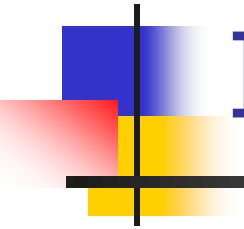


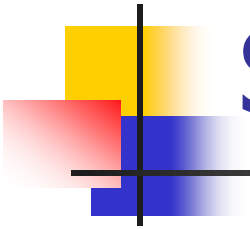
Introduction

Software

- Q : If you have to write a 10,000 line program in C 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…

- In a univ a student system is built while the commercial org builds industrial strength sw
- What is the difference between a student program and industrial strength sw for the same problem?
- Software (IEEE): collection of programs, procedures, rules, and associated documentation and data

Software…

Student

- Developer is the user
  - bugs are tolerable
  - UI not important
  - No documentation

Industrial Strength

- Others are the users
  - bugs not tolerated
  - UI v. imp. 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 for a problem & industrial strength software are two different things
- Key difference is in quality (including usability, reliability, portability, etc.)
- Brooks thumb-rule: Industrial strength sw costs 10 time more than student sw
- In this course, software means industrial strength software
- This software has some characteristics

Is Expensive

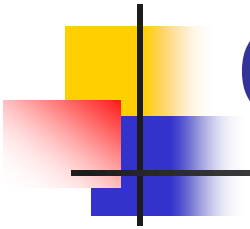
- Let us look at costs involved
  - Productivity = Appx 1000 LOC/PM
  - Cost = $3K to $10K/PM
  - Cost per LOC = $5 to $15
  - I.e, each line of delivered code costs many $s
- A simple application for a business may have 20KLOC to 50KLOC
  - Cost = $100K to $2.25Million
  - Can easily run on $10K-$20K hardware
  - So HW costs in an IT solution are small as compared to SW costs.

Requires tight Schedule

- Business requirements today demand short delivery times for software
- In the past, software products have often failed to be completed in time
- Along with cost, cycle time is a fundamental driving force

Productivity – for cost and schedule

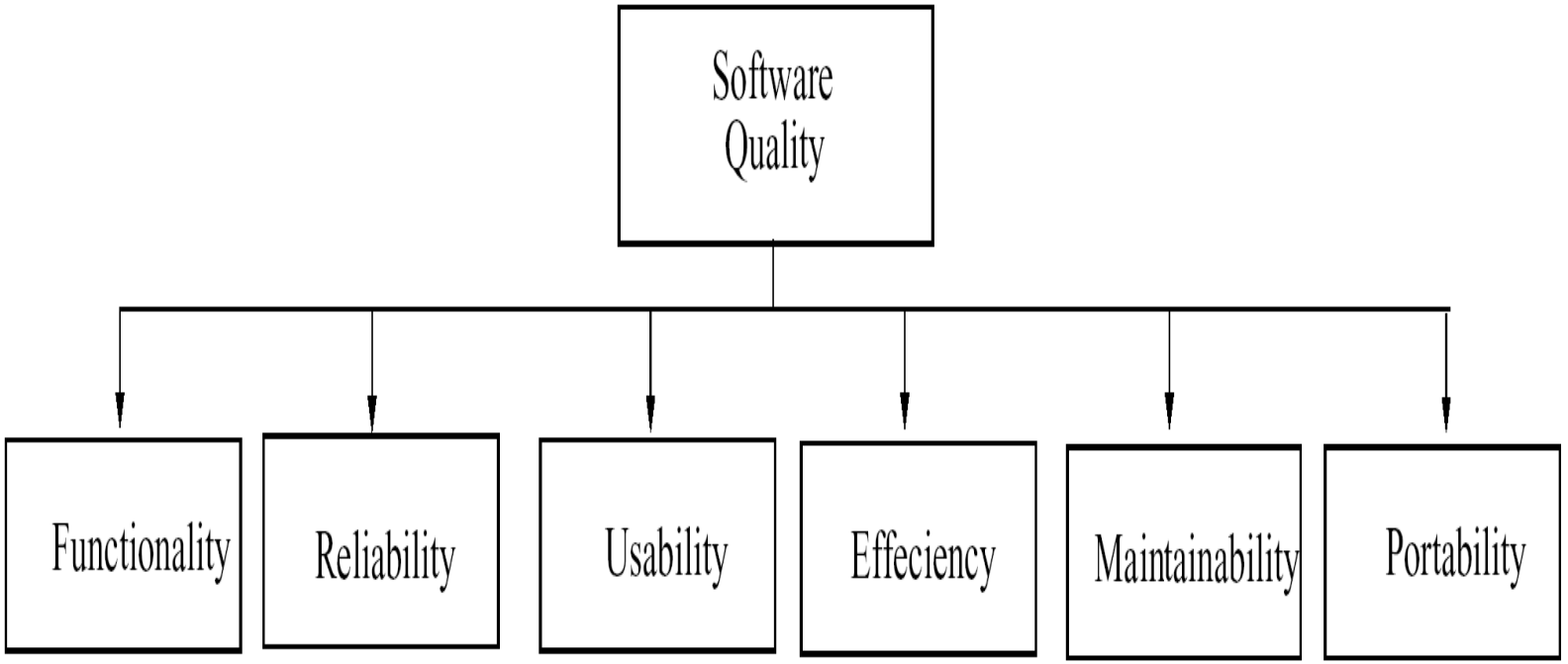
- An industrial strength software project is driven by cost and schedule
- Both can be modeled by productivity, measured in terms of output per unit effort (e.g. LOC per person month)
  - Higher productivity leads to lower cost
  - Higher productivity leads to lower cycle time
- Hence, for projects (to deliver software), quality and productivity are basic drivers

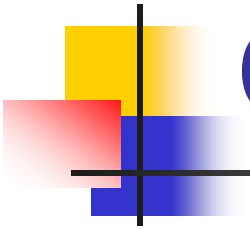


Quality

- Along with productivity, quality is the other major driving factor
- Developing high Q sw is a basic goal
- Quality of sw is harder to define

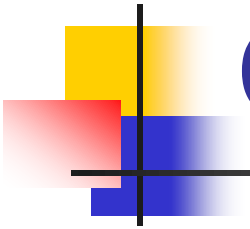
Quality – ISO standard





Quality – ISO std…

- ISO std has six attributes
  - Functionality
  - Reliability
  - Usability
  - Efficiency
  - Maintainability
  - Portability



Quality…

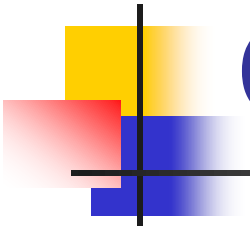
- Multiple dimensions mean that not easy to reduce Q to a single number
- Concept of Q is project specific
  - For some reliability is most important
  - For others usability may be more important
- Reliability is generally considered the main Q criterion

Quality…

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

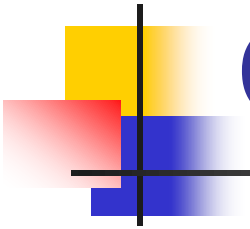
Quality – Maintainability

- Once sw delivered, it enters the maintenance phase, in which
  - Residual errors are fixed – this is corrective maintenance
  - Upgrades and environment changes are done – this is adaptive maintenance
- Maintenance can consume more effort than development over the life of the software (can even be 20:80 ratio!)
- Hence maintainability is another quality attribute of great interest



Quality and Productivity

- Hence, quality and productivity (Q&P) are the basic drivers in a sw project
- The aim of most methodologies is to deliver software with a high Q&P
- Besides the need to achieve high Q&P there are some other needs



Change

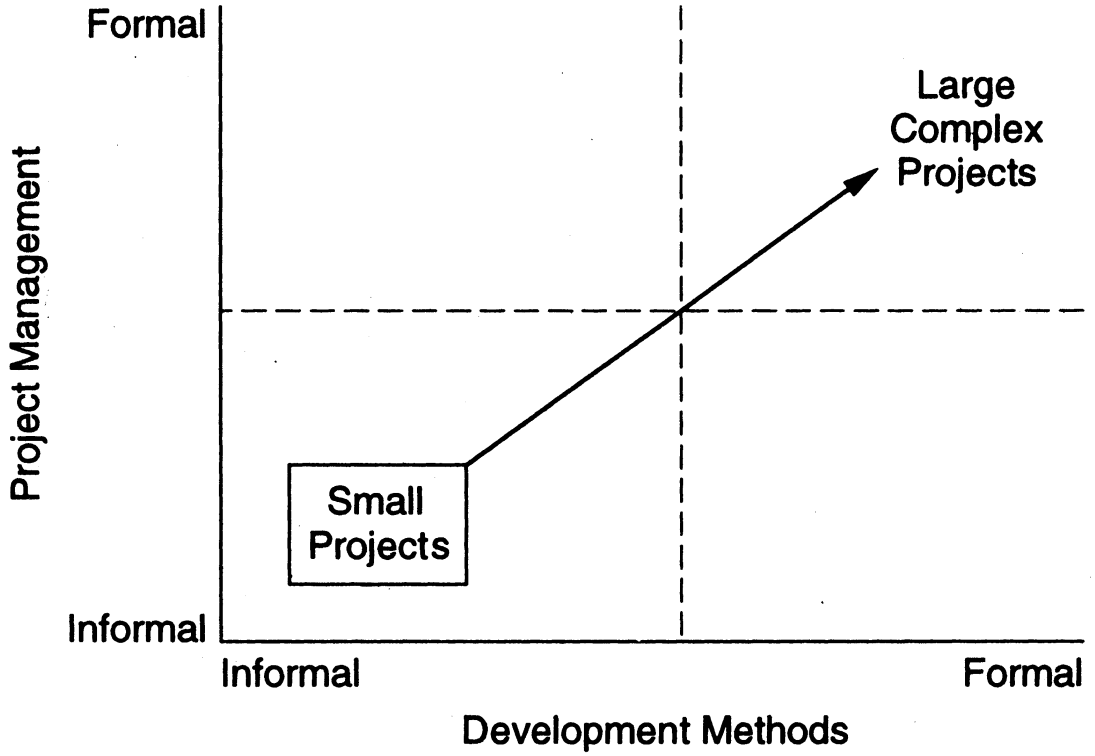
- Only constant in business is change!
- Requirements change, even while the project is in progress
- In a project, up to 40% of development effort may go in implementing changes

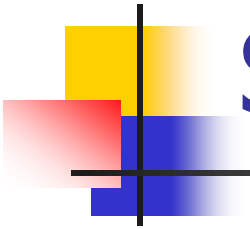
- Practices used for developing software must accommodate change

Scale

- Most industrial strength software tend to be large and complex
- Methods for solving small problems do not often scale up for large problems
- Two clear dimensions in a project
  - engineering
  - project management
- For small, both can be done informally, for large both have to be formalized

Scale…


Formal
Project Management
Large
Complex
Projects
Small
Projects
Informal
Informal
Formal
Development Methods

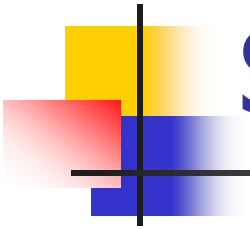


Scale…

- An illustration of 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

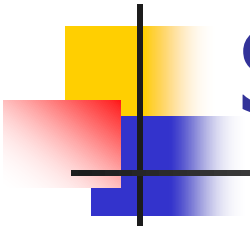
Scale: Examples

Gcc	980KLOC	C, C++, yacc
Perl	320 KLOC	C, perl, sh
Appache	100 KLOC	C, sh
Linux	30,000 KLOC	C, c++
Windows XP	40,000 KLOC	C, C++



Scale…

- As industry strength software tends to be large, hence methods used for building these must be able to scale up
- For much of the discussion, we will high Q&P as the basic objective



Summary

- The problem domain for SE is industrial strength software
- SE aims to provide methods for systematically developing (industrial strength) software
- Besides developing software the goal is to achieve high quality and productivity (Q&P)
- Methods used must accommodate changes, and must be able to handle large problems