

**Project “JOBIT - Innovative teaching methodologies and courseware for software development VET to reduce skills gap in IT”**

**Teaching Handbook For Software Development Subjects**

Last update: May 2017

Author(s): European University Cyprus (Cyprus)

BCS Koolitus (Estonia)

Contributors: Centro Internazionale per la Promozione dell’ Educazione e lo Sviluppo (Italy)



This project has been funded with support from the European Commission.

This publication [communication] 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.

Contents

[Objectives 3](#_Toc491701791)

[What is JOBIT? 4](#_Toc491701792)

[Acknowledgements 5](#_Toc491701793)

[Vocational Pedagogy 7](#_Toc491701794)

[Desirable Learning Outcomes 8](#_Toc491701795)

[Subjects to be taught 9](#_Toc491701796)

[Learning & teaching methods that work best in VET 10](#_Toc491701797)

[How to apply these in JSD subjects 11](#_Toc491701798)

[Scrum 13](#_Toc491701799)

[The methodology 13](#_Toc491701800)

[The roles in scrum 13](#_Toc491701801)

[Scrum in education 14](#_Toc491701802)

[How to use scrum in a VET environment 15](#_Toc491701803)

[References 16](#_Toc491701804)

[APPENDIX 1 – SAMPLES OF LESSON PLANS 17](#_Toc491701805)

[1st Sample Lesson Plan - Programming 17](#_Toc491701806)

[2nd Sample Lesson Plan – Software Engineering 23](#_Toc491701807)

# Objectives

Through this training program you will gain expertise in (i) **how to plan**, (ii) **design**, and (iii) **facilitate an effective training course** aimed for vocational students who will be following the course of Junior Software Development (JSD). This preliminary framework will be revised based on feedback received from the pilot delivery of the VET teacher training course. By the end of this training you will:

* Develop effective strategies for teaching vocational students by getting acquainted with the main principles of VET education, and the issues that distinguish VET education from general education;
* Construct their personal professional identity as VET teachers (e.g. closely tied to their pedagogical approaches; socio-political impact; human and ethical dimensions (Tran & Nguyen, 2013));
* Gain better understanding of the desired learning outcomes for VET based on current trends and needs as mentioned by the VET community as well as industry stakeholders;
* Get familiarized with the rationale and course content of the teacher-training pack prepared by the consortium especially for JSDs, and with ways to facilitate its use during the training course;
* Develop strategies for promoting industry engagement, and particularly for increasing the teachers’ industrial contacts and awareness of how to keep up with the everlasting changing industry and its trends;
* Get acquainted with the pedagogical methodologies that work best in VET as mentioned by the previous sections prepared by the JOBIT consortium;
* Learn how to introduce the learning and teaching methods that work best in VET and cultivate these methodologies within the course material using suggested approaches.

# What is JOBIT?

Software development in a wider sense is becoming an integrated part of many different jobs in the modern society. It is a well-known fact that programmers are the most sought professionals in almost all European countries. Different sources declare thousands and tens of thousands developers needed in the labour market.

Until recent years, in the majority of European countries, software development jobs have been regarded needing educational preparation on Master’s or minimum Bachelor’s level. But today there are many successful examples of teaching programming starting already in primary schools, and also successful software development curricula on EQF levels 3, 4 and 5 can be found in different European countries.

However they are not spread enough, and sometimes employers hesitate in hiring people coming from such curricula because they are not sure whether the vocational training provides its learners the skills usually needed in this field by the labour market.

The project Innovative teaching methodologies and courseware for software development VET to reduce skills gap in IT (or JOBIT for short) focuses on meeting the needs of the employers through the development of innovative and effective software development teaching methodologies for Vocational Education and Training providers of the EQF levels 3-5. At the end of the project, the teachers of the VET software development courses will be equipped with competences and skills needed to deliver quality teaching and to secure their learners` with skills which will meet the needs of the labour market.

The project aims to increase the relevance of software development VET courses to the actual labour market needs. This will enhance the employability of VET graduates and increase their placement rate.

**Innovative teaching methodologies and courseware for the software development VET to reduce skills gap in Information Technology**

The JOBIT project partners are:

* BCS Koolitus (Estonia)– www.bcskoolitus.ee
* CEIPES - Centro Internazionale per la Promozione dell’Educazione e lo Sviluppo (Italy)– www.ceipes.org
* European University Cyprus (Cyprus)– www.euc.ac.cy.

Project JOBIT is implementing a series of activities for the target groups:

* Research about the employers‘ expectations of skills and competence from the VET school graduates entering the labour market as Junior Software Developers in project partner countries.
* Developing a methodology for the teaching of crucial subjects in software development VET courses.
* Providing specific learning resources related to the developed syllabus, training plan and methodology.
* Teacher training in all partner countries including in-class training, and practise in an organisation, resulting in more competent software development teachers.

# Acknowledgements

For this project to be effective, the consortium firstly identified the need to increase the relevance of software development VET courses with the labour market’s needs as well as the need to reduce skills mismatches and shortages in software development VET course graduates.

In order to achieve this goal we focused our research towards several stakeholders in all partner countries. The JOBIT consortium would like to thank all the employers who agreed to be interviewed and to answer surveys related to their need for skills and competences of the VET school graduates who are expected to enter the labour market as Junior Software Developers.

Moreover we extend our gratitude towards the teaching personnel who also participated in our research in our attempt to pin down the situation towards the current educational background of a Junior Software Developer VET/or not graduate in order to compare it with the market’s needs.

Both educators as well as employers role was definitive to our project. Their time and comments have been invaluable to this research, and we extend our thanks to them for their involvement thus far.

Special thanks to the following companies who have agreed to allow our trainees to visit their environment as part of their one week apprenticeship program and thus enabling our researchers to test their ideas about our new methodology regarding software developers.

1.

2.

3.

We would also like to thank those who attended our pre-training meetings in an attempt to revise our steps before proceeding to the actual training programs.

Namely:

# Vocational Pedagogy

**You cannot develop a credible description or even a theoretical foundation for vocational pedagogy unless you are prepared to ask and answer some fundamental questions about vocational education**

[Cite your source here.]

By using the term “vocational pedagogy” we combine “the science, art and craft of teaching and learning vocational education” as mentioned in “Vocational Pedagogy - What it is, why it matters and what we can do about it” (Lucas, 2014: 2).

In a more specific content, vocational pedagogy refers to the sum of the many considerations which vocational teachers handle in their everyday teaching, adjusting their methods to meet the, everlasting, changing needs of their learners and to match the context in which they find themselves (Hansen, 2008).

Vocational pedagogy is evidently under-researched and under-theorised (Rojewski, 2009; Lucas, 2014). VET is all too often seen as the “poorer cousin” of academic education. Being clearer about what vocational pedagogy is matters because it forces us to think about the wider goals of vocational education and thus to improve its status.

Once described more comprehensively, vocational pedagogy enables us to develop models and tools which can help VET teachers more effectively to match teaching and learning methods to the needs of their students and their contexts. Through such means, vocational pedagogy can directly influence the quality of teaching and learning. Lucas (2014) has documented “a line of thought” as he describes it. This is demonstrated in the figure below.

# Desirable Learning Outcomes

There are a number of capabilities that make up the working competence of a vocational worker. Lucas (2013) modernizes this set and expands it as follows:

# Subjects to be taught

Our suggestions regarding the curriculum to be taught is based on current trends and scientific evidence regarding todays VET needs. The subjects are also based on the input from the JOBIT-R-O1. The following table shows a mapping of the desired learning outcomes mentioned in the previous section to the chosen subjects.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Subject | Desirable Learning Outcomes | | | | | |
|  | Routine Expertise | | | | | |
|  |  | Resourcefulness | | | | |
|  |  |  | Craftsmanship | | | |
|  |  |  |  | Functional Literacies | | |
|  |  |  |  |  | Business-like attitudes | |
|  |  |  |  |  |  | Wider Skills |
| Introduction to Computer Science | X | X | X |  |  |  |
| Programming principles 1 | X | X | X | X |  |  |
| Programming principles 2 | X | X | X | X |  |  |
| Web development |  | X | X | X |  | X |
| Databases |  |  | X | X | X | X |
| Software development 1 |  |  | X | X | X | X |
| Software development 2 |  |  | X | X | X | X |
| Working placement (1 or 2 semesters) | X | X | X | X | X | X |

# Learning & teaching methods that work best in VET

**Constructivist Learning Approaches:**

**Learning by…**

Constructivism puts emphasis on the learning process and not on the learning outcome. The learning activities and environment should be structured in such a way that learners can create and control the development of their own learning. In this perspective, the functions of teachers and trainers are closer to guidance and coaching rather than to instruction.

**Situated Learning Approaches:**

Bridging apprenticeships should be designed to reduce the gap between theoretical learning in the formal instructional setting of the classroom and the real-life applications of knowledge in the work environment.

**Authentic Assessment Activities**

Specifically designed assignments that apply standard driven knowledge as well as skills that refer to real world challenges. Billett (2013) has identified four key strengths in the potential for securing occupational capacities through authentic experiences:

1. Engagement in work tasks
2. Indirect guidance provided by the setting
3. Practice within that setting
4. The close guidance of other workers and experts.

**Learner Centred Approaches:**

One principal goal should be for the teacher to nurture and encourage self-directed learning, by empowering students to become autonomous, self-motivated, responsible for their own learning, able to make choices about how and what they will learn (King, 1999).

**Learning by…**

**Problem Based Approaches**

An educational strategy and a method to organise the learning process in such a manner that the students are actively engaged in finding solutions to problems by themselves (Graaff & Kolmos, 2007)

**Student Mentoring**

Rather than being told what to do, learners should be asked to determine what they think and, having made an approximation, be guided by an expert mentor (Billet, 1994: 10).

Expert mentors ought to provide the modelling, coaching and scaffolding which students need to engage in authentic tasks in a gradual way, usually starting with peripheral tasks, and eventually moving into more core ones.

# How to apply these in JSD subjects

JOBIT suggests that the constructivist learning phases identified by (Wolf, 2005) be adopted to accommodate the methodologies that best serve the purposes of the VET pedagogy. These phases promote a student-centered approach that aims to produce and instill knowledge relating to individual topics but is extensible to cover larger areas of interest. The phases are:

**Initial Exposure**

Carefully selected material (readings, videos, presentations) relating to the (weekly) learning outcome(s) is available for students to interact and study. This first view happens by the students on their own to obtain a first impression and a first understanding of the material.

**Brief review**

In this phase, taking place in class, students review and discuss the subject in a collaborative environment. The instructor assumes a moderator role but can also intervene and further clarify or qualify difficult or interesting point(s). This phase will lead to a practical activity to stimulate and enhance learning.

**Guided practice activity**

This phase ideally takes the form of a guided lab. The instructor acts a mentor and students are called on to follow and via ‘doing’ achieve a clearer understanding of the activity. Either based on individuals or segregated in groups, this active learning experience will be a prelude to a homework assignment.

**Individual or group programming assignment**

In this phase, students will have the opportunity to work on their own time and put to practice what they learned in class. This coursework could be completed individually (introductory and elementary levels) or in groups (upper intermediate levels). There are issues that need consideration with group assignments; it is possible to deal with these by setting explicit expectations and requiring everyone to ‘do their bit’.

**Evaluation of learning achievement**

Completion (not copying) of individual assignments can demonstrate sufficient knowledge on the learning outcomes. In the case of group work, careful planning, specific goals, in-group individual tasks and segregation of duties can ensure learning achievement for all involved.

# Scrum

## The methodology

Scrum is an agile framework for completing complex projects. Scrum originally was formalized for software development projects, but it works well for any complex, innovative scope of work. It is widely used in IT development; however, more and more professionals are exploring alternative areas where Scrum can be applied (EduScrum Guidebook).

Scrum is based on empirical process control theory that states that knowledge comes from experience and making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk. Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

**Transparency**: Significant aspects of the process must be visible to those responsible for the outcome. Transparency requires those aspects be defined by a common standard so observers share a common understanding of what is being seen.

**Inspection**: Scrum users must frequently inspect Scrum artifacts and progress toward a Sprint Goal to detect undesirable variances. Their inspection should not be so frequent that inspection gets in the way of the work. Inspections are most beneficial when diligently performed by skilled inspectors at the point of work.

**Adaptation**: If an inspector determines that one or more aspects of a process deviate outside acceptable limits, and that the resulting product will be unacceptable, the process or the material being processed must be adjusted. An adjustment must be made as soon as possible to minimize further deviation.

Scrum prescribes four formal events for inspection and adaptation: 1) sprint planning 2) daily scrum 3) sprint review and 4) sprint retrospective

## The roles in scrum

Scrum identifies three entities or roles

* **The product owner** who is responsible for maximizing the value of the product and the work of the team. In most instances the instructor can assume this role and have the following responsibilities:
  + Setting goals, user stories, defining problems, defining meanings
  + Time planning
  + Monitoring and quality assurance
  + Evaluation and feedback
  + Coaching rather than leading, the instructor does not teach. Provide help only when asked
* **The scrum master** who is responsible for ensuring the team adheres to scrum theory, practices and rules. The scrum master is a servant-leader for the team and acts as an intermediary with those outside the team (i.e. the product owner) helping realize which interactions with the team are helpful and which are not; in such a manner maximizing the value created by the team. Responsibilities of the scrum master also include:
  + Inspiriting and inspiring the team
  + Upkeep and update of the scrum board
  + Initiates meetings
  + Facilitates the team to work and achieve the maximum possible
* **The team (of developers)** which should be a self-organizing and cross-functional entity. The team model in Scrum is designed to optimize flexibility, creativity and productivity. Teams deliver products iteratively (maximizing opportunities for feedback) and incrementally (ensuring that a useful version of a working product is always available). The characteristics of the team are:
  + It consists by between 4 and 9 members with different skills that complement each other.
  + The team is responsible for work management.
  + Team members are equal (teamwork is of utmost importance).

Scrum identifies a ‘sprint’ as a period of time (duration to depend on circumstances) in which the team decide to accomplish a subset of the tasks associated with the project.

## Scrum in education

Scrum was for the first time used in education in Netherlands (EduScrum reference). Using scrum in education gives students more opportunities to get involved in their own education. By using more project work related to practical tasks, students can learn more effectively and develop themselves in an employable way. Scrum suits students of different ages and different skills and teaches them how to be responsible. In JOBIT, we use scrum as a software development model, but also as learning method. In the learning process, we should focus to skills, which our students need more and more in the future. The table below provides a listing of these skills

|  |  |  |  |
| --- | --- | --- | --- |
| Communication and the bravery to communicate | Entrepreneurship | Responsibility, taking decisions | Learning to learn |
| Work independently | Reading skills | Self-analysis | Routine tolerance |
| Time management | Ethics | Teamwork | Flexibility |
| Problem solving | Focusing | Information searching | Self-motivation |

In VET education scrum can be used, or at least be integrated, in some manner, in order to assist with: 1) hands-on tasks; 2) teaching entrepreneurship 3) understanding and practicing teamwork; 4) practical learning, practical lectures from specialists; 5) different age-group classes (due to small numbers); 6) segregation and delegation of duties.

Software development students need to learn scrum as methodology and as model of software lifecycle, but JOBIT uses Scrum as active learning and a practical training method. The most important result of using Scrum is that it increases student’s responsibility - students feel that they themselves control their learning process. Scrum merges students and helps teachers to follow the learning results visually. Scrum increases: 1) entrepreneurship and proactivity; 2) efficiency; 3) learning becomes the student’s own learning; 4) communication skills; 5) creativity / innovation / co-creativity / self-confidence; 6) project management skills

## How to use scrum in a VET environment

The description below outlines the steps of scrum

Step 1: **making the teams** – 1)class determines the scrum master 2)scrum master chooses his/her team 3)anonymous list of skills produced by all members, scrum master to match complementary team members

Step 2: **release planning and time plan** – 1) identify goals 2) write/analyze user stories 3) create time plan 4) evaluation 5) recommended learning resources

Step 3: **sprint planning** – 1)divide sprint in user stories 2) put user stories on scrum board 3) board columns must include: ToDo, Busy, Done 4) possible e-tools for e-board

Step 4: **stand-up meetings** – 1)2-4 minutes in duration 2) participation is important 3) what was the last thing the team did; 4) what is the team doing next 5) what problems were faced and how to avoid them

Step 5: **sprint release and review** – 1) quick review, poster presentation 2) feedback from listeners 3) learn from others, share and teach/help team members

Step 6: **sprint retrospective** – 1) back to step 3 (sprint planning) 2)analyze results, evaluate team members; evaluate relations; evaluate process 3)what went well, potential corrections, possible action points

# References

Based on Scrum training in February 2017, Harri Hautala and Linda Salminen from Helsinki Business College.

eduScrum homepage <http://eduscrum.nl/en/>

Linders, B. (2013). Scrum for Education - Experiences from eduScrum and Blueprint Education. [Electronic Version]. Available from: <https://www.infoq.com/articles/scrum-education>

Sutherland, J. (2012). Scrum: The Future for Education? [Electronic Version]. Available from: <https://www.scruminc.com/scrum-future-for-education-2/>

Solingen, R. A view into the future – using Scrum as teaching tool in high school (and prior?) [Electronic Version]. Available from: <http://rinivansolingen.nl/a-view-into-the-future-using-scrum-as-teaching-tool-in-high-school-and-prior/?lang=en>

The Scrum Guide - <https://www.scrumalliance.org/why-scrum/scrum-guide>

The Beginner’s Guide to Scrum And Agile Project Management <http://blog.trello.com/beginners-guide-scrum-and-agile-project-management>

The eduScrum Guide <http://eduscrum.nl/file/CKFiles/The_eduScrum_Guide_EN_December_2013_1.0.pdf>

# APPENDIX 1 – SAMPLES OF LESSON PLANS

|  |  |
| --- | --- |
|  | 1st Sample Lesson Plan - Programming |
|  |  |
| Lesson | Object orientation principles and class construction |
| We will need | Computers with respective language IDE |
| Number of student | 10-24 (working individually or in pairs) |
| Duration | * In class 150 minutes (with 15-20 minutes break) * At home (before main lesson) 20-30 minutes for initial exposure * At home (after main lesson) between 1 and 3 hours for assignments |
| Example of programs | Class Person, class Faculty, class Student |
| Class activity sheets | [Optional] Fill in the gaps for basic syntax for class files (language depended) |
| Example of videos | <https://www.youtube.com/watch?v=SS-9y0H3Si8> |
| Class organization | Students will work on their own in guided activity. Student might pair up for assimilation activity. |
| Skills taken for granted | * Basic language syntax * Data types * Procedural programming (methods/functions/subroutines) |
|  |  |
| Scope | Introduce the notion of object orientation to students and instruct them on how to create their own classes |
| Objectives | 1. To understand the notion of object as an individual entity 2. To grasp the meaning of the attributes of an object 3. To grasp the meaning of the behaviours (methods) of an object 4. To appreciate the need for the principle of least privilege ( [[1]](#footnote-1) ) 5. To construct their own class files    1. To edit and create the required file(s)    2. To insert data attributes    3. To create access methods ( 1 )    4. To create constructor and/or destructor methods ( 1 )    5. To create other class methods 6. To instantiate objects in a program and manipulate their details (data) and/or use their methods. |
|  |  |
|  | Pre-lesson activity |
| Initial exposure | Learning material will be made available to the students before the class. This might happen during the previous class or via some online learning platform. The main aim will be to introduce the object orientation paradigm. Sample material:   * Short PowerPoint slideshow to demonstrate the need for classes quite possibly linked to ‘bespoke’ data types specific for industrial (software or hardware) use. * An instructional video (3 -5 minutes in duration) to explain object orientation and how everything tangible in the real world is an object with attributes and methods. * [Optional] links to other instructional videos * [Optional] suggested readings from a textbook * [Optional] suggested links to online instructional websites |
|  |  |
|  | Main lesson |
|  | The instructor will confirm that all students did in fact have the initial exposure experience.  [Optional] in the eventuality that not all students did the required pre-work, the instructor will spend between 5-10 minutes in an attempt to establish a basic understanding of the object orientation paradigm. |
| Brief review | The instructor initializes a conversation (of some sort) in which students are invited to discuss the object orientation paradigm and verbalize their understanding of the subject matter.  Activity: Students can be asked to provide their own examples of   * Classes * Possible data attributes * Potential class methods * Usage environments for classes   Activity: The instructor might present possible (correct or mistaken) examples for classes and students can work in pairs trying to realize and decide whether the example is correct or wrong. |
|  | The instructor will confirm that all students are clear about the subject matter and are ready to see how to define classes in practice. |
| Guided practice activity | Initially the instructor will demonstrate a brief example of defining a class (for example, the <Person> class with just the <name> and <phone>). Students must follow the instructor in order to start to familiarize themselves with the process. Also presented will be a brief driver program to demonstrate usage of the class to instantiate one or more objects.  Activity: the instructor together with the students repeat another example of a class (for example, the <Faculty> class).  Activity: the students pair-up to work on a third example of a class (for example, the <Student> class). The instructor circulates the lab offering assistance to whoever might require it.  Activity: students work individually to work on a fourth example of a class. An example class is chosen from a student suggestion made in the ‘brief review’ part or anything else, but it must be with something that students are familiar with, possibly something that they use every day (e.g. mobile phone or laptop). |
|  | The instructor prepares to close the subject and enquires whether students are clear about everything and if anything needs clarification. |
|  | The instructor concludes the class briefly reviewing the object-oriented paradigm. The instructor summarizes the steps required in order to create and utilize the classes. |
|  | After main lesson |
| Individual or group programming activity | Students must work individually. The instructor assigns a considerable piece of coursework that will have the student working on classes.  [Option] the assignment can include a number of smaller-size classes that will give the students to opportunity to practice class creation for a number of times.  [Option] the assignment can consist of a single large-size, more realistic, example class which will give the students the opportunity to explore class creation for a more realistic environment.  The assignment must include a driver program that will use the created class (or classes). The driver (and the assignment in general) should consist of a real world (possibly industry related) scenario that will have the student creating a solution for a (if possible, real) problem.  [Optional] in an attempt to avoid or at least minimize copying, the instructor could devise two or maybe more versions of the assignment and assign them to students according to certain criteria. For example, have 2 assignments (A and B) of equal difficulty; use the registration number of the students; if the last digit of the registration number is odd, then the student must work on assignment A, if the last digit is even, then the student must work on assignment B.  This assignment should not group assigned. I is crucial that students work on their own and obtain a level of understanding that will allow them to create classes on their own and apply the object-oriented paradigm.  General guidelines/Tips:   * Always explain to the student the grading scheme used for the assignment. * Allow amble time for the completion of the assignment taking under consideration other loads that the students might have or any other factors that might affect coursework timelines (e.g. national holidays etc). * It is a fact that sometimes students copy from each other or other sources. The instructor should advise students that it is not to their benefit to copy since they are not actually learning and that in an exam situation they will not be able to perform. The instructor should also make it clear that such behavior is unethical as well as unacceptable. * When the deadline for the assignment has expired, make available your solution for the assignment. This will provide the students with a point of reference as to what you expected by them. * When correcting any assignment prepare an explanation sheet that will clearly document your marking. The marking should be fair (based on the initially allotted grading) and any deduction of marks clearly explained. Students can use the sheet to discover where they went wrong and in conjunction with your solution realize what to ‘do’ right next time. |
| Evaluation of learning achievement | Students that have completed the assignment must have sufficient attainment of the learning outcomes.  [Optional] after the submission of the assignment and before the immediately following class you might ask the students to have an online quiz (using an online learning platform) which might or might not relate directly to the assignment.  [Optional] In the immediately following lesson, you may use a brief quiz and ascertain the level of learning. It can be [options]   * A fill-in the blanks of a class * A write-up a complete class * A write-up a driver based on a given class * A what’s the output given a small class and a respective driver |

|  |  |
| --- | --- |
|  | 2nd Sample Lesson Plan – Software Engineering |
|  |  |
| Lesson | Requirements elicitation |
| We will need | Pads and pens |
| Number of student | 10-24 (working either individually, or in pairs, or in teams) |
| Duration | * In class 150 minutes (with 15-20 minutes break) * At home (before main lesson) 20-30 minutes for initial exposure * At home (after main lesson) between 2 and 4 hours for assignments |
| Example of programs | n/a |
| Class activity sheets | [Optional] |
| Example of videos | <https://www.youtube.com/watch?v=vSXn16qMEZo>  <https://www.youtube.com/watch?v=XUDtUpzxvvw> |
| Class organization | Varied depending on options used |
| Skills taken for granted | * Software development |
|  |  |
| Scope | Introduce requirements elicitation and approaches used for it |
| Objectives | 1. Understand explicit requirements 2. Understand implicit (or tacit) requirements 3. Know about qualitative elicitation techniques 4. Know about quantitative elicitation techniques 5. Perform a brainstorming session 6. [Optional] Perform an elicitation interview 7. Perform document analysis |
|  |  |
|  | Pre-lesson activity |
| Initial exposure | Students must have a basic understanding of the concept of requirements elicitation, the distinction between implicit and explicit requirements. Informative material will be available to the students before the class via some online learning platform. Sample material   * Short PowerPoint slideshow to introduce requirements elicitation and the approaches used for it. * Explanatory video(s) (short in duration) to explain related concepts. * [Optional] links to other instructional videos * [Optional] suggested links to online instructional websites   [Option-A] The instructor can liaise with industry contacts and ask a stakeholder to provide a real project under consideration.  [Option-B] the instructor can liaise with industry contact and ask a group of stakeholders to visit the class and provide a real project under consideration.  If [Option-A] or [Option-B] are used, any available material should be circulated to the students so that they are aware. |
|  |  |
|  | Main lesson |
|  | The instructor will confirm that all students did in fact have the initial exposure experience.  [Optional] in the eventuality that not all students did the required pre-work, the instructor will spend between 5-10 minutes in an attempt to establish a basic understanding. |
| Brief review | The instructor initializes a conversation (of some sort) in which students are asked to discuss requirements elicitation and verbalize their understanding of the subject matter.  Various available techniques are discussed and students can be asked to categorize which type of requirements each technique can be used for. |
|  | The instructor confirms that all students are clear about the subject matter and are ready to proceed. |
| Guided practice activity | [Default] The instructor assumes the role of a stakeholder …  [Option-A] a software industry stakeholder …  … who is after some target product. Students then go through the steps of applying some techniques in order to elicit requirements.  Activity: the class and stakeholder perform a brief ‘brainstorming’ session in order to identify project objectives and needs and possibly generate a requirement set.  Activity: The client presents some typical document relating to the target product and explains the use. ‘Document analysis’ can be performed (maybe a later stage, after the stakeholder is gone, or as part of an assignment).  [Optional] Activity: The class and stakeholder simulate an interview in an attempt to extract ‘tacit’ requirements. This can happen under the guidance of the instructor and the input of the students but it has been marked as optional as it is relatively too early for students to interview clients.  Activity: the instructor can provide students with a predefined set of questions and students (either individually or in groups) go through and identify which questions are best to use in an interview situation.  Activity: if [Option-B] is used, then the class can be separated in groups and create ‘focus groups’ in order to establish requirement for separate aspects of the system.  Activity: if [Option-B] is used, then a specially designed mini ‘requirements workshop’ might be possible where stakeholders and class can come together to identify system requirements. |
|  | The instructor concludes the class briefly reviewing any theoretical knowledge gained or any practical activities implemented. |
|  | After main lesson |
| Individual or group activity | Students can work either individually or in pre-determined teams to perform the following tasks:   * For the given fictitious or real [Options A or B] project:   + Given any submitted documents by the stakeholder, perform a ‘document analyses’ and realize certain data requirements for the product.   + Draw up a draft document outlining discovered requirements for the product. * For any product:   + Perform research and come up with questions to be used in an interview of a stakeholder.   + Perform research and come up with a questionnaire that can used in stakeholders and users of the system.   General guidelines/Tips:   * Always explain to the student the grading scheme used for the assignment. * Allow amble time for the completion of the assignment taking under consideration other loads that the students might have or any other factors that might affect coursework timelines (e.g. national holidays etc). * It is a fact that sometimes students copy from each other or other sources. The instructor should advise students that it is not to their benefit to copy since they are not actually learning and that in an exam situation they will not be able to perform. The instructor should also make it clear that such behaviour is unethical as well as unacceptable. |
| Evaluation of learning achievement | Before the immediately following class you might ask the students to have an online quiz (using an online learning platform).  [Optional] For the given fictitious or real [Options A or B] project, distribute to all students, the submitted assignments and discuss them inside the immediately following class. |

1. Optional – depending on the language paradigm [↑](#footnote-ref-1)