***Cognitive and cyclomantic complexity***

* Cyclomatic complexity
  + Works very well for measuring testability but not maintainability
  + Doesn’t give the measure on complexity of the code in human perspective
* Cognitive complexity
  + Increment when ther is a break in the linear flow of the code
  + Increment when structures that break the flow are nested
  + Ignore “shrthand” structures that readably condense multiple lines of code into one
  + Tells you how many test cases are needed to cover a given method

Gives a guideline on number of paths code gets through

For every split complexity gets increased

More the complexity in a method/function means more business logic built into it

Functions with bigger complexity value is a good candidate for refactoring/broken down into pieces

Functions/ files with bigger complexity may need extended testing

***Quality gates***

***Default maintainability rating***

* <=% of the time that has already gone into the application, the rating is A
* Between 6 to 10% the rating is B
* Between 11 to 20% the rating is C
* Between 21% to 50% the rating is D
* Anything over 50% is an E
* It is determined by the value of the technical debt ratio which compares the technical debt on a project to the cost it would take to rewrite the code form scratch

***Default reliability rating***

* *A = 0bugs*
* *B = at least 1 minor bug*
* *C = at least 1 major bug*
* *D = at least 1 critical bug*
* *E = at least 1 Blocker bug*

***Default security rating***

* A = 0 vulnerabilities
* B = 1 Minor vulnerabilities
* C = 1 major vulnerabilities
* D = 1 critical vulnerabilities
* E = 1 Blocker vulnerabilities

***Quality profiles***

Rules

***Rules and rule templates***

Quality profile tab

***Users, groups and permissions***

Administration -> security