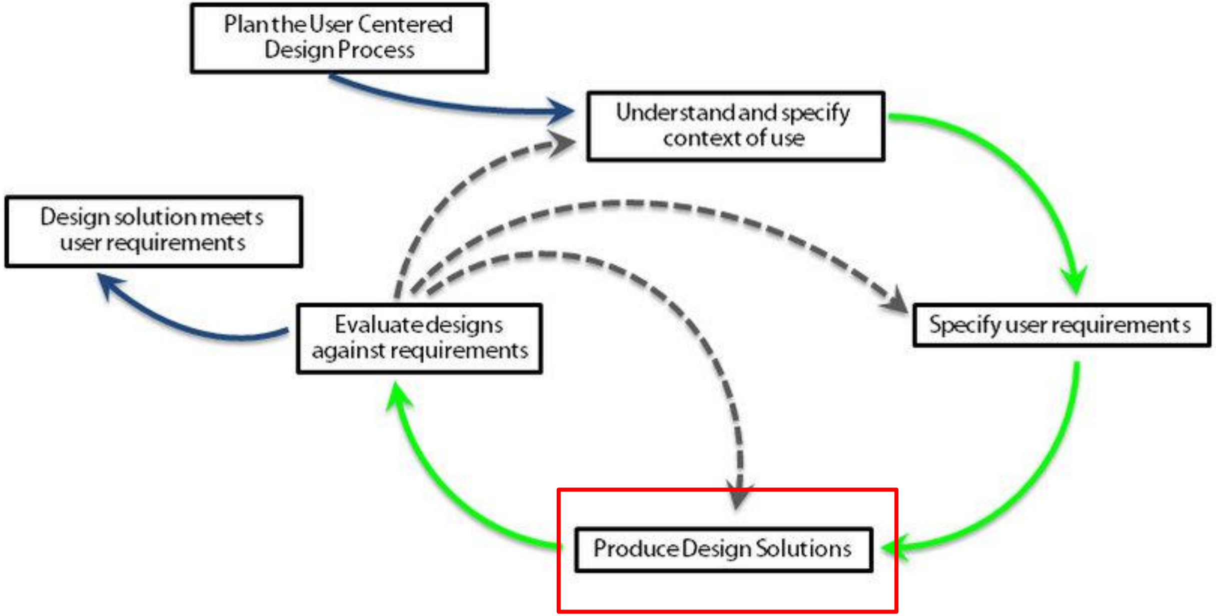
SEng 310
Lecture 11 - June 12th, 2023

Slide credits: Prof. Sowmya Somanath and Prof. Charles Perin

RECAP - HOW TO PROTOTYPE

HUMAN-CENTERED DESIGN PROCESS


Plan the User Centered Design Process
Understand and specify context of use
Design solution meets user requirements
Evaluate designs against requirements
Specify user requirements
Produce Design Solutions

https://www.researchgate.net/publication/283963125/figure/fig1/AS:671513787248651@1537112771357/The-User-Centered-Design-Process-according-to-DIN-EN-ISO-9241-210-8.jpg

HOW TO PROTOTYPE

PROTOTYPING TECHNIQUES

Storyboard

PICTIVE

Wizard-of-Oz

Video Prototyping

Role-Playing

Prototyping Toolkits

PROTOTYPING TECHNIQUES

Storyboard

PICTIVE

Wizard-of-Oz

Video Prototyping

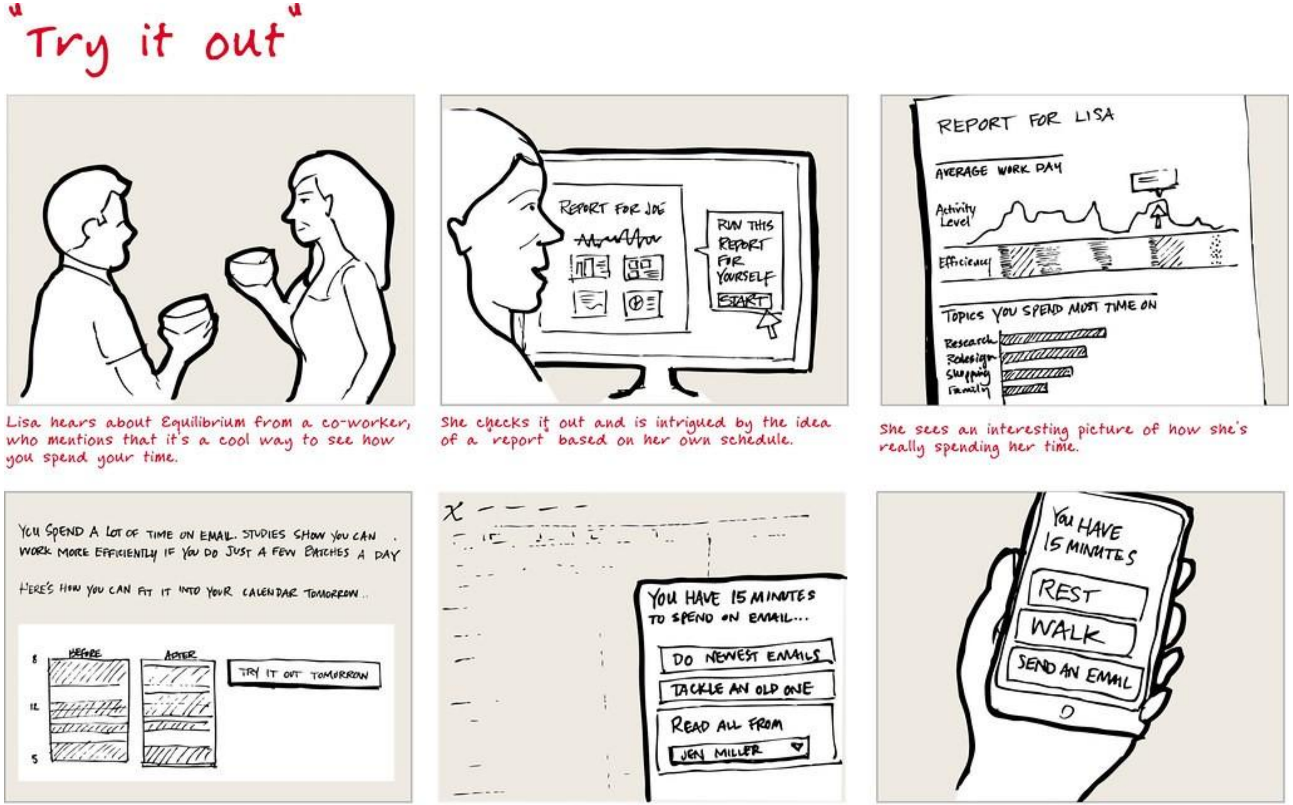
Role-Playing

Prototyping Toolkits

STORYBOARDS

Storyboard uses a sequence of images to tell the story of how character(s) interacts with an interface.

The concept comes from films.


"Try it out"
REPORT FOR JOE
RUN THIS REPORT FOR YOURSELF
START
REPORT FOR LISA
AVERAGE WORK DAY
Activity Level
Efficiency
TOPICS YOU SPEND MOST TIME ON
Research
Redesign
Shopping
Family
YOU SPEND A LOT OF TIME ON EMAIL. STUDIES SHOW YOU CAN WORK MORE EFFICIENTLY IF YOU DO JUST A FEW BATCHES A DAY
HERE'S HOW YOU CAN FIT IT INTO YOUR CALENDAR TOMORROW...
BEFORE
AFTER
8
12
5
TRY IT OUT TOMORROW
YOU HAVE 15 MINUTES TO SPEND ON EMAIL...
DO NEWEST EMAILS
TACKLE AN OLD ONE
READ ALL FROM
JEN MILLER
YOU HAVE 15 MINUTES
REST
WALK
SEND AN EMAIL

Lisa hears about Equilibrium from a co-worker, who mentions that it's a cool way to see how you spend your time.

She checks it out and is intrigued by the idea of a "report" based on her own schedule.

She sees an interesting picture of how she's really spending her time.

She sees that she can get simple suggestions based on her real calendar, and that she can easily "try out" Equilibrium's features.

The next day, she gets interesting and timely reminders.

She signs up to receive other reminders for good-for-her things throughout the day.

https://live.staticflickr.com/7454/9200995303_5053c78890_b.jpg

PROTOTYPING TECHNIQUES

Storyboard

PICTIVE

Wizard-of-Oz

Video Prototyping

Role-Playing

Prototyping Toolkits

PICTIVE

Plastic **I**nterface for **C**ollaborative **T**echnology **I**nitiatives through **V**ideo **E**xploration

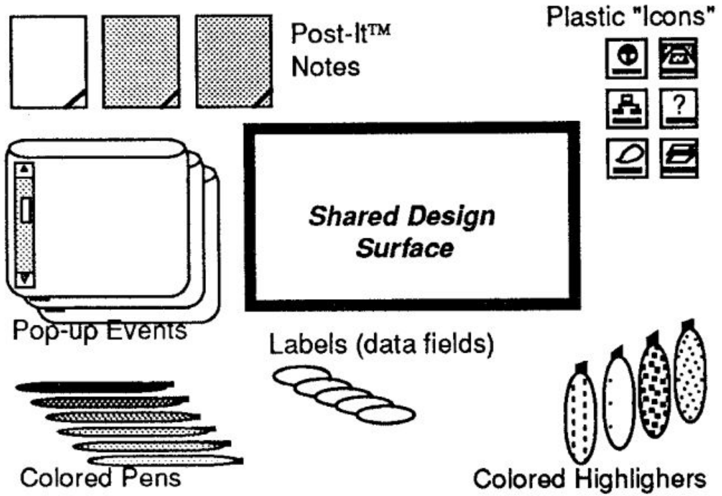


Figure 1. PICTIVE design objects.

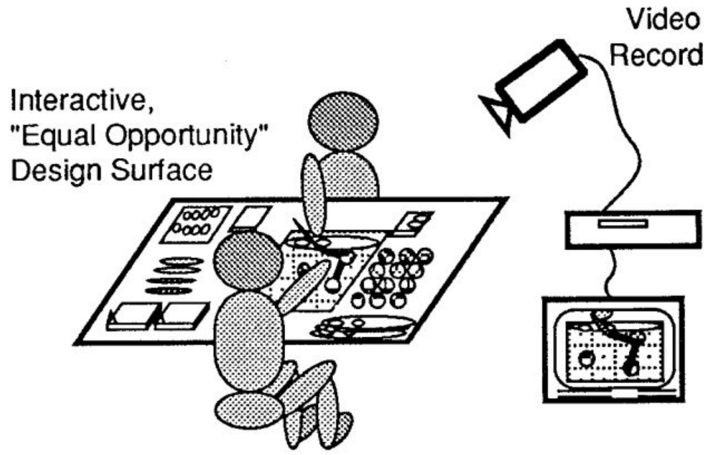


Figure 2. PICTIVE setting.

Muller, Michael J. "PICTIVE—an exploration in participatory design." *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1991.

PROTOTYPING TECHNIQUES

Storyboard

PICTIVE

Wizard-of-Oz

Video Prototyping

Role-Playing

Prototyping Toolkits

WIZARD OF OZ

Common problem: it’s difficult to prototype some piece of functionality

Need: test whether it is actually good

Solution: fake it with a person controlling the interface! Make the interaction as authentic as possible
Key: user has no idea that the interaction is being

faked Origin: Wizard of Oz book. See:

https://www.youtube.com/watch?time_continue=107&v=NZR64EF3OpA&feature=emb_logo

PROTOTYPING TECHNIQUES

Storyboard

PICTIVE

Wizard-of-Oz

Video Prototyping

Role-Playing

Prototyping Toolkits

VIDEO PROTOTYPING

- Video prototyping allows you to prototype functionality without needing to be physically present.

- Unlike wizard of oz prototypes which require you to be around to simulate functionality, a video prototype frees you from that, but constrains you to a limited scenario.

- You can also simulate this through a stitched together set of images (stop motion).

Mackay, Wendy E. "Video Prototyping: a technique for developing hypermedia systems." *CHI'88 Conference Companion Human Factors in Computing Systems*. Vol. 5. 1988.

PROTOTYPING TECHNIQUES

Storyboard

PICTIVE

Wizard-of-Oz

Video Prototyping

Role-Playing

Prototyping Toolkits

ROLE-PLAYING

- It can help people learn more about how an interaction can play out and reveal any unconscious actions
- Researchers can learn about new interaction techniques that emerge in that moment; “natural” interactions
- You can include props and costumes but it’s not always necessary

PROTOTYPING TECHNIQUES

Storyboard
PICTIVE
Wizard-of-Oz
Video Prototyping
Role-Playing

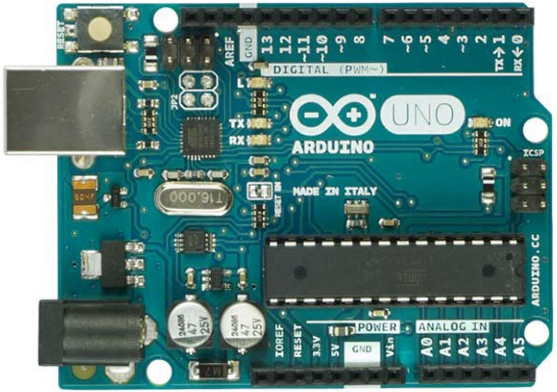
LOW-FIDELITY PROTOTYPES

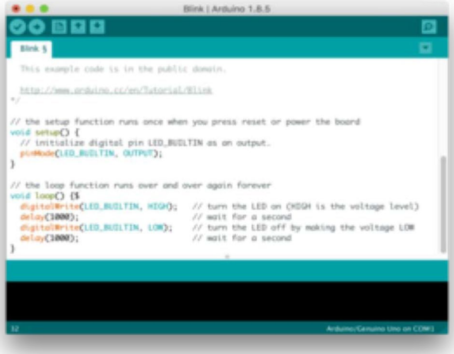
Prototyping Toolkits

PROTOTYPING TECHNIQUES

Storyboard PICTIVE Wizard-of-Oz Video Prototyping Role-Playing	**LOW-FIDELITY PROTOTYPES**
Prototyping Toolkits	**MEDIUM / HIGH-FIDELITY PROTOTYPES**

HIGHER FIDELITY PROTOTYPING



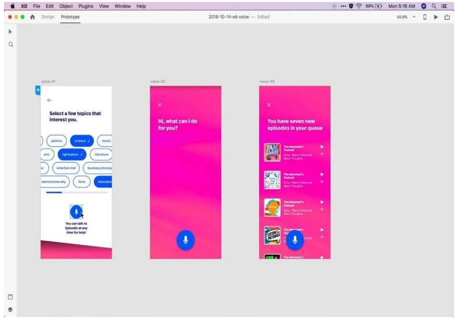


```
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```



Select a few topics that interest you.

You can talk to Episodic at any time for help!

Hi, what can I do for you?

You have seven new episodes in your queue

PROTOTYPING TOOLS AND TOOLKITS

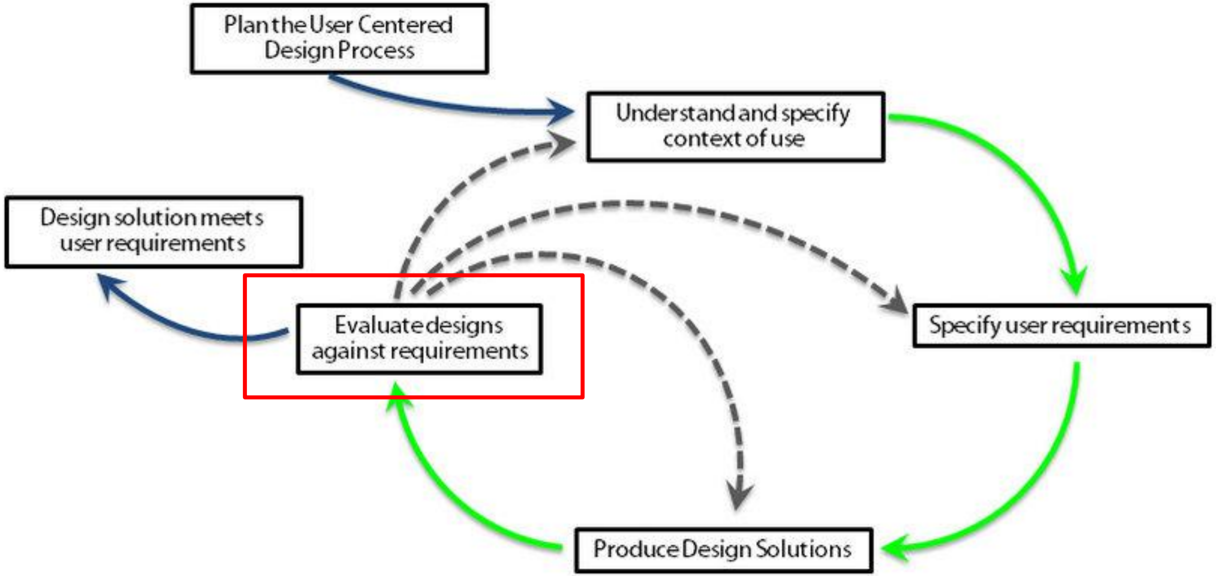
Toolkits are: “generative platforms designed to create new interactive artifacts, provide easy access to complex algorithms, enable fast prototyping of software and hardware interfaces, and/or enable creative exploration of design spaces.”

Ledo, David, et al. "Evaluation strategies for HCI toolkit research." *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018.

BENEFITS

- Toolkits make it easier for users to author new interactive systems by encapsulating concepts to simplify expertise
- Toolkits define rules or pathways for users to create new solutions, leading them to right solutions and away from wrong ones
- Given that toolkits reduce the effort to build new interactive solutions, they can enable new audiences to author these solutions
- Toolkits can align their ideas to existing infrastructure and standards, enabling power in combination
- Toolkits allow for replication of ideas that explore a concept

HUMAN-CENTERED DESIGN PROCESS


Plan the User Centered Design Process
Understand and specify context of use
Design solution meets user requirements
Evaluate designs against requirements
Specify user requirements
Produce Design Solutions

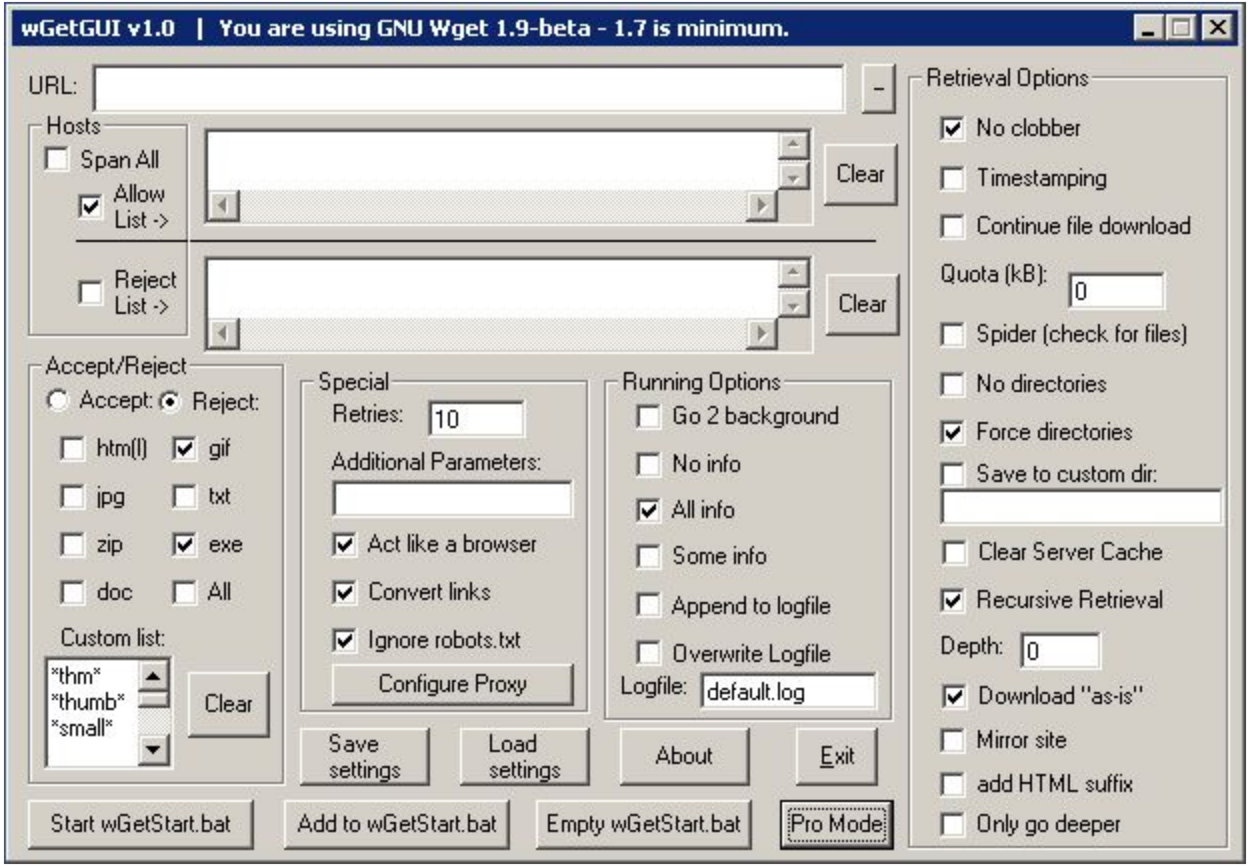
https://www.researchgate.net/publication/283963125/figure/fig1/AS:671513787248651@1537112771357/The-User-Centered-Design-Process-according-to-DIN-EN-ISO-9241-210-8.jpg

wGet

- GNU Wget is a free software package for retrieving files
- Non-interactive command-line tool
- GNU Wget has many features:
  - Resume downloads, using REST and RANGE
  - Use filename wildcards and recursively mirror directories
  - NLS-based message files for many different languages

…and more

wGetGUI v1.0



wGetGUI v1.0 | You are using GNU Wget 1.9-beta - 1.7 is minimum.

URL:

Hosts
- □ Span All
- ☑ Allow List ->
- □ Reject List ->

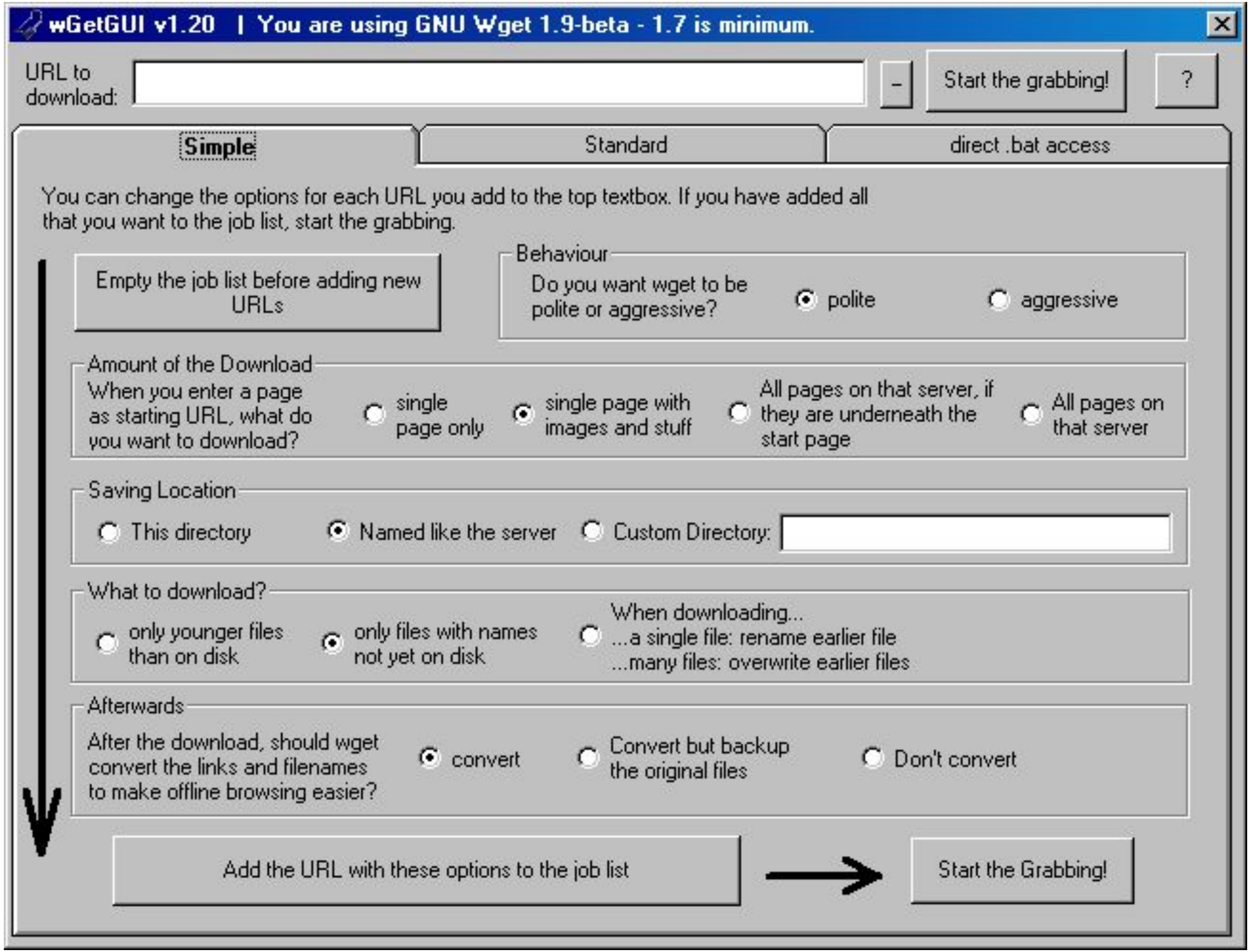
Accept/Reject
- ○ Accept: ● Reject:
- □ htm(l) ☑ gif
- □ jpg □ txt
- □ zip ☑ exe
- □ doc □ All

Custom list:
thm
thumb
small

Special
- Retries: 10
- Additional Parameters:
- ☑ Act like a browser
- ☑ Convert links
- ☑ Ignore robots.txt

Running Options
- □ Go 2 background
- □ No info
- ☑ All info
- □ Some info
- □ Append to logfile
- □ Overwrite Logfile
- Logfile: default.log

Retrieval Options
- ☑ No clobber
- □ Timestamping
- □ Continue file download
- Quota (kB): 0
- □ Spider (check for files)
- □ No directories
- ☑ Force directories
- □ Save to custom dir:
- □ Clear Server Cache
- ☑ Recursive Retrieval
- Depth: 0
- ☑ Download "as-is"
- □ Mirror site
- □ add HTML suffix
- □ Only go deeper



wGetGUI v1.20 | You are using GNU Wget 1.9-beta - 1.7 is minimum.

You can change the options for each URL you add to the top textbox. If you have added all that you want to the job list, start the grabbing.

Behaviour

Do you want wget to be polite or aggressive? ● polite ○ aggressive

Amount of the Download

When you enter a page as starting URL, what do you want to download? ○ single page only ● single page with images and stuff ○ All pages on that server, if they are underneath the start page ○ All pages on that server

Saving Location

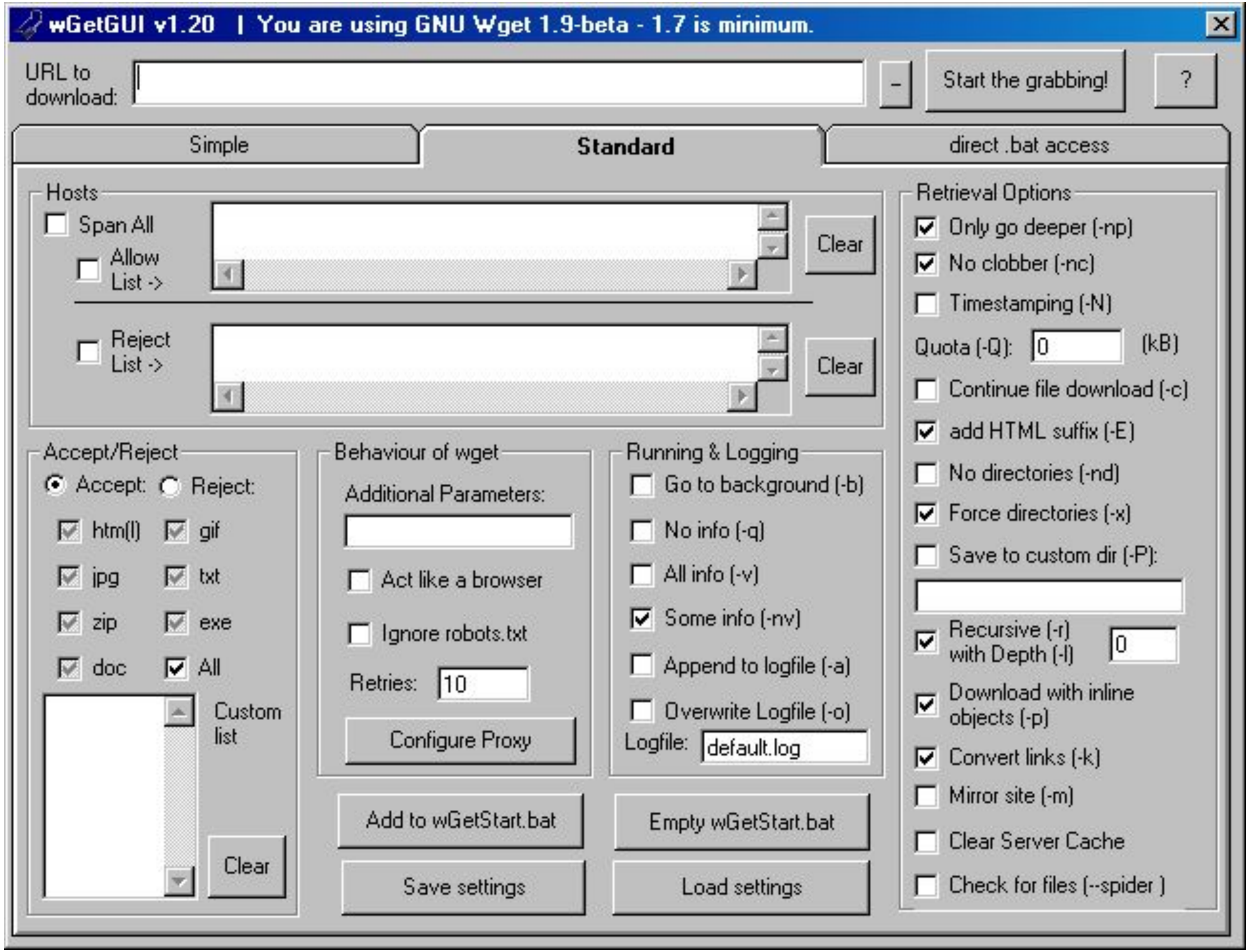
○ This directory ● Named like the server ○ Custom Directory:

What to download?

○ only younger files than on disk ● only files with names not yet on disk ○ When downloading... ...a single file: rename earlier file ...many files: overwrite earlier files

Afterwards

After the download, should wget convert the links and filenames to make offline browsing easier? ● convert ○ Convert but backup the original files ○ Don't convert



wGetGUI v1.20 | You are using GNU Wget 1.9-beta - 1.7 is minimum.

URL to download:

Hosts

☐ Span All

☐ Allow List ->

☐ Reject List ->

Accept/Reject

● Accept: ○ Reject:

☑ htm(l) ☑ gif

☑ jpg ☑ txt

☑ zip ☑ exe

☑ doc ☑ All

Custom list

Behaviour of wget

Additional Parameters:

☐ Act like a browser

☐ Ignore robots.txt

Retries: 10

Running & Logging

☐ Go to background (-b)

☐ No info (-q)

☐ All info (-v)

☑ Some info (-nv)

☐ Append to logfile (-a)

☐ Overwrite Logfile (-o)

Logfile: default.log

Retrieval Options

☑ Only go deeper (-np)

☑ No clobber (-nc)

☐ Timestamping (-N)

Quota (-Q): 0 (kB)

☐ Continue file download (-c)

☑ add HTML suffix (-E)

☐ No directories (-nd)

☑ Force directories (-x)

☐ Save to custom dir (-P):

☑ Recursive (-r) with Depth (-l) 0

☑ Download with inline objects (-p)

☑ Convert links (-k)

☐ Mirror site (-m)

☐ Clear Server Cache

☐ Check for files (--spider)

SO…

We probably don’t need to run a usability study on this design, because intuitively, we see that there are “things that are wrong” with the current design.

How can we formalize this idea of “using our intuition” so that it is more systematic, and less haphazard?

HEURISTIC EVALUATION

Systematic inspection of an interface design to see if an interface complies with a set of usability heuristics, or usability guidelines.

Generally:

- 3-5 inspectors (usability engineers, end users, experts…)
- inspect interface in isolation (~1-2 hr for simple interfaces)
- results are aggregated afterwards
  - single evaluator catches ~35% usability problems
  - 5 evaluators catch ~75%

HEURISTIC

A heuristic is a rule of thumb—a principle for solving a problem or making decisions

- Never chase after a bus, another one is coming…
- Stuck in traffic: car in the lane next to me passed me □ that lane must be moving faster

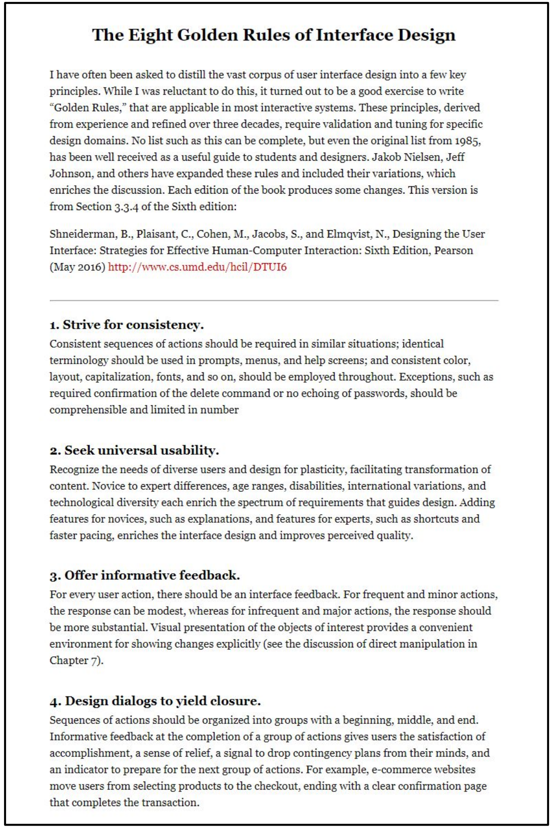
Not always right/true, but cognitive shortcuts

DESIGN HEURISTICS

Broad usability statements that can guide a developer’s design efforts

Derived from common design problems across many systems and several researchers and practitioners have developed different sets of heuristics (e.g. domain specific)

EXAMPLES OF DESIGN HEURISTICS



The Eight Golden Rules of Interface Design

I have often been asked to distill the vast corpus of user interface design into a few key principles. While I was reluctant to do this, it turned out to be a good exercise to write "Golden Rules," that are applicable in most interactive systems. These principles, derived from experience and refined over three decades, require validation and tuning for specific design domains. No list such as this can be complete, but even the original list from 1985, has been well received as a useful guide to students and designers. Jakob Nielsen, Jeff Johnson, and others have expanded these rules and included their variations, which enriches the discussion. Each edition of the book produces some changes. This version is from Section 3.3.4 of the Sixth edition:

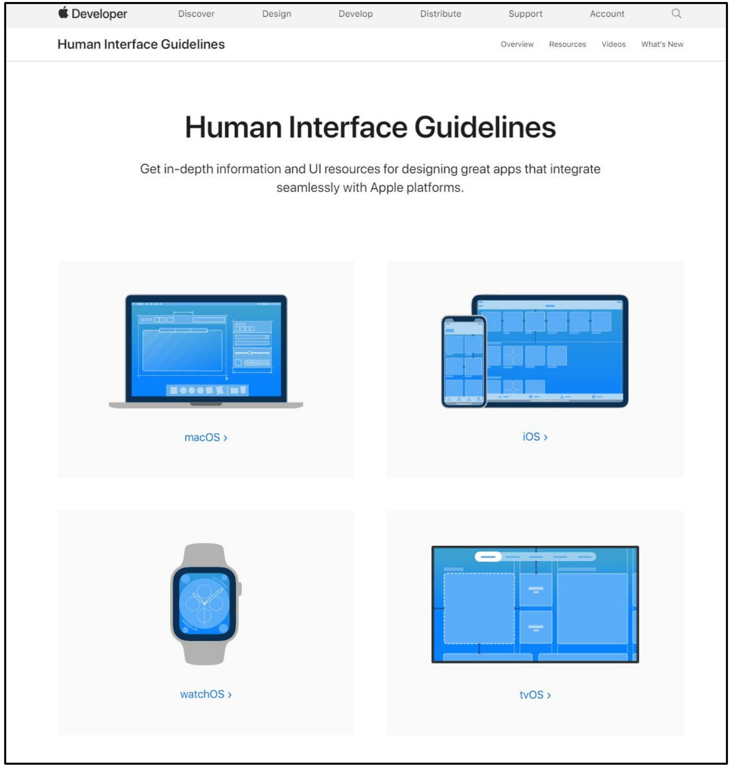
Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., and Elmqvist, N., Designing the User Interface: Strategies for Effective Human-Computer Interaction: Sixth Edition, Pearson (May 2016) http://www.cs.umd.edu/hcil/DTUI6

1. Strive for consistency.
Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent color, layout, capitalization, fonts, and so on, should be employed throughout. Exceptions, such as required confirmation of the delete command or no echoing of passwords, should be comprehensible and limited in number

2. Seek universal usability.
Recognize the needs of diverse users and design for plasticity, facilitating transformation of content. Novice to expert differences, age ranges, disabilities, international variations, and technological diversity each enrich the spectrum of requirements that guides design. Adding features for novices, such as explanations, and features for experts, such as shortcuts and faster pacing, enriches the interface design and improves perceived quality.

3. Offer informative feedback.
For every user action, there should be an interface feedback. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial. Visual presentation of the objects of interest provides a convenient environment for showing changes explicitly (see the discussion of direct manipulation in Chapter 7).

4. Design dialogs to yield closure.
Sequences of actions should be organized into groups with a beginning, middle, and end. Informative feedback at the completion of a group of actions gives users the satisfaction of accomplishment, a sense of relief, a signal to drop contingency plans from their minds, and an indicator to prepare for the next group of actions. For example, e-commerce websites move users from selecting products to the checkout, ending with a clear confirmation page that completes the transaction.



Human Interface Guidelines

Get in-depth information and UI resources for designing great apps that integrate seamlessly with Apple platforms.

CONDUCTING A HEURISTIC EVALUATION

- Pre-evaluation training
  - Provide evaluators with domain knowledge and information on scenario
- Evaluation
  - Individuals evaluate **and then** aggregate results
- Severity ratings
  - Determine how severe each problem is (priority)
  - Perform individually **and then** as a group
- Debriefing
  - Discuss outcome with design team

https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

EVALUATION

Each evaluator performs at least two passes

- First: get a feel for flow and scope of system
- Second: focus on specific elements

Assistance

For walk-up and use interfaces, no need; otherwise, supply evaluators with scenarios

Each evaluator produces a list of problems

- Explain why with respect to the heuristics or other information
- Be specific and list each problem separately

EVALUATION

Why list each violation?

Where problems may be found

- Single location in UI
- Two or more locations that need to be compared
- Overall structure of UI
- Something that is missing
  - Hard w/ paper prototypes

SEVERITY RATINGS

0 – don’t think this is a usability problem

1 – cosmetic problem

2 – minor usability problem

3 – major usability problem; important to fix

4 – usability catastrophe; must fix

SEVERITY RATING

Used to allocate resources to fix problems

Combination of:

- Frequency
- Impact
- Persistence (one time or repeating)

Should be estimated after all problems have been seen

Independently first is good

HEURISTIC EVALUATION: ADVANTAGES

Minimalist approach

- Few guidelines identify many common usability problems

“Discount” usability engineering

- End users not required
- Cheap and fast
- Can be done by usability experts as well as end users
- provides common evaluation template (to evaluate across systems)

HEURISTIC EVALUATION: DISADVANTAGES

Principles are general

- Not a simple checklist
- You may actually have the wrong design altogether

NIELSEN’S 10 HEURISTICS

1. VISIBILITY OF SYSTEM STATUS
2. MATCH BETWEEN SYSTEM AND REAL-WORLD
3. USER CONTROL AND FREEDOM
4. CONSISTENCY AND STANDARDS
5. HELP USERS RECOGNIZE, DIAGNOSE, AND RECOVER FROM ERRORS
6. ERROR PREVENTION
7. RECOGNITION RATHER THAN RECALL
8. AESTHETIC AND MINIMALIST DESIGN
9. FLEXIBILITY AND EFFICIENCY OF USE
10. HELP AND DOCUMENTATION