**7.2 Activity Space (Developer Guide)**

**Activity 1**

1. What is a developer guide?
2. Who reads developer guides? What do they use it for?
   * *Would they have all the required technical knowledge to understand everything in the developer guide?*
3. What contents are there in developer guides?
4. How do readers use the developer guide? What are their expectations?
5. What are the implications of all these on the DG?

Meet at 12:19 (be ready to share)

|  |  |
| --- | --- |
| Room 1 (Wei Li, Daryl, Musfirah) | Room 2 (Aileen & Emily & Wraine) |
| 1. A developer guide is a technical documentation for developers to reference from, it teaches them how the system was built 2. Potential users like companies who might be considering the use of the application. Developers of all experience level who are using the application. 3. Architecture of the app, acknowledgement, setting up (configurations & requirements), implementation, troubleshooting, testing. 4. They follow the steps given or they take reference from examples. Their expectation is that they are able to set up the app, implement it and use it. They might also want to learn what are some limitations of the platform. 5. It becomes very wordy because the DG may be detailed. | 1. Technical guide on how the application works and the flow of the program/ how it is developed. 2. Developers who want to recreate/maintain the application. [even for new developers with minimal knowledge] 3. Design and Architecture structure of the app. Documentation of the relationship of the different classes. Configuration steps of the application. 4. The readers use the developer guides as a textbook, when they read the guide, they should have a better understanding of the application such as knowing the logic and the design of the app. Their expectations could be to be able to set up the application successfully with the stated configurations. 5. A developer would become very lengthy and wordy. It is also technical which requires the reader to spend more time and effort to grasp the concept. |
| Room 3 (Jun Leong and Jun Lim) | Room 4 (Haofeng & Yu Zhong) |
| 1. Technical documentation on how the system is being run from an implementation point of view, and why certain approaches are being used. Explains to the user the overall architecture of their product. 2. Other developers who might want to make use of the product for other purposes, e.g., using certain open-source API for certain projects. Technical leads or hiring managers with similar technical capabilities, who are looking to either invest in the product or hire the people who developed it. 3. Overall architecture of the product, how to get started, how to troubleshoot, limitations/constraints faced by the product, additional properties/miscellaneous content. 4. Typically go through general overview to understand the product on a high level. After that use the doc to answer any questions regarding functionality or behaviour or the product. They expect to find most, ideally all, of their answers to their queries regarding the product there. 5. Content overload, lots of effort going through iterations of DGs that make sense to existing and future developers. Getting mired in the low-level details when they perhaps need a higher level answer. | 1. Expectation of the final product. How to use the development of the product 2. Developers who want to create a software using existing data structure. 3. All the required technical features will be inside to aid the developers to build their software. Ways to troubleshoot and test their software. 4. They will search for the specific features or function when they are not sure on the syntax or how to use. The developers expect the DG to contain examples on how to use the features and details of the features. 5. The development guide will be wordy and detailed |
| Room 5 (Alan and Shyun) | Room 6 (Haziq, Isaac) |
| 1. Technical documentation to walkthrough on how to develop application on a particular platform  2. People who wish to build an application using an existing framework.   * These people should not be expected to know everything about the framework that they are programming on, but they should have a basic level of programming competency.   3. Diagram/flowchart of the system components. Quick-start guide on how to set up the environment required for the developer to build and test applications.  4. Readers will generally use the developer guide as a manual/tutorial to guide them through with building an application/system/product using the specified framework. Readers probably expect the developer guide to be comprehensive yet concise for them to properly build their applications. Readers should also expect some technical jargons being used.  5. A possible implication is that since a developer guide can get very technical, potential developers who read the guide may get slightly confused and perhaps stressed by the unknown technical jargons used. | 1. Technical guide for the system and how it works. It also contains the rationale behind design decisions and implementation details.  2. People who want to know the technical details of the system (Potential collaborators who want to work with you/ People who want to learn about a specific system)  3. Design, implementation and testing of each aspect in the system.  4. Users can use the developer guide to understand the system. Users expect to know the rationale and thought process behind each part of the system and how it is implemented.  5. Developer guides must be informative and comprehensive such that users are able to understand every part of the implementation |
| Room 7 (Sharif and Tuan) |  |
| Developer Guides showcases how a particular software or feature was implemented.  Other developers, who are trying to improve on the current application or are trying to get inspiration from it.  Overview of the functions implemented in the software. Reasoning behind certain algorithms used. How functions interact with each other. Usage of certain programming methodologies.  Readers use the guide to allow them to replicate certain functions or features with ease. They expect user guides to be detailed and explanatory (explains complex algorithms or solutions that may not be obvious)  Implication: DG might to be too technical and reader experience may vary for developers of different experience levels. |  |

On your own, take a look at the AB3 Developer Guide.

<https://se-education.org/addressbook-level3/DeveloperGuide.html>

* What contents are there?

|  |  |
| --- | --- |
| **Sample A** | **Sample B** |
| Qn 1, 2 ,3, - Group 1  Qn 4, 5, 6- Group 2  Qn 7, 8, 9 - Group 3  Qn 10, 11 - Group 4 | Qn 1, - Group 4  Qn 2, - Group 1  Qn 3 – Group 2  Qn 4 - Group 3  Qn 5- Group 4  6 and 7 - Everybody |

**Activity 2**

**Developer Guide Journey of Discovery**

Work in your Project Teams.

**Sample A**

1. What is the purpose of this document? What software is it for? Who are its readers? Where is all this information stated?
2. Look at Design Goals on pages 5-6. What is written in this section? How is language used to evidence the writer’s critical thinking/thought process?
3. The writer takes into consideration that the reader may not be familiar with the terminology used in the software and in this document. How does the writer do this?
4. Look at the **Architecture** – **Project Overview** and **Module Overview, City Module, Weather module** sections on pages 8-12, and page 18. How is the information organized (what is the structure)?
5. How are the diagrams labelled? How does the writer make it clear which text goes with which diagram?
6. The diagrams in pages 11-18 are all of different colours. Why?
7. Look at the text on page 11. Why are some words written in a font that’s different from the other text in the paragraph? Example:

.

1. How does the writer use white space to make the document reader-friendly?
2. How does the writer make the document navigable?
3. What verb tense is primarily used in this document? Why?
4. How can the document be improved?

**Sample B:** <https://www.govinfo.gov/media/FDsys_Architecture.pdf>

1. What is the purpose of this document? What software is it for? Who are its readers? Where is all this information stated?
2. Is there a description of the software?
3. How does the writer make the document navigable?
4. Look at page 60. This section describes the Interactive Submission feature.
   1. What is written in the first paragraph?
   2. How does the writer refer to the figure in this section?
   3. What is the content of this section?
5. Look at page 80. How does the writer address the reader?
6. Look at the paragraph on **Parser Integration** page 76. What is the verb tense used there? Why does the writer use this verb tense instead of the present simple?
7. How can the document be improved?

**Activity 4**

In Project Teams, please discuss what you have learnt from this session and what you will be mindful of when writing your developer guide.

|  |  |
| --- | --- |
| **Group** | **Notes about the Developer Guide Thanks everyone!** |
| **1** | * The guide should be easy to navigate (use hyper-links wherever possible) * Content header must be clear * Make use of diagrams where applicable * Format of the guide should be straight-forward and clear * State the intended audience |
| **2** | * Decide on a standard format and language to use for the documentation. * Colouring and diagram formatting is important. * Must be consistent throughout * Easy navigation |
| **3** | * State the purpose and address the required audience * Hyperlinks * Colour-coded * Be consistent * Use clear headings * Glossary if need be * Clear structuring in terms of font/font size/indentation |
| **4** | - we will be clear and state our intended audience and purpose from the very start  - decide on the types of fonts and format for different purposes (normal text, markups, icons for special notes / tips) |