Chapter 2
Software Engineering History and State-of-Art

Contents

- 2.1 Evolution of Software Engineering
- 2.2 Software Engineering State-of-art

Evolution of Software Industry

- Independent Programming Service
- Software Product
- Enterprise Solution
- Packaged Software for the Mass

Independent Programming Services (Era 1)

- Feb 1955, Elmer Kubie and John Sheldon founded CUC
  - the First Software Company that devoted to the construction of software especially for hardware company.
- Promoting Software Industry: two Major Projects,
  - SABRE, airline reservation system, $30 million.
  - SAGE, air defense system (1949~1962)

    700/1000 programmers in the US. $8 billion.

Software Product (Era 2)

- 1964 Martin Goetz developed Flowchart Software -- Autoflow for RCA, but rejected.
    - Sale to the customer of RCA & IBM.
    - Develop and market software products not specifically designed for a particular hardware platform.
  - MARK IV, a pre-runner for the database management system.
- IBM unbundled software from hardware.

Enterprise Solutions (Era 3)

- Dietmar Hopp. IBM Germany
  - Systems, Applications and Products (SAP) $3.3billion (1997)
  - Setting up shop in Walldorf, Germany.
  - Marked by the emergence of enterprise solutions providers.

    e.g. Baan 1978. Netherlands. $680 million (1997)

    Oracle 1977. U.S.

    Larry Ellison.
  - ERP, $45 billion (1997)

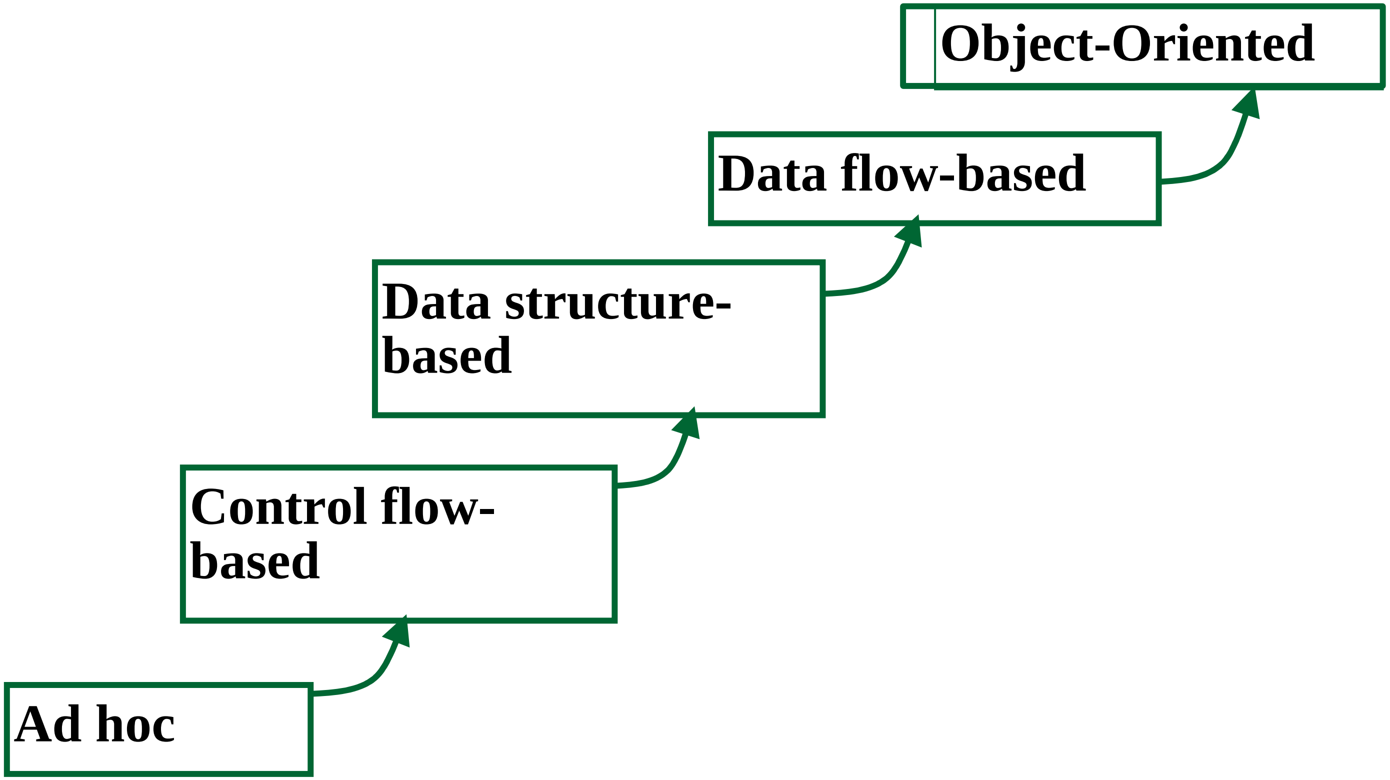
Packaged Software for the Masses (Era 4)

- Software products for the masses. 1979.
  - VisiCalc, Spreadsheet program.
- August 1981: The deal of the century.
  - Bill Gates bought the first version of the OS from a small firm called Seattle Computer Products for $50,000 without telling them it was for IBM.
  - The development of the IBM PC, 1981, initiated a 4th software era.
    - PC-based mass-market software. Few additional services are required for installation.
  - Microsoft reached revenues of $11.6 billion. Packaged Software Products, $57 billion (1997)

Internet Software and Services (Era 5)

- Internet and value-added services period, 1994. W
  - with Netscape's browser software for the internet.

Evolution of Design Techniques



Object-Oriented

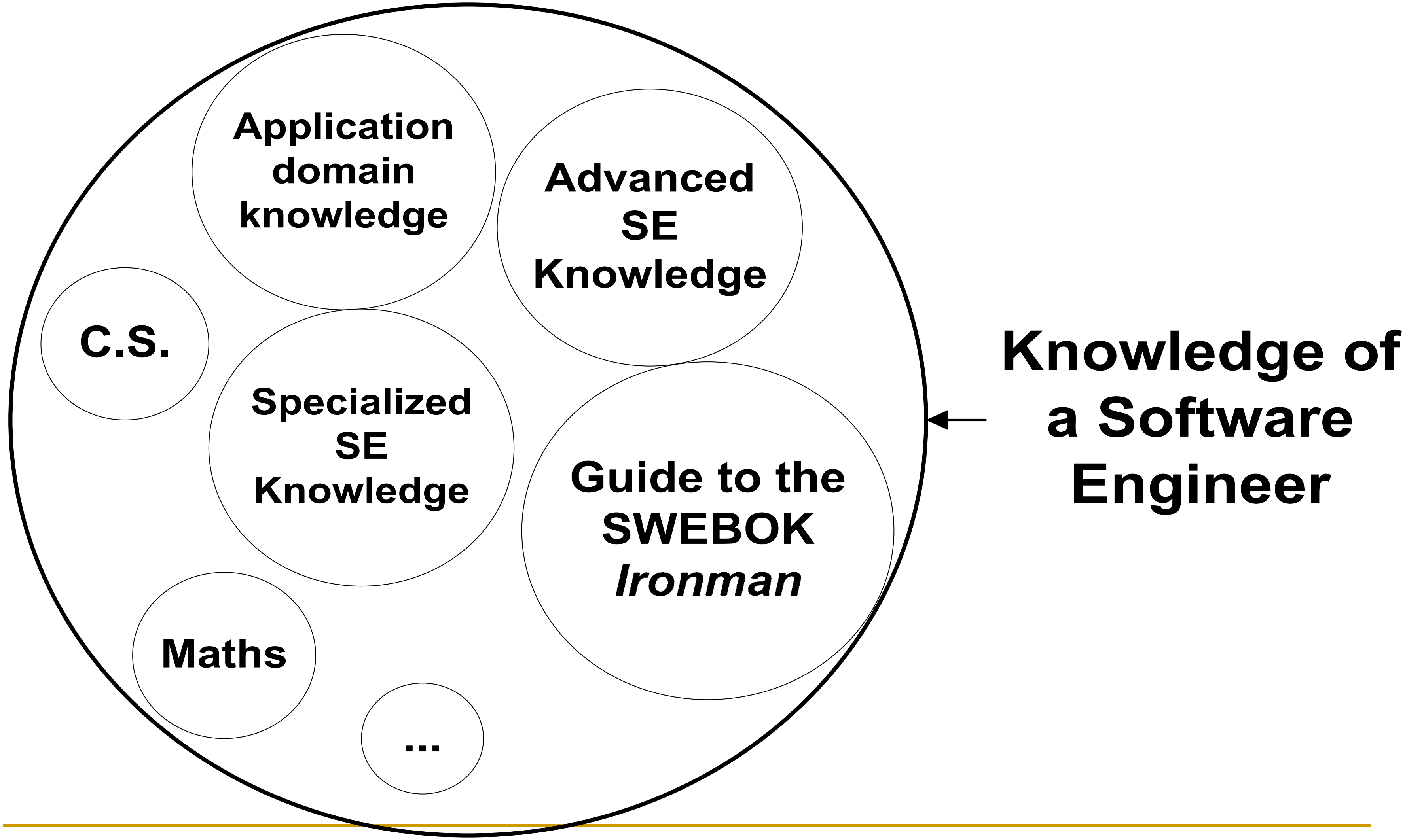
Data flow-based

Data structure-based

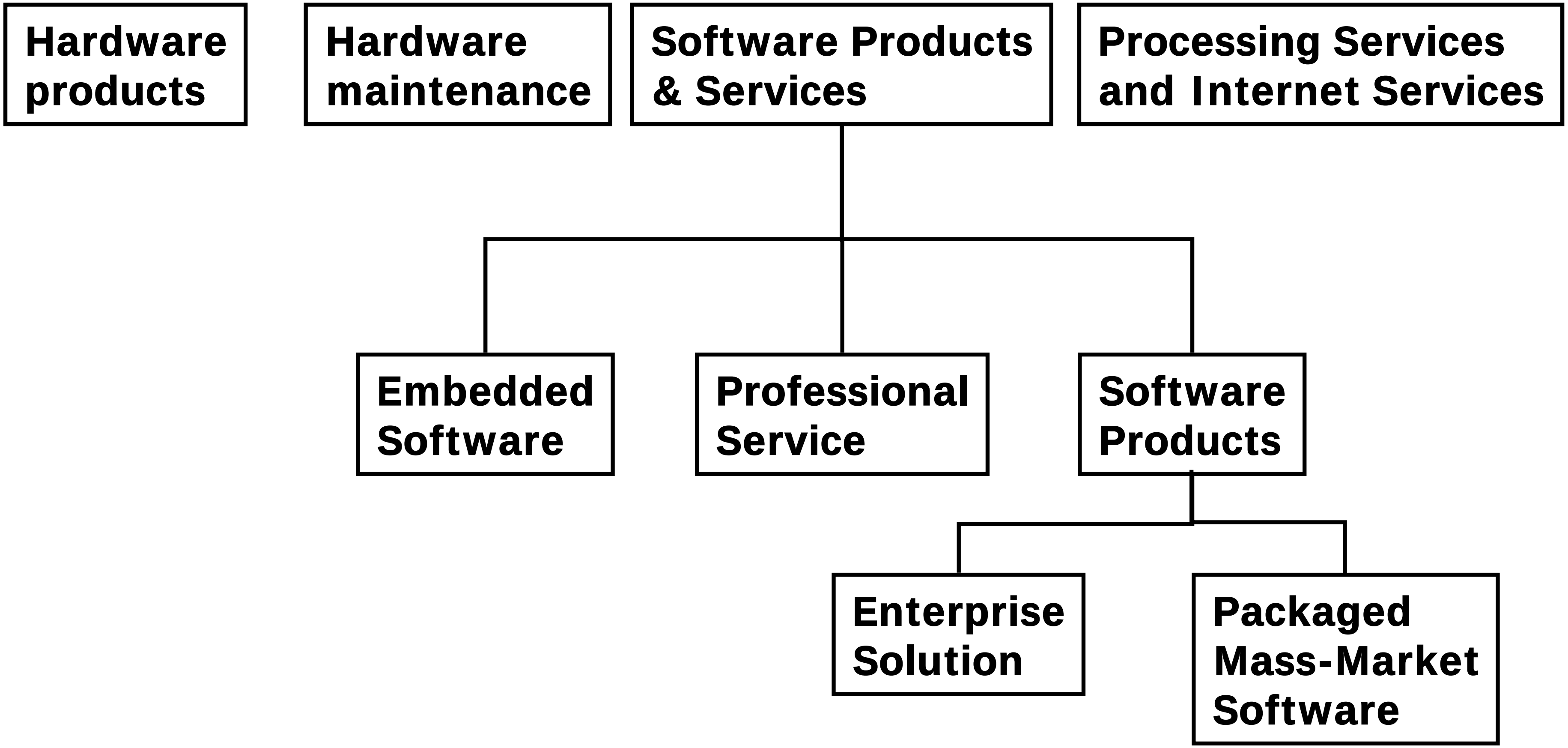
Control flow-based

Ad hoc

Related Knowledge


Application domain knowledge
Advanced SE Knowledge
C.S.
Specialized SE Knowledge
Guide to the SWEBOK *Ironman*
Maths
...
Knowledge of a Software Engineer

IT Market



Hardware products

Hardware maintenance

Software Products & Services

Processing Services and Internet Services

Embedded Software

Professional Service

Software Products

Enterprise Solution

Packaged Mass-Market Software

Software Products and Services

Professional Software Services

Anderson Consulting
IBM
EDS
CSC
Science Applications
Cap Gemini
Hp
DEC
Fujitsu
BSO Origin

Enterprise Solutions

IBM
Oracle
Computer Associates
SAP
HP
Fujitsu
Hitachi
Parametric Technology
People Soft
Siemens

Packaged Mars-Market Software

Microsoft
IBM
Computer Associates
Adobe
Novell
Symantec
Intuit
Autodesk
Apple
The Learning Company

Software Engineering Today?

- Organizations "go with what has worked in the past"
- Everyone is too busy getting product out the door to spend time in education or training or addressing these problems effectively
- "Out of date" practices become institutionalized

Software Engineering Today?

- Few people know, or can integrate, best practices
  - Unable to adopt and utilize proven methodologies in timely fashion
- Although significant improvements have been made in specific areas, the rapidly evolving nature of the software industry has resulted in little overall improvement in the overall situation.

Not Crisis, but a Chronic Problem

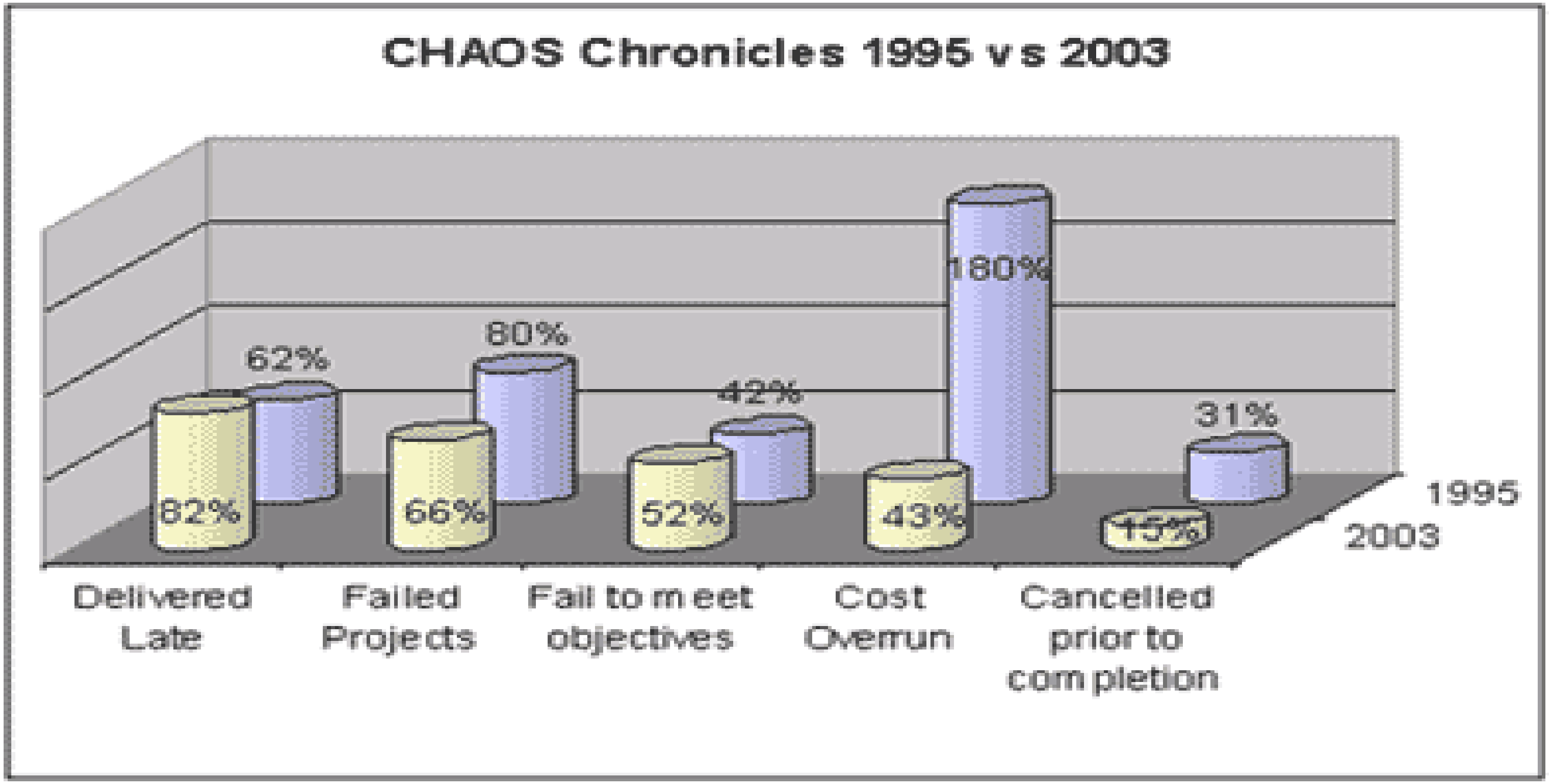
- The crisis persists
  - After 35 years later, the software “crisis” is still with us
  - Major problems are still the same:
    - poor quality (correctness, usability, maintainability, etc)
    - over budget
    - delivered late, or not at all
- It is not a crisis but a chronic problem
  - It becomes a persistent, chronic condition that software industry has to face with

What’s Wrong?

- Does software engineering have no progress at all? Not quite true.
  - We have indeed seen a lot of improvements, e.g. high level programming, object-oriented technology, etc.
- But it does not achieve its promise, why?
  - production of fault-free software, delivered on time and within budget, that satisfies the users’ needs, and is easy to maintain, etc.

A More Close Look

- The comparison with 1995's report does show that there is some progress in the past eight years.



So, What’s the Problem?

- Software issues: software industry has changes a lot in the past years
- Education issue: more emphasis on methods and tools but lack of sufficient education and training on people
- Process and quality issue: there lacks of a set of known proven practices for software engineers to follow with

Software Changes in the Past Years

- Changes in software over time:
  - grew in size from 10’s or 100’s of lines to 1000’s to 1,000,000’s of lines of code
  - operating environment changed from simple “batch” operations to complex multiprogramming systems, to time-sharing and distributed computing to today’s Internet network computing environment.

Software Changes in the Past Years

- As computer systems (both hardware and software) have become larger and more complex, the software development process has also become more and more complex
  - the simple art of “programming in the small” is no longer capable of coping with the task.

Situations for Software are Different Too

- Driven by intense market forces, including persistent pressure to deliver software on unrealistic time schedules
  - Rapidly changing requirements
  - Pressures for faster time to market
- Continuing rapid evolution of software methodologies and systems
  - Integration of new processes and techniques
  - Need to re-design major systems

Situations for Software are Different Too

- Talent shortage: needed software engineering skills often in short supply
- What even worse
  - Moore’s law means always trying new things
  - Complexity moves into software
  - Can’t find the limits except by trial and error

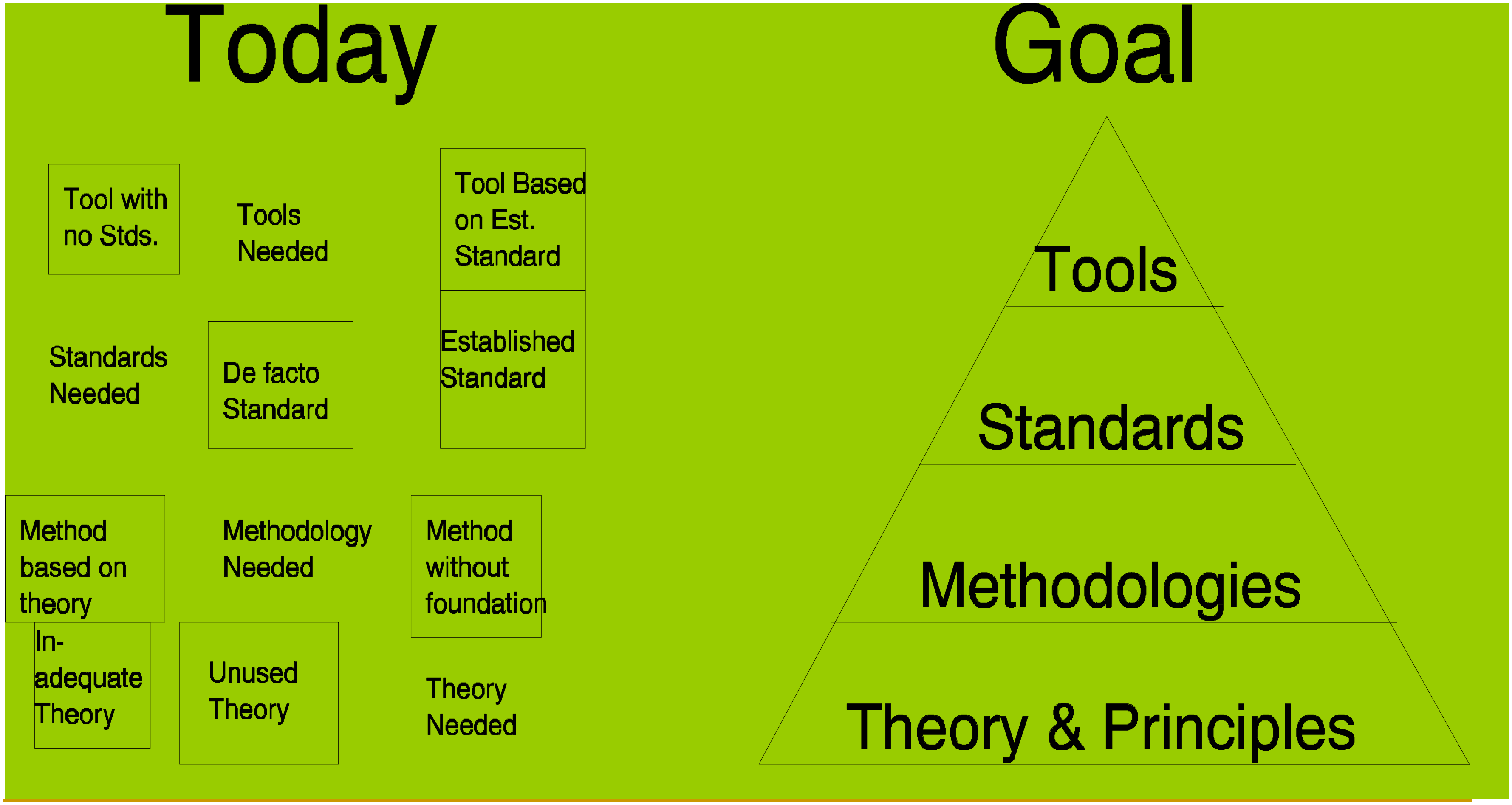
The Software Industry Today

- Even though much is now known about how to improve software production, the overall state is not much better than ever, due to the urgency of meeting unrealistic delivery schedules and the continuing rapid evolution of the software industry
    - **i.e. poor quality, late delivery, over budget**

The Software Industry Today

- Component-Based Engineering and Integration
- Technological Heterogeneity
- Enterprise Heterogeneity
- Greater potential for Dynamic Evolution
- Internet-Scale Deployment
- Many competing standards
- Much conflicting terminology

The Current State of Software Engineering



Today

Tool with no Stds.

Tools Needed

Tool Based on Est. Standard

Standards Needed

De facto Standard

Established Standard

Method based on theory

Methodology Needed

Method without foundation

In-adequate Theory

Unused Theory

Theory Needed

Goal

Tools

Standards

Methodologies

Theory & Principles

Three key Challenges

Software engineering in the 21st century faces three key challenges:

- **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

Ever-Present Difficulties

- Few guiding scientific principles
- Few universally applicable methods
- As much managerial / psychological / sociological as technological

Future of SE ...

- Process
- Requirements engineering
- Reverse engineering
- Testing
- Maintenance and Evolution
- Software architecture
- OO Modeling
- SE and Middleware
- Tools and environments
- Configuration management
- Databases and SE
- SE Education
- Software analysis
- Formal specification
- Mathematical foundations
- Reliability and Dependability
- Performance
- SE for Safety
- SE for security
- SE for mobility
- SE & the Internet
- Software economics
- Empirical studies of SE
- Software metrics