Co-funded by the Erasmus+ Programme of the European Union

Erasmus+ SkoPS

Intellectual Output 7 - Dependability and Safety

Project Title	Empowering the European Workforce Development through Online/Virtual Skills Training for Digital Transformation toward Mitigating the Impact of Pandemic Situations (SkoPS)		
Project Acronym	SkoPS	**Project Number**	2020-1-DE01-KA226-HE-005772
Date	2022-10-30	**Deliverable No.**	D0.7
Contact Person	Bahareh Kiamanesh Ali Behravan	**Organization**	USI
Phone		**E-Mail**	Bahareh.Kiamanesh@uni-siegen.de Ali.Behravan@uni-siegen.de
Version	1.1	**Confidentiality level**	Public

Version History

Version No.	Date	Change	Editor(s)
0.1	30.11.2021	Initial Version	Ali Behravan
1.1	30.10.2022	Final Version	Bahareh Kiamanesh

Contributors

Name	Organization
Bahareh Kiamanesh	USI
Ali Behravan	USI

Disclaimer

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

Version History 2

Contributors 2

1. Introduction 3

1.1 Abstract 4

1.2 Purpose of the document 4

1.3 Relations to other deliverables 4

2. Template 5

1 Introduction

Course Plan Template can help the partners to prepare the course materials that help us to organize the materials and contents of courses in the SkoPS project in standard form. Some other Forms and templates for the student assessment Plan and Evaluation Questionaries Plan for each Course can be added to this Template.

1.1 Abstract

Dependability is the ability of a system to deliver the intended service, including fault-tolerance techniques. In Real-time computing, dependability refers to service provision at a particular time. Functional safety is concerned with the safety-critical systems and concentrates on the characteristics of the extra systems added to a system with the purpose of making its operation safe. The main concept of the dependability theory is the threats, the attributes, and the techniques used to enforce dependability.

1.2 Purpose of the document

The purpose of this document is to have the same template for all courses. In this template, for each course, some features are considered.

1.3 Relation to other deliverables

This Report is part of the D0.7, which is the general structure of each course.

2 Template

Course Plan Template			
Course ID and Title:	Dependability and Safety		
Course Duration:	2/3 months	**Course ECTS:**	1/1.5
Leading Organization:	University of Siegen (USI)		
Course Media:	Text / Text File / Video / Website/ Internet		
Laboratory (Yes/No)	Yes		

Course Description:

Safety-critical applications require high reliability in computing and electronic systems, which is achieved by designing fault-tolerant systems. This module introduces the design and analysis of reliable computing systems and is represented in 5 weeks. This course presents the fundamentals of fault-tolerant systems and fault-tolerance tools at both hardware and software levels, e.g., redundancy and re-execution. It will give an understanding of how to detect faults, design computing components to tolerate faults, and measure the reliability of systems. This course will help students design and analyze reliable computing systems. This course includes five chapters.

It starts with introductory information about the faults, errors, failures, etc. In Chapter 2, a preliminary description of fault-tolerance techniques is given. Chapter 3 introduces the requirements of dependability. Chapter 4 presents hardware redundancy for fault tolerance and probabilistic tools to measure reliability. Chapter 5 describes fault-tolerance techniques for software systems.

Course Materials and Equipment (Prerequisite)

- This course is suitable for graduate and senior-level undergraduate students.
- Students should have basic knowledge of ICT systems design, analysis, and probability theory.

Teaching and Learning Activities:

- Providing lectures and content
- Using graphs, images, and videos to enhance the understanding of the subject
- Solving different examples in modeling and evaluation to clarify the problem
- Evaluation activities

Course activities:

- Revision and advancement of existing teaching/learning materials
- Type: Open Educational Resource (OER)
- Discussion and interactive activities

Course Objectives:

- Providing the introductory knowledge, terminologies, and concepts applied in the specification, design, evaluation, and principles of fault-tolerant operations
- Designing robust and resilient systems.
- Overviewing the dependability in the field of computing systems.
- Increasing the awareness of the potential impact of failures, dependability, and safety in the industrial environment.

Table of Contents:

1. Introduction of Dependability
1.1 Introduction
1.2 Dependability Concept
1.3 Origins of the Dependability Concepts
1.4 Dependability Tree
1.5 Dependability Threats: Faults
1.6 Dependability Threats: Faults Sources
1.7 Dependability Threats: Causes of Faults
1.8 Dependability Threats: Errors
1.9 Dependability Threats: Types of Errors
1.10 Dependability Threats: Failures
1.11 Dependability Threats: Failures Modes
1.12 Fault-Error-Failure Propagation Model
1.13 Dependability Attributes
1.14 Dependability Means
2. Dependability Means
2.1 Fault Hypothesis
2.2 Fault Classification
2.3 Fault Classification: Transient faults
2.4 Fault Classification: Intermittent faults
2.5 Fault Classification: Permanent faults
2.6 Fault Tolerance
2.7 Fault Tolerant versus High Availability
2.8 Fault Tolerance: Error Processing
2.9 Fault Tolerance: Fault Treatment
2.10 Error Detection
2.11 Error Recovery: Forward Recovery
2.12 Error Recovery: Backward Recovery
2.13 Fault Detection
2.14 Fault Isolation
2.15 Redundancy Techniques
2.15.1 Hardware Redundancy
2.15.2 Activation of Hardware Redundancy
2.15.3 Redundancy Techniques
2.15.4 Duplication with Comparison
2.15.5 Standby Sparing
2.15.6 Pair-and-a-Spare
2.15.7 Hardware versus Software Voters
2.15.8 Redundancy and Fault Tolerance
2.15.9 Software Redundancy
2.15.10 Static Software Redundancy Techniques
2.15.11 Dynamic Software Redundancy Techniques
2.15.12 Information Redundancy

2.15.13 Time Redundancy
2.16 Fault Removal
2.17 Fault Forecasting
3. Dependability Attributes
3.1 Dependability Attributes
3.2 Traditional Measures of Dependability
3.3 Network Measures
3.4 Reliability
3.5 Availability
3.6 Safety
3.7 Maintainability
4. Hardware Fault Tolerance
4.1 Hardware Fault Tolerance
4.2 Hardware Failures
4.3 Hardware Failure Rate and Reliability
4.4 Failure rate and Reliability
4.5 Resilient Structures
4.6 Series Systems
4.6.1 Reliability of Series Systems
4.7 Reliability of Series Systems
4.8 Parallel Systems
4.8.1 Reliability of Parallel Systems
4.9 Non-Series, Parallel systems
4.9.1 Reliability of Non-Series, Parallel Systems
4.10 M-of-N Systems
4.11 Triple Modular Redundant (TMR) structure
4.12 Reliability of TMR
4.13 Resilient Structures
4.14 Voters
4.15 Plurality Voting
4.16 K-Plurality Voting
4.17 Variations on N-Modular Redundancy (NMR)
4.17.1 Unit-Level Modular Redundancy
4.17.2 Dynamic Redundancy
4.17.3 Hybrid Redundancy
4.17.4 Duplex Systems
4.18 Reliability Evaluation Techniques
4.19 Reliability Evaluation Techniques: Poisson Processes
4.20 Reliability Evaluation Techniques
4.21 Reliability Evaluation Techniques: Markov Models
4.21.1 The Markov model for a duplex system with an inactive spare
4.21.2 The Markov model for a duplex system with an inactive
4.21.3 The Markov model for a duplex system with repair
4.22 Fault-Tolerance Processor-Level Techniques
4.22.1 Watchdog Processor
4.22.2 Simultaneous Multithreading for Fault Tolerance
4.23 Byzantine Failures
5. Software Fault Tolerance
5.1 Software Fault Tolerance
5.2 Acceptance Tests
5.2.1 Timing checks
5.2.2 Verification of Output

5.2.3 Range Checks

Laboratory Description and Equipment:

To cover the online format of the module design, remote labs are also designed, which are aligned with the pandemic situations. Bridge the gap from theory to practical implementation by performing a practical experiment in the lab. The practical part depends on the knowledge of practical tasks of the lecture contents (e.g., programming of embedded Systems with microcontrollers, scheduling, memory management, and time analysis). The Embedded Control Lab can be held on by the university of Siegen using an online platform.

Course References:

[1]. Shooman, M.L., 2003. Reliability of computer systems and networks: fault tolerance, analysis, and design. John Wiley & Sons.
[2]. Koren, I. and Krishna, C.M., 2020. Fault-tolerant systems. Morgan Kaufmann.
[3]. Sorin, D.J., 2009. Fault-tolerant computer architecture. Synthesis Lectures on Computer Architecture, 4(1), pp.1-104.
[4]. Avizienis, A., Laprie, J.C. and Randell, B., 2001. Fundamental concepts of dependability. Department of Computing Science Technical Report Series.
[5]. Obermaisser, R. and Peti, P., 2006, June. A fault hypothesis for integrated architectures. In 2006 International Workshop on Intelligent Solutions in Embedded Systems (pp. 1-18). IEEE.

Evaluation and Assessment Methods:

Assessments take various forms, including Quizzes and Multiple-choice questions after each module. The course will be passed by 80% of the quizzes.

The tasks lead to the production of the intellectual output and the applied methodology.

The methodology in creating the SkoPS courses will be as follows: The leading partner is mainly responsible for preparing the materials and will distribute tasks to the other partners based on their experiences and skills. All other partners will review the prepared materials and provide feedback to improve the materials' quality. A systematic review process with all partners involved will further enhance the quality of the courses. Right at the beginning of the project (directly after the kick-off meeting), a requirement analysis will be carried out by all project partners to precise the content of the courses which are outlined in this proposal. A second meeting will be conducted six months after the kick-off meeting as an internal course development Hackathon. Each partner will invite one or two external experts for this Hackathon to get their feedback for improving the quality of the course.