1. 1. Gender

- [ ] Female
- [ ] Male
- [ ] Other

---

1. 2. Age

- [ ] 18-25
- [ ] 26-30
- [ ] 31-35
- [ ] 36-40
- [ ] 41-45
- [ ] 46-50
- [ ] 51-55
- [ ] 55+

---

1. 3. Studies

- [ ] University/College Bachelor degree
- [ ] University Master degree
- [ ] Doctoral degree (PhD)
- [ ] Other

---

1. 4. Job Role

- [ ] Student
- [ ] Software Engineer / Developer
- [ ] Project Manager
- [ ] Analyst
- [ ] Other

---

1. 5. Years of Working Experience

- [ ] 0-2
- [ ] 3-5
- [ ] 6-10
- [ ] 10+

---

1. 6. Employment Status

- [ ] Full time
- [ ] Part time
- [ ] Freelance/Contractor
- [ ] Self-Employed
- [ ] Currently unemployed
- [ ] Student
- [ ] Other

---

1. 7. Number of Employees at your company

- [ ] 0-20
- [ ] 21-50
- [ ] 51-100
- [ ] 101-200
- [ ] 201-500
- [ ] 501-1000
- [ ] 1000+

---

2. 1. Is your Project an Open Source project?

- [ ] Yes
- [ ] No

---

2. 2. Number of Project members

- [ ] 0-3
- [ ] 4-7
- [ ] 8-10
- [ ] 11-13
- [ ] 14-17
- [ ] 20+

---

2. 3. The main target Platform of the Project

In case it targets more than one platform, please specify the one which is mainly focused

- [ ] Dektop App
- [ ] Mobile App
- [ ] Web App
- [ ] Other

---

2. 4. Your main area/focus on the Project

- [ ] Backend
- [ ] Frontend
- [ ] Both
- [ ] Other

---

2. 5. Project's main language

Select the language on your focused area (e.g. Backend)

- [ ] Java
- [ ] Python
- [ ] C/C++
- [ ] Javascript/Typescript (including any Framework i.e. Vue, React, Angular etc)
- [ ] .Net (ASP, C# etc)
- [ ] Other

---

2. 6. Main code editor you are using for the Project

Select the editor you are using on your focused area (e.g. Backend)

- [ ] IntelliJ (any product)
- [ ] VS Code
- [ ] Eclipse
- [ ] Visual Studio
- [ ] Netbeans
- [ ] Other

---

2. 7. Are there any UML diagrams for the Project?

- [ ] Yes
- [ ] No

---

3. 1. How much, each of the following, could assist you, on understanding a Code project, considering that you have just joined the project?

0=Not at all, 5=Very much

	0	1	2	3	4	5
One-to-one (in person) guidance with a more experienced project member, to go through the code	☐	☐	☐	☐	☐	☐
Inline Code Documentation (e.g. Javadoc, docStrings etc) that may help you - while reading the code	☐	☐	☐	☐	☐	☐
External Code Documentation (e.g. Wiki pages, mkdocs, pdf etc) - to have it as a reference before or during code reading	☐	☐	☐	☐	☐	☐
Functional Docs - to understand the business	☐	☐	☐	☐	☐	☐

value of the project						
Technical Docs - for high level technical details	☐	☐	☐	☐	☐	☐
UML diagrams - for better understanding the designed structure	☐	☐	☐	☐	☐	☐
Design Patterns in code - while reading the code	☐	☐	☐	☐	☐	☐
Usage of frameworks or tools you have worked with (e.g. Angular, Vue, React, Spring, Maven, Gradle, Ant etc)	☐	☐	☐	☐	☐	☐
Naming (for classes, packages, methods, variables) and Project Structure (packaging/ components schema) - while reading the code	☐	☐	☐	☐	☐	☐
Minor Code	☐	☐	☐	☐	☐	☐

tasks/exercises, that their solutions demonstrate various parts of the code						
Code execution paths of User Stories documented in any form (could be a set of code bookmarks, breakpoinst etc) - to follow the code flow of typical use case scenarios	☐	☐	☐	☐	☐	☐
Tests, that may reveal how some features are implemented (feature tests)	☐	☐	☐	☐	☐	☐

3. 2. Consider that you came up with a cool feature on an Open Source project, and you want to contribute on it, to implement it. However, you don't have anyone to provide you information directly, and you have limited time, so you need to do your contribution without spending too much time. Which of the following actions, would you prefer doing, in order to get the "quick win"?

Read all the options and re-order them to provide your preference

- [ ] a. Run the application and try to spot similar features (probably with debugger enabled), in order to understand how the work, and start your implementation based on them
- [ ] b. Navigate to the code, starting from a fixed entry point (e.g. a breakpoint) and trying to find out the logic, by following method calls, definitions etc
- [ ] c. Read Technical docs - to find the assets you need for your implementation
- [ ] d. Read UML diagrams - to check where your feature could be added

---

3. 3. Consider that you are responsible for the 1st week of training of a new member on your Project (the project you selected before). Which of the following, would you provide to your colleague, for that early stage? Please select only those that are available to your Company and your selected Project.

Note: Typically, you would provide him all of the following, but keep in mind the time constraint here

- [ ] One-to-one guidance (including App and Code demonstration)
- [ ] External Documentation (Wiki pages, mkdocs, pdf, ppt) - self paced
- [ ] Functional Docs - self paced
- [ ] Technical Docs - self paced
- [ ] Minor Code tasks/exercises - self paced
- [ ] Other

---

3. 4. Rate the degree of each of the following factors, that in your opinion, may affect the time that a new project member needs, in order to be productive.

0=Not at all, 5=Very much

	0	1	2	3	4	5
The quality of existing training/documentation	☐	☐	☐	☐	☐	☐
New member's experience (general working experience)	☐	☐	☐	☐	☐	☐
The Language/Framework that the project is written	☐	☐	☐	☐	☐	☐
It's a Soft Skill. It depends on his/her personality	☐	☐	☐	☐	☐	☐

3. 5. What is your expectation, regarding the period that a new project member would need, in order to become productive, based on his/her level?

	0-2 months	3-4 months	5-6 months	7-9 months	10-12 months	1+ year
Junior	☐	☐	☐	☐	☐	☐
Mid-Senior	☐	☐	☐	☐	☐	☐
Senior	☐	☐	☐	☐	☐	☐

---

3. 6. How many man-days in total, do you believe, you and your team would spend, to train a new project member, based on his/her level? (Consider only the time spend on training sessions and their preparation)

	0-1	2-3	4-5	6-7	8-9	10+
Junior	☐	☐	☐	☐	☐	☐
Mid-Senior	☐	☐	☐	☐	☐	☐
Senior	☐	☐	☐	☐	☐	☐

---

3. 7. Rate how much would the following tool, may assist on a Project joining process (based on your opinion): A tool that will stand as a Virtual guide for a new member able to navigate him throughout the code, demonstrating important code parts or features, based on the instructions (configuration) that an experienced project member would provide. Think of it, like a tutorial-wizard that would present the code with extra info, comments, images and maybe voice as well. The instructions would be as simple as adding a new breakpoint, and would be able to auto-adjust, so that on code changes, the instructions (steps) remain valid. Instructions would also be under version control, for maintainability.

	0	1	2	3	4	5	
Not at all	☐	☐	☐	☐	☐	☐	Very Much