will be posted for the 2017 edition of the course Scientific Method. Stay tuned!

Page responsible: Ola Leifler
Last updated: 2015-03-02

Resources



TDDD89 (Fall 2017)

- Course Information
- Course Description
- Resources
- Lectures
- Groups and topics
- Examination
- Timetable
- All Messages
- Contact

WEBREG

- Seminar groups (UPG2)
- Final submissions (UPG1)

INTERNAL

- IDA internal
- Student Pages
- Emergency

TDDD89 Scientific Method

Resources

- **Theses at IDA**
- **Theses at ISY** (also contains links to divisions where thesis proposals are posted)
- **Planning report for your Master's thesis**
- **Checklist for degree project at the second cycle (Master's) level**
- **Introduction to scientific methods**
- **Hederskodex för Sveriges Ingenjörer**
- **Portal för att söka exjobb i Mjärdevi**
- **Portal för exjobb på IDA**

Git/Gitlab

There is a **video tutorial** on how to use **Gitlab** in the course, and for those of you who are not familiar with the distributed ver
control system **Git**, there are **many resources available**.
Also, there is an **example project** that you may take as inspiration and copy the structure from.

Writing a report

- Merkel, M., Andersson U., och Önnegren B. (2011). *On writing a report (in Swedish)*. (**pdf**.
- Öberg J. (2015). *Instructions for final thesis reports*. (**English**, **Swedish**).
- Ramsey, N. '**Learning Technical Writing Using The Engineering Method**', Tufts University, 2016
- IEEE Editorial Style Manual, official manual by the Institute of Electrical and Electronics Engineers, used by engineers in Com Science and Electrical Engineering. (**pdf**).
- The **Academic Phrasebank** from Manchester University, UK, may be used to find synonyms and useful phrases in academic
- **Advice on academic writing in English** from **Academic English Support** @ LiU.
- In Swedish only: **Språkverkstäder vid Campus Valla och Campus US** har en hel del tips om både muntlig och skriftlig framställning på svenska och engelska. Språkverkstäderna är öppna för alla studenter vid Linköpings universitet som vill ha med muntlig eller skriftlig framställning på svenska och engelska.

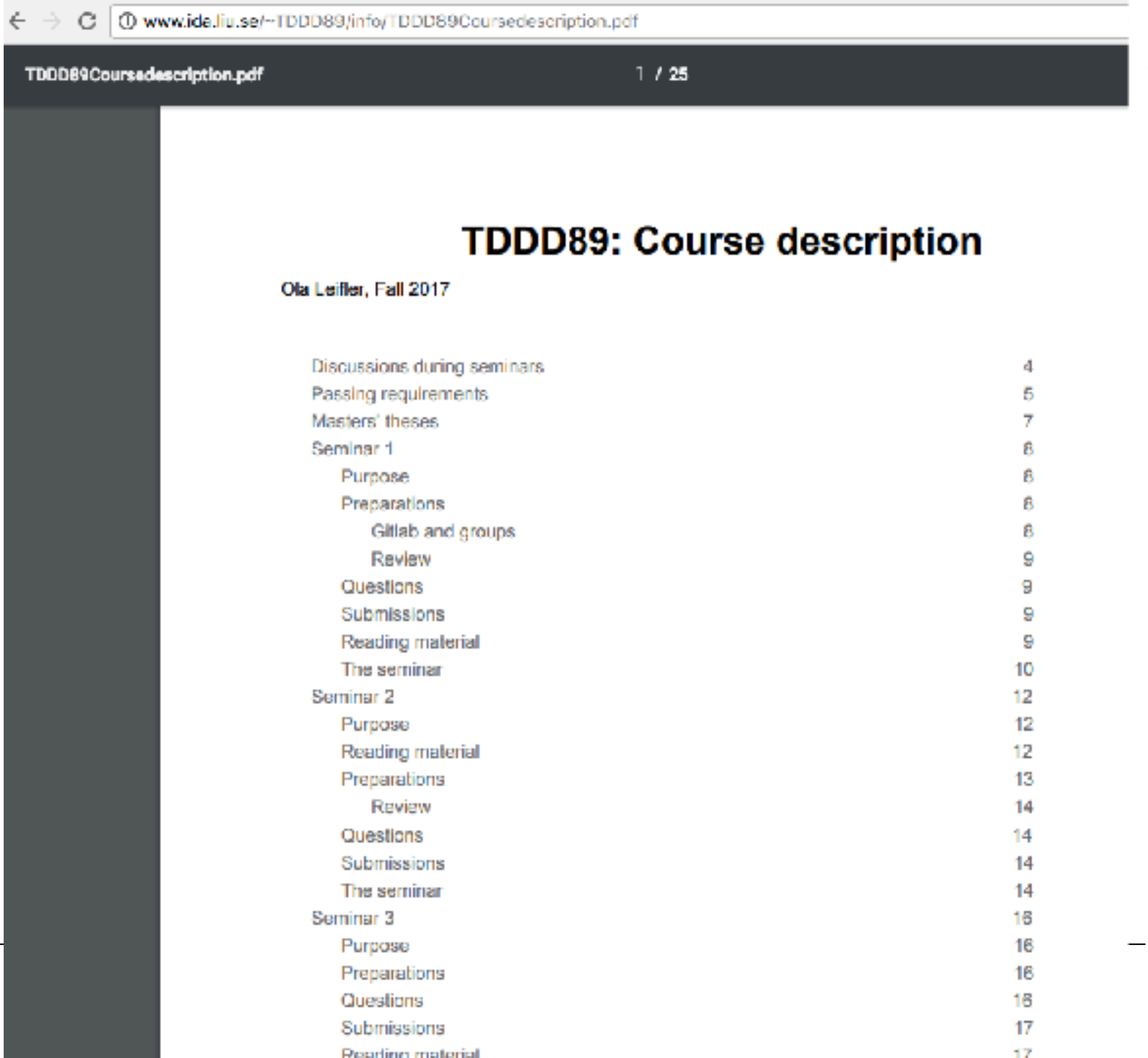
Reading papers

- **Vad är en vetenskaplig artikel?**
- Keshav, S. (2007). **How to read a paper**. ACM SIGCOMM Computer Communication Review, 37(3), 83-84.
- Greenhalgh, T. (1997). **How to read a paper. Statistics for the non-statistician. I: Different types of data need diffe statistical tests**. British Medical Journal, 315(7104), 364. (**pdf**)
- Greenhalgh, T. (1997). **How to read a paper. Statistics for the non-statistician. II:" Significant" relations and their pitfalls**. British Medical Journal, 315(7105), 422. (**pdf**)

References

- The reference manager **Mendeley**, for references in Word and LaTeX

Course description


www.ida.liu.se/~TDDD89/info/TDDD89Coursedescription.pdf
TDDD89Coursedescription.pdf
1 / 25

TDDD89: Course description

Ola Leifler, Fall 2017

Course format & staff

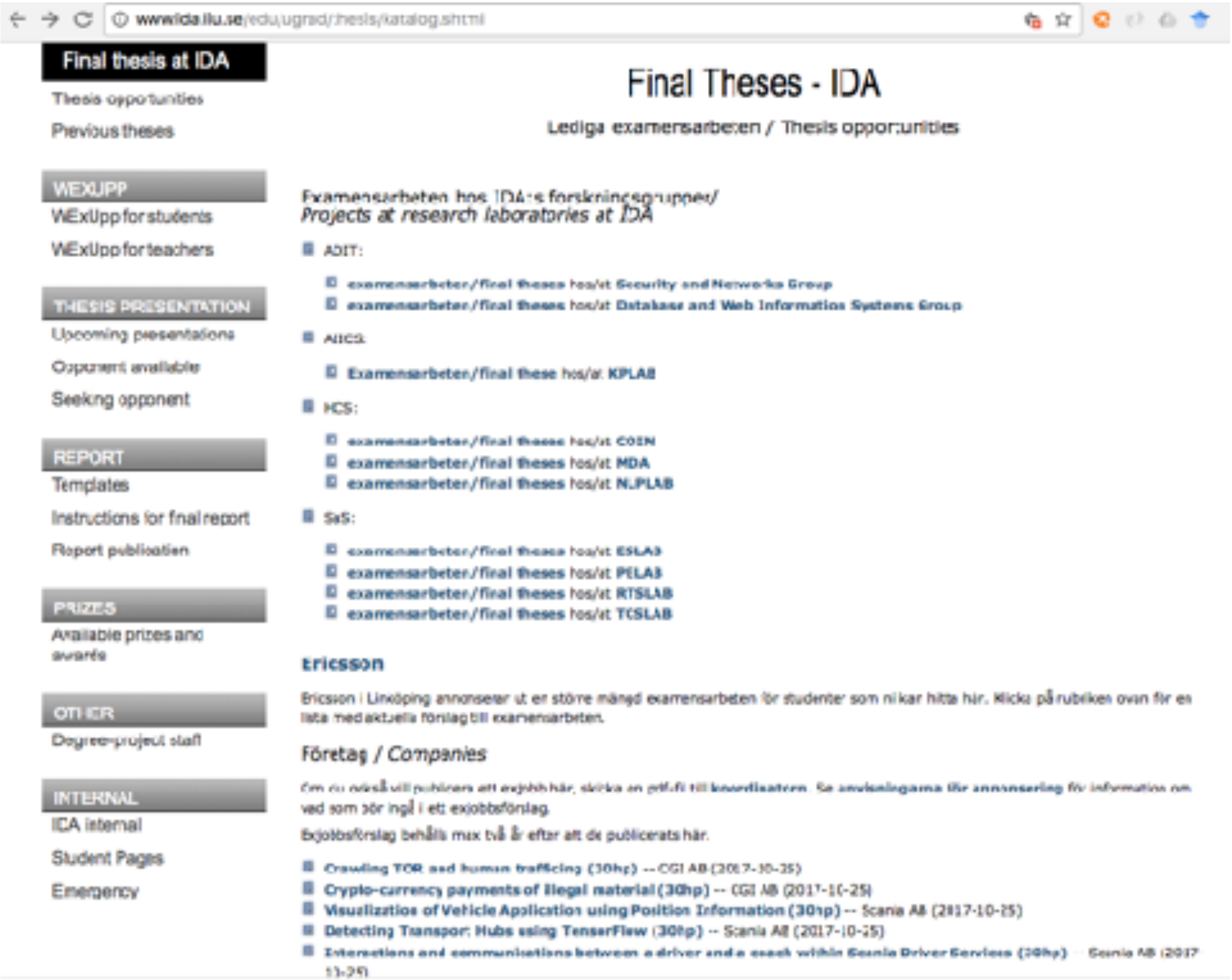
- 5 x 2h lectures
  - Lectures on academic English by Pamela Vang
- 6 x 2h seminars
  - theme-specific groups
  - Ola Leifler, Aseel Berglund, Azeem Ahmad & Oscar Gustafsson

Seminar	Read	Write
1	Introduction, Background	Thesis plan
2		Introduction
3	Theory, Method	
4		Theory
5	Results, Discussion, Conclusion	
6		Method

Lectures

1. Introduction
2. Information search and evaluation
3. Introduction to academic writing in English
4. Scientific methods
5. Feedback on English writing and grammar

Selecting a thesis topic



Final thesis at IDA

Thesis opportunities

Previous theses

WEXUPP

WExUpp for students

WExUpp for teachers

THESIS PRESENTATION

Upcoming presentations

Opponent available

Seeking opponent

REPORT

Templates

Instructions for final report

Report publication

PRIZES

Available prizes and awards

OTHER

Degree-project staff

INTERNAL

IDA internal

Student Pages

Emergency

Final Theses - IDA

Lediga examensarbeten / Thesis opportunities

Examensarbeten hos IDA:s forskningsgrupper/
Projects at research laboratories at IDA

- ADIT:
  - examensarbeten/final theses hos/at Security and Networks Group
  - examensarbeten/final theses hos/at Database and Web Information Systems Group
- AIICS:
  - Examensarbeten/final these hos/at KPLAB
- HCS:
  - examensarbeten/final theses hos/at COIN
  - examensarbeten/final theses hos/at MDA
  - examensarbeten/final theses hos/at NLPLAB
- SaS:
  - examensarbeten/final theses hos/at ESLAB
  - examensarbeten/final theses hos/at PELAB
  - examensarbeten/final theses hos/at RTSLAB
  - examensarbeten/final theses hos/at TCSLAB

Ericsson

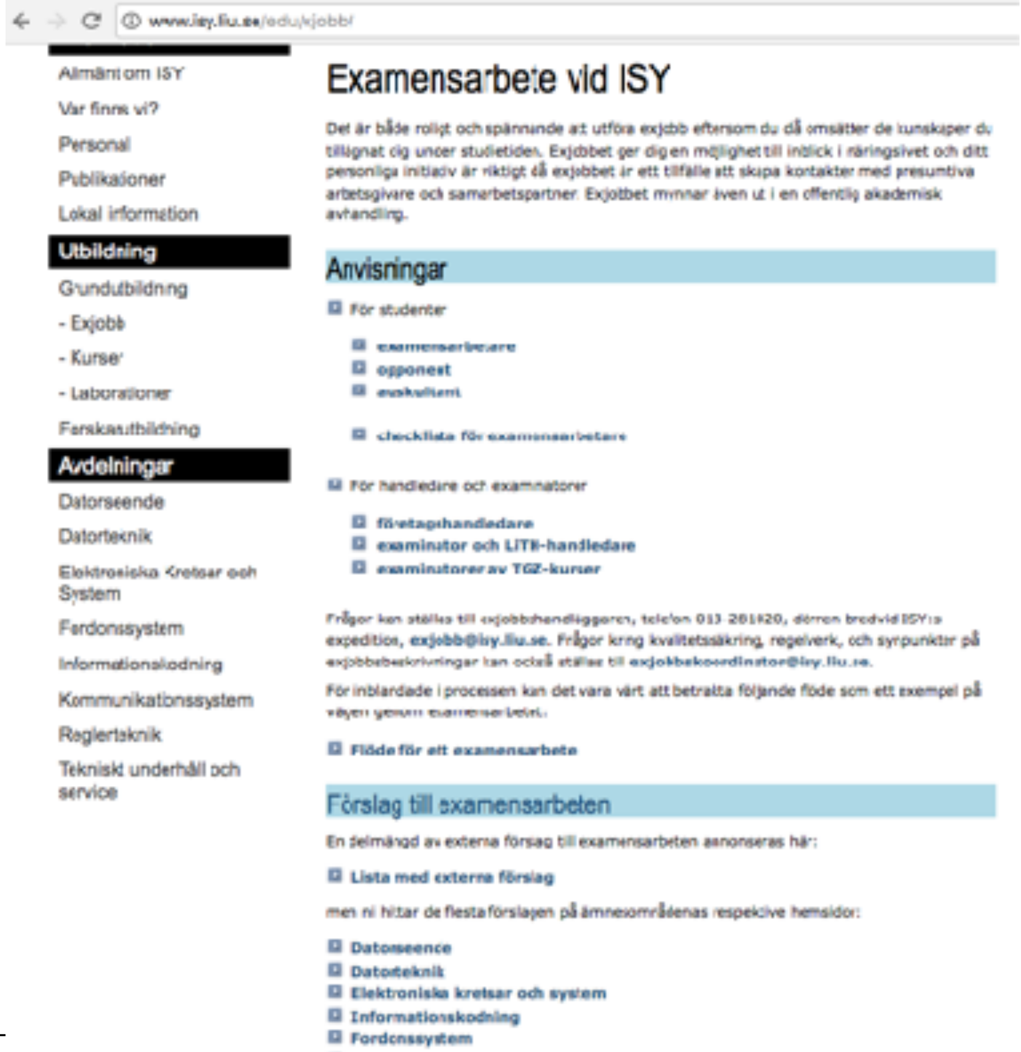
Ericsson i Linköping annonserar ut en större mängd examensarbeten för studenter som ni kan hitta här. Klicka på rubriken ovan för en lista med aktuella förslag till examensarbeten.

Företag / *Companies*

Om ni också vill publicera ett exjobb här, skicka en pdf-fil till koordinatorn. Se anvisningarna för annonsering för information om vad som bör ingå i ett exjobbsförslag.

Exjobbsförslag behålls max två år efter att de publicerats här.

- Crawling TOR and human trafficing (30hp) -- CGI AB (2017-10-25)
- Crypto-currency payments of illegal material (30hp) -- CGI AB (2017-10-25)
- Visualization of Vehicle Application using Position Information (30hp) -- Scania AB (2017-10-25)
- Detecting Transport Hubs using TensorFlow (30hp) -- Scania AB (2017-10-25)
- Interactions and communications between a driver and a coach within Scania Driver Services (30hp) -- Scania AB (2017-10-25)



Allmänt om ISY

Var finns vi?

Personal

Publikationer

Lokal information

Utbildning

Grundutbildning

- Exjobb

- Kurser

- Laborationer

Forskarutbildning

Avdelningar

Datorseende

Datorteknik

Elektroniska Kretsar och System

Fordonssystem

Informationskodning

Kommunikationssystem

Reglerteknik

Tekniskt underhåll och service

Examensarbete vid ISY

Det är både roligt och spännande att utföra exjobb eftersom du då omsätter de kunskaper du tillägnat dig under studietiden. Exjobbet ger dig en möjlighet till inblick i näringslivet och ditt personliga initiativ är viktigt då exjobbet är ett tillfälle att skapa kontakter med presumtiva arbetsgivare och samarbetspartner. Exjobbet mynnar även ut i en offentlig akademisk avhandling.

Anvisningar

- För studenter
  - examensarbetare
  - opponent
  - auskultant
  - checklista för examensarbetare
- För handledare och examinatorer
  - företagshandledare
  - examinator och LiTH-handledare
  - examinatorer av TGZ-kurser

Frågor kan ställas till exjobbshandläggaren, telefon 013-281420, dörren bredvid ISY:s expedition, exjobb@isy.liu.se. Frågor kring kvalitetssäkring, regelverk, och synpunkter på exjobbsbeskrivningar kan också ställas till exjobbskoordinator@isy.liu.se.

För inblandade i processen kan det vara värt att betrakta följande flöde som ett exempel på vägen genom examensarbetet.

- Flöde för ett examensarbete

Förslag till examensarbeten

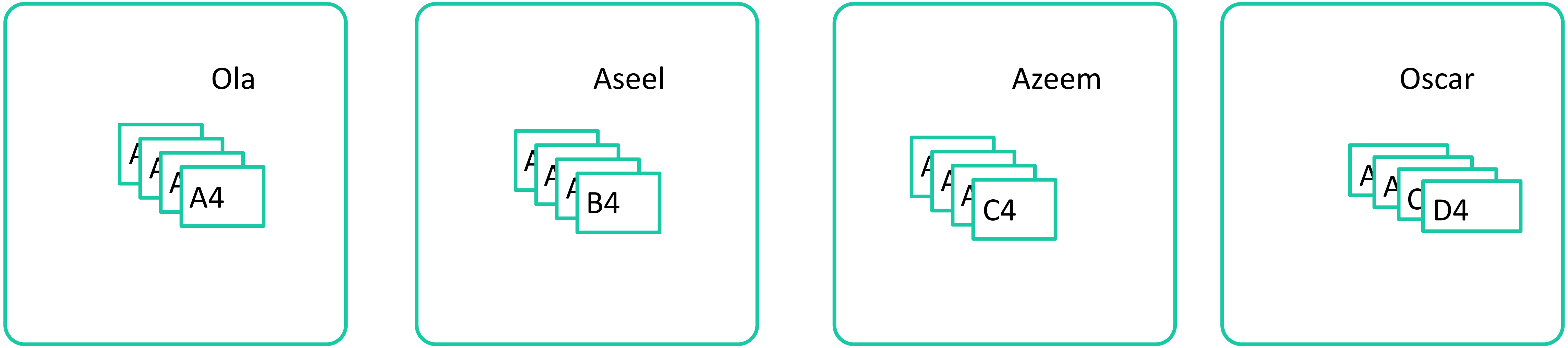
En delmängd av externa förslag till examensarbeten annonseras här:

- Lista med externa förslag

men ni hittar de flesta förslagen på ämnesområdenas respektive hemsidor:

- Datorseende
- Datorteknik
- Elektroniska kretsar och system
- Informationskodning
- Fordonssystem

Seminar groups



Ola

A A A A4

Aseel

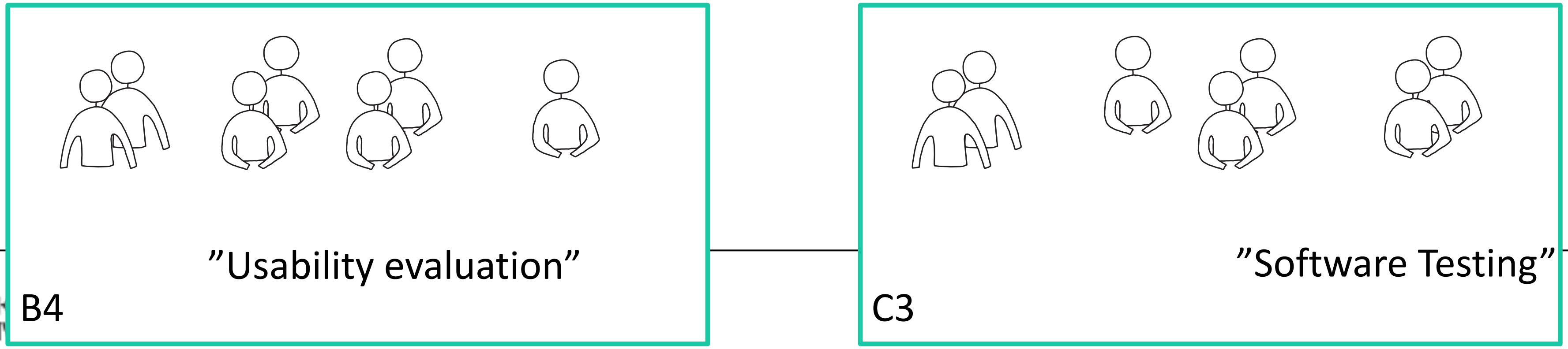
A A A B4

Azeem

A A A C4

Oscar

A A C D4



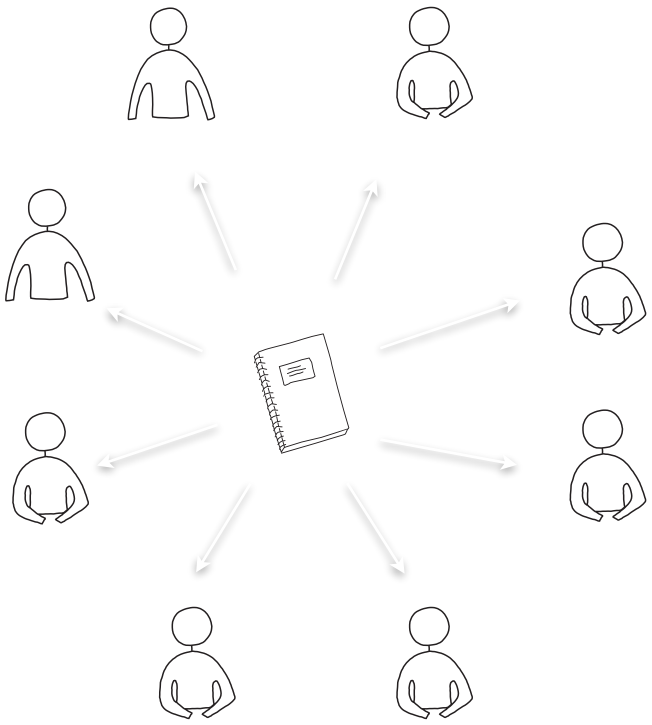
B4

”Usability evaluation”

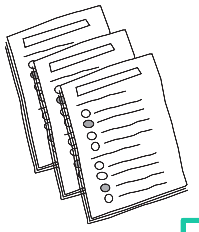
C3

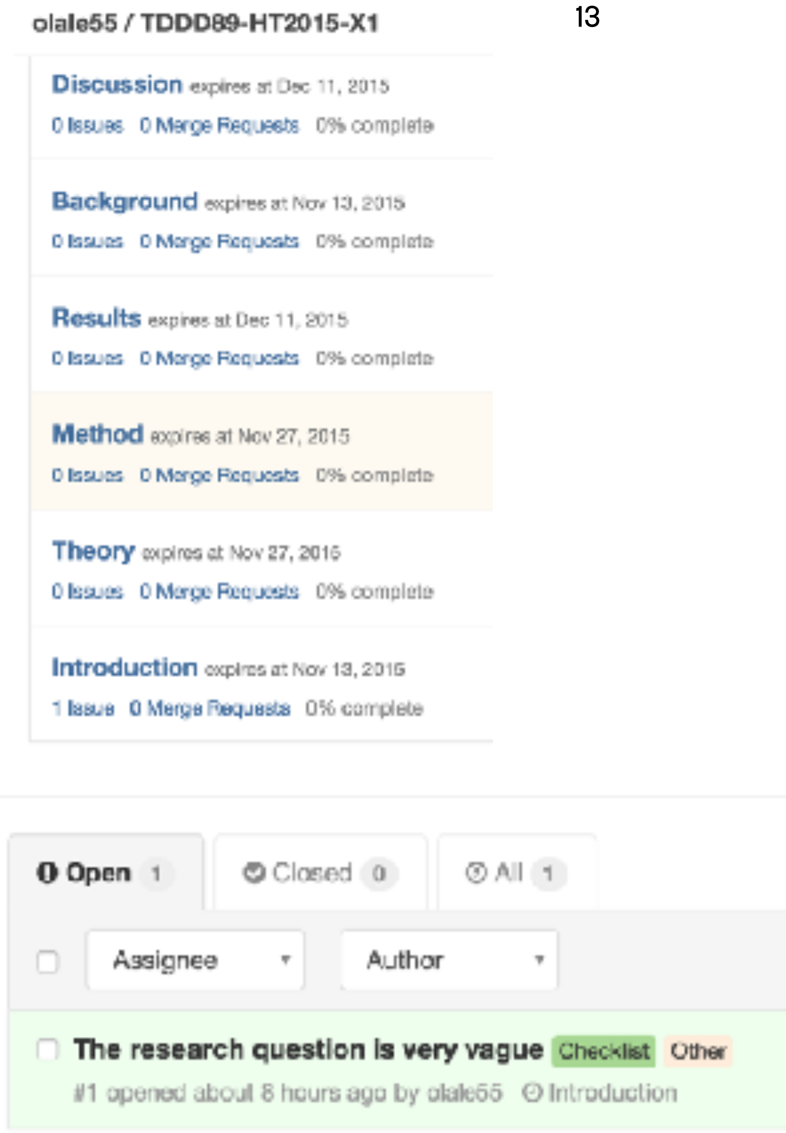
”Software Testing”

Seminar 1,3,5



Are the research questions in the published thesis easy to find, clear and with a reasonable scope, as required by the *instructions for final thesis reports* ?





olale55 / TDDD89-HT2015-X1

Discussion expires at Dec 11, 2015
0 Issues 0 Merge Requests 0% complete

Background expires at Nov 13, 2015
0 Issues 0 Merge Requests 0% complete

Results expires at Dec 11, 2015
0 Issues 0 Merge Requests 0% complete

Method expires at Nov 27, 2015
0 Issues 0 Merge Requests 0% complete

Theory expires at Nov 27, 2015
0 Issues 0 Merge Requests 0% complete

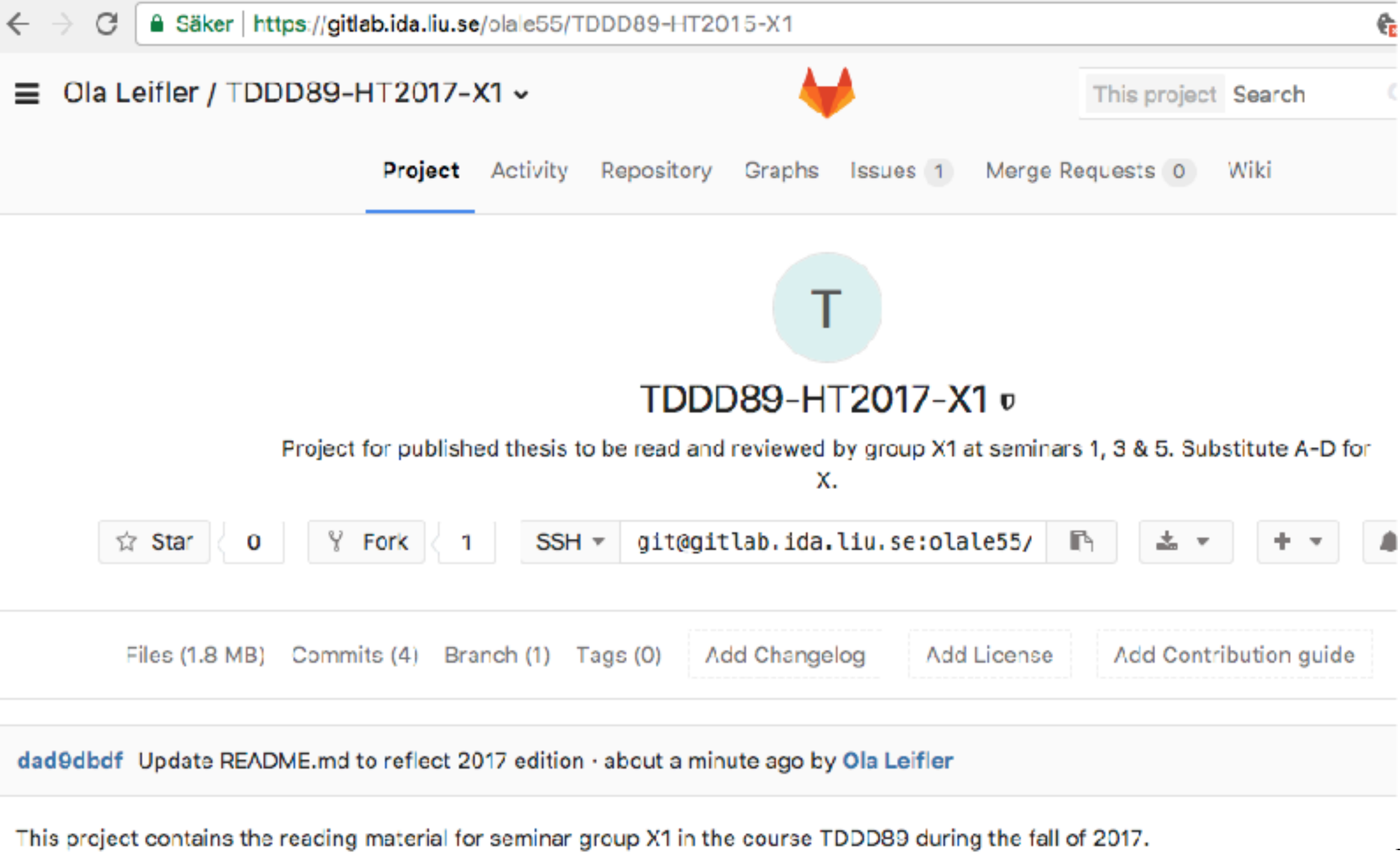
Introduction expires at Nov 13, 2015
1 Issue 0 Merge Requests 0% complete

□ **The research question is very vague** Checklist Other
#1 opened about 8 hours ago by olale55 Introduction

1

2

3



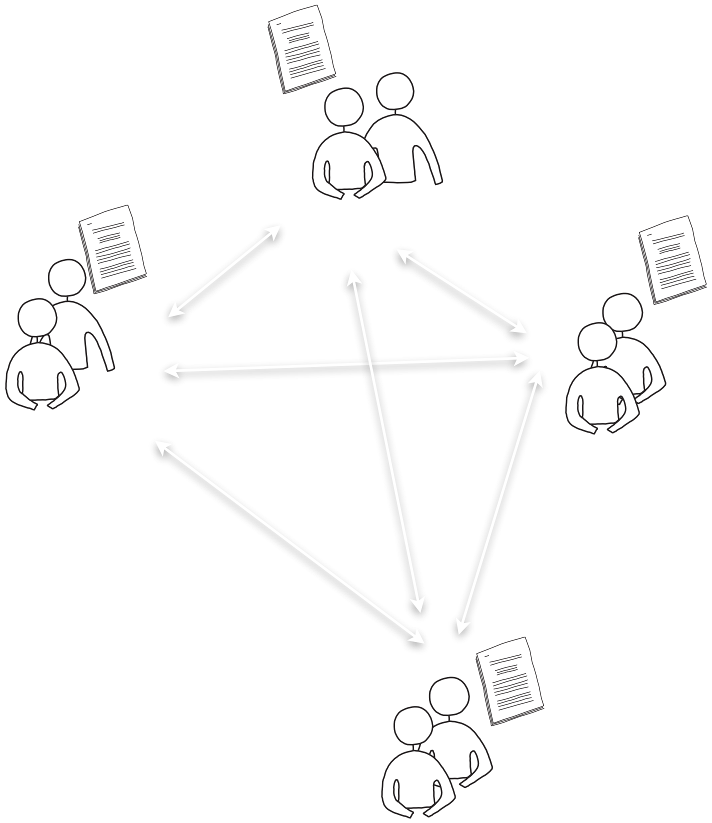
TDDD89-HT2017-X1

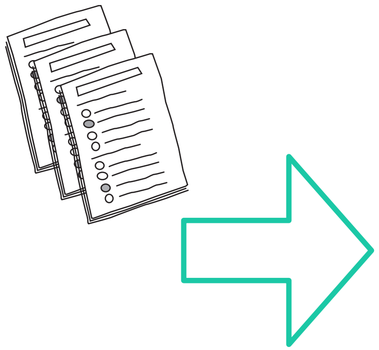
Project for published thesis to be read and reviewed by group X1 at seminars 1, 3 & 5. Substitute A-D for X.

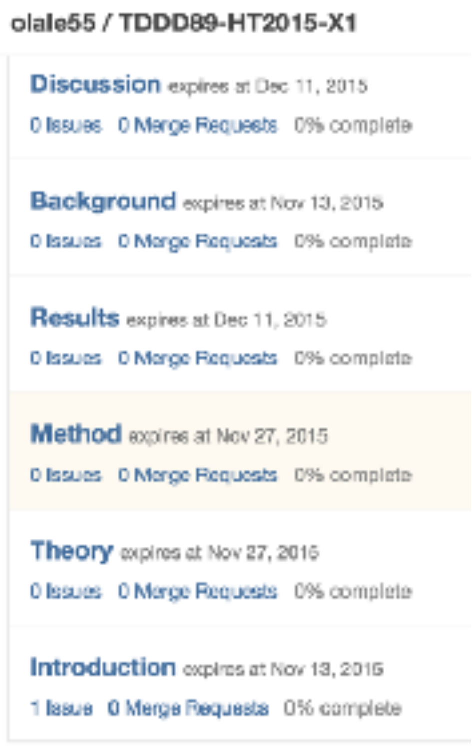
dad9dbdf Update README.md to reflect 2017 edition · about a minute ago by Ola Leifler

This project contains the reading material for seminar group X1 in the course TDDD89 during the fall of 2017.

Seminar 2,4,6







olale55 / TDDD89-HT2015-X1

Discussion expires at Dec 11, 2015
0 Issues 0 Merge Requests 0% complete

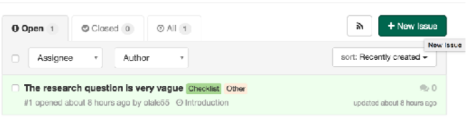
Background expires at Nov 13, 2015
0 Issues 0 Merge Requests 0% complete

Results expires at Dec 11, 2015
0 Issues 0 Merge Requests 0% complete

Method expires at Nov 27, 2015
0 Issues 0 Merge Requests 0% complete

Theory expires at Nov 27, 2015
0 Issues 0 Merge Requests 0% complete

Introduction expires at Nov 13, 2015
1 Issue 0 Merge Requests 0% complete



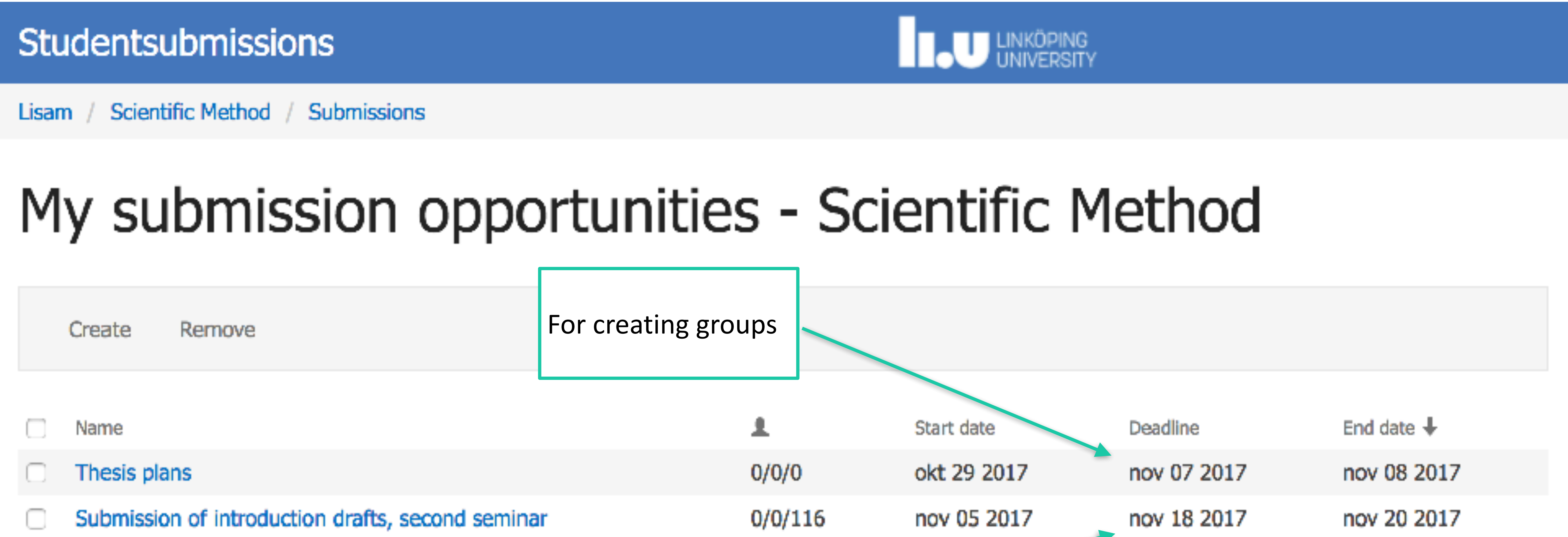
□ **The research question is very vague** Checklist Other 0
#1 opened about 8 hours ago by olale55 Introduction updated about 8 hours ago

1

2

3

LISAM



Studentsubmissions

Lisam / Scientific Method / Submissions

My submission opportunities - Scientific Method

Create   Remove

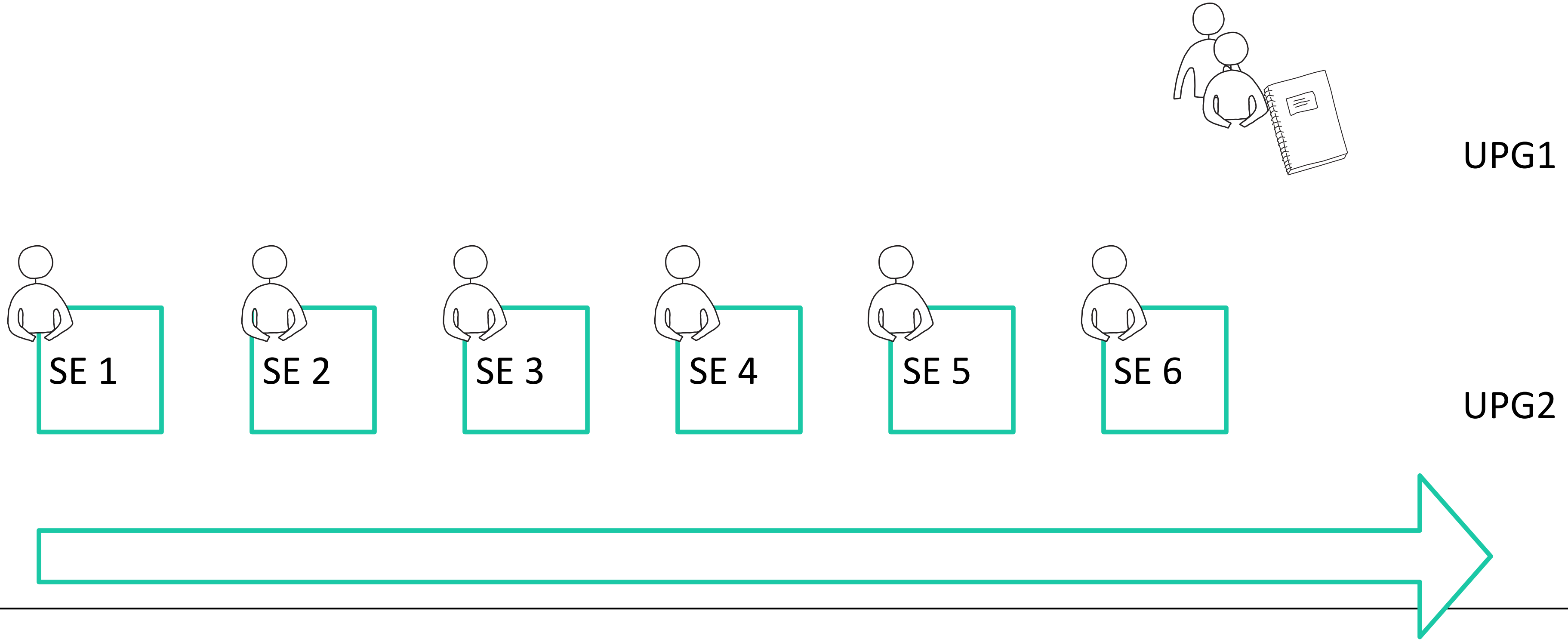
□	Name	👤	Start date	Deadline	End date ↓
□	Thesis plans	0/0/0	okt 29 2017	nov 07 2017	nov 08 2017
□	Submission of introduction drafts, second seminar	0/0/116	nov 05 2017	nov 18 2017	nov 20 2017

For creating groups

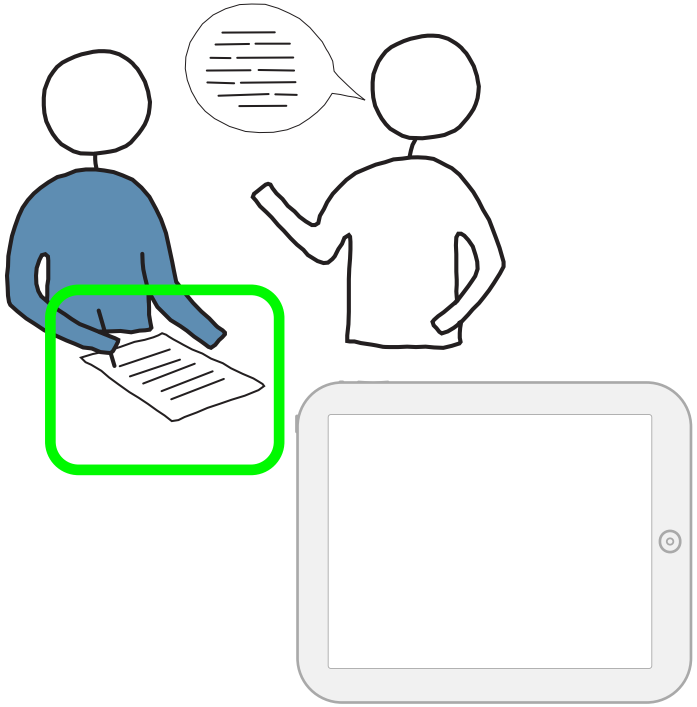
For review by Pamela Vang

Examination

- UPG1: First three chapter of your thesis report at the end of the course
- UPG2: Preparation and participation in seminars during the course



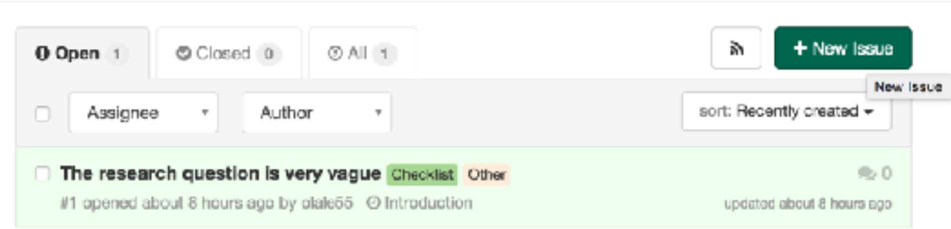
Seminars



Final submissions

Review, rewrite

SE 6



□ The research question is very vague Checklist Other
#1 opened about 8 hours ago by olale55 Introduction
0
updated about 8 hours ago

URKUND

End of HT2

January 6, 2018

Workload distribution

Seminar	work %
1	10
2	20
3	15
4	20
5	15
6	20

160h total:
Plan your time,
look ahead and
read the course description document thoroughly

Part 2

What is a great thesis?

Thesis = project results + written presentation

- A working, interesting application with proven and general value
  - A well-described application
  - of general interest
  - and with a clear description of ”proven” and ”value”

What is a great thesis?

Thesis = project results + written presentation

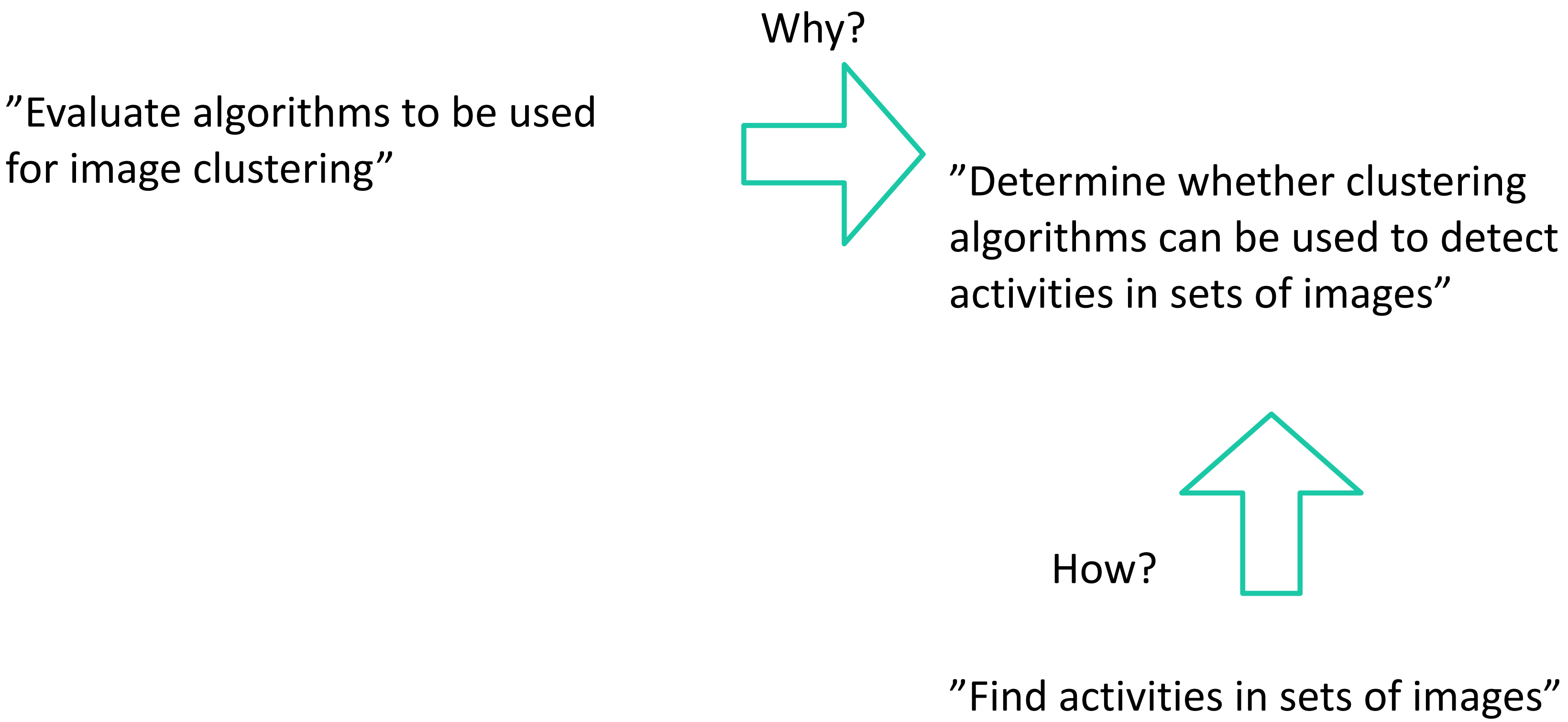
- An evaluation with general and interesting results
  - that others can **use**
  - that others will **believe**

What is a great thesis?

Thesis = project results + written presentation

- An authoritative report
  - with a good focus (questions!)
  - and results that answer the questions
  - through a transparent, thorough description of the process

Starting your thesis work



Why?

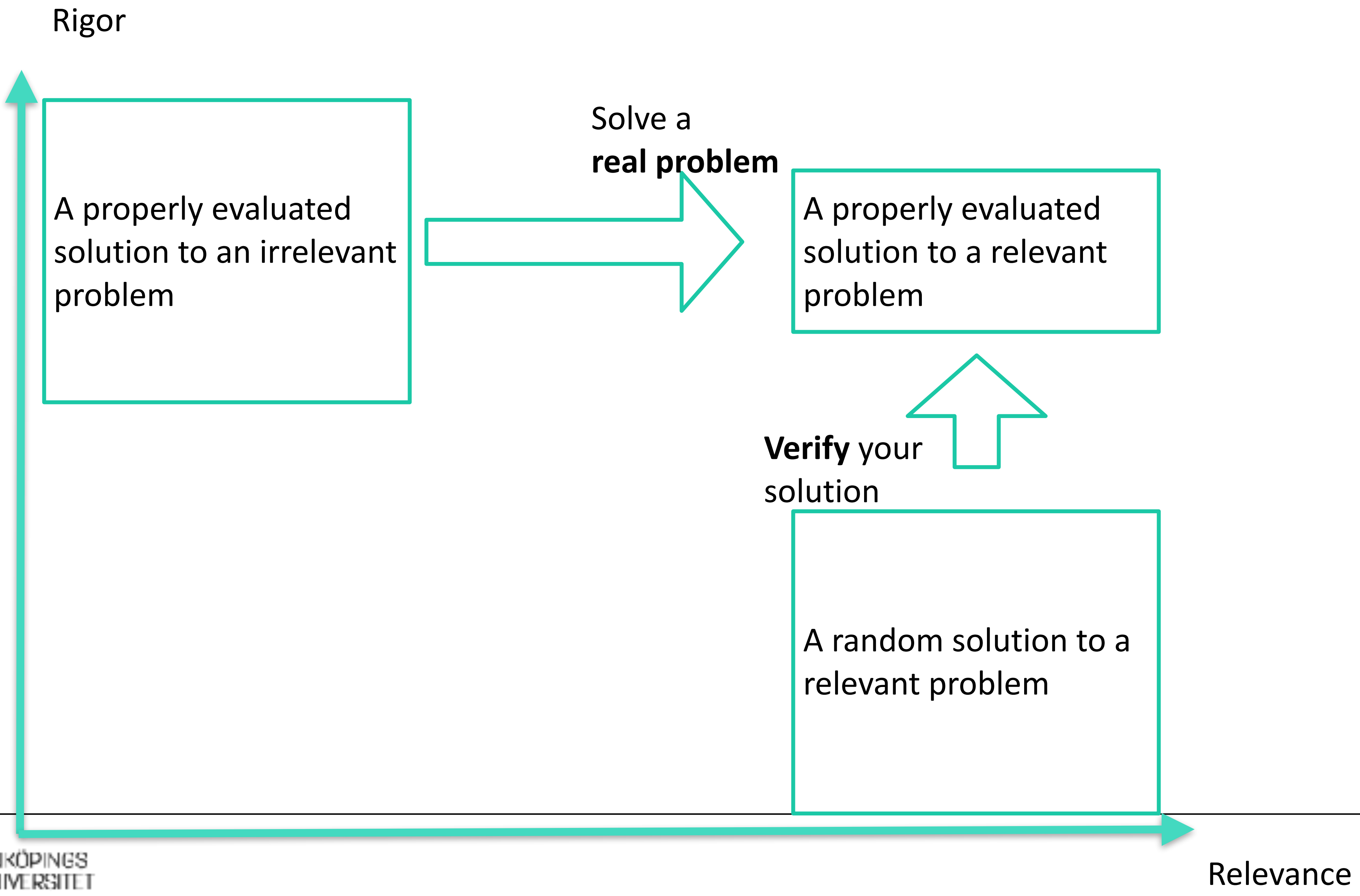
”Evaluate algorithms to be used for image clustering”

”Determine whether clustering algorithms can be used to detect activities in sets of images”

How?

”Find activities in sets of images”

Relevance/Rigor



Rigor

A properly evaluated solution to an irrelevant problem

Solve a **real problem**

A properly evaluated solution to a relevant problem

Verify your solution

A random solution to a relevant problem

Relevance

Thesis outline

Why should even I read this report?

What have you studied here?

What does this relate to?

Can I trust you?

What is built?

What have you found?

How can we explain the results?

How can I use these results in my work?

Thesis outline

Abstract

Research Questions

Theory, Background

Method

Implementation

Results

Discussion

Conclusion

What is a good research question in Computer Science?

Question type	Example question	Type of answer
Means of development	What is the most efficient software testing method for small teams?	Procedure
General method for analysis	How can you verify conformance to realtime constraints in a multi-threaded embedded system?	Analysis method
Specific evaluation of systems	When is PhoneGap more economical to use than NativeGoo for cross-platform mobile development?	Empirical predictive model based on data
Generalization or characterisation	Given recent results in tuning deep neural networks, which meta-heuristics should be used for exploring the parameter space?	Classification, taxonomy
Feasibility study	Can you automate a car?	Specific implementation, empirical observations

A Great Thesis

I - The Problem

The infamous ”How can you …?”, ”Is it possible to …?”

- ”How can you construct a web application?”
- ”Is it possible to construct a mobile application X?”
- ”How can you create a usable website?”

Is the answer any of the following?

By writing what has already been written many times before

Yes, there is no reason to believe it could not be done.

By adhering to published design guideline Y

Then come up with a better question..

Better questions

- Start exploring a field:
  - ”How can fuzz testing be used to find bugs in parallel embedded software?”
- Then, use existing technical approaches and theoretical models:
  - ”Combining dynamic and static analysis of scheduling embedded systems”
  - ”What is the efficiency of AFL at finding timing-related errors in parallel software applications?”

The journey to a better question

Question	Approach	Objection
How can you automate testing?	Applying automatic generation of test cases	You end up with 10^7 test cases, only some of which are necessary..
How can we select test cases?	Applying statistical/ML clustering techniques	Black-box solution, we need traceability
How can we prioritise test cases?	Optimize based on historical records to maximise APFD	No weight given to critical functionality
What do you really want to optimise?	Listen, observe, collect data	Requires an open mindset

2 - the theory

Theory

Validation

”There are seven dimensions of usability”

Does it describe the world?

”NP-hard problems are at least as hard as the hardest NP problems”

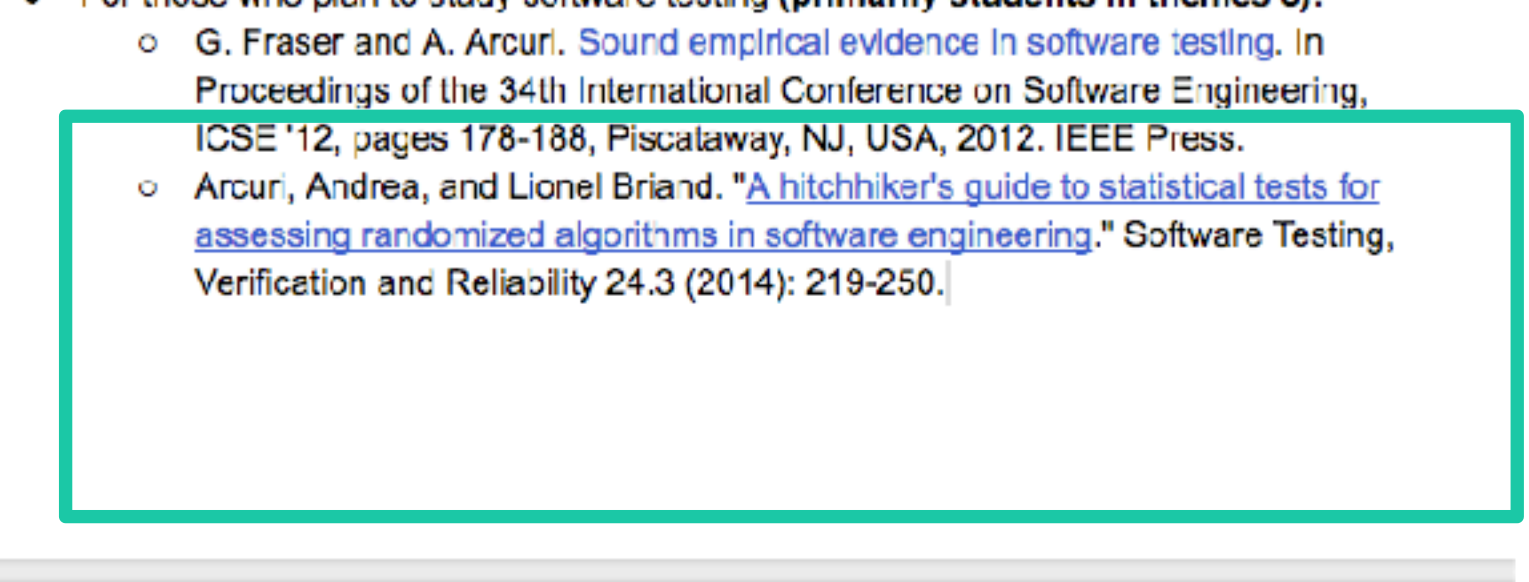
Is it proven?

Characterization of knowledge, accumulated through scientific studies, published in peer-reviewed journals

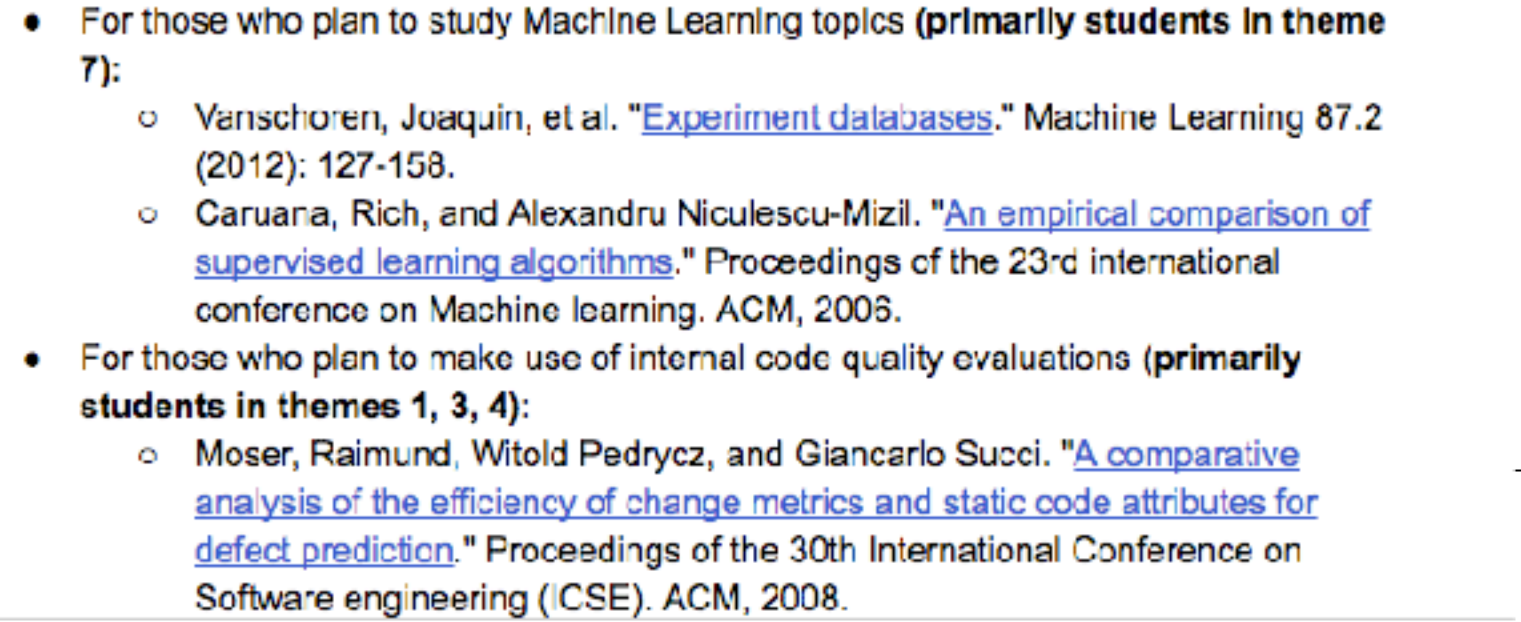
”It is just a theory”

- Empirical observations are based on expectations, informed by theoretical frameworks:
  - When reading **energy consumption** values of a user device in a 4G network, you expect frequent radio transmissions to be costly due to different **power states**
  - When assessing usability, you assume that **learnability** is affected by **the consistency** of an application
- Based on observations, you can **test claims** made by your theory

Strong persuasion skills (aka a Method)



- For those who plan to study software testing **(primarily students in themes 5):**
  - G. Fraser and A. Arcuri. Sound empirical evidence in software testing. In Proceedings of the 34th International Conference on Software Engineering, ICSE '12, pages 178-188, Piscataway, NJ, USA, 2012. IEEE Press.
  - Arcuri, Andrea, and Lionel Briand. "A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering." Software Testing, Verification and Reliability 24.3 (2014): 219-250.



- For those who plan to study Machine Learning topics **(primarily students in theme 7):**
  - Vanschoren, Joaquin, et al. "Experiment databases." Machine Learning 87.2 (2012): 127-158.
  - Caruana, Rich, and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms." Proceedings of the 23rd international conference on Machine learning. ACM, 2006.
- For those who plan to make use of internal code quality evaluations (**primarily students in themes 1, 3, 4)**:
  - Moser, Raimund, Witold Pedrycz, and Giancarlo Succi. "A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction." Proceedings of the 30th International Conference on Software engineering (ICSE). ACM, 2008.

Algorithms	Implementations	Number of runs	Hyper-parameters
nearest centroid classifier	`NearestCentroid`	3	`metric` ∈ {`"l1"`;`"l2"`;`"cosine"`}

naive Bayes classifier (multinomial distribution)	`MultinomialNB`	22	`alpha` ∈ `np.linspace(0,1,11)`
			`fit_prior` ∈ {`True`;`False`}

Linear SVM	`LinearSVC`	20	`C` ∈ `np.logspace(-4, 4, 10)`
			`loss` ∈ {`"squared_hinge"`;`"hinge"`}
			`class_weight="balanced"`

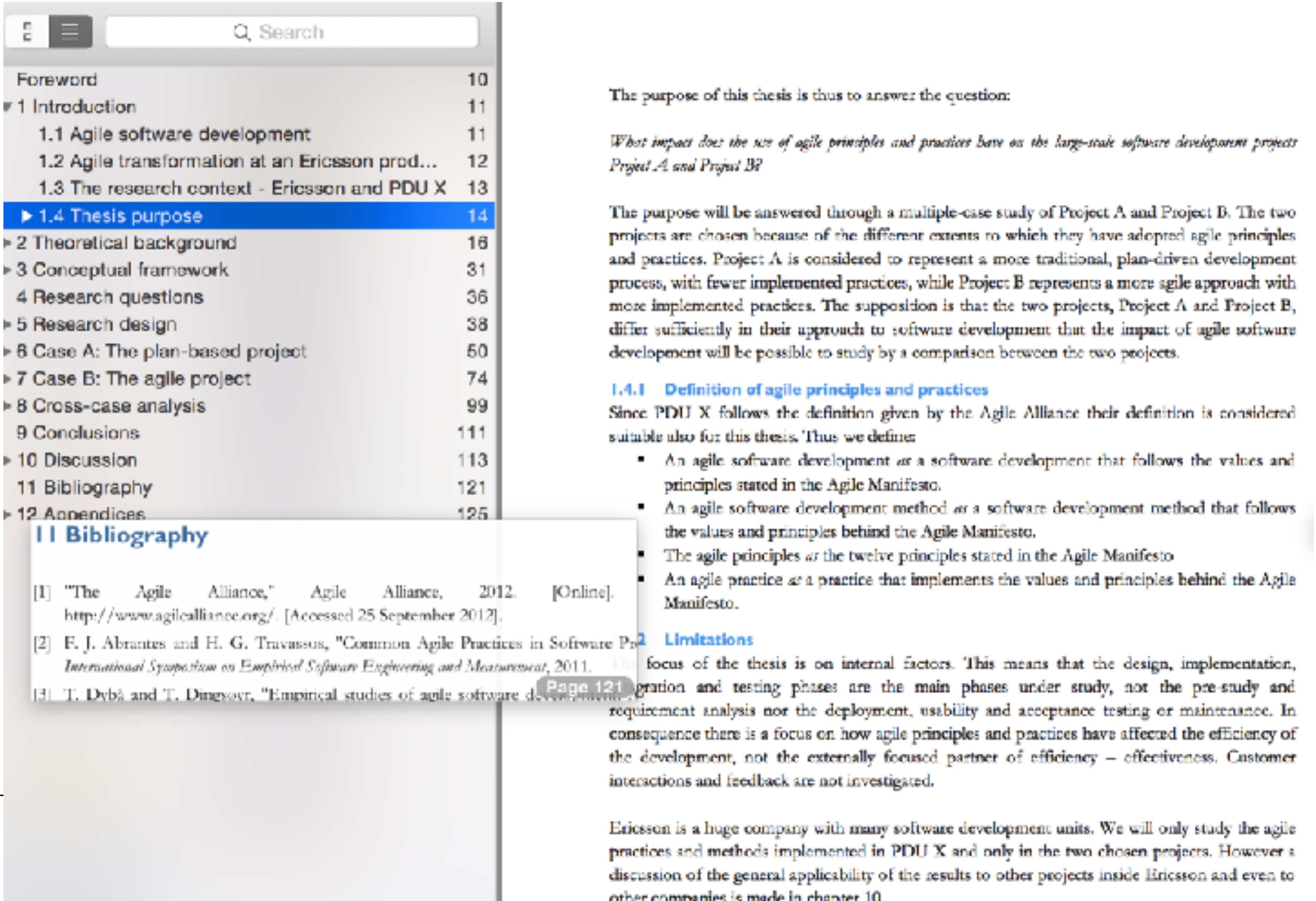
Logistic Regression	`LogisticRegression`	40	`dual=False`
			`C` ∈ `np.logspace(-4, 4, 10)`
			`class_weight="balanced"`
			`solver` ∈ {`"newton-cg"`;`"sag"`;`"lbfgs"`}
			`multi_class="multinomial"`
			`dual=True`
			`C` ∈ `np.logspace(-4, 4, 10)`
			`class_weight="balanced"`
			`solver="liblinear"`
			`multi_class="ovr"`

Perceptron	`Perceptron`	13	`penalty` ∈ {`"l2"`;`"elasticnet"`}
			`alpha` ∈ `10.0**-np.arange(1,7)`
			`class_weight = "balanced"`
			`penalty=None`
			`class_weight="balanced"`

Stochastic gradient descent	`SGDClassifier`	120	`loss` ∈ {`"hinge"`;`"log"`;`"modified_huber"`;
			`"squared_hinge"`;`"perceptron"`}
			`penalty` ∈ {`"l2"`;`"elasticnet"`}
			`alpha` ∈ `10.0**-np.arange(1,7)`
			`class_weight="balanced"`
			`average` ∈ {`True`;`False`}

Table 3.8: The different configurations of the experiment 4

Structure and format



11 Bibliography

[1] "The Agile Alliance," Agile Alliance, 2012. [Online]. http://www.agilealliance.org/. [Accessed 25 September 2012].

[2] F. J. Abrantes and H. G. Travassos, "Common Agile Practices in Software Pr *International Symposium on Empirical Software Engineering and Measurement*, 2011.

[3] T. Dybå and T. Dingsøyr, "Empirical studies of agile software develo

The purpose of this thesis is thus to answer the question:

What impact does the use of agile principles and practices have on the large-scale software development projects Project A and Project B?

The purpose will be answered through a multiple-case study of Project A and Project B. The two projects are chosen because of the different extents to which they have adopted agile principles and practices. Project A is considered to represent a more traditional, plan-driven development process, with fewer implemented practices, while Project B represents a more agile approach with more implemented practices. The supposition is that the two projects, Project A and Project B, differ sufficiently in their approach to software development that the impact of agile software development will be possible to study by a comparison between the two projects.

1.4.1 Definition of agile principles and practices

Since PDU X follows the definition given by the Agile Alliance their definition is considered suitable also for this thesis. Thus we define:

- An agile software development *as* a software development that follows the values and principles stated in the Agile Manifesto.
- An agile software development method *as* a software development method that follows the values and principles behind the Agile Manifesto.
- The agile principles *as* the twelve principles stated in the Agile Manifesto
- An agile practice *as* a practice that implements the values and principles behind the Agile Manifesto.

2 Limitations

The focus of the thesis is on internal factors. This means that the design, implementation, gration and testing phases are the main phases under study, not the pre-study and requirement analysis nor the deployment, usability and acceptance testing or maintenance. In consequence there is a focus on how agile principles and practices have affected the efficiency of the development, not the externally focused partner of efficiency – effectiveness. Customer interactions and feedback are not investigated.

Ericsson is a huge company with many software development units. We will only study the agile practices and methods implemented in PDU X and only in the two chosen projects. However a discussion of the general applicability of the results to other projects inside Ericsson and even to other companies is made in chapter 10.

A great thesis:

An interesting problem

A convincing theory

A reliable method

A working solution

Established effects

Great presentation

But I will become a Master of Engineering, right?

Engineering vs research

	Engineering	Research
Rationale	Solve a problem	Gain understanding
Activities	Design, implement, verify	interviews, experiments, proofs, ...
Goal	Satisfied customers	New/shared understanding

In order to **solve a problem**, you need to **gain understanding** of the problem

In order to **verify** your implementation, you may need to do **experiments, interviews or proofs**

In order to have **satisfied customers**, you need to achieve a **shared understanding** that the problem has been solved appropriately

Thesis types

Thesis types

- **Evaluations** of new techniques or methods to improve existing products or processes
- **Design** of a prototype application
- **Incremental** improvements of existing techniques of methods

Evaluation

General problem: Does the code quality deteriorate over time? How do we know?

Approach 1: Relate Git commits to code metrics such as cyclomatic complexity and draw a graph

Why is this not a good idea?

- We have not defined what we mean by ”code quality”, and hence, we have no way of knowing what to measure, or whether it relates to our desired quality.
- There is no clear sense of how to assess what we have done.
- There is no mention of how this would be useful to know.

Approach 2

General problem: Does the code quality deteriorate over time? How do we know?

Approach 2: Based on interviews, we define code quality as *detected faults*. Determine whether detected faults correlate with cyclomatic complexity.
We define the purpose as being able to answer the question ”What do we need to improve in order to produce long-term maintainable software?”

Why is this a better approach?

- We now have a definition of code quality
- The result can be assessed

Design

General problem: Create a new Foo application at our company

Approach 1: Read about the latest techniques that can be used on Wikipedia and on project sites, implement the system and ask the company supervisor if he/she is happy

Why is this not a good approach?

- We don’t know why the company wants the Foo application, how to evaluate it, or how long time it would take to implement it in full.
  - IF the requirements are not clear from the start, and the estimated time to implement the working, full solution is > 10 weeks, **do not aim for a full solution**

Approach 2

If the projected time to implement a full solution is > 10 weeks

- Conduct a set of semi-structured interviews to understand the problem domain and the goal,
- a literature survey to understand solutions to similar problems,
- and a few structured iterations of development and documented customer feedback, to produce *a set of requirements* based on the initial prototypes.

Approach 3

If the projected time to implement a full solution is <= 10 weeks

- Determine functional and **non-functional requirements**,
- a literature survey to understand solutions to similar problems and **how to assess them**,
- develop the application iteratively, and **evaluate the resulting application** based on the non-functional requirements

Incremental improvement

General problem: We would like to perform testing of Telecom equipment with less hardware resources

Approach 1: Implement a booking system that automatically releases resources upon expired time slots.

Why is this not a good approach?

- We do not know how and why people use hardware resources, so we do not know how to optimize something.
- Is this a technical, an organizational problem or a cultural issue?
- How do we even measure utilization?

Approach 2

- Conduct an interview series to establish how different people perceive the problem
- Conduct an observational study to determine how people actually use resources
- Find a suitable model for resource utilization in the literature and apply it
- Measure utilization and relate to the results of the interviews

What’s next?

Write a draft of your thesis plan by next Monday. Focus on the main topic, relate to previous courses, both the WHAT and the HOW.

Summary

- Write a draft thesis & plan, prepare for and participate in seminars
- A great thesis is a marriage between solid engineering skills, genuine scientific approach to validate your work, and a lucid presentation.
- There are three main types of industry theses: evaluations, prototypes and improvements.