

Reflection on Project-Oriented Practice in Software Engineering Curriculum

Arnon Sturm, Gera Weiss

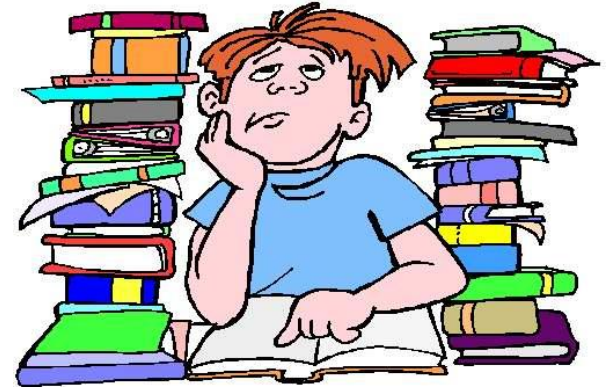


Projects are not the only self-learning tool...

2



Reading



Home Work



Laboratory



Project

Projects in SE curriculum?

3

Projects in SE@BGU:

- A sequence of ongoing programming or modelling tasks
- Students are given testable specifications + basic test-cases
- Grades are given based on automatic and manual inspection
- Usually, all teams get the same task

Programming Tasks

Project+Teaching

Project

Laboratory

מערכות הפעלה	עקרונות הקומפילציה
סדנא ליישום פרויקט תוכנה	יסודות הנדסת תוכנה
עיצוב מנשקי אדם מחשב	מבוא לרשתות מחשבים
אבטחת מחשבים ורשתות תקשורת	מבוא לשיטות חישוביות
סטטיסטיקה	פיסיקה 1 ב'

חדו"א א' 2'	חדו"א א' 1'
מבנים בדידים וקומבינטוריקה	מבוא ללוגיקה ותורת הקבוצות
עקרונות תכנות מונחה עצמים	אלגברה ליניארית
מבני נתונים	מבוא למדעי המחשב
אנגלית מתקדמים 2	הדרכה בספריה
מבוא להנדסת תוכנה	אנגלית מתקדמים 1

פרויקט בהנדסת תוכנה 2	פרויקט בהנדסת תוכנה 1
לימודים כלליים	אימות תוכנה
קורס בחירה מערכות מידע	קורס בחירה מערכות מידע
קורס בחירה מערכות מידע	קורס בחירה מערכות מידע
קורס/ בחירה מדעי המחשב	קורס/ בחירה מדעי המחשב
	הנדסת איכות תוכנה

תכנון אלגוריתמים	בסיסי נתונים
עקרונות שפות תכנות	אוטומטים
מעבדה לארכיטקטורה ותכנות מערכות	תכנות מערכות
ניתוח ועיצוב מערכות להנדסת תוכנה	הסתברות
	מערכות ספרתיות

Study Goals and Objectives

4



- The goal of this study is to
 - ▣ Uncover the perceived value of projects
- Research questions:
 - ▣ Do project “pay back” the invested work-load?
 - ▣ Do projects shift focus towards technicalities?
 - ▣ Do projects help in improving “soft skills”?
 - ▣ Do we need to add/cancel some projects?

Study Plan

5

□ Questionnaires to students

- 2nd to 4th year SE students
- Two parts:
 1. General perception
 2. Perception for each course
- Participation was voluntary

□ Interview with instructors

- Why do you use a project in your course?
- Did the project achieve its goals?



54 responses out of 247 requests

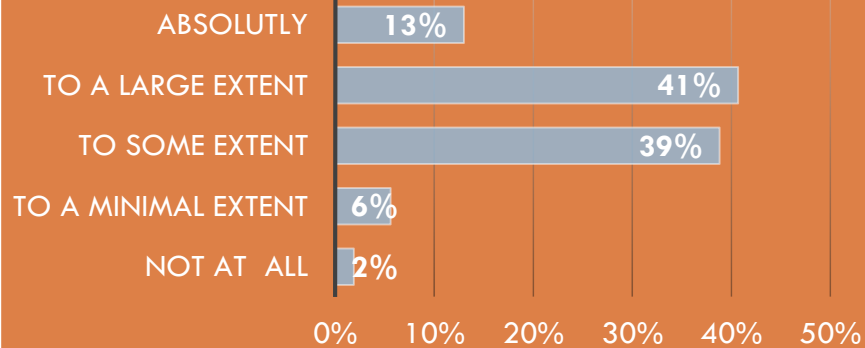


About an hour with each instructor

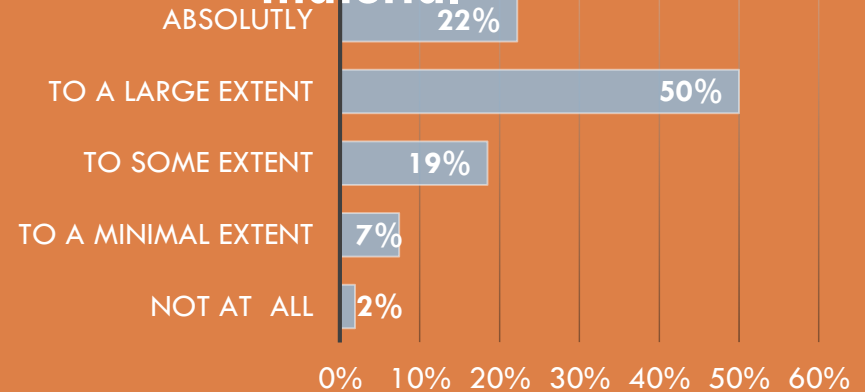
Results

6

Overall, are projects good?



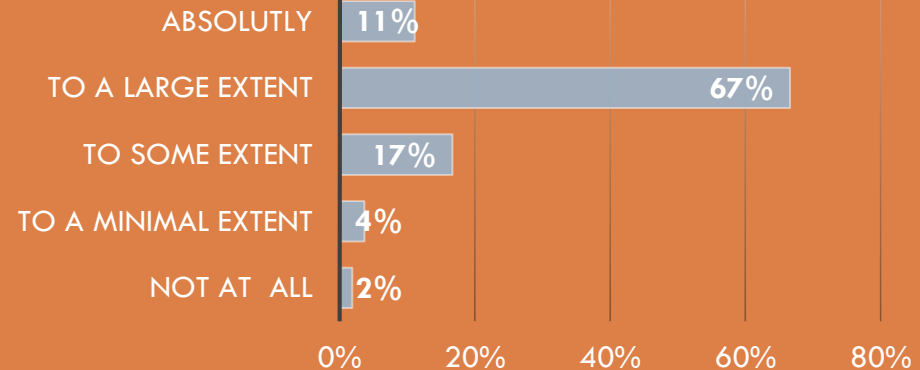
Project helped in understanding course material



Students like projects



Good Learning Experience



Results



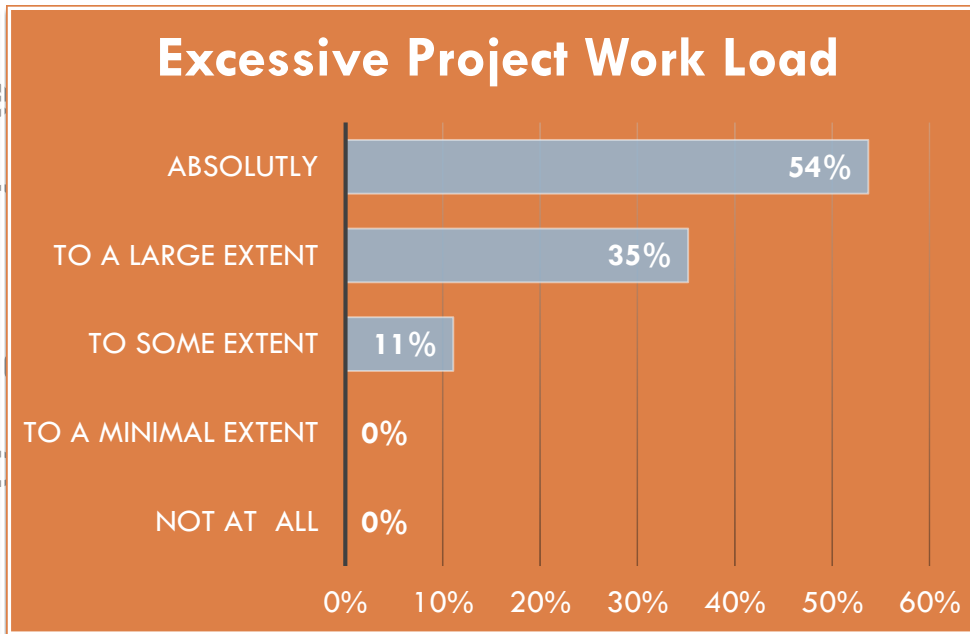
7

□ **Students** think that projects put excessive load

□ They pe
program

□ They pr
2-3 stud

□ They think that projects do not contribute to

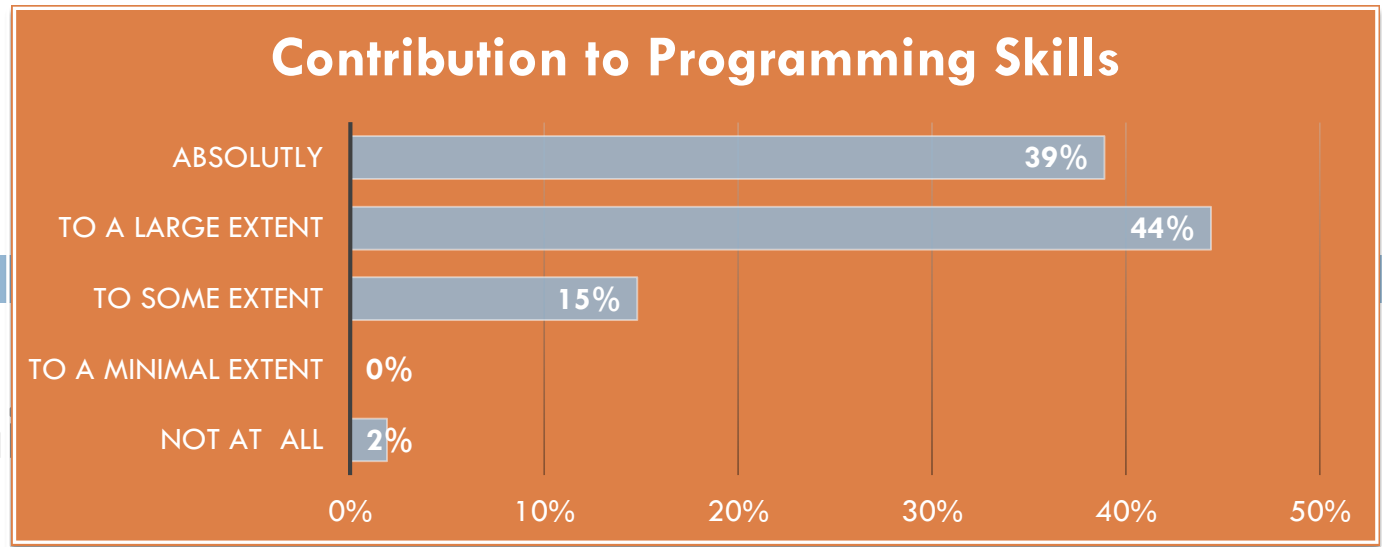


g to their
management skills
ams size of

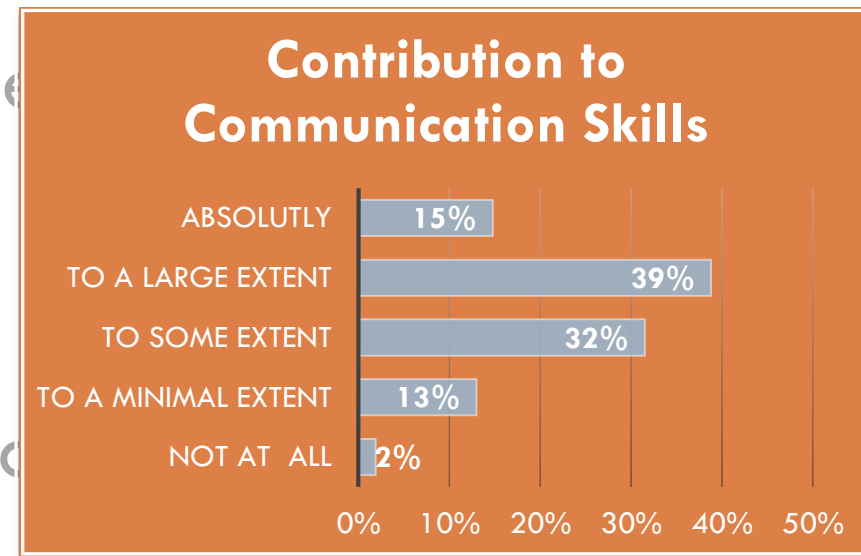
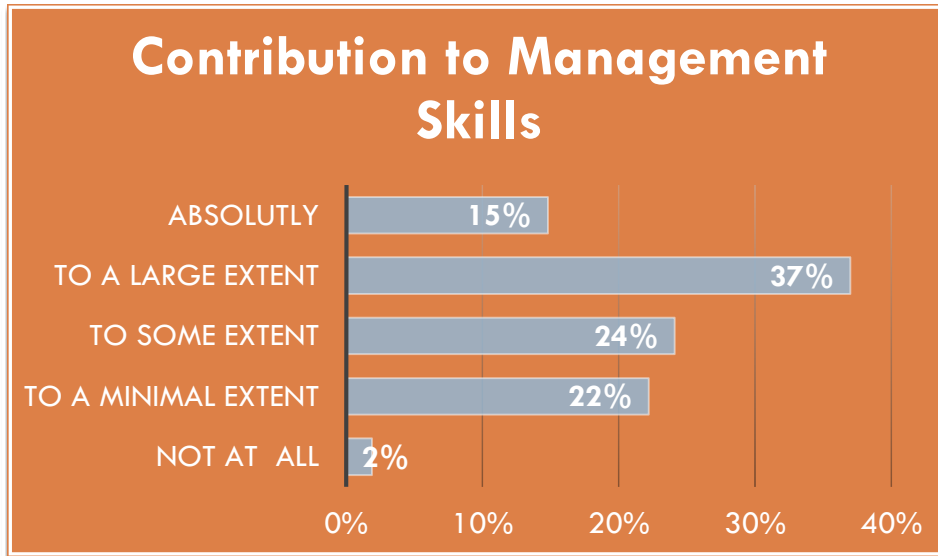
Instructors think that the load is within reasonable limits

Results

8



- Students perceive projects as contributing to their programming, communication, and management skills



Results

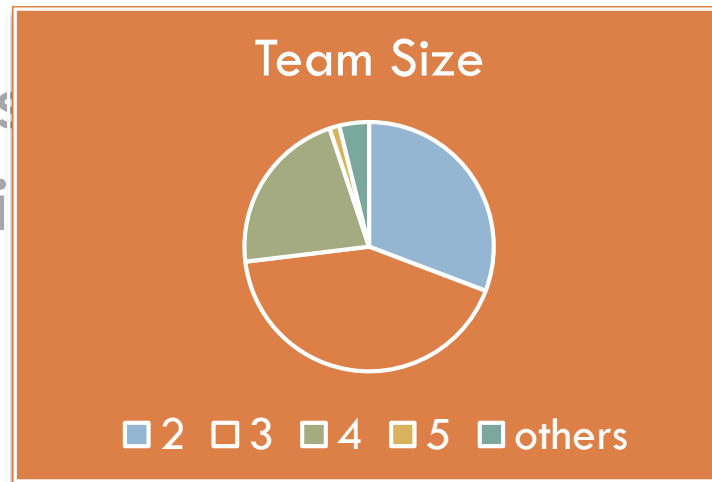
9

- Students' comments: excessive load
- They prefer projects with 2-3 students per team
- They think that projects help in understanding theoretical concepts, programming, communication, and management skills

Students' Comments:

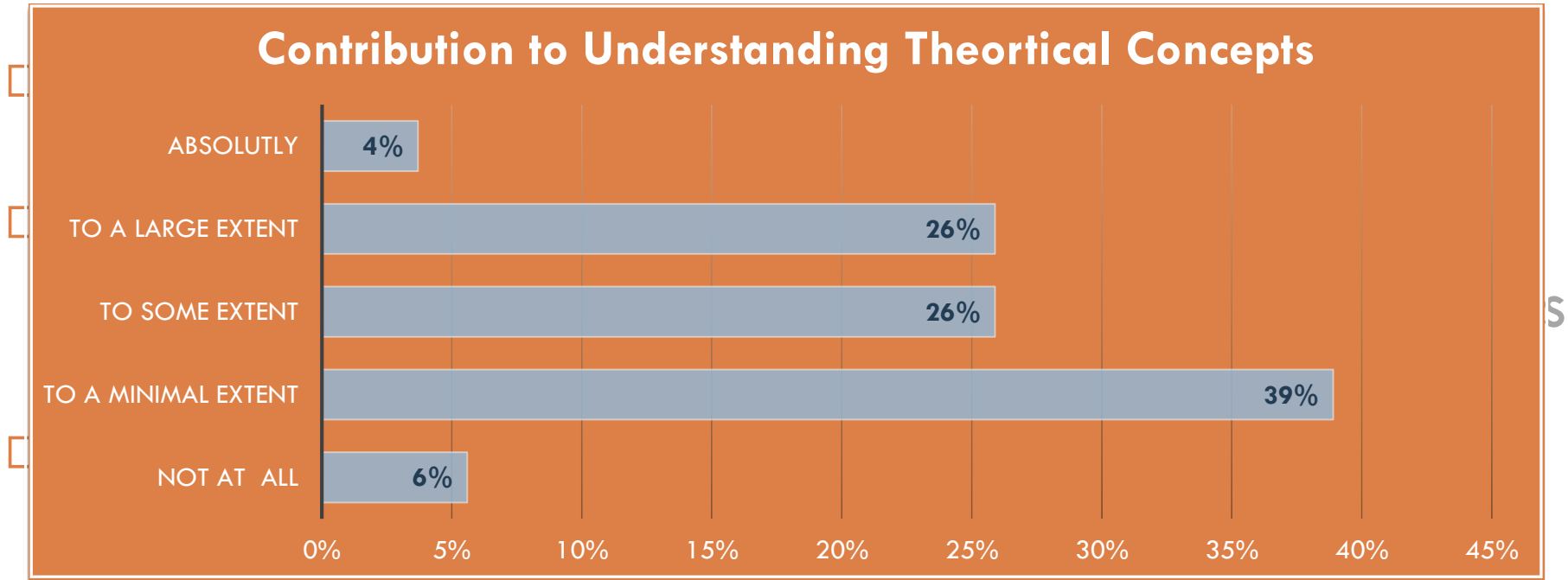
2-3 is an ideal number

- Team work
- Reasonable to manage
- Minimizes free riders



Results

10



- They think that projects do not contribute to understanding theoretical concepts

Results

11

- Students think that projects put **excessive load**
- They perceive projects as **contributing** to their programming, communication, and management skills
- They prefer **team of 2-3 students**
- They think that projects **do not contribute** to understanding theoretical concepts

Student comments

12

Advantages:

- Practical training (hands-on)
- Increased self learning abilities
- Iterative learning is good

Limitations:

- Not enough guidance

So, now let's drill down

13

- We looked at 5 courses:
 - ▣ Introduction to Software Engineering
 - ▣ Analysis and Design of Software Systems
 - ▣ Elements of Computing Systems
 - ▣ Compiler Principles
 - ▣ Formal Verification

Results of our per-course survey

14

- It takes time (years) to establish a good project task
- For courses that teach **software engineering processes** (Intro. to SE, Analysis and Design):
 - ▣ Projects help students see rationale and motivations
 - ▣ Risk of getting too much into technologies
 - ▣ Projects and theory are not always aligned
- For courses that teach **tool internals** (Compilers, Model Checkers, Hardware):
 - ▣ Student appreciate the contribution of developing a prototypical tool
 - ▣ Students enjoy the project despite high load
 - ▣ Projects put a good emphasize on theoretical issues

Results of our per-course survey

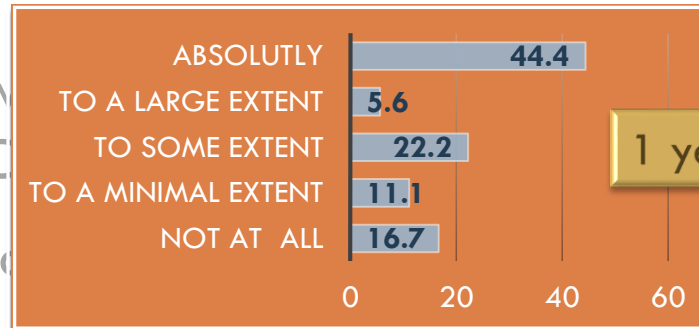
15

□ It takes time (years) to establish a good project task

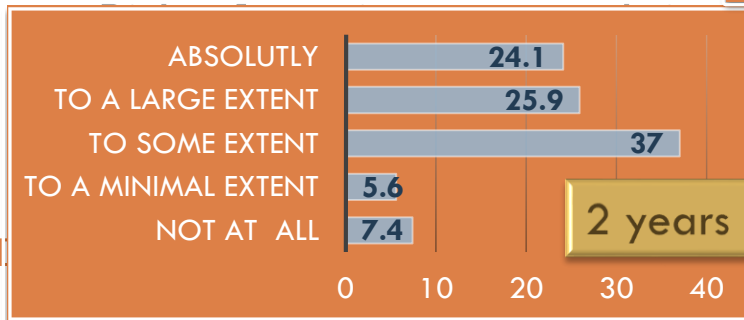
□ For courses that teach software

Can the project be improved?

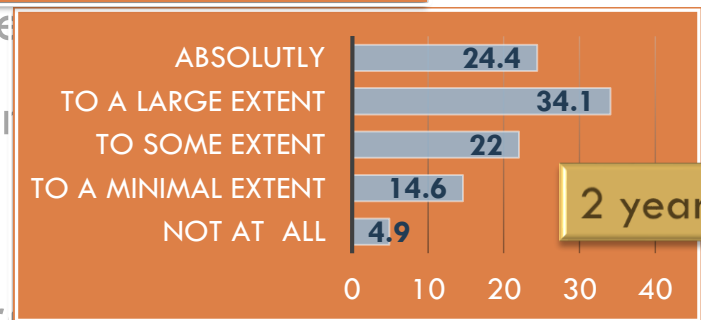
□ Projects help students see re



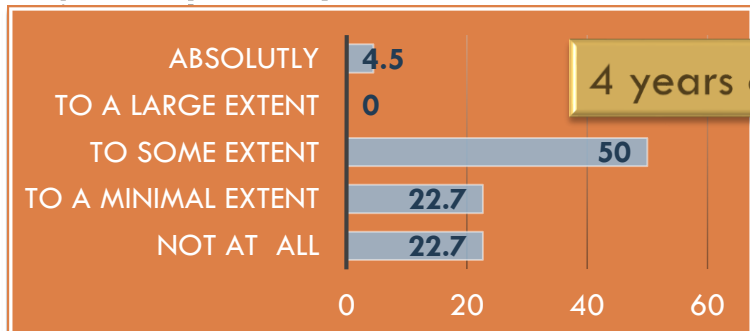
1 year old



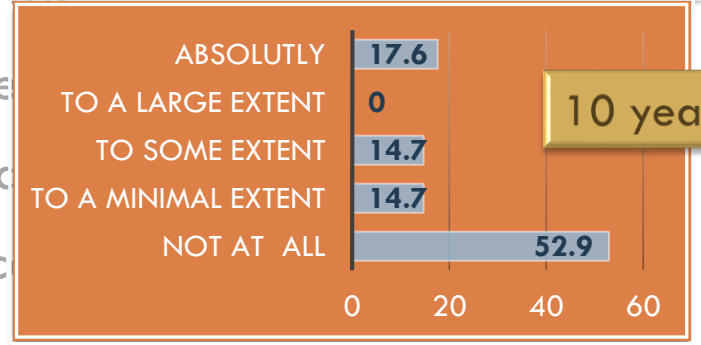
2 years old



2 years old



4 years old



10 years old

Results of our per-course survey

16

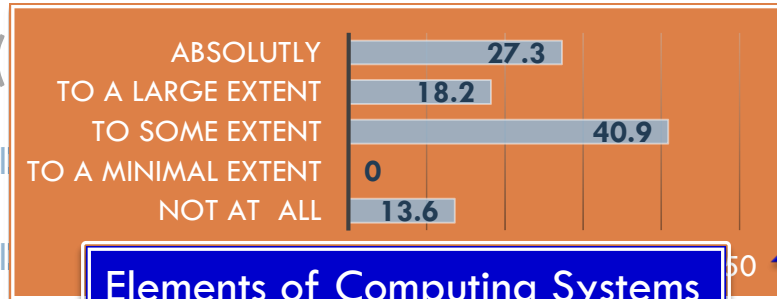
□ For courses that teach **software engineering processes**

- ▣ Risk of getting too much into technologies
- ▣ Projects and theory are not always aligned

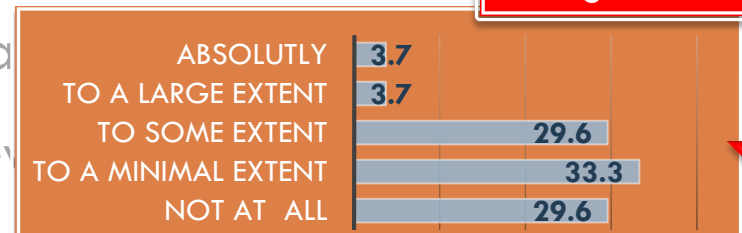


Design & Analysis

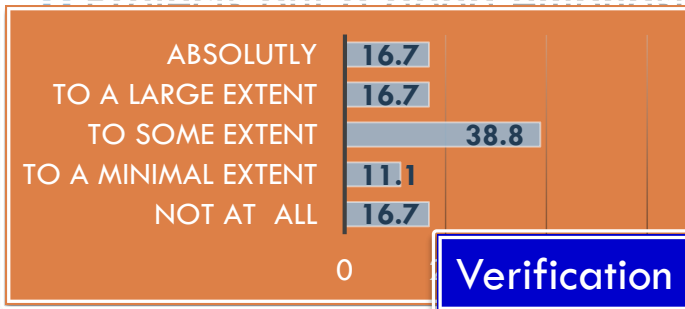
□ For courses that teach **tool internals**



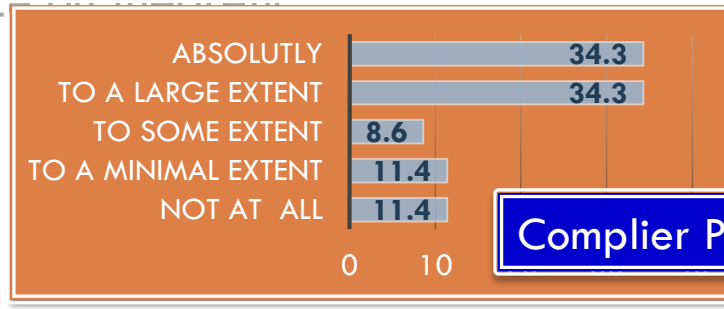
Elements of Computing Systems



Introduction to Software Engineering



Verification



Compiler Principles

Results of our per-course survey

17

- For courses that teach **software engineering processes**
 - ▣ Risk of getting too much into technologies
 - ▣ Projects and theory are not always aligned

- For courses that teach **tool internals**
(Compilers, Model Checkers, Hardware):
 - ▣ Student appreciate the contribution of developing a prototypical tool
 - ▣ Projects put a good emphasize on theoretical issues



Threats to Validity

18

- Question Bias
 - ▣ Does questioning about project attracts criticism?

- Limited number of participants
 - ▣ Is 54 responses out of 247 requests enough?

- No comparison to same courses without a project

Conclusions

19

- Projects are a good practice in SE courses
 - ▣ They provide a programming experience
 - ▣ They allow for better understanding of the material
 - ▣ Contribute to management & communication skills
 - ▣ **Students and instructors are in debate** regarding the load the projects pose on the students
- **The results indicate a risk** of having projects shift student attention from theory to technicalities
- We will verify this trend in future research

Questions???

20

Arnon Sturm, Information Systems Engineering, Ben-Gurion University of the Negev, Israel, sturm@bgu.ac.il



Gera Weiss, Computer Science, Ben-Gurion University of the Negev, Israel, geraw@cs.bgu.ac.il



Introduction to Software Engineering

21

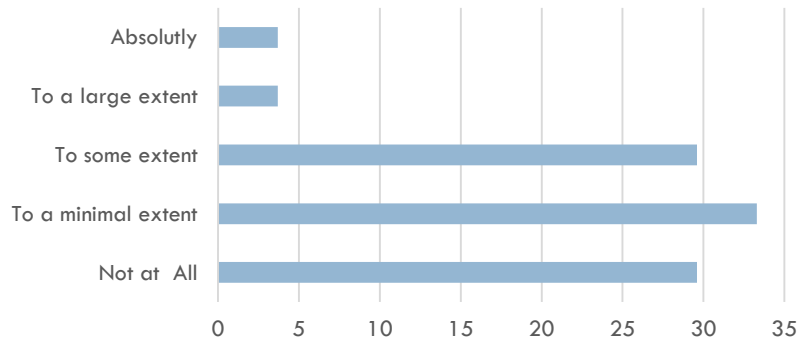
- Course goal: to introduce basic SE concepts
- Covers topics such as:
 - ▣ **Theory:** SDLC models, Software design,...
 - ▣ **Tools:** Version Control, Unit Testing, ...
 - ▣ **Practice:** Database, Web, ...
- The first course in SE. Given in the second semester
- This year was the **second round of the course**



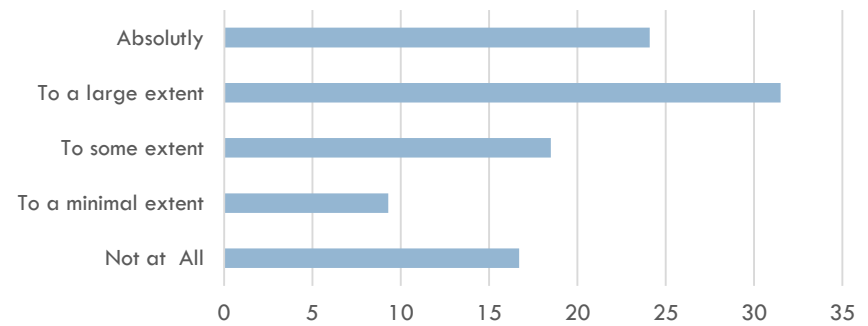
Introduction to Software Engineering

22

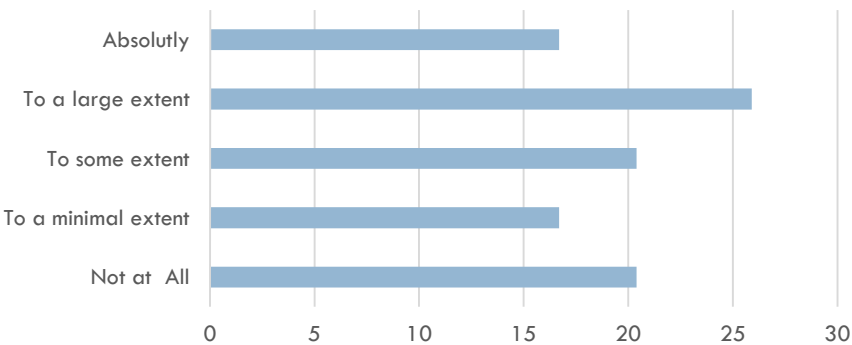
Contribution to Understanding



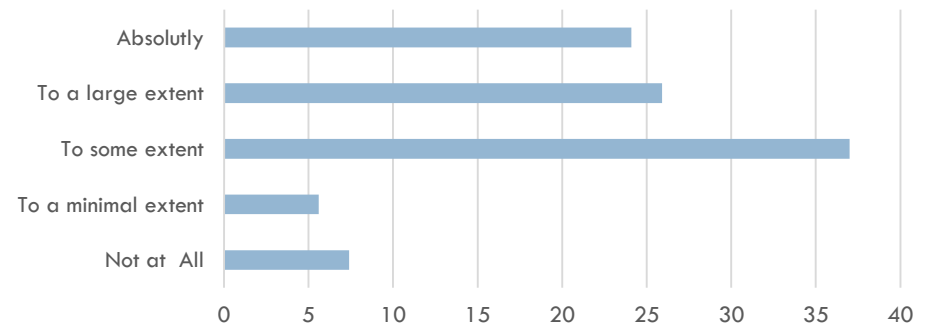
Contribution to Programming Skills



Could the course be taught without a project



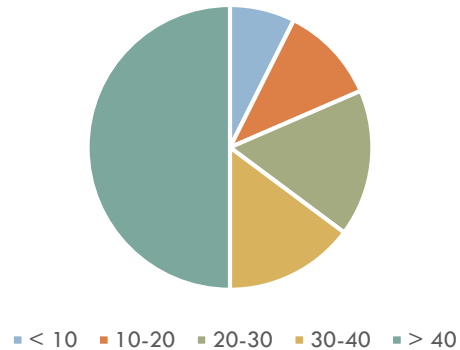
Could the instructor choose a better Project?



Introduction to Software Engineering

23

Effort Invested per Student



Students' Comments:

Advantages

- Teaches self-learning
- Provided a lot of insights
- Fun

Limitations:

- Self-learning/not enough guidance
- Not connected to the lectures
- Too loaded

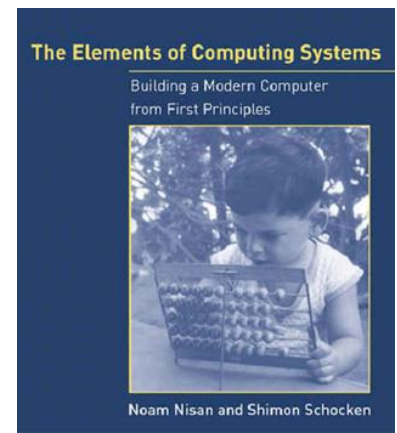
Instructor's Comments:

- Hard to teach abstract ideas at this stage
- Programming facilitate better understanding, not achievable by homework
- The course is loaded
- Goals achieved: design, multi versioning, and teamwork
- The students were proud of their achievements

Elements of Computing Systems

24

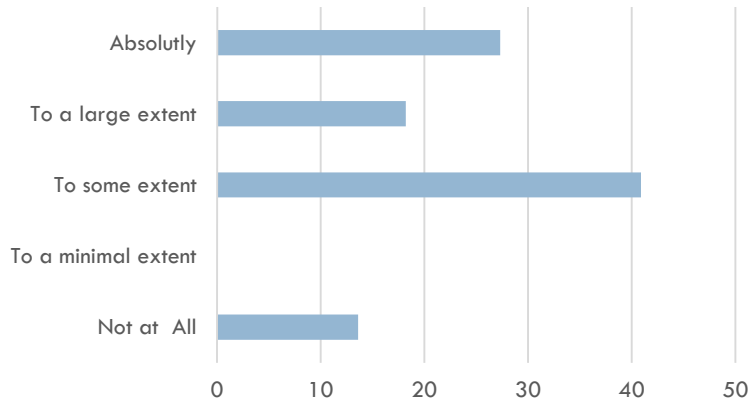
- Based on Nisan's & Shockey's book
- Building a computer from logic gates up:
 - ▣ starting with the hardware (combinational logic gates, arithmetic logic units, sequential logic gates, the CPU and memory) and then through the software hierarchy
- The course is given in the 3rd semester
- Given for the 4th time



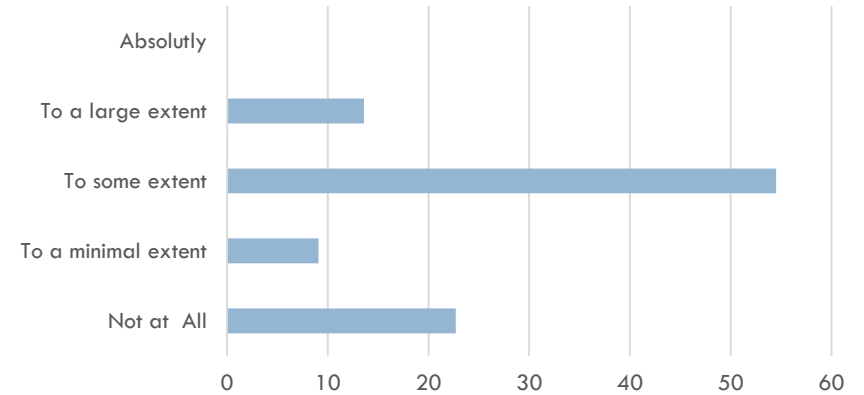
Elements of Computing Systems

25

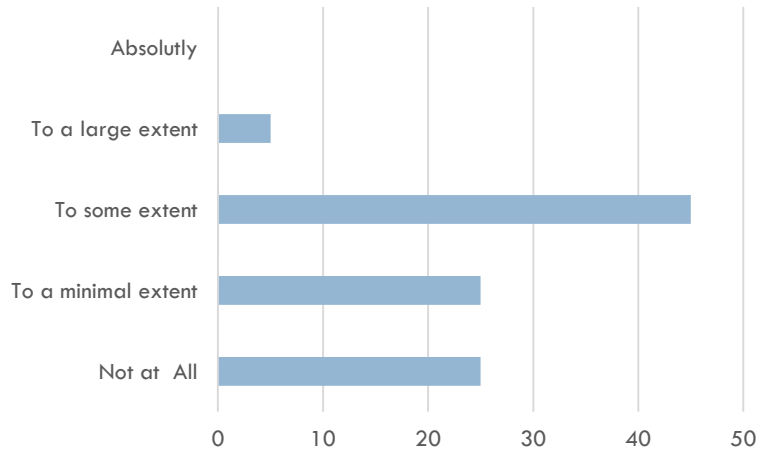
Contribution to Understanding



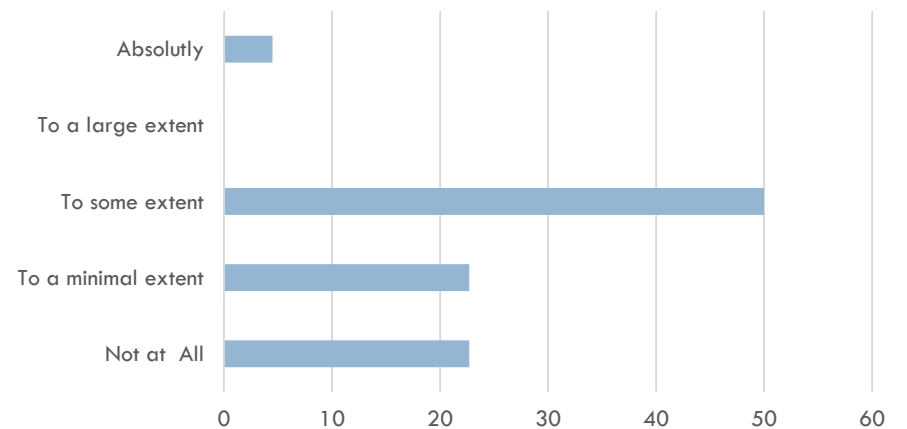
Contribution to Programming Skills



Teaching without Project



Better Project



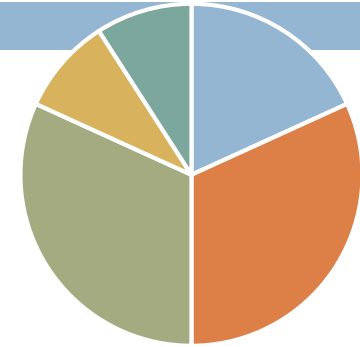
Elements of Computing Systems

Effort Invested per Student

26

Students' Comments:

- ▣ Effective learning
- ▣ Aligned with the material



▣ < 10 ▣ 10-20 ▣ 20-30 ▣ 30-40 ▣ > 40

Instructor's Comments:

- ▣ Goals: Allow for deeper understanding of the material and for programming practice
- ▣ Provided infrastructure to allow focus on the required material
- ▣ Students understanding was good (based on exam)
- ▣ No need for teamwork, there is no complex design
- ▣ The course teaches technical capabilities
- ▣ Students enjoyed the course

Analysis and Design of Software Systems

27

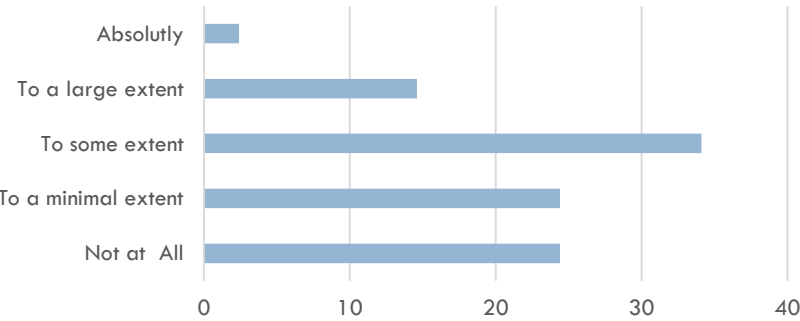
- **Course goal:** teach analysis and design techniques in software development
- **Project goal:** to demonstrate the relationship among code and design, to practice the implication of changes, and to cope with project and team management
- The course is given in the 4th semester.
- It was given for the third time in that configuration



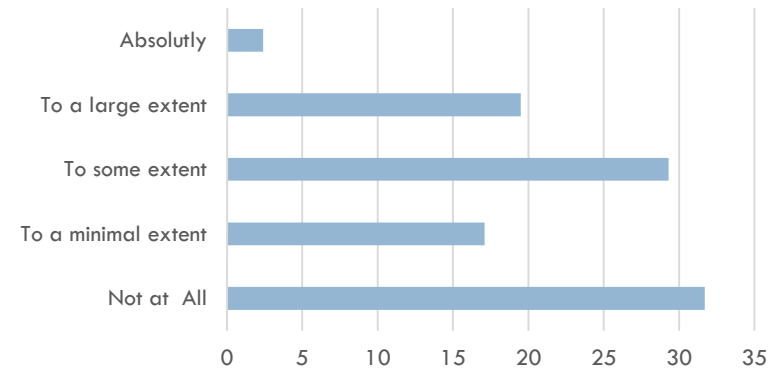
Analysis and Design of Software Systems

28

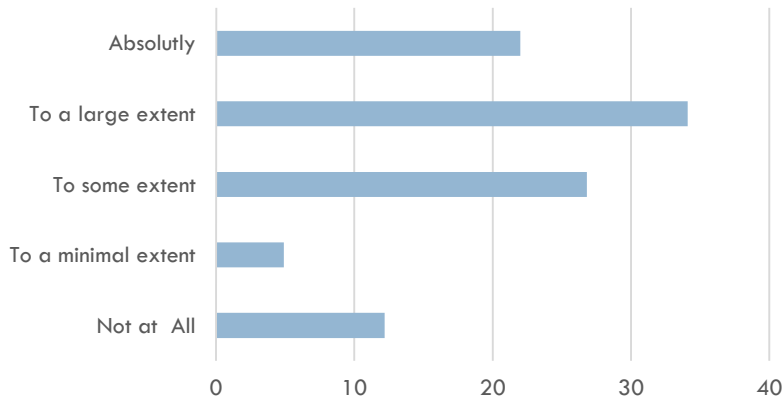
Contribution to Understanding



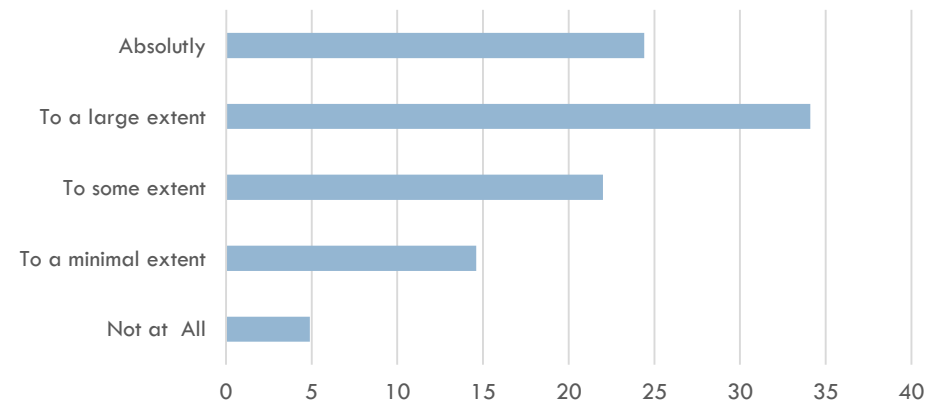
Contribution to Programming Skills



Teaching without Project



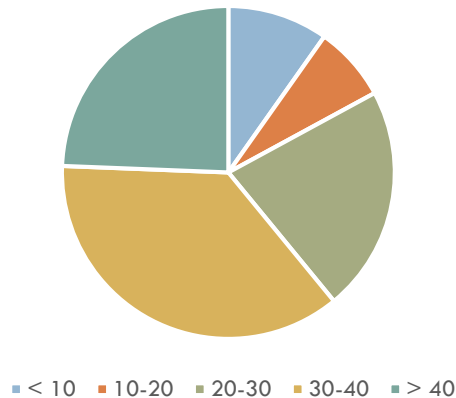
Better Project



Analysis and Design of Software Systems

29

Effort Invested per Student



Students' Comments:

- Limitations
 - Not relevant for that course
 - Not related to the course material
 - Free riders
 - Did not help in understanding the material
 - Loaded

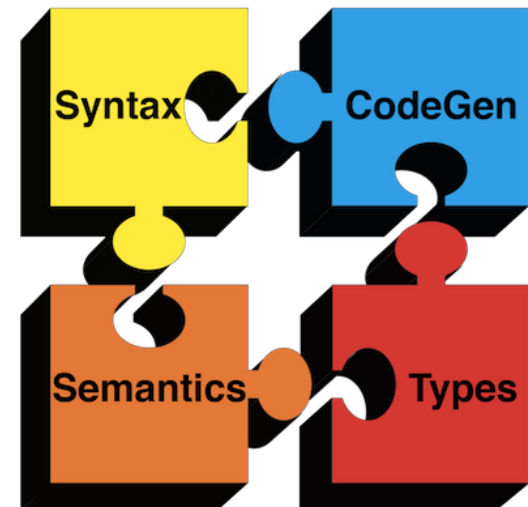
Instructor's Comments:

- The project load is not above required standards
- The project achieved part of its stated goals – in some cases the analysis in early stages reveal problems in the project. The connection among code and design was not demonstrated well.
- Can do the course without a project...
- The size of the ten should vary (increased) as the project evolve

Compiler Principles

30

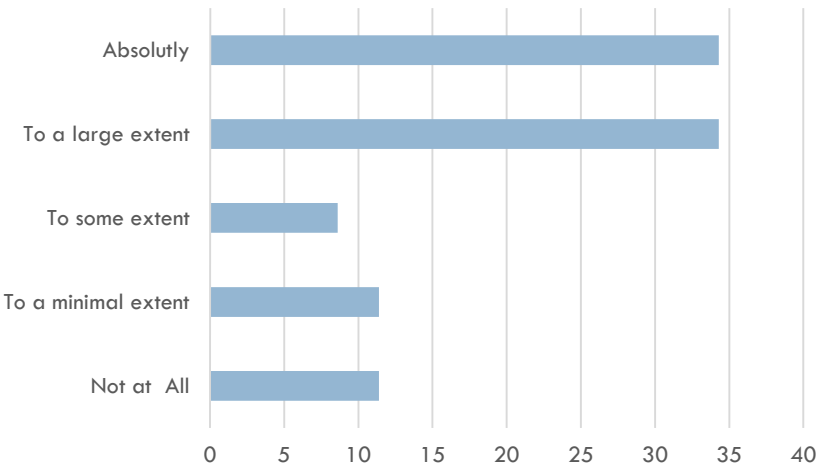
- The goals of the course is the following:
 - ▣ Have a deeper understanding of programming languages including their characteristics and limitations
 - ▣ Understanding the compilation process
- Given in the fifth semester
- Has an long history
- Led by a very dedicated person



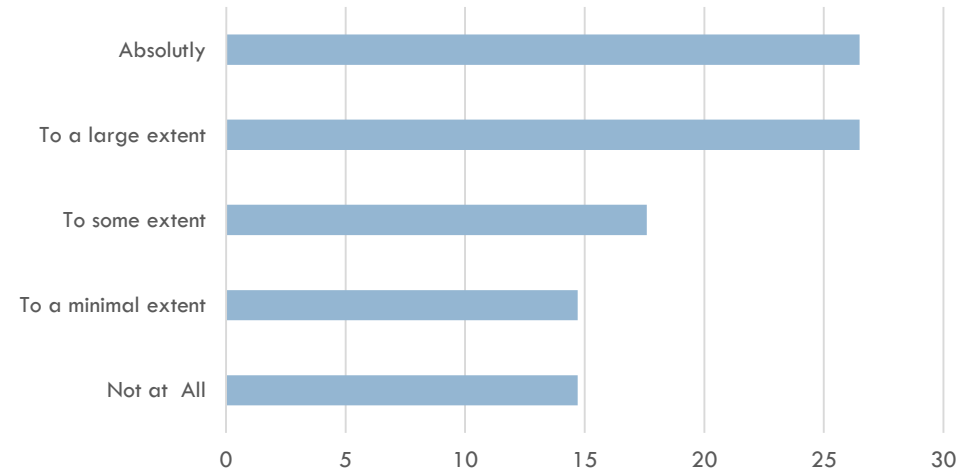
Complier Principles

31

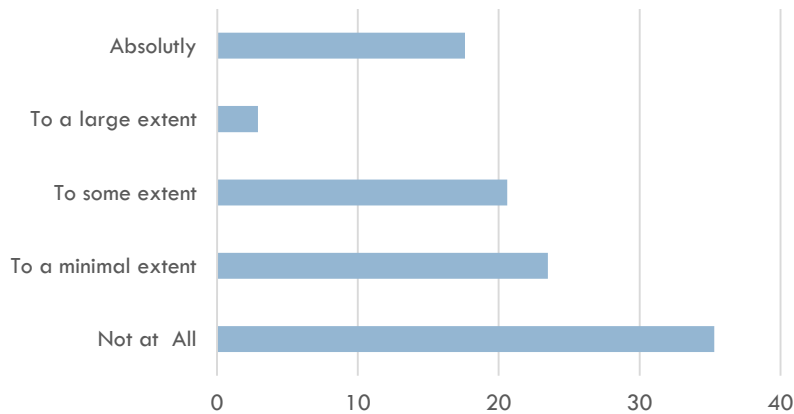
Contribution to Understanding



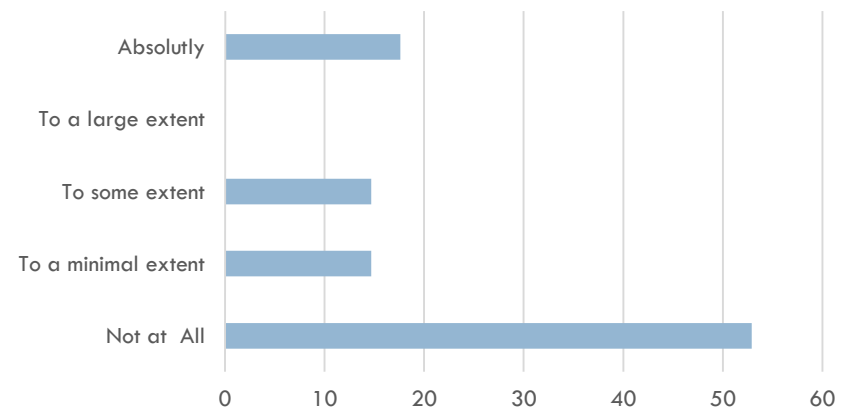
Contribution to Programming Skills



Teaching without Project



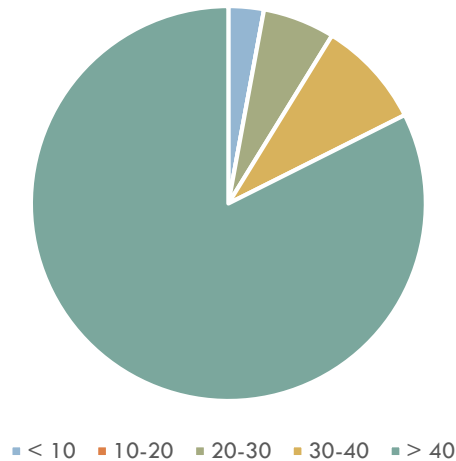
Better Project



Complier Principles

32

Effort Invested per Student



Students' Comments:

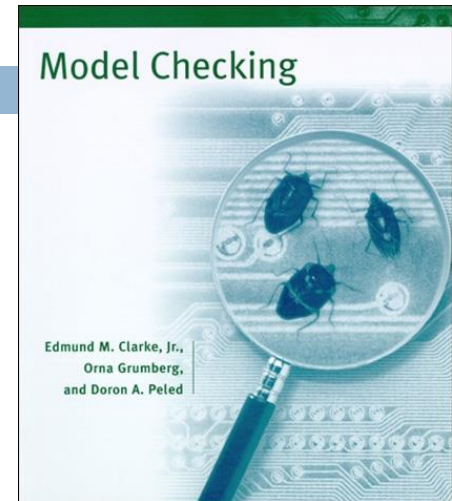
- Advantages
 - Well guided
 - Well fitted to the course theme
 - Help in understanding the course
 - Interesting
 - Challenging
- Limitations
 - Loaded
 - Not relevant

Instructor's Comments:

- The load of the project is reasonable; it is important to have a large and meaningful project to allow effective learning
- The project is based on an infrastructure provided to the students.
- The students understanding was good
- The ideal team size is too – otherwise to many management problems
- Students enjoyed the course and are proud of the outcome

Formal Verification

33

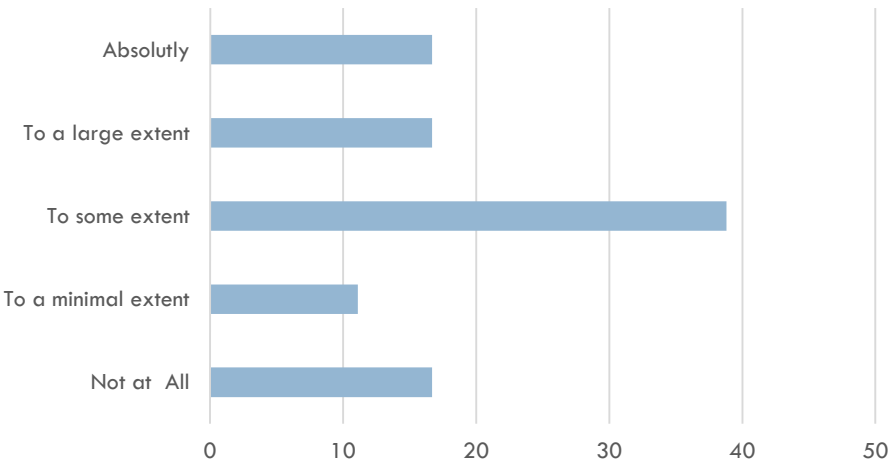


- An introduction to formal verification theory and practical methods
- The course contains both:
 - ▣ **Mathematical content:** Temporal Logic, Automata Theory, ...
 - ▣ **Practical content:** Algorithms, Modelling Languages, Methodologies,...
- The project consisted of programming a model-checker
- The course is given with a project for the 1st time

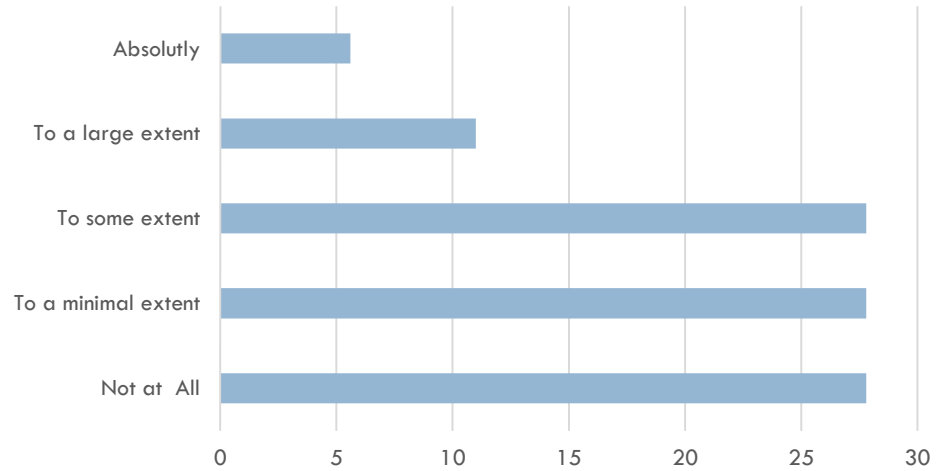
Formal Verification Methods

34

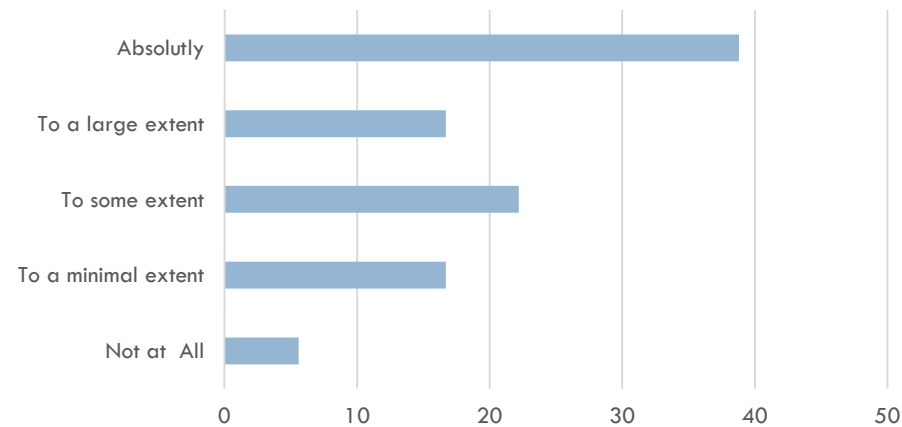
Contribution to Understanding



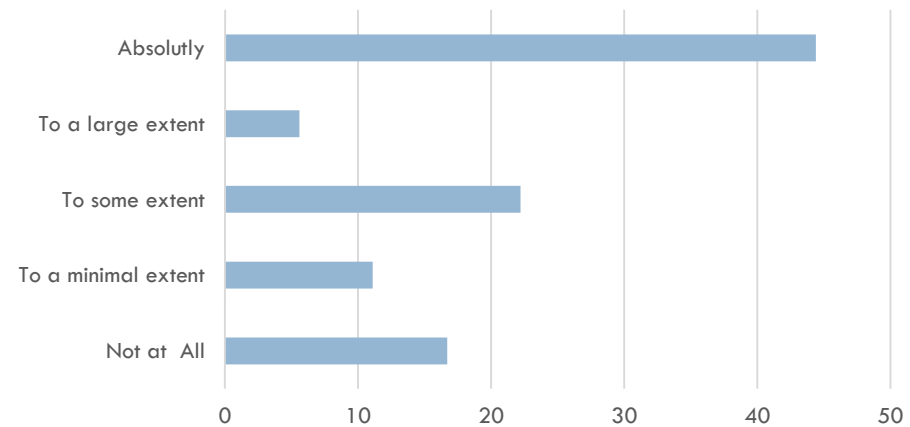
Contribution to Programming Skills



Teaching without Project



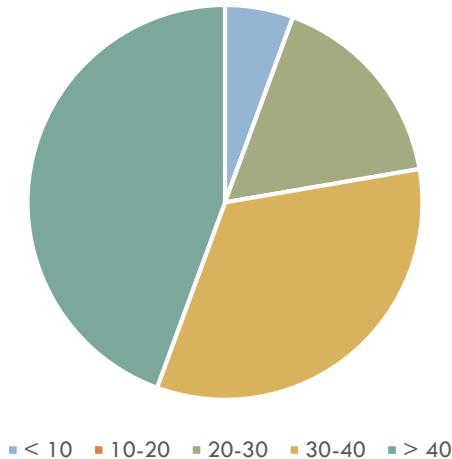
Better Project



Formal Verification Methods

35

Effort Invested per Student



Students' Comments:

- Advantages
 - Has a potential to contribute to understanding
 - Good in having a verification tool
- Limitations
 - Loaded
 - In its incubation phase
 - Did not help to understand the material

Instructor Viewpoint

Purpose (of weaving the project):

- Increase enjoyment
- Increase accessibility of material
- Implementation increases understandability

Scope & Load: OK

Achievement of goals: Partially

Student Enjoyment: Yes

Optimal Team Size: 2-3 to allow diversity

Type of Project: Standard to allow collaboration