# *2IMP40 - Empirical Methods in Software Engineering*

*Nathan Cassee & Alexander Serebrenik*

*Q2 – 2023*

2IMP40 is a Q2 master course on Empirical methods in Software Engineering. After concluding this course, you should be able to effectively and accurately apply techniques to conduct empirical research studies in Software Engineering. To achieve this we will combine on campus lectures on empirical research methods in Software Engineering with practical workshops in which you get to engage with empirical research methods.

Note: This course has been given in 2020 and 2021 as "Applications of Data Science to Software Engineering". Based on student feedback we changed the name to better reflect the contents of the course.

**Changelist:**

- Nov 21st: Added the primary sources for Design Science.
- Dec 5th: Updated describe a study assignment.
- Jan 9th: Added more details on the exam.
- Jan 23rd: Added the individual contribution section.

## *Course Overview*

THE COURSE ITSELF can be divided into three parts. The first part covers fundamentals, the second part discusses how to gather data used for empirical studies and the final part covers the various techniques we use to analyze the collected data.

The red thread through all of these three topics is our interest: *Software Engineering*. During this course we will talk, and learn how to apply, research techniques that can be used to study Software Engineering. We will see examples of academic studies that have sought to understand software development, and we will learn how to apply the techniques used in these studies.

An important part of this course is understanding what Software Engineering is, and how developers and companies develop Software. This is a vital aspect of the course, if you are not familiar with these topics please use self-study to learn more about these topics. The following video should provide an overview: `https://www.youtube.com/watch?v=1LSXEC0Clow&ab_channel=Indeed`

After concluding this course you should master all of the following learning objectives:

- Formulate and motivate research questions pertaining to software engineering, identify questions that can and that cannot be answered by means of empirical research
- Students should be able to independently design and execute a sound empirical study in Software Engineering given a state-of-the-art dataset.

- Students should be able to evaluate empirical studies in Software Engineering using tools accepted in the field and be able to to identify threats to validity.
- Students should be able to describe the results of empirical studies to practitioners not familiar with academic research.
- Students should be able to comprehend the research methods used for empirical studies in Software Engineering.

Using these lecture-notes we discuss the course schedule, assessment, lectures, and we reference the sources on which our work is based.

### *Schedule*

Table 1 contains an overview of all teaching activities and deadlines. For the activities, we distinguish between *Lectures*, *Workshops* and *Coffee-hours*. Lectures are classic on campus lectures, workshops are more interactive hands-on sessions, and coffee-hours are informal catch-ups where you can talk to us about the course. During the coffee-hours you can tell us what you like, and what you would like to improve about the course. In addition to talking about the course, you can also use the coffee-hours to ask questions to us about the course material or the assignments.

### *Feedback*

The design, planning, and execution of education is difficult, therefore, we like to hear what you think of the course. For this, feel free to walk by our offices (MF 6.095 & MF 6.086b), shoot us an email (`a.serebrenik@tue.nl` & `n.w.cassee@tue.nl`), visit the coffee-hours, or use an anonymous feedback form to share your concerns or ideas.[1]

## *Assessment*

YOUR GRADE FOR THIS course is based on the following two components:

- **Assignments**: There will be two different types of assignment during the course: Design a Study and Describe a study. For design a study you will team-up in groups and to design and execute an empirical study during the quartile. For describe a study you will read and summarize an academic study for practitioners not familiar with academic research.

[1] `https://forms.microsoft.com/e/42g98m81iv`

| When | What | Topic |
|---|---|---|
| Nov 14 13:30 | Lecture | Introduction to Empiricism and Rationalism |
| Nov 16 08:45 | Lecture | Research Questions and Research Strategies |
| Nov 21 13:30 | Guest-Lecture | Adyen |
| Nov 21 14:30 | Lecture | Design Science |
| Nov 23 08:45 | Lecture | Sampling |
| Nov 24 23:59 | Deadline | 1st deadline for Design a study |
| Nov 28 13:30 | Lecture | Interviews & Surveys |
| Nov 30 08:45 | Workshop | How to read an empirical paper |
| Dec 5 13:30 | Lecture | Mining Software Repositories I |
| Dec 7 08:45 | Workshop | Mining Software Repositories |
| Dec 8 23:59 | Deadline | 2nd deadline for Design a study |
| Dec 12 13:30 | Lecture | Quantitative Analysis |
| Dec 14 08:45 | Lecture | Qualitative Analysis |
| Dec 15 23:59 | Deadline | 1st deadline for Describe a study |
| Dec 19 13:30 | Lecture | Advanced Repository Mining |
| Dec 21 08:45 | Coffee-hour | Feedback and assignment Q&A session |
| Dec 22 23:59 | Deadline | 3rd deadline for Design a study |
| Jan 9 13:30 | Coffee-hour | Feedback and assignment Q&A session |
| Jan 11 08:45 | Lecture | Threats to Validity |
| Jan 12 23:59 | Deadline | 2nd deadline for Describe a Study |
| Jan 16 13:30 | Workshop | Recap & Threats & Trustworthiness |
| Jan 18 08:45 | Coffee-hour | Exam and Assignment Q&A session |
| Jan 23 09:00 | Exam | Final examination |
| Feb 2 23:59 | Deadline | Final deadline for Design a study |
| Apr 9 18:00 | Exam | Resit |

Table 1: Course schedule per timeslot per week. To see what room we will meet up in please use https://mytimetable.tue.nl.

- **Exam**: During the exam-week you will take a 120-minute,[2] closed-book written exam on the course material. This exam will contain a mix of open and closed questions based on research methods presented during this course. More information related to the exam will be provided during the course, including practice exams. The quizzes from last year have been copied over to this year (You can find them under the Canvas header quizzes) as practice material. They contain closed questions that are representative of the closed questions we will ask on the exam.

  Your assignment grade will be determined as follows:

$$assignment\_grade = 0.7 \times design\_grade + 0.3 \times describe\_grade$$

[2] The length and format of the exam is not set in stone yet, we will announce any updates to the exam during the course.

Your final course grade will be calculated using the following formula:

$$final\_grade = \begin{cases} 0.7 \times assignment\_grade + 0.3 \times exam\_grade & \text{if } exam\_grade \geq 5 \wedge \ assignment\_grade \geq 5 \\ min(5,\ 0.7 \times assignment\_grade + 0.3 \times exam\_grade) & \text{else} \end{cases}$$

**Note that both your $assignment\_grade$ and $exam\_grade$ have to be $\geq$ 5 to pass the course.** If either one is less than 5 your maximum grade for the course will be a 5.

For a full overview of the assignments please see the Section on assignments.

### *Exam*

A 120-minute closed book exam with a mix of closed and open questions. The exam will be an ANS paper-based exam, and in addition to the exam you will also receive a paper, in which the threats to validity section has been redacted.[3] To help prepare for the exam, we have published the resit exam from 2021/2022, and the exam from 2022/2022. For the exam 2021/2022 the correct answers have been uploaded as well. However, the uploaded exams are different in structure from this year's exam. Please see below for the structure of the exam.

**Part 1 (20%)** Closed questions on course theory. These questions are similar in wording and depth to the questions in the self-study quizzes on Canvas. Please be aware that some of these quizzes from 2021 cover topics that have been redesigned for this year's edition.

**Part 2 (40%)** The second part will contain a series of open questions of open questions on course theory. The goal of these questions is to verify whether you understand and can apply the theory of the course to study software engineering. In particular, for this part, we will give you the motivation for a study, potentially even research questions, and we will ask you to design several alternative study designs to answer the research questions. You will then have to explain the strengths and weaknesses of each of these designs and argue which design you would pick and why.

As an example you consider the following excerpt:

> For a study investigating the relation between developers' emotions and perceived productivity, we needed a reliable and effective way to measure developers' productivity and a setting in which these can be measured.

Based on this description we expect that you can come up with three distinct, alternative study designs. You should be able to pick research methodologies, explain to which strategies they belong and explain the strengths and weaknesses of each of these designs.

[3] This paper will be a 4-page paper.

**Part 3 (40%)** For the third part, we ask you to read and accompanying four-page paper with the threats to validity section redacted. You should be able to understand the methodology of the accompanying paper in enough detail to be able to draft several relevant threats to validity based on the methodology of the paper.

## *Teaching activities*

For 2IMP40 we will organize lectures, guest-lectures, workshops and coffee-hours. In this section you can find details on each of these activities, and what you should prepare.

### *Lectures*

Lectures are classical in-person lectures, they are supported by the online material that was recorded in 2020. Based on feedback from previous years we have decided to restructure and redesign some lectures. The following lectures have been redesigned:

The online lectures of 2020/2021 can be accessed at the YouTube channel of the course: `https://www.youtube.com/channel/UCUeRK8nJKyj_i_Yz81eHa8g`

- The *Interview & Surveys* lectures have been merged into a single lecture.
- The Mining Software Repositories lecture has been redesigned and split into one lecture and one workshop.
- The *Qualitative Analysis* lecture has been redesigned.
- The *Quantitative Analysis and ML4SE* lecture has been redesigned.
- The *Threats to Validity* lecture has been expanded to include material related to trustworthiness.
- We have added a lecture on *Design Science* to bridge the gap between descriptive research and design research.

This year the lectures will be recorded and each lecture should become available on `http://videocollege.tue.nl/`.[4] While the lectures will be recorded we do strongly encourage you to attend the lectures, we try to make the lecture as interactive as possible and we welcome questions and discussions during the lectures.

### *Guest-lecture*

In addition to organizing in-person lectures we also have the honor of welcoming two guest-speakers, Maurício Aniche and Carianne Pretorius, from Adyen to talk about empirical research in software engineering.[5]

[4] Workshops and coffee-hours won't be recorded.

[5] Adyen is a very successful Dutch startup providing payment services to many different companies. Their site can be found here: `https://www.adyen.com/`

### *Workshops*

To complement the lectures we also organize a set of workshops. Through the workshops, we aim to provide a more hands-on experience in which you can directly interact with the course material. For 2IMP40 we will organize the following set of workshops:

- How to read an empirical paper.
- Mining Software Repositories.
- Threats to Validity & Trustworthiness.

When attending a workshop please ensure you bring your laptop, and if you prefer some pen and paper.

## *Assignments*

TO SHOWCASE THAT YOU are able to apply the knowledge you have learned this course consists of two assignments, *Design a study* and *Describe a study*.

### *Design a Study*

In this assignment, you will work in a group of four to design and execute an empirical study.[6] For this assignment you will pick an existing, state-of-the-art, dataset. As a starting point we have selected a list of datasets that you can use for this assignment. Your final submission for this project is a 4 + 1 page paper[7] formatted as an academic paper.

To format your submissions as an academic paper, please use the IEEE publishing template that can be found at `https://www.ieee.org/conferences/publishing/templates.html`. You should specify your documentclass using:

`\documentclass[10pt,conference]{IEEEtran}`.

For this assignment there will be a total of four deadlines, the four deadlines will build up, and for each subsequent deadline we expect a more complete version of your report. The first three deadlines are not mandatory. We only give feedback on your work to help you improve your project. The fourth deadline is mandatory and will be graded.

Your final submission for this assignment will be the 4+1 page paper, it should be formatted as an academic paper and it should describe the execution of a single empirical software engineering study.[8] Therefore, your report should contain the following sections (please do not include an abstract):

[6] Please don't confuse this assignment with the notion of *Design Science*. For this assignment, we expect you to design a *descriptive* study without an engineering component.

[7] With 4 + 1 we mean that a maximum of four pages of content are allowed, with one additional page for bibliographic references.

[8] In 2021 one of the studies for Design a Study was so well executed that it was published as an academic paper at the conference Mining Software Repositories 2021.

- **Introduction**: In the introduction you should explain your motivation for your work, you should introduce and cite relevant context and most importantly you should introduce your research questions:
  - You should list at least two research questions and it is important that these research questions are not variations of the same question.[9]
  - For each research question you should provide a motivation that answers the questions: *Why should we study this research question?*, *Who benefits from the answer to this research question?*. **Answering these questions is difficult, but is very crucial to a good introduction.**
  - Consider the lecture on research strategies and research questions to draft your research questions. When deciding on a research question try to explicitly think about the different types of research questions that exist, and how your work relates to other studies.
- **Related work**: In related work you should describe relevant related work (peer-reviewed academic papers) and explain how the work relates to the current study, and how your study differs. You should find and cite at least 6 relevant peer-reviewed papers published in the last 5 years.
  - When describing the papers you cite in your related work you should keep in mind that the findings of the work you cite are more important than the methodology used. Please don't just describe the what the authors did, but focus on what they found!
- **Methodology**: This section should describe and motivate the chosen research strategies and methods for the study, concretely you should describe what steps you take to answer the research questions. For each method you use you should include sufficient details to facilitate replication.
  - In your methodology you should clearly operationalize the concepts in your research questions. Concepts you introduce in your research questions should be clearly defined.
- **Results**: The results section should contain the results you obtained by executing the steps described in the methodology. Additionally, it should contain an answer to each research question.
- **Discussion**: In the discussion you should interpret your results and discuss the implications of your findings for software engineering.

[9] For instance, we would consider the two questions: *Does gender influence commit-rate?* and *Does experience influence commit-rate?* to be two variations of the same questions: *How do developer characteristics influence commit-rate?*

You can achieve this by comparing and contrasting your results with other studies.

- **Threats to Validity**: Certain threats to validity might have impacted the validity of your study. In this section you should describe them according to the model of Wohlin *et al.* [10]
- **Conclusion**: The conclusion should summarize your work, from motivation to results.
- **Individual Contributions**: In this section, you should describe the individual contributions of each group member. You should describe both *What* each group member has done, and *How much* each group member has contributed by using percentage scores to estimate contributions. This section **does not** count towards your page-limit. In other words, you can include it on the 5th page of your report, before the references.

Your paper will be graded based on the adherence to the report outline above. Additionally, we will consider the alignment between the motivation, research questions, methodology, and results.

For instance, when you opt for a particular sampling approach you should consider and explain why it is appropriate for your motivation and research questions.

*Dataset papers*

- A Dataset of Bot and Human Activities in GitHub (`https://decan.lexpage.net/files/MSR-2023.pdf`)
- GitHub OSS Governance File Dataset (`GitHubOSSGovernanceFileDataset`)
- GIRT-DATA: Sampling GitHub Issue Report Templates (`https://arxiv.org/pdf/2303.09236.pdf`)
- A Time Series-Based Dataset of Open-Source Software Evolution (`https://drive.google.com/file/d/1svIexRcWrXBhb_pCVhfs7yKRzGsyaBpV/view`)
- A Versatile Dataset of Agile Open Source Software Projects (`https://solar.cs.ucl.ac.uk/pdf/tawosi2022msr.pdf`)
- DaSEA – A Dataset for Software Ecosystem Analysis (`https://itu.dk/~ropf/blog/assets/msr2022.pdf`)
- DISCO: A Dataset of Discord Chat Conversations for Software Engineering Research (`http://olgabaysal.com/pdf/MuthuSubash_MSR2022_DataShowcase.pdf`)
- The OCEAN mailing list data set: Network analysis spanning mailing lists and code repositories (`https://arxiv.org/pdf/2204.00603.pdf`)

[10] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Planning*, pages 89–116. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-29044-2_8

*First deliverable:* For the first deliverable we ask that you pick three of the above dataset papers, and that you summarize each paper in a single paragraph. In addition to the summary you should also propose four research questions that you can pose based on one (or more) of the datasets.

This deadline is a good starting point to ensure that you can generate a .pdf that adheres to the report outline.

This version of the report will not be graded, we will only give feedback. The goal of this exercise is to practice academic writing, ensure you are familiar with at least three of the dataset papers and explore possible research questions. We will give feedback on the wording of the research questions.

Deadline end of week 2.

*Second deliverable:* For the second deliverable we ask that you submit a draft of your report with the introduction, related work and methodology. To write this draft, you should, together with your group, pick two research questions and one or two of the datasets to answer the research questions.

This version of the report will not be graded, we will only give feedback. While giving feedback on your report we will pay attention to your research questions: Are they feasible? Are they novel? And we will focus on the alignment: Does the motivation in the introduction align with the research questions, and does the methodology and chosen dataset(s) align with your research questions?

Deadline end of week 4.

*Third deliverable:* We expect you to submit a more complete draft of your report in which you addressed the feedback on your second deliverable. This version of the report should again contain a draft of the introduction, related work, and methodology. However, where possible this version should also contain a sketch of the results section.

This version of the report will not be graded, we will only give feedback that you can use to sharpen and strengthen your work.

Deadline end of week 6.

*Fourth deliverable* For the final deliverable we ask that you submit a full report, wrt. the requirements for the report sketched in this section.

**This version will be graded, according to the requirements specified in this section.**

Deadline end of week 10.

### *Describe a study*

In addition to being able to conduct research yourself we believe it is also important that you can help transfer knowledge from research papers to practitioners. To that end, the *Describe a study* assignment asks you to take a research paper, or research output, and **individually** record a video of at most two minutes in which you pitch the research and its findings to a practitioner audience. For the pitch, you should pretend that you have been invited to speak at a large developer conference.

A good example of this knowledge transfer is *"It Will Never Work In Theory"* by Greg Wilson. `https://neverworkintheory.org/reviews/`

While  summarzing research for practitioners you should keep in mind several key thoughts:

- Practitioners are not familiar with academic research, or the methodologies used software engineering researchers.
- While researchers are interested in the exact methodology, practitioners will care less about how a study was conducted but instead care greatly about **what the study means for their work**.

For *Describe a study* you should record a two-minute pitch summarizing and describing a research paper. To record the video you are free to use any tool you like, however, in the video your face should be visible. Additionally, you are free to use any supporting material you like, such as slides or images, however, these should not distract too much from the pitch.

You are asked to do this in two rounds. For the first round you should pick a paper, and write a brief summary or script for your pitch. After your submission we will give you feedback on your script, and you can use this feedback to improve your pitch. For the second round you should record your pitch, and submit it. This submission will be graded.

To pick paper, please subscribe to one of the Canvas groups. While the assignment is an individual assignment, we use the Canvas groups to ensure that you are divided equally over the papers. By joining a group, you pick the paper, the .pdf of the paper can then be found by navigating to the Files section on Canvas.[11]

The grading criteria for your pitch are:

- **Content**: Does the pitch contain a clear and accurate summary of the research paper? Does the pitch contain a clear description of the findings of the paper?
- **Delivery**: Is the pitch delivered in a clear and concise manner? Is the pitch delivered in a way that is engaging for the audience? Does the pitch stay within the time limit?

[11] There is a folder "Papers" in the Canvas files section.

- **Relevance**: Does your pitch highlight the relevance of the research for practitioners? Does it accurately address the question: *Why should practitioners care about this research?*

*The deadlines for this assignment are Week 5 for the scripts and Week 7 for the pitches.*

## *Topics*

One can roughly divide this course into three main topics: *Fundamentals*, *Data Collection*, and *Data Analysis*. In this section we discuss these three topics and we outline the subjects we will discuss for each of the topics.

For particular topics you might see a bibliographic reference in the margin, this means that this was our primary source for this topic.

### *Fundamentals*

In the fundamentals we cover the definition of Empiricism, the role of empirical studies in Software Engineering research, Research Questions and Research Strategies. At the end of this you part should be able to:

- Explain what research strategies exist[12] and be able to identify what research strategies are used in an academic paper.
- Understand how Design Science[13] relates to Empirical Software Engineering.[14]
- Design suitable research questions given a motivation and be able to pick suitable research strategies to address the drafted research questions.
- Write a threats to validity section of an empirical Software Engineering study.[15]

[12] Klaas-Jan Stol and Brian Fitzgerald. A holistic overview of software engineering research strategies. pages 47–54. IEEE, 5 2015. ISBN 978-1-4673-7028-8. DOI: 10.1109/CESI.2015.15

[13] Roel J. Wieringa. *Design Science Methodology for Information Systems and Software Engineering*. Springer Berlin Heidelberg, 2014. ISBN 978-3-662-43838-1. DOI: 10.1007/978-3-662-43839-8

[14] Emelie Engström, Margaret Anne Storey, Per Runeson, Martin Höst, and Maria Teresa Baldassarre. How software engineering research aligns with design science: a review. *Empirical Software Engineering*, 25:2630–2660, 7 2020. ISSN 15737616. DOI: 10.1007/s10664-020-09818-7

[15] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Planning*, pages 89–116. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-29044-2_8

### *Data Collection*

For data collection we cover the techniques used to gather data for Empirical studies in Software Engineering. During this section we will cover data-collection techniques for both quantitative and qualitative data. After concluding this section you should be able to:

- Name what sampling techniques exists, and explain how each technique works, and the impact of selecting a particular sampling strategy. [16]
- Know how to design and run surveys [17] and interviews. [18]
- Be familiar with promises and perils related to the extraction of data from software repositories. [19]

After the conclusion of this section of the course you should be able to list, describe, and apply all methods discussed for data collection. Concretely, this means that you should understand a discussed technique well enough to be able to apply it in an academic study. In other words, given a research question you should be able to select an appropriate data collection technique, argue why the technique is appropriate, and describe how you use the technique.

### *Data Analysis*

After data collection the question rises: How can we synthesize the collected data to answer our research questions? During this section we distinguish two broad categories of analysis one can use to analyze data: *Qualitative analysis* and *Quantitative analysis*. In particular we discuss:

- Data analysis techniques for repository data.
- Qualitative analysis and Grounded Theory building. [20]
- Statistics and time-series analysis.

After the lectures of this section you should be able to list, describe and apply all the methods discussed for data analysis. In essence given a collected dataset you should be able to select an appropriate analysis technique and correctly apply it to answer a research question. While doing this you should be able to argue why your chosen technique is appropriate, and describe in sufficient detail how you applied the analysis technique.

Congratulations! You made it to the end of the syllabus. Both Alexander and I love pets, and to showcase that you made it to the end of the syllabus please send us your favorite pet picture (or a picture of your favorite animal) to us at a.serebrenik@tue.nl and n.w.cassee@tue.nl!

[16] Sebastian Baltes and Paul Ralph. Sampling in software engineering research: a critical review and guidelines. *Empirical Software Engineering*, 27:94, 7 2022. ISSN 1382-3256. DOI: 10.1007/s10664-021-10072-8. URL `https://link.springer.com/10.1007/s10664-021-10072-8`

[17] Norbert Schwarz and Daphna Oyserman. Asking questions about behavior: cognition, communication, and questionnaire construction. *The American Journal of Evaluation*, 22:127–160, 2001. ISSN 1098-2140. DOI: https://doi.org/10.1016/S1098-2140(01)00133-3. URL `http://www.sciencedirect.com/science/article/pii/S1098214001001333`

[18] Per Erik Strandberg. Ethical interviews in software engineering. pages 1–11. IEEE, 9 2019. ISBN 978-1-7281-2968-6. DOI: 10.1109/ESEM.2019.8870192. URL `https://ieeexplore.ieee.org/document/8870192/`

[19] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. An in-depth study of the promises and perils of mining github. *Empirical Software Engineering*, 21: 2035–2071, 10 2016. ISSN 1382-3256. DOI: 10.1007/s10664-015-9393-5; and Christian Bird, Peter C. Rigby, Earl T. Barr, David J. Hamilton, Daniel M. German, and Prem Devanbu. The promises and perils of mining git. pages 1–10. IEEE, 5 2009. ISBN 978-1-4244-3493-0. DOI: 10.1109/MSR.2009.5069475

[20] Rashina Hoda. Socio-technical grounded theory for software engineering. *IEEE Transactions on Software Engineering*, pages 1–1, 2021. ISSN 0098-5589. DOI: 10.1109/TSE.2021.3106280