CS 335: SOFTWARE ENGINEERING

jlungo@udsm.ac.tz

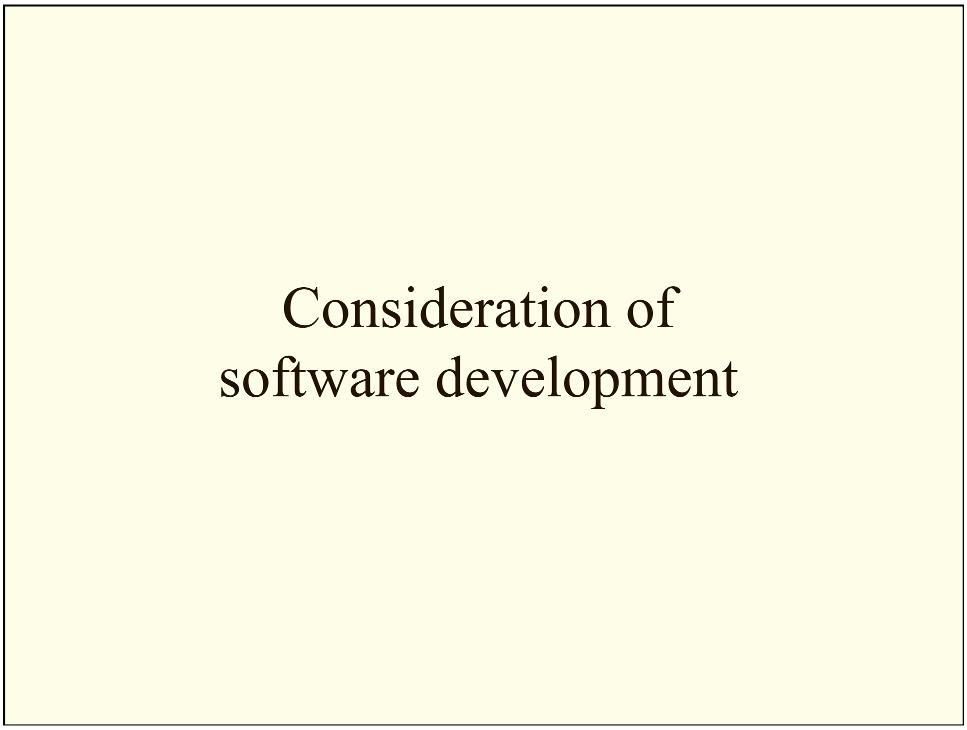
DR. J. H. LUNGO

TEACHING TIMETABLE

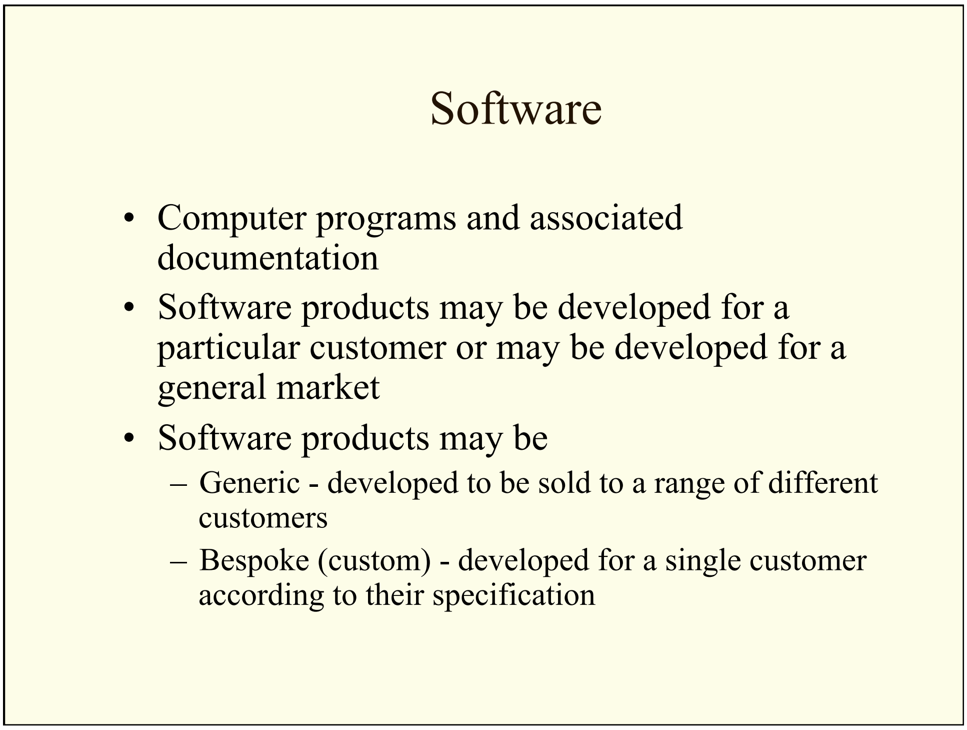
- IS 272 – Semester I

- Lecture Sessions
  - Wednesday 12:00 – 14:00 – B302

- Tutorial Sessions
  - Tuesday 12:00 – 13:00 – D01



Consideration of software development



Software

- Computer programs and associated documentation
- Software products may be developed for a particular customer or may be developed for a general market
- Software products may be
  - Generic - developed to be sold to a range of different customers
  - Bespoke (custom) - developed for a single customer according to their specification

Attributes of good software

- The software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable
- Maintainability
  - Software must evolve to meet changing needs
- Dependability
  - Software must be trustworthy
- Efficiency
  - Software should not make wasteful use of system resources
- Usability
  - Software must be usable by the users for which it was designed

Motivation

- The economies of ALL developed nations are dependent on software
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for professional software development
- Software engineering expenditure represents a significant fraction of GNP in all developed countries

What is software engineering?

- Software engineering is an engineering discipline which is concerned with all aspects of software production
- Software engineers should adopt a systematic and organised approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available

About Software Engineering

- Software Engineering is a Modeling Activity
  - In software development, software engineers build many different models of the system and of the application domain
- Software Engineering is a Problem-Solving Activity
  - Models are used to search for an acceptable solution
- Software Engineering is a Knowledge Acquisition Activity
  - In modeling the application and solution domain, software engineers collect data, organize it into information, and formalize it into knowledge
- Software Engineering is a Rationale-Driven Activity
  - Software engineers need to capture the context in which decisions were made and the rationale behind these decisions

software development process

- A set of activities whose goal is the development or evolution of software
- Generic activities in all software processes are:
  - Specification - what the system should do and its development constraints
  - Development - production of the software system
  - Validation - checking that the software is what the customer wants
  - Evolution - changing the software in response to changing demands

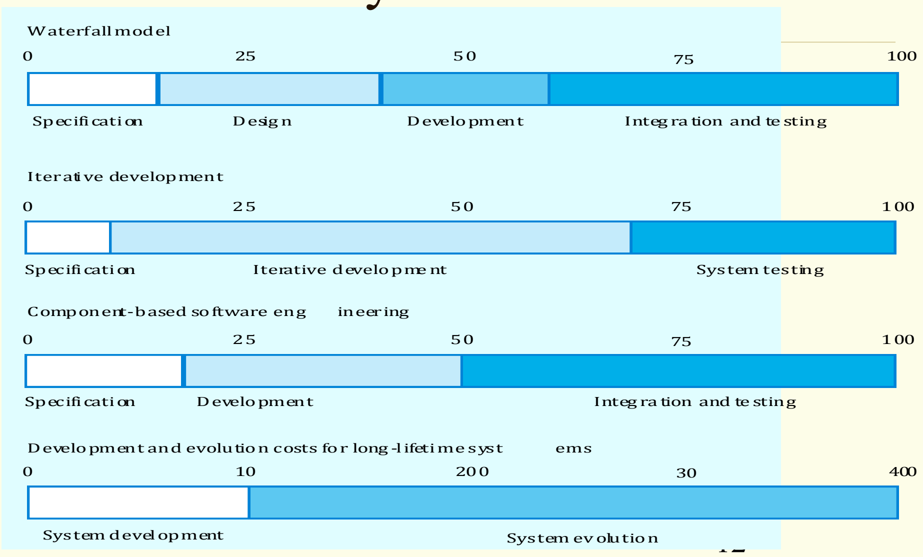
software development process model

- A simplified representation of a software process, presented from a specific perspective
- Examples of process perspectives are
  - Workflow perspective - sequence of activities
  - Data-flow perspective - information flow
  - Role/action perspective - who does what
- Generic process models
  - Waterfall
  - Evolutionary development
  - Formal transformation
  - Integration from reusable components

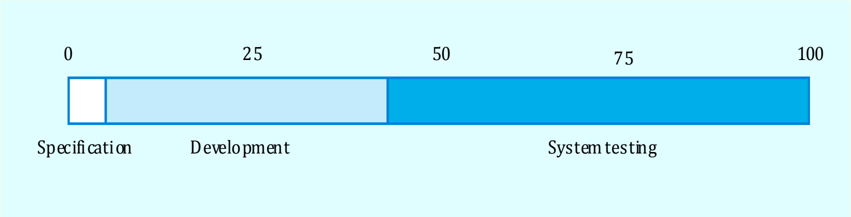
costs of software engineering

- Roughly 60% of costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs
- Costs vary depending on the type of system being developed and the requirements of system attributes such as performance and system reliability
- Distribution of costs depends on the development model that is used

Activity cost distribution



Product development costs



key challenges facing software engineering

- Coping with legacy systems, coping with increasing diversity and coping with demands for reduced delivery times
- Legacy systems
  - Old, valuable systems must be maintained and updated
- Heterogeneity
  - Systems are distributed and include a mix of hardware and software
- Delivery
  - There is increasing pressure for faster delivery of software

Professional and ethical responsibility

- Software engineering involves wider responsibilities than simply the application of technical skills
- Software engineers must behave in an honest and ethically responsible way if they are to be respected as professionals
- Ethical behaviour is more than simply upholding the law.

Issues of professional responsibility

- *Confidentiality*
  - Engineers should normally respect the confidentiality of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.
- *Competence*
  - Engineers should not misrepresent their level of competence. They should not knowingly accept work which is outwith their competence.

Issues of professional responsibility

- *Intellectual property rights*
  - Engineers should be aware of local laws governing the use of intellectual property such as patents, copyright, etc. They should be careful to ensure that the intellectual property of employers and clients is protected.
- *Computer misuse*
  - Software engineers should not use their technical skills to misuse other people's computers. Computer misuse ranges from relatively trivial (game playing on an employer's machine, say) to extremely serious (dissemination of viruses).

Example of Code of Ethics

- The Association of Computing Machinery (ACM) has a code of ethical practice.
- The Code contains eight Principles related to the behaviour of and decisions made by professional software engineers, including
  - practitioners, educators, managers, supervisors and policy makers, as well as trainees and students of the profession.

Code of ethics - preamble

- Preamble:
  - Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

Code of ethics - principles

- PUBLIC
  - Software engineers shall act consistently with the public interest.
- CLIENT AND EMPLOYER
  - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
- PRODUCT
  - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

Code of ethics - principles

- JUDGMENT
  - Software engineers shall maintain integrity and independence in their professional judgment.
- MANAGEMENT
  - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
- PROFESSION
  - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

Code of ethics - principles

- COLLEAGUES
  - Software engineers shall be fair to and supportive of their colleagues.
- SELF
  - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Ethical dilemmas

- Disagreement in principle with the policies of senior management.
- Your employer acts in an unethical way and releases a safety-critical system without finishing the testing of the system.
- Participation in the development of military weapons systems or nuclear systems.