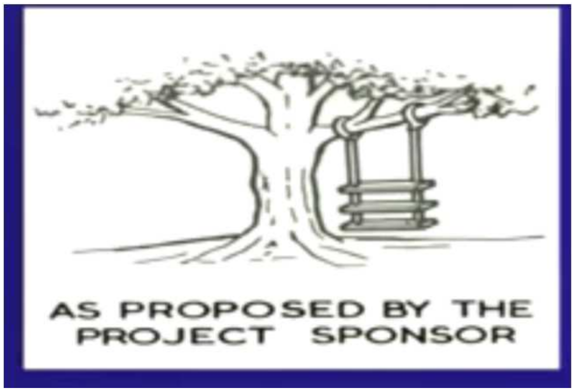
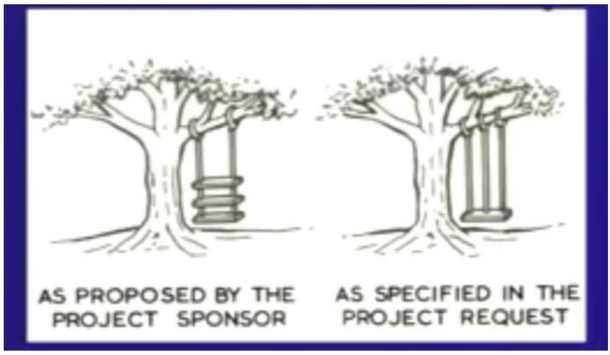
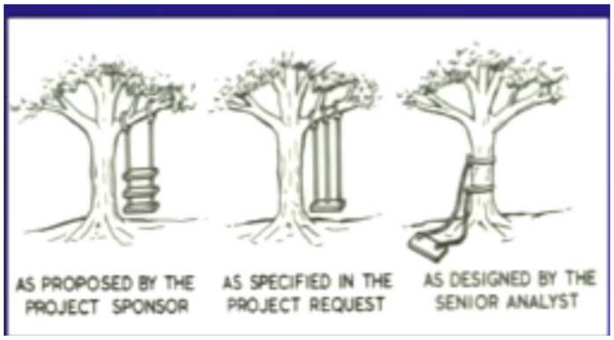
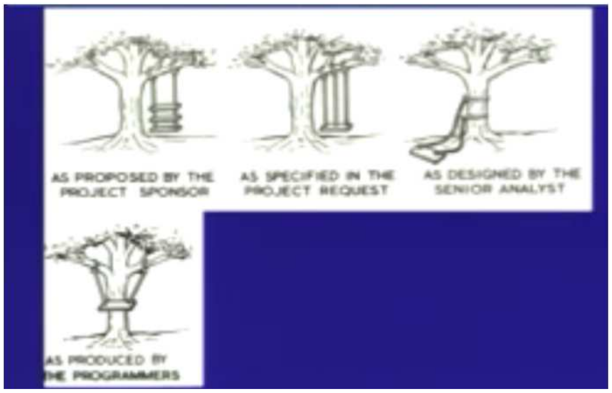
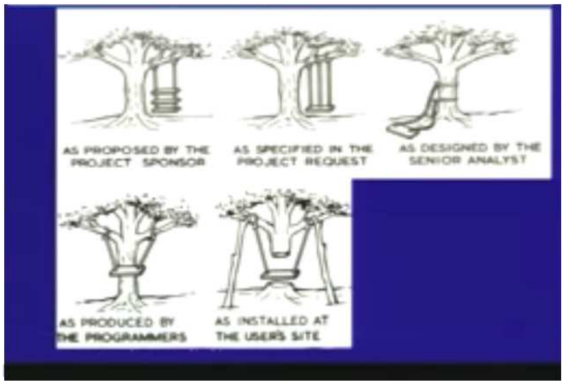
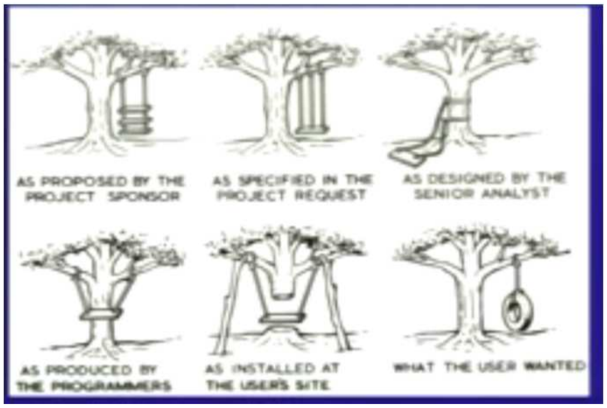

AS PROPOSED BY THE PROJECT SPONSOR


AS PROPOSED BY THE PROJECT SPONSOR
AS SPECIFIED IN THE PROJECT REQUEST

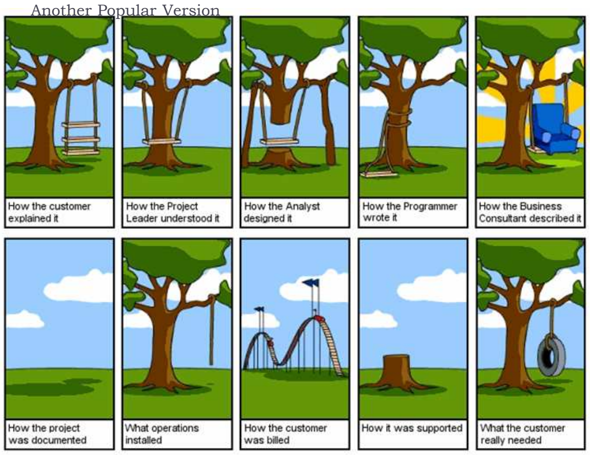

AS PROPOSED BY THE PROJECT SPONSOR
AS SPECIFIED IN THE PROJECT REQUEST
AS DESIGNED BY THE SENIOR ANALYST


AS PROPOSED BY THE PROJECT SPONSOR
AS SPECIFIED IN THE PROJECT REQUEST
AS DESIGNED BY THE SENIOR ANALYST
AS PRODUCED BY THE PROGRAMMERS


AS PROPOSED BY THE PROJECT SPONSOR
AS SPECIFIED IN THE PROJECT REQUEST
AS DESIGNED BY THE SENIOR ANALYST
AS PRODUCED BY THE PROGRAMMERS
AS INSTALLED AT THE USER'S SITE


AS PROPOSED BY THE PROJECT SPONSOR
AS SPECIFIED IN THE PROJECT REQUEST
AS DESIGNED BY THE SENIOR ANALYST
AS PRODUCED BY THE PROGRAMMERS
AS INSTALLED AT THE USER'S SITE
WHAT THE USER WANTED

Another Popular Version


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Practical

- Divide yourselves into groups of 5.
- Task:
  - Define precisely and unambiguously the process of making a cup of coffee in the morning.
  - Your specification should consider all possible problems and provide a solution to deal with each.
- We will then compare each group's solutions.

Introduction

Slides adapted from A. Rau Chaplin
& Kirstie Hawkey

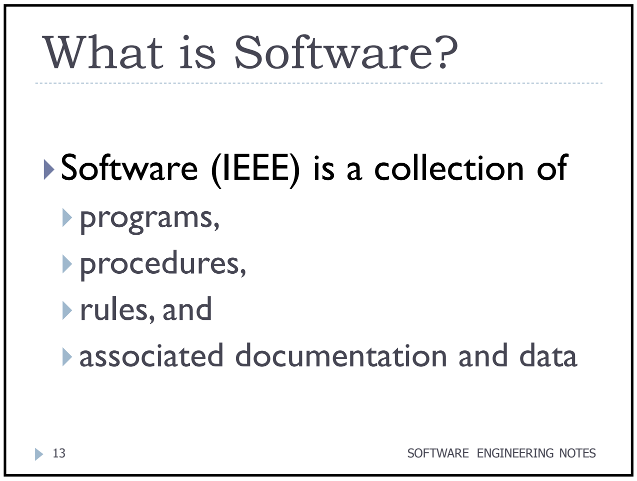
This Course

- Software Engineering is the systematic approach to development, deployment, operation, maintenance, and retirement of Software (SW).
- Basic Question in Software Engineering
  - *How to develop industrial-strength software?*

What this course will give?

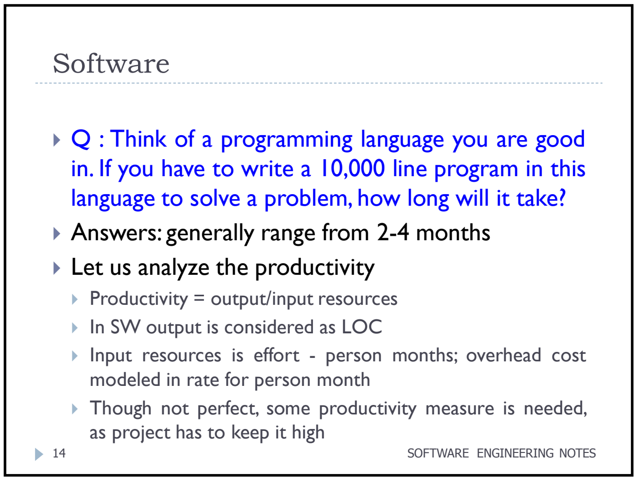
- **Main objective:** Give an idea of how industrial-strength software gets developed
- **At the end:** you should have the ability to plan, execute, and manage small software projects.
- **Lectures:** will discuss how to perform different tasks in a project

Let’s Begin…..



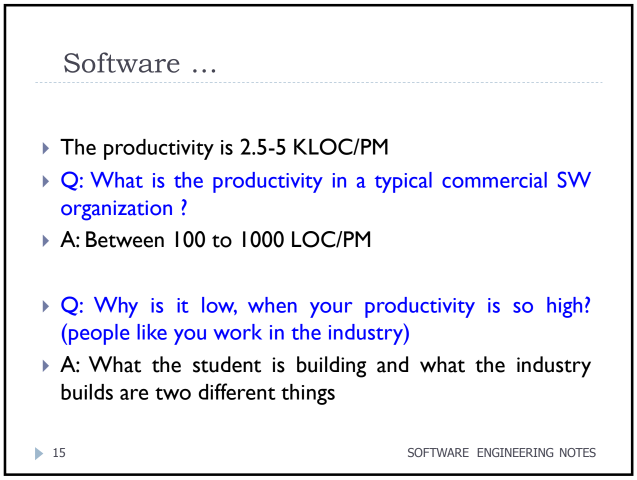
What is Software?

- Software (IEEE) is a collection of
  - programs,
  - procedures,
  - rules, and
  - associated documentation and data



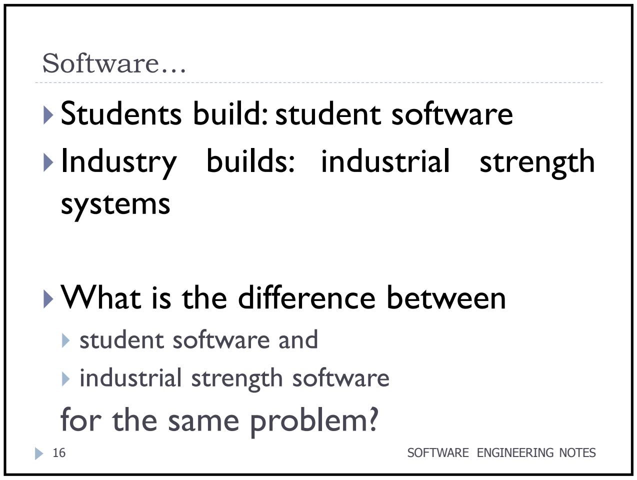
Software

- Q : Think of a programming language you are good in. If you have to write a 10,000 line program in this language to solve a problem, how long will it take?
- Answers: generally range from 2-4 months
- Let us analyze the productivity
  - Productivity = output/input resources
  - In SW output is considered as LOC
  - Input resources is effort - person months; overhead cost modeled in rate for person month
  - Though not perfect, some productivity measure is needed, as project has to keep it high



Software …

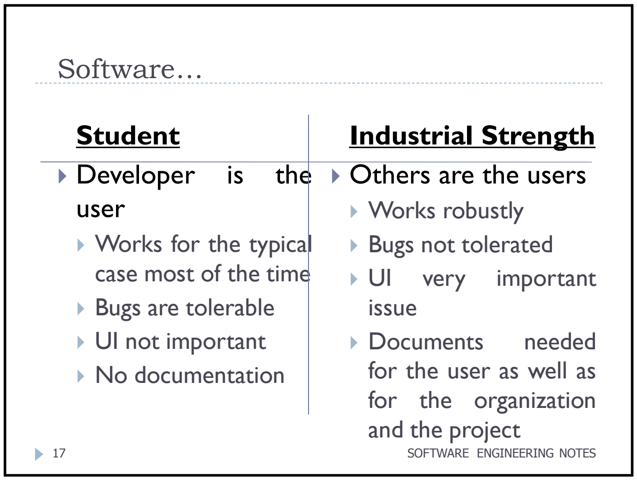
- The productivity is 2.5-5 KLOC/PM
- Q: What is the productivity in a typical commercial SW organization ?
- A: Between 100 to 1000 LOC/PM
- Q: Why is it low, when your productivity is so high? (people like you work in the industry)
- A: What the student is building and what the industry builds are two different things



Software…

- Students build: student software
- Industry builds: industrial strength systems
- What is the difference between
  - student software and
  - industrial strength software

  for the same problem?



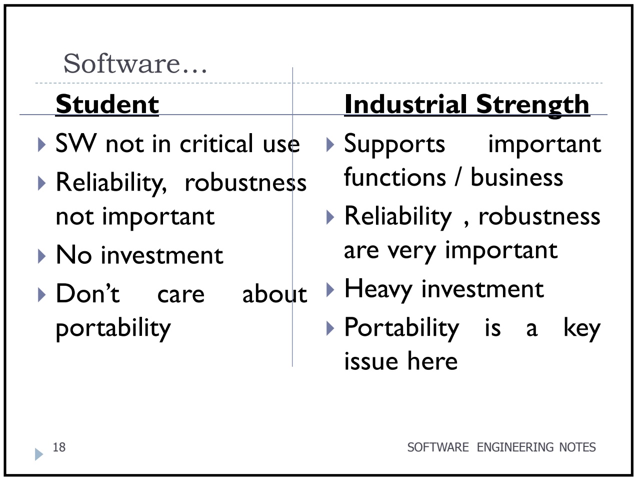
Software…

Student

- Developer is the user
  - Works for the typical case most of the time
  - Bugs are tolerable
  - UI not important
  - No documentation

Industrial Strength

- Others are the users
  - Works robustly
  - Bugs not tolerated
  - UI very important issue
  - Documents needed for the user as well as for the organization and the project



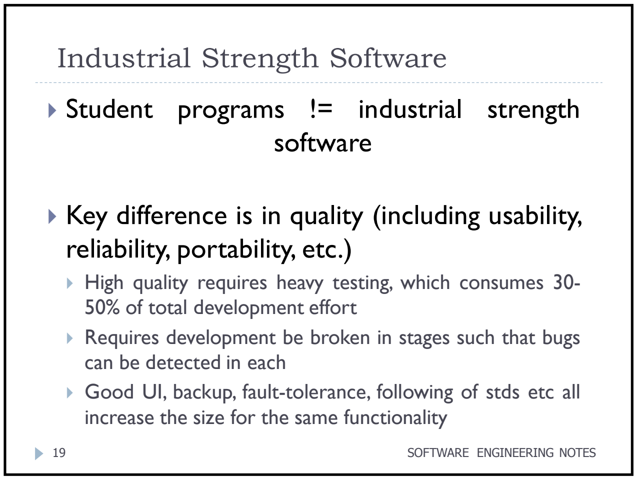
Software…

Student

- SW not in critical use
- Reliability, robustness not important
- No investment
- Don’t care about portability

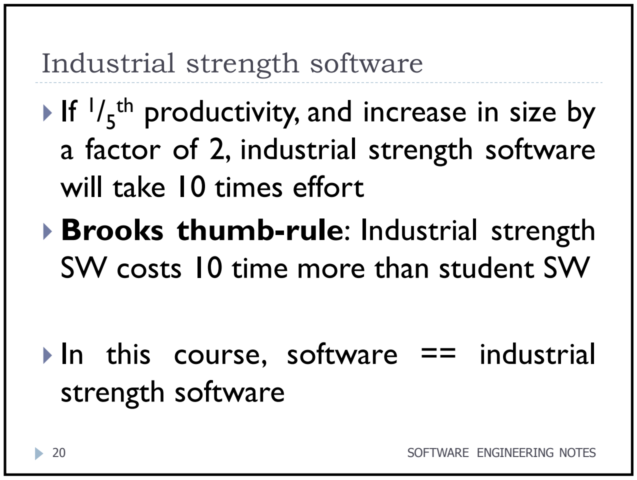
Industrial Strength

- Supports important functions / business
- Reliability , robustness are very important
- Heavy investment
- Portability is a key issue here



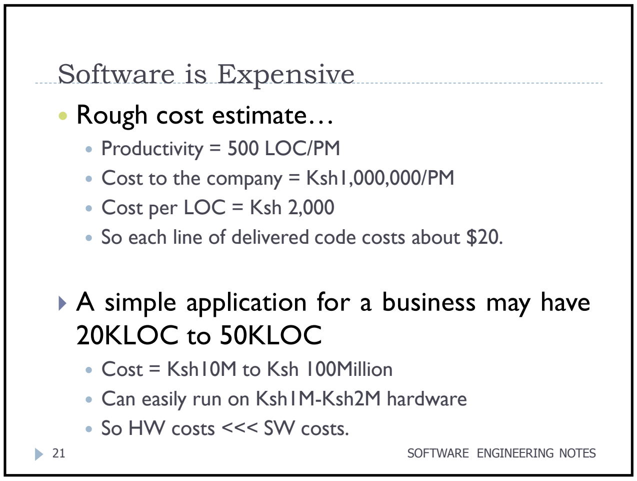
Industrial Strength Software

- Student programs != industrial strength software
- Key difference is in quality (including usability, reliability, portability, etc.)
  - High quality requires heavy testing, which consumes 30-50% of total development effort
  - Requires development be broken in stages such that bugs can be detected in each
  - Good UI, backup, fault-tolerance, following of stds etc all increase the size for the same functionality



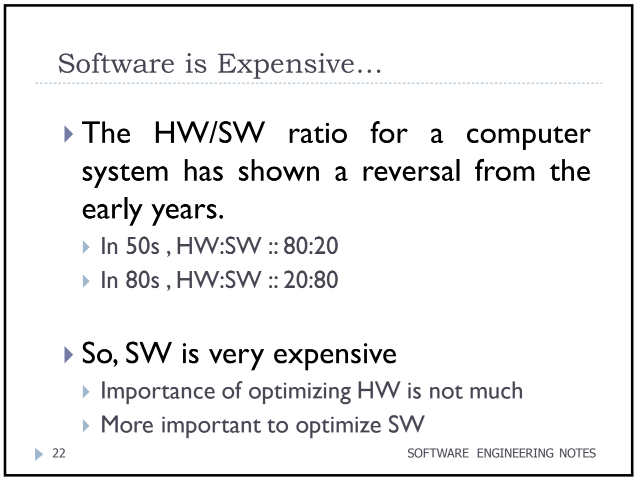
Industrial strength software

- If $^{1}/_{5}{}^{th}$ productivity, and increase in size by a factor of 2, industrial strength software will take 10 times effort
- **Brooks thumb-rule**: Industrial strength SW costs 10 time more than student SW
- In this course, software == industrial strength software



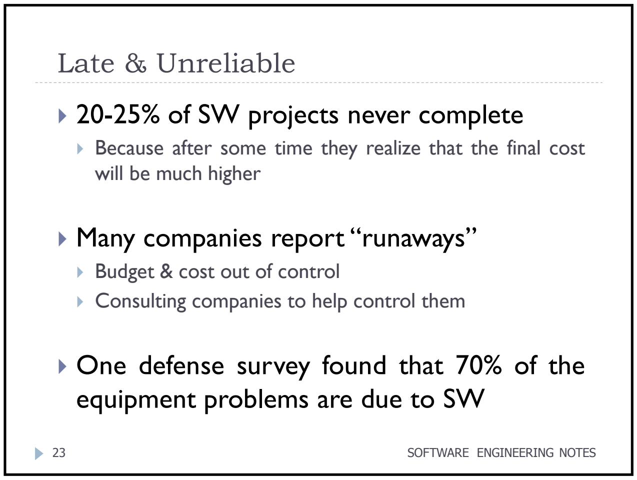
Software is Expensive

- Rough cost estimate…
  - Productivity = 500 LOC/PM
  - Cost to the company = Ksh1,000,000/PM
  - Cost per LOC = Ksh 2,000
  - So each line of delivered code costs about $20.
- A simple application for a business may have 20KLOC to 50KLOC
  - Cost = Ksh10M to Ksh 100Million
  - Can easily run on Ksh1M-Ksh2M hardware
  - So HW costs <<< SW costs.



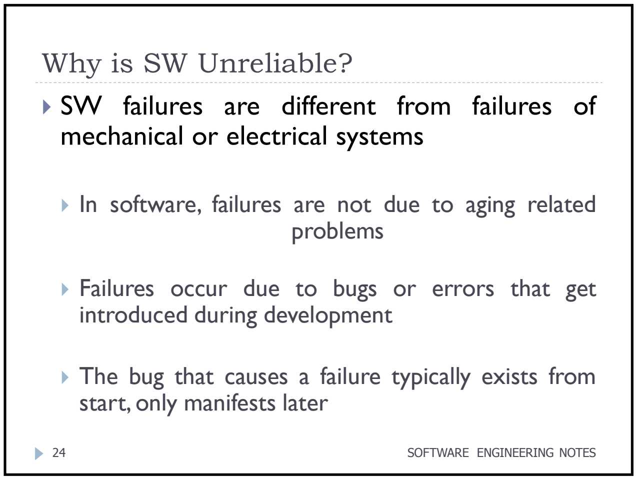
Software is Expensive…

- The HW/SW ratio for a computer system has shown a reversal from the early years.
  - In 50s , HW:SW :: 80:20
  - In 80s , HW:SW :: 20:80
- So, SW is very expensive
  - Importance of optimizing HW is not much
  - More important to optimize SW



Late & Unreliable

- 20-25% of SW projects never complete
  - Because after some time they realize that the final cost will be much higher
- Many companies report "runaways"
  - Budget & cost out of control
  - Consulting companies to help control them
- One defense survey found that 70% of the equipment problems are due to SW



Why is SW Unreliable?

- SW failures are different from failures of mechanical or electrical systems
  - In software, failures are not due to aging related problems
  - Failures occur due to bugs or errors that get introduced during development
  - The bug that causes a failure typically exists from start, only manifests later

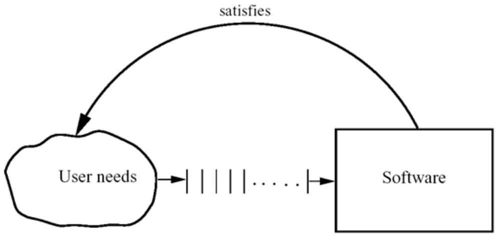
Maintenance

- Once SW delivered, it enters maintenance phase
- Why is maintenance needed for SW when it does not wear with age?
  - Residual errors requiring corrective maintenance
  - Upgrades and environment changes – adaptive maintenance
- Over SW lifetime, maintenance can cost more than the development cost of SW

What is Software Engineering?

- Problem domain discussed before, now we discuss the area of SE
- **SE (IEEE): systematic approach to development,…., *(fill the blanks)* of software**
- Systematic approach: methodologies and practices that can be used to solve a problem from problem domain

Basic Problem



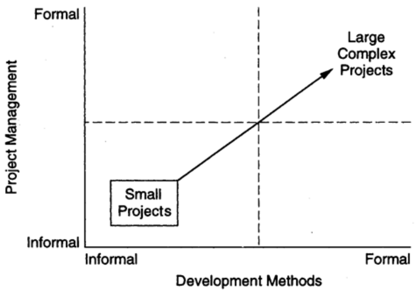
SE Challenges

- The problem of producing software to satisfy user needs drives the approaches used in SE
- Q: What other factors drive the selection of a SE approach?
  1. scale,
  2. productivity,
  3. quality,
  4. consistency,
  5. rate of change, …

1) Scale

- SE must deal with problem of scale
  - methods for solving small problems do not scale up for large problems
  - industrial strength SW problems tend to be large
- SE methods must be scalable
- Two clear dimensions in this
  1. **engineering methods**
  2. **project management**
- For small, both can be informal or ad-hoc, for large both have to be formalized

1) Scale…



1) Scale…

- An illustration of the issue of scale is counting the number of people in a room vs taking a census
  - Both are counting problems
  - Methods used in first not useful for census
  - For large scale counting problem, must use different techniques and models
  - Management will become critical

2) Productivity

- An engineering project is driven by **cost** and **schedule**
- **Cost:** In sw, cost is mainly manpower cost; hence, it is measured in person-months
- **Schedule** is in months/weeks – very important in business context
- In Biz context
  - Cost and Schedule can not be separated
  - SE must serve the Biz, NOT the other way around

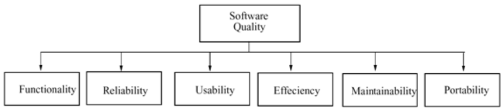
2) Productivity

- Productivity captures both Cost and Schedule
  - If P is higher, cost is lower
  - If P is higher, time taken can be less
- Approaches used by SE must deliver high Productivity

3) Quality

- Quality is the other major driving factor
- Developing high Quality SW is a basic goal
- Quality of SW is harder to define
- Approaches used should produce a high Quality software

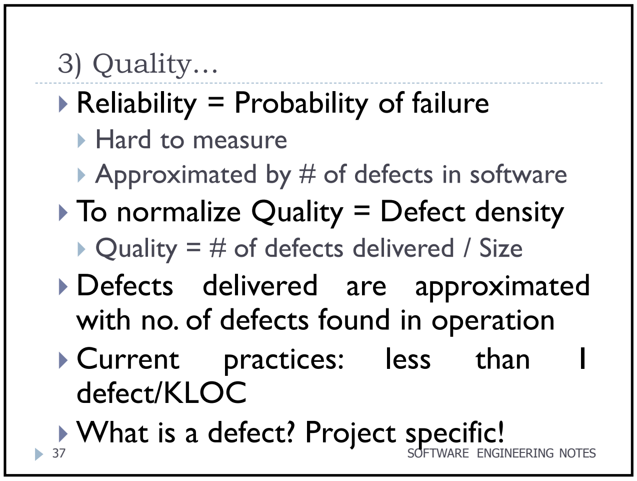
3) Quality – ISO standard



- ISO standard has six attributes
  1. Functionality
  2. Reliability
  3. Usability
  4. Efficiency
  5. Maintainability
  6. Portability

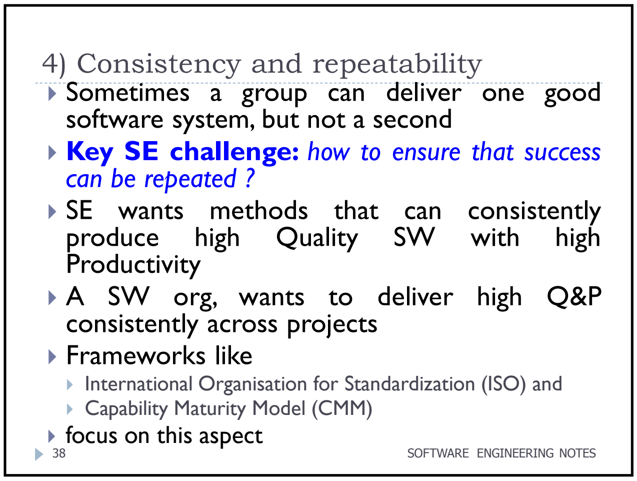
3) Quality…

- Multiple dimensions mean that it not not easy to reduce Q to a single number
- Concept of Q is project specific
  - For some reliability is most important
  - For others usability may be more important
- Reliability is generally considered the main Q criterion



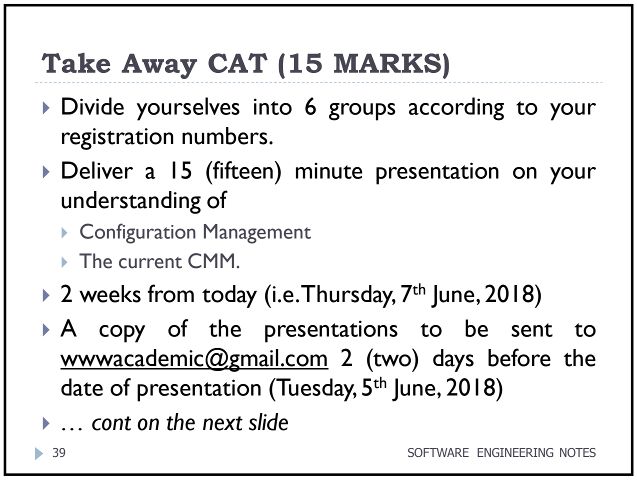
3) Quality…

- Reliability = Probability of failure
  - Hard to measure
  - Approximated by # of defects in software
- To normalize Quality = Defect density
  - Quality = # of defects delivered / Size
- Defects delivered are approximated with no. of defects found in operation
- Current practices: less than 1 defect/KLOC
- What is a defect? Project specific!



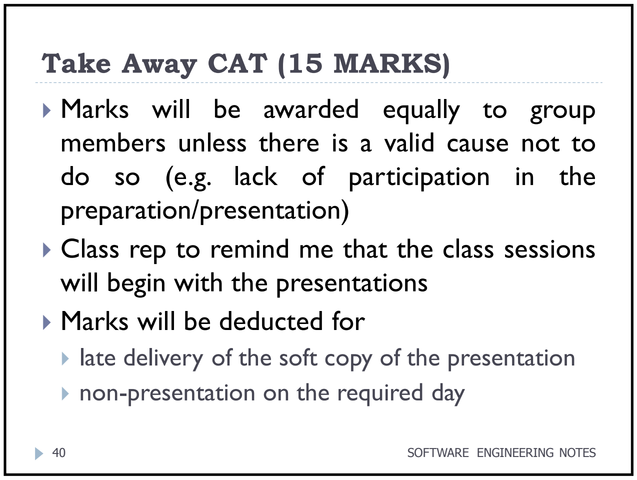
4) Consistency and repeatability

- Sometimes a group can deliver one good software system, but not a second
- **Key SE challenge:** *how to ensure that success can be repeated ?*
- SE wants methods that can consistently produce high Quality SW with high Productivity
- A SW org, wants to deliver high Q&P consistently across projects
- Frameworks like
  - International Organisation for Standardization (ISO) and
  - Capability Maturity Model (CMM)
- focus on this aspect



Take Away CAT (15 MARKS)

- Divide yourselves into 6 groups according to your registration numbers.
- Deliver a 15 (fifteen) minute presentation on your understanding of
  - Configuration Management
  - The current CMM.
- 2 weeks from today (i.e. Thursday, 7th June, 2018)
- A copy of the presentations to be sent to wwwacademic@gmail.com 2 (two) days before the date of presentation (Tuesday, 5th June, 2018)
- *… cont on the next slide*



Take Away CAT (15 MARKS)

- Marks will be awarded equally to group members unless there is a valid cause not to do so (e.g. lack of participation in the preparation/presentation)
- Class rep to remind me that the class sessions will begin with the presentations
- Marks will be deducted for
  - late delivery of the soft copy of the presentation
  - non-presentation on the required day

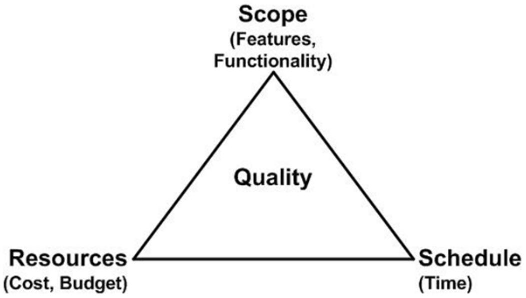
5) Rate of Change

- Only constant in business is change!
- Software must change to support the changing business needs
- SE practices must accommodate change
  - Methods that disallow change, even if high Q and P, are of little value

Goals of Industrial Strength SE

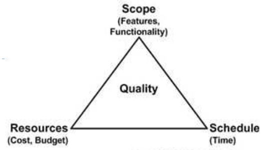
- **Consistently develop SW with high Q&P for large scale problems, under change**
- Q&P are the basic objectives to be achieved
- Q&P governed by people, processes, and technology

Iron Triangle



Copyright 2003-2006 Scott W. Ambler

Iron Triangle

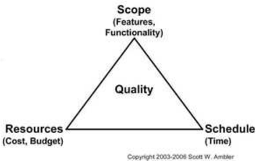


Copyright 2003-2006 Scott W. Ambler

- What happens when you break the triangle?
- **1) The project gets canceled**.
  - 15% of projects are cancelled before they deliver a system.
  - A study of 1,027 IT projects cited scope management related to serial practices as the single largest contributing factor to project failure in 82% of the projects and was given a overall weighted failure influence of 25%.

www.ambysoft.com/essays/brokenTriangle.htm

Iron Triangle



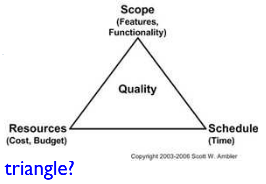
- What happens when you break the triangle?

2) The Project is delivered late, over budget, or both

- According to the **Chaos Report** 51% of projects are challenged (severely over budget and/or late), with an average cost overrun of 43%.

www.ambysoft.com/essays/brokenTriangle.htm

Iron Triangle



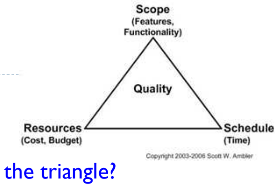
- What happens when you break the triangle?

3) The Project delivers poor quality software.

- When development teams are forced to deliver more functionality than they have time or resources for, they are often motivated to take short cuts which inevitably result in poor quality.

www.ambysoft.com/essays/brokenTriangle.htm

Iron Triangle



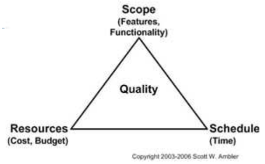
- What happens when you break the triangle?

4) The project under delivers.

- The team fails to deliver all of the required functionality.

www.ambysoft.com/essays/brokenTriangle.htm

Iron Triangle…



- What to do about it?
- ***Recognize that the iron triangle must be respected.***
- So
  - Vary the Scope
  - Vary the Schedule
  - Vary the Resources
  - Vary two or more factors

www.ambysoft.com/essays/brokenTriangle.htm

SE Methodology

- SE focuses mostly on processes for achieving the goals
- Process must be systematic
- SE separates process for developing sw from the developed product (i.e the sw)
- **Premise:** Process largely determines Q&P, hence suitable processes will lead to high Q&P

SE Methodology…

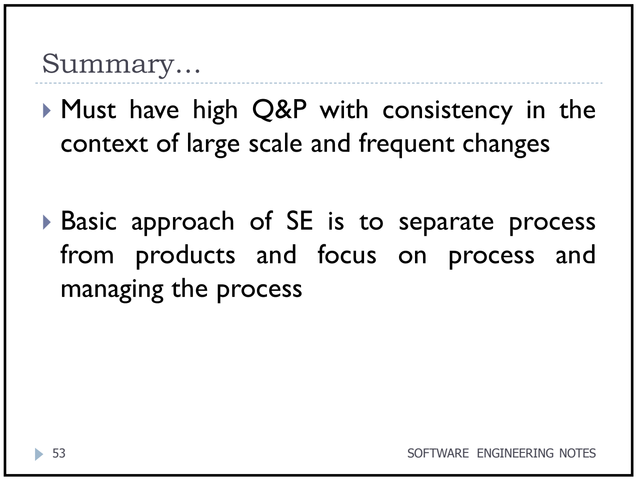
- Design of proper processes and their control is a key challenge SE faces
- SW process is the equivalent of manufacturing process
- This focus on process makes SE different from many CS courses

SE Methodology…

- The development process used in SE is typically phased
- Phases separate concerns with each phase focusing on some aspect
  - Requirements, architecture, design, coding, testing are key phases
- This phased process has to be properly managed to achieve the objectives
  - Metrics and measurement important for this

Summary

- The problem domain for SE is industrial strength software
- Software comprises programs, documentation, and data
- SE aims to provide methods for systematically developing SW
- **Main goal** – achieve high quality and productivity (Q&P)



Summary…

- Must have high Q&P with consistency in the context of large scale and frequent changes
- Basic approach of SE is to separate process from products and focus on process and managing the process