FALL 2014 GROUP 1

Requirements Phase Metrics Measurements

Twitter Project

Patrick Burton

Arvind Nair

Lakshmi Swathi Chavvakul

CSCI 50600

FALL 2014 GROUP 1

Metrics measured for the Requirements Phase:

In the requirements phase, the requirements specified for each iteration in the Software Requirements Specification have been measured.

We have measured the following 3 metrics:

1. Unambiguity Metric

2. Testability Metric

3. Comprehensiveness Metric

Unambiguity Metric Measurement:

Definition:

This metric tells us whether a requirement specified in the document can be fully understood and implemented in the system.

Meaning:

Requirements specified in the requirements must not be unambiguous. They must have a single meaning only. The developer must be told clearly what is to be implemented. Otherwise it will result in wastage of time and efforts and the final product will not meet expectations.

Formula:

$$\text{Unambiguity for each iteration} = \frac{\text{Sum of all unambiguity points for each Iteration}}{\text{2 X Number of Requirements for that Iteration}} \text{ X } 100$$

The unambiguity will be given in percentage.

Points: 0: Requirements could have many meanings.

1: Requirements are not clear enough(in between).

2: Requirements are unambiguous.

Iteration Number	Requirement ID	Notes	Points	% of Unambiguity
Iteration 1	ID 1	Not specific about the locating activity.	1	
	ID 2	Accurate	2	
	ID 3	Accurate	2	
	ID 4	Accurate	2	

	ID 5	Accurate	2	
	ID 6	Accurate	2	
	ID 7	Accurate	2	
	ID 8	Log out by clicking button or by closing window.	1	
	ID 9	Accurate	2	70%
	ID 10	Accurate	2	
	ID 11	Whether database or file system and which data?	0	
	ID 12	The type of data should be specified.	1	
	ID 13	Validation using mechanism not specified	1	
	ID 14	Functions not specified	0	
	ID 15	Not specified the types of vulnerabilities	1	
Iteration 2	ID 16	Accurate	2	
	ID 17	Accurate	2	100%
	ID 18	Accurate	2	
	ID 19	Accurate	2	
Iteration 3	ID 20	Accurate	2	90%
	ID 21	Accurate	2	
	ID 22	Not specified the number	1	
	ID 23	Accurate	2	
	ID 24	Accurate	2	
Iteration 4	ID 25	Accurate	2	100%
	ID 26	Accurate	2	
	ID 27	Accurate	2	
	ID 28	Accurate	2	
	ID 29	Accurate	2	
	ID 30	Accurate	2	
	ID 31	Accurate	2	

Table 1 : Unambiguity Metric Measurement

Analysis of Results:

As seen in the charts and tables, for the second and fourth iterations the requirements are clearly understood and implemented. But for iteration 1 and 3, the requirements may be ambiguous.

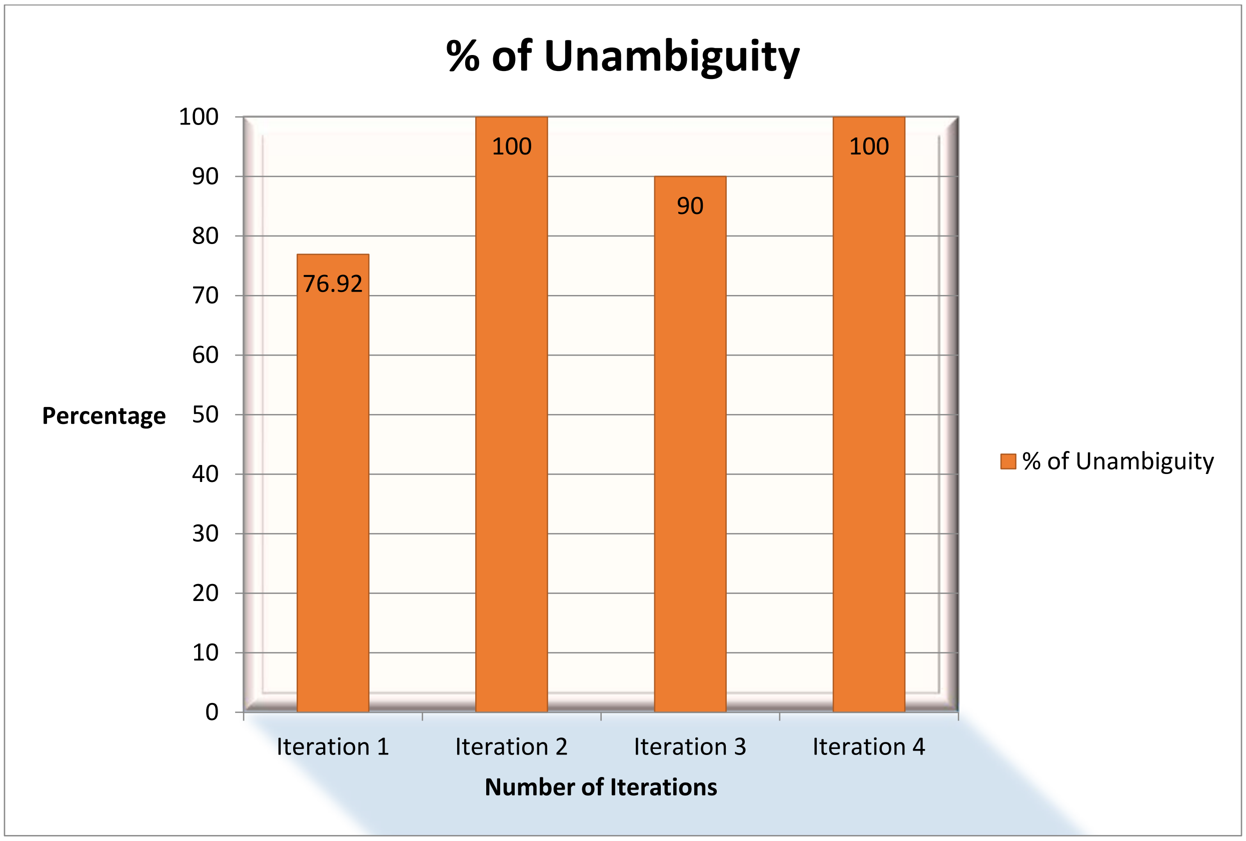

Figure 1 : Unambiguity Measurement Bar Graph

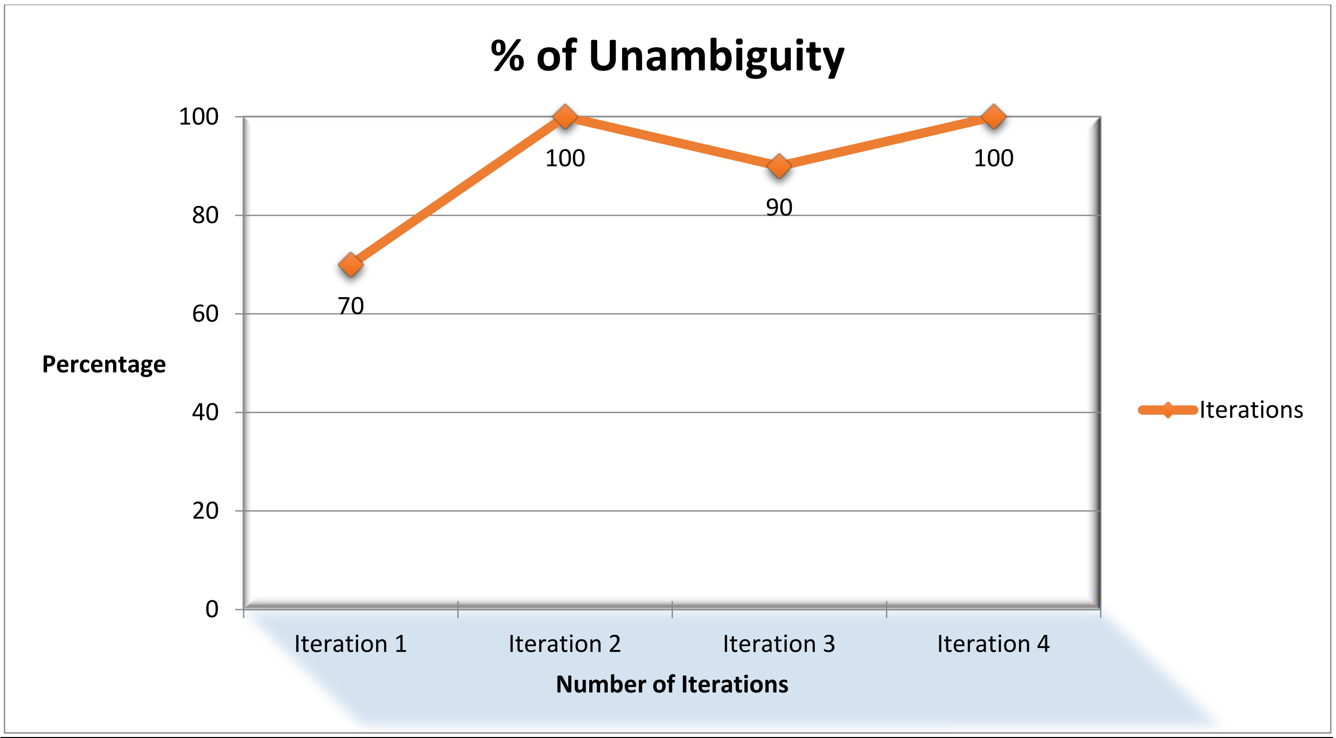

Figure 2 : Unambiguity Measurement Line Graph

Testability Metric Measurement:

Meaning:

This metric tells us whether the particular requirement mentioned in the requirements document can be tested. This helps us to understand what part of the requirements can be tested to find out if it is working in the implementation stage. This metric helps us to understand which of the requirements can be tested so that the final product can be deduced to be working on those basis.

Formula:

$$\%\text{ of Testability} = \frac{\text{Sum of total number of requirements that can be tested}}{\text{Total number of requirements}} \times 100$$

Iteration Number	Requirement ID	Can be Tested?	If yes how?	% of Testability
Iteration 1	ID 1	Yes	By going to the website's address.	73.33%
	ID 2	No	N/A	

	ID 3	Yes	By entering username into username field.	
	ID 4	Yes	By entering password into password field.	
	ID 5	Yes	By clicking on the Login button.	
	ID 6	Yes	By typing characters into message field.	
	ID 7	Yes	By checking screen.	
	ID 8	Yes	By closing the window or reloading the website.	
	ID 9	Yes	By checking with two devices.	
	ID 10	Yes	By checking if the user's name appears along with the tweet.	
	ID 11	No	N/A	
	ID 12	No	N/A	
	ID 13	Yes	When the user is taken to the tweet page of his profile it is successful.	
	ID 14	Yes	User should be only able to login, tweet and log out.	
	ID 15	No	N/A Cannot be checked at all times.	
Iteration 2	ID 16	No	N/A	75%
	ID 17	Yes	Check if all windows close and ask for login when retried.	
	ID 18	Yes	Check web browser secure connection icon.	
	ID 19	Yes	Check if user can retry more than certain number of	

			times.	
Iteration 3	ID 20	No	N/A	80%
	ID 21	Yes	Check if load more option is working.	
	ID 22	Yes	Check if the textbox takes more than that many characters.	
	ID 23	Yes	Check if send button works properly by typing tweets and clicking it	
	ID 24	Yes	Check if send box highlights on clicking it.	
Iteration 4	ID 25	Yes	Check if website is there on AWS.	71.42%
	ID 26	No	N/A Cannot be tested surely for all devices due to heterogeneity.	
	ID 27	Yes	Check if user name can be changed.	
	ID 28	Yes	Check if timestamp is recorded.	
	ID 29	Yes	Check if tweets are stored as real user ID	
	ID 30	Yes	Check if custom names are kept track of.	
	ID 31	No	N/A	

Table 2 : Testability Metric Measurement

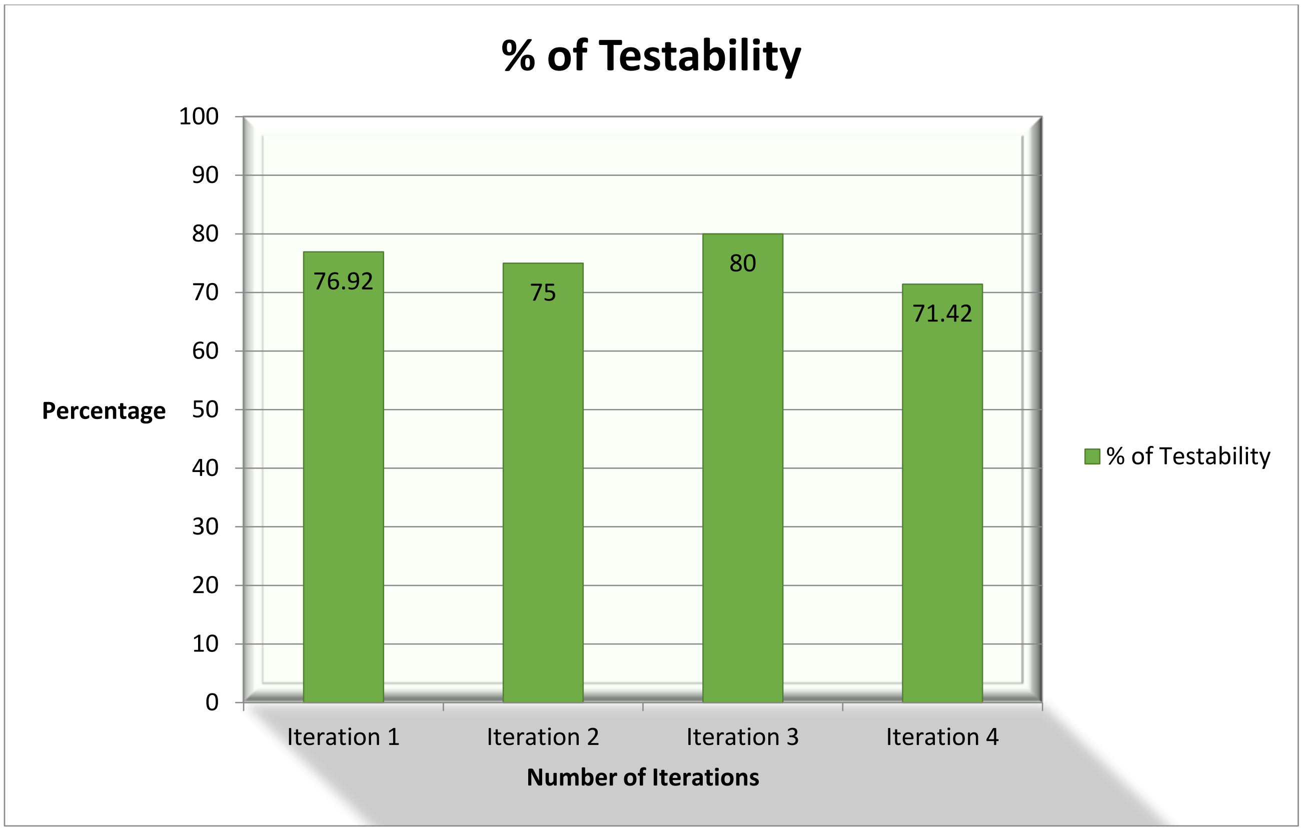

Figure 3 : Testability Measurement Bar Graph

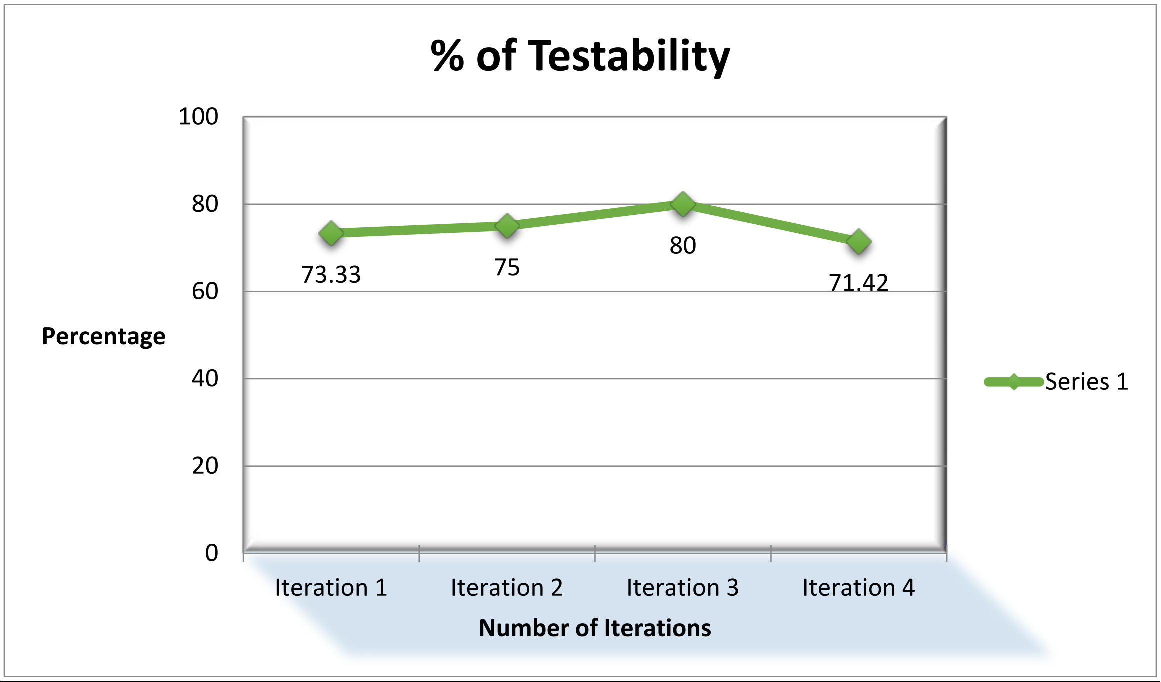

Figure 4 : Testability Measurement Line Graph

Analysis of Results:

The average of the testable requirements are around 75%. This means that most of the requirements can be tested and only 25% are non testable. Non testable requirements are not necessarily bad as some may be properties like using MySQL which need not be tested as it is sufficient to know that we have used MySQL.

Comprehensiveness Metric Measurement:

Definition:

The extent to which the wants and needs of the customer are included.

Meaning:

This metric is used for tracking the complete progress of the project. An appropriate measure would be the percentage of requirements implemented at every stage of the software project. It gives us a good idea of how much of the project gets completed in terms of all requirements implemented at every stage or iteration.

Formula:

Let T= total number of documented detailed requirements (all iterations)

$$\% \text{ of Requirements Implemented} = \frac{\text{Number of requirements implemented}}{T} \times 100$$

Iteration Number	Requirement ID	Will the requirement be implemented at each Iteration?	Final Percentage of Requirements Implemented in particular Iteration
Iteration 1	ID 1	✓	48.38%
	ID 2	✓	
	ID 3	✓	
	ID 4	✓	
	ID 5	✓	
	ID 6	✓	
	ID 7	✓	
	ID 8	✓	
	ID 9	✓	
	ID 10	✓	
	ID 11	✓	
	ID 12	✓	
	ID 13	✓	
	ID 14	✓	
	ID 15	✓	

Iteration 2	ID 16	✓	61.29%
	ID 17	✓	
	ID 18	✓	
	ID 19	✓	
Iteration 3	ID 20	✓	77.41%
	ID 21	✓	
	ID 22	✓	
	ID 23	✓	
	ID 24	✓	
Iteration 4	ID 25	✓	100%
	ID 26	✓	
	ID 27	✓	
	ID 28	✓	
	ID 29	✓	
	ID 30	✓	
	ID 31	✓	

Table 3 : Comprehensiveness Metric Measurement

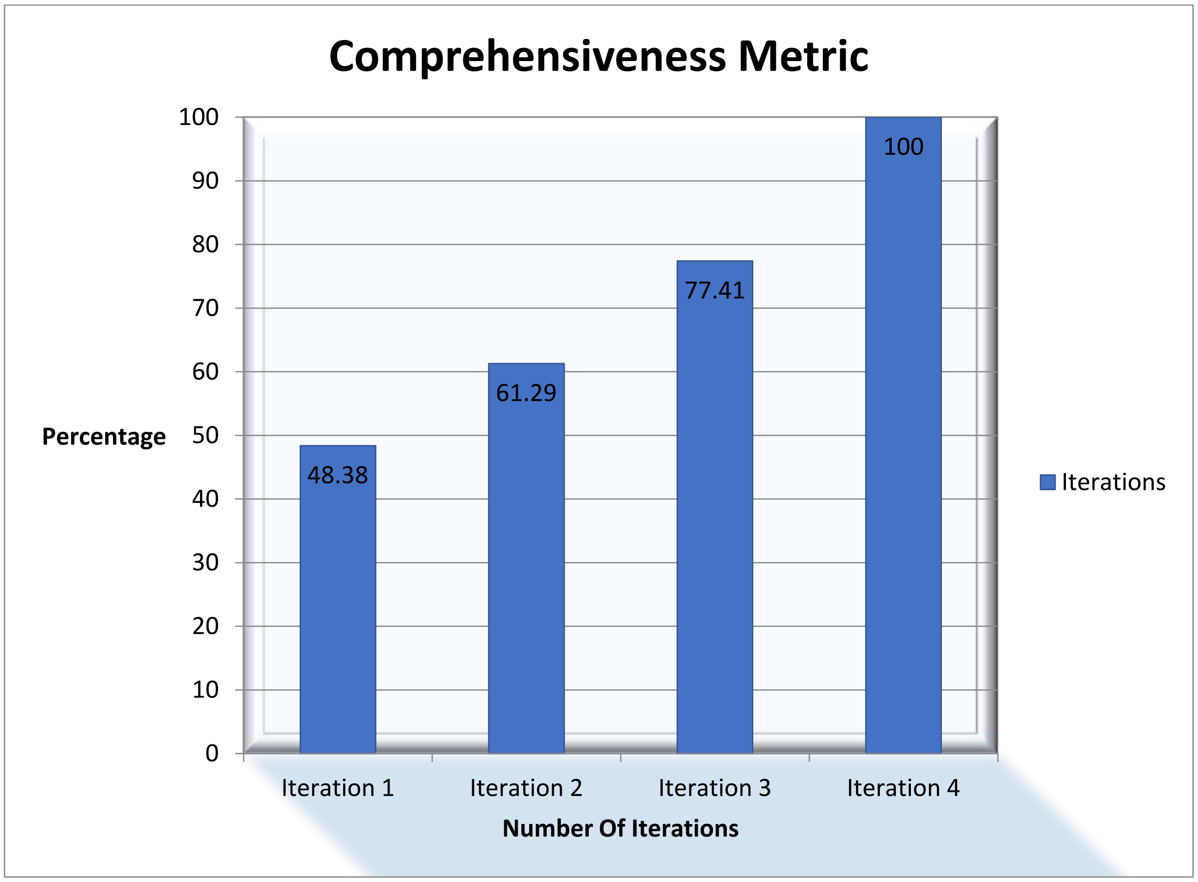

Figure 5 : Comprehensiveness Measurement Bar Graph

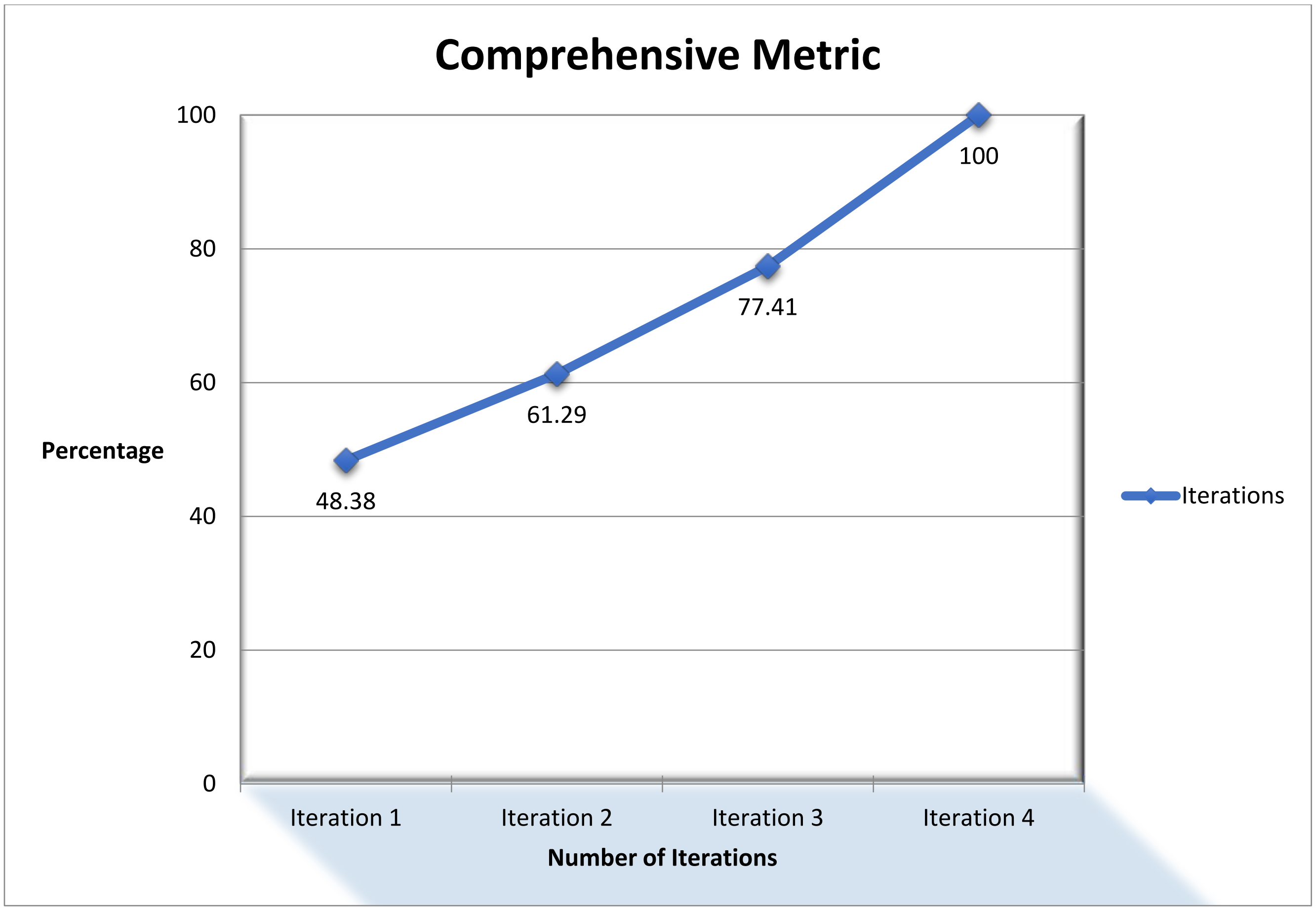

Figure 6 : Comprehensiveness Measurement Line Graph

Analysis of Results:

This would obviously show an increasing graph as at each iteration the requirements of the previous iteration would have been done. So, this metric gives us a measure of how much of the product would be implemented at each iteration and if an iteration implements less requirements or more requirements.

Conclusion:

The metric measurements should show an increasing trend but may vary depending on the metrics which are chosen. For example, the Comprehensiveness metric is showing an increasing trend but Testability is not. In an ideal scenario, Comprehensiveness should and Testability may or may not but it does not matter. For building software, the requirements phase is the hardest trying to figure out the requirements, especially at Iteration 1 as we do not know what small but essential features like load more tweets or timestamps it is missing but in the later iterations we realize. Hence it is best to follow an agile methodology.

References:

1. 1. *Software Engineering Modern Approaches Second Edition* Eric J. Braude and Michael E. Bernstein, Wiley Publications.

2. B. W. Boehm, J. R. Brown, M. Lipow, *Quantitative Evaluation of Software Quality*, TRW Systems and Energy Group, (1976).