

## ASSESSMENT BRIEF

<b>Module Name:</b>	Software Architecture
<b>Module Code:</b>	5N2772
<b>Section/technique:</b>	Skills demonstration
<b>Course:</b>	Software Development 5
<b>Teacher:</b>	Fachtna Roe
<b>Weighting:</b>	10%
<b>Title of brief:</b>	Skills Demo 2: Pair Programming
<b>Date:</b>	2024-02-09
<b>Due:</b>	
<b>Indicative duration:</b>	
<b>Brief:</b>	“Describe the role of modern methodologies in the software development process”
<b>Explainer:</b>	

---

Read the document below on the **Socratic Method**, and then answer and discuss:

- What is pair-programming?
- What advantages does pair programming have and where is it best suited for use?
- What do you think are typical pairings of people in pair-programming?
- What disadvantages does pair-programming have?
- How would pair-programming work in an Agile work environment?
- Is pair-programming compatible with the Agile manifesto?
- How does the Socratic Method benefit pair programming, when one programmer is far less experienced than the other?
- Why is the pure driver/navigator scenario unhelpful for an inexperienced programmer?

The **Socratic Method** is one of the oldest teaching methodologies. Learners are not fed answers to questions in such manner that only memory is filled. Questions are routinely answered with questions, and instead of a short and clinical "Ask, Answer" sequence something more akin to discussion and argument takes place.

The objective is not simply to answer the question put by the learners, but ideally to steer them through questioning to their own realization of an answer by connecting things already known. This is both more time consuming and more difficult for the educator, and does also require greater effort on the part of the learner. Whether the learner achieves the answer on any given occasion, or whether they only approach close to it before the answer is given, the extra effort is in actuality expended in service of answering not just the present question but the next question also, and the one beyond that. The objective of the process is to create more independent people, so that at some future point the learner will know how to construct an internal discussion in order to solve other future unseen questions and unknown problems.

Often, learners will complain.

Learners rarely see how much more effort the educator expends to avoid giving simple answers when there is a greater learning opportunity. Rather, learners will often object to the greater effort required of them also; at its simplest this reflects how all animals avoid energy expenditure.

The Socratic Method improves reasoning skills, without depending as much on mere memory. Cognition and recall are not the same. Memorizing one-off solutions is ultimately less useful than developing the ability to reason a path to a solution.

Reasoning is more energy intense than remembering. This is simply a fact and is reflected in the tendency we all have to operate in "auto-pilot"; it is also reflected in how stressed we tend to feel when required to make larger than usual numbers of decisions. Often times that stress is perceived by the learner as originating in or caused by the educator.

The Socratic Method is particularly useful for learners of Software Development. Developing software is largely a problem-solving task. The problems solved vary from day to day, from task to task, and from workplace to workplace. Where software users may well benefit from simply memorizing key and click sequences, software creators most certainly will not.

Learners should be exposed to as many problem-solving experiences as possible, to develop their reasoning skills. Over time the relative energy expenditure is adapted to and becomes natural.

Learners who cannot engage with the constant problem-solving of the method are less likely to become successful software developers.

In the case of pair programming, an expert always giving direct answers to the novice will prevent the novice from developing their own expertise in reasoning and problem solving.

**Submission:** Single HTML file (**index.html**) with all resources (images etc) linked in, in a folder called eg **red/1/SOFTARC2-RED1/** submitted via **t.fachtnaroe.net**.

<b>Software Architecture 5N2772</b>	<b>Learner Marking Sheet 1</b> <b>Skills Demonstration</b> <b>70%</b>
-------------------------------------	---

Learner's Name: \_\_\_\_\_

Learner's PPSN: \_\_\_\_\_

<b>Assessment Criteria</b>	<b>Maximum Mark</b>	<b>Learner Mark</b>
<b>Engage and document Agile Software Development Methodology / Methodologies to develop a piece of software</b>		
<ul style="list-style-type: none"> <li>Skills Demonstration 1 <b>(12 Marks) (1 Hour)</b> <ul style="list-style-type: none"> <li>Engage a modern methodology to aid the design of a system from initial requirements</li> </ul> </li> </ul>	<b>10</b>	
<ul style="list-style-type: none"> <li>Skills Demonstration 2 <b>(12 Marks) (1 Hour)</b> <ul style="list-style-type: none"> <li>Describe the role of modern methodologies in the software development process</li> </ul> </li> </ul>	<b>10</b>	
<ul style="list-style-type: none"> <li>Skills Demonstration 3 <b>(12 Marks) (1 Hour)</b> <ul style="list-style-type: none"> <li>Apply the main tools and techniques used in the gathering, recording and analysis of information relating to an existing information system</li> </ul> </li> </ul>	<b>10</b>	
	<b>10</b>	
<ul style="list-style-type: none"> <li>Skills Demonstration 4 <b>(12 Marks) (1 Hour)</b> <ul style="list-style-type: none"> <li>Generate Java (or suitable Object- Oriented language) classes from UML class diagram specifications</li> </ul> </li> </ul>	<b>10</b>	
<ul style="list-style-type: none"> <li>Skills Demonstration 5 <b>(12 Marks) (1 Hour)</b> <ul style="list-style-type: none"> <li>Use straightforward UML use-case diagrams to depict the interaction between an actor and system</li> </ul> </li> </ul>	<b>10</b>	
<ul style="list-style-type: none"> <li>Skills Demonstration 6 <b>(12 Marks) (1 Hour)</b> <ul style="list-style-type: none"> <li>Test the design specification of a new information system based on a provided use-case</li> </ul> </li> </ul>		
<b>Subtotal</b>	<b>60</b>	
<ul style="list-style-type: none"> <li>Skills Demonstration 7 <b>(10 Marks) (1 Hour)</b> <ul style="list-style-type: none"> <li>Explain the development of an information system using a traditional systems development life cycle model</li> </ul> </li> </ul>	<b>10</b>	
<b>Subtotal</b>	<b>10</b>	

Software Architecture 5N2772