**Reference Page**

Reginald Tremor

The University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Charmelia Butler

September 3, 2025

**References**

McConnell, S. (2004). *Code Complete: A Practical Handbook of Software Construction* (2nd

ed.). Microsoft Press. <https://archive.org/details/code-complete-2nd-edition/mode/2up>

Sommerville, I. (2015). *Software engineering (10th ed.).* Pearson Education.

[Software Engineering, 10th GLOBAL Edition](https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf)

Tsui, F., Karam, O., & Bernal, B. (2018). *Essentials of software engineering* (4th ed.). Jones &

Bartlett Learning.

Khleel, N. A. A., & Nehéz, K. (2022). *Comparison of version control system tools*. University of

Miskolc. <https://real.mtak.hu/164521/1/441_doi.pdf>

Koc, A., & Tansel, A. U. (2011). *A survey of version control systems.* International Conference

on Software Engineering and Management. <https://www.iiis.org/cds2011/cd2011imc/iceme_2011/paperspdf/fb394vz.pdf>

Maashi, M. S. (2022). *A comprehensive review of software testing methodologies based on*

*search-based software engineering*. Webology, 19(1), 51. [20220203075100pmwebology 19 (1) - 51.pdf](https://www.webology.org/data-cms/articles/20220203075100pmwebology%2019%20%281%29%20-%2051.pdf)

Myers, G. J., Badgett, C., & Sanders, T. (2011*). The art of software testing (3rd ed.)*. Wiley.

<https://onlinelibrary.wiley.com/doi/book/10.1002/9781119202486?msockid=1643d618d6636b042035c3a5d7e06af6>

Bertolino, A. (2007). *Software testing research: Achievements, challenges, dreams*. Future of

Software Engineering (FOSE '07), 85–103. <https://doi.org/10.1109/FOSE.2007.25>

Garousi, V., Felderer, M., Kuhrmann, M., Herkiloglu, K., & Eldh, S. (2020). *Exploring the*

*industry's challenges in software testing: An empirical study*. Journal of Software: Evolution and Process. <https://doi.org/10.1002/smr.2251>

Kazman, R., Klein, M., & Clements, P. (2000). *Evaluating software architectures: Methods and*

*case studies*. Addison-Wesley. <https://www.scribd.com/document/836369482/2012-Evaluating-Software-Architectures-Methods-and-Case-Studies-Klein-Mark-Kazman-Rick-Clements-Paul>

Bass, L., Clements, P., & Kazman, R. (2012). *Software architecture in practice* (3rd ed.).

Addison-Wesley.

Wiegers, K. E. (2003). *Software requirements (2nd ed.)*. Microsoft Press.

[Software requirements : practical techniques for gathering and managing requirements throughout the product development cycle : Wiegers, Karl Eugene, 1953- : Free Download, Borrow, and Streaming : Internet Archive](https://archive.org/details/softwarerequirem0002wieg)

OWASP Foundation. (2022). *OWASP top ten web application security risks*.

<https://owasp.org/www-project-top-ten/>

Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann.

[Usability Engineering: | Guide books | ACM Digital Library](https://dl.acm.org/doi/book/10.5555/2821575)