Towards a Theory of Software Development Expertise

Sebastian Baltes

@s_baltes

Universität Trier

Talk @ George Mason University

November 12, 2018, Fairfax, Virginia, USA

Corresponding Research Paper

Towards a Theory of Software Development Expertise

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

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

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 developers [63], or the time it takes for developers to become productive in software projects [117]. However, there is no theory combining those individual aspects. Such a theory could help structuring existing knowledge about SDExp in a concise way and hence facilitate its communication [44]. Despite [illegible] favor

ESEC/FSE 2018

Software Development Expertise?

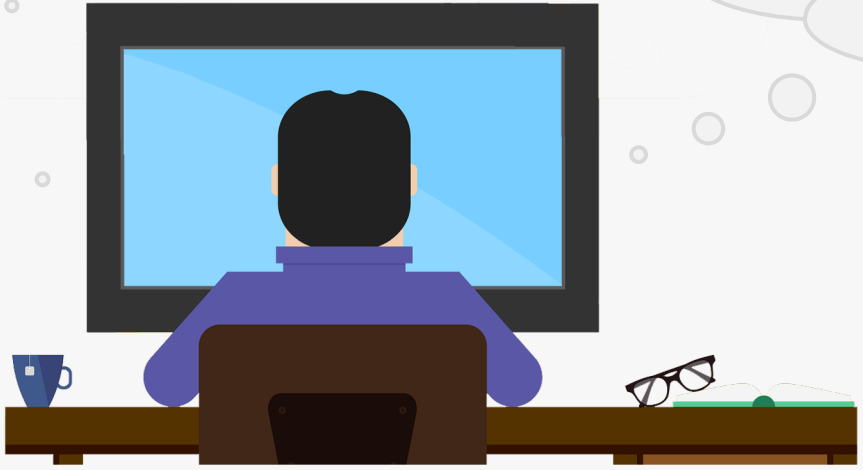
Implementing new features

Data structures

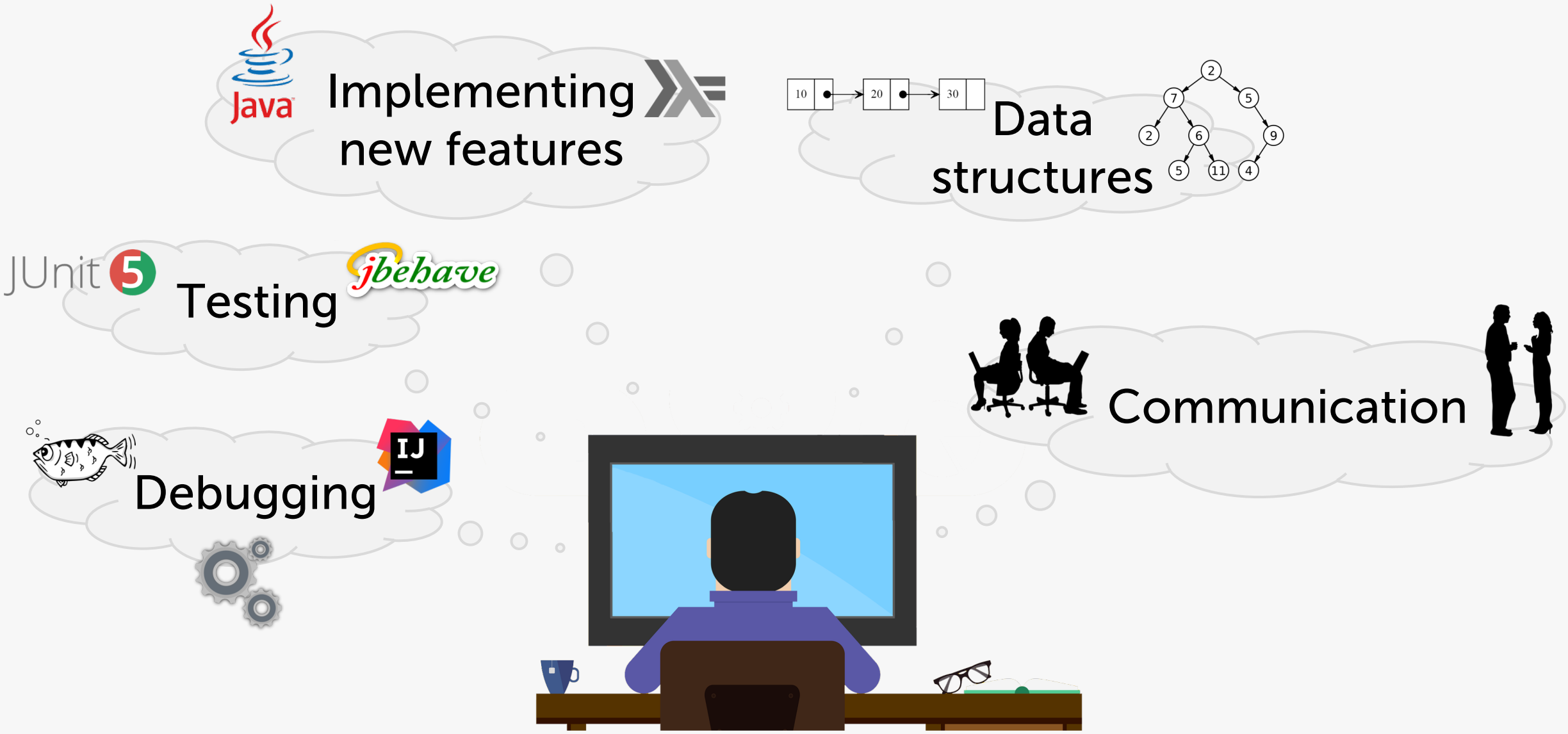
Testing

Communication

Debugging



Software Development Expertise?



How to structure all those expertise-related aspects?

Which factors influence expertise development over time?

How are experience and expertise related?

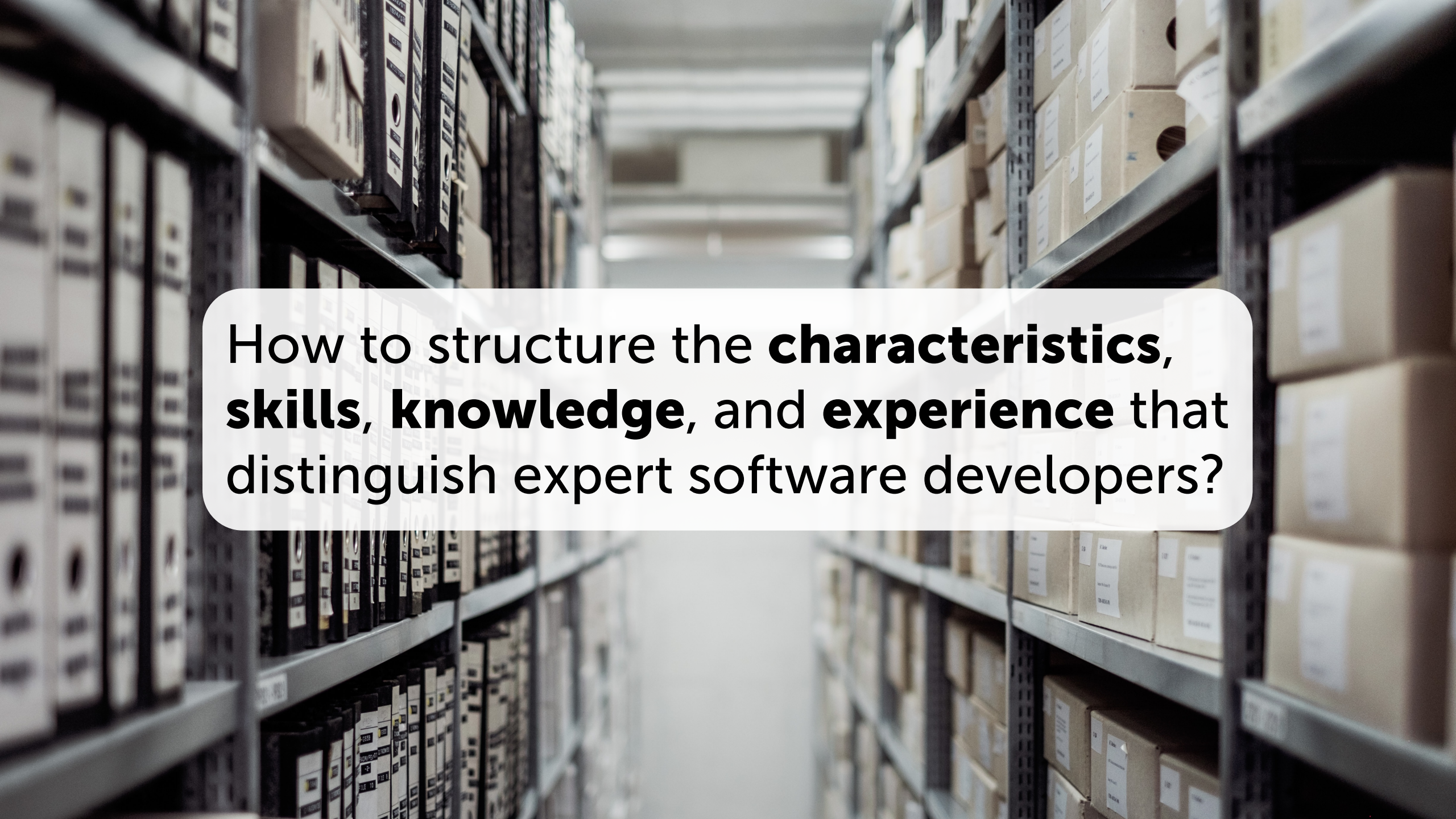
Definitions

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

Merriam-Webster

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

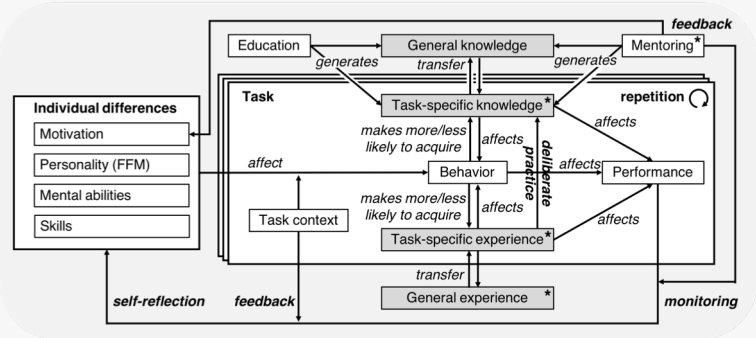
K. Anders Ericsson

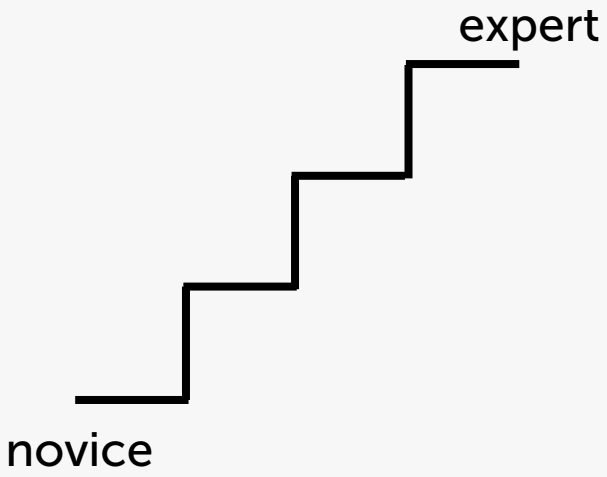


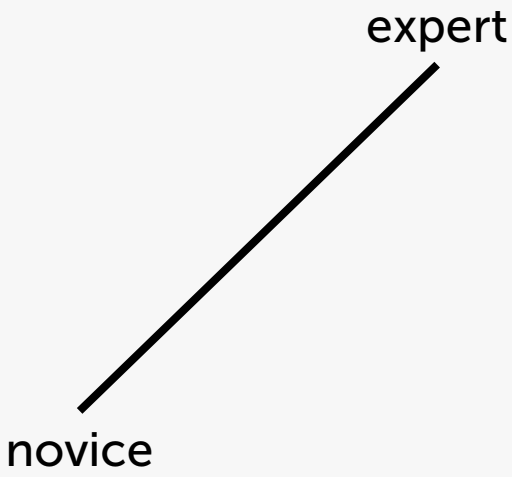
How to structure the **characteristics**, **skills**, **knowledge**, and **experience** that distinguish expert software developers?

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







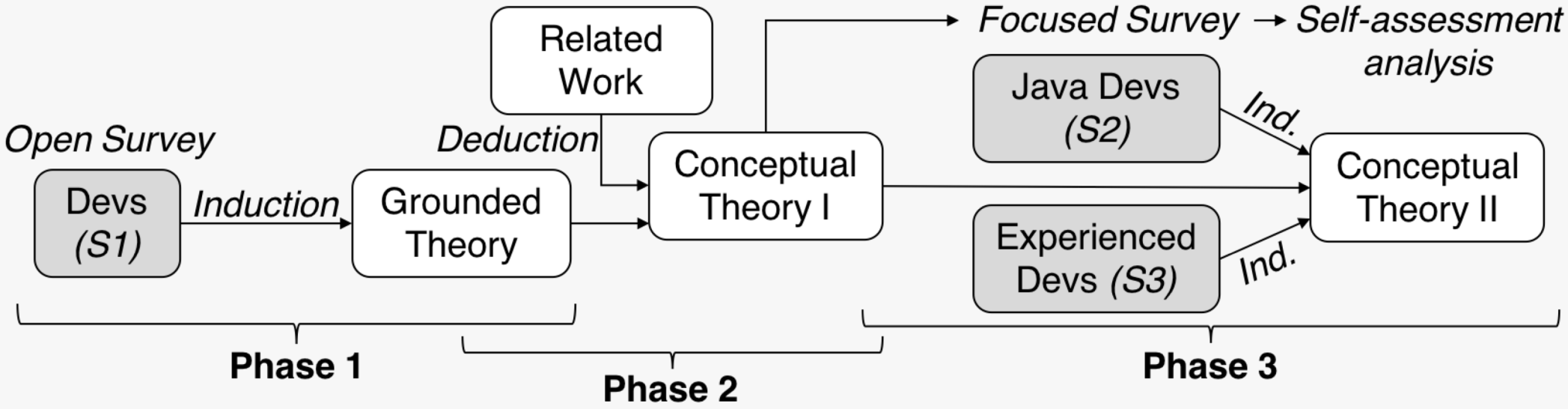
Theory Classification

- A **process theory** intends to explain and understand *"how an entity changes and develops"* over time (Ralph, 2018)

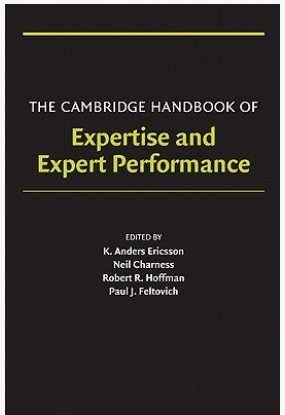
- In a **teleological process theory**, an entity *"constructs an envisioned end state, takes action to reach it, and monitors the progress"* (van de Ven and Poole, 1995)

- **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"*



Research Design

The Oxford Handbook of Expertise

Edited by Paul Ward, Jan Maarten Schraagen, Julie Gore, and Emilie M. Roth

Abstract

This handbook is currently in development, with individual articles publishing online in advance of print publication. At this time, we cannot add information about unpublished articles in this handbook, however the table of contents will continue to grow as additional articles pass through the review process and are added to the site. Please note that the online publication date for this handbook is the date that the first article in the title was published online. For more information, please read the site FAQs.

Keywords: gifted, gifted and talented, talent development, theories of intelligence, team expertise, expertise development, team reflection, team reflexivity, team debriefing, aging, development, knowledge representation, skill, cognition, self-regulation, skill decay, skill retention, enhancing retention, mitigating loss, training, expertise, skill acquisition, adaptable performance, transfer, skill reacquisition, experts, expertise, best practices, evidence-based performance, heuristics and biases, sociology, artificial intelligence

Bibliographic Information

ISBN: 9780198795872 Published online: Oct 2018

DOI: 10.1093/oxfordhb/9780198795872.001.0001

Find at OUP.com

Google Preview

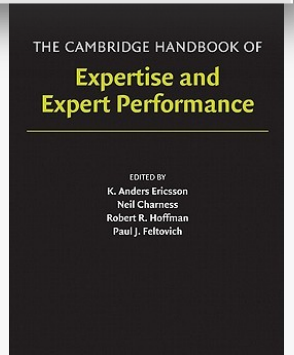
EDITORS

Paul Ward, *editor*
Paul Ward, University of Northern Colorado, USA

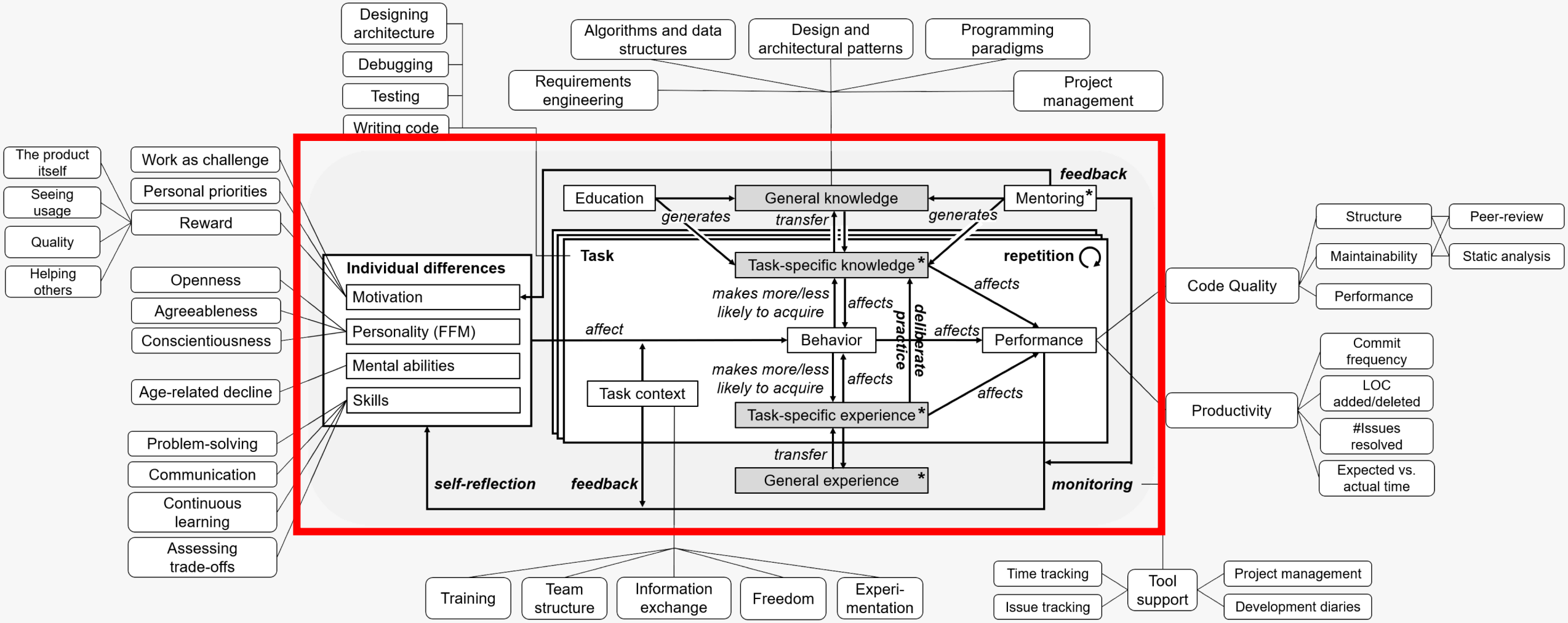
Jan Maarten Schraagen, *editor*
Jan Maarten Schraagen, University of Twente, Netherlands

Julie Gore, *editor*
Julie Gore, University More

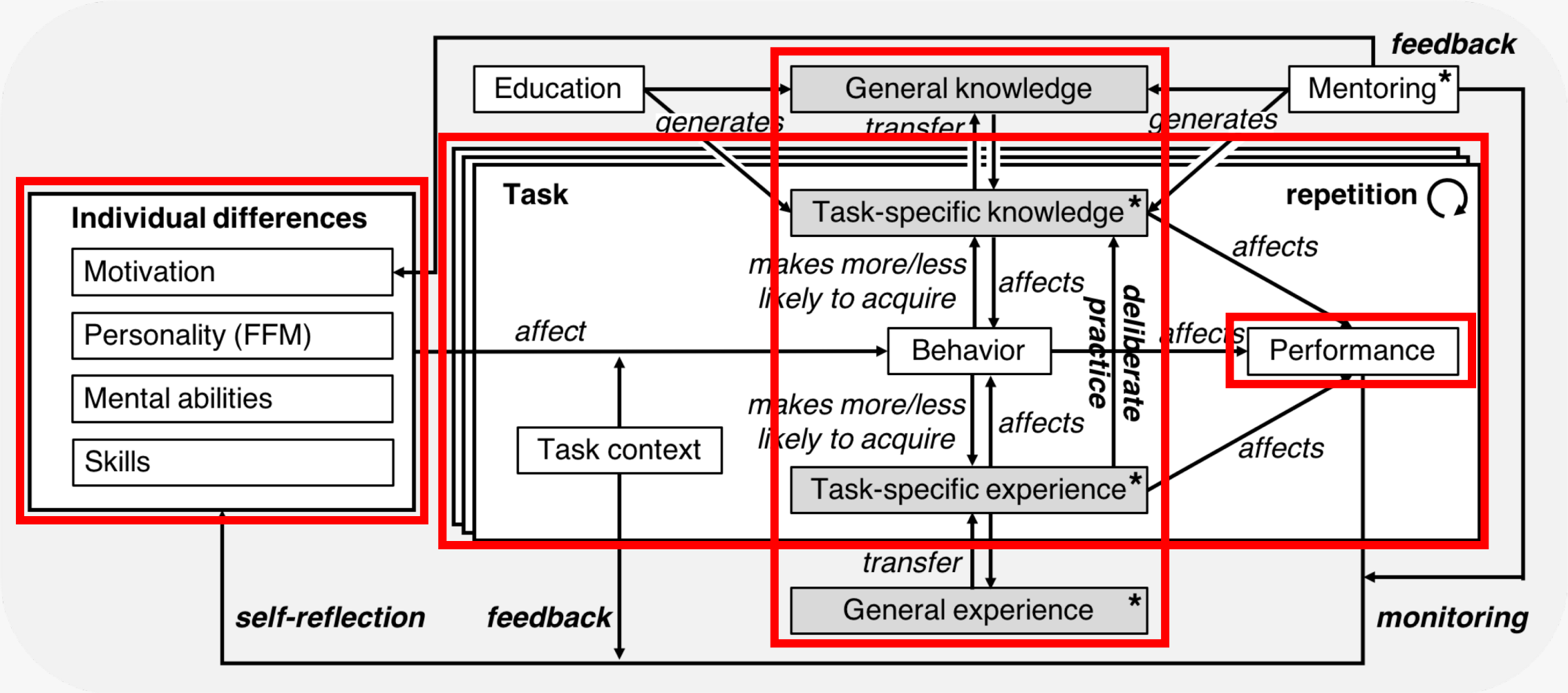
- **Deduction:** Main source *"Cambridge Handbook of Expertise and Expert Performance"*



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

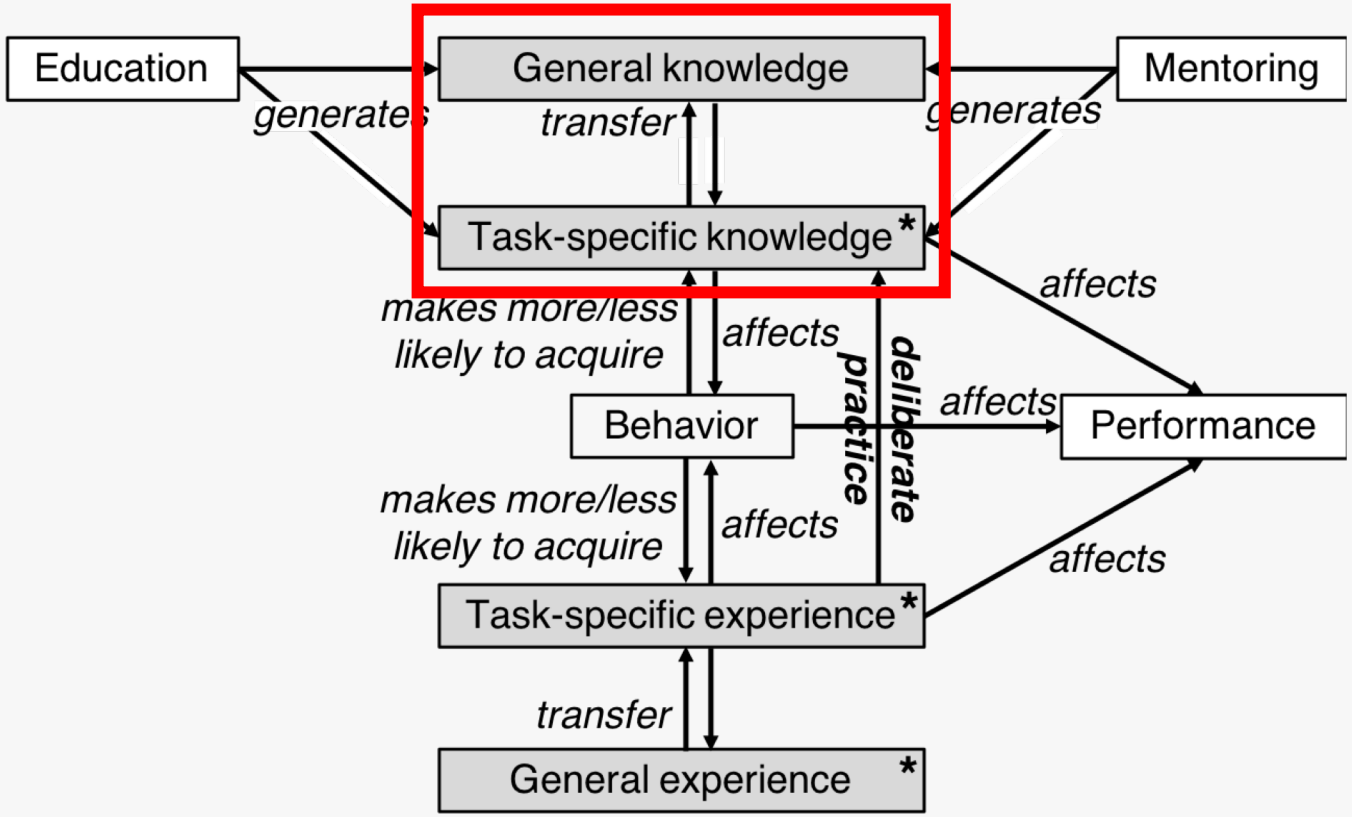
BILL CURTIS

Microelectronics and Computer Technology Corporation (MCC)
Austin, Texas

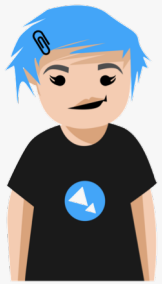
ICSE 1984
(Orlando, FL, USA)

Knowledge

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



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



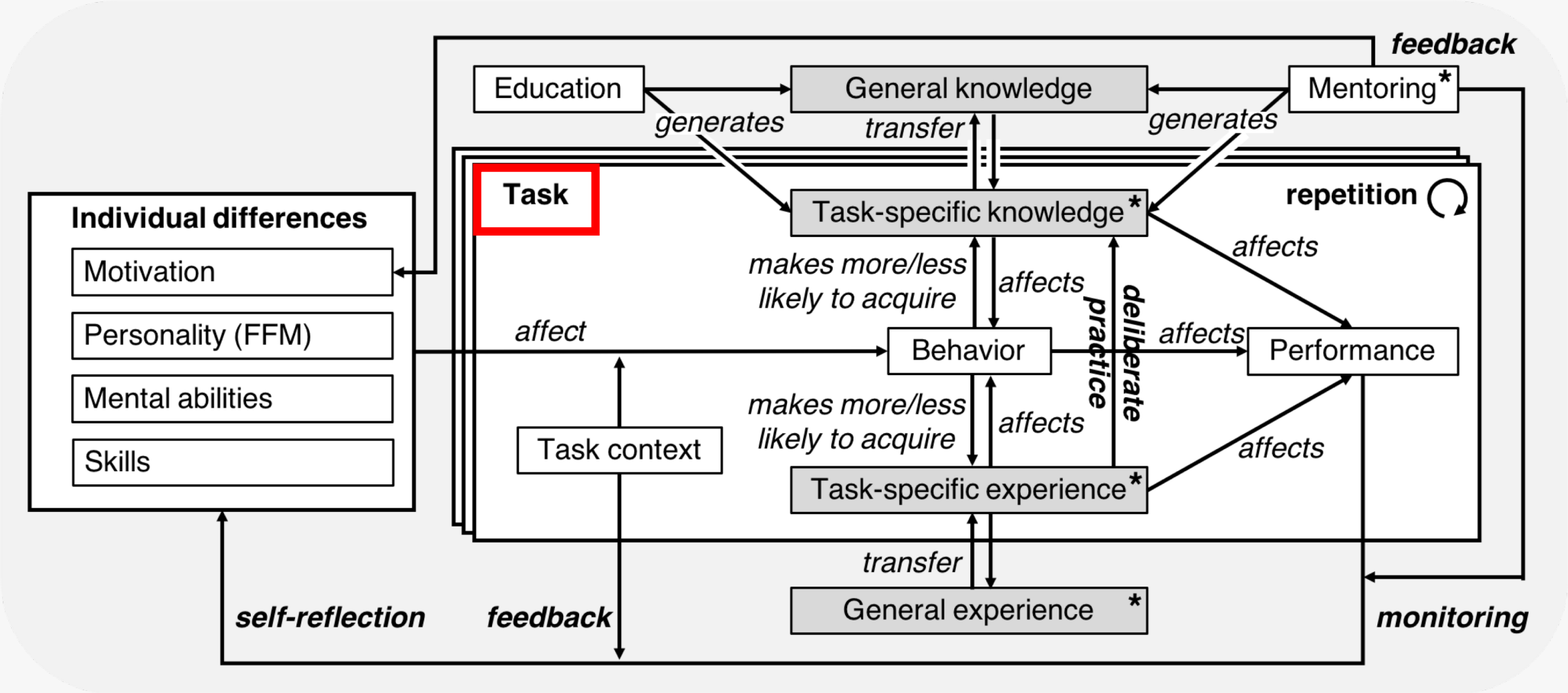
Experience

- 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



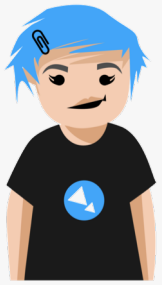
Tasks

- 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

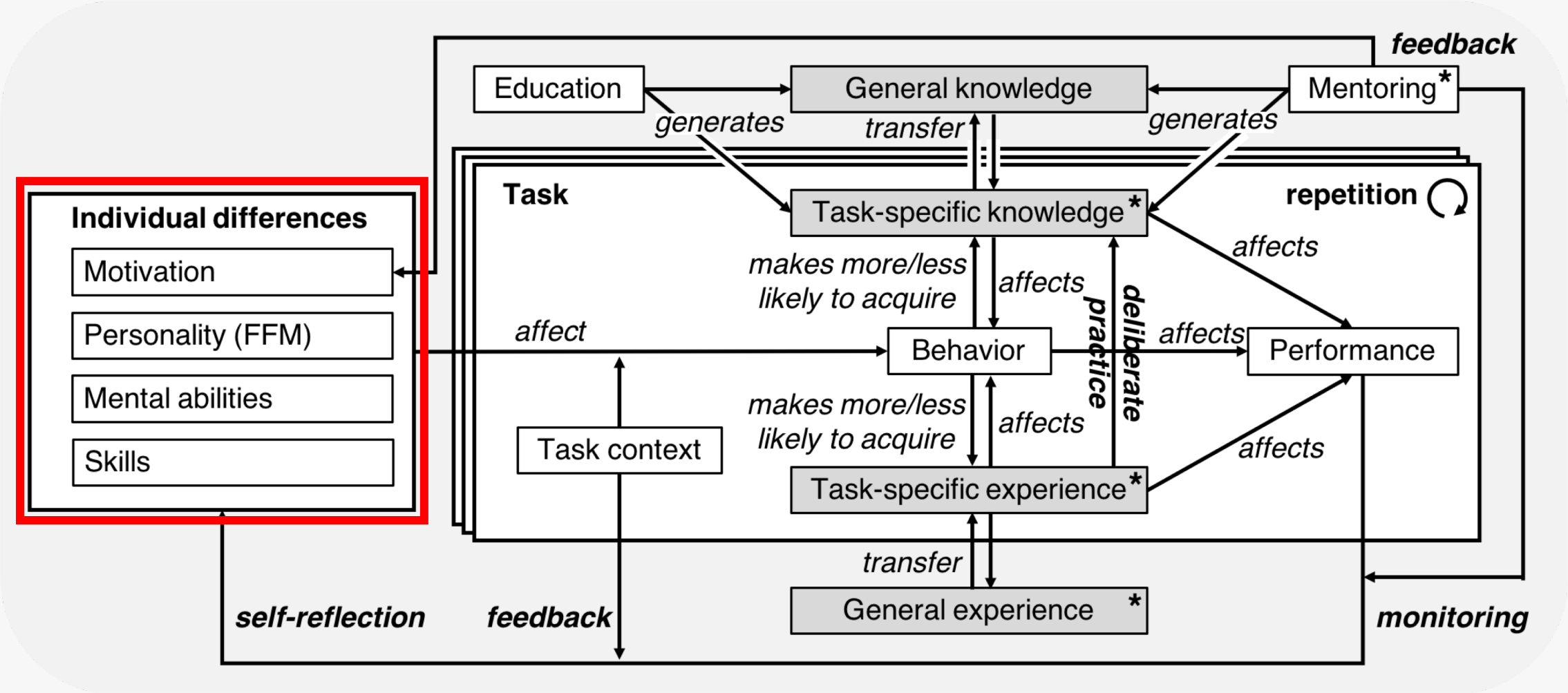
- Other mentioned tasks: testing, communicating, debugging

"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

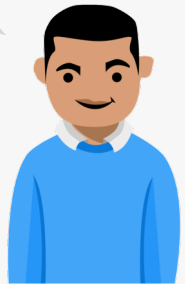


Individual Differences: Motivation

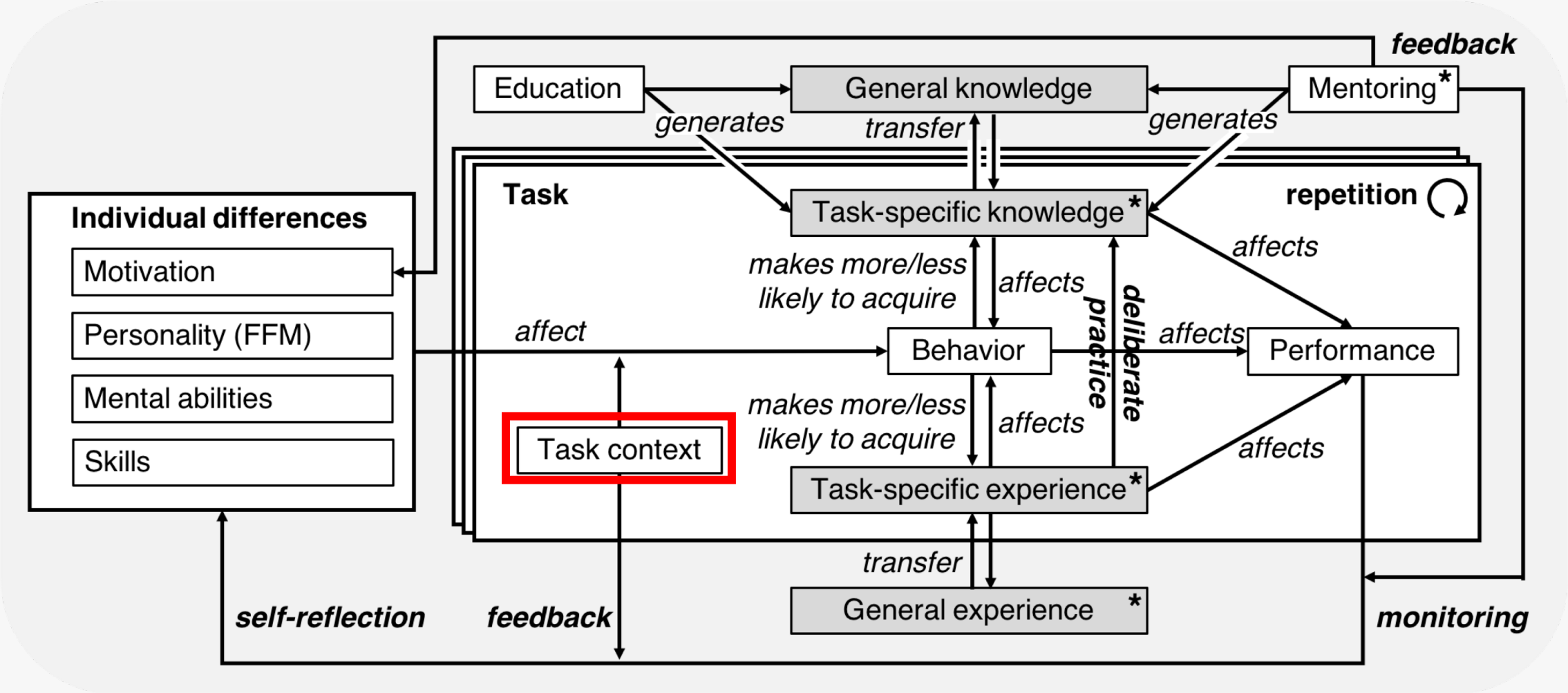
- Related work describes how **individual differences** affect expertise development
- Mental abilities and personality are relatively stable
- **Motivation can change** over time

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

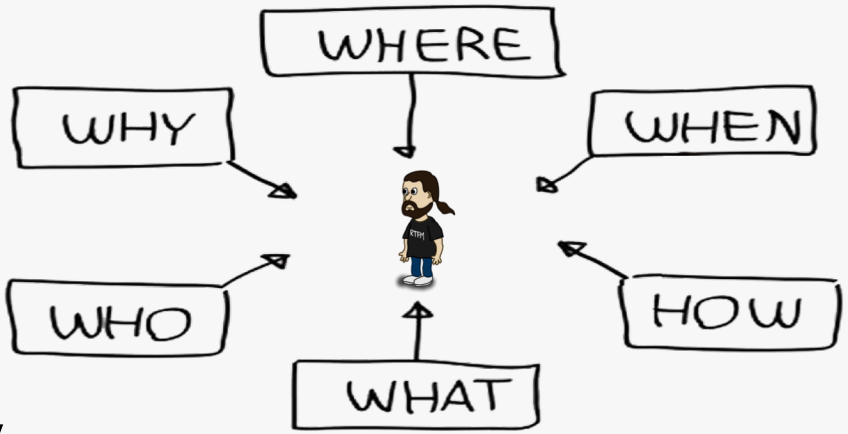
*"The initial design is fun, but what really is more rewarding is **refactoring**."*



Final Conceptual Theory



Task Context

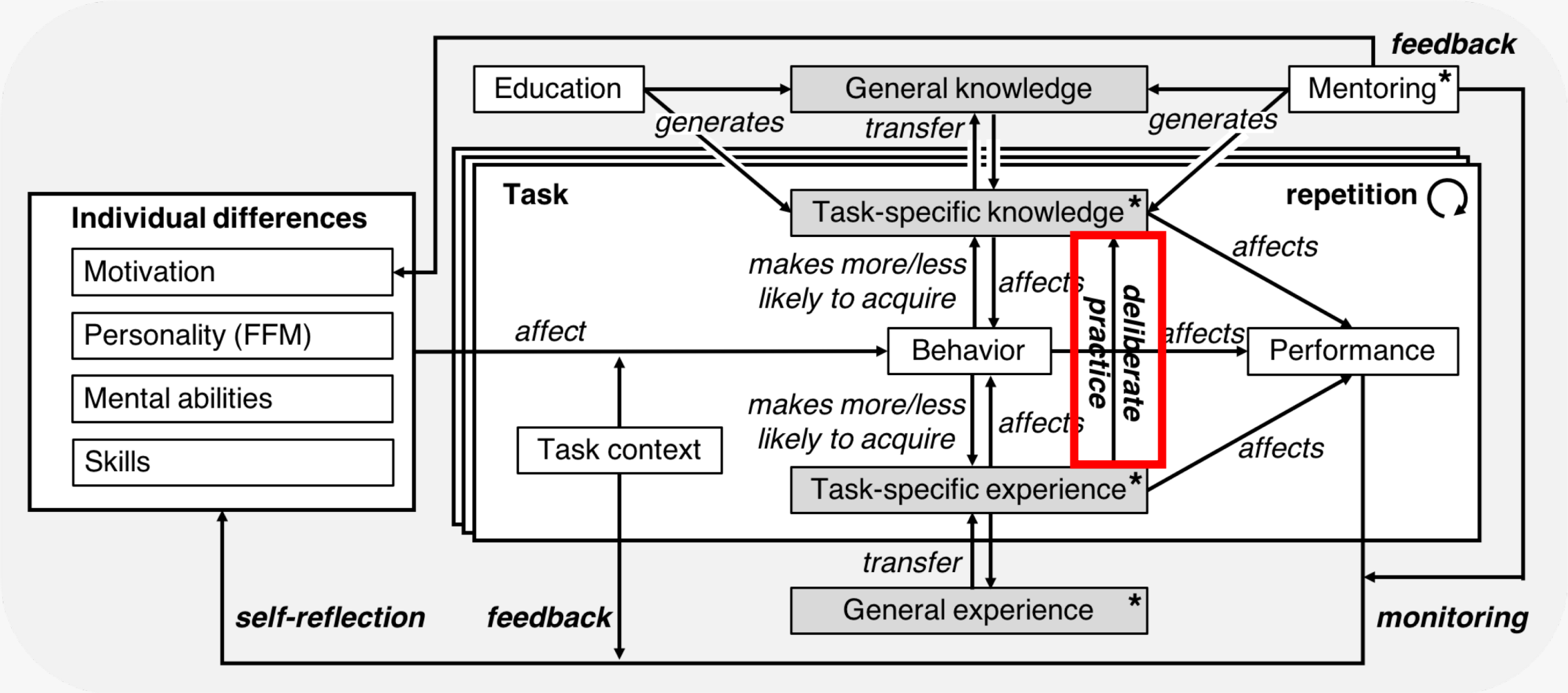


http://kratinmobile.com/blog/wp-content/uploads/2014/09/context3.png

- 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)

Final Conceptual Theory



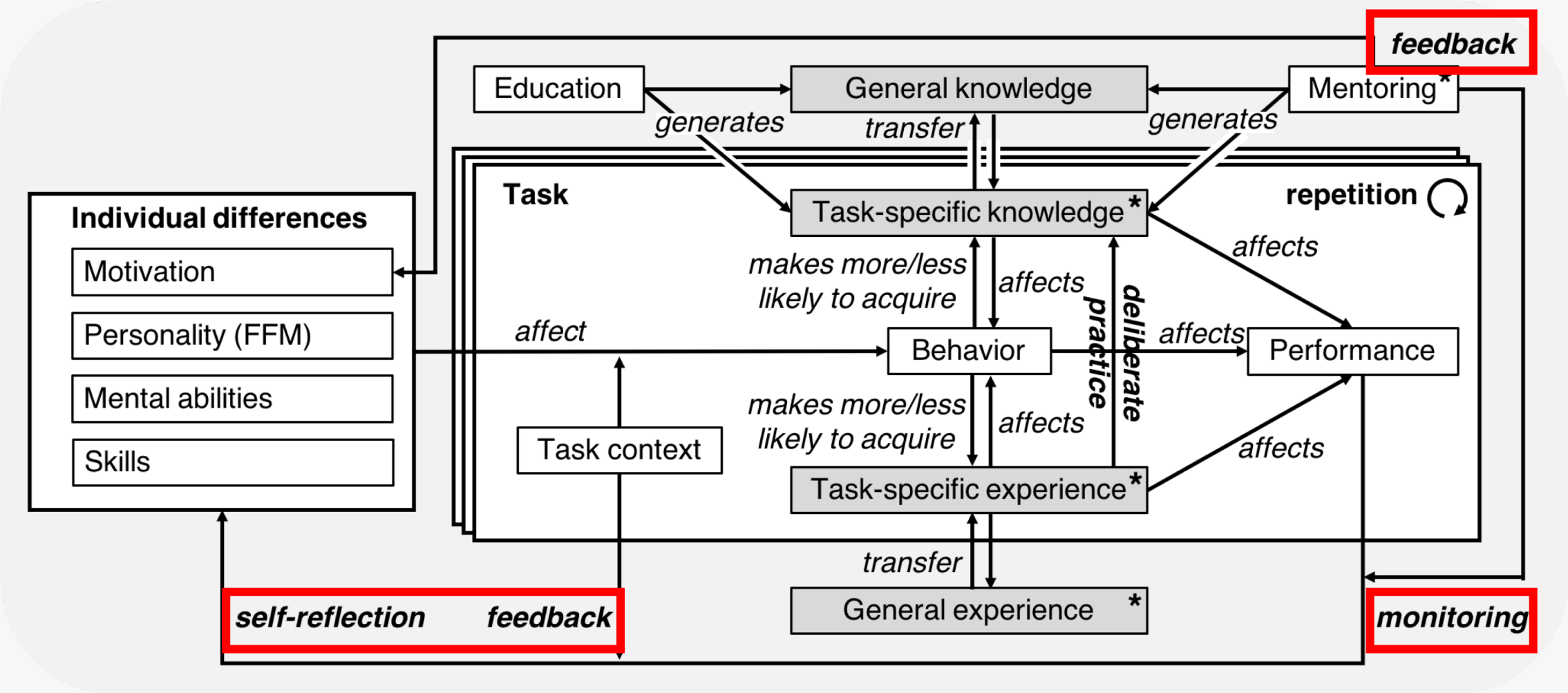
Deliberate Practice

- 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)

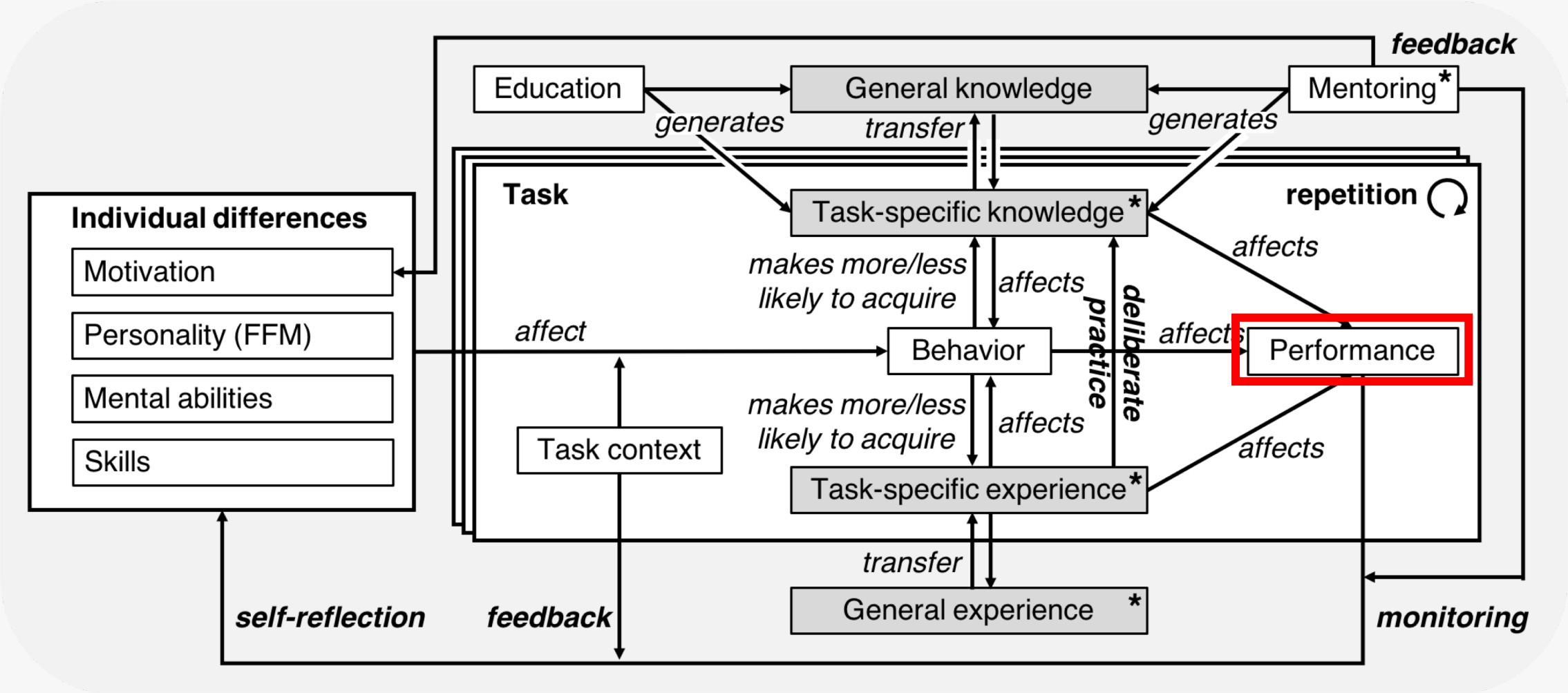
Deliberate Practice: Self-Reflection

- **(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

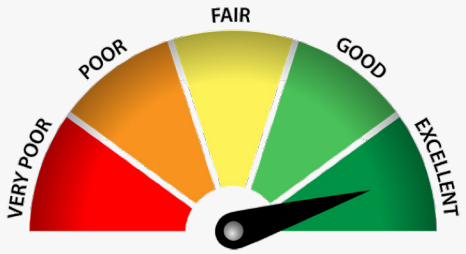
Final Conceptual Theory



Final Conceptual Theory



Performance



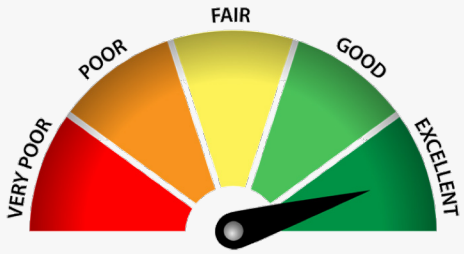
Scope of this work:

- We do **not** treat performance as a **dependent variable** that we try to explain or predict 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

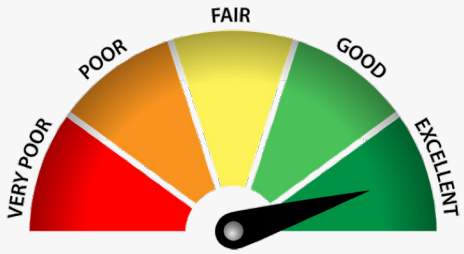
Performance



- 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]****."*

Expert Performance

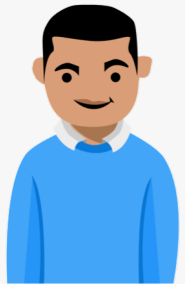


- In some areas (e.g., chess), there exist **representative tasks** and **objective criteria** for identifying experts
- Software development includes **many different tasks**
- Much more **difficult** to find objective measures for quantifying software development expert performance

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

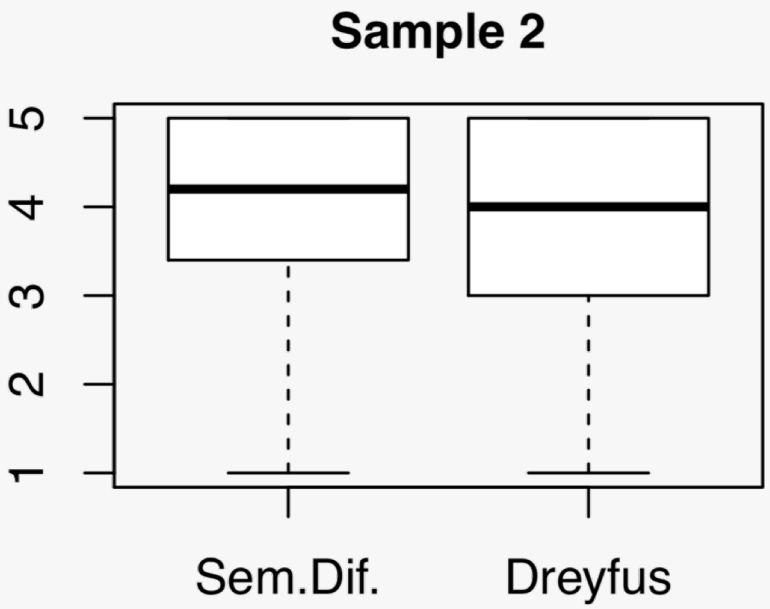
"Programming ability is based on **desire to achieve**. In the early years, it is a sort of **competition**. [...] I found that I lost a significant amount of my focus as I became 40, and started **using drugs such as ritalin** to enhance my abilities. This is pretty common among older programmers."

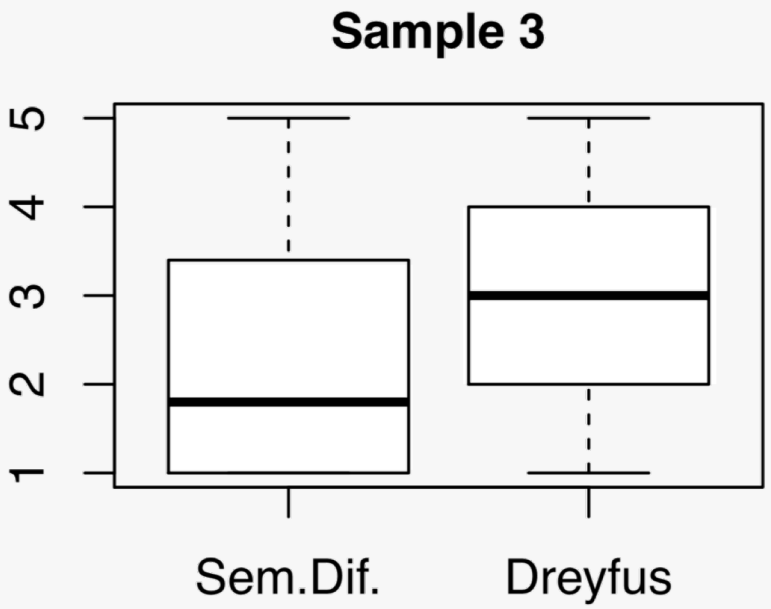
software developer, age 60

How are experience and expertise related?

Experience vs. Expertise

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

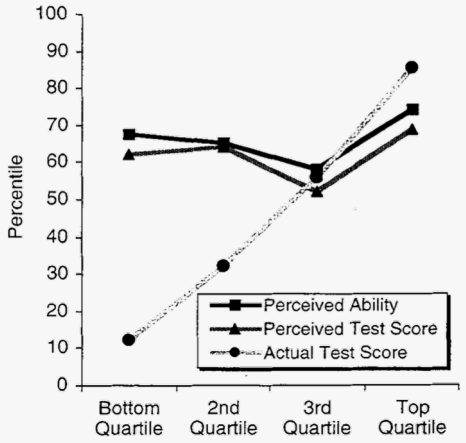




Experience vs. Expertise

- Analyzed correlation of experience (years) and self-assessed expertise and found **no consistent results**

- Possible explanation: **Dunning-Kruger effect**
  - Participants with a high skill-level underestimate their ability and performance relative to their peers
  - Context helped experienced developers to adjust their ratings to be more accurate



Experience vs. Expertise

- A
  a
- I

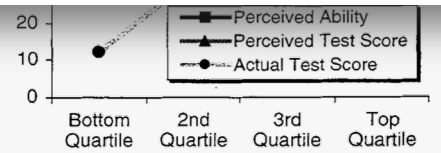
Journal of Personality and Social Psychology
1999, Vol. 77, No. 6, 1121–1134

0022-3514/99/$3.00

Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments

Justin Kruger and David Dunning
Cornell University

People tend to hold overly favorable views of their abilities in many social and intellectual domains. The authors suggest that this overestimation occurs, in part, because people who are unskilled in these domains suffer a dual burden: Not only do these people reach erroneous conclusions and make unfortunate choices, but their incompetence robs them of the metacognitive ability to realize it. Across 4 studies, the authors found that participants scoring in the bottom quartile on tests of humor, grammar, and logic grossly overestimated their test performance and ability. Although their test scores put them in the 12th percentile, they estimated themselves to be in the 62nd. Several analyses linked this miscalibration to deficits in metacognitive skill, or the capacity to distinguish accuracy from error. Paradoxically, improving the skills of participants, and thus increasing their metacognitive competence, helped them recognize the limitations of their abilities.

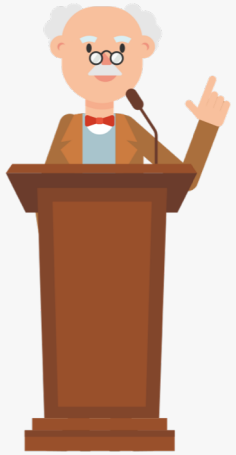


Takeaways

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



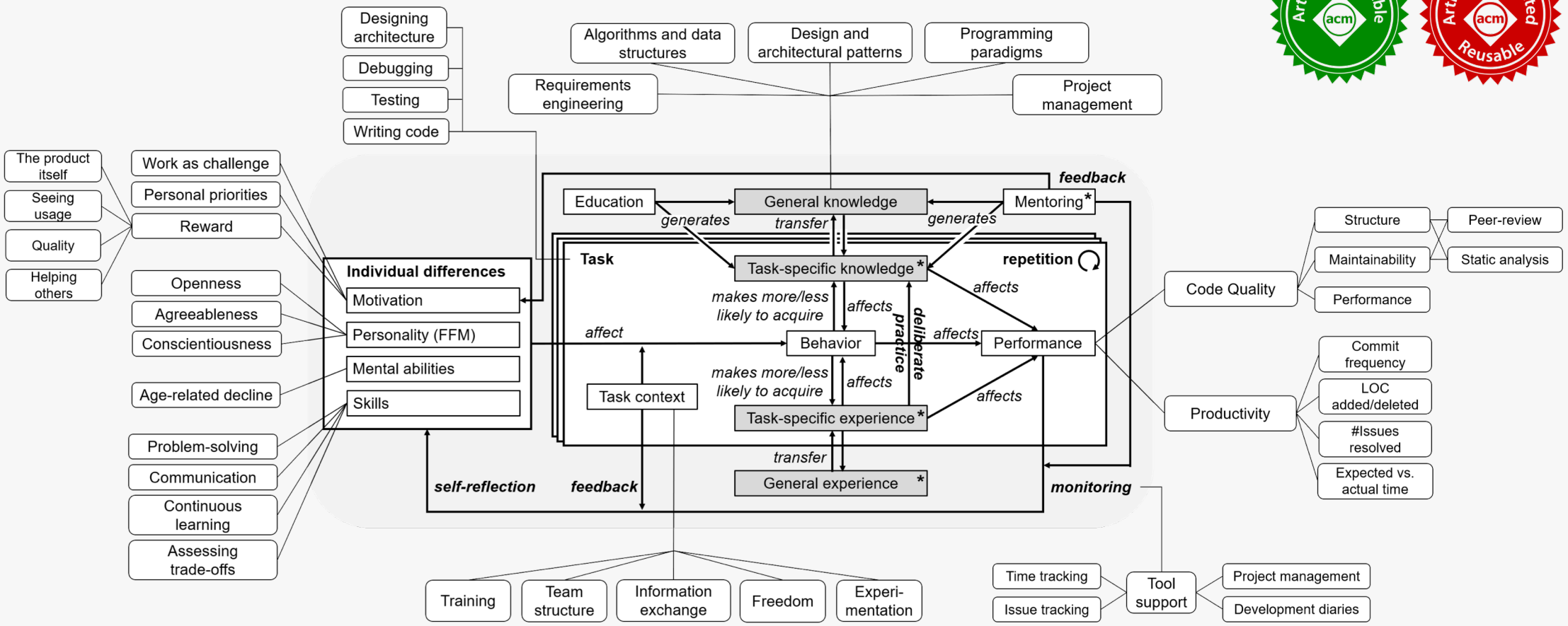
Summary for Developers

- 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
  - Facilitate meetings

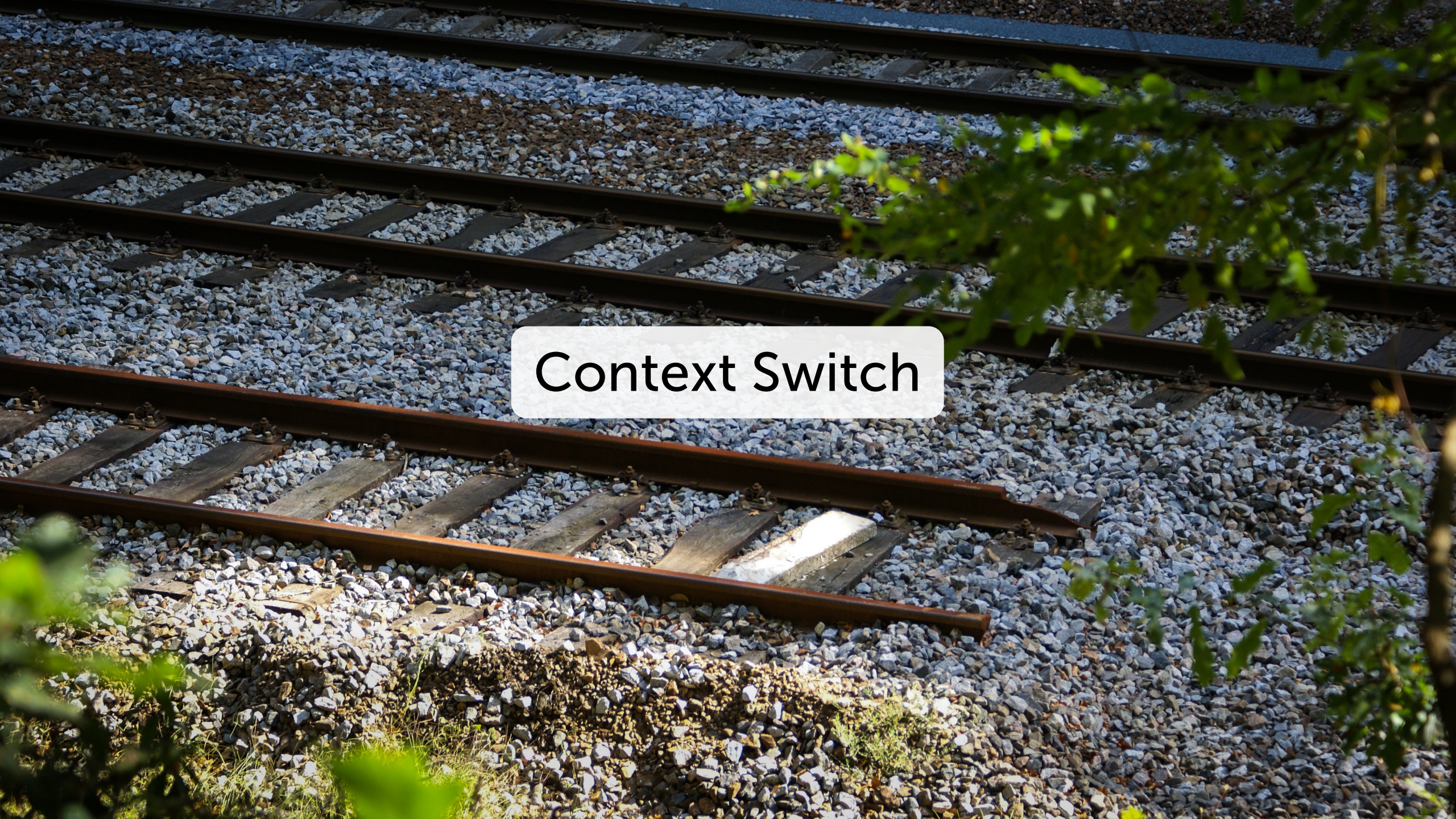


Sebastian Baltes

@s_baltes

expertise.sbaltes.com

Data and scripts available on Zenodo


Context Switch

SOTorrent

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

Sebastian Baltes
@s_baltes

sotorrent.org

Dataset available on Zenodo and BigQuery

Corresponding Research Papers

SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts

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

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 questio
site for software developers, providing a larg
snippets and free-form text on a wide variety o
software artifacts, questions and answers on S
for example when bugs in code snippets are fix
to work with a more recent library version, or
code snippet is edited for clarity. To be able to a
on SO evolves, we built *SOTorrent*, an open da
official SO data dump. *SOTorrent* provides acces
tory of SO content at the level of whole posts an
code blocks. It connects SO posts to other platfo
URLs from text blocks and by collecting refer

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

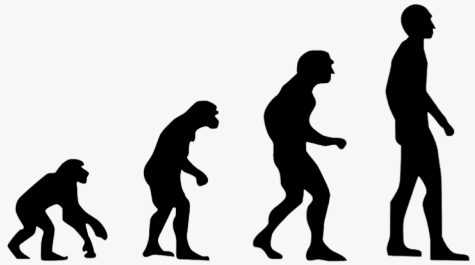
***Abstract*—Stack Overflow (SO) is the most popular question-and-answer website for software developers, providing a large amount of copyable code snippets. Like other software artifacts, code on SO evolves over time, for example when bugs are fixed or APIs are updated to the most recent version. To 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 versio
history of SO posts at the level of individual text and co
blocks (see Figure 1 for exemplary posts). The official S
data dump [1] keeps track of different versions of e
posts, but does not contain information about diffe
between versions at a more fine-grained level. In partic
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 eva uation of 134 different
string similarity metrics [15]. Beside providing access to the
version history, our dataset links SO p
in two ways: (1) by extracting linked U
of SO posts and from post comment

MSR 2018/2019

Why Reconstruct and Analyze SO Post Evolution?

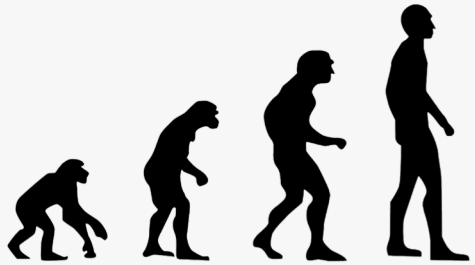
- The content of **14.3 million posts** has been **edited** after creation
  (SO data dump 2018-03-01)
- Like other **software artifacts**, SO posts **evolve over time**:
  - Bugs in code snippets are fixed
  - Clarifications are added in text documenting the code
  - Snippets are updated to new language/library versions
- **Copying code** from Stack Overflow (SO) is common, despite licensing, security, and maintainability implications



Why Reconstruct and Analyze SO Post Evolution?

Evolution of code on SO differs from regular software projects:

- **Short** code snippets (12 LOC on average)
- **No bug tracking** system (just comments and new answers)
- **No versioning** for individual snippets (just whole posts)



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() : "";
}
```

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

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 [illegible] ase we separate tokens using [illegible] boundary" (\A) thus giv[illegible] the entire contents of the str[illegible]

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[illegible]er

Post edits

edited Sep 2[illegible]

Reasons for edits

[illegible]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

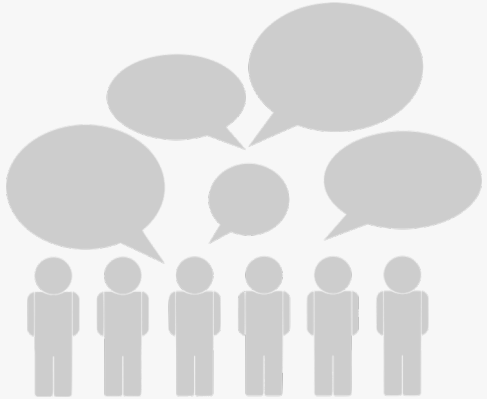
Alternative solution

Bug report

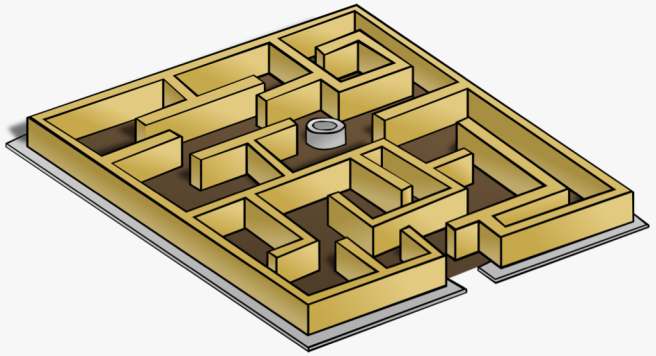
Bug report

Comment by author

This stuff is clearly a hack.



Even for such a simple code snippet, the **context** is quite **complex**:



- The snippet is based on an **external source**
- Hidden in the **comments**, the author acknowledges that his solution is *"clearly a hack"*
- There are several **bug reports** pointing to issues
- Has the snippet been **edited** to fix those issues?
- Is the snippet **safe** to use?


SOTorrent

Retrieve all versions of a code snippet:

```
SELECT PostHistoryId, Content, Length, LineCount, PredSimilarity
FROM PostBlockVersion
WHERE PostId=5445161 AND LocalId=2 AND PredEqual=0
ORDER BY PostHistoryId DESC;
```

Most recent version

PostHistoryId	Content	Length	LineCount	PredSimilarity
155295527	static String convertStreamToString(java.io.In…	192	4	0.7532467532467533
154620092	static String convertStreamToString(java.io.In…	352	13	0.7532467532467533
44935719	static String convertStreamToString(java.io.In…	192	4	0.9846153846153847
31249705	public static String convertStreamToString(jav…	199	4	0.9523809523809523
30827994	String convertStreamToString(java.io.InputStr…	185	4	0.6875
25270546	String convertStreamToString(java.io.InputStr…	239	7	0.9714285714285714
21289331	public String convertStreamToString(java.io.I…	246	7	0.8157894736842105
21230790	import java.util.Scanner; import java.util.No…	298	10	0.8405797101449275

Retrieve line-based difference for latest version:

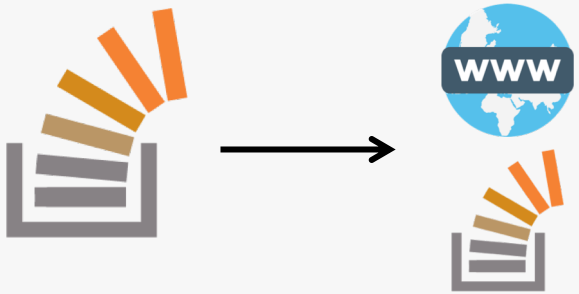
```
SELECT PostHistoryId, LocalId, PredLocalId, PostBlockDiffOperationId, Text
FROM PostBlockDiff
WHERE PostHistoryId=155295527 AND LocalId=2;
```

Changed lines

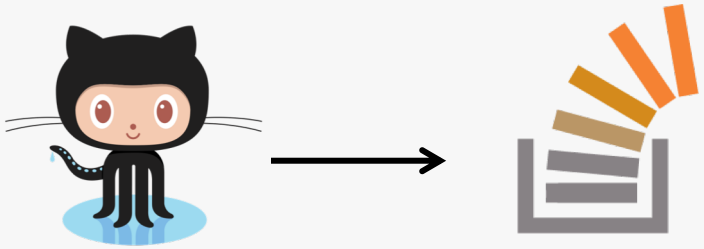
PostHistoryId	LocalId	PredLocalId	PostBlockDiffOperationId		Text
155295527	2	2	0	Equal	static String convertStreamToString(java.io.InputStream is) {
155295527	2	2	-1	**Delete**	java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");…
155295527	2	2	1	**Insert**	if (is == null) { return ""; } java.util.Scanner s…
155295527	2	2	0	Equal	}

Extracting Links From Stack Overflow Posts

- Extracted **31.4m links** from 11.6m posts, pointing to 567k different domains using a regular expression (SOTorrent 2018-05-04)



- Extracted **6.0m links** from 438k GitHub repos, pointing to 147k posts using Google BigQuery (SOTorrent 2018-05-04)



Retrieve links from a post version:

```
SELECT PostId, PostHistoryId, Domain, Url
FROM PostVersionUrl
WHERE PostHistoryId=155295527;
```

PostId	PostHistoryId	Domain	Url
5445161	155295527	community.oracle.com	https://community.oracle.com/blogs/pat/2004/10/23/stupid-scanner-tricks
5445161	155295527	download.oracle.com	http://download.oracle.com/javase/8/docs/api/java/util/Scanner.html
5445161	155295527	stackoverflow.com	https://stackoverflow.com/users/68127/jacob-gabrielson
5445161	155295527	stackoverflow.com	https://stackoverflow.com/users/101272/patrick

Retrieve links from GitHub repos to post:

```
SELECT PostId, RepoName, Branch, Path, FileExt, Size, Copies
FROM PostReferenceGH
WHERE PostId=5445161;
```

Referenced in 103 distinct repos

PostId	RepoName	Branch	Path	FileExt	Size
5445161	resource4j/resource4j	master	core/src/main/java/com/github/resource4j/object…	.java	2077
5445161	yugecin/opsu-dance	master	src/itdelatrisu/opsu/Utils.java	.java	16107
5445161	Roojin/persian-calendar-view	master	persiancalendar/src/main/java/ir/mirrajabi/persia…	.java	16833
5445161	FITeagle/sfa	master	src/main/java/org/fiteagle/north/sfa/dm/SFA_XM…	.java	5426
5445161	Steguer/ProjetAndroid	master	ProjetAndroid/libs/android-maps-utils/demo/src/…	.java	1140
5445161	ScottSWu/opsu	master	src/itdelatrisu/opsu/Utils.java	.java	17943
5445161	massimiliano76/freedomotic	master	plugins/devices/restapi-v3/src/main/java/com/fre…	.java	3315

…

MSR Mining Challenge 2019

Abstracts due Feb 1, 2019

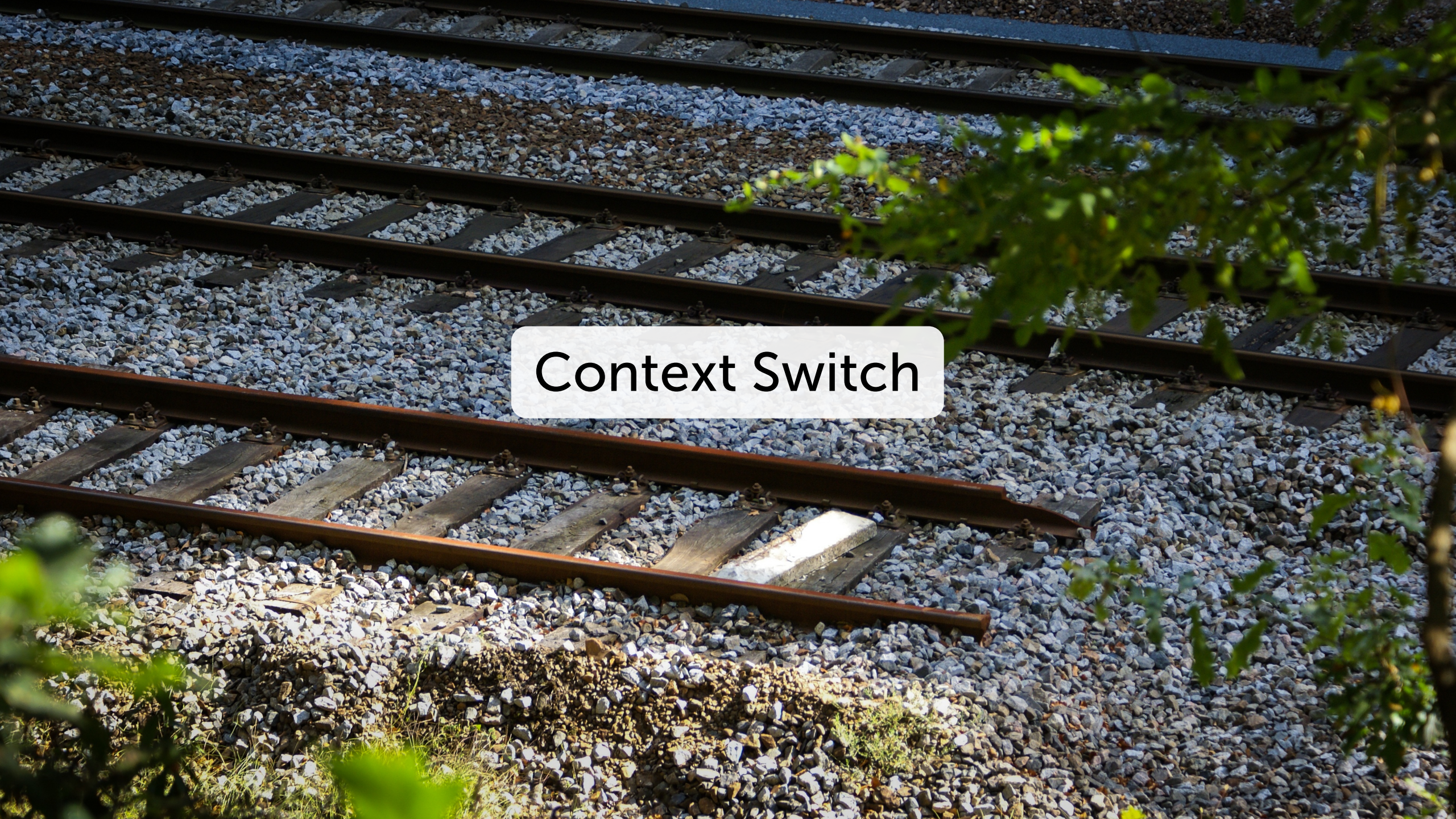
Papers due Feb 6, 2019

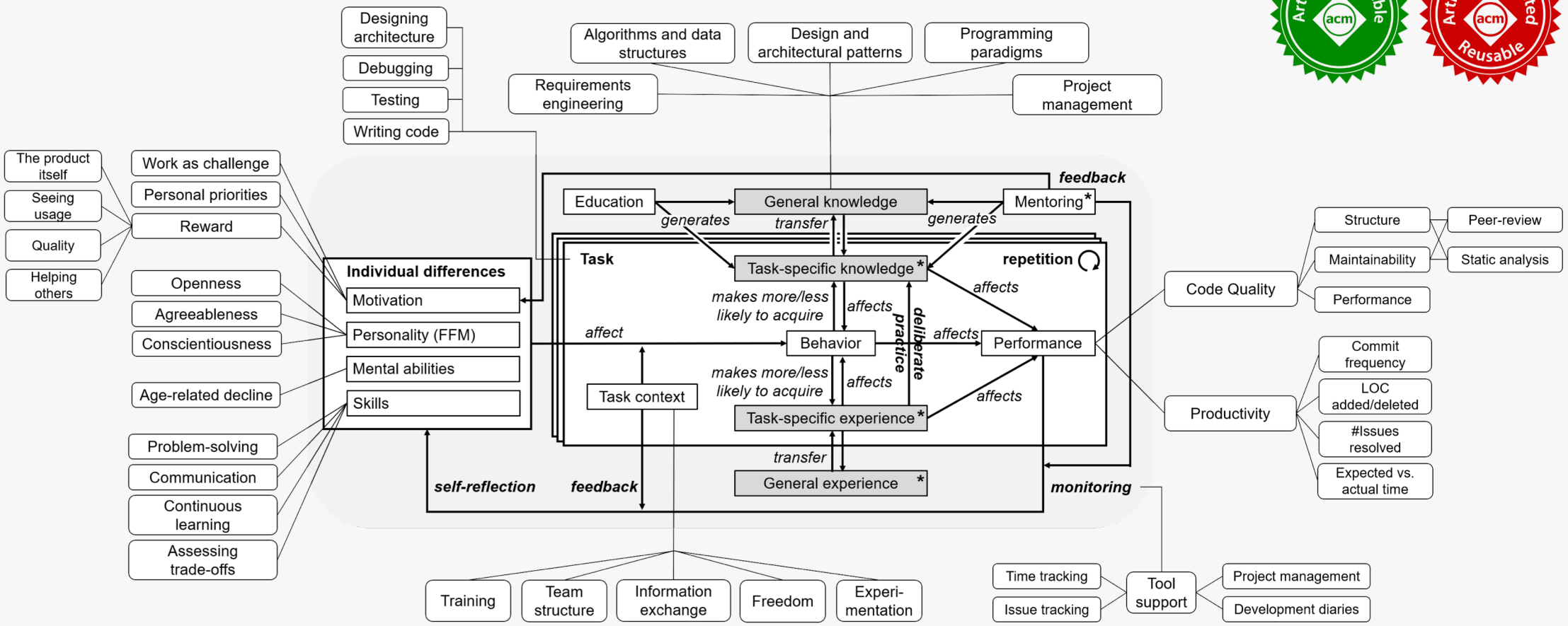
Sebastian Baltes

@s_baltes

sotorrent.org

Dataset available on Zenodo and BigQuery


Context Switch



Sebastian Baltes

@s_baltes

expertise.sbaltes.com

Data and scripts available on Zenodo