Differences between End-user software engineering and Professional software engineering

End-user programming

- To achieve program primarily for personal use.
- To support work or hobby.
- Not to be related with:
  - Inexperience
  - Only using simple programming languages.

(Ko et al., 2011)

Professional programming

- To achieve program for others to use.
- Paid to ship and maintain software over time.
- Probably:
  - Inexperience
  - Using simple programming languages.

  (Ko et al., 2011)

End-user development

- Lieberman et al. (2006): A set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify, or extend a software artifact.
- Not only ‘create’

Software engineering

- IEEE Standard 610.12: application of systematic, disciplined, quantifiable approaches to the development, operation, and maintenance of software.
- End-user software engineering: still involves systematic and disciplined activities.
  - But secondary to goal

(Ko et al., 2011)

Differences in activities

Software engineering activity	End-user	Professional
Requirements	implicit	explicit
Specifications	implicit	explicit
Reuse	unplanned	planned
Testing / Verification	overconfident	cautious
Debugging	opportunistic	systematic

Adapted from: Ko et al., 2011

References

- Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., ... & Rosson, M. B. (2011). The state of the art in end-user software engineering. ACM Computing Surveys (CSUR), 43(3), 21.
- Lieberman, H., Paternò, F., Klann, M., & Wulf, V. (2006). End-user development: An emerging paradigm (pp. 1-8). Springer Netherlands.