



# *COMP 354: Introduction to Software Engineering*

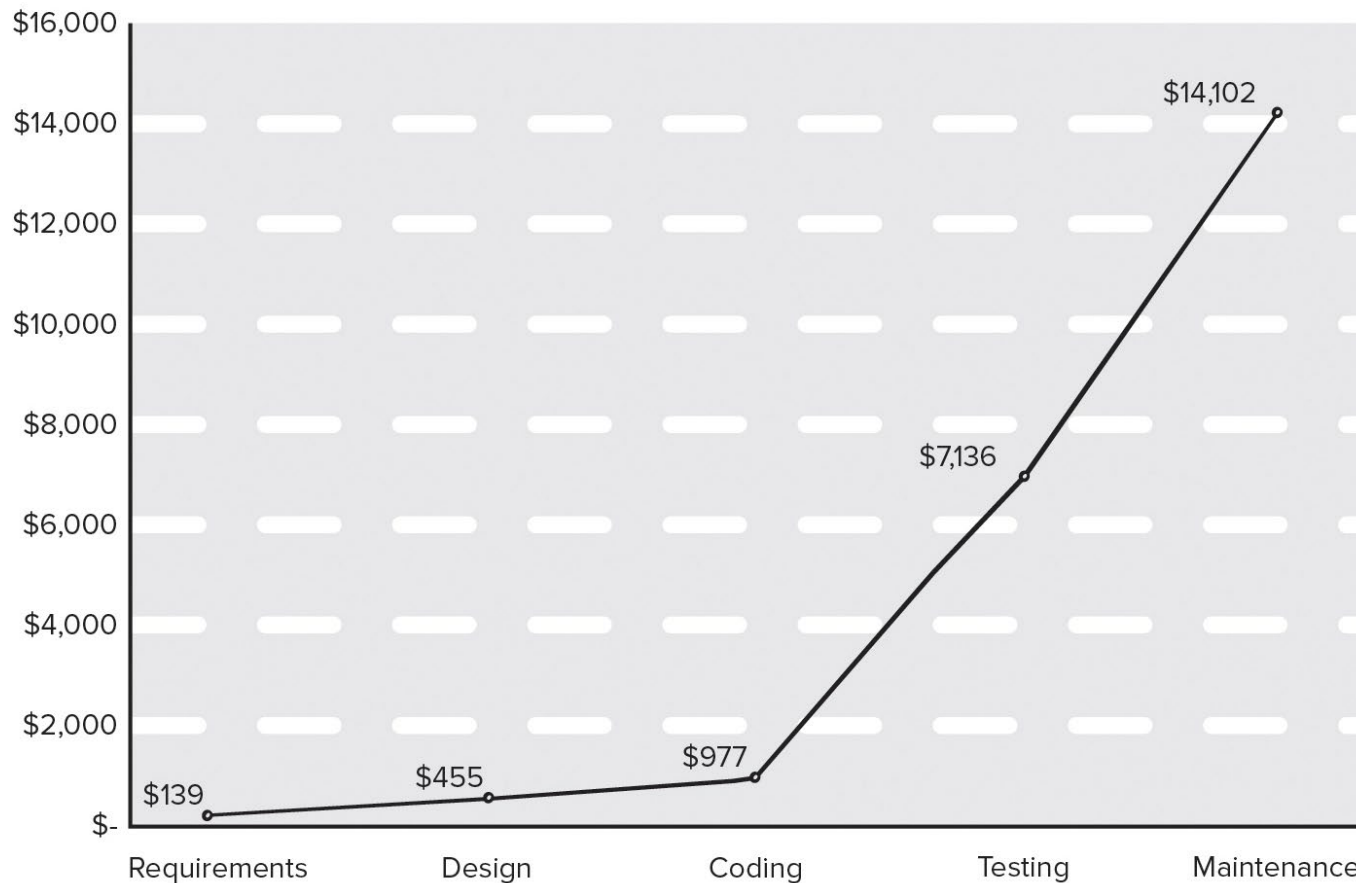
---

## Software Quality Concepts

Based on Chapter 15 of the textbook

# Relative Costs to Find and Repair a Defect

Copyright © McGraw-Hill Education. All rights reserved. No reproduction or distribution without the prior written consent of McGraw-Hill Education.



Source: Boehm, Barry and Basili, Victor R., "Software Defect Reduction Top 10 List," IEEE Computer, vol. 34, no. 1, January 2001.



# Software Quality

---

- Software quality can be defined as:
  - An effective software process applied in a manner that creates a useful product that provides measurable value for those who produce it and those who use it.
- Advantages of providing useful products:
  - Greater software product revenue.
  - Better profitability when an application supports a business process.
  - Improved availability of information that is crucial for the business.



# Software Quality – Effective Process

---

- An **effective software process** establishes infrastructure that supports building a high-quality software product.
- The management aspects of process create the checks and balances that help avoid project chaos—a key contributor to poor quality.
- Software engineering practices allow the developer to analyze the problem and design a solid solution—both critical to building high quality software.
- Umbrella activities such as change management and technical reviews have as much to do with quality as any other part of software engineering practice.



# Software Quality – Useful Product

---

- A **useful product** delivers the content, functions, and features that the end-user desires.
- But as important, it delivers these assets in a reliable, error free way.
- A useful product always satisfies those requirements that have been explicitly stated by stakeholders.
- A useful product satisfies a set of implicit requirement that are expected of all high-quality software.



# Software Quality – Adding Value

---

- By **adding value for both the producer and user** of a software product, high quality software provides benefits for the software organization and the end-user community.
- The software organization gains added value because high quality software requires less maintenance effort, fewer bug fixes, and reduced customer support.
- The user community gains added value because the application provides a useful capability in a way that expedites some business process.



# Achieving Software Quality

---

- Software quality is the result of good project management and solid engineering practice.
- To build high quality software you must understand the problem to be solved and be capable of creating a quality design that conforms to the problem requirements.
- Project management – project plan includes explicit techniques for quality and change management.
  - Use estimation to verify that delivery dates are achievable.
  - Schedule is understood and team avoids taking shortcuts.
  - Risk planning is conducted so problems do not breed chaos, software quality will be affected in a positive way.



# Achieving Software Quality

---

- Project plan should include explicit techniques for quality and change management.
- **Quality control** - series of inspections, reviews, and tests used to ensure conformance of a work product to its specifications.
- **Quality assurance** - consists of the auditing and reporting procedures used to provide management with data needed to make proactive decisions.
- Defect prediction is an important part of identifying software components that may have quality concerns.
- Machine learning and statistical models may help identify relationships between metrics and defection components.