**Further Literature:**

**1. J. Siegmund, A. Brechmann, S. Apel, C. Kästner, J. Liebig, T. Leich, and G. Saake, “Toward measuring program comprehension with functional magnetic resonance imaging,” Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering - FSE  ’12, p. 1, 2012.**

**short summary:**

In the research Paper " **Toward measuring program comprehension with functional magnetic resonance imaging** " authors first stated comprehension as a internal cognitive process. They view this process from a neuroscientist's point of view and used functional magnetic resonance imaging (fMRI) to visualize such internal cognitive process. They proposed experimental design of program comprehension which answers basic question like What makes a good programmer?

Furthermore, there are so many tools available to support comprehension of code and normally, they are selected based on code characteristics. also Think a loud strategy to understand how developer's try to understand code. But these tools doesn't help to know that what is happening inside mind of the programmer at the time of comprehension.

So, in this research paper, Writers mentioned program comprehension as cognitive internal process and then used fMRI to analyze this process.

They explained very clearly how fMRI is used to analyze comprehension process. An fMRI calculates difference of oxygen stages of blood that flows inside brain. When part of the mind is active, it needs more oxygen. With fMRI, we can come to know that which part of the brain is active. And we can also derive that how many regions of the brain is required for the cognitive process and how different cognitive processes are related to each other. It also helps in identifying the functional structure of the brain. A functional structure of the brain states that which section of the mind is contributed to which cognitive process.

They focused on the work of Parnin and Rugaber. Parnin and Rugaber describe how different types of memory located in different brain areas affects different programming activities.

In Sum, they measured which brain part is activated during program comprehension and how it is different from reading comprehension. They believed that program comprehension is higher cognitive task. So their main focus was on prefrontal cortex, which is active in higher cognitive tasks. They also observed eye movements with eye tracker for the reading comprehension.

In the future, they expect to find out good programmer from a bad programmer. According to them, maybe good programmers have a certain activation pattern that completely differs from bad programmers.

**2. Y.-G. Gueheneuc, “A Theory of Program Comprehension: Joining Vision Science and Program Comprehension,” International Journal of Software Science and Computational Intelligence (IJSSCI), vol. I, no. 2, pp. 54–72, 2009.**

**Short summary :**

Y.G. Gueheneuc combined two different domains: Program Comprehension and Vision Science

in his research article **''A Theory of Program Comprehension: Joining Vision Science and Program Comprehension** ''. For the first domain, he started with simple definition of program comprehension as the processes taking place in the software engineers' minds when they understand programs. Then he explained how software engineers process available information to perform their tasks but not how software engineers *acquire* this information. For the second domain, Vision Science deals with how User gets Visual information from her/his surroundings.

And finally, **Gueheneuc** thought of combining these two concepts to give more concrete idea for Software engineers about Comprehension process.

According to **Gueheneuc**, combined theoretical framework of Program Comprehension and Vision science will help programmer understanding code and predicting new facts about learning from the environment. For vision science, he also focused on cognitive informatics which encloses Program comprehension.

Program comprehension is one domain of the software engineering which focuses on how computer scientist creates mental representation of program while understanding it.(mind map!) This mind map is needed throughout the whole process of software life cycle, which includes stages like development, maintenance, deployment & use. Gueheneuc research also contains different situations of Novice and expert Programmer in understanding and identifying the patterns of code. UML Diagram example for Composite pattern of code for expert and novice user makes it easier to understand these situations.

Vision science theories helps to understand human's vision. Based on these theories Vision scientists predict new facts. There are so many kinds of vision like spatial vision, color vision,

perception of events, eye vision, visual memory etc.

Gueheneuc discussed some theories of Program comprehension and Vision Science. Then rationale for developing these theories. Continuing to known facts & trends in Program comprehension. He divided program comprehension as 3 lines of research

1. Based on program documentation analysis.
2. Context in which program comprehension take place
3. Theories based on which programmer used the extracted data in given context.

He also explained how Different design pattern helps Software engineer to comprehend the code. The interesting part of it was how he explained the example of Composite design pattern with various below scenarios.

\_The Software Engineer Does Not Know Design Patterns.

\_The Software Engineer Knows an Unused Design Pattern.

\_The Software Engineer Knows the Used Design Pattern.

In short, they discussed program comprehension theories, based on time and effort of novice and expert programmer to comprehend the code through vision science.

In future, they expect to have deep study of internal cognitive visualization of the comprehension process, their behaviors and their forms. They also plan to compare their theory with the theory of von Mayrhauser's and Brooks'.