

## Research period 2 Lesson-01

### Lesson 01 - Purpose

Based on [ictresearchmethods.nl](http://ictresearchmethods.nl)

This lesson focuses on formulating a research question within the field of Software Engineering. Students learn to distinguish between good and bad research questions and to formulate their own main and sub-questions.

### Lesson 01 — Research question (90 min)

Students can choose a concrete research topic and translate this into one main research question supported by appropriate sub-questions.

1. Explanation of theory: main question and sub-questions
2. Introduction of the Pizzeria organization
3. Presentation of the workflow using a video about a pizza business
4. Showing and explaining the BPMN process model

The pizzeria is considering implementing a kitchen management system to reduce administrative burden in the kitchen. Such a system provides:

- Allergen registration
- Recipe management
- Mise-en-place support
- Label printing

### In-class task:

Formulate three or four sub-questions that fit the pizzeria's main research question.

### Homework for Lesson 01 (part of the portfolio)

1. Formulate one research question within the domain of software engineering.
2. Provide an explanation of why the research question is relevant to the software engineering domain.
3. Formulate three or four supporting sub-questions.
4. Apply the research question checklist (see slides) to the sub-questions.

**Marking:**

When	Topic	Way to examine	Weight
Lesson-01			25%
	Formulate one research question within the software engineering domain	See five criteria for research question	3%
	Explain why the research question is relevant for the software engineering domain	The explanation demonstrates interest in the SE domain and current relevance.	3%
	Formulate three or four supporting sub-questions	Collectively, they must provide the full answer tot the main question	4%
	Apply the research question checklist (see slides) to the subquestions	Research question checklist	15%

## Backgroud information

**Bad research question: What is the best programming language?**

Why it is bad: Too broad, subjective, lacks context.

Improved version: Which programming language offers the best performance for real-time data processing in IoT applications within our organization?

**Bad research question: Why is agile better than waterfall?**

Why it is bad: Value-laden, general, not measurable.

Improved version: What is the impact of shifting from waterfall to agile on lead time and defect density in software projects at company X?

**Bad research question: How does artificial intelligence work?**

Why it is bad: Pure knowledge question.

Improved version: How can artificial intelligence be applied to improve bug detection in existing codebases?

**Bad research question: How can we make an app for students?**

Why it is bad: Design task, not research.

Improved version: What are the user requirements and success factors for a time-management app for students?

**Bad research question: Is Python more popular than Java?**

Why it is bad: Too superficial.

Improved version: Which factors explain the popularity of Python among software engineers in the Netherlands between 2020 and 2025?

## 2. Examples of main questions and sub-questions

### Example A - Technical focus

Main question: How can the loading time of our web application be reduced by 30% without negatively affecting the user experience?

Sub-questions:

1. What are the current performance bottlenecks?
2. Which optimization techniques exist?
3. How does caching influence performance?
4. Which solution offers the best balance between speed and usability?

### Example B - Process-focused

Main question: How does pair programming influence code quality in our team?

Sub-questions:

1. How do we define code quality?
2. How often and how is pair programming used?
3. What differences exist in defect density?
4. How do developers perceive pair programming?

### Example C - User-focused

Main question: To what extent does dark mode improve the experience of developers in an IDE?

Sub-questions:

1. Which usability criteria are relevant?
2. What research exists?
3. How do developers experience dark vs. light mode?
4. What are the implications for interface design?

### **3. What makes a question researchable?**

A researchable question:

- Is specific
- Is neutrally phrased
- Is measurable
- Is feasible
- Is relevant
- Is tied to a clear research aim

#### 4. Reflection – Your own research question

Main question:

How can the integration of different tools, software, and components ensure that at least 80% of our client's requirements are effectively met and implemented by the end of week 8?

Sub-questions:

1. What functional and non-functional requirements are currently prioritized by the client, and which additional requirements are needed to support future users? \_\_\_\_\_
2. Which tools, software, and components are most suitable for meeting at least 80% of the client's requirements? \_\_\_\_\_
3. How can the chosen tools and software components be integrated to ensure effective performance and reliability while making the product easily scalable to handle a future user usage capacity increase by two times? \_\_\_\_\_
4. What technical risks could affect the integration of the chosen tools and technologies, and how can we reduce them during the implementation of project? \_\_\_\_\_

Checklist:

☐ Is my question concrete and specific?

☐ Can I collect data for this?

☐ Is my question

neutral?

☐ Is it feasible?

☐ Does it contribute to improvement or new insight?

Reflection:

The question is concrete and specific. It clearly defines a measurable target and a fixed timeframe by stating "at least 80% of our client's requirements are effectively met and implemented by the end of week 8". It is also possible to collect data from it due to the specific wording of the question, making it possible to make a requirement list, testing results, and client feedback.

The question is also neutral, as it does not assume that a specific tool, technology or approach is better than another.

Additionally, the question is feasible within the context of the project, as it aligns with the project timeline and scope.

The research question invites us to examine how different tools and components can be integrated effectively. The research can provide practical recommendations for improving system design, performance, and scalability. The sub-questions support this by addressing client requirements, tool selection, scalability and risk management, which together lead to improvement of the project and potential new insight.