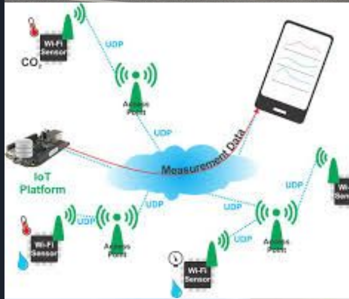
# Change Analysis & Bug Detection for CPS Dev

Interim Project Presentation

HCIRevival Group: Timothy Zimmermann, Andrea Meier, Tim Schluchter, Maximilian Böker

# Cyber-Physical Systems (CPS)

- Integration of digital and physical components
- Together they perform a well defined task
- Examples
    - Aviation
    - Automotive
    - Environmental Monitoring
    - Healthcare



# Change Analysis and Bug Detection

- Analyzation of programming artifacts such as commits or source code changes
- Change Distilling[1]:
  - Analyze changes in more detail
- Evolizer[2]:
  - Impact of change types
- DCI[3]:
  - Detect Behavioral Changes in Continuous Integration
  - Write automatically tests, that reflect the behavior of the system

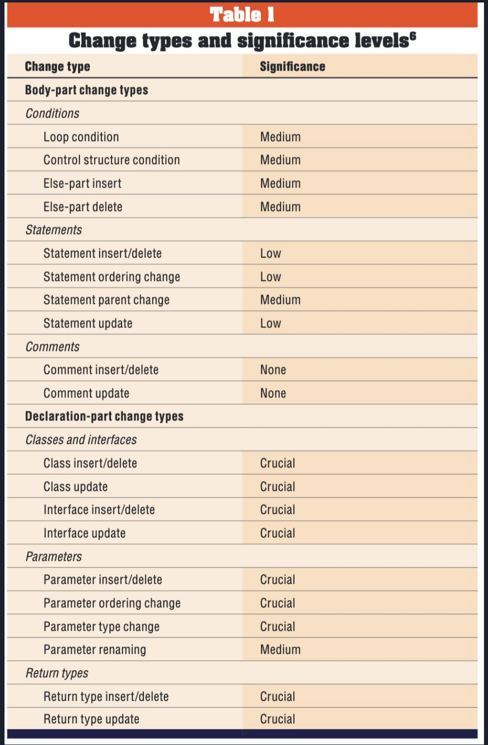


**Table 1**

**Change types and significance levels[6]**

| Change type | Significance |
|---|---|
| **Body-part change types** | |
| *Conditions* | |
| Loop condition | Medium |
| Control structure condition | Medium |
| Else-part insert | Medium |
| Else-part delete | Medium |
| *Statements* | |
| Statement insert/delete | Low |
| Statement ordering change | Low |
| Statement parent change | Medium |
| Statement update | Low |
| *Comments* | |
| Comment insert/delete | None |
| Comment update | None |
| **Declaration-part change types** | |
| *Classes and interfaces* | |
| Class insert/delete | Crucial |
| Class update | Crucial |
| Interface insert/delete | Crucial |
| Interface update | Crucial |
| *Parameters* | |
| Parameter insert/delete | Crucial |
| Parameter ordering change | Crucial |
| Parameter type change | Crucial |
| Parameter renaming | Medium |
| *Return types* | |
| Return type insert/delete | Crucial |
| Return type update | Crucial |

1) Fluri, Beat; Wursch, Michael; PInzger, Martin; Gall, Harald (2007): Change Distilling:Tree Differencing for Fine-Grained Source Code Change Extraction. In: *IIEEE Trans. Software Eng.* 33 (11), S. 725–743. DOI: 10.1109/TSE.2007.70731.
2) Gall, Harald C.; Fluri, Beat; PInzger, Martin (2009): Change Analysis with Evolizer and ChangeDistiller. In: *IEEE Softw.* 26 (1), S. 26–33. DOI: 10.1109/MS.2009.6.
3)Danglot, Benjamin; Monperrus, Martin; Rudametkin, Walter; Baudry, Benoit (2020): An approach and benchmark to detect behavioral changes of commits in continuous integration. In: *Empir Software Eng* 25 (4), S. 2379–2415. DOI: 10.1007/s10664-019-09794-7.

## Motivation (1/3)

- Code changes can have catastrophic consequences
- Boeing 747 Max crash[1], Tesla's autopilot crash[2]

1) https://nypost.com/2019/05/19/boeing-admits-to-flaw-in-737-max-flight-simulators/ ; https://www.engadget.com/2019/10/18/boeing-employees-may-have-mislead-faa-on-737-max/
2) https://www.foxnews.com/auto/tesla-smashes-overturned-truck-autopilot

## Motivation (2/3)

- Tesla's phantom breaking[1]

1) https://insideevs.com/news/406912/video-tesla-phantom-braking-crash/

## Motivation (3/3)

- Change analysis improves code quality, efficiency of software & hardware[1]
- Bugs may go live undetected with dramatic consequences[2]
- Many crucial factors in CPS: privacy, security, interoperability, data extraction, data correctness → "The safety and efficiency of the system rely on the proper software design, development, and management."[3]
- Analyzing software systems' history allows to understand software evolution and reduce maintenance costs[4]

⇒ achievable through change analysis and bug detection

1) M. Hilton, T. Tunnell, K. Huang, D. Marinov, and D. Dig. Usage, costs, and benefits of continuous integration in open-source projects. In Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, ASE 2016, pages 426–437, New York, NY, USA, 2016. ACM.
2) Danglot B., Monperrus M., Rudametkin W., and Baudry B. An Approach and Benchmark to Detect Behavioral Changes of Commits in Continuous Integration. arXiv:1902.08482v3 [cs.SE]. 2019.
3) Haque S. A., Aziz S. M., and Rahman M. Review of Cyber-Physical System in Healthcare, Hindawi Publishing Corporation International Journal of Distributed Sensor Networks Volume 2014, Article ID 217415, 20 pages http://dx.doi.org/10.1155/2014/217415. 2014.
4) Gall H. C., Fluri B., and Pinzger M. Change Analysis with Evolizer and ChangeDistiller. IEEE Software 2009, 26(1):26-33. 2009.

# Study Definition & Planning (1/2)

- Research Questions: Taxonomy for CPS code changes & bugs
  a. Understand how bugs/code changes affect CPS
  b. Specify and categorize significant and behavioral CPS changes
  c. Recognize critical changes affecting behavior of functionality in real life
- Data Extraction Process:
  - Scripts to collect issues and commits from GitHub repos in .csv format
  - Combining both with timestamp and commit ID

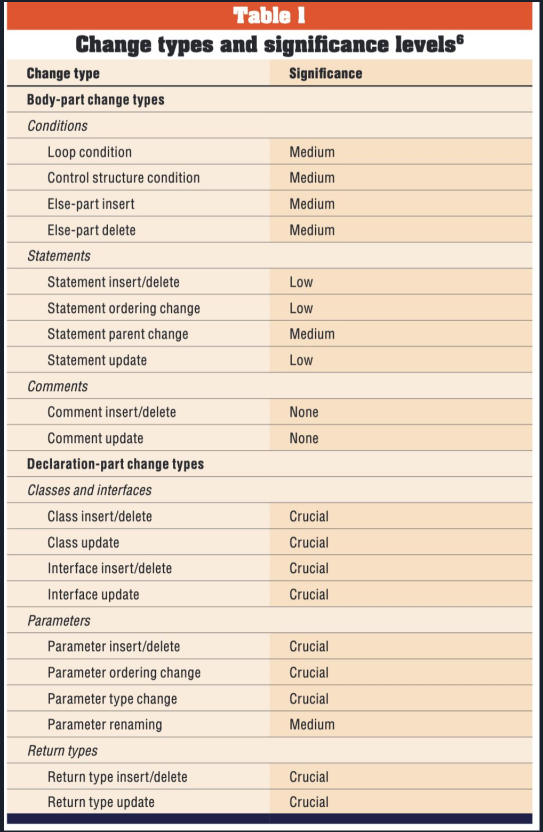
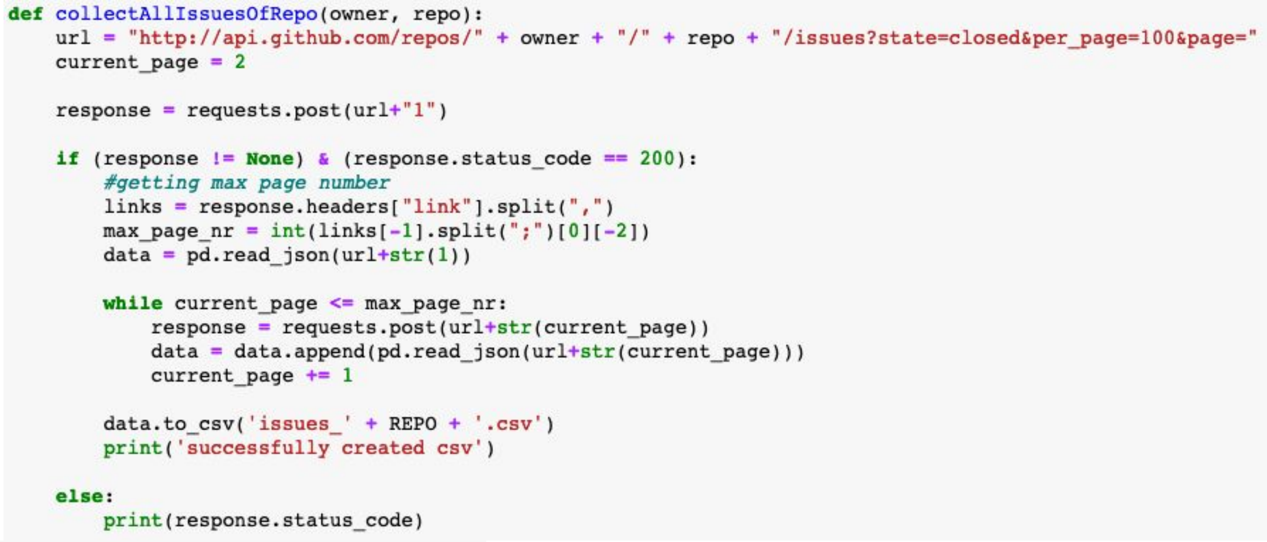


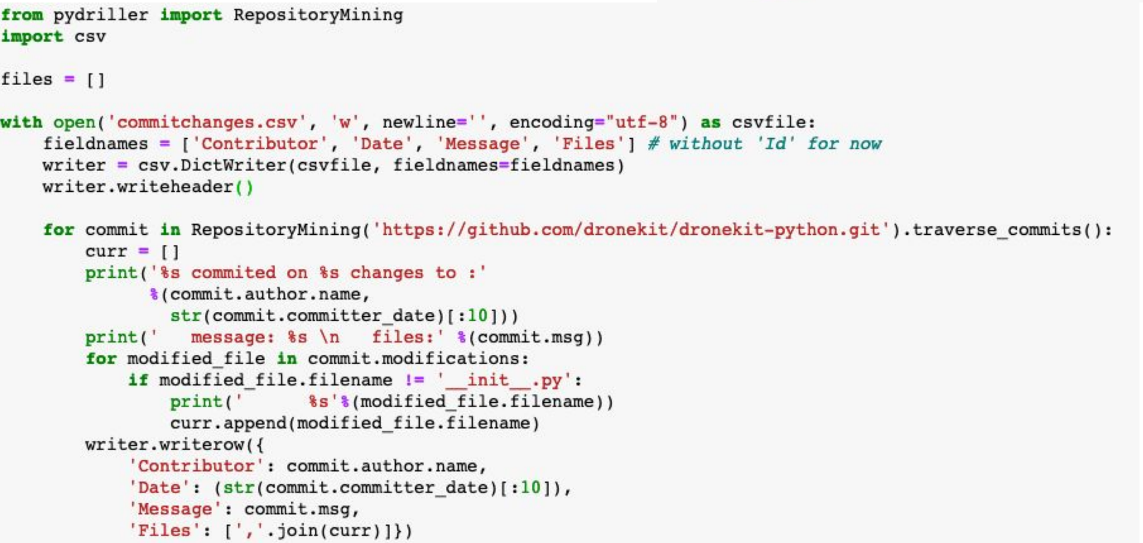
**Table I**

**Change types and significance levels[6]**

| Change type | Significance |
|---|---|
| **Body-part change types** | |
| *Conditions* | |
| Loop condition | Medium |
| Control structure condition | Medium |
| Else-part insert | Medium |
| Else-part delete | Medium |
| *Statements* | |
| Statement insert/delete | Low |
| Statement ordering change | Low |
| Statement parent change | Medium |
| Statement update | Low |
| *Comments* | |
| Comment insert/delete | None |
| Comment update | None |
| **Declaration-part change types** | |
| *Classes and interfaces* | |
| Class insert/delete | Crucial |
| Class update | Crucial |
| Interface insert/delete | Crucial |
| Interface update | Crucial |
| *Parameters* | |
| Parameter insert/delete | Crucial |
| Parameter ordering change | Crucial |
| Parameter type change | Crucial |
| Parameter renaming | Medium |
| *Return types* | |
| Return type insert/delete | Crucial |
| Return type update | Crucial |



```python
def collectAllIssuesOfRepo(owner, repo):
    url = "http://api.github.com/repos/" + owner + "/" + repo + "/issues?state=closed&per_page=100&page="
    current_page = 2

    response = requests.post(url+"1")

    if (response != None) & (response.status_code == 200):
        #getting max page number
        links = response.headers["link"].split(",")
        max_page_nr = int(links[-1].split(";")[0][-2])
        data = pd.read_json(url+str(1))

        while current_page <= max_page_nr:
            response = requests.post(url+str(current_page))
            data = data.append(pd.read_json(url+str(current_page)))
            current_page += 1

        data.to_csv('issues_' + REPO + '.csv')
        print('successfully created csv')

    else:
        print(response.status_code)
```



```python
from pydriller import RepositoryMining
import csv

files = []

with open('commitchanges.csv', 'w', newline='', encoding="utf-8") as csvfile:
    fieldnames = ['Contributor', 'Date', 'Message', 'Files'] # without 'Id' for now
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()

    for commit in RepositoryMining('https://github.com/dronekit/dronekit-python.git').traverse_commits():
        curr = []
        print('%s commited on %s changes to :'
              %(commit.author.name,
                str(commit.committer_date)[:10]))
        print('    message: %s \n    files:' %(commit.msg))
        for modified_file in commit.modifications:
            if modified_file.filename != '__init__.py':
                print('        %s'%(modified_file.filename))
                curr.append(modified_file.filename)
        writer.writerow({
            'Contributor': commit.author.name,
            'Date': (str(commit.committer_date)[:10]),
            'Message': commit.msg,
            'Files': [','.join(curr)]})
```

## Study Definition & Planning (2/2)

- Potential Hypothesis:
    - Categorization of CPS code changes/bugs is possible
    - CPS have characteristics of bugs that can be separated from bugs of other domains, specific taxonomy ⇒ use defined taxonomies to design models for CPS and feed them into machine learning to predict types of behavioral changes and failures

## Next Steps

- Gather all issues and commits
- Randomized commit sampling
- Analyze these commits
  - Classify based on the previously shown classification
  - Find a connection between issues and relevant commit