

Undertake real-time software solution development

Overview

This standard covers the competences needed to undertake real-time software solution development. It covers the principles of real-time software solutions design, implementation, testing and deployment, and how the desired functionality and real-time data streaming of an embedded system can be realised through software applications. This includes concurrent programming, communication and synchronisation. This also includes development of real-time software for embedded systems.

In order to do meet this standard, you are required to have the knowledge, skills and understanding necessary to design, implement, test and deploy real-time systems software applications; ensure that your work complies with all legal, statutory, industrial and organisational requirements, and follow relevant industry codes of practice. You will be required to work autonomously and to take responsibility for the development work that you carry out.

Your underpinning knowledge and understanding of the principles of real-time and embedded systems will enable you to undertake programming tasks including for example; scheduling, asynchronous events, real-time threads, asynchronous transfer of control and resource control.

This type of activity is likely to be undertaken by someone whose work role involves delivering real-time software solutions and the development of associated tools and methods. This includes people working as Real-Time Software Developers, Real-Time Programmers, Real-Time Data Engineers, Embedded Systems Software Engineers etc. You will likely work within a team of developers, and in a sector or organisation that works with real-time data, such as real-time financial trading, building control, transport and logistics, manufacturing processes, chemical processing, health monitoring systems or on-line gaming for example.

Performance criteria

You must be able to:

1. Review customer requirements for real-time software applications in order to agree upon the function of the software system
2. Review hardware architecture and develop a software architecture definition to meet customer requirements
3. Design scalable real-time software configurations and solutions to match the real-time operating system and its scheduling mechanism
4. Specify the message interactions between different tasks in the system in line with organisational feature design guidelines
5. Design real-time software modules that deliver the correct communication, synchronisation and processing features required
6. Develop software modules using real-time programming languages, tools and methods, in order to deliver specified functionality
7. Integrate software modules to produce efficient real-time system solutions
8. Test the functionality of real-time software applications to confirm they operate in line with the software requirements specification
9. Deploy real-time software applications into production environments in line with organisational requirements

Knowledge and understanding

You need to know and understand:

1. The design context in which real-time software System Designers operate
2. The software functionality required to achieve real-time operation of a software system
3. The principles, role and limitations of timing analysis in the construction of real-time software systems
4. How to design real-time software that incorporates; communication and synchronisation, representation of timing constraints, and asynchronous processing
5. The industry standard design approaches for modular real-time software development
6. The underlying mathematical and statistical analysis techniques that can be used to undertake timing analysis of real-time programs
7. The different approaches to programming fault tolerance in real-time software systems and their impact on timing analysis
8. The role of scheduling in facilitating timing analysis
9. Schedulability analysis for predicting the response time of tasks in multi-task applications
10. Both fixed priority pre-emptive scheduling (FPPS) and earliest-deadline first (EDF) scheduling approaches for real-time systems
11. Worst-case execution time (WCET) analysis techniques and how to apply them
12. Real-time modelling techniques software and Java concurrency models
13. The features of the real-time specification for Java (RTSJ)
14. Testing techniques for real-time software systems and how to apply them
15. The formal design methods for modelling real-time systems

Undertake real-time software solution development

Developed by ODAG

Version Number 1

Date Approved March 2020

Indicative Review Date March 2023

Validity Current

Status Original

Originating Organisation ODAG Consultants Ltd

Original URN TECIS507401

Relevant Occupations Software Development

Suite IT and Telecoms

Keywords Real time software
