

European Software
Skills Alliance.

Train the Trainer Programme

Annex IV

DevOps expert EQF 6

30 November 2023
Status: Final version

**di
SS
e**

Copyright © 2023 by the European Software Skills Alliance. The project resources contained herein are publicly available under the [Creative Commons license 4.0 B.Y](https://creativecommons.org/licenses/by/4.0/)

Train the Trainer Programme – Annex IV – DevOps expert EQF 6, 2023.

Deliverable 13: “ESSA Train the Trainer Programme & Materials”– Annex IV

This document is a draft version and is subject to change after review coordinated by the European Education and Culture Executive Agency (EACEA).

Authors: Federica D’Armini (Adecco/Mylia)

Editors/Reviewers: Federica D’Armini (Adecco/Mylia), Sabine Boesen – Mariani (Utrecht University of Applied Sciences), Gruden Zala (University of Ljubljana), Soltes Tjasa (University of Ljubljana), Niels Sellings (Digital Europe), Wanda Saabel (Global Knowledge), Ants Slid (BCS Koolitus), Jutta Breyer (Digital Europe), Paul Aertsen (Global Knowledge)softwareskills.eu

Legal Disclaimer

The European Commission’s support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



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

About ESSA

The European Software Skills Alliance (ESSA) is a four-year transnational project funded under the EU’s Erasmus+ programme. It ensures the skills needs of the rapidly evolving Software sector can be met — today and tomorrow.

ESSA provides current and future software professionals, learning providers and organisations with software needs with the educational and training instruments they need to meet the demand for software skills in Europe.

ESSA will develop a European Software Skills Strategy and learning programmes for Europe. It will address skill mismatches and shortages by analysing the sector in depth and delivering future-proof curricula and mobility solutions; tailored to the European software sector’s reality and needs.

Project partners

The ESSA consortium is led by DIGITALEUROPE. It is composed of academic and non-academic partners from the education, training, and software sectors.

View all project partners: [ESSA Partners](#) | [ESSA Associated Partners](#)

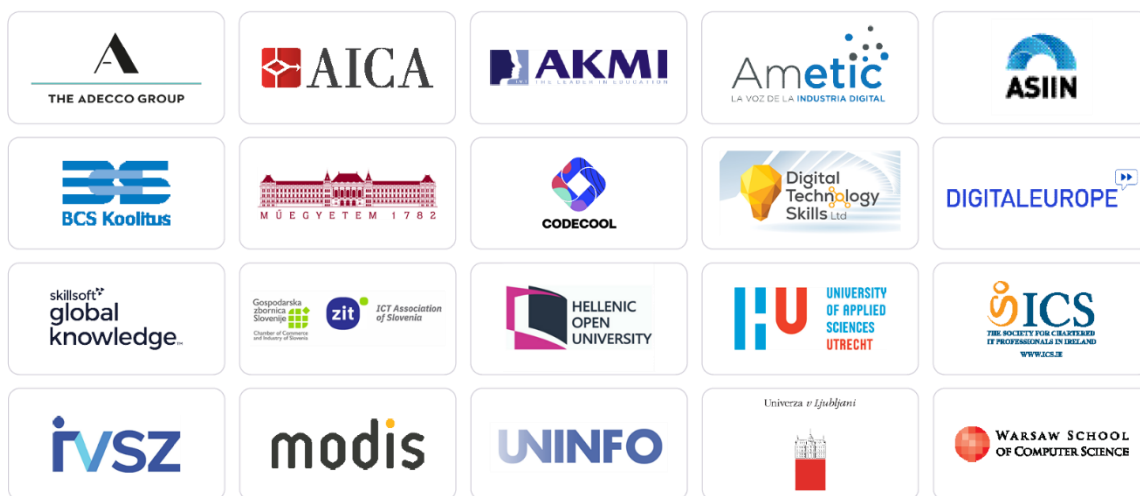


Table of Contents

1. How to use the ESSA Learning programme for DevOps expert EQF 6 profile.....	6
1.1 Introduction	6
1.2 TARGET	6
2. How to deliver the ESSA DevOps expert EQF 6 profile.....	7
2.1 IT-oriented students	7
2.1.1 Overall Information about the Learning Programme.....	7
2.1.2 Learning Programme PLO 1 – Application Development [e-3].....	7
2.1.2.1 Learning Units PLO 1 – Application Development [e-3].....	9
2.1.3 Learning Programme PLO 2 – Component integration [e-3].....	10
2.1.3.1 Learning Units PLO 2 – Component integration [e-3].....	12
2.2 People with ICT knowledge in need of reskilling to become DevOps expert..	13
2.2.1 Overall Information about the Learning Programme.....	13
2.2.2 Learning Programme PLO 1 – Application Development [e-3].....	13
2.2.2.1 Learning Units PLO 1 – Application Development [e-3].....	15
2.2.3 Learning Programme PLO 8 – Soft competences [EQF6].....	15
2.2.3.1 Learning Units PLO 8 – Soft competences [EQF6].....	17
2.3 Students and professionals in need of upskilling/reskilling.....	17
2.3.1 Overall Information about the Learning Programme.....	18
2.3.2 Learning Programme PLO 1 – Application Development [e-3].....	19
2.3.2.1 Learning Units PLO 1 – Application Development [e-3].....	20
2.3.3 Learning Programme PLO 2 – Component integration [e-3].....	21
2.3.3.1 Learning Units PLO 2 – Component integration [e-3].....	22
2.3.4 Learning Programme PLO 3 – Testing [e-3].....	24
2.3.4.1 Learning Units PLO 3 – Testing [e-3].....	25
2.3.5 Learning Programme PLO 4 – Solution Deployment [e-3].....	25
2.3.5.1 Learning Units PLO 4 – Solution Deployment [e-3].....	26
2.3.6 Learning Programme PLO 5 – ICT Systems Engineering [e-3].....	28
2.3.6.1 Learning Units PLO 5 – ICT Systems Engineering [e-3].....	29

List of abbreviations and acronyms

Abbreviation	Term
e-CF, EN 16234-1	European e-Competence Framework, European Norm 16234 - Part 1: Framework
ECTS	European Credit Transfer and Accumulation System
EQF	European Qualifications Framework
ESSA	European Software Skills Alliance
LO	Learning Outcome
PLO	Programme Learning Outcome

1. How to use the ESSA Learning programme for DevOps expert EQF 6 profile

1.1 Introduction

In this Annex trainers, teachers and educators are provided with all information necessary to deliver the ESSA Learning Programmes designed for the DevOps expert EQF 6 Educational profile.

The proposed three learning paths follow a modular and flexible structure based on Programme Learning Outcomes (PLOs). Each PLO includes self-consistent Learning Units (LUs) supported by specific learning materials.

In particular, this document provides:

- overall information for Learning Programme - Objective, Total number of Programme Learning Outcomes (PLOs) concerned, Total Learning Units (LUs), Overall duration (hours); Total number of ECTS; Targeted Institutions (learning providers);
- detailed Learning Programme including the Learning Units for each Programme Learning Outcome (PLO).

In this regard, more specifically, the following is provided for each Programme Learning Outcome (PLO):

- overall information (N. of Learning Units, Duration in hours, Total number of ECTS, Recommendations for Micro-credentials, possible integration with studies related to other PLOs, Recommended Didactical Approach, Recommended Delivery methods, etc);
- detailed information for each Learning Unit (Title, Duration in hours, Didactical Approach and delivery method, type of Assessment, Title of the related Learning material proposed, Link to access to the learning material - ESSA Platform).

This Annex is strictly related to the document “Train the Trainer Programme. DELIVERABLE 13 – ESSA Train the Trainer Programme & Materials”.

As a further support, it is advised to consult the documents indicated in the paragraph “Sources of reference” of the Deliverable 13 above mentioned, through the available links.

Learning materials developed to support the delivery of the ESSA Learning Programmes for this Educational Profile are available on the ESSA platform at the following link: <https://learn.softwareskills.eu/>.

1.2 TARGET

The following three Learning Programmes address three type of target groups:

- **IT-oriented students**

- People with ICT knowledge in need of reskilling to become DevOps expert
- Students and professionals in need of upskilling/reskilling

2. How to deliver the ESSA DevOps expert EQF 6 profile

2.1 IT-oriented students

2.1.1 Overall Information about the Learning Programme

Objective	<i>The programme aims to train university students to become an ICT specialist. Students learn smart and devise creative ICT solutions for business issues. The students don't get lessons nor exams, but work fully on challenging projects for real clients from the start of the studies – it is therefore a practice based learning approach. The students are coached in their learning, both on skills and competencies. The learning outcomes of the Open-ICT training program are based on the HBO-I professional tasks (elaborated by the HBO-I Foundation). This foundation is a partnership between the universities of applied sciences in the Netherlands that provide ICT education and the business community. The curriculum leverages a blended learning model, combining the presence classroom and virtual classroom. HBO Open-ICT lasts 8 semesters and the curriculum here present is addressing first and second year students.</i>
Total number of PLOs concerned	2
Total Learning Units	9
Duration	13,5 hours
Total number of ECTS	starting from 1 ECTS
Targeted Institutions	Higher Education Institutions

2.1.2 Learning Programme PLO 1 – Application Development [e-3]

Overall information PLO 1 – Application Development [e-3]	
N. of Learning Units	5
Learning Outcomes	<ul style="list-style-type: none"> - Organises data and creates a structured dataset - Writes code and related documentation to it, using programming languages (e.g., Java, Javascript, PHP, Python) and tools (e.g., GitHub), applying programming principles (e.g., clean coding, green coding, secure programming) and other relevant practices (e.g. continuous integration, test-driven development), principles or constraints (e.g., privacy legislation, intellectual property law) - Efficiently creates a working software component/ application taking into account design requirements (e.g., scalability, availability) and other relevant constraints (e.g., architecture, efficiency, cost, quality, energy consumption) and applying relevant tools and techniques (e.g., object-oriented programming; IDE, CASE; editors, compilers; version

	<p><i>control management and tools; multimedia integration tools; app development tools; reuse of proved solutions)</i></p> <ul style="list-style-type: none"> - <i>Modifies an existing software component/ application, in order to optimise it (e.g., to improve scalability, maintenance, performance, security)</i> 				
Duration	-				
Total number of ECTS	<i>starting from 1 ECTS</i>				
Recommendations for Micro-credentials	<i>This PLO is currently deployed in a 4 year bachelor programme and delivered for students in the second year.</i>				
Often integrated with studies of PLO	PLO 2				
Recommended Didactical Approach	Presence Classroom Work placement				
Additional comments	-				
Recommended Delivery methods	<table> <tr> <td><i>Lecture</i></td> <td>20%</td> </tr> <tr> <td><i>Case study. Individual/team project</i></td> <td>80%</td> </tr> </table>	<i>Lecture</i>	20%	<i>Case study. Individual/team project</i>	80%
<i>Lecture</i>	20%				
<i>Case study. Individual/team project</i>	80%				
Additional comments	-				
Work Based Learning Task (If foreseen) and Follow-up, learning reinforcement	<p><i>Open-ICT training program are based on the HBO-I professional tasks (elaborated by the HBO-I Foundation). This foundation is a partnership between the universities of applied sciences in the Netherlands that provide ICT education and the business community.</i></p> <p><i>Open-ICT is characterized by agile project-driven education. Students therefore always work on real projects for our clients. Agile stands for short cyclical. Every two weeks the team thinks about what will be made and each student in the team looks at what he or she needs to learn for this. During the two weeks, making and learning alternate and at the end of each two weeks the work is delivered and you receive feedback on your work and your learning ability. Through this form of education, you learn new general and ICT skills every two weeks and deliver real products every two weeks. With this working method we are 100% in line with how a company works and learns later. The materials are supporting the students learning.</i></p>				
Important (new) approaches and technologies to consider	<p><i>Open ICT is based on new approaches to education, based on intrinsic motivation. The intrinsic motivation is maximal when students are allowed to make their own choices: autonomy, when students feel included in a learning community: connectedness, and when they develop self-confidence by learning in challenging tasks: feeling competent. Every semester the student chooses a professional role they want to deepen in line with the HBO-I professional tasks. In a development team, together with the client, they determine what they will make.</i></p>				

	<i>Students work incorporating ways of working implemented in companies such as agile methods. The work and learning process of Open-ICT comes from the agile method of the software development industry, called SCRUM. Every two weeks, students think about what they are going to create as a team, by user stories. They will think of the necessary tasks for their own contribution within the team and what they have to learn in order to be able to perform a certain task (learning stories). By dividing this into 'sprints' and properly guiding students, they can achieve learning objectives every two weeks and deliver working products. These quick results boost confidence and motivation.</i>
Training facilities (Link to ESSA learning material Platform)	https://learn.softwareskills.eu/

2.1.1.1 Learning Units PLO 1 – Application Development [e-3]

LU1	CISQ - 1 Introduction: Continuous Integration & Software Quality
Duration	<i>1 hour and 30 minutes</i>
Didactical Approach and delivery method	<i>Lecture and practical exercises</i>
Additional information	<i>Live classes</i>
Assessment	<i>Practical assessment & Portfolio</i>
Title of the Learning material	<i>Introduction: Continuous Integration & Software Quality</i>

LU2	CISQ - 3 Test principles and patterns
Duration	<i>1 hour and 30 minutes</i>
Didactical Approach and delivery method	<i>Lecture and practical exercises</i>
	<i>Live classes</i>
Additional information	<i>Practical assessment & Portfolio</i>
Assessment	<i>Test principals and patterns</i>

LU3	CISQ - 4 Code coverage
Duration	<i>1 hour and 30 minutes</i>
Didactical Approach and delivery method	<i>Lecture and practical exercises</i>
Additional information	<i>Live classes</i>
Assessment	<i>Practical assessment & Portfolio</i>
Title of the Learning material	<i>Coverage and mutation testing</i>

LU4	CISQ - 7 Security
Duration	<i>1 hour and 30 minutes</i>
Didactical Approach and delivery method	<i>Lecture and practical exercises</i>
Additional information	<i>Live classes</i>
Assessment	<i>Practical assessment & Portfolio</i>
Title of the Learning material	<i>Security</i>

LU5	CISQ - 8 Bonus
Duration	<i>1 hour and 30 minutes</i>
Didactical Approach and delivery method	<i>Lecture and practical exercises</i>
Additional information	<i>Live classes</i>
Assessment	<i>Practical assessment & Portfolio</i>
Title of the Learning material	<i>Bonus-lecture on Continuous Integration & Software Quality</i>

2.2.2 Learning Programme PLO 2 – Component integration [e-3]

Overall information PLO 2 - Component Integration [e-3]	
N. of Learning Units	4
Learning Outcomes	<ul style="list-style-type: none"> - <i>Integrates a solution, software application or component applying relevant practices, methods, techniques and tools, compliant with appropriate standards and procedures (e.g. configuration management, version management, change control, packaging and distribution, virtualisation, containerisation)</i> - <i>Monitors, verifies and tests system capacity and performance, using appropriate techniques and tools</i> - <i>Writes an installation report/ installation documentation</i>
Duration	-
Total number of ECTS	<i>starting from 1 ECTS</i>
Recommendations for Micro-credentials	<i>This PLO is currently deployed in a 4-year bachelor programme and delivered for students in the second year.</i>
Often integrated with studies of PLO	PLO 1

Recommended Didactical Approach	Presence Classroom Work placement
Additional comments	-
Recommended Delivery methods	Lecture 20% Case study. Individual/team project 80%
Additional comments	-
Work Based Learning Task (If foreseen) and Follow-up, learning reinforcement	<p><i>Open-ICT training program are based on the HBO-I professional tasks (elaborated by the HBO-I Foundation). This foundation is a partnership between the universities of applied sciences in the Netherlands that provide ICT education and the business community.</i></p> <p><i>Open-ICT is characterized by agile project-driven education. Students therefore always work on real projects for our clients. Agile stands for short cyclical. Every two weeks the team thinks about what will be made and each student in the team looks at what he or she needs to learn for this. During the two weeks, making and learning alternate and at the end of each two weeks the work is delivered and you receive feedback on your work and your learning ability. Through this form of education, you learn new general and ICT skills every two weeks and deliver real products every two weeks. With this working method we are 100% in line with how a company works and learns later. The materials are supporting the students learning</i></p>
Important (new) approaches and technologies to consider	<p><i>Open ICT is based on new approaches to education, based on intrinsic motivation. The intrinsic motivation is maximal when students are allowed to make their own choices: autonomy, when students feel included in a learning community: connectedness, and when they develop self-confidence by learning in challenging tasks: feeling competent. Every semester the student chooses a professional role they want to deepen in line with the HBO-I professional tasks. In a development team, together with the client, they determine what they will make.</i></p> <p><i>Students work incorporating ways of working implemented in companies such as agile methods. The work and learning process of Open-ICT comes from the agile method of the software development industry, called SCRUM. Every two weeks, students think about what they are going to create as a team, by user stories. They will think of the necessary tasks for their own contribution within the team and what they have to learn in order to be able to perform a certain task (learning stories). By dividing this into 'sprints' and properly guiding students, they can achieve learning objectives every two weeks and deliver working products. These quick results boosts confidence and motivation.</i></p>
Training facilities (Link to ESSA learning material Platform)	https://learn.softwareskills.eu/

2.2.2.1 Learning Units PLO 2 – Component integration [e-3]

LU1	CISQ - 2 Test automation
Duration	<i>1 hour and 30 minutes</i>
Didactical Approach and delivery method	<i>Lecture and practical exercises</i>
Additional information	<i>Live classes</i>
Assessment	<i>Practical assessment & Portfolio</i>
Title of the Learning material	<i>Test automation</i>

LU2	CISQ - 3 Test principles and patterns
Duration	<i>1 hour and 30 minutes</i>
Didactical Approach and delivery method	<i>Lecture and practical exercises</i>
Additional information	<i>Live classes</i>
Assessment	<i>Practical assessment & Portfolio</i>
Title of the Learning material	<i>Test principles and patterns</i>

LU3	CISQ - 5 Testing with collaborators
Duration	<i>1 hour and 30 minutes</i>
Didactical Approach and delivery method	<i>Lecture and practical exercises</i>
Additional information	<i>Live classes</i>
Assessment	<i>Practical assessment & Portfolio</i>
Title of the Learning material	<i>Testing with collaborators</i>

LU4	CISQ - 6 Structure and test approach
Duration	<i>1 hour and 30 minutes</i>
Didactical Approach and delivery method	<i>Lecture and practical exercises</i>
Additional information	<i>Live classes</i>
Assessment	<i>Practical assessment & Portfolio</i>

Title of the Learning material	<i>Structure and test approach</i>
---------------------------------------	------------------------------------

2.3 People with ICT knowledge in need of reskilling to become DevOps expert

2.3.1. Overall Information about the Learning Programme

Objective	<i>The programme aims to reskill people to become a DevOps expert at level EQF6. The programme target group are people with some ICT knowledge that want to reskill themselves quickly. The programme is well suited to be offered by VET and training institutes, both in more traditional settings, and also, for example, in more time- and location-independent settings, for example, in the case of distance learning institutes. As the programme focuses on a rapid reskilling and short route to the labour market, cooperation can be sought with labour market intermediaries.</i>
Total number of PLOs concerned	2
Total Learning Units	2
Duration	168 hours
Total number of ECTS	Starting from 6 ECTS
Targeted Institutions	VET and training institutes and Higher Education Institutions

2.3.2 Learning Programme PLO 1 – Application Development [e-3]

Overall information PLO 1 – Application Development [e-3]	
N. of Learning Units	1
Learning Outcomes	<ul style="list-style-type: none"> - Organises data and creates a structured dataset - Writes code and related documentation to it, using programming languages (e.g., Java, Javascript, PHP, Python) and tools (e.g., GitHub), applying programming principles (e.g., clean coding, green coding, secure programming) and other relevant practices (e.g. continuous integration, test-driven development), principles or constraints (e.g., privacy legislation, intellectual property law) - Efficiently creates a working software component/ application taking into account design requirements (e.g., scalability, availability) and other relevant constraints (e.g., architecture, efficiency, cost, quality, energy consumption) and applying relevant tools and techniques (e.g., object-oriented programming; IDE, CASE; editors, compilers; version control management and tools; multimedia integration tools; app development tools; reuse of proved solutions)

	<ul style="list-style-type: none"> - <i>Modifies an existing software component/ application, in order to optimise it (e.g., to improve scalability, maintenance, performance, security)</i>
Duration	<i>152 hours</i>
Total number of ECTS	<i>starting from 5 ECTS as an absolute minimum. If bigger and more complex practical assignments are used such as groupwork and a team project, then 7 ECTS is a minimum.</i>
Recommendations for Micro-credentials	<p><i>This PLO and its subsequent parts can be offered as a micro-credential as part of a modular (re)skilling programme for learners with no prior knowledge of application development. This PLO is also recommended as an independent stand-alone micro-credential for skilling and reskilling (ICT) professionals that are interested in learning the basics of application development. Parts of this PLO are also good candidates for micro-credentials, such as:</i></p> <ul style="list-style-type: none"> • <i>Object design</i> • <i>SQL basics</i> • <i>Python programming</i> • <i>Software & hardware architecture</i> • <i>Testing introduction</i> • <i>DevSecOps</i>
Often integrated with studies of PLO	<i>PLO 2. Component Integration, PLO 3. Testing, PLO 4. Solution Deployment and PLO 5. ICT Systems Engineering and if there are practical assignments, groupwork and a team project involved also PLO's: 7. Profession related competences, 8. Soft competences and 9. Functioning in organisations may be involved.</i>
Recommended Didactical Approach	<p><i>F2F classroom</i> <i>Virtual classroom</i> <i>Blended</i> <i>e-Learning</i> <i>In-company</i></p>
Additional comments	<i>To maximise accessibility and flexibility it is recommended that different didactical approaches are used as much as possible, so that the individual learner can decide what suits best. Besides this, offering in-company courses and training supports accessibility and flexibility.</i>
Recommended Delivery methods	<p><i>Lectures F2F</i> <i>Lectures virtual</i> <i>Lectures blended</i> <i>Virtual instructor-led training (VILT)</i> <i>Practical exercises</i> <i>Group/ teamwork</i> <i>Team project</i></p>
Additional comments	<i>Lectures, e-learning and virtual instructor-led training are recommended for learning the basic principles, terminology, and methods of software design. These should be reinforced through practical tasks, individual and group assignments, and if possible, a team project. The ratio between on the one hand lectures/ VILT and on the other hand practical work done by the learners should be appr. 60% - 40% respectively.</i>

Work Based Learning Task (If foreseen) and Follow-up, learning reinforcement	<p>After learning the basic principles, terminology, and models of application development, the programme should focus on analysing and simulating real work-life-like tasks as, for example:</p> <ul style="list-style-type: none"> • Practical exercises, based on real life situations, e.g., case studies
Important (new) approaches and technologies to consider	n/a
Training facilities (Link to ESSA learning material Platform)	https://learn.softwareskills.eu/

2.3.2.1 Learning Units PLO 1 – Application Development [e-3]

LU1	DevOps Fundamentals: -Object design -SQL basics - Python Programming -Software & hardware architecture - Introduction to testing - DevSecOps
Duration	152 hours (6,1 ECTS)
Didactical Approach and delivery method	Training lectures, Virtual instructor-led training (VILT), Practical exercises
Additional information	F2F, virtual, blended, e-learning/ videos
Assessment	Practical exercises, Exam
Title of the Learning material	Course materials, workbook; “Object design and UML”, “Object design exercise - Mini Bank”, “Object design exercise solution - Mini Bank”

2.3.3 Learning Programme PLO 8 – Soft competences [EQF6]

Overall information PLO 8 – Soft competences [EQF6]	
N. of Learning Units	1
Learning Outcomes	<ul style="list-style-type: none"> - Manages teamwork processes and facilitates collaboration to reach common objectives, e.g., handles conflicts, negotiates, motivates, and persuades. - Communicates with peers, colleagues, supervisors and or relevant others, specialists and non-specialists, and clients, appropriately to the scientific and professional community, using conventions which are relevant. Applies communication to the objective and the target group. - Masters the English language at level B2. Can understand the main ideas of complex text on both concrete and abstract topics, including technical discussions in his/her field of specialisation - Related to the occupation and knowledge domain, critically collects detailed professional and/or scientific information on a limited range of basic theories, principles and concepts, as well as limited information on some important current issues and topics. Analyses, evaluates, and

	<p><i>combines critically this information, knowledge and insights and presents this. Critically applies/ translates/ interprets results of research (possibly executed by others) to the own context (the occupation and/or knowledge domain). Executes applied and practice-oriented research.</i></p> <ul style="list-style-type: none"> - <i>Identifies and analyses complex and unpredictable problems. Solves these problems in a tactical, strategic and creative way by selecting and using data and by using one's creativity, flexibility and inventiveness.</i> - <i>Exercises self-management in complex technical or professional activities or projects, taking responsibility for decision making in unpredictable work or study contexts. Is able to cope with change (positive or negative) and to adapt to a considerable level of variety in the workplace. Handles pressure and setbacks and maintains composure. Shows initiative, creativity and some originality and carries responsibility for the results of own activities, work and or study and for the work results of others. Works correctly and carefully, fully aware of the importance of trustworthiness and accountability.</i> - <i>Realises learning and personal development on one's own initiative, by reflecting on and evaluating personal (learning) results. Selects and uses training/instructional methods and procedures appropriate for the situation when learning.</i>
Duration	16 hours
Total number of ECTS	starting from n.0,5 ECTS
Recommendations for Micro-credentials	<p><i>Due to its strong intertwining with other PLOs, it may be difficult to offer this PLO in its entirety as a stand-alone micro-credential. However, this PLO can be given explicit attention in combination with other PLOs such as mentioned above. Any more theoretical parts of this PLO could be offered separately as micro-credential for skilling (ICT) professionals, such as:</i></p> <ul style="list-style-type: none"> • <i>Time management</i> • <i>Creativity development</i>
Often integrated with studies of PLO	<p><i>This PLO addresses competences that relate to more generic aspects in work related contexts. It is often integrated with: PLO 9. Functioning in organisations. The problem solving and critical analysis aspects of this PLO are often intertwined with PLO's 1. Application development, PLO 5. Systems Engineering and PLO 6. Change Support.</i></p>
Recommended Didactical Approach	<p>F2F classroom Virtual classroom Blended e-Learning In-company</p>
Additional comments	<p><i>To maximise accessibility and flexibility it is recommended that different didactical approaches are used as much as possible, so that the individual learner can decide what suits best. Besides this, offering in-company courses and training supports accessibility and flexibility.</i></p>
Recommended Delivery methods	<p><i>Lectures virtual Virtual instructor-led training (VILT) Practical exercises</i></p>

	<i>Group/teamwork Team project</i>
Additional comments	Lectures, e-learning and virtual instructor-led training are recommended for learning the basic principles, terminology, and methods related to soft competences. These should be reinforced through practical tasks, individual and group assignments, and if possible, a team project. The ratio between on the one hand lectures/ VILT and on the other hand practical work done by the learners should be appr. 20% (or less) - 80% (or more) respectively.
Work Based Learning Task (If foreseen) and Follow-up, learning reinforcement	<i>After learning the basic principles of the different soft competences, the programme should focus on real work-life-like tasks as, for example:</i> <ul style="list-style-type: none"> · <i>Practical exercises, based on real life situations, e.g., case studies</i> · <i>Working together in a group or a team to analyse and solve specific problems</i>
Important (new) approaches and technologies to consider	<i>n/a</i>
Training facilities (Link to ESSA learning material Platform)	<i>https://learn.softwareskills.eu/</i>

2.3.3.1 Learning Units PLO 8 – Soft competences [EQF6]

LU1	Get trained and informed: - Managing time and priorities -Developing creativity
Duration	<i>16 Hours (0,64 ECTS)</i>
Didactical Approach and delivery method	<i>Virtual instructor-led training (VILT), Training lecture, practical exercises, work in groups</i>
Additional information	<i>F2F, virtual, blended, e-learning/ videos</i>
Assessment	<i>Practical exercises</i>
Title of the Learning material	<i>Course materials:</i> <ul style="list-style-type: none"> - <i>Creative thinking for professional efficiency - Trainee booklet</i> - <i>Creative thinking for professional efficiency – Exercises</i> - <i>Manage your time and priorities - Trainee booklet</i> - <i>Manage your time and priorities - Exercises</i>

2.4 Students and professionals in need of upskilling/reskilling

2.4.1 Overall Information about the Learning Programme

<p>Objective</p>	<p><i>The programme aims to train students and professionals in need of upskilling/reskilling. It addresses the domain of software engineering as a key area where up-skilling would be necessary on domestic grounds. The targeted professionals are those that are already engaged in the software engineering industry and need to upgrade their software engineering skills, or in any other technical industry sector where professionals would need to acquire new software engineering skills that are currently needed due to business processes modernization.</i></p> <p><i>The DevOps Expert programme is designed to provide advanced knowledge in the field of DevOps, involving a critical understanding of important theories and principles, The courses are designed to develop advanced skills, demonstrating mastery and innovation, required to solve complex and unpredictable problems in the DevOps field, All attendees are expected to autonomously manage complex DevOps projects, taking responsibility for decision-making in unpredictable software engineering contexts; This study programme will enable the participants to responsibly apply the DevOps culture and practice within groups of developers.</i></p>
<p>Total number of PLOs concerned</p>	<p>5</p>
<p>Total Learning Units</p>	<p>15</p>
<p>Duration</p>	<p>-</p>
<p>Total number of ECTS</p>	<p>Starting from 10 ECTS</p>
<p>Targeted Institutions</p>	<p>Training providers and Higher Education Institutions</p>

2.4.2 Learning Programme PLO 1 – Application Development [e-3]

Overall information PLO 1 – Application Development [e-3]	
N. of Learning Units	2
Learning Outcomes	<ul style="list-style-type: none"> - Organises data and creates a structured dataset - Writes code and related documentation to it, using programming languages (e.g., Java, Javascript, PHP, Python) and tools (e.g., GitHub), applying programming principles (e.g., clean coding, green coding, secure programming) and other relevant practices (e.g. continuous integration, test-driven development), principles or constraints (e.g., privacy legislation, intellectual property law) - Efficiently creates a working software component/ application taking into account design requirements (e.g., scalability, availability) and other relevant constraints (e.g., architecture, efficiency, cost, quality, energy consumption) and applying relevant tools and techniques (e.g., object-oriented programming; IDE, CASE; editors, compilers; version control management and tools; multimedia integration tools; app development tools; reuse of proved solutions) - Modifies an existing software component/ application, in order to optimise it (e.g., to improve scalability, maintenance, performance, security)
Duration	-
Total number of ECTS	starting from 2 ECTS in order to offer a balanced programme that covers all the 10 PLOs.
Recommendations for Micro-credentials	<p>UL will provide qualitative microcredentials that represent the key competences acquired by the student within this PLO, subject to further refinement and agreement at project level. Examples include:</p> <p>“Writes code and related documentation to it, using programming languages (e.g., Java, Javascript, PHP, Python) and tools (e.g., GitHub).”</p> <p>“Applies programming principles (e.g., clean coding, green coding, secure programming) and other relevant practices (e.g. continuous integration, test-driven development), principles or constraints (e.g., privacy legislation, intellectual property law).”</p> <p>And similar to all other competences provided by this PLO.</p>
Often integrated with studies of PLO	PLO 2, 5, 9
Recommended Didactical Approach	Presence Classroom Virtual Classroom Blended e-learning Work placement
Additional comments	-

Recommended Delivery methods	<i>Lecture</i> 50% <i>Case study. Individual/team project (WBL)</i> 50%
Additional comments	Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects.
Work Based Learning Task (If foreseen) and Follow-up, learning reinforcement	<i>After learning the basic principles, terminology, and models of software design, the study should focus on analysing and simulating real work-life-like tasks as, for example:</i> <ul style="list-style-type: none"> • <i>Designing simple design models & diagrams for a real-life-like customer project</i> • <i>Internship tasks of interpreting designs for applications</i>
Important (new) approaches and technologies to consider	<ul style="list-style-type: none"> • <i>Understanding the basics of application lifecycle</i> • <i>Getting to know the process of application development</i> • <i>Getting to know tools and applications used in the process of application development</i>
Training facilities (Link to ESSA learning material Platform)	https://learn.softwareskills.eu/

2.4.2.1 Learning Units PLO 1 – Application Development [e-3]

LU1	Web Development: Backend
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, problem-based learning, collaborative-based learning.</i>
Assessment	<i>Exam, Practical assignment and Presentation of Practical assignment</i>
Title of the Learning material	<i>Node, MVC, Other environments Rest API User Authentication</i>

LU2	Web Development: Frontend
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>

Additional information	<i>Lectures, problem-based learning, collaborative-based learning.</i>
Assessment	<i>Practical assessment</i>
Title of the Learning material	<i>Angular SPA Designing with Swagger and OpenAPI Type script and Angular User Authentication</i>

2.4.3 Learning Programme PLO 2 – Component integration [e-3]

Overall information PLO 2 - Component Integration [e-3]	
N. of Learning Units	5
Learning Outcomes	<ul style="list-style-type: none"> - <i>Integrates a solution, software application or component applying relevant practices, methods, techniques and tools, compliant with appropriate standards and procedures (e.g. configuration management, version management, change control, packaging and distribution, virtualisation, containerisation)</i> - <i>Monitors, verifies and tests system capacity and performance, using appropriate techniques and tools</i> - <i>Writes an installation report/ installation documentation</i>
Duration	-
Total number of ECTS	<i>starting from 2 ECTS</i>
Recommendations for Micro-credentials	<i>This PLO should be an integral part of the initial studies for students with no prior knowledge of software development. Recommended as an independent micro-credential for upskilling junior developers and junior DevOps (EQF 4, 5). Microcredentials will be assigned in qualitative form according to the competences acquired.</i>
Often integrated with studies of PLO	PLO's 4, 5, 8, 9, 10
Recommended Didactical Approach	Presence Classroom Virtual Classroom Blended e-learning Work placement
Additional comments	-
Recommended Delivery methods	<i>Lecture</i> 50% <i>Case study. Individual/team project</i> 50%
Additional comments	Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects in the same

	amount (time and ECTS-wise) as theoretical work. Lectures should be supported by laboratory sessions.
Work Based Learning Task (If foreseen) and Follow-up, learning reinforcement	<p><i>After learning the basic principles, terminology, and models of components integration, the study should focus on analysing and simulating real work-life-like tasks as, for example:</i></p> <ul style="list-style-type: none"> • <i>Each participant chooses a custom domain (from work or life) and create an application that can later be used in real life</i> • <i>Internship tasks of interpreting designs for applications</i>
Important (new) approaches and technologies to consider	<ul style="list-style-type: none"> • <i>Upgrade of custom domain project</i> • <i>Learning about modern approaches and demand on the labour market</i> • <i>Learning about different tools available</i>
Training facilities (Link to ESSA learning material Platform)	https://learn.softwareskills.eu/

2.4.3.1 Learning Units PLO 2 – Component integration [e-3]

LU1	Web development: Frontend
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, laboratory sessions, problem-based learning, collaborative-based learning.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Angular SPA Designing with Swagger and OpenAPI Type script and Angular User Authentication</i>

LU2	Web development: Backend
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>

Additional information	<i>Lectures, laboratory sessions, problem-based learning, collaborative-based learning.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Node, MVC, Other environments Rest API User authentication</i>

LU3	Virtualization
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, laboratory sessions, problem-based learning, collaborative-based learning.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Modern Infrastructures - Virtualization</i>

LU4	Orchestration
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, laboratory sessions, problem-based learning, collaborative-based learning.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Modern Infrastructures - Orchestration</i>

LU5	Containerization
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, laboratory sessions, problem-based learning, collaborative-based learning.</i>
Assessment	<i>Exam,</i>

	<i>Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Modern Infrastructures - Containers Modern Infrastructures - Kubernetes</i>

2.4.4 Learning Programme PLO 3 – Testing [e-3]

Overall information PLO 3 - Testing [e-3]					
N. of Learning Units	1				
Learning Outcomes	<ul style="list-style-type: none"> - <i>Writes (parts of) test related documentation (e.g. test plan, test strategy/approach, test case, test script, test scenario, test conditions)</i> - <i>Constructs tests, by selecting appropriate test methods, techniques, and tools (e.g. test automation tools)</i> - <i>Defines and configures a test environment</i> - <i>Executes test cases and performs test activities related to different sorts of tests</i> - <i>Records and interprets test outcomes and writes test result documentation/ test report</i> 				
Duration	-				
Total number of ECTS	<i>starting from 2 ECTS</i>				
Recommendations for Micro-credentials	<i>Recommended as an independent micro-credential for upskilling junior developers and junior DevOps (for example EQF 4 and 5).</i>				
Often integrated with studies of PLO	PLO's 3, 9, 10				
Recommended Didactical Approach	Presence Classroom Virtual Classroom Blended e-learning Work placement				
Additional comments	-				
Recommended Delivery methods	<table style="width: 100%; border: none;"> <tr> <td style="width: 60%;"><i>Lecture</i></td> <td style="text-align: right;"><i>50%</i></td> </tr> <tr> <td><i>Case study. Individual/team project</i></td> <td style="text-align: right;"><i>50%</i></td> </tr> </table>	<i>Lecture</i>	<i>50%</i>	<i>Case study. Individual/team project</i>	<i>50%</i>
<i>Lecture</i>	<i>50%</i>				
<i>Case study. Individual/team project</i>	<i>50%</i>				
Additional comments	Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects in the same amount (time and ECTS-wise) as theoretical work. Lectures should be supported by laboratory sessions.				
Work Based Learning Task (If foreseen) and Follow-up, learning reinforcement	<i>After learning the basic principles, terminology, and models of software design, the study should focus on analysing and simulating real work-life-like tasks as, for example:</i>				

	<ul style="list-style-type: none"> • Each participant chooses a custom domain (from work or life) and create an application that can later be used in real life • Internship tasks of interpreting designs for applications
Important (new) approaches and technologies to consider	<ul style="list-style-type: none"> • Creating a test plan • Understanding the importance of testing • Getting to know tools available for testing
Training facilities (Link to ESSA learning material Platform)	https://learn.softwareskills.eu/

2.4.4.1 Learning Units PLO 3 – Testing [e-3]

Software testing	
LU1	
Duration	3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations
Didactical Approach and delivery method	Lectures, laboratory sessions
Additional information	Lectures, problem-based learning, collaborative-based learning.
Assessment	Exam, Practical assignment, Presentation of Practical Assignment
Title of the Learning material	Testing

2.4.5 Learning Programme PLO 4 – Solution Deployment [e-3]

Overall information PLO 4 - Solution Deployment [e-3]	
N. of Learning Units	4
Learning Outcomes	<ul style="list-style-type: none"> - Writes a release plan (e.g., including solution verification and validation, documentation, supporting activities; deployment workflow and product roll-out activities) - Writes a plan for data management/ migration - Executes (parts of) a solution/ software release, applying appropriate methods, techniques, and tools (e.g., CI/CD tools; tools related to automated software release, software packaging and distribution) - Writes (parts of) release documentation
Duration	-
Total number of ECTS	starting from 2 ECTS

Recommendations for Micro-credentials	<i>Recommended as an independent micro-credential for upskilling junior developers and junior DevOps (for example EQF 4, 5).</i>
Often integrated with studies of PLO	PLO's 2, 5, 8, 9
Recommended Didactical Approach	Presence Classroom Virtual Classroom Blended e-learning Work placement
Additional comments	-
Recommended Delivery methods	<i>Lecture</i> 50% <i>Case study. Individual/team project</i> 50%
Additional comments	Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects in the same amount (time and ECTS-wise) as theoretical work. Lectures should be supported by laboratory sessions.
Work Based Learning Task (If foreseen) and Follow-up, learning reinforcement	<i>After learning the basic principles, terminology, and models of components integration, the study should focus on analysing and simulating real work-life-like tasks as, for example:</i> <ul style="list-style-type: none"> • <i>Each participant chooses a custom domain (from work or life) and create an application that can later be used in real life</i> • <i>Internship tasks of interpreting designs for applications</i>
Important (new) approaches and technologies to consider	<ul style="list-style-type: none"> • <i>Understanding integration of information systems</i> • <i>Understanding the pipeline in real-life contexts</i> • <i>Learning about managing applications</i> • <i>Getting to know approaches and most used tools</i>
Training facilities (Link to ESSA learning material Platform)	https://learn.softwareskills.eu/

2.4.5.1 Learning Units PLO 4 – Solution Deployment [e-3]

LU1	Web development: Frontend
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>

Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, Laboratory sessions, problem-based learning, collaborative-based learning, WBL.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Angular SPA Designing with Swagger and OpenAPI Type script and Angular User Authentication</i>

LU2	Virtualization
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, Laboratory sessions, problem-based learning, collaborative-based learning, WBL.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Modern Infrastructures - Virtualization</i>

LU3	Orchestration
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, Laboratory sessions, problem-based learning, collaborative-based learning, WBL.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Modern Infrastructures - Orchestration</i>

LU4	Containerization
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>

Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, Laboratory sessions, problem-based learning, collaborative-based learning, WBL.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Modern Infrastructures - Containers Modern Infrastructures - Kubernetes</i>

2.4.6 Learning Programme PLO 5 – ICT Systems Engineering [e-3]

Overall information PLO 5 - ICT Systems Engineering [e-3]					
N. of Learning Units	3				
Learning Outcomes	<ul style="list-style-type: none"> - <i>Describes software and hardware components, tools and architectures, including network components, topologies, protocols and interconnections</i> - <i>Analyses existing system infrastructures, applying appropriate methods, techniques, and tools (e.g. related to interoperability of components, security, energy consumption)</i> - <i>Proposes improvements to an existing system infrastructure, to better meet requirements (e.g., cloud solutions, Infrastructure as Code, Containers as a Service, virtual machines, load balancers)</i> - <i>Designs (parts of) a DevOps pipeline, by formulating a set of practices and tools that the development and operations teams may implement to build, test, and deploy software</i> 				
Duration	-				
Total number of ECTS	<i>starting from 2 ECTS</i>				
Recommendations for Micro-credentials	<i>Recommended as an independent micro-credential for upskilling junior developers and junior DevOps (for example EQF 4, 5).</i>				
Often integrated with studies of PLO	PLO's 2, 4, 8, 9				
Recommended Didactical Approach	Presence Classroom Blended Work placement				
Additional comments	-				
Recommended Delivery methods	<table style="width: 100%; border: none;"> <tr> <td style="border: none;"><i>Lecture and laboratory sessions</i></td> <td style="border: none; text-align: right;"><i>up to 50%</i></td> </tr> <tr> <td style="border: none;"><i>Practical assignment</i></td> <td style="border: none; text-align: right;"><i>up to 50%</i></td> </tr> </table>	<i>Lecture and laboratory sessions</i>	<i>up to 50%</i>	<i>Practical assignment</i>	<i>up to 50%</i>
<i>Lecture and laboratory sessions</i>	<i>up to 50%</i>				
<i>Practical assignment</i>	<i>up to 50%</i>				

Additional comments	Lectures, e-learning are recommended for learning the basic principles, terminology, and models of software design. These should be reinforced through practical tasks, case studies, individual/team-projects in the same amount (time and ECTS-wise) as theoretical work. Lectures should be supported by laboratory sessions.
Work Based Learning Task (If foreseen) and Follow-up, learning reinforcement	<p><i>After learning the basic principles, terminology, and models of software design, the study should focus on analysing and simulating real work-life-like tasks as, for example:</i></p> <ul style="list-style-type: none"> • <i>Internship tasks of interpreting designs for applications</i> • <i>Each participant chooses a custom domain (from work or life) and create an application that can later be used in real life</i>
Important (new) approaches and technologies to consider	<ul style="list-style-type: none"> • <i>Learning about modern approaches</i> • <i>Getting to know different tools</i> • <i>Learning structural and behavioural system specifications</i>
Training facilities (Link to ESSA learning material Platform)	https://learn.softwareskills.eu/

2.4.6.1 Learning Units PLO 5 – ICT Systems Engineering [e-3]

LU1	Orchestration
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, Laboratory sessions, problem-based learning, collaborative-based learning, WBL.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Modern Infrastructures - Orchestration</i>

LU2	Containerization
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>

Additional information	<i>Lectures, Laboratory sessions, problem-based learning, collaborative-based learning, WBL.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Modern Infrastructures - Containers Modern Infrastructures - Kubernetes</i>

LU3	Virtualization
Duration	<i>3 weeks altogether: 1 week of lectures and lab work, 1 weeks of individual student work on practical assignment, 1 week exam and presentations</i>
Didactical Approach and delivery method	<i>Lectures, laboratory sessions</i>
Additional information	<i>Lectures, Laboratory sessions, problem-based learning, collaborative-based learning, WBL.</i>
Assessment	<i>Exam, Practical assignment, Presentation of practical assignment</i>
Title of the Learning material	<i>Modern Infrastructures - Virtualization</i>

www.softwareskills.eu



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

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.