## Unit 9: Packaging and Testing

The reading this week focusses on how to assess the complexity of code, such that it may be examined for quality prior to release to a customer.

### Required Reading

Pezze, M. & Young, M. (2008) *Software Testing and Analysis: Process, Principles, and Techniques*.

- Chapters 3, 4, 9 and 10.

Jacobson, I., Rumbaugh, J. & Booch, G. (2004) *The Unified Modeling Language Reference Manual.* 2nd ed. Pearson Education.

- Chapter 4.

Hourani, H., Wasmi, H. & Alrawashdeh, T. (2019) A Code Complexity Model of Object Oriented Programming (OOP). IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). 560-564. DOI: 10.1109/JEEIT.2019.8717448.

Ebert, C., Cain, J., Antoniol, G., Counsell, S. & Laplante, P. (2016) Cyclomatic Complexity. *IEEE Software* 33(6):27-29.

Smith, M. D,  &  Robson, D.J. (1990) Object-oriented programming-the problems of validation. Proceedings of the Conference on Software Maintenance. 272-281.

ISO/IEC (2011) ISO/IEC 25010:2011 Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models.

Pylint (n.d) Pylint Homepage.

## Additional Reading

Cheikhi, L., Al-Qutaish, R. & Idri, A. & Sellami, A. (2014) Chidamber and Kemerer Object-Oriented Measures: Analysis of their Design from the Metrology Perspective. *International Journal of Software Engineering and Its Applications* 8: 359-374.

Van Rossum, G., Warsaw, B. & Coghlan, N. (2001) PEP 8 – Style Guide for Python Code.

Goodger, D. & van Rossum, G. (2002) PEP 257 – Docstring Conventions.

Docs.python.org (n.d) unittest – Unit Testing Framework.

Docs.pytest.org (n.d) pytest: Helps you Write Better Programs.