On Hard and Soft Skills for Surviving Your PhD Journey

Walid Maalej

About myself

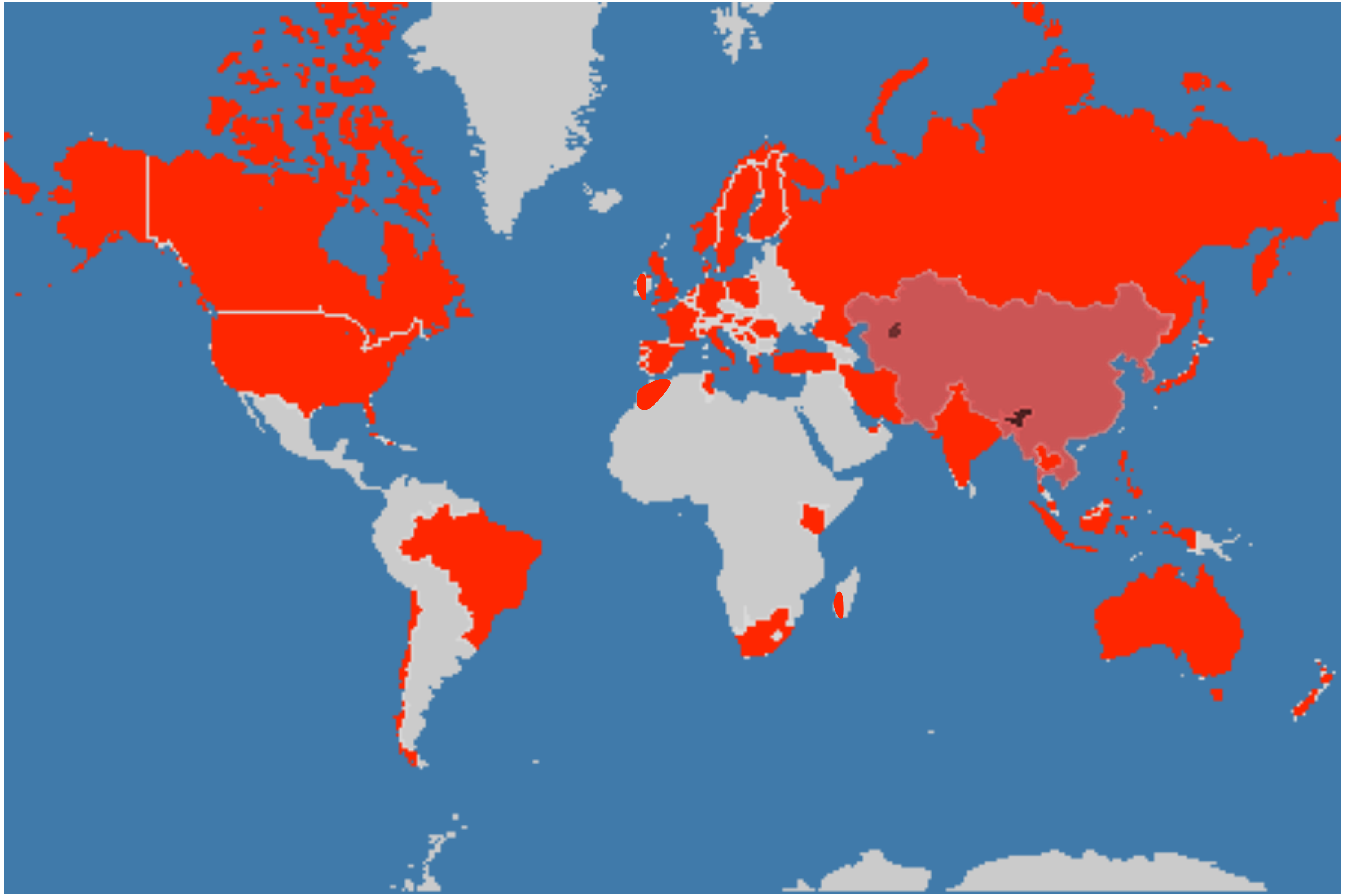
UHH Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

HITeC Hamburger Informatik Technologie-Center

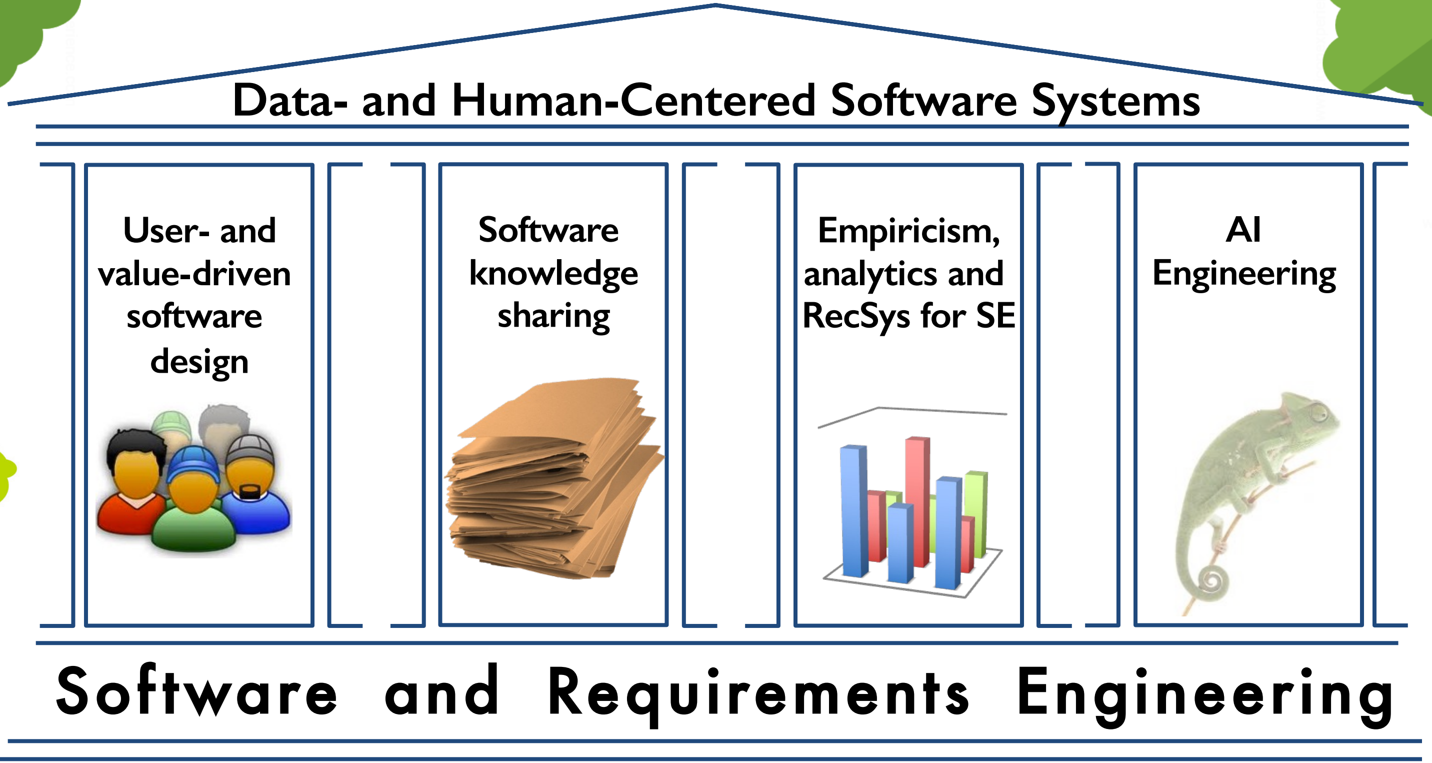
acm

IFB | innovations starter

2022 RE



Research Blueprint



Data- and Human-Centered Software Systems

User- and value-driven software design

Software knowledge sharing

Empiricism, analytics and RecSys for SE

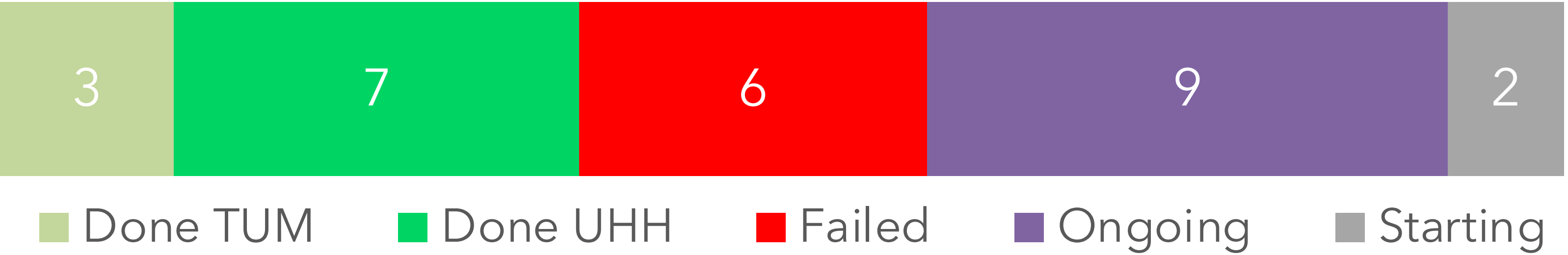
AI Engineering

Software and Requirements Engineering

My experience with PhD projects

1. Own PhD in 2010

2. More than two dozens PhD students (direct supervisor)



3. PhD theses reviewer or examiner around the globe

🇸🇪 🇩🇪 🇦🇹 🇺🇸 🇬🇧 🇦🇺 🇮🇹

Disclaimer:
Almost all my experience is on
Empirical Software Engineering

What is empirical research?

Observation

+

Data

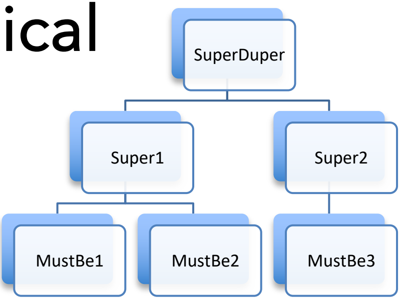
Systematic

The “new standard” in the SE/CS

Other research approaches

Engineering and design

Analytical



SuperDuper

Super1

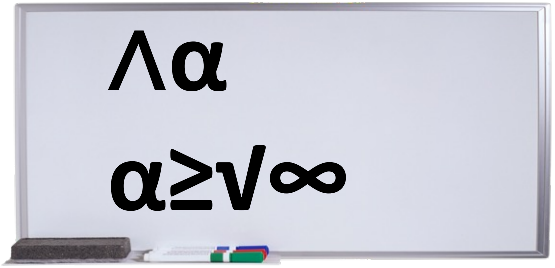
Super2

MustBe1

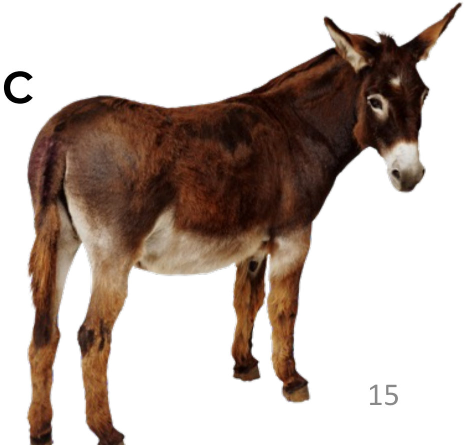
MustBe2

MustBe3

Mathematical, formal



Anecdotic



Goals of empirical studies

A. Explore

Understand phenomena and identify problems

B. Evaluate

Check and improve hypotheses, measure impact

Research strategies

1. Qualitative

2. Quantitative

Both are important. If you want to excel, combine!

BOTH QUALITATIVE AND QUANTITATIVE ARE IMPORTANT

If you want to excel, combine!

Quick dive into selected methods

Interviews

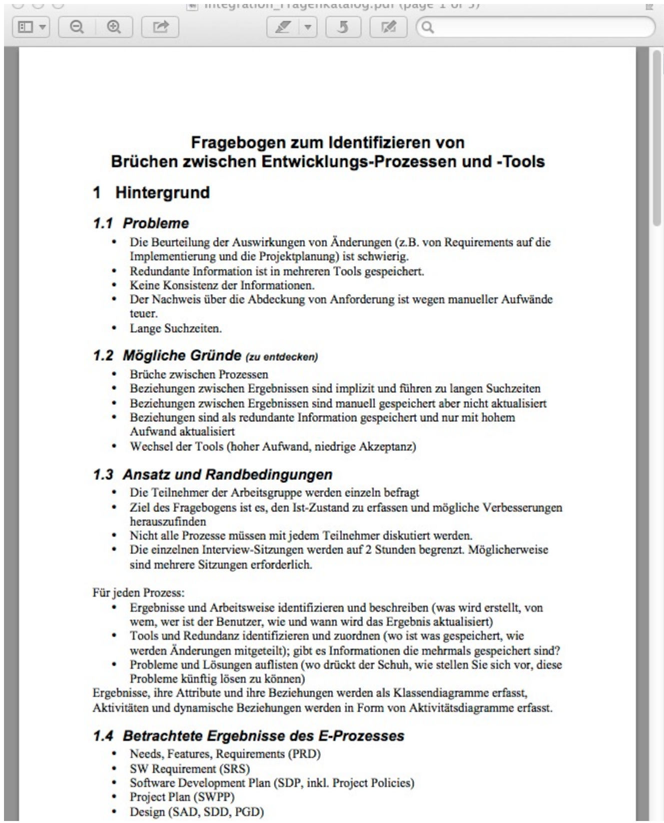
- Include open questions
- Goes in depth (what and why)
- Are rather…
  - Subjective
  - Exploratory
  - Involve users!

INTERVIEW GUIDELINES [..]

Interview guideline

- Prepare questions and first drafts
  - Use **Templates**!
- Iterate
- Make dry runs
- Do not influence subjects
- Stay in scope
- Listen actively and suggest ideas

1. Prepare and structure questions use templates...



Fragebogen zum Identifizieren von Brüchen zwischen Entwicklungs-Prozessen und -Tools

1 Hintergrund

1.1 *Probleme*

- Die Beurteilung der Auswirkungen von Änderungen (z.B. von Requirements auf die Implementierung und die Projektplanung) ist schwierig.
- Redundante Information ist in mehreren Tools gespeichert.
- Keine Konsistenz der Informationen.
- Der Nachweis über die Abdeckung von Anforderung ist wegen manueller Aufwände teuer.
- Lange Suchzeiten.

1.2 *Mögliche Gründe (zu entdecken)*

- Brüche zwischen Prozessen
- Beziehungen zwischen Ergebnissen sind implizit und führen zu langen Suchzeiten
- Beziehungen zwischen Ergebnissen sind manuell gespeichert aber nicht aktualisiert
- Beziehungen sind als redundante Information gespeichert und nur mit hohem Aufwand aktualisiert
- Wechsel der Tools (hoher Aufwand, niedrige Akzeptanz)

1.3 *Ansatz und Randbedingungen*

- Die Teilnehmer der Arbeitsgruppe werden einzeln befragt
- Ziel des Fragebogens ist es, den Ist-Zustand zu erfassen und mögliche Verbesserungen herauszufinden
- Nicht alle Prozesse müssen mit jedem Teilnehmer diskutiert werden.
- Die einzelnen Interview-Sitzungen werden auf 2 Stunden begrenzt. Möglicherweise sind mehrere Sitzungen erforderlich.

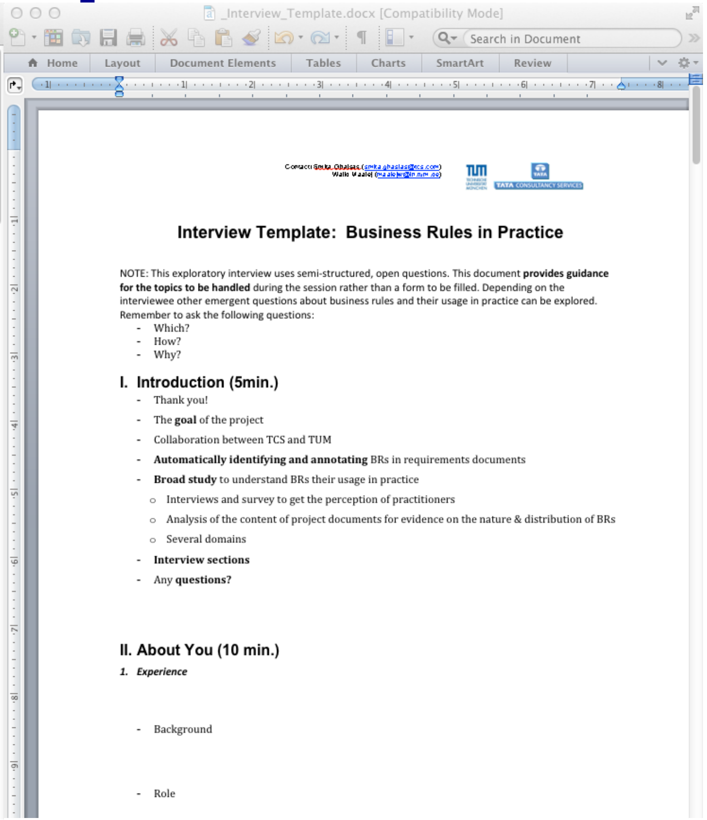
Für jeden Prozess:

- Ergebnisse und Arbeitsweise identifizieren und beschreiben (was wird erstellt, von wem, wer ist der Benutzer, wie und wann wird das Ergebnis aktualisiert)
- Tools und Redundanz identifizieren und zuordnen (wo ist was gespeichert, wie werden Änderungen mitgeteilt); gibt es Informationen die mehrmals gespeichert sind?
- Probleme und Lösungen auflisten (wo drückt der Schuh, wie stellen Sie sich vor, diese Probleme künftig lösen zu können)

Ergebnisse, ihre Attribute und ihre Beziehungen werden als Klassendiagramme erfasst, Aktivitäten und dynamische Beziehungen werden in Form von Aktivitätsdiagramme erfasst.

1.4 *Betrachtete Ergebnisse des E-Prozesses*

- Needs, Features, Requirements (PRD)
- SW Requirement (SRS)
- Software Development Plan (SDP, inkl. Project Policies)
- Project Plan (SWPP)
- Design (SAD, SDD, PGD)



Contact: Smita Ghaisas (smita.ghaisas@tcs.com)
Walid Maalej (maalejw@in.tum.de)

Interview Template: Business Rules in Practice

NOTE: This exploratory interview uses semi-structured, open questions. This document **provides guidance for the topics to be handled** during the session rather than a form to be filled. Depending on the interviewee other emergent questions about business rules and their usage in practice can be explored. Remember to ask the following questions:

- Which?
- How?
- Why?

I. Introduction (5min.)

- Thank you!
- The **goal** of the project
- Collaboration between TCS and TUM
- **Automatically identifying and annotating** BRs in requirements documents
- **Broad study** to understand BRs their usage in practice
  - Interviews and survey to get the perception of practitioners
  - Analysis of the content of project documents for evidence on the nature & distribution of BRs
  - Several domains
- **Interview sections**
- Any **questions?**

II. About You (10 min.)

1. Experience

- Background
- Role

2. Iterate and make a make dry runs!

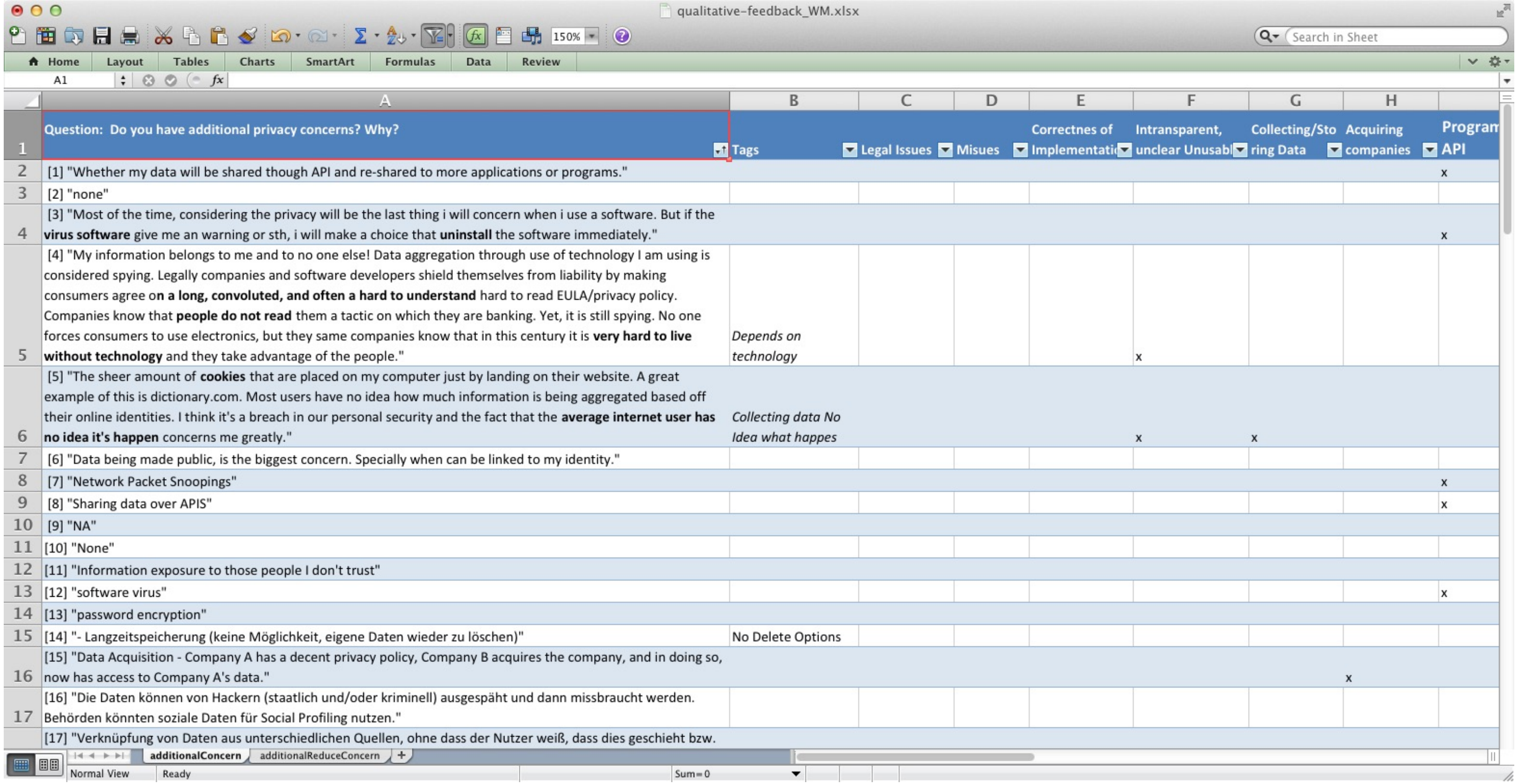
3. Do not influence your subjects!

- By explicitly or implicitly stating how they should answer
- In the way you pose the questions

4. Tighten scope and help subjects to focus!

6. Spend 60 to 90 minutes for one session!

Most difficult part: analysis!



	A	B	C	D	E	F	G	H	
1	Question: Do you have additional privacy concerns? Why?	Tags	Legal Issues	Misues	Correctnes of Implementati	Intransparent, unclear Unusabl	Collecting/Storing Data	Acquiring companies	Program API
2	[1] "Whether my data will be shared though API and re-shared to more applications or programs."								x
3	[2] "none"								
4	[3] "Most of the time, considering the privacy will be the last thing i will concern when i use a software. But if the **virus software** give me an warning or sth, i will make a choice that **uninstall** the software immediately."								x
5	[4] "My information belongs to me and to no one else! Data aggregation through use of technology I am using is considered spying. Legally companies and software developers shield themselves from liability by making consumers agree on **a long, convoluted, and often a hard to understand** hard to read EULA/privacy policy. Companies know that **people do not read** them a tactic on which they are banking. Yet, it is still spying. No one forces consumers to use electronics, but they same companies know that in this century it is **very hard to live without technology** and they take advantage of the people."	*Depends on technology*				x			
6	[5] "The sheer amount of **cookies** that are placed on my computer just by landing on their website. A great example of this is dictionary.com. Most users have no idea how much information is being aggregated based off their online identities. I think it's a breach in our personal security and the fact that the **average internet user has no idea it's happen** concerns me greatly."	*Collecting data No Idea what happes*				x	x		
7	[6] "Data being made public, is the biggest concern. Specially when can be linked to my identity."								
8	[7] "Network Packet Snoopings"								x
9	[8] "Sharing data over APIS"								x
10	[9] "NA"								
11	[10] "None"								
12	[11] "Information exposure to those people I don't trust"								
13	[12] "software virus"								x
14	[13] "password encryption"								
15	[14] "- Langzeitspeicherung (keine Möglichkeit, eigene Daten wieder zu löschen)"	No Delete Options							
16	[15] "Data Acquisition - Company A has a decent privacy policy, Company B acquires the company, and in doing so, now has access to Company A's data."							x	
17	[16] "Die Daten können von Hackern (staatlich und/oder kriminell) ausgespäht und dann missbraucht werden. Behörden könnten soziale Daten für Social Profiling nutzen."								
	[17] "Verknüpfung von Daten aus unterschiedlichen Quellen, ohne dass der Nutzer weiß, dass dies geschieht bzw.								

Questionnaires

Are rather…

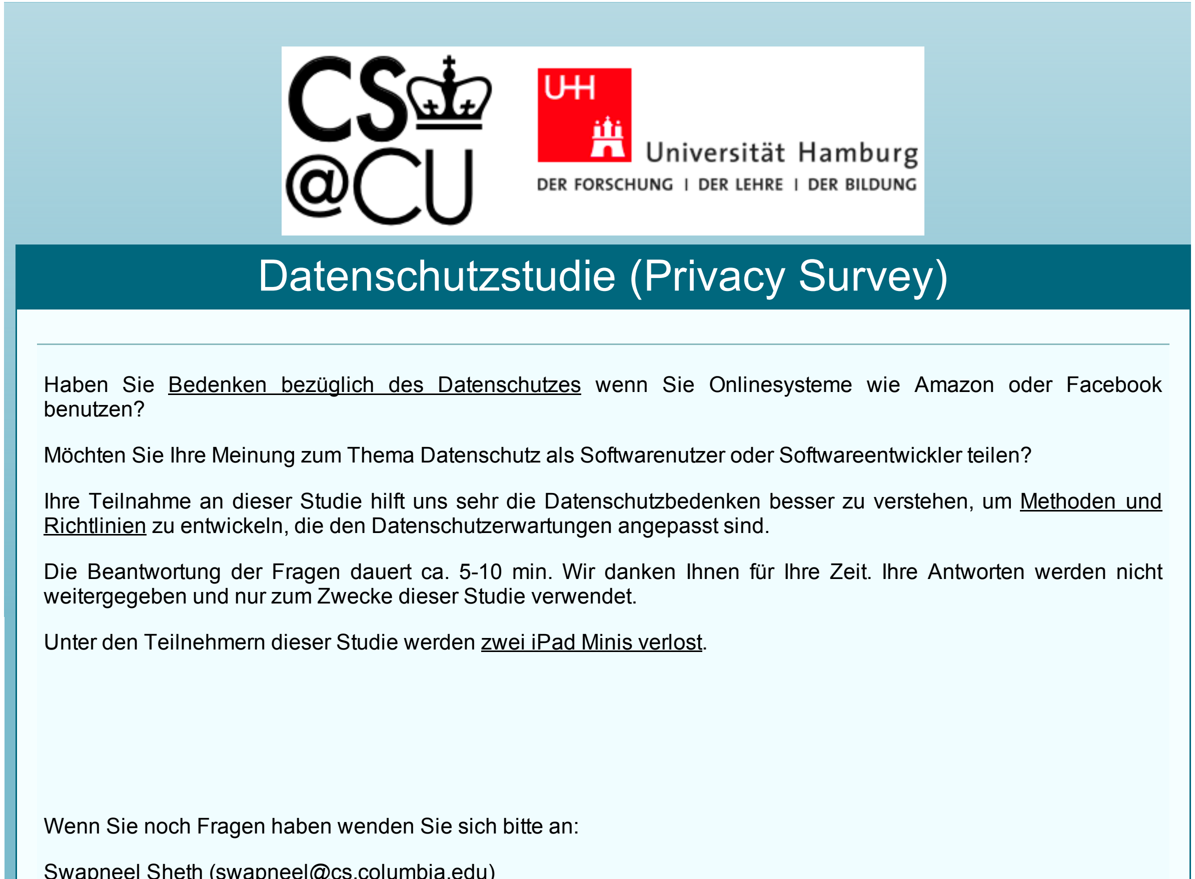
- Subjective
- Quantitative
- Evaluative

Involve MANY users

QUESTIONNAIRE GUIDELINES [..]

Iterate and make a make dry runs!

Describe your objective!



Datenschutzstudie (Privacy Survey)

Haben Sie Bedenken bezüglich des Datenschutzes wenn Sie Onlinesysteme wie Amazon oder Facebook benutzen?

Möchten Sie Ihre Meinung zum Thema Datenschutz als Softwarenutzer oder Softwareentwickler teilen?

Ihre Teilnahme an dieser Studie hilft uns sehr die Datenschutzbedenken besser zu verstehen, um Methoden und Richtlinien zu entwickeln, die den Datenschutzerwartungen angepasst sind.

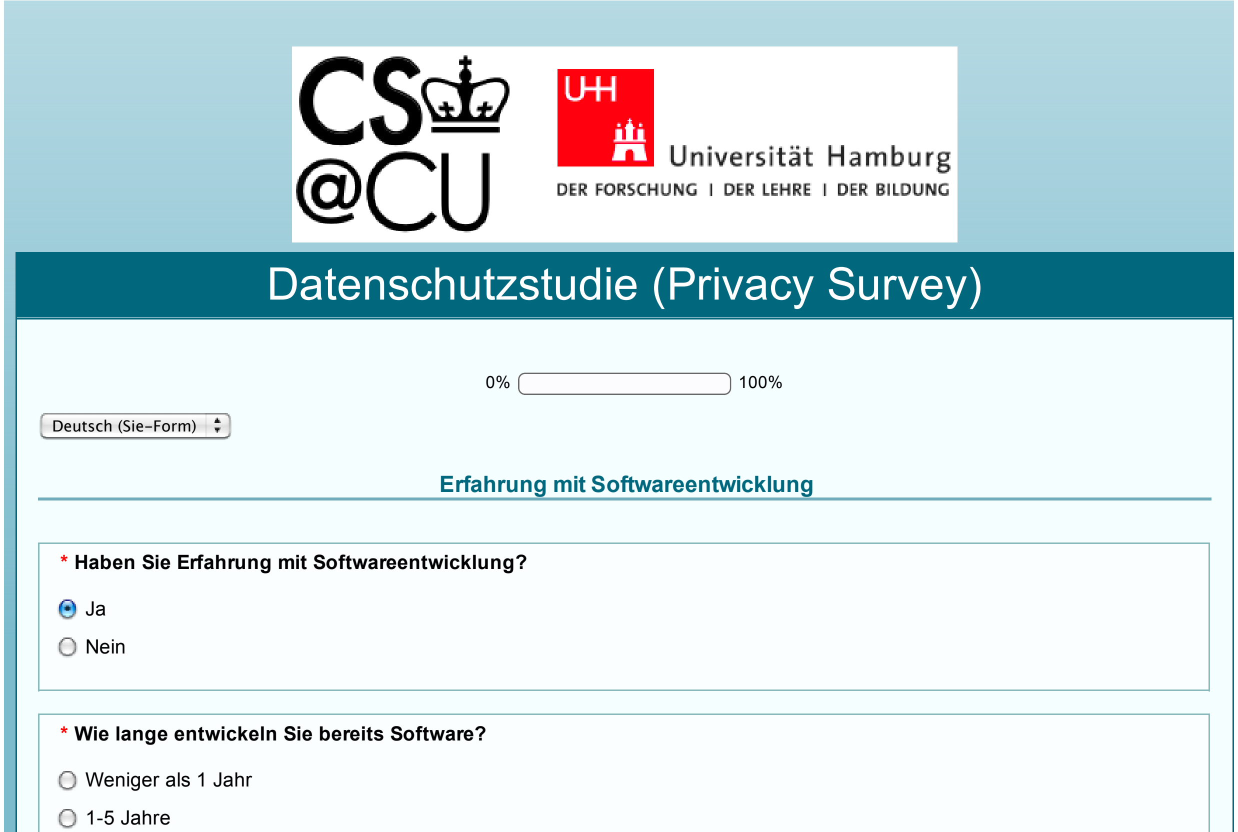
Die Beantwortung der Fragen dauert ca. 5-10 min. Wir danken Ihnen für Ihre Zeit. Ihre Antworten werden nicht weitergegeben und nur zum Zwecke dieser Studie verwendet.

Unter den Teilnehmern dieser Studie werden zwei iPad Minis verlost.

Wenn Sie noch Fragen haben wenden Sie sich bitte an:

Swapneel Sheth (swapneel@cs.columbia.edu)

Make it short (max 15 minutes)!



Datenschutzstudie (Privacy Survey)

0% 100%

Deutsch (Sie-Form)

Erfahrung mit Softwareentwicklung

* **Haben Sie Erfahrung mit Softwareentwicklung?**

● Ja

○ Nein

* **Wie lange entwickeln Sie bereits Software?**

○ Weniger als 1 Jahr

○ 1-5 Jahre

Perfection your questions!

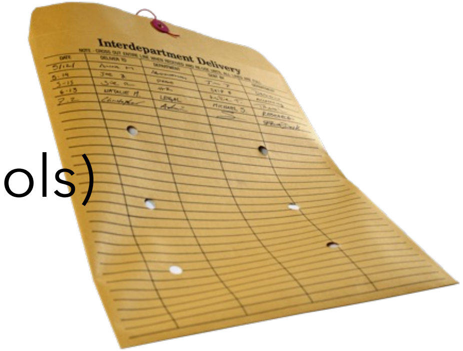
1. Remove unclear questions!
2. Put the least important last!
3. Match questions with answers
4. Think about the outliers

5. Exclude non-serious subjects!

- Filter incomplete answers?
- Use “check” questions
- Remove "noise answers”
- Random order of the questions and answers
- ….

6. Carefully think about incentives

1. Share results (information and tools)



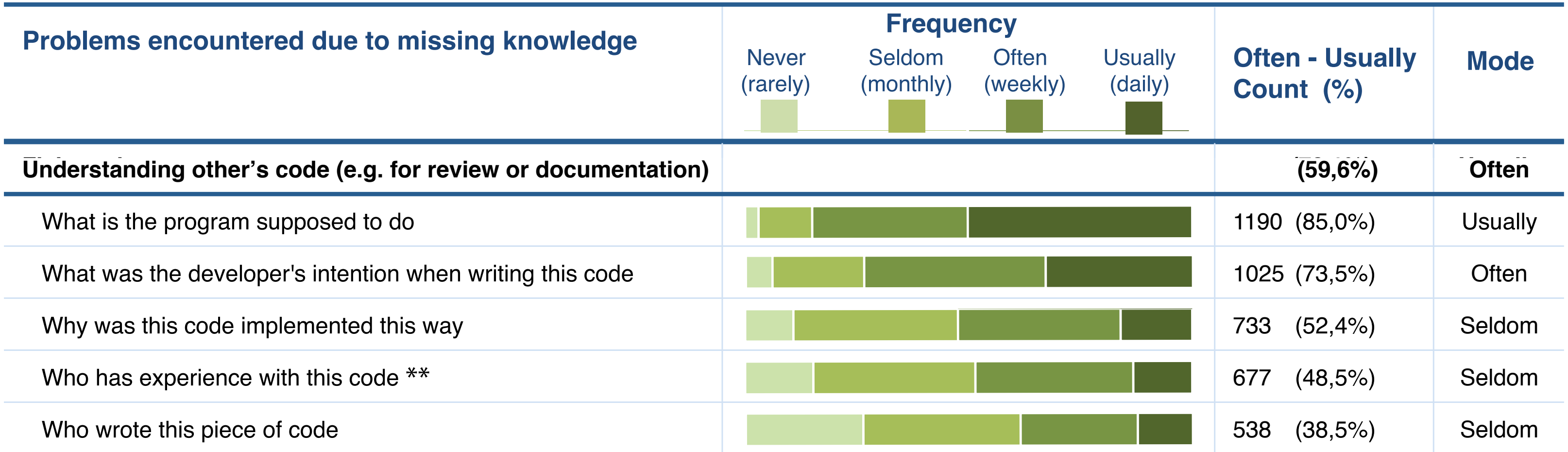
2. Raffle gifts

3. Offer dedicated analysis

4. Show the importance of your research

Use Likert or semantic scales!

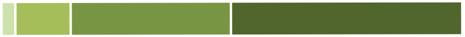
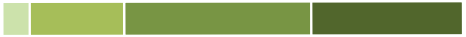
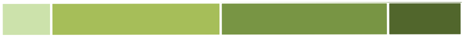
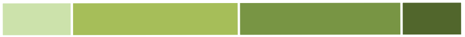
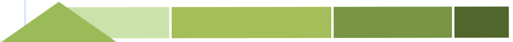
When I am trying to understand other's code ***I need to know…***	Never/ Rarely	Seldom ≈ monthly	Often ≈ weekly	Usually ≈ daily	I don't know
What was the coder's intention as he wrote this	○	○	○	○	○

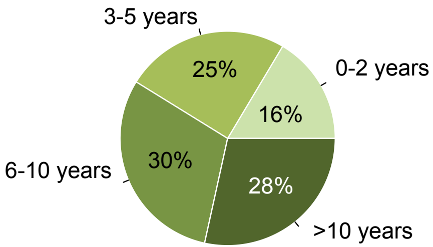


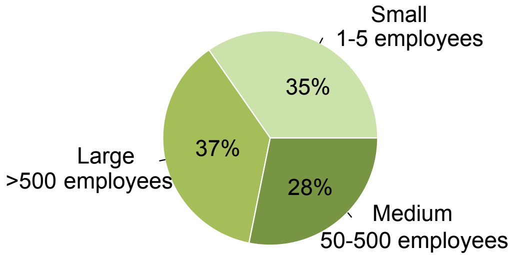
Problems encountered due to missing knowledge	**Frequency** Never (rarely) 🟩 Seldom (monthly) 🟩 Often (weekly) 🟩 Usually (daily) 🟩	**Often - Usually Count (%)**	**Mode**
Understanding other's code (e.g. for review or documentation)		**(59,6%)**	**Often**
What is the program supposed to do		1190 (85,0%)	Usually
What was the developer's intention when writing this code		1025 (73,5%)	Often
Why was this code implemented this way		733 (52,4%)	Seldom
Who has experience with this code **		677 (48,5%)	Seldom
Who wrote this piece of code		538 (38,5%)	Seldom

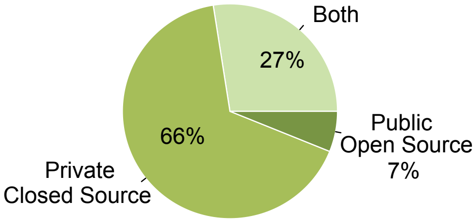
Run statistical tests to remove random results!

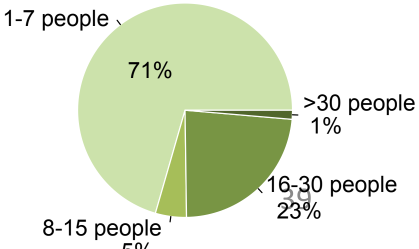
9. Focus on quasi-experimentation instead of summative statistics!

Problems encountered due to missing knowledge	Frequency Never (rarely) 🟩 · Seldom (monthly) 🟩 · Often (weekly) 🟩 · Usually (daily) 🟩	Often - Usually Count (%)	Mode
Understanding other's code (e.g. for review or documentation)		**(59,6%)**	**Often**
What is the program supposed to do		1190 (85,0%)	Usually
What was the developer's intention when writing this code		1025 (73,5%)	Often
Why was this code implemented this way		733 (52,4%)	Seldom
Who has experience with this code **		677 (48,5%)	Seldom
Who wrote this piece of code		538 (38,5%)	Seldom









Observation

Is rather…

- Objective
- Quali-/quantitative
- Exploratory

With users and >1 researchers!

OBSERVATION GUIDELINES [..]

1. Observe less but in realistic environment

How many subjects do we need?

2. Use an observation template!

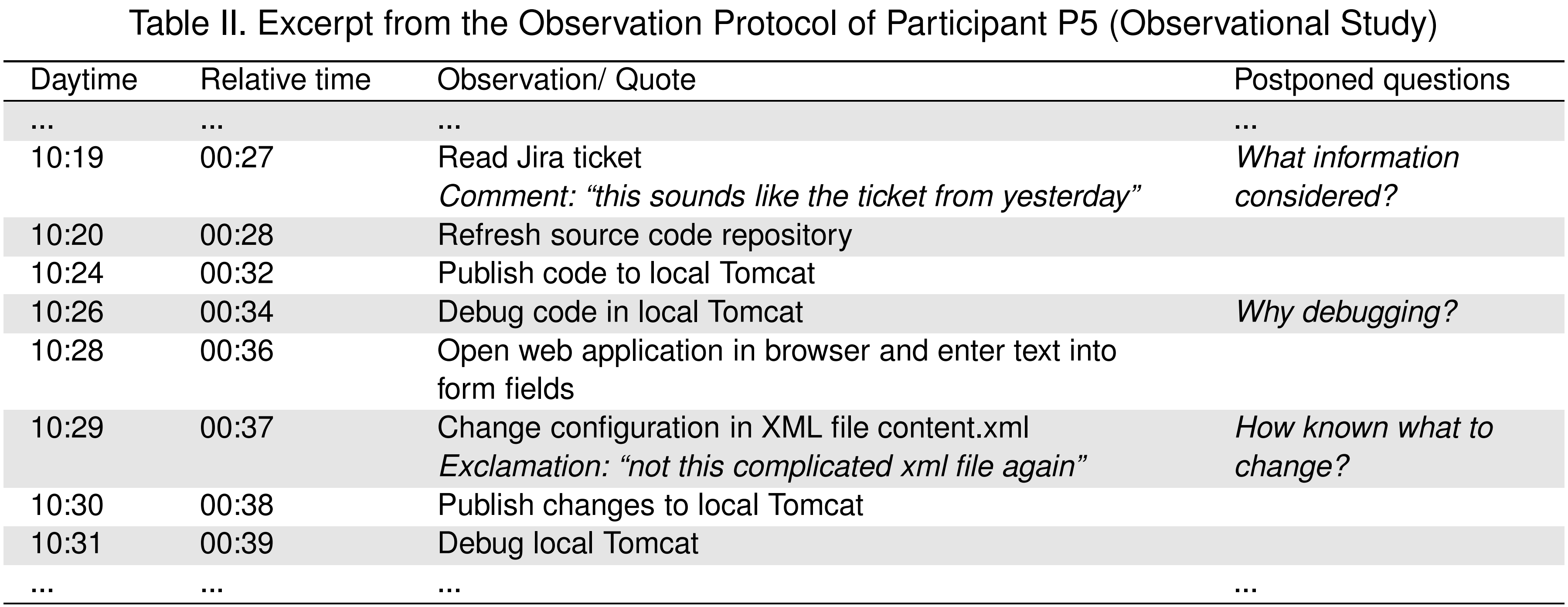


Table II. Excerpt from the Observation Protocol of Participant P5 (Observational Study)

Daytime	Relative time	Observation/ Quote	Postponed questions
...	...	...	...
10:19	00:27	Read Jira ticket *Comment: “this sounds like the ticket from yesterday”*	*What information considered?*
10:20	00:28	Refresh source code repository	
10:24	00:32	Publish code to local Tomcat	
10:26	00:34	Debug code in local Tomcat	*Why debugging?*
10:28	00:36	Open web application in browser and enter text into form fields	
10:29	00:37	Change configuration in XML file content.xml *Exclamation: “not this complicated xml file again”*	*How known what to change?*
10:30	00:38	Publish changes to local Tomcat	
10:31	00:39	Debug local Tomcat	
...	...	...	...

Prepare codes for observations!

3. Report only what you observe at least twice!

4. Talk about your observation in peer debriefing

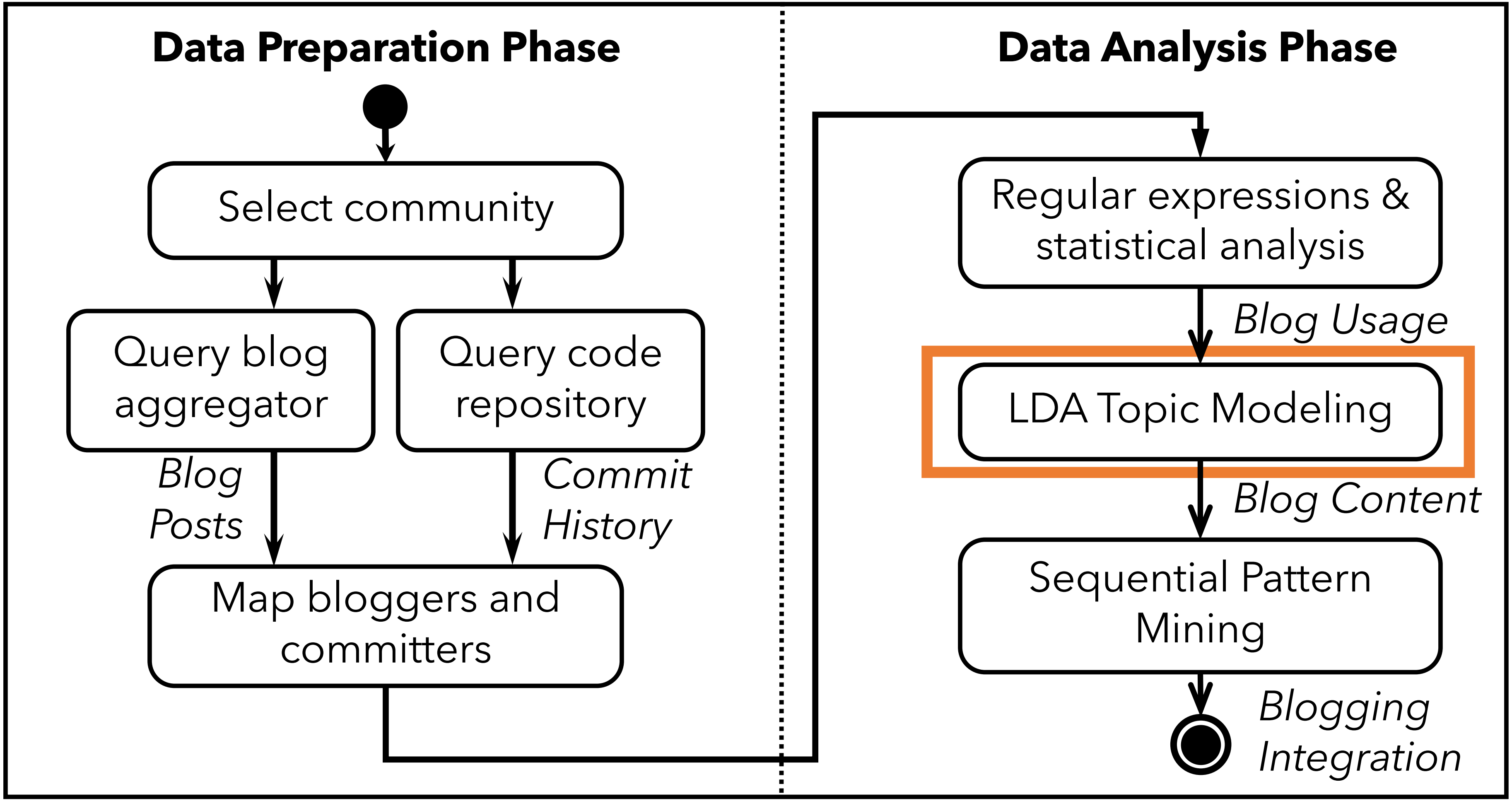
- This helps to identify the relevant observations and to group observation
- Avoid talking to subjects during observation

MASTER AND COMBINE EMPIRICAL METHODS FOR IMPACT

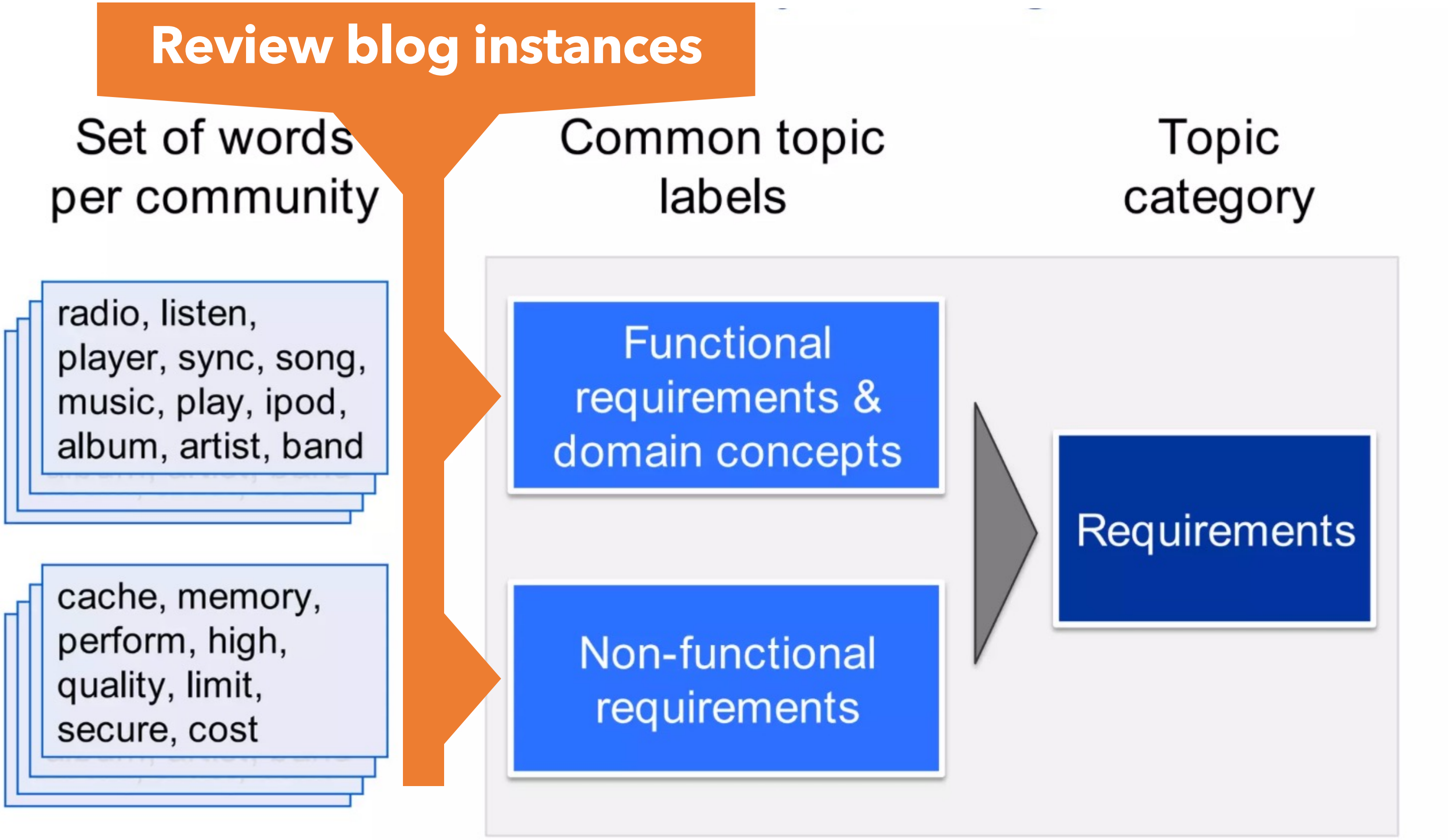
Some examples from my research career so far

Mining Informal Knowledge (Developers’ Blogs and App Reviews)

[MSR’11, EMSE’13, MSR’14, RE’15, RE’15…]


Data Preparation Phase
Data Analysis Phase
Select community
Query blog aggregator
Query code repository
Blog Posts
Commit History
Map bloggers and committers
Regular expressions & statistical analysis
Blog Usage
LDA Topic Modeling
Blog Content
Sequential Pattern Mining
Blogging Integration

Method to analyze the blog content



Review blog instances

Set of words per community

radio, listen, player, sync, song, music, play, ipod, album, artist, band

cache, memory, perform, high, quality, limit, secure, cost

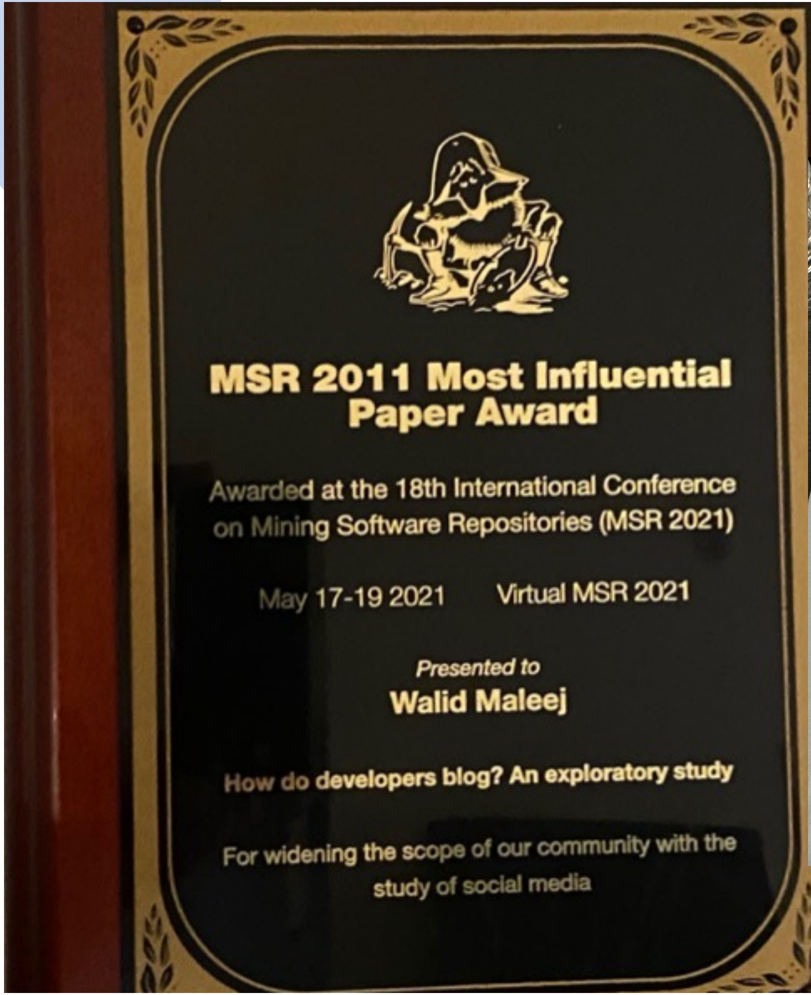
Common topic labels

Functional requirements & domain concepts

Non-functional requirements

Topic category

Requirements



MSR 2011 Most Influential Paper Award

Awarded at the 18th International Conference on Mining Software Repositories (MSR 2021)

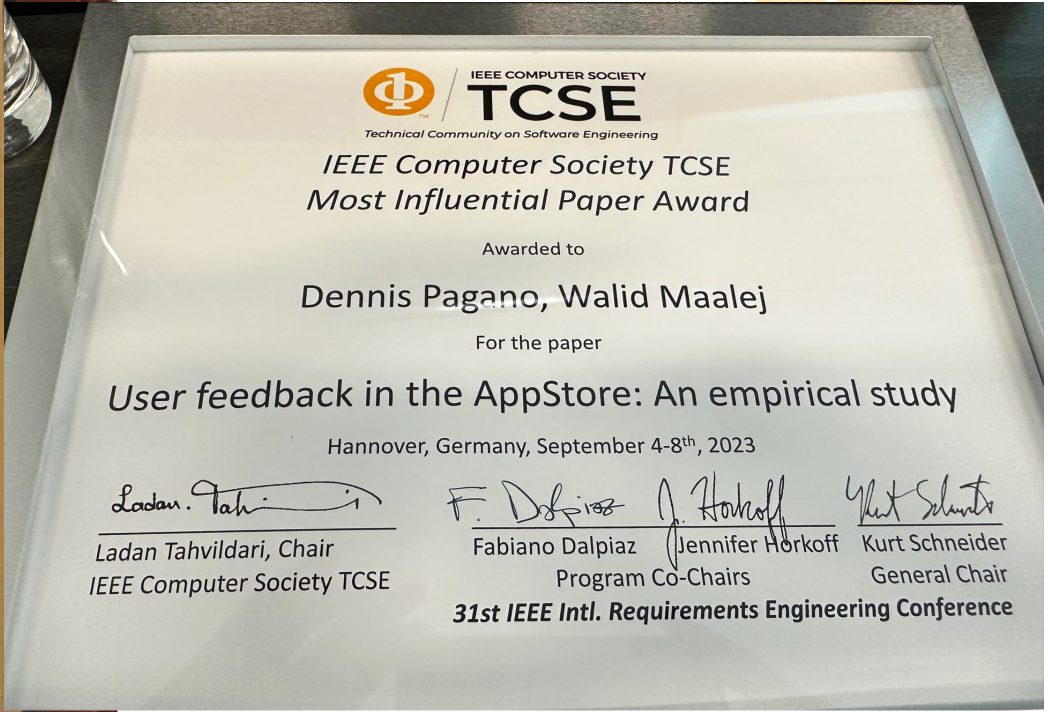
May 17-19 2021   Virtual MSR 2021

Presented to

Walid Maleej

How do developers blog? An exploratory study

For widening the scope of our community with the study of social media



IEEE COMPUTER SOCIETY
TCSE
Technical Community on Software Engineering

IEEE Computer Society TCSE
Most Influential Paper Award

Awarded to

Dennis Pagano, Walid Maalej

For the paper

User feedback in the AppStore: An empirical study

Hannover, Germany, September 4-8th, 2023

Ladan. Tah

Ladan Tahvildari, Chair
IEEE Computer Society TCSE

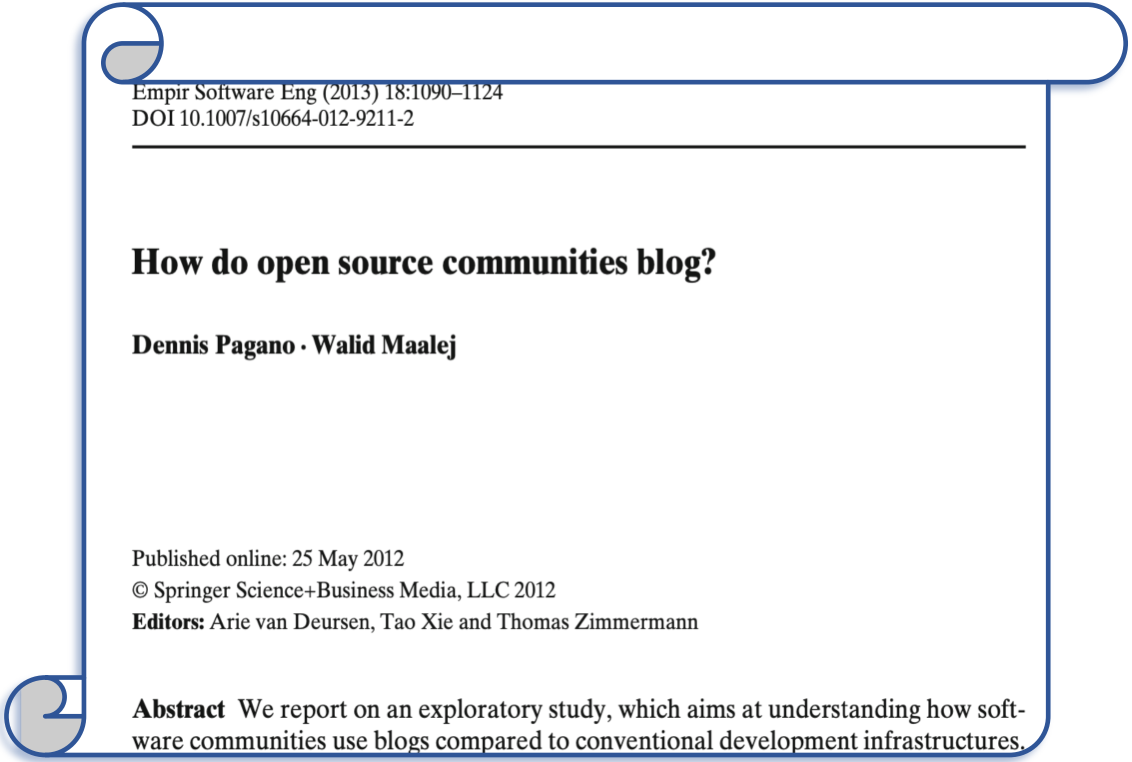
F. Dalpiaz   J. Horkoff

Fabiano Dalpiaz   Jennifer Horkoff
Program Co-Chairs

Kurt Schneider
General Chair

31st IEEE Intl. Requirements Engineering Conference

Read more…



Empir Software Eng (2013) 18:1090–1124
DOI 10.1007/s10664-012-9211-2

How do open source communities blog?

Dennis Pagano · Walid Maalej

Published online: 25 May 2012
© Springer Science+Business Media, LLC 2012
Editors: Arie van Deursen, Tao Xie and Thomas Zimmermann

Abstract We report on an exploratory study, which aims at understanding how software communities use blogs compared to conventional development infrastructures.

doi.org/10.1145/1985441.1985461

Image created with Stable Diffusion 2.1

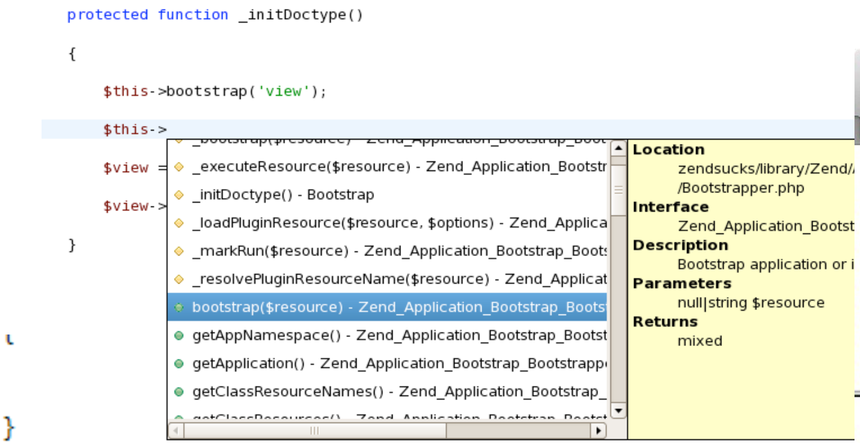
```
rror_mod.use_x = False        MIRROR_Y":
irror_mod.use_y = True
irror_mod.use_z = False
 operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

election at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.acti
("Selected" + str(modifie
irror_ob.select = 0
 bpy.context.selected_ob
ata.objects[one.name].se

int("please select exactly

- OPERATOR CLASSES ----
```

doi.org/10.1109/TSE.2013.12

Mining API Documentation

[TSE’13, MSR’14, FSE’19, ICSME’22]

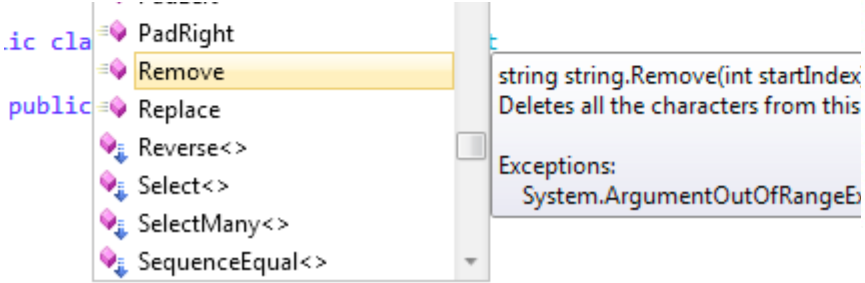
API reference documentation is an important source of knowledge



```
protected function _initDoctype()
{
    $this->bootstrap('view');
    $this->
    $view =
    $view->
}
```

_executeResource($resource) - Zend_Application_Bootstr
_initDoctype() - Bootstrap
_loadPluginResource($resource, $options) - Zend_Applica
_markRun($resource) - Zend_Application_Bootstrap_Boots
_resolvePluginResourceName($resource) - Zend_Applicat
bootstrap($resource) - Zend_Application_Bootstrap_Boots
getAppNamespace() - Zend_Application_Bootstrap_Bootst
getApplication() - Zend_Application_Bootstrap_Bootstrapp
getClassResourceNames() - Zend_Application_Bootstrap_

Location
zendsucks/library/Zend/
/Bootstrapper.php
Interface
Zend_Application_Bootst
Description
Bootstrap application or i
Parameters
null|string $resource
Returns
mixed

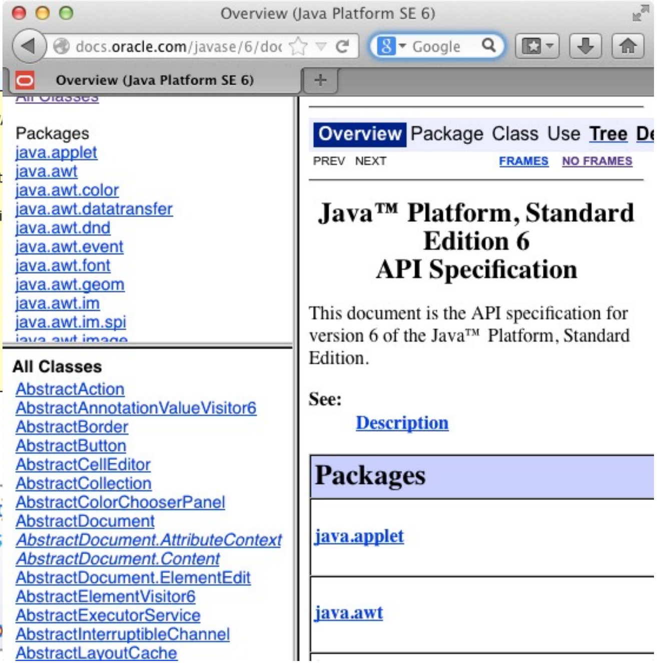


```
.ic cla
public
```

PadRight
Remove
Replace
Reverse<>
Select<>
SelectMany<>
SequenceEqual<>

string string.Remove(int startIndex
Deletes all the characters from this

Exceptions:
System.ArgumentOutOfRangeE



Overview (Java Platform SE 6)

docs.oracle.com/javase/6/doc

Overview (Java Platform SE 6)

Packages
java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font
java.awt.geom
java.awt.im
java.awt.im.spi

All Classes
AbstractAction
AbstractAnnotationValueVisitor6
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeContext
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractElementVisitor6
AbstractExecutorService
AbstractInterruptibleChannel
AbstractLayoutCache

Overview Package Class Use Tree De
PREV NEXT FRAMES NO FRAMES

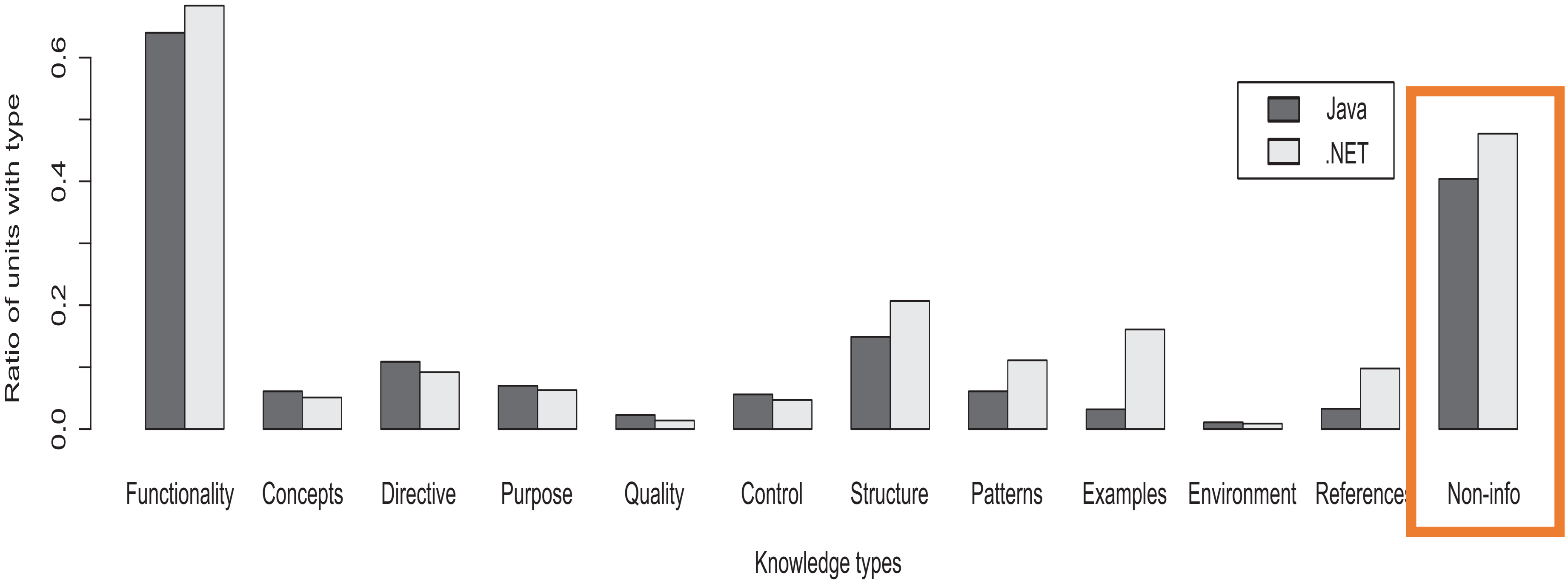
Java™ Platform, Standard Edition 6 API Specification

This document is the API specification for version 6 of the Java™ Platform, Standard Edition.

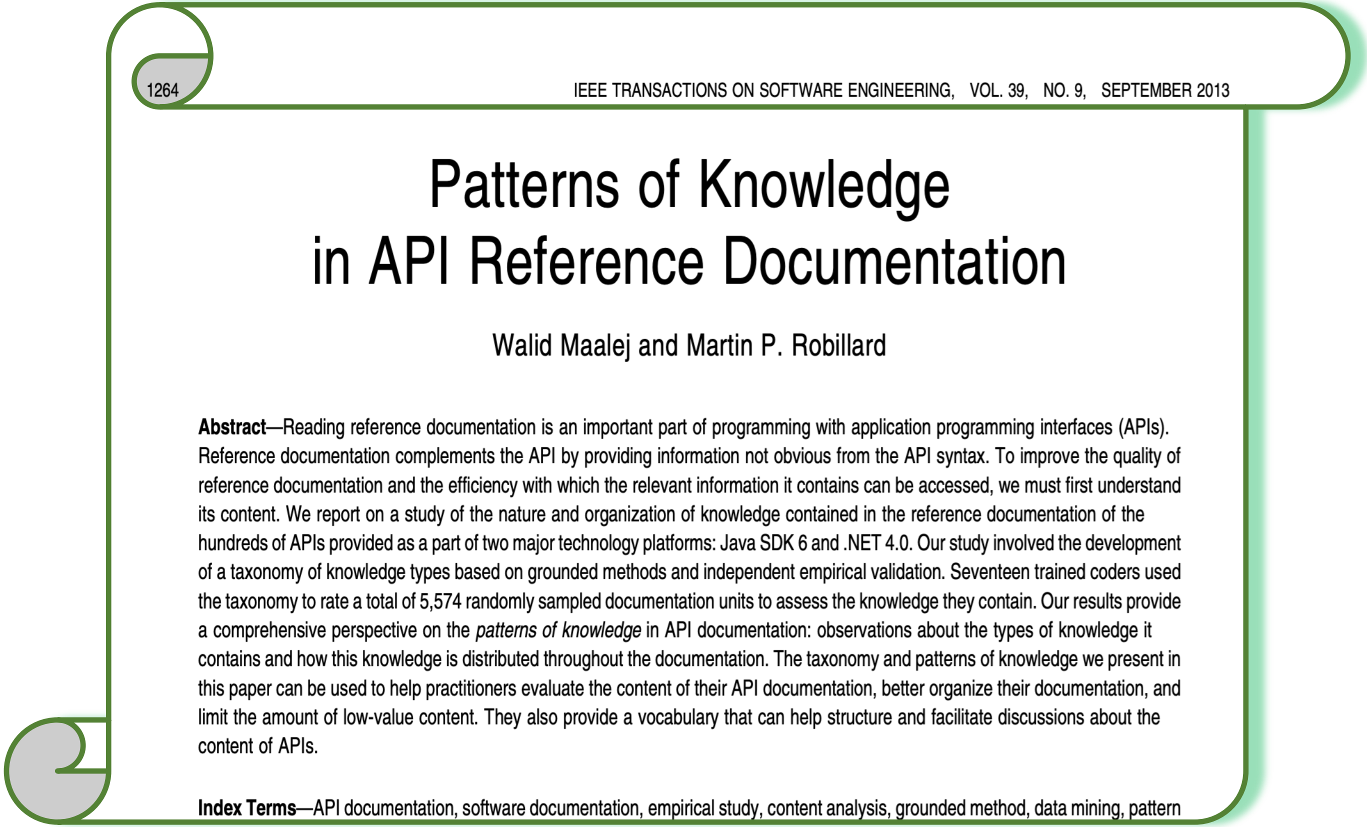
See:
Description

Packages
java.applet
java.awt

Proportion of knowledge type by documentation unit



Read more…



1264 IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 39, NO. 9, SEPTEMBER 2013

Patterns of Knowledge in API Reference Documentation

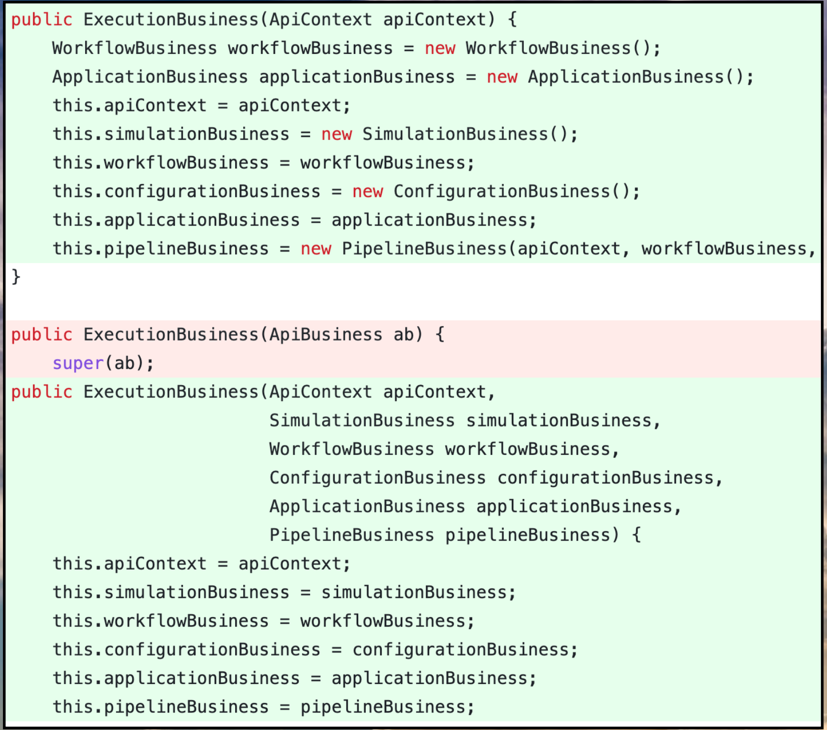
Walid Maalej and Martin P. Robillard

Abstract—Reading reference documentation is an important part of programming with application programming interfaces (APIs). Reference documentation complements the API by providing information not obvious from the API syntax. To improve the quality of reference documentation and the efficiency with which the relevant information it contains can be accessed, we must first understand its content. We report on a study of the nature and organization of knowledge contained in the reference documentation of the hundreds of APIs provided as a part of two major technology platforms: Java SDK 6 and .NET 4.0. Our study involved the development of a taxonomy of knowledge types based on grounded methods and independent empirical validation. Seventeen trained coders used the taxonomy to rate a total of 5,574 randomly sampled documentation units to assess the knowledge they contain. Our results provide a comprehensive perspective on the *patterns of knowledge* in API documentation: observations about the types of knowledge it contains and how this knowledge is distributed throughout the documentation. The taxonomy and patterns of knowledge we present in this paper can be used to help practitioners evaluate the content of their API documentation, better organize their documentation, and limit the amount of low-value content. They also provide a vocabulary that can help structure and facilitate discussions about the content of APIs.

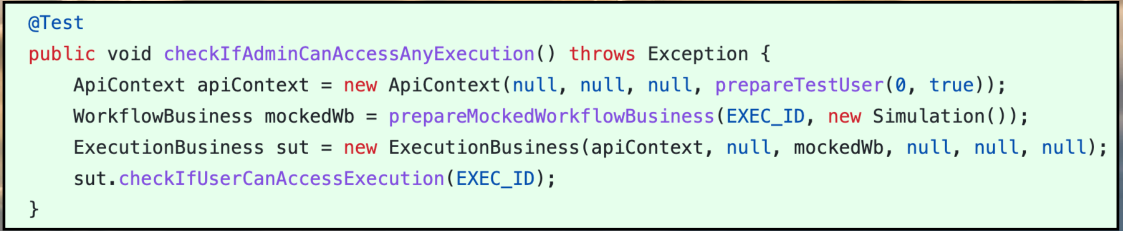
Index Terms—API documentation, software documentation, empirical study, content analysis, grounded method, data mining, pattern

doi.org/10.1109/TSE.2013.12

3. Mining Pull Requests for Testability



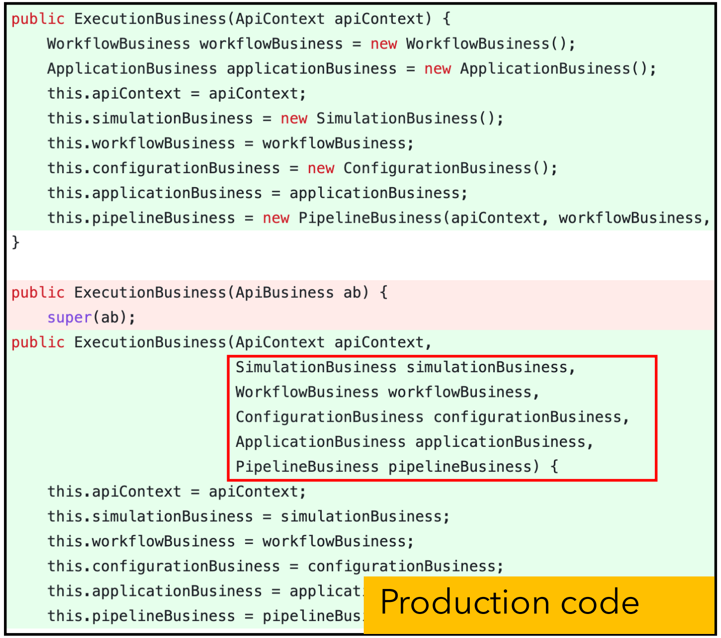
```java
public ExecutionBusiness(ApiContext apiContext) {
    WorkflowBusiness workflowBusiness = new WorkflowBusiness();
    ApplicationBusiness applicationBusiness = new ApplicationBusiness();
    this.apiContext = apiContext;
    this.simulationBusiness = new SimulationBusiness();
    this.workflowBusiness = workflowBusiness;
    this.configurationBusiness = new ConfigurationBusiness();
    this.applicationBusiness = applicationBusiness;
    this.pipelineBusiness = new PipelineBusiness(apiContext, workflowBusiness,
}

public ExecutionBusiness(ApiBusiness ab) {
    super(ab);
public ExecutionBusiness(ApiContext apiContext,
                         SimulationBusiness simulationBusiness,
                         WorkflowBusiness workflowBusiness,
                         ConfigurationBusiness configurationBusiness,
                         ApplicationBusiness applicationBusiness,
                         PipelineBusiness pipelineBusiness) {
    this.apiContext = apiContext;
    this.simulationBusiness = simulationBusiness;
    this.workflowBusiness = workflowBusiness;
    this.configurationBusiness = configurationBusiness;
    this.applicationBusiness = applicationBusiness;
    this.pipelineBusiness = pipelineBusiness;
```



```java
@Test
public void checkIfAdminCanAccessAnyExecution() throws Exception {
    ApiContext apiContext = new ApiContext(null, null, null, prepareTestUser(0, true));
    WorkflowBusiness mockedWb = prepareMockedWorkflowBusiness(EXEC_ID, new Simulation());
    ExecutionBusiness sut = new ExecutionBusiness(apiContext, null, mockedWb, null, null, null);
    sut.checkIfUserCanAccessExecution(EXEC_ID);
}
```

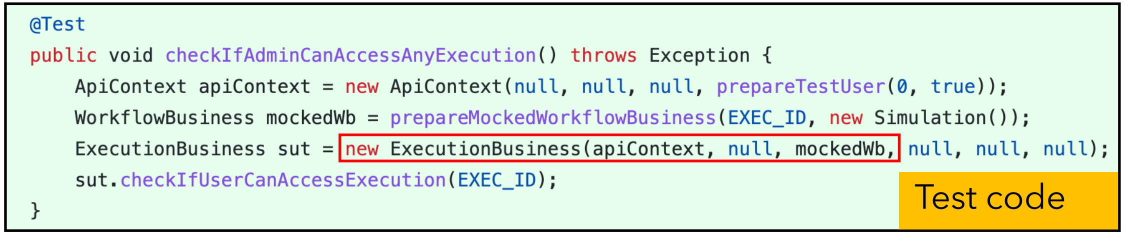
[Reich and Maalej, ICSE’23]

Motivating example: create constructor



```
public ExecutionBusiness(ApiContext apiContext) {
    WorkflowBusiness workflowBusiness = new WorkflowBusiness();
    ApplicationBusiness applicationBusiness = new ApplicationBusiness();
    this.apiContext = apiContext;
    this.simulationBusiness = new SimulationBusiness();
    this.workflowBusiness = workflowBusiness;
    this.configurationBusiness = new ConfigurationBusiness();
    this.applicationBusiness = applicationBusiness;
    this.pipelineBusiness = new PipelineBusiness(apiContext, workflowBusiness,
}

public ExecutionBusiness(ApiBusiness ab) {
    super(ab);
public ExecutionBusiness(ApiContext apiContext,
                         SimulationBusiness simulationBusiness,
                         WorkflowBusiness workflowBusiness,
                         ConfigurationBusiness configurationBusiness,
                         ApplicationBusiness applicationBusiness,
                         PipelineBusiness pipelineBusiness) {
    this.apiContext = apiContext;
    this.simulationBusiness = simulationBusiness;
    this.workflowBusiness = workflowBusiness;
    this.configurationBusiness = configurationBusiness;
    this.applicationBusiness = applicat
    this.pipelineBusiness = pipelineBus
```

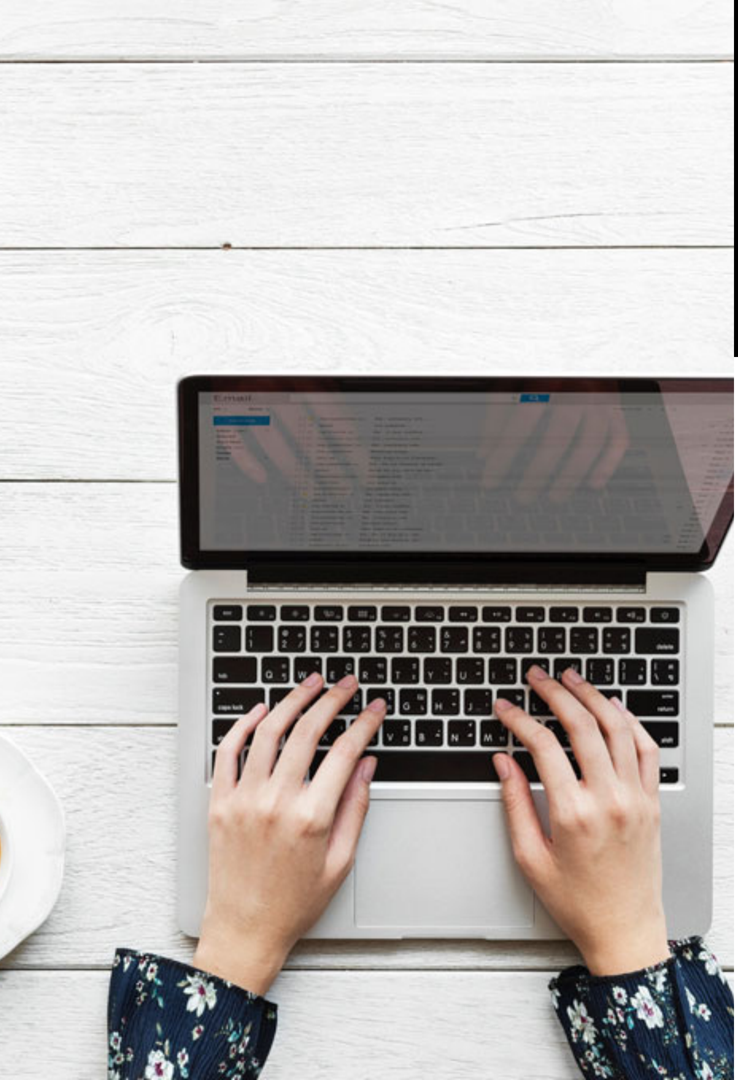
Production code



```
@Test
public void checkIfAdminCanAccessAnyExecution() throws Exception {
    ApiContext apiContext = new ApiContext(null, null, null, prepareTestUser(0, true));
    WorkflowBusiness mockedWb = prepareMockedWorkflowBusiness(EXEC_ID, new Simulation());
    ExecutionBusiness sut = new ExecutionBusiness(apiContext, null, mockedWb, null, null, null);
    sut.checkIfUserCanAccessExecution(EXEC_ID);
}
```

Test code

github/virtual-imaging-platform/pull_request_52

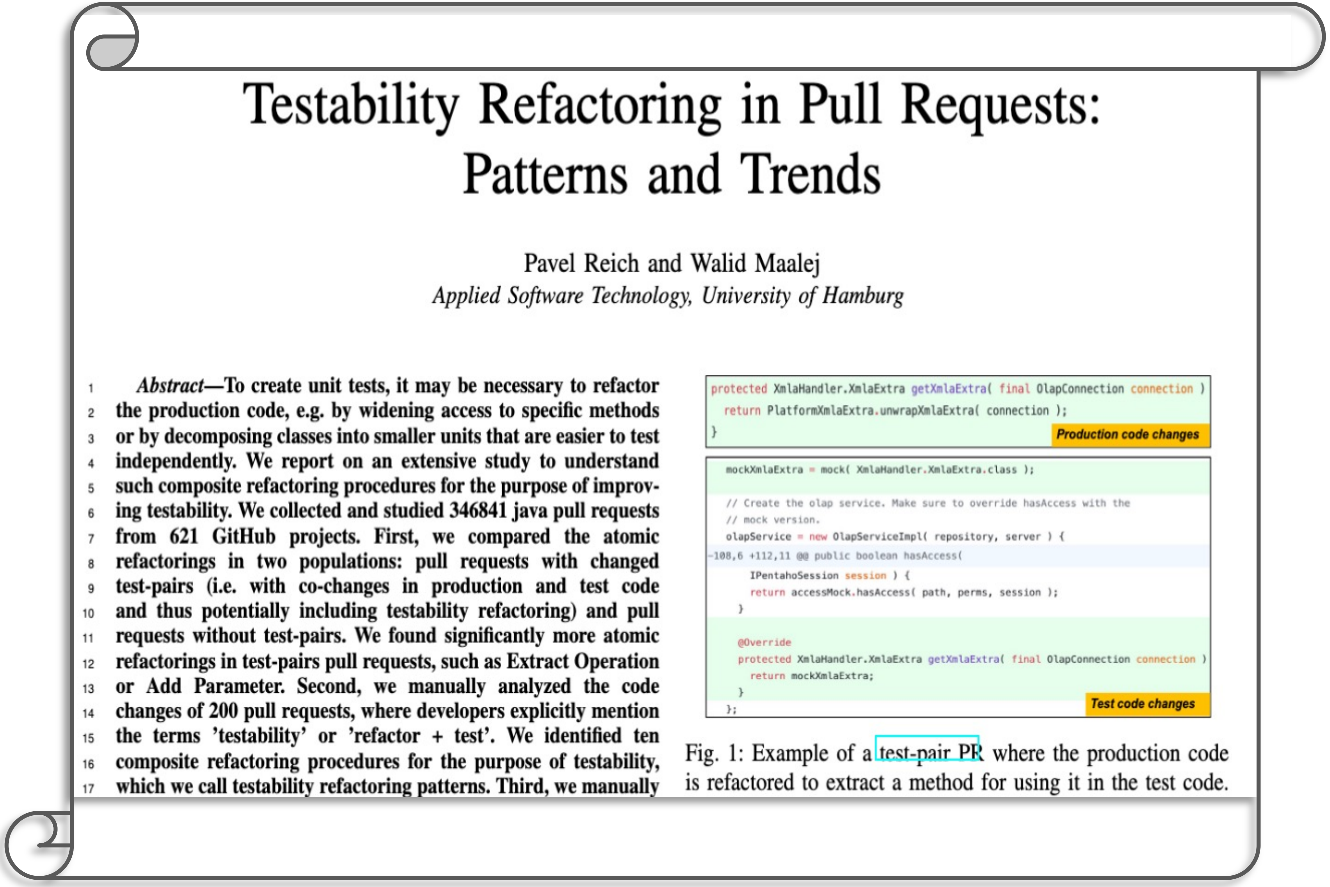


MANUAL ANALYSIS

- PRs categorized as
  - Changes in production code **only** to improve testability
  - Same as above + features/bugfixes (**incl.** refactoring for test)
  - Changes are **irrelevant** for testability (changes, bugfixes, refactorings in test code…)
- Testability-relevant PRs can contain one or more testability refactoring patterns

TESTABILITY REFACTORING PATTERNS IN PRS

Pattern name	#	%
P1. extract_method_for_**override**	51	22.2
P2. extract_method_for_invocation	39	17.0
P3. widen_access_for_invocation	35	15.2
P4. extract_class_for_invocation	29	12.6
P5. add_constructor_param	25	10.9
P6. extract_class_for_override	15	6.5
P7. create_constructor	10	4.3
P8. widen_access_for_override	9	3.9
P9. override_system_time	4	1.7
P10.extract_attribute_for_assertion	3	1.3
Total	230	100



Testability Refactoring in Pull Requests: Patterns and Trends

Pavel Reich and Walid Maalej
Applied Software Technology, University of Hamburg

Abstract—To create unit tests, it may be necessary to refactor the production code, e.g. by widening access to specific methods or by decomposing classes into smaller units that are easier to test independently. We report on an extensive study to understand such composite refactoring procedures for the purpose of improving testability. We collected and studied 346841 java pull requests from 621 GitHub projects. First, we compared the atomic refactorings in two populations: pull requests with changed test-pairs (i.e. with co-changes in production and test code and thus potentially including testability refactoring) and pull requests without test-pairs. We found significantly more atomic refactorings in test-pairs pull requests, such as Extract Operation or Add Parameter. Second, we manually analyzed the code changes of 200 pull requests, where developers explicitly mention the terms 'testability' or 'refactor + test'. We identified ten composite refactoring procedures for the purpose of testability, which we call testability refactoring patterns. Third, we manually

```
protected XmlaHandler.XmlaExtra getXmlaExtra( final OlapConnection connection )
  return PlatformXmlaExtra.unwrapXmlaExtra( connection );
}
```

Production code changes

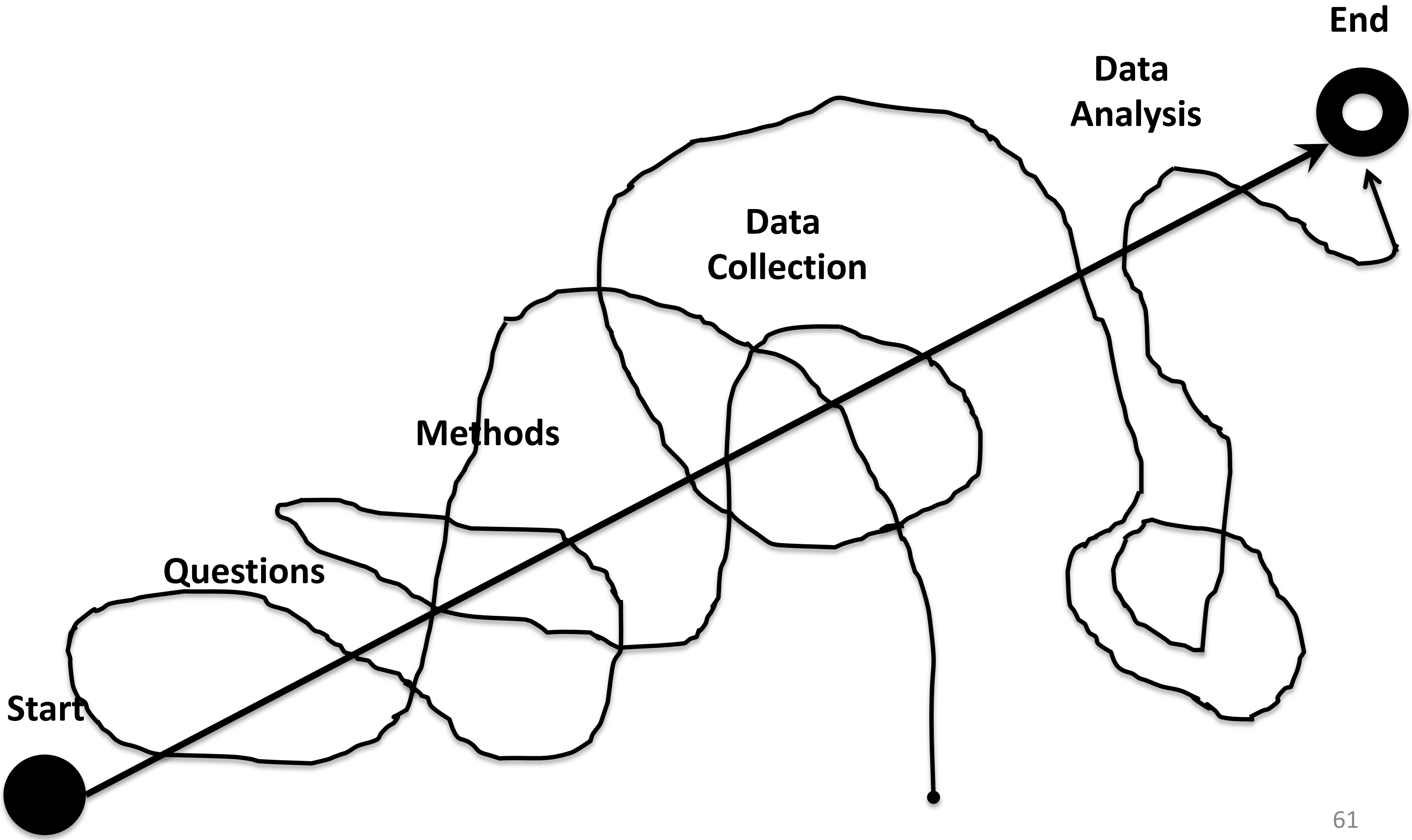
```
  mockXmlaExtra = mock( XmlaHandler.XmlaExtra.class );

  // Create the olap service. Make sure to override hasAccess with the
  // mock version.
  olapService = new OlapServiceImpl( repository, server ) {
-108,6 +112,11 @@ public boolean hasAccess(
      IPentahoSession session ) {
      return accessMock.hasAccess( path, perms, session );
    }

    @Override
    protected XmlaHandler.XmlaExtra getXmlaExtra( final OlapConnection connection )
      return mockXmlaExtra;
    }
  };
```

Test code changes

Fig. 1: Example of a test-pair PR where the production code is refactored to extract a method for using it in the test code.

GitHub Repo

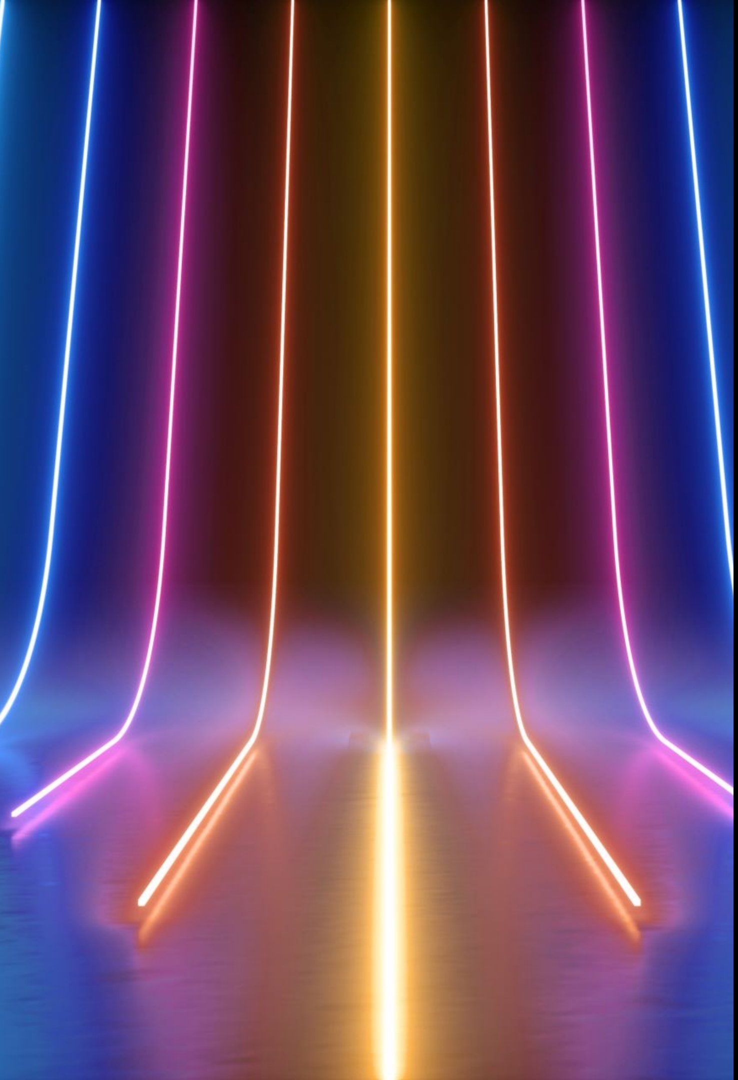
ICSE2023-Paper


End
Data
Analysis
Data
Collection
Methods
Questions
Start

Lessons learned

- Research is not linear
- Data science is not a merely quantitative discipline (but also qualitative)
- Integrating multiple sources of data makes a big difference

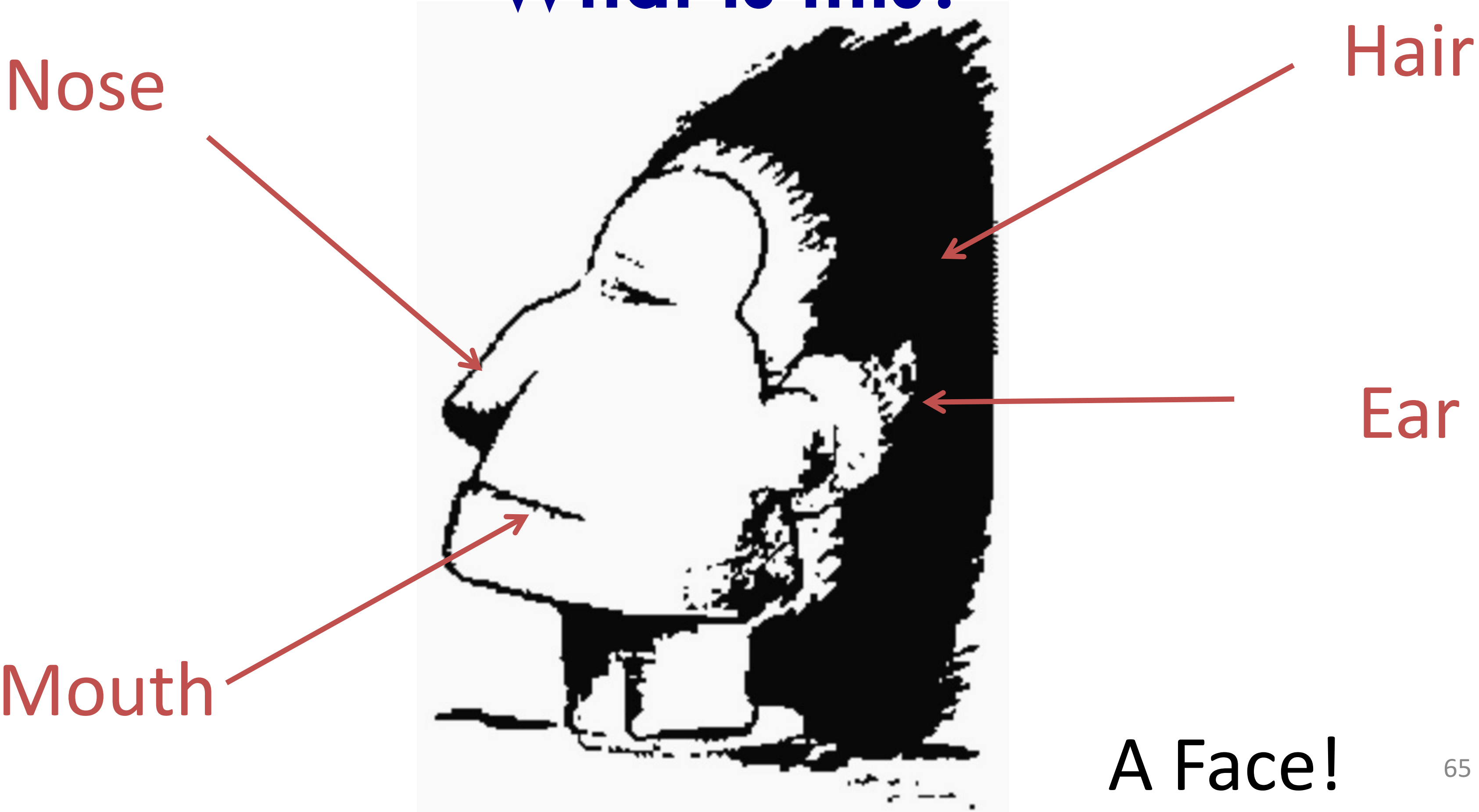
A PhD journey is more than a research challenge



1. IT’S A MARATHON, NOT A SPRINT

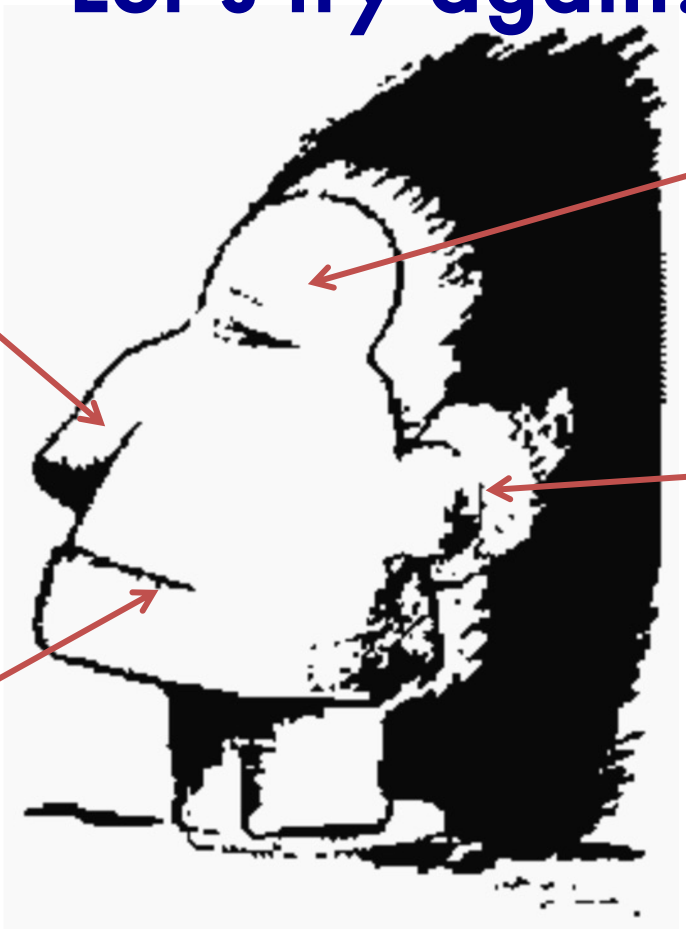
Prepare yourself, be patient, and keep your speed!

What is this?



A Face!

Let's try again!



Elbow

Head

Arm

Coat

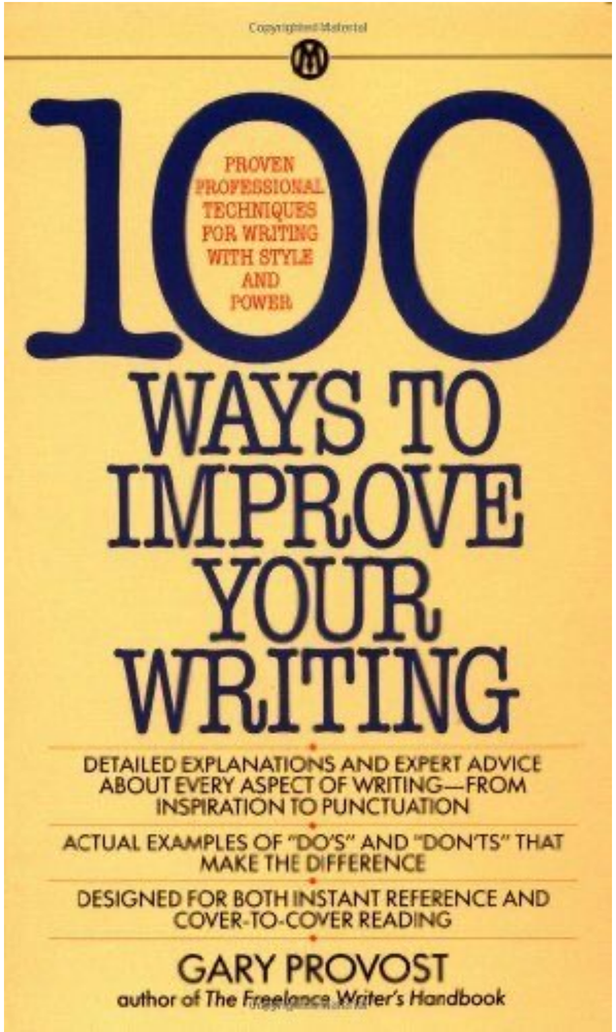
An Inuit
Entering a Cave!

2. IT’S YOUR PHD

Shape it, enjoy it, but know when you are done!

3. CONTENT & PRESENTATION: ABOUT 50/50

Boost your presentation, communication and writing skills!



100 WAYS TO IMPROVE YOUR WRITING

PROVEN PROFESSIONAL TECHNIQUES FOR WRITING WITH STYLE AND POWER

DETAILED EXPLANATIONS AND EXPERT ADVICE ABOUT EVERY ASPECT OF WRITING—FROM INSPIRATION TO PUNCTUATION

ACTUAL EXAMPLES OF "DO'S" AND "DON'TS" THAT MAKE THE DIFFERENCE

DESIGNED FOR BOTH INSTANT REFERENCE AND COVER-TO-COVER READING

GARY PROVOST

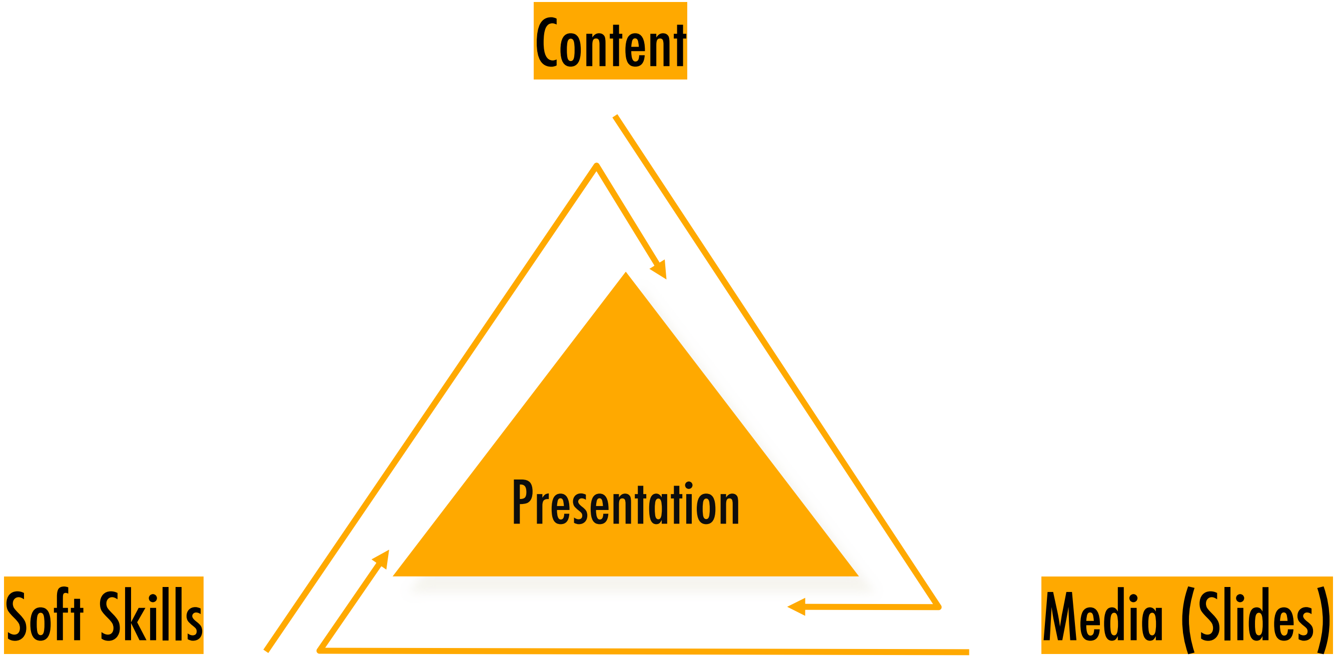
author of *The Freelance Writer's Handbook*

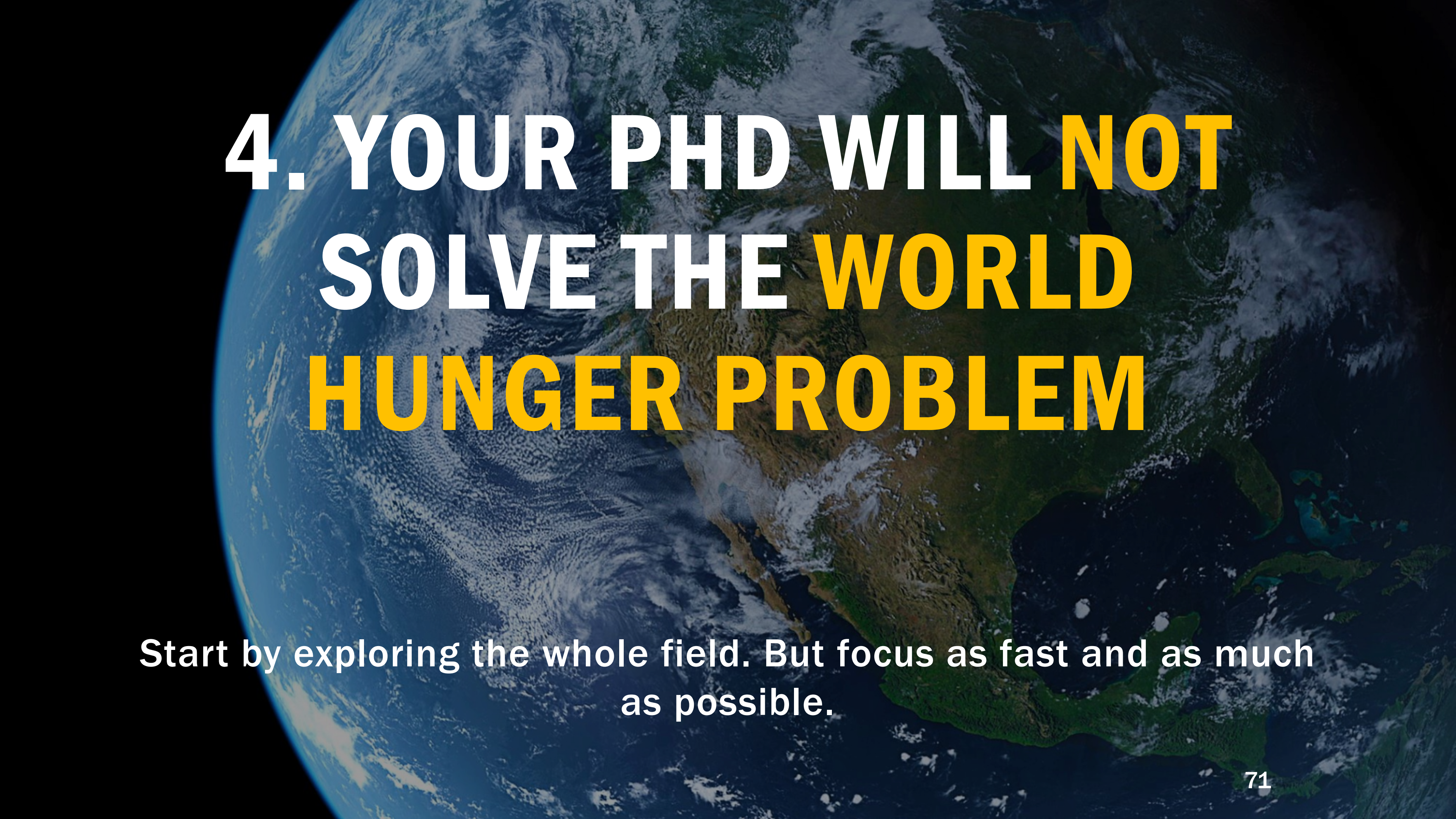
Talk Smart: The Podcast

Smart: Communication Technic

0:00 -2:22

The “golden triangle” of a good presentation





4. YOUR PHD WILL NOT SOLVE THE WORLD HUNGER PROBLEM

Start by exploring the whole field. But focus as fast and as much as possible.

One last thing…

CONNECTING THE DOTS

Trust in yourself and your context, that the dots will connect.

You can't connect the dots looking forward; you can only connect them looking backwards [...]

So, you have to trust that the dots will somehow connect in your future.

Steve Jobs