Contract cheating in Higher Education

Findings from a Survey of Australian Students and Staff

TOny maSon,
aDA gAVRILOVSKA
DaVId a. joyneR

This Strategic Priority Project (SP16-3283) is supported by the Australian Government Department of Education and Training

contract cheating and assessment design

EXPLORING THE CONNECTION

Source: *https://www.acode.edu.au/mod/resource/view.php?id=676*

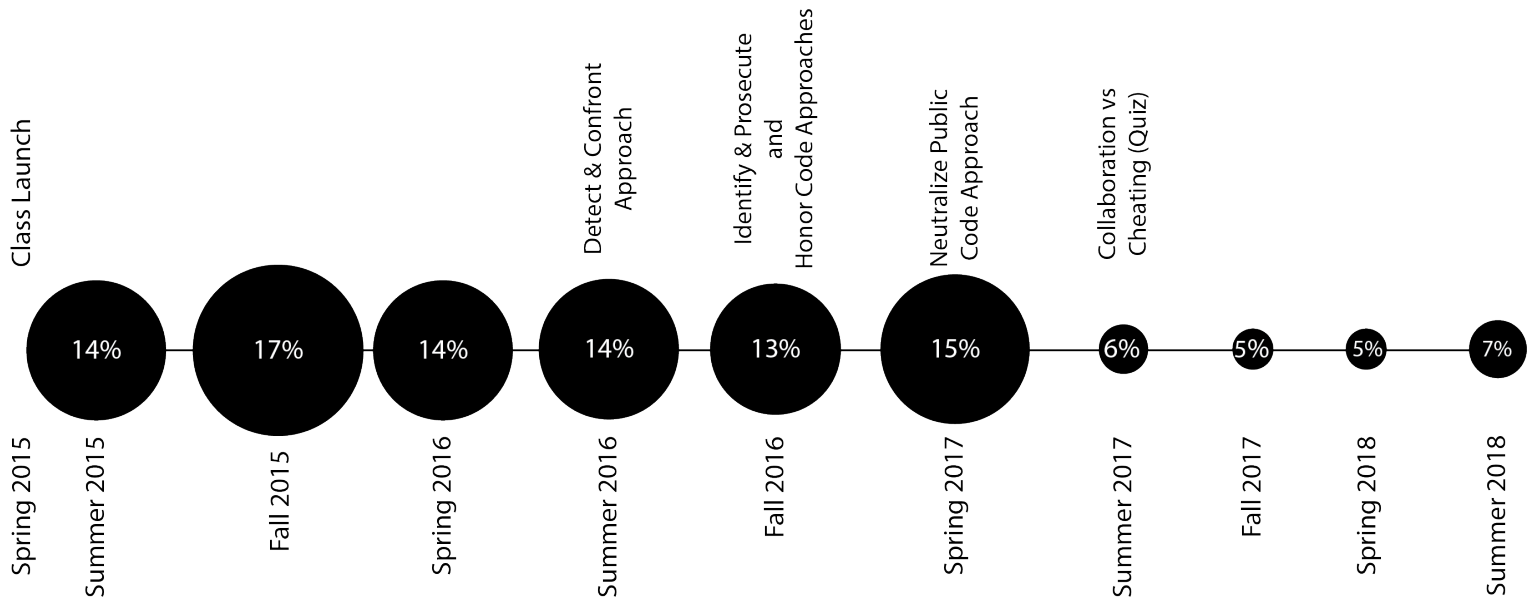
Georgia Tech
CREATING THE NEXT

Collaboration versus Cheating

Tony Mason, Ada Gavrilovska, David A. Joyner

February 1, 2019

We reduced Detected Plagiarism Rate



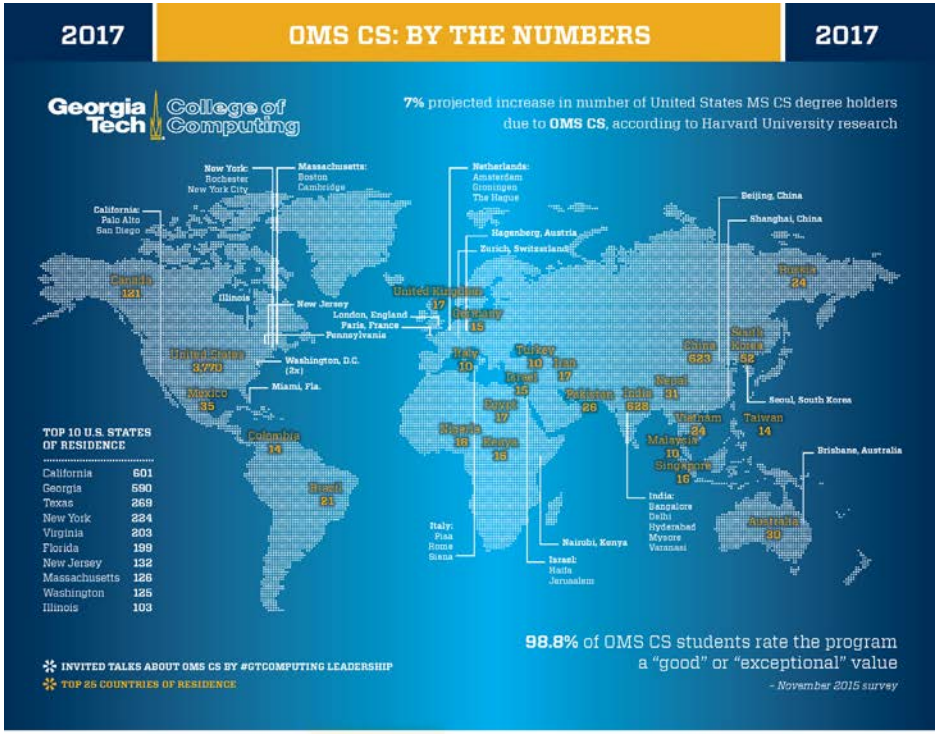
Academic Honesty

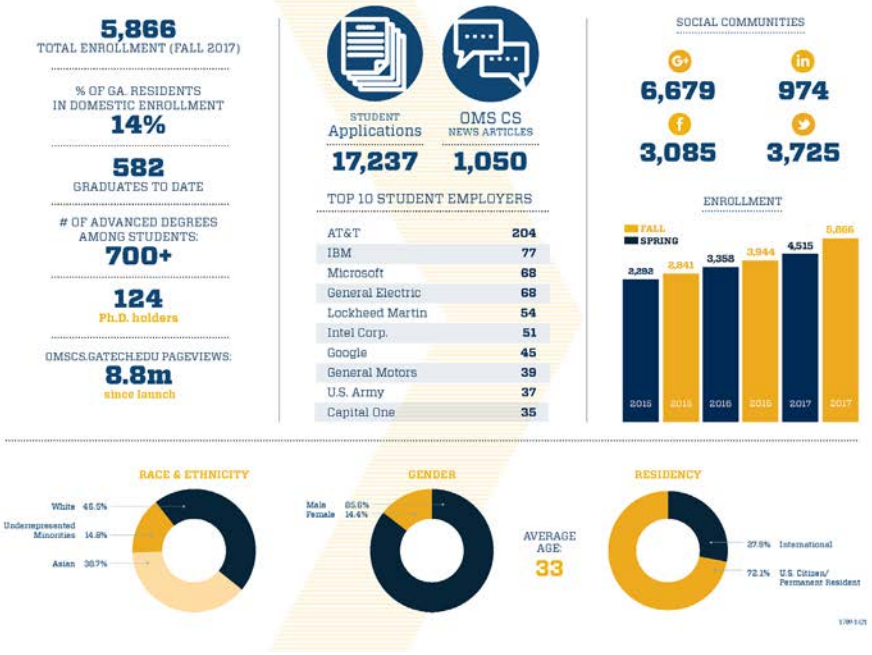
*As a society, we rely on the **academic** and journalistic **integrity** of other people's work. The **whole point of academic research** is to share knowledge with others and learn from one another. Since knowledge and ideas are the primary product produced by academic communities, it is **essential that this knowledge is accurate** and gives credit to those who created it.*

University of Ontario. 2016. Why is Academic Integrity and Honesty Important ?
https://secure.apa.uoit.ca/academic_integrity/module1/Module13.html

OMSCS

Online Master of Science in Computer Science





CS 6200

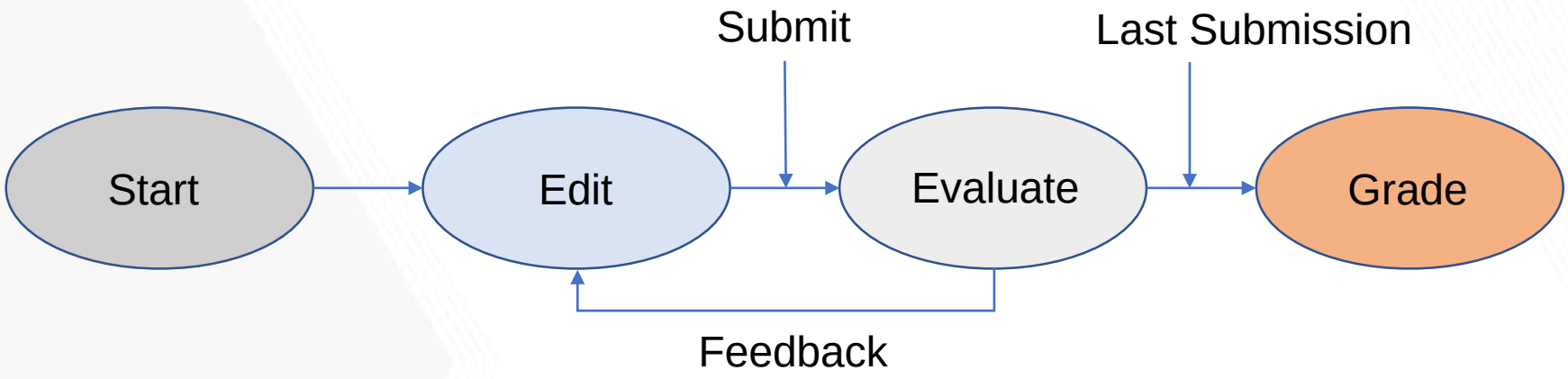
Graduate Introduction to Operating Systems

Often the first course in the program for new students

Challenging, requires **C** programming ability

High drop rate – approximately 40%

Project Workflow

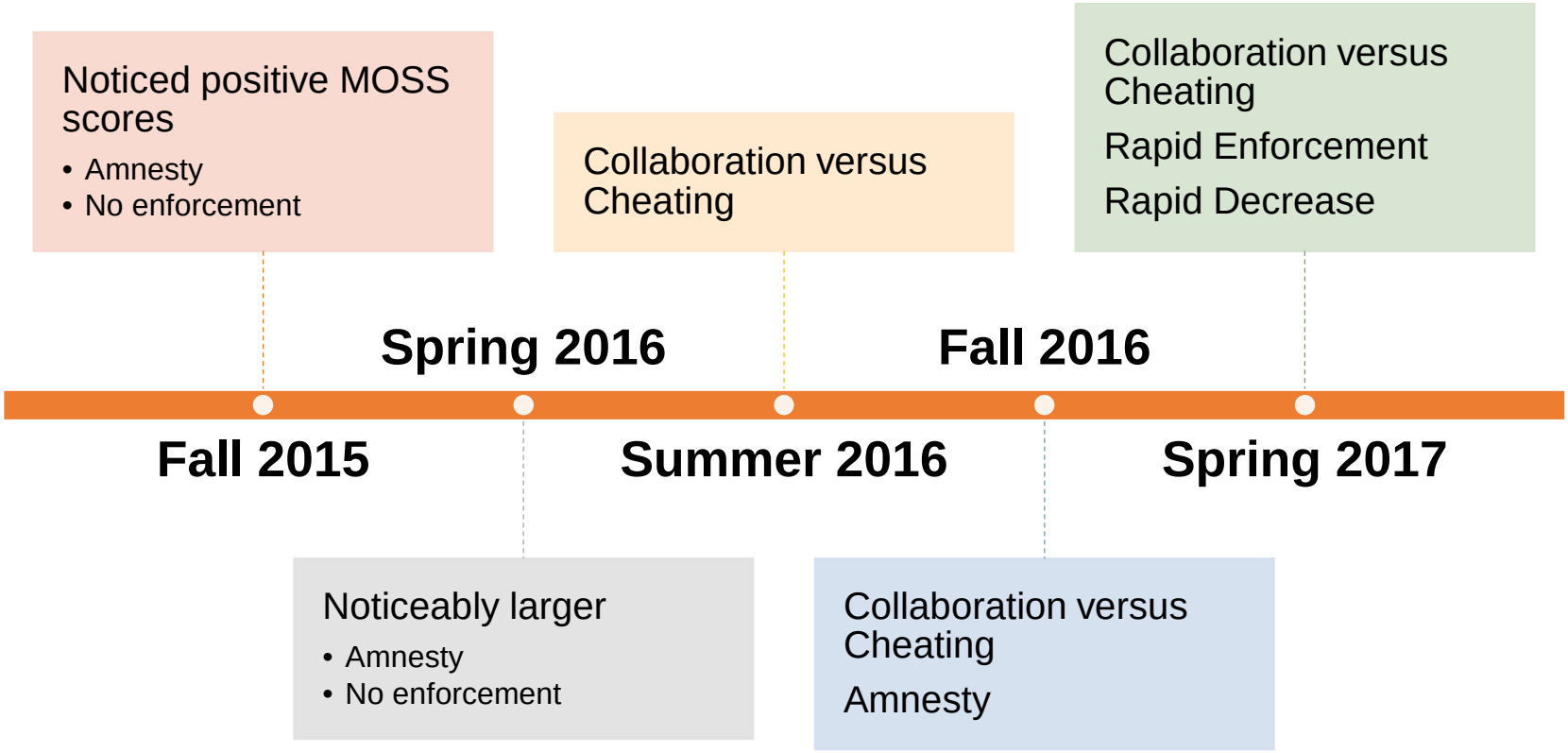


Automated grading using Docker Containers in AWS

Feedback is given (but not grades) to students

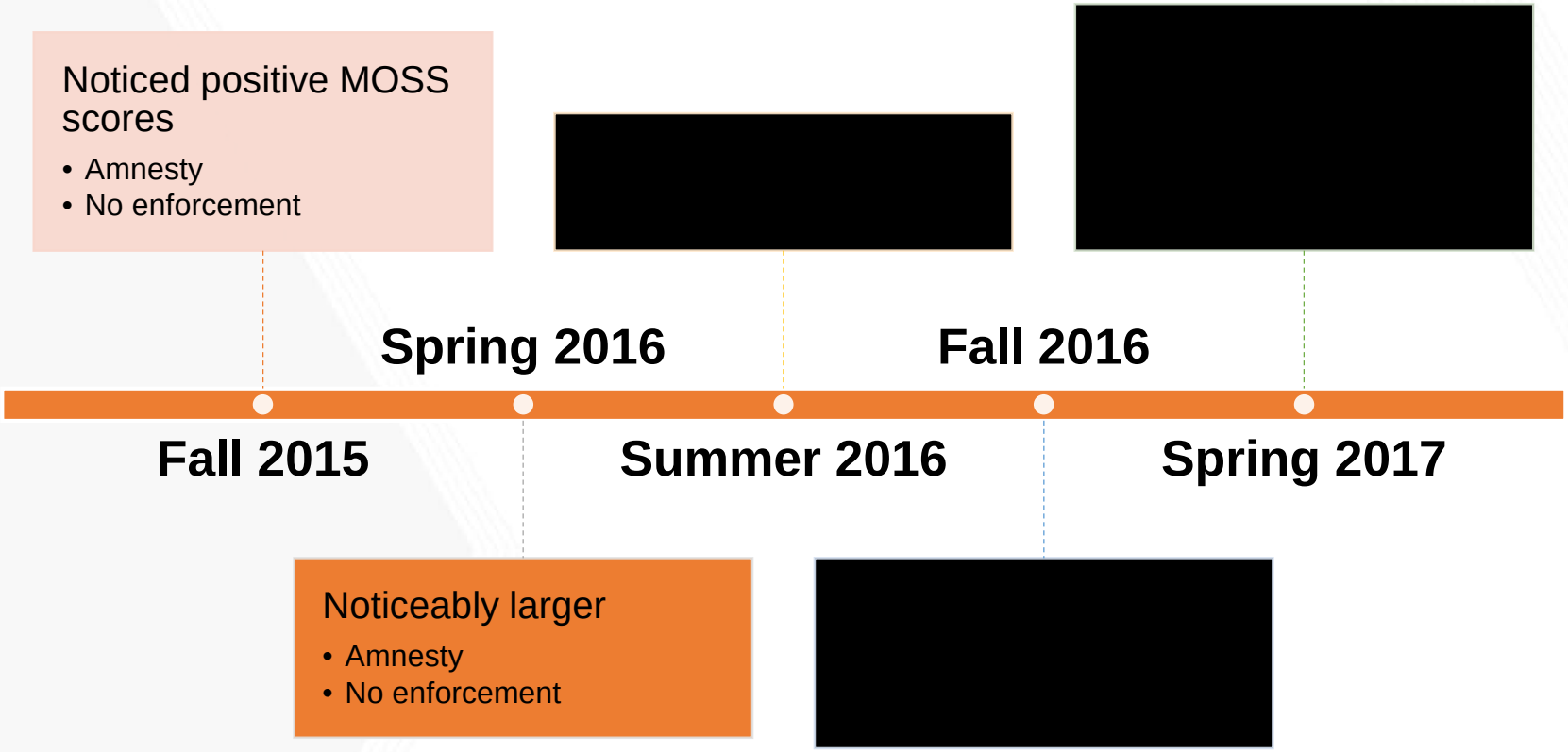
All submissions are saved; final submission prior to deadline is graded

Two major solo projects (no groups)



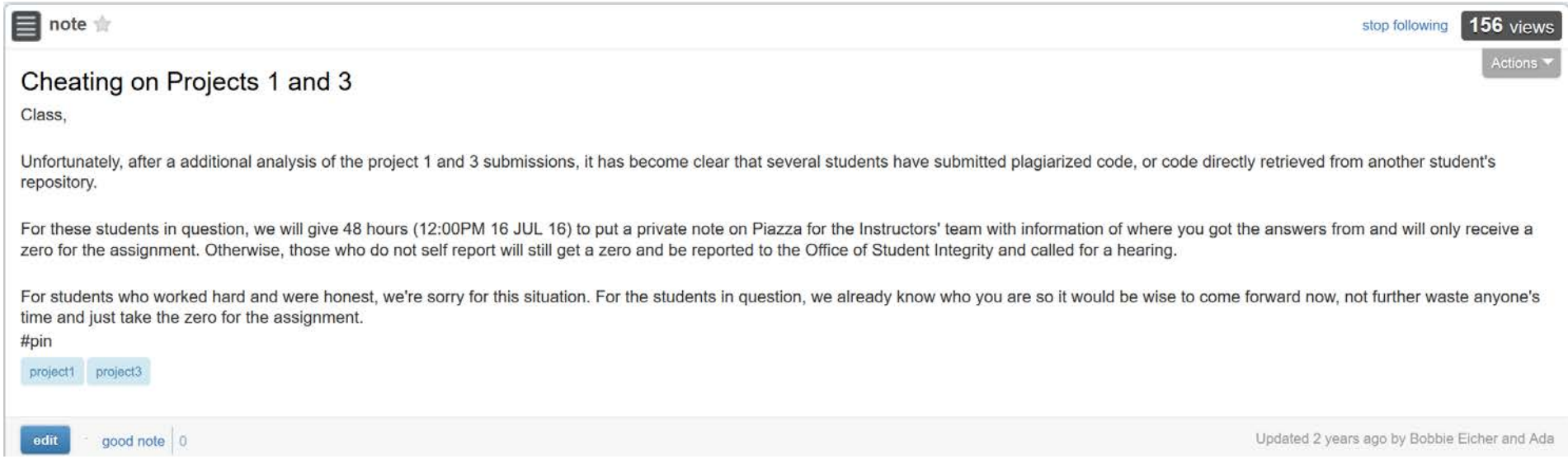
Increasing Plagiarism

What we tried



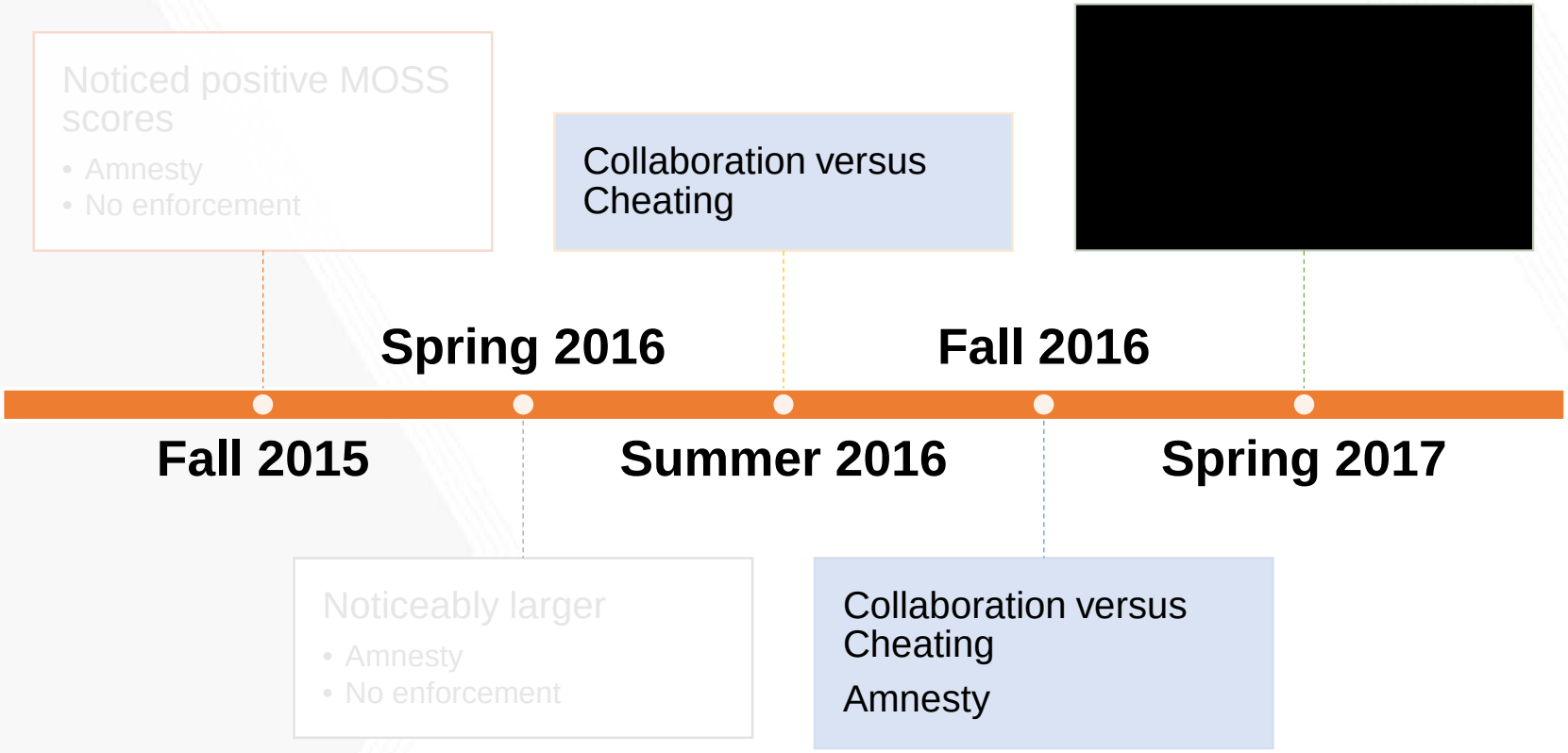
Increasing Plagiarism

Georgia Tech
CREATING THE NEXT



Amnesty

- Not **one** student we suspected came forward
- **Multiple** students came forward but had not cheated.



Increasing Plagiarism

Georgia Tech
CREATING THE NEXT

Collaboration versus Cheating

Our goal in CS-6200 is to provide you with a great opportunity to learn more about operating systems. Part of that is a series of projects that you will be asked to implement in the class.

Collaboration is a very good thing. On the other hand, cheating is considered a very serious offense and is vigorously prosecuted. Vigorous prosecution requires that you be advised of the cheating policy of the course before the offending act.

If you obtain help of any kind, always write the name(s) of your sources in the project README file.

This document describes what is acceptable collaboration versus unacceptable plagiarism for this class. Note that ultimately, this is an essential part of the Georgia Tech Honor Code.

Collaboration

Collaboration is essential to both the learning experience in this class as well as in the *real* world. Collaboration is working with other people cooperatively to enhance understanding.

We encourage you to:

- Share ideas
- Explain your code to someone to see if they know why it doesn't work.
- Help someone else debug if they've run into a wall.

Ideally, we'd like to see you do this via Piazza, in public threads. That creates transparency and allows other students to also participate in the conversation as well as learn from it.

Plagiarism

Plagiarism is using someone else's work instead of doing your own work:

- Never share code or text on the project.
- Never use someone else's code or text in your solutions.
- Never consult project code or text that you find anywhere, such as the Internet

The one exception here is that you may use small snippets of code for commonly performed tasks (e.g., creating a socket, or translating a host name to an IP address), provided that you properly reference the usage and explain what the code does in your README file.

Note: a "snippet" is less than 10 lines of code. If we find you used 11 lines of code once in your project, we aren't likely to say anything. But let me show you a real-world example of this:

What do you think the odds are that two different students would write exactly the same code? Here's an experiment for you: try rewriting code that you previously have written without looking at that older code. Then compare them. If you can't write the same code twice, what do you think the odds are that someone else will.

That image is from one of the tools that we use in this course. It is not the only mechanism that we use, it just happens to be the most well known of them. There's even a public github that discusses techniques for bypassing it. https://github.com/gencheng1234/How-to-cheat-in-computer-science-101

Here's the interesting bit of it - If you actually do all the work necessary to escape detection, by the time you're done, you'll have done more work than if you had just done your own work.

Here's one of the best descriptions about cheating (and why every excuse doesn't work) I've ever read: http://www.cs.ubc.ca/~lmm/courses/cheat.html

Here's an amazing description of why this matters: https://secure.sps.ucf.ca/academic_integrity/module1/Module13.html

If in doubt, ask.

ıst efforts, we sometimes find students cheating. While this comes in a number of forms, we do not tolerate it because it is not
help you learn the material and cheating does not achieve that goal.

ılity for cheating is that you deprive yourself of the learning opportunity. In addition to disappointing yourself, you also disappoir
ıe material.

ıred as scholarly misconduct. The actual penalty for cheating may consist of not receiving any points on the assignment, fa
es to your status as a Georgia Tech student, including expulsion from the program. The details are on the OSI website.

want to do - but we have done it in the past. Nobody wants to face a formal review board hearing, presented with evidence of

ecause if we don't, OMSCS gets a poor reputation and that diminishes the value of the program (and the degree) for the vast r

folks: this is our Nash Equilibrium. We're trying to be transparent about it because we want *your* Nash Equilibrium to be not

ıstions, please ask us.

e else is cheating, alert us immediately.

cheated, come forward. We will be fair in working it out with you. We're far less forgiving when we have to come to you wit

ts on ~ 5% of the class last semester. If you can find a repository with code from a prior semester, please know that I have a
class since its inception and the process of checking is now automated. I'd much rather spend the time helping you understan

roject4 logistics

Establishing Expectations

Collaboration versus Cheating

Our goal in CS-6200 is to provide you with a great **opportunity to learn** more about operating systems. Part of that is a series of projects that you will be asked to implement in the class.

Collaboration is a very good thing. On the other hand, cheating is considered a very serious offense and is vigorously prosecuted. Vigorous prosecution requires that you be advised of the cheating policy of the course before the offending act.

If you obtain help of any kind, **always** write the name(s) of your sources in your project report.

This document describes what is acceptable collaboration versus unacceptable plagiarism for this class. Note that ultimately, this is an essential part of the **Georgia Tech Honor Code**.

Plagiarism is using someone else's work instead of doing your own work:

Never share code or text on the project.

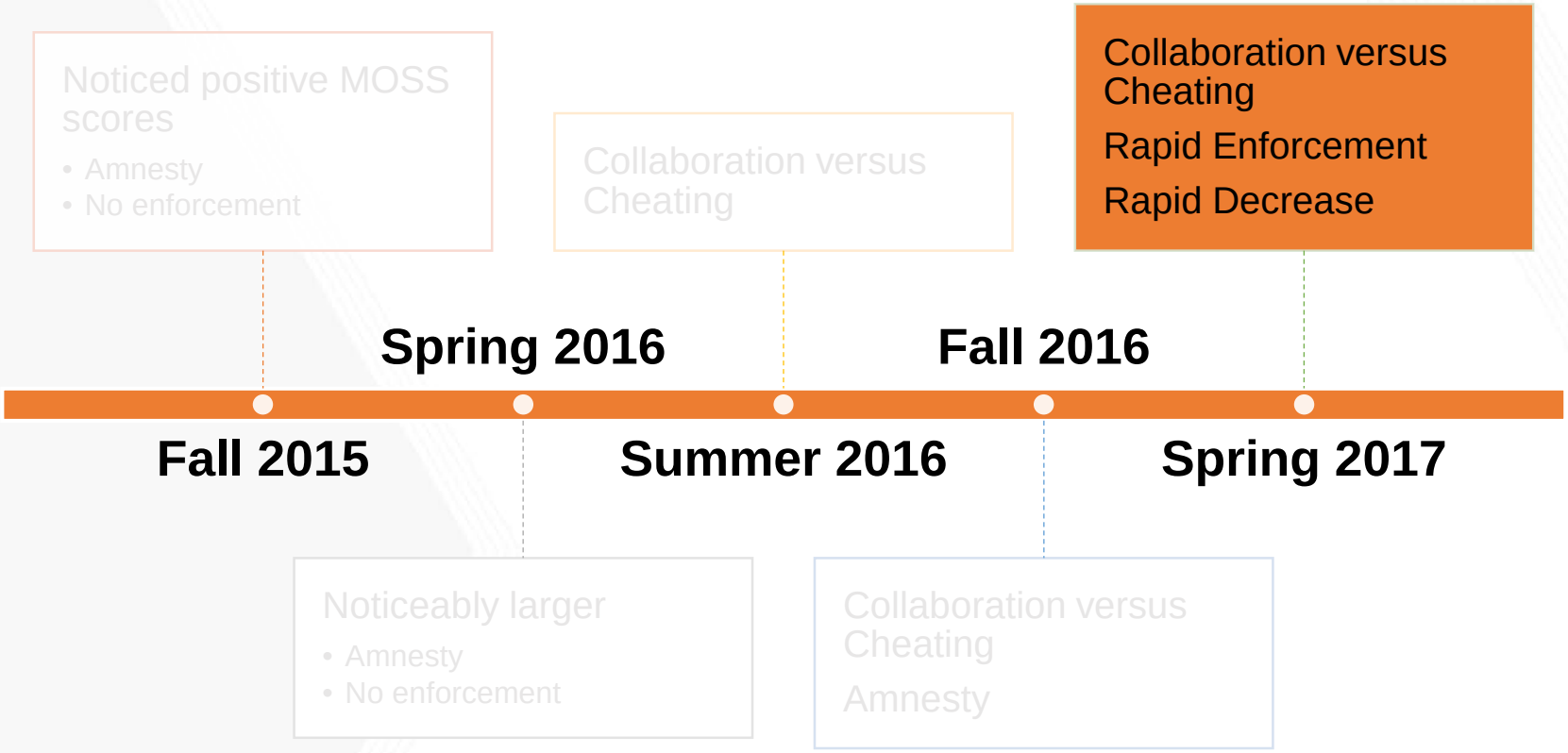
Never use someone else's code or text in your solutions.

Never consult project code or text that you find anywhere, such as the Internet

Cheating is considered as scholarly misconduct. The actual penalty for cheating may consist of not receiving any points on the assignment, failing the class, or may have more serious consequences to your status as a Georgia Tech student, including expulsion from the program. The details are on the OSI website.

This is **not** what we want to do - but we have done it in the past. Nobody wants to face a formal review board hearing, presented with evidence of cheating and forced to defend their actions.

Why do we do it? Because if we don't, OMSCS gets a poor reputation and that diminishes the value of the program (and the degree) for the vast majority of students that don't cheat.



Increasing Plagiarism

Georgia Tech
CREATING THE NEXT

Enforcement

Spring 2016

- Project 1 – Rapid contact/enforcement
- Project 2 – 0% measured plagiarism rate
- Enforcement is time-intensive

Goal: find a less time-intensive mechanism

The Quiz

Quiz Instructions

In an effort to ensure you understand the policy on collaboration versus cheating, please confirm that you have read and understood the following messages.

Question 1 — 1 pts

From: "Galil, Zvi" <galil@cc.gatech.edu>

Subject: [Oms-fac] Note for OMS students

Date: March 14, 2017 at 11:45:51 AM EDT

To: "omscs-official@cc.gatech.edu" <omscs-official@cc.gatech.edu>

Hello OMS CS students,

Most of my messages to you are about great news concerning the College or OMS CS. Unfortunately today's note is a bit more serious, and it's one I hope all of you read and take to heart. It's come to my attention that incidents of plagiarism are rising among our OMS student population, and as a community we must not only reverse this trend but eliminate it. It should go without saying that when you present someone else's work as your own, you are violating Georgia Tech's **Academic Honor Code (https://policylibrary.gatech.edu/student-affairs/academic-honor-code)** , and this will not be tolerated. It could even impact your career.

One particular source of concern is when students post their assignments publicly on Github (or similar services) and leave them there—even after they graduate. Other students find this code, and some have copied it for their own assignments. All of you should be aware that we use sophisticated software to compare your work to what is "out there" on the web, and it is as easy for us to detect plagiarism of this kind as it is dangerous for you to engage in it.

The simple message is: Don't do it. And for the students who utilize Github as a professional portfolio, you should know that if an investigation reveals your code has been used in someone else's assignment, you will be implicated in the situation and asked questions. If you'd like to avoid this headache, consider limiting access to your Github repository.

Tomorrow I'll send a much more positive message about more great press OMS CS has received recently. The reason we get this good coverage is that Georgia Tech and OMS CS enjoy a sterling academic reputation, and one cornerstone of that reputation is academic integrity. Please help us maintain and nurture that reputation.

Thanks for your help!

Zvi Galil

John P. Imlay Jr. Dean of Computing

Please confirm that you have read and understood the above statement from Dean Galil:

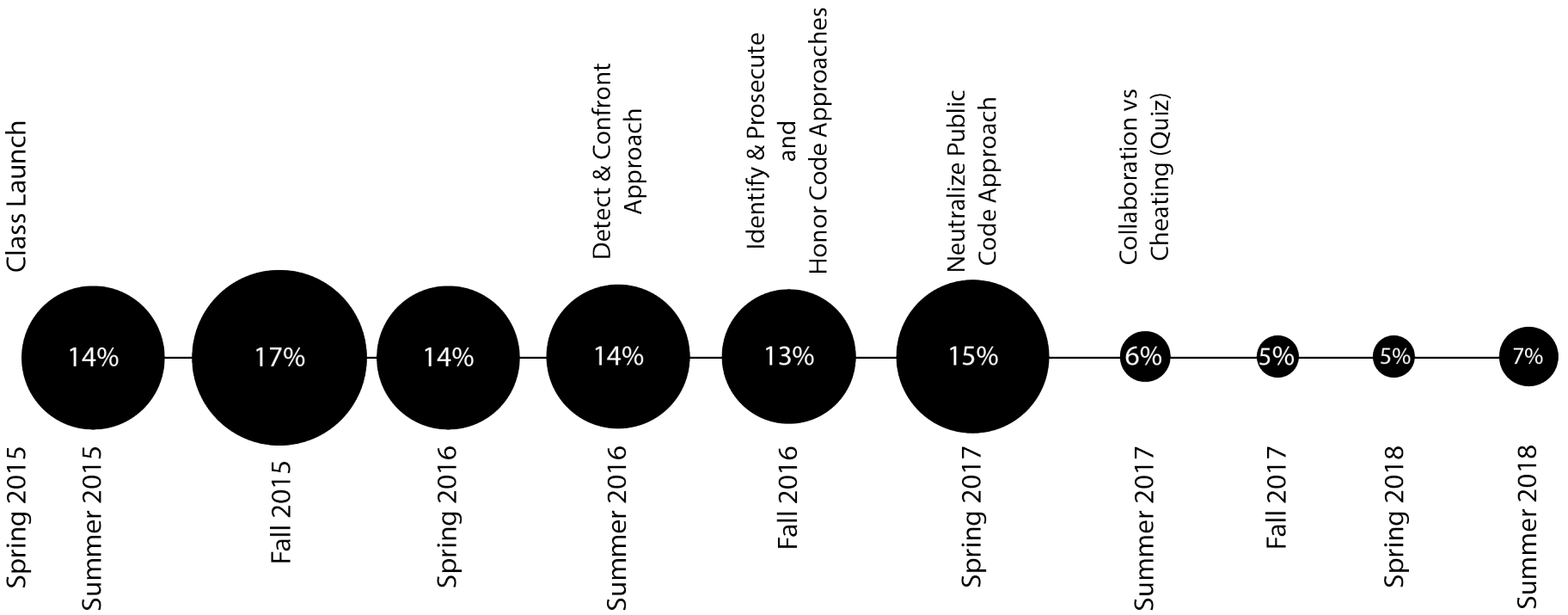
- ○ I confirm
- ○ I do not confirm

Question 2 — 1 pts

Please confirm that you have read and understood the clarification on Collaboration vs. Cheating posted in Piazza post @11 (https://piazza.com/class/jh58qa7wyb92kh?cid=11) .

- ○ I confirm
- ○ I do not confirm

It worked!


Class Launch
Detect & Confront Approach
Identify & Prosecute and Honor Code Approaches
Neutralize Public Code Approach
Collaboration vs Cheating (Quiz)
14%
17%
14%
14%
13%
15%
6%
5%
5%
7%
Spring 2015
Summer 2015
Fall 2015
Spring 2016
Summer 2016
Fall 2016
Spring 2017
Summer 2017
Fall 2017
Spring 2018
Summer 2018

Evaluation

Plagiarism: MOSS similarity score of > 30%

Project 1 has four distinct components

Project 2 has two distinct components

Enforcement Revisited

Fall 2018 – resumed rapid enforcement

- Improved automation
- Scripted MOSS anonymization, report generation

Fewer cases = lower burden

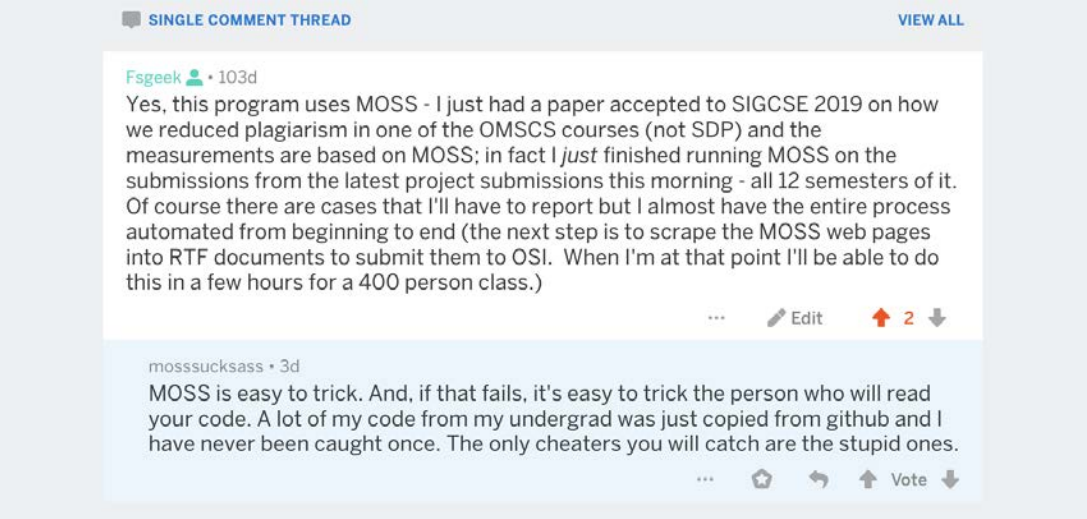
Delayed enforcement = ineffective

Future Work

Beyond MOSS

What more can we do to detect plagiarism?

How to detect false negatives?



CS 6200 Detection Improvements (1)

Automatically build git repo from submission history

- Each of the past 50 submissions are saved
- Time series analysis is useful for enforcement

Automate evaluating checksums

- Each submission contains file checksums
- Detects submission script tampering
- Detects prior semester code used

CS 6200 Detection Improvements (2)

Improve "take down"

- Starter code now includes a LICENSE file

Automate analysis of watermarks

- Starter code is modified each semester
- Change constants, order of headers, order of options
- Change function signatures

Contract Programming

Contracting out Homework is another issue

- More difficult to detect
- Prior work

Using AI techniques: *Jack Watson*

- Evaluate projects on contract websites
- Pose as contractors
- Other?

Note: this is ongoing research

Summary

Clearly explain class policy

- What constitutes plagiarism
- Penalties of getting caught

Include an assessment

Statistically significant decrease in detected plagiarism


Questions?

About the Authors

Tony Mason is an Instructional Associate for the OMSCS program as well as a PhD Student in Computer Science at the University of British Columbia (fsgeek@{gatech.edu,cs.ubc.ca})

Ada Gavrilovska is an Associate Professor in the College of Computing at Georgia Tech and the Instructor for CS6200.

David A. Joyner is the Associate Director of Student Experience at Georgia Tech's College of Computing.