Chapter 1

I. History of SE. (Early, Middle, Late)
II. Scope of SE

I. History of SE.
(a) Early years of computing. (1940 – 1960)
- Addition, subtraction, multiplication, …
  EN1AC → Eckert & MAuchly
- 1952 → First compiler – A0. by Grace Hopper
- 1958 ALGOL 58. First compiler / compiler.
- 1957 –1961 SAGE Sponsored by DOD.
  Semi Automated Ground Environment
  Detect ICBM.
  Characteristics:
  ~ $250m
  ~ 1000 people
  ~ 1m assembly instructions
  Problems:
  - Flexibility – requirement changes
  - People → skills vary.
    → organization. (structure).
  - Coding does not begin until SPEC.
  Solution / Suggestions:
  - Early spec is a must
  - Early system integration plan
    (precise interface, agreement among users)
  - Early Test Planning
  - Lean staffing in the beginning.
  - Have project plan and reviews.

(b) Middle years (1960 – 1980)
Wave of bigger project (>Million lines).
- The manned space program (1967)
  Apollo project by NASA (IBM)
  Characteristics:
  ~300 – 600 people
  ~ more than 1 billion $
  ~ more than 1 million Fortran / Assembly code.
- SABRE by United / American Airlines
- OS360 by IBM (F.P. Brooks)
  Characteristics:
  Over 200 M $
  More than 1 M instructions
  Did not stabilize until 16 releases (40% rewrite)
  Problems:
  - How to specify big systems.
  - How to produce quality software.
  - How to produce flexible software. (different platforms)
  - Education of people
1968 1st conference in SE in Garmisch.
50 people from industry, academic, gov.
sponsored by NATO.
SE is 32 years old.
→ structured Design Techniques (1970 - )
   (classical)
→ OOA, OOD, OOI. (1990 ~ )
   [Booch, Rambough, Jacobson]

(c) Late years (1980 - )
Problems

Conjecture: The cost of HW decreases by 50% every 12-18 months.

II. Scope of SE
(a) SE:
   * Schach: A discipline whose aim is the production of fault-free software that
     satisfies the user’s needs and that is delivered on time and within budget.

Quality, Cost, Time
IEEF: SE is application of systematic and disciplined, quantifiable approach
   to the development, operation and maintenance of software.
   i.e. Application of engineering of software and the study of approaches in it.

Foundation of SE.
The process, i.e., the way we build the software which includes life-cycle
   models, methods & tools & management.
1. Life cycle models
   Software life cycle
2%  ° Requirement Analysis
5% Specification
6% { Preiminary Design
 { Detailed Design
20% { Implementation
 { Testing (Validation) \rightarrow correction proof
67% Operation Maintenance Retire (Kill it)

From here, we have to focus on 2 things:
1. We have to look at the M issues (maintenance)
2. We have to do good REQ / SPEC Engineering.

Cost to Correct an Error

REQ SPEC DES IMP M

(b) Management
- Hiring / firing programmer
- Predicting cost (cocomo) by Boehm
- Predicting time
- Organizational problems
- Technology Transfer

1. Technology
   life cycle models
   methods
   Tools

2. Management
   organizational problems
   projecting cost / time