Software Construction and User Interfaces (SE/ComS 319)

Ali Jannesari

Department of Computer Science

Iowa State University, Spring 2019

USER INTERFACES

Outline

- User interfaces
- Data visualization
- UI Design process
- User interfaces construction

User interface

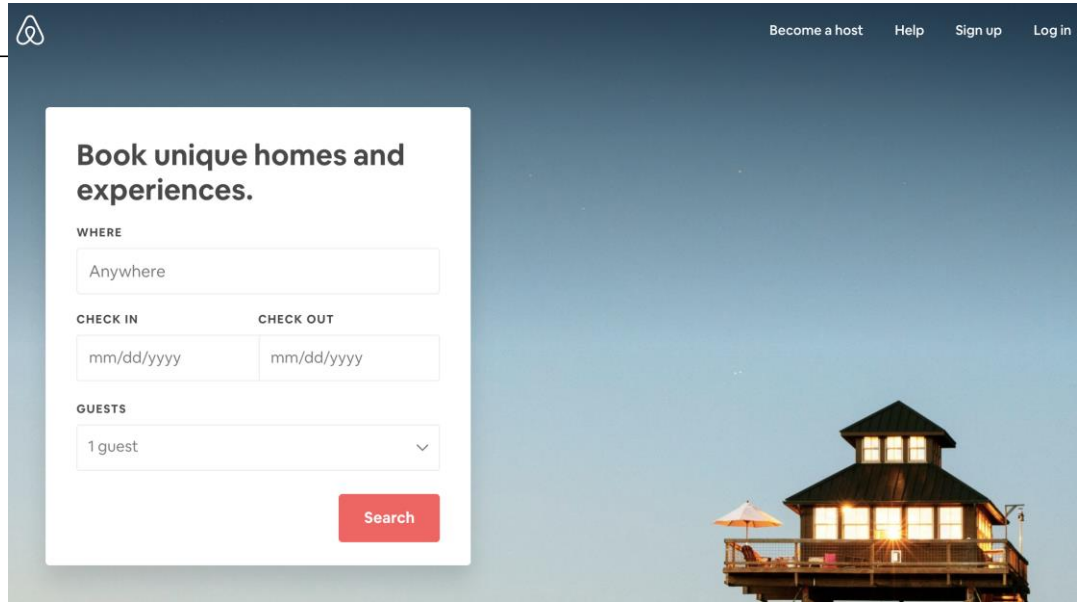
- User interface: Way by which end-users will interact with your software
- Should take into consideration users' expectations, experience and skills
- Bad interface ➔ low usability

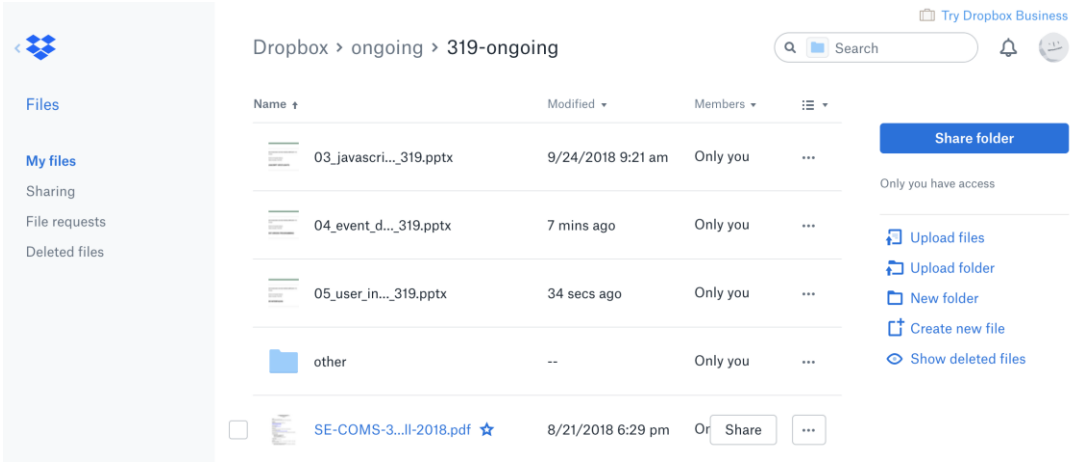
User interface

- Human errors have been correlated to the usability of user interface (J. Galliers and et al, ACM TCHI).
  - Confirmation dialog box:

User interface

- Examples of excellent UI:
  - AirBnB, DropBox, …





User interface

- Examples of poor UI
  - IBM Lotus Notes
  - Windows 8,…

Visible language

- **Visible language** refers to all of the **graphical techniques** used to communicate the message or context:
- **Layout**: formats, proportions, grids and 2-D/3-D organization
- **Typography**: selection of typefaces (fonts) and typesetting, including variable width and fixed width
  - Monospaced fonts are same width opposed to variable-width fonts, where the w and m are wider than most letters, and the i is narrower!
- **Color and Texture**: color, texture and light that convey complex information and pictorial reality
- **Imagery**: signs, icons and symbols, from the photographically real to the abstract

Visible language (2)

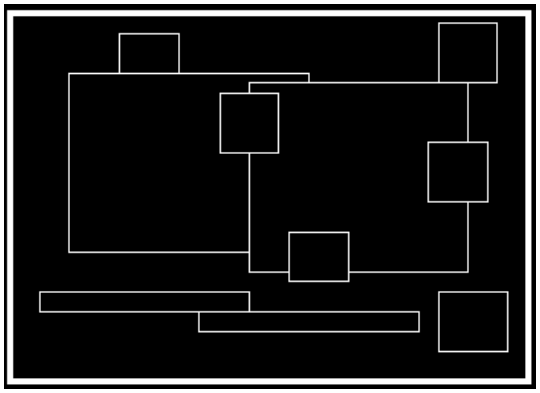
- **Sequencing**: the overall approach to visual storytelling
- **Sound**: abstract, vocal, concrete, or musical cues
- **Visual identity**: the additional, unique rules that lend overall consistency to a user interface
  - The overall decisions as to how the corporation or the product line expresses itself in visible language
    - E.g. ISU wen Red & gold

Use of a visible language – Fundamental principles

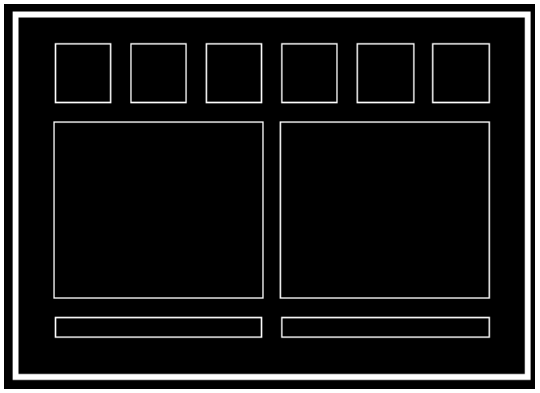
- **Organize**: provide the user with a clear and consistent conceptual structure
  - Consistency, screen layout, relationships and navigability
- **Economize**: do the most with the least amount of cues
  - Simplicity, clarity, distinctiveness, and emphasis
    - Emphasis: The most important elements should be easily perceived. Non-critical elements should be de-emphasized.
- **Communicate**: match the presentation to the capabilities of the user
  - In order to communicate successfully keep in balance legibility, readability, typography, symbolism, multiple views, and color/texture

Use of a visible language – Fundamental principles (2)

- **Organize**: provide the user with a clear and consistent conceptual structure (consistency, screen layout, relationships and navigability)
  - Layout

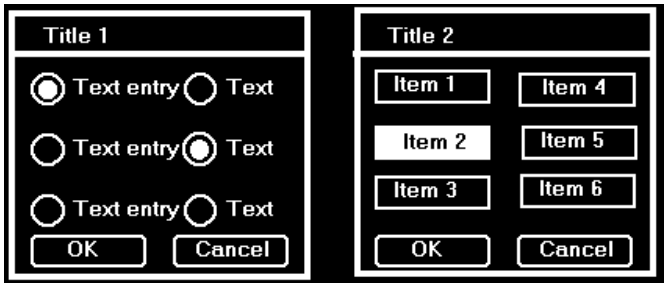


Chaotic Screen



Ordered Screen

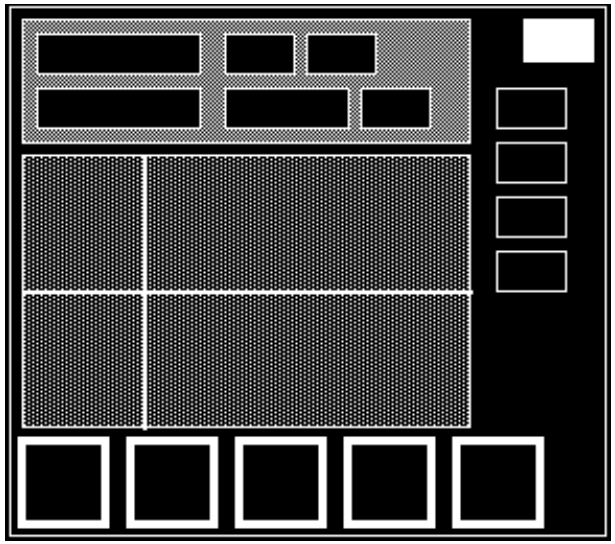
  - Consistency: Same kinds of elements are shown in the same places

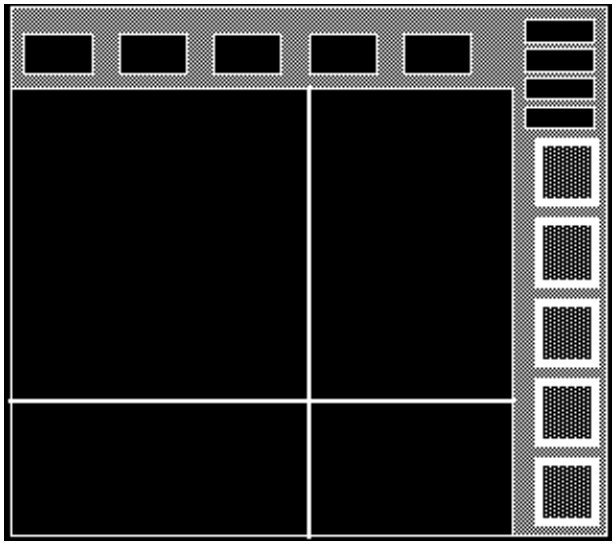


http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html

Use of a visible language – Fundamental principles (3)

- **Organize**: provide the user with a clear and consistent conceptual structure (consistency, screen layout, relationships and navigability)
  - Relationships: Linking related items and disassociating unrelated items can help achieve visual organization

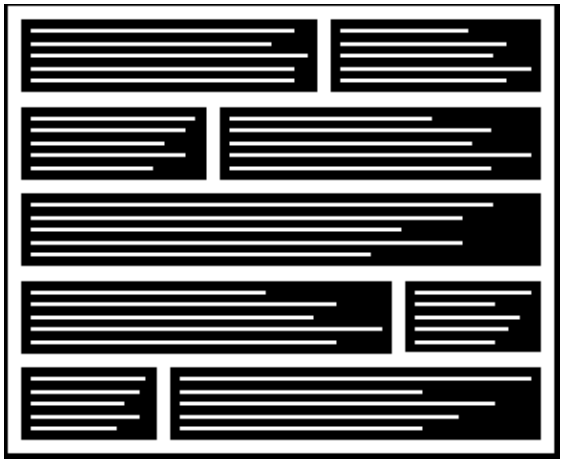




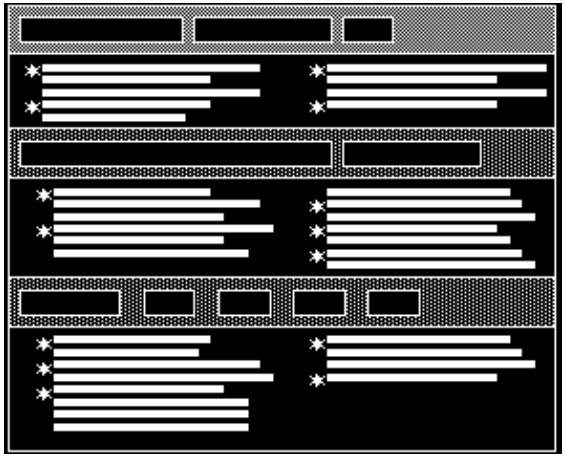
http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html

Use of a visible language – Fundamental principles (4)

- **Organize**: provide the user with a clear and consistent conceptual structure (consistency, screen layout, relationships and navigability)
- Navigability



Poor design



Improved design

- Spatial layout and color help focus viewer's attention to most important areas
- Bulleted items guide the viewer through the secondary contents

http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html

Use of a visible language – Fundamental principles (5)

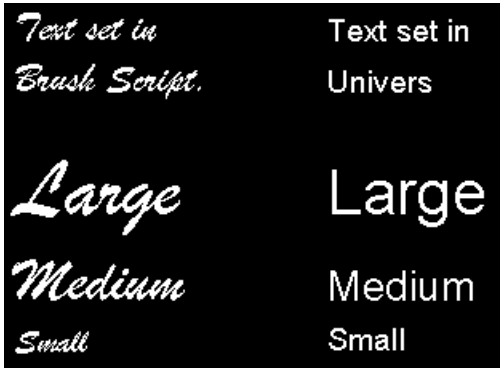
- **Economize**: do the most with the least amount of cues
  - Simplicity, clarity, distinctiveness, and emphasis
  - Simplicity
    - Includes only the elements that are most important for communication
    - It should also be as unobtrusive as possible

Complicated and Simpler Designs

  - Emphasis: The most important elements should be easily perceived. Non-critical elements should be de-emphasized.

Use of a visible language – Fundamental principles (6)

- **Communicate**: match the presentation to the capabilities of the user
  - In order to communicate successfully keep in balance legibility, readability, typography, symbolism, multiple views, and color/texture



Illegible and Legible Texts

  - Developing better visual (color or black-and-white) communication is an important part of making computer graphics that communicate effectively and efficiently through graphic design

Design principles for user interfaces

- UI Design Issues
- UI Design Process
- UI Evaluation

UI Design Issues

- Human factors
- Interaction styles (to/from the user)
  - Visualization
  - Error/warnings
  - Color
  - . . .

Human Factors

- Limited short-term memory
  - How many items of information can one remember instantaneously?
- Familiarity
  - Use terms and concepts from the domain of the application
- Consistency
  - Similar/comparable operations should be activated in the same way
- Error recovery & guidance
  - Provide meaningful, unambiguous feedback when errors occur

Interaction styles

- Obtaining information from the user
- Presenting information to the user
- **Direct Manipulation**
- **Menu-based**
- **Form-based**
- **Natural language**
- **Command language**

Interaction styles – Input

- Direct manipulation
  - Advantage: Intuitive interaction
  - Disadvantage: Hard to implement, requires visual metaphor
  - Applications: Games, CAD

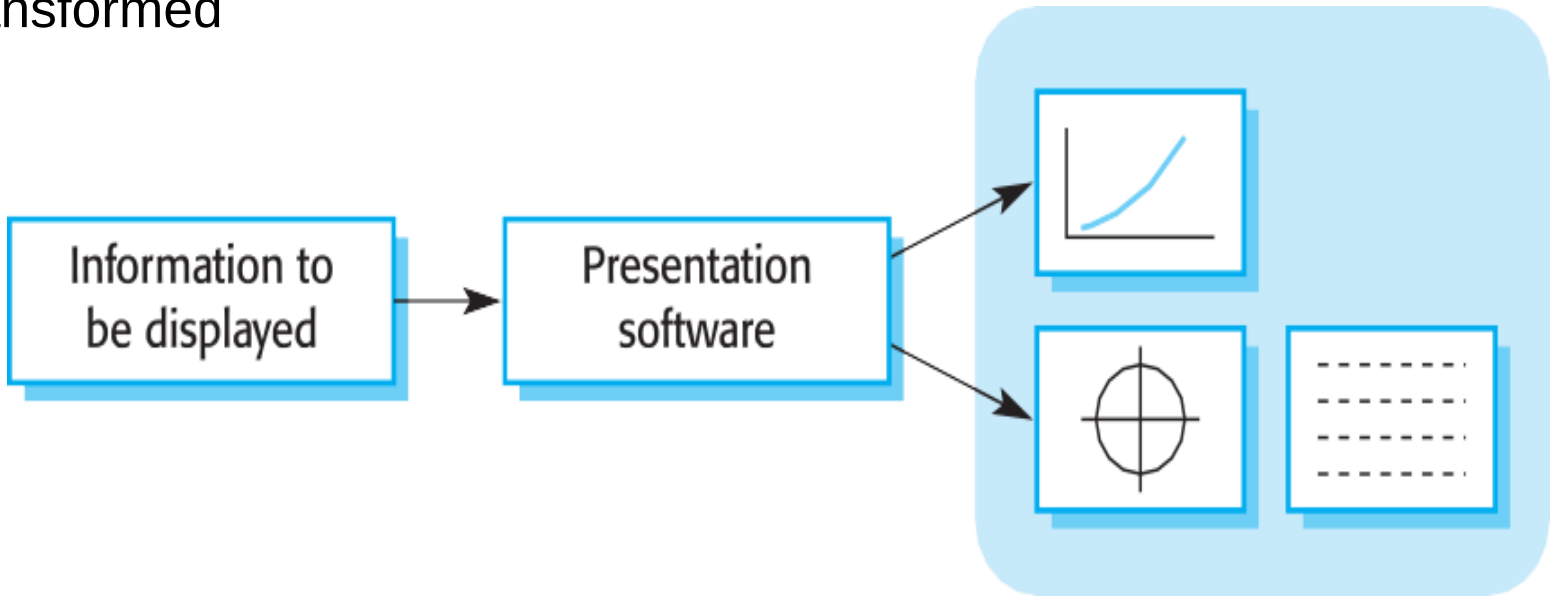
- Menu-based
  - Advantage: Avoids user error
  - Disadvantage: Can be slow and/or complex
  - Applications: Most systems

Interaction styles – Input

- Form-based
  - Advantage: Simple and Checkable
  - Disadvantage: Can be long
  - Applications: e-Commerce
- Natural Language
  - Advantage: easy and natural
  - Disadvantage: Natural language processing (NLP)
  - Applications: Information retrieval systems, apps
- Command-Line Language
  - Advantage: : Easy to implement
  - Disadvantage: Hard to understand/remember all commands

Interaction styles – output

- Direct
- Transformed



DATA VISUALIZATION

Data visualization

- Techniques for displaying large amounts of information
- May reveal relationships and/or trends of data
  - Could improve human problem-solving performance
  - Could influence Business Decisions
  - **Wrong inferences!!** (Risks of Visualization)

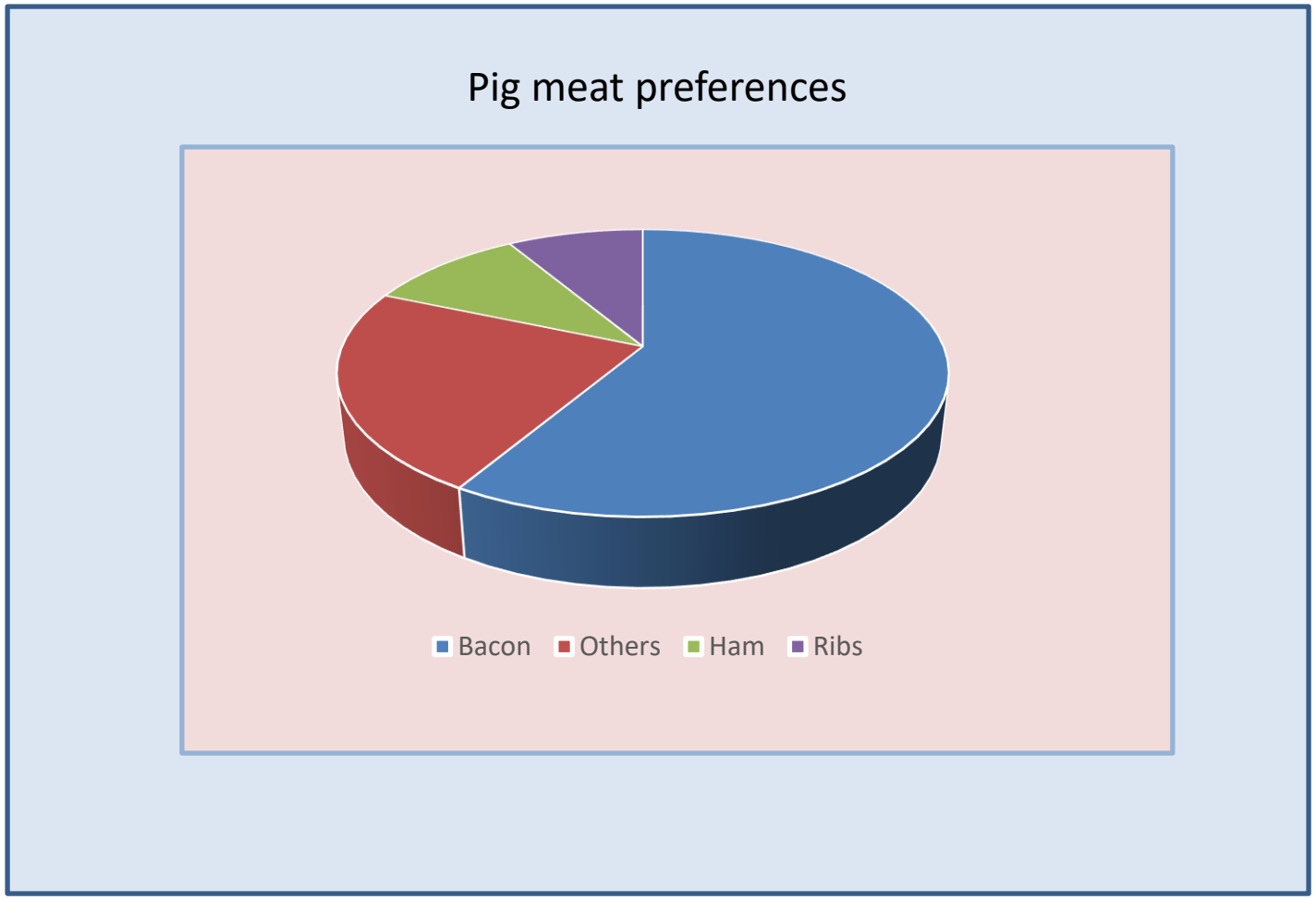
Why data visualization?

- Data exploration is easy
- Easy to communicate idea
- Easy to share data and results with audience
- Easy to share findings with stakeholders

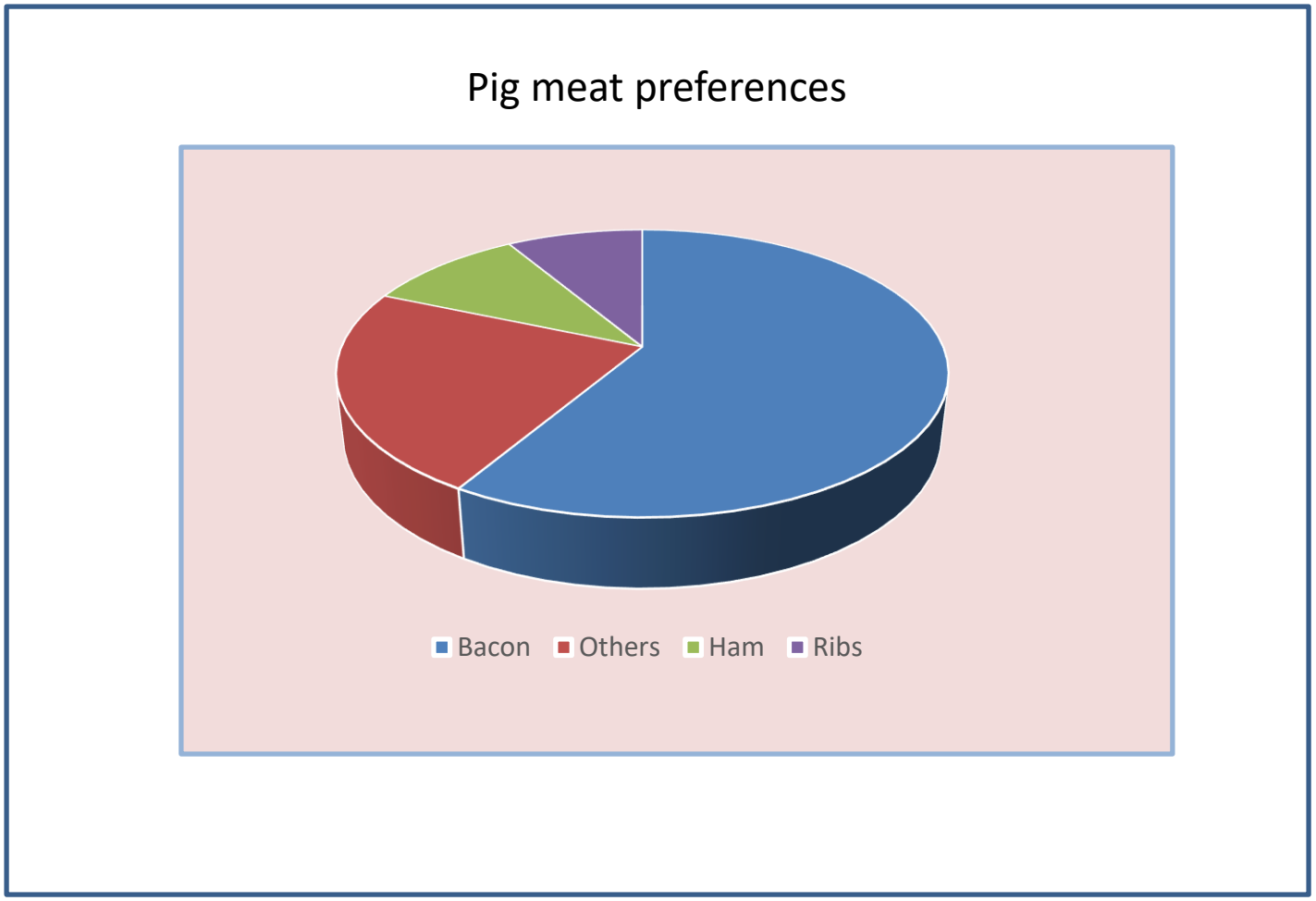
Best practices for data visualization

- Less is more effective
- Less is more attractive
- Less is more impactive

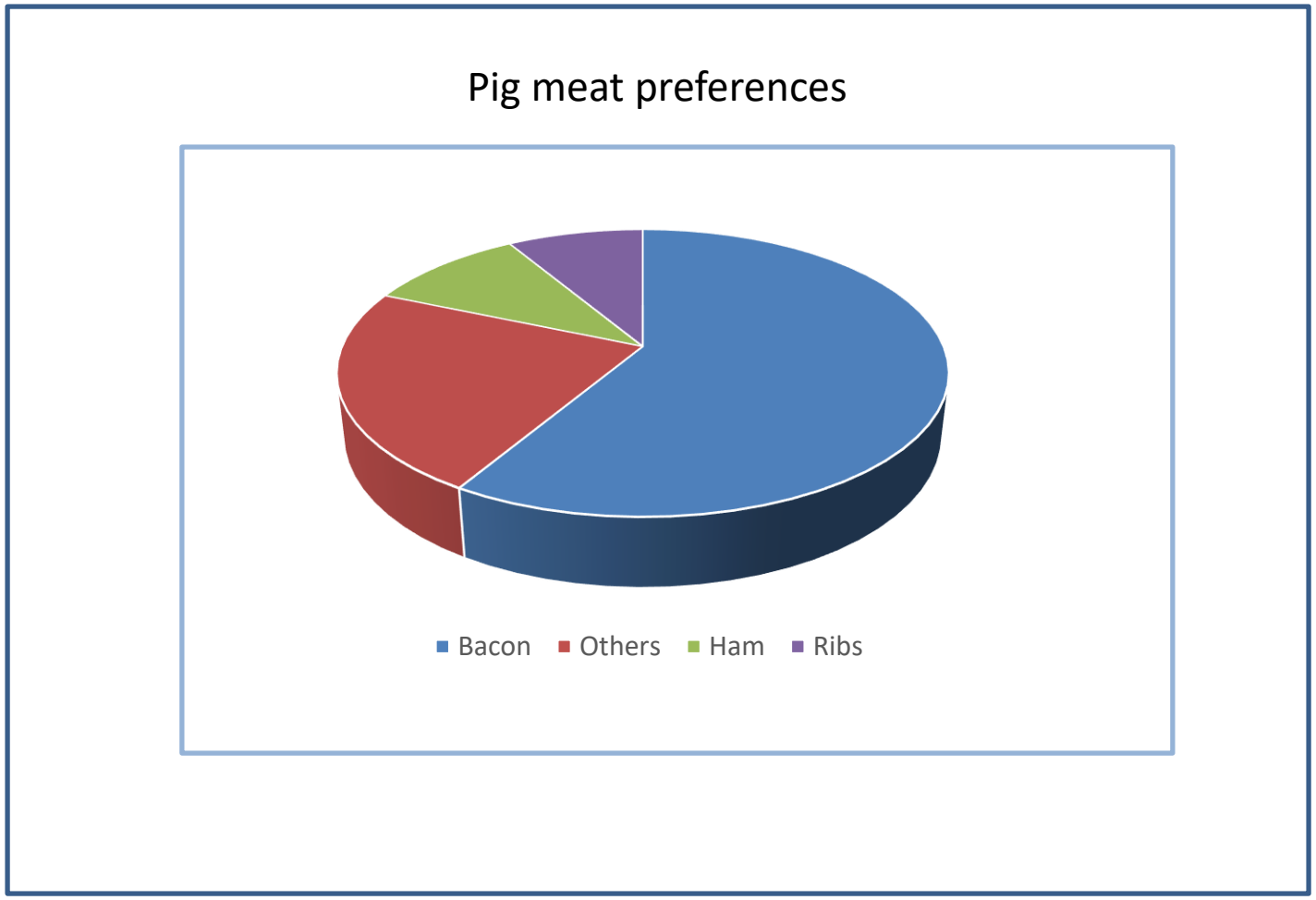
Best practices – Example (1)



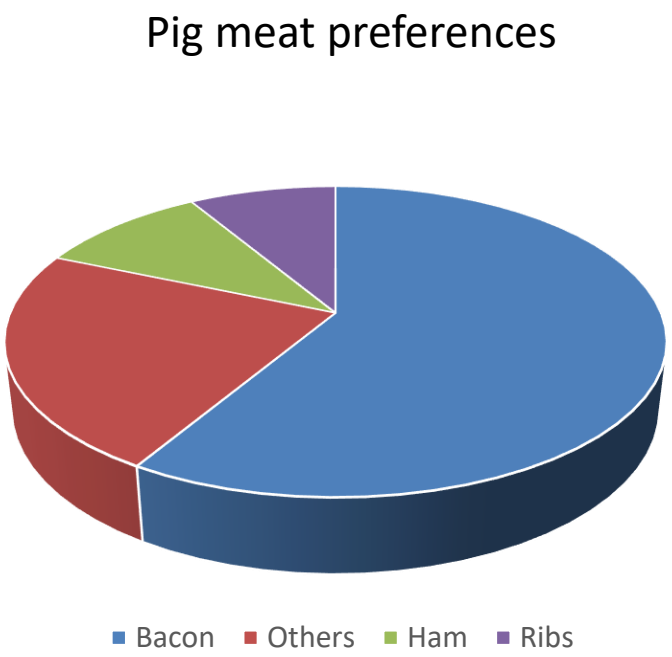
Best practices – Example (2)



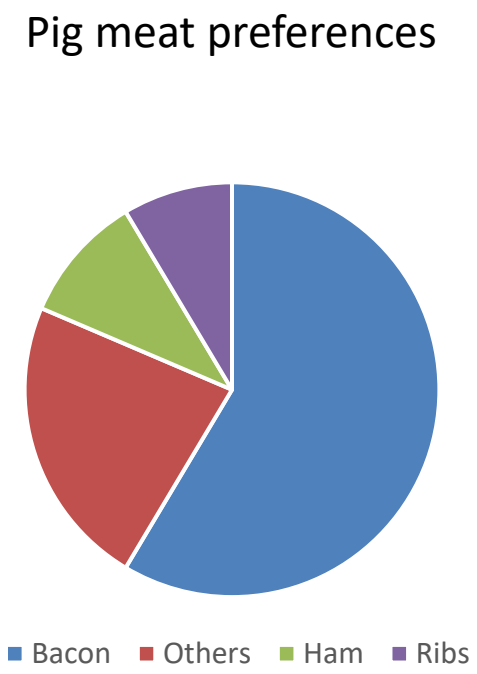
Best practices – Example (3)



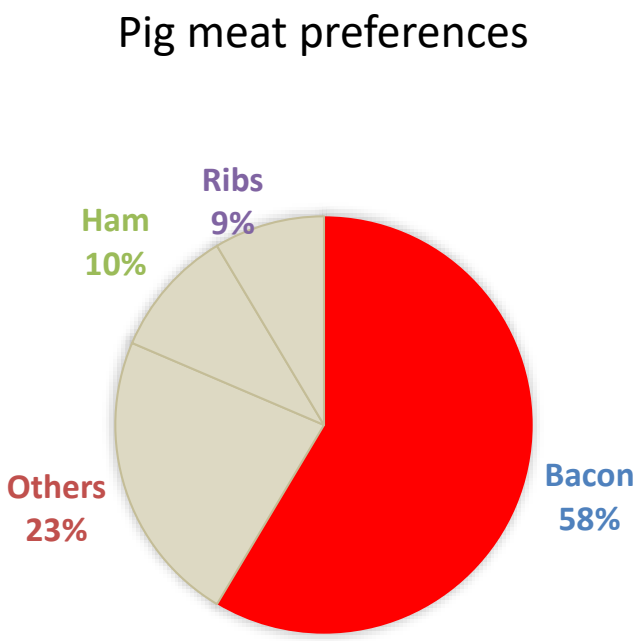
Best practices – Example (4)



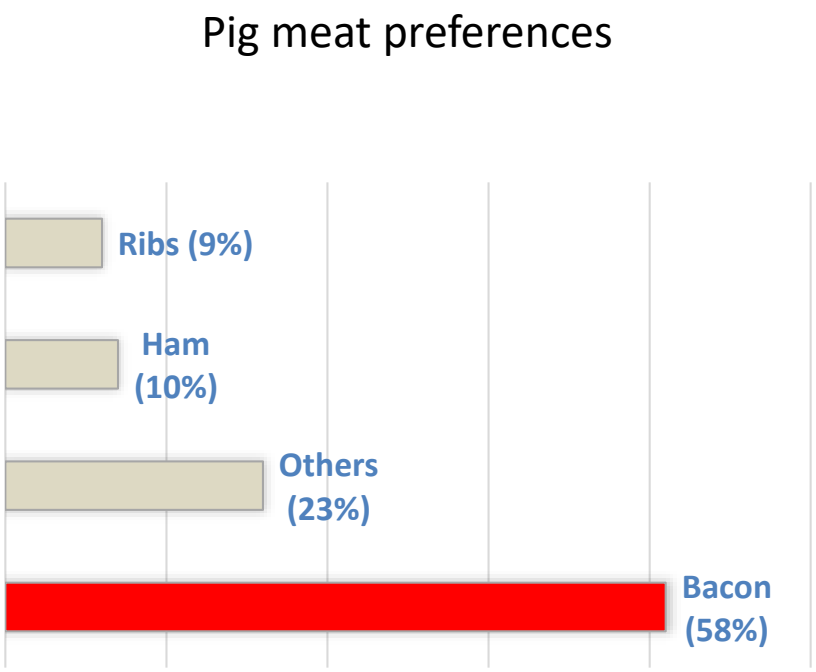
Best practices – Example (5)



Best practices – Example (6)

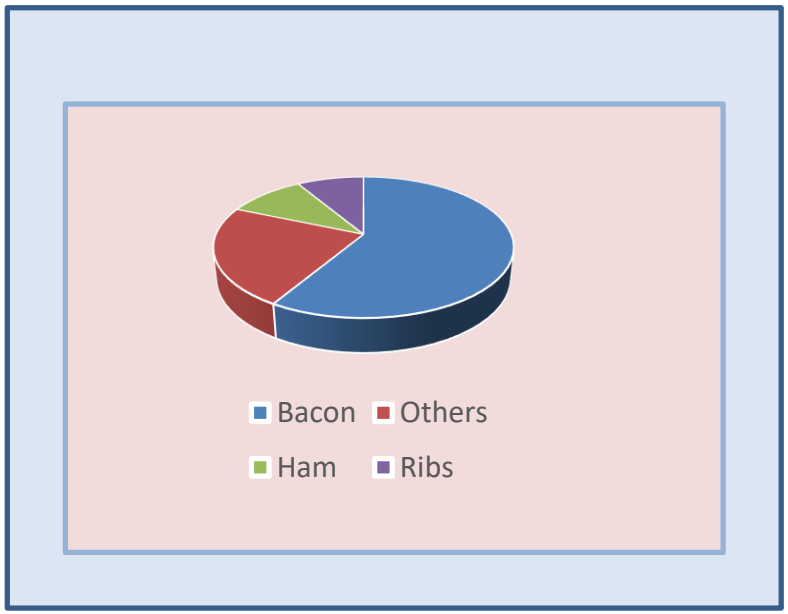


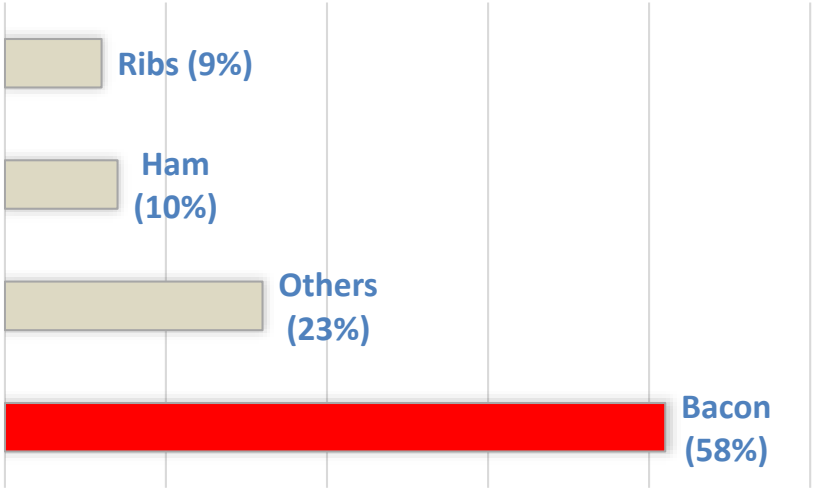
Best practices – Example (7)



Best practices – Example (8)

Pig meat preferences



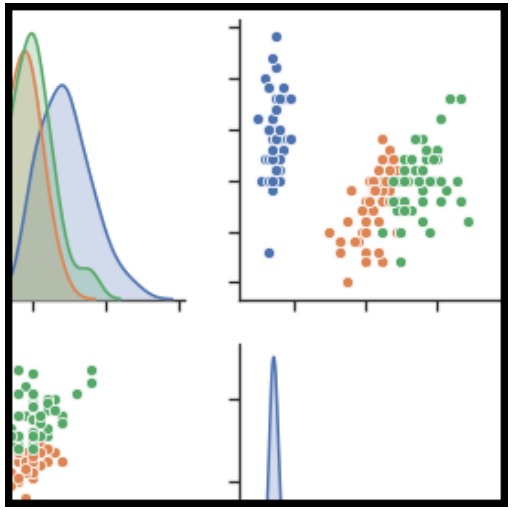


Pie vs. bar and charts

- Pie charts are less crunchy
- Bar and Charts are more crunchy and effective

Python libraries for visualization

- Seaborn
  - A Python data visualization library based on Matplotlib
  - It provides a high-level interface for drawing attractive and informative statistical graphics
  - https://seaborn.pydata.org
- Matplotlib
  - One of the most widely used, visualization library in Python
  - http://aosabook.org/en/matplotlib.html



Matplotlib architecture

Scripting layer
(Pyplot)

Automates the process of defining a canvas and defining a figure artist instance and connecting them

Artist layer
(Artist)

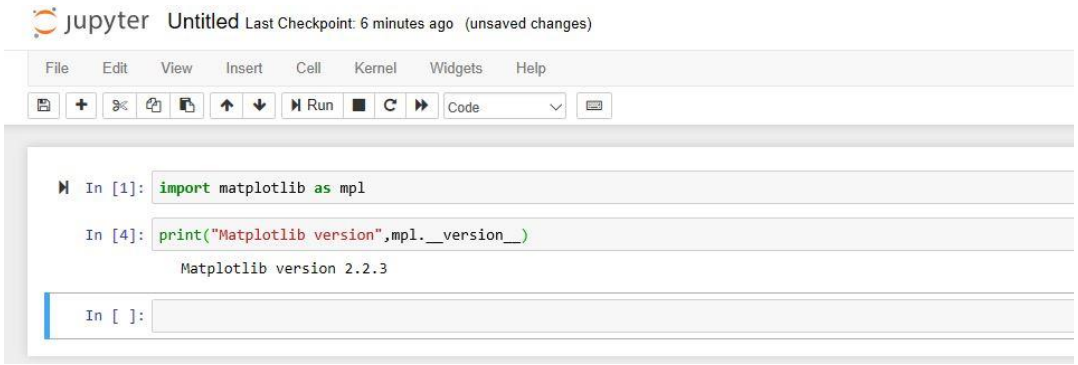
Knows how to use renderer to draw on canvas

Backend layer
(FigureCanvas, Render, Event)

Defines a canvas and knows how to draw figure on canvas

Jupyter notebook

- Open source web application that allows live code visualizations
- Numerical simulation, statistical modelling, data visualization, machine learning
- https://jupyter.org
- Jupyter has some specialized support for Matplotlib

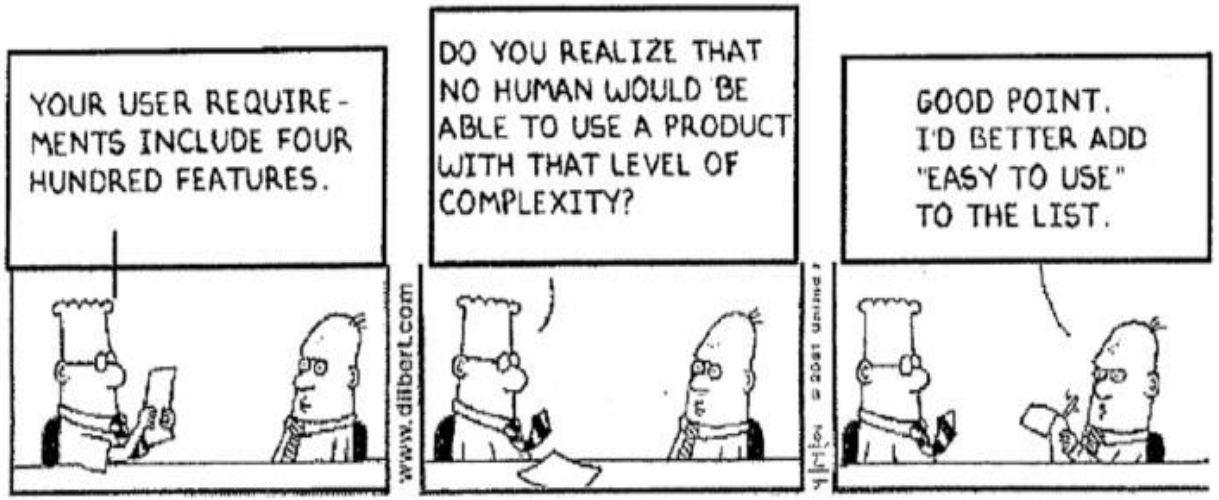


UI DESIGN PROCESS

Design process of UI

- User Analysis
  - Understand what users will do with the system
- Prototyping
  - Develop (many) prototypes
- Evaluation
  - Experiment with the prototypes

User Analysis – Humor



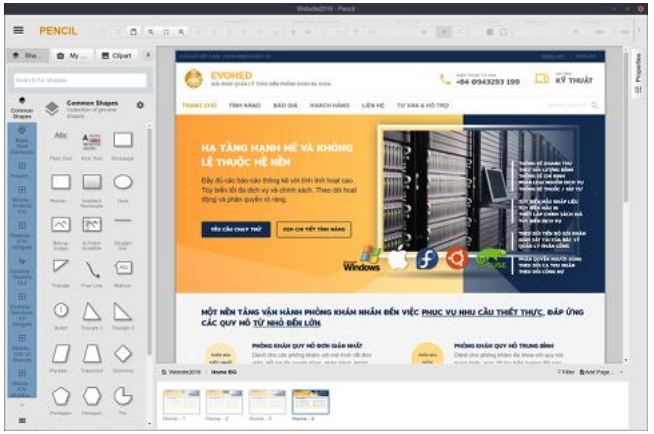
User Analysis

- Ask questions, show examples, explain what can (more importantly cannot) be done, . . .
  - Requirements solicitation
  - Ethnography (Observe the user at work)

- **Tangible** information (feedback) from user:
  - I want to control my computing device using voice commands
  - I want to control my computing device using voice commands and it must only follow my voice commands
  - I want to mind-control my computing device

Prototyping

- Provide users a direct experience with the interface
- Helps in getting users' judgment
- Simple prototypes
  - Paper + pencil
    - Story-boards, scenarios, use-cases, etc
  - Digital with dummy buttons
    - e.g., Pencil Project http://pencil.evolus.vn
      - free and open-source GUI prototyping tool
  - Digital with some functionality
    - e.g., scripting, visual language, etc



Evaluation – Usability

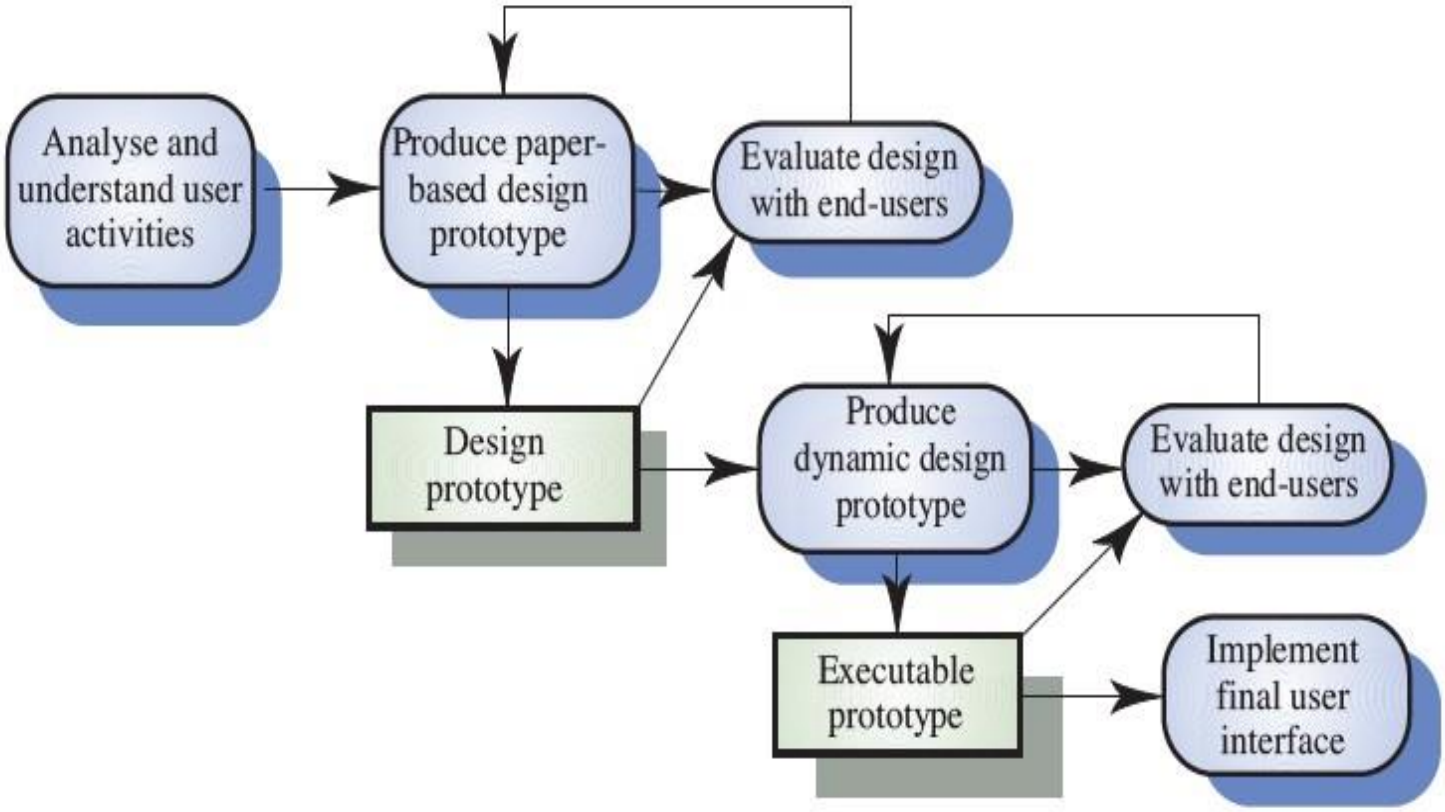
- Conformance to domain-specific vocabulary
- Recognition of options
- Consistency
- Visibility of system status
- Error prevention
- Error information
- Easy recovery methods
- Precise and concise information
- Help and manuals
- Flexibility for experts

Evaluation – Usability

Designing the User Interface (Ben Shneiderman criteria):

1. Strive for consistency.
2. Give shortcuts to the user.
3. Offer informative feedback.
4. Make each interaction with the user yield a result.
5. Offer simple error handling.
6. Permit easy undo of actions.
7. Let the user be in control.
8. Reduce short-term memory load on the user.

Typical UI Design – UI Development cycle



User interface principles – Recap

- Keep the users in mind
- Get feedback often!
- Prepare multiple (progressively advanced) prototypes

Construction of UI

- UI allows users interact with the data
  - Manage
  - View
  - Modify
- Location of data & type of interaction
  - Stand-alone applications: data hosted on the client
  - Client-server applications
    - Data hosted on the server, user-interface and computations on the client
    - Data hosted on the server, computations on the server, user-interface on the client
    - Data hosted on the server, computations partitioned between client and server (data requested when needed or pre-fetched), user-interface on the client.

Software development process (1)

- System analysis & requirements elicitation
  - Analysis: What are the functional and non-functional requirements of the desired system?
    - Domain model: Relationship of the software with the real-world
    - Application model: Description of application functionality
- System design: High-level architecture of the application
  - Relationships: UML and modularization
  - Class diagrams: Organize the data (information hiding, interface specification)
  - Interface design (interactions between software modules)
- Implementation & Testing
- Deployment

Software development process (2)

- Design considers domain & data, modularization and interface
- Distinction between data, components and interface during design
  - Data design ➔ data structures
  - Component (package) design ➔ separation of functionalities
    - Decomposability
    - Composability
    - Understandability (Individuality)
    - Continuity (Extensibility)
    - Protection (Security)
  - Interface design ➔ reduction of communication complexity
    - Example of interfaces in Java: Runnable – The Runnable interface should be implemented by any class whose instances are intended to be executed by a thread.

Summary

- User interfaces
- Data visualization
- UI Design process
- User interfaces construction

Literature – User Interfaces

- https://www.interaction-design.org/literature/topics/ui-design
- https://blog.teamtreehouse.com/10-user-interface-design-fundamentals
- http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html

- Designing the User Interface (6th Edition) by Ben Shneiderman and Catherine Plaisant