Freie Universität Berlin

Explaining Pair Programming Session Dynamics from Knowledge Gaps

Technical Paper presented at the 42nd Int'l. Conf. on Software Engineering (ICSE 2020)

Franz Zieris
zieris@inf.fu-berlin.de

Lutz Prechelt
prechelt@inf.fu-berlin.de

Practitioner expectations:

Better design and fewer defects in less time

Learning from each other and together

Understanding legacy parts of the software system

Pair Programming

Motivation

- Expectations in industry:
  Why pair-program?

Better design and fewer defects in less time

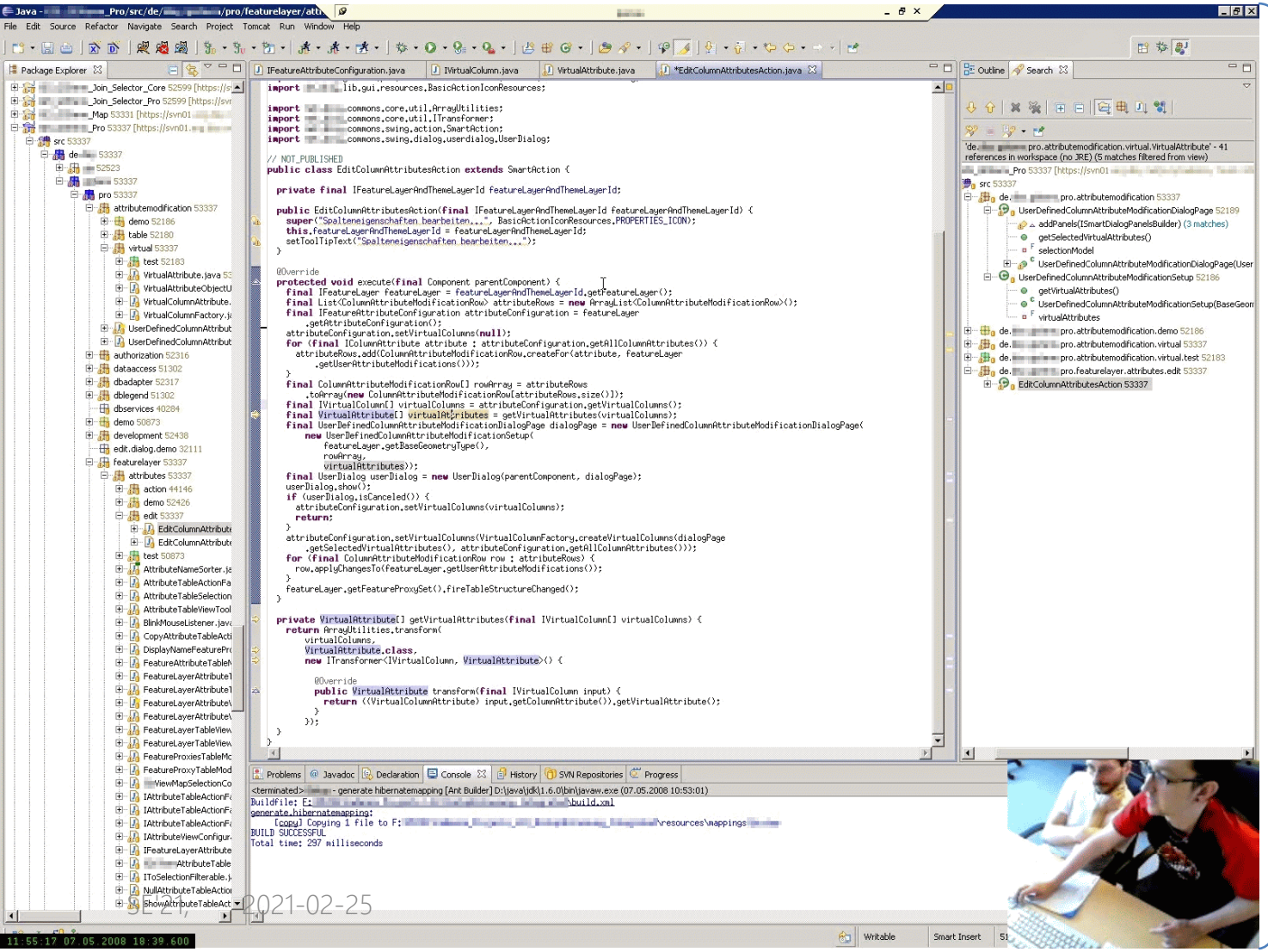
Learning from each other and together

Understanding legacy parts of the software system

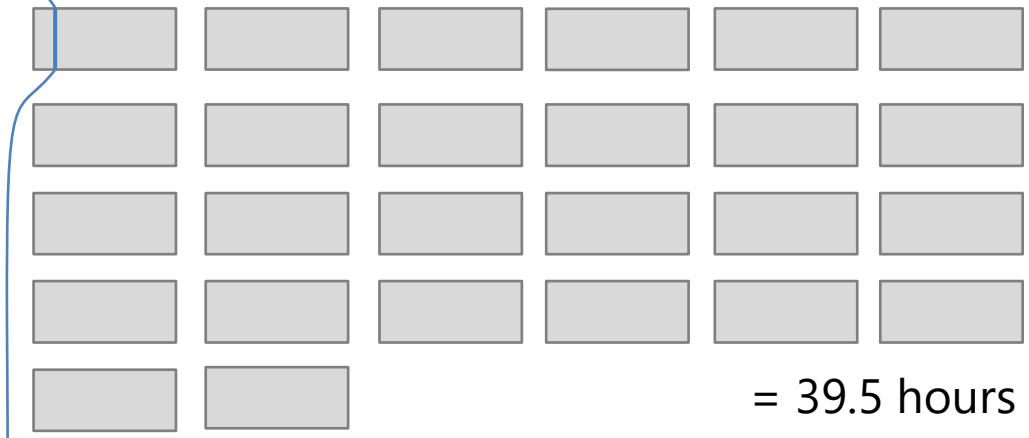
- Our Overall Research Goal
  Understand how industrial pair programming actually works
- Research Question

  What are the underlying mechanisms of **knowledge transfer** in pair programming?
- Intended Outcome
  - Advise practitioners
  - behavioral (anti-)patterns

Qualitative Data Analysis



- **Grounded Theory** approach
- Recorded **industrial** PP sessions (audio, webcam, screen)



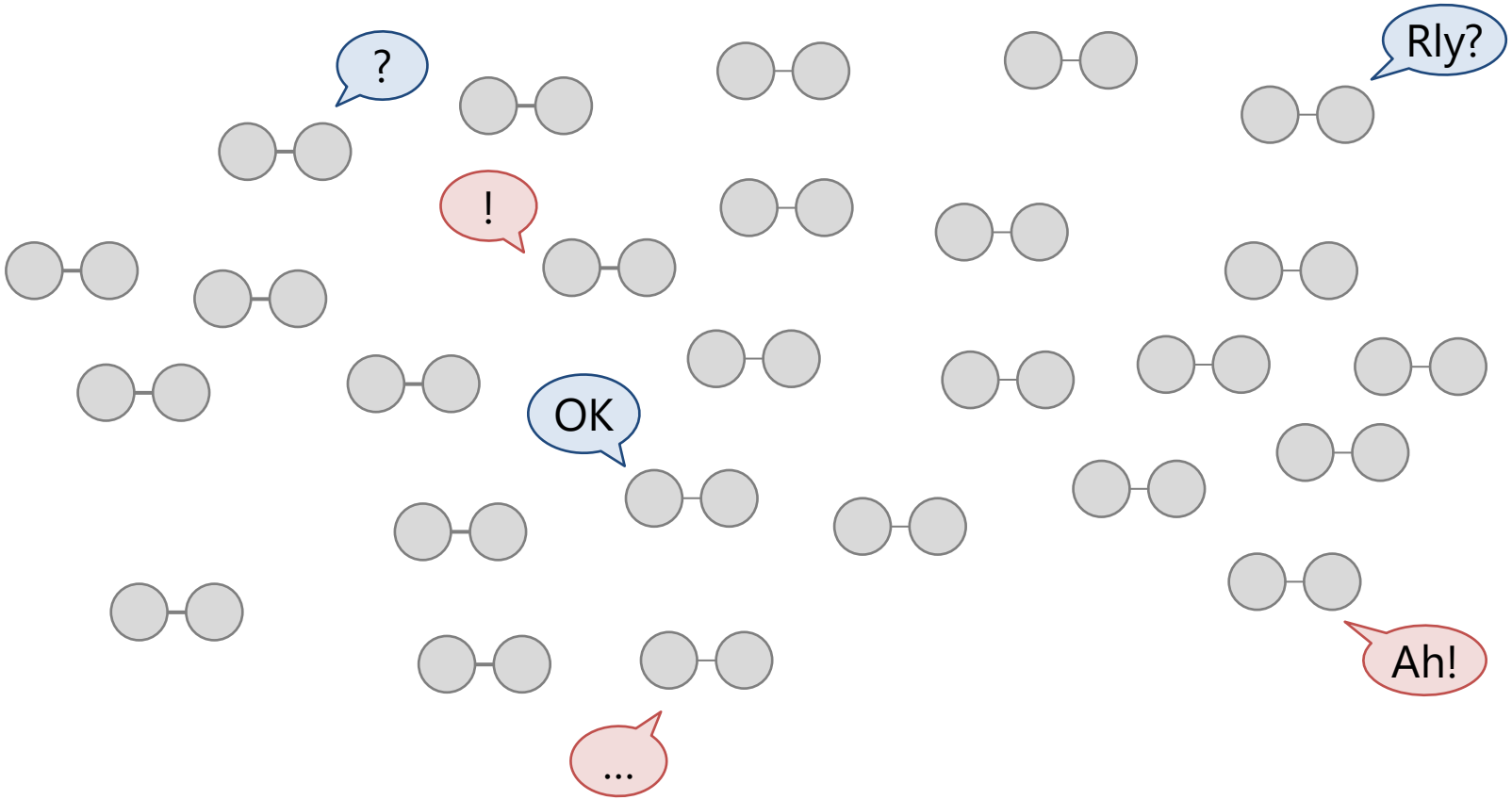
= 39.5 hours

Theoretical sampling: 26 sessions (9 companies, 16 pairs)

- from a total of 67 sessions

Qualitative Data Analysis

26 pairings of professional developers

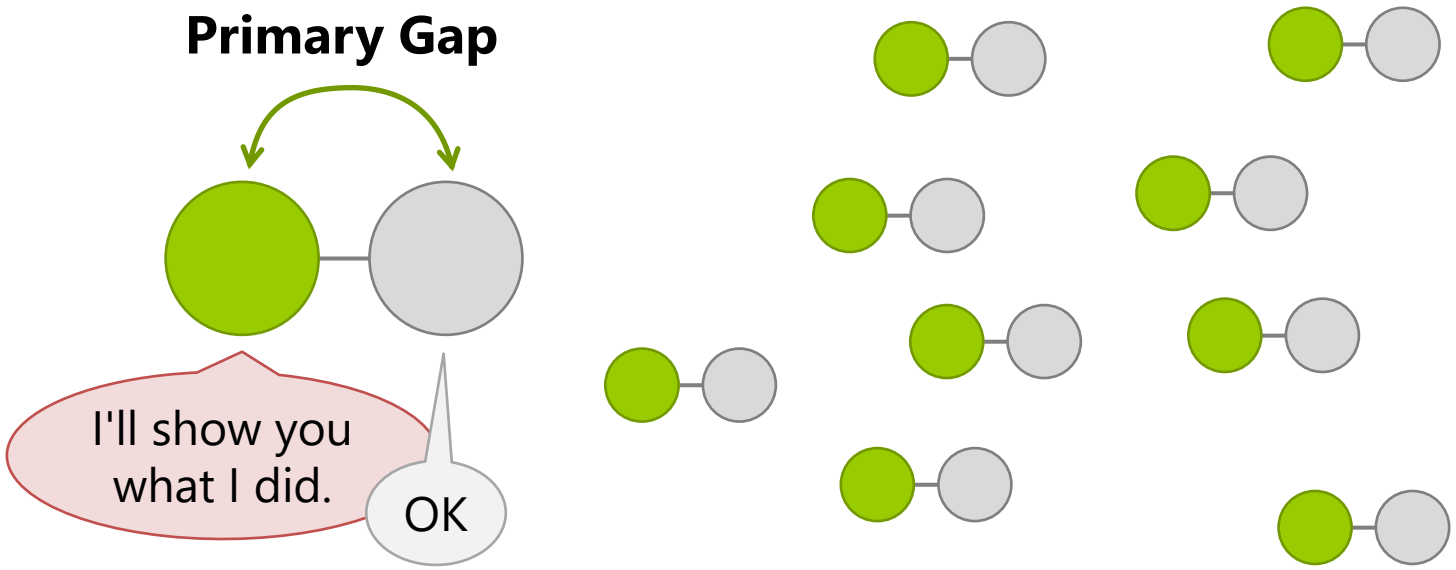


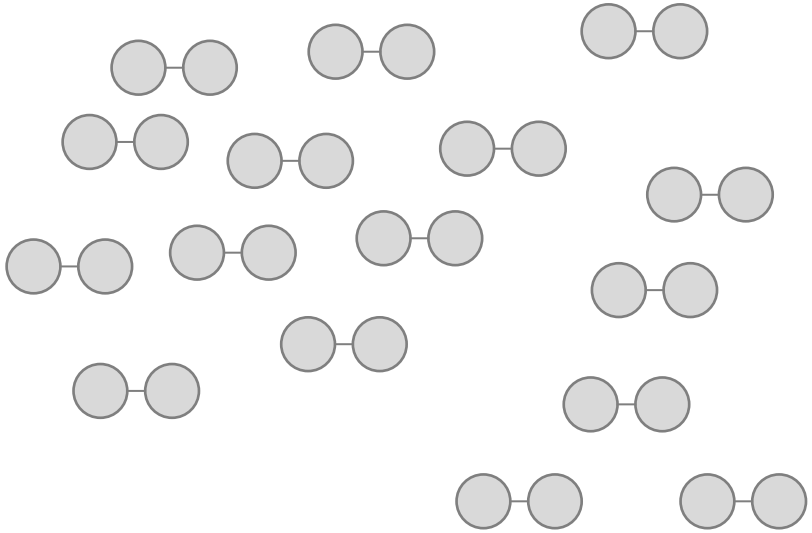
Analysis of pair programmers' dialog:

- What do they ask for? What do they explain?
- What do they (not) know? What do they learn?

Observation 1: The Primary Gap

One partner already worked on the task



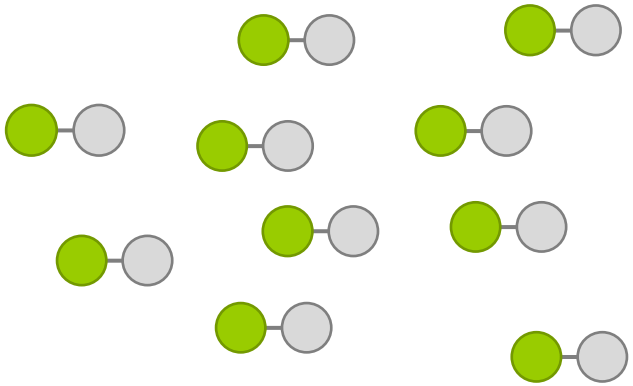


= task-relevant system knowledge

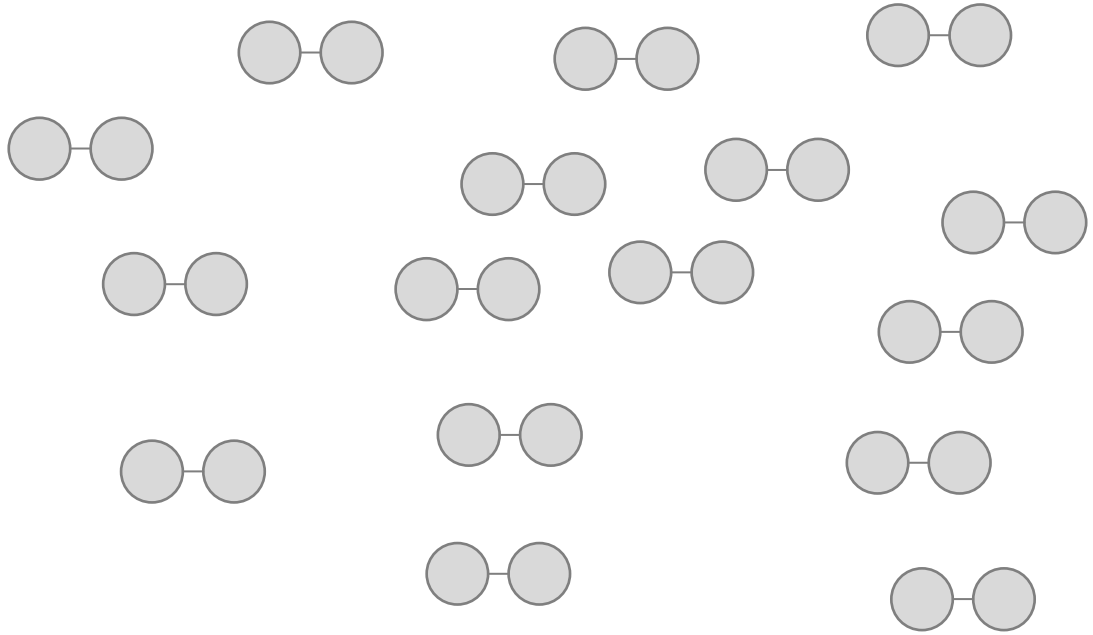
current state of implementation, classes, call hierarchies, defects, test/build setup, configuration state, ...

What about more homogenous pairs?

One partner already worked on the task



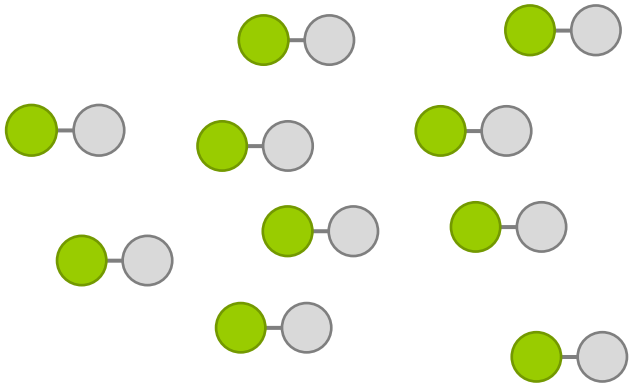
Both partners with similar prior involvement



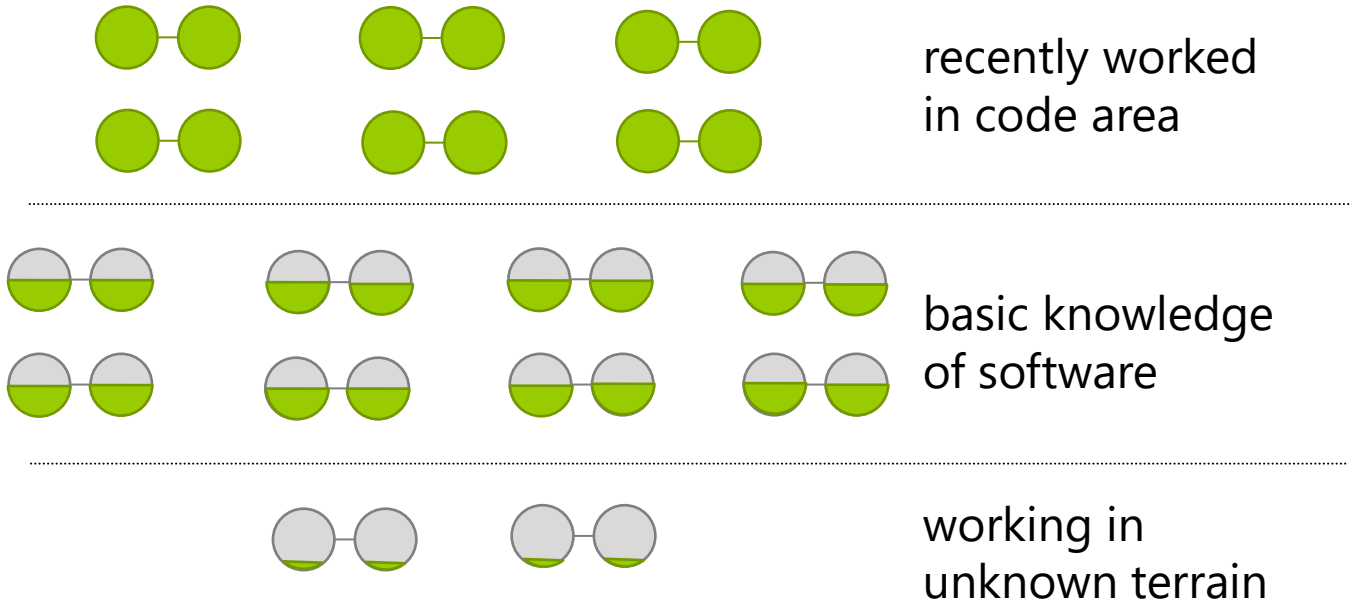
= task-relevant system knowledge

What about more homogenous pairs?

One partner already worked on the task

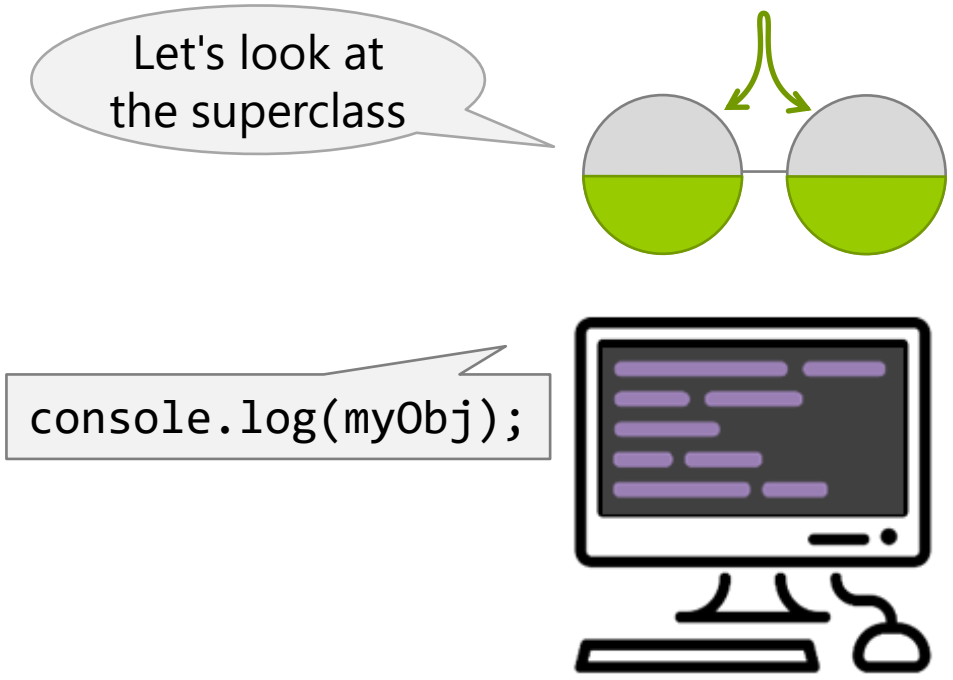


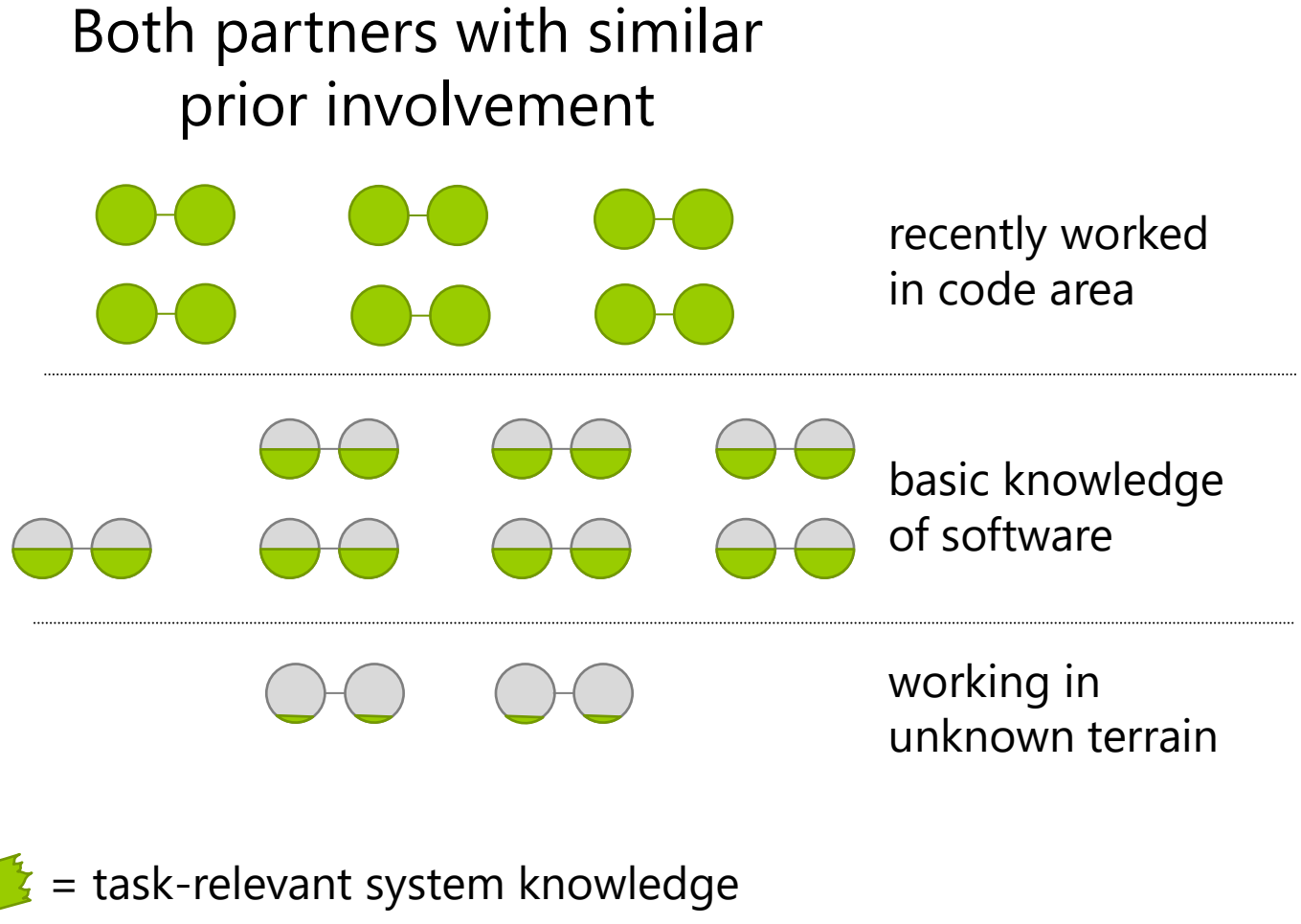
Both partners with similar prior involvement



= task-relevant system knowledge

Observation 2: The Secondary Gap

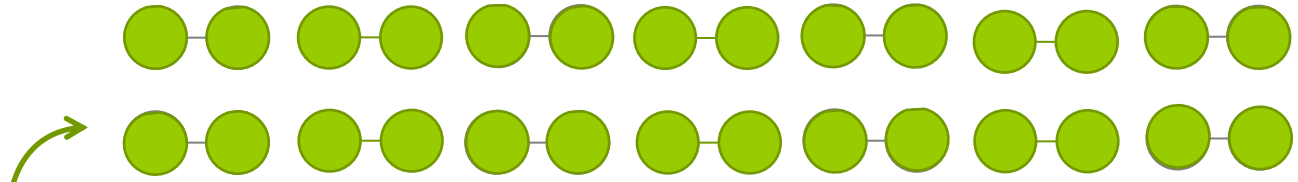




Observation 2: The Secondary Gap

Both partners with similar prior involvement

Acquiring System Knowledge together works for most pairs



Good system understanding

Medium system understanding

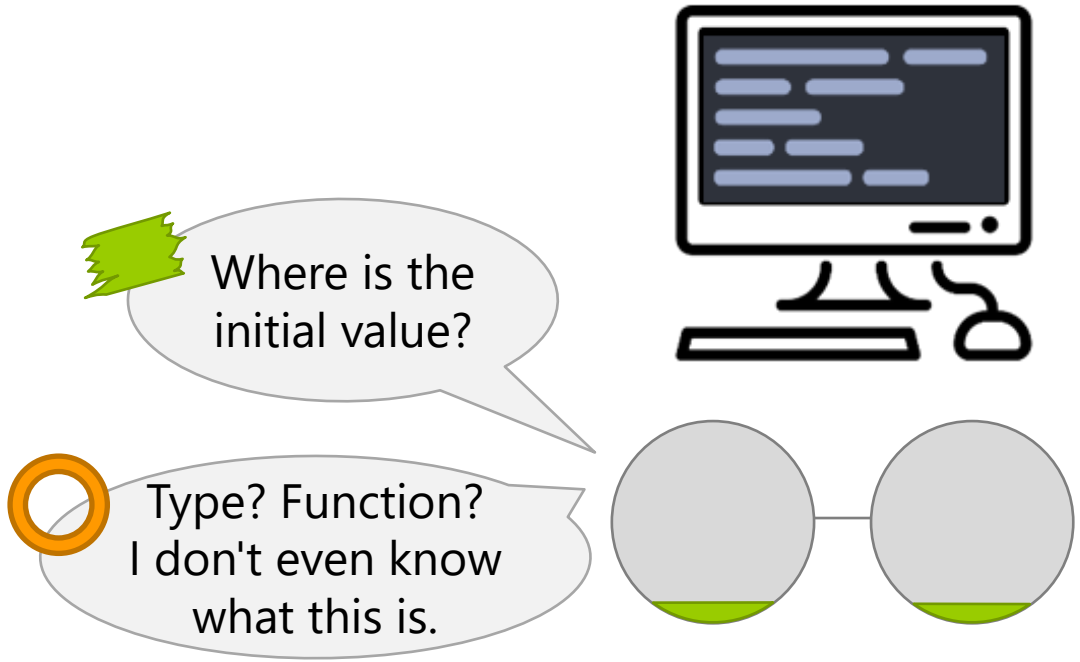
… but not for all

Litte system understanding

 = task-relevant system knowledge

A Different Kind of Knowledge

Task: implement test case



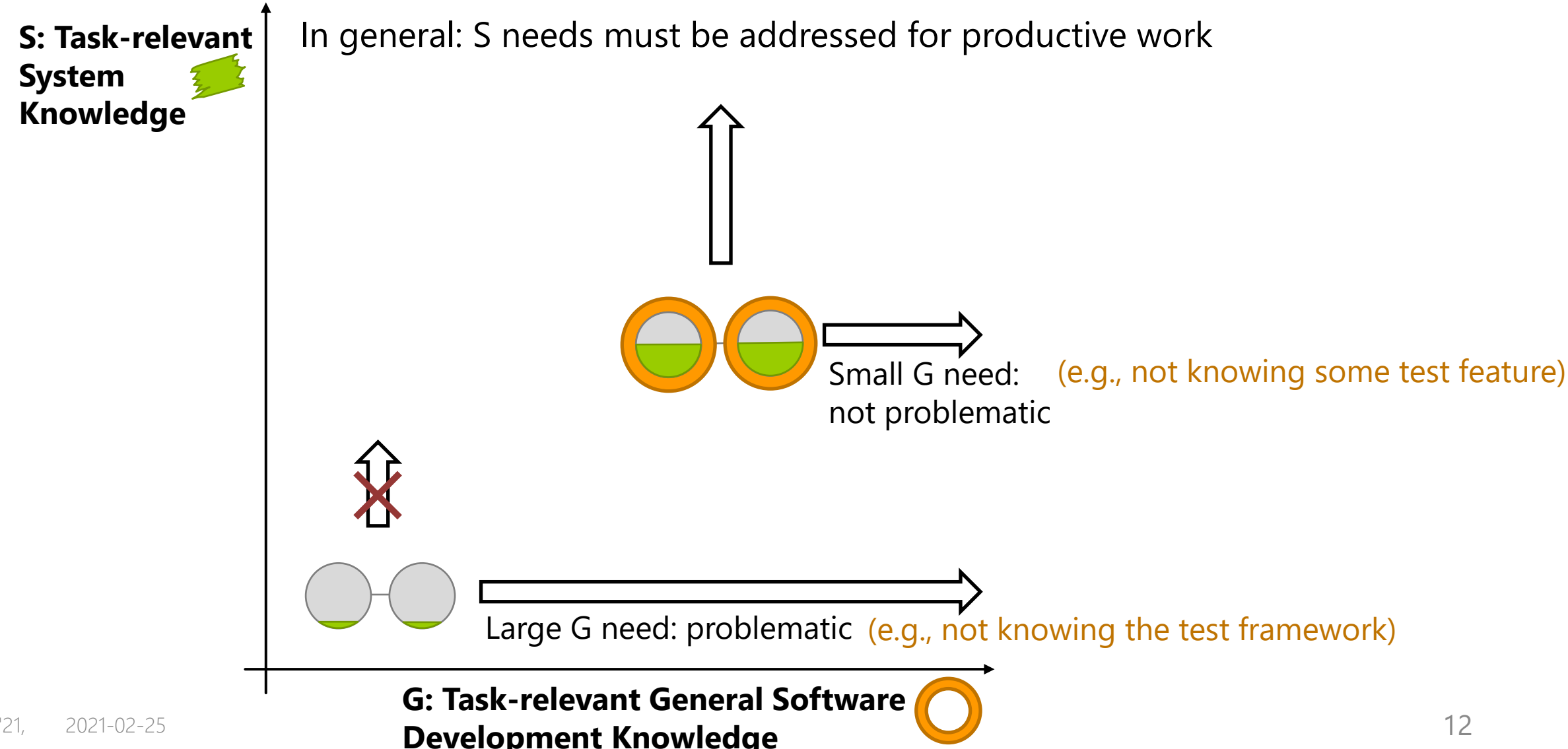
S knowledge (task-relevant system understanding)

Data structure holding the application state?
How to modify and read the state?
…

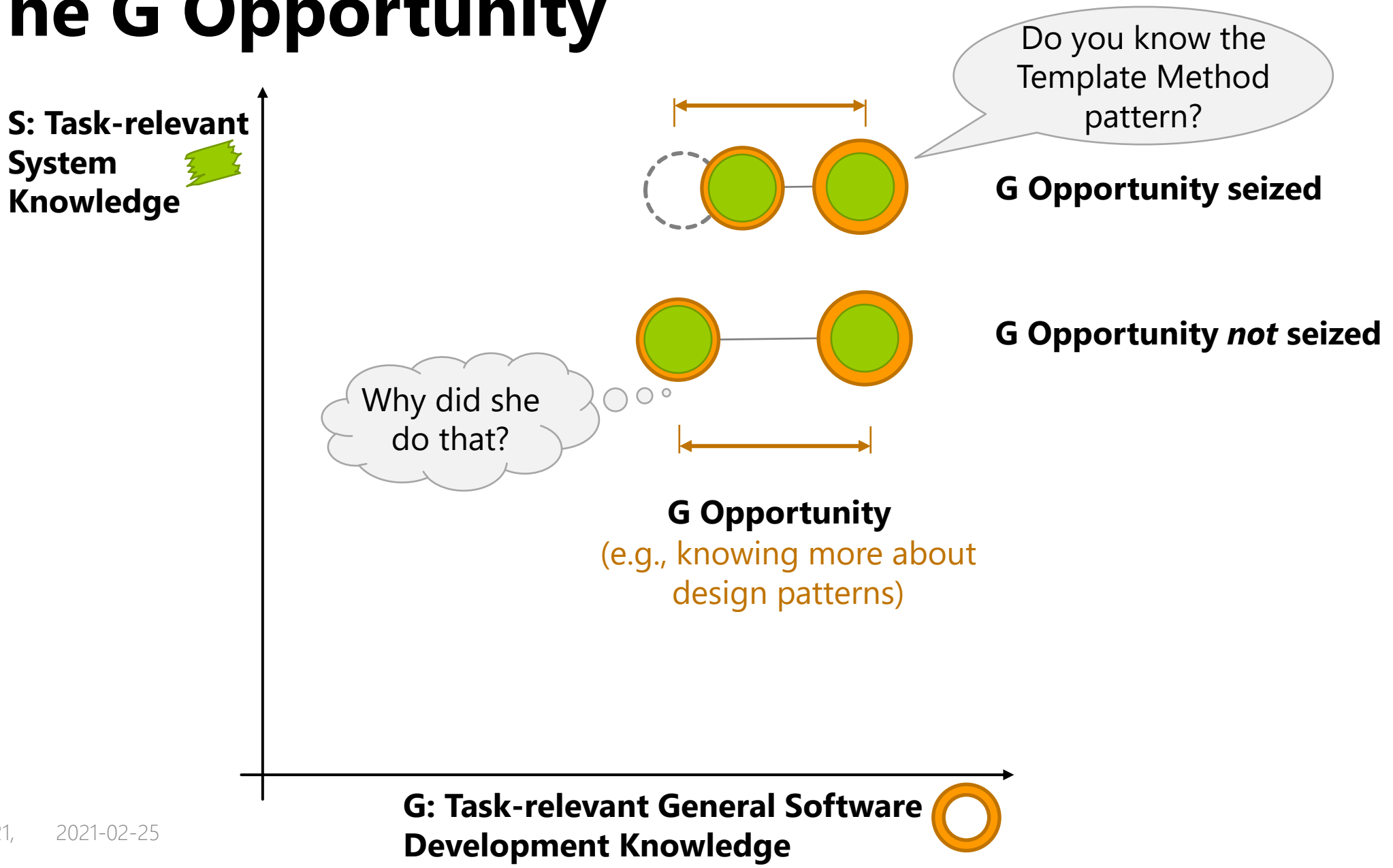
G knowledge (task-relevant general software development knowledge)

Syntax of programming language?
Higher-order functions?
Application framework?
Test framework?
…

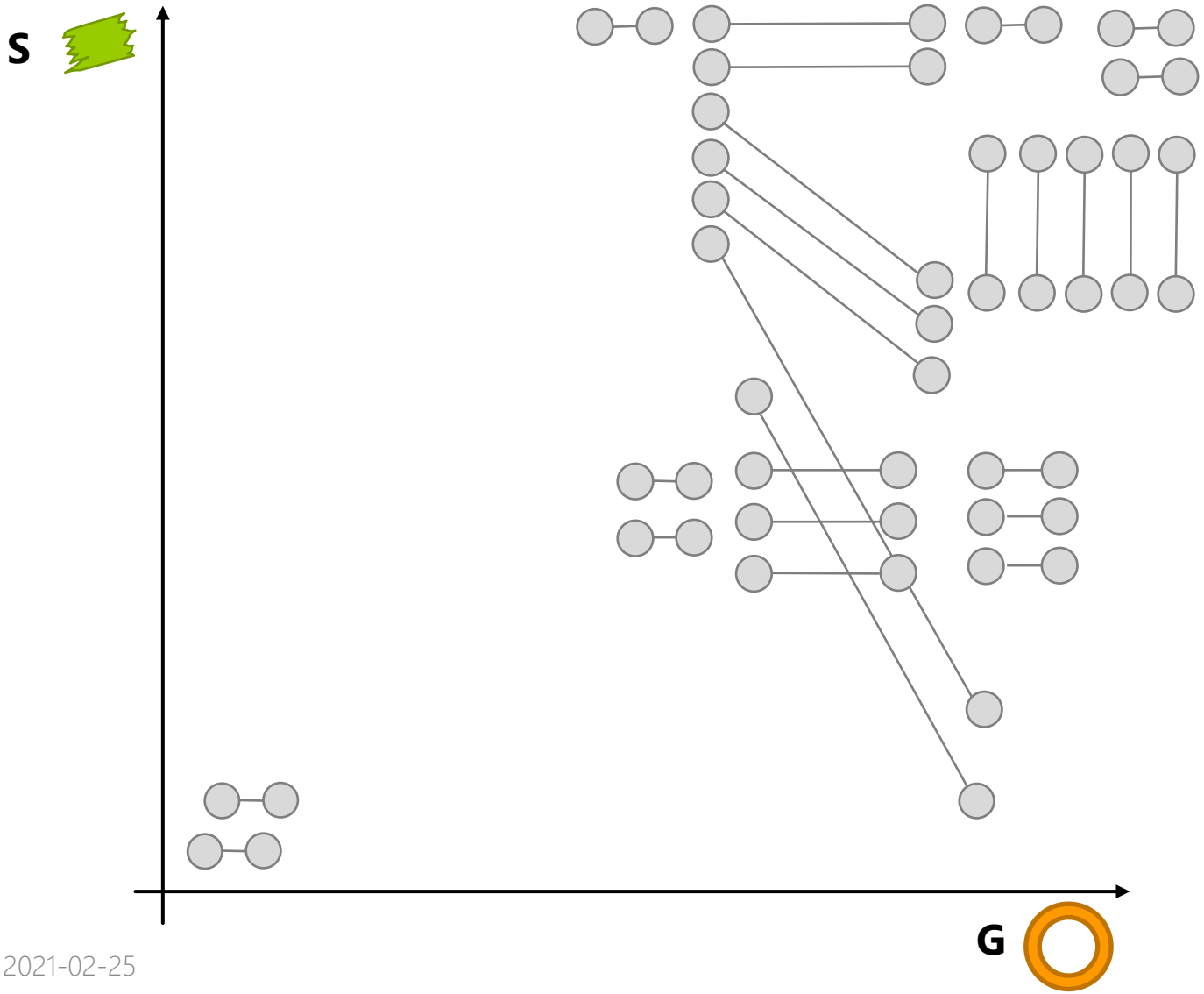
Roles of S and G knowledge



The G Opportunity



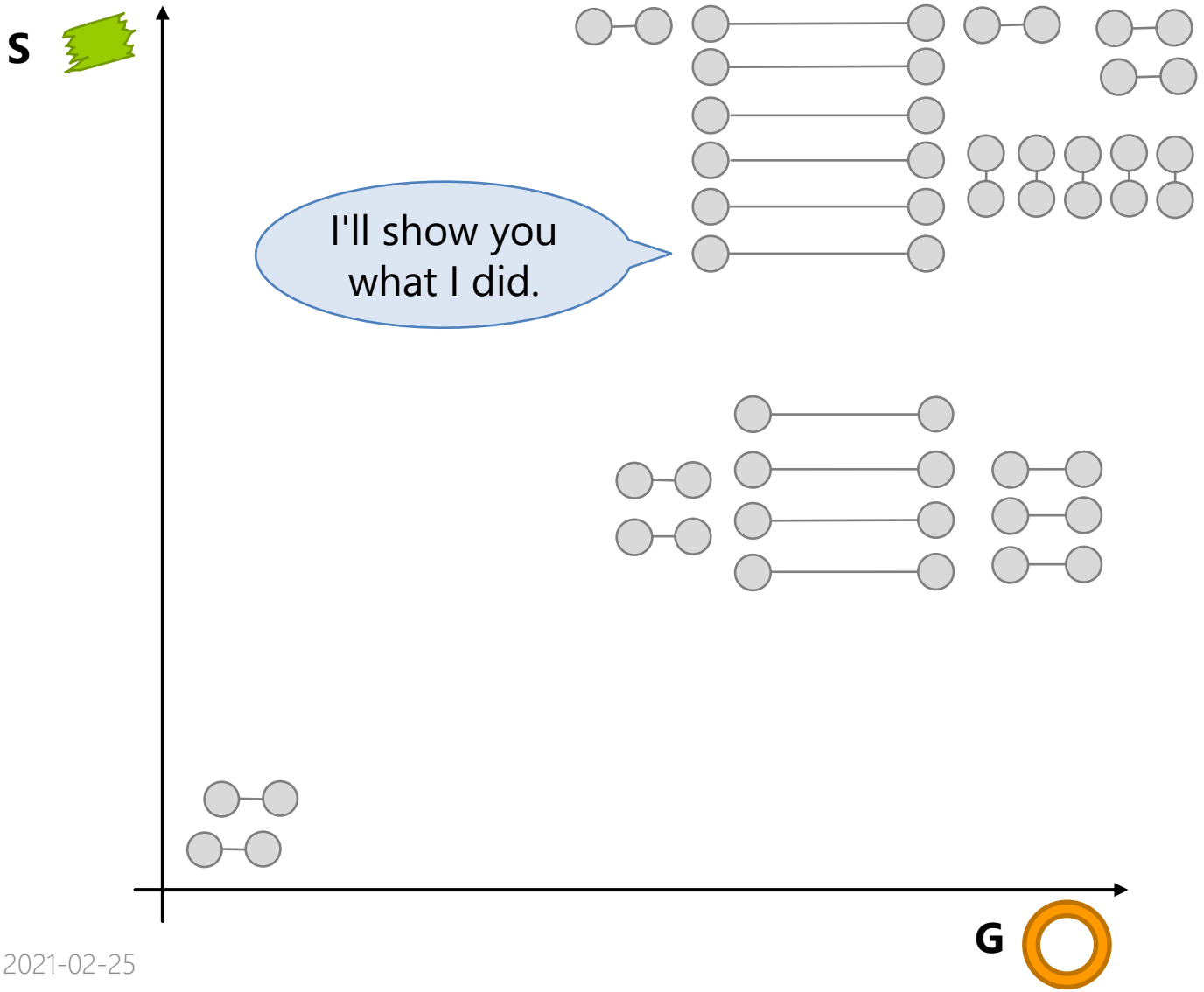
Overall Session Dynamic



Overall Dynamic

1. Close **Primary Gap**
2. Close **Secondary Gap**
3. Seize **G Opportunity**

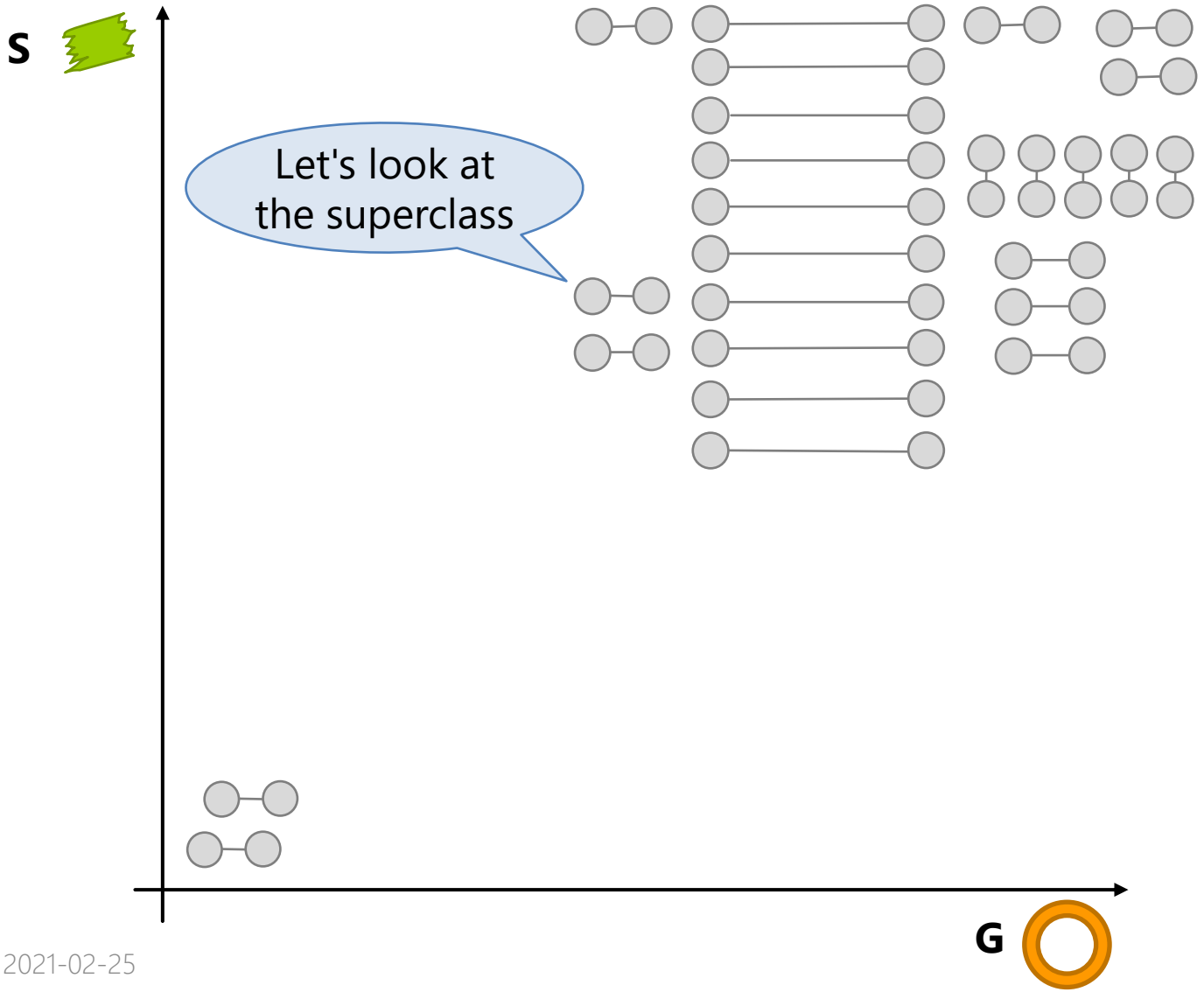
Overall Session Dynamic



Overall Dynamic

1. Close **Primary Gap**
2. Close **Secondary Gap**
3. Seize **G Opportunity**

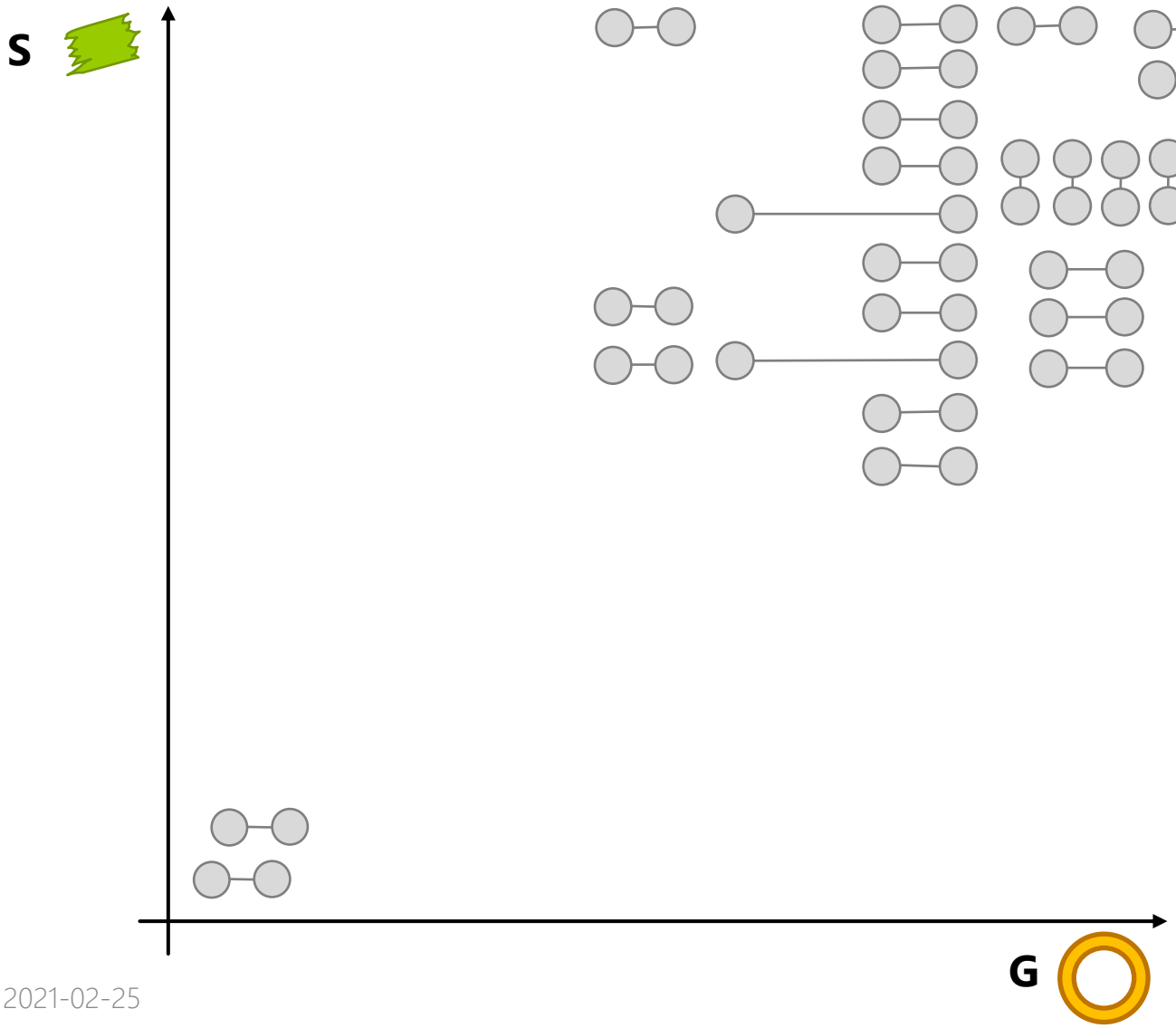
Overall Session Dynamic



Overall Dynamic

1. Close **Primary Gap**
2. Close **Secondary Gap**
3. Seize **G Opportunity**

Overall Session Dynamic

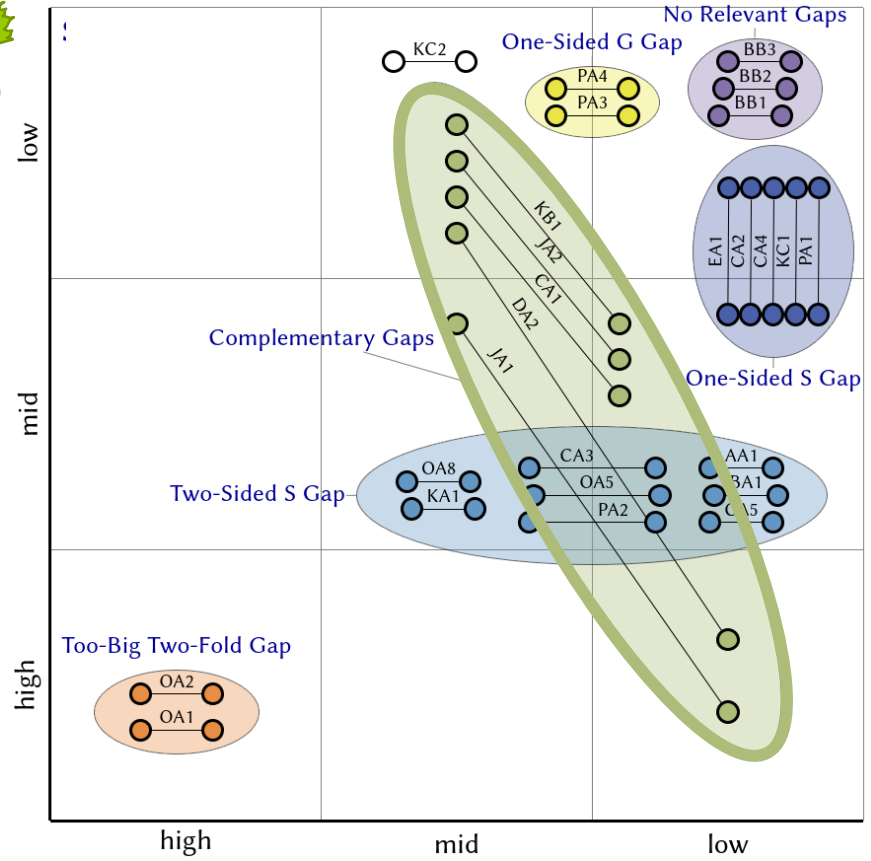


Overall Dynamic

1. Close **Primary Gap**
2. Close **Secondary Gap**
3. Seize **G Opportunity**

Summary

S: Task-relevant System Knowledge



G: Task-relevant General Software Development Knowledge

- Differences in S knowledge affect pairs more than in G knowledge
- What matters is **task-relevant** knowledge
  - ➔ different knowledge needs, different session dynamic
- Practical idea: consider team members' task-specific knowledge needs to *form pairs* and *tailor tasks*
  - Mutually satisfactory constellation: **Complementary Gaps**

Thank you!

Images

 https://web.archive.org/web/20080509191418/http://www.cenqua.com/pairon/

 Icon "Computer" by Denis Shumaylov from the Noun Project

 Icon "Bug" by Minh Do from the Noun Project

 Icon "design" by Adrien Coquet from the Noun Project

 Icon "knowledge" by Olivia from the Noun Project

 Icon "corner webs" by Kate Maldjian from the Noun Project

 Icon "Box" by No More Heroes from the Noun Project

 Photo by "Startup Stock Photos" used under CC0 license