Draft Report of the Special Committee of the Council for Higher Education on Defining Study Programs in Computer Engineering, Electrical Engineering, Software Engineering, and Computer Science

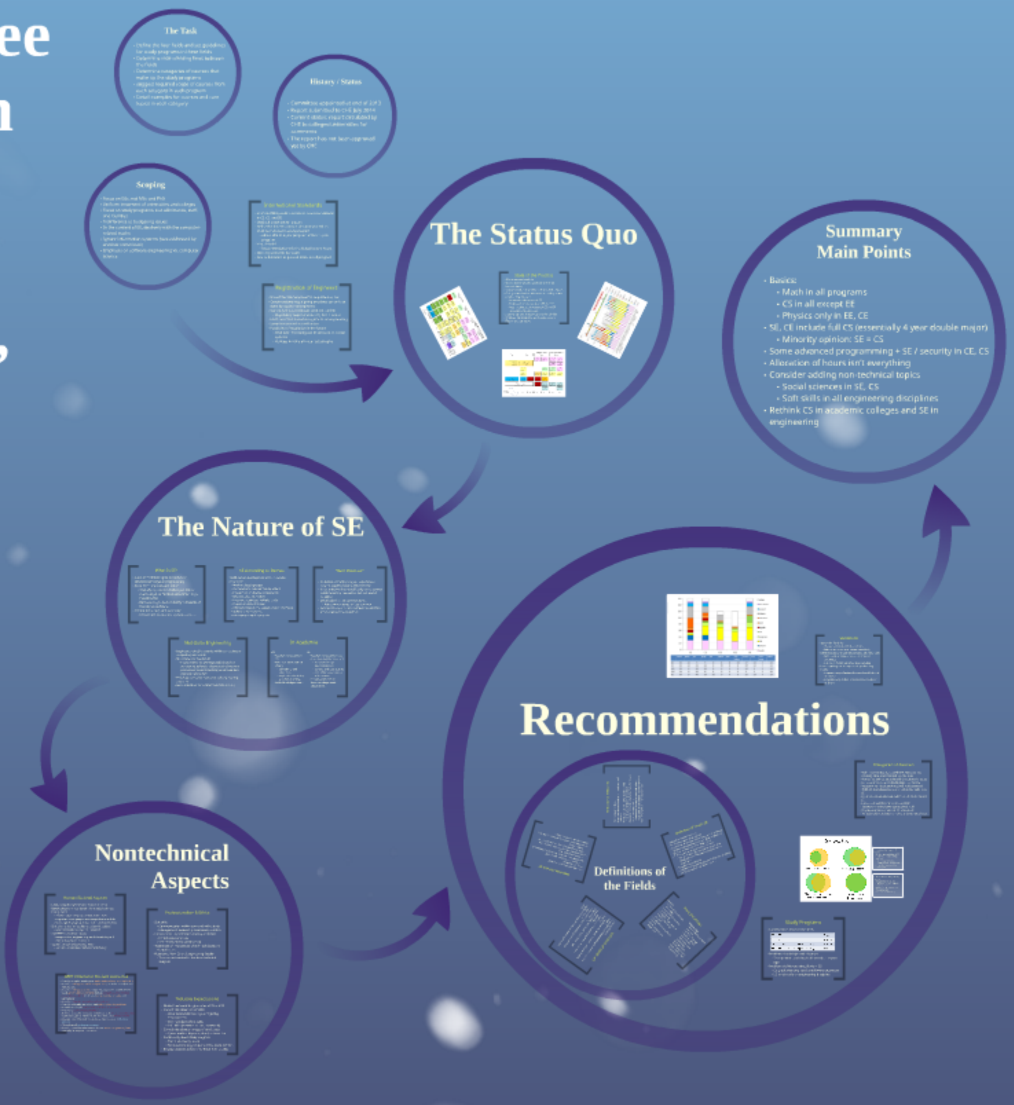
General principles
(not specific program)

Under discussion
Not approved yet

Responsible for academic study programs
(not vocational training)

Many existing programs
Many requests for new programs
Lots of overlap

Prof. Dror Feitelson, HU (chair)
Prof. Catriel Beeri, HU
Prof. Yehuda Leviatan, Technion
Prof. Sivan Toledo, TAU
Dr. Nathan Weiss, IAI/Elta



Summary
Main Points

- Basics:
  - Math in all programs
  - CS in all except EE
  - Physics only in EE, CE
- SE, CE include full CS (essentially 4 year double major)
  - Minority opinion: SE = CS
- Some advanced programming + SE / security in CE, CS
- Allocation of hours isn't everything
- Consider adding non-technical topics
  - Social sciences in SE, CS
  - Soft skills in all engineering disciplines
- Rethink CS in academic colleges and SE in engineering

General principles
(not specific program)

Draft Report of the Special Committee of the Council for Higher Education on Defining Study Programs in Computer Engineering, Electrical Engineering, Software Engineering, and Computer Science

Under discussion
Not approved yet

Responsible for academic study programs
(not vocational training)

Many existing programs
Many requests for new programs
Lots of overlap

Prof. Dror Feitelson, HU (chair)
Prof. Catriel Beeri, HU
Prof. Yehuda Leviatan, Technion

- Define the
for study p
- Determine
the fields
- Determine
make up th
- suggest ne
each catego
- Detail exam
topics in ea
Scopin
- Focus on BSc, not MSc and Ph
- Uniform treatment of univers
- Focus on study programs, not
and facilities
- Indifference to budgeting iss
- In the context of EE, deal only
related tracks
- Ignore information systems (
another committee)
- Emphasis on software engine
science

General principles
(not specific program)

Special Committe

Higher Education

Responsible for academic study programs
(not vocational training)

dy Programs in

eering, Electrical

Many existing programs

General principles
(not specific program)

Special Committee
Higher Education

Responsible for academic study programs
(not vocational training)

dy Programs in

ncil for Higher Educatio

Responsible for academic study programs
(not vocational training)

ing Study Programs in r Engineering, Electrical

Many existing programs
Many requests for new programs
Lots of overlap

g, Software Engineering Computer Science

Feitelson, HU (chair)

Draft Report of the Coun on Defini

Under discussion
Not approved yet

Draft Report of the Special Committee of the Council for Higher Education on Defining Study Programs in Computer Engineering, Electrical Engineering, Software Engineering, and Computer Science

General principles
(not specific program)

Under discussion
Not approved yet

Responsible for academic study programs
(not vocational training)

Many existing programs
Many requests for new programs
Lots of overlap

Prof. Dror Feitelson, HU (chair)
Prof. Catriel Beeri, HU
Prof. Yehuda Leviatan, Technion

- Define the
for study p
- Determine
the fields
- Determine
make up th
- suggest re
each catego
- Detail exam
topics in ea

Scopin

- Focus on BSc, not MSc and Ph
- Uniform treatment of univers
- Focus on study programs, no
and facilities
- Indifference to budgeting iss
- In the context of EE, deal only
related tracks
- Ignore information systems (
another committee)
- Emphasis on software engine
science

The Task

- Define the four fields and set guidelines for study programs in these fields
- Determine main dividing lines between the fields
- Determine categories of courses that make up the study programs
- suggest required scope of courses from each category in each program
- Detail examples for courses and core topics in each category

Scoping

- Focus on BSc, not MSc and PhD
- Uniform treatment of universities and colleges
- Focus on study programs, not admittance, staff, and facilities
- Indifference to budgeting issues
- In the context of EE, deal only with the computer-related tracks
- Ignore information systems (was addressed by another committee)
- Emphasis on software engineering vs. computer science

History / Status

- Committee appointed at end of 2013
- Report submitted to CHE July 2014
- Current status: report circulated by CHE to colleges/universities for comments
- The report has not been approved yet by CHE

International Standards

- ACM and IEEE publish curricular recommendations in CS, CE, and SE
- Updated about every 10 years
- Define the lowest common denominator which must be included in every program
  - About 28% of 3-year program, 21% of 4-year program
- Very detailed
  - Recommendations for individual lecture hours
- Not very useful for our work
- Goal is definition of general topics in full program

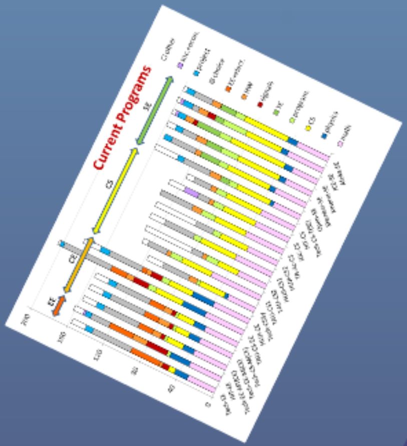
Registration of Engineers

- Use of the title “engineer” is regulated by law
- Certain actions (e.g. signing on plans) can only be done by registered engineers
- Currently this is irrelevant to EE, CE, and SE
  - Regulatory requirements only for EE-power
- Most countries do not recognize SE as engineering
- Some initiatives for certification
- Possibility of regulation in the future
  - With ever increasing use of software in critical systems
  - Perhaps in wake of major catastrophe

The Status Quo

State of the Practice

- All universities teach CS
- Most universities also teach EE and/or CE (except Haifa)
- Few universities teach SE (Technion, BGU, Open)
- Colleges are divided into academic colleges and engineering colleges
  - Academic colleges teach CS
  - Engineering colleges teach EE and SE
  - Result: colleges do not teach CS and SE (exception: Machon Lev)
- Study programs in each topic rather similar
- Engineering programs are 4 years, CS is 3 (Technion also has 4)



האוניברסיטה העברית – הנדסת מחשבים

שנה ד'	שנה ג'	שנה ב'	שנה א'

State of the Practice

- All universities teach CS
- Most universities also teach EE and/or CE (except Haifa)
- Few universities teach SE (Technion, BGU, Open)
- Colleges are divided into academic colleges and engineering colleges
  - Academic colleges teach CS
  - Engineering colleges teach EE and SE
  - Result: colleges do not teach CS and SE (exception: Machon Lev)
- Study programs in each topic rather similar
- Engineering programs are 4 years, CS is 3 (Technion also has 4)

מכללת כנרת – הנדסת תוכנה

שנה א'		שנה ב'		שנה ג'		שנה ד'	
חדו"א 1 10-101, 5 נ"ז	חדו"א 2 10-102, 5 נ"ז		משוואות דיפרנציאליות 10-104, 3.5 נ"ז		XML (נבחר) 14-323, 2.5נ"ז		אנדרואיד (נבחר) 12-069, 2.5נ"ז
אלגברה 1 10-103, 5 נ"ז	מתמטיקה דיסקרטית 10-110, 2.5 נ"ז	הסתברות 10-211, 4 נ"ז	לוגיקה 12-220, 2.5 נ"ז	בינה מלאכותית 14-327, 3.5 נ"ז	מערכות לומדות 14-316, 3.5 נ"ז	גרפיקה ועיבוד תמונה 14-426, 3 נ"ז	אותות ומערכות 14-320, 4 נ"ז
מבוא למדעי המחשב 14-110, 4.5 נ"ז	תכנות מונחה עצמים 12-111, 4 נ"ז	תכנות מערכות 14-219, 4 נ"ז	שפות תכנות 14-318, 4 נ"ז	ביואינפורמטיקה 14-315, 3 נ"ז	רשתות מחשבים 12-331, 2.5 נ"ז	רשתות מחשבים מתקדם 14-428, 3 נ"ז	ארכיטקטורה של תוכנה 14-432, 3 נ"ז
מערכות ספרתיות 14-112, 3 נ"ז		ארגון ותכנות מחשב 14-211, 3 נ"ז	מבנה מחשבים 14-213, 3 נ"ז	מערכות הפעלה 14-317, 4 נ"ז	מערכות הפעלה מתקדם 14-321, 3 נ"ז	מערכות מבוזרות 14-424, 4 נ"ז	אימות חומרה ותוכנה 14-430, 3 נ"ז
		מבני נתונים 14-212, 3 נ"ז	אלגוריתמים 14-218, 3 נ"ז	חישוביות 12-326, 3 נ"ז		ניהול פרוייקטי תוכנה 14-431, 3 נ"ז	אבטחת מידע ותקשורת 14-425, 2.5 נ"ז
	פיסיקה 1 10-124, 4.5 נ"ז	פיסיקה 2 10-225, 4.5 נ"ז	אוטומטים שפות פורמאליות 14-214, 3 נ"ז	מסדי נתונים 12-324, 3 נ"ז	הנדסת מערכות עתירות תכנה 14-322, 4 נ"ז	הנדסת תכנה זריזה 14-433, 3 נ"ז	הבטחת איכות תוכנה 14-421, 3.5 נ"ז
חברה ומדינה 2 נ"ז	כלכלה למהנדסים 14-319, 2.5 נ"ז	חברה ומדינה 2 נ"ז	ארגון ועיבוד קבצים 12-222, 3 נ"ז	מנשק אדם מחשב 13-323, 2.5 נ"ז	יזמות 10-041, 2.5 נ"ז	פרוייקט 1 14-423, 4 נ"ז	פרוייקט 2 14-427, 4 נ"ז
19.5נ"ז	18.5 נ"ז	20.5 נ"ז	22 נ"ז	19 נ"ז	18 נ"ז	20 נ"ז	22.5 נ"ז

ס"ה 160 נ"ז

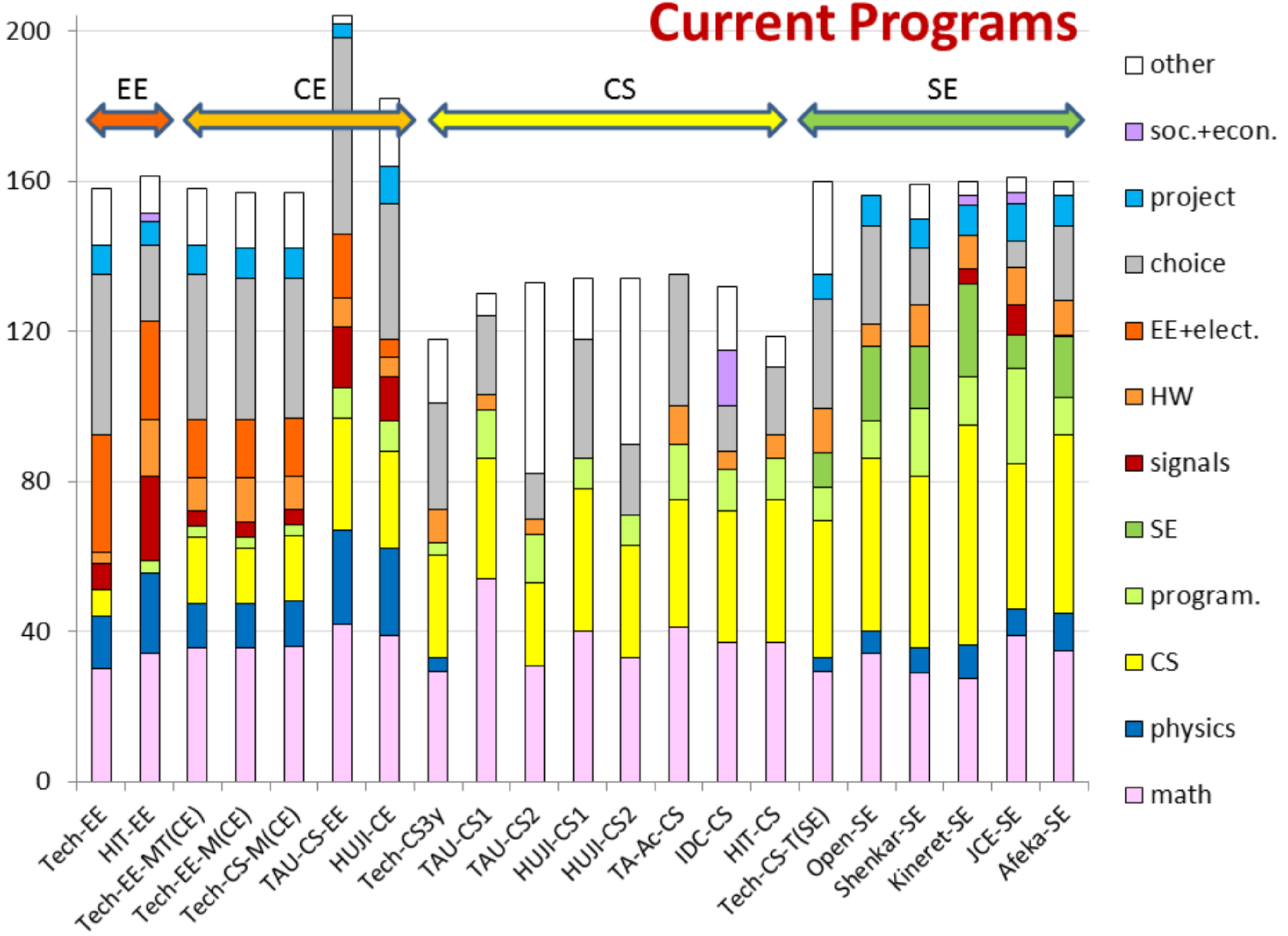
האוניברסיטה העברית – הנדסת מחשבים

שנה א'		שנה ב'		שנה ג'		שנה ד'
אלגברה לפיסיקאים 80152, 6 נ"ז	אינפי להנדסה ומדעים 80177, 7 נ"ז	הסתברות וסטט. למחשב 52005, 6 נ"ז	פונקציות מרוכבות 80314, 4 נ"ז			
מתמטיקה שימושית 1 80114, 6 נ"ז	מתמטיקה שימושית 2 80157, 6 נ"ז			חובת בחירה 2 אשכולות לפחות 15 נ"ז כ"א לפחות 36 נ"ז ס"ה 1. מערכות מידע ואינטרנט 2. מערכות נבונות 3. מערכות ראיה וגרפיקה 4. מערכות חומרה 5. מערכות תקשורת		
מתמטיקה דיסקרטית 80181, 4 נ"ז	תכנות מונחה עצמים 67125, 4 נ"ז	C/C++ 67316/7, 4נ"ז			מערכות הפעלה 67808, 4 נ"ז	
מבוא למדעי המחשב 67101, 7 נ"ז	מבני נתונים 67109, 4 נ"ז	אלגוריתמים 67504, 5 נ"ז	מבנה המחשב 67200, 5 נ"ז		חישוביות וסיבוכיות 67521, 5 נ"ז	
		מבוא להנדסת חשמל 83335, 5 נ"ז	מערכות ליניאריות 67310, 5 נ"ז	עיבוד אותות ספרתיים 67567, 4 נ"ז	פרויקט א' 67546, 2 נ"ז	פרויקט ב' 67547, 8 נ"ז
מכניקה קלאסית 77115, 7 נ"ז	חשמל ומגנטיות להנדסה 83313, 6 נ"ז	גלים ויסודות הפיסיקה 83325, 6 נ"ז	אותות ומשתנים אקראיים 67652, 4 נ"ז	בחירה כללית 18 נ"ז (מתוכם עד 12 מחוץ לפקולטה מתוכם 8 "אבני פינה")		
סדנת עזר בפיסיקה 83110, 1 נ"ז		מעבדת פיסיקה להנדסה 83315, 3 נ"ז				
31 נ"ז	27 נ"ז	29 נ"ז	18נ"ז	4 נ"ז	9 נ"ז	

	בחובה	בחוץ
חובה	118	
פרוייקט	10	
אשכולות	36	
בחירה	0	18
ס"ה	**164**	**18**

ס"ה 180 נ"ז

Current Programs



The Nature of SE

What Is SE?

- A set of methodologies for software development *beyond programming*
- Born from "the software crisis"
  - A lot of projects are challenged (late or over budget) or fail (cancelled after large investments)
  - But booming hi-tech industry + all walks of like rely on software
- Critical for society and economy
  - Infrastructure, medical systems, cars, ...

SE According to Parnas

"multi-person development of multi-version programs"

- Work in (large) groups
- Contend with code written by others
- Integration of diverse components
- Effective documentation
- Extensive testing at multiple levels
- Requirements extraction
- Characterization and design of user interfaces
- Software maintenance
- Managing complex projects

"Best Practices"

- Collection of methodologies / experience / insights acquired laboriously over time
- No quantitative theoretical background: cannot calculate testing needed for desired level of reliability
- Often looks like just common sense
  - But common sense isn't so common
- Perhaps stronger human-related aspects than other engineering disciplines

Not Quite Engineering

- Engineers trained to operate within constraints of compulsory standards
- SE is more of a free for all
  - Hoare (1981): "*In any respectable branch of engineering, failure to observe such elementary precautions [bounds checking] would have long been against the law.*"
- Widely perceived as not (yet) a real engineering discipline
- Some initiatives for certification (NCEES, IEEE)

In Academia

CS

- Faculty have degrees in CS
- Based on mathematical theory
  - Complexity and algorithms
  - Applications in vision, graphics, learning, ...
- Established department

SE

- Faculty have degrees in CS
- based on empirical research
  - Proof of concept demonstrations
  - Surveys and case studies
  - Controlled experiments with humans
- Growing out of CS to become independent department

What Is SE?

- A set of methodologies for software development *beyond programming*
- Born from “the software crisis”
    - A lot of projects are challenged (late or over budget) or fail (cancelled after large investments)
    - But booming hi-tech industry + all walks of like rely on software
- Critical for society and economy
    - Infrastructure, medical systems, cars, …

SE According to Parnas

“multi-person development of multi-version programs”

- Work in (large) groups
- Contend with code written by others
- Integration of diverse components
- Effective documentation
- Extensive testing at multiple levels
- Requirements extraction
- Characterization and design of user interfaces
- Software maintenance
- Managing complex projects

"Best Practices"

- Collection of methodologies / experience / insights acquired laboriously over time
- No quantitative theoretical background: cannot calculate testing needed for desired level of reliability
- Often looks like just common sense
    - But common sense isn't so common
- Perhaps stronger human-related aspects than other engineering disciplines

Not Quite Engineering

- Engineers trained to operate within constraints of compulsory standards
- SE is more of a free for all
    - Hoare (1981): “*In any respectable branch of engineering, failure to observe such elementary precautions [bounds checking] would have long been against the law*.”
- Widely perceived as not (yet) a real engineering discipline
- Some initiatives for certification (NCEES, IEEE)

In Academia

CS

- Faculty have degrees in CS
- Based on mathematical theory
  - Complexity and algorithms
  - Applications in vision, graphics, learning, …
- Established department

SE

- Faculty have degrees in CS
- based on empirical research
  - Proof of concept demonstrations
  - Surveys and case studies
  - Controlled experiments with humans
- Growing out of CS to become independent department

Nontechnical Aspects

Human/Societal Aspects

- Many computer systems are human-oriented
- Developing them may benefit from social sciences background
    - Psychology: how people think and behave
    - Cognition: what people can comprehend and do
    - Sociology: how groups of people react and behave
- Snir: alternative for "additional scientific subject"
    - Science based on empirical research
- ACM/IEEE: one of four bases
    - computation, engineering, math+statistics, and psychology+social sciences
- Opposition by hardcore engineers
    - Lack of quantitative mathematical theory

Professionalism & Ethics

- Soft skills
    - Communication within team and with clients
    - Management, leadership, teamwork, conflicts
- Commitment to profession, society, and client
    - Ethical considerations
    - Self improvement and learning
- Awareness of implications of technical decisions
    - Legal issues
- Resistance from CS and engineering faculty
    - Time better invested in hardcore technical subjects

ABET Criteria for Student Outcomes

- An ability to apply knowledge of mathematics, science, and engineering
- An ability to design and conduct experiments, as well as to analyze and interpret data
- An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- An ability to function on multidisciplinary teams
- An ability to identify, formulate, and solve engineering problems
- An understanding of professional and ethical responsibility
- An ability to communicate effectively
- the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- A recognition of the need for, and an ability to engage in life-long learning
- A knowledge of contemporary issues
- An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Industry Expectations

- Similar treatment for graduates of CS and SE
- Complaints about lack of skills
    - Work habits (unit testing, configuring environment)
    - Cutting edge technologies
    - Soft skills (communication, teamwork)
- Complaints about shortage of candidates
    - Cyber, mobile, big data, cloud, automation
- Controversy about study program
    - Too much theory, math
    - But academic degree appreciated (used as filter)
- Employ students before they finish their studies

Human/Societal Aspects

- Many computer systems are human-oriented
- Developing them may benefit from social sciences background
  - Psychology: how people think and behave
  - Cognition: what people can comprehend and do
  - Sociology: how groups of people react and behave
- Snir: alternative for “additional scientific subject”
  - Science based on empirical research
- ACM/IEEE: one of four bases
  - computation, engineering, math+statistics, and psychology+social sciences
- Opposition by hardcore engineers
  - Lack of quantitative mathematical theory

Professionalism & Ethics

- Soft skills
  - Communication within team and with clients
  - Management, leadership, teamwork, conflicts
- Commitment to profession, society, and client
  - Ethical considerations
  - Self improvement and learning
- Awareness of implications of technical decisions
  - Legal issues
- Resistance from CS and engineering faculty
  - Time better invested in hardcore technical subjects

ABET Criteria for Student Outcomes

- An ability to apply knowledge of mathematics, science, and engineering
- An ability to design and conduct experiments, as well as to analyze and interpret data
- An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability
- An ability to function on multidisciplinary teams
- An ability to identify, formulate, and solve engineering problems
- An understanding of professional and ethical responsibility
- An ability to communicate effectively
- the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context
- A recognition of the need for, and an ability to engage in life-long learning
- A knowledge of contemporary issues
- An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

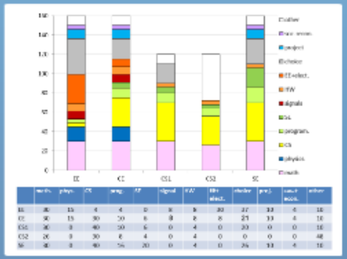
Industry Expectations

- Similar treatment for graduates of CS and SE
- Complaints about lack of skills
  - Work habits (unit testing, configuring environment)
  - Cutting edge technologies
  - Soft skills (communication, teamwork)
- Complaints about shortage of candidates
  - Cyber, mobile, big data, cloud, automation
- Controversy about study program
  - Too much theory, math
  - But academic degree appreciated (used as filter)
- Employ students before they finish their studies

ices"

ns / experience /
ly over time
background: cannot
v desired level of

on sense
t so common
elated aspects than
es



	math.	phys.	CS	prog.	SE	signal	HW	EE+ elect.	choice	proj.	soc.+ econ.	other
EE	30	15	4	4	0	8	8	30	37	10	4	10
CE	30	15	30	10	6	8	8	8	21	10	4	10
CS1	30	0	40	10	6	0	4	0	20	0	0	10
CS2	26	0	30	8	4	0	4	0	0	0	0	48
SE	30	0	40	16	20	0	4	0	26	10	4	10

Variations

- Option for flexibility
  - Changes of up to 25% in each area
  - But do not scrap small areas completely
- Differentiation and competition possible / desirable
  - Different institutions can select different emphases
  - E.g. CS with theoretical vs. practical slant
- Not everything can be expressed by allocating hours
  - Emphasis on professionalism and work habits in all courses
  - Emphasis on practical exercises and feedback on them

Recommendations

Categories of Courses

- Math – discrete (CS, SE, CE) and continuous (EE, CE) including calculus and probability / statistics
- Physics – EE and CE, possible scientific subject for CS, SE
- CS – SE as in CS, CE as in double major, EE minimal
- Programming – basic OOP required in all, advanced (OOD, UI, event-based, parallel) in SE and partially in CS, CE
- SE – intro and security required in CS, CE, much more in SE
- Signals – CE and EE, information and DSP
- Hardware – minimal coverage required in all
- EE and electronics – mainly in EE, some in CE
- Professionalism and society – skills and theoretical basis

Definitions of the Fields

Definition of Fields: CE

Definition of Fields: SE

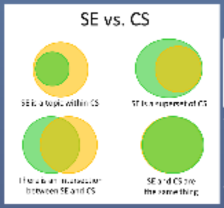
- Debatable engineering field
- Focus on software development
- Description, definition, and construction of software systems and applications, largely divorced from the physical platform on which they will run
- Difference from programming is in scale
  - You can build a doghouse from some wood planks yourself, but need an engineer for a skyscraper
- Based on mastery of CS
- Additional engineering aspects include requirements modeling, design, verification, QA, ...

Basic Distinction

Definition of Fields: CS

Definition of Fields: EE

SE vs. CS


SE is a topic within CS
SE is a superset of CS
There is an intersection between SE and CS
SE and CS are the same thing

SE is a superset of CS (majority opinion)
- SE required practically all of CS
- There is significant additional material
  - Planning, estimation, configuration, management, ...
- This is sometimes justifying another year

SE and CS are the same (minority opinion)
- CS graduates work as SE
- Implication: 3-year program is sufficient for SE
- Make this explicit: CS should include some SE
- "SE" is just for marketing

Study Programs

- Largely correspond to the fields

CS	CS	3 years
SE	SE, CS	4 years
CE	EE, electronics, signal processing, CS	4 years
EE	EE, electronics, signal processing	4 years

- Problem of rapid dynamic changes
  - Smartphones and cloud did not exist 10 years ago
- Problem of differentiating SE from CS
  - CS graduates employed as software engineers
  - SE is not really an engineering discipline

Definitions of the Fields

Definition of Fields: CE

- An engineering field
- Focus on the combination of hardware and software
- Tendency toward higher levels than EE
  - Computer systems, not devices
- Co-design of hardware platform and software system or application
  - Interact and complement each other
  - E.g. embedded systems, sensor networks
- Can be viewed as combining CS and EE

Definition of Fields: SE

- Debatable engineering field
- Focus on software development
- Description, definition, and construction of software systems and applications, largely divorced from the physical platform on which they will run
- Difference from programming is in scale
  - You can build a doghouse from some wood planks yourself, but need an engineer for a skyscraper
- Based on mastery of CS
- Additional engineering aspects include requirements modeling, design, verification, QA, …

Basic Distinction

Science

- Create and accumulate knowledge
- Use experimentation to build theory
- Study the physical world (physics, chemistry, biology)
- Study abstract structures (math)
- Study computerized systems (CS)

Engineering

- Build reliable economic systems
- Do this efficiently
- Reconcile budget, requirements, regulations, and technology
- Diverges from underlying science
- May carry societal responsibility

Definition of Fields: CS

- A scientific field
- Principles and understanding of computer systems
- Theory
  - Models of computation, upper and lower bounds
- Practice
  - Architecture, operating systems, performance evaluation
- Programming considered required basic skill
- Churn of topics
  - Compilers / programming languages no longer major topics
  - Applicative areas like vision and learning become major

Definition of Fields: EE

- An undisputed engineering field
- Focus on hardware
  - Irrespective of how it will be used
- Level of devices
  - Transistors, physical effects, and manufacture
  - Transistors, logic, VLSI
- Level of construction and system architecture
  - ISA, ALU, pipelining, super-scalar and out-of-order, caching
- EE includes many other non-computer tracks (RF, control, signals, communication, …)

Study

- Largely correspond

CS	CS
SE	SE, CS
CE	EE, electronics,
EE	EE, electronics,

- Problem of rapid dy
  - Smartphones an ago
- Problem of differen
  - CS graduates em
  - SE is not really a

Basic Distinction

Science

- Create and accumulate knowledge
- Use experimentation to build theory
- Study the physical world (physics, chemistry, biology)
- Study abstract structures (math)
- Study computerized systems (CS)

Engineering

- Build reliable economic systems
- Do this efficiently
- Reconcile budget, requirements, regulations, and technology
- Diverges from underlying science
- May carry societal responsibility

Definition of Fields: CS

- A scientific field
- Principles and understanding of computer systems
- Theory
    - Models of computation, upper and lower bounds
- Practice
    - Architecture, operating systems, performance evaluation
- Programming considered required basic skill
- Churn of topics
    - Compilers / programming languages no longer major topics
    - Applicative areas like vision and learning become major

Definition of Fields: EE

- An undisputed engineering field
- Focus on hardware
  - Irrespective of how it will be used
- Level of devices, physical effects, and manufacture
  - Transistors, logic, VLSI
- Level of construction and system architecture
  - ISA, ALU, pipelining, super-scalar and out-of-order, caching
- EE includes many other non-computer tracks (RF, control, signals, communication, ...)

Definition of Fields: CE

- An engineering field
- Focus on the combination of hardware and software
- Tendency toward higher levels than EE
  - Computer systems, not devices
- Co-design of hardware platform and software system or application
  - Interact and complement each other
  - E.g. embedded systems, sensor networks
- Can be viewed as combining CS and EE

Definition of Fields: SE

- Debatable engineering field
- Focus on software development
- Description, definition, and construction of software systems and applications, largely divorced from the physical platform on which they will run
- Difference from programming is in scale
  - You can build a doghouse from some wood planks yourself, but need an engineer for a skyscraper
- Based on mastery of CS
- Additional engineering aspects include requirements modeling, design, verification, QA, …

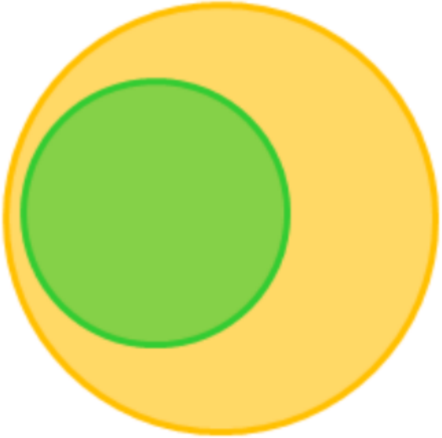
Study Programs

- Largely correspond to the fields

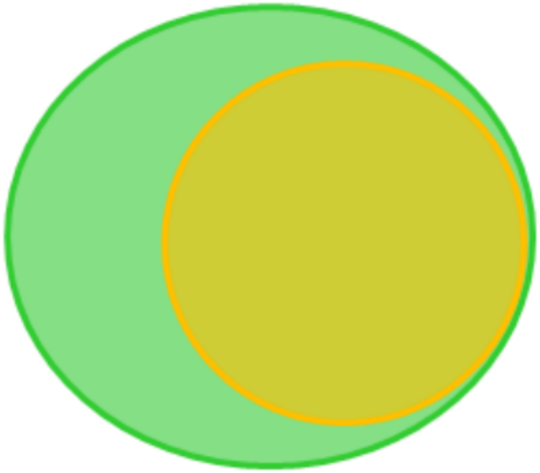
CS	CS	3 years
SE	SE, CS	4 years
CE	EE, electronics, signal proc’ing, CS	4 years
EE	EE, electronics, signal proc’ing	4 years

- Problem of rapid dynamic changes
  - Smartphones and cloud did not exist 10 years ago
- Problem of differentiating SE from CS
  - CS graduates employed as software engineers
  - SE is not really an engineering discipline

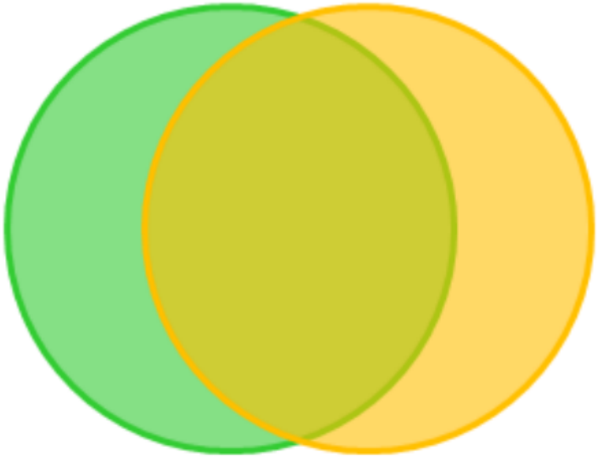
SE vs. CS



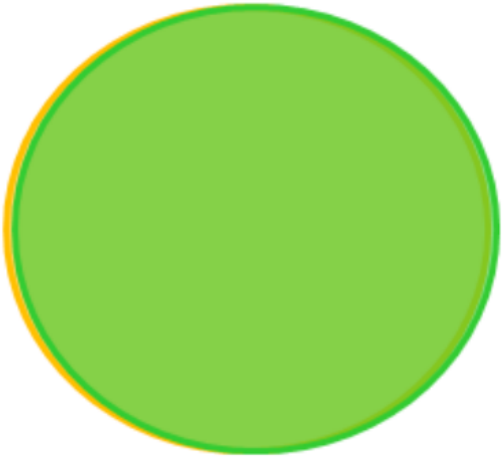
SE is a topic within CS



SE is a superset of CS



There is an intersection between SE and CS



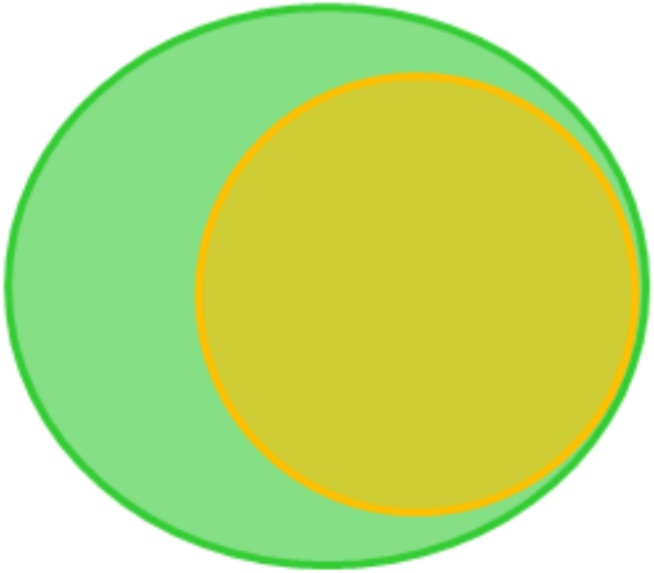
SE and CS are the same thing

SE is a su
opinion)

- SE rec
- There
mater
  - Pla
  co
- This is
anoth

SE and CS
opinion)

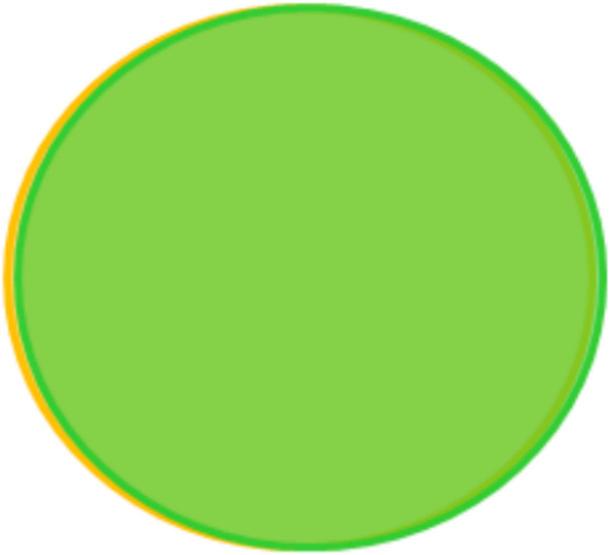
- CS gra
- Implic
suffici
- Make
incluc
- “SE” is



SE is a superset of CS

SE is a superset of CS (majority opinion)

- SE requires practically all of CS
- There is significant additional material
  - Planning, estimation, configuration, management, …
- This is non-trivial, justifying another year



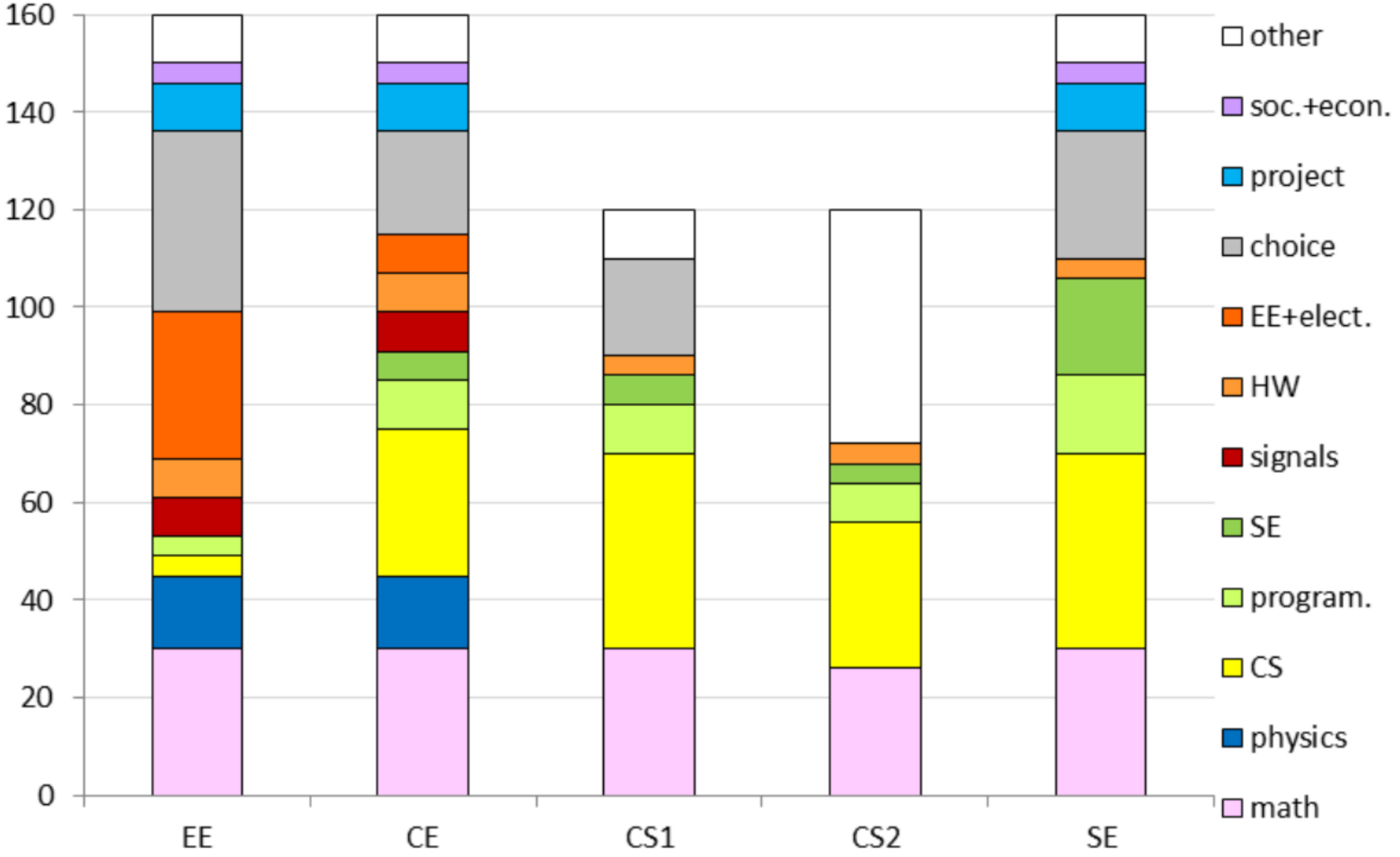
SE and CS are the same thing

SE and CS are the same (minority opinion)

- CS graduates work as SE
- Implication: 3-year program is sufficient for SE
- Make this explicit: CS should include some SE
- “SE” is just for marketing

Categories of Courses

- Math – discrete (CS, SE, CE) and continuous (EE, CE) including calculus and probability / statistics
- Physics – EE and CE, possible scientific subject for CS, SE
- CS – SE as in CS, CE as in double major, EE minimal
- Programming – basic OOP required in all, advanced (OOD, UI, event-based, parallel) in SE and partially in CS, CE
- SE – intro and security required in CS, CE, much more in SE
- Signals – CE and EE, information and DSP
- Hardware – minimal coverage required in all
- EE and electronics – mainly in EE, some in CE
- Professionalism and society – skills and theoretical basis



	math.	phys.	CS	prog.	SE	signal	HW	EE+ elect.	choice	proj.	soc.+ econ.	other
EE	30	15	4	4	0	8	8	30	37	10	4	10
CE	30	15	30	10	6	8	8	8	21	10	4	10
CS1	30	0	40	10	6	0	4	0	20	0	0	10
CS2	26	0	30	8	4	0	4	0	0	0	0	48
SE	30	0	40	16	20	0	4	0	26	10	4	10

Variations

- Option for flexibility
  - Changes of up to 25% in each area
  - But do not scrap small areas completely
- Differentiation and competition possible / desirable
  - Different institutions can select different emphases
  - E.g. CS with theoretical vs. practical slant
- Not everything can be expressed by allocating hours
  - Emphasis on professionalism and work habits in all courses
  - Emphasis on practical exercises and feedback on them

Summary
Main Points

- Basics:
  - Math in all programs
  - CS in all except EE
  - Physics only in EE, CE
- SE, CE include full CS (essentially 4 year double major)
  - Minority opinion: SE = CS
- Some advanced programming + SE / security in CE, CS
- Allocation of hours isn't everything
- Consider adding non-technical topics
  - Social sciences in SE, CS
  - Soft skills in all engineering disciplines
- Rethink CS in academic colleges and SE in engineering