

UCR

A Many Small Programs (MSP) Approach in a CS1 Course

Joe Michael Allen

Advisor: Frank Vahid

Dept. of Computer Science and Engineering

University of California, Riverside

UNIVERSITY OF CALIFORNIA, RIVERSIDE

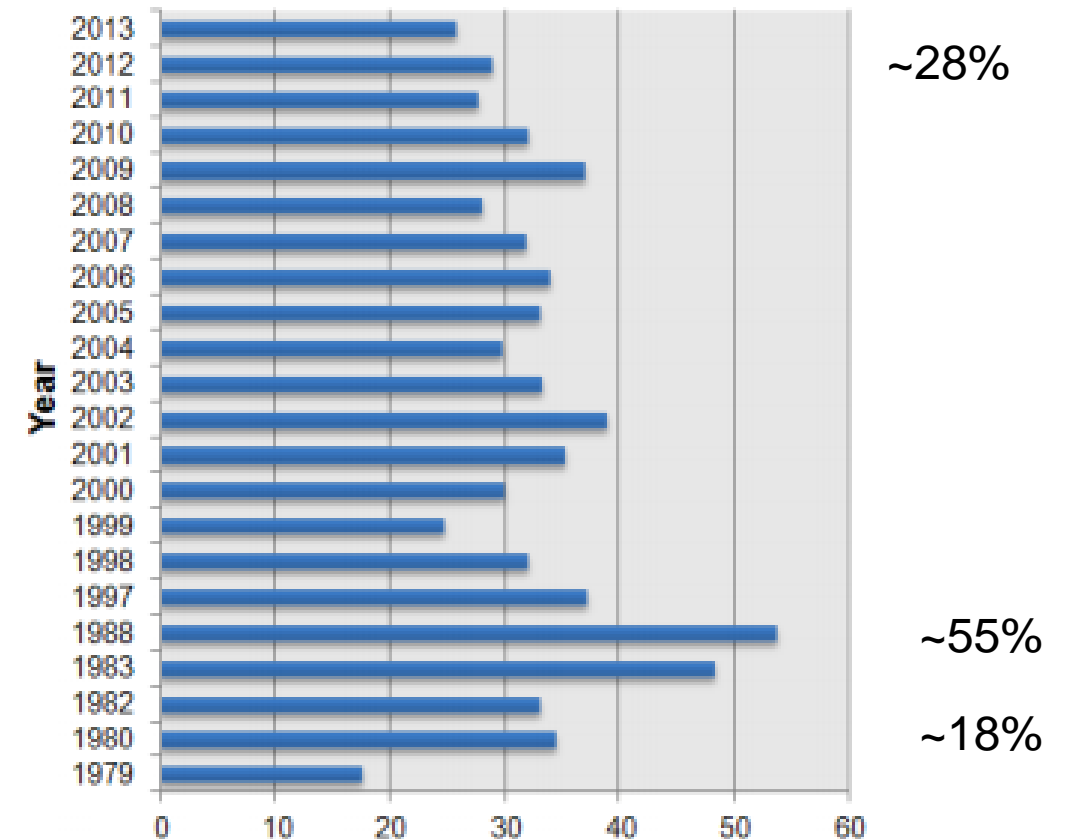
Problem

- › CS1 issues
 - › High student stress
 - › Student dissatisfaction
 - › Academic dishonesty
 - › Low grades
 - › High non-passing rates

Our Solution

- › Improve students' experience
 - › Improve student satisfaction & happiness
 - › Without worsening performance
- › Focus on weekly programming assignments
 - › Large part of the students' experience
 - › Key source of issues – student struggle/fear

~ 30% non-passing rate over the past 30 years



Watson, C. and Li, F. "Failure Rates in Introductory Programming Revisited," iTiCSE, 2014
<http://dro.dur.ac.uk/19223/1/19223.pdf%3FDDDD10%2Bd74ks0%2Bdcs0lw>

Outline

- › Related work
- › What is an MSP teaching approach
- › Study 1 – Satisfaction & grade performance
- › Study 2 – MSP usage analysis - UCR
- › Study 3 – MSP usage analysis - Other universities
- › Study 4 – MSP pivot analysis
- › Current/Future work
- › Conclusion

Related work

- › Improving CS1
 - › Pair programming [1]
 - › Use of different programming languages [2]
 - › Various tutoring models [3]
 - › Student self agency in assignments [4]
 - › Plagiarism detection [5]

- › Focus on programming assignments
 - › Real-world programming assignments (web-crawler, analyze DNA, predict election outcomes) [6, 7]
 - › Gamify programming assignments (missions, leader boards, badges) [8, 9]

- › No recent work published on our proposed solution

What is an MSP Teaching Approach?

- ▶ Traditional: One Large Program (OLP) each week
 - ▶ Solution 50-200 lines
 - ▶ Long spec
 - ▶ Multiple topics

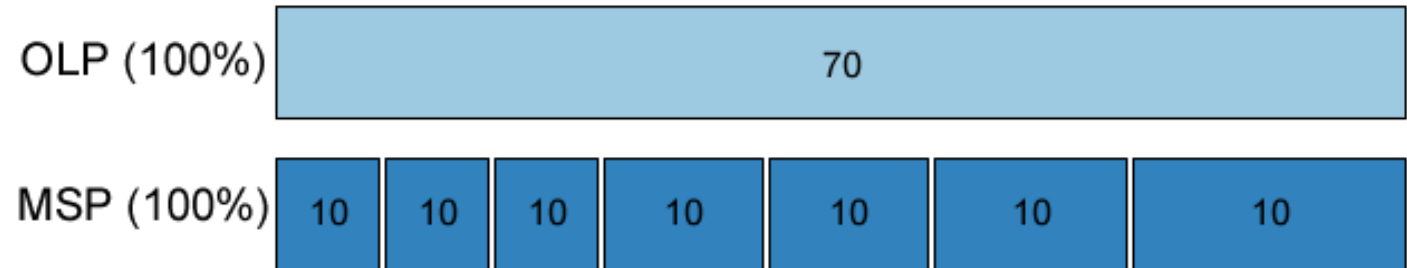
OLP (100%)



70

Many Small Programs (MSPs) each week

- › MSP teaching approach: 5-7 MSPs
 - › Solution 10-50 lines each
 - › Short & concise spec



- › Benefits
 - › Less intimidating
 - › Pivot if stuck
 - › Build confidence, more practice
- › Enabled by new auto-graders
 - › Easy to create / Instant feedback
 - › zyLabs (zyBooks): ~30 min create lab

MSPs – prompt

5.10 LAB: Count input length without spaces, periods, or commas

Visible to students [Edit lab](#) [Share](#) [Note](#)

Given a line of text as input, output the number of characters excluding spaces, periods, or commas.

Ex: If the input is "Listen, Mr. Jones, calm down." (excluding the quotes), the output is:

```
21
```

Note: Account for all characters that aren't spaces, periods, or commas (Ex: "r", "2", "!").

MSPs – solution

Solution

Add a solution and run your test cases against it before assigning to students. Solutions can also be revealed to students if desired. (Optional)

[Upload a solution](#)

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main() {
6     string userText;
7     unsigned int i;
8     int charCount;
9
10    getline(cin, userText); // Gets entire line, including spaces.
11
12    charCount = 0;
13    for (i = 0; i < userText.length(); ++i) {
14        if ( (userText.at(i) != ' ') && (userText.at(i) != '.') && (userText.at(i) != ',') ) {
15            charCount += 1;
16        }
17    }
18
19    cout << charCount << endl;
20
21    return 0;
22 }
23
```


MSPs – test cases

1. Compare output (3 points)

When input is

```
Listen, Mr. Jones, calm down.
```

Standard output exactly matches

```
21
```

2. Compare output (3 points)

When input is

```
Howdy!
```

Standard output exactly matches

```
6
```

3. Compare output (2 points)

When input is

```
abcd,,,,efgh....ijkl ....
```

Standard output exactly matches

```
12
```

4. Compare output (2 points)

When input is

```
,
```

Standard output exactly matches

```
0
```

› Test cases:

- › 10 points per MSP
- › Input/output tests
- › Unit tests

MSP sample 1 – week 1

1.19 LAB: Input: Mad Lib

Visible to students



Edit lab



Share



Note

Mad Libs are activities that have a person provide various words, which are then used to complete a short story in unexpected (and hopefully funny) ways.

Complete the program to read the needed values from input, that the existing output statement(s) can use to output a short story.

Ex: If the input is "Eric Chipotle 12 cars" (excluding the quotes), the output is:

```
Eric went to Chipotle to buy 12 different types of cars.
```

Solution

Add a solution and run your test cases against it before assigning to students. Solutions can also be revealed to students if desired. (Optional)

[Upload a solution](#)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     string firstName;
6     string genericLocation;
7     int wholeNumber;
8     string pluralNoun;
9
10    cin >> firstName;
11    cin >> genericLocation;
12    cin >> wholeNumber;
13    cin >> pluralNoun;
14
15    cout << firstName << " went to " << genericLocation << " to buy " << wholeNumber << " different types of " << pluralNoun << "." << endl;
16
17    return 0;
18 }
```

MSP sample 2 – week 6

6.8 LAB: Step counter

Visible to students



Edit lab

Share

Note

A pedometer treats walking 2,000 steps as walking 1 mile. Write a program whose input is the number of steps, and whose output is the miles walked.

Ex: If the input is 5345, the output is:

2.6725

Your program must define and call a function:

```
double StepsToMiles(int userSteps)
```

Solution

Add a solution and run your test cases against it before assigning to students. Solutions can also be revealed to students if desired. (Optional)

[Upload a solution](#)

```
1 #include <iostream>
2 using namespace std;
3
4 double StepsToMiles(int userSteps) {
5     return userSteps / 2000.0;
6 }
7
8 int main() {
9     int inputSteps;
10
11     cin >> inputSteps;
12     cout << StepsToMiles(inputSteps) << endl;
13
14     return 0;
15 }
16
```

MSP sample 3 – week 8

8.15 LAB: Word frequencies

Visible to students



Edit lab

Share

Note

Write a program that reads a list of words. Then, the program outputs those words and their frequencies. The input begins with an integer indicating the number of words that follow. Assume that the list will always contain less than 20 words.

Ex: If the input is:

```
5 hey hi Mark hi mark
```

Then, the output is:

```
hey 1
hi 2
Mark 1
hi 2
mark 1
```

Hint: Use two vectors, one vector for the strings and one vector for the frequencies.

Solution

Add a solution and run your test cases against it before assigning to students. Solutions can also be revealed to students if desired. (Optional)

[Upload a solution](#)

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 using namespace std;
5
6 int main() {
7     vector<string> userWords;
8     vector<int> wordFreq;
9     string currWord;
10    unsigned int i;
11    unsigned int j;
12    unsigned int numWords;
13
14    // Integer indicating the number of integers that follow
15    cin >> numWords;
16
17    // Gets list of words from input and initialize word's frequency to 0
18    for(i = 0; i < numWords; ++i) {
19        cin >> currWord;
20        userWords.push_back(currWord);
21        wordFreq.push_back(0);
22    }
23
24    // Determine frequency of each word
25    for (i = 0; i < userWords.size(); ++i) {
26        for (j = 0; j < userWords.size(); ++j) {
27            if (userWords.at(i) == userWords.at(j)) {
28                wordFreq.at(i) = wordFreq.at(i) + 1;
29            }
30        }
31    }
32
33    // Outputs words and their frequencies
34    for(i = 0; i < userWords.size(); ++i) {
35        cout << userWords.at(i) << " " << wordFreq.at(i) << endl;
36    }
37
38    return 0;
39 }
40 }
```

Study 1 – Satisfaction & grade performance

› RQ's:

- › Do MSPs impact student satisfaction?
- › Do MSPs impact student grade performance?

J.M. Allen, F. Vahid, K. Downey, A. Edgcomb.
Weekly Programs in a CS1 Class: Experiences with Auto-graded Many-small Programs (MSP). ASEE Annual Conference, 2018. **(Best paper nominee)**

› Methods

› Student “stress” survey

- › Given week 8 of the quarter
- › Ask students about their experience
- › 18 questions: Strongly agree (6) to Strongly disagree (0)

› Student grade performance

- › Participation, Challenge, and Programming Activities, Midterm, Final, Total grade

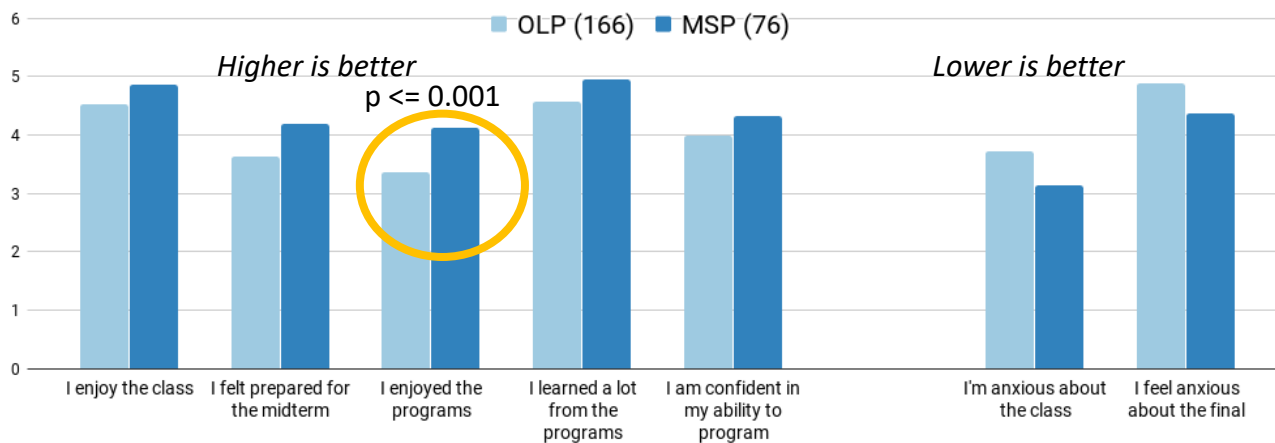
The screenshot shows a survey interface with a purple header. At the top right, it says 'QUESTIONS' and 'RESPONSES 1,213'. The title of the survey is 'Anonymous class survey'. Below the title is a 'Form description' field. The first question is 'Which course are you enrolled in?' with two radio button options: 'CS 10 (in-person section) Spring18' and 'CS 10 (online section) Spring18'. The second question is 'Please answer the following about the class *' with a Likert scale from 'Strongly agree' to 'Strongly disagree'. Three items are listed with their corresponding scales:

	Strongly agree	Agree	Slightly agree	Slightly disagr...	Disagree	Strongly disag...
I enjoy the clas...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am often anxl...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I spend a lot of...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

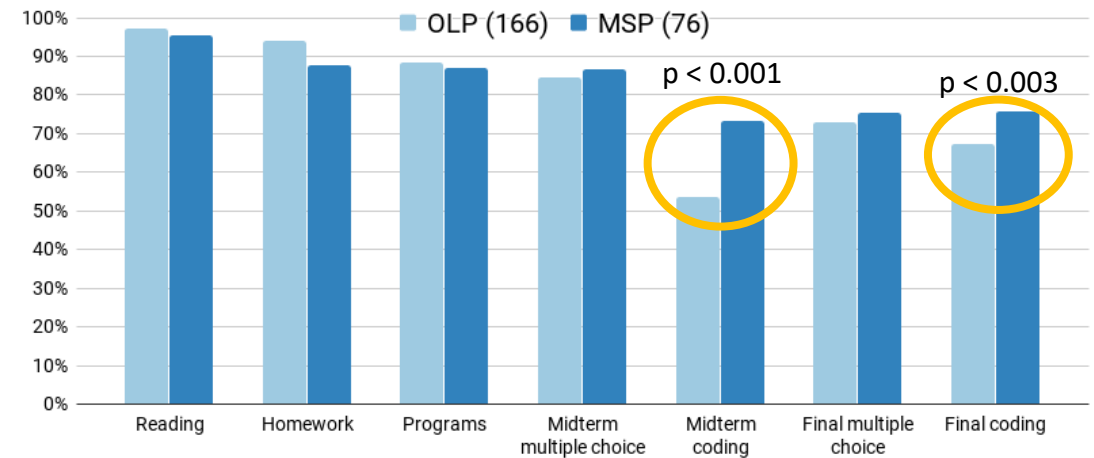
Study 1 - Conclusion

- MSP group had more favorable responses and enjoyed the class more
- MSP group student grade performance did not worsen (higher coding scores on exams)
- Results: UCR CS1 use MSPs, ~200 universities use MSPs, and zyBooks mimicked and maintains MSPs

Student satisfaction – stress survey results



Student grade performance results



Study 2 – MSP usage analysis - UCR

› RQ's:

- › How do students interact with MSPs?

J.M. Allen, F. Vahid, A. Edgcomb, K. Downey, and K. Miller. An Analysis of Using Many Small Programs in CS1, ACM SIGCSE Technical Symposium on Computer Science Education, 2019.

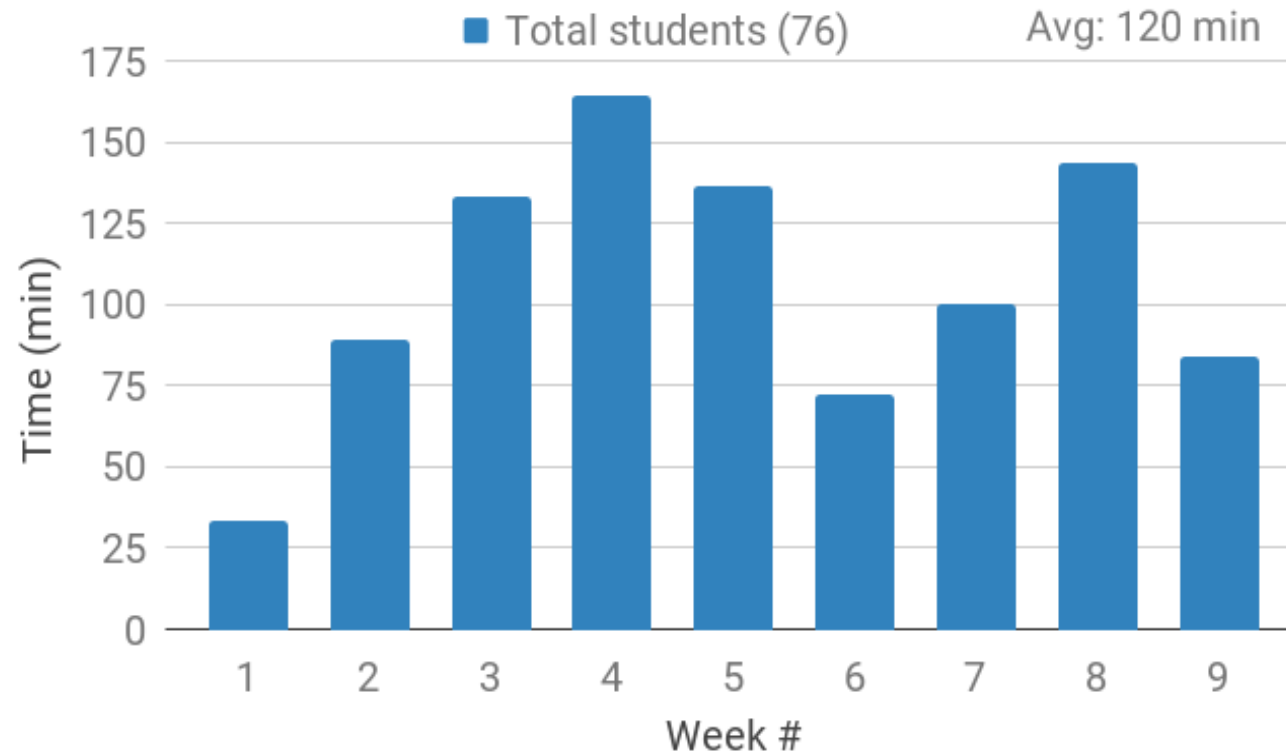
› Methods

- › UCR CS1 Spring 2017 MSP section: 76 students
- › zyLab metadata

labID		userID	score	maxScore	timestamp	
14	CH1 LAB: Formatted output: No parking sign		31228		4/8/2018 22:55	
15	CH1 LAB: Formatted output: No parking sign		31228		4/8/2018 22:55	
16	CH1 LAB: Formatted output: No parking sign		31228	10	10	4/8/2018 22:55
17	CH1 LAB: Input: Welcome message		31228		4/8/2018 22:57	
18	CH1 LAB: Input: Welcome message		31228	10	10	4/8/2018 22:58
19	CH1 LAB: Input: Mad Lib		31228		4/8/2018 23:01	
20	CH1 LAB: Input: Mad Lib		31228		4/8/2018 23:02	
21	CH1 LAB: Input: Mad Lib		31228		4/8/2018 23:02	
22	CH1 LAB: Input: Mad Lib		31228		4/8/2018 23:03	
23	CH1 LAB: Input: Mad Lib		31228	10	10	4/8/2018 23:03
24	CH1 LAB: Input and formatted output: House real estate summary		31228		4/8/2018 23:08	
25	CH1 LAB: Input and formatted output: House real estate summary		31228		4/8/2018 23:08	

Q: How much time do students spend working on MSPs each week?

A: At least 120 min / week

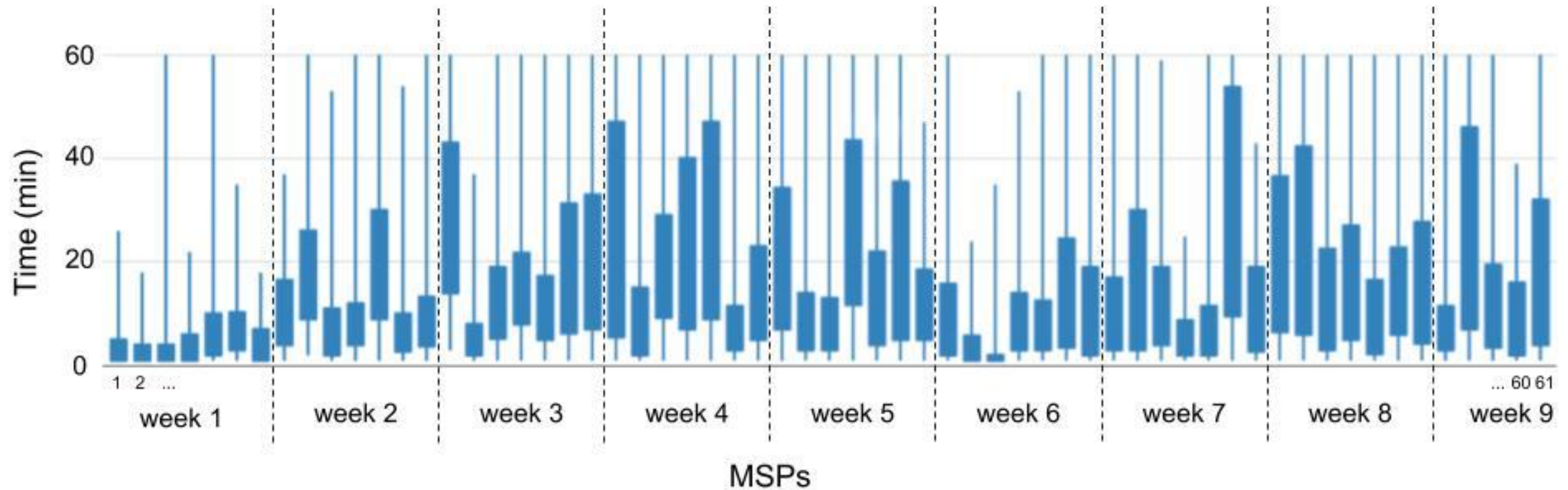


NOTE: *Underestimate.*

Students with 0 subs or 0 time excluded. Avg is for weeks 2-8.

Q: How much time do students spend working on each MSP?

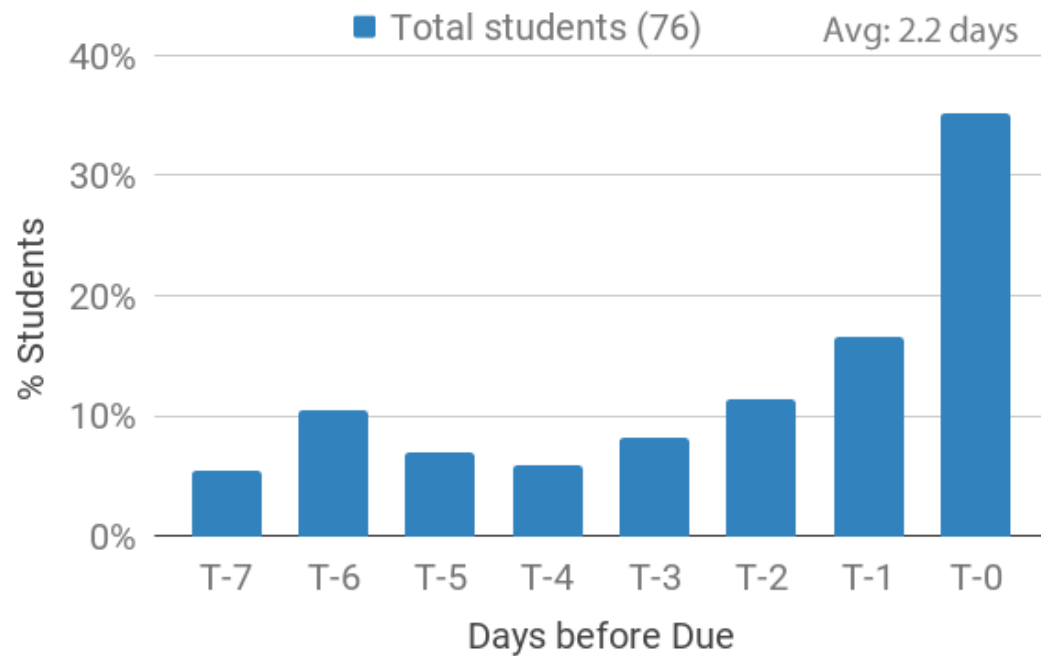
A: About 17 min / MSP



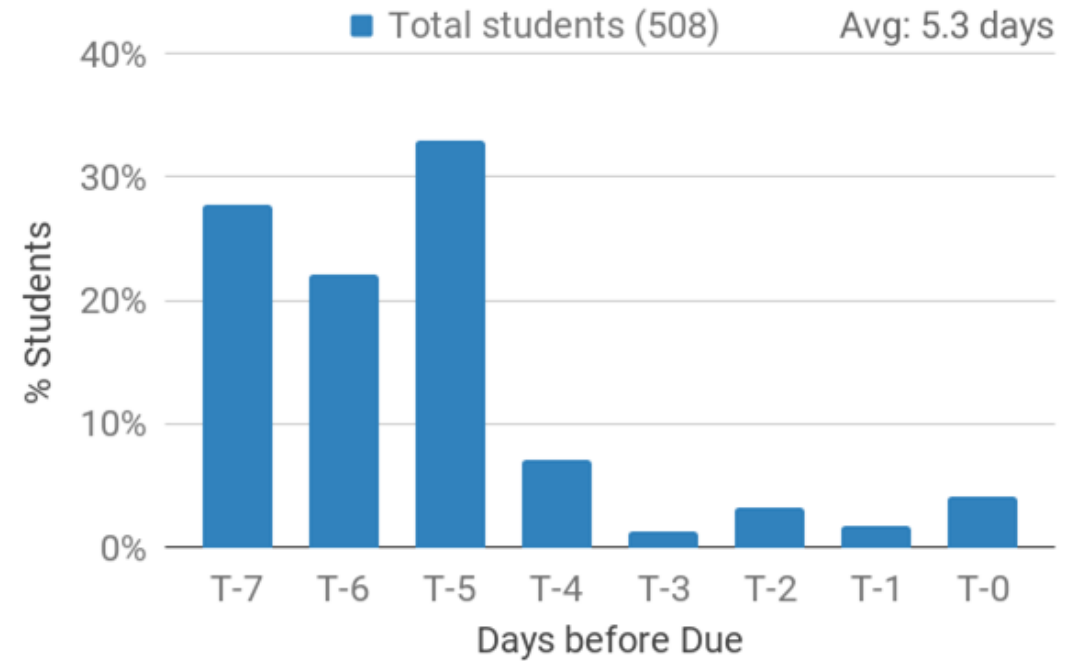
Average time spent per MSP - 17 min / MSP (weeks 1 and 9 excluded).

Q: How many days before the due date do students start working on MSPs?

A: MSPs started 2.2 days before due date

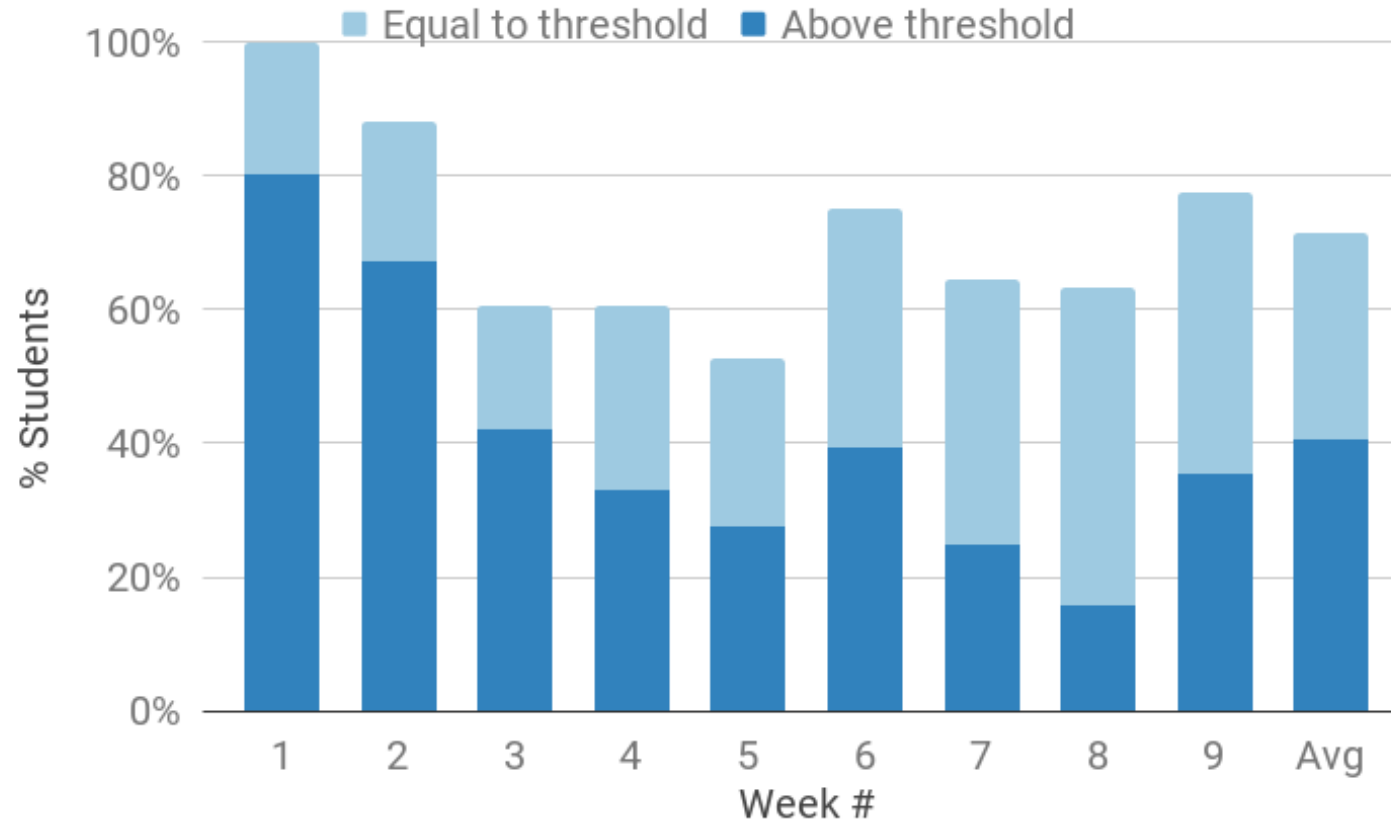


A: With policy adjustment in Fall 2018, started 5.3 days before



Q: Given a full-credit threshold, do students complete more MSPs than required?

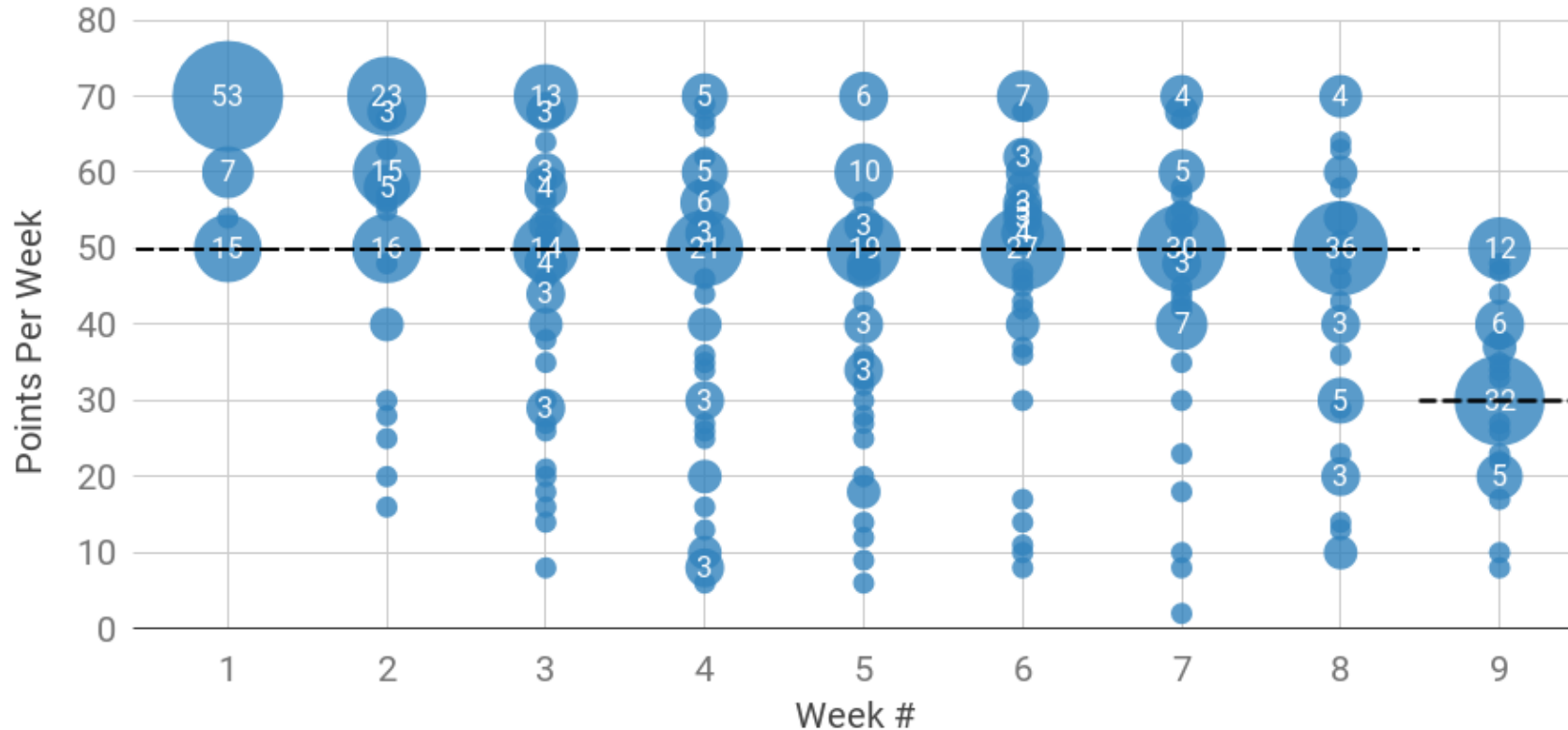
A: 40% of students completed more MSPs than required



No extra credit given for exceeding full-credit threshold

Q: Given a full-credit threshold, how many points do students score each week?

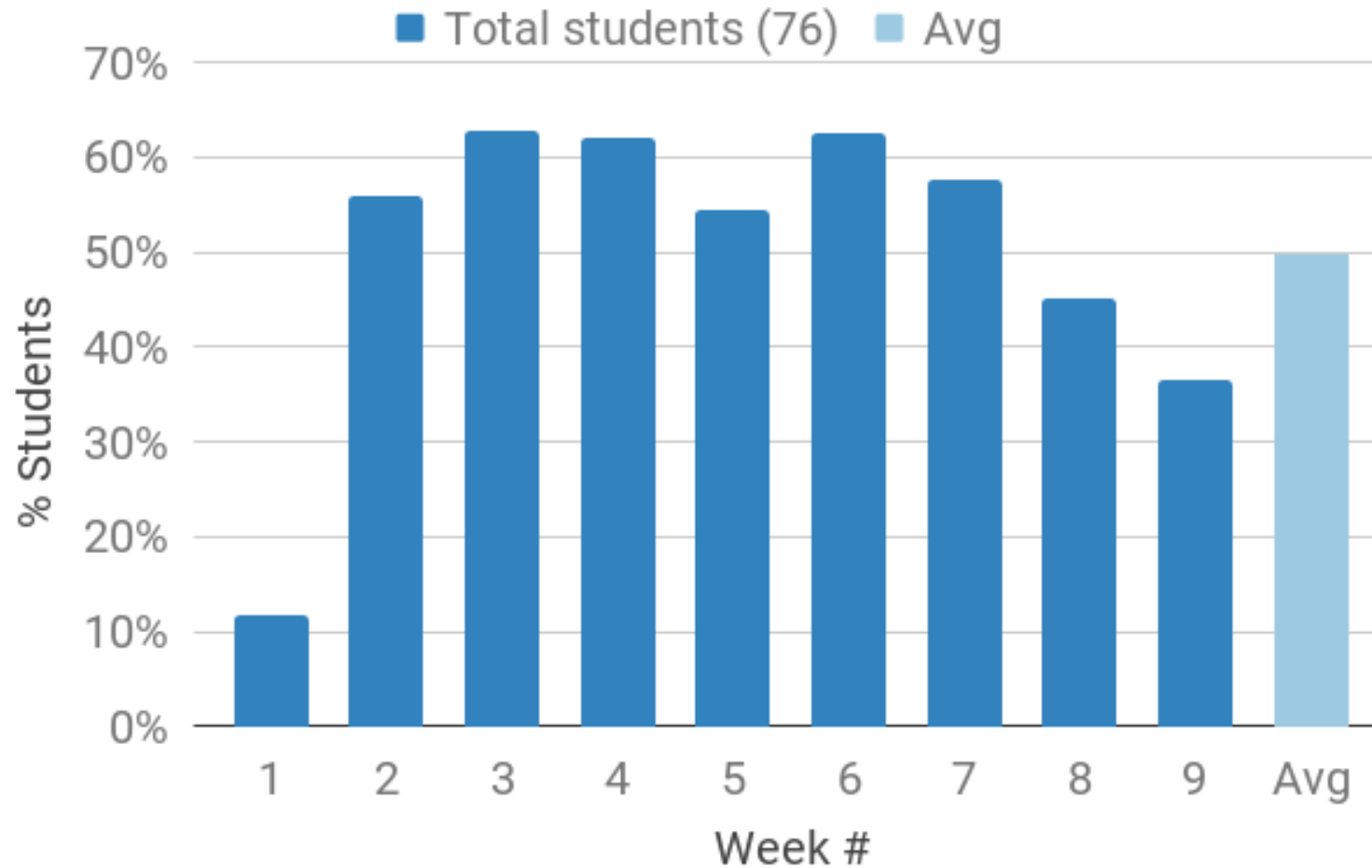
A: Total points per week – Avg 13 more points



Bubble size represents number of students. Dashed line indicates full-credit threshold. Students who scored 0 points for a week excluded.

Q: Do students pivot, or help themselves when stuck?

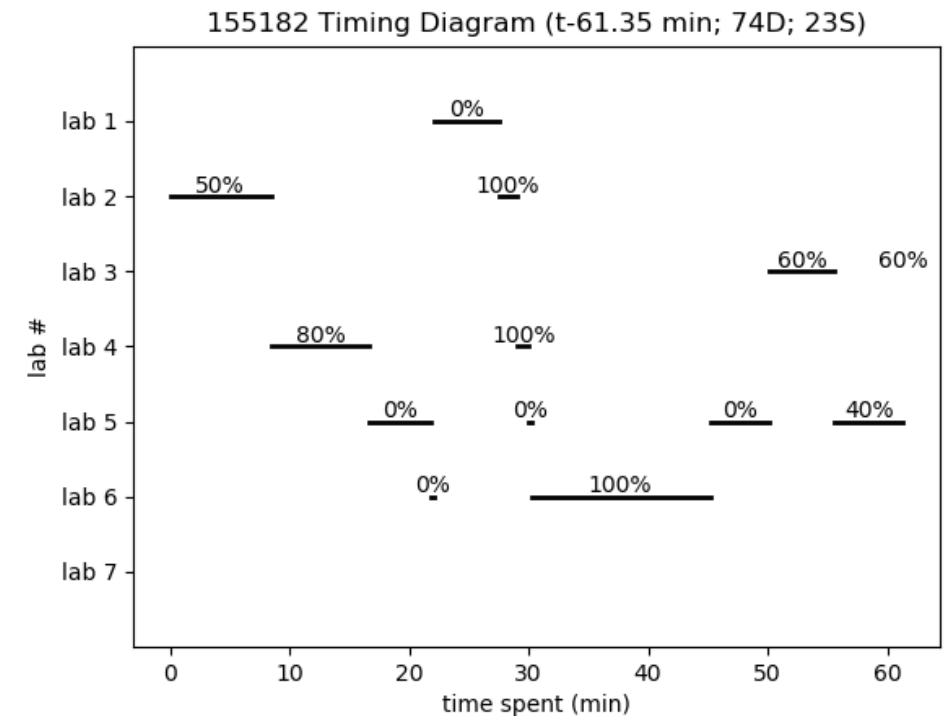
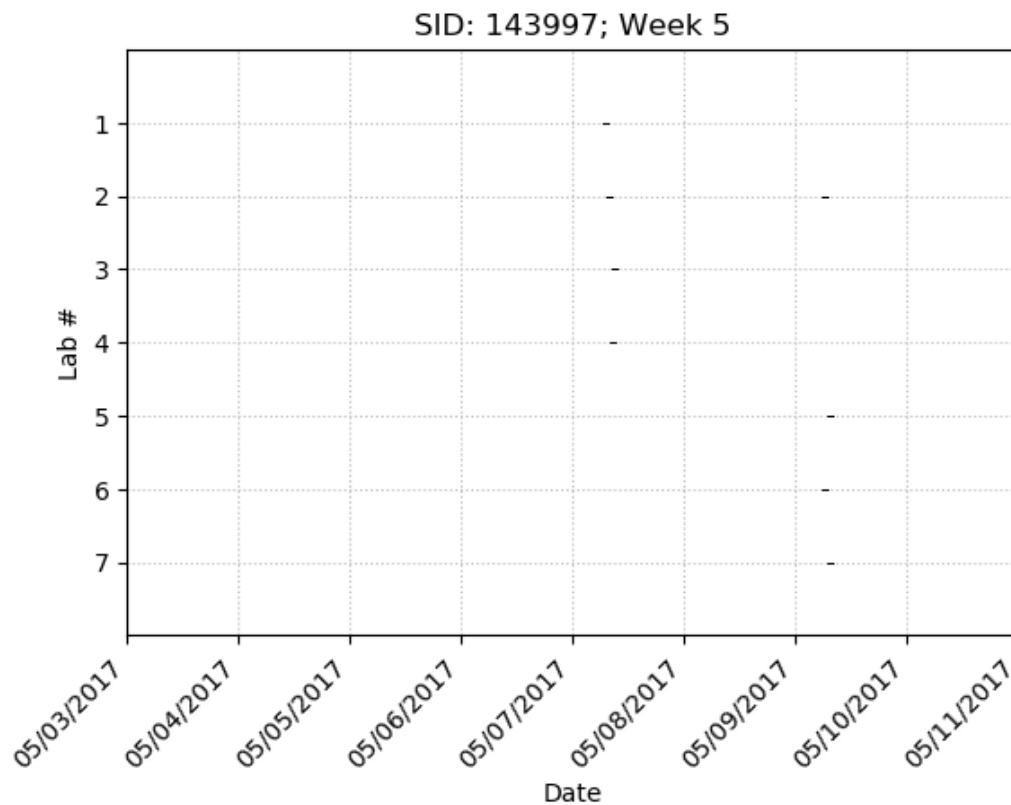
A: Each week, 50% of students pivoted (avg. 1.3 pivots)



Q: Can we visually represent pivots?

A: Yes - Student workflow charts (GANTT charts)

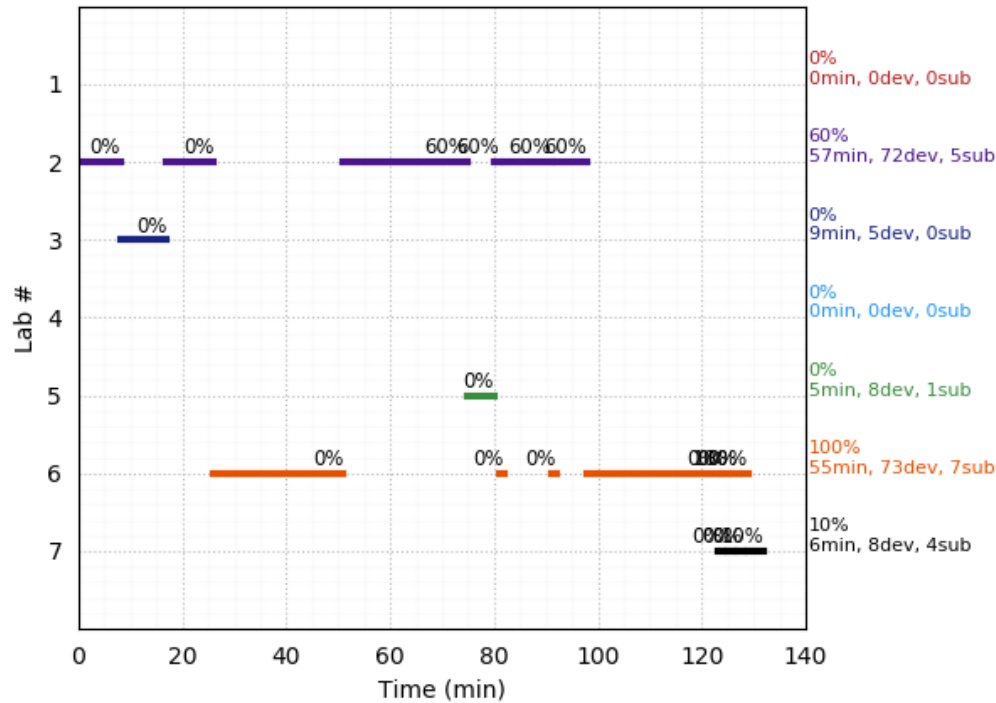
J.M. Allen and F. Vahid. Concise Graphical Representations of Student Effort on Weekly Many Small Programs, ACM SIGCSE Technical Symposium on Computer Science Education, 2020.



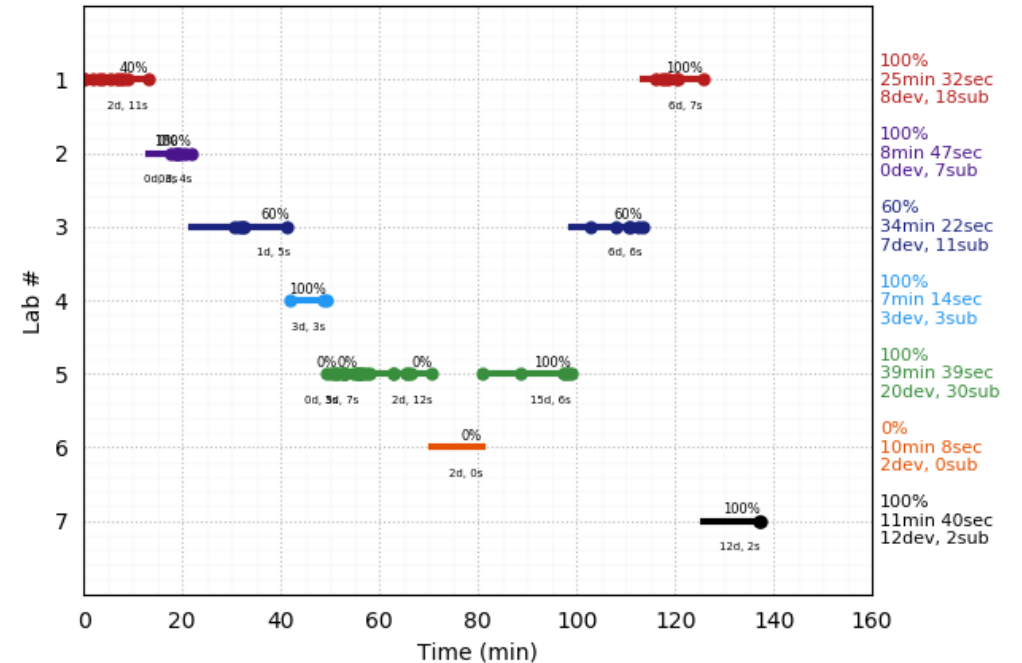
Q: Can we visually represent pivots?

A: Yes - Student workflow charts (GANTT charts)

135322 Week 4 Lab Gantt Chart (132min, 166dev, 17sub)



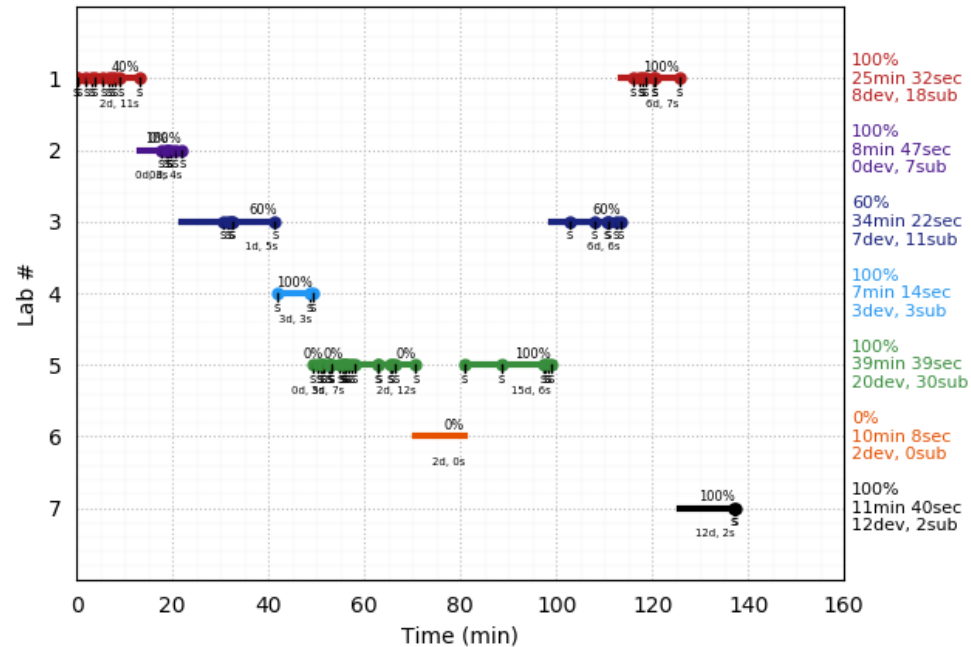
SID: 64046; Week 2 LA Gantt Plot (137min 22sec; 52dev; 71sub)



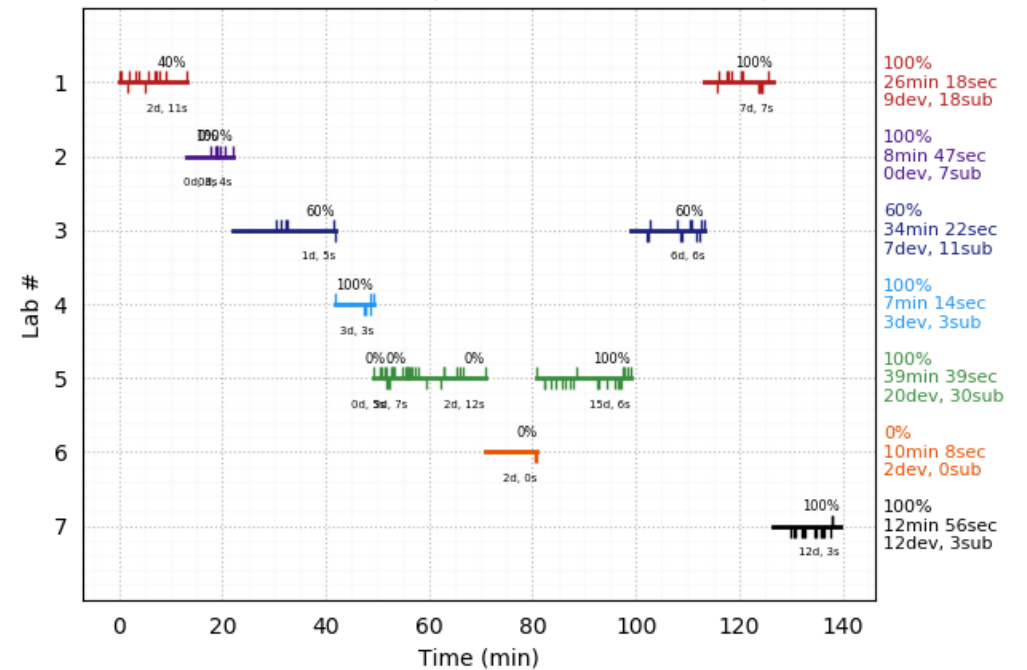
Q: Can we visually represent pivots?

A: Yes - Student workflow charts (GANTT charts)

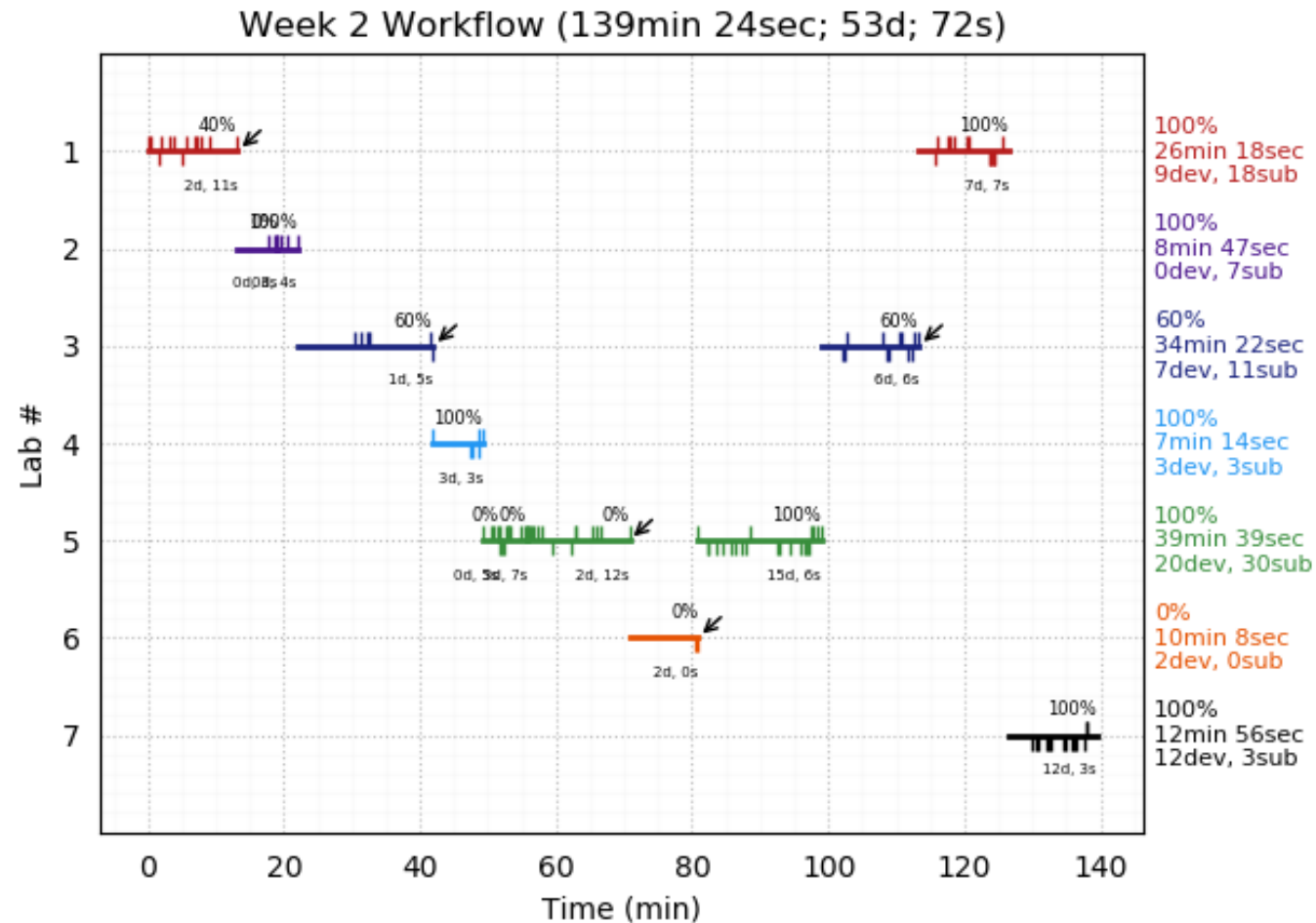
SID: 64046; Week 2 LA Gantt Plot (137min 22sec; 52dev; 71sub)



Week 2 Workflow (139min 24sec; 53d; 72s)



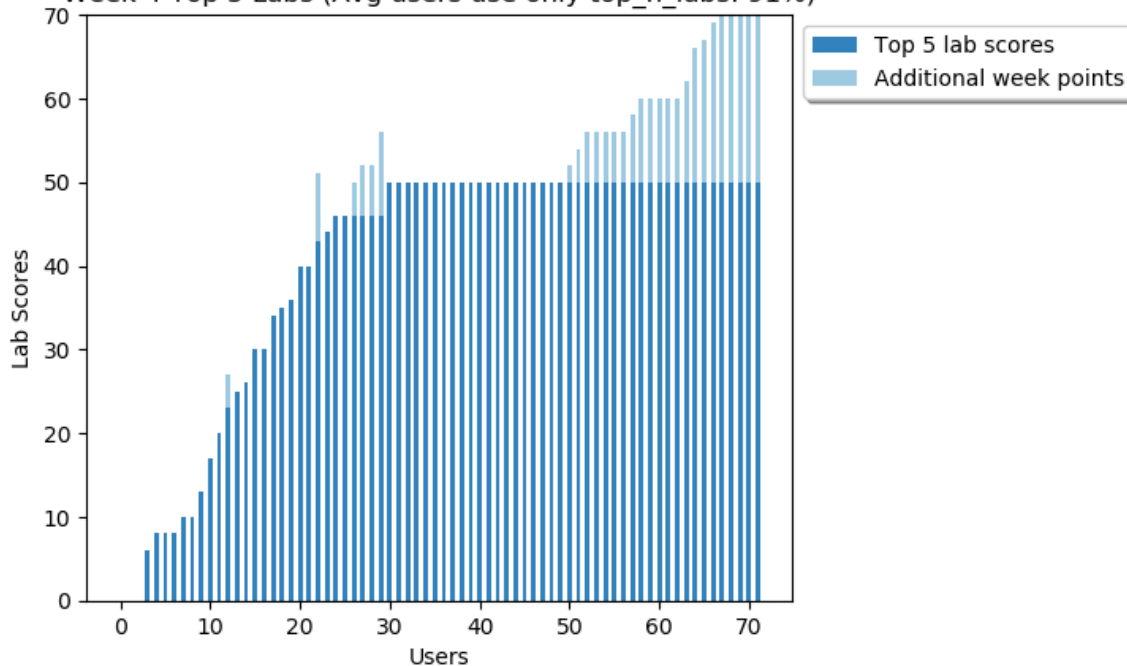
Student workflow chart



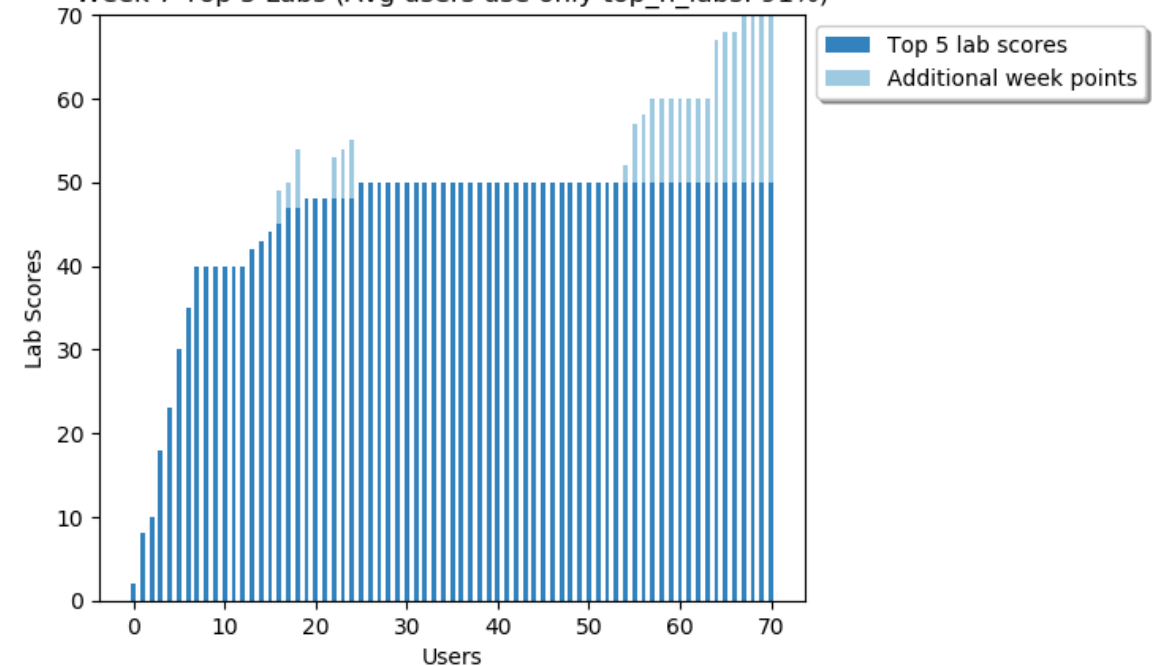
Common question – do students skip the ‘hard parts’?

A: ~95% of students score full credit using their top 5 highest scores

Week 4 Top 5 Labs (Avg users use only top_n_labs: 91%)



Week 7 Top 5 Labs (Avg users use only top_n_labs: 91%)



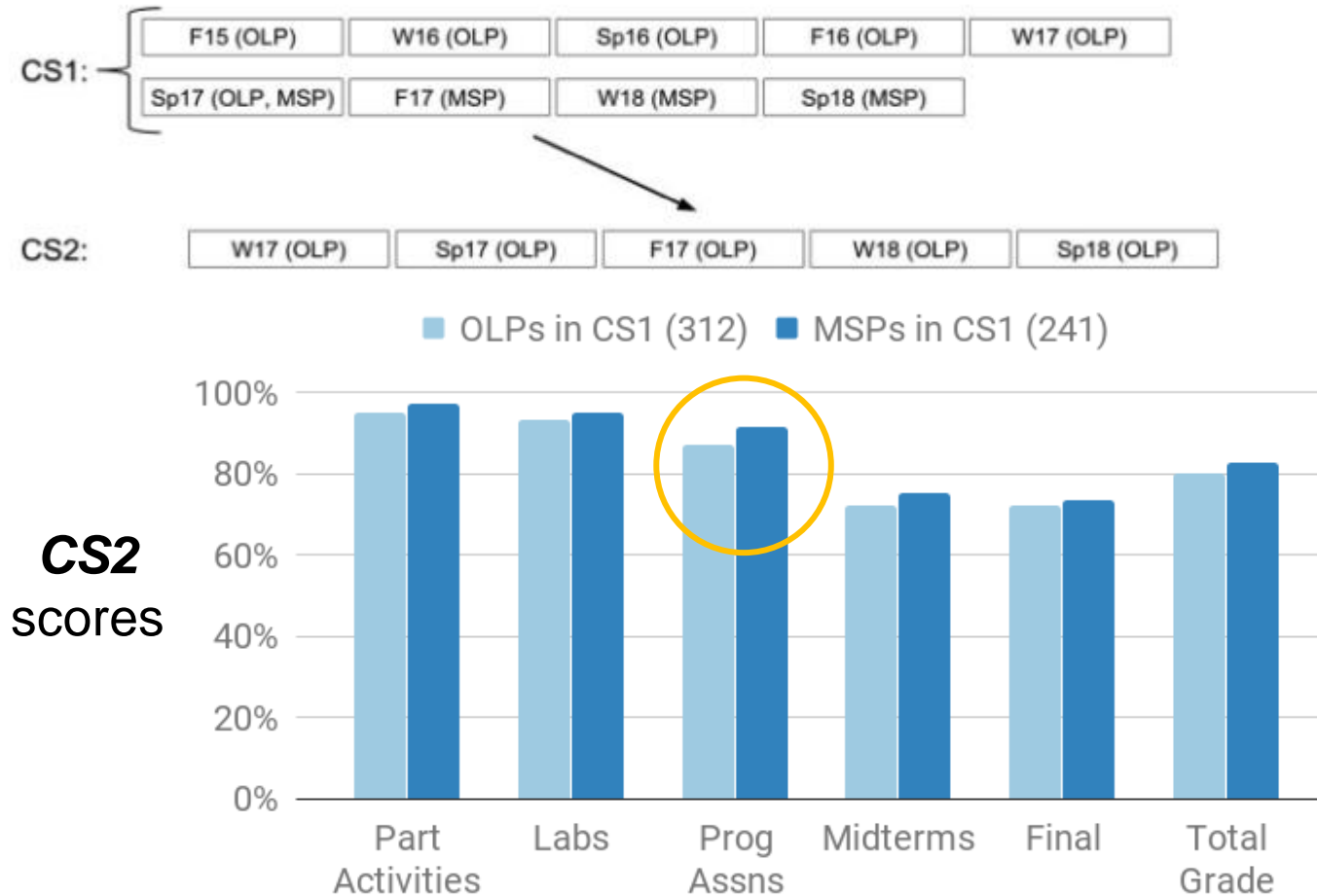
Q: Do students use MSPs to study for exams?

A. Yes, students use MSPs to study for exams

Total number of students	76
Total number of MSPs	61
% of students that used MSPs to study for the midterm	38%
% of students that used MSPs to study for the final	37%
% of students that used MSPs to study for either exam	54%
% of MSPs that were used to study for the midterm	97%
% of MSPs that were used to study for the final	90%
% of MSPs that were used to study for either exam	98%

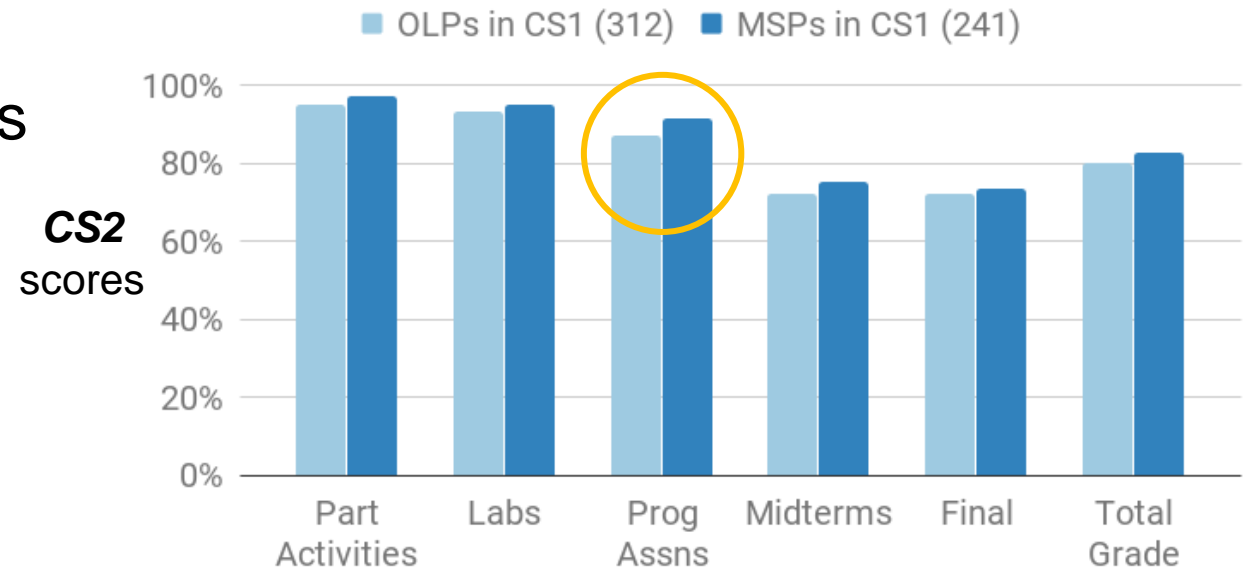
Q: *Won't MSP CS1 students do poorly in an OLP CS2?*

A. MSP CS1 students do fine in an OLP CS2, in fact slightly better



Study 2 - Conclusion

- › Students make good use of MSPs
 - › Sufficient time
 - › Started early
 - › Completed more than necessary
 - › Pivoted to help selves when stuck
 - › Used MSPs to study for exams



- › *And*, MSP CS1 students do just as well as OLP CS1 students in an OLP CS2.

Study 3 – MSP usage analysis - Other universities

- › RQ's:
 - › How do students interact with MSPs at other universities?

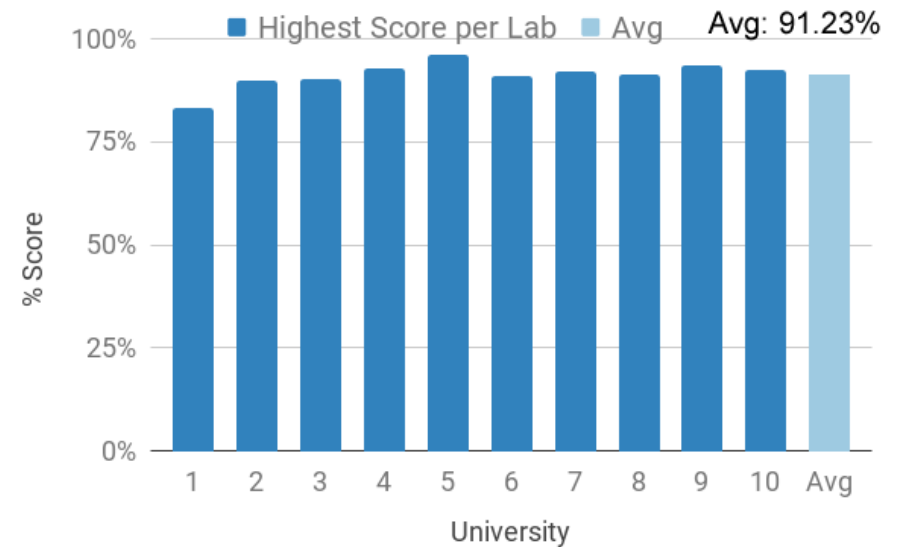
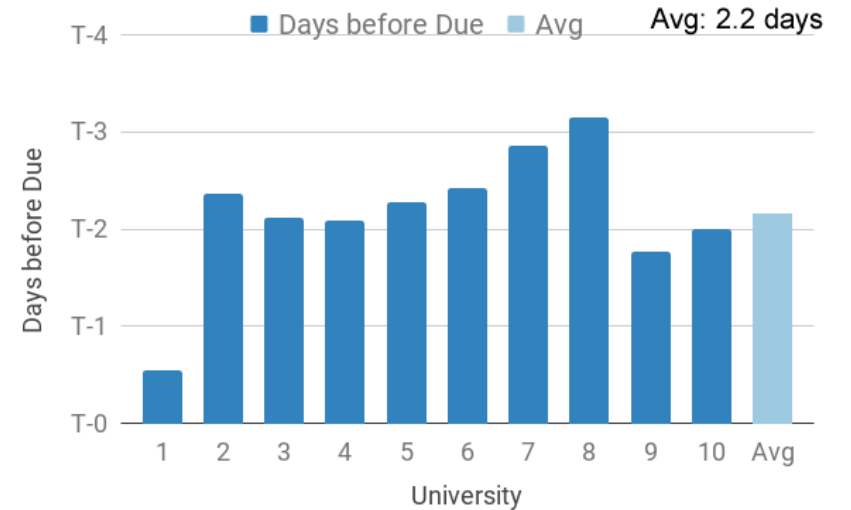
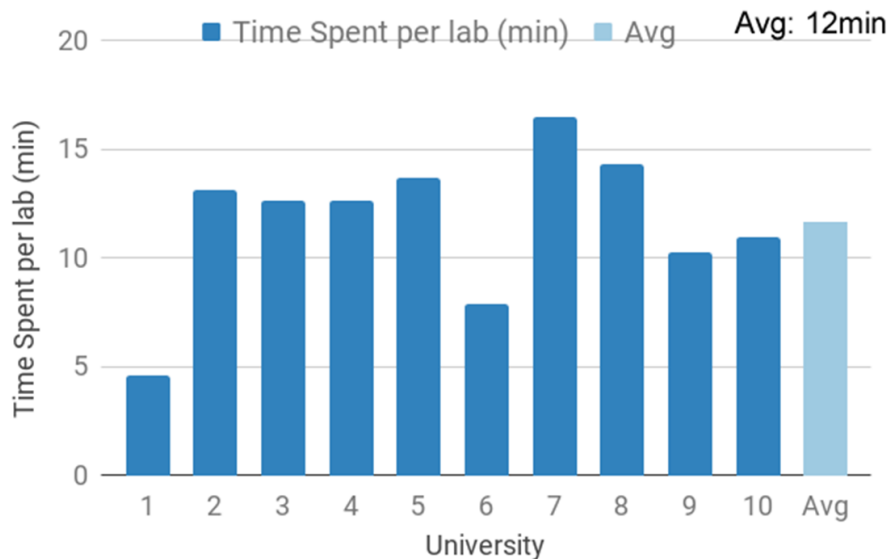
J.M. Allen, F. Vahid, K. Downey, K. Miller, and A. Edgcomb. Many Small Programs in CS1: Usage Analysis from Multiple Universities, Proceedings of ASEE Annual Conference, 2019.

› Methods

	Prog Language	#Students	# MSPs	# Submissions collected	# Develops collected
University 1	C++	20	98	3177	5635
University 2	Python	81	69	19244	19707
University 3	C++	30	19	2397	3416
University 4	C++	14	61	1675	5104
University 5	Java	11	51	643	3535
University 6	C++	234	77	21451	40573
University 7	Python	333	43	88981	103089
University 8	C++	79	25	7315	9298
University 9	Java	56	59	7454	18505
University 10	Java	321	65	40320	96721

Study 3 - Conclusion

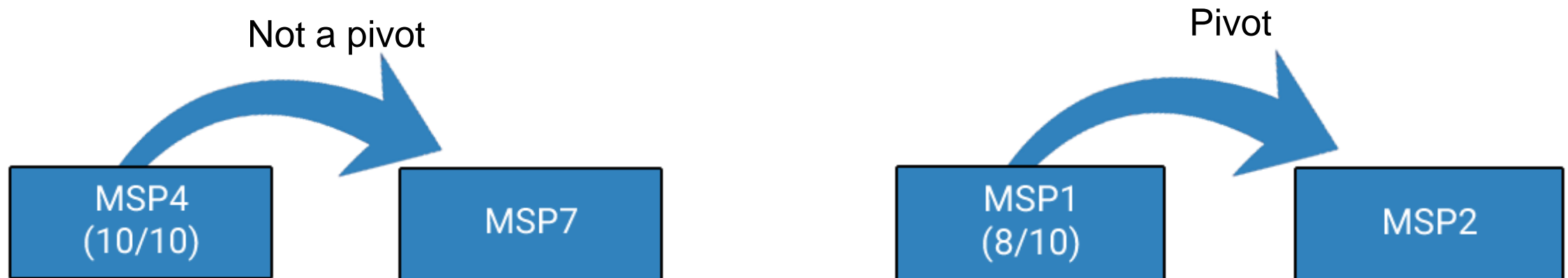
- ▶ Similar results from other universities
 - ▶ Spend sufficient time (avg 12min per lab)
 - ▶ Start early (avg 2.2 days)
 - ▶ Complete most MSPs (avg 91% completion)



Study 4 – MSP pivot analysis

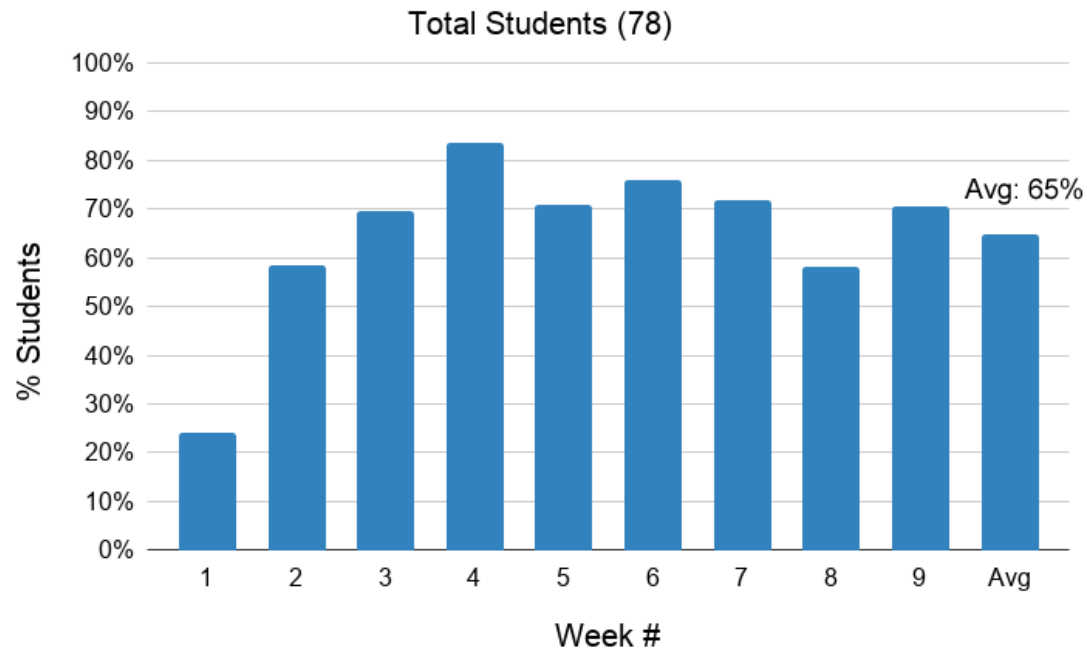
- › RQ's:
 - › Do students make use of pivoting with MSPs?
- › Pivot definition:
 - › When a student switches to a different activity before completing the previous one first

J.M. Allen and F. Vahid. Analyzing Pivoting Among Weekly Many Small Programs in a CS1 Course, Proceedings of ASEE Annual Conference, 2020.

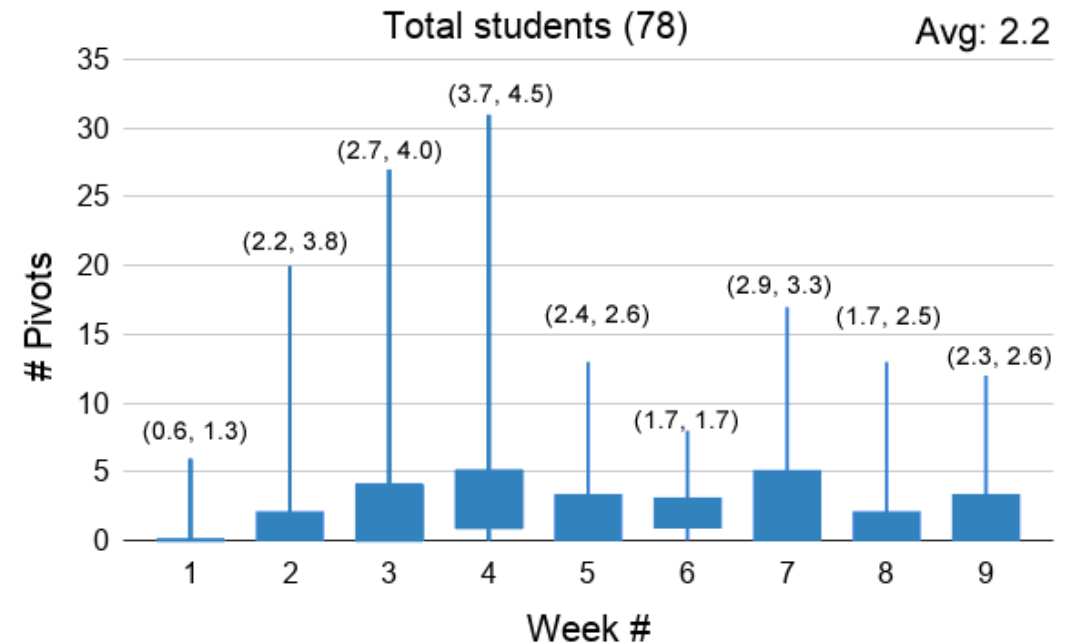


Pivot outcomes

% Students that pivot each week

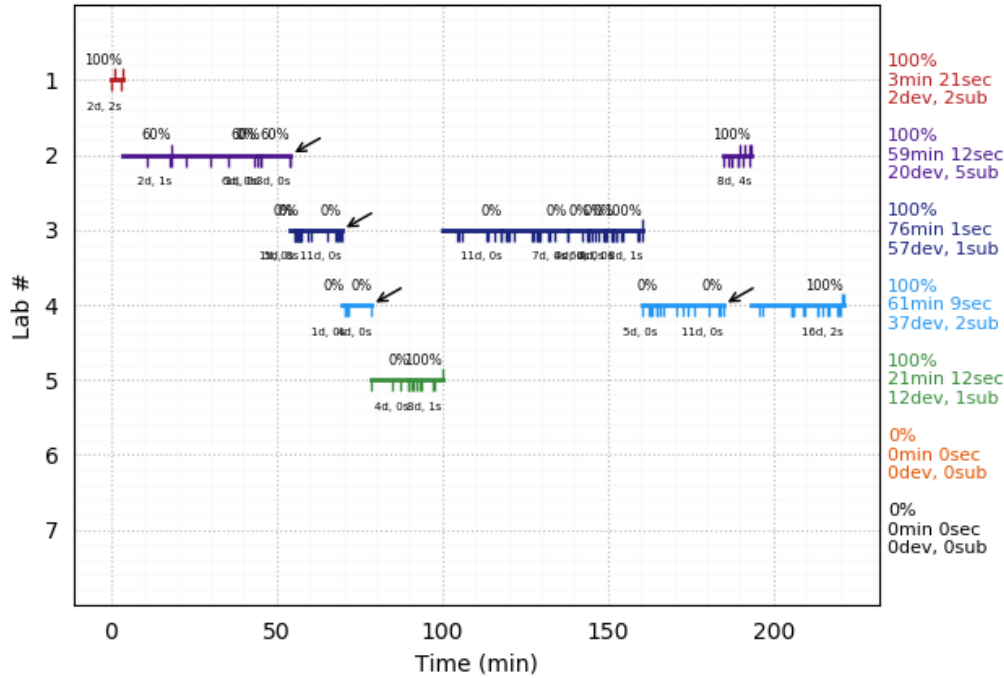


Pivots each week (avg, stdev)

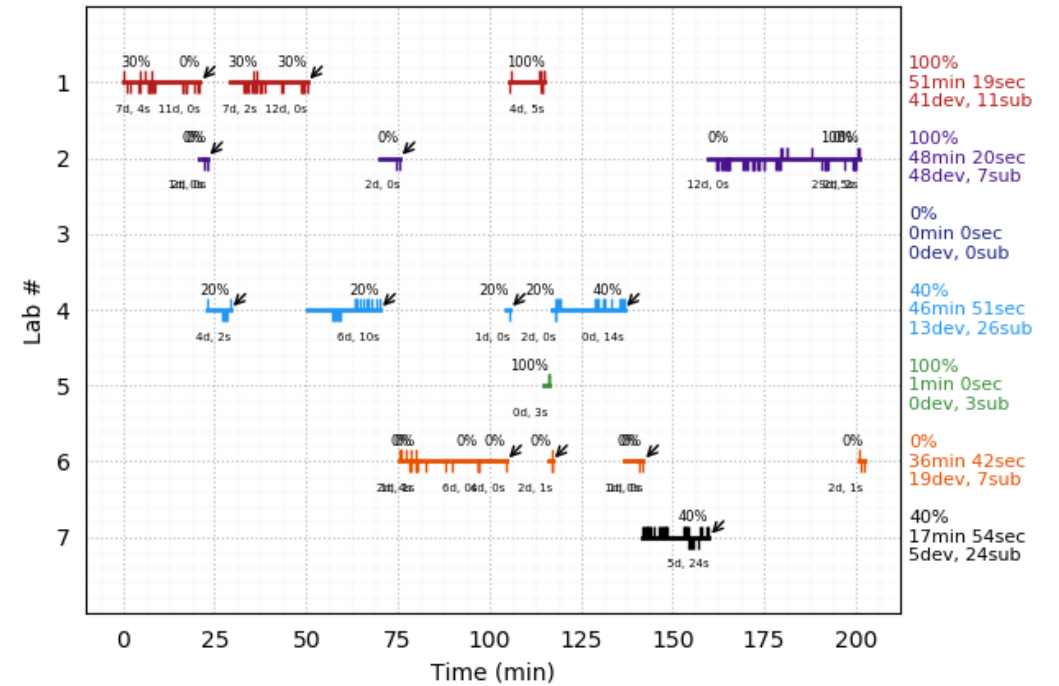


Pivot outcomes

Week 4 Workflow (220min 55sec; 128d; 11s)



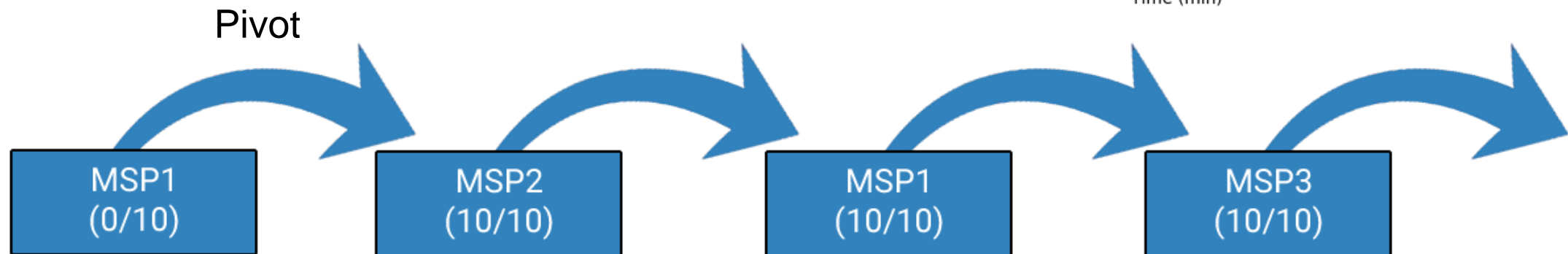
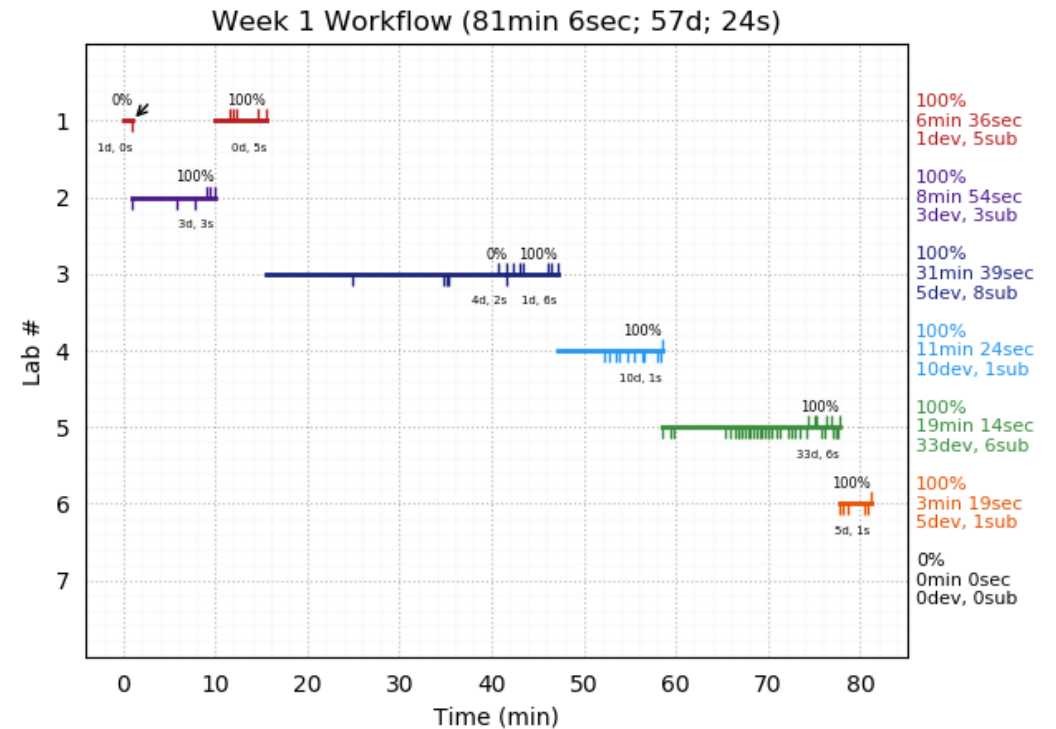
Week 8 Workflow (202min 6sec; 126d; 78s)



Study 4 – Conclusion

- > Students make use of pivoting
 - > 65% of students pivot each week
 - > Avg 2.2 pivots each week

- > Student workflow charts help us visually recognize pivot patterns



Current/Future work

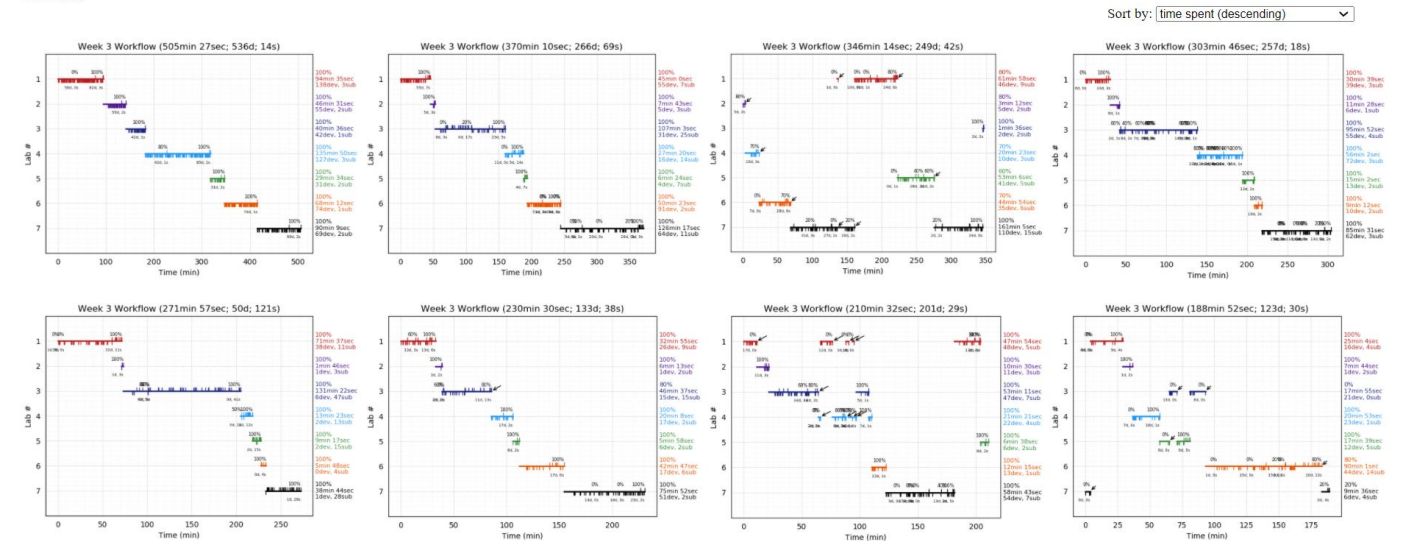
- › Does showing student their workflow charts impact their behavior
- › MSP student workflow chart classifications
- › Make all tools and analyses available online

Does showing student their workflow charts impact their behavior

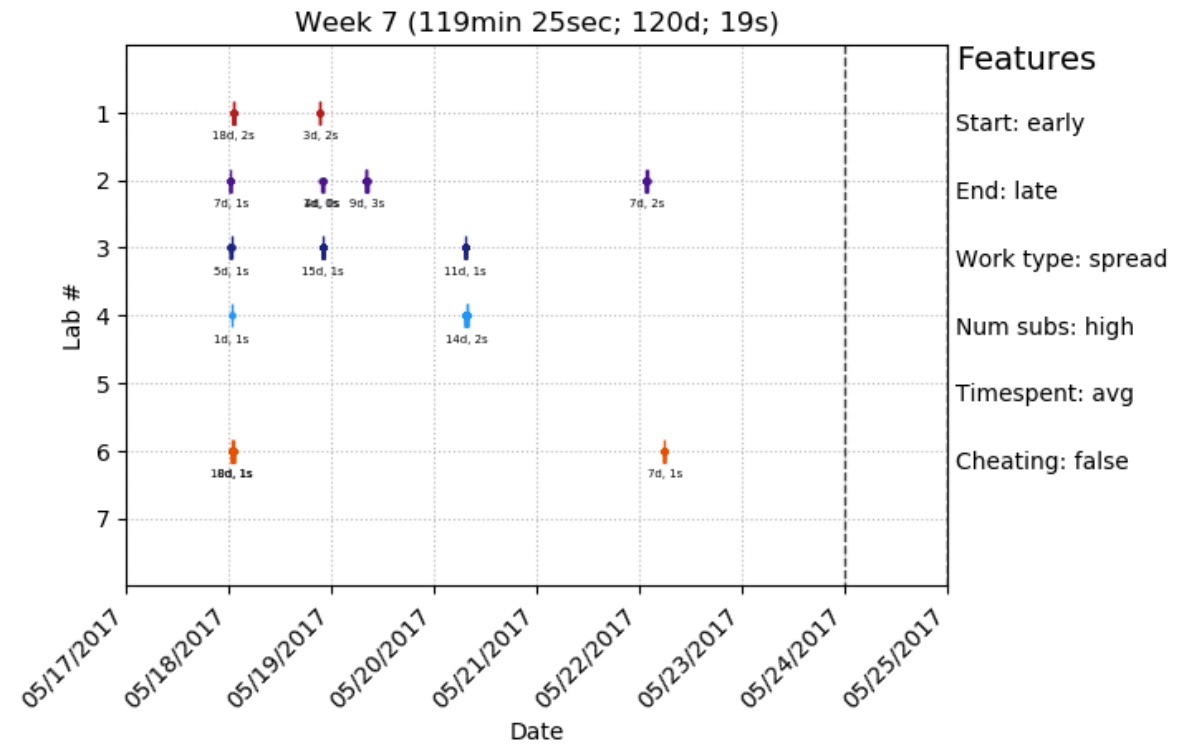
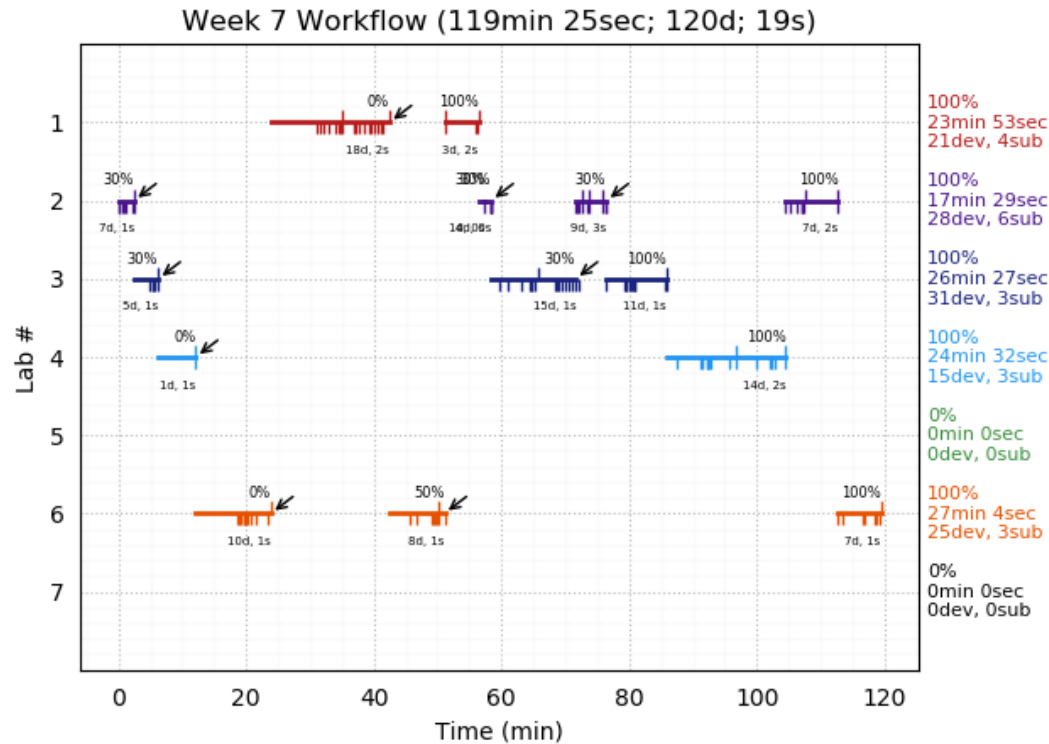
- Fall 2020 – CS10A
- Are there any impacts in:
 - Timespent
 - Number of develops / submits
 - Cheating

➤ [Interactive website](#)

Week 3



MSP student workflow chart classifications



Make all the tools and analysis available online

UCR - zyBooks zyLab Analysis

Lorem ipsum.

Questions? Email: analysis-support@cs.ucr.edu

- Upload zyBooks zyLab log: No file chosen
- [sample_1.csv](#) ([View file](#)) ([Download file](#))
- [sample_2.csv](#) ([View file](#)) ([Download file](#))

Analysis options

- Run all analyses
- Run time spent analysis
- Run score analysis

ANALYZE

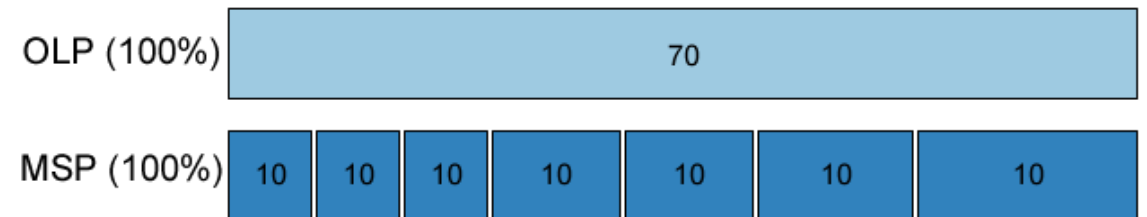
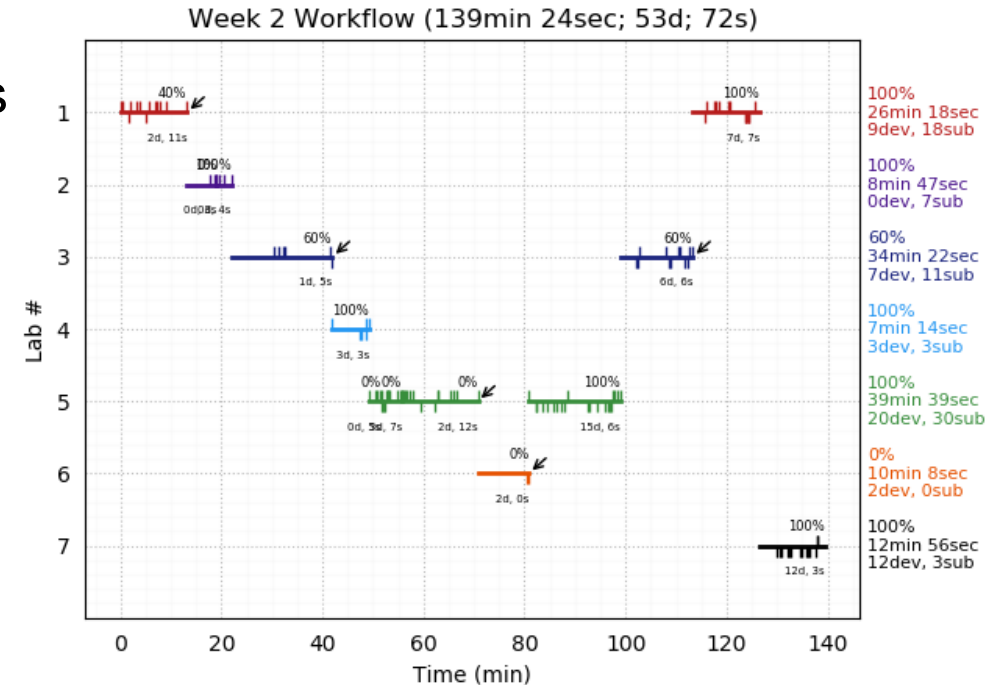
Conclusion

- > Using an MSP approach in CS1 has positive results
 - > Students are more satisfied & grades are not harmed

- > Students make good use of MSPs
 - > Sufficient time, start early, complete more than required, pivot when stuck, study for exams
 - > MSP CS1 students do just as well as OLP CS1 students in an OLP CS2

- > Other universities show similar results

- > Students pivot when stuck
 - > Workflow charts provide deep insight into student engagement



References and Publications

References

- Watson, C. and Li, F. "Failure Rates in Introductory Programming Revisited," iTiCSE, 2014
<http://dro.dur.ac.uk/19223/1/19223.pdf%3FD0D10%2Bd74ks0%2Bdcs0lw>
- J. Bennedsen and M. E. Caspersen: Failure rates in introductory programming: 12 years later. ACM Inroads 10, 2 (June 2019), 30–36
- [1] N. Nagappan, L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, S. Balik: Improving the CS1 Experience with Pair Programming. ACM SIGCSE Technical Symposium on Computer Science Education, 2013.
- [2] R. J. Enbody and W. F. Punch: Performance of Python CS1 Students in Mid-level non-Python CS Courses. ACM SIGCSE Technical Symposium on Computer Science Education, 2010.
- [3] R. Das, U. Z. Ahmed, A. Karkare, and S. Gulwani: Prutor: A System for Tutoring CS1 and Collecting. 2016
- [4] S. Sharmin, D. Zingaro, L. Zhang, and C. Brett: Impact of Open-Ended Assignments on Student Self-Efficacy in CS1. In Proceedings of the ACM Conference on Global Computing Education (CompEd '19), 2019.
- [5] F. Rosales, A. García, S. Rodríguez, J. L. Pedraza, R. Méndez, and M. M. Nieto: Detection of Plagiarism in Programming Assignments, 2008
- [6] D. E. Stevenson and P. J. Wagner: Developing Real-World Programming Assignments for CS1. ITiCSE'06, 2006.
- [7] R. Anderson, M. D. Ernst, R. Ordóñez, P. Pham, S. A. Wolfman: Introductory Programming Meets the Real World: Using Real Problems and Data in CS1. ACM SIGCSE Technical Symposium on Computer Science Education, 2016
- [8] A. Goldman: Using Daily Missions to Promote Incremental Progress on Programming Assignments. Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University
- [9] R. Giguette: Pre-Games: Games Designed to Introduce CS1 and CS2 Programming Assignments. ACM SIGCSE Technical Symposium on Computer Science Education, 2013

References and Publications

› Included Publications

- › J.M. Allen, F. Vahid, K. Downey, A. Edgcomb. Weekly Programs in a CS1 Class: Experiences with Auto-graded Many-small Programs (MSP). ASEE Annual Conference, 2018. (Best paper nominee)
- › J.M. Allen, F. Vahid, A. Edgcomb, K. Downey, and K. Miller. An Analysis of Using Many Small Programs in CS1, ACM SIGCSE Technical Symposium on Computer Science Education, 2019.
- › J.M. Allen and F. Vahid. Concise Graphical Representations of Student Effort on Weekly Many Small Programs, ACM SIGCSE Technical Symposium on Computer Science Education, 2020. (Pending, but accepted)
- › J.M. Allen, F. Vahid, K. Downey, K. Miller, and A. Edgcomb. Many Small Programs in CS1: Usage Analysis from Multiple Universities, Proceedings of ASEE Annual Conference, 2019.
- › J.M. Allen and F. Vahid. Analyzing Pivoting Among Weekly Many Small Programs in a CS1 Course, Proceedings of ASEE Annual Conference, 2020

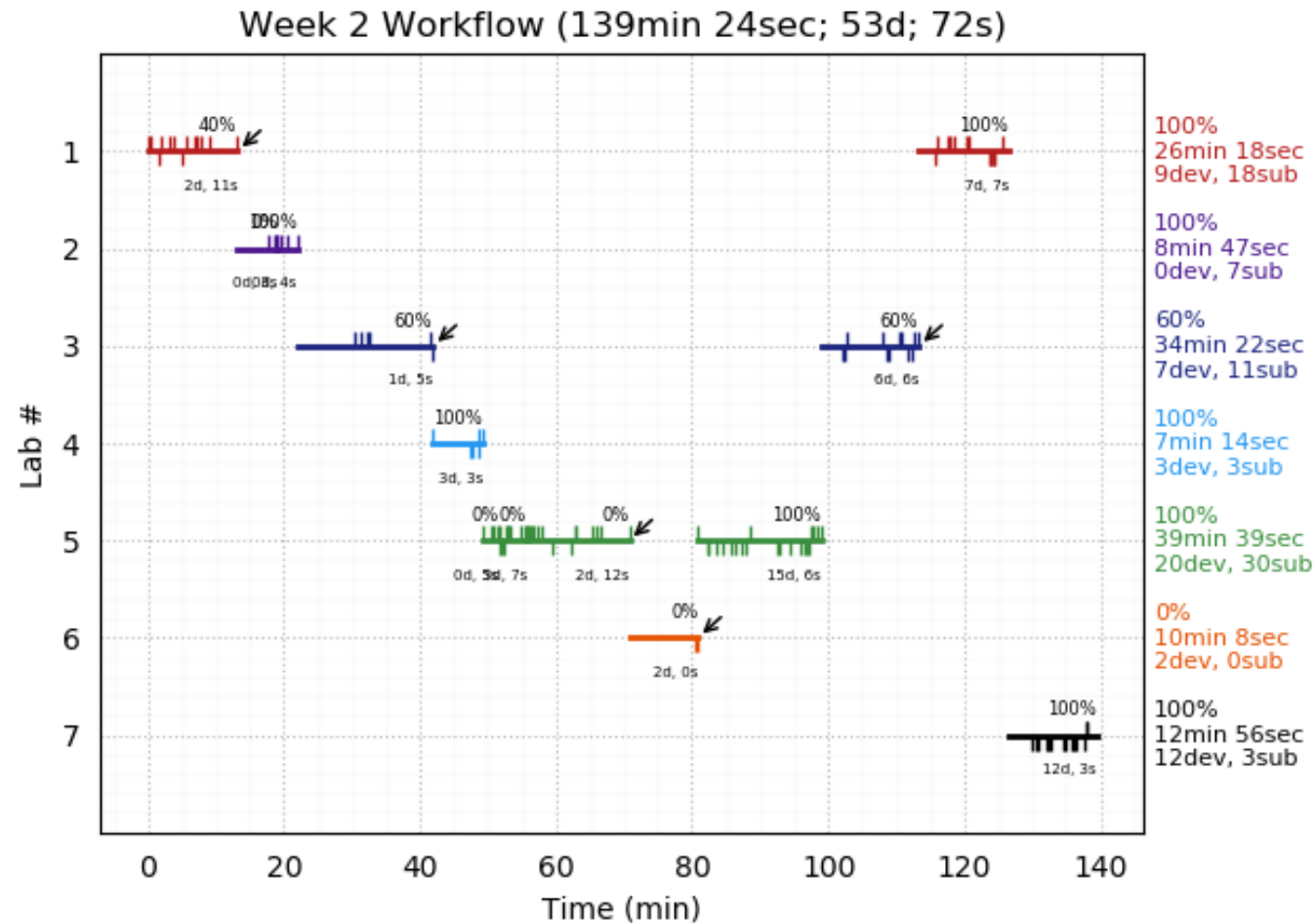
› Other Publications

- › F. Vahid, J.M. Allen, A. D. Edgcomb, and R. Lysecky. Using the free Coral language and simulator to simplify first-year programming courses, Proceedings of FYEE Annual Conference, 2020.
- › F. Vahid and J.M. Allen. An online course for freshmen? The evolution of a successful online CS1 course, Proceedings of FYEE Annual Conference, 2020.
- › J.M. Allen and F. Vahid. Teaching Coral before C++ in a CS1 Course, Proceedings of ASEE Annual Conference, 2020.
- › J.M. Allen and F. Vahid. Experiences in Developing a Robust Popular Online CS1 Course for the Past 7 Years, Proceedings of ASEE Annual Conference, 2020.

› Other Talks

- › J.M. Allen, F. Vahid, K. Downey, and A. Edgcomb. Weekly Programs in CS 1: Experiences with Many-Small Auto-Graded Programs, UCR Symposium, 2018.

Appendix A: Student workflow chart



Appendix B: Online tools

UCR - zyBooks zyLab Analysis

Lorem ipsum.

Questions? Email: analysis-support@cs.ucr.edu

[Interactive website](#)

- Upload zyBooks zyLab log: No file chosen
- sample_1.csv ([View file](#)) ([Download file](#)) **Week 3**
- sample_2.csv ([View file](#)) ([Download file](#))

Analysis options

- Run all analyses
- Run time spent analysis
- Run score analysis

