On Continuous Threat Modeling of Cyber-physical Systems

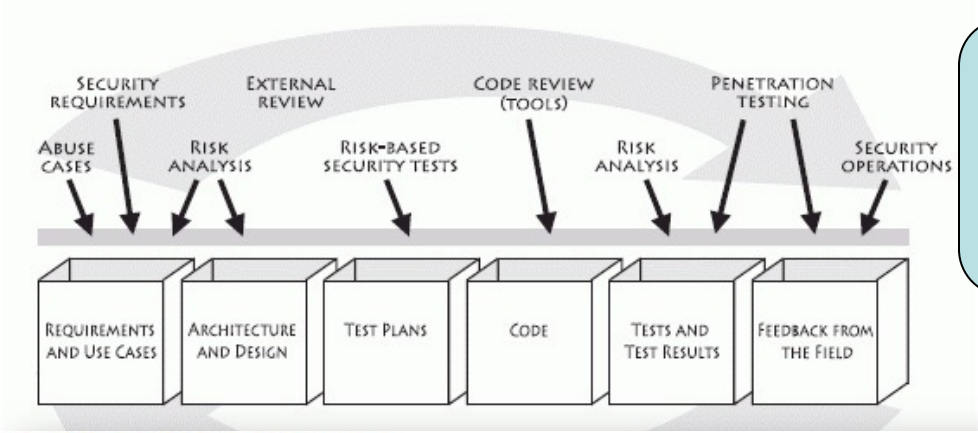
Dr. Lotfi ben Othmane

DevSecOps Days

Washington DC – December 16, 2021

IOWA STATE UNIVERSITY

Practices of Changing Secure Software

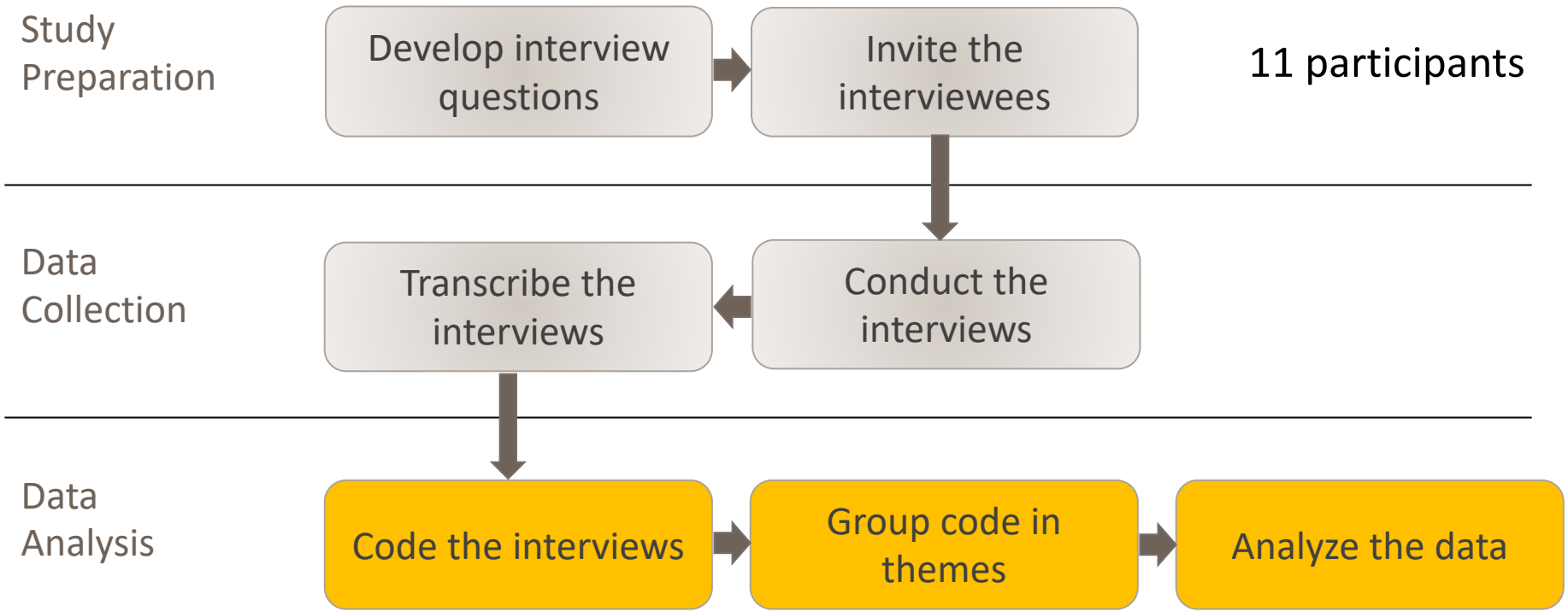



```
public class MyClass
{
public void Method_A()
    {  // Do Something
        FileIOPermission myPerm =
            new FileIOPermission(PermissionState.Read);

        myPerm.Demand();
        // Do Something
    }}
```

How do practitioners deal with changing secure software?

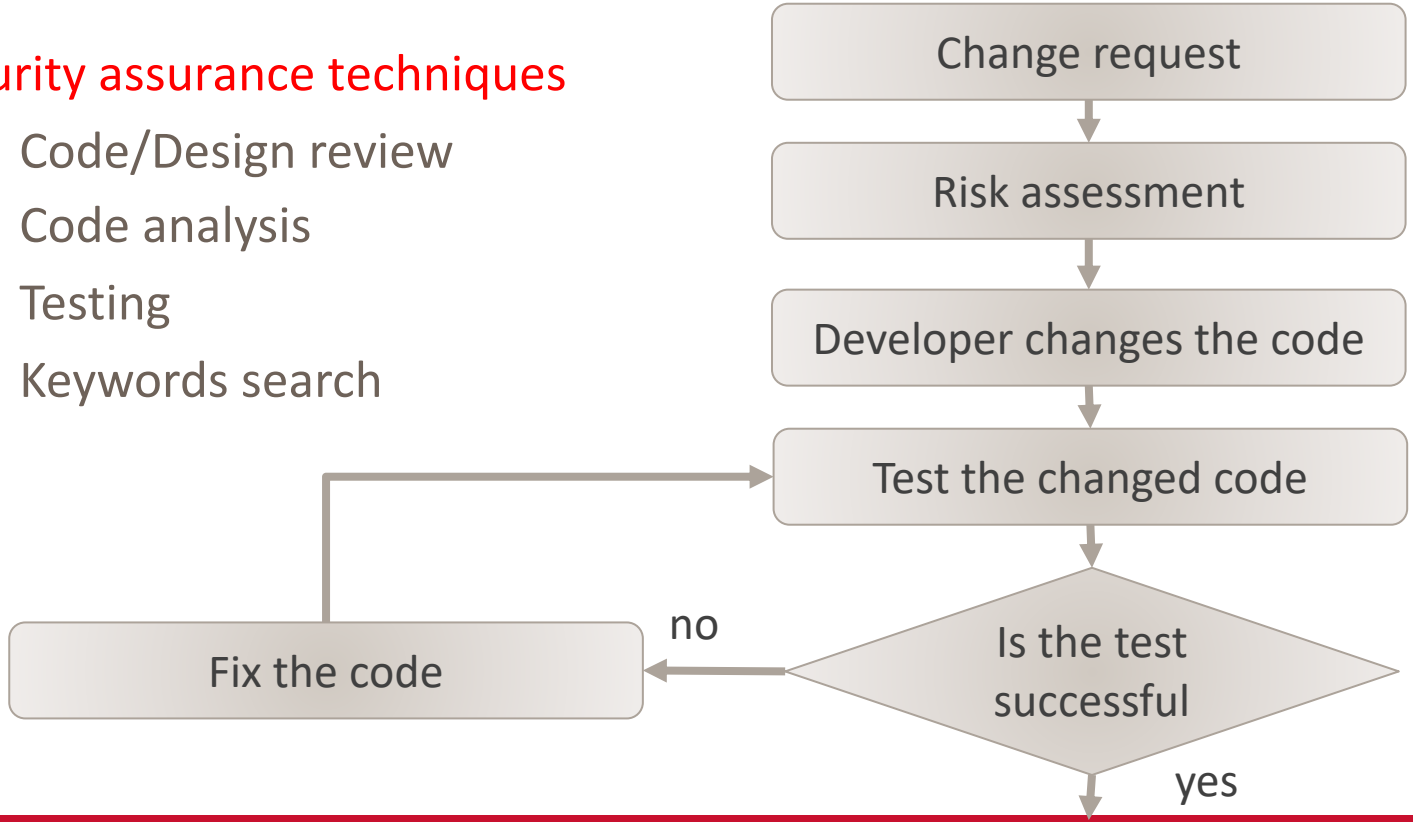
Empirical Study - Practices of Changing Secure Software



The process of Changing Secure Software

Security assurance techniques

1. Code/Design review
2. Code analysis
3. Testing
4. Keywords search



Practices of Changing Secure Software – Reflection

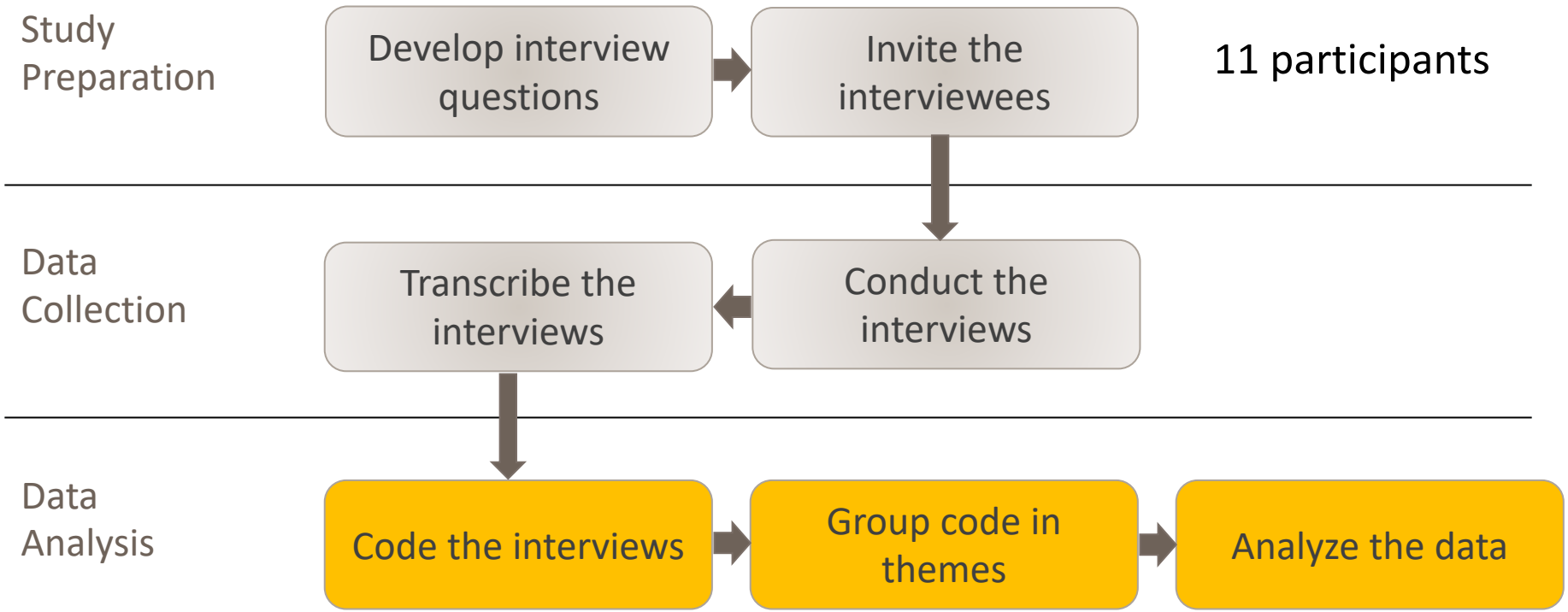
Only few participants do threat modeling

What are the practices of threat modeling for CPS?

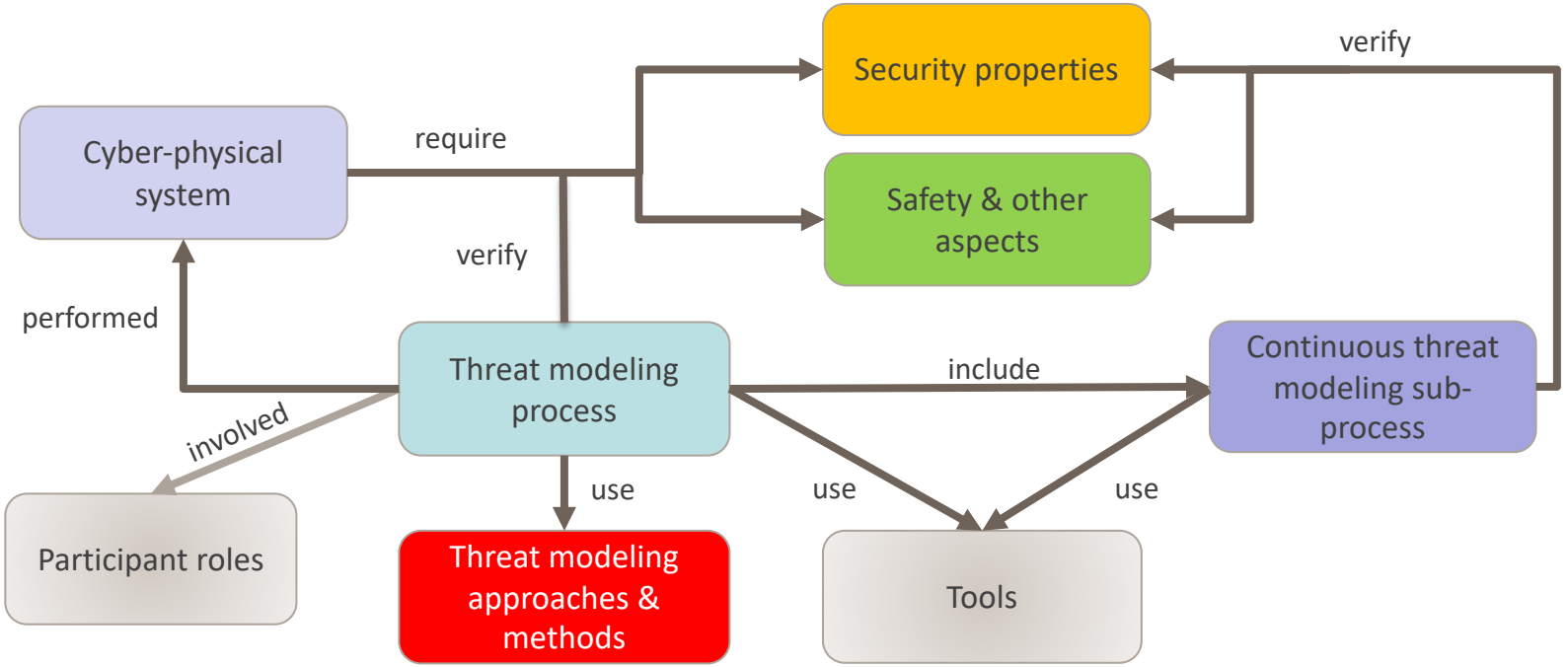
They do not use an effective methods for assessment of the security of changed code

Jamil et al., 2019

Empirical Study – Focus on the Practices of Threat Modeling



Identified Themes and Their Relationship



Empirical Study - Practices of Threat Modeling - Findings

Used approaches

- Control system background - Focus on malicious controllability of the physical components
- IT background - Classic threat modeling approaches

Used methods

- Known methods
  - STRIDE
  - PASTA
  - LINDDUN
  - Attack-tree
- Combination of known methods and approaches
- Combination of threat modeling standards and known approaches

Problems with Threat Modeling of CPSs

- The focus is on exploitable system state and not security of data
- CPS are complex – require interaction of many components
- Threat modeling is time consuming

- CPSs evolve and change continuously with limited control
  - Threat models are performed using the given architecture

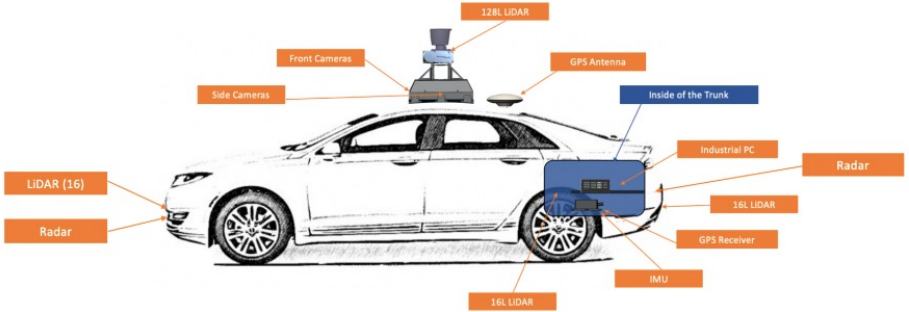
  => Requires frequent updates

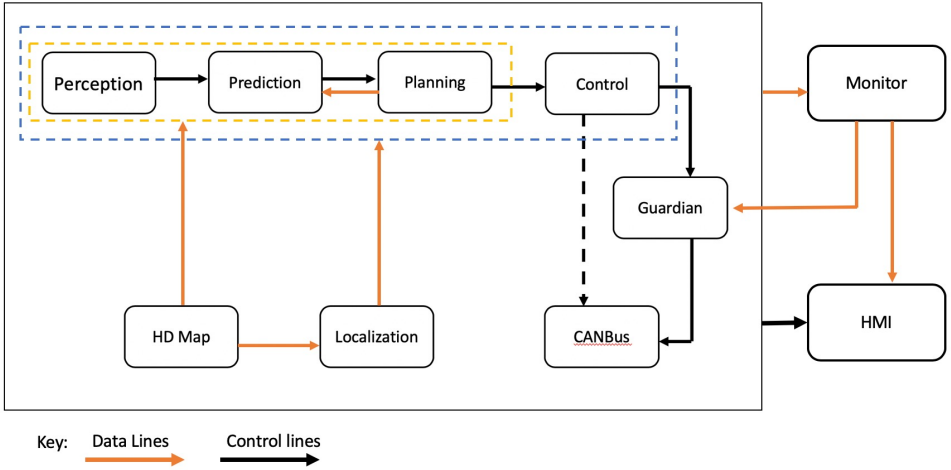
➔ Threat modeling is not practiced

Software Evolution

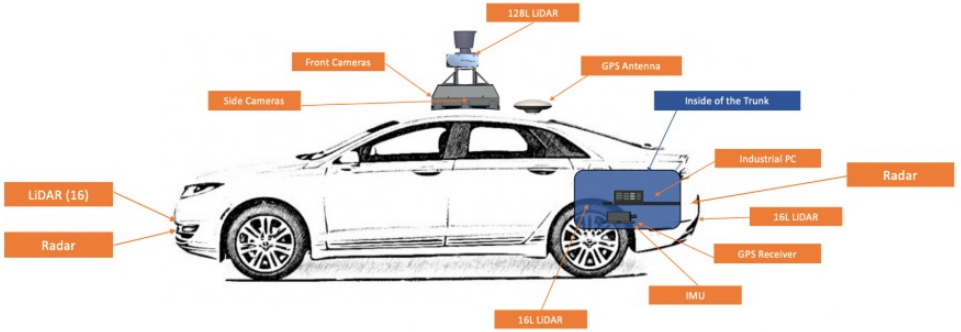
The architecture diagram is outdated

Have the code changes impacted the security of the software?

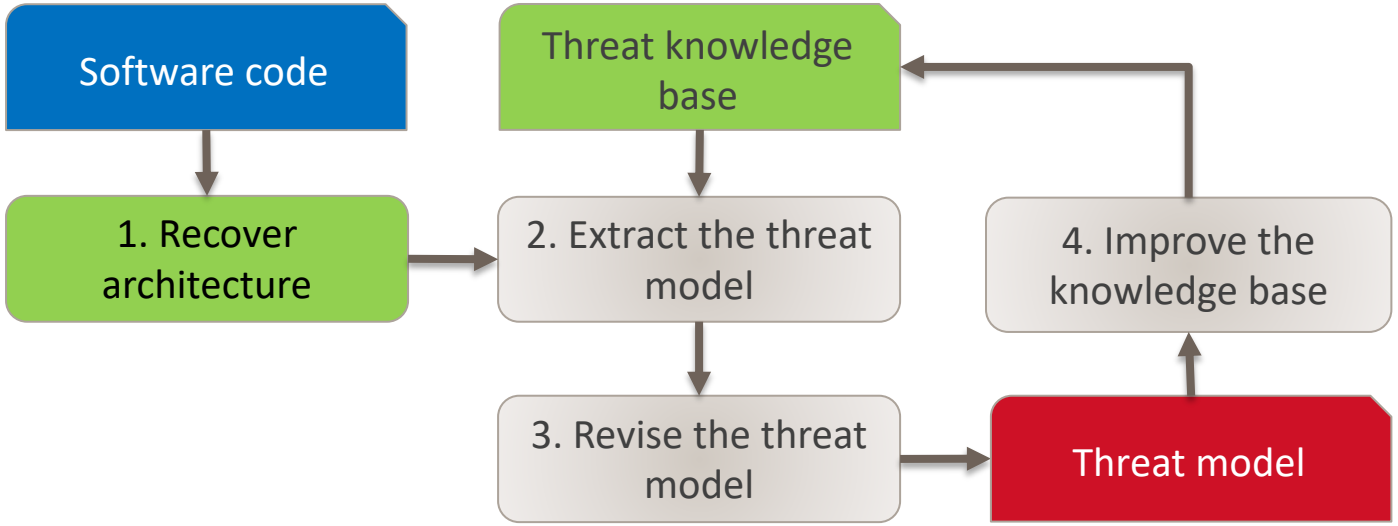




Can we automate threat modeling of a given CPS?



Threat Modeling Approach



Limitation: The focus is only on the software stack.

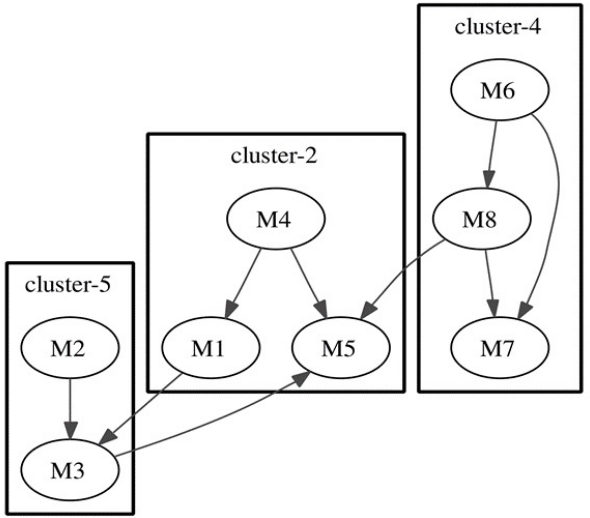
Architecture Recovery

Architecture recovery: Extract the architecture of the application from its implementation.

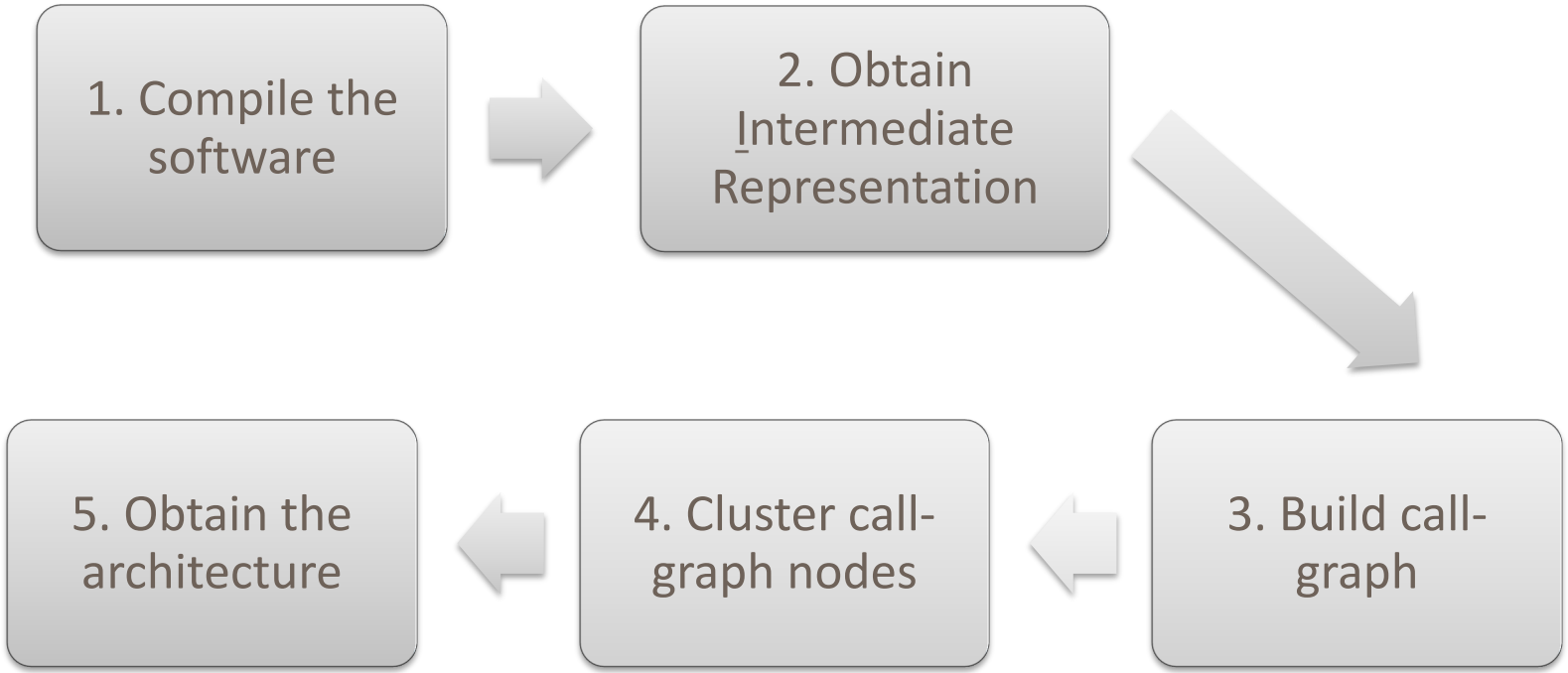
Call graph: Model the relationship between components/methods of the system.

Clustering: Identify the components of the system

Goal: Have a large number of internal connections to the components and few connections between the clusters.



Architecture Recovery



Case Study 1– universalAAL Lightning Example

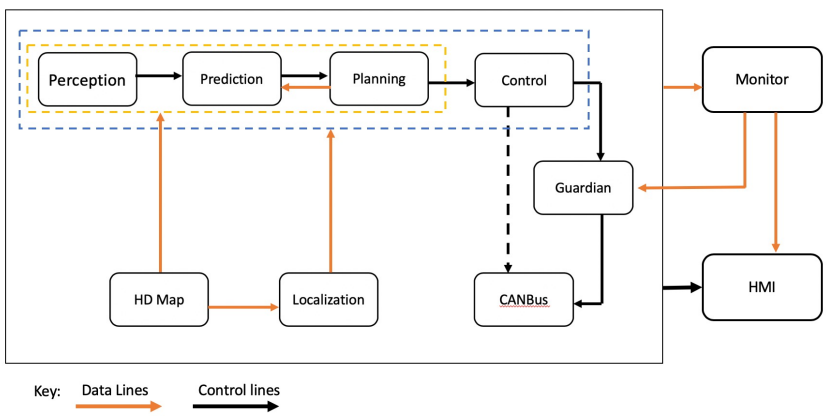
- Small Java application of universAAL (Ambient Assisted Living)
- Client GUI application to turns on/off the light.
- Call-graph nodes: 378 nodes.
- Bunch managed to cluster the call-graph and was able to recover the architecture of the application

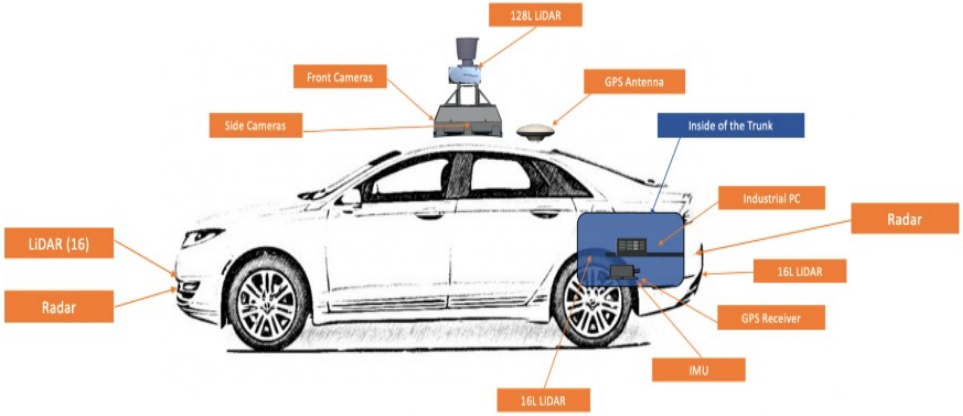
(Ali 2016)

=> Shows success for small applications.

Case Study 2: Apollo Auto

Attackers attack the CPS through the target interfaces





Given architecture

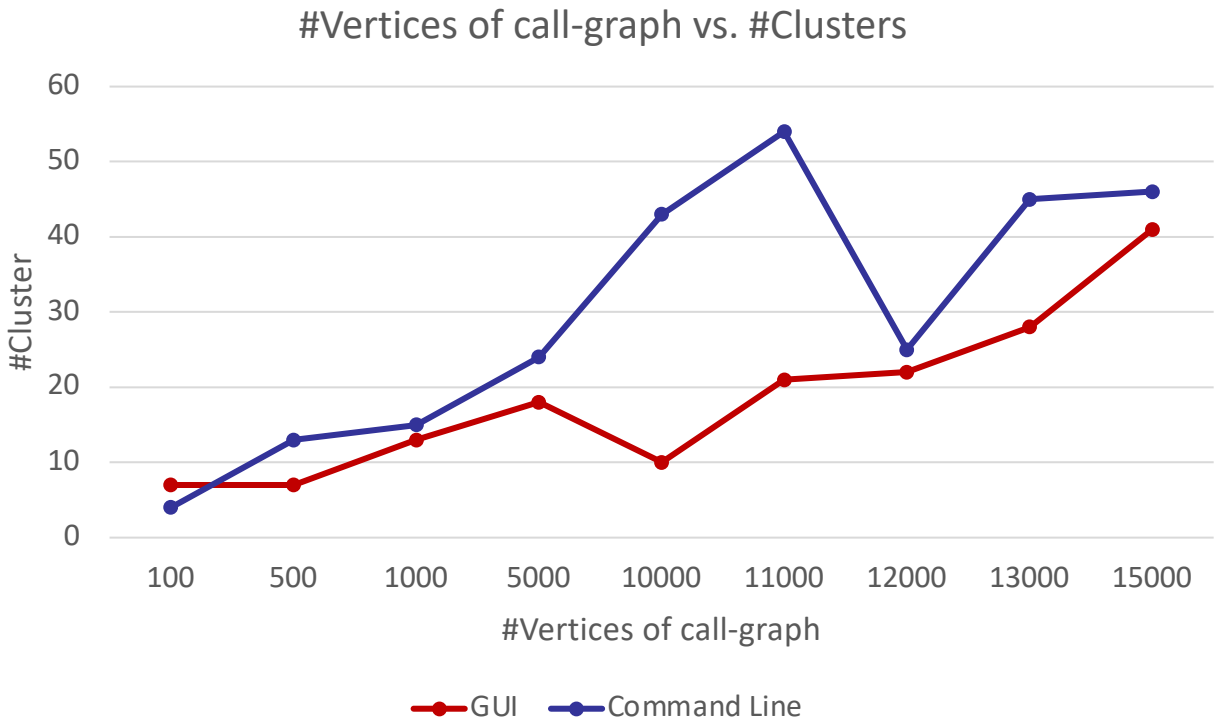
Impact of Software Change and the Target Surfaces

Target Surfaces	#Components	
	Given Arch.	Ground-truth Arch.
Map	7	7
LiDAR	4	4
Machine vision	3	3
Radar	4	4
Infrastructure sign	4	5

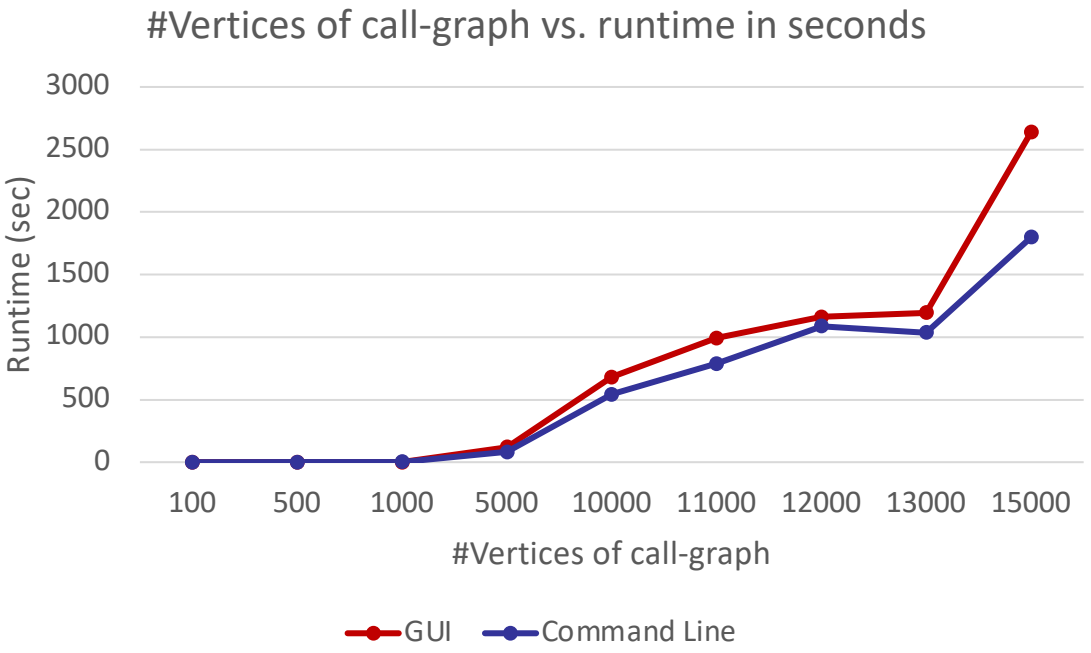
Impact of Software Change and the Target Surface

Target Surfaces	#Components	
	Given Arch.	Ground-truth Arch.
GPS	5	5
Road	1	2
In-vehicle sensor	2	2
Electronic device	1	1
Acoustic sensor	1	1

Architecture Recovery Challenges - Clustering Capabilities



Architecture Recovery Challenges - Clustering Performance



- Run Bunch on AWS x-large instance: no end for a month
- Modified Bunch to create command-line version: no end

Jamil et al., 2021

Conclusions

1. Practitioners do not use effective security assurance methods for changing software.

2. Threat modeling, unlike code analysis and penetration testing, is not commonly used.

3. Practitioners rely on their own experience to complement the outcomes of the used threat modeling methods for cyber-physical systems.

4. The performance limitation of the architecture recovery (using Bunch) is a big problem for the automation of threat modeling from source code.

Lotfi ben Othmane
othmanel@iastate.edu

Azmat Ali

Shifa Khan

Jian Lee

Ameerah-Muhsinah Jamil

Papers are available at: https://works.bepress.com/lotfi-benothmane/