

# Software & Software Engineering

*Slide Set to accompany*  
*Software Engineering: A Practitioner's Approach, 7/e*  
**by Roger S. Pressman**

**Slides copyright © 1996, 2001, 2005, 2009 by Roger S. Pressman**

***For non-profit educational use only***

May be reproduced ONLY for student use at the university level when used in conjunction with *Software Engineering: A Practitioner's Approach, 7/e*. Any other reproduction or use is prohibited without the express written permission of the author.

All copyright information MUST appear if these slides are posted on a website for student use.

# Quick Recap

- What is a Software?
- Why Software Engineering Discipline?
- Definition of Software Engineering

# Today Agenda

- Characteristics of Software
- Legacy Software
- Another reason for Software Engineering
- Software Myths
- Recommended Textbook and Course Outline

# Characteristics of Software

## ❑ Characteristics of Software

- Intangible
- Software Developed not manufactured
- Software does not wear out
- Software mostly custom built

# Software Developed not Manufactured

❑ In spite of similarities, Not manufactured in classical way

**HW cannot be corrected in case of quality issues**

**SW can be corrected**

**To meet deadlines more people can be inducted**

**Not possible in case of SW**

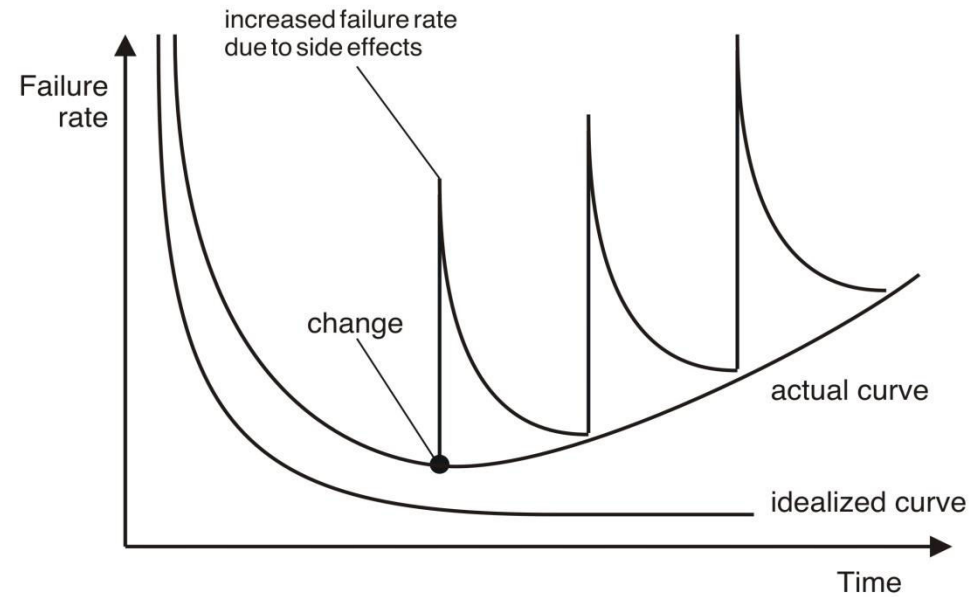
**HW cost concentrated in material**

**SW concentrated in engineering**

# Software are Custom Build

- ❑ Products other than Software Products are not custom build
- ❑ However Software Products are mostly custom build
- ❑ In present era, however 3<sup>rd</sup> Party modules are being used

# Software does not wear out



**Software Product**

# Software does not wear out but Legacy

❑ The older program often referred to as Legacy Software....





# Legacy Software

- ❑ Have poor quality
  - Inextensible design
  - Complicated code
  - Poor/no documentation
- ❑ They still perform core business functions & indispensable
- ❑ We do nothing until they go for **significant change**

# Legacy Software

## ❑ Significant change/ maintenance

- Adaptation
- Enhancement
- Correction

# Software Myths

- ❑ **Erroneous beliefs** about software and the process that is used to build it
- ❑ misleading attitudes - that have **caused serious problems** for managers and practitioners
- ❑ Today, most knowledgeable software engineering professionals **recognize myths**
- ❑ However, old attitudes and habits are **difficult to modify**, and remnants of software myths remain

# Management Myths

- ☐ We already have a book that's full of standards and procedures for building software. Won't that provide my people with everything they need to know?
- ☐ If we get behind schedule, we can add more programmers and catch up
- ☐ If I decide to outsource the software project to a third party, I can just relax and let that firm build it

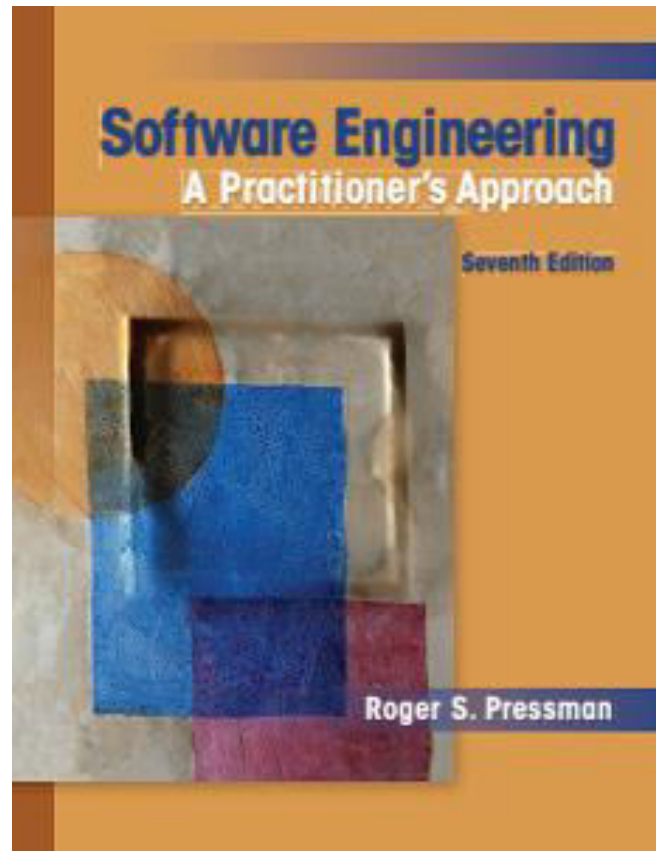
# Customer Myths

- ❑ A general statement of objectives is sufficient to begin writing programs—we can fill in the details later
- ❑ Software requirements continually change, but change can be easily accommodated because software is flexible

# Practitioners Myths

- ❑ Once we write the program and get it to work, our job is done
- ❑ Until I get the program “running” I have no way of assessing its quality
- ❑ The only deliverable work product for a successful project is the working program
- ❑ Software engineering will make us create big and unnecessary documentation and will invariably slow us down

# Recommended Book



**PART ONE      THE SOFTWARE PROCESS    29**

---

CHAPTER 2	Process Models	30
CHAPTER 3	Agile Development	65

---

**PART TWO      MODELING    95**

---

CHAPTER 4	Principles that Guide Practice	96
CHAPTER 5	Understanding Requirements	119
CHAPTER 6	Requirements Modeling: Scenarios, Information, and Analysis Classes	148
CHAPTER 7	Requirements Modeling: Flow, Behavior, Patterns, and WebApps	186
CHAPTER 8	Design Concepts	215
CHAPTER 9	Architectural Design	242
CHAPTER 10	Component-Level Design	276
CHAPTER 11	User Interface Design	312
CHAPTER 12	Pattern-Based Design	347
CHAPTER 13	WebApp Design	373

---

**PART THREE      QUALITY MANAGEMENT    397**

---

CHAPTER 14	Quality Concepts	398
CHAPTER 15	Review Techniques	416
CHAPTER 16	Software Quality Assurance	432
CHAPTER 17	Software Testing Strategies	449
CHAPTER 18	Testing Conventional Applications	481
CHAPTER 19	Testing Object-Oriented Applications	511
CHAPTER 20	Testing Web Applications	529
CHAPTER 21	Formal Modeling and Verification	557
CHAPTER 22	Software Configuration Management	584
CHAPTER 23	Product Metrics	613



# Summary

- ❑ Characteristics of Software
- ❑ Legacy Software
- ❑ Another reason for Software Engineering
- ❑ Software Myths
- ❑ Recommended textbook and course outline