Typing Exercises as Interactive Worked Examples for Deliberate Practice in CS Courses

Adam M. Gaweda, Collin F. Lynch, Nathan Seamon, Gabriel Silva de Oliveira, Alay Deliwa
North Carolina State University

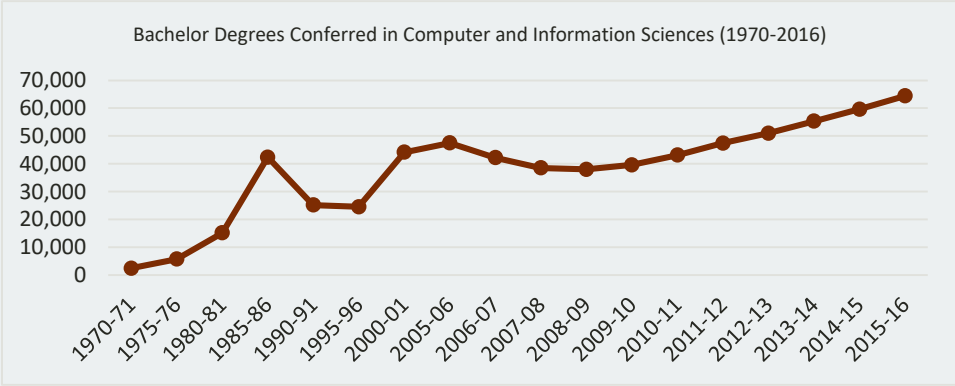
Slides: go.ncsu.edu/typos-ace2020

CS Popularity and Attrition

Computer Science enrollment is currently at **an all-time high** in the US (Bureau of Labor Statistics, 2017)

However, early CS courses commonly experience a **30-50% drop/fail** attrition rate

Novice CS students struggle with **syntax errors**, rather than problem-solving (Altadmri, Brown, 2015)



Worked Examples and Templates

Computer Scientists store **recurring basic plan** mental models to apply a learned concept to a similar problem (Soloway, Ehrlich, 1984)

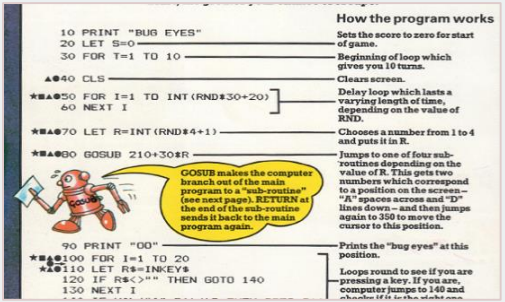
Debugging can take the majority of novices' time but can be mitigated with **canned code** (Brooks, 1977)

Multiple **worked examples** improve students' abstract mental models (Zhi et al, 2018)

Novel CS Exercises

The emergence of **novel CS exercises** has created exercises like Parson's Puzzles, Output Prediction, and Bug Localization

Early computer magazines provided worked examples needed to be retyped for making computer games (Usbourne, 1982; Antic, 1982-90)



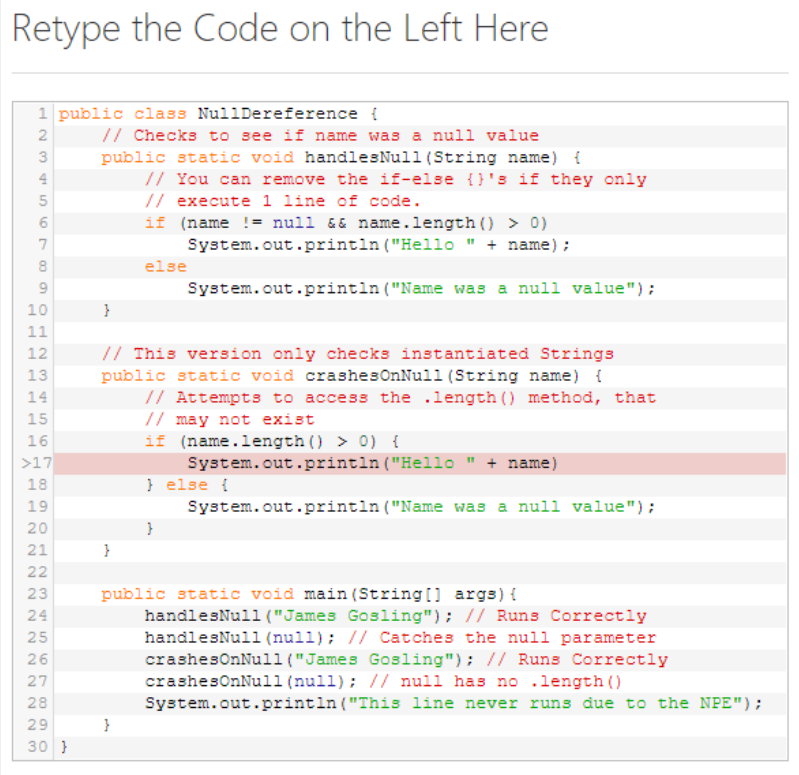
Computer Space Games, 1982

Antic, 1985

Typing Exercises

CS novices can benefit from **retyping example code** to build their own recurring basic plans without the risk of simply **copying and pasting** the code

Typing exercises give students canned code, which can be used as a **template for future exercises**



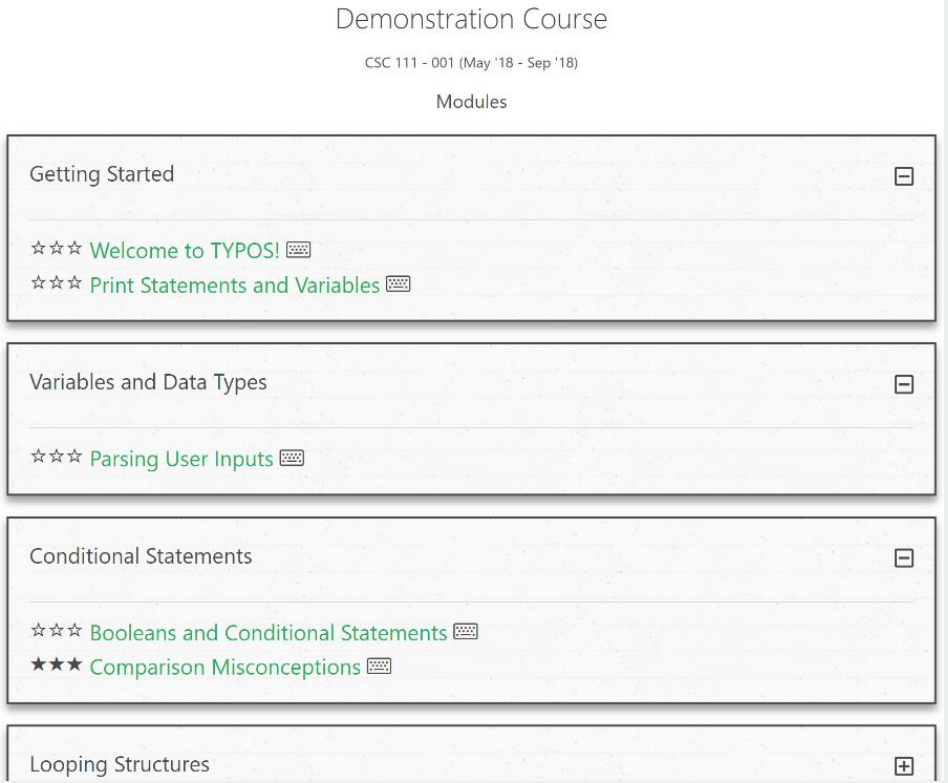
TYPOS, 2017

TYPOS

TYPOS is a low-level CS Exercise Platform

Students could **complete exercises** as often as they wanted and **download copies** after completion

Information and Images Available at go.ncsu.edu/typos



Research Questions

RQ1 - Do students refer to typing exercises while working on other tasks?

RQ2 - How do different patterns of use correlate with student performance?

RQ3 - Is there a correlation between completion of typing exercises and build failures on assignments?

Study

TYPOS was deployed into **three offerings** of a second semester CS course which taught Java

Homework and Projects were graded through unit and code coverage tests on the Jenkins Continuous Integration system

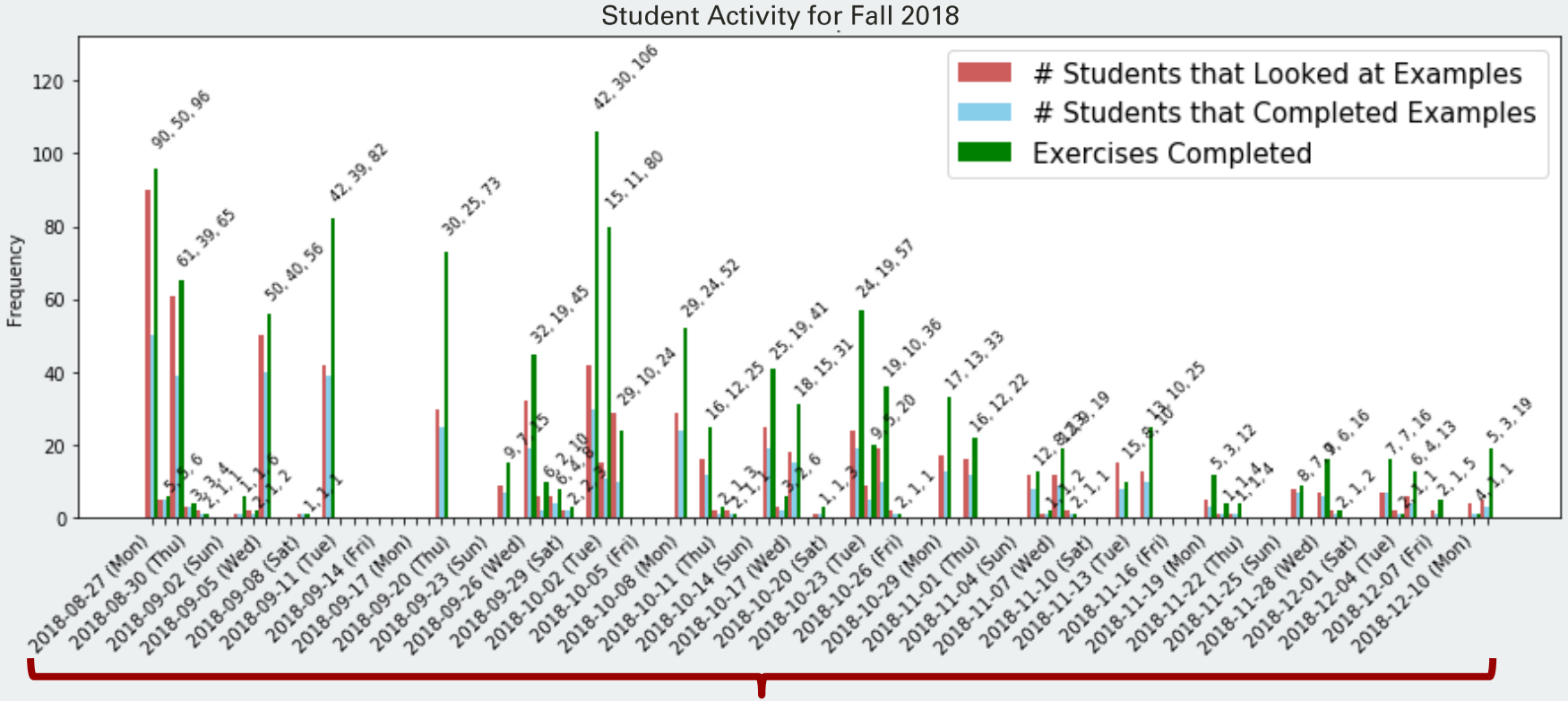
- Comments were graded by TAs

3 - 4 optional typing exercises were released each lecture mirroring the day's topic

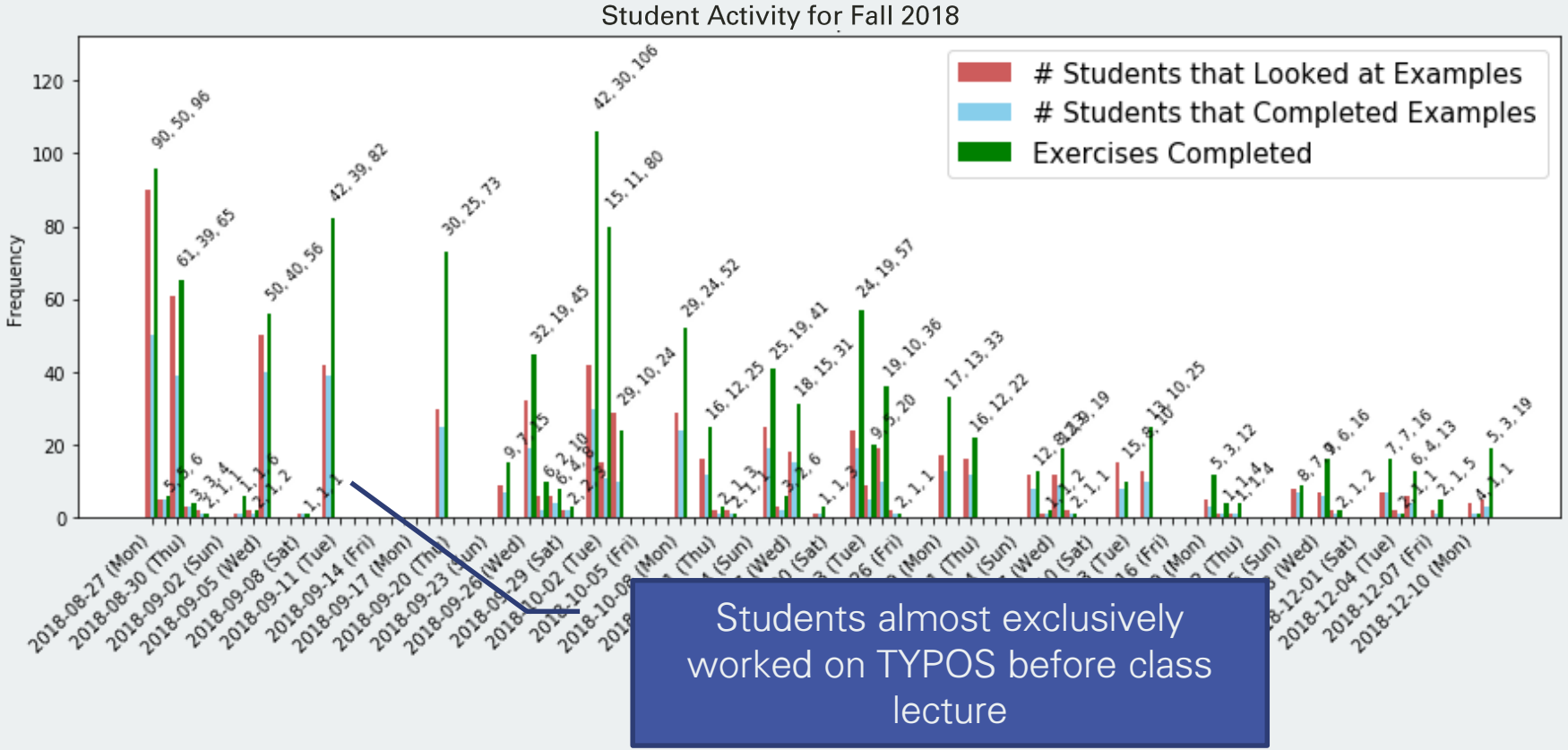
- **Total Number of Exercises Released per Offering:** 66

In total, 337 students made 7,334 views, 10,097 attempts, and 3,024 completions

Usage Analysis

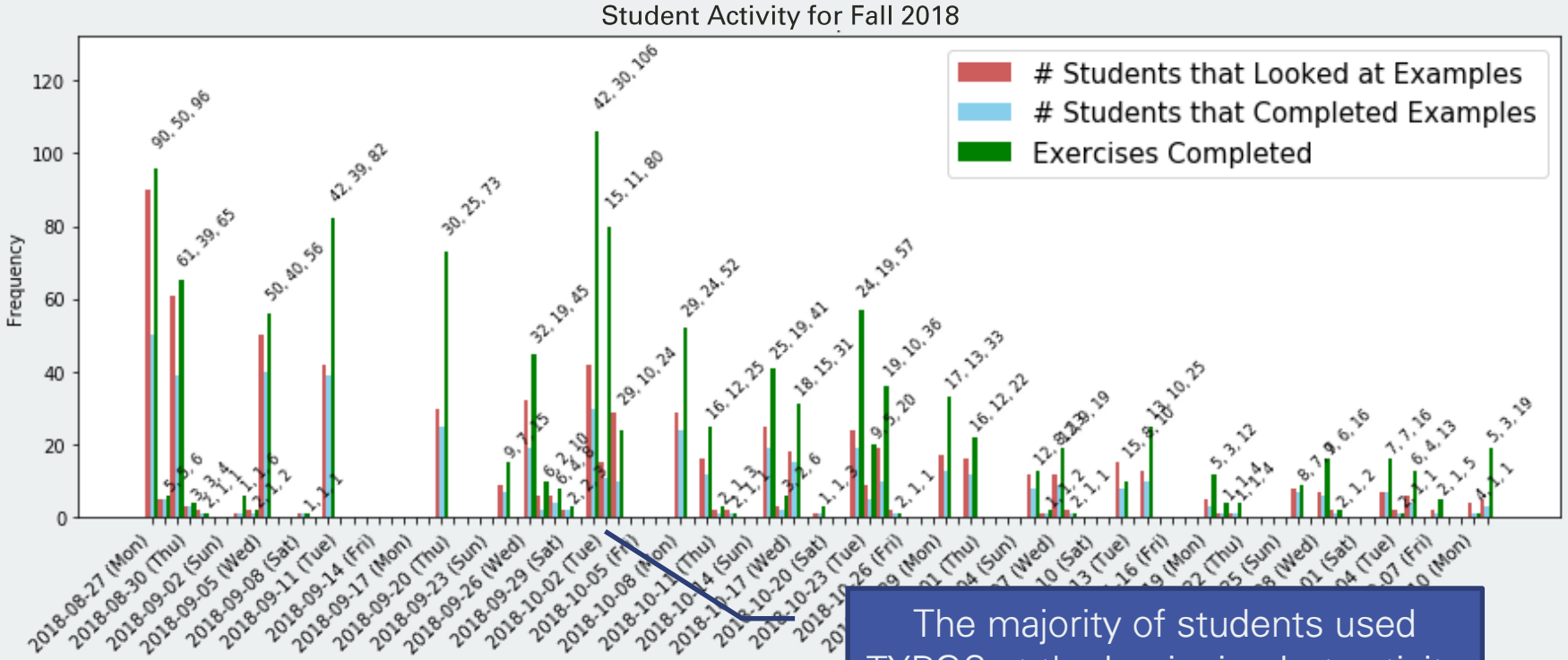


Usage Analysis



Students almost exclusively worked on TYPOS before class lecture

Usage Analysis



The majority of students used TYPOS at the beginning but activity shifted to prior to exams

Usage Analysis

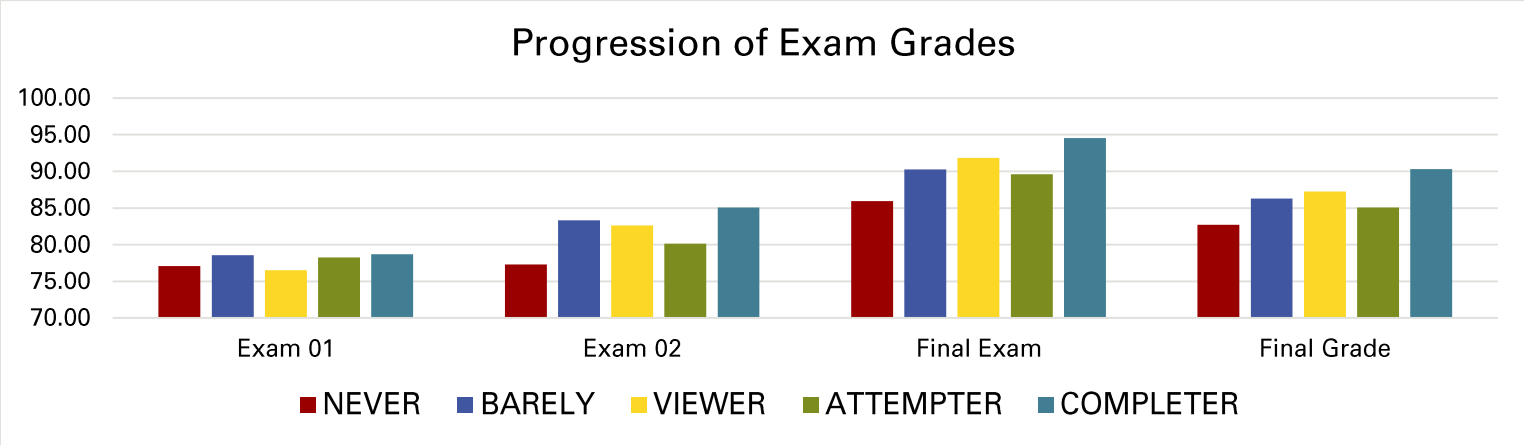
Active Users (77) comprised 75% of all activity on TYPOS

Category	Total	A	B	C	D	F	Avg Views	Avg Attempts	Avg Completes
Never	25	11	8	2	-	4	0	0	0
Barely	235	109	85	21	3	17	7.5	12.5	3.1
Viewers	17	9	4	3	-	1	58.1	13.7	3.4
Attempters	26	11	11	2	-	2	60.2	56.2	18.0
Completers	34	19	14	1	-	-	88.4	160.8	52.1

Active Users were classified as **viewers**, **attempters**, or **completers**

Student Performance Analysis

Regular completers **earned higher exam grades and final course grades** when compared to other regular users



Student exam and course grades by category

Student Performance Analysis

Initially **low performing completers benefited** from the additional low-level practice

Category	Total	Course Grades	Exam Learning Gains
Active Completers	34	0.065	0.161
Initially Low-Performing Students	16	**0.038***	**0.041***

P-values for all and lower-performing completers across course grade and learning gains (* $p<0.05$)

Assignment Implementation Analysis

This course used the **Jenkins** Continuous Integration system to assess student submissions with three feedbacks

Red Ball

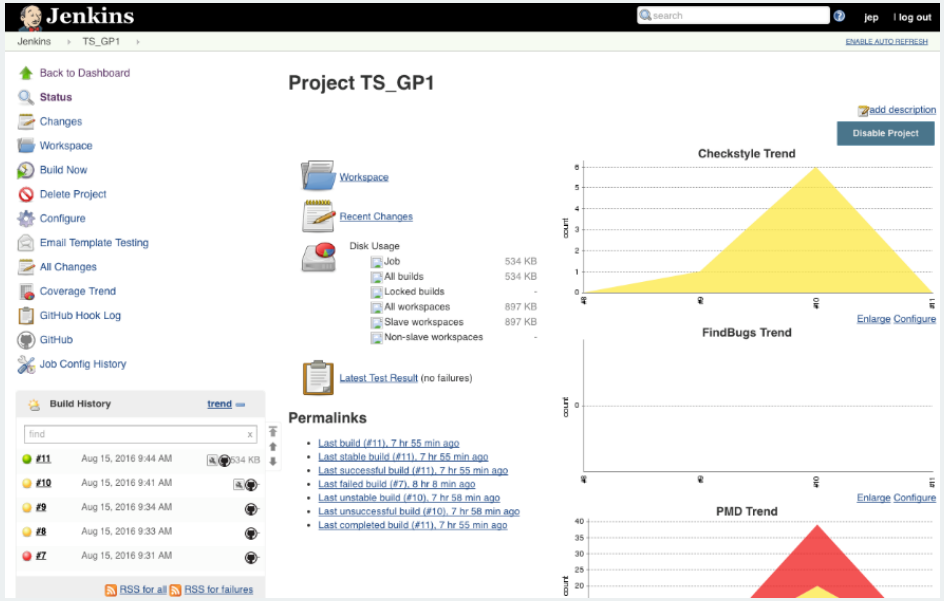
Build could not compile

Yellow Ball

Build compiled, but failed tests

Green Ball

Build successfully completed all tests



Assignment Implementation Analysis

We assessed all active users TYPOS behavior with Jenkins' red ball status

All TYPOS usage was **negatively correlated** with build failures

Category	# Viewed	# Attempted	# Completed
Active Students	-0.045**	-0.048**	**-0.110****
Initially Low-Performing Students	**-0.132***	-0.050**	**-0.144****

Pearson's r Comparing TYPOS Activity to Build Failures (* p<0.05, ** p<0.01)

The more exercises completed, the lower number of build failures

Conclusions

RQ1 - How do different patterns of use correlate with student performance?

Regular completers earned higher course grades, especially in initially low-performing students

RQ2 - Do students refer to typing exercises while working on other tasks?

Students reviewed TYPOS prior to exams and completed exercises prior to lecture

RQ3 - Is there a correlation between completion of typing exercises and build failures in other work?

Reviewing and retyping worked examples reduced the number of build failures on Jenkins

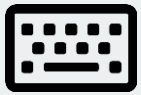
Future Work

In a perfect world, everyone would master topics immediately.

 Slides: go.ncsu.edu/typos-ace2020

Future Work


Lecture Video Introducing Topic


Typing Exercise


Self Explanation


Fix the Bug


Find the Bug


Output Prediction


Coding Exercise


Parson's Puzzles


Fill in the Blank

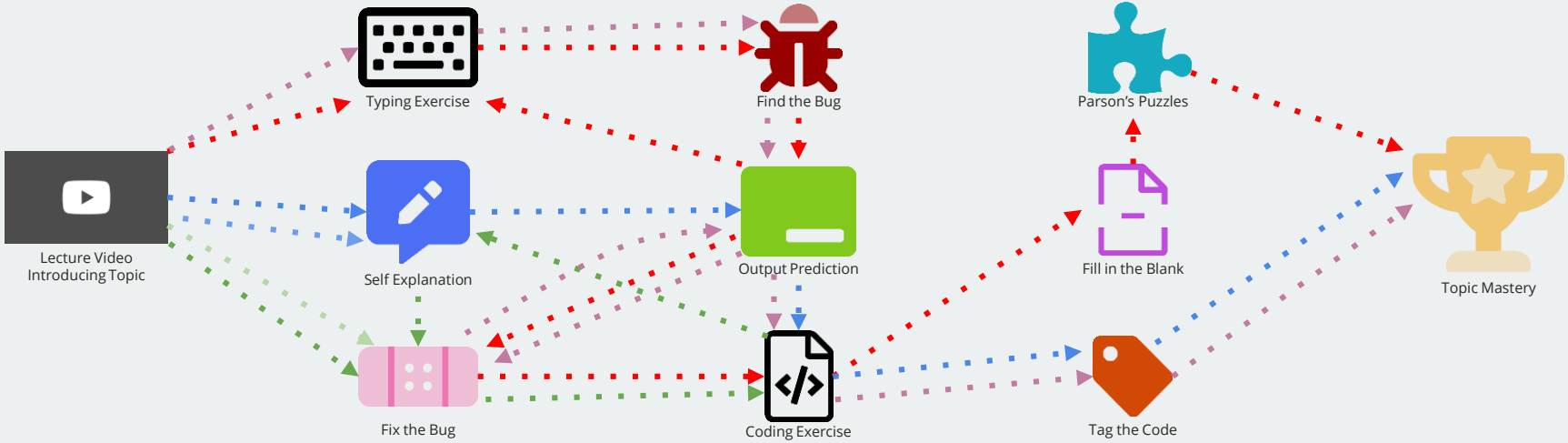

Tag the Code


Topic Mastery

However, in reality, there are multiple ways to practice

Slides: go.ncsu.edu/typos-ace2020

Future Work



By mapping prior student **activity sequences**, we seek to provide students with their **next best step** to ensure they are mastering topics efficiently.

Slides: go.ncsu.edu/typos-ace2020