

Human Computer Interaction

- Software needs suitable mechanism for interacting with its users
- Need to think about the nature of the interaction from the user perspective – their goals/priorities when using the software
- Need to consider user resources available (cpu, bandwidth)
- Need to consider user environment (noise, lighting, distraction)
- Need to think about how the nature of the interaction impacts behaviour of system (e.g. error prevention)
- How can system best interact with them to reflect all this?

Suitable communication

- Consider correct level of formality, humour for user and situation
- Ensure communication and guidance never leaves user feeling lost or confused about next steps
- Consider suitable metaphors for design
- Balance best input mechanisms to support/guide newer user yet still be efficient enough for power user
- Think about audio vs visual cues, and analog vs digital displays, and pros/cons of each for situation

Design vs tools

- Most languages have some form of support for graphical interface elements (radio buttons, pop ups, spinners, forms, etc)
- Those are secondary to first thinking carefully about how your system and users will interact, what information needs to be communicated in each direction and under what circumstances

Testing

- Adding more complex interactions also results in more complex testing
- Need to test that the behaviour is technically correct (all the right elements appear in all the right places, and do all the right things when selected)
- Also need to test the behaviour is practical/usable: how well suited is it for the users in reality? (usability testing)

GUI tools

- Of course, we need some experience actually developing with graphical components
- Will look at zenity to provide quick gui interface elements for bash scripts
- Will look at gtk library to provide more complex gui interface elements with a C++ program