



Software Developers' Work Habits and Expertise

Sebastian Baltes

 @s_baltes

 empirical-software.engineering

Talk @ Microsoft Research
Redmond, WA, USA, March 21, 2019



Sweden

Växjö

Germany

Luxembourg
Trier
Saarbrücken

Australia

Adelaide

Trier



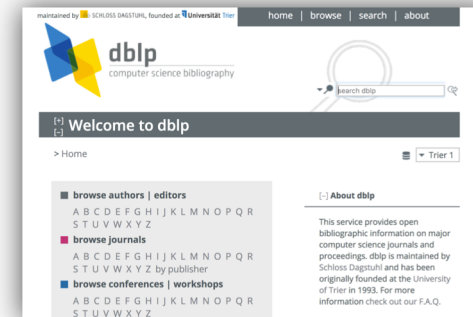
Roman Heritage



Karl Marx



Riesling Wine



DBLP

Software Developers' Work Habits and Expertise

Empirical Studies on Sketching, Code Plagiarism,
and Expertise Development



Why Study Developers' Work Habits?

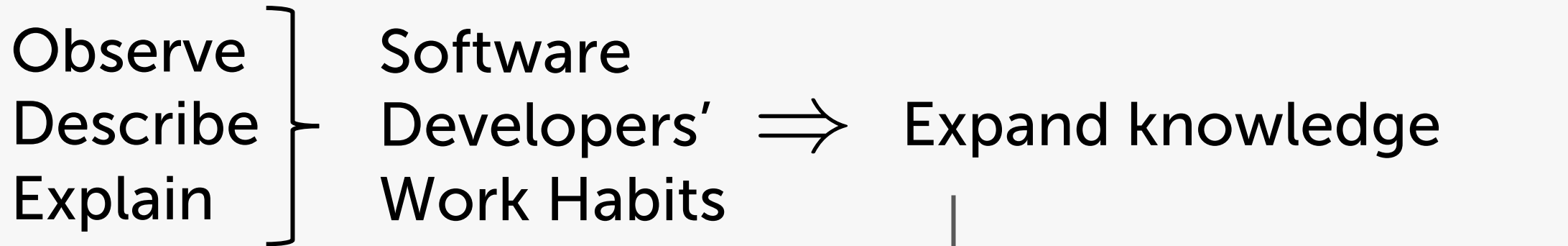
*“For me, thoroughly analyzing and **understanding the state-of-practice** is an essential first step towards **improving** how software is being developed. Too often, decisions are still rather opinion-based than **data-informed.**”*





Evidence-based Practice through **Practice-based Evidence**

Goal of my PhD Research



Habits?



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

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

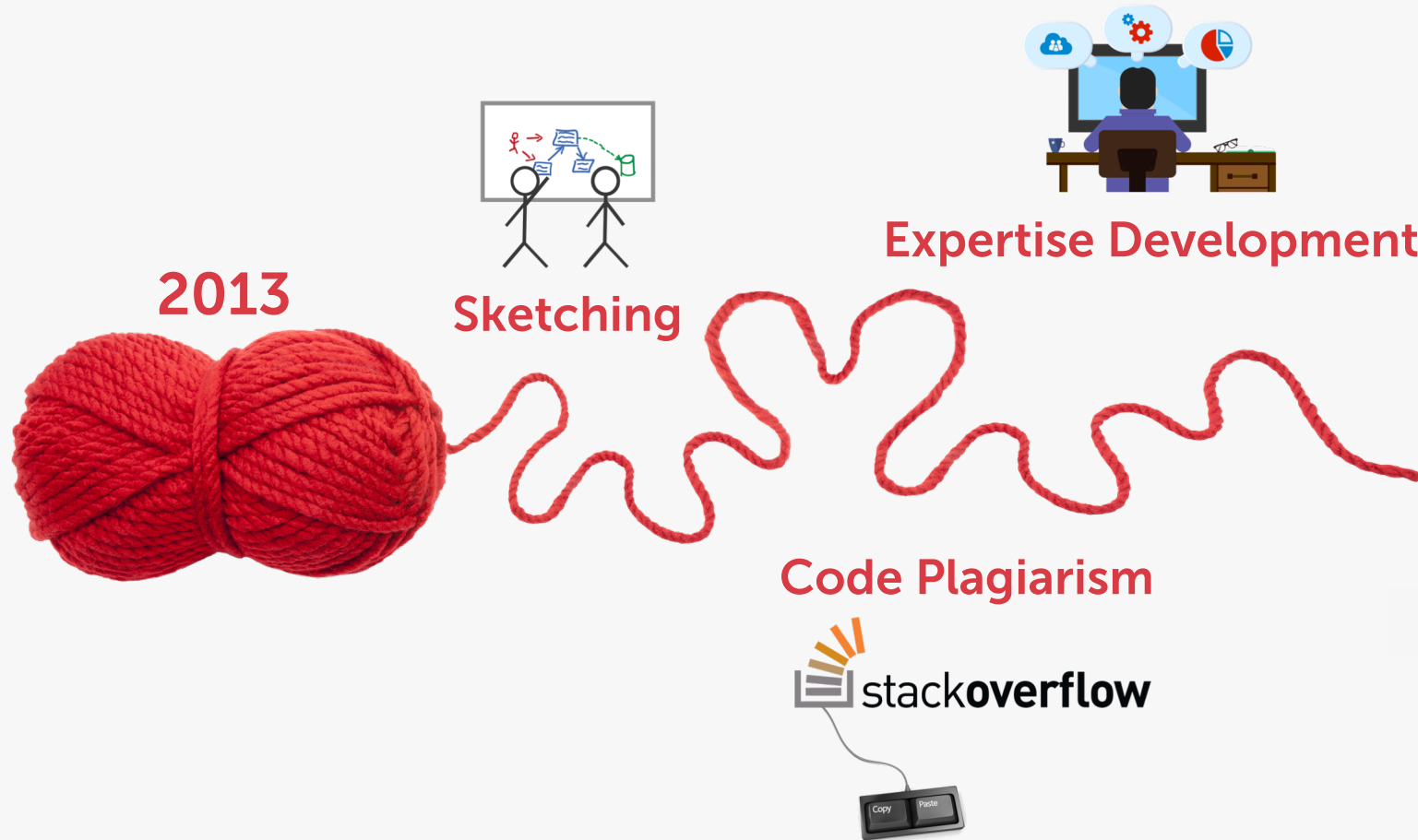
Personal habits



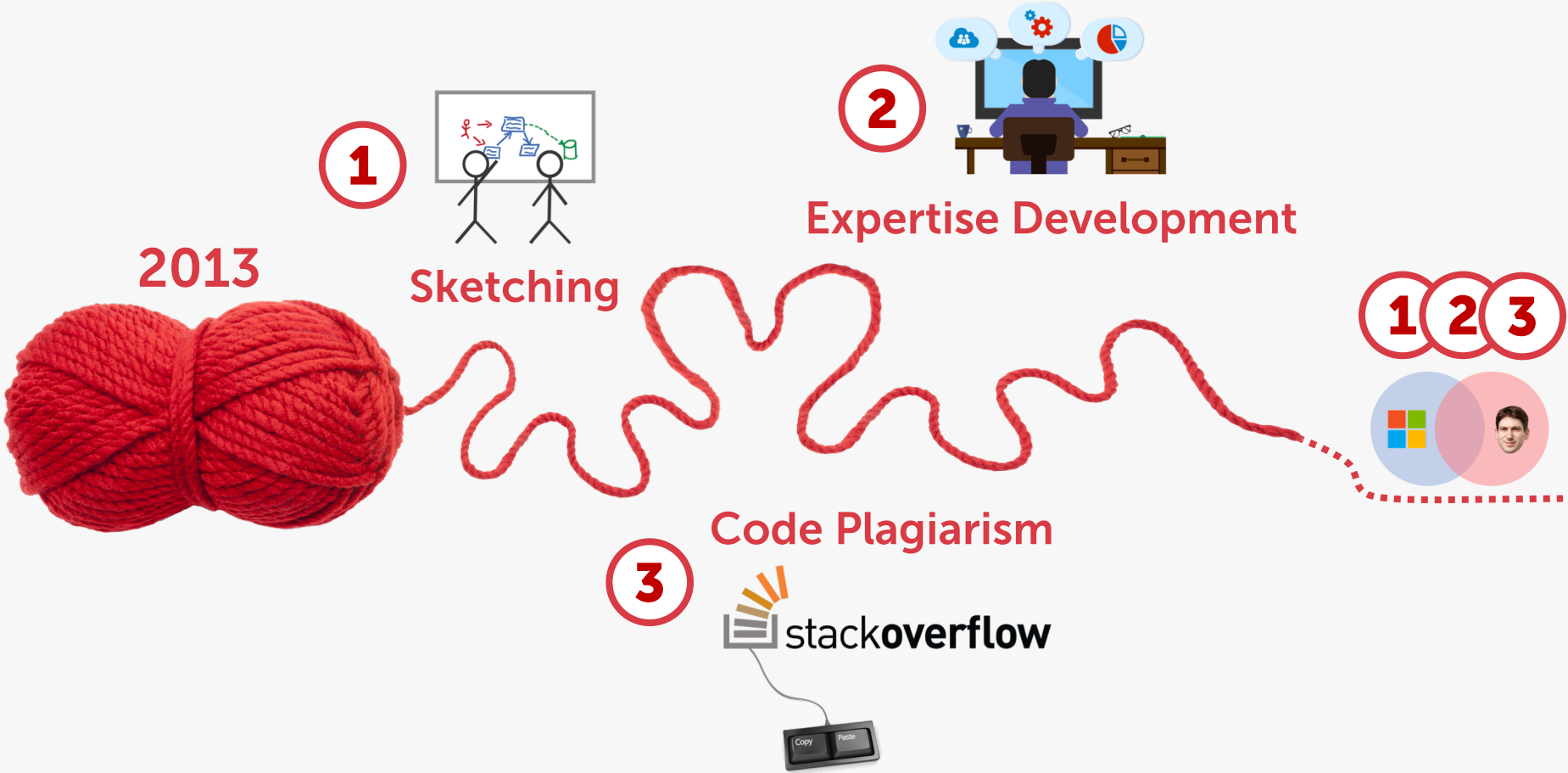
Work habits



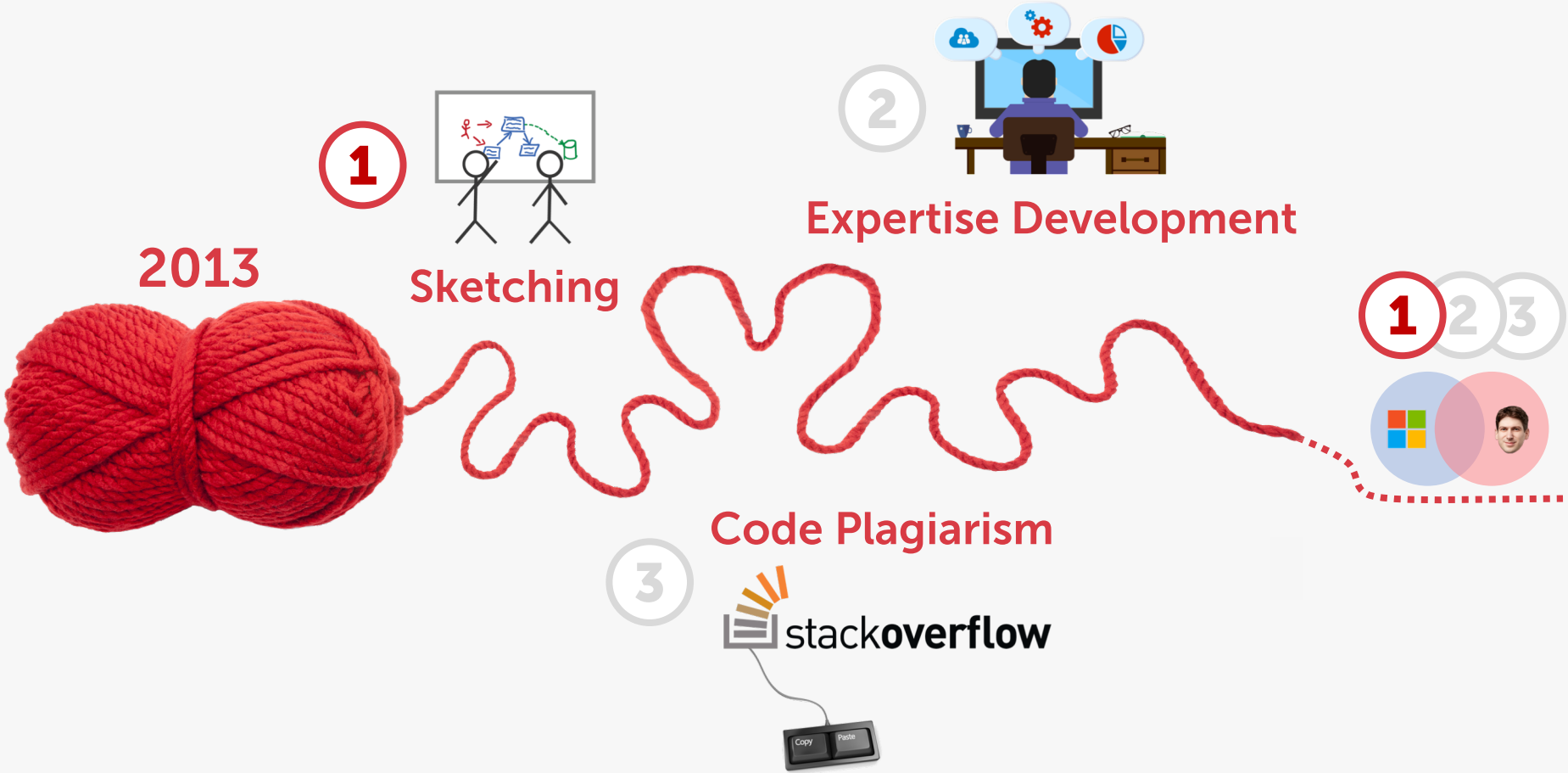
Studied Habits

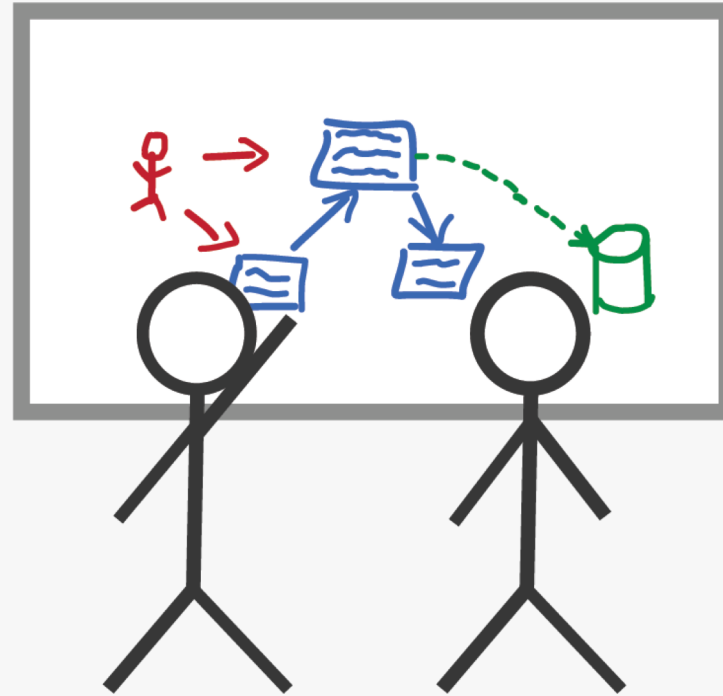


Overview of this Talk

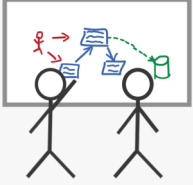


Overview of this Talk





Sketching



Research Questions



Questions:

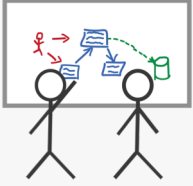
How and **why** do software practitioners use sketches and diagrams?

How are they related to **source code**?

How can we provide better **tool support**?

Approach:

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



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

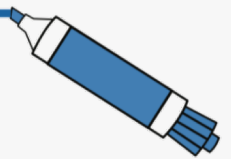
Over the past years, studies have shown the importance of sketches and diagrams in software development [6,11,43]. Most of these visual artifacts do not follow formal conventions like the *Unified Modeling Language* (UML), but have an informal, ad-hoc nature [6,11,23,25]. Sketches and diagrams are important 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/>

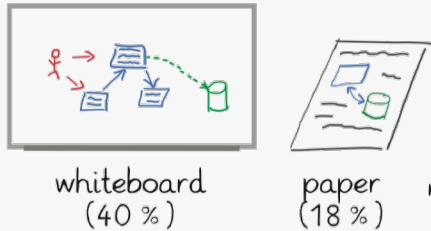
Sketching



Sketches and Diagrams in Practice



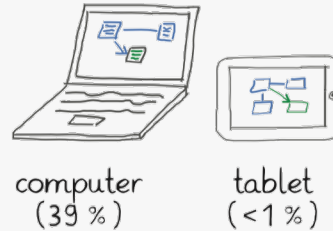
Revision



Media

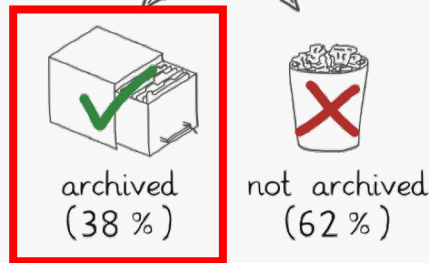
transitions between media are common

Revision

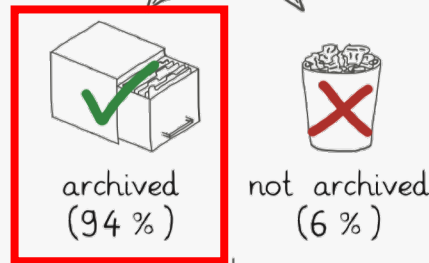


analog (58%)

digital (40%)

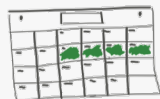


Archiving

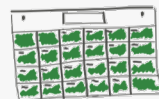


Lifespan




several work days



several months



Purpose

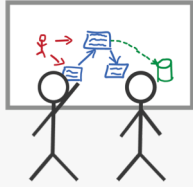
- Designing (75%) 
- Explaining (60%) 
- Understanding (56%) 
- Analyzing Requirements (45%)

Relation to Source Code

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



Sketching



SketchLink

```
SDRaytracer.java x
100 Ray eye_ray = new Ray();
101
102 /**
103  * Creates an two-dimensional image of pixels and computes the pixel colors by
104  * creating a ray for each pixel and tracing this ray through the scene.
105  * @sketchlink
106  */
107 void renderImage() {
108     System.out.println("Render Image: " + fov);
109     double tan_fovx = Math.tan(fov);
110     double tan_fovy = Math.tan(fov);
111     for (int i = 0; i < width; i++)
112         for (int j = 0; j < height; j++) {
113             image[i][j] = new RGB(0, 0, 0);
114             k = 0; k < rayPerPixel; k++) {
115                 double di = i + (Math.random() / 2 - 0.25);
116                 double dj = j + (Math.random() / 2 - 0.25);
117                 Ray ray = new Ray(0, 0, 0, 1, 0, 0, 1);
118                 ray.setStart(startX, startY, startZ); // ro
119                 ray.setDir((float) (((0.5 + di) * tan_fovx * 2.0) / width - tan_fovx),
120                         (float) (((0.5 + dj) * tan_fovy * 2.0) / height - tan_fovy), (float) 1f);
121                 eye_ray.normalize();
122                 image[i][j] = addColors(image[i][j], rayTrace(eye_ray, 0), 1.0f / rayPerPixel);
123             }
124         }
125     }
126 }
127 }
128 }
129 }
```

Sketchlink - Linked Sketches for Method "renderImage"

2013-12-02 18:12:11 "SDRaytracer Overview" (diehl)

Bildschirmraster Szene

raytrace() lighting() hitobjekt() raytrace()

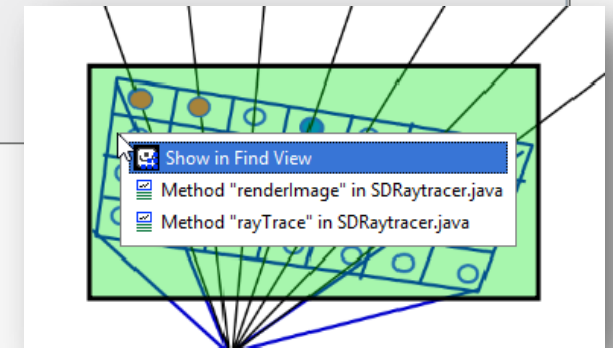
renderImage()

2013-11-05 14:21:16 "Pixel Raster" (diehl)

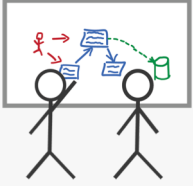
Found sketches:

- 2013-12-02 SDRaytracer Overview
- 2013-11-05 Pixel Raster

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



Sketching



Visual Thinking In Action: Visualizations As Used On Whiteboards

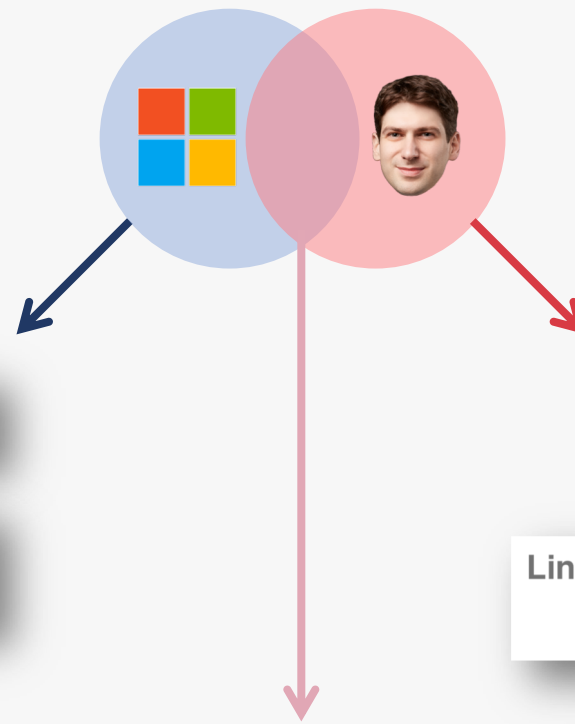
Jagoda Walny, Sheelagh Carpendale, Nathalie Henry Riche, Gina Venolia, and Philip Fawcett

**Let's Go to the Whiteboard:
How and Why Software Developers Use Drawings**

Mauro Cherubini

Gina Venolia and Rob DeLine

Andrew J. Ko



Sketches and Diagrams in Practice

Sebastian Baltes

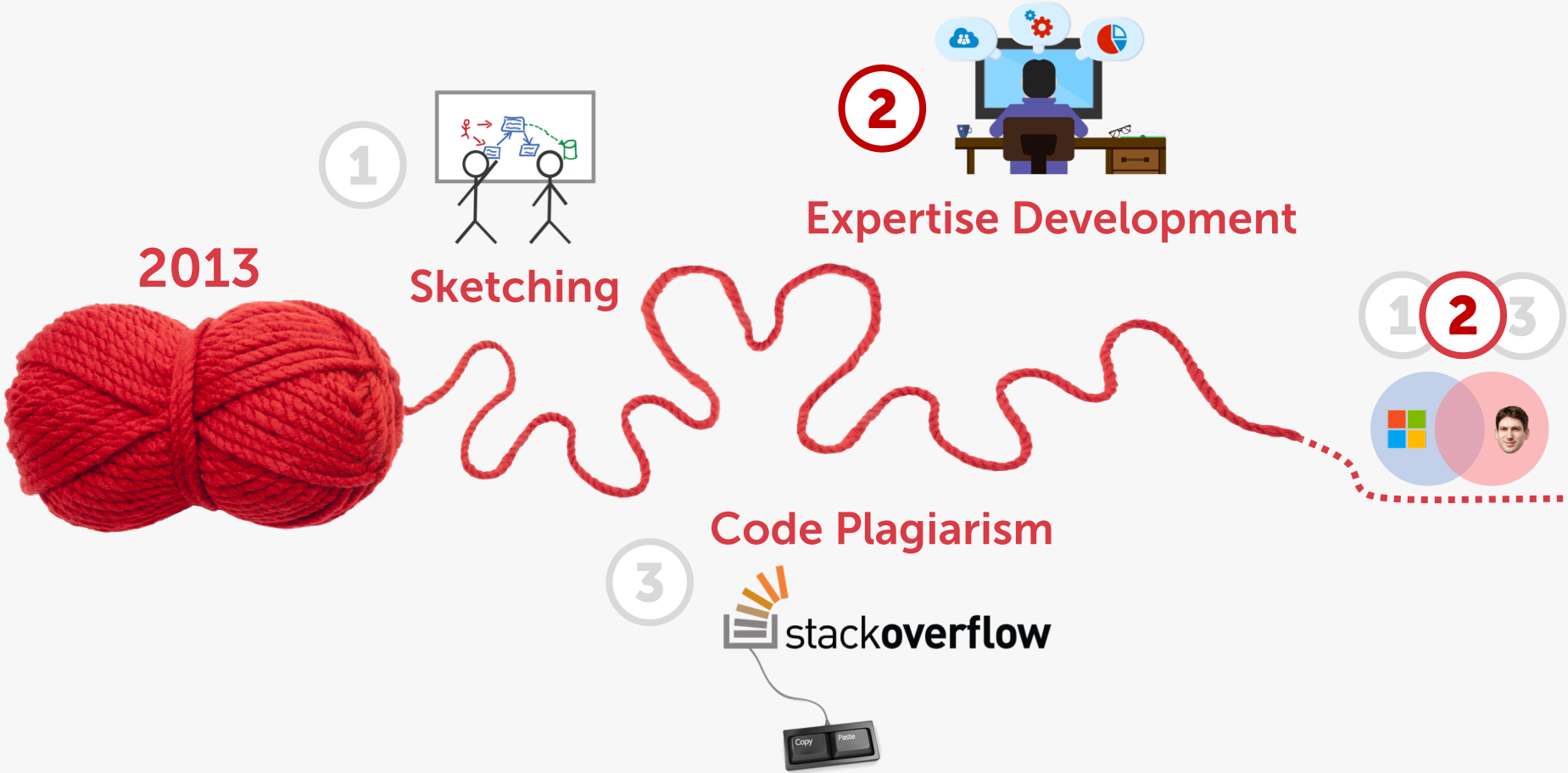
Stephan Diehl

Linking Sketches and Diagrams to Source Code Artifacts

Sebastian Baltes, Peter Schmitz, and Stephan Diehl

- *Follow-up study*: What has changed since the widespread adoption of **smartphones and tablets**?
- *SketchLink*: Get **feedback** from Microsoft developers, port to **Visual Studio**, evaluate
- *Graphic Facilitation*: **Explore use cases** within Microsoft, use preliminary results from interviews

Overview of this Talk





Expertise Development



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 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

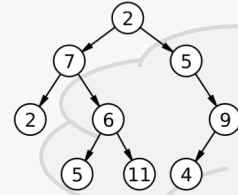
Debugging



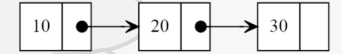
Software Development Expertise?



Implementing new features



Algorithms & Data structures



JUnit 5 Testing *jbehave*

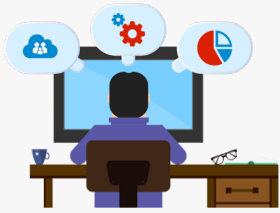


Debugging



Communication





Questions:

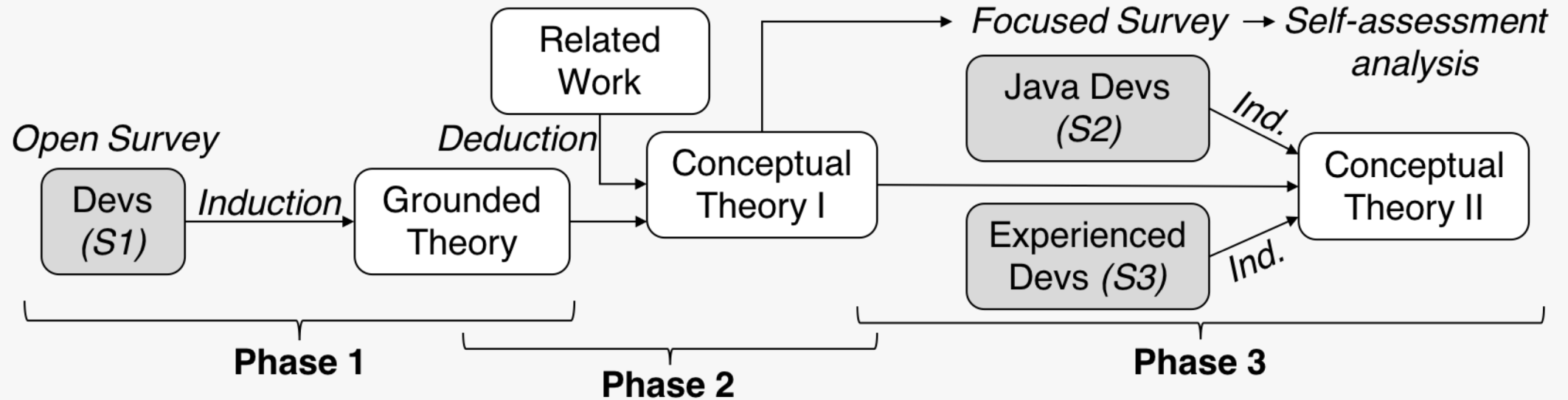
How to **structure** all those expertise-related aspects?

Which factors influence **expertise development** over time?

Approach:

Iterative theory building

Research Design



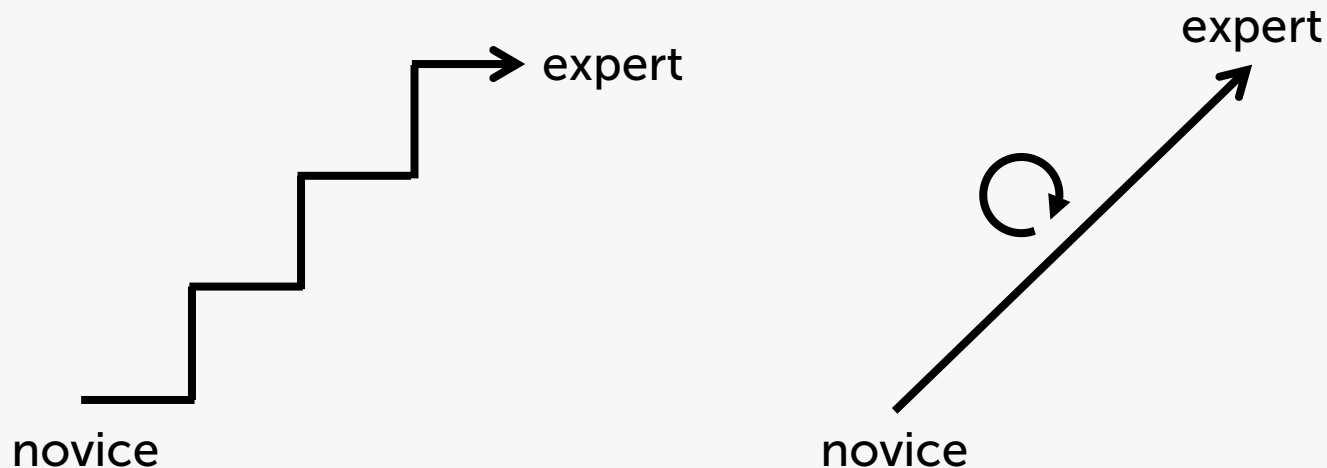
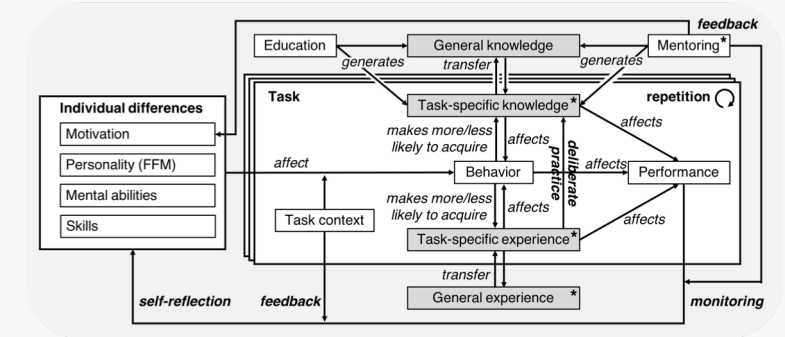
- **Induction:** 335 online survey participants in total
- **Deduction:** Main source "*Cambridge Handbook of Expertise and Expert Performance*"

THE CAMBRIDGE HANDBOOK OF
**Expertise and
Expert Performance**

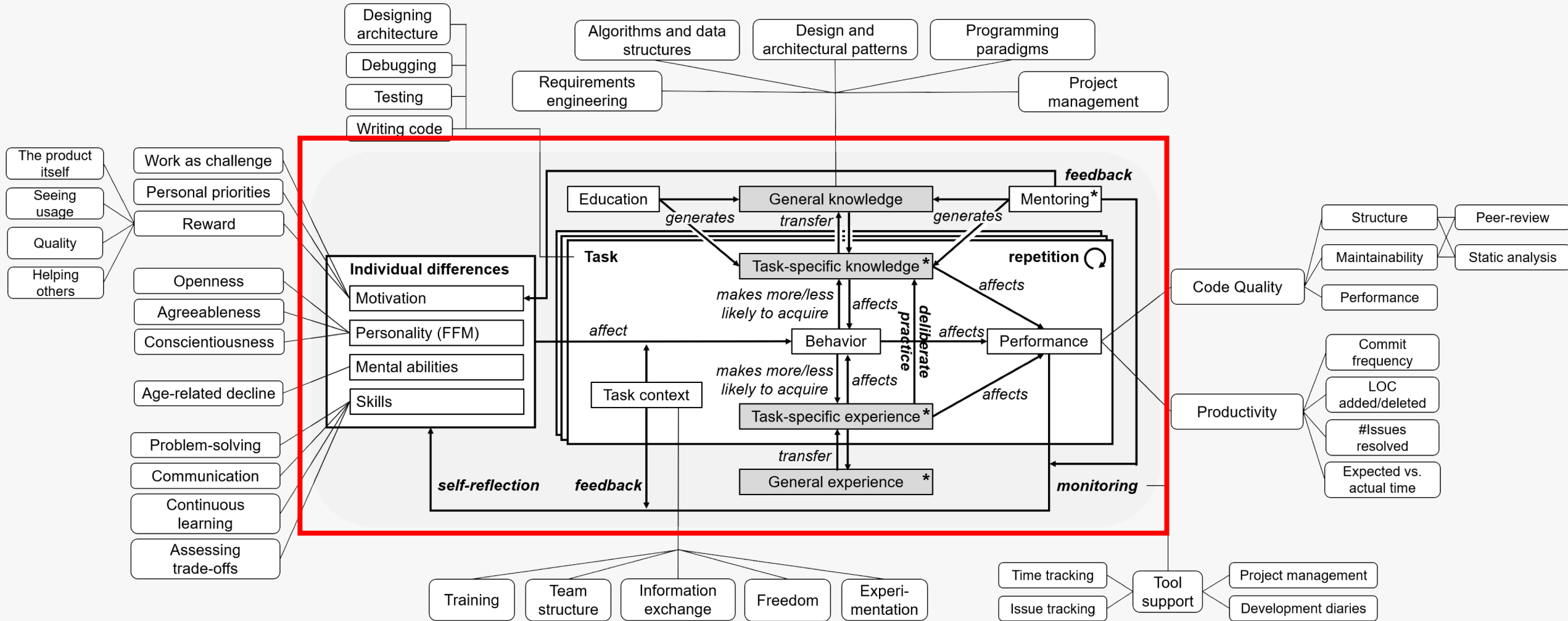
EDITED BY
K. Anders Ericsson
Neil Charness
Robert R. Hoffman
Paul J. Feltoich

Our Expertise Model

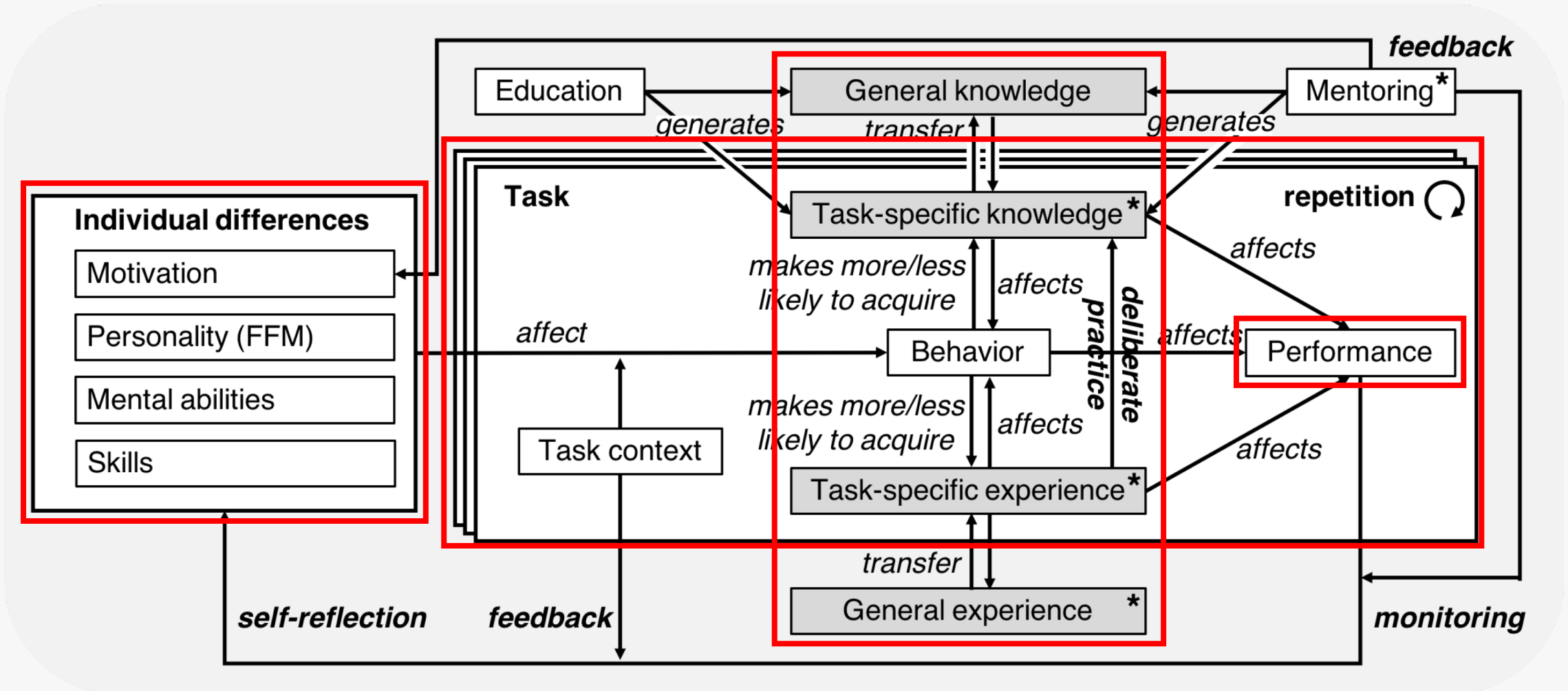
- **Task-specific** (e.g., writing code, debugging, testing)
- Focuses on **individual developers**
- **Process** view (repetition of tasks)
- Notion of **transferable knowledge and experience** from related fields or tasks
- **Continuum** instead of discrete expertise steps



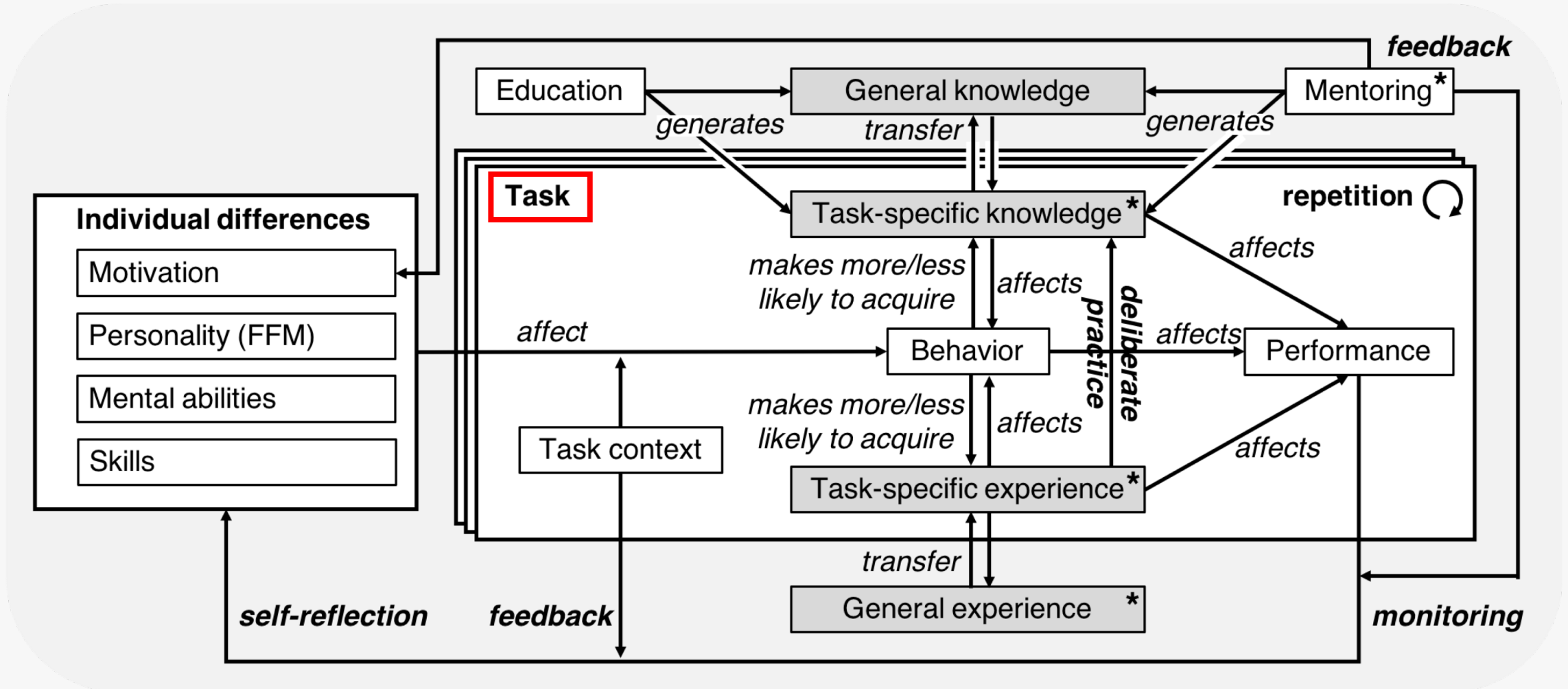
Conceptual Theory



Conceptual Theory



Conceptual Theory



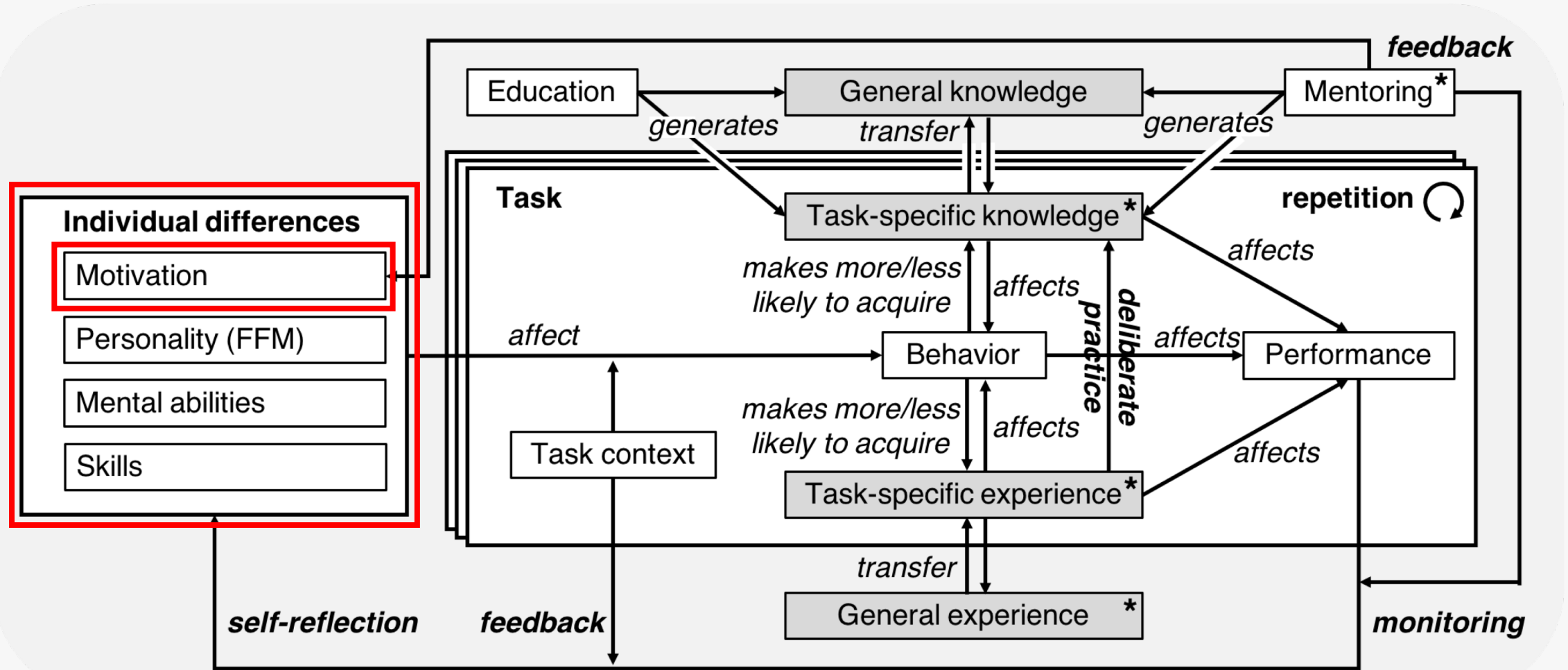
Tasks

- 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”*



Conceptual Theory



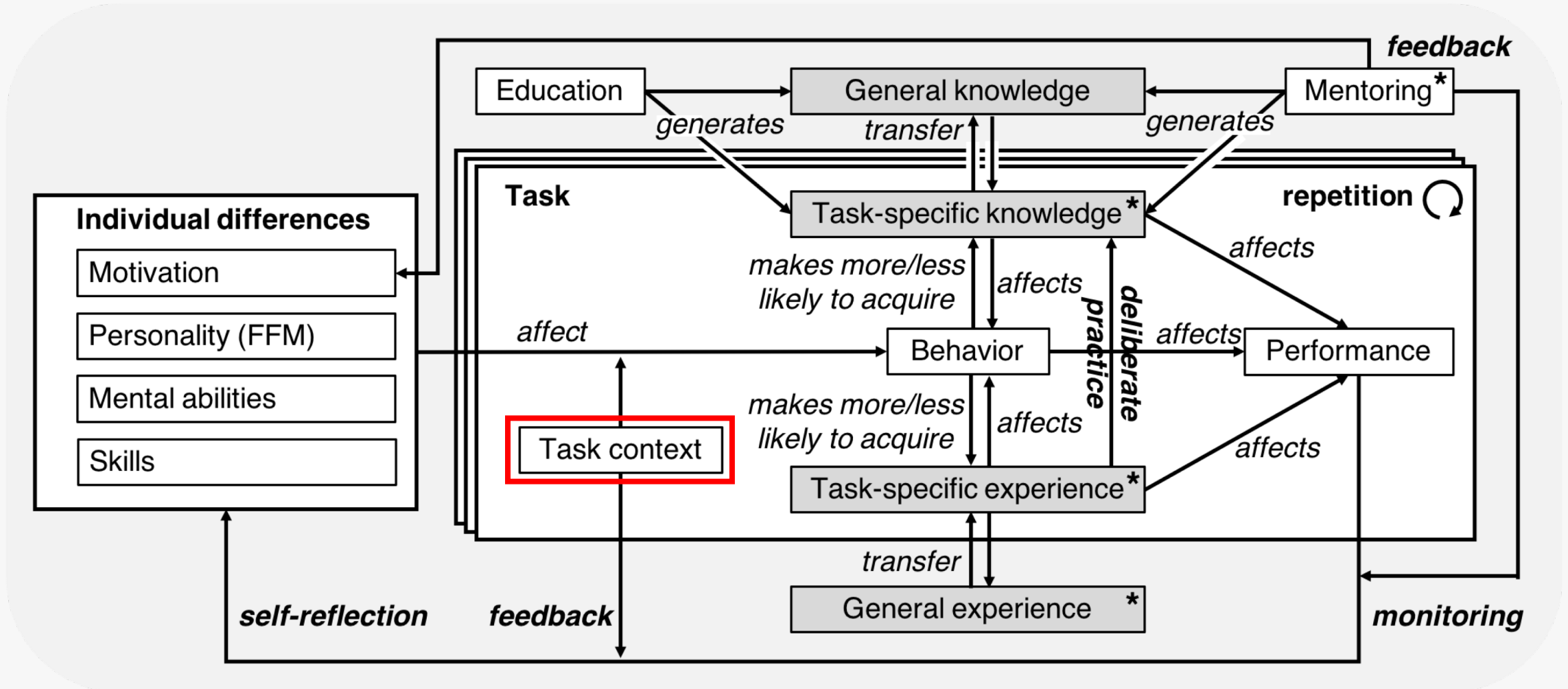
Individual Differences: Motivation

- 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**."*

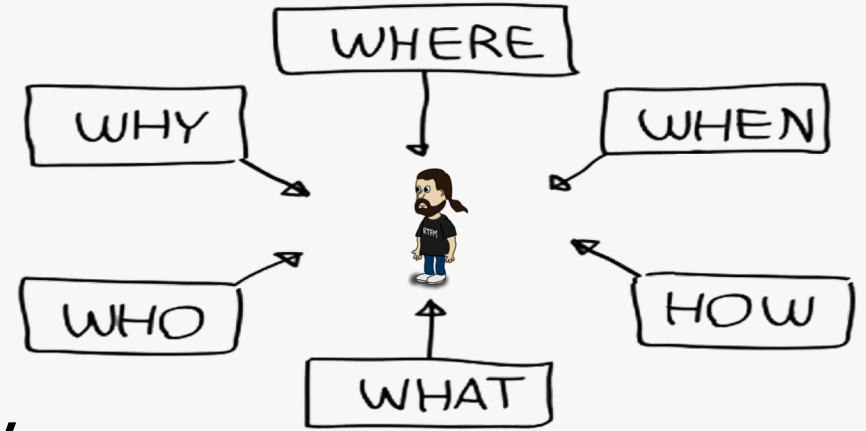


Conceptual Theory

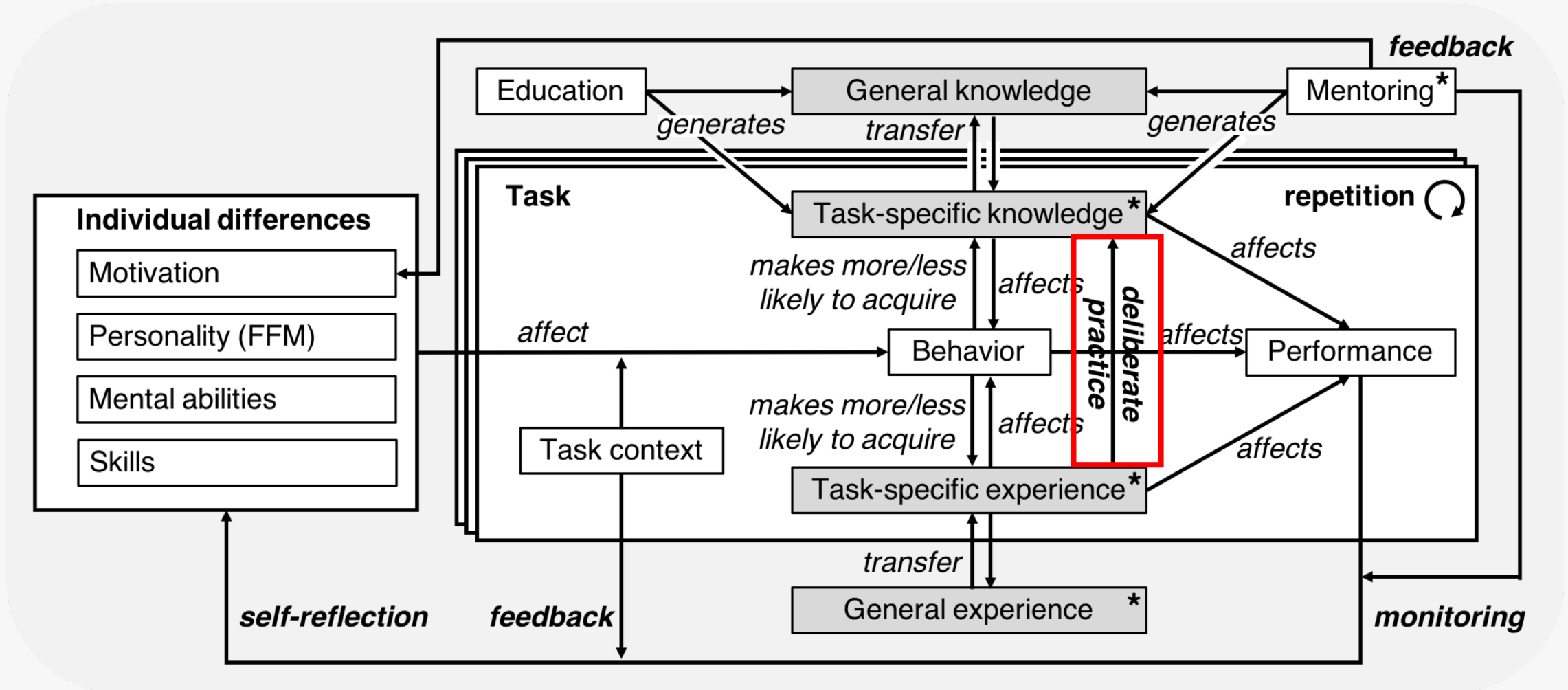


Task Context

- **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)



Conceptual Theory

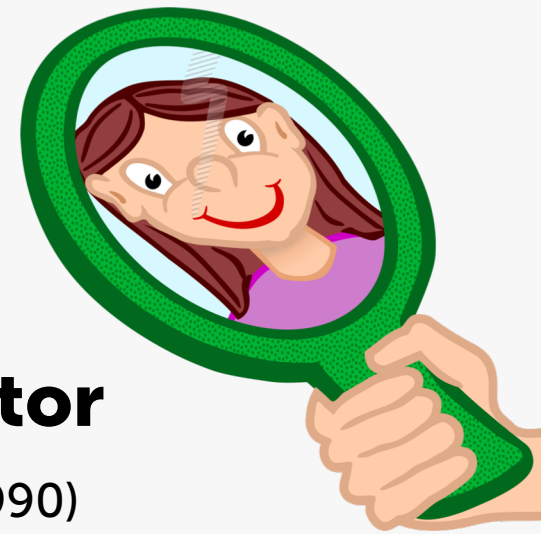


Deliberate Practice



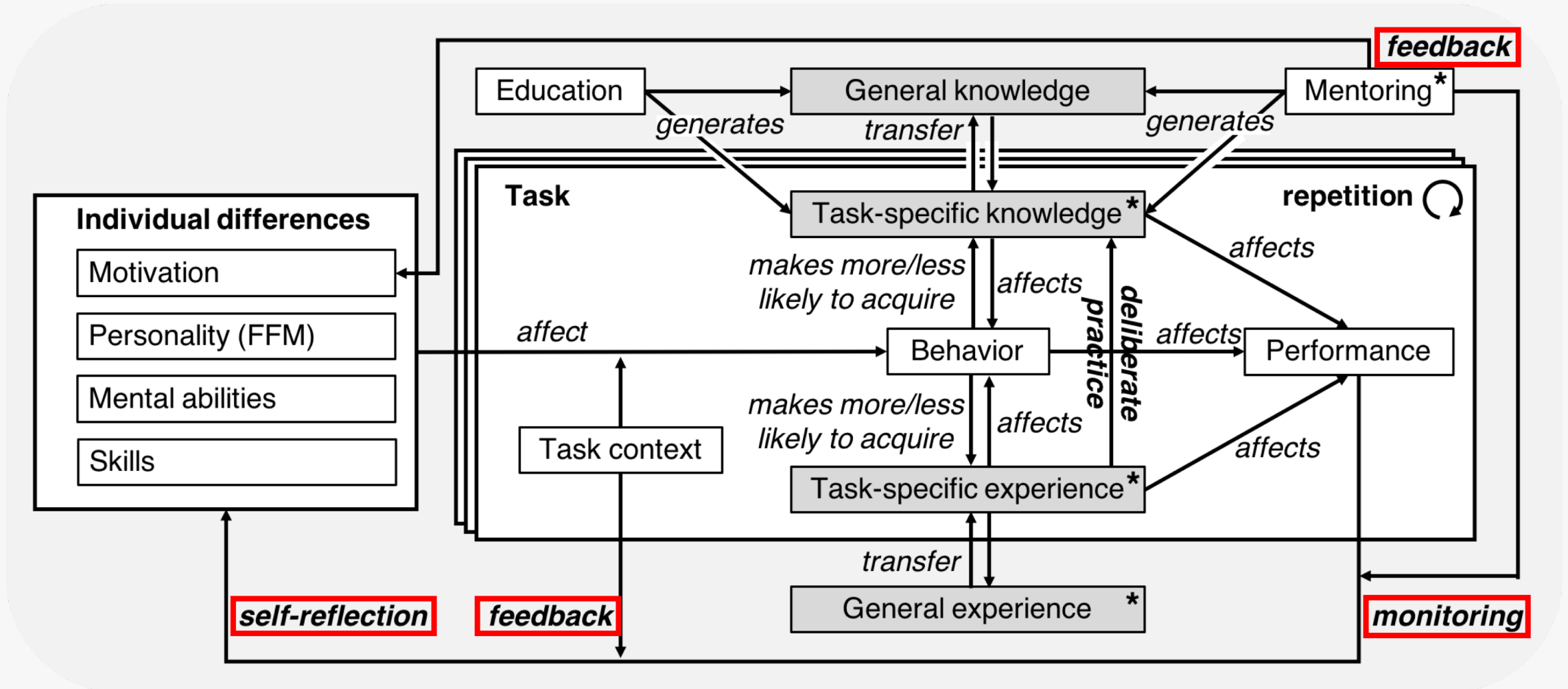
- Having **more experience** does not automatically lead to **better performance** (Ericsson et al., 1993)
- Performance may even **decrease** over time (Feltovich, 2006)
- Length of experience only weak correlate of job performance (Ericsson, 2006)
- Deliberate practice: „***Prolonged efforts to improve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

Deliberate Practice: Self-Reflection

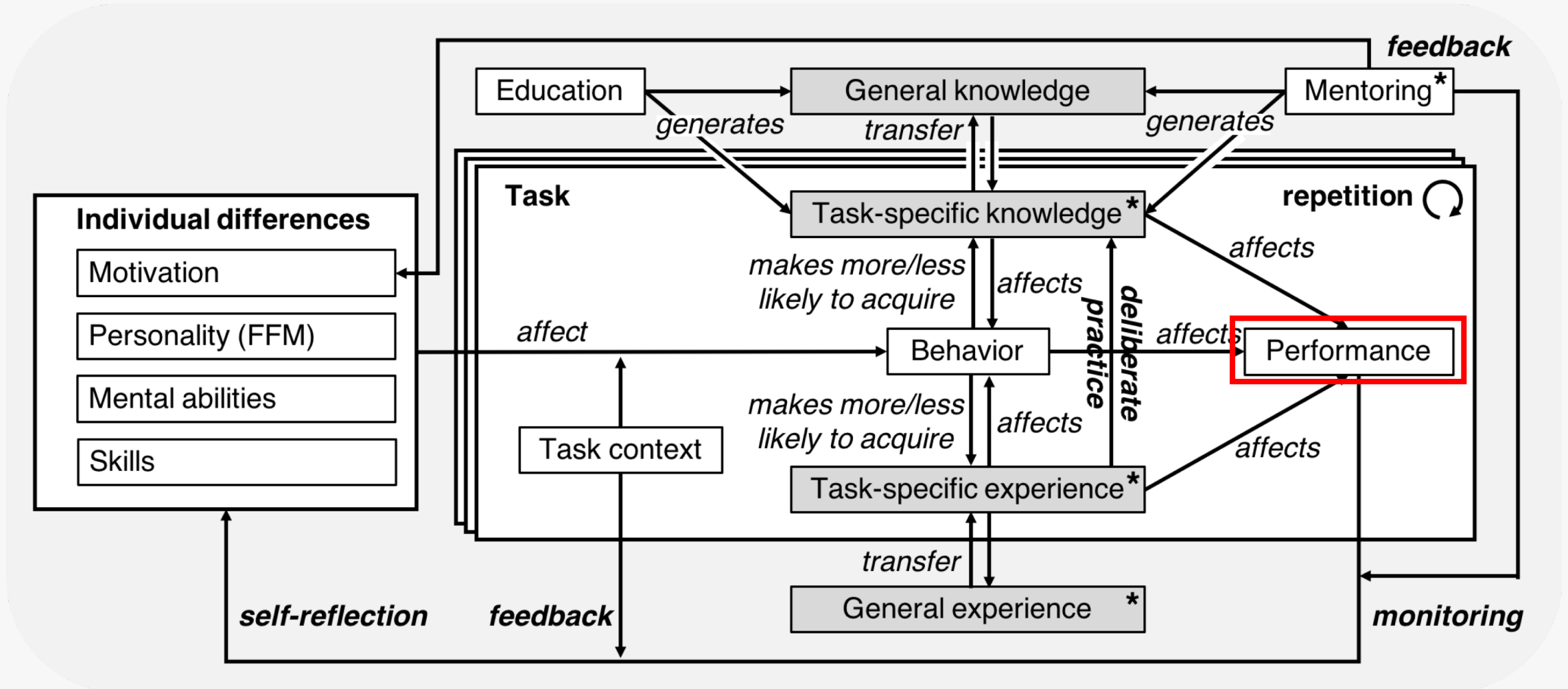


- **(Self-)reflection** and **feedback** important to **monitor** progress towards goal achievement (Locke and Latham, 1990)
- *“[T]he more **channels of accurate and helpful feedback** we have access to, the better we are likely to perform.”*
(Tourish and Hargie, 2003)
- **Mentors**, teachers, and peers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

Conceptual Theory



Conceptual Theory



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 in 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.”*

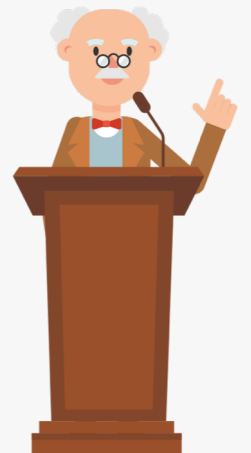


software developer, age 60

Summary for Researchers

Researchers can...

- Use our theory to **design studies** on expertise development
(first operationalizations described in the paper)
- Adopt our **theory building** approach
- Consider our results when asking for expertise **self-assessments**
(task-specific view, context important, explicitly mention characteristics of experts)



Summary for Developers

Developers can...

- Learn what other developers expect from **experts** and **mentors**
- Learn which **behaviors** may lead to becoming an expert
 - Deliberate practice
 - Having challenging goals
 - Building and maintaining a supportive work environment
 - Asking for feedback from peers
 - Explicitly reflecting about what one knows and what not



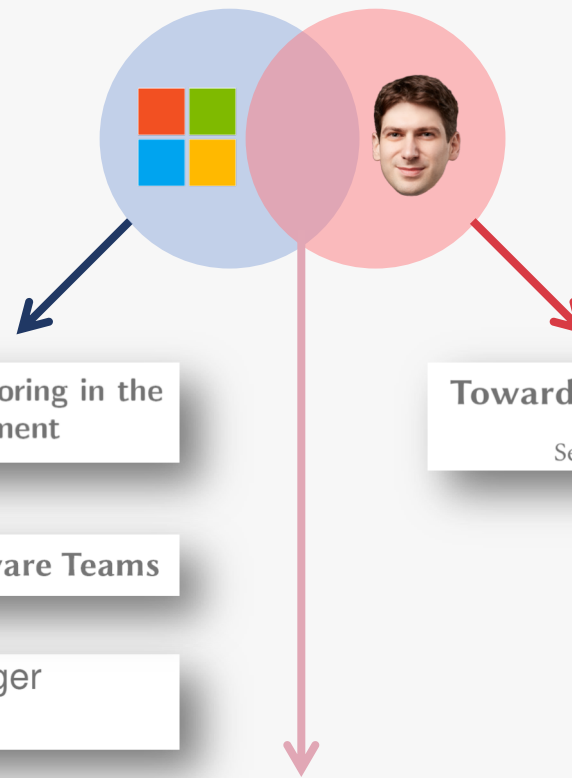
Summary for Employers

Employers can...

- Learn what **(de)motivates** employees and thus hinders expertise development
 - problem solving vs. non-challenging work
- Reflect on ideas to build a work environment **supporting self-improvement** of their staff:
 - Good mix of continuity and change in processes
 - Communicating clear visions, directions, and goals
 - Rewarding high-quality work wherever possible
 - Revisiting information sharing in company



Expertise Development



Design Recommendations for Self-Monitoring in the Workplace: Studies in Software Development

Towards a Theory of Software Development Expertise

Sebastian Baltes

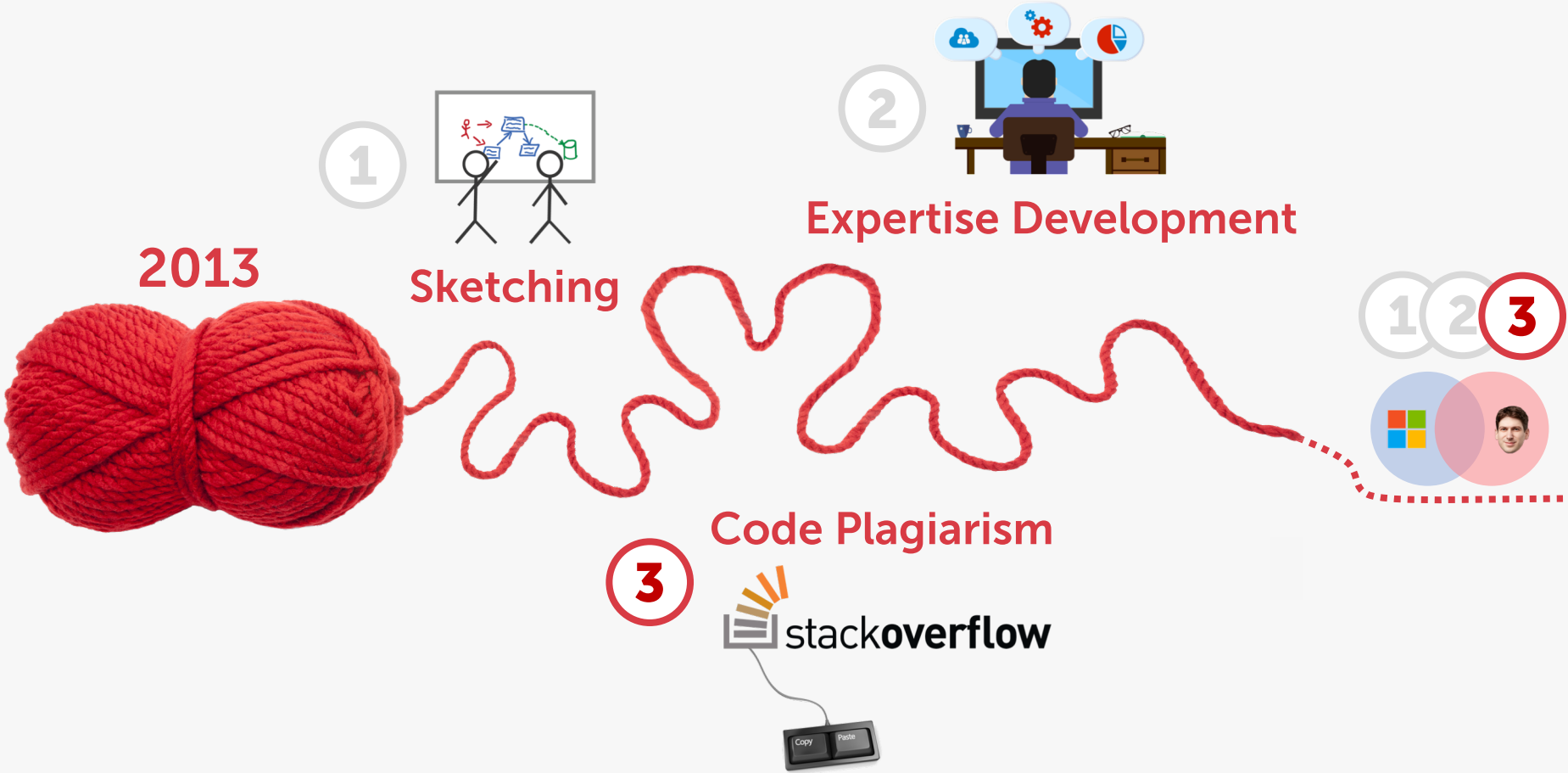
Stephan Diehl

A Study of the Organizational Dynamics of Software Teams

What Makes a Great Manager of Software Engineers?

- *Study:* Identify **specific challenges** that **older developers** face to prevent such experienced knowledge workers from dropping out of software development
- *Study:* Further investigate role of **feedback** and **team changes** in expertise development
- *Study:* Investigate expertise development and ageing from a **sociological perspective** (team expertise, discourse analysis)

Overview of this Talk



Code Plagiarism





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



Usage and attribution of Stack Overflow code snippets in GitHub projects

Sebastian Baltes¹  · Stephan Diehl¹ 

Published online: 01 October 2018
© Springer Science+Business Media, LLC, part of Springer Nature 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/>

Question for the Audience I

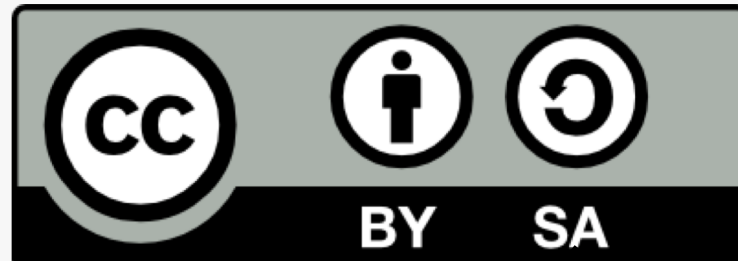
Who admits regularly copying non-trivial code snippets from Stack Overflow?



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."*



Attribution

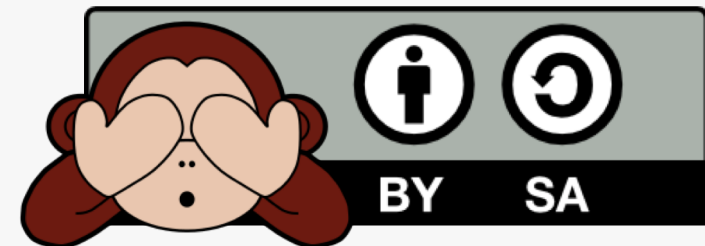
Share-alike

*"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]



Here's what I do:

889

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 getLocation(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



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

    public boolean getLocation(Context context, LocationResult result)
    {
        // Use LocationResult callback class to pass location value from MyLocation to user code.
        locationResult=result;
        if(lm==null)
            lm = (LocationManager) context.getSystemService(Context.LOCATION_SERVICE);

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

        //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, locationListenerGps);
        if(network_enabled)
            lm.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListenerNetwork);
        timer1=new Timer();
        timer1.schedule(new GetLastLocation(), 20000);
        return true;
    }

    LocationListener locationListenerGps = new LocationListener() {
        public void onLocationChanged(Location location) {
            timer1.cancel();
            locationResult.getLocation(location);
            lm.removeUpdates(this);
            lm.removeUpdates(locationListenerNetwork);
        }
        public void onProviderDisabled(String provider) {}
        public void onProviderEnabled(String provider) {}
        public void onStatusChanged(String provider, int status, Bundle extras) {}
    };

    LocationListener locationListenerNetwork = new LocationListener() {
        public void onLocationChanged(Location location) {
            timer1.cancel();
            locationResult.getLocation(location);
            lm.removeUpdates(this);
            lm.removeUpdates(locationListenerGps);
        }
        public void onProviderDisabled(String provider) {}
        public void onProviderEnabled(String provider) {}
        public void onStatusChanged(String provider, int status, Bundle extras) {}
    };

    class GetLastLocation extends TimerTask {
        @Override
        public void run() {
            lm.removeUpdates(locationListenerGps);
            lm.removeUpdates(locationListenerNetwork);

            Location net_loc=null, gps_loc=null;
            if(gps_enabled)
                gps_loc=lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);
            if(network_enabled)
                net_loc=lm.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);


            //if there are both values use the latest one
            if(gps_loc!=null && net_loc!=null){
                if(gps_loc.getTime()>net_loc.getTime())
                    locationResult.getLocation(gps_loc);
                else
                    locationResult.getLocation(net_loc);
                return;
            }

            if(gps_loc!=null){
                locationResult.getLocation(gps_loc);
                return;
            }
            if(net_loc!=null){
                locationResult.getLocation(net_loc);
                return;
            }
            locationResult.getLocation(null);
        }
    }



    public static abstract class LocationResult{
        public abstract void getLocation(Location location);
    }
}
```



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

Microsoft / ApplicationInsights-Home

Code Issues 72 Pull requests 0 Projects 0 Wiki Insights

Unclear licensing situation for code in AccountController.cs #328

Open

Microsoft / rDSN

Code Issues 5 Pull requests 1 Projects 0 Wiki Insights

Unclear licensing situation for code in csproj.template.php #209

Open

Microsoft / Windows-universal-samples

Code Issues 42 Pull requests 55 Projects 0 Wiki Insights

Unclear licensing situation for code in **BindableFlyout.cs** #1070

Open sbaltes opened this issue a day ago · 1 comment

Yes.

11 I put together a simple solution for developers who desire this functionality. It uses an attached property to identify the ItemsSource and the ItemTemplate for a Flyout control. If the developer elects to use a MenuFlyoutItem or something else, it is up to them.

Here's the attached property:

```
public class BindableFlyout : DependencyObject
{
    #region ItemsSource
    public static IEnumerable GetItemsSource(DependencyObject obj)
    {
        return obj.GetValue(ItemsSourceProperty) as IEnumerable;
    }
    public static void SetItemsSource(DependencyObject obj, IEnumerable value)
    {
        obj.SetValue(ItemsSourceProperty, value);
    }
    public static readonly DependencyProperty ItemsSourceProperty =
        DependencyProperty.RegisterAttached("ItemsSource", typeof(IEnumerable),
            typeof(BindableFlyout), new PropertyMetadata(null, ItemsSourceChanged));
    private static void ItemsSourceChanged(DependencyObject d, DependencyPropertyChangedEventArgs e)
    {
        Setup(d as Windows.UI.Xaml.Controls.Flyout);
    }
    #endregion

    #region ItemTemplate
    public static DataTemplate GetItemTemplate(DependencyObject obj)
    {
        return (DataTemplate)obj.GetValue(ItemTemplateProperty);
    }
    public static void SetItemTemplate(DependencyObject obj, DataTemplate value)
    {
        obj.SetValue(ItemTemplateProperty, value);
    }
    public static readonly DependencyProperty ItemTemplateProperty =
        DependencyProperty.RegisterAttached("ItemTemplate", typeof(DataTemplate),
            typeof(BindableFlyout), new PropertyMetadata(null, ItemTemplateChanged));
    private static void ItemTemplateChanged(DependencyObject d, DependencyPropertyChangedEventArgs e)
    {
    }
    #endregion
}
```


"But do Stack Overflow authors care about attribution?"

<https://meta.stackexchange.com/q/273168>

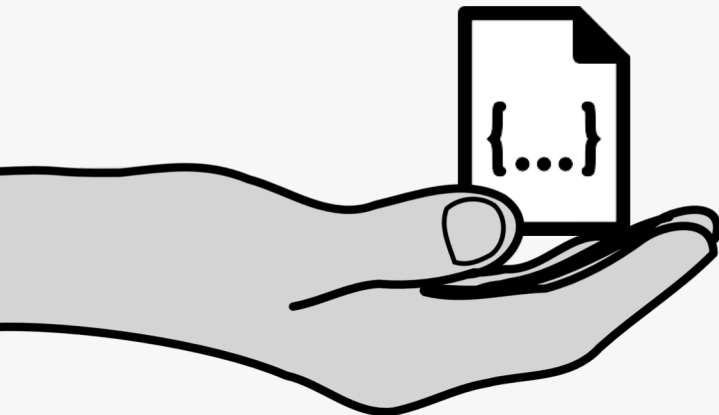
The screenshot shows a Meta Stack Exchange thread. The main title is "The MIT License – Clarity on Using Code on Stack Overflow and Stack Exchange". The thread has 505 votes and is marked as a star. The content includes several updates: an update from December regarding digesting answers, an update from January postponing the decision, and a "Welcome!" message. A "TLDR" section explains the purpose of the post: to address concerns about attribution requirements for code posted on Stack Overflow. The TLDR states that the purpose is to address the concern expressed most frequently in response to the initial proposal: no attribution requirement. Also, we want to make sure everyone has ample opportunity to provide feedback and we have time to consider it. We are more concerned with doing this right than doing it fast, so please let us know what you think about this proposed change.

<https://meta.stackexchange.com/q/272956>

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.

Method 1: Regular Expressions



209m files in
4.1m projects

.java

13m Java files in
336k projects

...stackoverflow\.com...

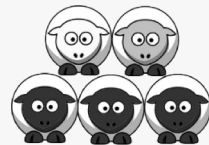
10 most frequently
referenced answers



Check if attributed
(URL to answer or
corresponding question)

Check for false
positives

4,198 files
with matches



Check external
availability



Manually build regular
expressions matching
code snippets
(referenced usages as test cases)

Exemplary Regex

```
public static String humanReadableByteCount(long bytes, boolean si) {  
    int unit = si ? 1000 : 1024;  
    if (bytes < unit) return bytes + " B";  
    int exp = (int) (Math.log(bytes) / Math.log(unit));  
    String pre = (si ? "KMGTPE" : "KMGTPE").charAt(exp-1) + (si ? "" : "i");  
    return String.format("%.1f %sB", bytes / Math.pow(unit, exp), pre);  
}
```



```
((?i:String[\s]+\w+\([^{\}*long[^{\}]+\)[\s]*\{[\s\S]+if[\s]*\([^<]+<[^\\)]+\)[\s\S]*return[^;]+\+[^;]*"\ B"[\s\S]+int[\s][^=]+\=[\s]*\([\s]*int[\s]*\)[\s]*\([\s]*Math[\s]*\.[\s]*log[\s]*\([^\\)]+\)[\s]*\[/\s]*Math[\s]*\.[\s]*log[\s]*\([^\\)]+\)[\s]*\)[\s\S]+return[^}]+String[\s]*\.[\s]*format[\s]*\([^{\}]+\})
```

<https://stackoverflow.com/a/3758880>

Results

Rank	Matches				Recall	Attribution	
	ALL	DISTINCT	REF	NO-REF	REF/ F_{AQ}	REF/DISTINCT	F_{AQ} /DIST.
1	997	448	97	351	79.5%	21.7%	27.2%
2	1,843	913	60	853	60.0%	6.6%	11.0%
3	2,662	902	87	815	80.6%	9.6%	12.0%
4	420	170	18	152	94.7%	10.6%	11.2%
5	1,492	402	25	377	73.5%	6.2%	8.5%
6	2,642	807	65	742	87.8%	8.1%	9.2%
7	160	124	12	112	29.3%	9.7%	33.1%
8	355	174	22	152	61.1%	12.6%	20.7%
9	295	225	5	220	10.6%	2.2%	20.9%
10	65	33	11	22	42.3%	33.3%	78.8%
All	10,931	4,198	402	3,796	<i>M</i> 61.9%	<i>M</i> 12.1%	<i>M</i> 23.2%

Method 2: Code Clone Detector

- **Goal:** Use code clone detector to find clones of a sample of Stack Overflow snippets in a sample of GitHub projects

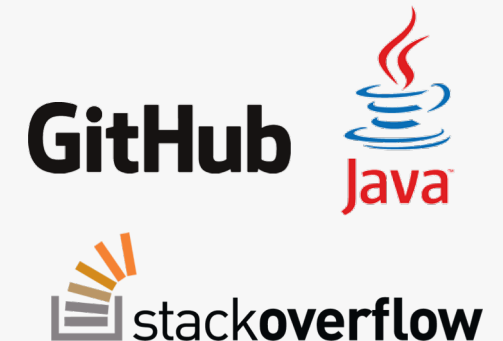
- *Why samples?*

- Code clone detection is computationally expensive



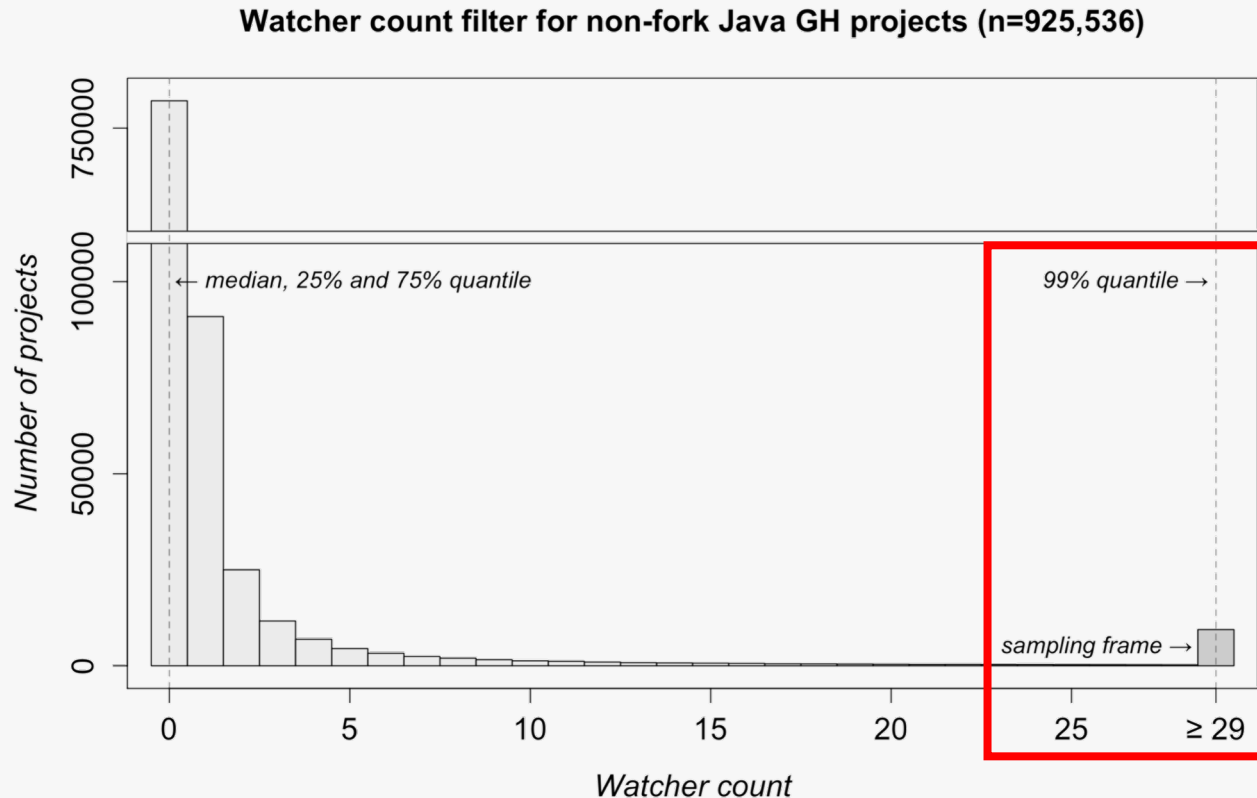
- *Which snippets and projects to select?*

- Random samples: Many **toy projects** on GitHub and many **irrelevant snippets** on Stack Overflow
- Purposive sampling: Limited generalizability



GitHub Project Sample

- Focus on **popular** GitHub projects
- High precision in selecting “engineered” software projects [Munaiah et al. 2017]
- Greater (potential) impact of licensing issues



Sample size:
3,000 / 2,313



Stack Overflow Snippet Samples

- Non-trivial snippets retrieved from 100 most frequently referenced answers (n=111)
⇒ S_{top100}
- Non-trivial snippets retrieved from answers referenced in GitHub projects (n=137)
⇒ S_{gh}
- *External sources*: Only three snippets available under a more permissive license than CC BY-SA

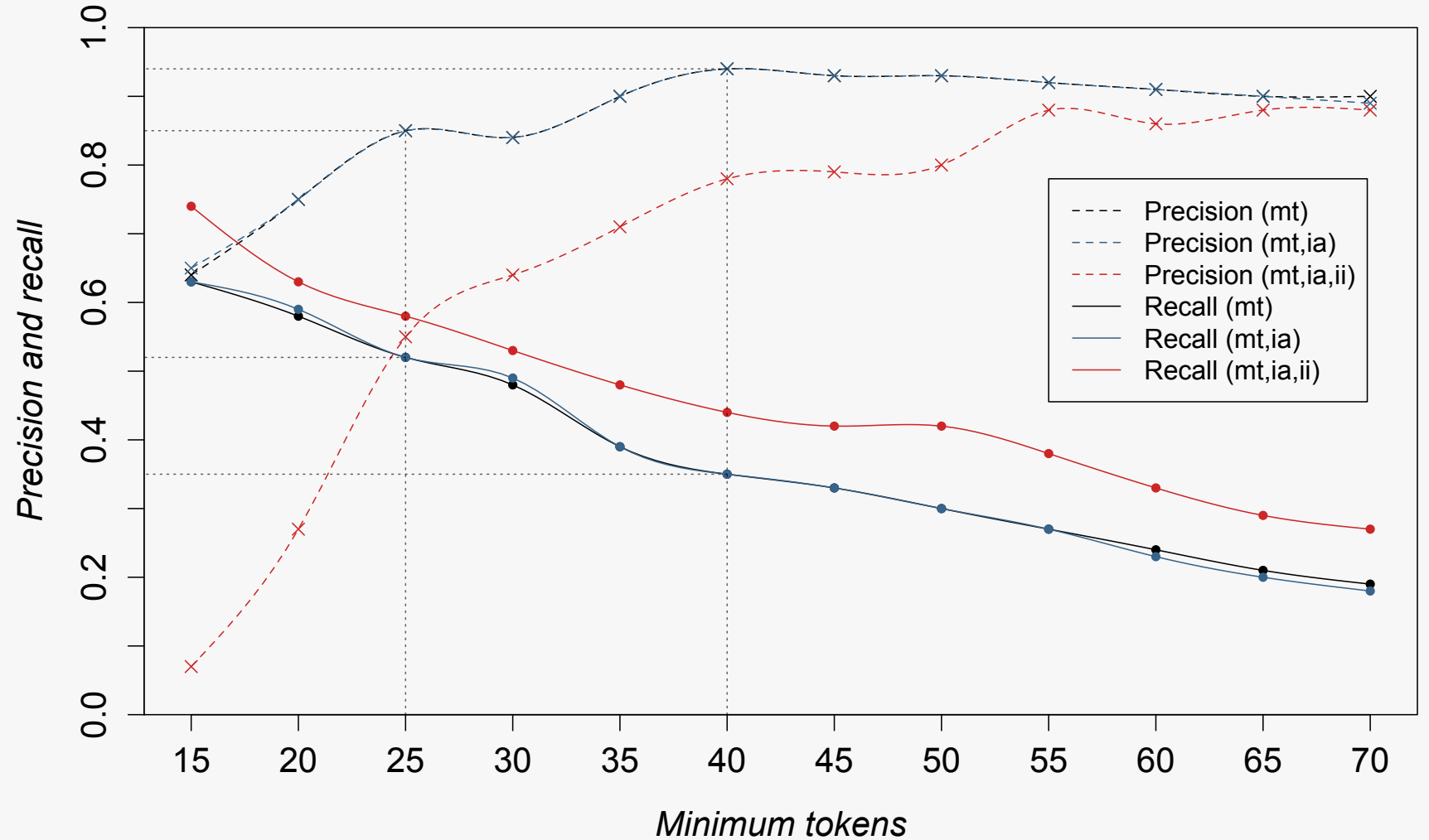


Code Clone Detector Calibration



<https://pmd.github.io/>

Comparison of CPD configurations



Results

Set	Snippets				Files		Repos
	ALL	MATCHED	ANSWERS	MATCHED	MATCH.	REF	MATCHED
S_{gh}	137	53 (39%)	102	52 (51%)	163	58 (36%)	124 (5%)
S_{top100}	111	48 (43%)	85	46 (54%)	173	25 (14%)	125 (5%)
US	222	101 (46%)	169	86 (51%)	297	70 (24%)	199 (9%)

Method 3: Exact Matches

- **Goal:** Address shortcomings of Method 1 and 2
 - Increase sample sizes
 - Exclude snippets available on external sources
 - Systematically exclude short snippets
- Select as many projects and snippets as possible and search for (almost) exact matches



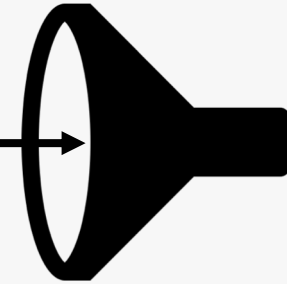
Method 3: Exact Matches


 Google Cloud

GitHub 

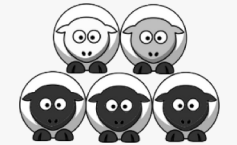
209m files in
4.1m projects

- ✓ Project is not a fork, has ≥ 5 Java files and ≥ 1 watcher(s)
- ✓ File has ending `.java` has ≥ 68 NLOC (Q_3)




1.7m Java files
in 64k projects

Normalization and
substring search



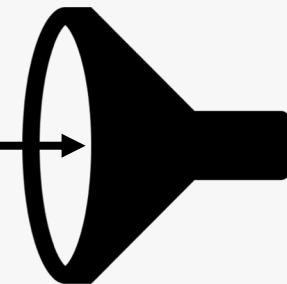
10,358
matches

 Google Cloud

 **stackoverflow**

21m answers

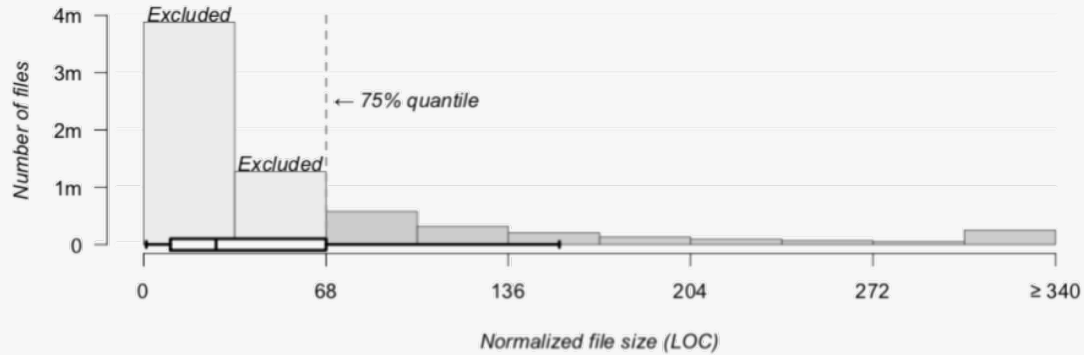
- ✓ Question tagged `java` or `android`
- ✓ Answer score ≥ 10
- ✓ Code block ≥ 6 NLOC



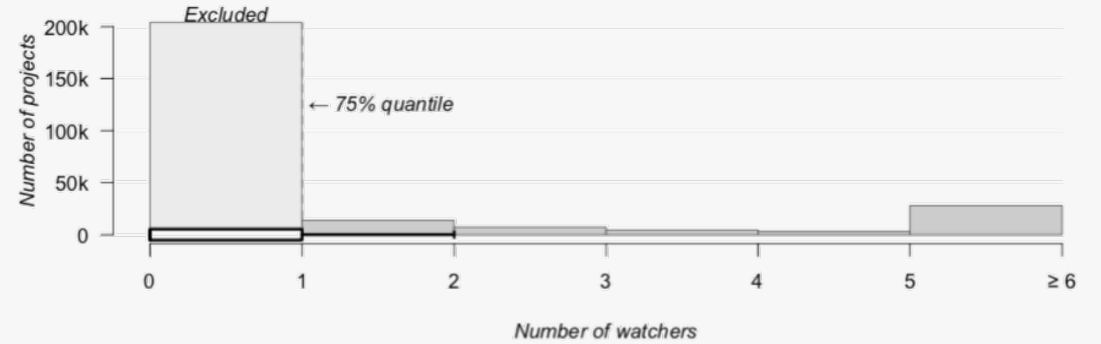

29k snippets from
24k answers

Details: Filtering of GitHub Projects

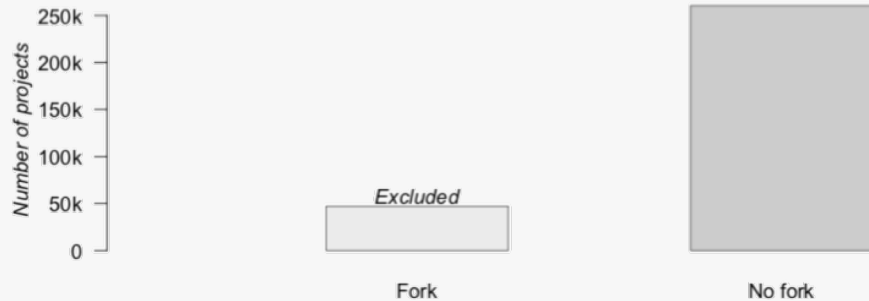
File size filter for GH Java files (n=6,851,022)



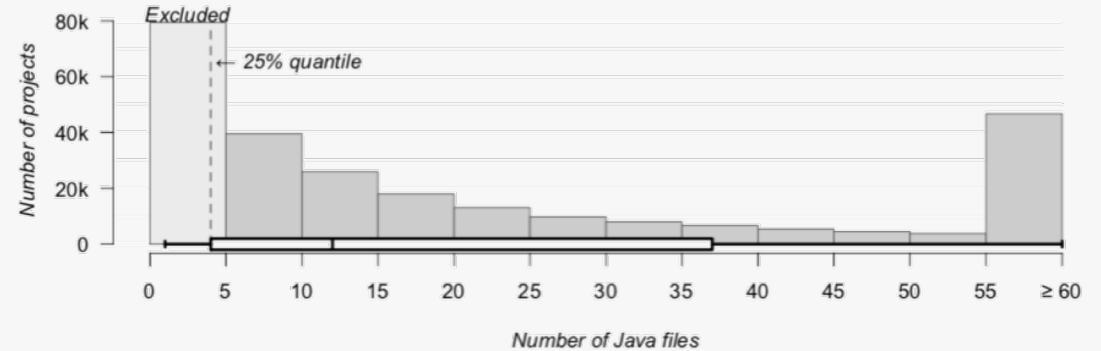
Watcher count filter for GH Java projects (n=260,498)



Fork filter for GH projects containing Java files (n=307,489)

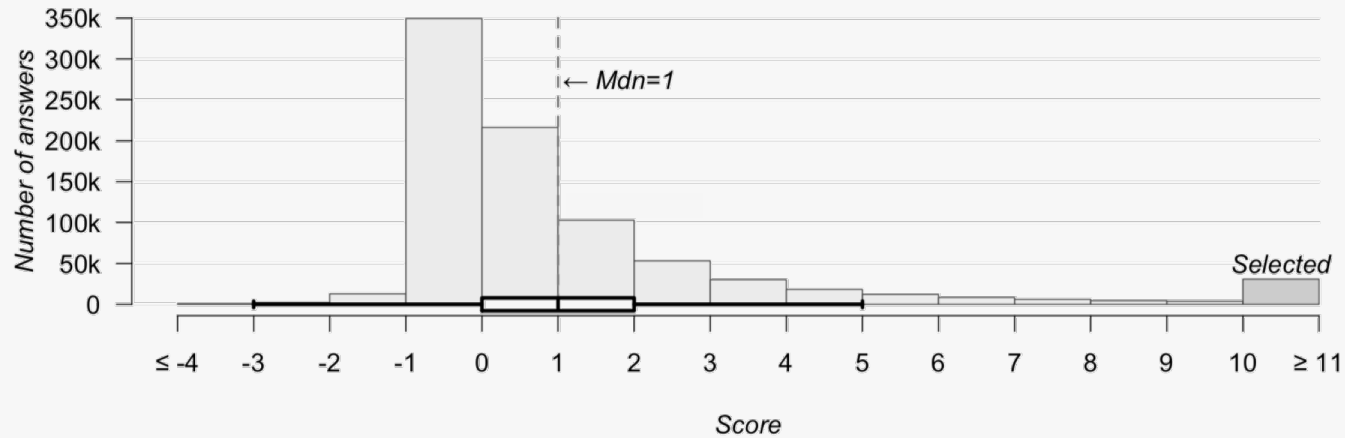


File count filter for GH Java projects (n=260,498)

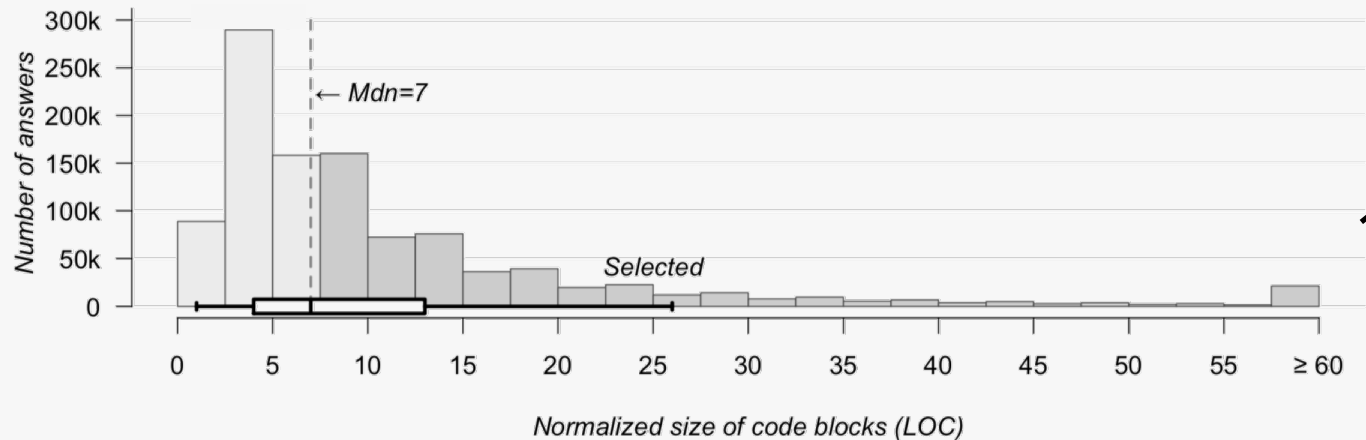


Details: Filtering of Stack Overflow Snippets

Score filter for SO Java answers (n=851,795)

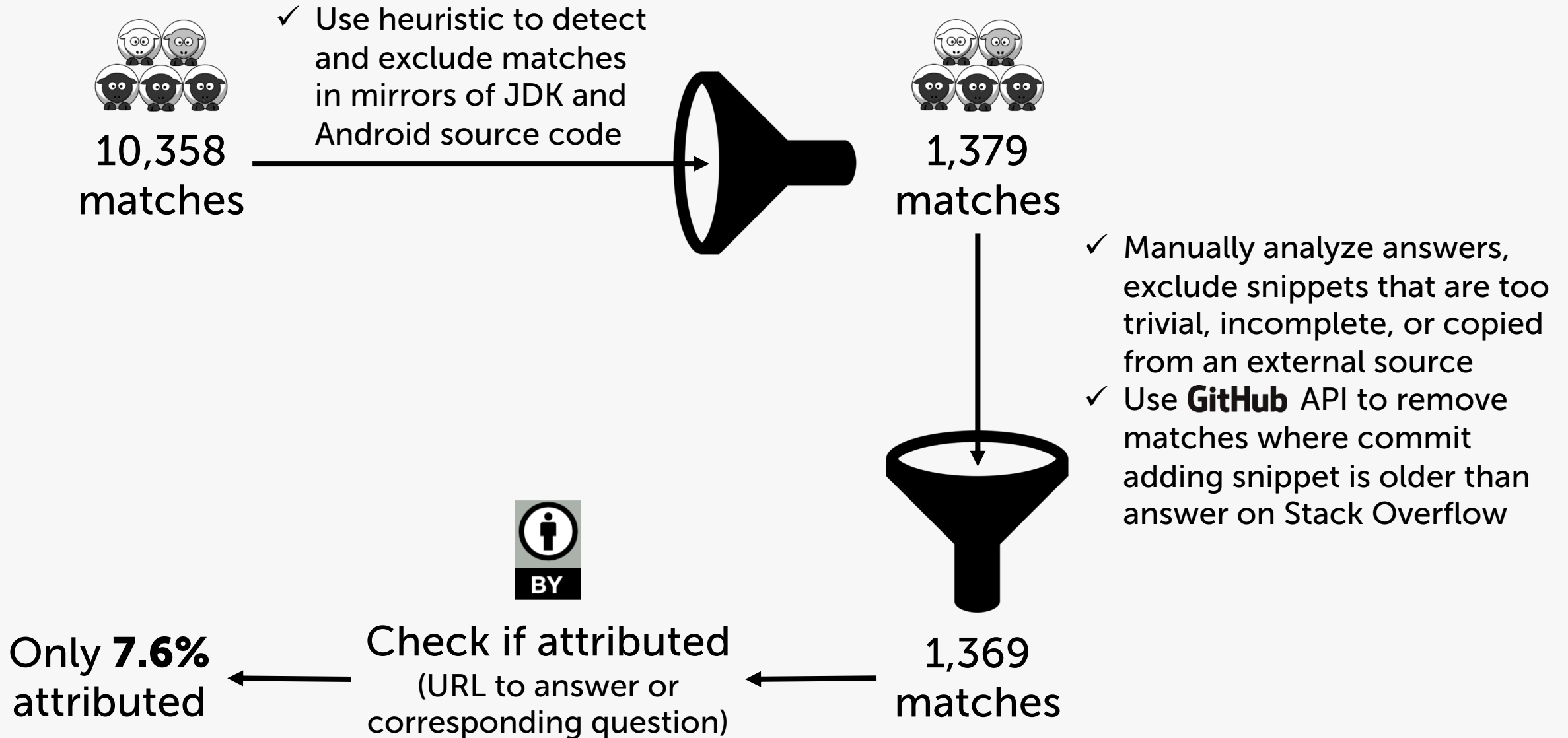


Length filter for SO Java code blocks (n=1,063,993)



Proxies for originality

Method 3: Filtering of Matches



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 (<i>n</i> = 2,962)	attributed (<i>n</i> = 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 (<i>n</i> = 144)	attributed (<i>n</i> = 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 (<i>n</i> = 1,169)	attributed (<i>n</i> = 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%)

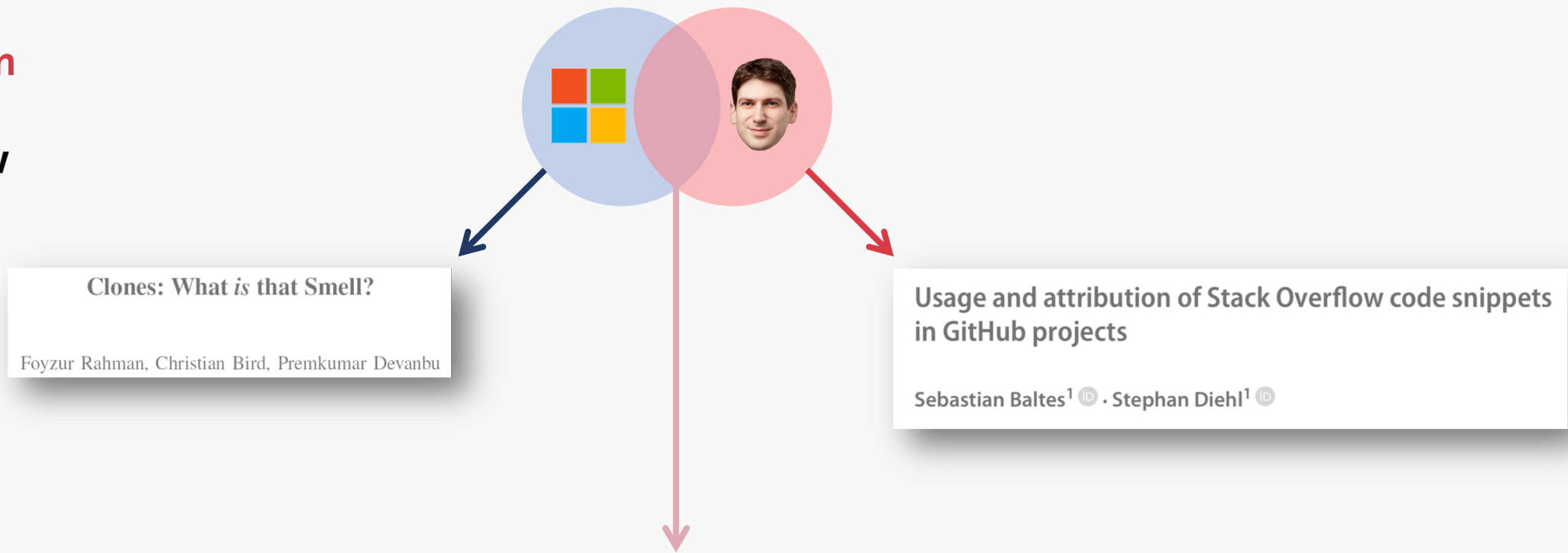
Method 3

Reaching out to Developers

- **Contacted owners** of GitHub projects containing copies of Stack Overflow snippets
- **75% not aware** of CC BY-SA licensing
(see slide about online surveys)
- Many thankful responses



Code Plagiarism



- *Study*: Investigate **code reuse from online resources** within Microsoft (Stack Overflow, tutorials, official documentation resources, etc.)
- *Tool support*: Support **maintainability** of copied snippets by automatically adding links to sources, integration into CI tools
- *Education*: Help developers **understand complex licensing situations** (not only for complete libraries but also for individual snippets)

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

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 site for software developers, providing snippets and free-form text on a wide range of software artifacts, questions and answers for example when bugs in code snippet work with a more recent library version. As code snippet is edited for clarity. To be able to track how SO evolves, we built *SOTorrent*, an official SO data dump. *SOTorrent* provides a history of SO content at the level of whole code blocks. It connects 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.

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.

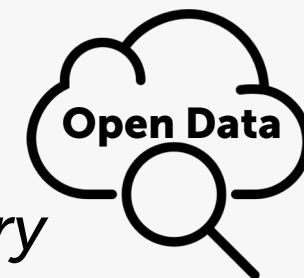
dataset [16] that enables researchers to analyze the version history of SO posts at the level of individual text and code blocks (see Figure 1 for exemplary posts). The official SO data dump [1] keeps track of different versions of entire posts, but does not contain information about differences between 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



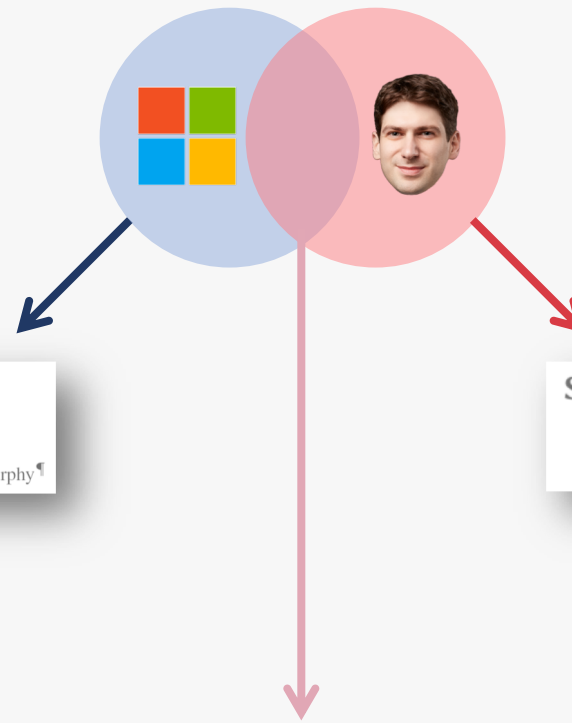
MSR 2018/19

sotorrent.org

Dataset available on Zenodo and BigQuery

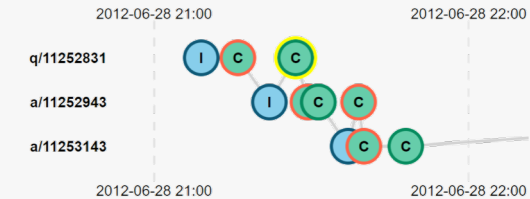


Mining Stack Overflow

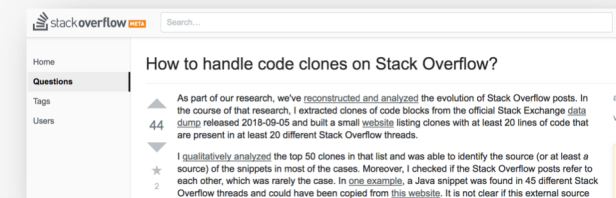


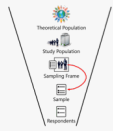
Change Bursts as Defect Predictors
Nachiappan Nagappan* Andreas Zeller† Thomas Zimmermann‡ Kim Herzig§ Brendan Murphy¶

SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts
Sebastian Baltes Christoph Treude Stephan Diehl



- *Study:* Analyze **communication and edit patterns** related to code on Stack Overflow, identify buggy snippet versions
- *Study:* Analyze links to better understand Stack Overflow's role in the **ecosystem** of documentation resources
- *Tool:* **Clones** within Stack Overflow affect maintainability of code snippets, may indicate duplicate posts. Not clear yet which tool support is desired by the community

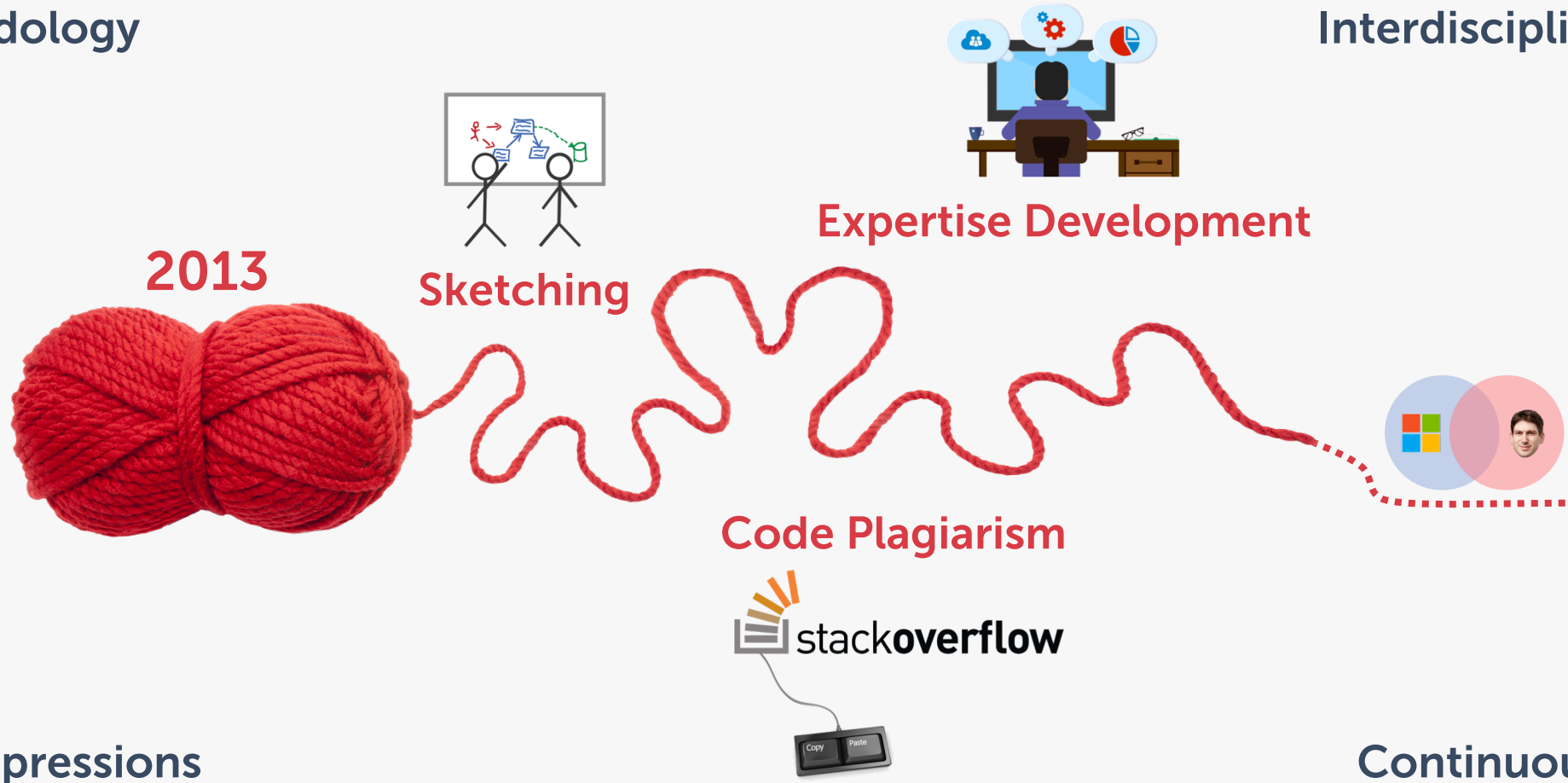




Issues in Sampling Software Developers Methodology



Constructing Urban Tourism Space Digitally Interdisciplinary Research



Regular Expressions RegViz

Evidence-based Practice
through
Practice-based Evidence

