End-user computing

**End-user computing** (**EUC**) refers to systems in which non-programmers can create working [applications](https://en.wikipedia.org/wiki/Application_software).[[1]](https://en.wikipedia.org/wiki/End-user_computing#cite_note-EUC-1) EUC is a group of approaches to computing that aim to better integrate [end users](https://en.wikipedia.org/wiki/User_(computing)) into the computing environment. These approaches attempt to realize the potential for high-end computing to perform [problem-solving](https://en.wikipedia.org/wiki/Problem_solving) in a trustworthy manner.[[2]](https://en.wikipedia.org/wiki/End-user_computing#cite_note-2)[[3]](https://en.wikipedia.org/wiki/End-user_computing#cite_note-3)

End-user computing can range in complexity from users simply clicking a series of buttons, to writing scripts in a controlled [scripting language](https://en.wikipedia.org/wiki/Scripting_language), to being able to modify and execute code directly.

Examples of end-user computing are systems built using [fourth-generation programming languages](https://en.wikipedia.org/wiki/Fourth-generation_programming_language), such as [MAPPER](https://en.wikipedia.org/wiki/MAPPER) or [SQL](https://en.wikipedia.org/wiki/SQL), or one of the [fifth-generation programming languages](https://en.wikipedia.org/wiki/Fifth-generation_programming_language), such as [ICAD](https://en.wikipedia.org/wiki/ICAD_(software)).

Factors

Factors contributing to the need for further EUC research include [knowledge](https://en.wikipedia.org/wiki/Knowledge) processing, [pervasive computing](https://en.wikipedia.org/wiki/Pervasive_computing), issues of [ontology](https://en.wikipedia.org/wiki/Upper_ontology_(computer_science)), [interactive visualization](https://en.wikipedia.org/wiki/Interactive_visualization), and the like.

Some of the issues related to end-user computing concern [software architecture](https://en.wikipedia.org/wiki/Software_architecture) ([iconic](https://en.wikipedia.org/wiki/Graphical_user_interface) versus language interfaces, open versus closed,[[*clarification needed*](https://en.wikipedia.org/wiki/Wikipedia:Please_clarify)] and others). Other issues relate to [intellectual property](https://en.wikipedia.org/wiki/Intellectual_property), configuration and maintenance. End-user computing allows more user-input into system affairs that can range from [personalization](https://en.wikipedia.org/wiki/Personalization) to full-fledged ownership of a system.

EUC strategy

EUC applications should not be evolved by accident, but there should be a defined EUC strategy.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] Any [Application Architecture](https://en.wikipedia.org/wiki/Application_Architecture) Strategy / IT Strategy should consider the white spaces in automation (enterprise functionality not automated by [ERP](https://en.wikipedia.org/wiki/Enterprise_resource_planning) / [Enterprise Grade Applications](https://en.wikipedia.org/w/index.php?title=Enterprise_Grade_Applications&action=edit&redlink=1)). These are the potential areas where EUC can play a major role.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] Then ASSIMPLER parameters[[*which?*](https://en.wikipedia.org/wiki/Wikipedia:Avoid_weasel_words)] should be applied[[*by whom?*](https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Words_to_watch#Unsupported_attributions)] to these white spaces to develop the EUC strategy. (ASSIMPLER stands for availability, scalability, security, interoperability, maintainability, performance, low [cost of ownership](https://en.wikipedia.org/wiki/Cost_of_ownership), extendibility and reliability.[[4]](https://en.wikipedia.org/wiki/End-user_computing#cite_note-4))