Software Engineering (COMP433)

Introduction

Section: 1
Location: N/A;
Time: Tuesday & Thursday::10:00-11:15

Prof.Dr. Adel Taweel
ataweel@birzeit.edu

web-page:
http://

Why Software Engineering?

- **Software development is Complex**!
- **Important to distinguish "small" systems** (*one developer, one user, experimental use only*) **from "Complex" systems** (*multiple developers, multiple users, products*)
- **Experience with "small" systems is misleading**
  - *One person techniques do not scale up*
- **Analogy with bridge building:**
  - A bridge over a stream = easy, one person job
  - A bridge over a River … ? (*the techniques do not scale)*

Why Software Engineering ?

The problem is ***complexity***

Complexity depends on many factors, but ***size*** is a key factor:

UNIX:

v 1 (1971) contains 10,000 lines of code

v 10 (1989) contains 4 million lines of code

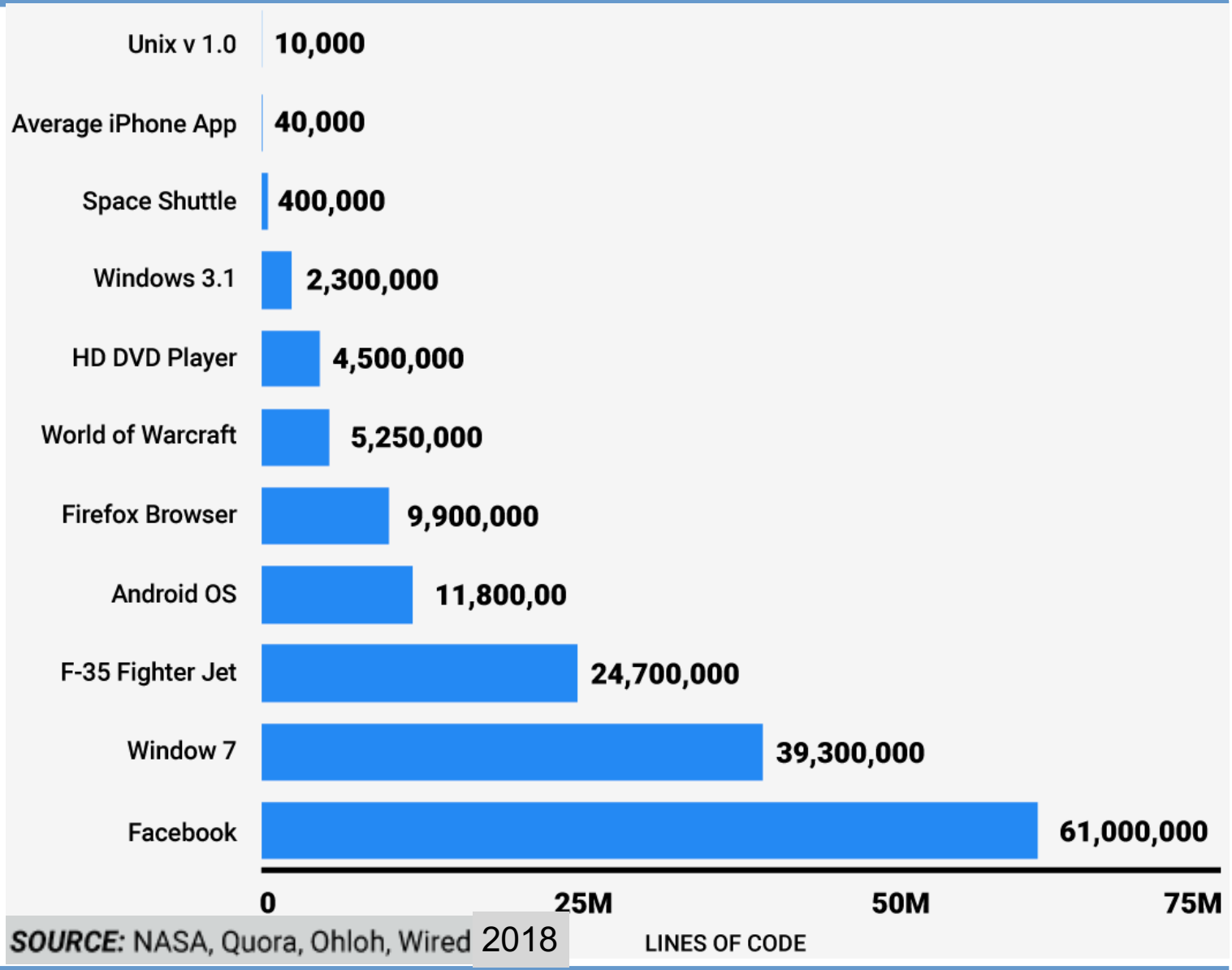
Windows:

Windows 2000 contains 100 million lines of code

Windows 7 contains 39.3 million lines of code (?)

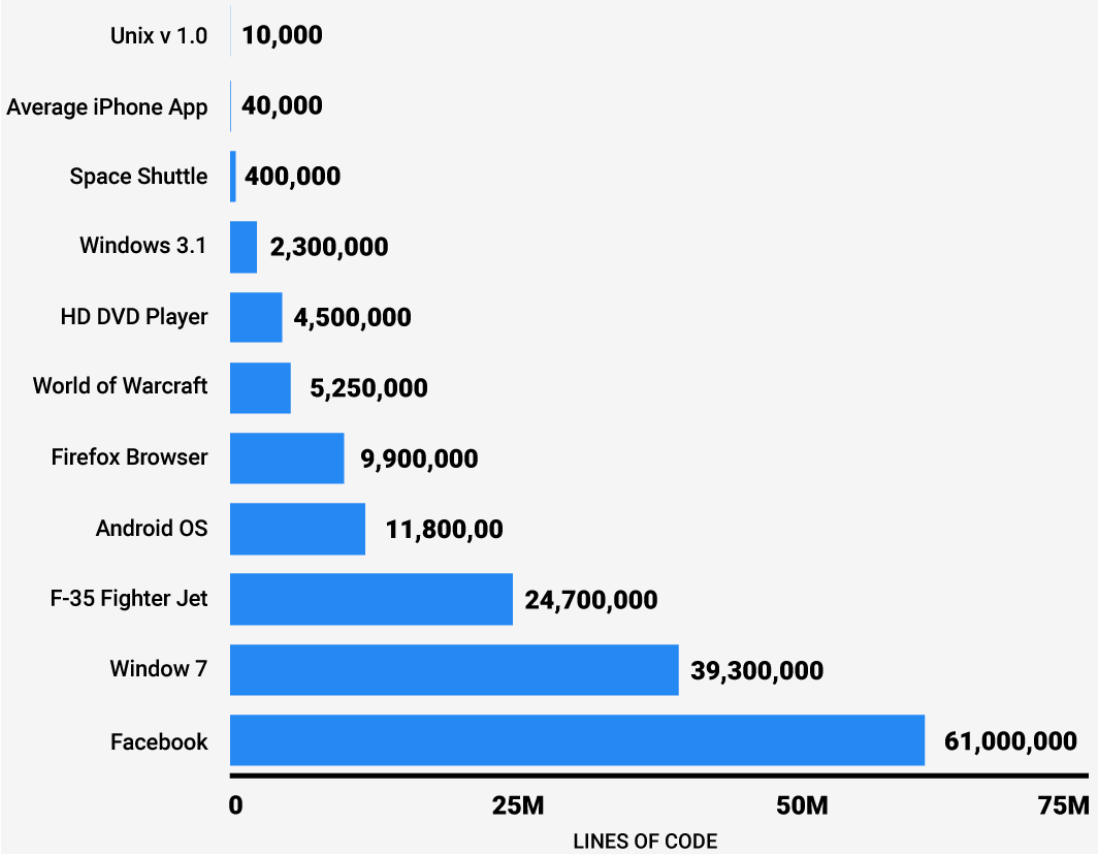
Why Software Engineering ?

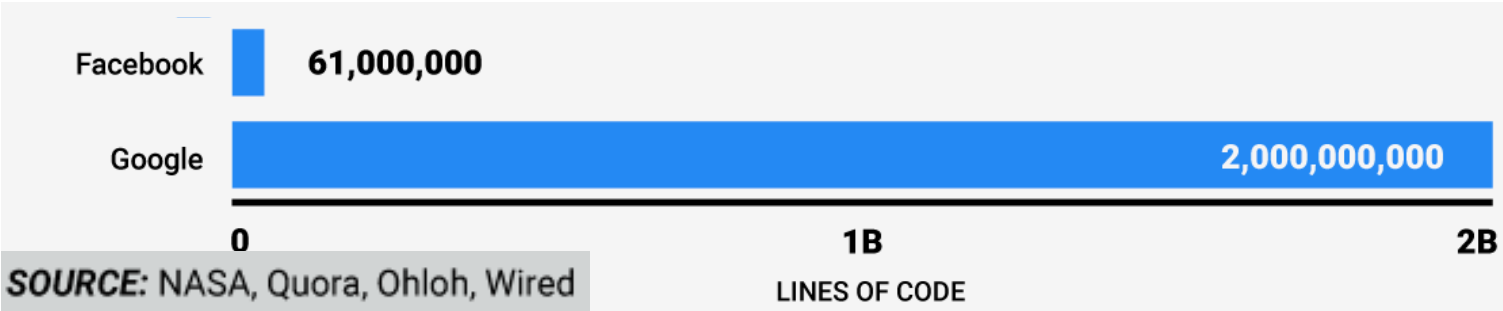
Complexity
and
Size
matter!



Why Software Engineering ?

Complexity ***increases*** *as* ***Size*** **increases!**



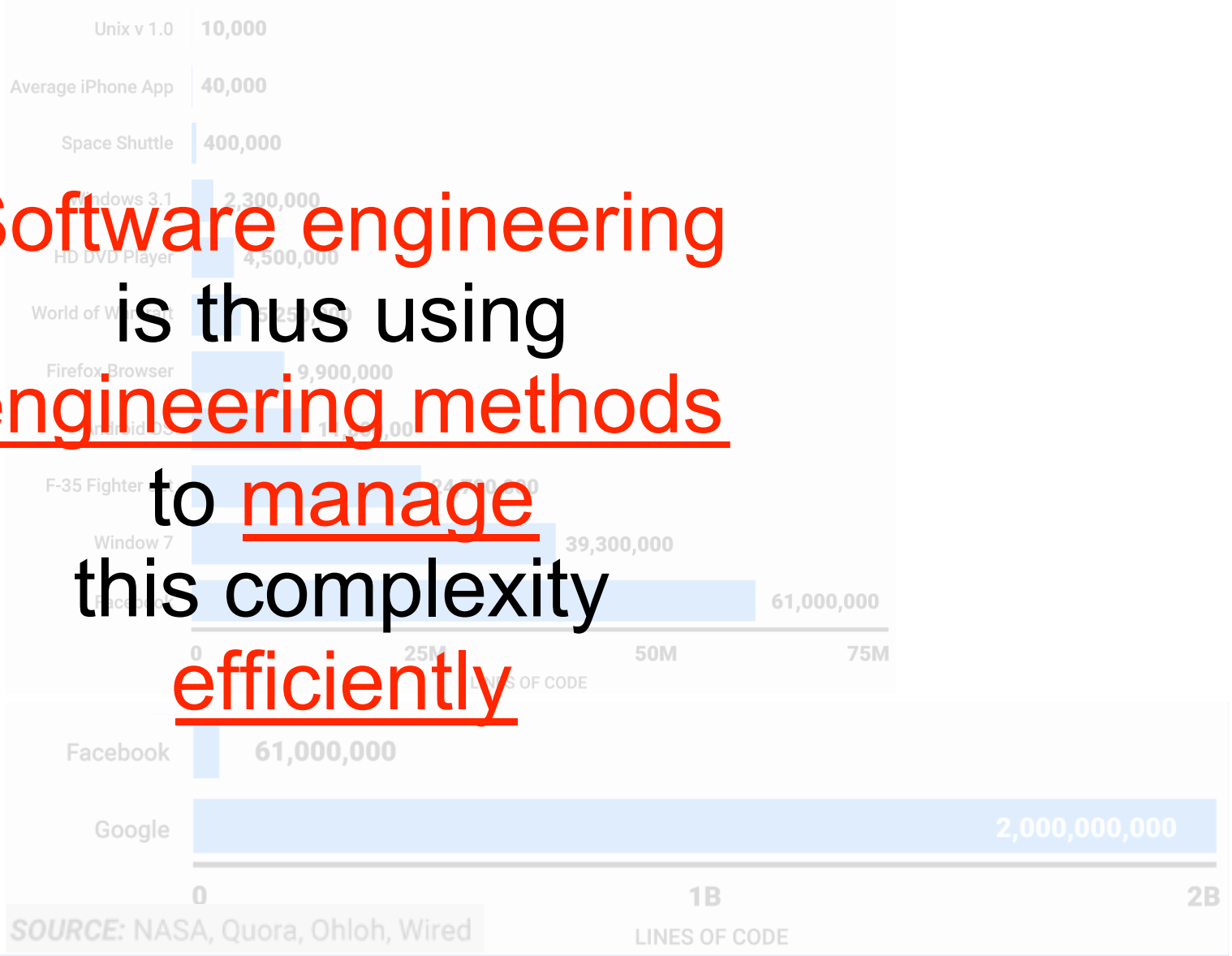


SOURCE: NASA, Quora, Ohloh, Wired

Why Software Engineering ?

Complexity increases as Size increases!

Software engineering is thus using engineering methods to manage this complexity efficiently



Teaching method

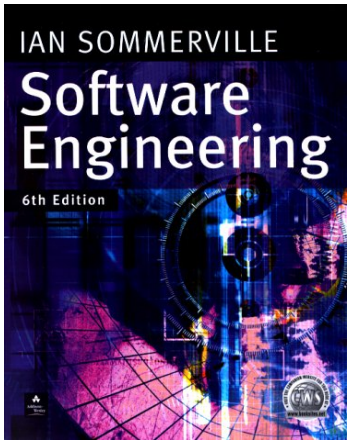
- Lectures **(~ 3hrs per week)**
- Independent Student Reading
- Practical work **(a group project)**
- Tutorials **(in lectures)** – Analytical/ Cognitive Analysis

---------- Course Assessment ----------

• Mid-term + Quizzes	**30%**
• Group Project/Assignment	**35%**
• Final	**35%**

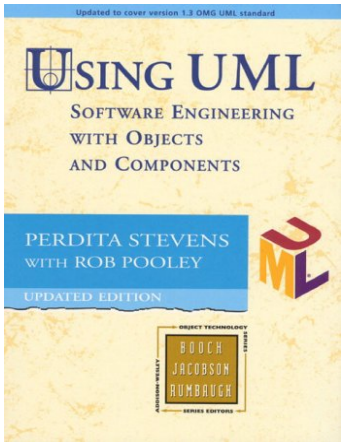
Recommended Course Textbooks

- Sommerville I. (2010)
  Software Engineering 9th Edition, Addison-Wesley, Harlow, Essex,UK (6th ,7th,or 8th would suffice)



- Bruegge and Dutoit, ***Object-Oriented Software Engineering Using UML, Patterns, and Java***, Prentice Hall **3rd** Edition

- Stevens P. with Pooley, R. (2005)
  Using UML: Software Engineering with Objects and Components,
  2nd Ed., Addison-Wesley, Harlow, Essex, UK



- Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich. (2005) ***Modern System Analysis and Design*** 4th - 6th Edition, Prentice Hall.

- *Roger Pressman (2014),* ***Software Engineering: A Practitioner's Approach*** *6-8th Edition, McGraw-Hill.*

What is the difference between software engineering and computer science?

Computer Science	Software Engineering
is concerned with	
theory fundamentals Algorithms, date structures, complexity theory, numerical methods	Understanding domain challenges the practicalities of developing and delivering useful quality software SE deals with practical problems in complex software products

Computer science theories are currently *insufficient* to act as a complete underpinning for software engineering, BUT they provide a foundation for practical aspects of software engineering