

# **Interview Guideline**

## **Introduction & Welcome (5 minutes)**

- Round of introductions
- Explanation of the objective and background
- Consent to recording

## **Demographics (15 minutes)**

### **Personal background**

- Programming experience
- How did you come to be involved in the software project?
- Do you have prior experience in software engineering?
- How long have you been involved in the project?
- How much of your time are you involved in software development?
- Are these hours part of your working time or do you do them "on top" in your free time?
- Would you consider yourself a software developer? And why?
- What distinguishes you from "traditional" software developers?

### **Software background**

- History of the project
- Reasoning for choice of programming languages in this project
- Purpose of the software
- Users of the software (number)
- Plans for the future?
  - Development
  - Usage

### **People involved in the software**

- Number of people
- Average duration of participation
- Which persons (student assistants, academic staff, professors, external) are involved?
- Who makes up the largest proportion?
- Main responsible person/expert for the software
- Hierarchy among employees (within the project)
- Communication in the development team
- Regular meetings?
- Onboarding of new employees
  - Description
  - Duration
  - Biggest challenges
- To what extent is the success of the software dependent on the developers?  
What are the implications if these people then leave the team?

# **Requirements and Requirements-related Activities**

## **Requirements Engineering (10 minutes)**

- How do you elicit requirements?
- How do you document requirements?
- How regularly do you update them?
- How do you manage your requirements?
- Would you like to do more in this area?
  - If yes: What currently prevents you from doing so?

## **Change Management (10 minutes)**

- How do you manage changes in your software?
- Do you have a roadmap for future events & deadlines?
- Do you have a central list for open tasks, ideas, known bugs?
- Do you have a change log?
- Would you like to do more in this area?
  - If yes: What currently prevents you from doing so?

## **Software Testing (10 Minutes)**

- How do you manage tests in your software?
- How do you document it?
- Have you implemented functional tests?
- Do you regularly analyze the trend of your test results?
- Would you like to do more in this area?
  - If yes: What currently prevents you from doing so?

## **Reproducibility (10 Minutes)**

- Do you pay attention to reproducibility in your software?
- How often does it happen in your area that publishing the software is required?
- What do you do to achieve reproducibility?
- What prevents you from making your software reproducible?
- Would you like to do more here?