Does Reducing Curricular Complexity Impact Student Success in Computer Science?

SIGCSE Technical Symposium 2025

Sumukhi Ganesan
Khoury College of Computer Sciences
Northeastern University
ganesan.su@northeastern.edu

Albert Lionelle
Khoury College of Computer Sciences
Northeastern University
a.lionelle@northeastern.edu

Catherine Gill
Center for Inclusive Computing
Northeastern University
c.gille@northeastern.edu

Carla Brodley
Center for Inclusive Computing
Northeastern University
c.brodley@northeastern.edu

Northeastern
Khoury College of
Computer Sciences

Northeastern University
Center for Inclusive Computing

Colorado State University

Curricular Redesign

2021: Colorado State University new BSCS curriculum

Reasons of the revamp:

- Course prerequisite overly complicated (2018)
  - Advising / forcing when something is taken
  - Hidden prerequisite nightmare (e.g. 21 credit minor in CS was 40+ credits)
- Wanted to add
  - More flexibility to students
  - Concentrations
  - Additional courses

Q1: What did they do?

- What are some metrics to help with comparison? (curricular complexity)

Q2: Did it hurt or help retention of students?

Q3: Did it attract more students from undecided?

Redesign Philosophy

Reduce Prerequisites to:

- Only what is needed for the majority of content
- Remove any prereq that is about ‘when’ not ‘what’
- Make some prerequisites either/or for later courses

Pathway approach

- Try to let students progress even if stuck on topics
- All pathways required, but keep mostly independent
  - Algorithms (math kept to this pathway)
  - Software Engineering
  - Systems

Add required ethics course (Gen Ed)

Add optional CS 0 course (Gen Ed)

Add additional upper division options

Course Outcomes – Fixed
Intentionally no change!

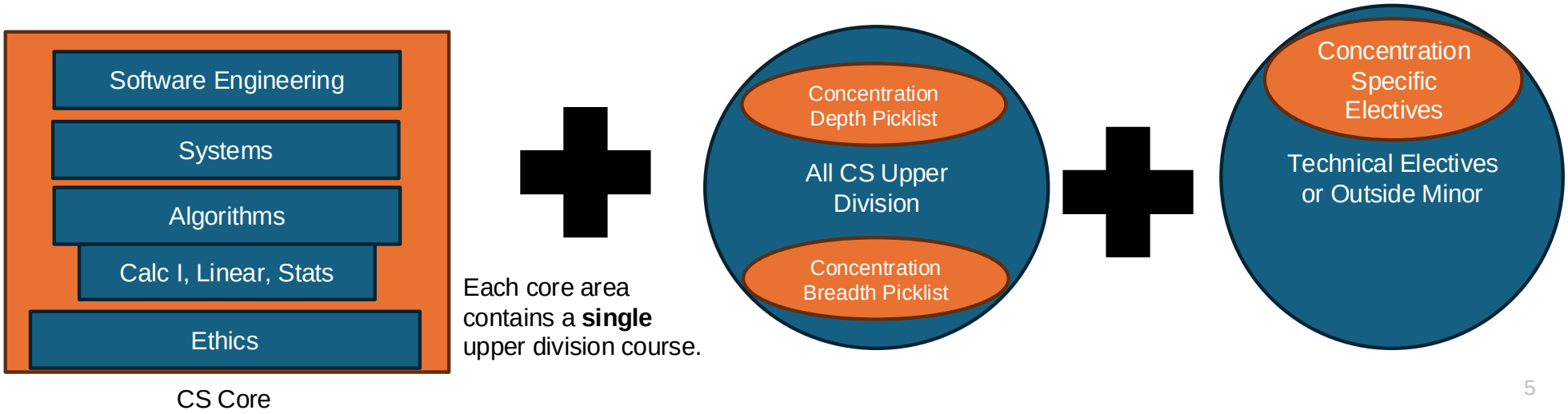
Overall Design

Core Requirement

CS Upper Division
- Minimize Prerequisites

Technical Electives

Concentrations added – specific to primary research areas in the department
- General, AI/ML, Cybersecurity, Education, Human Centered Computing, Software Engineering, Systems



How did this change the general BSCS?

	Old Curriculum (2017)	# of Credits	New Curriculum (2021)	# of Credits	Change Summary?
Core Lower Division Courses	CS1, CS2, Discrete Math, Computer Org, Software Development	5 courses 20 credits	CS 1, CS 2, Discrete Math, Computer Org, Software Dev, Ethics (GE), Optional: CS 0 (GE)	6-7 courses 23-26 Credits 3-6 credits overlap with general university requirements (GE)	**INCREASED** + 1-2 CS courses Overlapped with GE * Improves transfers
Core Upper Division	Algorithms, Operating Sys, Security, Software Eng, 4 'pick list' CS courses	8 courses 28 credits	Algorithms, Operating Sys, Software Eng, 6 'pick list' CS courses	9 courses 31-32 credits	**INCREASED** + 1 course + added more flexibility
Math	Calc I, Calc II, Stats/Probability, Linear Algebra	4 courses 13-15 credits	Calc I, Stats/Probability, Linear Algebra	3 courses 11 credits	REDUCED Moved Calc II as optional!

Additional modifications to technical electives and natural science credits – in paper
Calc II was moved to technical electives, but *required for AI concentration*

Dataset

Six years of student data

- Fall 2017 – Fall 2023

All students taking CS or math courses

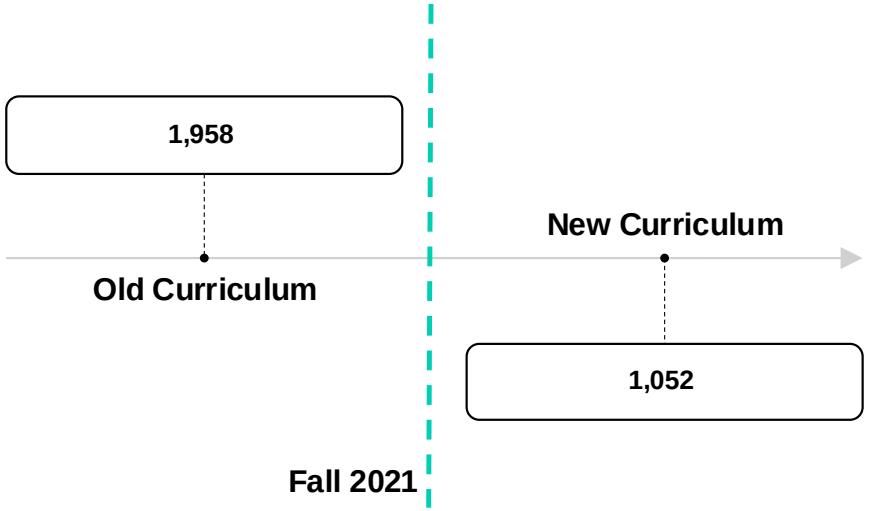
- ~40,000 students

Students who majored in BSCS

- Old program (Fa17- Fa21) – 1958
- New program (Fa21- Fa23) – 1052

Majored in Undeclared Seeking CS (USCS)

- Old program (Fa17- Fa21) – 307
- New program (Fa21- Fa23) – 214



Measuring the Impact

Degree Changes

- Curricular Analytics for comparing changes
- Degree Structure Complexity
- Course Centrality

Attrition / Retention

- Declared CS but later left.
- Terminal courses
  - Last course taken before leaving
- Correlation between course complexity and attrition?

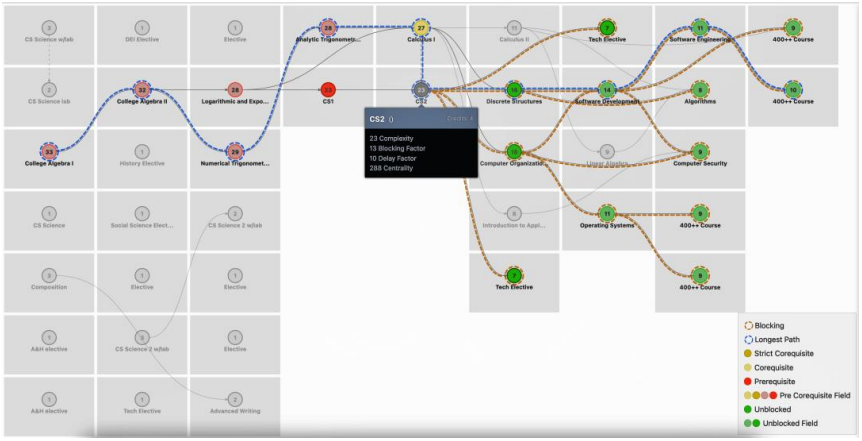
Attraction

- Undeclared Seeking Computer Science (USCS)
- Compared conversion rates USCS → CS Major

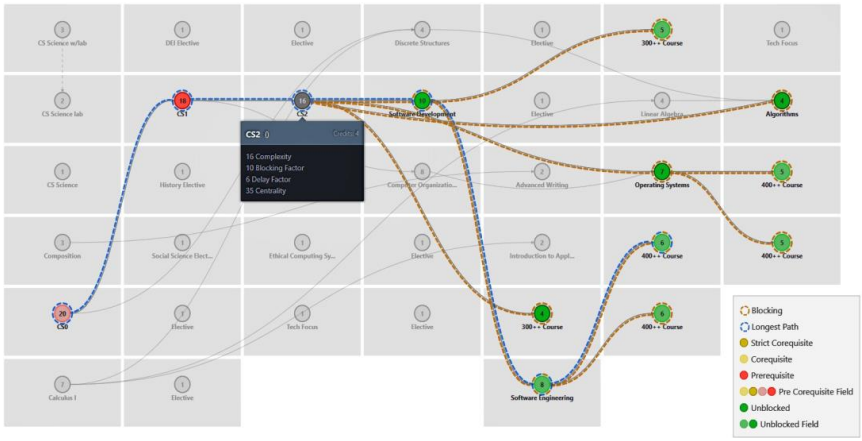
curricularanalytics.org

Measuring Degree Structure Changes

60% reduction in program structural complexity!

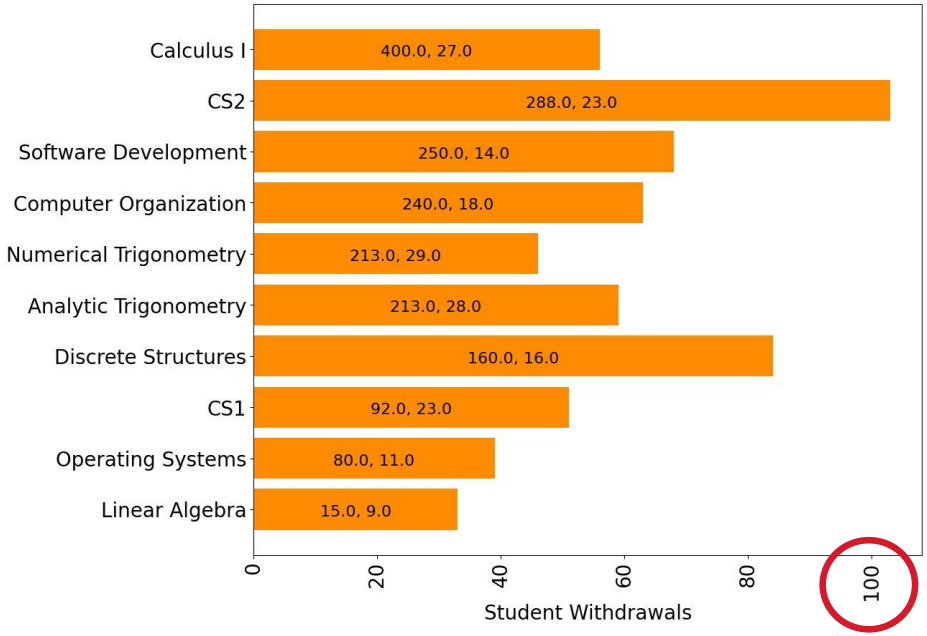


Complexity = 415
multiple blocking courses

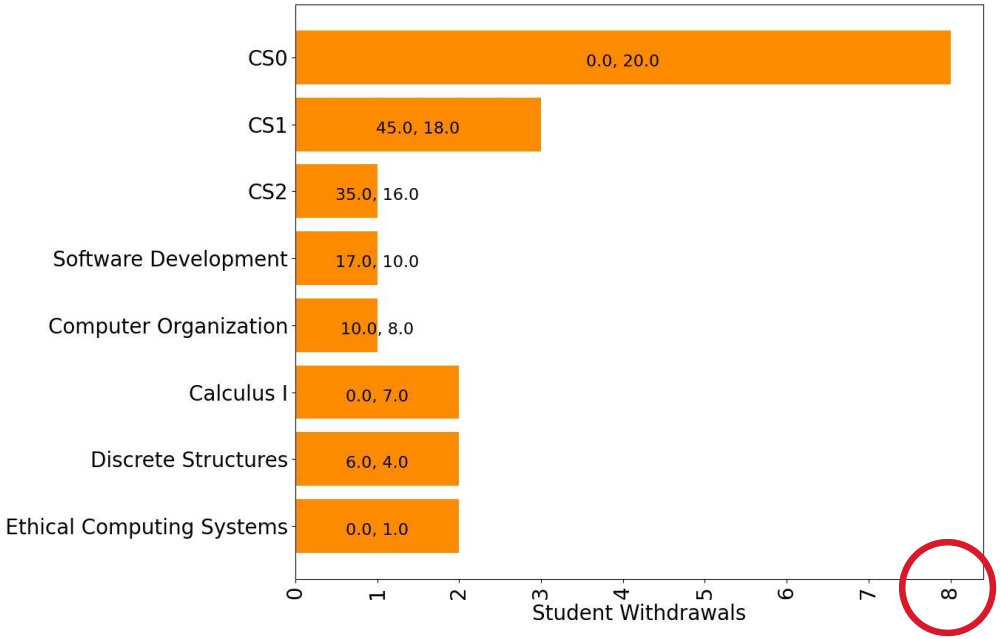


Complexity = 164
minimal blocking courses

Terminal Courses?



Driven out by Math



Deciding after CS0

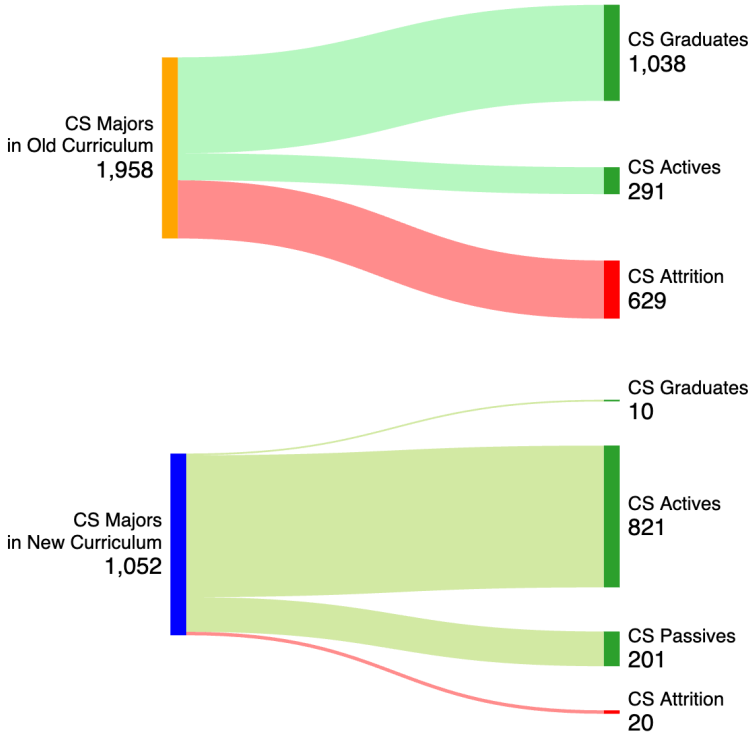
Did retention/attraction increase?

Program retention increased!
- 67% Old degree
- 98% New degree

Undeclared seeking CS increased attraction!
- 47% converted in old degree
- 69% converted in new degree

Overall numbers increasing
- Fall 2015, 725, 12% women (88)
- Fall 2023, 1022, 19% women (198)

Students still reporting similar job placements



Key Takeaways

- ✅ Reducing Complexity works
  - 98% retention rate (up from 67%).
  - 69% Undeclared converted to CS (up from 47%).
- ✅ Flexibility does not compromise Rigor
  - Course outcomes & content remained unchanged.
  - Pathways model allow continued progress
- ✅ Math need not be a barrier
  - Math-heavy prerequisites were moved until later
  - CS 0 gave students a *computer science focused entry point*.
  - Fewer students left CS due to calculus requirements.
- ✅ Use Curricular Complexity tools
  - Identify easier, helping remove bottlenecks.

Presentations Available @
https://github.com/NeuCurricularAnalytics/papers

Paper Available @
https://dl.acm.org/doi/10.1145/3641554.3701915

Thank you!
And
Questions

Northeastern University
Center for Inclusive Computing

Northeastern University
Khoury College of Computer Sciences

Colorado State University