Enhancing Productivity by Example

Ohad Barzilay

Tel Aviv University

Based on a PhD research under the supervision of:

Amiram Yehudai, Blavatnik School of Computer Science, Tel-Aviv University

Orit Hazzan, Department of Education in Technology and Science, Technion

April 2011

Agile Israel 2011

Session: Open Your Mind...

Agenda

□ Examples are good

□ We should use them more!

Examples and related areas

- We define example usage as using an already existing code fragment (the example) within a new context. For example:
  - Looking for code in the Internet
  - Examples that are part of the documentation
  - Examining code in the code base of the organization
  - And much more…

- There is some overlap with the following areas:
  - Reuse
  - Copy and paste
  - Patterns

Benefits of Example Usage

- Developers use examples in their work
- Examples are not unique to software development

- Using examples has many potential benefits:
  - Enhancing **software reuse**
  - Promotes **learning** and **understanding**
  - **Productivity** increase
  - Improvement of **code quality**
  - Enforcement of **consistency of design** and of **coding standards**
  - Establishment of an effective **knowledge transfer** mechanism inside the organization and outside of it

Diversity of Example Usage

- Despite its benefits, we find that example usage is diverse:
  - Its benefits are **not consensually appreciated** by the developers
  - The software development community does not address the **barriers** and **challenges** that accompany extensive example usage
  - Different developers are **attentive** to different uses of examples
  - There is no explicit, widely accepted set of best practices for using examples methodologically nor systematically

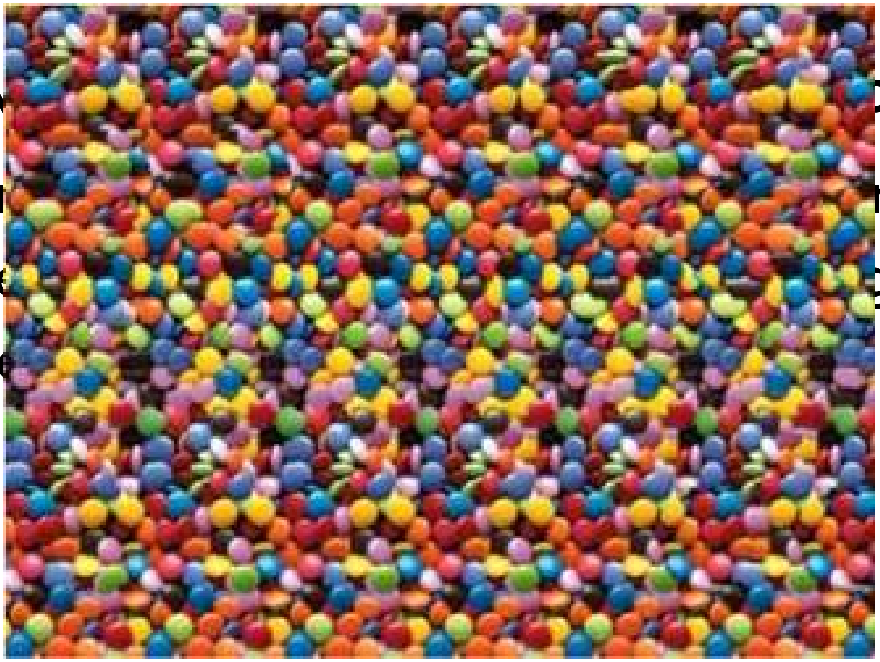
By becoming **aware** to the **diversity** of example usage we could find ways to increase example usage in our development process

Using examples in professional software development **is like teenage sex…**

- No one is sure what it is, but they hear that **it's great**
- Everyone thinks that everyone else is doing it
- Those who say they are doing it all the time are probably lying
- Although it is **natural**, those who do it sometimes feel **guilty**
- Once they start doing it,
  - **safety** is not their main concern
  - they'll discover that it is going to take a while to figure out how to get really **good at it**
  - they'll realize that they'll have to try to discover a whole bunch of **new methods** of doing it to really figure it all out

Introduction

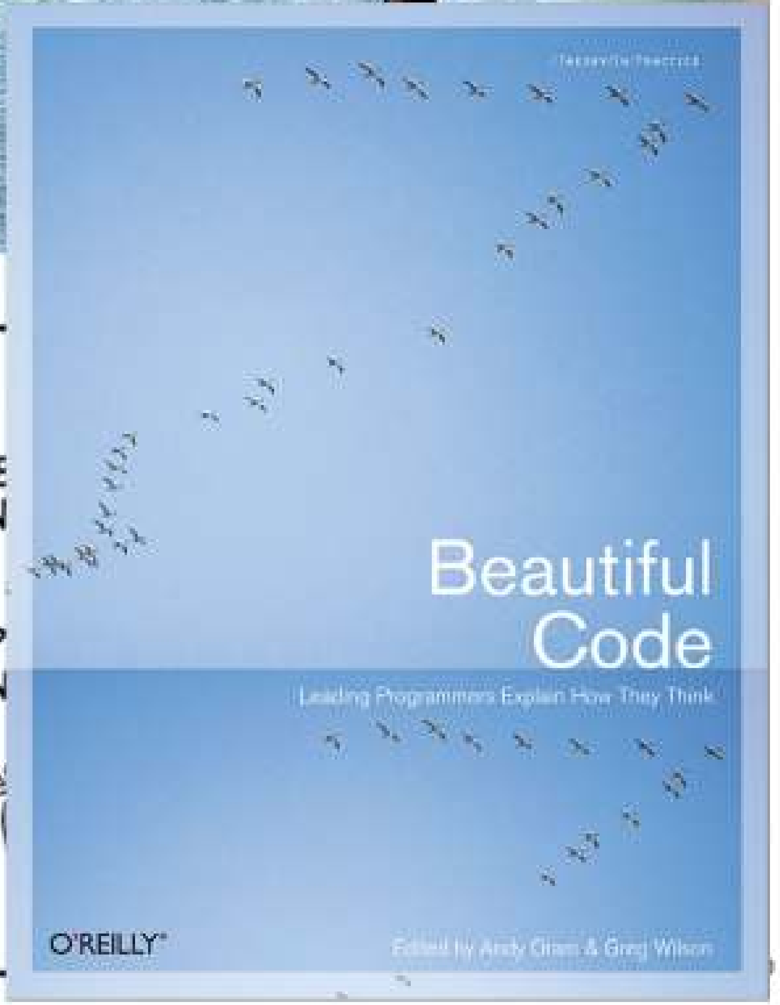
- We did not embark on this research with example usage in mind



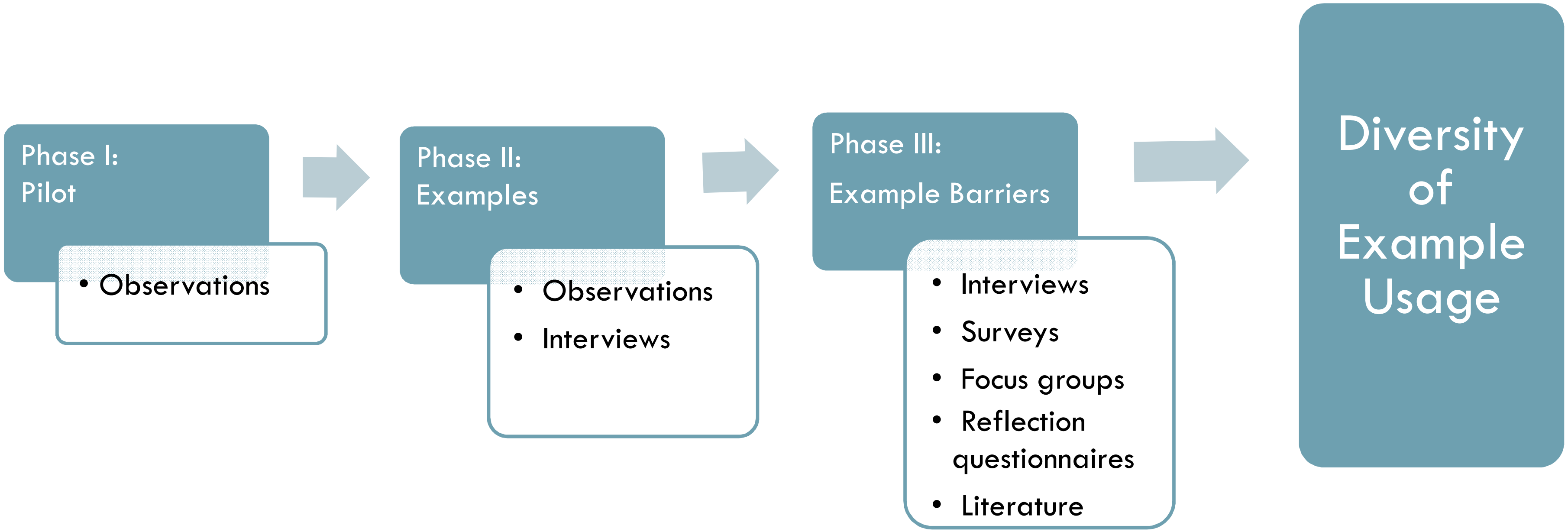
- Our motiv ctivity
  - Refactor nformally for years be given a name [Opdyke

Looking for “The Next Refactoring”

I'M ANALYZING SOME PIECE OF CODE I'VE FOUND
I LOVE TO DO THIS. CRAWLING THROUGH SOME VERY OLD CODE AND ASKIN MYSELF:
- WHO WROTE IT?
- WHEN DID HE WRITE IT?
- WHAT DID HE THINK THEN


THEORY IN PRACTICE
Beautiful Code
Leading Programmers Explain How They Think
O'REILLY
Edited by Andy Oram & Greg Wilson

Research overview



Phase I:
Pilot

- Observations

Phase II:
Examples

- Observations
- Interviews

Phase III:
Example Barriers

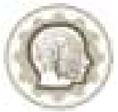
- Interviews
- Surveys
- Focus groups
- Reflection questionnaires
- Literature

Diversity of Example Usage

□ Grounded Theory

Diversity of Example Usage

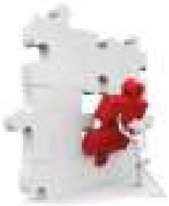
- We identify 9 **human and social factors** that dominate developers' approach to example usage



- We identify 3 types of lack of developer **attentiveness** to example usage

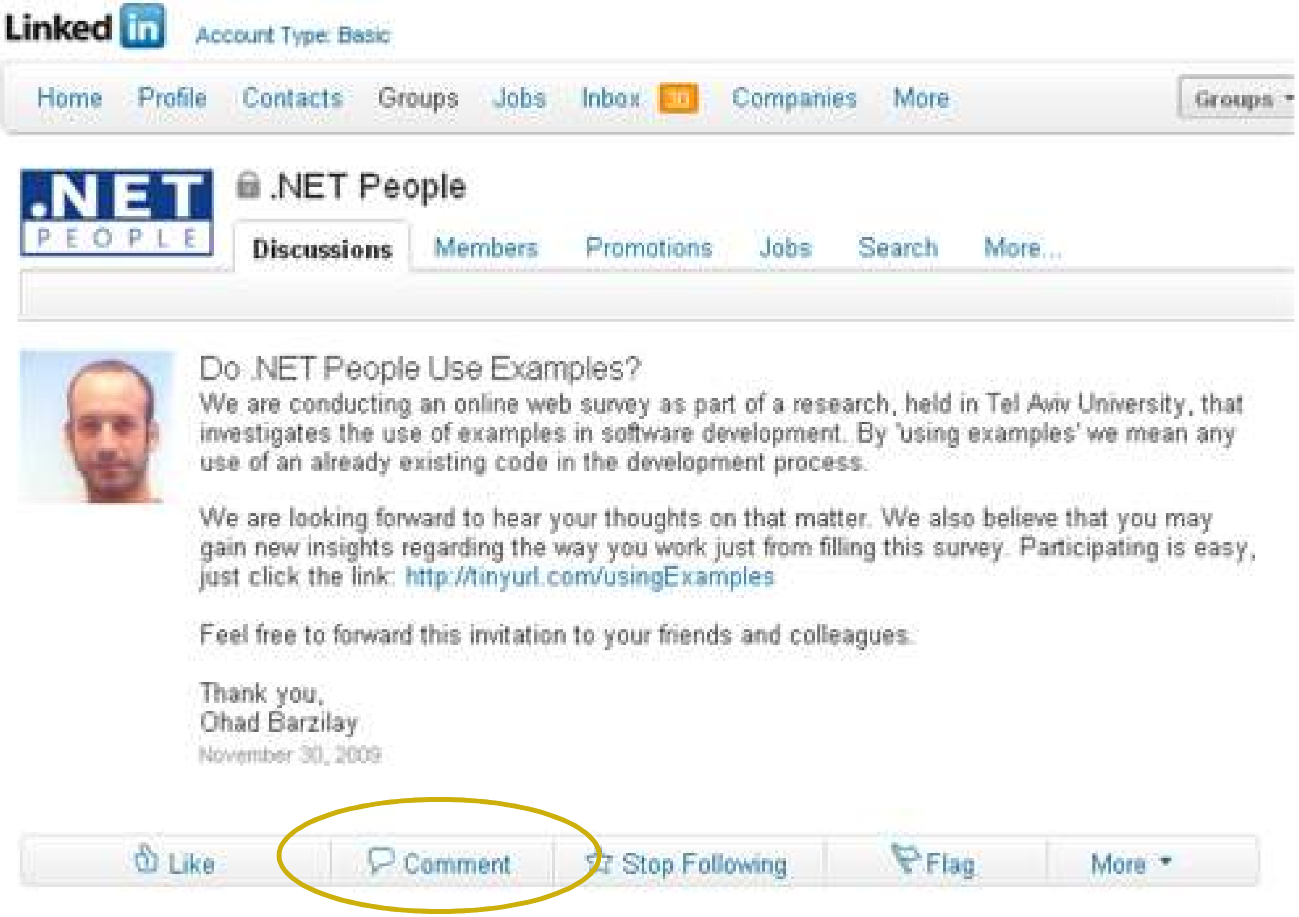
- We identify 15 **motivations** for using examples by professional developers, and divide them into three categories

- We propose a comprehensive approach that would conceptualize **example embedding**



Human and Social Aspects of Example Usage

LinkedIn Virtual Focus Group



Do .NET People Use Examples?

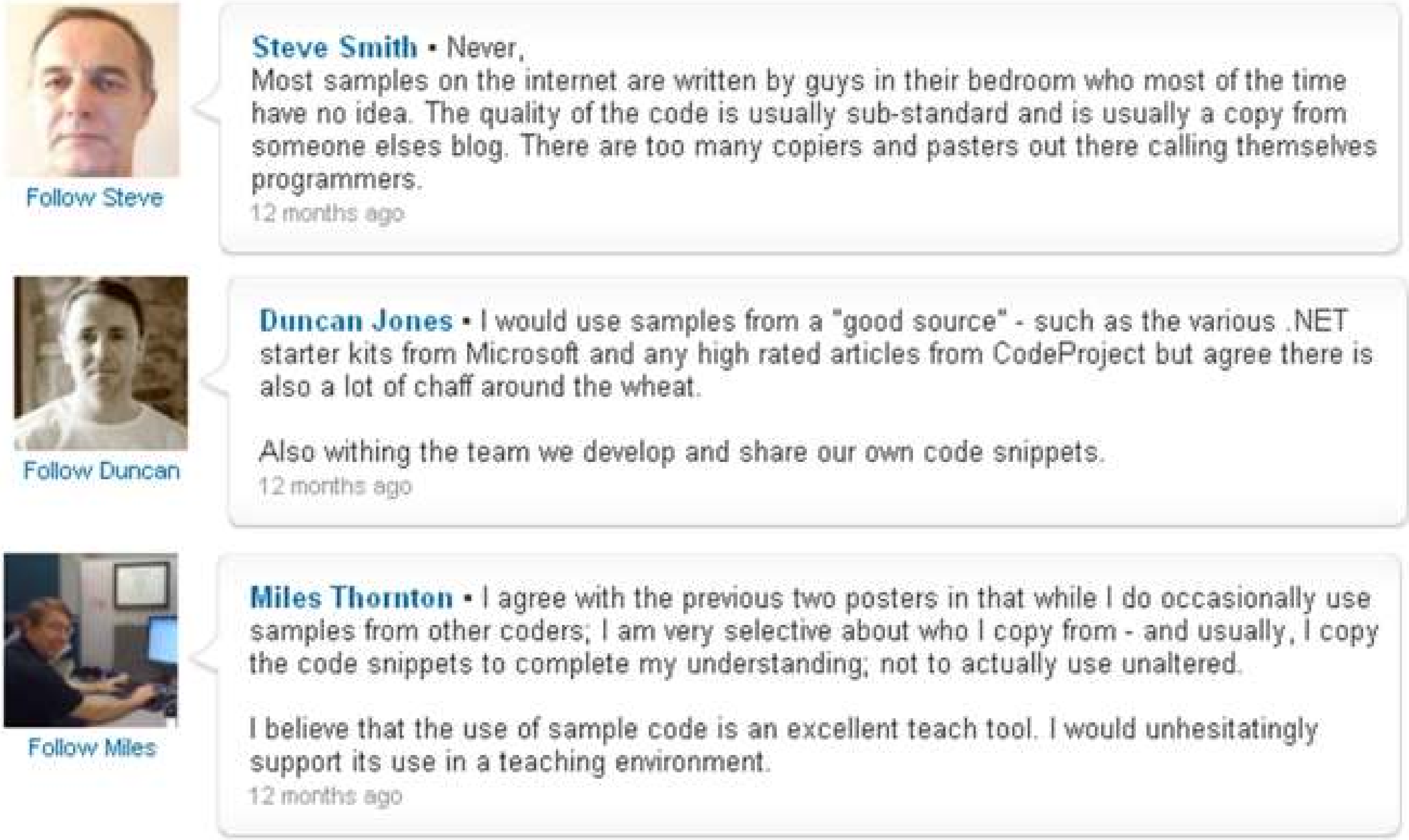
We are conducting an online web survey as part of a research, held in Tel Aviv University, that investigates the use of examples in software development. By 'using examples' we mean any use of an already existing code in the development process.

We are looking forward to hear your thoughts on that matter. We also believe that you may gain new insights regarding the way you work just from filling this survey. Participating is easy, just click the link: http://tinyurl.com/usingExamples

Feel free to forward this invitation to your friends and colleagues.

Thank you,
Ohad Barzilay
November 30, 2009

Not everyone loves examples…



Steve Smith • Never,
Most samples on the internet are written by guys in their bedroom who most of the time have no idea. The quality of the code is usually sub-standard and is usually a copy from someone elses blog. There are too many copiers and pasters out there calling themselves programmers.
12 months ago

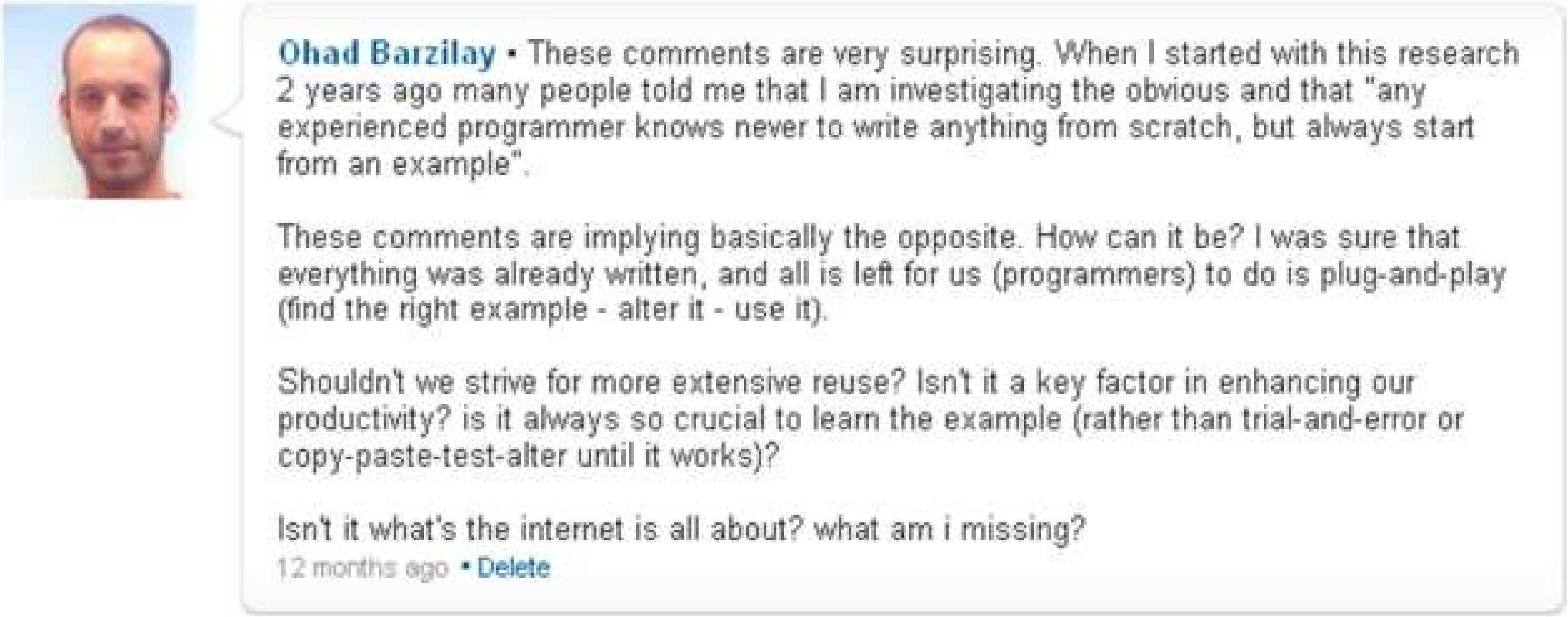
Duncan Jones • I would use samples from a "good source" - such as the various .NET starter kits from Microsoft and any high rated articles from CodeProject but agree there is also a lot of chaff around the wheat.

Also withing the team we develop and share our own code snippets.
12 months ago

Miles Thornton • I agree with the previous two posters in that while I do occasionally use samples from other coders; I am very selective about who I copy from - and usually, I copy the code snippets to complete my understanding; not to actually use unaltered.

I believe that the use of sample code is an excellent teach tool. I would unhesitatingly support its use in a teaching environment.
12 months ago

Starting a fire…



Ohad Barzilay • These comments are very surprising. When I started with this research 2 years ago many people told me that I am investigating the obvious and that "any experienced programmer knows never to write anything from scratch, but always start from an example".

These comments are implying basically the opposite. How can it be? I was sure that everything was already written, and all is left for us (programmers) to do is plug-and-play (find the right example - alter it - use it).

Shouldn't we strive for more extensive reuse? Isn't it a key factor in enhancing our productivity? is it always so crucial to learn the example (rather than trial-and-error or copy-paste-test-alter until it works)?

Isn't it what's the internet is all about? what am i missing?

12 months ago • Delete

- □ Within the following weeks we received 134 comments of 67 unique users

Virtual Focus Group

- This LinkedIn discussion group serves as a virtual discussion group with interesting properties regarding:
  - Subject details
  - Reaching potential subjects
  - Discussion visibility

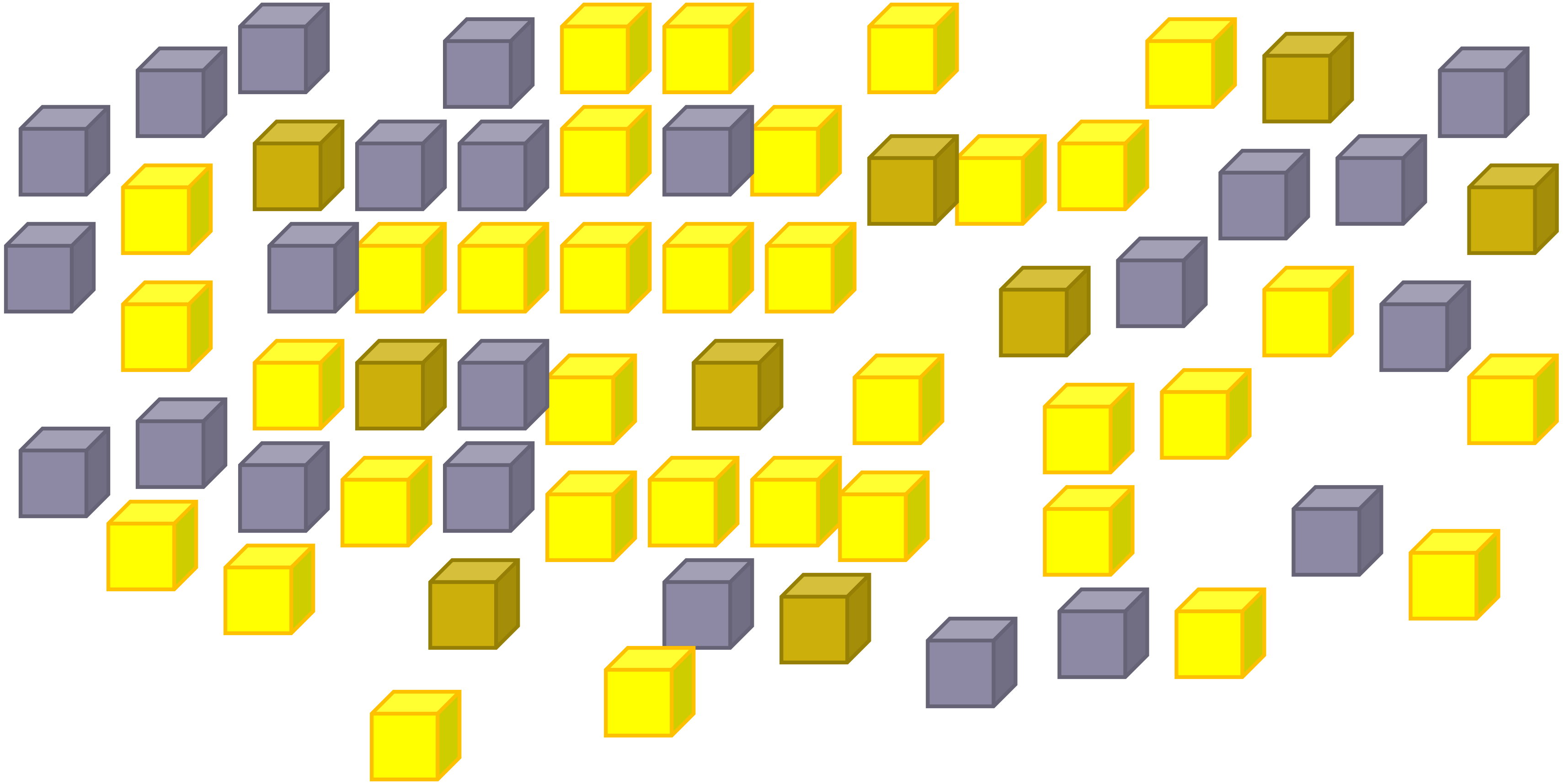
Virtual Focus Group Properties

- **Subject details are useful for researchers:**
  - Correlations between answers and background
  - Lowers the barrier of research participation
  - Subjects are traceable

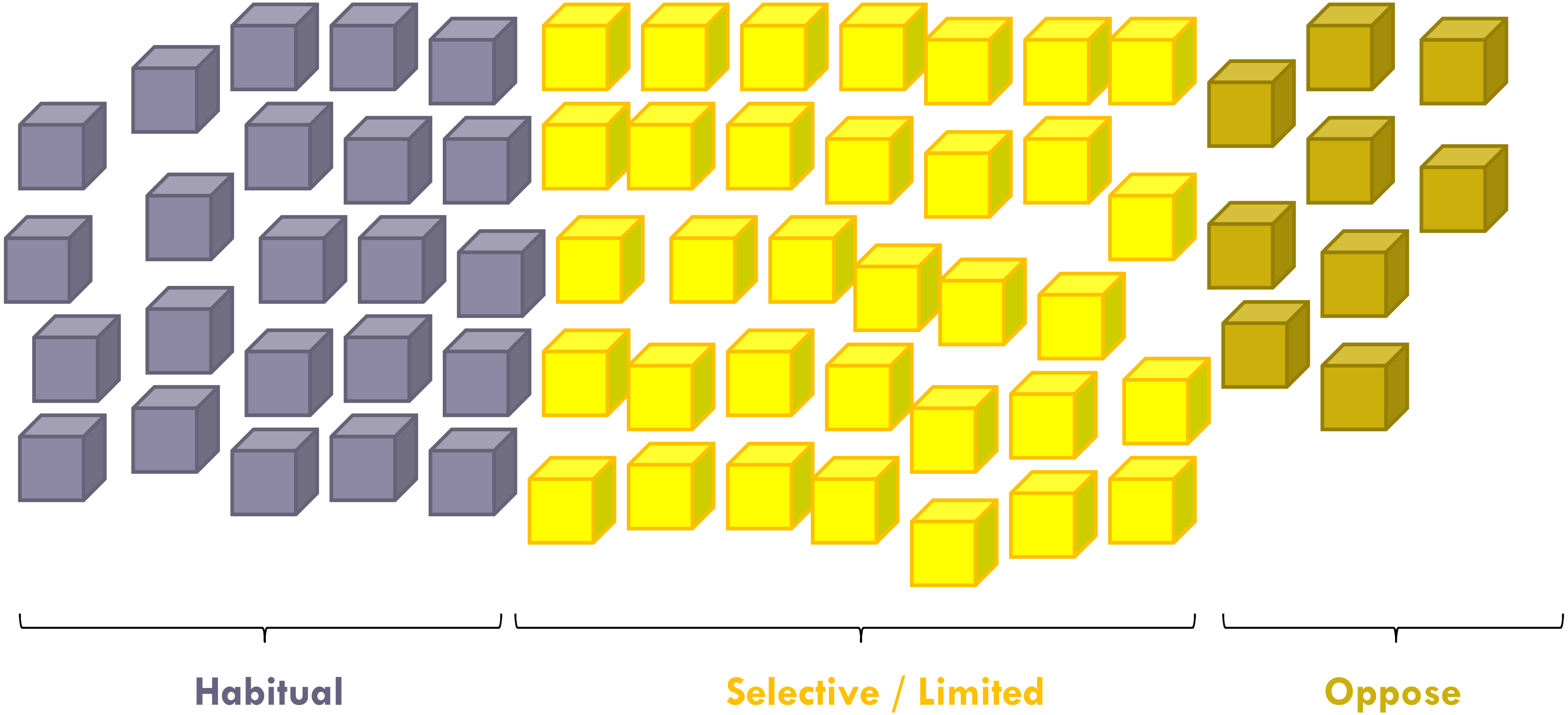
- **Reaching potential subjects**
  - Segmented community
  - Social platform
    - Encourages participation for increasing web presence
    - Viral distribution, hot topics promotion

- **Discussion visibility**
  - Participants can see each other answers
  - Other researcher could re-validate results
  - Potential future employers

Analysis



Vertical Analysis



Vertical Analysis

- Habitual example users (35%):
  - Using examples is more efficient
  - Examples promote pragmatic reuse
- Example antagonists (14%):
  - Examples' poor quality
  - Bad experience they had with example users
  - Lack of important properties: testability, understandability, and documentation.
- Using examples in limited context (51%):
  - Learning purposes*
  - Not using the example code*
  - Only for specific kind of development task*
  - Based on the example size and
  - Based on the example source

* Analyzed further

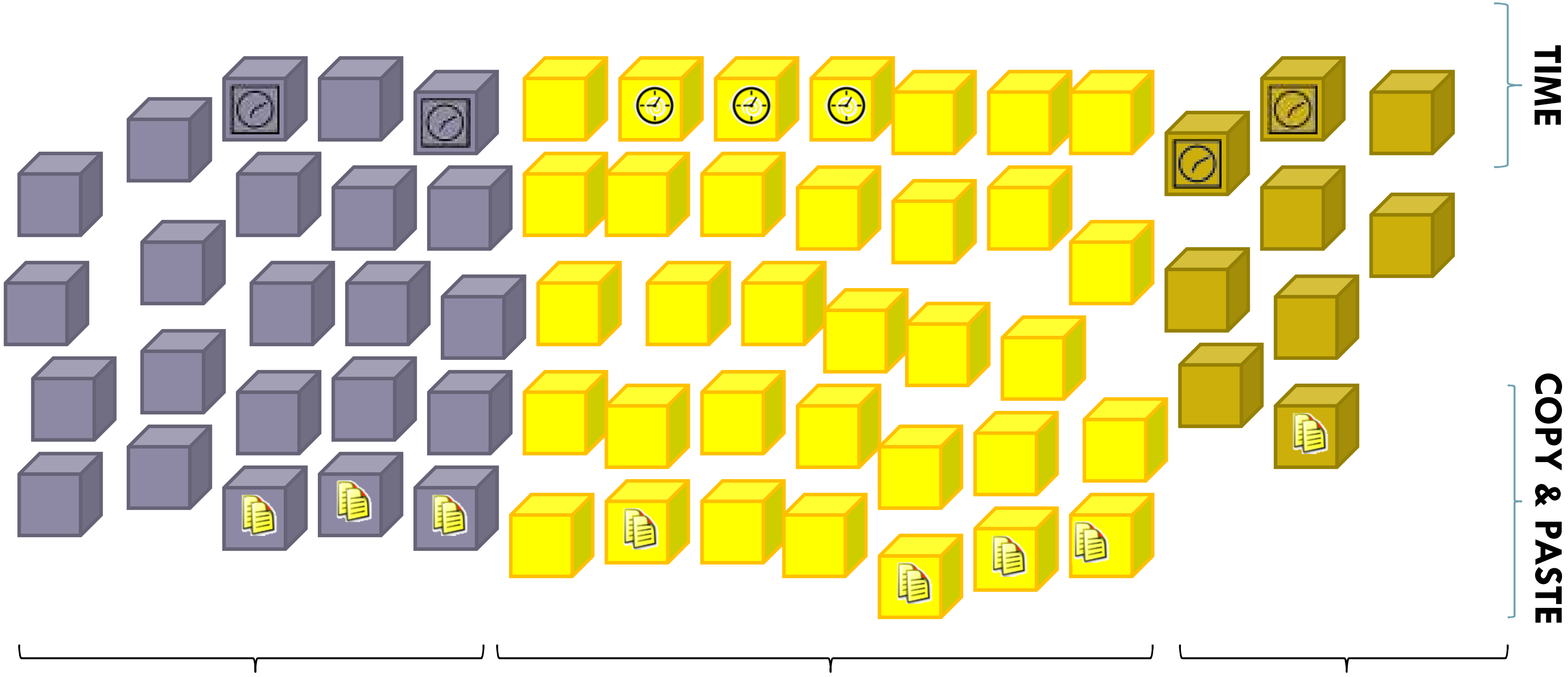
Limited Example Usage

- **Learning (65%)**
  - Environment/Mode
  - Centrality of examples in learning
  - Learning what? (API/technology/technique)
  - Interfere with learning

- **Development mode (Mental state)**
  - Hello world
  - Default vs. 'last resort‘
  - Level of familiarity
  - Debugging
  - Code comprehension
  - Periodic example reviewing

- **Using the example code (copy & paste)**
  - Yes (35%)
  - No (47%)
  - Selective (18%)

Horizontal Analysis



TIME

COPY & PASTE

Habitual

Selective / Limited

Oppose

Human and Social Aspects Dominate Developers Approach to Example Usage

- □ ***Conformance to organization goals***
- □ ***Personal development***
- □ ***Acknowledging example dexterity***

Human and Social Aspects Dominate Developers Approach to Example Usage

□ ***Analytical skills***

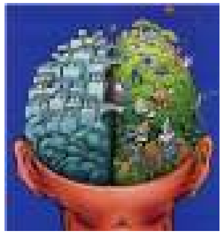
□ ***Perception of Role and Profession***

□ ***Ego***

□ ***Community identity***

□ ***Ownership***

□ ***Trust***



Conformance to organization goals

□ “Anyone that isn't looking for code for reuse is reinventing the wheel and wasting time"

□ “They [developers who use examples] do sometimes copy and paste a few lines of code. But that's only to avoid typing in something that's obvious“

□ “Yes, it may be reinventing the wheel or a little time consuming but I am not pressed for time"

Personal development

- In some cases developers' desire to **master the underlying technology** (so it could be added to their professional toolbox), might prevent them from taking a faster, more pragmatic approach

- Habitual example users, on the other hand, see the personal development exists in **taking the pragmatic approach time after time**, which makes them faster and more efficient programmers

Acknowledging example dexterity

- □ “I can spend less time writing my own free javascript plugins or just raw css tricks than I would trying to customize some picky little feature the client is looking for in the RAD tools“

- □ “There's nothing worse than cribbing a sample, spending a lot of time with it and then finding it doesn't actually work…“

- □ “when I am working with a programmer that productizes a sample, he/she is always tweaking and debugging the code until they end up rewriting […] It is more cost effective to write your own code, rather than cutting and pasting someone else's code“

- □ "The sample code usually needs to be fortified, but it saves me all the typing. Even if I take half baked code that is backed by a good idea or good design, I am better off starting from the sample then starting from scratch“

Ego

- "…But I never use 'examples' as a part of the real code“
- "I'd never use code samples in any commercial product“
- "…But never use them at work“
- "Hmmmm copy paste code is bad programming in fact that is not programming“
- "I have in fact seen direct cut and paste jobs that don't fill even the requirements or run code in production. Many times“
- "mindlessly copy-paste“
- "blindly copy-pasting“
- "cut-and-paste monkey"

Community identity

- Developers who oppose example usage do not consider themselves as **example *writers***
- "I personally look at forums more often than not because they are posts usually by real programmers or at least answered by real programmers. I also answer questions on forums so that karma (bad luck) if you will not catch up to me. I rather enjoy actually answering questions about things I know".
- "I equate some of these statements similarly to saying... I don't listen to others' opinions because they are often wrong"
- “Maybe it's my bad experience, but I've worked with as many bad coders per capita as I find on the internet as well”

Ownership and trust

- Some developers prefer to write the code by themselves and take responsibility for it than to trust others, and maybe lose control over their code

- "the websites I **trust** are msdn.com, jquery.com, codeproject“

- "…I grab a ton of code from old projects. But it is generally **my code** (or my team's code)“

- Taking ownership ‘rituals’:
  - Understanding
  - Testing
  - Convert the example style
  - Commenting
  - Going line by line

Role definition

- "A job of a programmer is to program, not play with Lego blocks where every block fits everywhere (nearly)“
- “the ultimate goal should be to ‘code’. That means Code most of the code yourself and figure it out....push yourself”
- "The problem I have is leads saying always code it from scratch, even if you're going to essentially code 70% of it the same and add custom code to it“
- "Some people will try to justify their existence by thinking that [they] have to code every line of every module they are working on, but that is just silly and a waste of time"

Profession definition

- A **creation** process is often considered nobler than a **reformation** process
- Some even consider computer programming to be an **art**
- The same way that an architect got more prestige than a handyman, so the classic programmers wish not to be downgraded
- "at the end of the day, we are all merely plumbers...“ [1]

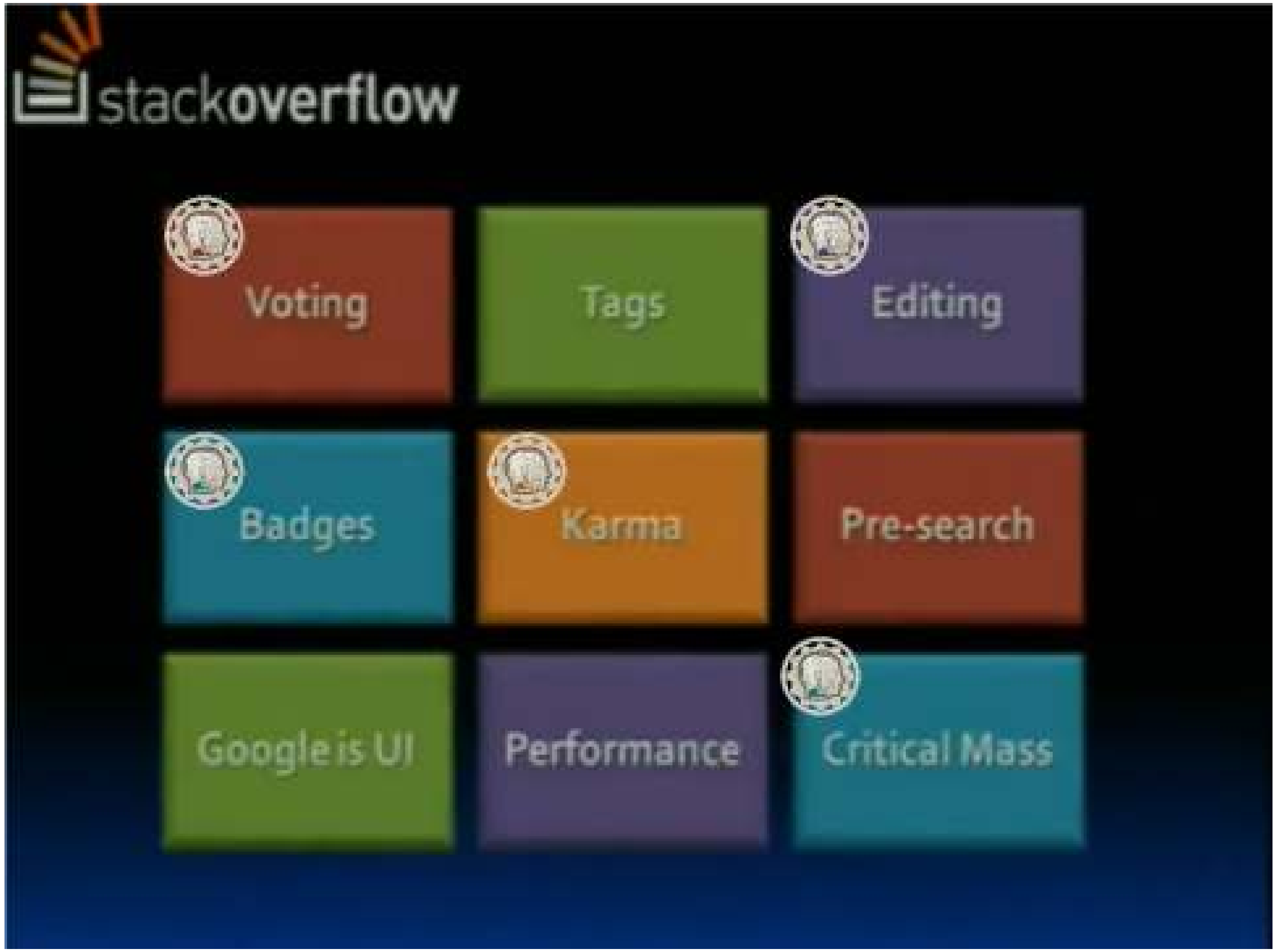
Analytical skills

- "Copy-Paste [is] not advised as it will never help a programmer to improve, but it will help in improving the analytical skills"

- "I'm describing myself as a Problem Solver as opposed to a Solution Repository! I may agree that the reverse would produce better experts, but I think, with less creativity... "

- Z. Obrenovic, D. Gasevic, and A. Eliëns. **Stimulating creativity through opportunistic software development.** *IEEE Softw.*, 25(6):64–70, 2008.

Addressing human and social concerns in Stack Overflow



Voting
Tags
Editing

Badges
Karma
Pre-search

Google is UI
Performance
Critical Mass

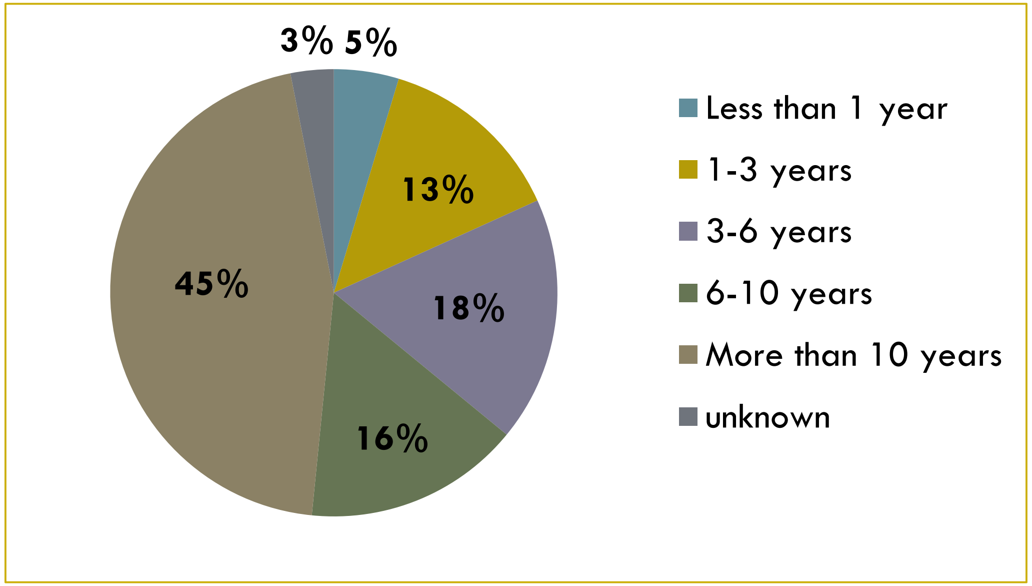
Why developers use examples?

Diversity of developers' motivations for example usage

Online Survey

- 311 survey submissions
- 192 answered the question "For what purposes do you use examples? Could you think of other purposes for using examples?"
- Most participants were experienced developers that use examples frequently
- We identified 3 axes that affect example usage motivation:
  - Properties of the task
  - The development activity
  - Software engineering considerations

Frequency	*No. of participants*
All the time, every few minutes	24
Once an hour	34
Once a day	69
Once a week	28
Less than once a week	16
Other	21



Classifying developers motivations for example usage

Task property	*No. of participants*
Unfamiliar	50
Specific	50
Complicated	16
Common	7
Hello world	5

Example usage by task property

Aspect	*No. of participants*
Reuse	32
Speed	28
Quality	8
Pragmatic	7

Software engineering considerations of example usage

Activity	*No. of participants*
Learning	57
Implementation	54
Problem solving	33
Design	33
Comprehension	32
Self improvement	6

Example usage by development activity

Example usage by task property

- □ **Unfamiliar, new** "[I use examples] mostly to get an initial familiarity with a broad API I don't know yet. Given 300 functions in a flat API, it's sometimes hard to focus on the main/important ones. A good example helps identify those, and lets you save the rest for later study"

- □ **Specific** Among the 49 examples of the specific tasks, we find language specific tasks, framework specific tasks and operation specific tasks.

- □ **Complicated** "Examples are these small bites of code you will never remember in full. For example usage of HTTP request in PHP or usage of any module of any CMS system"

- □ **Common** "[I use example for] implementing something general that 'someone must have implemented already'"

- □ **Hello world** "[it is] hard to remember [when was the] last time I started a project from scratch".

Example usage by development activity

□ **Learning** "to solve a problem I have" vs. "to learn how to solve a problem I have"

□ **Implementation** "i use them when i write new code".

□ **Problem solving** "I use them for two reasons, to see how other people solve some kind of new problem that arises in front of me, or to compare other ways of solving problems I already had"

□ **Design** "Examples of other developers gives different point of view on problem I'm working on .Also you can learn other techniques of development. This helps you to create your own style - by filtering mostly interesting ideas and techniques .Examples also give you opportunity to get more experience - you see other projects, other problems, other strategy for design and different solutions of same problem".

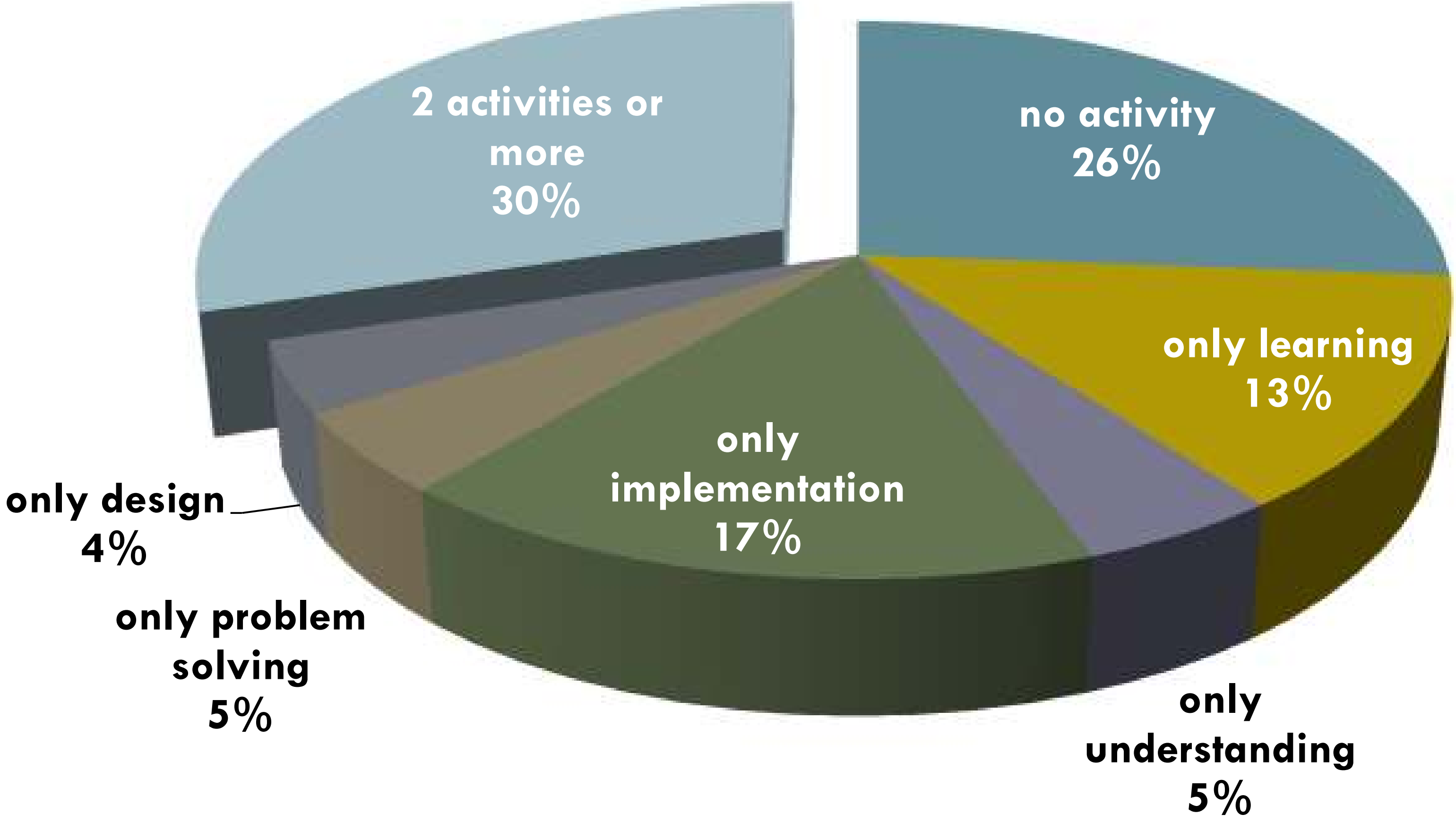
□ **Comprehension** "I use examples to try and understand the overall design of a system and build a mental map between expressions I do not understand to expressions I do understand".

□ **Self improvement** "Mostly learning, and improving my problem solving mentality. Examples can provide different perspectives than those you already have, so can give you additional insight into your problem space and your solution space".

Software engineering aspects of example usage

- **Reuse** “avoiding reinventing the wheel”.
- **Speed** “to speed up coding, to spend less time reading and understanding documentation, and copy pasting existing code to see if it works”.
- **Quality** “Use well proven code”.
- **Pragmatic** “to get things done”.

Using examples for multiple purposes



Implications

- Developers may not be attentive to using examples in multiple contexts

- Example usage is not acknowledged as a fundamental software activity

- Provides code search with context, and motivate further advancements:
  - Bookmarking mechanism
  - Catalogs for common tasks
  - Code search for design decisions

Developers attentiveness to example usage

Example Attentiveness Observed

- Phase I:
  - 2 teams in 2 large world wide software companies
  - 1 month, 14 sessions of 2-3 hours each
  - Observed 10 developers

- In this talk we focus on one session (during phase I), and we interpret it from the perspective of example attentiveness

Developers Attentiveness Observed

- We revisited our observation reports and identified 3 types of lack of attentiveness:
  - Context dependant
  - Utilization
  - Scale

Addressing Attentiveness Issues

- Two alternatives:
  - Take a proactive approach to increase developers awareness and attentiveness
  - Build an ecosystem in which theses issues are already weaved-in

Focus group case study

- Discussing example usage with developers
- Part of the mechanism we establish to collect industry feedback
- 20 developers
- Agenda:
  - Research review
  - Discussion
  - Reflection questionnaires
- A follow up questionnaire (after 3 months)

Focus group case study

- Research Review
  - Potential **benefits** of the systematic use of examples
  - **Barriers** preventing example usage from being applied more extensively
  - Example related **techniques**

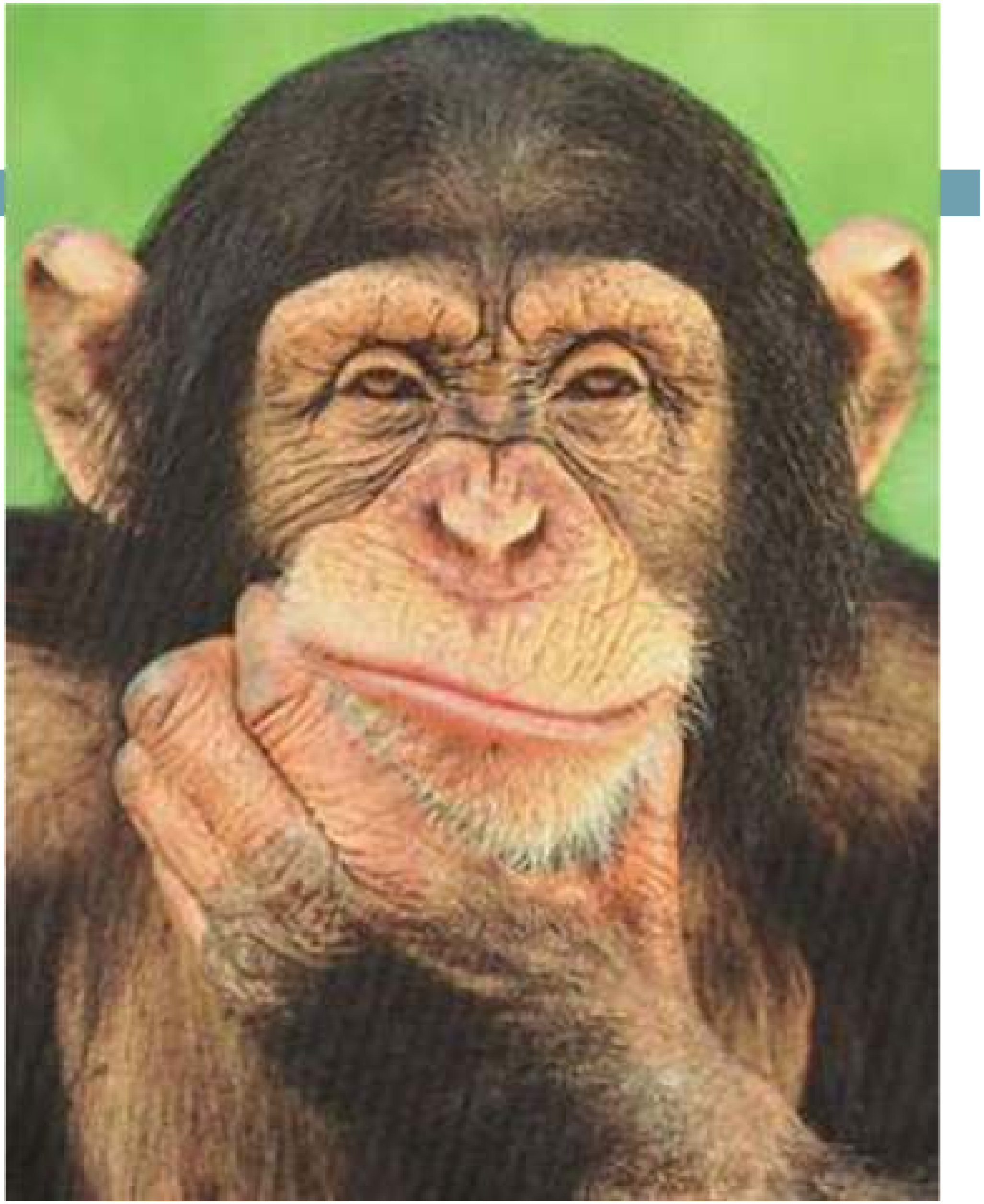
- In the reflection questionnaires we asked the participants:
  - Whether they **use** examples in their work
  - Whether they are **in favor** of using examples
  - Whether they were **influenced** by the session
  - How they estimate the session will affect their work **in the future**

- A follow up (after 3 months)
  - Did the talk (or completing the questionnaire) affect your awareness to reuse and example usage? If yes, in what way?
  - Have you incorporated any new techniques or practices in your work with respect to example usage? If yes - which?

Reflection

- A ‘**reflective practitioner**’ [Schön, 1983,1987] is someone who, at regular intervals, looks back at the work done, and the work process, and considers how they can be improved

- **Reflective practitioners** are not happy to carry on at the current standard, they want to improve



Summary of reflection questionnaires

- 7 of the 15 subjects stated that the session increased their level of **awareness** to **new opportunities** for example usage
  - Corresponding to the 3 types of awareness discussed earlier

- 4 of the 8 other subjects mentioned that following the session, they had some **new ideas about example usage** that they considered using in their work

New Opportunities

"Till now I used examples only 'as an inspiration'. Following the talk I would start using the **example code** as well.".

Selective example usage

- In another case study we identified additional types of selective example usage (though some of them are not related to attentiveness)

- The main variability factors are:
  - Reusing the example code or not
  - Developer's mental state (development mode)
  - Example size
  - Example source
  - Learning factor

- Indeed, the first 3 factors correspond to the 3 attentiveness aspects presented above: example context, scale and utilization.

Conceptualization and abstraction

"The talk helped [me] formalize some ideas I already had. I had been a learner-by-example for years but, as you know, putting a name to something makes it much more real and relevant."

4 of the 15 subjects addressed the **conceptualization** and **abstraction** of example usage in software development

Conceptualization and abstraction

- These quotes (and others) suggest addressing example usage as an **abstract fundamental software activity** and not merely as a **programming technique**

- The focus group participants consider examples in a wider context:
  - Developer productivity
  - Development speed
  - Code quality

- The focus group participants consider examples for:
  - Documentation purposes
  - Client training
  - Example-aware development process

Conceptualization and abstraction

- Implication on the nature of software development
  - "...at the end of the day, we are all merely plumbers...."

- To exploit to full potential of the example usage concept we propose to weave it into the **software engineering ecosystem**

- We demonstrate this idea using the **refactoring** concept

Refactoring Revisited

much more than a a fundamental activity

- □ **Refactoring** is ~~a disciplined technique~~ for restructuring an existing body of code, altering its internal structure without changing its external behavior [www.refactoring.com/]

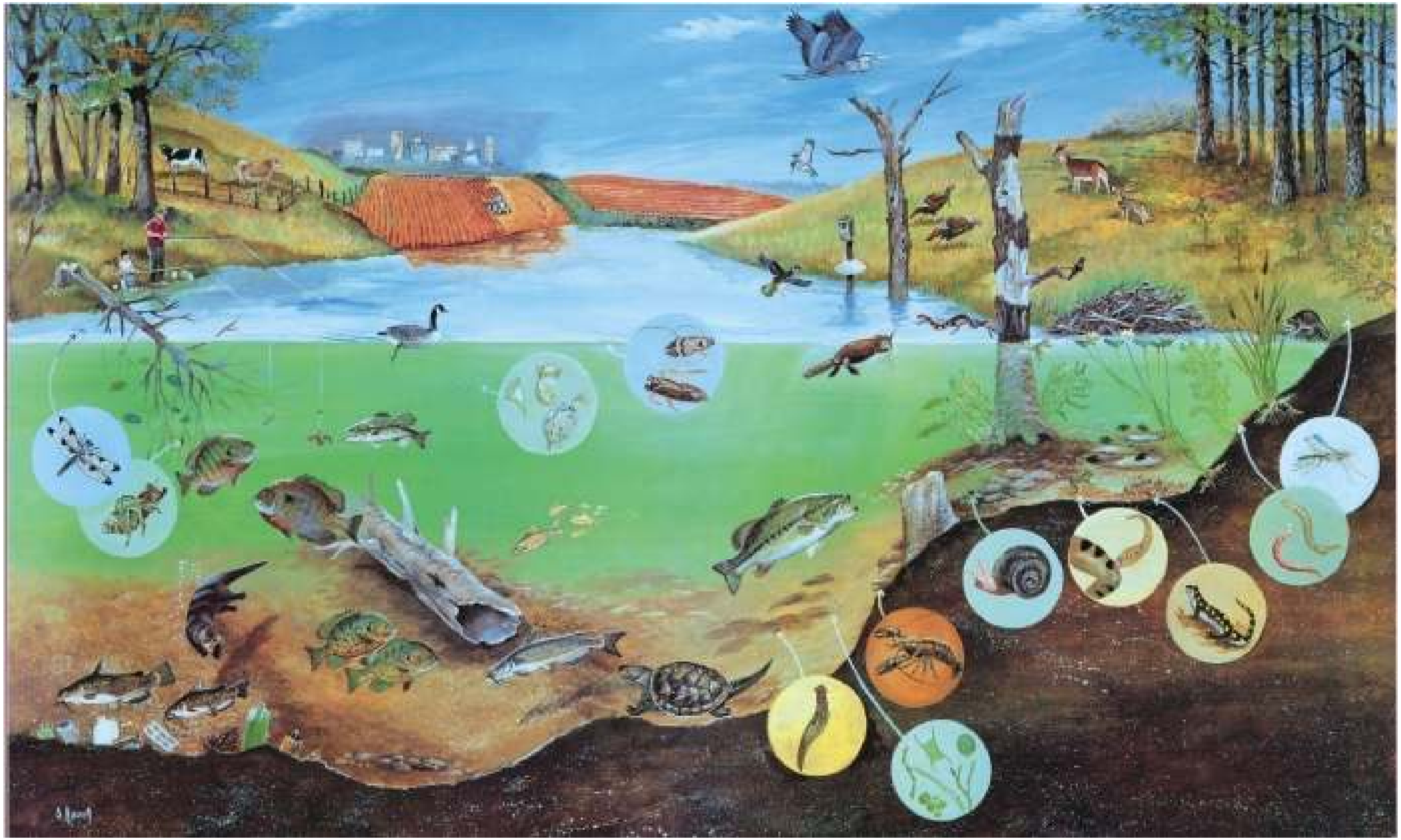
Appreciating *Refactoring*

The mere **identification of refactoring** promoted the following important processes:

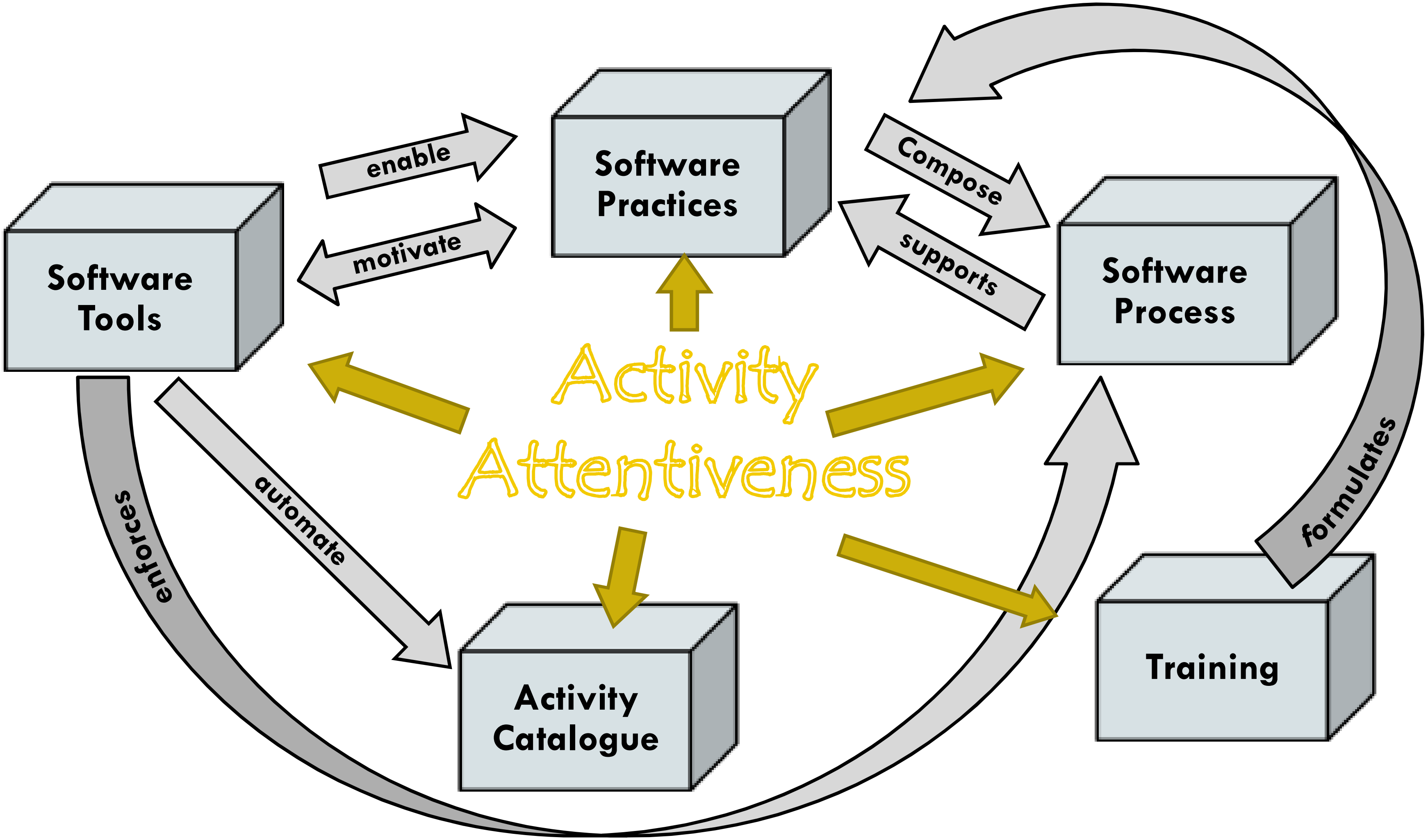
- □ Provided **name** and **definition** for the activity
- □ Laid the foundations for others to build a **catalogue**
- □ Enabled the development of **software tools**
- □ Promoted new **coding practices**
- □ Influenced the **development process**

These various aspects serve as an **ecosystem** that exploits the use of refactoring **systematically** and **methodically** to leverage its full potential and eliminate its **pitfalls** and **deficiencies**

Ecosystem



The Software Development Ecosystem



Example Embedding

- We define **Example Embedding** as the notion of using an already existing code fragment (the example) within a new context

- We (you!) should look for ways to
  - **Weave** example embedding into the software engineering ecosystem
  - Use examples more **systematically** more **extensively** and more **effectively** to exploit their full potential

- We believe that **productivity** benefits from using examples **habitually** and **correctly** in example supportive ecosystem

Summary

- □ Human and social aspects of example usage
- □ Developers motivations
- □ Example Attentiveness Observed
  - ◘ Context, Utilization, Scale
- □ Focus group case study
- □ Addressing diversity by weaving example attentiveness into the software engineering ecosystem

Thank You

ohadbr@gmail.com

Question: "isn't this research more suitable for social sciences / psychology / management?"

- Answer:
  - We investigate **software development**
  - We use **research methods**, which are not common in the computer science community

- We should ask ourselves – who is in charge for software engineering?
  - If it is computer science – it should not ignore important aspects of it