Change Analysis & Bug Detection for CPS Dev

Final Project Presentation

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

Content

- Intro to CPS
- Motivation
- Study & Methodology
- Results
- Future Work

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 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

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

- Code changes can have catastrophic consequences
- Examples: Boeing 737 Max crash[1], Tesla's autopilot crash[2]
- Change analysis improves code quality, efficiency of software & hardware[3]
- Early defect detection contributes significantly to quality assurance[4]
- Analyzing software systems' history reduces maintenance costs[5]

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
3) 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.
4) 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.
5) 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

- Research Questions: Taxonomy for CPS code changes & bugs
  a. Specify and categorize significant and behavioral CPS changes
  b. Recognize critical changes affecting behavior of functionality in real life
- Hypothesis:
  a. Categorization of CPS code changes/bugs is possible
  b. CPS have specific taxonomy
     - use taxonomy to design models for CPS
     - feed into ML to predict types of behavioral changes and failures

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

Approach & Methodology

- Identified area of interest: CPS repos
- Filtered repos: safety critical & between 100 & 3000 commits

  ⇒ 12 repositories

- Extract issues & commits from repos, stored as CSV file

```
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

        #if more than 1 page
        if "link" in response.headers:
            links = response.headers["link"].split(",")
            max_page_nr = int(links[-1].split(";")[0][-2])

        #if only 1 page
        else:
            max_page_nr = current_page

        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 for: ', repo)

    else:
        print(response.status_code)
```

```
## The commit gatherer mines the GitHub repositories given within "respositories.csv" for relevant commit changes,
## it filters out commit changes that include certains files (see ignore_filenames for details) + only considers commits that involve 1 to 11 files,
## for each repositories it creates one CSV file named "Commit_CSVs_{repositoryname}.csv" in which information on each commit change is listed.

from pydriller import RepositoryMining
import pandas as pd
import csv

repositories = pd.read_csv("Additional Resources/repositories.csv")

files = []
ignore_filenames = ['__init__.py', 'readme.md', '.gitignore', '', '__main__.py']
for key, repository in repositories.iterrows():
    print('Gathering commits for {}'.format(repository['Name']))
    with open('Commit_CSVs/Commit_CSVs_{}.csv'.format(repository['Name']), 'w+', newline='', encoding="utf-8") as csvfile:
        fieldnames = ['Commit_ID','Contributor', 'Date', 'Message', 'Files', 'Branch','Repository']  # without 'Id' for now
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()
        bug_commits = []
        # print(RepositoryMining(repository['URL']).branches())
        for commit in RepositoryMining(repository['URL']).traverse_commits():
            curr = []
            for modified_file in commit.modifications:
                # ignore certain files common on each git repo, but unnecessary
                if modified_file.filename.lower() not in ignore_filenames:
                    curr.append(modified_file.filename)
            commit_mess = commit.msg.replace('\n','')
            commit_mess.replace('\t', '')
            if len(curr) in range(1, 11):
                writer.writerow({
                    'Commit_ID': commit.hash,
                    'Contributor': commit.author.name,
                    'Date': (str(commit.committer_date)[:10]),
                    'Message': commit_mess,
                    'Files': [','.join(curr)],
                    'Branch': str(commit.branches),
                    'Repository': repository['Name'],
                    })
            if any(word in commit.msg.lower() for word in ['bug', 'error', 'problem']):
                bug_commits.append([commit.hash, commit.author.name, (str(commit.committer_date)[:10]), commit_mess, [','.join(curr)]])
```

Approach & Methodology

- Defined taxonomy[1)]
  - added significance level, extended with low level classification “deletion of code”
- Created random sample from commits (113)
- Commit classification
  - Multiple changes: highest priority
  - Threshold of 5 minutes: discarded
  - If unuseful file format: discarded

```
import pandas as pd
import glob
import math
```

```
SAMPLE = "./Commit_CSVs/Sample_Commits.csv"
PATH = './Commit_CSVs/[!Sample]*.csv'
PERCENTAGE = 0.01
```

Read all files from Commit_CSVs folder and create a dataframe

```
df = pd.concat(map(pd.read_csv, glob.glob(PATH)))
```

Takes a sample of all commits. Size is dependet on the indicated percentage. If there is already an existing Sample_Commits.csv file it gets replaced.

```
number_of_commits = df.shape[0]
sample_size = math.ceil(number_of_commits * PERCENTAGE)
sample = df.sample(sample_size)
sample.to_csv(SAMPLE)
sample
```

1) Panichella, S., & Zaugg, N. (2020). An empirical investigation of relevant changes and automation needs in modern code review. *Empirical Software Engineering*, *25*(6), 4833-4872.

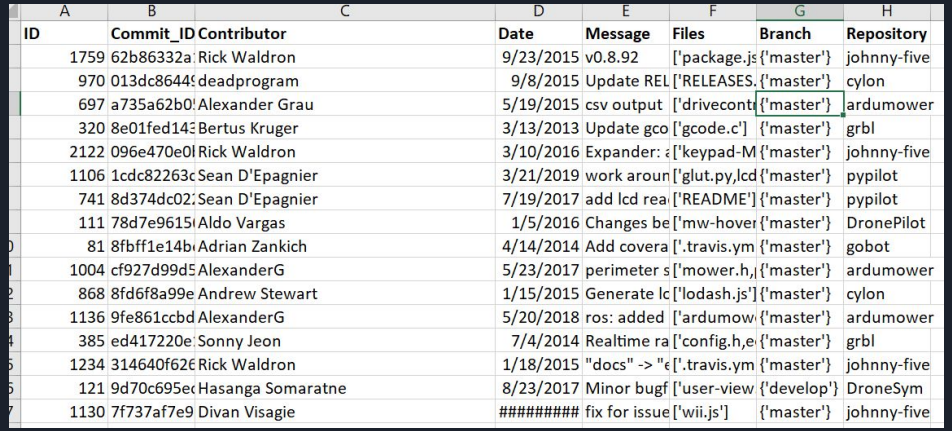
Category	High-level Change	Low-level Change	Significance
Documentation (D)	Textual Documentation (D)	D.1 Naming	Low
		D.2 Comments	Low
		D.3 License Header	Low
		D.4 Typos	Low
		D.5 Other	Low
	Language Supported Documentation (D)	D.6 Immutability	Low
		D.7 Visibility (Modifiers)	Low
Style (S)	Style (S)	S.1 Brackets & Braces	Medium
		S.2 Indentation	Medium
		S.3 Blank Lines	Low
		S.4 Long Lines	Low
		S.5 Whitespace Usage	Low
		S.6 Grouping	Low
		S.7 Commented out code	Medium
Structure (STR)	Re-implementation (STR)	STR.1 Semantic Duplication	Medium
		STR.2 Semantic Dead Code	Low
		STR.3 Change Function	Crucial
		STR.4 Standard Coding Conventions	Low
		STR.5 New Functionality	Crucial
		STR.6 Strings (Wording)	Low
		STR.7 Logging	Low
		STR.8 Testing	Crucial
	Organization (STR)	STR.9 Imports	Crucial
		STR.10 Move Function	Medium
		STR.11 Long Subroutine	Medium
		STR.12 Dead Code	Low
		STR.13 Duplication / Redundant Code	Low
		STR.14 Complex Code / Simplification	Medium
		STR.15 Statement Issue	Medium
		STR.16 Consistency	Medium
		STR.17 Architectural changes	Crucial
		STR.18 Deletion of Code	Crucial

Interface (I)	Interface (I)	I.1 Function Call	Medium
		I.2 Parameter	Crucial
Logic (L)	Logic (L)	L.1 Compare	Crucial
		L.2 Computation	Crucial
		L.3 Wrong Location	Medium
		L.4 Algorithm/Performance	Crucial
Resource (R)	Resource (R)	R.1 Variable Initialization	Medium
		R.2 Memory Management	Medium
		R.3 Data & Resource Manipulation	Medium
		R.4 Security	Crucial
		R.5 Concurrency	Medium
Check (C)	Check (C)	C.1 Check Function	Crucial
		C.2 Check Variable	Crucial
		C.3 Check User Input	Crucial
Larger Defects (LD)	Larger Defects (LD)	LD.1 Completeness	Crucial
		LD.2 GUI	Medium
		LD.3 Check outside code / Domino Effects	Medium

Extracting Commit Changes

commit_gatherer.py → Sampler.ipynb → reverse_commit_search.py

- Commit_CSVs_ardumower
- Commit_CSVs_cylon
- Commit_CSVs_dronekit-python
- Commit_CSVs_DronePilot
- Commit_CSVs_DroneSym
- Commit_CSVs_gobot
- Commit_CSVs_grbl
- Commit_CSVs_johnny-five
- Commit_CSVs_node-ar-drone
- Commit_CSVs_pypilot
- Commit_CSVs_TurtleBot
- Commit_CSVs_Valetudo
- Sample_Commits



	A	B	C	D	E	F	G	H
	ID	Commit_ID	Contributor	Date	Message	Files	Branch	Repository
	1759	62b86332a	Rick Waldron	9/23/2015	v0.8.92	['package.js	{'master'}	johnny-five
	970	013dc8644	deadprogram	9/8/2015	Update REL	['RELEASES.	{'master'}	cylon
	697	a735a62b0	Alexander Grau	5/19/2015	csv output	['drivecont	{'master'}	ardumower
	320	8e01fed143	Bertus Kruger	3/13/2013	Update gco	['gcode.c']	{'master'}	grbl
	2122	096e470e0	Rick Waldron	3/10/2016	Expander: a	['keypad-M	{'master'}	johnny-five
	1106	1cdc82263	Sean D'Epagnier	3/21/2019	work aroun	['glut.py,lcd	{'master'}	pypilot
	741	8d374dc02	Sean D'Epagnier	7/19/2017	add lcd rea	['README']	{'master'}	pypilot
	111	78d7e9615	Aldo Vargas	1/5/2016	Changes be	['mw-hover	{'master'}	DronePilot
0	81	8fbff1e14b	Adrian Zankich	4/14/2014	Add covera	['.travis.ym	{'master'}	gobot
1	1004	cf927d99d5	AlexanderG	5/23/2017	perimeter s	['mower.h,/	{'master'}	ardumower
2	868	8fd6f8a99e	Andrew Stewart	1/15/2015	Generate lc	['lodash.js']	{'master'}	cylon
3	1136	9fe861ccbd	AlexanderG	5/20/2018	ros: added	['ardumow	{'master'}	ardumower
4	385	ed417220e	Sonny Jeon	7/4/2014	Realtime ra	['config.h,e	{'master'}	grbl
5	1234	314640f626	Rick Waldron	1/18/2015	"docs" -> "e	['.travis.ym	{'master'}	johnny-five
6	121	9d70c695e	Hasanga Somaratne	8/23/2017	Minor bugf	['user-view	{'develop'}	DroneSym
7	1130	7f737af7e9	Divan Visagie	#########	fix for issue	['wii.js']	{'master'}	johnny-five

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

Name

- drivecontrol.ino_after
- drivecontrol.ino_before
- motorcontrol.cpp_after
- motorcontrol.cpp_before
- motorcontrol.h_after
- motorcontrol.h_before

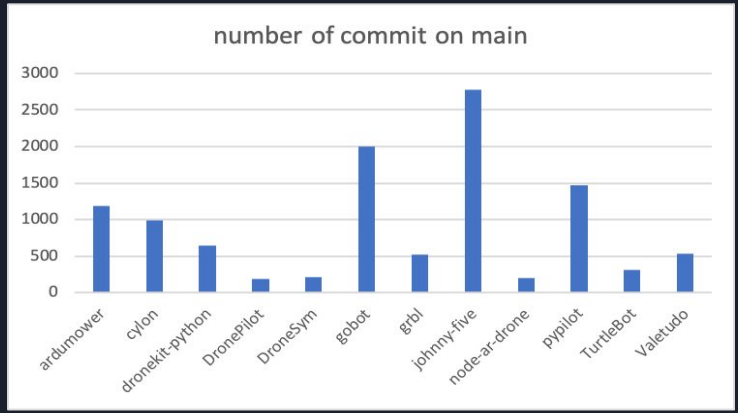
Results

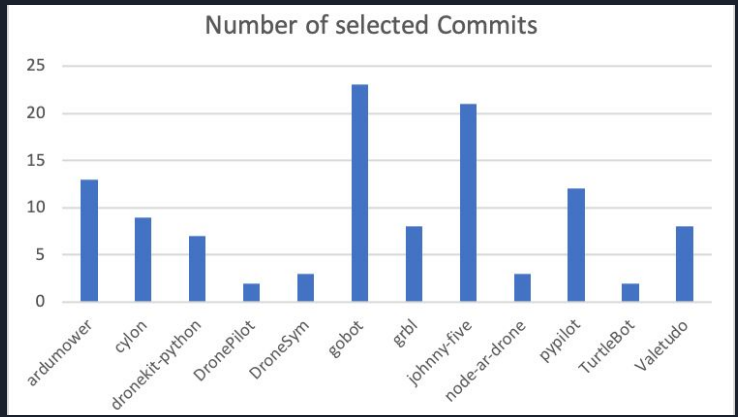
4 repositories > 1000 commits

4 repositories ∈ [1000; 500]

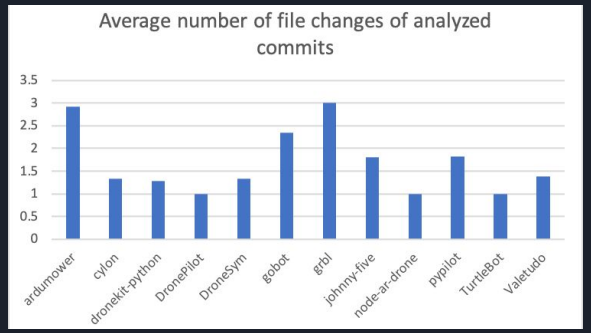
4 repositories < 500 commits

Sampled 113 commits out of 11’000





Results



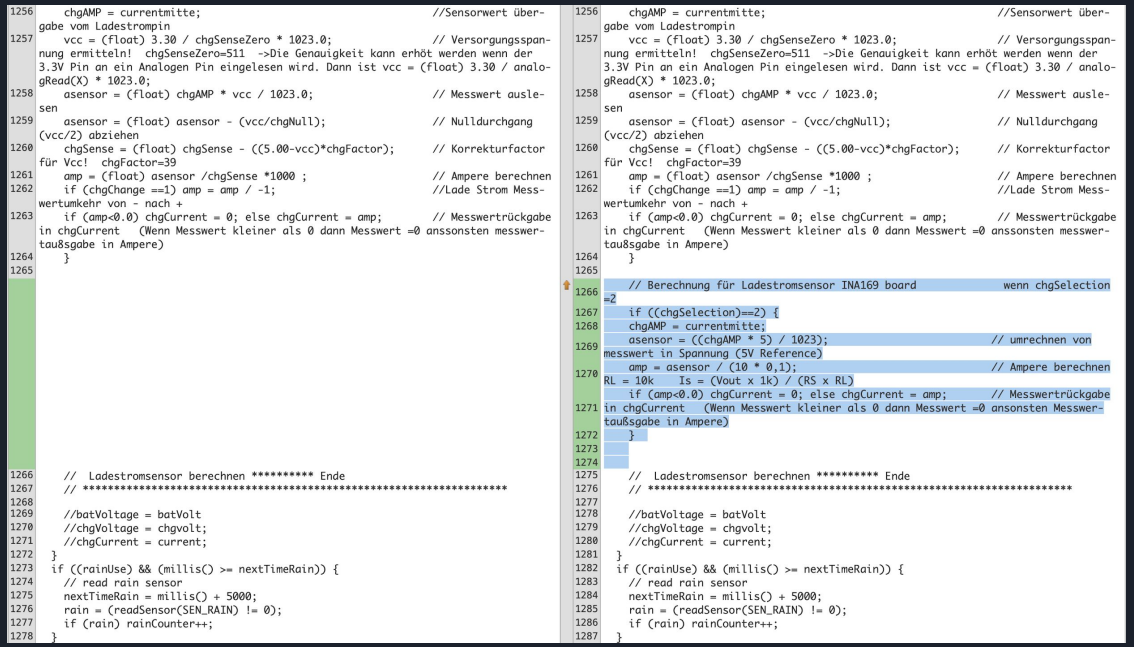
Bigger projects tend to have more files changed per commit with exception of GRBL

File Changes	Count
Source code files (.cpp, .py, .js, .h, .c, .ino, .sh, .go)	116**
Text files* (.txt, .md, .html)	12
package.json	5
Unknown filetype	3

* README.md files were already filtered out and thus not displayed in our table.
** Some commits contained both file types and were counted twice

Example 1

Ardumower
Added new computation which computes the charge on batteries for a new PCB



```
    chgAMP = currentmitte;                                  //Sensorwert über-
gabe vom Ladestrompin
    vcc = (float) 3.30 / chgSenseZero * 1023.0;             // Versorgungsspan-
nung ermitteln!  chgSenseZero=511  ->Die Genauigkeit kann erhöt werden wenn der
3.3V Pin an ein Analogen Pin eingelesen wird. Dann ist vcc = (float) 3.30 / analo-
gRead(X) * 1023.0;
    asensor = (float) chgAMP * vcc / 1023.0;                // Messwert ausle-
sen
    asensor = (float) asensor - (vcc/chgNull);              // Nulldurchgang
(vcc/2) abziehen
    chgSense = (float) chgSense - ((5.00-vcc)*chgFactor);   // Korrekturfactor
für Vcc!  chgFactor=39
    amp = (float) asensor /chgSense *1000 ;                 // Ampere berechnen
    if (chgChange ==1) amp = amp / -1;                      //Lade Strom Mess-
wertumkehr von - nach +
    if (amp<0.0) chgCurrent = 0; else chgCurrent = amp;     // Messwertrückgabe
in chgCurrent   (Wenn Messwert kleiner als 0 dann Messwert =0 anssonsten messwer-
tau8sgabe in Ampere)
    }
```

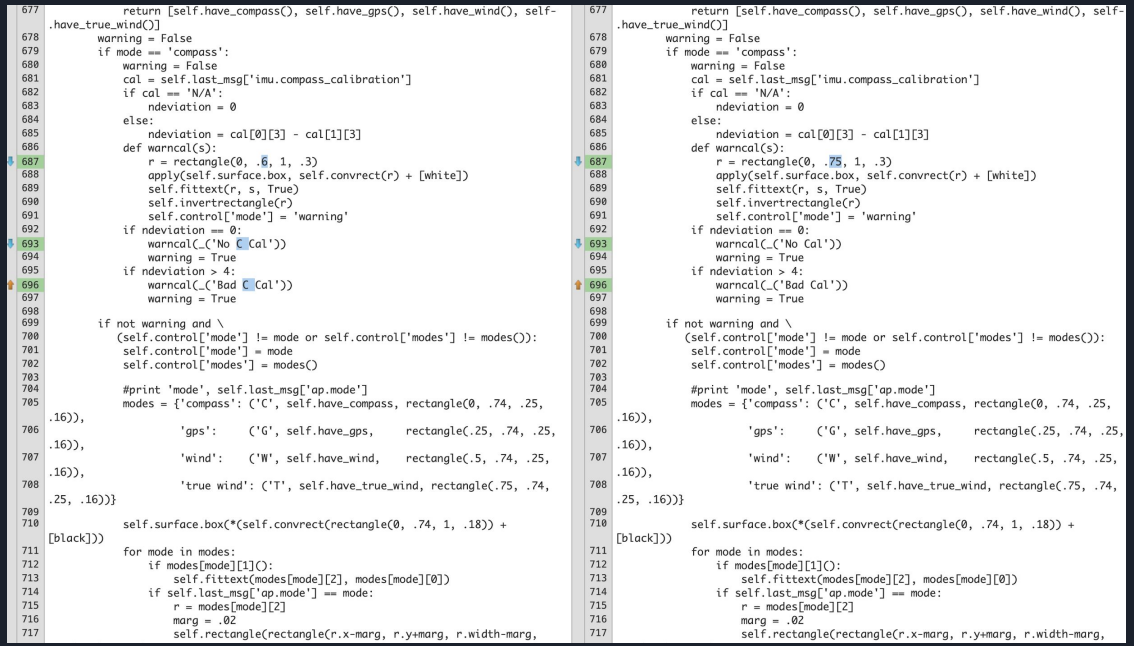
```
    //  Ladestromsensor berechnen ********** Ende
    // ************************************************************************

    //batVoltage = batVolt
    //chgVoltage = chgvolt;
    //chgCurrent = current;
  }
  if ((rainUse) && (millis() >= nextTimeRain)) {
    // read rain sensor
    nextTimeRain = millis() + 5000;
    rain = (readSensor(SEN_RAIN) != 0);
    if (rain) rainCounter++;
  }
```

```
    chgAMP = currentmitte;                                  //Sensorwert über-
gabe vom Ladestrompin
    vcc = (float) 3.30 / chgSenseZero * 1023.0;             // Versorgungsspan-
nung ermitteln!  chgSenseZero=511  ->Die Genauigkeit kann erhöt werden wenn der
3.3V Pin an ein Analogen Pin eingelesen wird. Dann ist vcc = (float) 3.30 / analo-
gRead(X) * 1023.0;
    asensor = (float) chgAMP * vcc / 1023.0;                // Messwert ausle-
sen
    asensor = (float) asensor - (vcc/chgNull);              // Nulldurchgang
(vcc/2) abziehen
    chgSense = (float) chgSense - ((5.00-vcc)*chgFactor);   // Korrekturfactor
für Vcc!  chgFactor=39
    amp = (float) asensor /chgSense *1000 ;                 // Ampere berechnen
    if (chgChange ==1) amp = amp / -1;                      //Lade Strom Mess-
wertumkehr von - nach +
    if (amp<0.0) chgCurrent = 0; else chgCurrent = amp;     // Messwertrückgabe
in chgCurrent   (Wenn Messwert kleiner als 0 dann Messwert =0 anssonsten messwer-
tau8sgabe in Ampere)
    }

    // Berechnung für Ladestromsensor INA169 board                wenn chgSelection
=2
    if ((chgSelection)==2) {
    chgAMP = currentmitte;
    asensor = ((chgAMP * 5) / 1023);                        // umrechnen von
messwert in Spannung (5V Reference)
    amp = asensor / (10 * 0,1);                             // Ampere berechnen
RL = 10k    Is = (Vout x 1k) / (RS x RL)
    if (amp<0.0) chgCurrent = 0; else chgCurrent = amp;     // Messwertrückgabe
in chgCurrent   (Wenn Messwert kleiner als 0 dann Messwert =0 ansonsten Messwer-
taußsgabe in Ampere)
    }


    //  Ladestromsensor berechnen ********** Ende
    // ************************************************************************

    //batVoltage = batVolt
    //chgVoltage = chgvolt;
    //chgCurrent = current;
  }
  if ((rainUse) && (millis() >= nextTimeRain)) {
    // read rain sensor
    nextTimeRain = millis() + 5000;
    rain = (readSensor(SEN_RAIN) != 0);
    if (rain) rainCounter++;
  }
```

Commit	Commit ID	Commit Notes	File Change	High-level Change	Low-level Change	Significance	Evaluator	Reviewer	Comment
70	99df7e58b25e3a827b6ab3284004e859c118060c	Ladestromsensor INA169 hinzugefügt	Mower.cpp robot.cpp	Logic (L)	L.2 Computation	Crucial	Max	Timothy	- mower.cpp --> comment - robot.cpp --> new if-condition / computation

Example 2

Adjusted display/design of error message on the UI



```
677             return [self.have_compass(), self.have_gps(), self.have_wind(), self.have_true_wind()]
678         warning = False
679         if mode == 'compass':
680             warning = False
681             cal = self.last_msg['imu.compass_calibration']
682             if cal == 'N/A':
683                 ndeviation = 0
684             else:
685                 ndeviation = cal[0][3] - cal[1][3]
686             def warncal(s):
687                 r = rectangle(0, .6, 1, .3)
688                 apply(self.surface.box, self.convrect(r) + [white])
689                 self.fittext(r, s, True)
690                 self.invertrectangle(r)
691                 self.control['mode'] = 'warning'
692             if ndeviation == 0:
693                 warncal(_('No C Cal'))
694                 warning = True
695             if ndeviation > 4:
696                 warncal(_('Bad C Cal'))
697                 warning = True
698
699         if not warning and \
700            (self.control['mode'] != mode or self.control['modes'] != modes()):
701             self.control['mode'] = mode
702             self.control['modes'] = modes()
703
704             #print 'mode', self.last_msg['ap.mode']
705             modes = {'compass': ('C', self.have_compass, rectangle(0, .74, .25, .16)),
706                      'gps':     ('G', self.have_gps,     rectangle(.25, .74, .25, .16)),
707                      'wind':    ('W', self.have_wind,    rectangle(.5, .74, .25, .16)),
708                      'true wind': ('T', self.have_true_wind, rectangle(.75, .74, .25, .16))}
709
710             self.surface.box(*(self.convrect(rectangle(0, .74, 1, .18)) + [black]))
711             for mode in modes:
712                 if modes[mode][1]():
713                     self.fittext(modes[mode][2], modes[mode][0])
714                 if self.last_msg['ap.mode'] == mode:
715                     r = modes[mode][2]
716                     marg = .02
717                     self.rectangle(rectangle(r.x-marg, r.y+marg, r.width-marg,
```

```
677             return [self.have_compass(), self.have_gps(), self.have_wind(), self.have_true_wind()]
678         warning = False
679         if mode == 'compass':
680             warning = False
681             cal = self.last_msg['imu.compass_calibration']
682             if cal == 'N/A':
683                 ndeviation = 0
684             else:
685                 ndeviation = cal[0][3] - cal[1][3]
686             def warncal(s):
687                 r = rectangle(0, .75, 1, .3)
688                 apply(self.surface.box, self.convrect(r) + [white])
689                 self.fittext(r, s, True)
690                 self.invertrectangle(r)
691                 self.control['mode'] = 'warning'
692             if ndeviation == 0:
693                 warncal(_('No Cal'))
694                 warning = True
695             if ndeviation > 4:
696                 warncal(_('Bad Cal'))
697                 warning = True
698
699         if not warning and \
700            (self.control['mode'] != mode or self.control['modes'] != modes()):
701             self.control['mode'] = mode
702             self.control['modes'] = modes()
703
704             #print 'mode', self.last_msg['ap.mode']
705             modes = {'compass': ('C', self.have_compass, rectangle(0, .74, .25, .16)),
706                      'gps':     ('G', self.have_gps,     rectangle(.25, .74, .25, .16)),
707                      'wind':    ('W', self.have_wind,    rectangle(.5, .74, .25, .16)),
708                      'true wind': ('T', self.have_true_wind, rectangle(.75, .74, .25, .16))}
709
710             self.surface.box(*(self.convrect(rectangle(0, .74, 1, .18)) + [black]))
711             for mode in modes:
712                 if modes[mode][1]():
713                     self.fittext(modes[mode][2], modes[mode][0])
714                 if self.last_msg['ap.mode'] == mode:
715                     r = modes[mode][2]
716                     marg = .02
717                     self.rectangle(rectangle(r.x-marg, r.y+marg, r.width-marg,
```

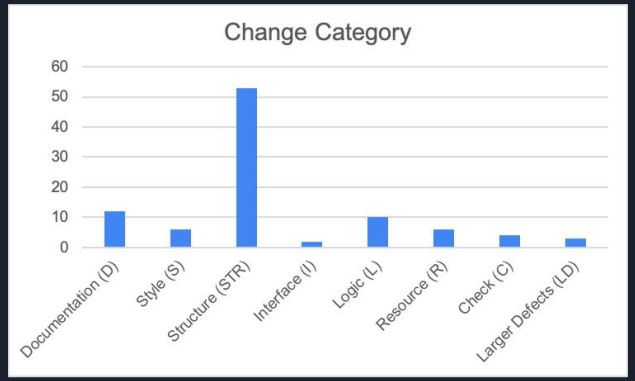
Commit	Commit ID	Commit Notes	File Change	High-level Change	Low-level Change	Significance	Evaluator	Reviewer	Comment
47	fe08d3b6fe2dac9b661beeb66bd71dbdfd675c9	nicer cal error display	['client.py']	Re-implementation (STR)	STR.6 Strings (Wording)	Low	Tim	Andrea	changed interface element size and wordings

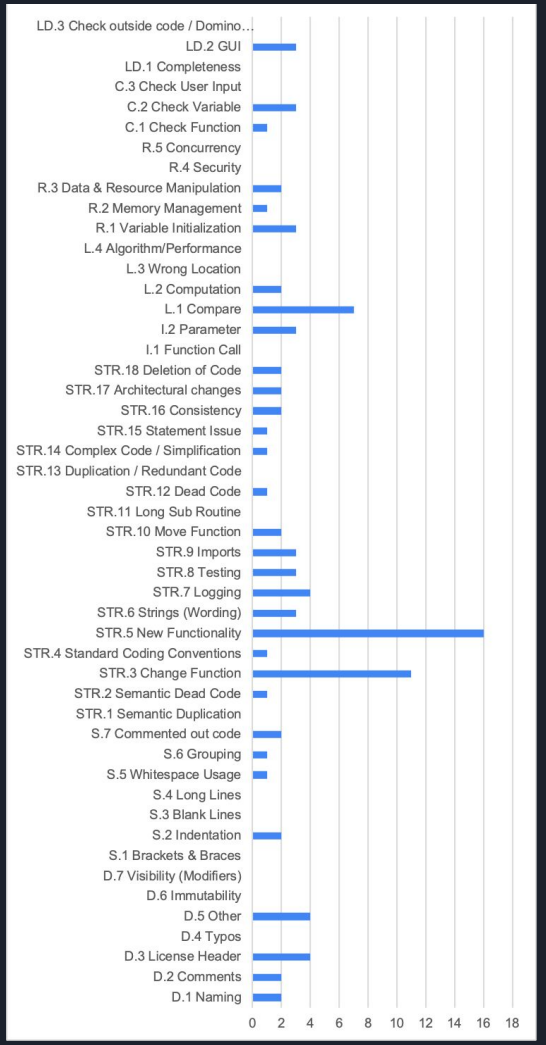
Results

Commit	Commit ID	Commit Notes	File Change	High-level Change	Low-level Change	Significance	Evaluator	Reviewer	Comment
...	...	...	...	...	...	...	...	...	...
47	fe08d3b6fe2dac9b661ebeeb66bd71dbdfd675c9	nicer cal error display	['client.py']	Re-implementation (STR)	STR.6 Strings (Wording)	Low	Tim	Andrea	changed interface element size and wordings
...	...	...	...	...	...	...	...	...	...
70	99df7e58b25e3a827b6ab3284004e859c118060c	Ladestromsensor INA169 hinzugefügt	Mower.cpp robot.cpp	Logic (L)	L.2 Computation	Crucial	Max	Timothy	- mower.cpp --> comment - robot.cpp --> new if-condition / computation
...	...	...	...	...	...	...	...	...	...

Results

Significance	Count
Low	26
Medium	19
Critical	51
N/A*	17





* Contained more than 3 files that were changed and therefore discarded

GitHub Repository

https://github.com/mboeke/hcirevivalgroup

Future Work

- Future Goal: System predicts how errorprone commit to CPS is.
- Improve data/Limitations of our work
  - More Data
  - Improve labeling guidelines
  - More features per commit
- What to predict?
  - First: binary classifier
  - Goal: multiclass classification

Thanks!

Any questions?