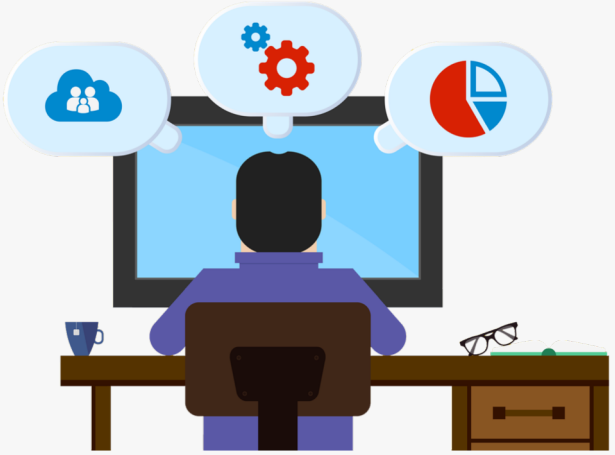


Software Developers' Work Habits and Expertise

Sebastian Baltes

@s_baltes

empirical-software.engineering

THE UNIVERSITY of ADELAIDE

Interaction

My Background


Evidence-based Practice through Practice-based Evidence

Studying Developers' Work Habits

Observe
Describe
Explain

Software Developers' Work Habits

$\Rightarrow$ Expand knowledge

- Derive requirements for better tool support
- Identify possible process improvements
- Communicate findings back to practitioners

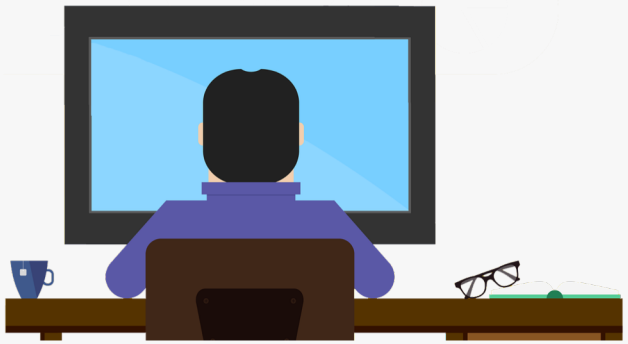
Habits?

A habit is a „**settled tendency or usual manner of behavior**“

https://www.merriam-webster.com/dictionary/habit

Personal habits

Work habits



Studied Habits

Issues in Sampling Software Developers

Methodology

airbnb

Constructing Urban Tourism Space Digitally

Interdisciplinary Research

Expertise Development

2013

Sketching

Code Plagiarism

stackoverflow

Regular Expressions

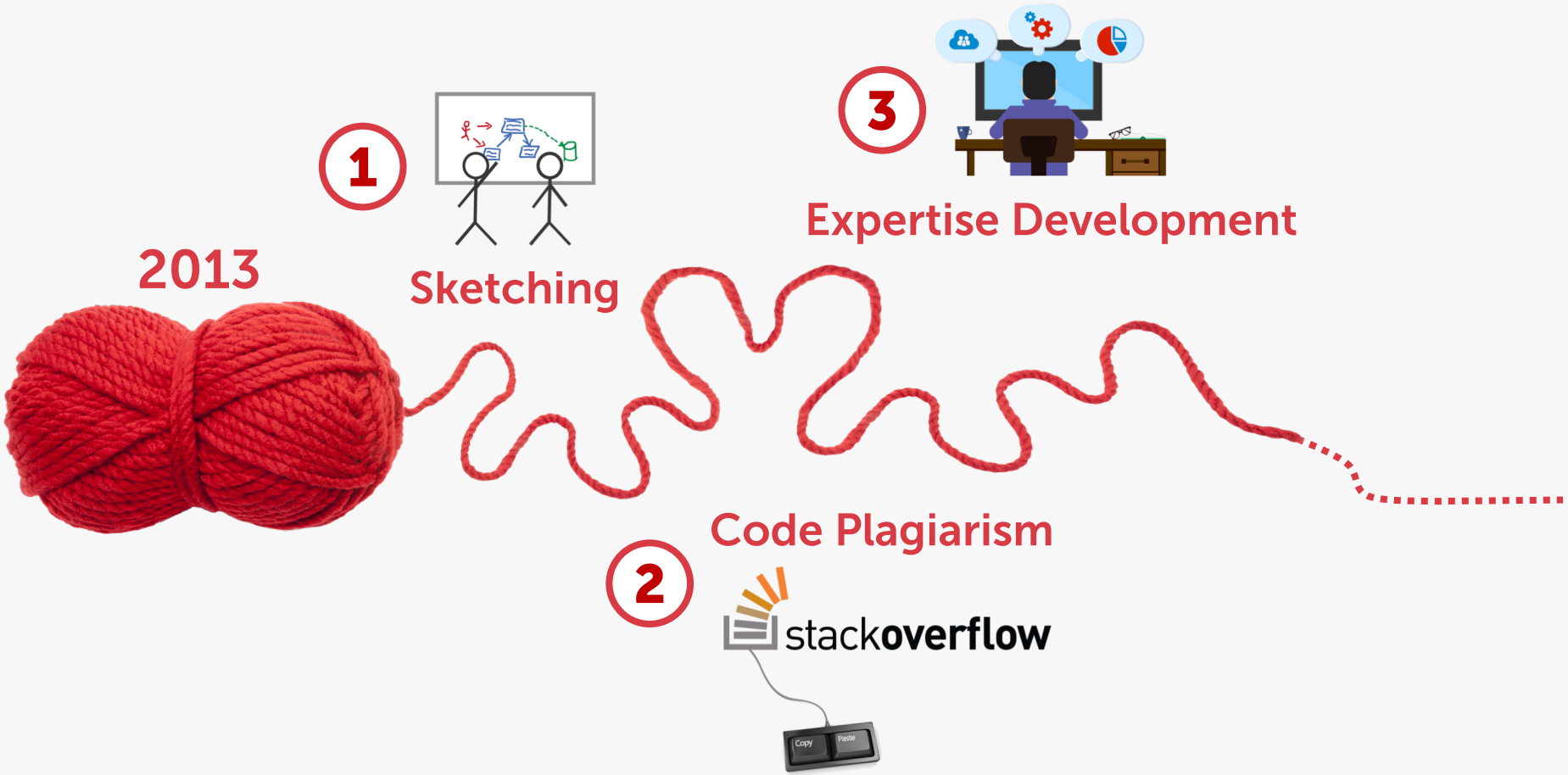
RegViz

Continuous Integration

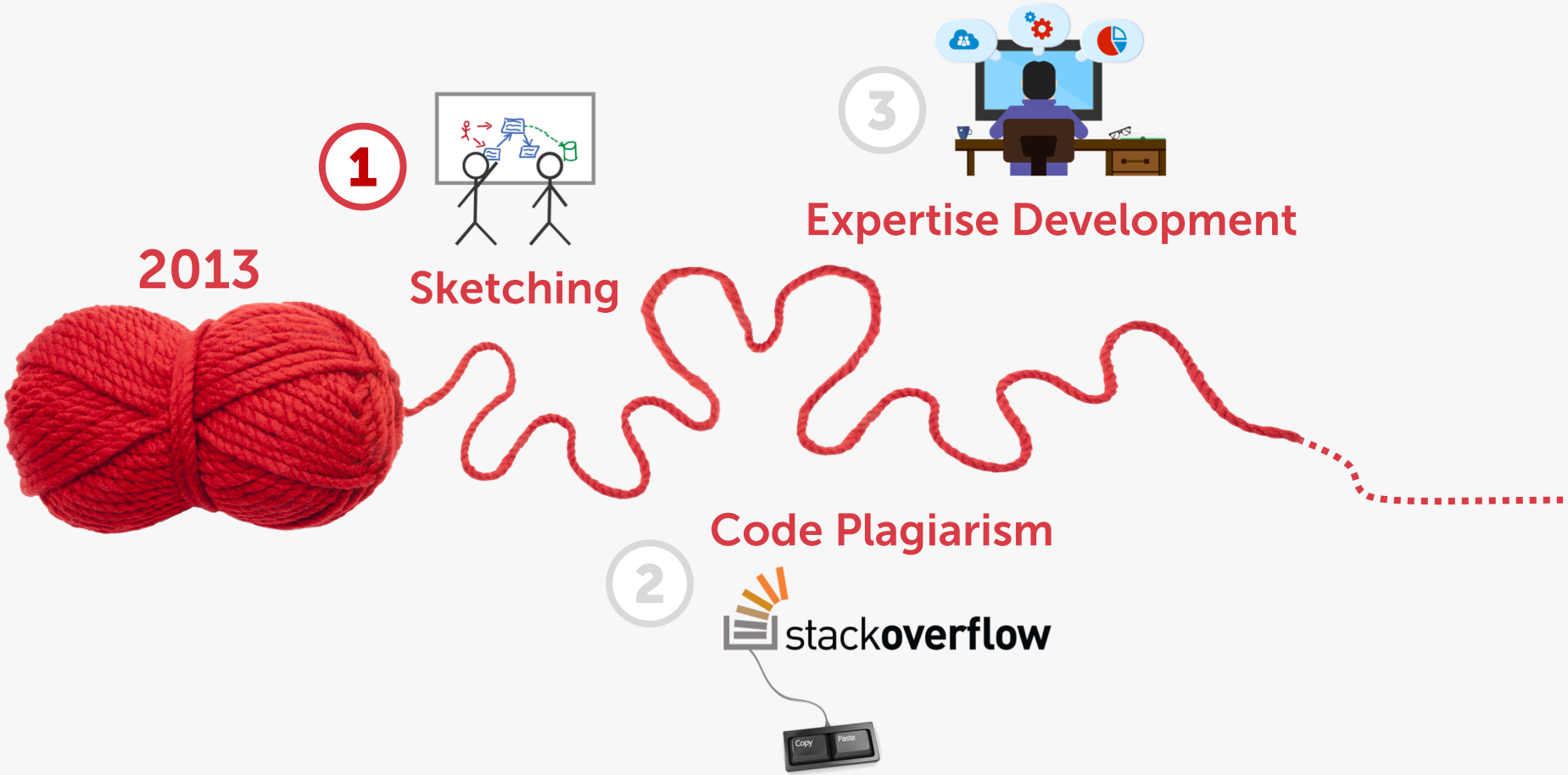
GitHub

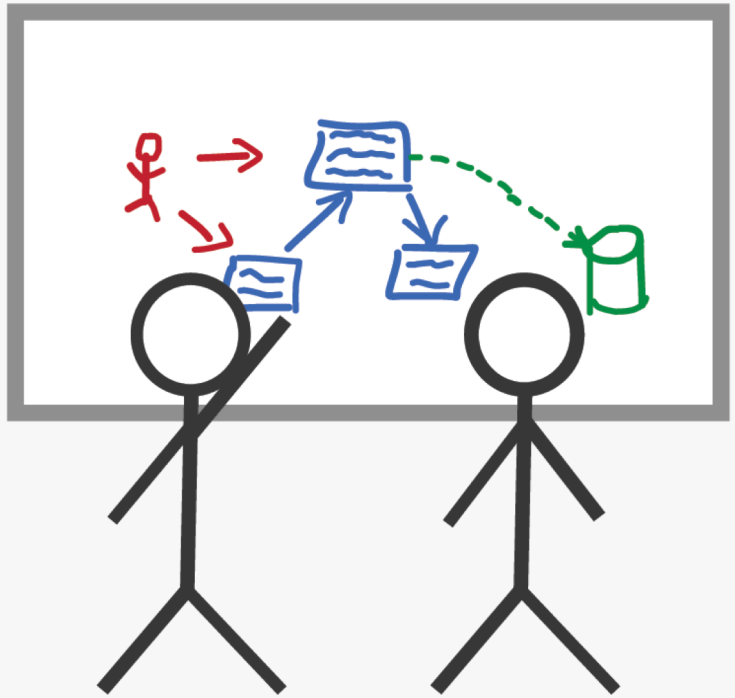
Travis CI

Overview of this Talk



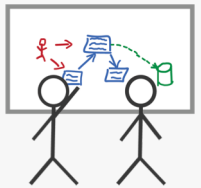
Overview of this Talk





Sketching

Sketching



Research Questions

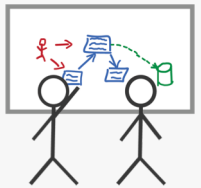
Questions:

How and **why** do software practitioners use sketches and diagrams?
How are they related to **source code**?
How can we provide better **tool support**?

Approach:

Field study, online survey, lab study, formative tool evaluations

Sketching



Sketches and Diagrams in Practice

Sebastian Baltes
Computer Science
University of Trier
Trier, Germany
s.baltes@uni-trier.de

Stephan Diehl
Computer Science
University of Trier
Trier, Germany
diehl@uni-trier.de

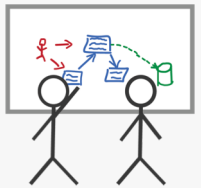
ABSTRACT

Sketches and diagrams play an important role in the daily work of software developers. In this paper, we investigate the use of sketches and diagrams in software engineering practice. To this end, we used both quantitative and qualitative methods. We present the results of an exploratory study in three companies and an online survey with 394 participants. Our participants included software developers, software architects, project managers, consultants, as well as researchers. They worked in different countries and on projects from a wide range of application areas. Most questions in the survey were related to the last sketch or diagram that the participants had created. Contrary to our expectations and previous work, the majority of sketches and

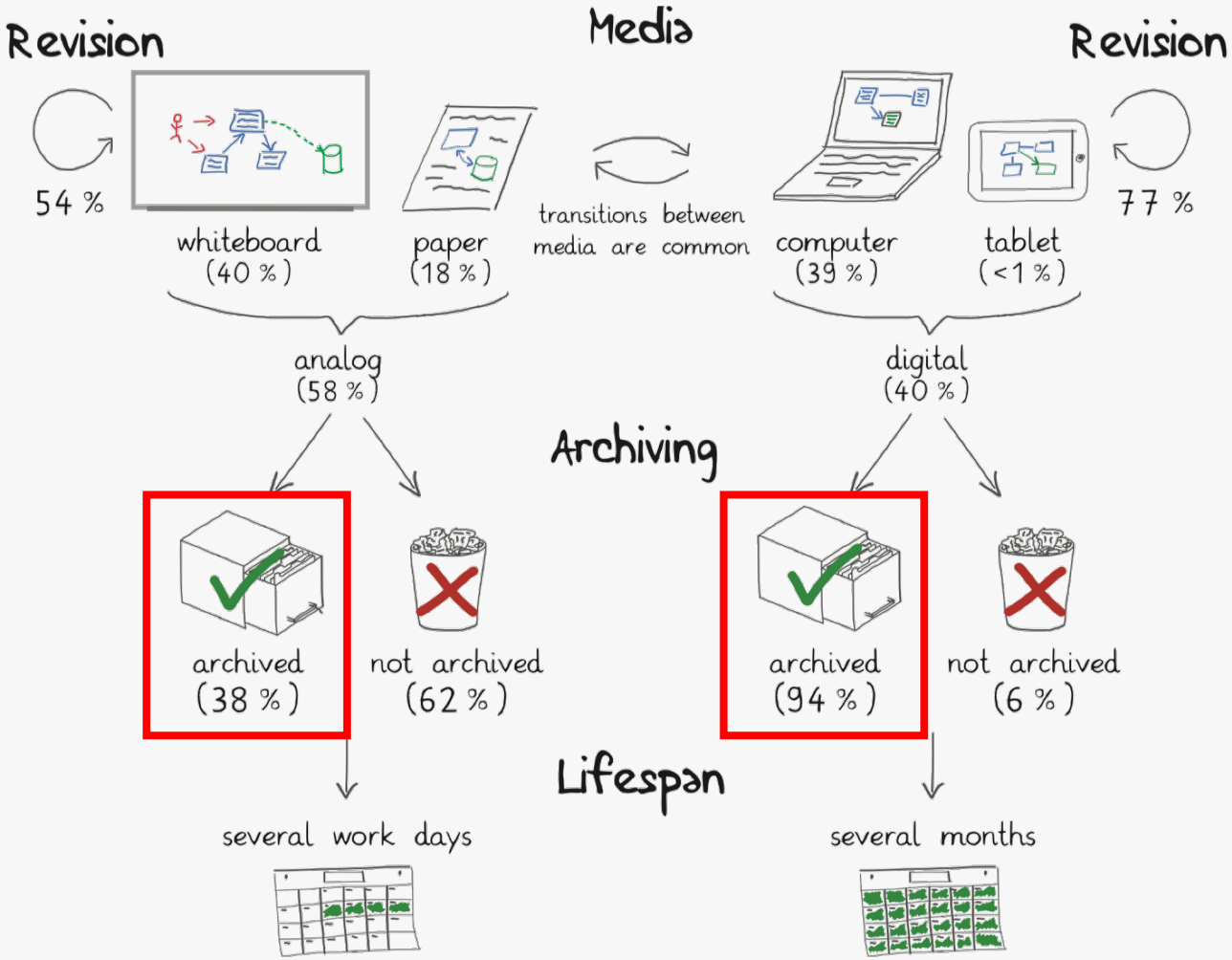
1. INTRODUCTION

Over the past years, studies have shown the importance of sketches and diagrams in software development [6,11,43]. Most of these visual artifacts do not follow formal conventions like the *Unified Modeling Language* (UML), but have an informal, ad-hoc nature [6,11,23,25]. Sketches and diagrams are important because they depict parts of the mental model developers build to understand a software project [21]. They may contain different views, levels of abstraction, formal and informal notations, pictures, or generated parts [6, 11,41,42]. Developers create sketches and diagrams mainly to understand, to design, and to communicate [6]. Media for sketch creation include whiteboards, engineering notebooks, scrap papers, but also software tools like Photoshop

https://empirical-software.engineering/projects/sketches/



Sketches and Diagrams in Practice



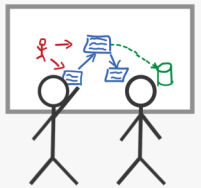
Purpose

Designing (75 %)
Explaining (60 %)
Understanding (56 %)
Analyzing Requirements (45 %)

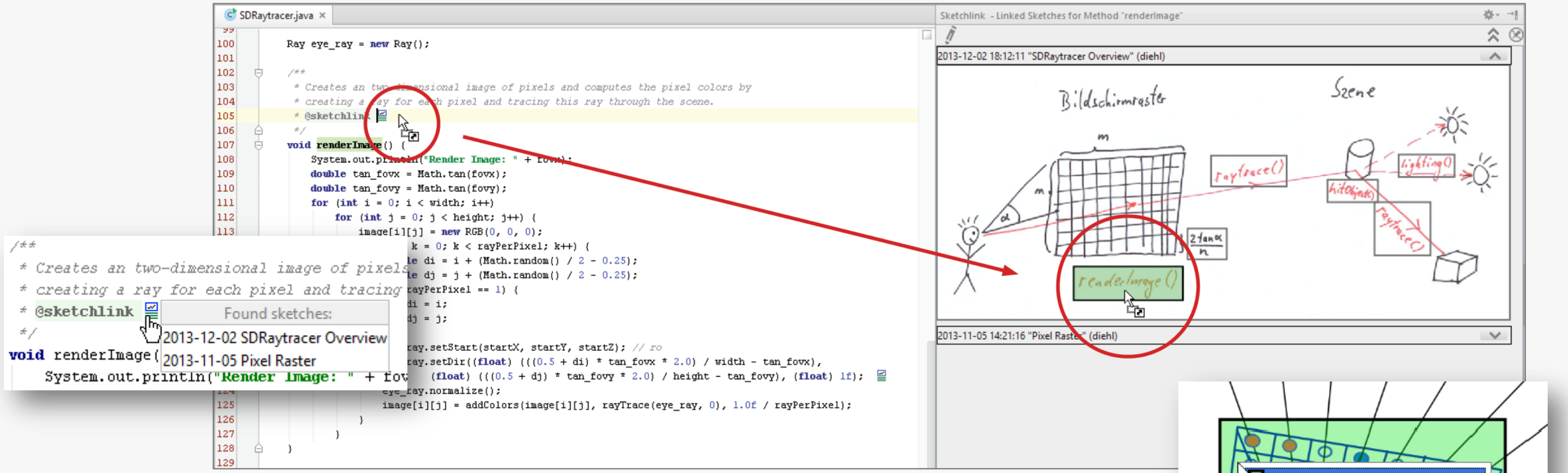
Relation to Source Code

47 % of the sketches are rated as helpful for others to understand the related source code artifacts.

Sketching

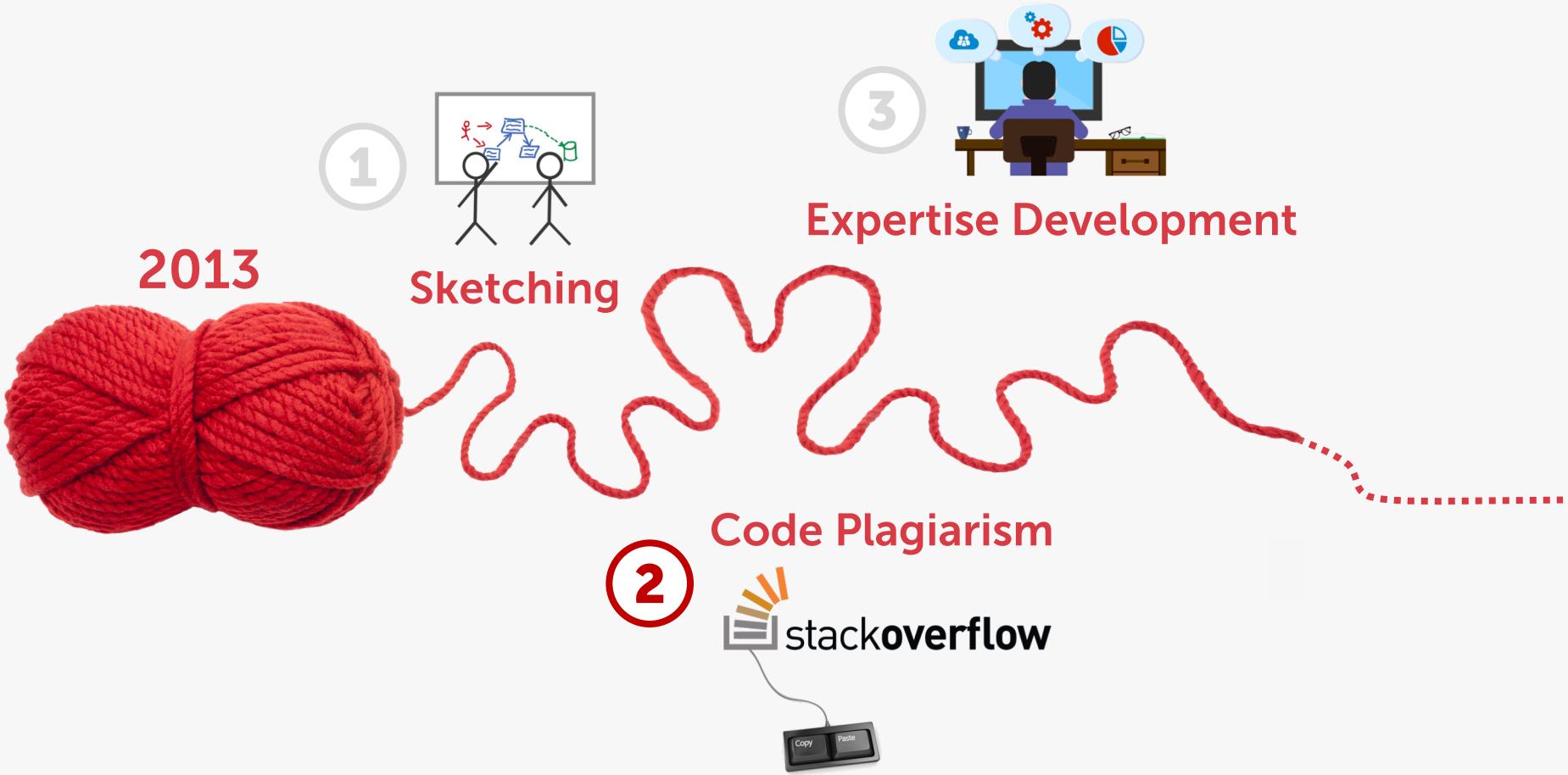


SketchLink



https://www.youtube.com/watch?v=mG6xCiQpS80

Overview of this Talk



Code Plagiarism

Code Plagiarism

Empirical Software Engineering
https://doi.org/10.1007/s10664-018-9650-5

CrossMark

Usage and attribution of Stack Overflow code snippets in GitHub projects

Sebastian Baltes[1] · Stephan Diehl[1]

Published online: 01 October 2018

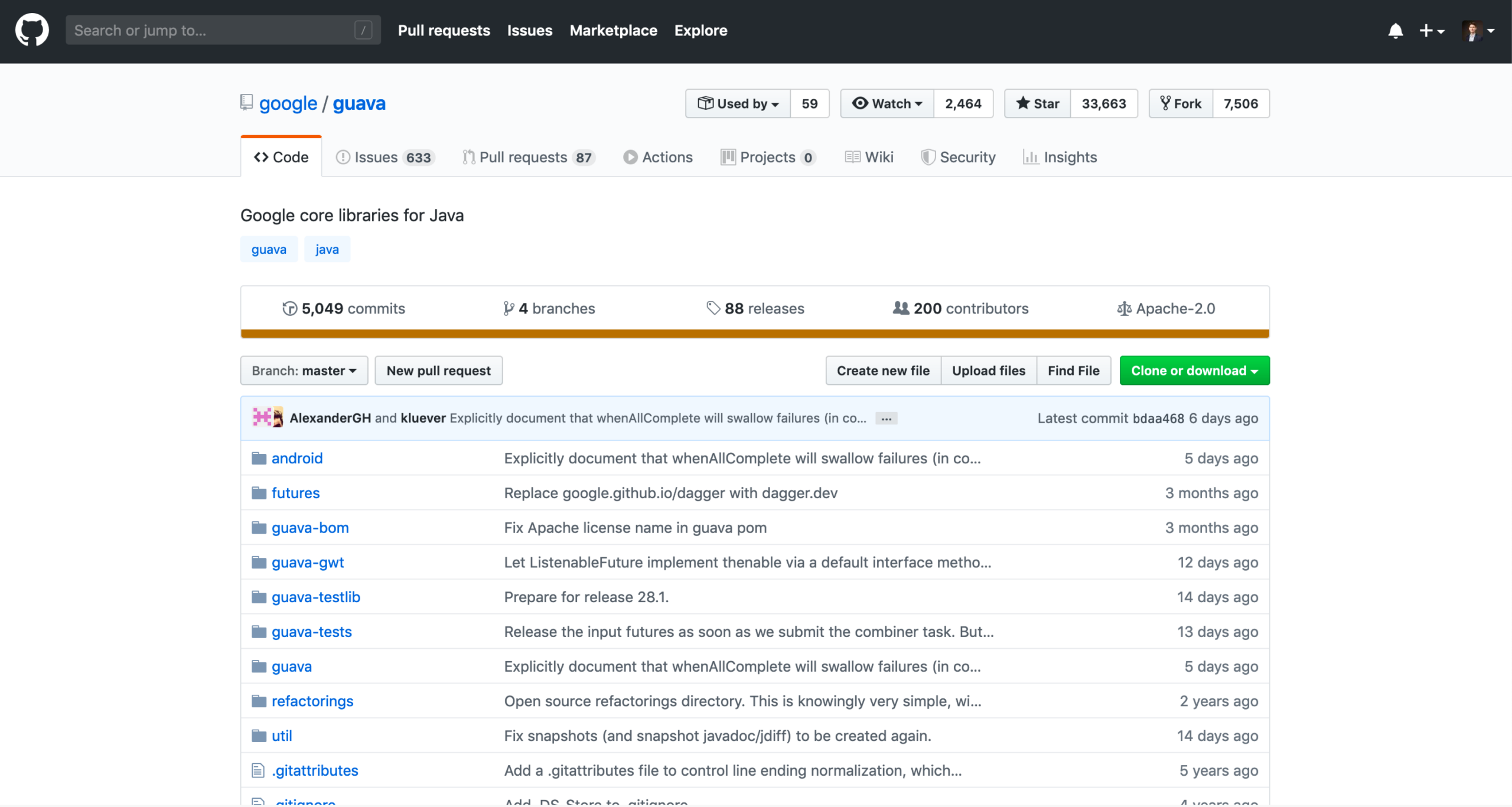
Abstract

Stack Overflow (SO) is the most popular question-and-answer website for software developers, providing a large amount of copyable code snippets. Using those snippets raises maintenance and legal issues. SO's license (CC BY-SA 3.0) requires attribution, i.e., referencing the original question or answer, and requires derived work to adopt a compatible license. While there is a heated debate on SO's license model for code snippets and the

https://empirical-software.engineering/projects/snippets/

GitHub

- **Hosted version control** platform for (software) projects
- Features include access control, **collaboration features** such as **issue tracking**, wikis, gamification of development activity
- **Public** projects and **private** projects with up to three collaborators are **free**
- As of May 2019: **>37m users** and **>100m projects**


Search or jump to...
Pull requests
Issues
Marketplace
Explore
google / guava
Used by 59
Watch 2,464
Star 33,663
Fork 7,506
Code
Issues 633
Pull requests 87
Actions
Projects 0
Wiki
Security
Insights
Google core libraries for Java
guava
java
5,049 commits
4 branches
88 releases
200 contributors
Apache-2.0
Branch: master
New pull request
Create new file
Upload files
Find File
Clone or download
AlexanderGH and kluever Explicitly document that whenAllComplete will swallow failures (in co...
Latest commit bdaa468 6 days ago
android Explicitly document that whenAllComplete will swallow failures (in co... 5 days ago
futures Replace google.github.io/dagger with dagger.dev 3 months ago
guava-bom Fix Apache license name in guava pom 3 months ago
guava-gwt Let ListenableFuture implement thenable via a default interface metho... 12 days ago
guava-testlib Prepare for release 28.1. 14 days ago
guava-tests Release the input futures as soon as we submit the combiner task. But... 13 days ago
guava Explicitly document that whenAllComplete will swallow failures (in co... 5 days ago
refactorings Open source refactorings directory. This is knowingly very simple, wi... 2 years ago
util Fix snapshots (and snapshot javadoc/jdiff) to be created again. 14 days ago
.gitattributes Add a .gitattributes file to control line ending normalization, which... 5 years ago

Stack Overflow

- **Question and answer** website for software developers
- Covers a **wide variety** of **programming-related topics**
- Posts can be commented, edited, and up-/down-voted
- **Gamification** through reputation points awarded for different kinds of contributions
- **Jobs** section for advertising employment opportunities
- As of June 2019 **>10.5m** registered users and **>17.7m questions**

Home
PUBLIC
Stack Overflow
Tags
Users
Jobs
TEAMS What's this?
First 10 Free

How do I read / convert an InputStream into a String in Java?

Ask Question

Asked 10 years, 9 months ago Active 2 days ago Viewed 2.0m times

3775

1107

If you have a `java.io.InputStream` object, how should you process that object and produce a `String`?

Suppose I have an `InputStream` that contains text data, and I want to convert it to a `String`, so for example I can write that to a log file.

What is the easiest way to take the `InputStream` and convert it to a `String`?

```
public String convertStreamToString(InputStream is) {
    // ???
}
```

java string io stream inputstream

share improve this question

edited Jan 5 at 10:28

Peter Mortensen
14.4k ● 19 ● 88 ● 117

asked Nov 21 '08 at 16:47

Johnny Maelstrom
19.3k ● 5 ● 17 ● 17

781 Boy, I'm absolutely in love with Java, but this question comes up so often you'd think they'd just figure out that the chaining of streams is somewhat difficult and either make helpers to create various combinations or rethink the whole thing. – Bill K Nov 21 '08 at 17:16

29 The answers to this question only work if you want to read the stream's contents *fully* (until it is closed). Since that is not always intended (http requests with a keep-alive connection won't be closed), these method calls block (not giving you the contents). – f1sh Jul 14 '10 at 13:32

Blog
"They Didn't Teach Us This": A Crash Course for Your First Job in Software
Featured on Meta
Stack Exchange and Stack Overflow are moving to CC BY-SA 4.0
Planned maintenance scheduled for Wednesday September 11, 2019 at 1:00 UTC…
What is this new DEV share button integrated with SO?
Experiment (ENDED): closing and reopening happens at 3 votes for the next 30…

Linked
51 BufferedInputStream To String Conversion?
29 Read text from InputStream

2 IOUtils.toString() has long been deprecated. This answer definitely is not the recommended way any more. – Roshan Feb 15 at 16:08

2 then edit it to explain why it is deprecated to help future readers. – Jean-François Fabre ♦ Apr 3 at 20:23

show 10 more comments

2251

Here's a way using only the standard Java library (note that the stream is not closed, your mileage may vary).

```
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
```

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in the stream, and in this case we separate tokens using "beginning of the input boundary" (\A), thus giving us only one token for the entire contents of the stream.

Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` constructor that indicates what character set to use (e.g. "UTF-8").

Hat tip goes also to Jacob, who once pointed me to the said article.

share improve this answer

edited Jan 5 at 10:33

Peter Mortensen
14.4k ● 19 ● 88 ● 117

answered Mar 26 '11 at 20:40

Pavel Repin
28.3k ● 1 ● 29 ● 37

8 Thanks, for my version of this I added a finally block that closes the input stream, so the user doesn't have to since you've finished reading the input. Simplifies the caller code considerably. – user486646 Apr 21 '12 at 17:07

Why is there a が in 深淵に臨むが如し?
Georgian capital letter "Ƥ" ("tar") in pdfLaTeX
Do Milankovitch Cycles fully explain climate change?
Are programming languages necessary/useful for operations research practitioner?
Which ping implementation is cygwin using?
Calculate time difference between two dates
Can I use ratchet straps to lift a dolly into a truck bed?
Does the word "uzi" need to be capitalized?
Dividing Divisive Divisors
How does Vivi differ from other Black Mages?
Is there a sentence that begins with "them"?
Character theory and Quantum Chemistry
What's the biggest difference between these two photos?
Should I use my toaster oven for slow roasting?
Which currencies does Wizz Air use in-flight?
Gas pipes - why does gas burn "outwards?"
2.5 year old daughter refuses to take medicine
What are the advantages and disadvantages of Preprints.org compared with arXiv?
Random Variable with IID always Gaussian?
Why would "an mule" be used instead of "a mule"?

Example

Read/convert an InputStream to a String

3101

929

If you have `java.io.InputStream` object, how should you process that object and produce a `String` ?

Suppose I have an `InputStream` that contains text data, and I want to convert this to a `String` . For example, so I can write the contents of the stream to a log file.

What is the easiest way to take the `InputStream` and convert it to a `String` ?

```
public String convertStreamToString(InputStream is) {
    // ???
}
```

java string io stream inputstream

share improve this question

edited May 19 '17 at 8:58 Pehlaj 4,824 ● 6 ● 25 ● 43

asked Nov 21 '08 at 16:47 Johnny Maelstrom 15.9k ● 5 ● 16 ● 17

Question

https://stackoverflow.com/q/309424

2034

Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
```

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in the stream, and in this case we separate tokens using "beginning of the input boundary" (\A) thus giving us only one token for the entire contents of the stream.

Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` constructor that indicates what charset to use (e.g. "UTF-8").

Hat tip goes also to Jacob, who once pointed me to the said article.

EDITED: Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share improve this answer

edited Sep 2 '17 at 1:27

answered Mar 26 '11 at 20:40 Pavel Repin 25.3k ● 1 ● 27 ● 36

Answer

https://stackoverflow.com/a/5445161

2034

Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
```

Code snippet

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in ase we separate tokens using boundary" (\A) thus giv the entire contents of the st

Source of snippet

Reference to JDK

Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` constructor that indicates what charset to use (e.g. "UTF-8").

Hat tip goes also to Jacob, who once pointed me to the said article.

EDITED: Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share in er

Post edits

edited Sep 2

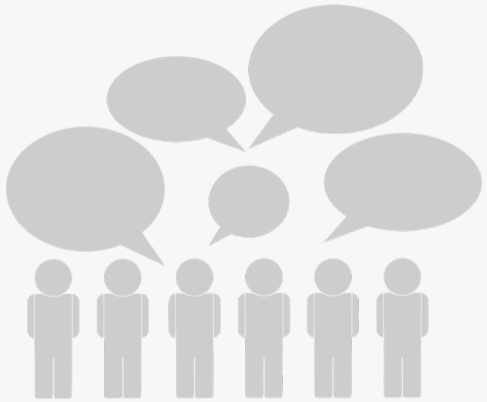
Reasons for edits

ered Mar 26 '11 at 20:40

Pavel Repin

25.3k ● 1 ● 27 ● 36

Comments



EDITED: Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share improve this answer — edited Sep 2 '17 at 1:27 — answered Mar 26 '11 at 20:40 Pavel Repin 25.3k ●1 ●27 ●36

7 Thanks, for my version of this I added a finally block that closes the input stream, so the user doesn't have to since you've finished reading the input. Simplifies the caller code considerably. – user486646 Apr 21 '12 at 17:07

4 @PavelRepin @Patrick in my case, an empty inputStream caused a NPE during Scanner construction. I had to add `if (is == null) return "";` right at the beginning of the method; I believe this answer needs to be updated to better handle null inputStreams. – CFL_Jeff Aug 9 '12 at 13:36

The problem with this approach I find is it does not handle CR/LF translations too well. So you have to make sure your line endings are consistent. – Archimedes Trajano Feb 28 '13 at 12:13

@ArchimedesTrajano does `IOUtils.copy(inputStream, writer, encoding)` deal with CR/LF translations better? I think CR/LF consistency is entirely unrelated issue. Not saying it isn't an issue. – Pavel Repin Mar 1 '13 at 9:18

95 For Java 7 you can close in a try-with: `try(java.util.Scanner s = new java.util.Scanner(is)) { return s.useDelimiter("\\A").hasNext() ? s.next() : ""; }` – earcam Jun 13 '13 at 5:24

3 Unfortunately this solution seems to go and lose the exceptions thrown in my underlying stream implementation. – Taig Jul 16 '13 at 7:59

excellent trick! any ideas about performance of Scanner vs reading the stream in a more verbose way? – isapir Aug 28 '13 at 19:54

@Igal I didn't measure it. If you do, gist it and I'll append your results to the answer. – Pavel Repin Aug 28 '13 at 23:13

11 FYI, hasNext blocks on console input streams (see here). (Just ran into this issue right now.) This solution works fine otherwise... just a heads up. – Ryan Feb 24 '14 at 5:36

1 @earcam thanks for the tip! For those wondering how this works, it's thanks to try-with-resources – Mark Mar 14 '15 at 21:33

1 looks like a neat trick, but it seems there are some limitations. For me it hangs when reading InputStream from Socket. When testing with something like ByteArrayInputStream it works nicely. Reading from socket results in a hang. – Normunds Kalnberzins Dec 16 '15 at 14:16

If the `Scanner` is going to be *"giving us only one token for the entire contents of the stream"* anyways, why not use a normal stream reader? `Scanner` is meant to pre-parse tokens out of the stream, not for being the stream reader (without any parsing being done). – XenoRo Dec 28 '15 at 14:06

@AlmightyR `Scanner` has built-in stream reading logic and we're telling it that the stream has just one token. A special case of Scanner usage. Fair game. Good point though. This stuff is clearly a hack. – Pavel Repin Jan 15 '16 at 1:23

1 be careful ,using this method with socket stream is slow ! Scanner#next() hangs for a little while. – WestFarmer Apr 20 '16 at 10:22

1 nice answer, the article link is on oracle website community.oracle.com/blogs/pat/2004/10/23/stupid-scanner-tricks – Eng. Samer T Jul 23 '17 at 16:04

Bug report

Alternative solution

Bug report

Bug report

Comment by author

This stuff is clearly a hack.

SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts

Sebastian Baltes
Lorik Dumani
research@sbaltes.com
dumani@uni-trier.de
University of Trier, German

Christoph Treude
christoph.treude@adelaide.edu.au
University of Adelaide, Australia

Stephan Diehl
diehl@uni-trier.de
University of Trier, Germany

ABSTRACT
Stack Overflow (SO) is the most popular
site for software developers, providin
snippets and free-form text on a wide v
software artifacts, questions and answe
for example when bugs in code snippet
to work with a more recent library ver
code snippet is edited for clarity. To be a
on SO evolves, we built *SOTorrent*, an
official SO data dump. *SOTorrent* provid
tory of SO content at the level of whole
code blocks. It connects SO posts to othe
URLs from text blocks and by collectin

SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets

Sebastian Baltes
University of Trier, Germany
research@sbaltes.com

Christoph Treude
University of Adelaide, Australia
christoph.treude@adelaide.edu.au

Stephan Diehl
University of Trier, Germany
diehl@uni-trier.de

***Abstract*—Stack Overflow (SO) is the most popular question-and-answer website for software developers, providing a large amount of copyable code snippets. Like other software artifacts, code on SO evolves over time, for example when bugs are fixed or APIs are updated to the most recent version. To be able to analyze how code and the surrounding text on SO evolves, we built *SOTorrent*, an open dataset based on the official SO data dump. *SOTorrent* provides access to the version history of SO content at the level of whole posts and individual text and code blocks. It connects code snippets from SO posts to other platforms by aggregating URLs from surrounding text blocks and comments, and by collecting references from GitHub files to SO posts. Our vision is that researchers will use *SOTorrent* to investigate and understand the evolution and maintenance of code on SO and its relation to other platforms such as GitHub.**

dataset [16] that enables researchers to analyze the version history of SO posts at the level of individual text and code blocks (see Figure 1 for exemplary posts). The official SO data dump [1] keeps track of different versions of entire posts, but does not contain information about differences between versions at a more fine-grained level. In particular, extracting different versions of the same code snippet from the history of a post is challenging and required us to develop a complex strategy, involving the evaluation of 134 different string similarity metrics [15]. Beside providing access to the version history, our dataset links SO posts to external resources in two ways: (1) by extracting linked URLs from text blocks of SO posts and from post comments and (2) by providing

MSR 2018/19

sotorrent.org

Open Data

Dataset available on Zenodo and BigQuery


stackoverflow
Copy
Paste

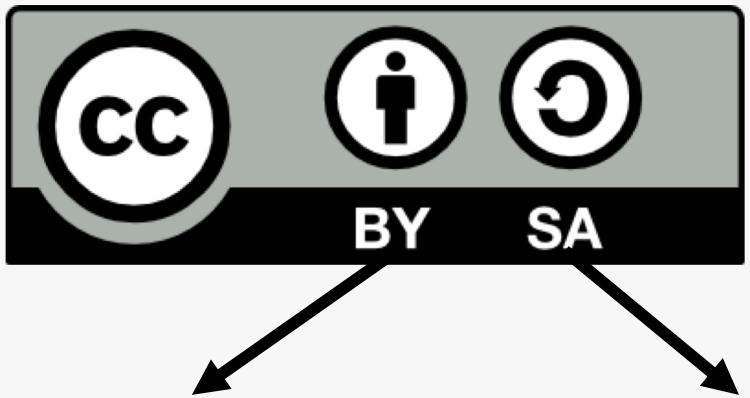
Question for the Audience I

Who admits regularly copying non-trivial code snippets from Stack Overflow?

Question for the Audience II

Who knew that all content on Stack Overflow is licensed under CC BY-SA?

*"You must give **appropriate credit** [...] and indicate if changes were made."*

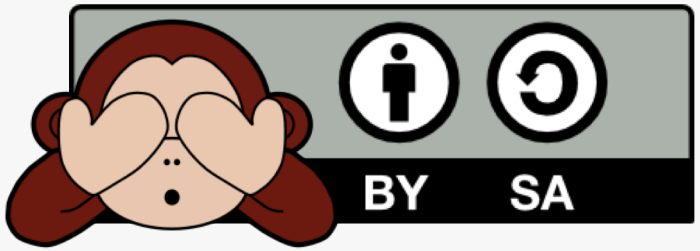


*"If you [...] **build upon** the material, you must **distribute your contributions** under the same license as the original."*

Results from our Online Surveys

- **46%** of the participants admitted copying code from Stack Overflow **without attribution**
- **75%** did **not know** that content on SO is licensed under **CC BY-SA**
- **67%** did **not know** that **attribution is required**

→ **Lack of awareness**



Background

"Well, but these snippets are rather trivial and not protected by copyright."

- Not all code snippets on Stack Overflow are copyrightable
- "A snippet that is more than one or two lines of standard function calls would typically be creative enough for copyright" [Engelfriet 2016]
- But no "international standard for originality" [Creative Commons 2017b]

http://theconversation.com/why-universities-cant-be-expected-to-police-copyright-infringement-82677

889

Here's what I do:

1. First of all I check what providers are enabled. Some may be disabled on the device, some may be disabled in application manifest.
2. If any provider is available I start location listeners and timeout timer. It's 20 seconds in my example, may not be enough for GPS so you can enlarge it.
3. If I get update from location listener I use the provided value. I stop listeners and timer.
4. If I don't get any updates and timer elapses I have to use last known values.
5. I grab last known values from available providers and choose the most recent of them.

Here's how I use my class:

```
LocationResult locationResult = new LocationResult(){
    @Override
    public void gotLocation(Location location){
        //Got the location!
    }
};
MyLocation myLocation = new MyLocation();
myLocation.getLocation(this, locationResult);
```

And here's MyLocation class:

```
import java.util.Timer;
import java.util.TimerTask;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;

public class MyLocation {
    Timer timer1;
    LocationManager lm;
    LocationResult locationResult;
    boolean gps_enabled=false;
    boolean network_enabled=false;

    public boolean getLocation(Context context, LocationResult result)
    {
        //I use LocationResult callback class to pass location value from MyLocat
        locationResult=result;
        if(lm==null)
            lm = (LocationManager) context.getSystemService(Context.LOCATION_SERV

        //exceptions will be thrown if provider is not permitted.
        try{gps_enabled=lm.isProviderEnabled(LocationManager.GPS_PROVIDER);}catch
        try{network_enabled=lm.isProviderEnabled(LocationManager.NETWORK_PROVIDER

        //don't start listeners if no provider is enabled
        if(!gps_enabled && !network_enabled)
            return false;

        if(gps_enabled)
            lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, locatio
        if(network_enabled)
            lm.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, loc
```

Somebody may also want to modify my logic. For example if you get update from Network provider don't stop listeners but continue waiting. GPS gives more accurate data so it's worth waiting for it. If timer elapses and you've got update from Network but not from GPS then you can use value provided from Network.

One more approach is to use LocationClient http://developer.android.com/training/location/retrieve-current.html. But it requires Google Play Services apk to be installed on user device.

share improve this answer

edited Jun 25 '13 at 9:33

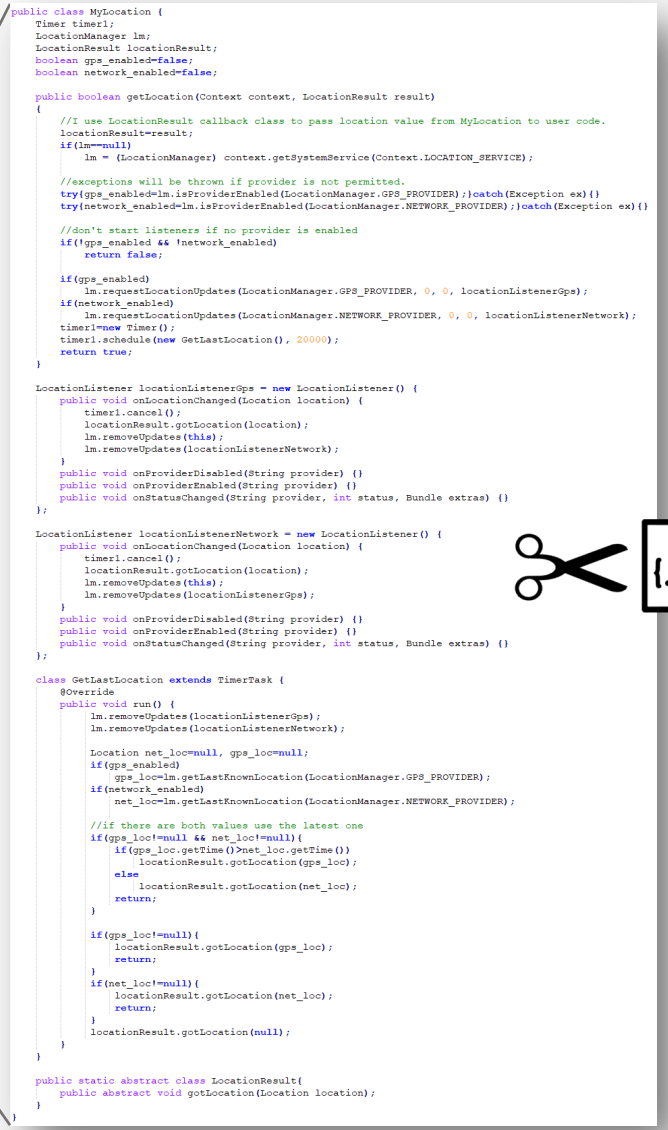
answered Jun 30 '10 at 0:07

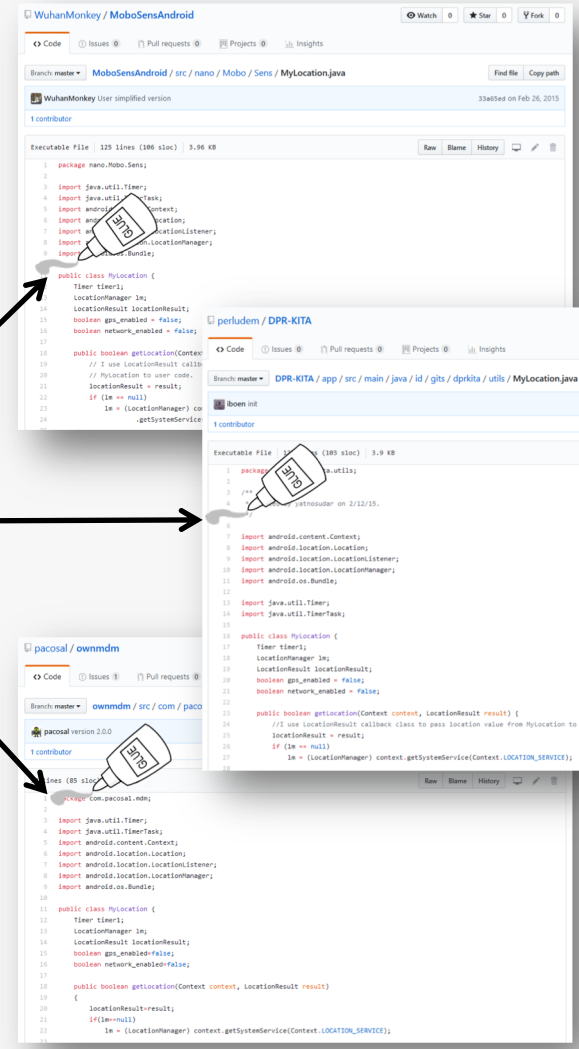
Fedor
40k ● 9 ● 71 ● 86

https://stackoverflow.com/a/3145655

stackoverflow

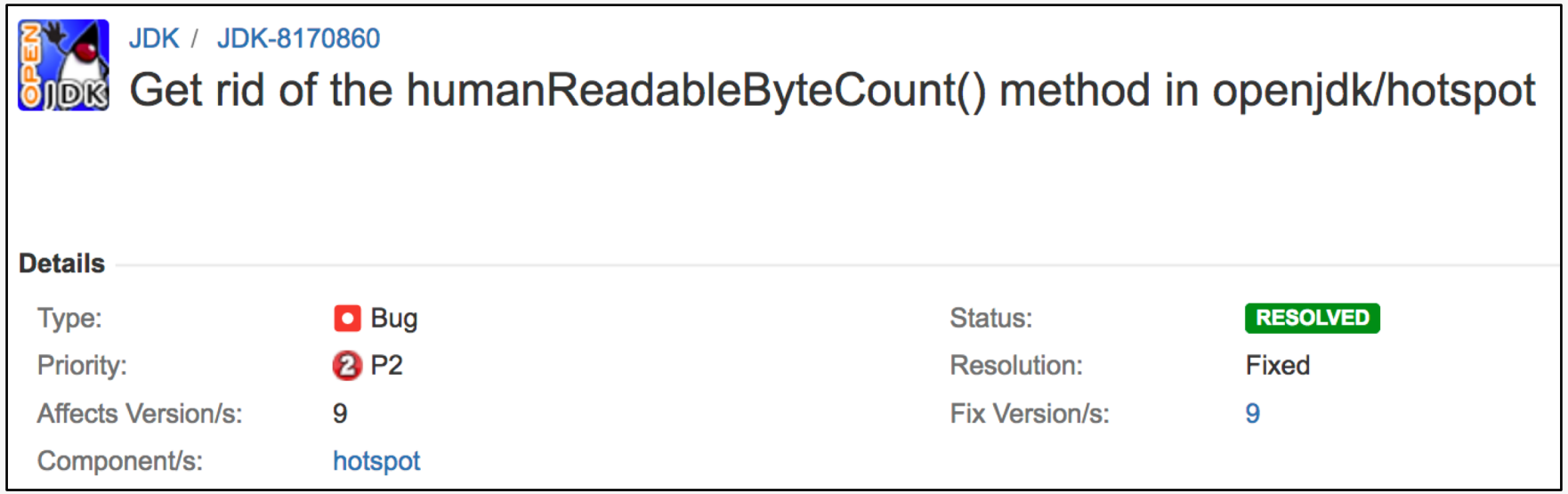
GitHub





https://github.com/pacosal/ownmdm/blob/master/src/com/pacosal/mdm/MyLocation.java

Stack Overflow Code in the OpenJDK



JDK / JDK-8170860

Get rid of the humanReadableByteCount() method in openjdk/hotspot

Details

Type:	Bug	Status:	RESOLVED
Priority:	P2	Resolution:	Fixed
Affects Version/s:	9	Fix Version/s:	9
Component/s:	hotspot		

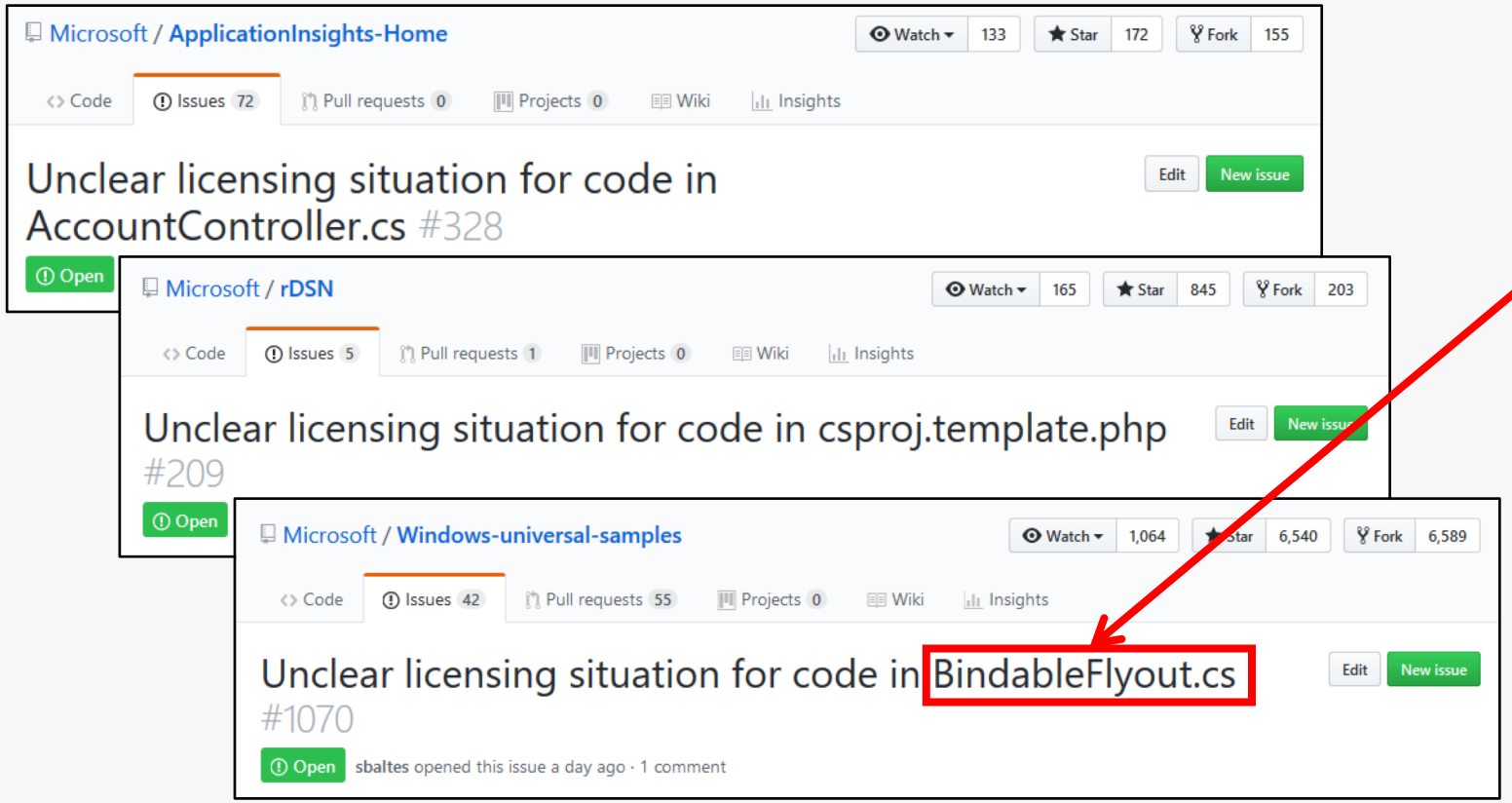
implement the method humanReadableByteCount which body was copied from the Stack Overflow site: https://stackoverflow.com/a/3758880

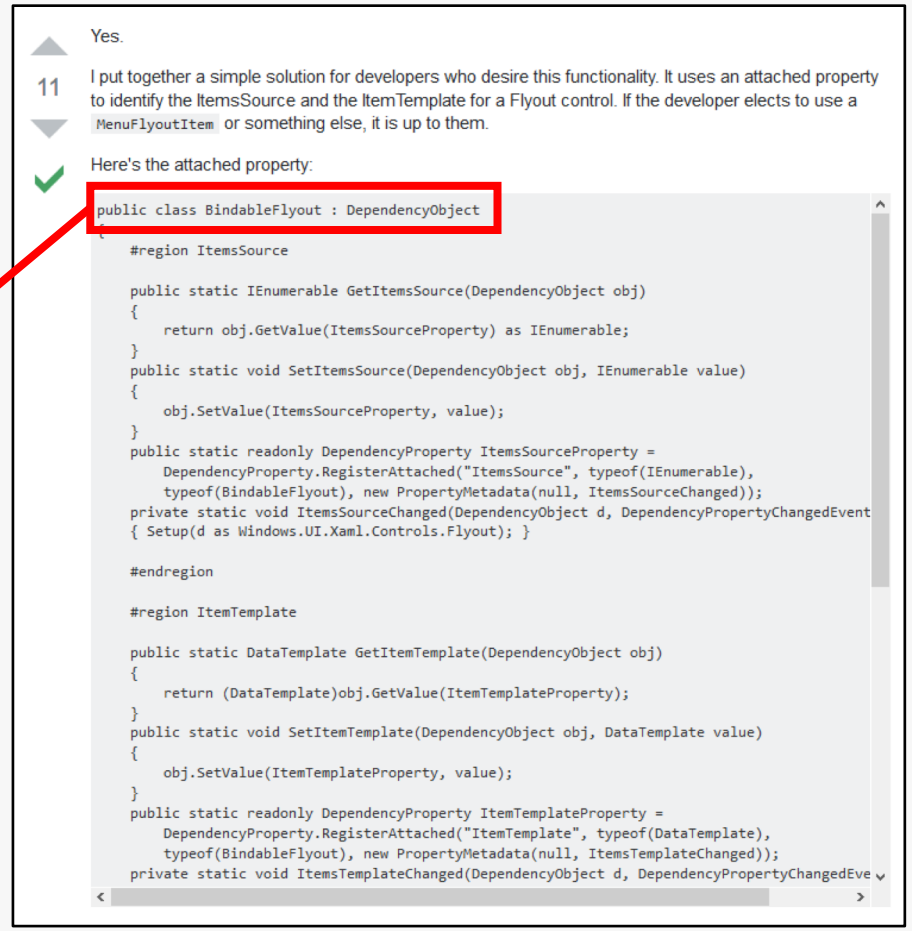
It's just a few lines of code, but it could cause legal issues. The method should be either re-implemented or removed.

Besides the potential legal issues, duplicating a code is not a good practice.

https://bugs.openjdk.java.net/browse/JDK-8170860

… and in Microsoft GitHub Repos

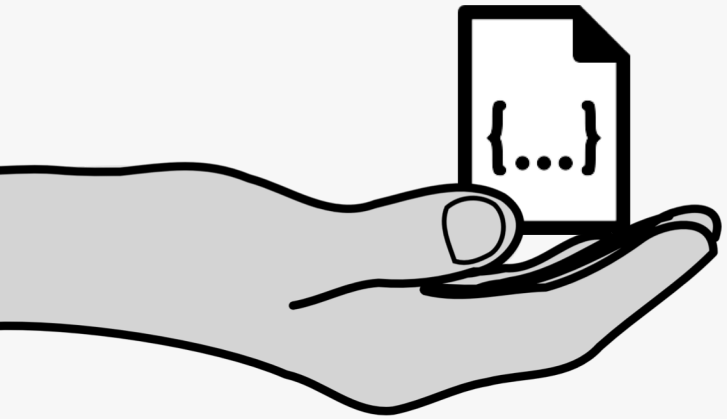




Implications of Stack Overflow's License

Permissive Licenses

- Permit using the licensed source code in proprietary software **without publishing changes** or the **derived work**
- *Examples:* MIT, Apache, and BSD license families



Copyleft Licenses

- Requires either modifications to the licensed content or the complete derived work to be **published under the same or a compatible license** (share-alike)
- *Examples (weak copyleft):* Mozilla/Eclipse Public Licenses
- *Examples (viral copyleft):* GNU General Public Licenses, Creative Commons Share-Alike Licenses (e.g., **CC BY-SA**)

Enforceability of Copyleft Licenses

- Courts in the US and Europe ruled that open source licenses are **enforceable contracts**
- Authors are able to **sue** when terms such as the share-alike requirement are violated:
  - **Interdict distribution** of derived work
  - **Claim monetary damages**
- USA: DMCA takedown notices for allegedly infringed copyright
  - Example: https://github.com/github/dmca
- Risk in mergers and acquisitions of companies
  - Example: FSF vs. Cisco lawsuit

Research Question

Question:

How **frequently** is code from Stack Overflow posts used in public GitHub projects **without** the required **attribution**?

Approach:

Triangulate an estimate for the attribution ratio using three different methods.

Attribution

Attribution ratio:

- Method 1 (regular expressions): 23 %
- Method 2 (code clone detector): 24 %
- Method 3 (exact matches): 8 %

Conservative estimate:

- **Attribution ratio $\leq$ 25%**

Share-alike

Only **2%** of all analyzed repositories (all methods) containing code from Stack Overflow **attributed** its source and used a **compatible license** (not CC BY-SA, but GPL 3.0).

SPDX license name	Number of repos containing a SO code snippet clone that was: unattributed ($n = 2,962$)	attributed ($n = 329$)
Apache-2.0	921 (31.1%)	99 (30.1%)
MIT	621 (21.0%)	72 (21.9%)
GPL-3.0	435 (14.7%)	60 (18.2%)
GPL-2.0	284 (9.6%)	21 (6.4%)
BSD-3-Clause	82 (2.8%)	9 (2.7%)

Method 1

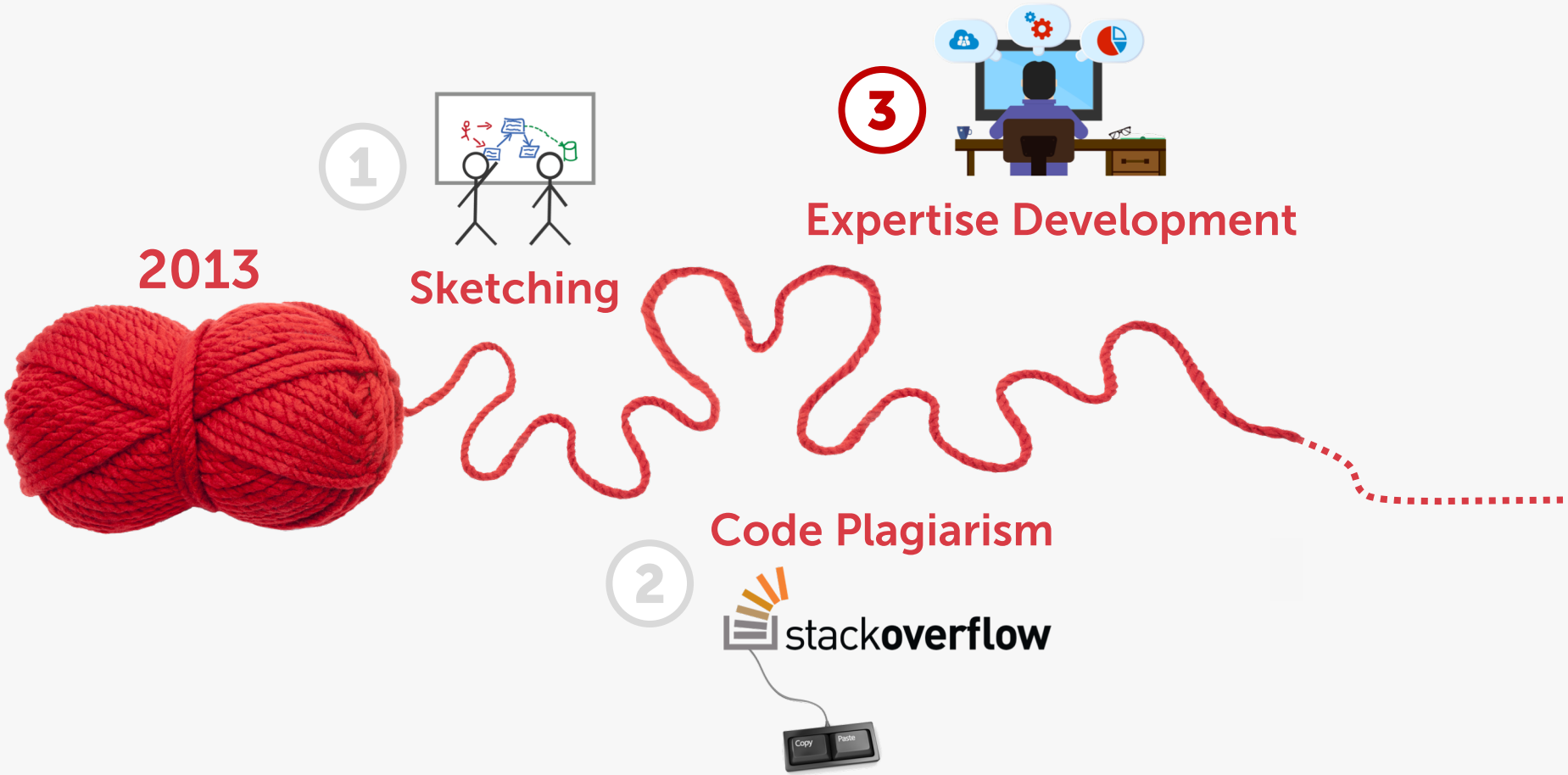
SPDX license name	Number of repos containing a SO code snippet clone that was: unattributed ($n = 144$)	attributed ($n = 55$)
None	56 (38.9%)	18 (32.7%)
Apache-2.0	33 (22.9%)	15 (27.3%)
GPL-3.0	17 (11.8%)	6 (10.9%)
MIT	6 (4.2%)	4 (7.3%)
GPL-2.0	4 (2.8%)	2 (3.6%)

Method 2

SPDX license name	Number of repos containing a SO code snippet clone that was: unattributed ($n = 1,169$)	attributed ($n = 163$)
Apache-2.0	353 (30.2%)	36 (37.4%)
MIT	239 (20.4%)	25 (15.3%)
GPL-3.0	211 (18.0%)	19 (11.7%)
None	153 (13.1%)	61 (37.4%)
GPL-2.0	89 (7.61%)	8 (4.9%)

Method 3

Overview of this Talk



Expertise Development

Expertise Development

Towards a Theory of Software Development Expertise

Sebastian Baltes
University of Trier
Trier, Germany
research@sbaltes.com

ESEC/FSE 2018

Stephan Diehl
University of Trier
Trier, Germany
diehl@uni-trier.de

ABSTRACT

Software development includes diverse tasks such as implementing new features, analyzing requirements, and fixing bugs. Being an expert in those tasks requires a certain set of skills, knowledge, and experience. Several studies investigated individual aspects of software development expertise, but what is missing is a comprehensive theory. We present a first conceptual theory of software development expertise that is grounded in data from a mixed-methods survey with 335 software developers and in literature on expertise and expert performance. Our theory currently focuses on programming, but already provides valuable insights for researchers, developers, and employers. The theory describes important properties of software development expertise and which factors foster or hinder its formation, including how developers' performance may decline over time. Moreover, our quantitative results show that developers' expertise self-assessments are context-dependent and that experience is not necessarily related to expertise.

expert performance [78]. Bergersen et al. proposed an instrument to measure programming skill [9], but their approach may suffer from learning effects because it is based on a fixed set of programming tasks. Furthermore, aside from programming, software development involves many other tasks such as requirements engineering, testing, and debugging [62, 96, 100], in which a software development expert is expected to be good at.

In the past, researchers investigated certain aspects of software development expertise (SDExp) such as the influence of programming experience [95], desired attributes of software engineers [63], or the time it takes for developers to become "fluent" in software projects [117]. However, there is currently no theory combining those individual aspects. Such a theory could help structuring existing knowledge about SDExp in a concise and precise way and hence facilitate its communication [44]. Despite many arguments in favor of developing and using theories [46, 56, 85, 109], theory-driven research is not very common in software engineering [97].

https://empirical-software.engineering/projects/expertise/

Software Development Expertise?

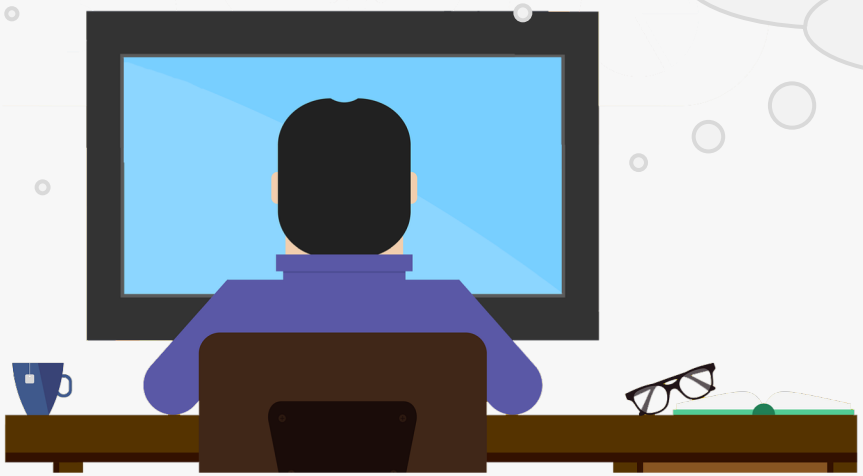
Implementing new features

Algorithms & Data structures

Testing

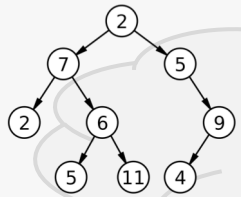
Communication

Debugging

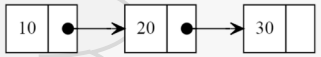


Software Development Expertise?

Implementing new features



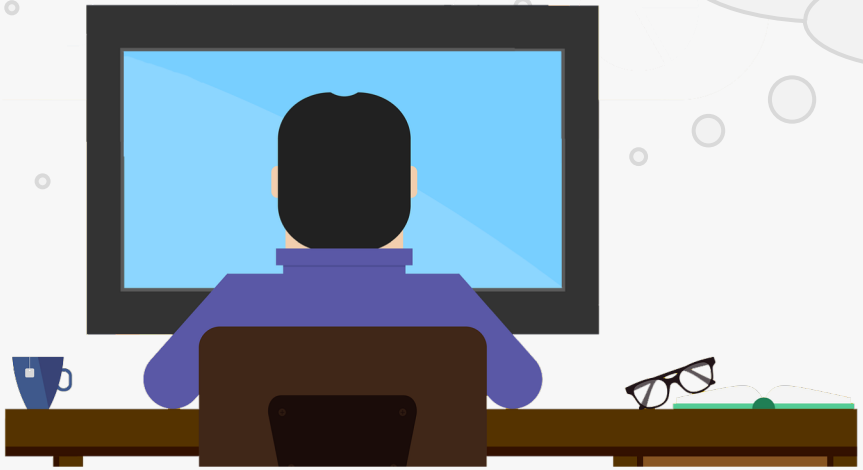
Algorithms & Data structures



JUnit 5 Testing jbehave

Communication

Debugging IJ




How to structure all those expertise-related aspects?


Which factors influence expertise development over time?

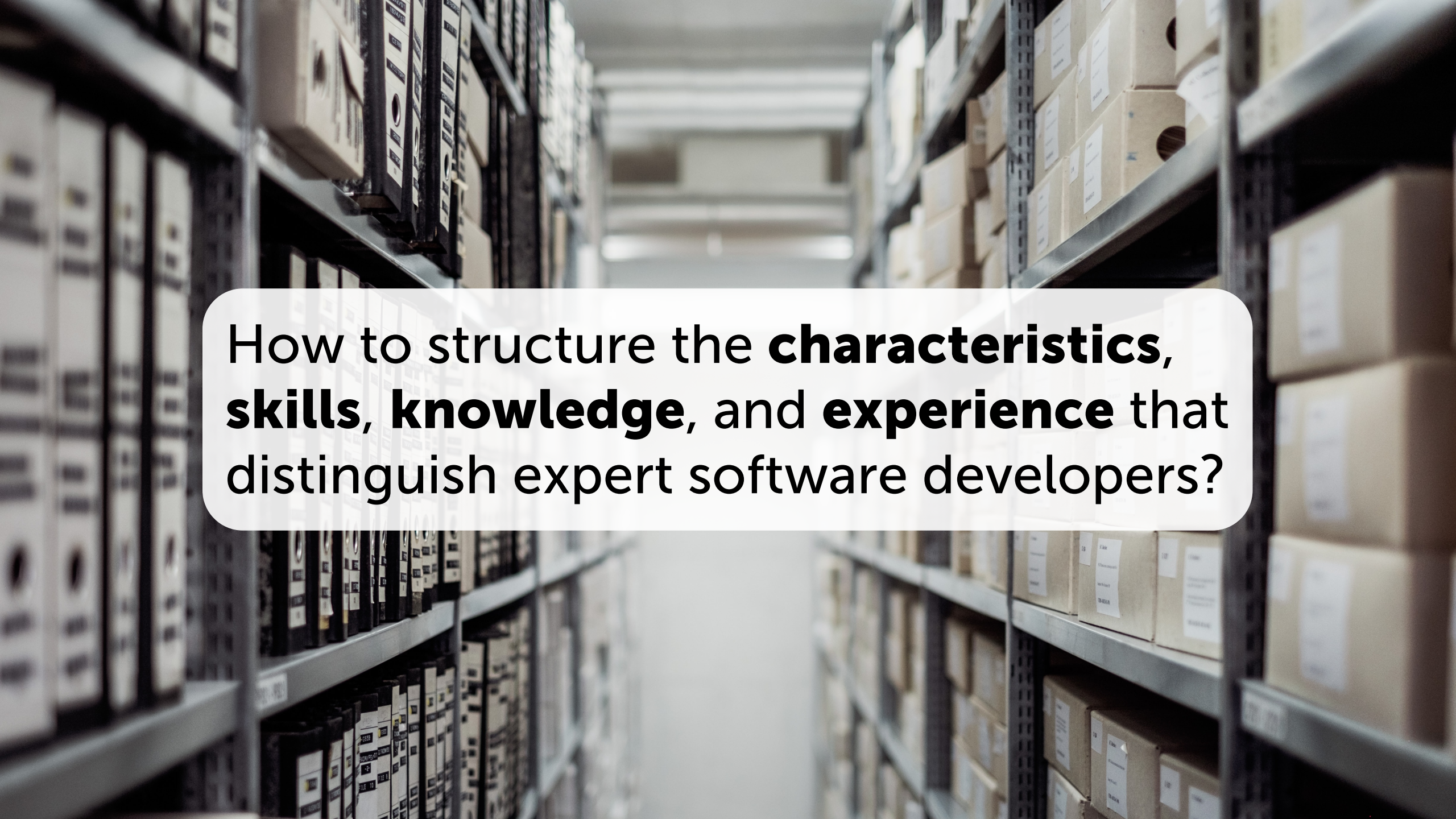
How are experience and expertise related?

Definitions

An expert is someone *"with the special **skill** or **knowledge** representing mastery of a **particular subject**"*

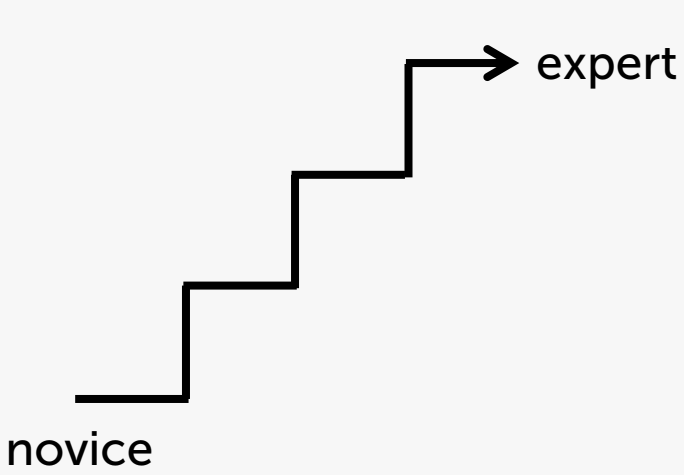
Expertise are *„the **characteristics**, **skills**, and **knowledge** that distinguish experts from novices and less **experienced** people."*

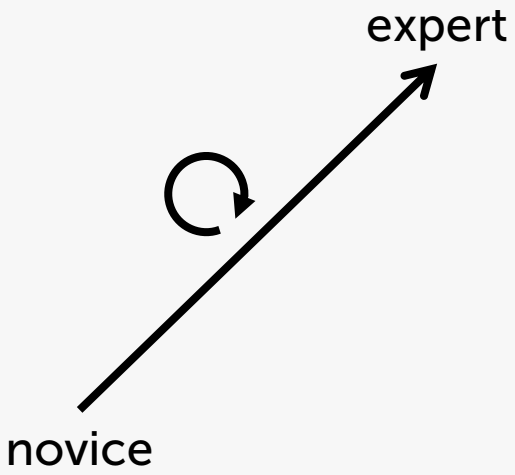
K. Anders Ericsson


How to structure the **characteristics**, **skills**, **knowledge**, and **experience** that distinguish expert software developers?

Our Expertise Model

- **Task-specific** (e.g., writing code, debugging, testing)
- Focuses on **individual developers**
- **Process** view (repetition of tasks)
- Notion of **transferable knowledge and experience** from related fields or tasks
- **Continuum** instead of discrete expertise steps

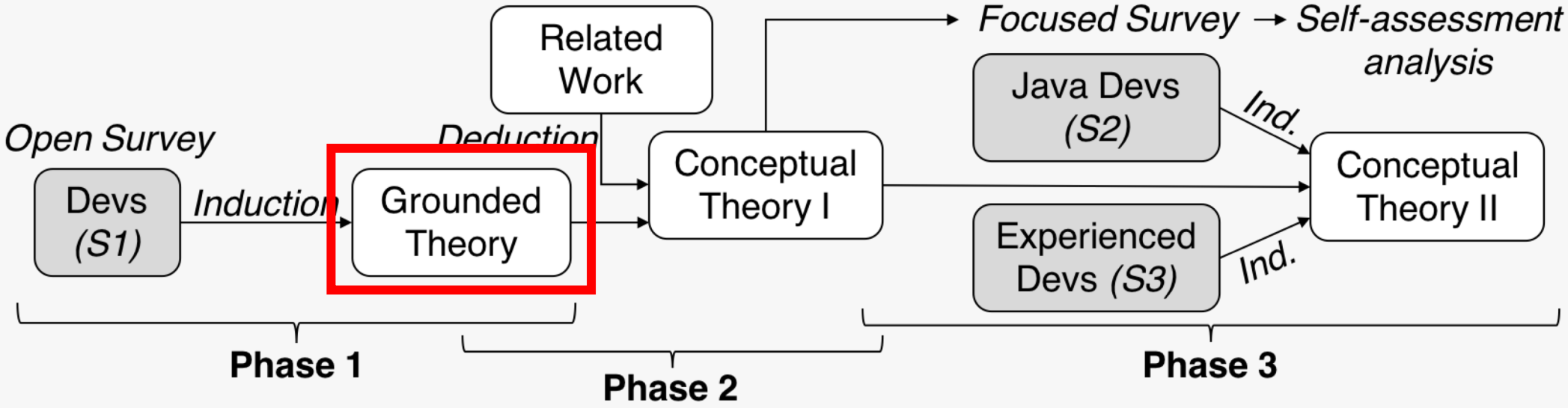




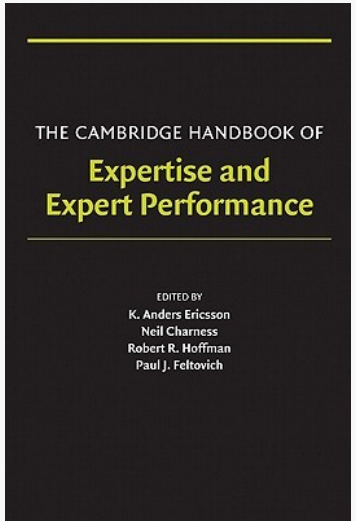
Theory Classification

- A **process theory** intends to explain and understand *"how an entity changes and develops"* over time (Ralph, 2018)
- In a **teleological process theory**, an entity *"constructs an envisioned end state, takes action to reach it, and monitors the progress"* (van de Ven and Poole, 1995)
- **Our theory:**
  - *Entity:*
    Individual software developer working on different software development tasks
  - *Envisioned end state:*
    Being an expert in (some of) those tasks

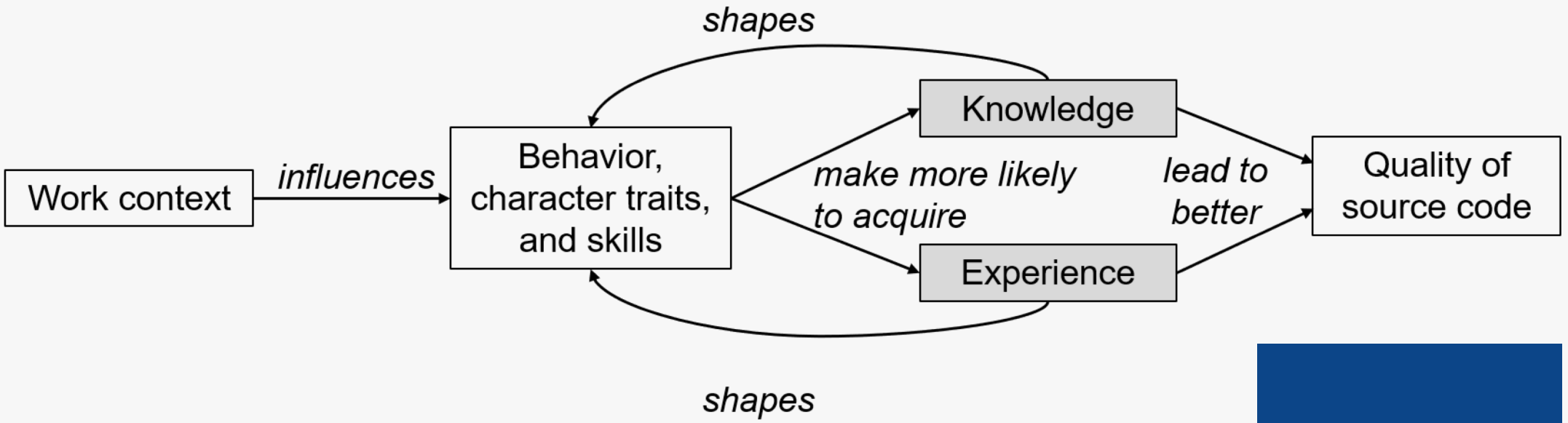
Research Design

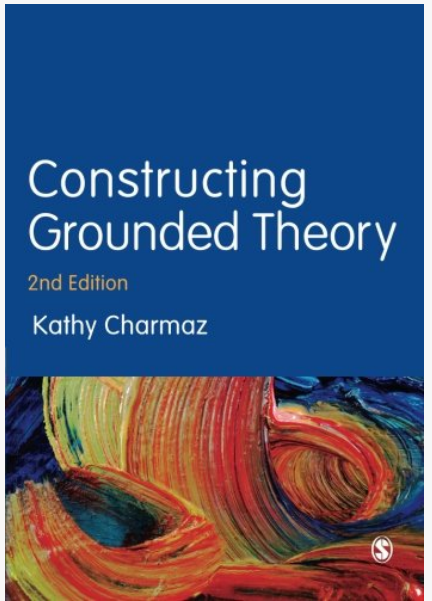


- **Induction:** 335 online survey participants in total
- **Deduction:** Main source *"Cambridge Handbook of Expertise and Expert Performance"*

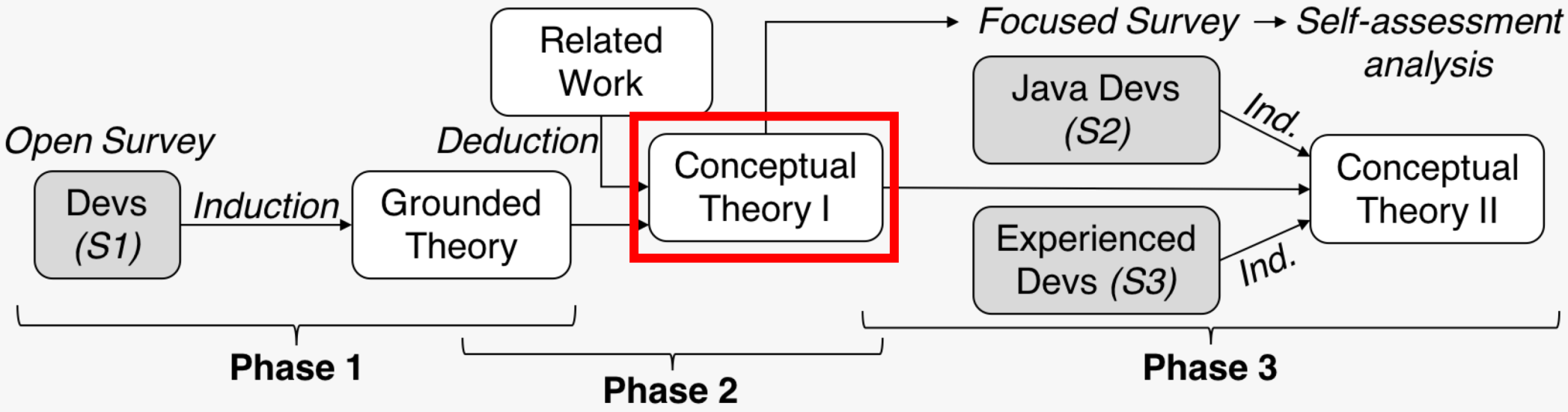


Grounded Theory

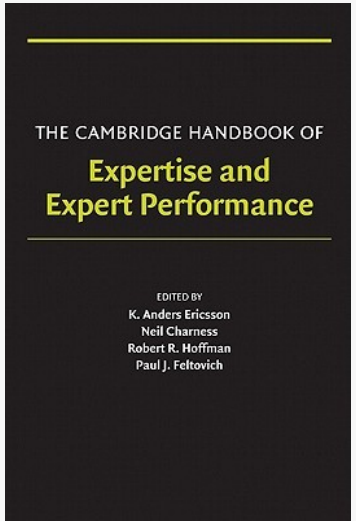




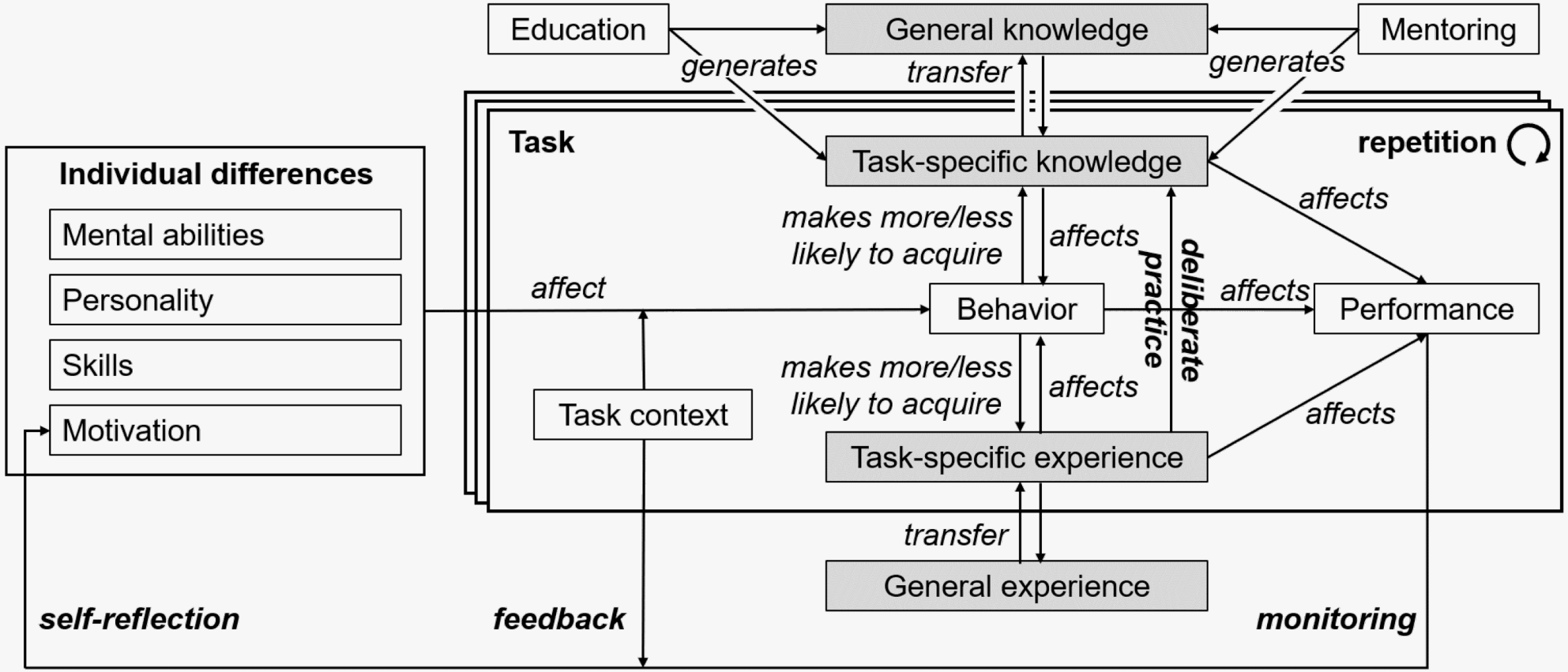
Research Design



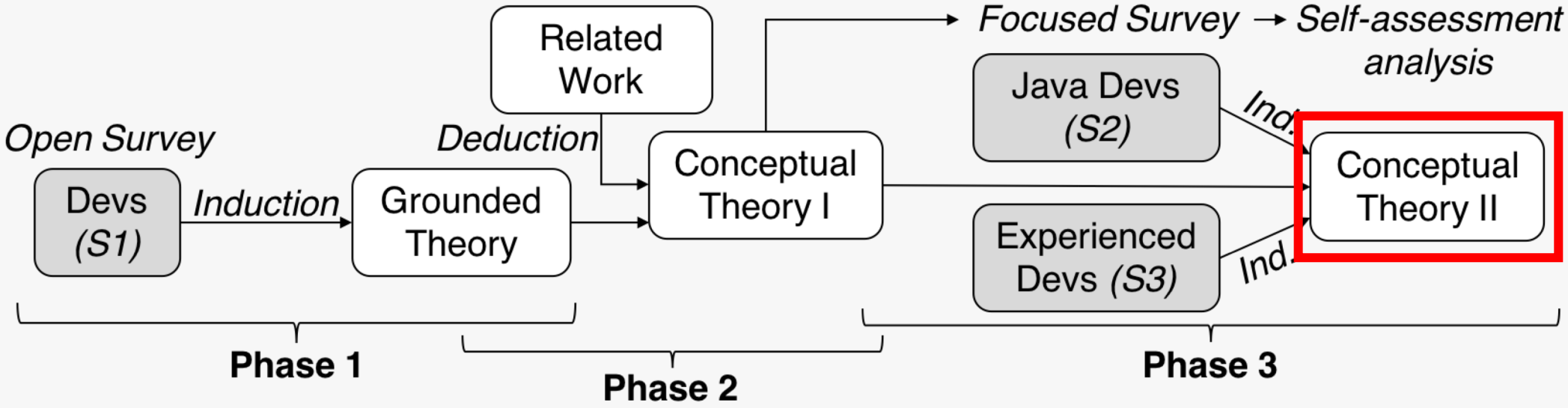
- **Induction:** 335 online survey participants in total
- **Deduction:** Main source *"Cambridge Handbook of Expertise and Expert Performance"*



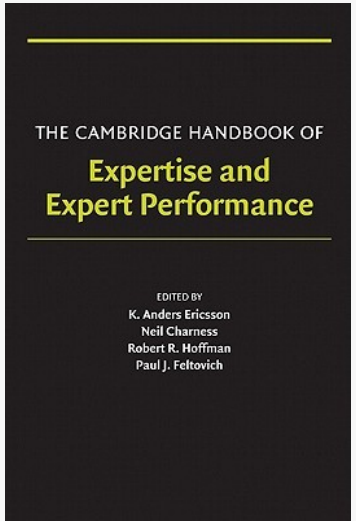
Preliminary Conceptual Theory



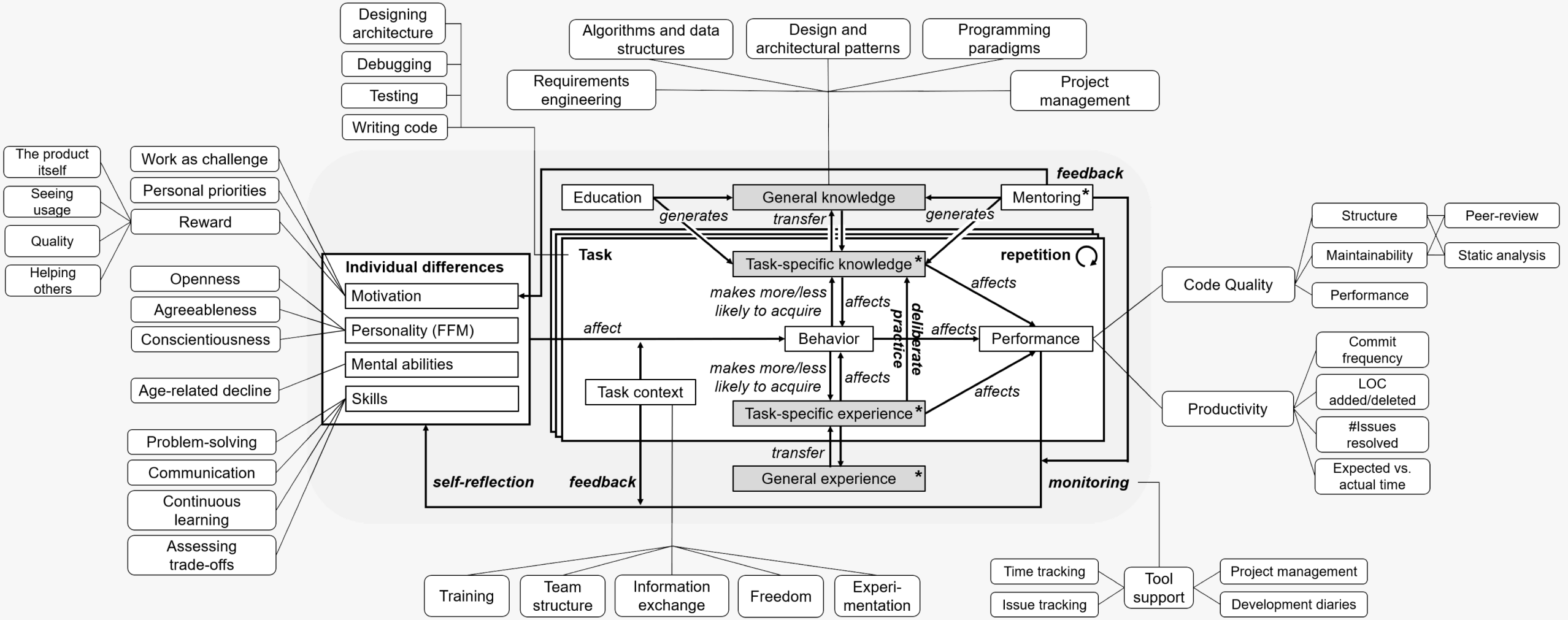
Research Design



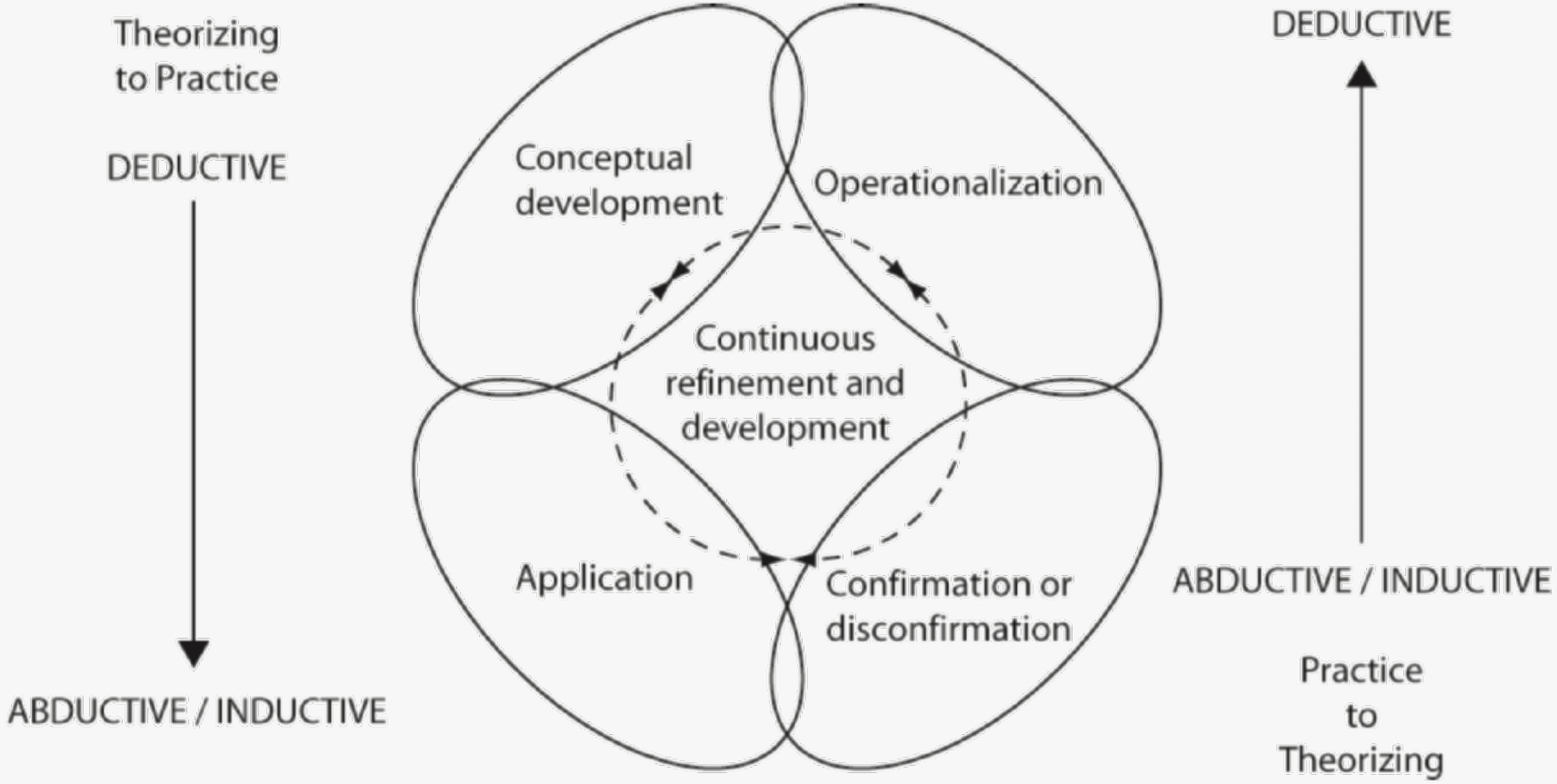
- **Induction:** 335 online survey participants in total
- **Deduction:** Main source *"Cambridge Handbook of Expertise and Expert Performance"*



Final Conceptual Theory

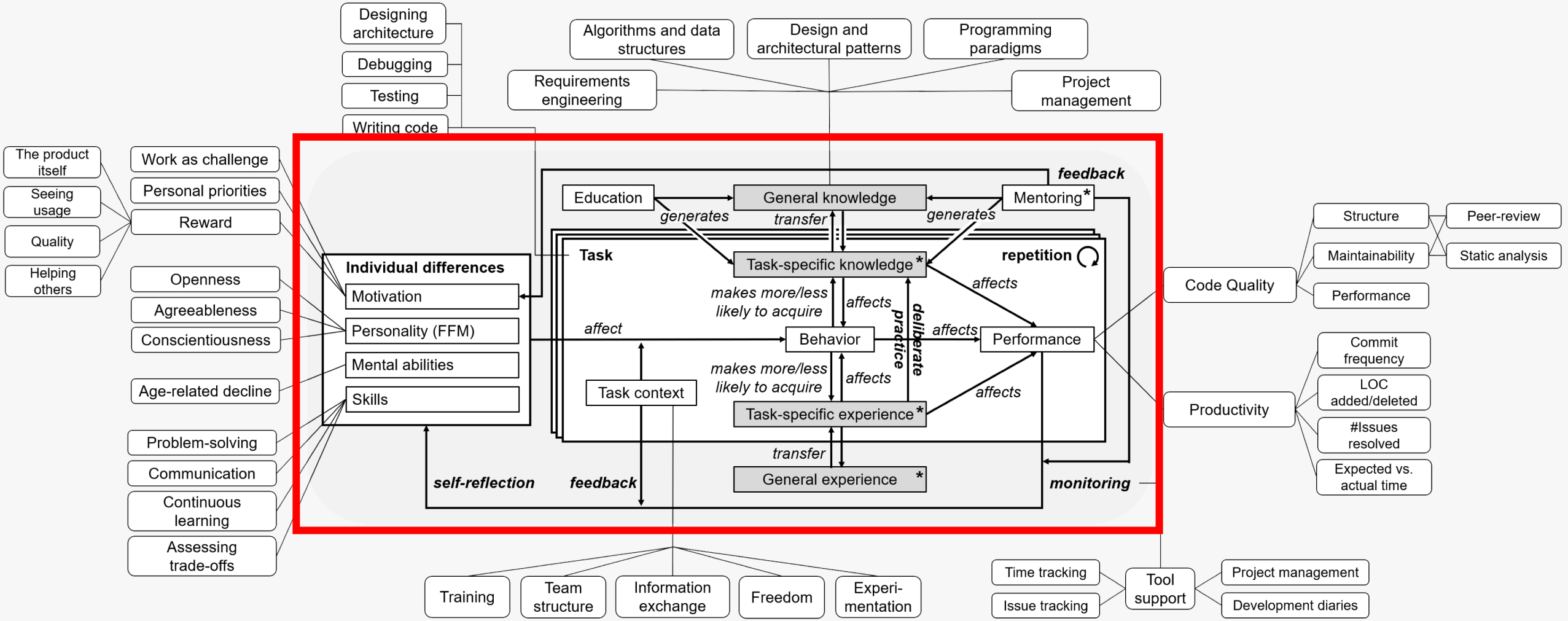


Conceptual Theory?

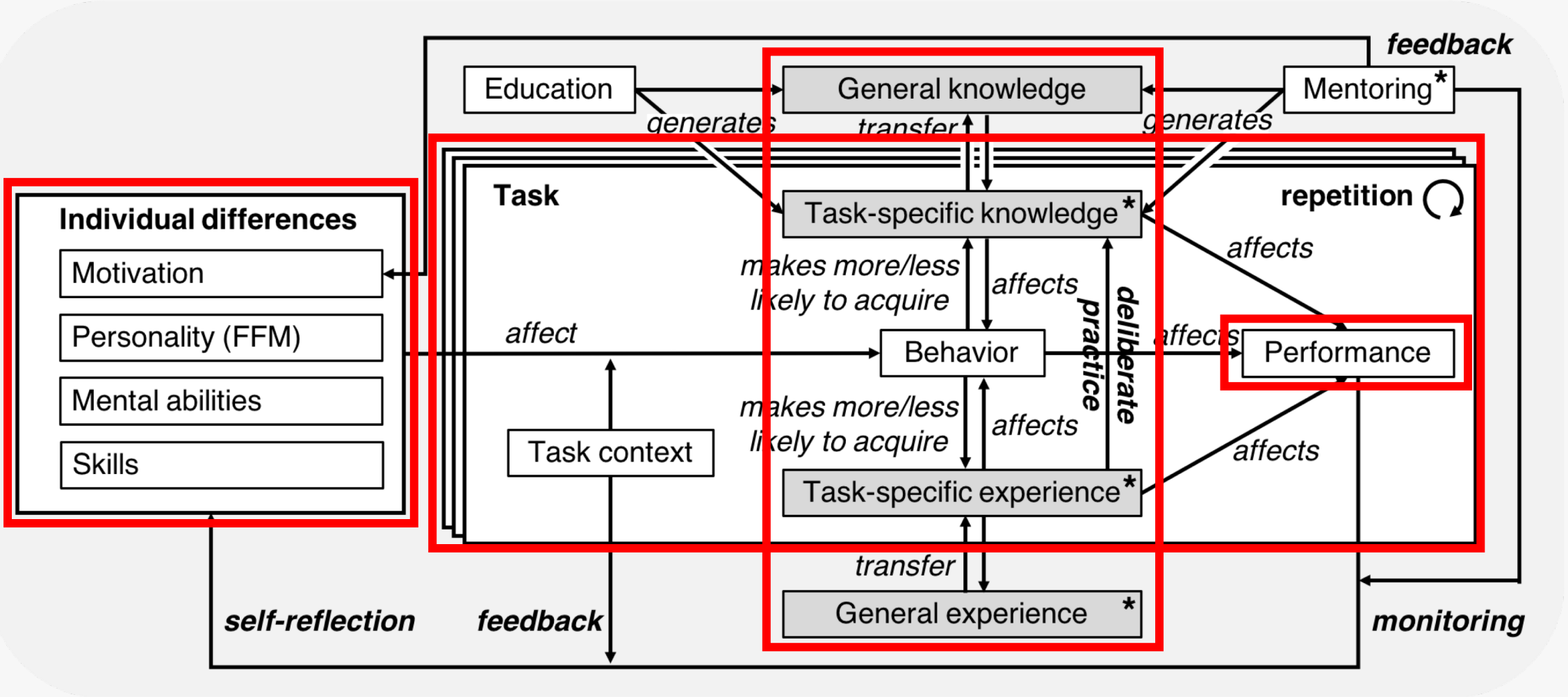


Building Theories in Software Engineering
Dag I.K. Sjøberg, Tore Dybå, Bente C.D. Anda, and Jo E. Hannay
in *Guide to Advanced Empirical Software Engineering* (2008).

Final Conceptual Theory



Final Conceptual Theory

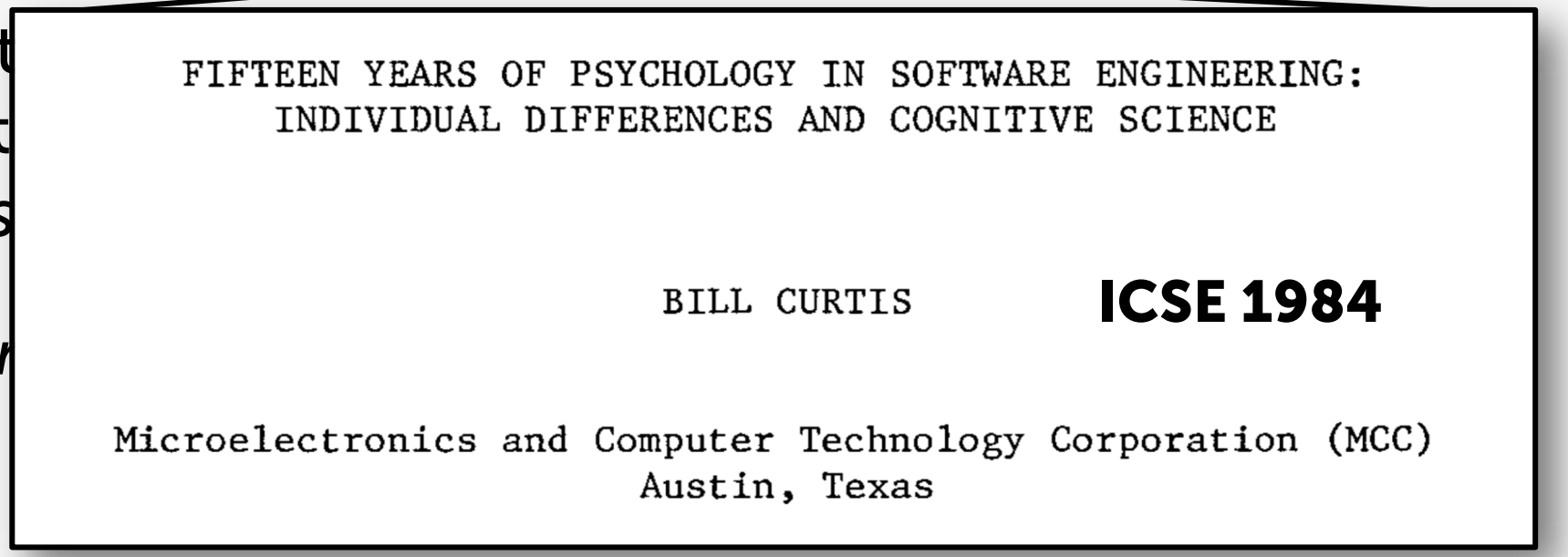


Knowledge

- **Knowledge** is a *"permanent structure of information stored in memory"* (Robillard, 1995)
- Developer's knowledge base considered (most) important factor influencing **performance** (Curtis, 1984)
- Studies suggest that this knowledge base is *"highly **language dependent**",* but experts also have *"abstract, **transferable** knowledge and skills"* (Sonnentag et al., 2006)
- *"Semantic"* vs. *"syntactical"* knowledge (Shneiderman and Mayer, 1978)

Knowledge

- **Knowledge** is a *"permanent structure of information stored in memory"* (Robillard, 1995)
- Developer's knowledge base considered (most) important factor influencing **performance** (Curtis, 1984)
- Studies suggest t
  dependent", but
  knowledge and s
- *"Semantic"* vs. *"syn*



FIFTEEN YEARS OF PSYCHOLOGY IN SOFTWARE ENGINEERING:
INDIVIDUAL DIFFERENCES AND COGNITIVE SCIENCE

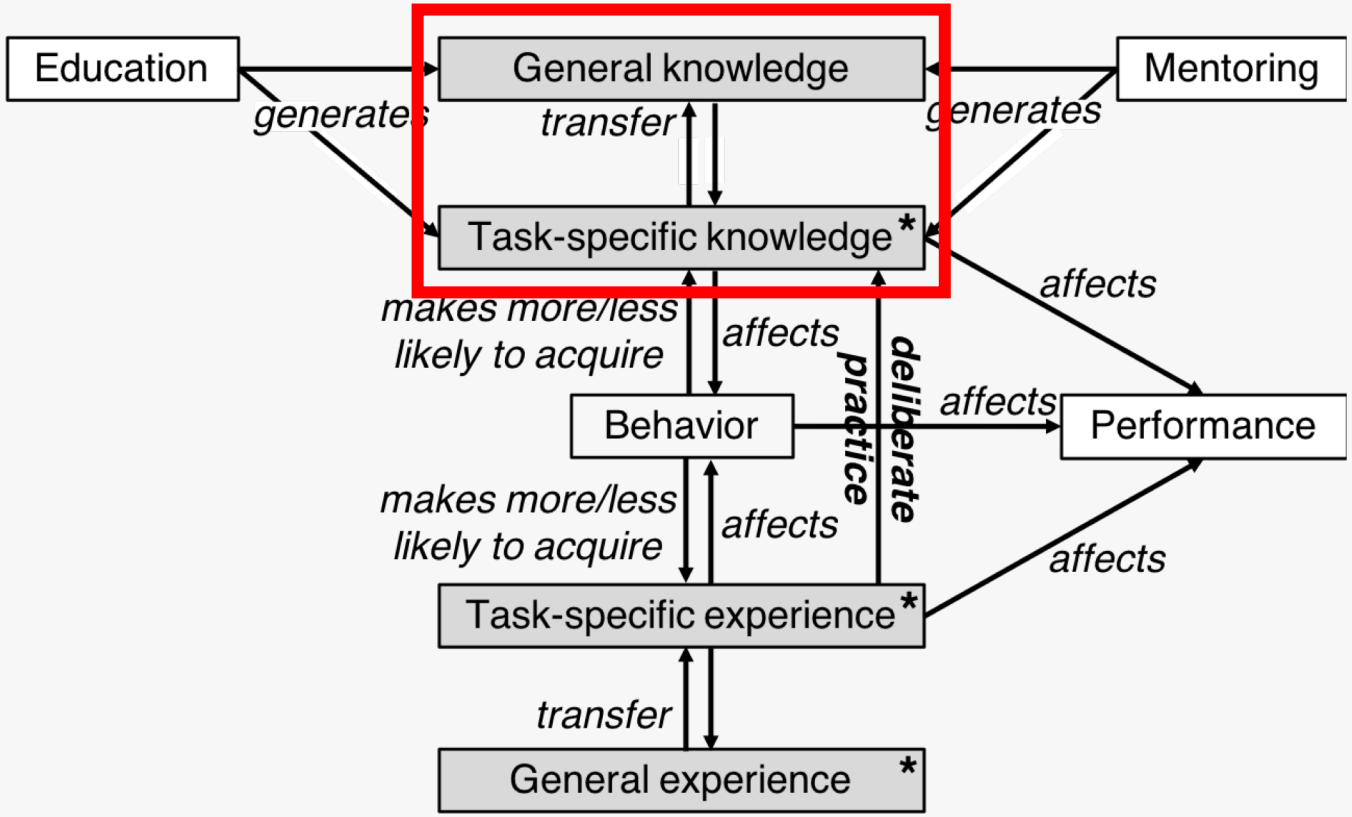
BILL CURTIS

ICSE 1984

Microelectronics and Computer Technology Corporation (MCC)
Austin, Texas

Knowledge

Knowledge about *"paradigms [...], data structures, algorithms, computational complexity, and design patterns"*

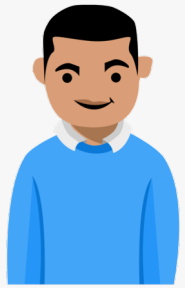


An *"intimate knowledge of the design and philosophy of the language"*

Experience

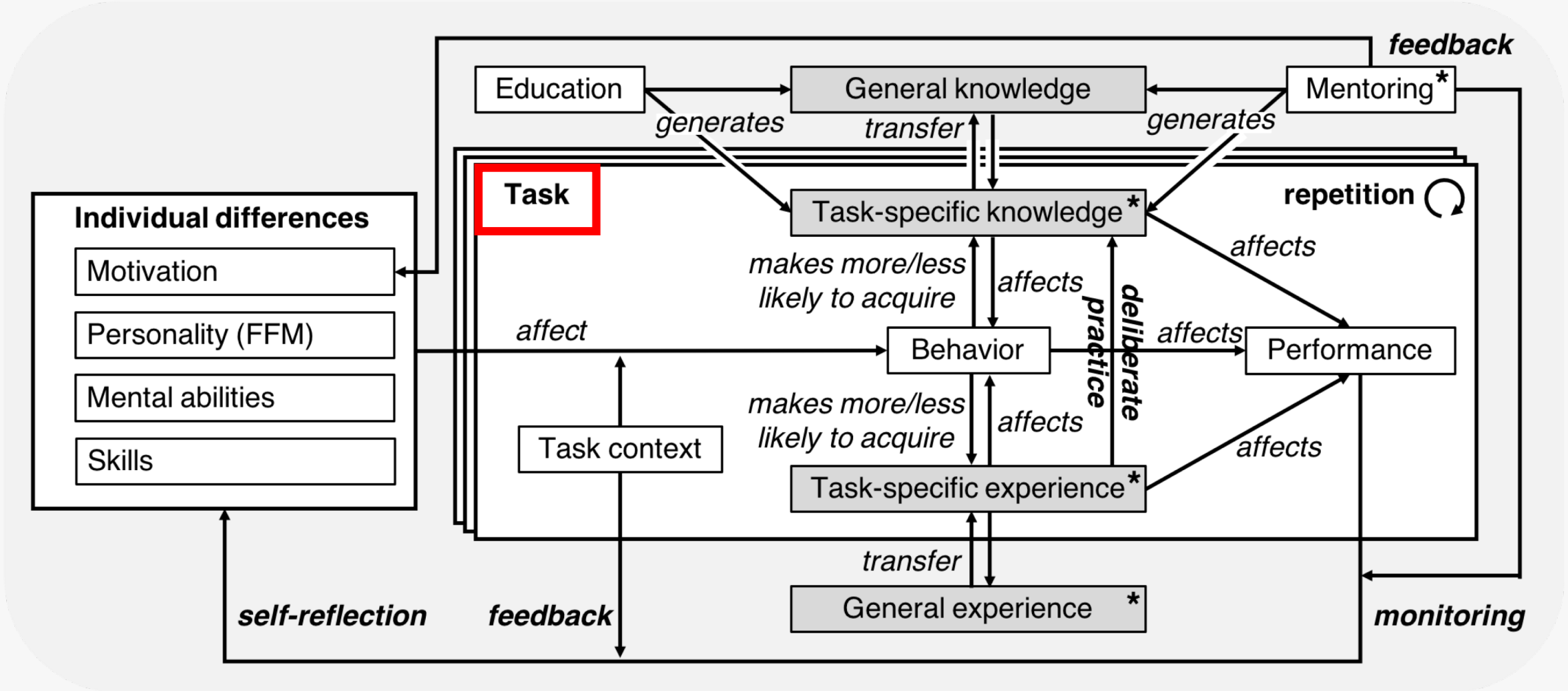
- Many participants mentioned not only the **quantity**, but also the **quality of experience**

Having built *„everything from small projects to enterprise projects“*



Having shipped *„a significant amount of code to production or to a customer“*

Final Conceptual Theory



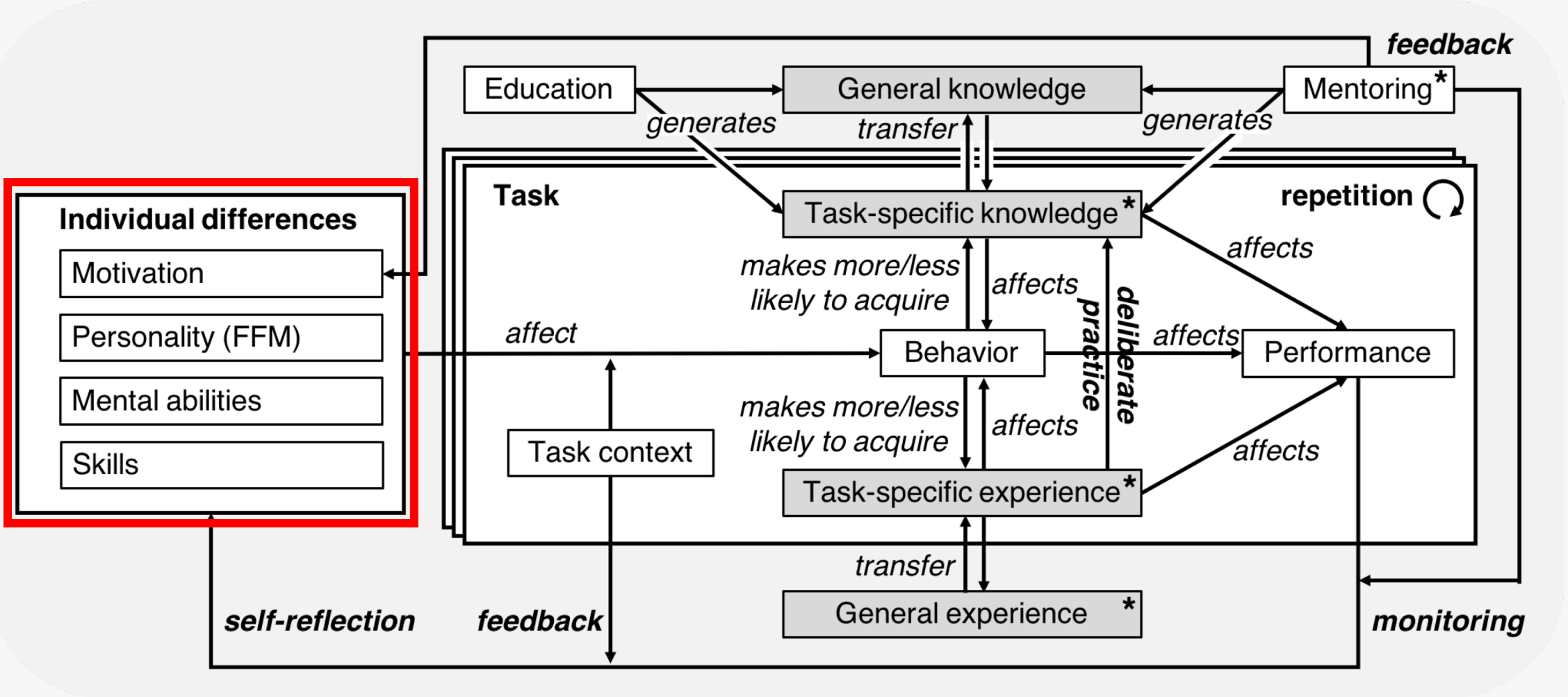
Tasks

- Asked participants to name the **three most important tasks** that a software development expert should be good at
- Most frequently mentioned:
  1. Designing a software architecture
  2. Writing source code
  3. Analyzing and understanding requirements
- Other mentioned tasks: testing, communicating, debugging

"Architecting the software in a way that allows flexibility in project requirements and future applications of the components"


Which factors influence expertise development over time?

Final Conceptual Theory

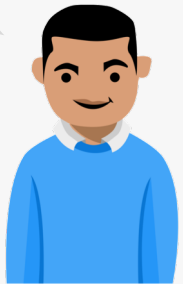


Individual Differences: Motivation

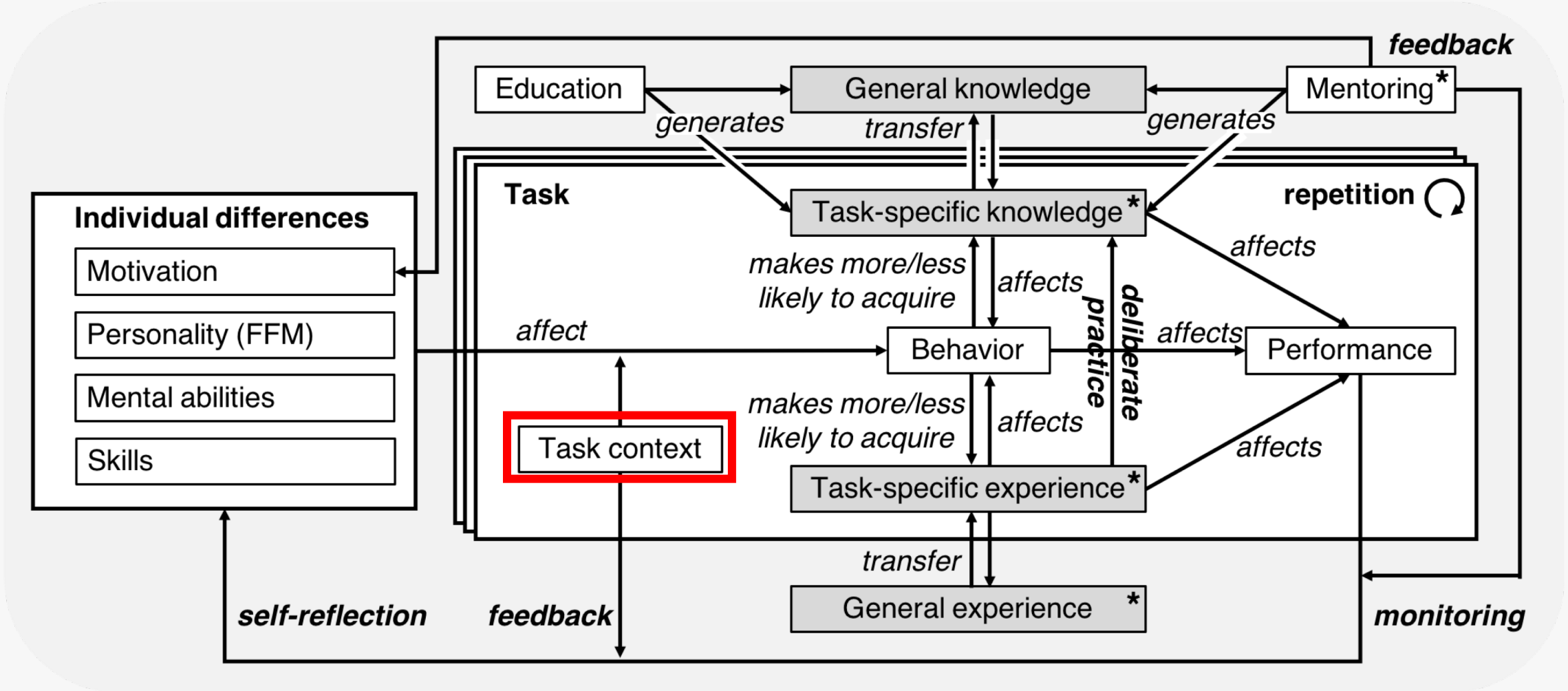
- Related work describes how **individual differences** affect expertise development
- Mental abilities and personality are relatively stable
- **Motivation can change** over time

- Many participants **intrinsically motivated:**
  - Problem solving
  - Seeing a high-quality solution
  - Creating something new
  - Helping others

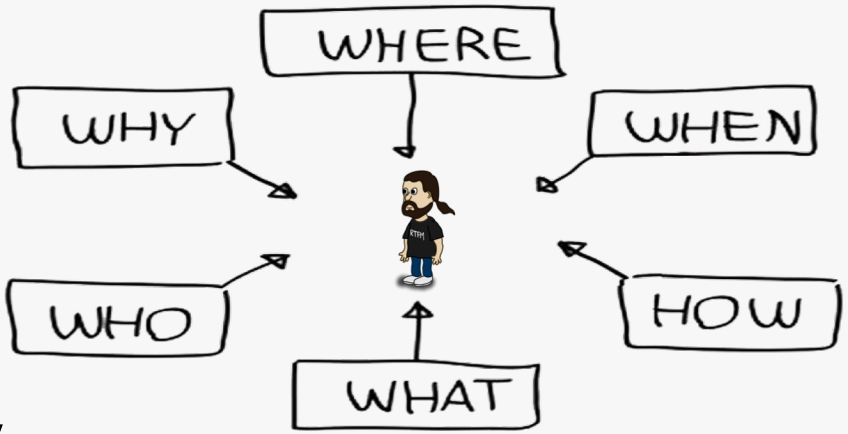
*"The initial design is fun, but what really is more rewarding is **refactoring**."*



Final Conceptual Theory



Task Context

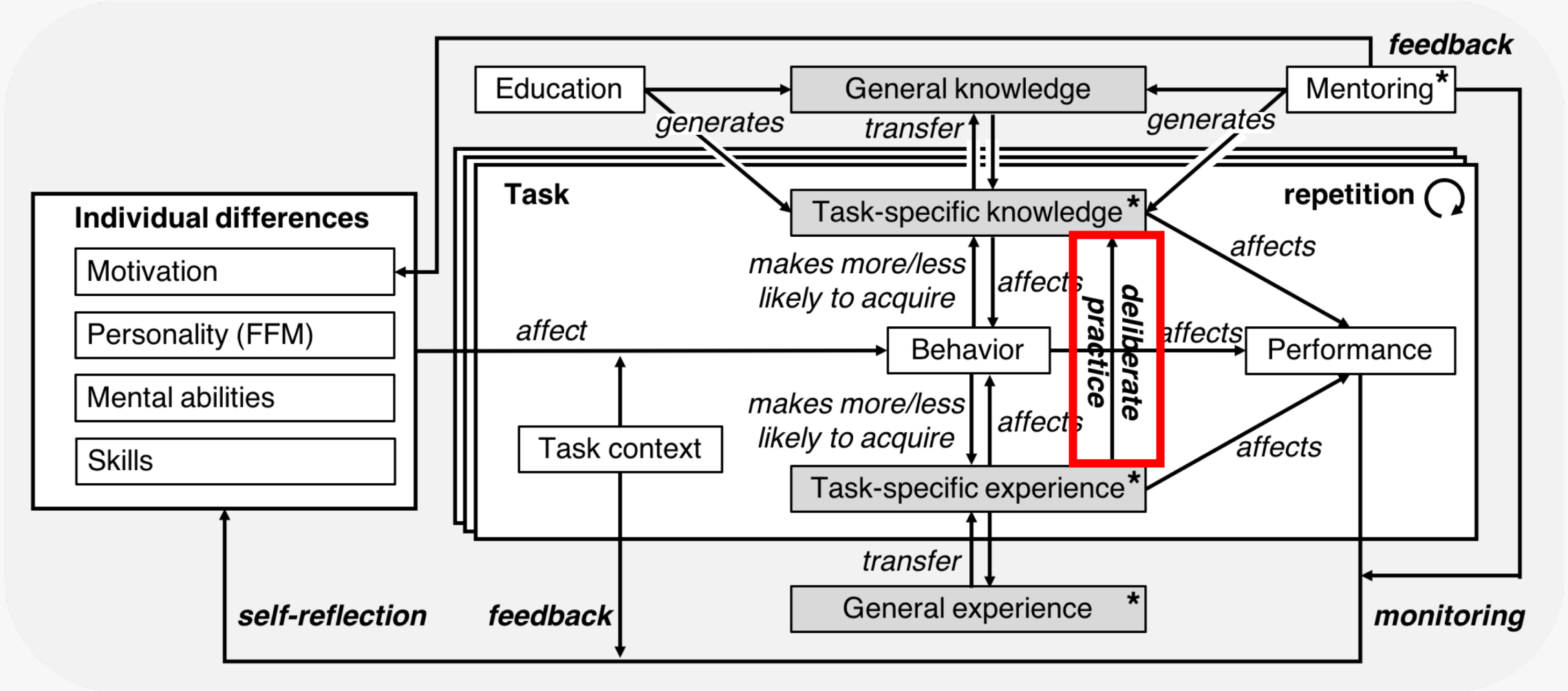


- Work **environment**
  (office, coworkers, customers etc.)
- Project **constraints**
  (external dependencies, time, etc.)
- Can either **foster or hinder** expertise dev.
- We asked: *What can employers do?*

1. Encourage learning
   (training courses, library, monetary incentives)
2. Encourage experimentation
   (side projects, being open to new ideas/technologies)
3. Improve information exchange
   (facilitate meetings, rotating between teams/projects)
4. Grant freedom
   (less time pressure)

http://kratinmobile.com/blog/wp-content/uploads/2014/09/context3.png

Final Conceptual Theory



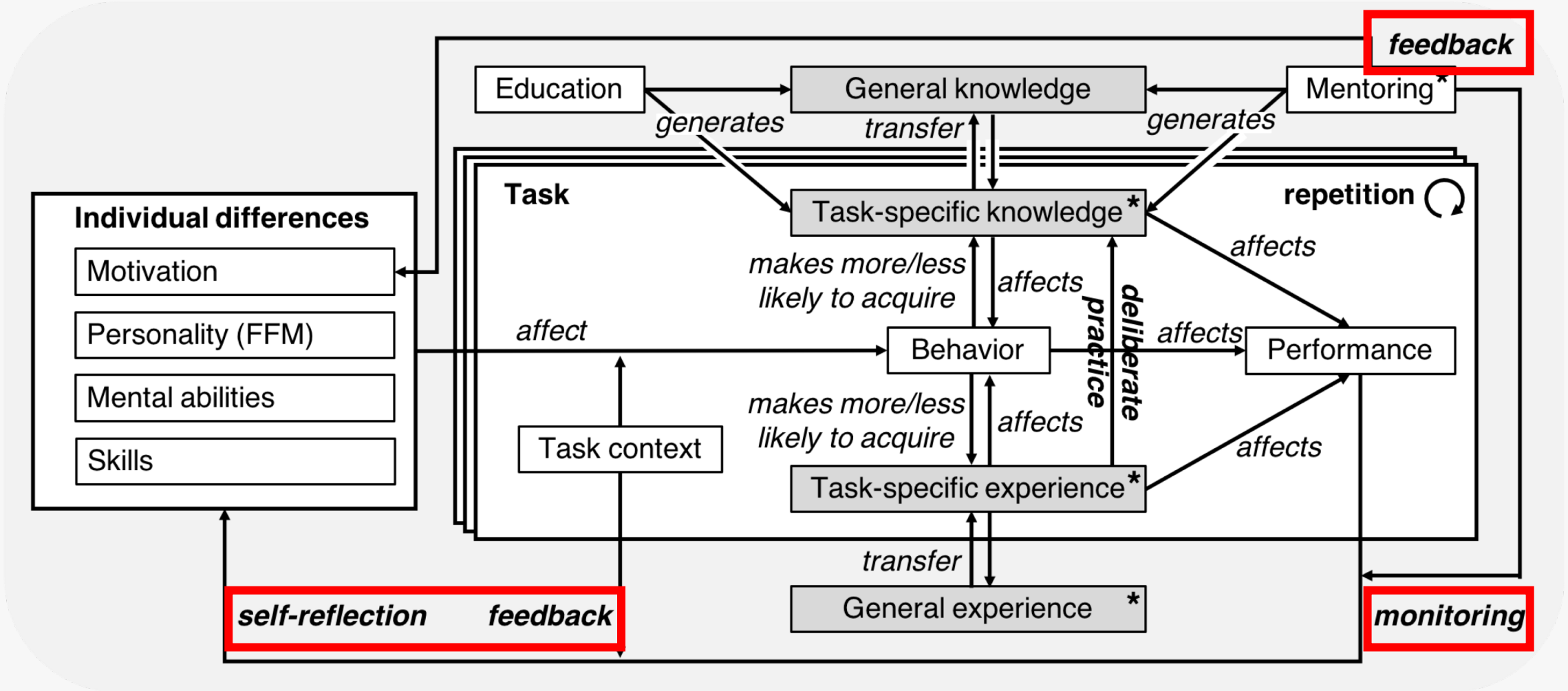
Deliberate Practice

- Having **more experience** does not automatically lead to **better performance** (Ericsson et al., 1993)
- Performance may even **decrease** over time (Feltovich, 2006)
- Length of experience only weak correlate of job performance (Ericsson, 2006)
- Deliberate practice: *„**Prolonged efforts to improve** performance while negotiating motivational and external constraints“* (Ericsson et al., 1993)

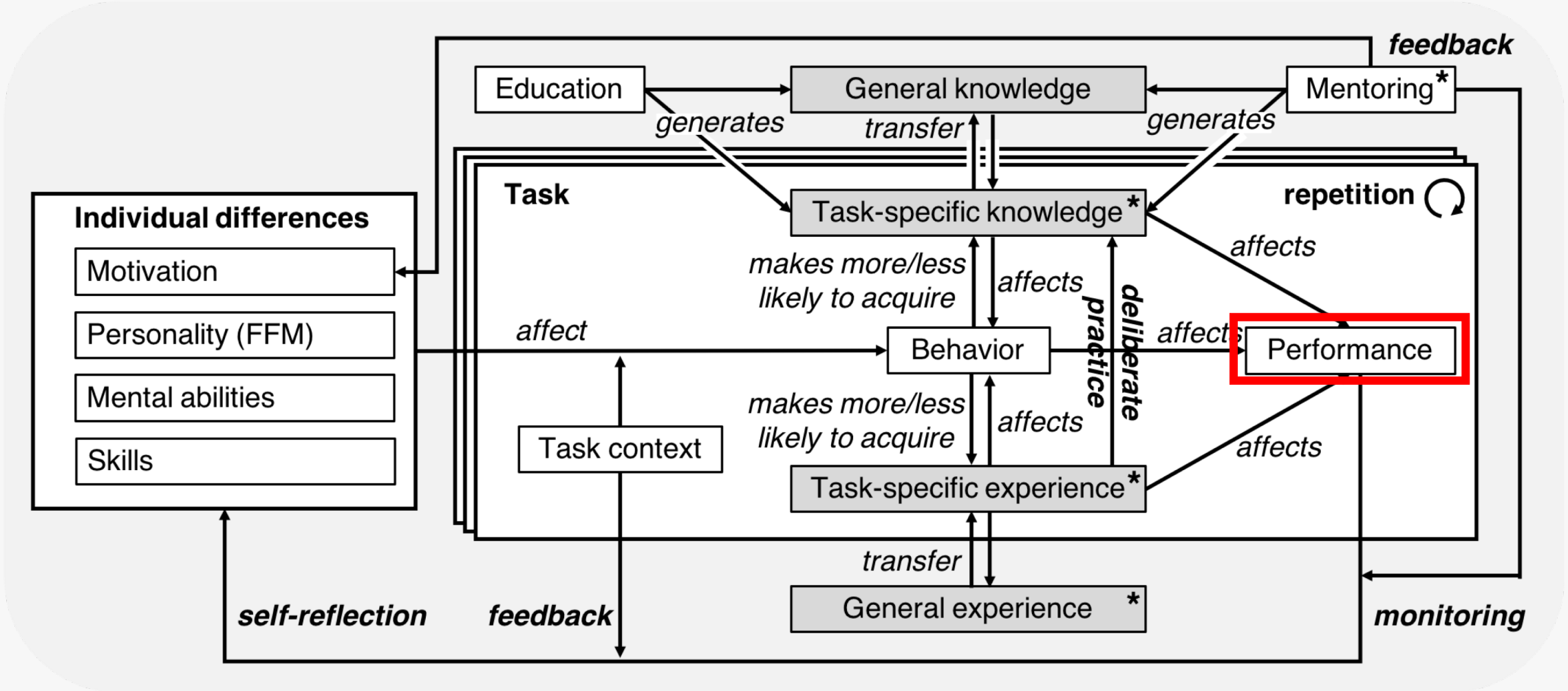
Deliberate Practice: Self-Reflection

- **(Self-)reflection** and **feedback** important to **monitor** progress towards goal achievement (Locke and Latham, 1990)
- *"[T]he more **channels of accurate and helpful feedback** we have access to, the better we are likely to perform."*
  (Tourish and Hargie, 2003)
- **Mentors**, teachers, and peers are an important sources for feedback

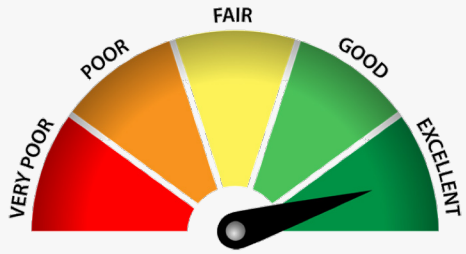
Final Conceptual Theory



Final Conceptual Theory



Performance



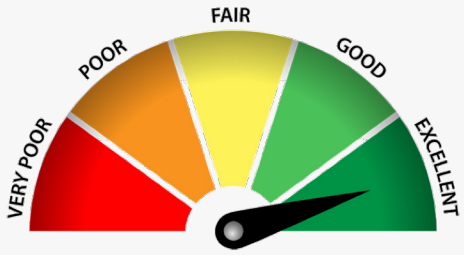
Scope of this work:

- We do **not** treat performance as a **dependent variable** that we try to explain for individual tasks
- We consider different **performance monitoring** approaches to be a means for feedback and self-reflection

Long-term goal:

- Build **variance theory** for explaining and predicting the development of expertise

Performance



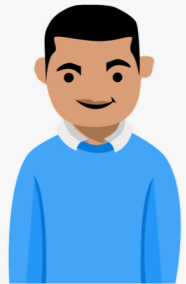
- Participants described different **properties of expert's source code** (well-structured, readable, maintainable, etc.)

„Everyone can write [...] code which a machine can read and process but the key lies in writing concise and understandable code which [...] ***people who have never used that piece of code before [can read]****."*

Performance Decline

- Goal: Identify factors **hindering** expertise development
- **41.5%** of participants observed a **significant performance decline** over time (for themselves or others)
- Reasons:
  - Demotivation
  - Changes in the work environment
  - Age-related decline
  - Changes in attitude
  - Shifting towards other tasks

*"I perceived an **increasing procrastination** in me and in my colleagues, by **working on the same tasks** over a relatively long time [...] **without innovation and environment changes**."*



Age-Related Performance Decline

"For myself, it's mostly the effects of aging on the brain. At age 66, ***I can't hold as much information short-term memory****, for example. [...] I can compensate for a lot of that by writing simpler functions with clean interfaces. The results are still good, but* ***my productivity is much slower than when I was younger****."*

software architect, age 66

"Programming ability is based on **desire to achieve**. In the early years, it is a sort of **competition**. [...] I found that I lost a significant amount of my focus as I became 40, and started **using drugs such as ritalin** to enhance my abilities. This is pretty common among older programmers."

software developer, age 60

How are experience and expertise related?

Experience vs. Expertise

- Self-assessment with **semantic differential** (novice to expert) and **Dreyfus expertise model**

Semantic Differential Scale

- Beginning of survey:
  Please rate your Java programming expertise on the following scale:

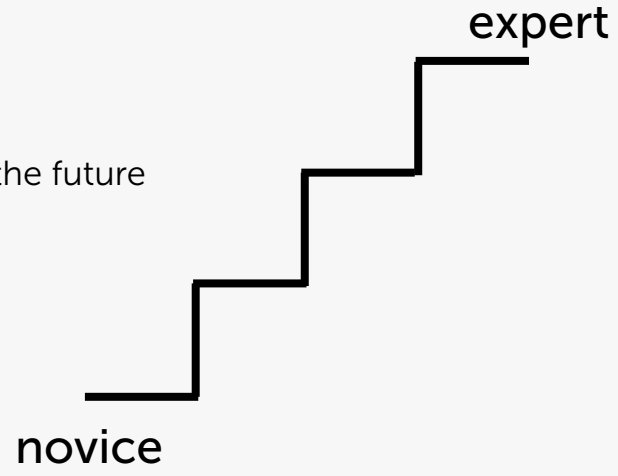
1 (Novice)	*2*	*3*	*4*	*5*	*6 (Expert)*
○	○	○	○	○	○

...

- End of survey:
  Please rate your own Java programming expertise according to the five stages described below.

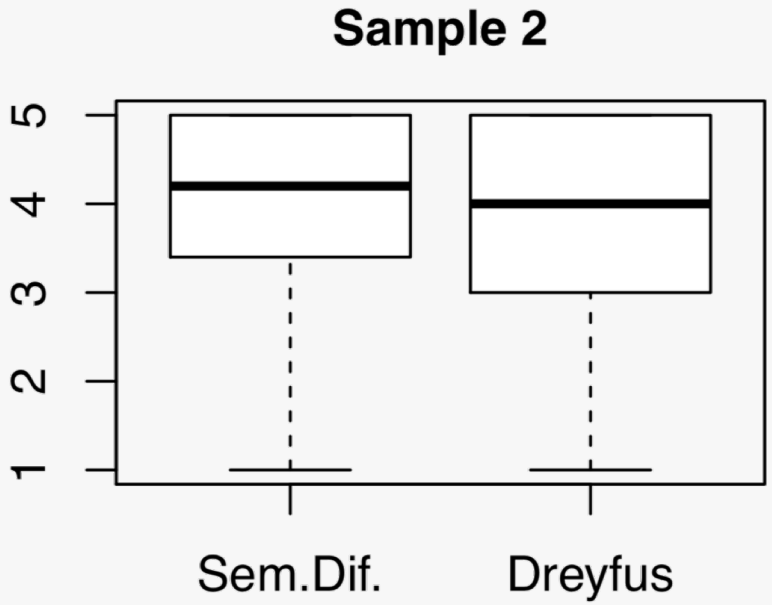
Discrete Expertise Model

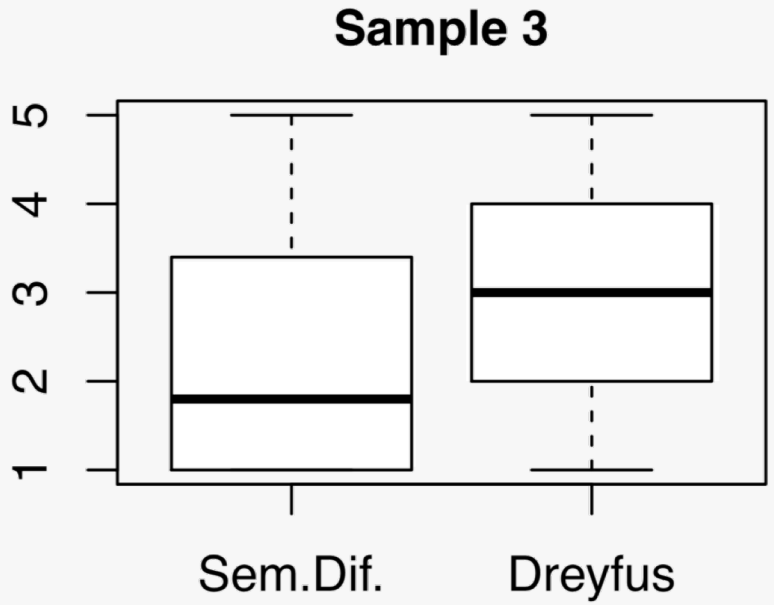
- Stage 1 (Novice):
  - has little or no experience
  - wants unambiguous rules to accomplish his/her tasks
  - is able to handle small, isolated tasks
- Stage 2 (Advanced Beginner):
  - has gained some experience
  - can work more independently than a novice
  - knows general principles in a limited context, but does not have a holistic understanding ("big picture")
- Stage 3 (Competent):
  - has a holistic understanding of the problem domain
  - bases his/her work on deliberate planning and extensive past experience
  - can apply general maxims (e.g. design patterns) easily to specific contexts
- Stage 4 (Proficient):
  - has a vast amount of experience that he/she can intuitively apply to new contexts
  - can easily differentiate between irrelevant and important details
  - constantly reflects on what he/she has done and revises own approach to perform better in the future
- Stage 5 (Expert):
  - he/she is a major source of knowledge and information for others
  - primarily works from his/her intuition



Experience vs. Expertise

- Self-assessment with **semantic differential** (novice to expert) and **Dreyfus expertise model**
- More experienced developers **adjusted** their ratings when context was provided, less experienced not

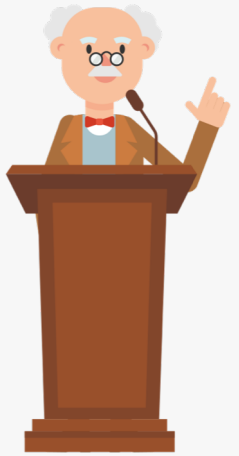




https://www.mirror.co.uk/news/weird-news/how-mcdonalds-takeaway-bag-ended-9664800

Summary for Researchers

- Can use our results when **designing studies** involving expertise **self-assessments** or our **theory building** approach
- Clear understanding what distinguishes novices and experts: **Provide** this **context** when asking for **self-assessed expertise** and later report it together with the results
- Can use theory to **design experiments** (first operationalizations described in paper)
- Future Work: Operationalization, develop **standardized description** of novice and expert for certain tasks

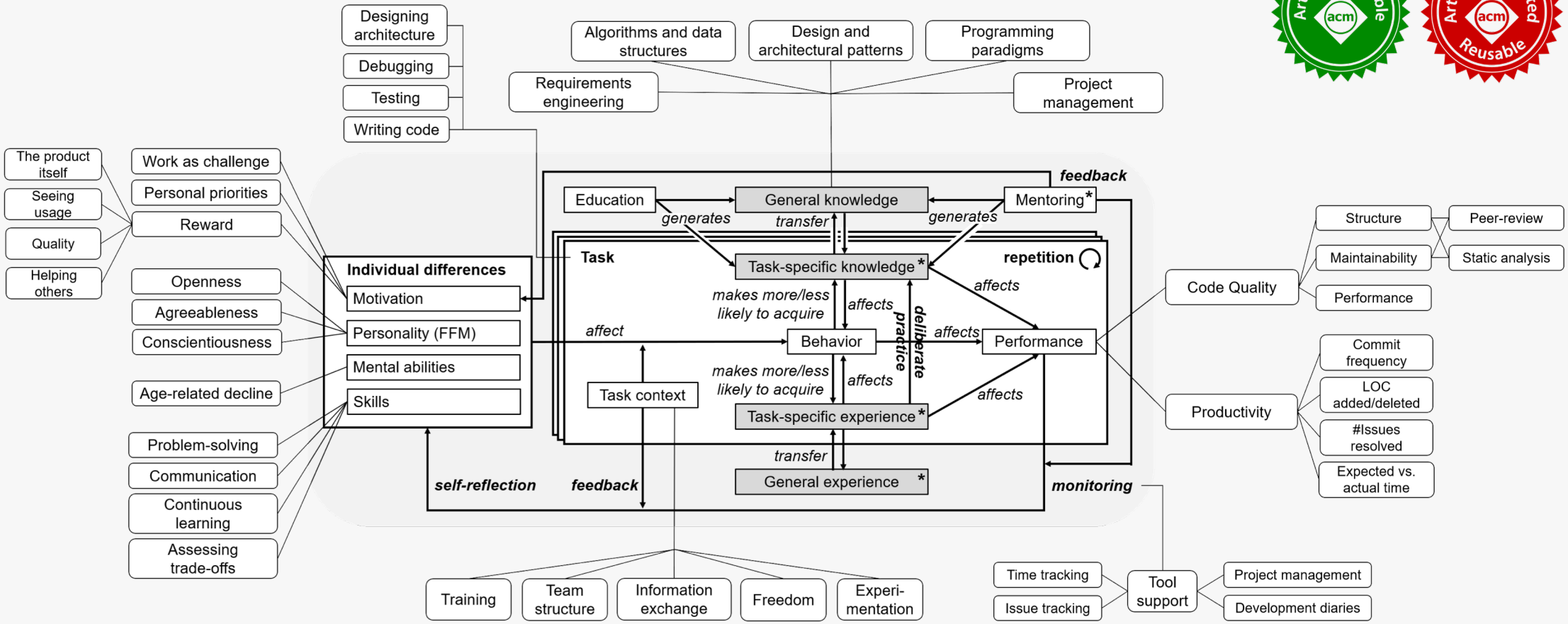


Summary for Developers

- See which **attributes** other developers assign to experts
- Learn which **behaviors** may lead to becoming a better software developer:
  - Deliberate practice
  - Have challenging goals
  - Build or maintain a supportive work environment (also for others)
  - Ask for feedback from peers
  - Reflect about what one knows and what not

Summary for Employers

- Learn what **(de)motivates** their employees:
  - Main motivation: problem solving
  - Main demotivation: non-challenging work
- Ideas on how to build supportive work environment **supporting self-improvement** of staff:
  - Good mix of continuity and change in software development process
  - Communicate clear visions, directions, and goals
  - Reward high-quality work wherever possible
  - Revisit information sharing in company



Sebastian Baltes

@s_baltes

expertise.sbaltes.com

Data and scripts available on Zenodo