



# EE102: Software Engineering I

## Section 6 – Software Engineering

### 6.1 Definition:

- **Software Engineering = Programming?**
  - No
  - Software Engineering is a more complex process
  - Programming is mainly focused on coding as part of the Software Engineering Process
  
- **Definitions**
  - An engineering discipline concerned with all practical aspects of software production
  - A discipline that uses sound engineering principles in order to produce:
    - useful, reliable, understandable, modifiable and efficient software
    - in a cost-effective manner and
    - on a timely basis

## 6.2 Motivation:

- **Complex software is difficult to manage**
  - Tens to hundreds of programmers
  - Thousands to millions of lines of code
  - Duration from months to few years
  - High costs
  - Often time pressure, safety and security issues
  
- **Avoid total or partial project failures (very high rate)**
  - 31.1% never completed, cancelled during development (according to US NIST)
  - 52.7% delivered less than fully functional, late or over-budget
  - 16.2% delivered in time, fully functional and on budget
  - U.S. economy loses an estimated \$59.5 billion annually because of software errors
  
- **Systematic work may improve things**
  - There are no guarantees
  - Stage-based development
  - Design-before-code approach
  - Good project management
  - Use tested techniques

- **Well known software-caused failures**

- 1960-70s: USAF F-16 fighter aircraft
  - after \$85 million for software development and \$250 million for software maintenance had a navigational software bug
- 1980s: Therac-25 – a software-driven radiation therapy machine
  - due to a software bug increased the radiation dose and killed several patients
- 1990s: Taurus - the London Stock Exchange system
  - after 3 years of development and £480 million was cancelled in March 1993

## 6.3 Phases in Software Engineering:

- **Requirements analysis and definition phase**

- The user specifies what he/she wants
- The software developer formalises what the user wants
- 85% of errors are made during the requirements phase

- **Design phase**

- Includes planning, architectural/system design and detailed design

- **Implementation (Coding) phase**

- The various components are coded and tested separately

- **Integration phase**

- The various components are integrated and tested as a whole

- **Testing phase**

- Verification (have we built the product right?)
- Validation (have we built the right product?)
- Acceptance Testing (the client tests it)

- **Maintenance phase**

- All modifications necessary after the client agreed with the product
- Includes corrections, perfective enhancements and adaptive enhancements

- **Retiring phase**

- The product is removed from service

## 6.4 Software Process Model

- **Definition**

- Organizes the software development process by specifying major phases and their order, the activities of each phase, the deliverables, (eventually) exit criteria

- **Necessary**

- Not, but recommended for obtaining a good software

- **Some Models**

- Build and fix model, V-Model
- Prototype model, Spiral model
- Object oriented model, **Waterfall model**, etc.

