University of Sadat City

Faculty of Computers and Artificial Intelligence (FCAI)

Software Engineering-2 (CS307)

Lecture 1

Presented By:

Dr. Ahmed Tealeb

Information Systems Department

2nd Semester 2022-2023

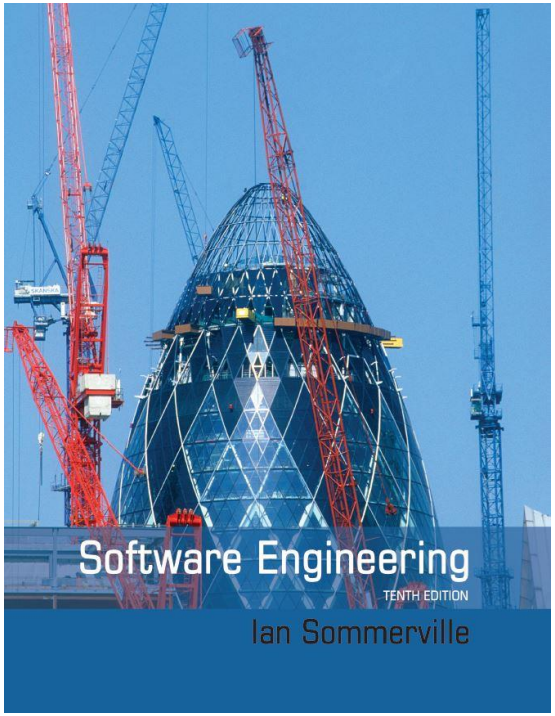
Course Objectives

The main objective of this course is to provide students with the critical systems:

- Dependability, critical systems specification, critical systems development.
- Security engineering, Distributed software engineering,
- Project management, Quality management, Process improvement, and Configuration management.

Textbook

Sommerville, Ian. (2016) "Software Engineering" 10th Ed., Pearson, ISBN 978-0133943030



Assessment:

Assessment	weight
Mid term exam	20%
Quizzes	10%
Practical exam (Project)	15%
Attendance	5%
Final exam	50%

Table of Contents

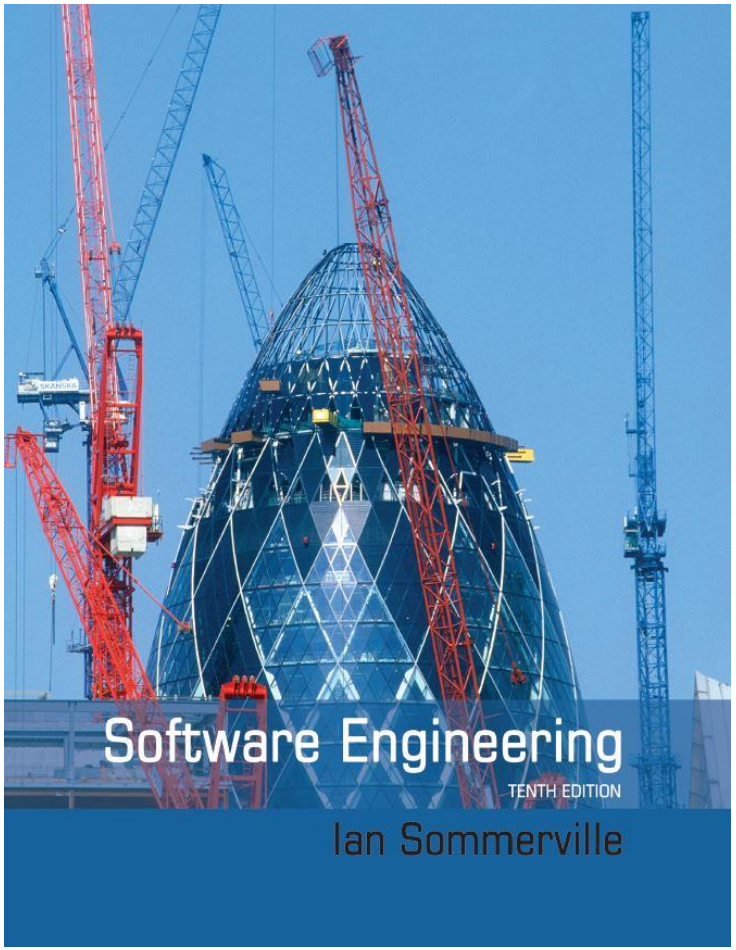
Table of Contents

Table of Contents Cont.

Table of Contents

Software Engineering

Tenth Edition



Chapter 10

Dependable systems

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Learning Objectives

10.1 Dependability properties

10.2 Sociotechnical systems

10.3 Redundancy and diversity

10.4 Dependable processes

10.5 Formal methods and dependability

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

System Dependability

- For many computer-based systems, the most important system property is the dependability of the system.
- The dependability of a system reflects the user's degree of trust in that system. It reflects the extent of the user's confidence that it will operate as users expect and that it will not 'fail' in normal use.
- Dependability covers the related systems attributes of reliability, availability and security. These are all inter-dependent.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Importance of Dependability

- System failures may have widespread effects with large numbers of people affected by the failure.
- Systems that are not dependable and are unreliable, unsafe or insecure may be rejected by their users.
- The costs of system failure may be very high if the failure leads to economic losses or physical damage.
- Undependable systems may cause information loss with a high consequent recovery cost.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Causes of Failure

- Hardware failure
  - Hardware fails because of design and manufacturing errors or because components have reached the end of their natural life.
- Software failure
  - Software fails due to errors in its specification, design or implementation.
- Operational failure
  - Human operators make mistakes. Now perhaps the largest single cause of system failures in socio-technical systems.

Pearson

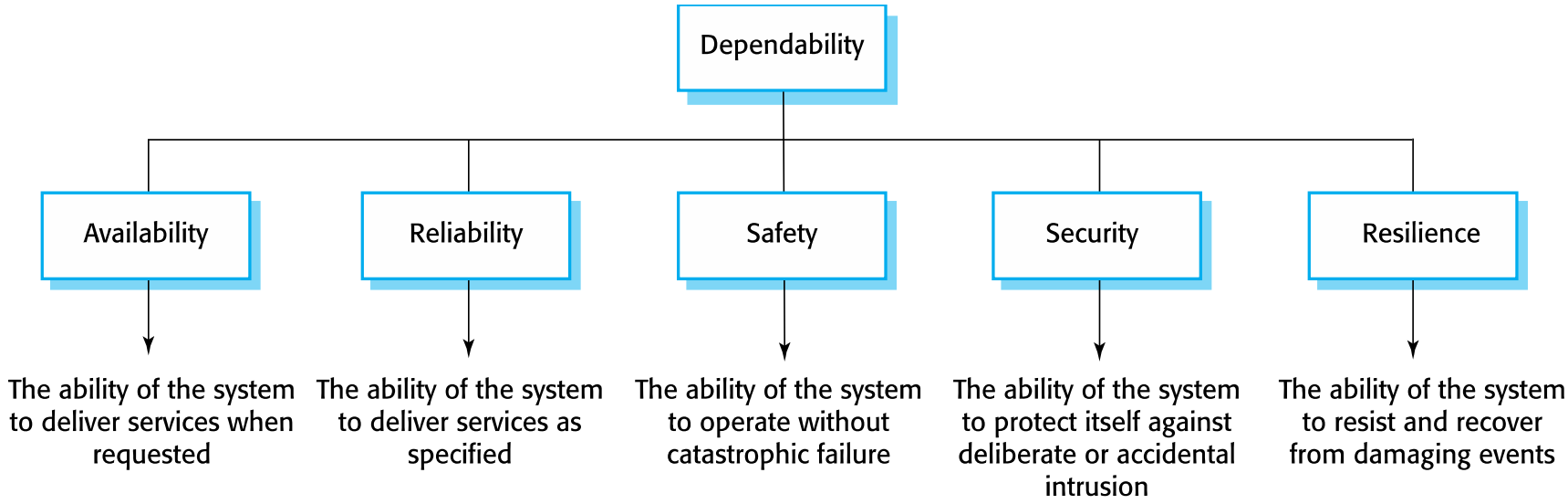
Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependability Properties

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

The Principal Dependability Properties



Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Principal Properties (1 of 2)

- Availability
  - The probability that the system will be up and running and able to deliver useful services to users.
- Reliability
  - The probability that the system will correctly deliver services as expected by users.
- Safety
  - A judgment of how likely it is that the system will cause damage to people or its environment.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Principal Properties (2 of 2)

- Security
  - A judgment of how likely it is that the system can resist accidental or deliberate intrusions.
- Resilience
  - A judgment of how well a system can maintain the continuity of its critical services in the presence of disruptive events such as equipment failure and cyberattacks.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Other Dependability Properties

- Repairability
  - Reflects the extent to which the system can be repaired in the event of a failure
- Maintainability
  - Reflects the extent to which the system can be adapted to new requirements;
- Error tolerance
  - Reflects the extent to which user input errors can be avoided and tolerated.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependability Attribute Dependencies

- Safe system operation depends on the system being available and operating reliably.
- A system may be unreliable because its data has been corrupted by an external attack.
- Denial of service attacks on a system are intended to make it unavailable.
- If a system is infected with a virus, you cannot be confident in its reliability or safety.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependability Achievement (1 of 2)

- Avoid the introduction of accidental errors when developing the system.
- Design V & V processes that are effective in discovering residual errors in the system.
- Design systems to be fault tolerant so that they can continue in operation when faults occur
- Design protection mechanisms that guard against external attacks.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependability Achievement (2 of 2)

- Configure the system correctly for its operating environment.
- Include system capabilities to recognise and resist cyberattacks.
- Include recovery mechanisms to help restore normal system service after a failure.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

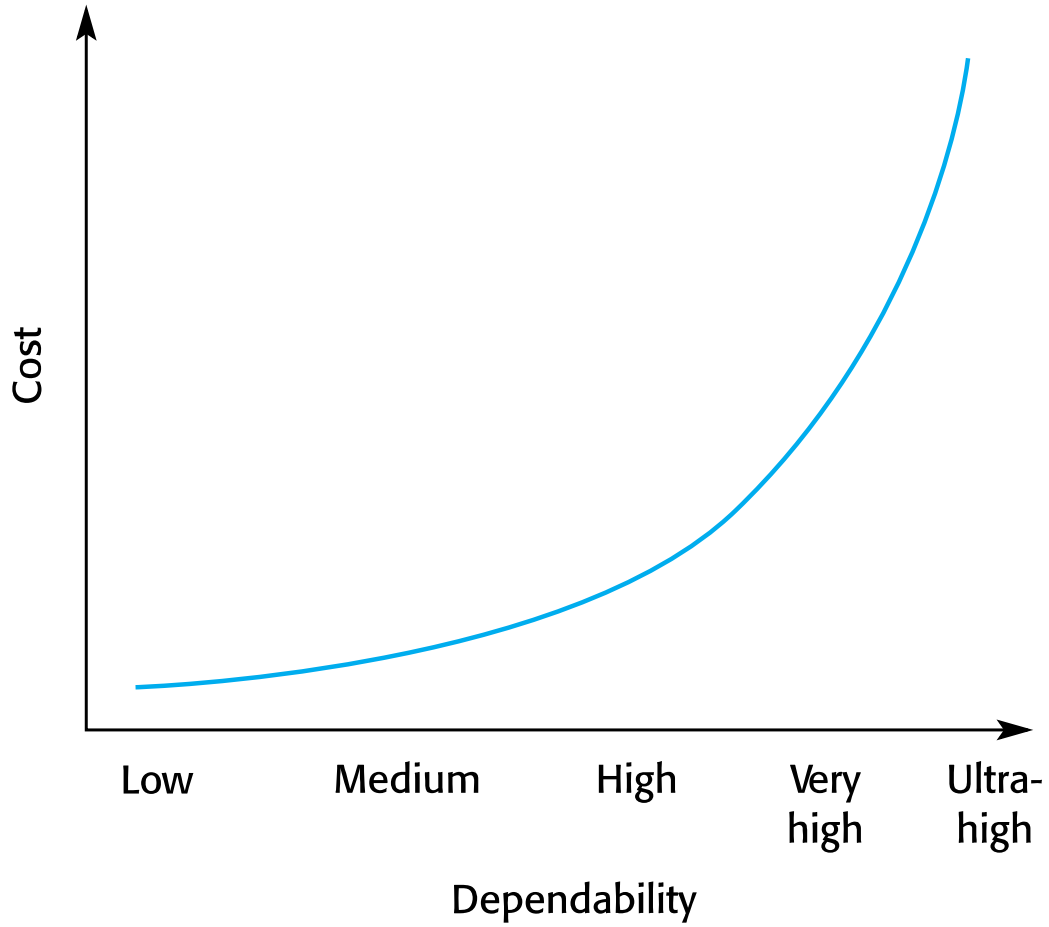
Dependability Costs

- Dependability costs tend to increase exponentially as increasing levels of dependability are required.
- There are two reasons for this
  - The use of more expensive development techniques and hardware that are required to achieve the higher levels of dependability.
  - The increased testing and system validation that is required to convince the system client and regulators that the required levels of dependability have been achieved.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Cost/dependability Curve



Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependability Economics

- Because of very high costs of dependability achievement, it may be more cost effective to accept untrustworthy systems and pay for failure costs
- However, this depends on social and political factors. A reputation for products that can’t be trusted may lose future business
- Depends on system type - for business systems in particular, modest levels of dependability may be adequate

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Redundancy and Diversity

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Redundancy and Diversity

- Redundancy
  - Keep more than a single version of critical components so that if one fails then a backup is available.
- Diversity
  - Provide the same functionality in different ways in different components so that they will not fail in the same way.
- Redundant and diverse components should be independent so that they will not suffer from ‘common-mode’ failures
  - For example, components implemented in different programming languages means that a compiler fault will not affect all of them.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Diversity and Redundancy Examples

- Redundancy. Where availability is critical (e.g. in e-commerce systems), companies normally keep backup servers and switch to these automatically if failure occurs.
- Diversity. To provide resilience against external attacks, different servers may be implemented using different operating systems (e.g. Windows and Linux)

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Process Diversity and Redundancy

- Process activities, such as validation, should not depend on a single approach, such as testing, to validate the system.
- Redundant and diverse process activities are important especially for verification and validation.
- Multiple, different process activities the complement each other and allow for cross-checking help to avoid process errors, which may lead to errors in the software.

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Problems with Redundancy and Diversity

- Adding diversity and redundancy to a system increases the system complexity.
- This can increase the chances of error because of unanticipated interactions and dependencies between the redundant system components.
- Some engineers therefore advocate simplicity and extensive V & V as a more effective route to software dependability.
- Airbus FCS architecture is redundant/diverse; Boeing 777 FCS architecture has no software diversity

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependable Processes

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependable Processes

- To ensure a minimal number of software faults, it is important to have a well-defined, repeatable software process.
- A well-defined repeatable process is one that does not depend entirely on individual skills; rather can be enacted by different people.
- Regulators use information about the process to check if good software engineering practice has been used.
- For fault detection, it is clear that the process activities should include significant effort devoted to verification and validation.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependable Process Characteristics

- Explicitly defined
  - A process that has a defined process model that is used to drive the software production process. Data must be collected during the process that proves that the development team has followed the process as defined in the process model.
- Repeatable
  - A process that does not rely on individual interpretation and judgment. The process can be repeated across projects and with different team members, irrespective of who is involved in the development.

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Attributes of Dependable Processes

Process characteristic	Description
Auditable	The process should be understandable by people apart from process participants, who can check that process standards are being followed and make suggestions for process improvement.
Diverse	The process should include redundant and diverse verification and validation activities.
Documentable	The process should have a defined process model that sets out the activities in the process and the documentation that is to be produced during these activities.
Robust	The process should be able to recover from failures of individual process activities.
Standardized	A comprehensive set of software development standards covering software production and documentation should be available.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependable Process Activities (1 of 2)

- Requirements reviews to check that the requirements are, as far as possible, complete and consistent.
- Requirements management to ensure that changes to the requirements are controlled and that the impact of proposed requirements changes is understood.
- Formal specification, where a mathematical model of the software is created and analyzed.
- System modeling, where the software design is explicitly documented as a set of graphical models, and the links between the requirements and these models are documented.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependable Process Activities (2 of 2)

- Design and program inspections, where the different descriptions of the system are inspected and checked by different people.
- Static analysis, where automated checks are carried out on the source code of the program.
- Test planning and management, where a comprehensive set of system tests is designed.
  - The testing process has to be carefully managed to demonstrate that these tests provide coverage of the system requirements and have been correctly applied in the testing process.

Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependable Processes and Agility (1 of 2)

- Dependable software often requires certification so both process and product documentation has to be produced.
- Up-front requirements analysis is also essential to discover requirements and requirements conflicts that may compromise the safety and security of the system.
- These conflict with the general approach in agile development of co-development of the requirements and the system and minimizing documentation.

Pearson

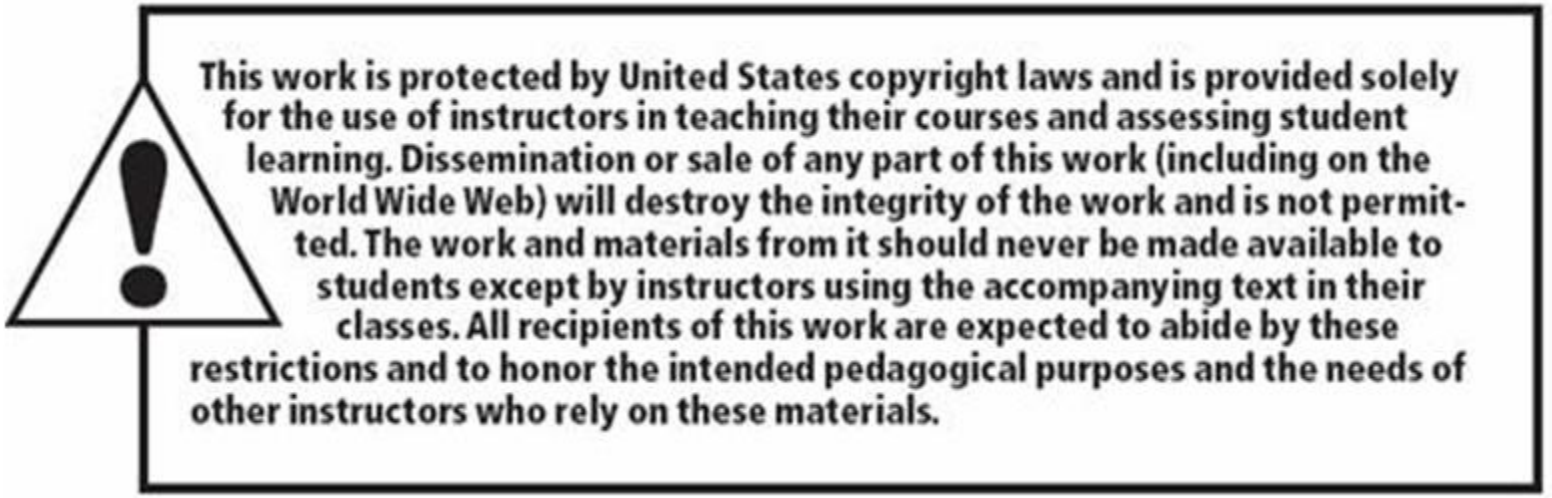
Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Dependable Processes and Agility (2 of 2)

- An agile process may be defined that incorporates techniques such as iterative development, test-first development and user involvement in the development team.
- So long as the team follows that process and documents their actions, agile methods can be used.
- However, additional documentation and planning is essential so ‘pure agile’ is impractical for dependable systems engineering.

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved

Copyright



Pearson

Copyright © 2016, 2011, 2006 Pearson Education, Inc. All Rights Reserved