Developer Planning and Self-Reflection
What was I looking for?
What do I know now?
What should I do first?
Do I understand?
Have I seen this before?
What was that again?
What's the next thing I should be doing?
Where am I (in the code)?
Is this worth my time?
Is this my responsibility?
Why do I care?
Have I used this package before?
Am I making the right decision for my code?
How long have we been talking?

Vulnerability Severity/Ranking
How serious is this vulnerability?
Are all these vulnerabilities the same severity?
How do the rankings compare?
What do the vulnerability rankings mean?

Data Storage and Flow
How is data put into this variable?
Does data from this method/code travel to the database?
Where does this information/data go?
How do I find where the information travels?
How does the information change as it travels through the program?
What does this variable contain?
Is any of the data malicious?
Where is the data coming from?
What are the contents of this sql file?
Where along the data pipeline is this method?
Where along the pipeline do you want to make sure the data is secure?

Application Context/Usage
What is this method/variable used for in the program?
Are we handling secure data in this context?
What is the context of this vulnerability/code?
How does the system work?
Will usage of this method change?
Is this method/variable ever being used?
Is this code used to test the program/functionality?
Does test utils get sent to production code?
How many layers of code/security have passed through before reaching this point?

**Control Flow/Call Information**
What is the call hierarchy?
How can I get calling information?
Who can call this?
Where is the method being called?
What causes this to be called?
Are all calls coming from the same class?
What gets called when this method gets called?
How often is this code called?
Is this called from another less secure API?
How frequently does this method get called?
How many calls does it take to reach the vulnerability?
What are the parameters called?
Is the entire set of possible parameters known in advance?

**Resources/Documentation**
What type of information does this resource link me to?
What is the documentation?
Can my team members/resources provide me with more information?
Where can I get more information?
What information is in the documentation?
Is this a reliable/trusted resource?
How do resources prevent or resolve this?
How should I word my search to get the right information?
Is this the OWASP site?
Does the method have a javadoc?

**Understanding and Interacting with Tools**
Why is the tool complaining?
What is the tool output telling me?
Can I verify the information the tool provides?  
What is the tool key binding?  
What is the tool's confidence?  
What tool do I need for this?  
How is the information presented by the tool organized?  
What is the related stack trace for this method?  
How can I annotate that these strings have been escaped and the tool should ignore the warning?  

**Understanding Alternative Fixes and Approaches**  
Why should I use this alternative method/approach to fix the vulnerability?  
What are the alternatives for fixing this?  
Does the alternative function the same as what I'm currently using?  
When should I use the alternative?  
Is the alternative slower?  
Are there other considerations to make when using the alternative(s)?  
How does my code compare to the alternative code in the example I found?  
Is secure random good?  
Is prepared statement a Java thing or a 3rd party library?  
How secure is secure random?  

**Code Background and Function**  
Who wrote this code?  
Why is this code needed?  
Is this library code?  
Are there tests for this code?  
Why was this code written this way?  
What does this code do?  
Is this code doing anything?  
How much effort was put into this code?  
Why are we using this API in the code?  
Is there a list of files we are going to iterate over?  
Is rand static or final?  
Why are we collecting this information in the first place?  
Is there a variable called "dir" at the top?  
How does the code accomplish design goals?  
What is the name of the current class?  
Could the pointer be null?
Does it throw an exception instead

**Locating Information in the Code**

Where is this used in the code?
Where are other similar pieces of code?
Where is this artifact?
Is this artifact located in this class?
Where is this method defined?
Where is this class?
Where is the next occurrence of this variable?
How do I track this information in the code?
How do I navigate to other open files?
Where is this class instantiated
Where was the class instantiated

**Assessing the Application of the Fix**

How hard is it to apply a fix to this code?
How do I use this fix in my code?
How do I fix this vulnerability?
Is there a quick fix for automatically applying a fix?
Will the code work the same after I apply the fix?
Can these fix suggestions be applied to my code?
Will the notification go away when I apply this fix?
Does the code stand up to additional tests prior to/after applying the fix?
What other changes do I need to make to apply this fix?

**Preventing and Understanding Potential Attacks**

How can this vulnerability lead to an attack?
How can I replicate an attack that exploits this vulnerability?
Why is this a vulnerability?
What are the possible attacks that could occur?
How can I prevent this attack?
How should I address this problem?
What is the problem (potential attack)?
Is this a real vulnerability?
How do I find out if this is a real vulnerability?
How much value would you get from randomly injecting stuff?
Is releaseForms handling data securely?
**Relationship Between Vulnerabilities**
Does this other piece of code have the same vulnerability as the code I'm working with?
Are all the vulnerabilities related in my code?
Are all of these notifications vulnerabilities?

**End-User Interaction**
Is there input coming from the user?
Does the user have access to this code?
Does user input get validated/sanitized?

**Notification Text**
What is the relationship between the notification and the code?
What code caused this notification to occur?
What does the notification say?

**Understanding Concepts**
What is the term for this concept?
Do these words have special meaning related to this concept/problem?
How does this concept work?
What is this concept?
Is Java's true/false considered enum?
What is the nomenclature for testing suites?

**Confirming Expectations**
Is this doing what I expect it to?

**Uncategorized**
Have I exhausted all references in this method?
Is this method deprecated?
What does the constructor and datatype look like?
Was the author of this code aware of security issues?
Has the package I'm using been tested?
Is it done the same way in Python?
How can I transfer text from the IDE to the browser?
What parts of the code do you trust with this data?
What do you want to trust?
Are there non-malicious ways the code could break?