# Detecting Anti-Patterns of Continuous Integration

Thomas Dahlgren
University of Minnesota Morris

# Introduction

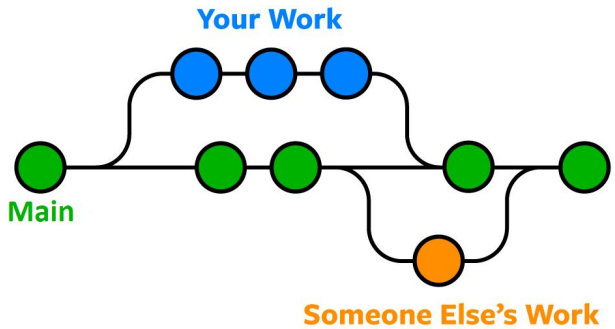
- Continuous integration (CI) is a way of developing software
  - Small changes integrate faster
  - Improves feedback
  - Improves development time
- Anti-patterns take away the advantages
  - Can slowly enter a project
  - Might not know about them
  - Appear in all parts of a project
- Can we detect them?

# Outline

- Introduction
- **Background**
  - Github and version control
  - Continuous Integration (CI)
  - Anti-patterns and CI
- Detection of anti-patterns with CI-Odor and developer's feedback
- Detection of anti-patterns with Hansel and Gretel and developer's feedback
- Conclusion

# Background: GitHub

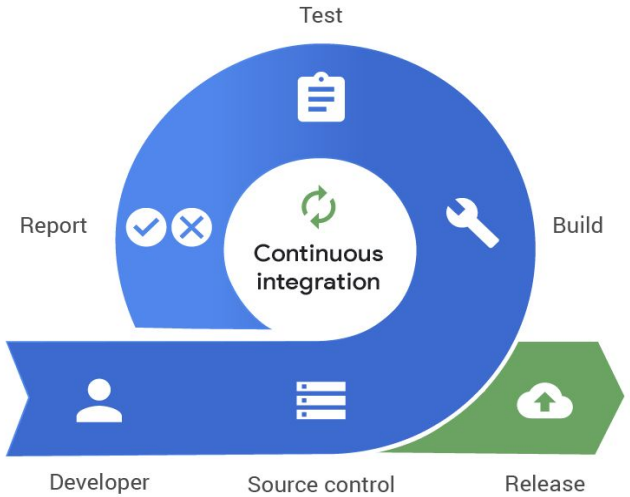
- Developers can store their code
- Branches are used for teams to integrate their code
- Keeps everything stored




https://www.nobledesktop.com/blog/what-is-git-and-why-should-you-use-it

# Background: Continuous Integration

- Continually integrate code into main
- Run builds and tests
  - A build is runnable code
  - Tests are made by the team for the code
- Repeat
- Release




https://www.pagerduty.com/resources/learn/what-is-continuous-integration/

# Background: CI configuration Files

- Used to build the project on the cloud (Travis CI, GitHub actions)
- Includes the build order of jobs
- Specifies the tools needed to build
- Is updated with the project as it grows

```
before_install:
  - sudo apt-get update -qq

install:
  - wget https://bit.ly/hj67 -O /tmp/casper.tar.gz
  - tar -xvf /tmp/casper.tar.gz
  - bower install
```

Source: Gallaba (2020)

# Background: Anti-Patterns

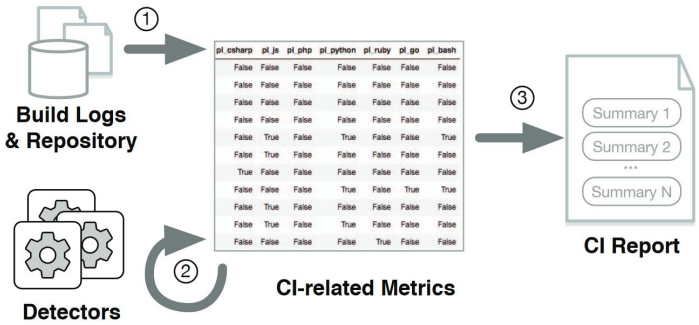
- Appear to be good or neutral, but are bad
- Slowly enter a project
- Hard for developers to see

# Outline

- Introduction
- Background
  - Github and version control
  - Continuous Integration (CI)
  - Anti-patterns and CI
- **Detection of anti-patterns with CI-Odor and developer's feedback**
- Detection of anti-patterns with Hansel and Gretel and developer's feedback
- Conclusion

# CI-Odor

- CI-Odor detects anti-patterns developers make during the development process
- Name similar to “code smells”
- Detects four anti-patterns
- Builds a report for the developers




Source: Vassallo (2019)

# CI-Odor: Deciding on Anti-Patterns

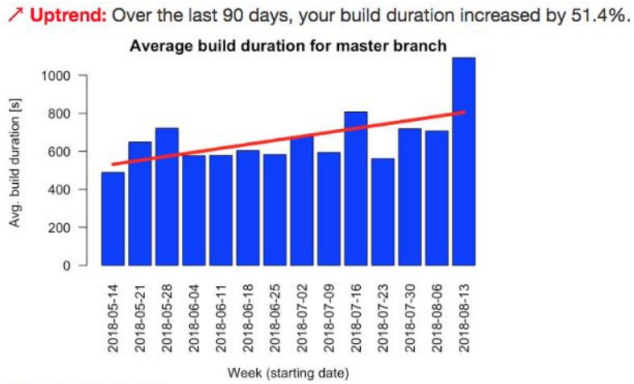
- Collected a list of anti-patterns from Paul Duvall
- Put out a survey that 144 developers responded to
- Decided on the 4 anti-patterns

# CI-Odor: What Are the Anti-Patterns?

- Increasing build times
  - Slows feedback to the team
- Late branch merging
  - Makes integrating harder
- Skipped failed tests
  - Removes safety of tests
- Broken release branches
  - Slows feedback on new feature

# CI-Odor: Detection

- CI-Odor uses build logs in the projects history for finding the anti-patterns
  - Slow build times, broken release branch, and skipped failed tests
- Late branch merging looks at the activity on the branches
- Slow build times and late merging of branches were given severity warnings




Source: Vassallo (2019)

# CI-Odor: Finding Projects

- Searched for projects that were active
  - Needed to use CI
  - Java and Maven
  - 5 members
  - Communication channel
- Got 36 projects that fit the criteria

# CI-Odor: Detection

- 8,520 detected incidents
  - 3,823 high severity
- Slow Build Times
  - 18,474 builds across the projects
  - 20 projects had increasingly slow builds
  - 3% of builds were high severity
- Broken Release Branches
  - 11.51% of release branches were broken

# CI-Odor: Detection

- Skipped Failed Tests
  - 15 of 36 projects had them
  - In only 56 of the builds
- Late Branch Merging
  - Affected 35 of 36 projects
  - Median of 2 for each
  - 115 high severity warnings

# CI-Odor: Developer feedback

- Sent out another survey
  - Sent to the teams of the 36 projects
  - Sent out on forums as well
  - Asked about the reports
- Heard back from 13 of the developers from 7 of the 36 projects
  - 8 agreed reports were useful, 3 disagreed, 2 neutral
  - Disagreed over project specifics
  - 5 were made curious

# CI-Odor: Developer Feedback

- 42 people responded overall (29 general devs) to questions about what they agreed with
  - Slow Build: 81.1%
  - Skip Failed Tests: 84.4%
  - Late Merging: 75%
  - Broken Release Branch: 86.4%
- Should CI-Odor break the build?
  - No, 59.6% disagree for slow builds
  - 65.9% disagree for late merges

# Outline

- Introduction
- Background
  - Github and version control
  - Continuous Integration (CI)
  - Anti-patterns and CI
- Detection of anti-patterns with CI-Odor and developer's feedback
- **Detection of anti-patterns with Hansel and Gretel and developer's feedback**
- Conclusion

# Hansel and Gretel

- Two tools to detect anti-patterns in the configuration files of Travis CI projects
  - Hansel detects them
  - Gretel removes them
- Made to determine 3 things:
  - How prevalent are anti-patterns in the configuration files
  - Can they be removed automatically
  - Will developers accept these changes

# Hansel and Gretel: Which anti-patterns

- The researchers browsed for anti-patterns
  - Looked over official Travis CI documentation
  - Used the TravisLint tool (only detects syntax errors)
  - Looked at community forums
- Decided on 4 anti-patterns
- Using Irrelevant Properties
  - Can be typos or old
  - Travis CI ignores them in runtime

# Hansel and Gretel: Which anti-patterns

- Bypassing Security Checks
  - Using SSH insecurely
- Redirecting Scripts into Interpreters
  - Downloading a script from URL
  - Can fail
- Commands Unrelated to the Phase
  - Misplaced commands cause maintenance issues
  - Travis CI optimized for commands under certain phases (slower builds)

```
install:
  - wget https://bit.ly/hj67 -O /tmp/casper.tar.gz
  - tar -xvf /tmp/casper.tar.gz
  - bower install

before_script:
  - export PATH=$PATH:$PWD/casperjs-1.0.2/bin/

script:
  - npm test
```

Source: Gallaba (2020)

# Hansel and Gretel: Finding and Removing

- Hansel's detection
  - Parses the configuration file for anti-patterns
  - Knows which phases have which commands
- Gretel's removal
  - Deletes the anti-pattern completely
  - Moves the command to where it should be
  - Makes a phase if it doesn't exist

# Hansel and Gretel: Finding Projects

- Found 9,312 projects that
  - Use Travis CI
  - Have at least 500 files
  - Aren't copies
- Ran Hansel and Gretel on the Projects
  - 894 had an anti-pattern (9.6%)

# Hansel and Gretel: Detection

- Redirecting Scripts into Interpreters
  - 206 instances
- Bypassing Security Checks
  - 63 instances
- Using Irrelevant Properties
  - 242 insances
- Commands Unrelated to the Phase
  - 519 commands in unrelated phases

| Expected in \ Observed in | Install | Script | Deploy |
|---|---|---|---|
| **Install** | - | 467 | 0 |
| **Script** | 0 | - | 0 |
| **Deploy** | 0 | 52 | - |

Source: Gallaba (2020)

# Hansel and Gretel: Detection Rate

- 100 of the 9,312 projects were chosen for manual review
  - The researchers checked them for anti-patterns
  - Hansel detected 24 of the 29 anti-patterns (82.76%)
  - 3 of them were edge cases for script redirection
  - 2 were missed bindings (composer tool also goes to install phase)

# Hansel and Gretel: Removal

- Selected 250 anti-patterns from the original projects
  - Chose until adding more wouldn't add anything to the test
- Used TravisLint to make sure the configuration files were correct
  - Was run before and after using Gretel
- 174 of the 250 were able to be removed automatically
  - 69 of the 76 require manual verification
  - The remaining 7 use yarn, which Gretel didn't support

```
language: node_js
node_js:
- '0.10'
before_script:
- cd frontend
- npm install -g
    bower grunt-cli
- npm install
- bower install
script:
- grunt test
```

```
language: node_js
node_js:
- '0.10'
install:
- cd frontend
- npm install -g bower
    grunt-cli
- npm install
- bower install
script:
- grunt test
```

Source: Gallaba (2020)

# Hansel and Gretel: Developer Feedback

- Sent out 174 pull requests with Gretel fixes
  - 49 received responses, 36 accepted and integrated
- Rejection reasons for the 13
  - 2 rejected for build breaks
  - 2 didn't want the commands moved
  - 2 were projects no longer developed
  - 1 because developer disagreed
  - 6 received no feedback on rejection

# Outline

- Introduction
- Background
  - Github and version control
  - Continuous Integration (CI)
  - Anti-patterns and CI
- Detection of anti-patterns with CI-Odor and developer's feedback
- Detection of anti-patterns with Hansel and Gretel and developer's feedback
- **Conclusion**

# Conclusion

- Have gone over the use of anti-pattern detection tools
- The two discussed found different anti-patterns
  - Observation: one anti-pattern could lead to another
- Developers liked being alerted to anti-patterns
- Developers accepted the changes suggested from Gretel (73.47%)
- Future work

# Acknowledgements

Thanks to my advisor Kristin Lamberty for her help during my research process.

# References


- Keheliya Gallaba and Shane McIntosh. 2020. Use and Misuse of Continuous Integration Features: An Empirical Study of Projects That (Mis)Use Travis CI. IEEE Transactions on Software Engineering 46, 1 (2020), 33–50.
- Carmine Vassallo, Sebastian Proksch, Harald C. Gall, and Massimiliano Di Penta. 2019. Automated Reporting of Anti-Patterns and Decay in Continuous Integration. In Proceedings of the 41st International Conference on Software Engineering (Montreal, Quebec, Canada) (ICSE ’19). IEEE Press, 105–115.

# Questions?